

# AWS Certified Solution Architect Associate



# Getting Started

In 28  
Minutes



Amazon S3



EC2



Amazon EBS



ELB



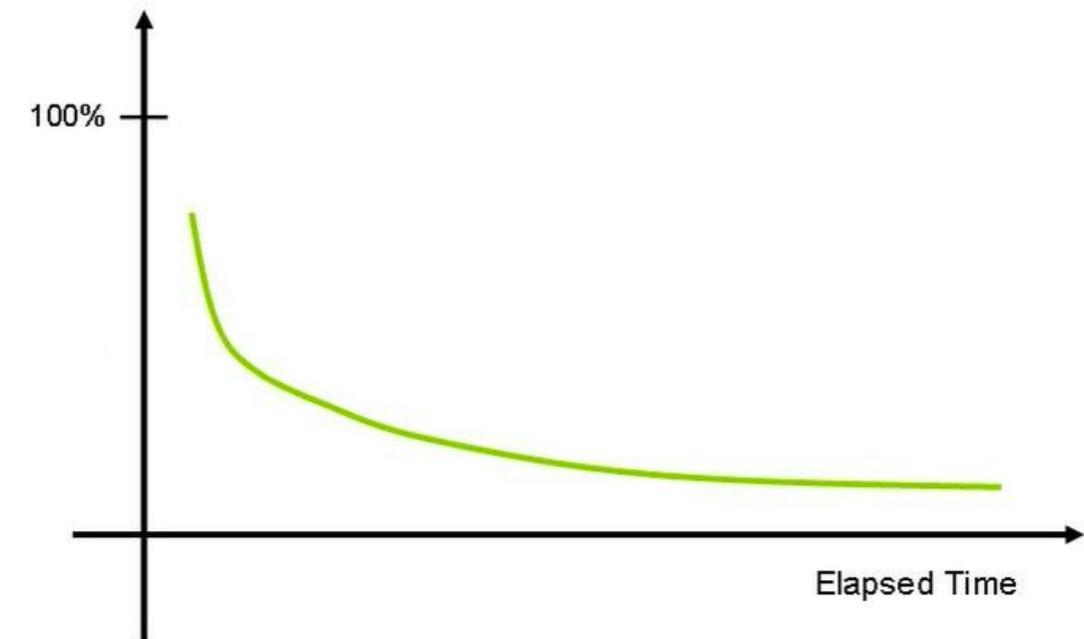
ECS

- AWS has 200+ services. Exam expects knowledge of 40+ services.
- Exam **tests your decision making abilities:**
  - Which service do you choose in which situation?
  - How do you **trade-off** between resilience, performance and cost while not compromising on security and operational excellence?
- This course is **designed** to help you *make these tough choices*
- **Our Goal :** Enable you to architect amazing solutions in AWS

# How do you put your best foot forward?

In 28  
Minutes

- Challenging certification - Expects you to understand and **REMEMBER** a number of services
- As time passes, humans forget things.
- How do you improve your chances of remembering things?
  - Active learning - think and make notes
  - Review the presentation every once in a while



# Our Approach

In 28  
Minutes

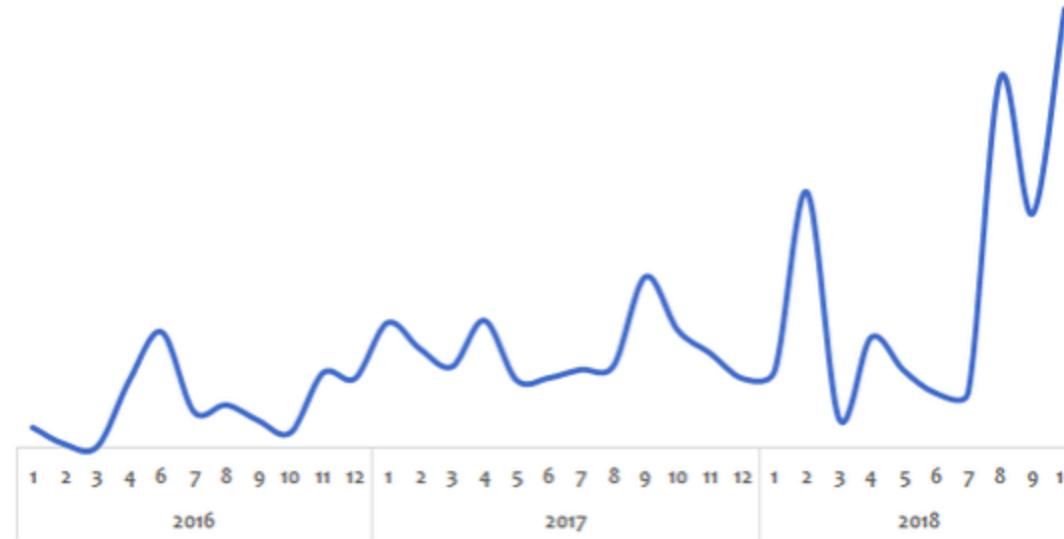
- Videos with presentations and demos
- Two kinds of quizzes to reinforce concepts:
  - Text quizzes : Traditional text quiz
  - Video quizzes : Discuss the question and the solution in a video
- Practice test at the end of the course
- (Recommended) Take your time. Do not hesitate to replay videos!
- (Recommended) Have Fun!



# AWS - Getting started

# Before the Cloud

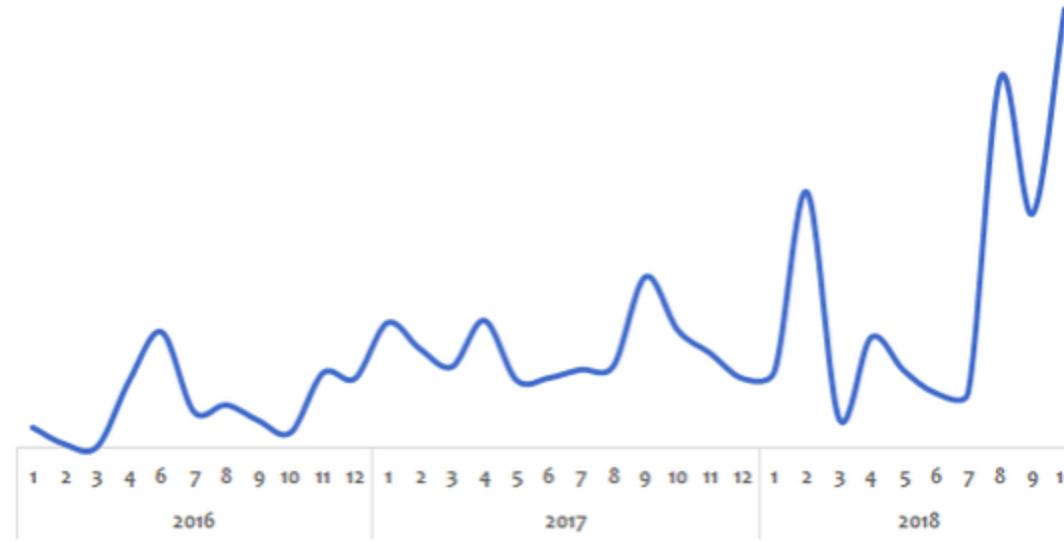
In 28  
Minutes



- Consider a Online Shopping Application:
  - Peak usage during holidays and weekends. Less load during rest of the time.
- A startup suddenly becomes popular:
  - How does it handle the sudden increase in load?
- Enterprises **procured (bought)** infrastructure **for peak load**
  - Startups procured infrastructure assuming they would be successful

# Before the Cloud - Challenges

In 28  
Minutes



- Low infrastructure utilization
- Needs ahead of time planning (**Can you guess the future?**)
- High costs of procuring infrastructure
- Dedicated infrastructure maintenance team (**Can a startup afford it?**)

# Silver Lining in the Cloud

In 28  
Minutes

- How about provisioning (renting) resources when you want them and releasing them back when you do not need them?
  - On-demand resource provisioning
- Advantages
  - Lower costs (Pay per use)
  - No upfront planning needed
  - Avoid "undifferentiated heavy lifting"
- Challenge
  - Building cloud enabled applications



# Amazon Web Services (AWS)

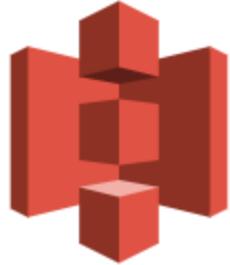
In 28  
Minutes

- Leading cloud service provider
- Provides most (200+) services
- Reliable, secure and cost-effective
- The entire course is all about AWS. You will learn it as we go further.



# Best path to learn AWS!

In 28  
Minutes



Amazon S3



EC2



Amazon EBS



ELB



ECS

- Cloud applications make use of multiple AWS services.
- There is **no single path** to learn these services independently.
- **HOWEVER**, we've worked out a simple path!

# Setting up AWS Account

In 28  
Minutes

- Create AWS Account
- Setup an IAM user

# Regions and Zones

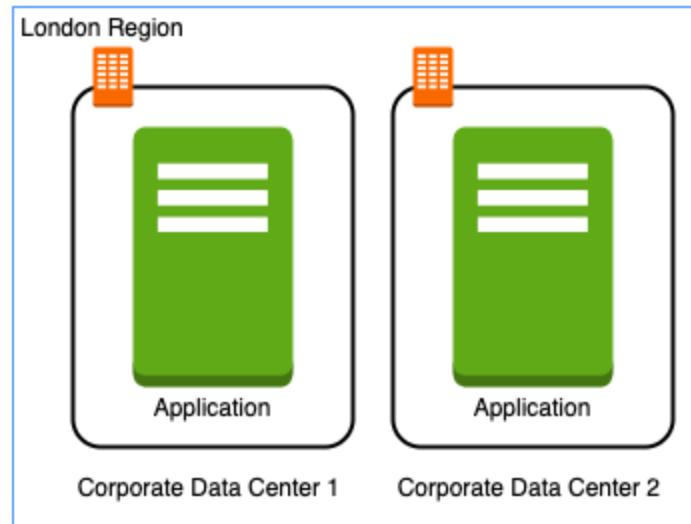
# Regions and Zones

In 28  
Minutes



- Imagine that your application is deployed in a data center in London
- What would be the challenges?
  - Challenge 1 : Slow access for users from other parts of the world (**high latency**)
  - Challenge 2 : What if the data center crashes?
    - Your application goes down (**low availability**)

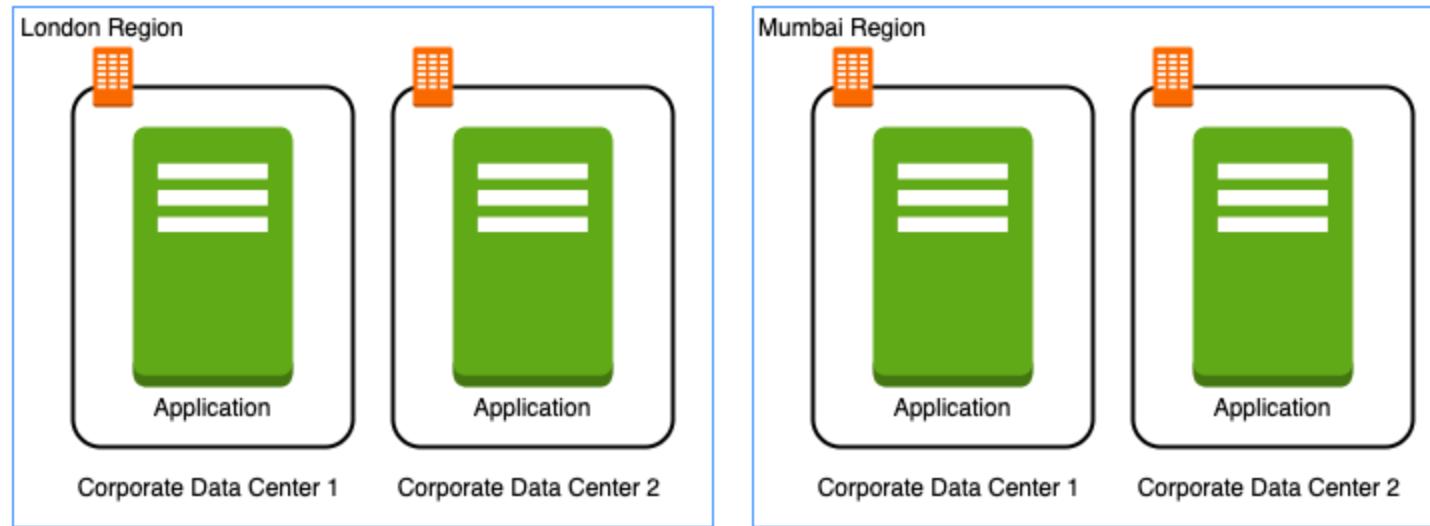
# Multiple data centers



- Let's add in one more data center in London
- What would be the challenges?
  - Challenge 1 : Slow access for users from other parts of the world
  - Challenge 2 (**SOLVED**) : What if one data center crashes?
    - Your application is **still available** from the other data center
  - Challenge 3 : What if **entire region** of London is unavailable?
    - Your application goes down

# Multiple regions

In 28  
Minutes



- Let's add a new region : Mumbai
- What would be the challenges?
  - Challenge 1 (**PARTLY SOLVED**) : Slow access for users from other parts of the world
    - You can solve this by adding deployments for your applications in other regions
  - Challenge 2 (**SOLVED**) : What if one data center crashes?
    - Your application is still live from the other data centers
  - Challenge 3 (**SOLVED**) : What if entire region of London is unavailable?
    - Your application is served from Mumbai

# Regions

In 28  
Minutes



- Imagine setting up your own data centers in different regions around the world
  - Would that be easy?
- (Solution) AWS provides **20+ regions** around the world (expanding every year)

# Regions - Advantages



- High Availability
- Low Latency
- Adhere to government **regulations**

# Regions

In 28  
Minutes

- Choose the right region(s) based on:
  - Where are your users located?
  - Where is your data located?
  - Regulatory and security compliance needs
- AWS Services can be:
  - Regional (OR)
  - Global



# Availability Zones

In 28  
Minutes

- Isolated locations in a Region
- Each AWS Region has at least two Availability Zones
- Increase availability of applications in the same region



# Regions and Availability Zones examples

In 28  
Minutes

*New Regions and AZs are constantly added*

Region Code	Region	Availability Zones	Availability Zones List
us-east-1	US East (N. Virginia)	6	us-east-1a us-east-1b us-east-1c us-east-1d us-east-1e us-east-1f
eu-west-2	Europe (London)	3	eu-west-2a eu-west-2b eu-west-2c
ap-south-1	Asia Pacific(Mumbai)	3	ap-south-1a ap-south-1b ap-south-1c

# EC2 Fundamentals

# EC2(Elastic Compute Cloud)

In 28  
Minutes

- In corporate data centers, applications are deployed to physical servers
- Where do you deploy applications in the cloud?
  - Rent virtual servers
  - **EC2 instances** - Virtual servers in AWS (billed by second)
  - **EC2 service** - Provision EC2 instances or virtual servers

# EC2 Features

In 28  
Minutes



EC2 Instances



ELB



Amazon EBS

- Create and manage lifecycle of EC2 instances
- Load balancing and auto scaling for multiple EC2 instances
- Attach storage (& network storage) to your EC2 instances
- Manage network connectivity for an EC2 instance
- Our Goal:
  - Setup EC2 instances as HTTP Server
  - Distribute load with Load Balancers

# EC2 Hands-on

In 28  
Minutes

- Let's create a few EC2 instances and play with them
- Let's check out the lifecycle of EC2 instances
- Let's use EC2 Instance Connect to SSH into EC2 instances

# EC2 Instance Types

In 28  
Minutes

- Optimized combination of **compute(CPU, GPU), memory, disk (storage) and networking** for specific workloads
- 270+ instances across 40+ instance types for different workloads
- **t2.micro:**
  - t - Instance Family
  - 2 - generation. Improvements with each generation.
  - **micro** - size. (*nano < micro < small < medium < large < xlarge < .....*)
- (Remember) As size increases, compute(CPU, GPU), memory and networking capabilities increase proportionately

# EC2 - Instance Metadata Service and Dynamic Data

In 28  
Minutes

## Instance Metadata Service:

- Get details about EC2 instance **from inside** an EC2 instance:
  - AMI ID, storage devices, DNS hostname, instance id, instance type, security groups, IP addresses etc
- URL: *<http://169.254.169.254/latest/meta-data/>*
- URL Paths: network, ami-id, hostname, local-hostname, local-ipv4 , public-hostname, public-ipv4, security-groups, placement/availability-zone

## Dynamic Data Service:

- Get dynamic information about EC2 instance:
- URL: *<http://169.254.169.254/latest/dynamic/>*
- Example: *<http://169.254.169.254/latest/dynamic/instance-identity/document>*

# EC2 Hands-on : Setting up a HTTP server

In 28  
Minutes

```
sudo su
yum update -y
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "Getting started with AWS" > /var/www/html/index.html
echo "Welcome to in28minutes $(whoami)" > /var/www/html/index.html
echo "Welcome to in28minutes $(hostname)" > /var/www/html/index.html
echo "Welcome to in28minutes $(hostname -i)" > /var/www/html/index.html
```

# Security Groups

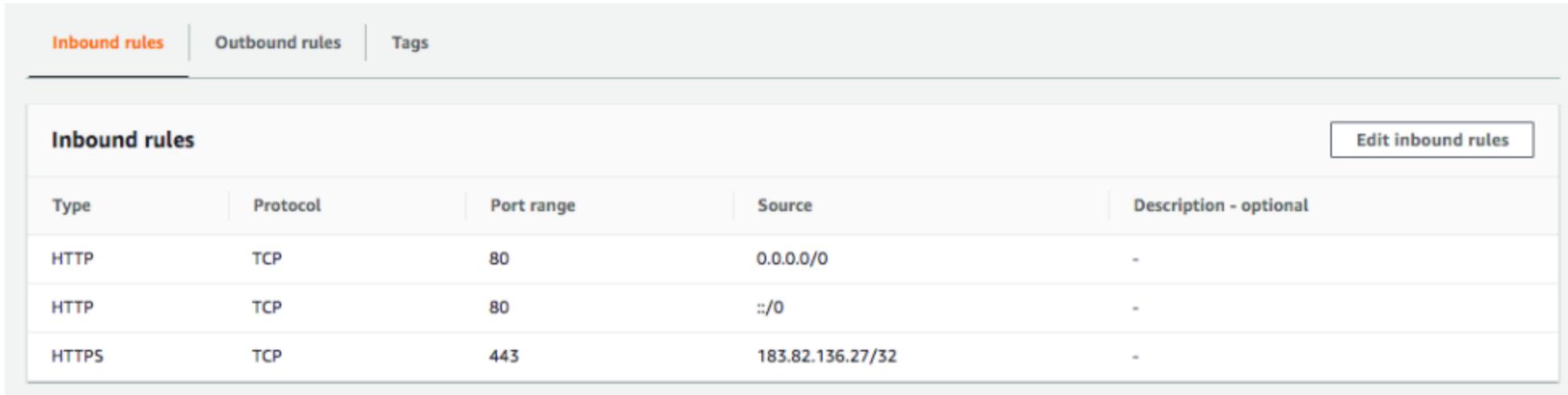
In 28  
Minutes



- **Virtual firewall** to control **incoming** and **outgoing** traffic to/from AWS resources (EC2 instances, databases etc)
- Provides additional layer of security - Defense in Depth

# Security Groups Rules

In 28  
Minutes



The screenshot shows the AWS Management Console interface for managing security group rules. The top navigation bar has tabs for 'Inbound rules' (which is selected and highlighted in orange), 'Outbound rules', and 'Tags'. Below the tabs, there's a section titled 'Inbound rules' with a 'Edit inbound rules' button. A table lists three rules:

Type	Protocol	Port range	Source	Description - optional
HTTP	TCP	80	0.0.0.0/0	-
HTTP	TCP	80	::/0	-
HTTPS	TCP	443	183.82.136.27/32	-

- Security groups are **default deny**
  - If there are no rules configured, no outbound/inbound traffic is allowed
- You can specify **allow rules ONLY**
- You can configure **separate rules** for inbound and outbound traffic
- You can assign multiple (upto five) security groups to your EC2 instances

# Security Groups

In 28  
Minutes

- You can add and delete security groups to EC2 instances at any time.
  - Changes are immediately effective
- Traffic NOT explicitly allowed by Security Group will not reach the EC2 instance
- Security Groups are **stateful**:
  - If an outgoing request is allowed, the incoming response for it is automatically allowed.
  - If an incoming request is allowed, an outgoing response for it is automatically allowed

# Security Group - Trivia

In 28  
Minutes

- What if there are no security group rules configured for inbound and outbound?
  - Default DENY. No traffic is allowed in and out of EC2 instance.
- Can I change security groups at runtime?
  - Yes. Changes are immediately effective.

# EC2 IP Addresses

In 28  
Minutes

- Public IP addresses are internet addressable.
- Private IP addresses are **internal** to a corporate network
- You CANNOT have two resources with same public IP address.
- HOWEVER, two different corporate networks CAN have resources with same private IP address
- All EC2 instances are assigned private IP addresses
- Creation of public IP addresses can be enabled for EC2 instances in public subnet
- (Remember) When you stop an EC2 instance, public IP address is lost
- **DEMO:** EC2 public and private addresses

# Elastic IP Addresses

In 28  
Minutes

- Scenario : How do you get a **constant public IP address** for a EC2 instance?
  - Quick and dirty way is to **use an Elastic IP!**
- **DEMO:** Using Elastic IP Address with an EC2 instance

# Elastic IP Addresses - Remember

In 28  
Minutes

- Elastic IP can be switched to another EC2 instance **within the same region**
- Elastic IP **remains attached** even if you stop the instance. You have to manually detach it.
- Remember : You are charged for an Elastic IP when you are NOT using it! Make sure that you explicitly release an Elastic IP when you are not using it
- You will be charged for Elastic IP when:
  - Elastic IP is NOT associated with an EC2 instance OR
  - EC2 instance associated with Elastic IP is stopped

# Simplify EC2 HTTP server setup

In 28  
Minutes

- How do we reduce the number of steps in creating an EC2 instance and setting up a HTTP Server?
- Let's explore a few options:
  - Userdata
  - Launch Template
  - AMI

# Bootstrapping with Userdata

In 28  
Minutes

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
curl -s http://169.254.169.254/latest/dynamic/instance-identity/document > /var/www/html/ir
```

- **Bootstrapping:** Install OS patches or software when an EC2 instance is launched.
- In EC2, you can configure **userdata** to bootstrap
- Lookup user data - *http://169.254.169.254/latest/user-data/*
- **DEMO** - Using Userdata

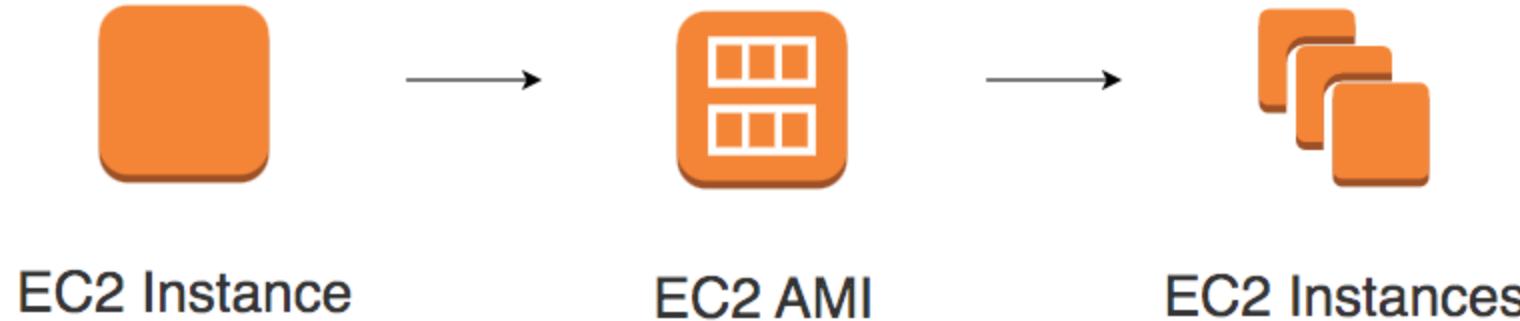
# Launch Templates

In 28  
Minutes

- Why do you need to specify all the EC2 instance details (AMI ID, instance type, and network settings) **every time** you launch an instance?
- How about creating a **Launch Template**?
- Allow you to launch Spot instances and Spot fleets as well
- **DEMO** - Launch EC2 instances using Launch Templates

# Reducing Launch Time with Customized AMI

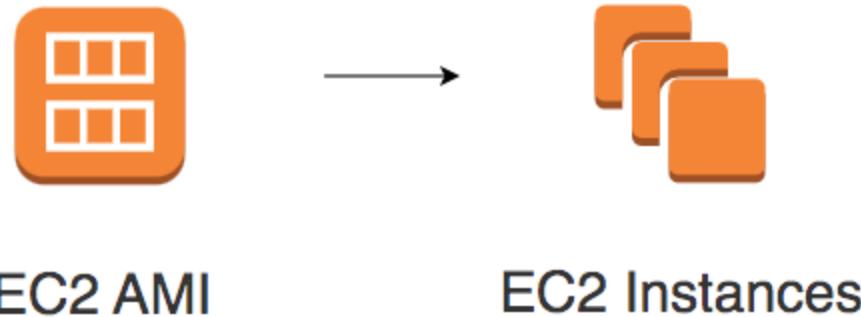
In 28  
Minutes



- Installing OS patches and software using userdata at launch of EC2 instances **increases boot up time**
- How about creating customized AMIs with OS patches and software **pre-installed**?
  - Hardening an Image - Customize EC2 images to your corporate security standards
- Prefer using Customized AMI to userdata
- **DEMO** : Create a Customized AMI and using it in Launch Template

# AMI - Amazon Machine Image

In 28  
Minutes



- What operating system and what software do you want on the instance?
- Three AMI sources:
  - Provided by AWS
  - **AWS Market Place:** Online store for customized AMIs. Per hour billing
  - **Customized AMIs:** Created by you.

# EC2 Amazon Machine Image - AMI - Remember

In 28  
Minutes

- AMIs contain:
  - Root volume block storage (OS and applications)
  - Block device mappings for non-root volumes
- You can configure launch permissions on an AMI
  - Who can use the AMI?
  - You can share your AMIs with other AWS accounts
- AMIs are stored in Amazon S3 (**region specific**).
- **Best Practice:** Backup upto date AMIs in multiple regions
  - Critical for Disaster Recovery

# EC2 Security - Key Pairs

In 28  
Minutes

- EC2 uses public key cryptography for protecting login credentials
- Key pair - public key and a private key
  - Public key is stored in EC2 instance
  - Private key is stored by customer

# Connecting to EC2 instance(s) - Troubleshooting

In 28  
Minutes

- You need to have the **private key** with you
- Change permissions to **0400** (*chmod 400 /path/my-key-pair.pem*)
  - Default permissions on private key - 0777 (**VERY OPEN**)
- (Windows Instances) In addition to private key, you need admin password
  - (At Launch) Random admin password is generated and encrypted using public key
  - Decrypt the password using the private key and use it to login via RDP
- Security Group should allow inbound SSH or RDP access:
  - Port 22 - Linux EC2 instance (SSH)
  - Port 3389 - RDP (Remote Desktop - Windows)
- Connect to your instance using its Public DNS: `ec2-**-**-**-**.compute.amazonaws.com`

# Important EC2 Scenarios - Quick Review

In 28  
Minutes

Scenario	Solution
You want to identify all instances belonging to a project, to an environment or to a specific billing type	Add Tags. Project - A. Environment - Dev
You want to change instance type	Stop the instance. Use "Change Instance Type" to change and restart.
You don't want an EC2 instance to be automatically terminated	Turn on Termination Protection. (Remember) EC2 Termination Protection is not effective for terminations from a) Auto Scaling Groups (ASG) b) Spot Instances c) OS Shutdown
You want to update the EC2 instance to a new AMI updated with latest patches	Relaunch a new instance with an updated AMI
Create EC2 instances based on on-premise Virtual Machine (VM) images	Use VM Import/Export. You are responsible for licenses.

# Important EC2 Scenarios - Quick Review

In 28  
Minutes

Scenario	Solution
Change security group on an EC2 instance	Assign at launch or runtime. Security Group changes are immediately effective.
You get a timeout while accessing an EC2 instance	Check your Security Group configuration
You are installing a lot of software using user data slowing down instance launch. How to make it faster?	Create an AMI from the EC2 instance and use it for launching new instances
I've stopped my EC2 instance. Will I be billed for it?	ZERO charge for a stopped instance (If you have storage attached, you have to pay for storage)

## AMI

- What operating system and what software do you want on the instance?
- Reduce boot time and improve security by creating customized hardened AMIs.
- Region specific.
- Backup AMIs in multiple regions.
- You can share AMIs with other AWS accounts.

## EC2 Instance Types

- Optimized combination of compute(CPU, GPU), memory, disk (storage) and networking for specific workloads.

## Security Groups

- Virtual firewall to control incoming and outgoing traffic to/from AWS resources (EC2 instances, databases etc)
- Default deny. Separate allow rules for inbound and outbound traffic
- Stateful and immediately effective

## Key Pairs

- Public key cryptography (Key Pairs) used to protect your EC2 instances
- You need private key with right permissions (chmod 400) to connect to your EC2 instance. (Windows EC2 instances only) You need admin password also.
- Security group should allow SSH(22) or RDP(3389)

# Quick Review

In 28  
Minutes

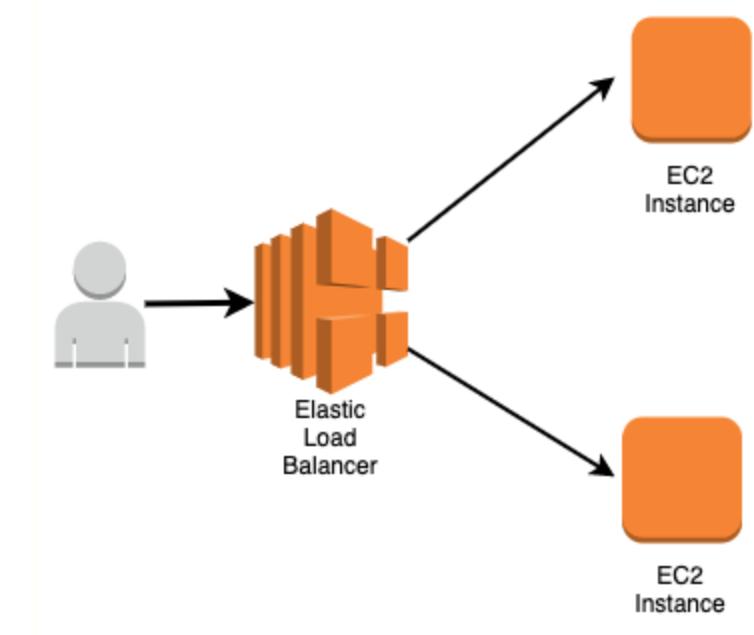
- **Instance Metadata Service** - Get details about EC2 instance from inside an EC2 instance. <http://169.254.169.254/latest/meta-data/>
- **Userdata** - Used for bootstrapping. Install OS patches or software when an EC2 instance is launched.
- **Elastic IP Addresses** - Static public IP address for EC2 instance.
- **Launch Templates** - Pre-configured templates (AMI ID, instance type, and network settings) simplifying the creation of EC2 instances.

# Load Balancing

# Elastic Load Balancer

In 28  
Minutes

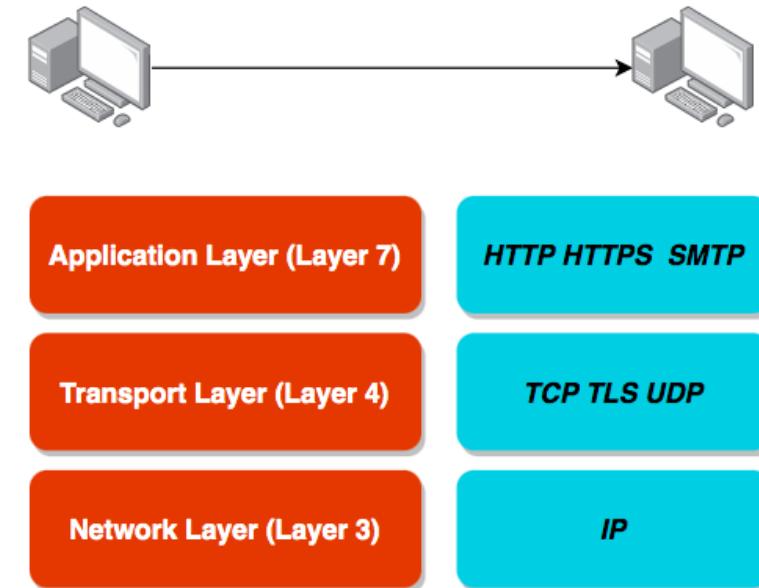
- Distribute traffic across EC2 instances in one or more AZs in a single region
- **Managed service** - AWS ensures that it is highly available
- Auto scales to handle huge loads
- Load Balancers can be **public or private**
- **Health checks** - route traffic to healthy instances



# HTTP vs HTTPS vs TCP vs TLS vs UDP

In 28  
Minutes

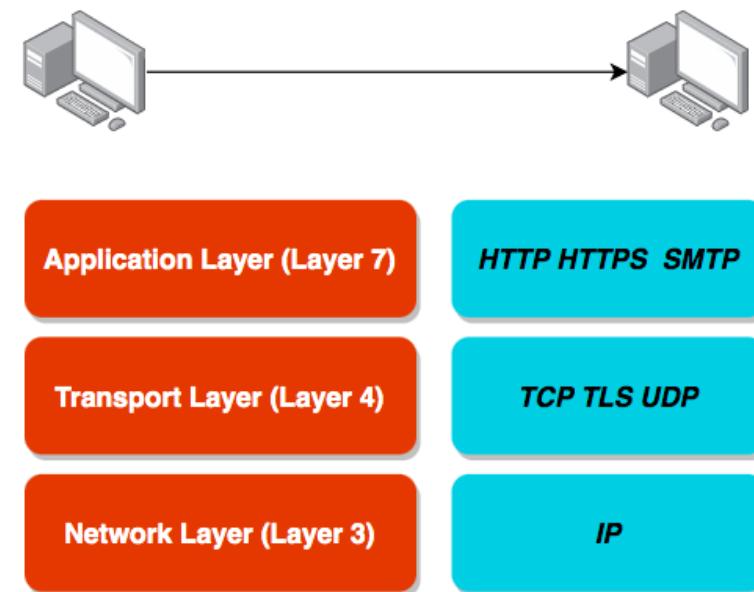
- Computers use protocols to communicate
- Multiple layers and multiple protocols
- **Network Layer** - Transfer bits and bytes
- **Transport Layer** - Are the bits and bytes transferred properly?
- **Application Layer** - Make REST API calls and Send Emails
- (Remember) Each layer makes use of the layers beneath it
- (Remember) Most applications talk at application layer. BUT some applications talk at transport layer directly(high performance).



# HTTP vs HTTPS vs TCP vs TLS vs UDP

In 28  
Minutes

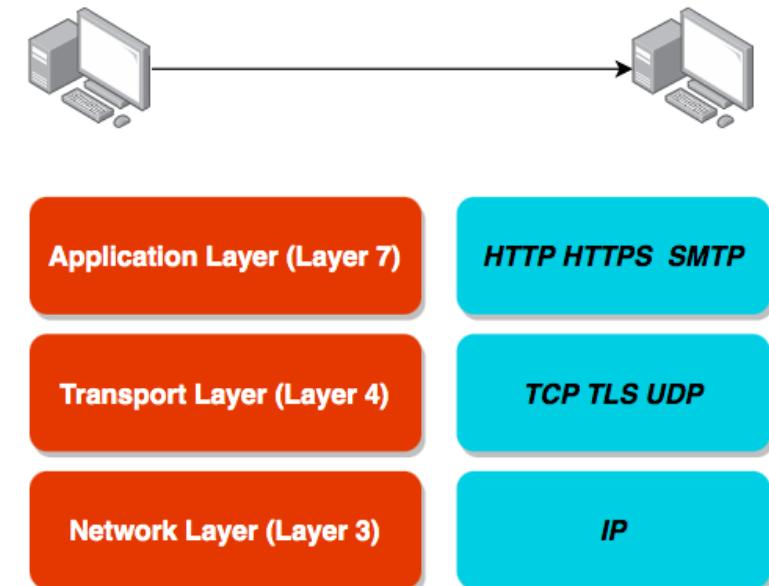
- Network Layer:
  - IP (Internet Protocol): Transfer bytes. Unreliable.
- Transport Layer:
  - TCP (Transmission Control): Reliability > Performance
  - TLS (Transport Layer Security): Secure TCP
  - UDP (User Datagram Protocol): Performance > Reliability
- Application Layer:
  - HTTP(Hypertext Transfer Protocol): Stateless Request Response Cycle
  - HTTPS: Secure HTTP
  - SMTP: Email Transfer Protocol
  - and a lot of others...



# HTTP vs HTTPS vs TCP vs TLS vs UDP

In 28  
Minutes

- Most applications typically communicate at application layer
  - Web apps/REST API(HTTP/HTTPS), Email Servers(SMTP), File Transfers(FTP)
  - All these applications use TCP/TLS at network layer(for reliability)
- **HOWEVER** applications needing high performance directly communicate at transport layer:
  - Gaming applications and live video streaming use UDP (sacrifice reliability for performance)
- **Objective:** Understand Big Picture. Its OK if you do not understand all details.



# Three Types of Elastic Load Balancers

In 28  
Minutes

- **Classic Load Balancer ( Layer 4 and Layer 7)**
  - Old generation supporting Layer 4(TCP/TLS) and Layer 7(HTTP/HTTPS) protocols
  - Not Recommended by AWS
- **Application Load Balancer (Layer 7)**
  - New generation supporting HTTP/HTTPS and advanced routing approaches
- **Network Load Balancer (Layer 4)**
  - New generation supporting TCP/TLS and UDP
  - Very high performance usecases

# Classic Load Balancer

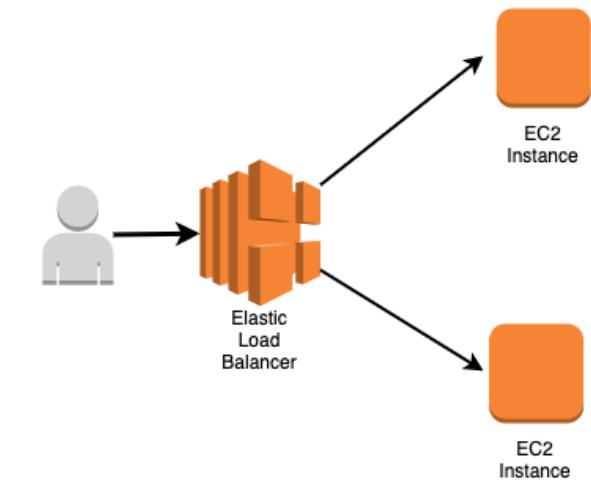
In 28  
Minutes

- Older version of ELB
- Not recommended anymore
- Supports TCP, SSL/TLS and HTTP(S) (Layer 4 and 7)
- Demo: Create a Classic Load Balancer

# Application Load Balancer

In 28  
Minutes

- Most popular and frequently used ELB in AWS
- Supports WebSockets and HTTP/HTTPS (Layer 7)
- Supports all important load balancer features
- Scales **automatically** based on demand (Auto Scaling)
- Can load balance between:
  - EC2 instances (AWS)
  - Containerized applications (Amazon ECS)
  - Web applications (using IP addresses)
  - Lambdas (serverless)
- **Demo** : Create an Application Load Balancer



# Load Balancers - Security Group Best Practice

In 28  
Minutes

Load Balancer allow traffic from everywhere!

Inbound rules			
Type	Protocol	Port range	Source
HTTP	TCP	80	0.0.0.0/0

EC2 Security Group **ONLY** allows traffic from Load Balancer Security Group

Inbound rules			
Type	Protocol	Port range	Source
HTTP	TCP	80	sg-03eb042440351fdad (awseb-e-grzepvhv3-stack-AWSEBLoadBalancerSecurityGroup-1EG2SPQRTIQ02)

(Best Practice) Restrict allowed traffic using Security Groups

# Listeners

In 28  
Minutes

The screenshot shows the 'Listeners' tab selected in a navigation bar. Below the navigation bar, a descriptive text states: '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.' Below this text are three buttons: 'Add listener' (blue), 'Edit' (grey), and 'Delete' (grey). The main area displays a table with one row. The columns are: Listener ID, Security policy, SSL Certificate, and Rules. The first column contains a checkbox and the text 'HTTP : 80'. The second column contains 'N/A'. The third column contains 'N/A'. The fourth column contains 'Default: forwarding to awseb-AWSEB-77UXO29Z6IMQ' and a 'View/edit rules' link.

<input type="checkbox"/> Listener ID	Security policy	SSL Certificate	Rules
<input type="checkbox"/> HTTP : 80 arn...6d4fd3790d1b96d9 ▾	N/A	N/A	Default: forwarding to awseb-AWSEB-77UXO29Z6IMQ <a href="#">View/edit rules</a>

- Each Load Balancer has **one or more listeners** listening for connection requests from the client
- Each listener has:
  - a protocol
  - a port
  - a set of rules to route requests to targets

# Multiple Listeners

In 28  
Minutes

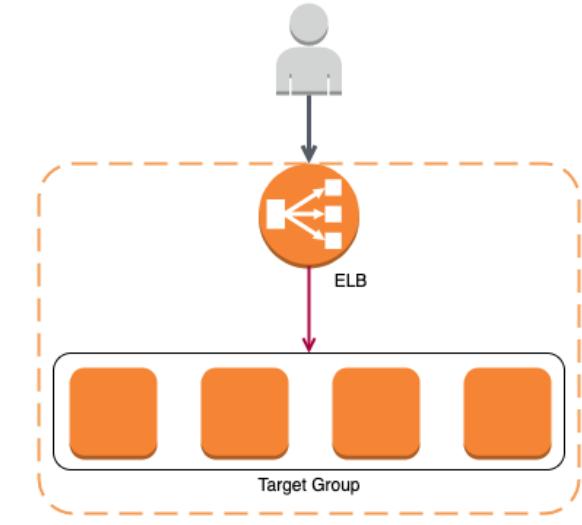
<input type="checkbox"/>	Listener ID	Security policy	SSL Certificate	Rules
<input type="checkbox"/>	HTTP : 80 arn...6d4fd3790d1b96d9▼	N/A	N/A	Default: forwarding to awseb-AWSEB-77UXO29Z6IMQ <a href="#">View/edit rules</a>
<input type="checkbox"/>	HTTP : 443 ▲ arn...770a0d2977599957▼	N/A	N/A	Default: redirecting to HTTP://#{host}:80/#{path}?#{query} <a href="#">View/edit rules</a>
<input type="checkbox"/>	HTTP : 8080 arn...8659e53f87d96af9▼	N/A	N/A	Default: returning fixed response 400 <a href="#">View/edit rules</a>

- You can have multiple listeners listening for a different protocol or port
- In the above example:
  - HTTP requests on port 80 are routed to the EC2 instances target group
  - HTTPS requests on port 443 are routed to port 80
  - HTTP requests on port 8080 get a fixed response (customized HTML)

# Target Groups

In 28  
Minutes

- How to group instances that ALB has to distribute the load between?
  - Create a Target Group
- A target group can be:
  - A set of EC2 instances
  - A lambda function
  - Or a set of IP addresses

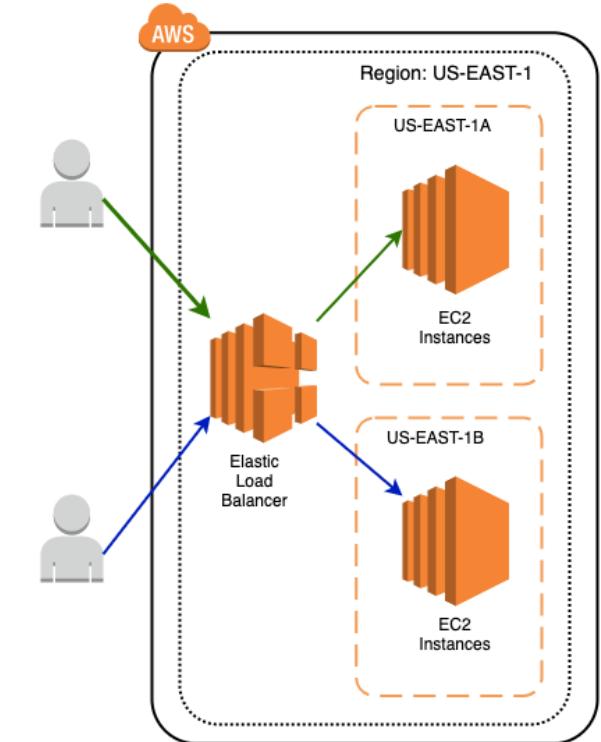


# Target Group Configuration - Sticky Session

In 28  
Minutes

*Enable sticky user sessions*

- Send all requests from one user to the same instance
- Implemented using a cookie
- Supported by ALB and CLB



# Target Group Configuration - Deregistration delay

In 28  
Minutes

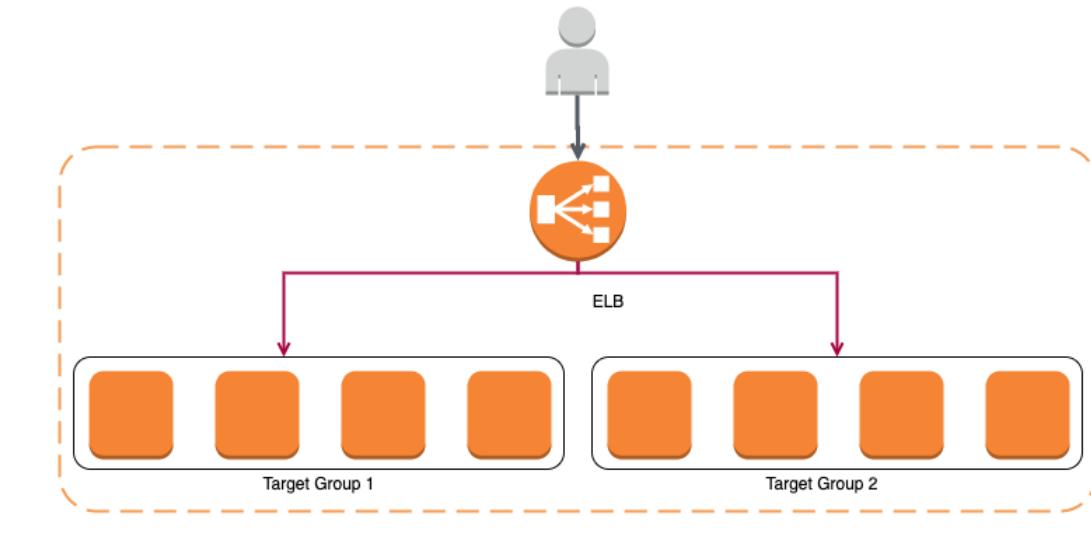
*How long should ELB wait before de-registering a target?*

- Load balancer stops routing new requests to a target when you unregister it
- What about requests that are **already in progress** with that target?
- This setting ensures that load balancer gives **in-flight requests** a chance to complete execution
- 0 to 3600 seconds (default 300 seconds)
- Also called Connection Draining

# Microservices architectures - Multiple Target Group(s)

In 28  
Minutes

- Microservices architectures have 1000s of microservices
  - <http://www.xyz.com/microservice-a>
  - <http://www.xyz.com/microservice-b>
- Should we create multiple ALBs?
- **Nope.** One ALB can support multiple microservices!
- Create separate target group for each microservices
- (Remember) Classic Load Balancer, **does NOT** support multiple target groups.



# Listener Rules

In 28  
Minutes

The screenshot shows the AWS Lambda Listener Rules configuration interface. It displays two rules, each consisting of an IF condition and a THEN action.

**Rule 1:**

- IF:** Path is /microservice-a
- THEN:** Forward to TARGET\_GROUP\_A: 1 (100%)  
Group-level stickiness: Off

**Rule 2:**

- IF:** Path is /microservice-b
- THEN:** Forward to TARGET\_GROUP\_B: 1 (100%)  
Group-level stickiness: Off

- How do I identify which request should be sent to which target group?
- Configure multiple listener rules for the same listener
- Rules are executed in the order they are configured.
- Default Rule is executed last.

# Listener Rules - Possibilities

In 28  
Minutes

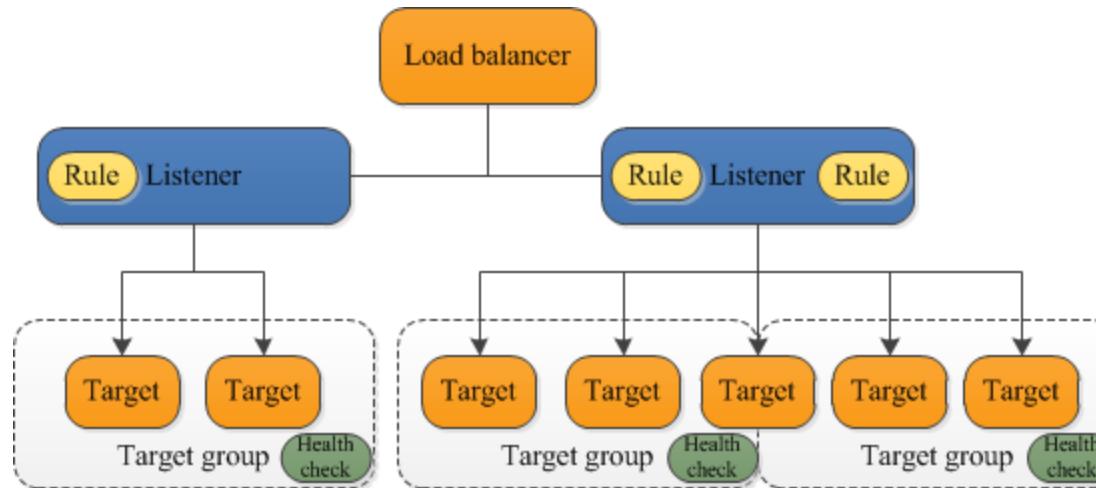
The screenshot shows two listener rules defined in AWS Lambda:

- Rule 1:** ARN: arn:aws:lambda:region:account-id:function:rule-1. IF Path is /microservice-a THEN Forward to TARGET\_GROUP\_A: 1 (100%). Group-level stickiness: Off.
- Rule 2:** ARN: arn:aws:lambda:region:account-id:function:rule-2. IF Path is /microservice-b THEN Forward to TARGET\_GROUP\_B: 1 (100%). Group-level stickiness: Off.

- Based on **path** - in28minutes.com/a to target group A and in28minutes.com/b to target group B
- Based on **Host** - a.in28minutes.com to target group A and b.in28minutes.com to target group B
- Based on **HTTP headers** (Authorization header) and methods (POST, GET, etc)
- Based on **Query Strings** (/microservice?target=a, /microservice?target=b)
- Based on **IP Address** - all requests from a range of IP address to target group A. Others to target group B

# Architecture Summary

In 28  
Minutes

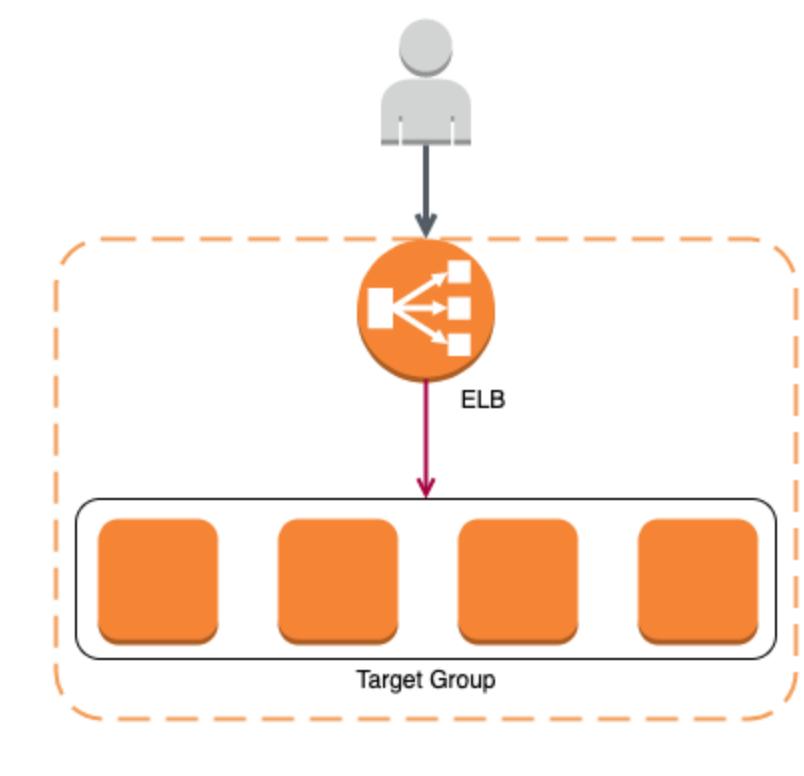


[https://docs.amazonaws.cn/en\\_us/elasticloadbalancing/latest/application/introduction.html](https://docs.amazonaws.cn/en_us/elasticloadbalancing/latest/application/introduction.html)

- Highly decoupled architecture
- Load balancer can have multiple listeners (protocol + port combinations).
- Each listener can have multiple rules each routing to a target group based on request content.
- A target can be part of multiple target groups.

# Introducing Auto Scaling Groups

In 28  
Minutes

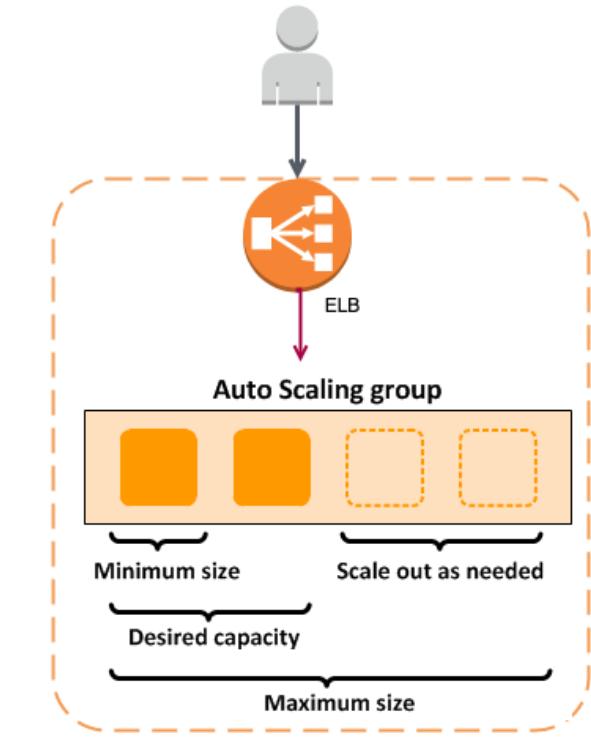


- Target Groups are configured with a static set of instances. How do you scale out and scale in **automatically**?
  - Configure a Auto Scaling Group

# Auto Scaling Groups

In 28  
Minutes

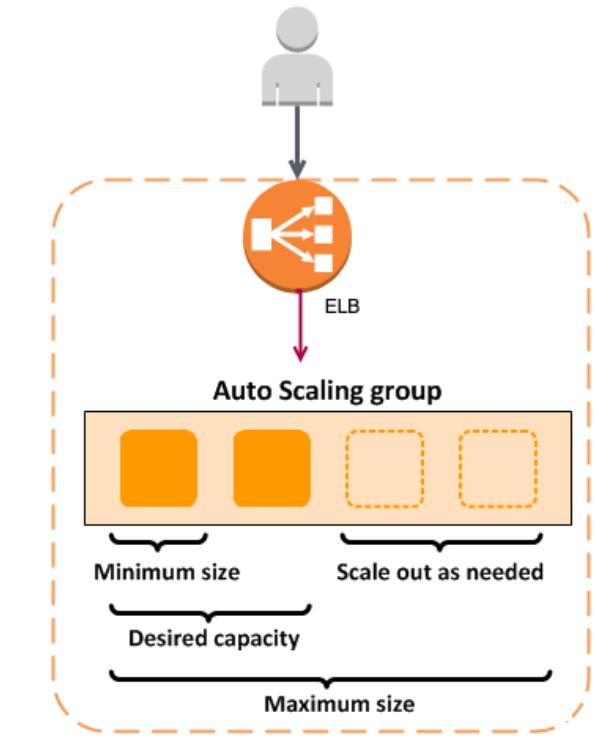
- Auto Scaling Group responsibilities:
  - Maintain configured number of instances (using periodic health checks)
    - If an instance goes down, ASG launches replacement instance
  - Auto scale to adjust to load (scale-in and scale-out based on auto scaling policies)
- ASG can launch On-Demand Instances, Spot Instances, or both
  - Best Practice: Use Launch Template
- An ELB can distribute load to active instances as ASG expands and contracts based on the load
- DEMO: Creating Auto Scaling Groups



# Auto Scaling Components

In 28  
Minutes

- **Launch Configuration/Template**
  - EC2 instance size and AMI
- **Auto Scaling Group**
  - Reference to Launch Configuration/Template
  - Min, max and desired size of ASG
  - EC2 health checks by default. Optionally enable ELB health checks.
  - **Auto Scaling Policies**
    - When and How to execute scaling?



# Auto Scaling Group - Use Cases

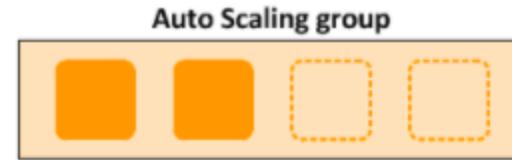
In 28  
Minutes



ASG Use case	Description	More details
Maintain current instance levels at all times	$\text{min} = \text{max} = \text{desired} = \text{CONSTANT}$ When an instance becomes unhealthy, it is replaced.	Constant load
Scale manually	Change desired capacity as needed	You need complete control over scaling
Scale based on a schedule	Schedule a date and time for scaling up and down.	Batch programs with regular schedules
Scale based on demand (Dynamic/Automatic Scaling)	Create scaling policy (what to monitor?) and scaling action (what action?)	Unpredictable load

# Dynamic Scaling Policy Types

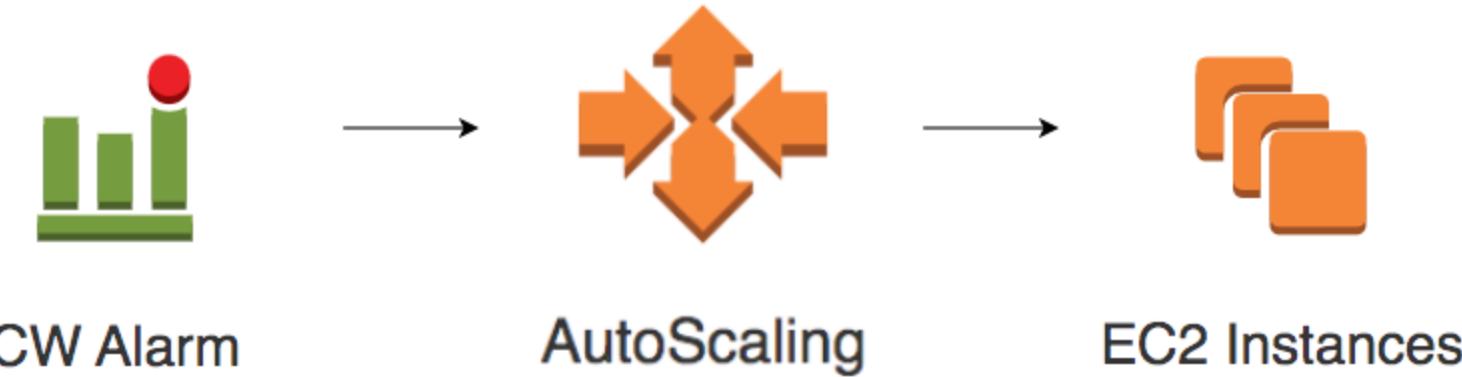
In 28  
Minutes



Scaling Policy	Example(s)	Description
Target tracking scaling	Maintain CPU Utilization at 70%.	Modify current capacity based on a target value for a specific metric.
Simple scaling	+5 if CPU utilization > 80% -3 if CPU utilization < 60%	Waits for cooldown period before triggering additional actions.
Step scaling	+1 if CPU utilization between 70% and 80% +3 if CPU utilization between 80% and 100% Similar settings for scale down	Warm up time can be configured for each instance

# Scaling Policies - Background

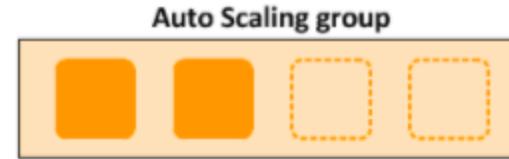
In 28  
Minutes



- Two parts:
  - CloudWatch alarm (Is CPU utilization  $>80\%$ ? or  $< 60\%$ ).
  - Scaling action (+5 EC2 instances or -3 EC2 instances)

# Auto Scaling - Scenarios

In 28  
Minutes



## Scenario

**Change instance type or size of ASG instances**

## Solution

Launch configuration or Launch template cannot be edited. Create a new version and ensure that the ASG is using the new version. Terminate instances in small groups.

**Roll out a new security patch (new AMI) to ASG instances**

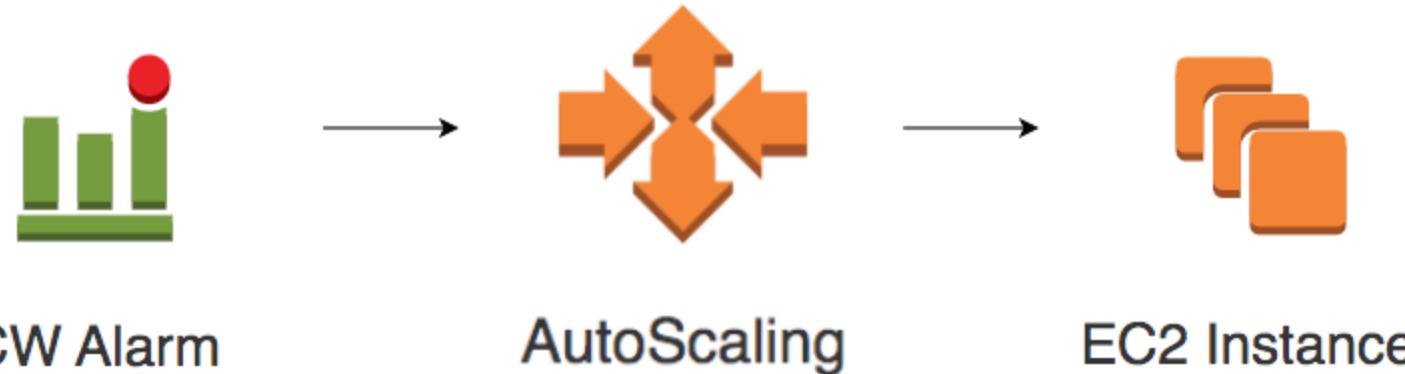
Same as above.

**Perform actions before an instance is added or removed**

Create a Lifecycle Hook. You can configure CloudWatch to trigger actions based on it.

# Auto Scaling - Scenarios

In 28  
Minutes



## Scenario

Which instance in an ASG is terminated first when a scale-in happens?

Preventing frequent scale up and down

I would want to protect newly launched instances from scale-in

## Solution

(Default Termination Policy) Within constraints, goal is to distribute instances evenly across available AZs. Next priority is to terminate older instances.

Adjust cooldown period to suit your need (default - 300 seconds). Align CloudWatch monitoring interval

Enable instance scale-in protection

# Network Load Balancer

In 28  
Minutes

- Functions at the **Transport Layer** - Layer 4 (Protocols TCP, TLS and UDP)
- For **high performance** use cases (millions of requests per second)
- Can be assigned a **Static IP/Elastic IP**
- Can load balance between:
  - EC2 instances
  - Containerized applications (Amazon ECS)
  - Web applications (using IP addresses)
- Demo

## Elastic Load Balancer

- Distribute traffic across EC2 instances in one or more AZs in a single region
- **Managed Service** - highly available, Auto scales, public or private

## Classic Load Balancer

- Layer 4(TCP/TLS) and Layer 7(HTTP/HTTPS)
- **Old.** Not Recommended by AWS

## Network Load Balancer

- Layer 4(TCP/TLS and UDP)
- **Very high performance usecases**
- Can be assigned a Static IP/Elastic IP

## Application Load Balancer

- Layer 7(HTTP/HTTPS)
- Supports **advanced routing approaches** (path, host, http headers, query strings and origin IP addresses)
- Load balance between EC2 instances, containers, IP addresses and lambdas

## Concepts

- Each Load Balancer has one or more **listeners** (different protocol or port) listening for connection requests from the client
- **Target group** is a group representing the targets (ex: EC2 instances)
- One ALB or NLB can support multiple microservices (multiple target groups)!

## Concepts

- **Auto Scaling Group** - Maintain configured number of instances (using periodic health checks). Auto scale to adjust to load.
- **Dynamic Scaling Policies** - Target tracking scaling, Simple scaling and Step scaling.
- **CloudWatch alarms** track the metric (Is CPU utilization >80%? or < 60%) and trigger the auto scaling action (+5 EC2 instances or -3 EC2 instances)

# EC2 & ELB for Architects

*It is not sufficient to get things working. We want more!*

- High Availability
- High Scalability
- Improve Performance
- Improve Security
- Low Costs
- and .....

# Availability

- Are the applications available **when the users need them?**
- **Percentage of time** an application provides the operations expected of it
- **Example:** 99.99% availability. Also called four 9's availability

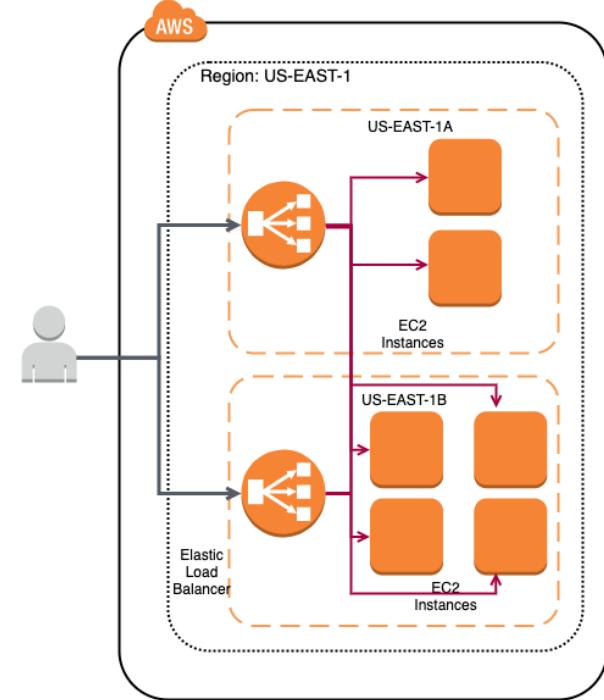
## Availability Table

Availability	Downtime (in a month)	Comment
99.95%	22 minutes	
99.99% (four 9's)	4 and 1/2 minutes	Most online apps aim for 99.99% (four 9's)
99.999% (five 9's)	26 seconds	Achieving 5 9's availability is tough

# Availability Basics - EC2 and ELB

In 28  
Minutes

- Deploy to multiple AZs
- Use Cross Zone Load Balancing
- Deploy to multiple regions
- Configure proper EC2 and ELB health checks



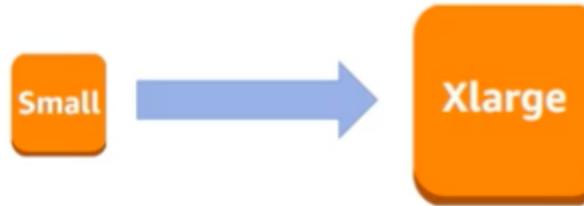
# Scalability

In 28  
Minutes

- A system is handling 1000 transactions per second. Load is expected to increase 10 times in the next month
  - Can we handle a **growth in users, traffic, or data size** without any drop in performance?
  - Does ability to serve more growth increase **proportionally** with resources?
- Ability to **adapt** to changes in demand (users, data)
- What are the options that can be considered?
  - Deploy to a bigger instance with bigger CPU and more memory
  - Increase the number of application instances and setup a load balancer
  - And a lot more.

# Vertical Scaling

In 28  
Minutes



- Deploying application/database to **bigger instance**:
  - A larger hard drive
  - A faster CPU
  - More RAM, CPU, I/O, or networking capabilities
- There are limits to vertical scaling

# Vertical Scaling for EC2

In 28  
Minutes

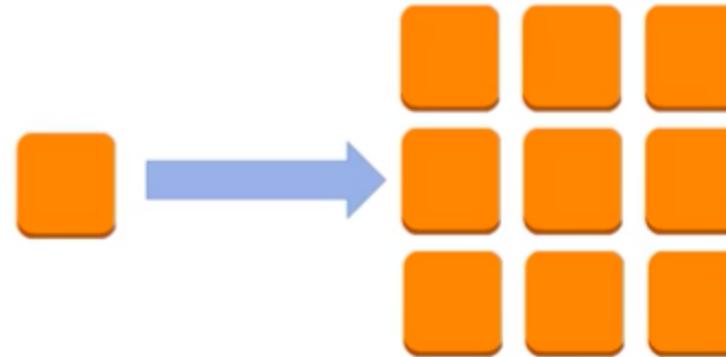
Instance	vCPU*	CPU Credits / hour	Mem (GiB)	Storage	Network Performance
t2.nano	1	3	0.5	EBS-Only	Low
t2.micro	1	6	1	EBS-Only	Low to Moderate
t2.small	1	12	2	EBS-Only	Low to Moderate
t2.medium	2	24	4	EBS-Only	Low to Moderate
t2.large	2	36	8	EBS-Only	Low to Moderate
t2.xlarge	4	54	16	EBS-Only	Moderate
t2.2xlarge	8	81	32	EBS-Only	Moderate

- Increasing EC2 instance size:

- *t2.micro* to *t2.small* or
- *t2.small* to *t2.2xlarge* or
- ...

# Horizontal Scaling

In 28  
Minutes

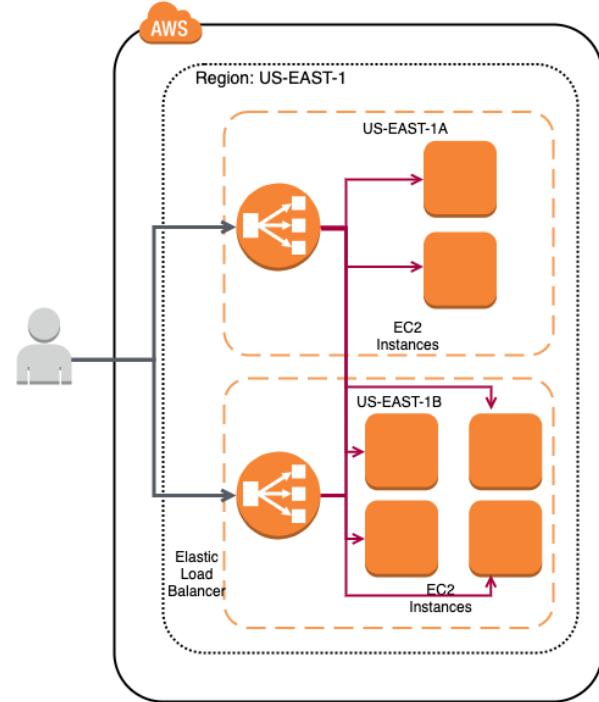


- Deploying multiple instances of application/database
- (Typically but not always) Horizontal Scaling is preferred to Vertical Scaling:
  - Vertical scaling has limits
  - Vertical scaling can be expensive
  - Horizontal scaling increases availability
- (BUT) Horizontal Scaling needs additional infrastructure:
  - Load Balancers etc.

# Horizontal Scaling for EC2

In 28  
Minutes

- Distribute EC2 instances
  - in a single AZ
  - in multiple AZs in single region
  - in multiple AZs in multiple regions
- Auto scale: Auto Scaling Group
- Distribute load : Elastic Load Balancer, Route53



# EC2 Instance Families

In 28  
Minutes

Instance Family	Details	Use Cases
m (m4, m5, m6)	<b>General Purpose.</b> Balance of compute, memory, and networking.	General Purpose : web servers and code repositories
t (t2, t3, t3a)	<b>Burstable performance instances</b> (accumulate CPU credits when inactive). Unlimited burstable mode (new feature)	Workloads with spikes : web servers, developer environments and small databases
c (c4, c5, c5n)	<b>Compute optimized.</b> High performance processors.	Batch processing, high performance http servers, high performance computing (HPC)
r (r4, r5, r5a, r5n)	<b>Memory (RAM) optimized</b>	Memory caches, in-memory databases and real time big data analytics
i (i3, d2)	<b>Storage (I/O) optimized</b>	NoSQL databases and data warehousing

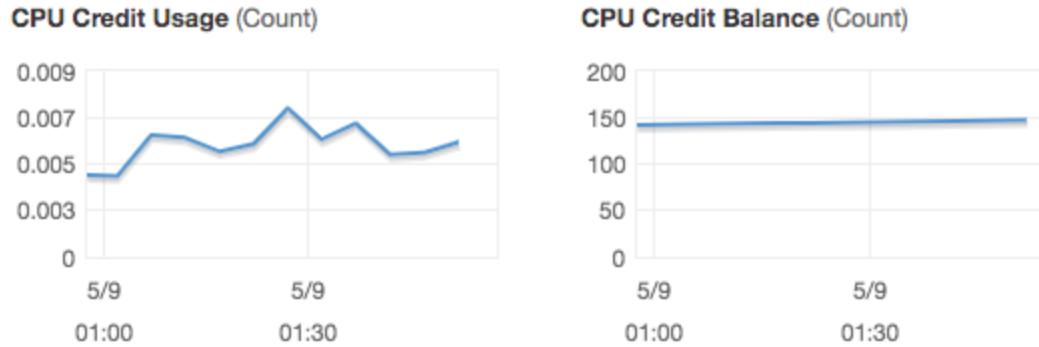
# EC2 Instance Families

In 28  
Minutes

Instance Family	Details	Use Cases
g (g3, g4)	GPU optimized	Floating point number calculations, graphics processing, or video compression
f (f1)	FPGA instances - customizable field programmable gate arrays	Applications needing massively parallel processing power, such as genomics, data analytics, video processing and financial computing
inf (inf1)	Machine learning ASIC instances	Machine learning applications such as image recognition, speech recognition, natural language processing and personalization

# EC2 Burstable Instances (T family - T2, T3 etc)

In 28  
Minutes

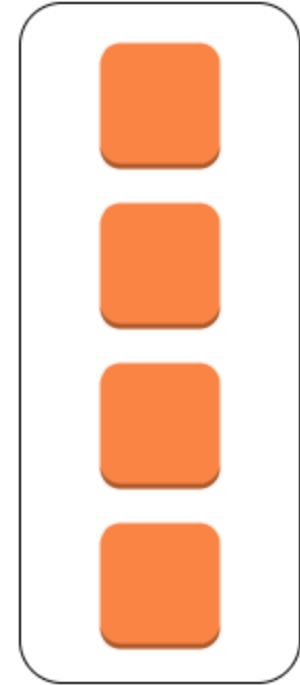


- Instances gather CPU Credits while they are idle
- CPU credits can be consumed at a later point in time (upto a maximum CPU Credit)
- Use case: Workloads with sudden spikes - Test environments
- (Feature) Unlimited Mode - Spike beyond CPU credit at additional cost:
  - For T2 instances, Unlimited Mode is disabled by default.
  - For T3 instances, Unlimited Mode is enabled by default

# EC2 Tenancy - Shared vs Dedicated

In 28  
Minutes

- **Shared Tenancy (Default)**
  - Single host machine can have instances from multiple customers
- **EC2 Dedicated Instances**
  - Virtualized instances on hardware dedicated to one customer
  - You do NOT have visibility into the hardware of underlying host
- **EC2 Dedicated Hosts**
  - Physical servers dedicated to one customer
  - You have visibility into the hardware of underlying host (sockets and physical cores)
  - (Use cases) Regulatory needs or server-bound software licenses like Windows Server, SQL Server



# EC2 Dedicated Instances vs Hosts

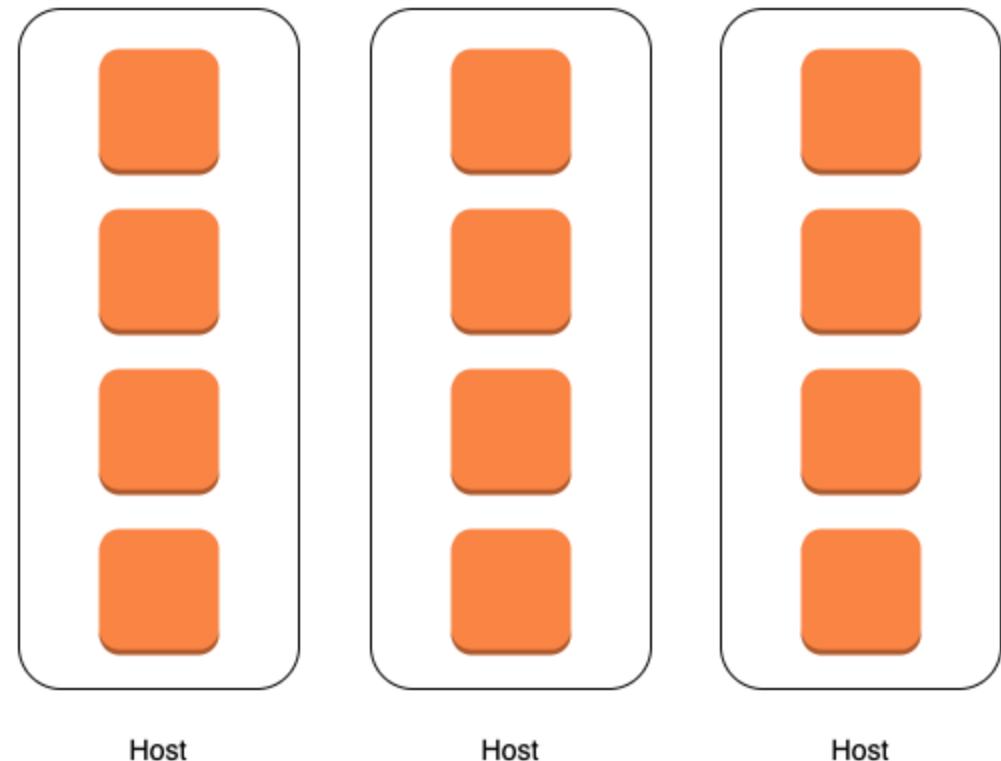
In 28  
Minutes

Feature	Dedicated Instance	Dedicated Host
Billing	Per instance	Per host
Targeted Instance Placement		✓
Access to Underlying Host Hardware		✓

# EC2 Placement Groups

In 28  
Minutes

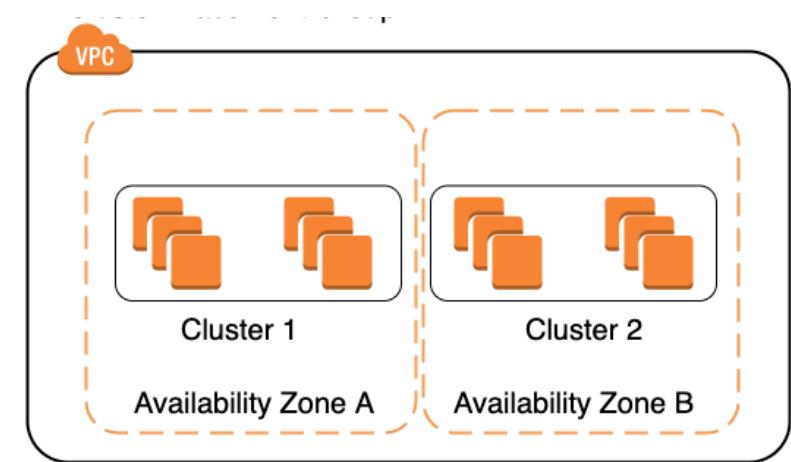
- Certain usecases need control over placement of a group of EC2 instances
  - Low latency network communication
  - High availability
- You DO NOT want EC2 to decide that for you!
- Go for EC2 placement groups!
  - Cluster (low network latency )
  - Spread (avoid simultaneous failures)
  - Partition (multiple partitions with low network latency)



# EC2 Cluster Placement Group

In 28  
Minutes

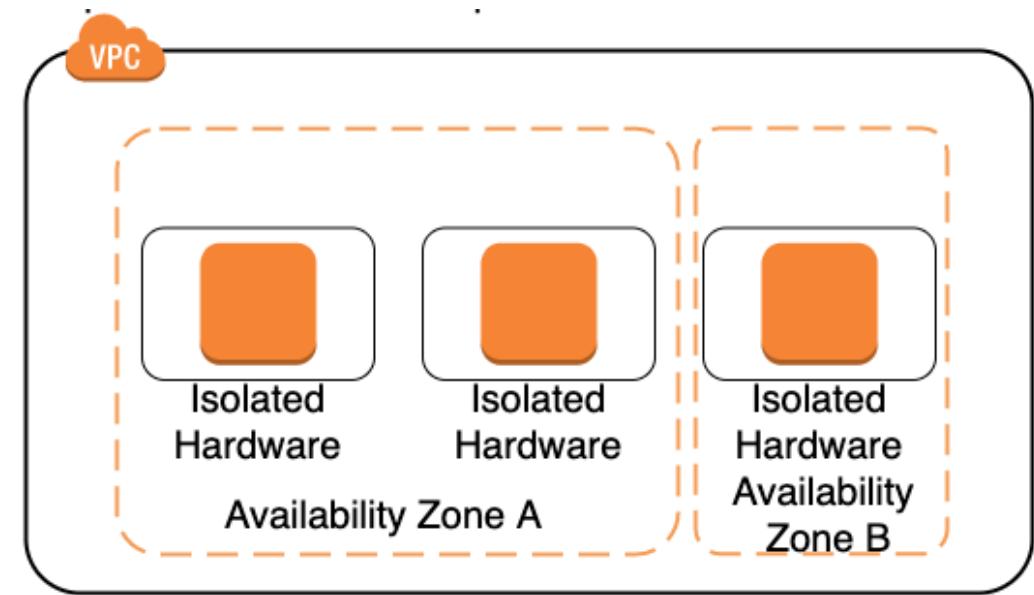
- When low latency network communication between EC2 instances is critical
- Example: Big Data or High Performance Computing needing extreme low latency
- EC2 instances placed near to each other in single AZ
- **High Network Throughput:** EC2 instances can use 10 Gbps or 25Gbps network
- (Disadvantage) Low Availability (Rack crashes => All EC2 instances fail)



# EC2 Spread Placement Group

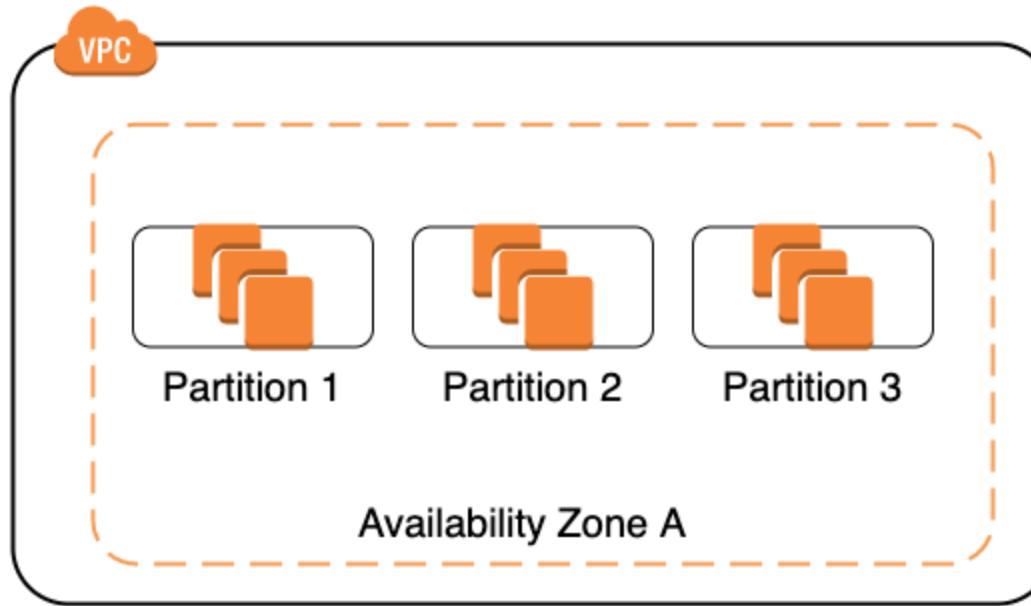
In 28  
Minutes

- Spread EC2 instances across distinct racks
- Each rack has its own network and power source
- **Avoid simultaneous failures** of EC2 instances
- Can be spread across different AZs in same region
- Maximum of seven running instances per AZ in a spread placement group



# EC2 Partition Placement Group - Use Case

In 28  
Minutes

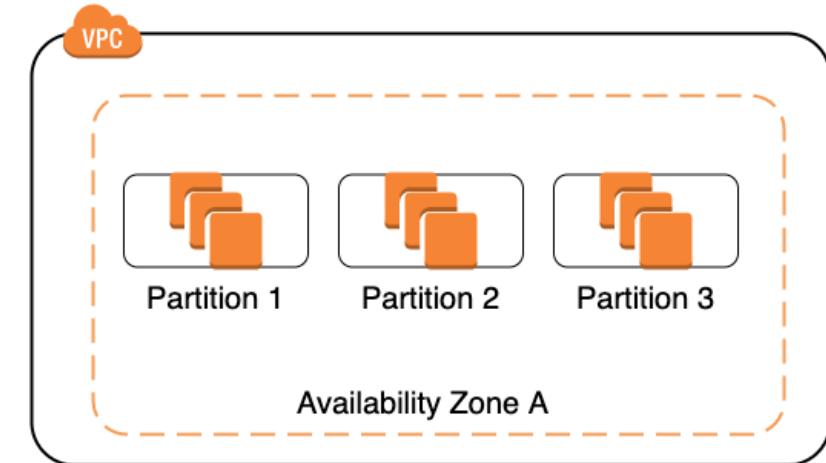


- In large distributed and replicated workloads (HDFS, HBase, and Cassandra), EC2 instances need to be **divided into multiple groups**:
  - Low latency communication between instances in a group
  - Each group is placed on a different rack

# EC2 Partition Placement Group

In 28  
Minutes

- A partition is a group of EC2 instances
- Each partition will be placed on a different rack
- You can choose the partition where EC2 instance is launched into
- Can be spread across **different AZs** in same region
- Maximum of seven partitions per Availability Zone per group



# EC2 Placement Groups - Best Practice

In 28  
Minutes

- **Insufficient capacity error** can happen when:
  - New instances are added in (OR)
  - More than one instance type is used (OR)
  - An instance in placement group is stopped and started
- If you receive a capacity error:
  - Stop and start all instances in the placement group (OR)
  - Try to launch the placement group again
  - Result: Instances may be migrated to a rack that has capacity for all the requested instances
- **Recommendation:**
  - Have only one instance type in a launch request AND
  - Launch all instances in a single launch request together

# Elastic Network Interface

In 28  
Minutes

- Logical networking component that represents a **virtual network card**
- Support IPv4 (110.120.120.145) and IPv6 (2001:0db8:85a3:0000:0000:8a2e:0370:7334)
- Each Elastic Network Interface can provide:
  - One primary and multiple secondary private IP addresses
  - One public address
  - One **Elastic IP address** per private IPv4 address
  - One or more **security groups**



# Elastic Network Interface - Primary and Secondary

In 28  
Minutes

- Each EC2 instance is connected to primary network interface (eth0)
- You can create and attach a secondary network interface - eth1
- Allows an instance to be **dual homed** - present in two subnets in a VPC
- Used to create a **management network** or a low budget high availability solution
- Terminology :
  - **Hot attach**: Attaching ENI when EC2 instance is running
  - **Warm attach**: Attaching ENI when EC2 instance is stopped
  - **Cold attach**: Attaching ENI at launch time of EC2 instance
- Demo



# EC2 Pricing Models Overview

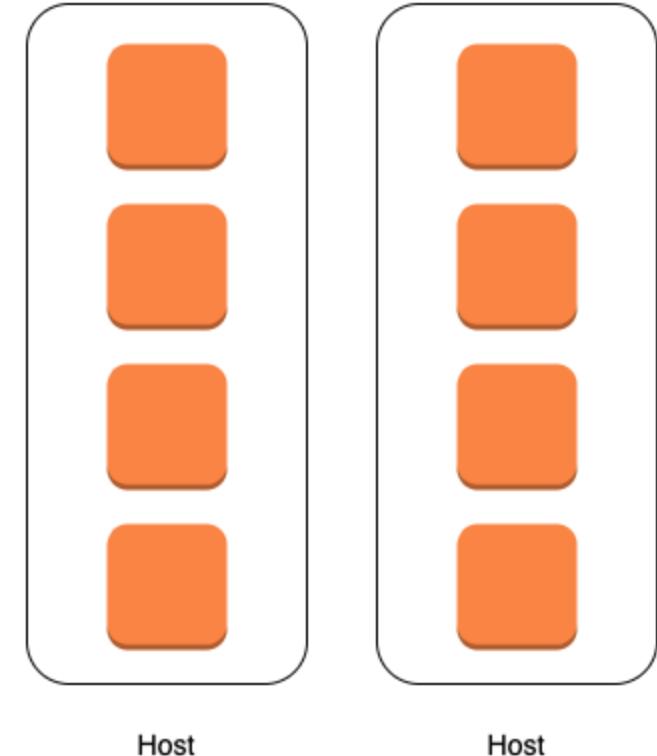
In 28  
Minutes

Pricing Model	Description	Details
On Demand	Request when you want it	Flexible and Most Expensive
Spot	Quote the maximum price	Cheapest (upto 90% off) BUT NO Guarantees
Reserved	Reserve ahead of time	Upto 75% off. 1 or 3 years reservation.
Savings Plans	Commit spending \$X per hour on (EC2 or AWS Fargate or Lambda)	Upto 66% off. No restrictions. 1 or 3 years reservation.

# EC2 On-Demand

In 28  
Minutes

- On demand resource provisioning - Use and Throw!
- Highest cost and highest flexibility
- This is what we have been using until now in this course
- **Ideal for:**
  - A web application which receives spiky traffic
  - A batch program which has unpredictable runtime and cannot be interrupted
  - A batch program being moved from on-premises to cloud for the first time



# EC2 Spot instances

In 28  
Minutes

- (Old Model) Bid a price. Highest bidder wins
- (New Model) Quote your maximum price. Prices decided by long term trends.
- Up to 90% off (compared to On-Demand)
- Can be terminated with a **2 minute notice**
- Ideal for **Non time-critical workloads that can tolerate interruptions** (fault-tolerant)
  - A batch program that does not have a strict deadline AND can be stopped at short notice and re-started
- (Best Practice) Stop or Hibernate EC2 instance on receiving interruption notice

## Spot Block

- Request Spot instances for a **specific duration** (1 or 2 or .. or 6 hours)
- For jobs that take finite time to complete

## Spot Fleet

- Request spot instances across a **range of instance types**
- The more instance types that you specify, the better your chances of having your target capacity fulfilled

# EC2 Linux Spot Instances - Pricing

In 28  
Minutes

- ZERO charge if terminated or stopped by Amazon EC2 in the first instance hour
- Otherwise you are charged by second
- Examples:
  - If EC2 terminates spot instance after 50 minutes, you pay ZERO
  - If you terminate spot instance after 50 minutes, you pay for 50 minutes
  - If either EC2 or you yourselves terminate spot instance after 70 minutes, you pay for 70 minutes

# EC2 Spot Instances - Remember

In 28  
Minutes

- Spot instances can be terminated, stopped, or hibernated when interrupted
  - Default - terminated
  - Use **maintain option** while creating spot request for stop and hibernate options
  - Hibernating a Spot instance allows you to save state of EC2 instances and **quickly start up**
- To **completely close** a spot request:
  - Step 1. Cancel Spot Request
  - Step 2. Terminate all Spot Instances
  - (**Remember**) Canceling a spot request might not terminate active spot instances

# EC2 Reserved Instances

In 28  
Minutes

- Reserve EC2 instances ahead of time!
- **Three types** of reserved instances:
  - Standard
  - Convertible
  - Scheduled
- **Payment models:**
  - No Upfront - \$0 upfront. Pay monthly installment.
  - Partial Upfront - \$XYZ upfront. Pay monthly installment
  - All Upfront - Full amount upfront. \$0 monthly installment.
  - **Cost wise** : Earlier you pay, more the discount. All Upfront < Partial Upfront < No Upfront
  - A difference upto 5%

# EC2 Standard Reserved Instances

In 28  
Minutes

- **Commitment** : (In a region) I reserve an EC2 instance with a **specific platform**(for example, Linux), a **specific instance type**(for example, t2.micro) for a term of **1 year or 3 years**
- You can switch to other instance sizes within the same instance family (t2.large to t2.xlarge)
- You can switch Availability Zones
- You **CANNOT** change instance families, operating systems, or tenancies (default or dedicated)

# EC2 Convertible Reserved Instances

In 28  
Minutes

- **Commitment** : (In a region) I reserve an EC2 instance for a term of **1 year or 3 years**
- You **can change** instance families, operating systems, or tenancies (default or dedicated)
- You can switch Availability Zones and instance size

# EC2 Scheduled Reserved Instances

In 28  
Minutes

- Commitment : (In a region) I reserve an EC2 instance part-time for a year - X hours every month/week/day at a specific time ZZ:ZZ
- (Restriction) Only available for a **few instance types** (ex: C3, C4, M4, R3) **in a few regions** (ex: US East (N. Virginia), US West (Oregon), Europe (Ireland))
- (Use case) Bills are generated on the first day of the month
- (Use case) A batch program runs for a few hours every day
- (Use case) Weekend batch program runs for a few hours every week

# EC2 Reserved Instances - Summary

In 28  
Minutes

- Standard: Commit for a EC2 platform and instance family for 1 year or 3 years. (Up to 75% off)
- Convertible: Standard + **flexibility** to change EC2 platform and instance family. (Up to 54% off)
- Scheduled: Reserve for **specific time period** in a day. (5% to 10% off)
- You can **sell reserved instances** on the AWS Reserved instance marketplace if you do not want to use your reservation

# EC2 Compute Savings Plans

In 28  
Minutes

- **Commitment** : I would spend X dollars per hour on AWS compute resources (Amazon EC2 instances, AWS Fargate and/or AWS Lambda) for a 1 or 3 year period
- Up to 66% off (compared to on demand instances)
- Provides **complete flexibility**:
  - You can change instance family, size, OS, tenancy or AWS Region of your Amazon EC2 instances
  - You can switch between Amazon EC2, AWS Fargate and/or AWS Lambda

# EC2 Instance Savings Plans

In 28  
Minutes

- **Commitment** : I would spend X dollars per hour on Amazon EC2 instances of a specific instance family (General Purpose, for example) within a specific region (us-east-1, for example)
- Up to 72% off (compared to on demand instances)
- You can switch operating systems (Windows to Linux, for example)

# EC2 Pricing Models Overview

In 28  
Minutes

<https://www.ec2instances.info/>

Pricing Model	Use cases
On Demand	Spiky workloads.
Spot	Cost sensitive, Fault tolerant, Non immediate workloads.
Reserved	Constant workloads that run all the time.
Savings Plans	Constant workloads that run all the time and you want more flexibility.

# Monitoring EC2 instances

In 28  
Minutes

CloudWatch metrics: Basic monitoring. [Enable Detailed Monitoring](#)

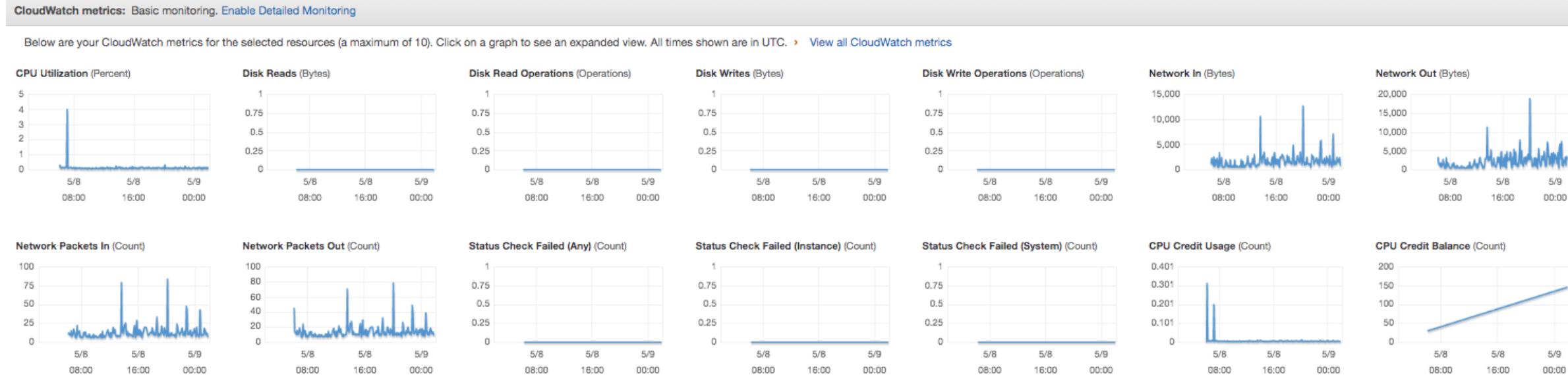
Below are your CloudWatch metrics for the selected resources (a maximum of 10). Click on a graph to see an expanded view. All times shown are in UTC. > [View all CloudWatch metrics](#)



- Amazon CloudWatch is used to **monitor** EC2 instances
- (FREE) **Basic monitoring** ("Every 5 minutes") provided for all EC2 instance types
- (\$\$\$) **EC2 Detailed Monitoring** can be enabled for detailed metrics every 1 minute

# Monitoring EC2 instances - Metrics

In 28  
Minutes



- **EC2 System level metrics (CPU, Disk, Network) are tracked by CloudWatch:**
  - CPU utilization
  - Network In and Out
  - Disk Reads & writes
  - CPU Credit Usage & Balance (For Burstable Instances)

# Monitoring EC2 instances - Custom Metrics

In 28  
Minutes



- CloudWatch does **NOT** have access to **operating system metrics** like memory consumption.
- You can provide those metrics to CloudWatch:
  - Install CloudWatch Agent to send OS metrics to CloudWatch. (OR)
  - Use CloudWatch collectd plug-in

# Elastic Load Balancers

# Secure Communication - HTTPS

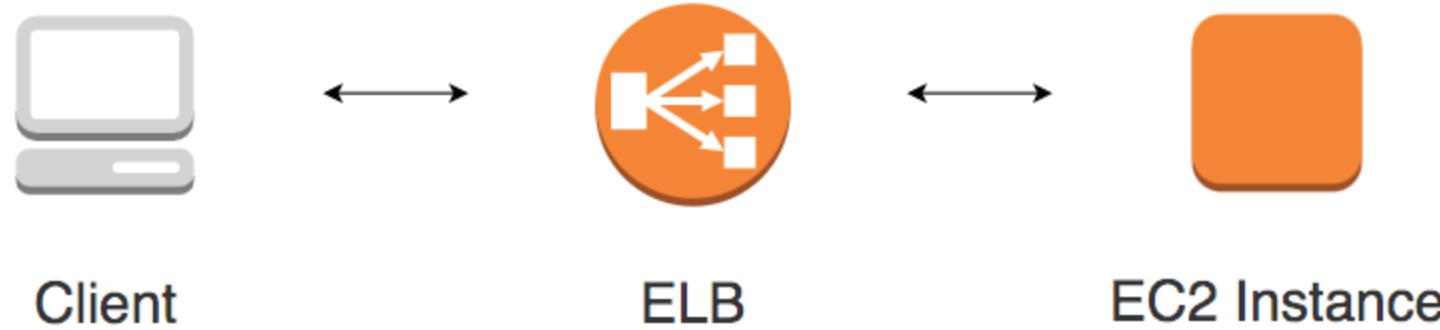
In 28  
Minutes



- Using HTTPS secures the communication on the internet
- To use HTTPS, install SSL/TLS certificates on the server
- In AWS, SSL certificates can be managed using AWS Certificate Manager

# Elastic Load Balancer - Two Communication Hops

In 28  
Minutes



- Client to ELB:
  - Over internet.
  - HTTPS recommended
  - ELB requires X.509 certificates (SSL/TLS server certificates)
- ELB to EC2 instance:
  - Through AWS internal network.
  - HTTP is ok. HTTPS is preferred.

# Elastic Load Balancer - SSL/TLS Termination

In 28  
Minutes

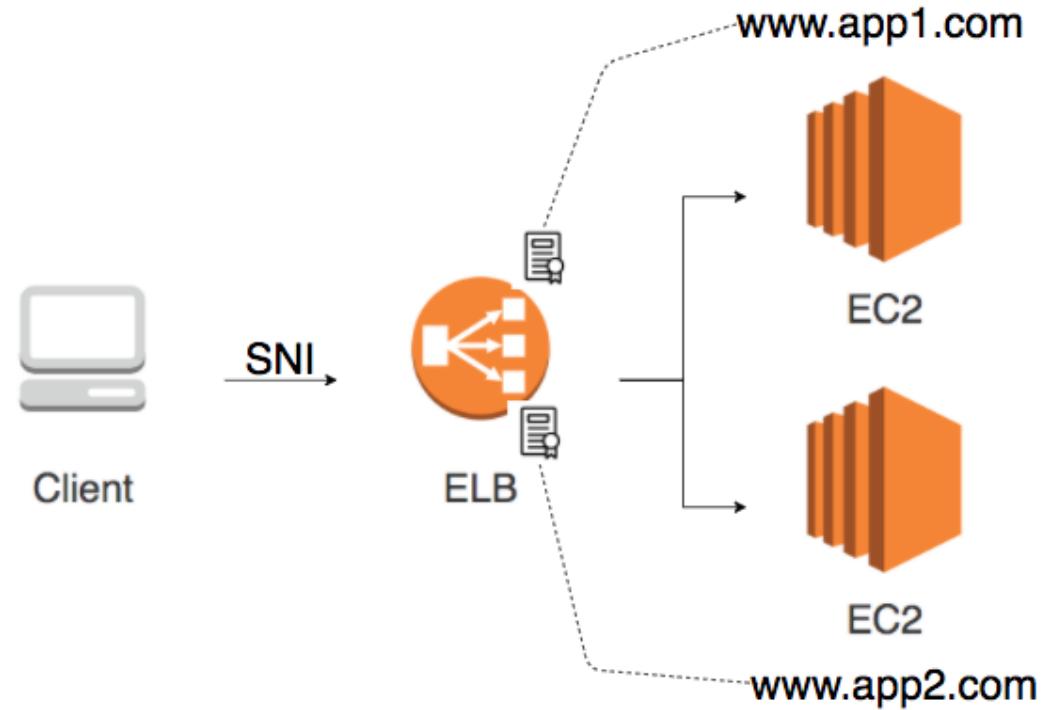


- Application/Classic Load Balancer - SSL Termination
  - Client to ELB: HTTPS
  - ELB to EC2 instance: HTTP
- Network Load Balancer - TLS Termination
  - Client to ELB: TLS
  - ELB to EC2 instance: TCP

# Server Name Indication

In 28  
Minutes

- ALB can provide load balancing for multiple target groups
- Each of these targets can be separate websites with different SSL/TLS certificates
- Each Listener can be associated with multiple SSL certificates(one for each website) to enable this
- Server Name Indication is automatically enabled when multiple SSL certificates are associated with a listener
- Server Name Indication is an extension to TLS protocol
  - Client indicates the host name being contacted at the start of interaction



# Elastic Load Balancer - Logs and Headers

In 28  
Minutes

- You can enable access logs on ELB to capture:
  - Time request was received
  - Client's IP address
  - Latencies
  - Request Paths, and
  - Server Response
- Network Load Balancer allows the EC2 instance to see the client details
- **HOWEVER** Application Load Balancer does NOT
  - Client details are in request headers:
    - X-Forwarded-For: Client IP address
    - X-Forwarded-Proto: Originating Protocol - HTTP/HTTPS
    - X-Forwarded-Port: Originating Port

# ALB vs NLB vs CLB -1

In 28  
Minutes

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Version	New v2	New v2	Old v1
Use cases	Web apps, microservices & containers	Extreme performance - millions of requests with less latency (100ms)	Avoid if possible. Not recommended by AWS.
Protocols Supported	HTTP, HTTPS (Layer 7)	TCP, UDP, TLS (Layer 4)	TCP, SSL/TLS, HTTP, HTTPS(Layer 4 & 7)
Connection draining	✓	✓	✓
Dynamic Host Port Mapping	✓	✓	

# ALB vs NLB vs CLB - 2

In 28  
Minutes

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
Cross-zone load balancing	✓(Always Enabled)	✓(Default Disabled)	✓(Default Disabled)
Sticky sessions	✓	✓	✓
Server Name Indication (SNI)	✓	✓	
Static IP		✓	
Elastic IP address		✓	
Preserve Source IP address		✓	
WebSockets	✓	✓	

# ALB vs NLB vs CLB - Routing

In 28  
Minutes

Feature	Application Load Balancer	Network Load Balancer	Classic Load Balancer
IP addresses as targets	✓	✓ (TCP, TLS)	
Source IP address range (CIDR) based routing	✓		
Path(Route) Based Routing	✓		
Host-Based Routing	✓		
Fixed response	✓		
Lambda functions as targets	✓		
HTTP header-based routing	✓		
HTTP method-based routing	✓		
Query string parameter-based routing	✓		

# Important Load Balancer Scenarios - Quick Review

In 28  
Minutes

Scenario	Solution
You want to maintain sticky sessions	Enable stickiness on ELB(cookie name: AWSELB)
You want to distribute load only to healthy instances	Configure health check. Health check can be a ping, connection or a web page request. You can configure interval, max wait time, threshold for number of failures. An instance can be InService/OutOfService.
Distribute load among two AZs in same region	Enable Cross Zone Load Balancing
How to ensure that in-flight requests to unhealthy instances are given an opportunity to complete execution?	Enable connection draining (1 to 3600 seconds. Default timeout - 300 seconds)
Give warm up time to EC2 instances before they start receiving load from ELB	Configure Health Check Grace Period

# Important Load Balancer Scenarios - Quick Review

In 28  
Minutes

Scenario	Solution
Protect ELB from web attacks - SQL injection or cross-site scripting	Integrate with AWS WAF (Web Application Firewall)
Protect web applications from DDoS attacks	Application Load Balancer (ALB) protects you from common DDoS attacks, like SYN floods or UDP reflection attacks.

## Security

- Use **Security Groups** to restrict traffic
- Place EC2 instances in **private subnets**
- Use **Dedicated Hosts** when you have regulatory needs

## Performance

- Choose right **instance family** (Optimized combination of compute, memory, disk (storage) and networking)
- Use appropriate placement groups
- Prefer creating an **custom AMI** to installing software using userdata
- Choose the right ELB for your use case
  - Prefer Network Load Balancer for **high performance** load balancing

## Cost Efficiency

- Have optimal number and type of EC2 instances running
- Use the **right mix** of:
  - Savings Plans
  - Reserved Instances
  - On demand Instances
  - Spot Instances

## Resiliency

- Configure the right **health checks**
- Use CloudWatch for monitoring
- (Disaster recovery) Upto date AMI copied to multiple regions

# Reference

# Application Load Balancer - Health Check Settings

In 28  
Minutes

- (Goal) Route traffic to healthy instances only!
- Periodic requests are sent to targets to test their status
- Important Settings:
  - **HealthCheckProtocol**: Which protocol?
  - **HealthCheckPort**: Which port?
  - **HealthCheckPath**: Destination path (default - /)
  - **HealthCheckTimeoutSeconds** - Maximum wait time
  - **HealthCheckIntervalSeconds** - How often should a health check be performed?
  - **HealthyThresholdCount** - How many health check successes before marking an instance as healthy?
  - **UnhealthyThresholdCount** - How many health check failures before marking an instance as unhealthy?

# Elastic Load Balancer Terminology

In 28  
Minutes

- **Connection draining** - Before an instance is terminated, requests in execution are given time to complete (deregistration\_delay.timeout\_seconds)
- **Dynamic Host Port Mapping** - Useful with containers. Two instances of the same task can be running on the same ECS container instance
- **Cross-zone load balancing** - Distribute load between available instances in multiple AZs in One Region
- **Sticky sessions** - Send requests from same user to same instance (cookies with configurable expiration - Stickiness duration default - 1 day)
- **Preserve Source IP address** - Allows instances to know where the request is coming from

# Elastic Load Balancer Terminology 2

In 28  
Minutes

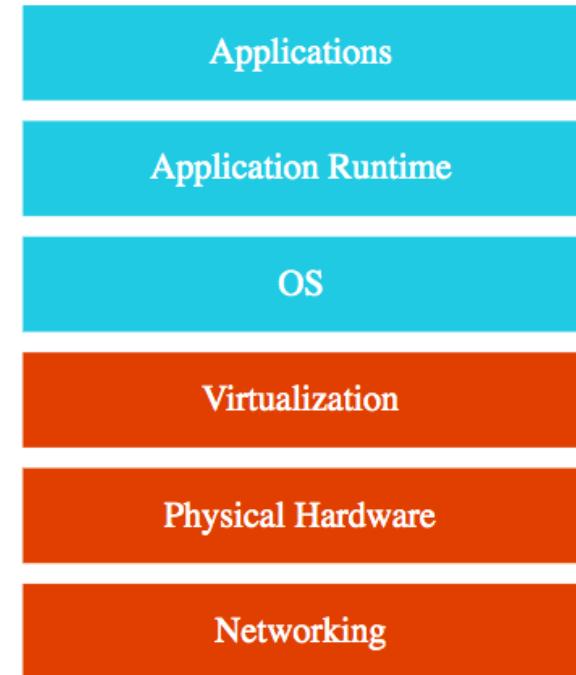
- **WebSockets** - Allows full-duplex communication over a single TCP connection
- **Source IP range (CIDR) based routing** - Redirect to different targets based on the Source CIDR block
- **Path(Route) Based Routing** - Send traffic to different targets based on the path of the request
- **Query string parameter-based routing** - /user?target=target1 vs /user?target=target2
- **Server Name Indication (SNI)** - Support multiple websites with different SSL certificates with one Load Balancer

# AWS Managed Services

# IAAS (Infrastructure as a Service)

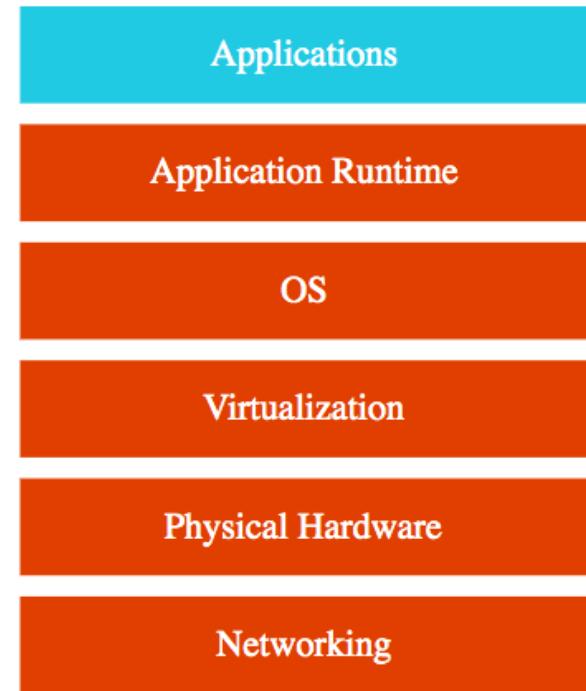
In 28  
Minutes

- Use **only infrastructure** from cloud provider
- Also called "**Lift and Shift**"
- **Example:** Using EC2 to deploy your applications or databases
- You are responsible for:
  - Application Code and Runtime
  - Configuring load balancing
  - Auto scaling
  - OS upgrades and patches
  - Availability
  - etc.. ( and a lot of things!)



# PAAS (Platform as a Service)

- Use a platform provided by cloud
- **Cloud provider** is responsible for:
  - OS (incl. upgrades and patches)
  - Application Runtime
  - Auto scaling, Availability & Load balancing etc..
- **You** are responsible for:
  - Application code
  - Configuration
- **CAAS (Container as a Service)**: Containers instead of Applications
- **FAAS (Function as a Service) or Serverless**: Functions instead of Applications



# AWS Managed Service Offerings

In 28  
Minutes



ELB



ECS



Amazon RDS

- **Elastic Load Balancing** - Distribute incoming traffic across multiple targets
- **AWS Elastic Beanstalk** - Run and Manage Web Apps
- **Amazon Elastic Container Service (ECS)** - Containers orchestration on AWS
- **AWS Fargate** - Serverless compute for containers
- **Amazon Elastic Kubernetes Service (EKS)** - Run Kubernetes on AWS
- **Amazon RDS** - Relational Databases - MySQL, Oracle, SQL Server etc
- And a lot more...

# AWS Elastic Beanstalk

# AWS Elastic BeanStalk

In 28  
Minutes

- **Simplest way** to deploy and scale your web application in AWS
  - Provides end-to-end web application management
- Supports Java, .NET, Node.js, PHP, Ruby, Python, Go, and Docker applications
- **No usage charges** - Pay only for AWS resources you provision
- **Features:**
  - Automatic load balancing
  - Auto scaling
  - Managed platform updates
  - Application health monitoring

# AWS Elastic Beanstalk Demo

In 28  
Minutes

- Deploy an application to cloud using AWS Elastic Beanstalk

# AWS Elastic Beanstalk Concepts

In 28  
Minutes

- **Application** - A container for environments, versions and configuration
- **Application Version** - A specific version of deployable code (stored in S3)
- **Environment** - An application version deployed to AWS resources. You can have multiple environments running different application versions for the same application.
- **Environment Tier:**
  - For batch applications, use **worker tier**
  - For web applications, use **web server tier**

# AWS Elastic Beanstalk - Remember

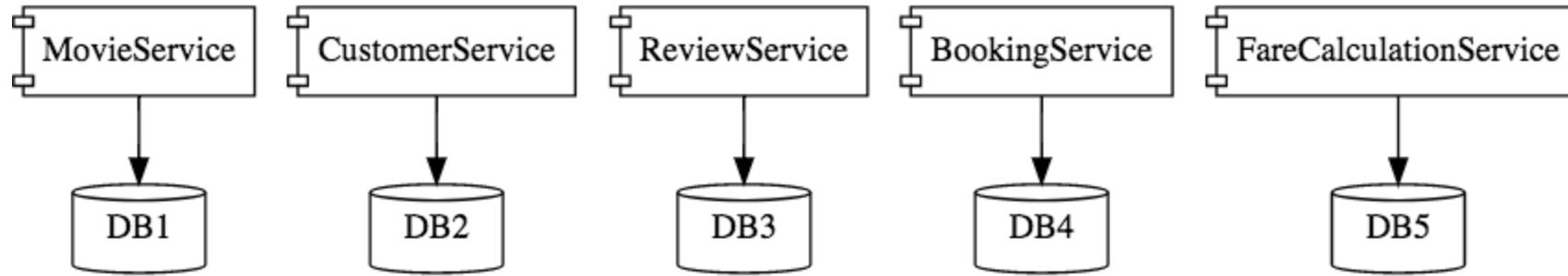
In 28  
Minutes

- You retain full control over AWS resources created
- Ideal for simple web applications
- NOT ideal for microservices architectures
- You can access server logs without logging into the server
- Logs can be stored in Amazon S3 or in CloudWatch Logs
- You can choose to apply patches and platform updates automatically
- Metrics are send to Amazon CloudWatch
- You can configure SNS notifications based on health
- Delete your environment!

# Containers and Container Orchestration

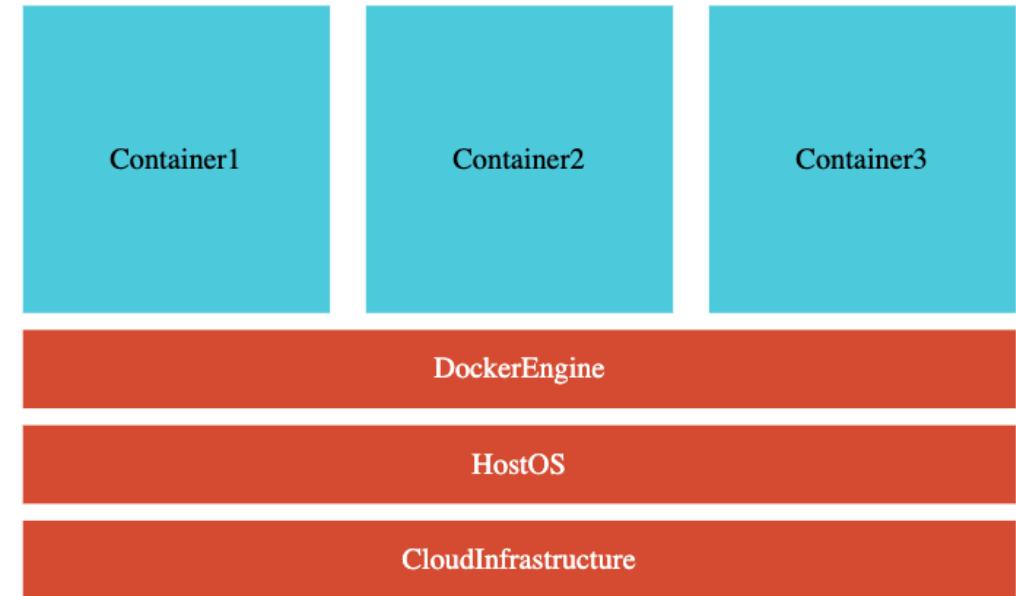
# Microservices

In 28  
Minutes



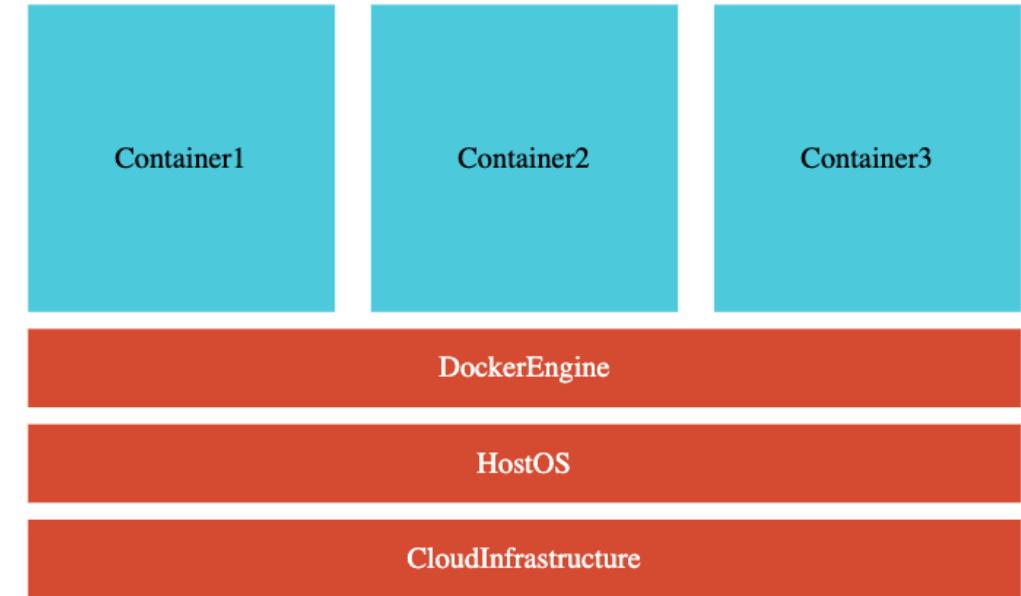
- Enterprises are heading towards microservices architectures
- Build small focused microservices
- **Flexibility to innovate** and build applications in different programming languages (Go, Java, Python, JavaScript, etc)
- **BUT deployments become complex!**
- How can we have **one way of deploying** Go, Java, Python or JavaScript .. microservices?
  - Enter containers!

- Create Docker images for each microservice
- Docker image **contains everything a microservice needs** to run:
  - Application Runtime (JDK or Python or NodeJS)
  - Application code
  - Dependencies
- You can run these docker containers **the same way** on any infrastructure
  - Your local machine
  - Corporate data center
  - Cloud



# Docker - Advantages

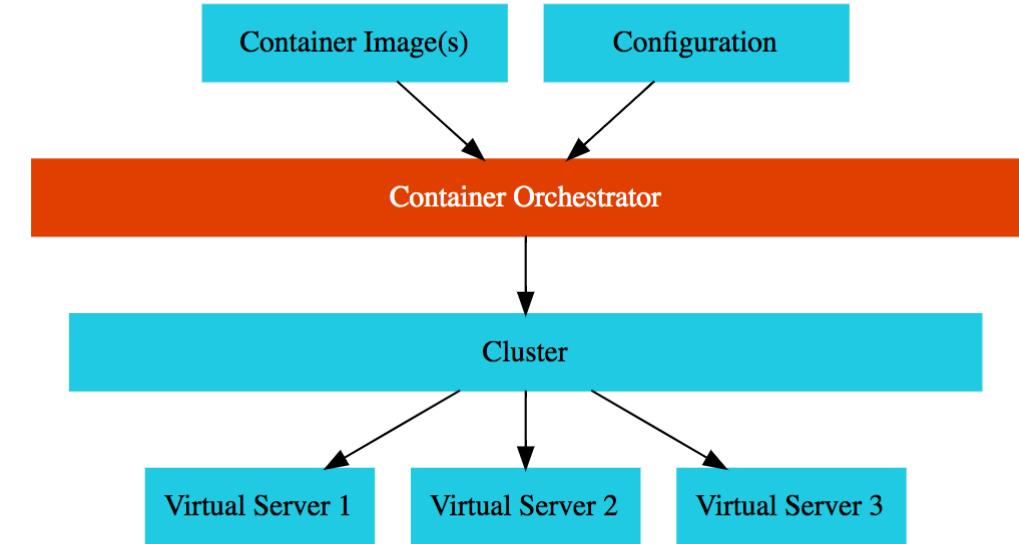
- Docker containers are **light weight** (compared to Virtual Machines)
- Docker provides **isolation** for containers
- Docker is **cloud neutral**
- (NEW CHALLENGE) How do you manage 1000's of containers belonging to multiple microservices?
  - Enter Container Orchestration!



# Container Orchestration

In 28  
Minutes

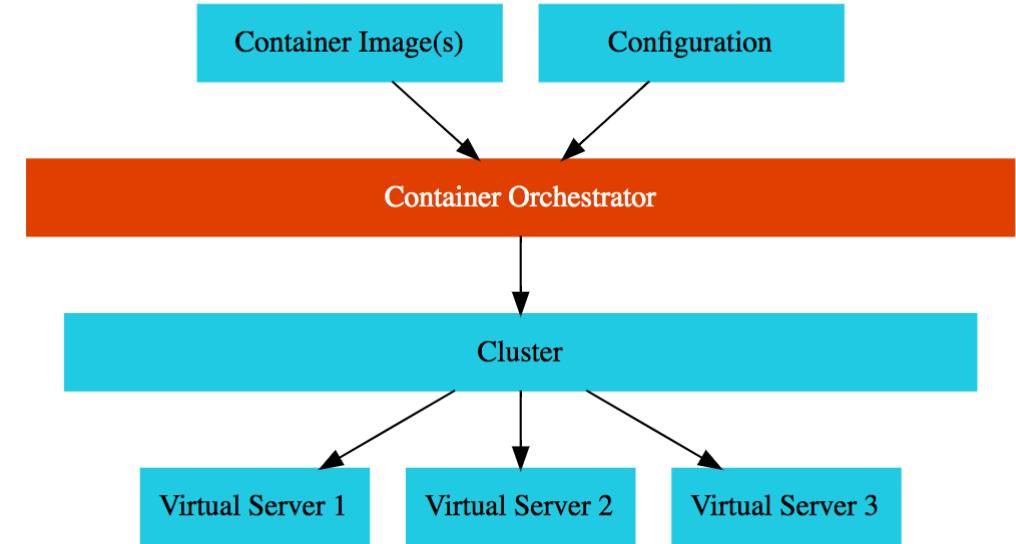
- **Requirement :** I want 10 instances of Microservice A container, 15 instances of Microservice B container and ....
- **Typical Features:**
  - **Auto Scaling** - Scale containers based on demand
  - **Service Discovery** - Help microservices find one another
  - **Load Balancer** - Distribute load among multiple instances of a microservice
  - **Self Healing** - Do health checks and replace failing instances
  - **Zero Downtime Deployments** - Release new versions without downtime



# Container Orchestration Options

In 28  
Minutes

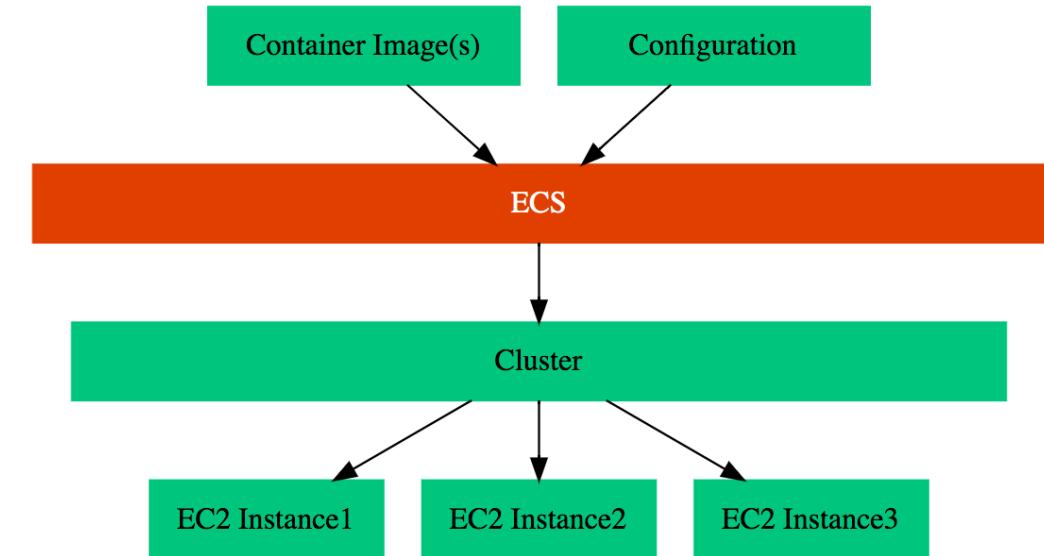
- **Cloud Neutral**
  - Kubernetes
  - AWS service - AWS Elastic Kubernetes Service (EKS)
  - EKS does not have a free tier
- **AWS Specific**
  - AWS Elastic Container Service (ECS)
  - AWS Fargate : Serverless version of AWS ECS
  - AWS Fargate does not have a free tier



# Amazon Elastic Container Service (Amazon ECS)

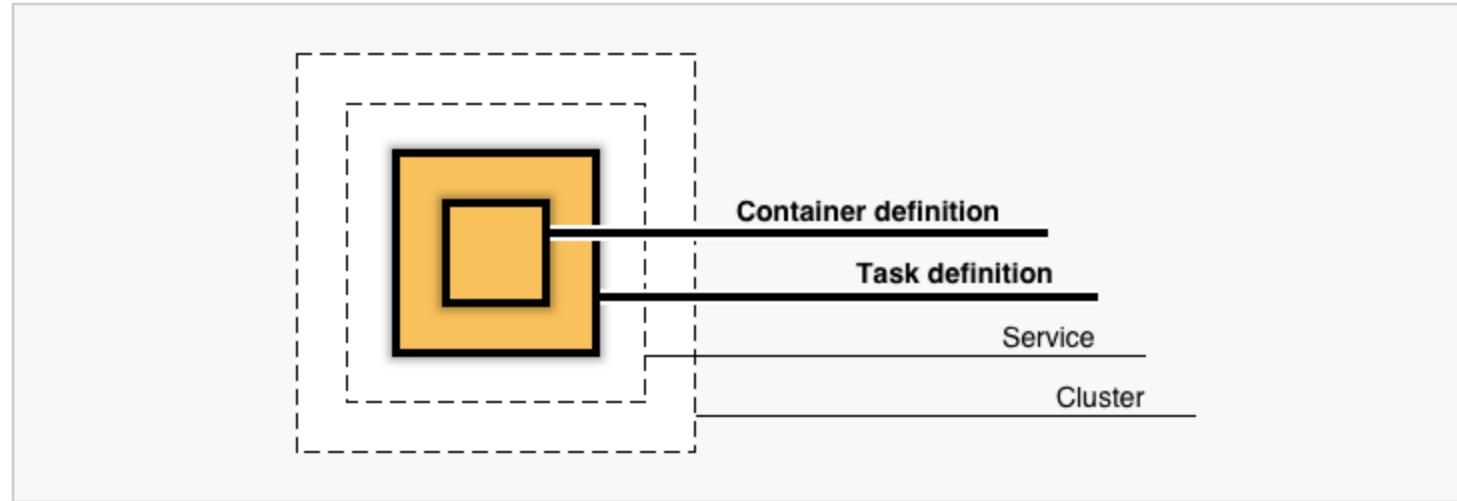
In 28  
Minutes

- Fully managed service for container orchestration
- Serverless option - **AWS Fargate**
- Use cases:
  - Microservices Architectures - Create containers for your microservices and orchestrate them using ECS or Fargate
  - Batch Processing. Run batch workloads on ECS using AWS Batch
- DEMO



# Amazon ECS - Task Definition

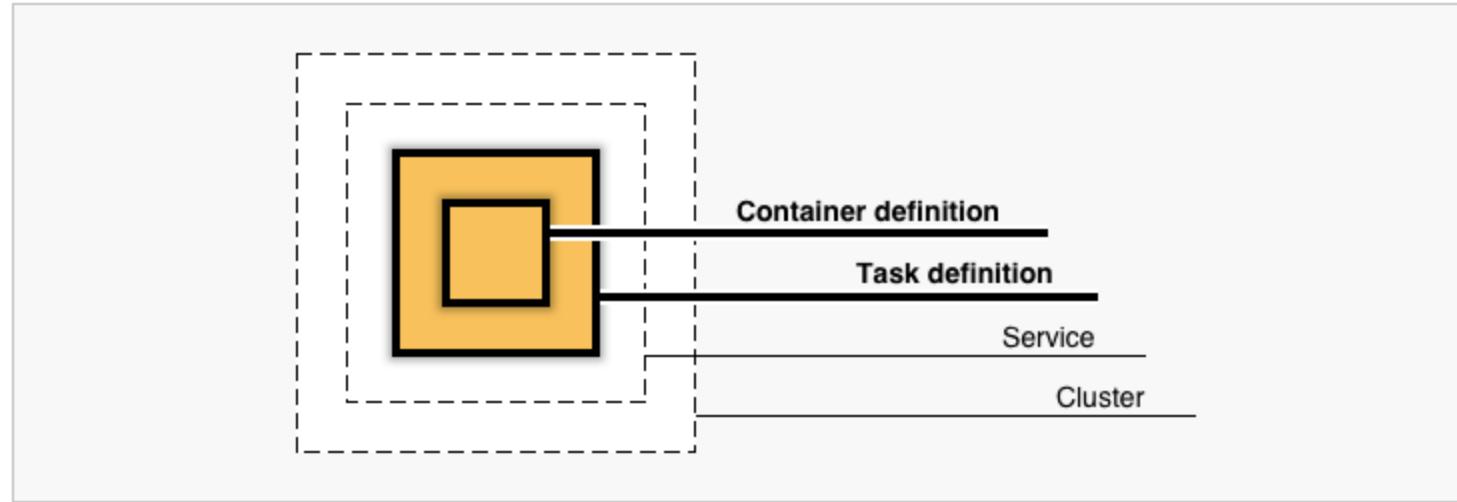
In 28  
Minutes



- **Container Definition(s)**
  - What is the image you want to use?
  - What resources does the container use (memory, CPU and ports)?
- **Task Role (Optional):** If you need access to AWS services (Amazon RDS etc)
- **Task execution IAM role:** Provides permissions to pull container images and publish container logs to Amazon CloudWatch

# Amazon ECS - Terminology

In 28  
Minutes



- **Service**
  - Allows you to run and **maintain** a specified number (the "desired count") of tasks
- **ECS cluster**
  - Grouping of one or more container instances (EC2 instances) where you run your tasks
  - For AWS Fargate (serverless ECS), you DON'T need to worry about EC2 instances

# Amazon Elastic Container Service - Remember

In 28  
Minutes

- **Container Instance** - EC2 instance in the cluster running a **container agent** (helps it communicate with the cluster)
  - AWS provides ECS ready AMIs with container agents pre-installed.
- AWS Fargate does NOT give you visibility into the EC2 instances in the cluster.
- You can use On-Demand instances or Spot instances to create your cluster.
- You can load balance using Application Load Balancers
- Two features of ALB are important for ECS:
  - **Dynamic host port mapping**: Multiple tasks from the same service are allowed per EC2 (container) instance
  - **Path-based routing**: Multiple services can use the same listener port on same ALB and be routed based on path ([www.app.com/microservice-a](http://www.app.com/microservice-a) and [www.app.com/microservice-b](http://www.app.com/microservice-b))
- Delete the cluster!

# Running Docker Containers in AWS

In 28  
Minutes

- **Elastic Beanstalk**
  - Single container or multiple containers in same EC2 instance
  - Recommended for simple web applications
- **Amazon ECS**
  - AWS specific solution for container orchestration
  - Ideal for microservices
- **Amazon Fargate**
  - Serverless version of Amazon ECS
  - You want to run microservices and you don't want to manage the cluster
- **Amazon EKS**
  - AWS managed service for Kubernetes
  - Recommended if you are already using Kubernetes and would want to move the workload to AWS

# Amazon ECR (Elastic Container Registry)

In 28  
Minutes

- You've created docker images for your microservices:
  - Where do you store them?
- You need a **Container Registry**
- **Amazon ECR** is a Fully-managed Docker container registry provided by AWS
- (Alternative) Docker Hub

# Additional Resources

- Docker: <https://www.youtube.com/watch?v=Rt5G5Gj7RP0>
- Kubernetes: <https://www.youtube.com/watch?v=rTNR7vDQDD8>
- AWS Fargate and ECS: <https://www.youtube.com/watch?v=2oXVYxIPs88>

# Serverless Fundamentals - Lambda and API Gateway



AWS Lambda



Lambda Fn



API Gateway

- What are the things we think about when we develop an application?
  - Where do we deploy the application?
  - What kind of server? What OS?
  - How do we take care of scaling the application?
  - How do we ensure that it is always available?
- What if we do not need to worry about servers and focus on building our application?
- Enter Serverless

- Remember: Serverless does NOT mean "No Servers"
- **Serverless for me:**
  - You don't worry about infrastructure
  - Flexible scaling
  - Automated high availability
  - Pay for use:
    - You don't have to provision servers or capacity!
- **You focus on code** and the cloud managed service takes care of all that is needed to scale your code to serve millions of requests!



AWS Lambda



Lambda Fn

- World before Lambda - ELB with EC2 Servers!
- You don't worry about servers or scaling or availability
- You only worry about your code
- You pay for what you use
  - Number of requests
  - Duration of requests
  - Memory consumed

# AWS Lambda - Supported Languages

In 28  
Minutes

- Java
- Go
- PowerShell
- Node.js
- C#
- Python,
- Ruby
- and a lot more...

# AWS Lambda Event Sources

In 28  
Minutes

- Amazon API Gateway
- AWS Cognito
- Amazon DynamoDB (event)
- Amazon CloudFront (Lambda@Edge)
- AWS Step Functions
- Amazon Kinesis (event)
- Amazon Simple Storage Service
- Amazon Simple Queue Service (event)
- Amazon Simple Notification Service
- The list is endless...

# AWS Lambda Demo

In 28  
Minutes

- Let's get our hands dirty!

# AWS Lambda Function

In 28  
Minutes

- Stateless - store data to Amazon S3 or Amazon DynamoDB
- 500MB of non-persistent disk space (/tmp directory)
- Allocate memory in 64MB increments from 128MB to 3GB
  - Lambda cost increases with memory
  - CPU Power increases with memory allocated
  - Inexpensive - <https://aws.amazon.com/lambda/pricing/>
    - Free tier - 1M free requests per month
- Monitor function executions through Amazon CloudWatch
- Maximum allowed time for lambda execution is 900 seconds (default - 3 seconds)
- Integrates with AWS X-Ray(tracing), AWS CloudWatch (monitoring and logs)

# REST API Challenges

In 28  
Minutes



- Most applications today are built around REST API
- Management of REST API is not easy:
  - You've to take care of authentication and authorization
  - You've to be able to set limits (rate limiting, quotas) for your API consumers
  - You've to take care of implementing multiple versions of your API
  - You would want to monitor your API calls
  - You would want to be able to cache API requests

# Amazon API Gateway

In 28  
Minutes



- How about a **fully managed service** with auto scaling that can act as a "**front door**" to your APIs?
- Welcome "**Amazon API Gateway**"

# Amazon API Gateway

In 28  
Minutes



- **"publish, maintain, monitor, and secure APIs at any scale"**
- Integrates with AWS Lambda, Amazon EC2, Amazon ECS or any web application
- Supports HTTP(S) and WebSockets (two way communication - chat apps and streaming dashboards)
- Serverless. Pay for use (API calls and connection duration)
- Demo!

# Amazon API Gateway - Remember

In 28  
Minutes



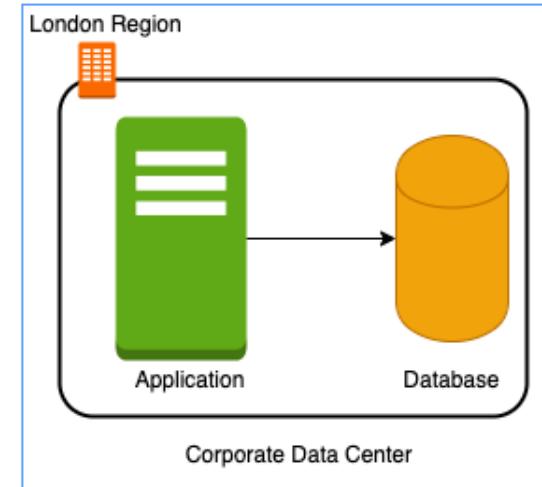
- API Lifecycle Management for RESTful APIs and WebSocket APIs
- Run multiple versions of the same API
- Rate Limits(request quota limits), throttling and fine-grained access permissions using API Keys for Third-Party Developers
- Authorization integration with:
  - AWS IAM (for AWS users using signature version 4)
  - Amazon Cognito
  - Lambda authorizer (custom authorization with JWT tokens or SAML)

# Virtual Private Cloud (VPC) Fundamentals

# Need for Amazon VPC

In 28  
Minutes

- In a corporate network or an on-premises data center:
  - Can anyone on the internet see the data exchange between the application and the database?
    - No
  - Can anyone from internet directly connect to your database?
    - Typically NO.
    - You need to connect to your corporate network and then access your applications or databases.
- Corporate network provides a **secure internal network** protecting your resources, data and communication from external users
- How do you do create **your own private network** in the cloud?
  - Enter **Virtual Private Cloud (VPC)**



# Amazon VPC (Virtual Private Cloud)

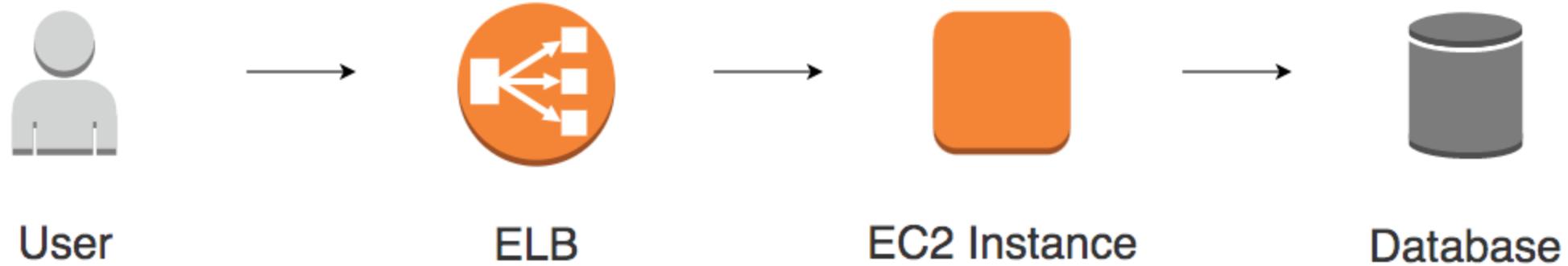
In 28  
Minutes

- Your own **isolated network** in AWS cloud
  - Network traffic within a VPC is isolated (not visible) from all other Amazon VPCs
- You **control all the traffic** coming in and going outside a VPC
- **(Best Practice)** Create all your AWS resources (compute, storage, databases etc) **within a VPC**
  - Secure resources from unauthorized access AND
  - Enable secure communication between your cloud resources



# Need for VPC Subnets

In 28  
Minutes

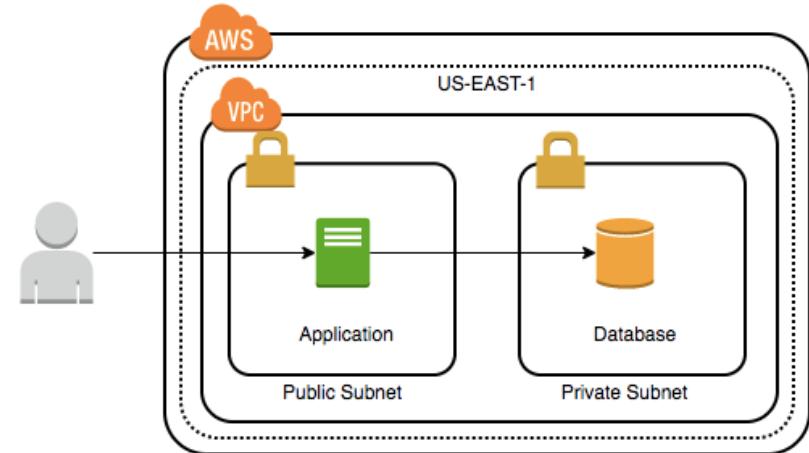


- Different resources are created on cloud - databases, compute (EC2) etc
- Each type of resource has **its own access needs**
- Public Elastic Load Balancers are accessible from internet (**public resources**)
- Databases or EC2 instances should NOT be accessible from internet
  - ONLY applications within your network (VPC) should be able to access them(**private resources**)
- How do you **separate public resources from private resources** inside a VPC?

# VPC Subnets

In 28  
Minutes

- (Solution) Create different subnets for public and private resources
  - Resources in a public subnet **CAN** be accessed from internet
  - Resources in a private subnet **CANNOT** be accessed from internet
  - BUT resources in public subnet can talk to resources in private subnet
- Each VPC is created in a Region
- Each Subnet is created in an Availability Zone
- **Example :** VPC - us-east-1 => Subnets - AZs us-east-1a or us-east-1b or ..



# Addressing for Resources - IP address

In 28  
Minutes

- How do you identify resources on a network ( public (internet) or private(intranet) )?
  - Each resource has an IP address
- There are **two IP address formats:**
  - **IPv4** (Internet Protocol version 4 - numeric 32 bit)
    - Example : 127.255.255.255
  - **IPv6** (Internet Protocol version 6 - alphanumeric 128 bit).
    - Example : 2001:0db8:85a3:0000:0000:8a2e:0370:7334
- IPv4 allows a total of 4.3 billion addresses
- We are running out of the IPv4 address space => IPv6 is introduced as an extension
- While IPv4 and IPv6 are supported on AWS, IPv4 is the most popularly used address format within an AWS VPC.



ELB



EC2 Instance

# CIDR (Classless Inter-Domain Routing) Blocks

In 28  
Minutes

- Typically resources in same network use similar IP address to make routing easy:
  - Example: Resources inside a specific network can use IP addresses from 69.208.0.0 to 69.208.0.15
- How do you express a **range of addresses** that resources in a network can have?
  - CIDR block
- A **CIDR block** consists of a **starting IP address(69.208.0.0)** and a **range(/28)**
  - Example: CIDR block 69.208.0.0/28 represents addresses from 69.208.0.0 to 69.208.0.15 - a total of 16 addresses
- **Quick Tip:** 69.208.0.0/28 indicates that the first 28 bits (out of 32) are fixed.
  - Last 4 bits can change =>  $2^{4} = 16$  addresses

# CIDR Exercises

CIDR	Start Range	End Range	Total addresses	Bits selected in IP address
69.208.0.0/24	69.208.0.0	69.208.0.255	256	01000101.11010000.00000000.*****
69.208.0.0/25	69.208.0.0	69.208.0.127	128	01000101.11010000.00000000.0*****
69.208.0.0/26	69.208.0.0	69.208.0.63	64	01000101.11010000.00000000.00*****
69.208.0.0/27	69.208.0.0	69.208.0.31	32	01000101.11010000.00000000.000****
69.208.0.0/28	69.208.0.0	69.208.0.15	16	01000101.11010000.00000000.0000***
69.208.0.0/29	69.208.0.0	69.208.0.7	8	01000101.11010000.00000000.00000**
69.208.0.0/30	69.208.0.0	69.208.0.3	4	01000101.11010000.00000000.000000*
69.208.0.0/31	69.208.0.0	69.208.0.1	2	01000101.11010000.00000000.000000*
69.208.0.0/32	69.208.0.0	69.208.0.0	1	01000101.11010000.00000000.0000000

- Exercise : How many addresses does **69.208.0.0/26** represent?
  - $2 \text{ to the power } (32-26 = 6) = 64$  addresses from 69.208.0.0 to 69.208.0.63
- Exercise : How many addresses does **69.208.0.0/30** represent?
  - $2 \text{ to the power } (32-30 = 2) = 4$  addresses from 69.208.0.0 to 69.208.0.3
- Exercise : What is the difference between **0.0.0.0/0** and **0.0.0.0/32**?
  - 0.0.0.0/0 represent all IP addresses. 0.0.0.0/32 represents just one IP address 0.0.0.0.

# CIDR Block Example - Security Group

In 28  
Minutes

Direction	Protocol	Port Range	Source/Destination
Inbound	TCP	443	172.31.0.0/16
Inbound	TCP	22	183.82.143.132/32
Outbound	All	All	0.0.0.0/0

- Allows HTTPS (TCP - 443) requests from a range of addresses (172.31.0.0/16)
- Allows SSH (TCP - 22) from a single IP address (183.82.143.132/32)
- Allows all outbound communication
- All other inbound/outbound traffic is denied
- CIDR Demo : Security Groups

# VPC CIDR Blocks

In 28  
Minutes

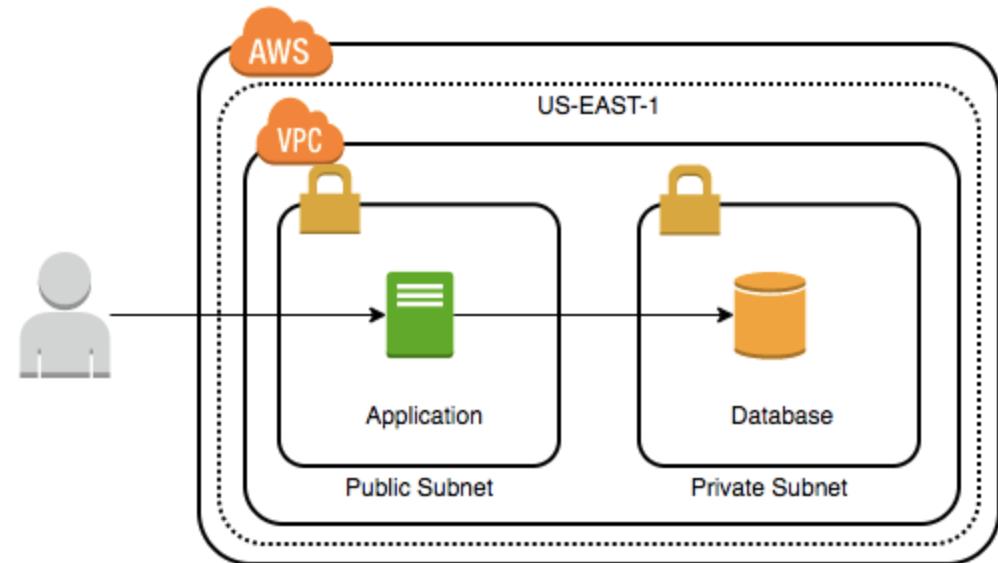
CIDR	Start Range	End Range	Total addresses	Bits selected in IP address
69.208.0.0/24	69.208.0.0	69.208.0.255	256	01000101.11010000.00000000.*****
69.208.0.0/25	69.208.0.0	69.208.0.127	128	01000101.11010000.00000000.0*****
69.208.0.0/26	69.208.0.0	69.208.0.63	64	01000101.11010000.00000000.00*****
69.208.0.0/27	69.208.0.0	69.208.0.31	32	01000101.11010000.00000000.000****
69.208.0.0/28	69.208.0.0	69.208.0.15	16	01000101.11010000.00000000.0000***
69.208.0.0/29	69.208.0.0	69.208.0.7	8	01000101.11010000.00000000.00000**
69.208.0.0/30	69.208.0.0	69.208.0.3	4	01000101.11010000.00000000.000000*
69.208.0.0/31	69.208.0.0	69.208.0.1	2	01000101.11010000.00000000.000000*
69.208.0.0/32	69.208.0.0	69.208.0.0	1	01000101.11010000.00000000.0000000

- Each VPC is associated with a **CIDR Block**
- CIDR block of VPC can be from /16 (65536 IP addresses) to /28 (16 IP addresses)
- **Example 1 :** VPC with CIDR block 69.208.0.0/24 - 69.208.0.0 to 69.208.0.255
- **Example 2 :** VPC with CIDR block 69.208.0.0/16 - 69.208.0.0 to 69.208.255.255

# Choosing a CIDR Block for VPC

In 28  
Minutes

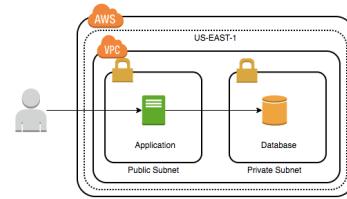
- Be careful in choosing a CIDR block. **Choose a wider range than you would need.**
- There **CANNOT** be an overlap of a VPC CIDR block with any other connected network
- All addresses inside a VPC CIDR range are **private addresses**:
  - Cannot route to private addresses from internet
  - Assign and use public IP addresses to communicate with VPC resources from internet



# Choosing a CIDR Block for a Subnet

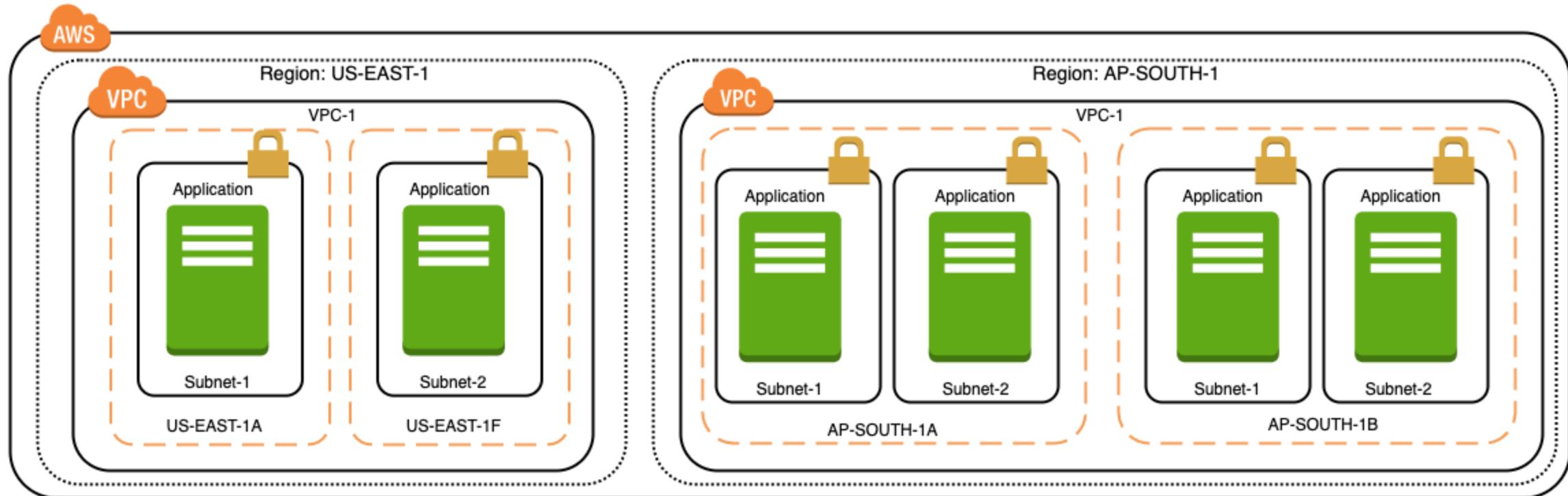
In 28  
Minutes

- Subnet provides a **grouping** for resources inside a VPC
- The CIDR block of a subnet **must be a subset or the same** as the CIDR block for the VPC
- **Minimum** subnet range is /28 (16 addresses)
- In each subnet, 5 IP address (first four and the last) are **reserved** by AWS
- Every new AWS account has a default VPC (/16) in every region with a public subnet(/20) in every AZ
- (Remember) Address range of a VPC CAN be extended (Add new CIDR Block)
- (Remember) Address range of a Subnet CANNOT be changed.



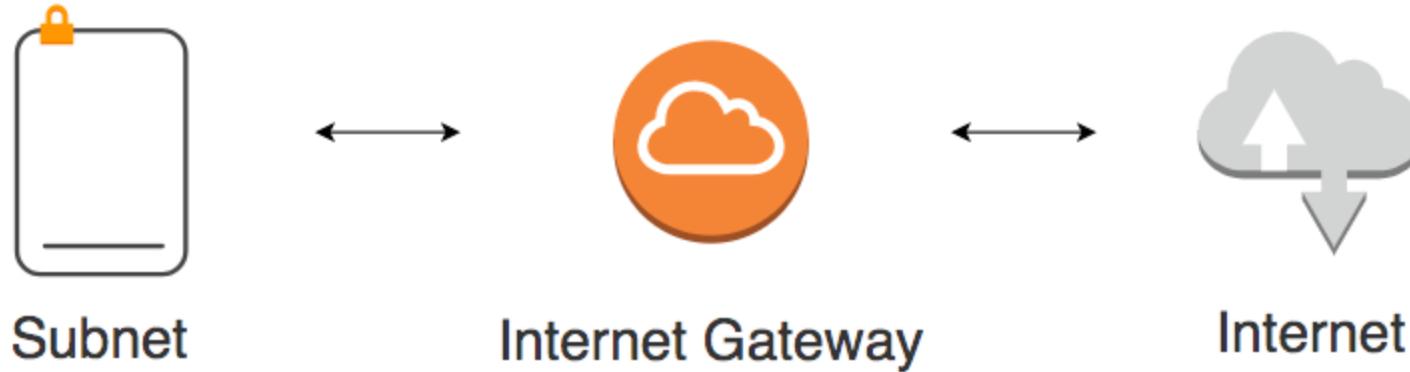
# VPC and Subnets Demo

In 28  
Minutes



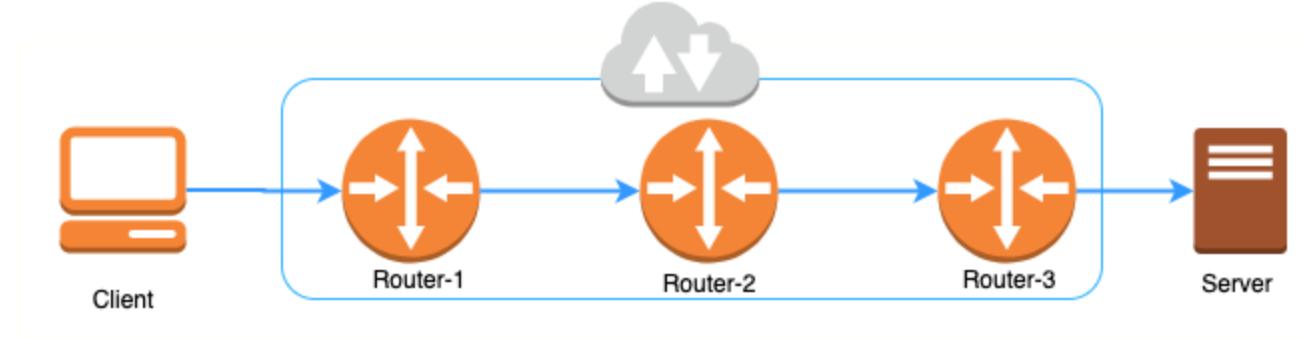
# Public Subnet

In 28  
Minutes



- Communication allowed from subnet to internet
- Communication allowed from internet to subnet

# Routing on the internet



- You have an IP address of a website you want to visit
- There is **no direct connection** from your computer to the website
- Internet is actually a **set of routers** routing traffic
- Each router has a set of rules that help it decide the path to the destination IP address

# Routing inside AWS

In 28  
Minutes

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	igw-1234567

- In AWS, **route tables** are used for routing
- Route tables can be associated with VPCs and subnets
- Each route table consists of a **set of rules** called routes
  - Each route or routing rule has a **destination** and **target**
  - What CIDR blocks (range of addresses) should be routed to which target resource?
- **Rule 1** - Route requests to VPC CIDR 172.31.0.0/16 (172.31.0.0 to 172.31.255.255) to local resources within the VPC
- **Rule 2** - Route all other IP addresses (0.0.0.0/0) to internet (internet gateway)

# Execution of Route Table

In 28  
Minutes

Destination	Target
172.31.0.0/16	Local
0.0.0.0/0	igw-1234567

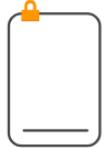
- What happens if I search for an address **172.31.0.10**?
  - Two destinations match - 172.31.0.0/16 (172.31.0.0 to 172.31.255.255) and 0.0.0.0/0
  - The most specific rule wins. 172.31.0.0/16 is more specific
  - Result : Routing to a local resource
- What happens if I search for an address **69.209.0.10**?
  - One destination match - 69.208.0.10
  - Result : Routing to internet
- The **most specific matching route wins**

# Public Subnet vs Private Subnet

In 28  
Minutes

Name	Destination	Target	Explanation
RULE 1	172.31.0.0/16	Local	Local routing
RULE 2	0.0.0.0/0	igw-1234567	Internet routing

- An **Internet Gateway** enables internet communication for subnets
- Any subnet which has a route to an internet gateway is called a **public subnet**
- Any subnet which **DOES NOT** have route to an internet gateway is called a **private subnet**



Subnet



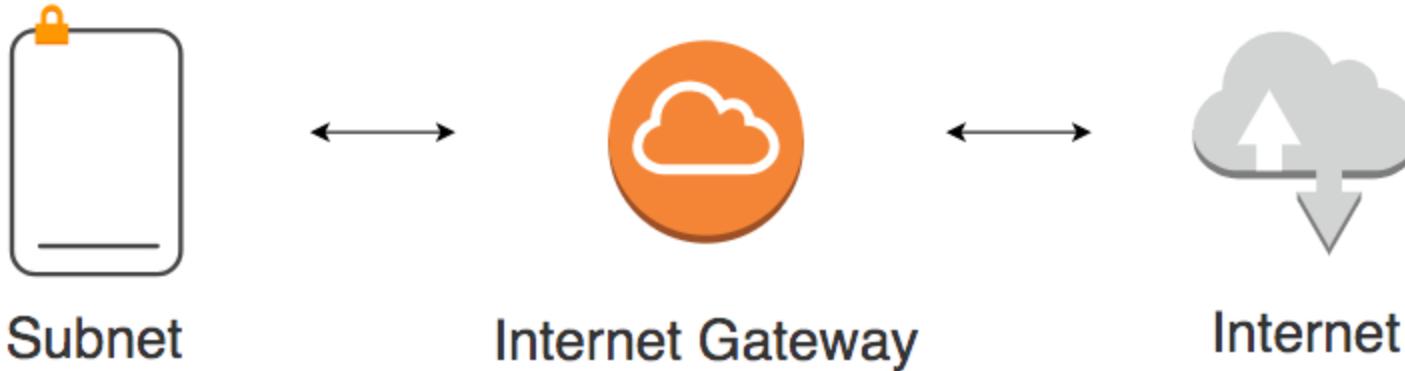
Internet Gateway



Internet

# More about Internet Gateway

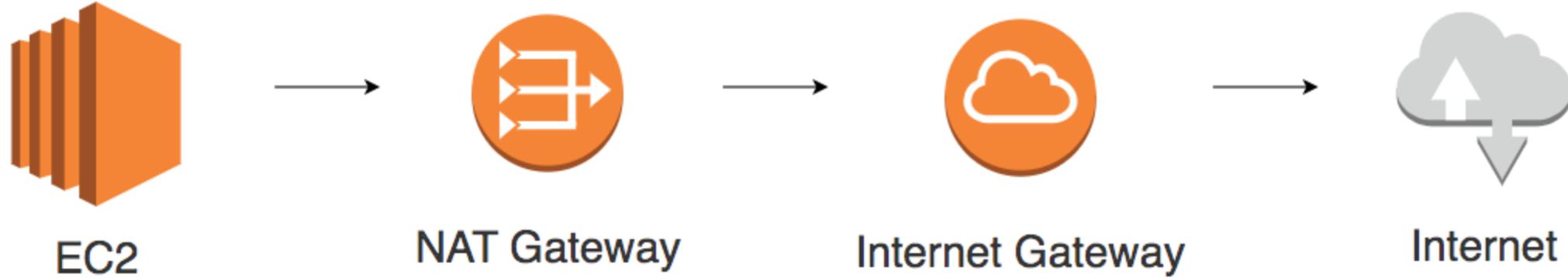
In 28  
Minutes



- Sits between subnet (VPC) resources and internet
- **One to one mapping** with a VPC
- Supports IPv4 and IPv6
- Translate private IP address to public IP address and vice-versa
- **DHCP option set**: Assign custom host names and IP addresses to EC2 instances

# Private Subnet - Download Patches

In 28  
Minutes



- Cannot be accessed from internet.
- Might allow traffic to internet using a NAT Device.

# Network Address Translation(NAT) Instance and Gateway

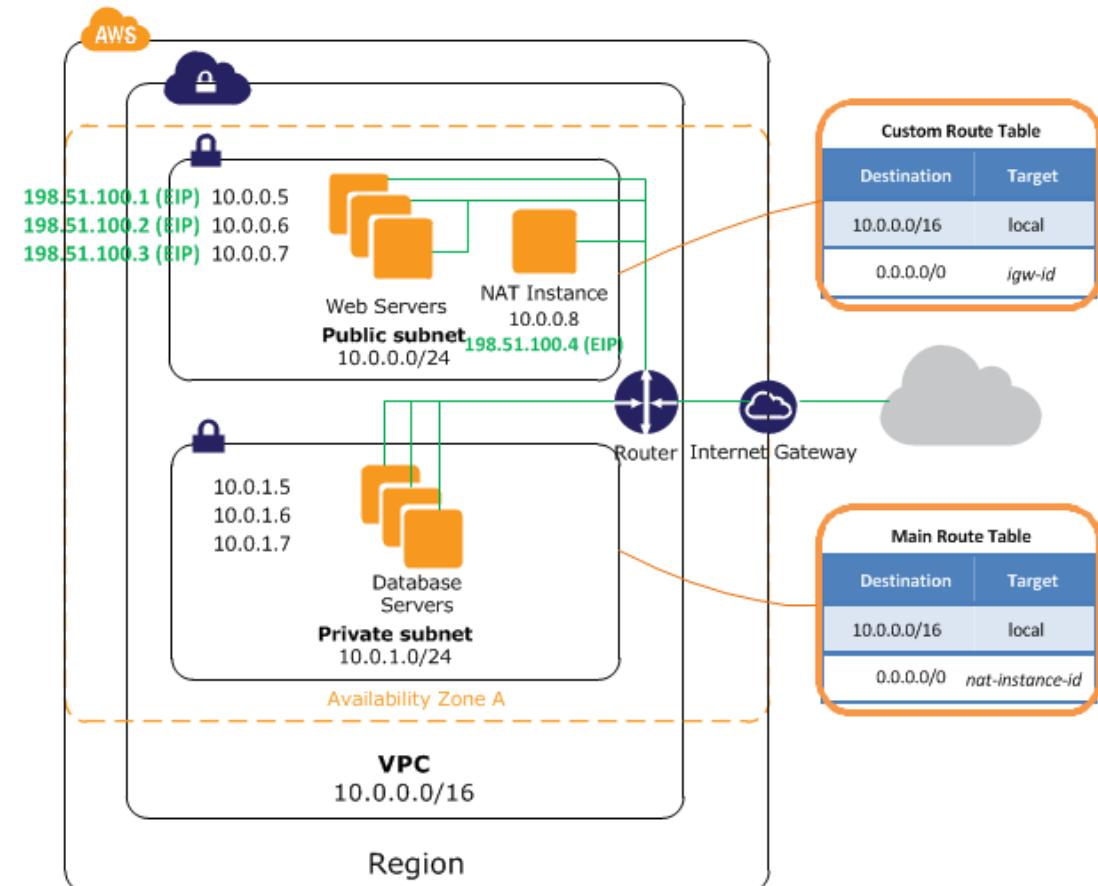
in 28 Minutes

- How do you allow instances in a private subnet to download software updates and security patches while denying inbound traffic from internet?
- How do you allow instances in a private subnet to connect privately to other AWS Services outside the VPC?
- Three Options:
  - NAT Instance: Install a EC2 instance with specific NAT AMI and configure as a gateway
  - NAT Gateway: Managed Service
  - Egress-Only Internet Gateways: For IPv6 subnets

# NAT instance

In 28  
Minutes

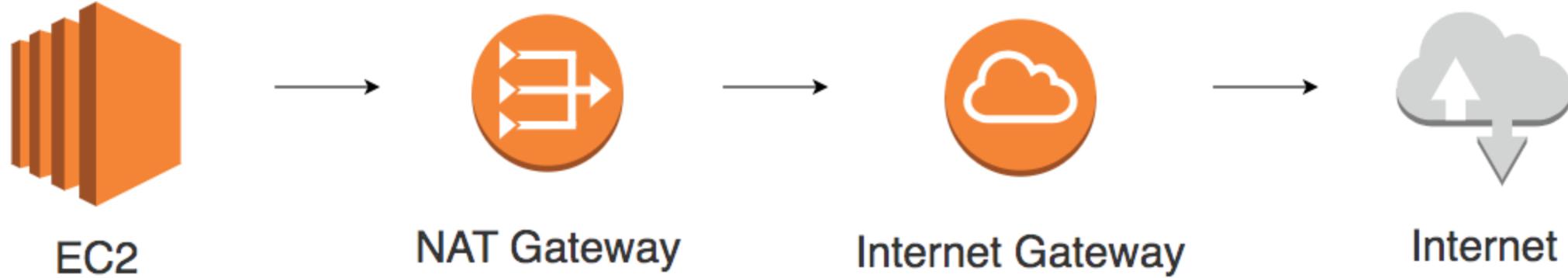
- **Step 1: Create EC2 instance**
  - AMI - Linux \*amzn-ami-vpc-nat
  - Public subnet with public IP address or Elastic IP
- **Step 2: Assign Security Group**
  - Inbound - HTTP(80) HTTPS(443) from private subnet
  - Outbound - HTTP(80) & HTTPS(443) to internet (0.0.0.0/0)
- **Step 3: Private Subnet Route Table**
  - Redirect all outbound traffic (0.0.0.0/0) to the NAT instance



[https://docs.aws.amazon.com/vpc/latest/userguide/VPC\\_NAT\\_Instance.html](https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html)

# NAT gateway

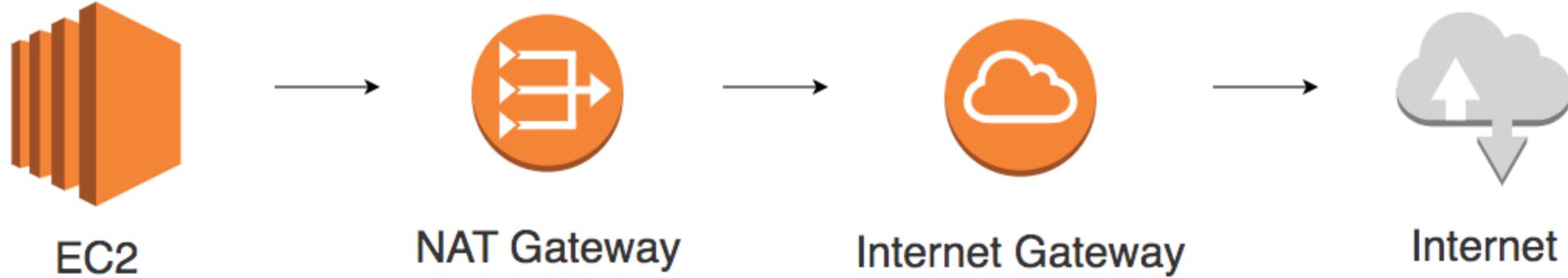
In 28  
Minutes



- AWS Managed Service
- Step 1: Get an **Elastic IP Address**
- Step 2: Create NAT gateway in a **PUBLIC subnet** with the Elastic IP Address.
- Step 3: Private subnet route - **all outbound traffic (0.0.0.0/0)** to NAT gateway.
- DEMO

# NAT gateway - Remember

In 28  
Minutes



- Prefer **NAT gateway over NAT instance**
  - Less administration, more availability and higher bandwidth
  - NAT Gateway does not need any security group management.
- NAT Gateway supports **IPv4 ONLY**.
  - Use Egress-Only Internet Gateways for IPv6.
- NAT Gateway uses the Internet Gateway.

# NAT gateway vs NAT instance

In 28  
Minutes

Feature	NAT gateway	NAT instance
Managed by	AWS	You
Created in	Public Subnet	Public Subnet
Internet Gateway	Needed	Needed
High Availability	Yes (in an AZ) Multi AZ (higher availability)	You are responsible.
Bandwidth	Up to 45 Gbps	Depends on EC2 instance type
Public IP addresses	Elastic IP address	Elastic IP address OR Public IP Address
Disable source destination check	No	Required
Security groups	No specific configuration needed	Needed on NAT instance
Bastion servers	No	Can be used as a Bastion server

# VPC and Subnets - Questions

In 28  
Minutes

Question	Answer
Can I have a VPC spread over two regions?	No
Can I have multiple VPCs in same region?	Yes
Is communication between two resources in a VPC visible outside VPC?	No
Can you allow external access to your resources in a VPC?	Yes
Can I have a subnet spread over two regions?	No
Can I have a subnet spread over two availability zones?	No
Can I have two subnets in one availability zone?	Yes
Can I have a subnet in availability zone ap-south-1a if it's VPC is in the region us-east-1?	No. Subnet should be in AZs belonging to the VPC's region

# VPC Main Route Table

In 28  
Minutes

Destination	Target
172.31.0.0/16	Local

- Each VPC has a **main route table**, by default
- **Main route table** has a default route enabling communication between resources in all subnets in a VPC
- Default route rule CANNOT be deleted/edited
- HOWEVER you can add/edit/delete other routing rules to the main route table



VPC

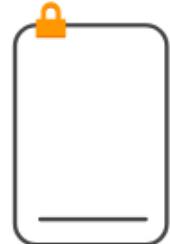


VPC Router

# Subnet Route Tables

In 28  
Minutes

- Each subnet can have its **own** route table OR **share** its route table with the VPC
- If a subnet does not have a route table associated with it, it **implicitly** uses the route table of its VPC
- Multiple subnets can share a route table
- **HOWEVER** at any point in time, a subnet can be associated with one route table **ONLY**



Subnet



VPC Router

# Quick Review of Security Groups - Default Security Group<sup>in 28 Minutes</sup>

Direction	Protocol	Port Range	Source/Destination
Inbound	All	All	Security Group ID (sg-xyz)
Outbound	All	All	0.0.0.0/0

- **Default security group** is created when you create a VPC
  - Allows all outbound traffic
  - Allows communication between resources assigned with the default security group
  - Denies all other inbound traffic (other than resources with the default security group)
  - Can be edited but not be deleted
- EC2 instances, by default, are assigned the default security group of the VPC
  - However, you can change it at any point - during launch or later
- Security Group has a **many to many relationship** with Resources (in same VPC)

# New Security Groups

In 28  
Minutes

Direction	Protocol	Port Range	Source/Destination
Outbound	All	All	0.0.0.0/0

- You can create **new security groups**
- By default:
  - There are no inbound rules
  - Denies all inbound traffic
  - Allows all outbound traffic
- You can add, edit and delete outbound and inbound rules

# Security Groups - Important Ports

Service	Port
SSH (Linux instances)	22
RDP (Remote Desktop - Windows)	3389
HTTP	80
HTTPS	443
PostgreSQL	5432
Oracle	1521
MySQL/Aurora	3306
MSSQL	1433

# Security Group Scenario Questions

In 28  
Minutes

## Scenario

Can source/destination of a security group be another security group?

A new security group is created and assigned to a database and an ec2 instance. Can these instances talk to each other?

The default security group (unchanged) in the VPC is assigned to a database and an ec2 instance. Can these instances talk to each other?

## Solution

Yes. It can even be same security group.

No. New security group does not allow any incoming traffic from same security group. Two resources associated with same security group can talk with each other only if you configure rules allowing the traffic.

Yes. Default security group (by default) has a rule allowing traffic between resources with same security group.

# Network Access Control List

In 28  
Minutes

- Security groups control traffic to a specific resource in a subnet.
- How about stopping traffic from **even entering the subnet?**
- NACL provides **stateless firewall** at subnet level.
- Each subnet **must** be associated with a NACL.
- **Default NACL** allows all inbound and outbound traffic.
- **Custom created NACL** denies all inbound and outbound traffic by default.
- Rules have a priority number.
  - Lower number => Higher priority.
- Hands-on



# Security Group vs NACL

In 28  
Minutes



Feature	Security Group	NACL
Level	Assigned to a specific instance(s)/resource(s)	Configured for a subnet. Applies to traffic to all instances in a subnet.
Rules	Allow rules only	Both allow and deny rules
State	Stateful. Return traffic is automatically allowed.	Stateless. You should explicitly allow return traffic.
Evaluation	Traffic allowed if there is a matching rule	Rules are prioritized. Matching rule with highest priority wins.

# Scenario: EC2 instance cannot be accessed from internet

In 28  
Minutes



NACL



Subnet



Security Group



EC2



ElasticIP

- Does the EC2 instance have a public IP address or an Elastic IP address assigned?
- Check the network access control list (ACL) for the subnet. Is inbound and outbound traffic allowed from your IP address to the port?
- Check the route table for the subnet. Is there a route to the internet gateway?
- Check your security group rules. Are you allowing inbound traffic from your IP address to the port?

# Amazon S3 Fundamentals

# Amazon S3 (Simple Storage Service)

In 28  
Minutes

- Most popular, very flexible & inexpensive storage service
- Store large objects using a **key-value** approach
- Also called **Object Storage**
- Provides REST API to access and modify objects
- Provides **unlimited storage**:
  - (S3 storage class) **99.99% availability** & (**11 9's - 99.999999999**) durability
  - Objects are replicated in a single region (across multiple AZs)
- **Store all file types** - text, binary, backup & archives:
  - Media files and archives
  - Application packages and logs
  - Backups of your databases or storage devices
  - Staging data during on-premise to cloud database migration



Amazon S3

# Amazon S3 Demo

In 28  
Minutes

- DEMO

# Amazon S3 - Objects and Buckets

In 28  
Minutes

- Amazon S3 is a **global service**. NOT associated with a region.
  - HOWEVER a bucket is created in a specific AWS region
- Objects are stored in buckets
  - Bucket names are **globally unique**
  - Bucket names are used as part of object URLs => Can contain ONLY lower case letters, numbers, hyphens and periods.
  - Unlimited objects in a bucket
- Each object is identified by a **key value pair**
  - Key is **unique** in a bucket
  - Max object size is **5 TB**
- (Remember) **No hierarchy** of buckets, sub-buckets or folders

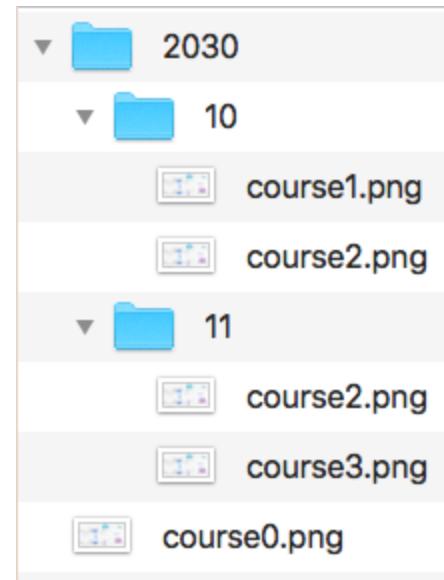


Amazon S3

# Amazon S3 Key Value Example

In 28  
Minutes

Key	Value
2030/course0.png	image-binary-content
2030/10/course1.png	image-binary-content
2030/10/course2.png	image-binary-content
2030/11/course2.png	image-binary-content
2030/11/course3.png	image-binary-content



# Amazon S3 Versioning

In 28  
Minutes

- Protects against **accidental deletion**
- Versioning is **optional** and is enabled at bucket level
- You can turn on versioning on a non versioned bucket
  - All old objects will have a version of null
- You cannot turn off versioning on a versioned bucket
  - You can only **suspend** versioning



S3 Bucket

# Amazon S3 Static Website Hosting

In 28  
Minutes

- Use S3 to host a static website using a bucket
- Step 1 : Upload website content
- Step 2 : Enable **Static website hosting**
- Step 3 : Disable "Block public access"
- Step 4 : Configure "Bucket policy" to enable public read access

# Resource-based policies - Bucket policies

In 28  
Minutes

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Sid": "PublicRead",  
            "Effect": "Allow",  
            "Principal": "*",  
            "Action": ["s3:GetObject"],  
            "Resource": ["arn:aws:s3:::mybucket/*"]  
        }  
    ]  
}
```

- Control access to your bucket and objects
- Can grant **cross account** and **public** access

# Amazon S3 - Tags

In 28  
Minutes

- Tags can be assigned to most AWS resources
- Can be used for **automation, security (policies), cost tracking** etc
- **Key-value pairs** applied to S3 objects:
  - Environment=Dev
  - Classification=Secure
  - Project=A
- Can be used in creating **lifecycle policies**
- Can be updated continuously during the lifetime of an object



S3 Bucket

# Amazon S3 Event Notifications

In 28  
Minutes

- Configure notifications when certain events happen in your bucket
- **Event Sources**
  - New object created events
  - Object removal events
  - Reduced Redundancy Storage (RRS) object lost events
  - Replication events
- **Event Destinations**
  - Amazon SNS topic
  - Amazon SQS queue
  - AWS Lambda function



S3 Bucket

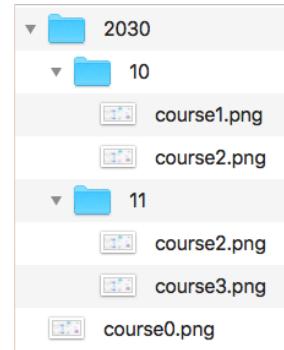


Lambda Fn

# Amazon S3 - Prefix

In 28  
Minutes

- Allows you to search for keys **starting with a certain prefix**
- Searching with prefix 2030/10 returns
  - 2030/10/course1.png & 2030/10/course2.png
- URL - ***http://s3.amazonaws.com/my-bucket-ranga?prefix=2030/10/***
  - Above URL would work only when public access is allowed
- Supported by REST API, AWS SDK, AWS CLI and AWS Management Console
- Used in IAM and Bucket Policies to restrict access to specific files or group of files



# Bucket ACLs and Object ACLs

In 28  
Minutes

- Bucket/Object ACLs
  - Access for bucket/object owner
  - Access for other AWS accounts
  - Public access
- Use **object ACLs (object level access)**
  - When bucket owner is not the object owner
  - When you need different permissions for different objects in the same bucket
- (Remember) Bucket/Object ACLs
  - CANNOT have conditions while policies can have conditions
  - CANNOT explicitly DENY access
  - CANNOT grant permissions to other individual users
- (Remember) ACLs are **primarily** used to grant permissions to public or other AWS accounts

# Amazon S3 Storage Classes - Introduction

In 28  
Minutes

- Different kinds of data can be stored in Amazon S3
  - Media files and archives
  - Application packages and logs
  - Backups of your databases or storage devices
  - Long term archives
- Huge variations in access patterns
- Trade-off between access time and cost
- S3 storage classes help to optimize your costs while meeting access time needs
  - Designed for durability of 99.999999999%(11 9's)



Amazon S3



Amazon Glacier

# Amazon S3 Storage Classes

In 28  
Minutes

Storage Class	Scenario	AZs
Standard	Frequently accessed data	>=3
Standard-IA	Long-lived, infrequently accessed data (backups for disaster recovery)	>=3
One Zone-IA	Long-lived, infrequently accessed, non-critical data (Easily re-creatable data - thumbnails for images)	1
Intelligent-Tiering	Long-lived data with changing or unknown access patterns	>=3
Glacier	Archive data with retrieval times ranging from minutes to hours	>=3
Glacier Deep Archive	Archive data that rarely, if ever, needs to be accessed with retrieval times in hours	>=3
Reduced Redundancy (Not recommended)	Frequently accessed, non-critical data	>=3

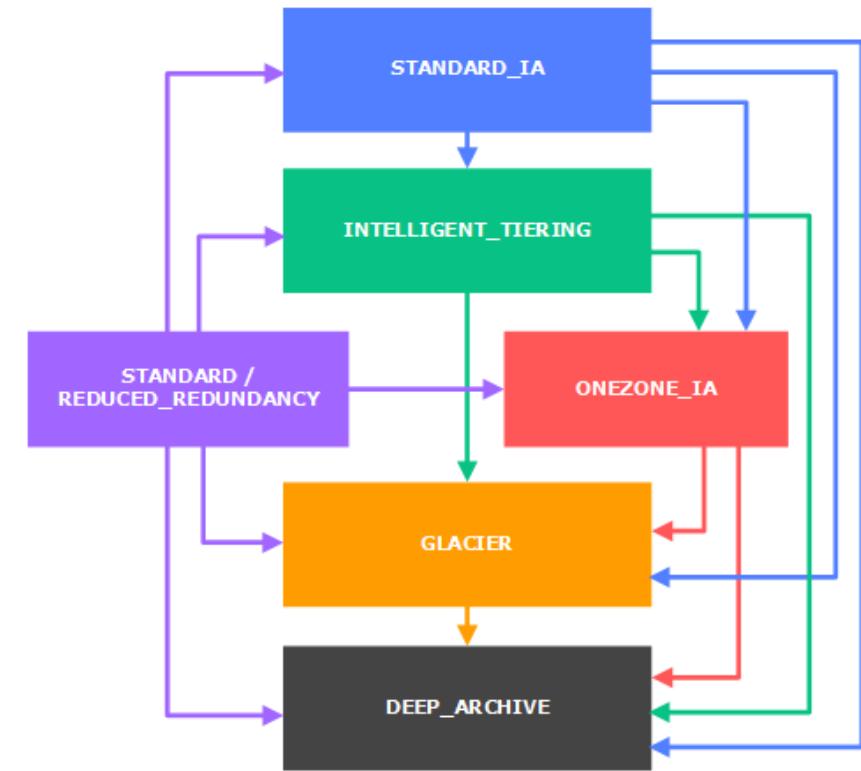
# Amazon S3 Storage Classes - Comparison

In 28  
Minutes

Feature	Standard	Intelligent Tiering	Standard IA	One Zone IA	Glacier	Glacier Deep Archive
Availability (Designed)	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
Availability (SLA)	99.9%	99%	99%	99%	99.9%	99.9%
Replication AZs	>=3	>=3	>=3	1	>=3	>=3
First byte: ms (milliseconds)	ms	ms	ms	ms	minutes or hours	few hours
Min object size (for billing)	NA	NA	128KB	128KB	40KB	40KB
Min storage days (for billing)	NA	30	30	30	90	180
Per GB Cost (varies)	\$0.025	varies	\$0.018	\$0.0144	\$0.005	\$0.002
Encryption	Optional	Optional	Optional	Optional	Mandatory	Mandatory

# S3 Lifecycle configuration

- Files are frequently accessed when they are created
- Generally usage reduces with time
- How do you save costs and move files automatically between storage classes?
  - Solution: S3 Lifecycle configuration
- Two kinds of actions:
  - transition actions (one storage class to another)
  - expiration actions (delete objects)
- Object can be identified by tags or prefix.



<https://docs.aws.amazon.com/AmazonS3/latest/transition-general-considerations.html>

*transition-general-considerations.html*

# Amazon S3 Replication - Same Region and Multiple Regions

28  
Minutes

- Replicate objects between buckets in same or different regions
  - Could be cross account
  - Can be configured at bucket level, a shared prefix level, or an object level using S3 object tags
  - Access to destination bucket is provided using IAM Policy
- Versioning should be enabled on BOTH source and destination
- ONLY new objects are replicated (Explicitly copy existing objects)
- (Advantage) Reduces latency and helps you meet regulations
- (USECASE) Object replication between dev & test environments



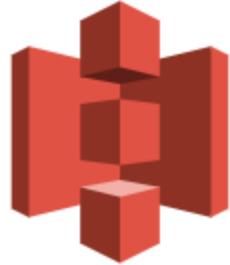
S3 Bucket



S3 Bucket

# Amazon S3 - Object level configuration

In 28  
Minutes



Amazon S3



S3 Bucket

- You can configure these at **individual object level** (overriding bucket level configuration):
  - Storage class
  - Encryption
  - Objects ACLs

# Amazon S3 Consistency

In 28  
Minutes

- S3 is distributed - maintains **multiple copies** of your data in a Region to ensure durability
- Distributing data **presents a challenge**
  - How do you ensure data is consistent?
- **S3 Consistency Model**
  - READ AFTER WRITE for PUTS of new objects
  - Eventual Consistency for Overwrites PUTS and DELETES
- (In simplified words) S3 Data is highly distributed across multiple AZs and (possibly) multiple regions:
  - When you create a new object, it is immediately available
  - You might get a previous version of data immediately after an object update using PUT/DELETE
  - You will never get partial or inconsistent data



Amazon S3

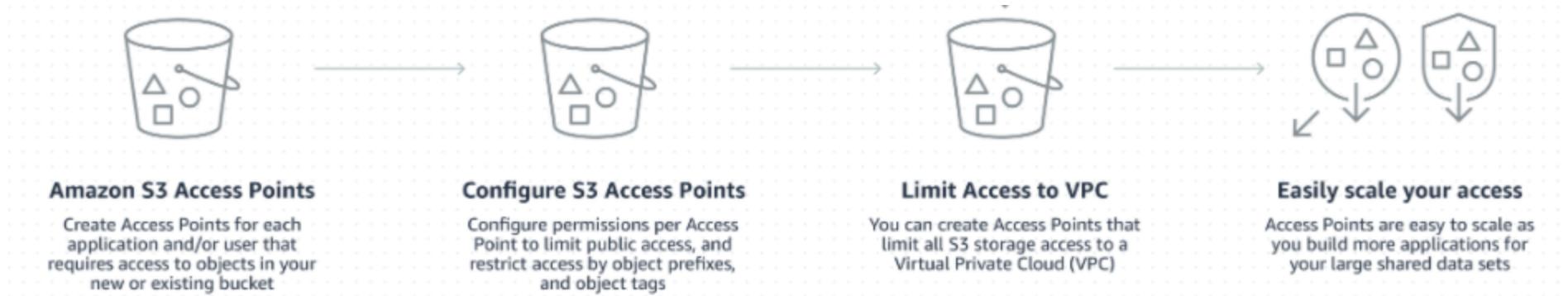
# Amazon S3 Presigned URL

In 28  
Minutes

- Grant **time-limited permission** (few hours to 7 days) to download objects
- **Avoid** web site scraping and unintended access
- Specify:
  - Your security credentials
  - Bucket name
  - Object key
  - HTTP method and
  - Expiration date and time
- Created using AWS SDK API
  - Java code
    - *GeneratePresignedUrlRequest(bucketName, objectKey).withMethod(HttpMethod.GET).withExpiration(expiration);*

# Amazon S3 Access Points

In 28  
Minutes



- Simplifies bucket policy configuration
- Create application specific access points with an application specific policy
- Provide multiple customized paths with unique hostname and access policy for each bucket
- “dual-stack” endpoint supports IPv4 and IPv6 access

# Amazon S3 Scenarios - Security

In 28  
Minutes

Scenario	Solution
Prevent objects from being deleted or overwritten for a few days or forever	Use Amazon S3 Object Lock. Can be enabled only on new buckets. Automatically enables versioning. Prevents deletion of objects. Allows you to meet regulatory requirements.
Protect against accidental deletion	Use Versioning
Protect from changing versioning state of a bucket	Use MFA Delete. You need to be an owner of the bucket AND Versioning should be enabled.
Avoid content scraping. Provide secure access.	Pre Signed URLs. Also called Query String Authentication.
Enable cross domain requests to S3 hosted website (from www.abc.com to www.xyz.com)	Use Cross-origin resource sharing (CORS)

# Amazon S3 Cost

In 28  
Minutes

- Important pricing elements:
  - Cost of Storage (per GB)
  - (If Applicable) Retrieval Charge (per GB)
  - Monthly tiering fee (Only for Intelligent Tiering)
  - Data transfer fee
- **FREE of cost:**
  - Data transfer into Amazon S3
  - Data transfer from Amazon S3 to Amazon CloudFront
  - Data transfer from Amazon S3 to services in the same region



Amazon S3

# Amazon S3 Scenarios - Costs

In 28  
Minutes

Scenario	Solution
Reduce Costs	Use proper storage classes. Configure lifecycle management.
Analyze storage access patterns and decide the right storage class for my data	Use Intelligent Tiering. Use Storage Class Analysis reports to get an analysis.
Move data automatically between storage classes	Use Lifecycle Rules
Remove objects from buckets after a specified time period	Use Lifecycle Rules and configure Expiration policy

# Amazon S3 Performance

In 28  
Minutes

- Amazon S3 is serverless
- Recommended for large objects
- Amazon S3 supports upto
  - 3,500 requests per second to add data
  - 5,500 requests per second to retrieve data
  - Zero additional cost
  - With each S3 prefix
- **Transfer acceleration**
  - Enable fast, easy and secure transfers of files to and from your bucket



Amazon S3

# Amazon S3 Scenarios - Performance

In 28  
Minutes

Scenario	Solution
Improve S3 bucket performance	Use <b>Prefixes</b> . Supports upto 3,500 RPS to add data and 5,500 RPS to retrieve data with each S3 prefix.
Upload large objects to S3	Use <b>Multipart Upload API</b> . <b>Advantages:</b> 1. Quick recovery from any network issues 2. Pause and resume object uploads 3. Begin an upload before you know the final object size. <b>Recommended</b> for files >100 MB and <b>mandatory</b> for files >4 GB
Get part of the object	Use <b>Byte-Range Fetches</b> - Range HTTP header in GET Object request <b>Recommended:</b> GET them in the same part sizes used in multipart upload
Is this recommended: EC2 (Region A) <-> S3 bucket (Region B)	No. <b>Same region recommended</b> . Reduce network latency and data transfer costs

# Amazon S3 Scenarios - Features

In 28  
Minutes

Scenario	Solution
Make user pay for S3 storage	Requester pays - The requester (instead of the bucket owner) will pay for requests and data transfer.
Create an inventory of objects in S3 bucket	Use S3 inventory report
I want to change object metadata or manage tags or ACL or invoke Lambda function for billions of objects stored in a single S3 bucket	Generate S3 inventory report Perform S3 Batch Operations using the report
Need S3 Bucket (or Object) Access Logs	Enable S3 Server Access Logs (default: off). Configure the bucket to use and a prefix (logs/).

# S3 Glacier

# Amazon S3 Glacier

In 28  
Minutes

- In addition to existing as a S3 Storage Class, S3 Glacier is a separate AWS Service on its own!
- **Extremely low cost storage** for archives and long-term backups:
  - Old media content
  - Archives to meet regulatory requirements (old patient records etc)
  - As a replacement for magnetic tapes
- High durability (11 9s - 99.99999999%)
- High scalability (unlimited storage)
- High security (**encrypted** at rest and in transfer)
- Cannot upload objects to Glacier using Management Console
  - Use REST API, AWS CLI, AWS SDK



Amazon Glacier

# Amazon S3 vs S3 Glacier

In 28  
Minutes

Feature	Amazon S3	S3 Glacier
Terminology	Objects (files) are stored in Buckets (containers)	Archives (files) are stored in Vaults (containers)
Keys	Objects keys are user defined	Archive keys are system generated identifiers
Mutability	(Default) Allows uploading new content to object	After an archive is created, it cannot be updated (Perfect for regulatory compliance)
Max size	Each object can be upto 5TB	Each archive can be upto 40TB
Management Console	Almost all bucket and object operations supported	Only vault operations are supported. You cannot upload/delete/update archives.
Encryption	Optional	Mandatory using AWS managed keys and AES-256. You can use client side encryption on top of this.
WORM Write Once Read Many Times	Enable Object Lock Policy	Enable Vault lock policy

# Retrieving archives from S3 Glacier

In 28  
Minutes

- Asynchronous two step process (Use REST API, AWS CLI or SDK)
  - Initiate a archive retrieval
  - (After archive is available) Download the archive
- Reduce costs by **optionally specify a range, or portion, of the archive to retrieve**
- Reduce costs by **requesting longer access times**
  - Amazon S3 Glacier:
    - Expedited (1 – 5 minutes)
    - Standard (3 – 5 hours)
    - Bulk retrievals (5–12 hours)
  - Amazon S3 Glacier Deep Archive:
    - Standard retrieval (upto 12 hours)
    - Bulk retrieval (upto 48 hours)



Amazon Glacier

# IAM - Fundamentals

# Typical identity management in the cloud

In 28  
Minutes

- You have **resources** in the cloud (examples - a virtual server, a database etc)
- You have **identities (human and non-human)** that need to access those resources and perform actions
  - For example: launch (stop, start or terminate) a virtual server
- How do you **identify users** in the cloud?
- How do you configure resources they can access?
- How can you configure what actions to allow?
- In AWS, *Identity and Access Management (IAM)* provides this service



# AWS Identity and Access Management (IAM)

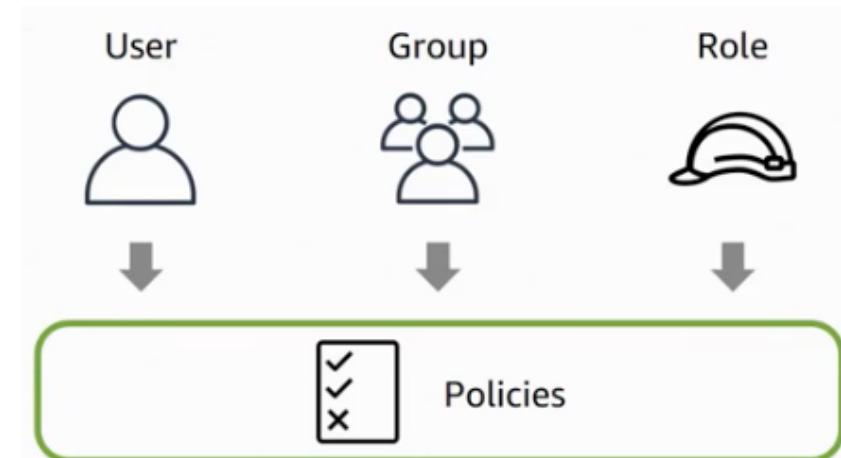
In 28  
Minutes

- **Authentication** (is it the right user?) and
- **Authorization** (do they have the right access?)
- **Identities** can be
  - AWS users or
  - Federated users (externally authenticated users)
- Provides very **granular** control
  - Limit a single user:
    - to perform single action
    - on a specific AWS resource
    - from a specific IP address
    - during a specific time window



# Important IAM Concepts

- **IAM users:** Users created in an AWS account
  - Has credentials attached (name/password or access keys)
- **IAM groups:** Collection of IAM users
- **Roles:** Temporary identities
  - Does NOT have credentials attached
  - (Advantage) Expire after a set period of time
- **Policies:** Define permissions
  - **AWS managed policies** - Standalone policy predefined by AWS
  - **Customer managed policies** - Standalone policy created by you
  - **Inline policies** - Directly embedded into a user, group or role



# AWS IAM Policies - Authorization

In 28  
Minutes

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": "*",  
            "Resource": "*"  
        }  
    ]  
}
```

- Policy is a JSON document with one or more permissions
  - Effect - Allow or Deny
  - Resource - Which resource are you providing access to?
  - Action - What actions are allowed on the resource?
  - Condition - Are there any restrictions on IP address ranges or time intervals?
- Example above: AWS Managed Policy : AdministratorAccess

# AWS Managed Policy : AmazonS3ReadOnlyAccess

In 28  
Minutes

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:Get*",  
                "s3>List*"  
            ],  
            "Resource": "*"  
        }  
    ]  
}
```

# Customer Managed Policy : ReadSpecificS3Bucket

In 28  
Minutes

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Action": [  
                "s3:Get*",  
                "s3>List*"  
            ],  
            "Resource": "arn:aws:s3:::mybucket/somefolder/*"  
        }  
    ]  
}
```

# IAM Scenarios

In 28  
Minutes

Scenario	User/Role	Recommendation
You're the only one in your account	IAM user	Do not use ROOT user
Your team needs access to your AWS account and there is no other identity mechanism	IAM users	Use IAM Groups to manage policies
EC2 instance talks with Amazon S3 or a database	IAM role	
Cross Account Access	IAM role	

# Instance Profile

In 28  
Minutes

- A Container (A Box) for an IAM role
- Used to pass role information to an EC2 instance
- AWS Management Console:
  - An instance profile is automatically created when you create a role for EC2 instance
- From CLI or API
  - Explicitly manage Instance Profiles - CreateInstanceProfile etc
- (REMEMBER) Instance profile is a simple container for IAM Role



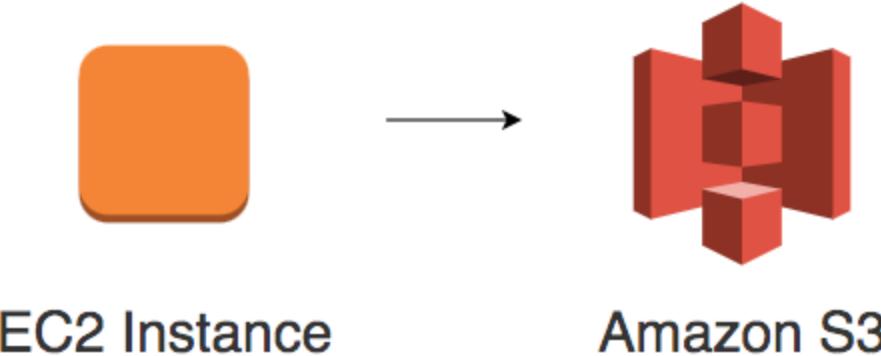
EC2 Instance



Amazon S3

# IAM Role Use case 1 : EC2 talking with S3

In 28  
Minutes



- Create IAM role with access to S3 bucket
- Assign IAM role to EC2 instance
- No need to store credentials in config files
- No need for rotation of keys

# IAM Role Use case 2: Cross Account Access

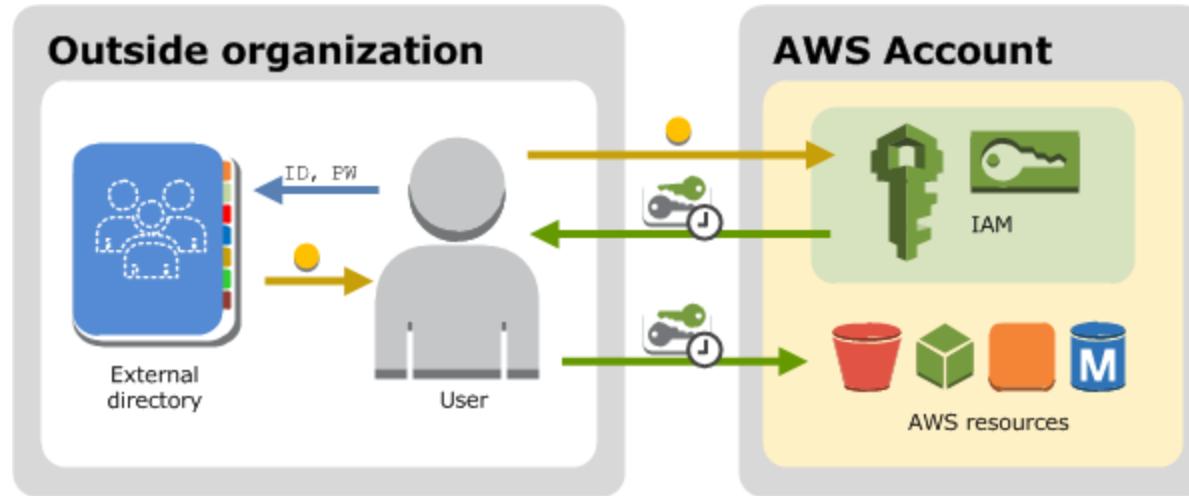
In 28  
Minutes

- PROD Account (111111111111)
  - Create IAM role (ProdS3AccessRole) with right permissions and establish trust relationship with AWS account 222222222222
- DEV Account (222222222222)
  - Grant users (Operations Group) permissions to assume the ProdS3AccessRole in PROD Account
    - Create a customer managed policy ProdS3AccessPolicy allowing access to call STS AssumeRole API for ProdS3AccessRole(arn:aws:iam::111111111111:role/ProdS3AccessRole)
    - Assign the policy to users (Operations Group)
    - (Optional) Enable MFA for assuming the role
- Operations user requests access to the role
  - Background: Call is made to AWS Security Token Service (AWS STS) AssumeRole API for the ProdS3AccessRole role
  - Gets access!



# IAM - Corporate Directory Federation

In 28  
Minutes



- Users authenticated with a **corporate directory**
  - For SAML2.0 compliant identity systems, establish a trust relationship with IAM
  - For enterprise using Microsoft AD (Active Directory), use AWS Directory Service to establish trust
  - Otherwise, set up a custom proxy server to translate user identities from enterprise to IAM roles

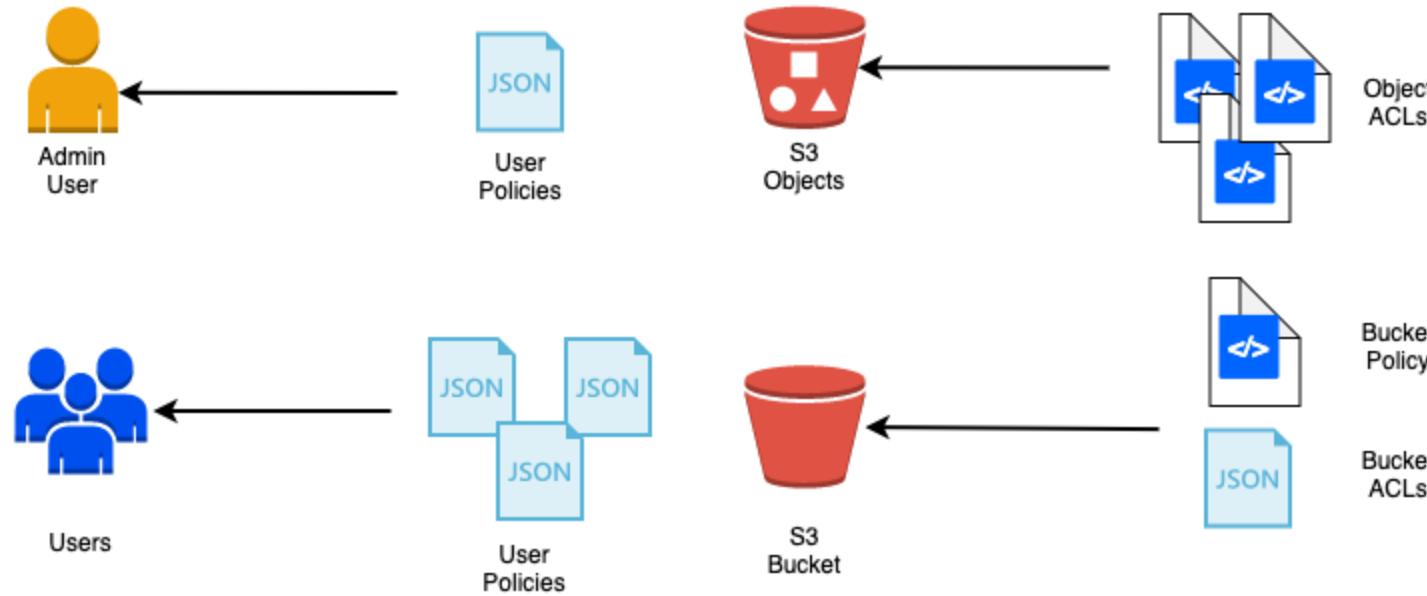
# IAM - Web Identity Federation

In 28  
Minutes

- Authenticate users using **web identities** - mobile or web apps
- Example: Open ID (Facebook, Google, etc)
- **Amazon Cognito** supports login with Facebook, Google, or other OpenID Connect compatible IdP
- Configure Role to use Web Identity as trusted entity
  - Authentication tokens exchanged using **STS AssumeRoleWithWebIdentity API**

# Identity-based and Resource-based policies

In 28  
Minutes



- By default only account owner has access to a S3 bucket
- Access policies enable other users to access S3 buckets and objects:
  - **Identity-based policies** : Attached to an IAM user, group, or role
  - **Resource-based policies and ACLs** : Attached to a resource - S3 buckets, Amazon SQS queues, and AWS KMS keys

# Identity-based and Resource-based policies

In 28  
Minutes

Policy Type	Identity-based	Resource-based
Attached with	IAM user, group, or role	A resource
Type	Managed and Inline	Inline only
Focus	What resource? What actions?	Who? What actions?
Example	Can list S3 buckets with name XYZ	Account A can read and modify. Public can read.
Cross-account access	User should switch role	Simpler. User accesses resource directly from his AWS account
Supported by	All services	Subset of services - S3, SQS, SNS, KMS etc
Policy Conditions	When (dates), Where(CIDR blocks), Enforcing MFA	When (dates), Where(CIDR blocks), Is SSL Mandatory?

# IAM Scenario Questions

In 28  
Minutes

Scenario	Solution
<b>How to rotate access keys without causing problems</b>	Create new access key Use new access key in all apps Disable original access key Test and verify Delete original access key
<b>How are multiple permissions resolved in IAM Policy</b>	If there is an explicit deny - return deny If there is no explicit deny and there is an explicit allow - allow If there is no explicit allow or deny - deny
<b>Which region are IAM users created in</b>	<b>IAM Users are global entities.</b> Can use AWS services in any geographic region
<b>What is the difference between IAM user, Federated user and Web identity federation user</b>	<b>IAM users</b> - created and maintained in your AWS account <b>Federated users</b> - managed outside of AWS - corporate directory <b>Web identity federation users</b> - Amazon Cognito, Amazon, Google, or any OpenID Connect-compatible provider Accounts

# Authentication with IAM - Remember

In 28  
Minutes

- IAM Users identities exist until they are **explicitly deleted**
- IAM allows you to create a **password policy**
  - What characters should your password contain?
  - When does a password expire?
- Access key's should be **constantly rotated**
- Two access keys can be **active simultaneously**. Makes rotation of keys easier.
- An IAM role can be added to already running EC2 instances. **Immediately effective**.
- An IAM role is **NOT** associated with an IAM user.
- An IAM user can assume an IAM role temporarily.

# Authentication with IAM - Remember

In 28  
Minutes

- An IAM role is **NOT associated** with long-term credentials
  - When a user, a resource (For example, an EC2 instance) or an application assumes a Role, it is provided with temporary credentials
- **Do NOT** use AWS IAM root user for regular everyday tasks. Lock it away after creating an admin IAM user.
- Enable **Multi Factor Authentication** for all important IAM operations
  - Extra layer of security
  - MFA Devices
    - Hardware device - Gemalto
    - Virtual device - An app on a smart phone

# Data Encryption

## KMS & Cloud HSM

# Data States

In 28  
Minutes

- **Data at rest:** Stored on a device or a backup
  - Examples : data on a hard disk, in a database, backups and archives
- **Data in motion:** Being transferred across a network
  - Also called Data in transit
  - **Examples :**
    - Data copied from on-premise to cloud storage
    - An application in a VPC talking to a database
  - **Two Types:**
    - In and out of AWS
    - Within AWS
- **Data in use:** Active data processed in a non-persistent state
  - Example: Data in your RAM



EC2 Instance

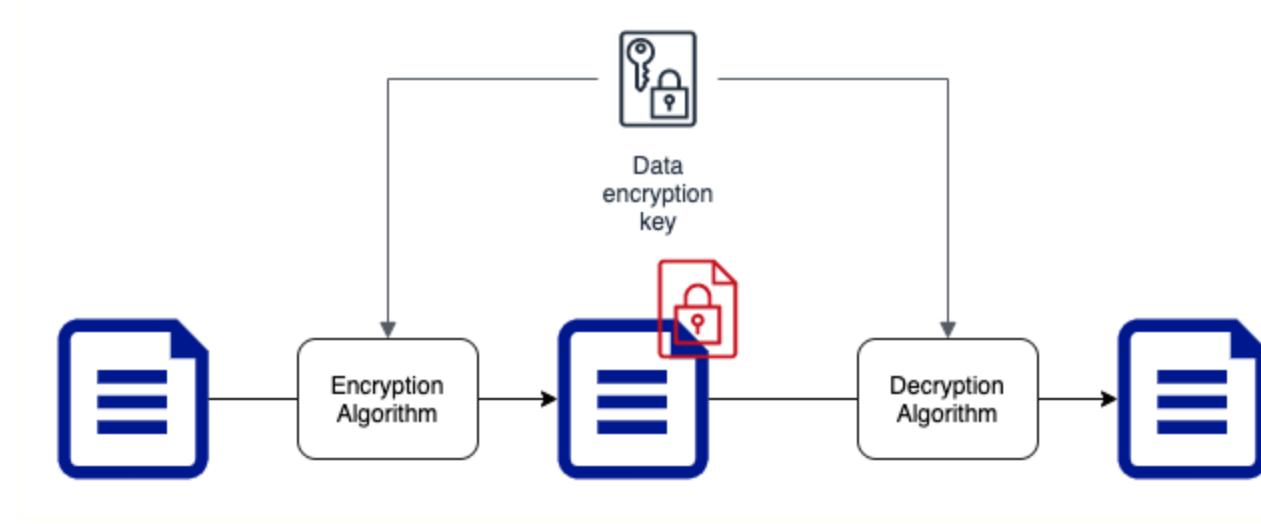


Database

- If you store data as is, what would happen if an **unauthorized entity gets access to it?**
  - Imagine losing an unencrypted hard disk
- **First law of security : Defense in Depth**
- Typically, enterprises encrypt all data
  - Data on your hard disks
  - Data in your databases
  - Data on your file servers
- Is it sufficient if you encrypt data at rest?
  - **No.** Encrypt data in transit - between application to database as well.

# Symmetric Key Encryption

In 28  
Minutes

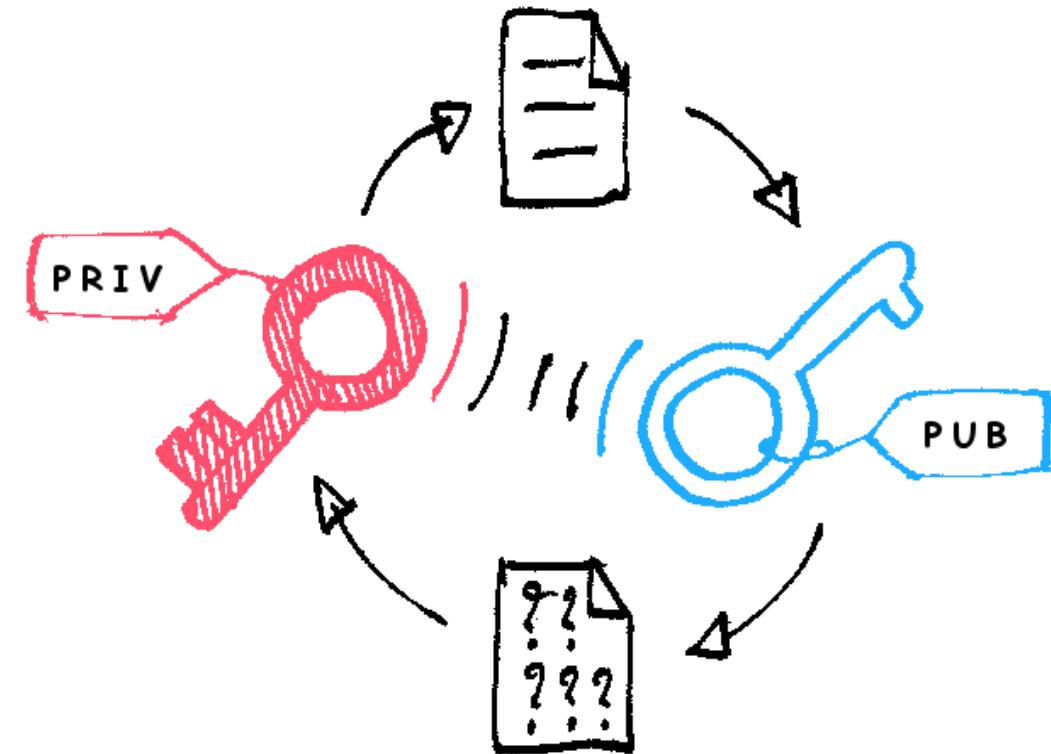


- Symmetric encryption algorithms use the **same key for encryption and decryption**
- Key Factor 1: Choose the **right encryption algorithm**
- Key Factor 2: How do we **secure the encryption key?**
- Key Factor 3: How do we **share the encryption key?**

# Asymmetric Key Encryption

In 28  
Minutes

- Two Keys : Public Key and Private Key
- Also called **Public Key Cryptography**
- Encrypt data with Public Key and decrypt with Private Key
- Share Public Key with everybody and keep the Private Key with you(YEAH, ITS PRIVATE!)
- No crazy questions:
  - Will somebody not figure out private key using the public key?
- How do you create Asymmetric Keys?



[https://commons.wikimedia.org/wiki/File:Asymmetric\\_encry](https://commons.wikimedia.org/wiki/File:Asymmetric_encry)

# KMS and Cloud HSM

In 28  
Minutes

- How do you generate, store, use and replace your keys?
- AWS provides two important services - KMS and Cloud HSM
  - Manage your keys
  - Perform encryption and decryption



AWS KMS



Cloud HSM

- Create and manage **cryptographic keys** (symmetric and asymmetric)
- **Control their use** in your applications and AWS Services
- Define key usage permissions (including **cross account** access)
- Track key usage in AWS CloudTrail (regulations & compliance)
- **Integrates with almost all AWS services** that need data encryption
- **Automatically rotate master keys** once a year
  - No need to re-encrypt previously encrypted data (versions of master key are maintained)
- **Schedule key deletion** to verify if the key is used
  - Mandatory minimum wait period of 7 days (max-30 days)



AWS KMS

# Server Side Encryption with KMS

In 28  
Minutes

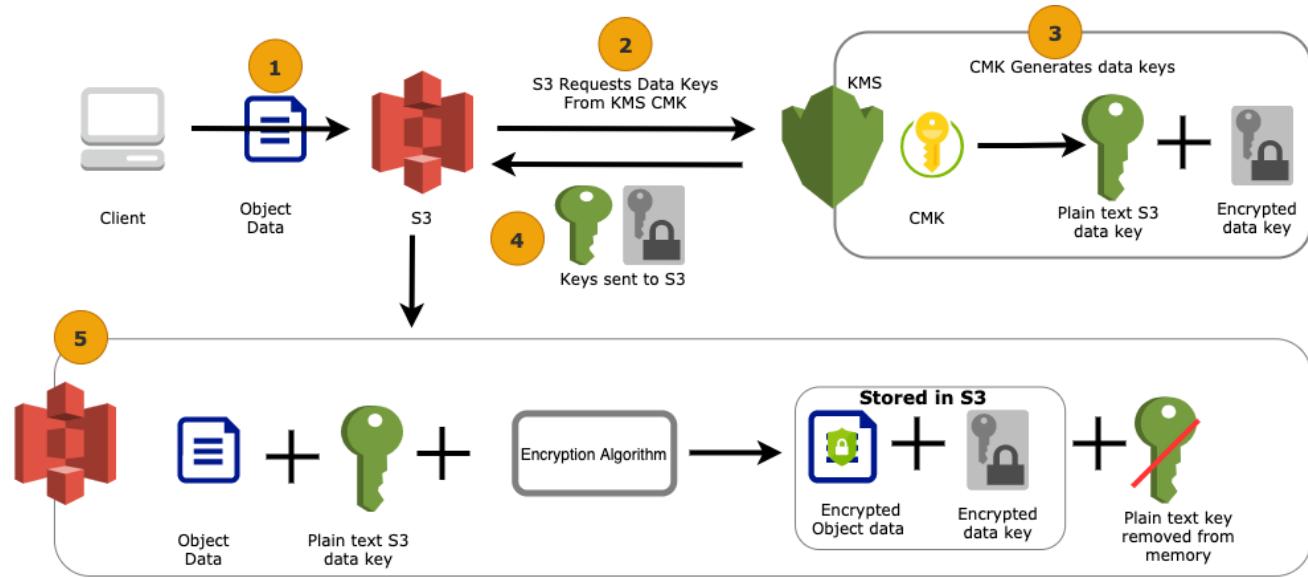
- Create Customer Master Key.  
Map to AWS service (S3)

- Steps

- Data sent to S3
- S3 receives **data keys** from KMS
- S3 encrypts data
- Stores encrypted data & data key

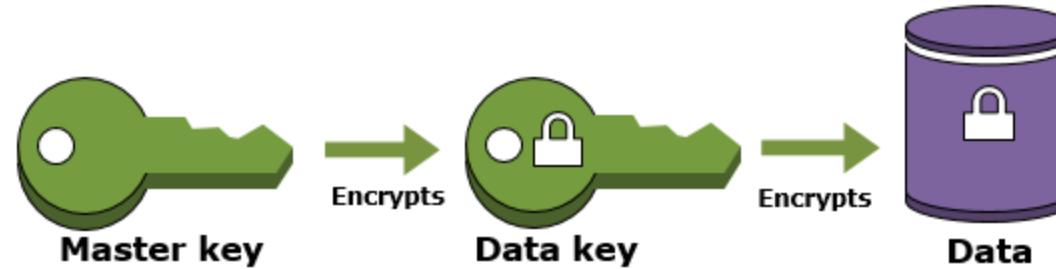
- Remember

- CMK never leaves KMS
- **Encryption of data key - KMS** using CMK
- **Encryption of data - AWS Service - Amazon S3** using data key



# Envelope Encryption

In 28  
Minutes



<https://docs.aws.amazon.com/kms/latest/developerguide/>

- The process KMS uses for encryption is called **Envelope Encryption**
  - Data is encrypted using **data key**
  - Data key is encrypted using Master key
  - Master key **never leaves KMS**
- KMS encrypts small pieces of data (usually data keys) <4 KB

# Decryption of data using KMS

In 28  
Minutes

- AWS service (Amazon S3) sends encrypted data key to KMS
- KMS uses Customer Master Key (CMK) to decrypt and return plain-text data key
- AWS service (Amazon S3) uses the plain-text data key to perform decryption
- (TIP) Remove plain-text data key from memory asap
- (TIP) AWS service needs IAM permissions to use the CMK
- Remember:
  - (Optional) You can associate a key/value map called **encryption context** with any cryptographic operation
  - (TIP) If encryption context is different, decryption will NOT succeed



Amazon S3



AWS KMS

# AWS CloudHSM

In 28  
Minutes

- Managed (highly available & auto scaling) **dedicated single-tenant** Hardware Security Module(HSM) for regulatory compliance
  - (Remember) AWS KMS is a multi-tenant service
- FIPS 140-2 Level 3 compliant
- AWS **CANNOT** access your encryption master keys in CloudHSM
  - In KMS, AWS can access your master keys
  - Be ultra safe with your keys when you are using CloudHSM
  - **(Recommendation)** Use two or more HSMs in separate AZs in a production cluster



Cloud HSM

# AWS CloudHSM

In 28  
Minutes

- AWS KMS can use CloudHSM cluster as "custom key store" to store the keys:
  - AWS Services can continue to talk to KMS for data encryption
  - (AND) KMS does the necessary integration with CloudHSM cluster
- (Best Practice) CloudWatch for monitoring and CloudTrail to track key usage
- Use cases
  - (Web servers) Offload SSL processing
  - Certificate Authority
  - Digital Rights Management
  - TDE for Oracle databases



Amazon S3



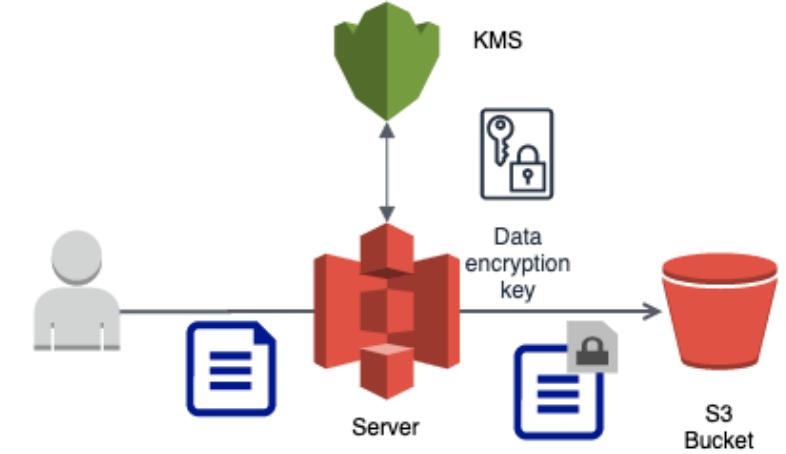
AWS KMS



Cloud HSM

# Server Side Encryption

- Client sends data (as is) to AWS service
- AWS service interacts with KMS to perform encryption on the server side
- Recommended to **use HTTPS endpoints** to ensure encryption of data in transit
  - All AWS services (including S3) provides HTTPS endpoints
  - Encryption is optional with S3 but highly recommended in flight and at rest



# Server Side Encryption - S3

In 28  
Minutes

- **SSE-S3:**

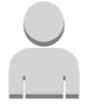
- AWS S3 manages its own keys
- Keys are rotated every month
- Request Header - *x-amz-server-side-encryption(AES256)*

- **SSE-KMS:**

- Customer manages keys in KMS
- Request Headers - *x-amz-server-side-encryption(aws:kms)* and *x-amz-server-side-encryption-aws-kms-key-id(ARN for key in KMS)*

- **SSE-C:**

- Customer sends the key with every request
- S3 performs encryption and decryption without storing the key
- HTTPS is mandatory



User



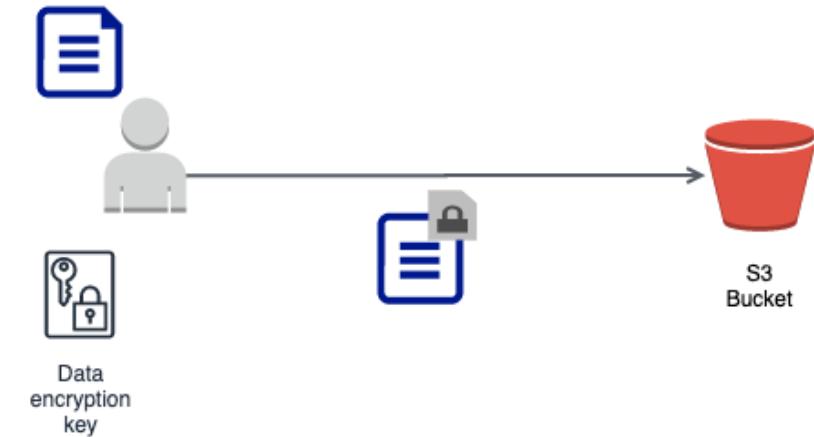
Amazon S3



AWS KMS

# Client Side Encryption

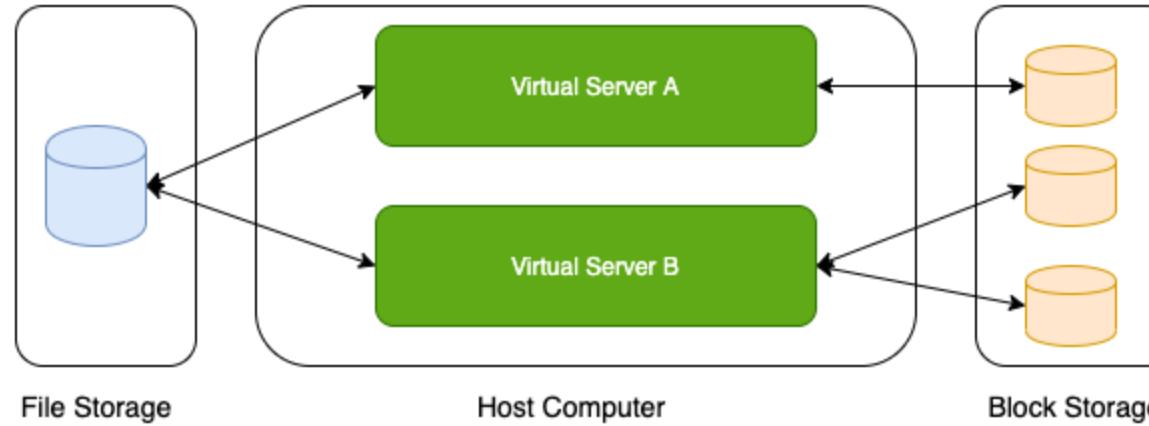
- Client manages encryption process and sends encrypted data to AWS service
  - AWS will not be aware of master key or data key
- AWS service stores data as is
- For Amazon S3, you can use a client library (Amazon S3 Encryption Client)



# Storage Fundamentals

# Storage Types - Block Storage and File Storage

In 28  
Minutes

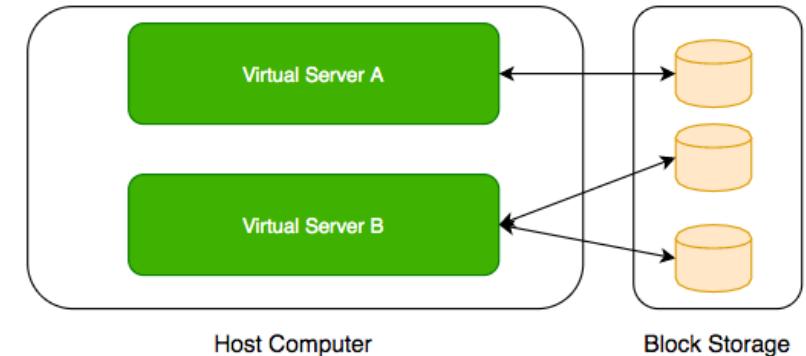


- What is the type of storage of your hard disk?
  - Block Storage
- You've created a file share to share a set of files with your colleagues in a enterprise. What type of storage are you using?
  - File Storage

# Block Storage

In 28  
Minutes

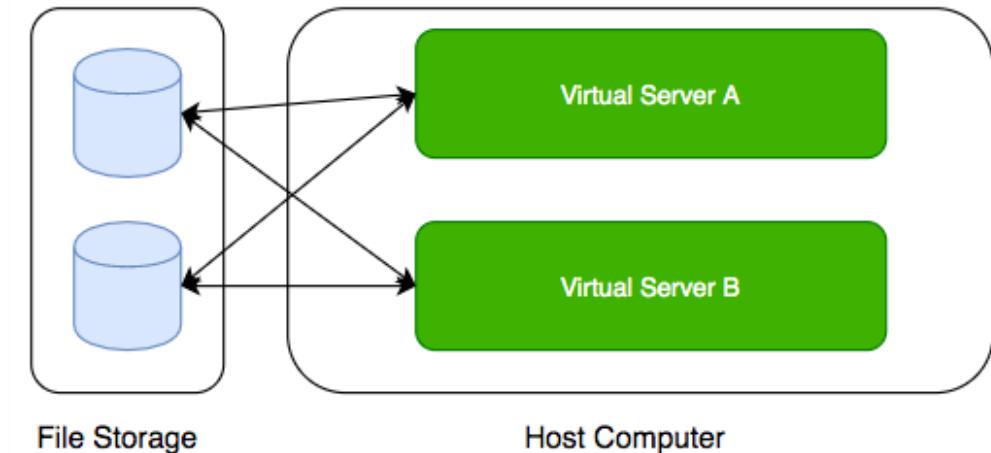
- Use case: Harddisks attached to your computers
- Typically, ONE Block Storage device can be connected to ONE virtual server
- HOWEVER, you can connect multiple different block storage devices to one virtual server
- Used as:
  - **Direct-attached storage (DAS)** - Similar to a hard disk
  - **Storage Area Network (SAN)** - High-speed network connecting a pool of storage devices
    - Used by Databases - Oracle and Microsoft SQL Server



# File Storage

In 28  
Minutes

- Media workflows need huge shared storage for supporting processes like video editing
- Enterprise users need a quick way to share files in a secure and organized way
- These file shares are shared by several virtual servers



# AWS - Block Storage and File Storage

In 28  
Minutes



Amazon EFS



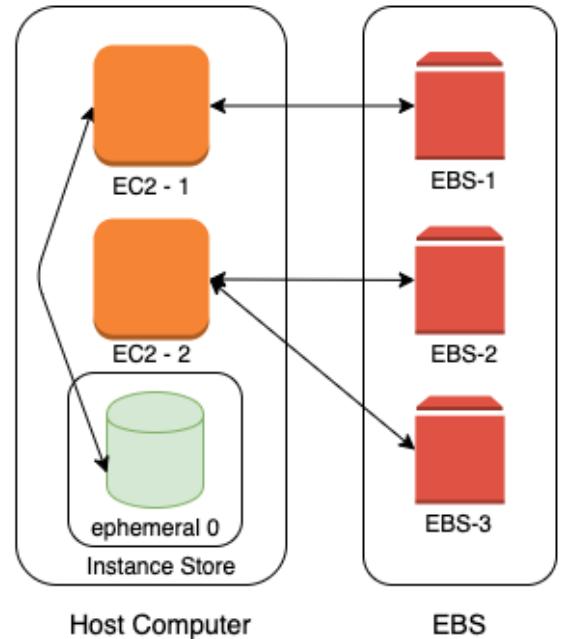
Amazon EBS

- **Block Storage:**
  - Amazon Elastic Block Store (EBS)
  - Instance store
- **File Storage:**
  - Amazon EFS (for Linux instances)
  - Amazon FSx Windows File Servers
  - Amazon FSx for Lustre (high performance use cases)

# EC2 - Block Storage

In 28  
Minutes

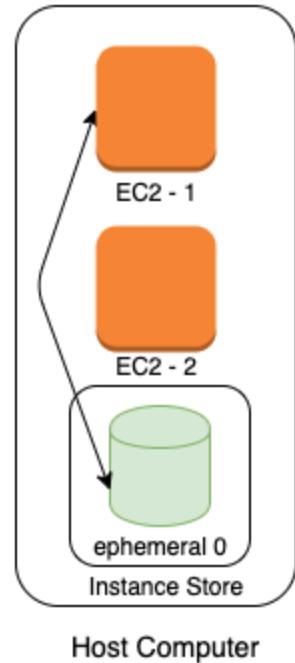
- Two popular types of block storage can be attached to EC2 instances:
  - **Elastic Block Store (EBS)**
  - **Instance Store**
- **Instance Stores** are physically attached to the EC2 instance
  - Temporary data
  - Lifecycle tied to EC2 instance
- **Elastic Block Store (EBS)** is network storage
  - More durable
  - Lifecycle NOT tied to EC2 instance



# Instance Store

In 28  
Minutes

- Physically attached to your EC2 instance
- **Ephemeral storage**
  - Temporary data.
  - Data is lost when hardware fails or an instance is terminated.
  - Use case: cache or scratch files
- Lifecycle is tied to EC2 instance
- Data is NOT lost on instance reboot
- Only some of the EC2 instance types support **Instance Store**



# Instance Store - Advantages and Disadvantages

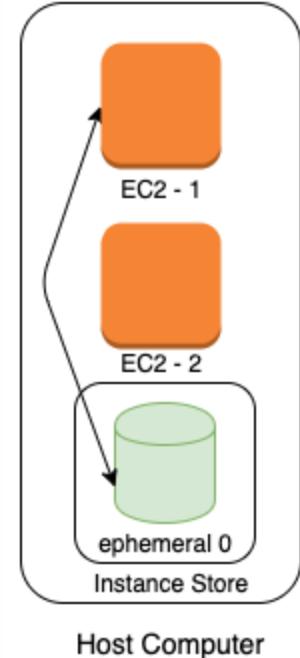
In 28  
Minutes

- **Advantages**

- Very Fast I/O (2-100X of EBS)
- (Cost Effective) **No extra cost.** Cost is included in the cost of EC2 instance
- Ideal for storing **temporary information** - cache, scratch files etc

- **Disadvantages**

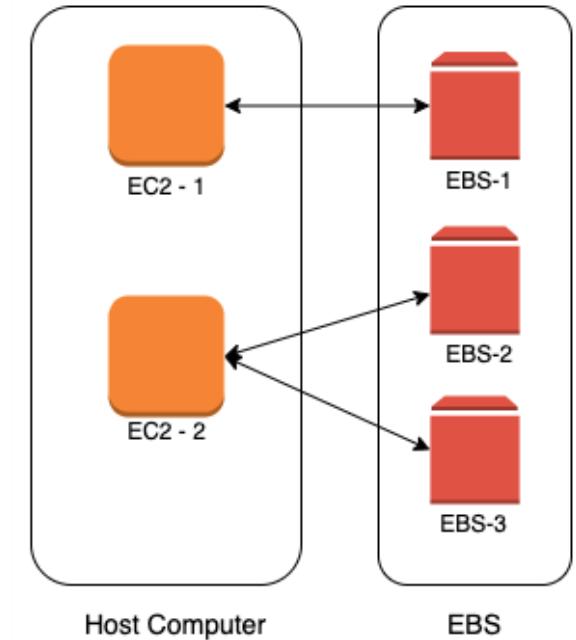
- Slow boot up (up to 5 minutes)
- **Ephemeral storage** (data is lost when hardware fails or instance is terminated)
- **CANNOT take a snapshot** or restore from snapshot
- Fixed size based on instance type
- You cannot detach and attach it to another EC2 instance



# Amazon Elastic Block Store (EBS)

In 28  
Minutes

- Network block storage attached to your EC2 instance
- Provisioned capacity
- Very flexible.
  - Increase size when you need it - when attached to EC2 instance
- Independent lifecycle from EC2 instance
  - Attach/Detach from one EC2 instance to another
- 10X more durable compared to an usual hard disk (annual failure rate of 0.1% - 0.2%)
- 99.999% Availability & replicated within the same AZ
- Use case : Run your custom database



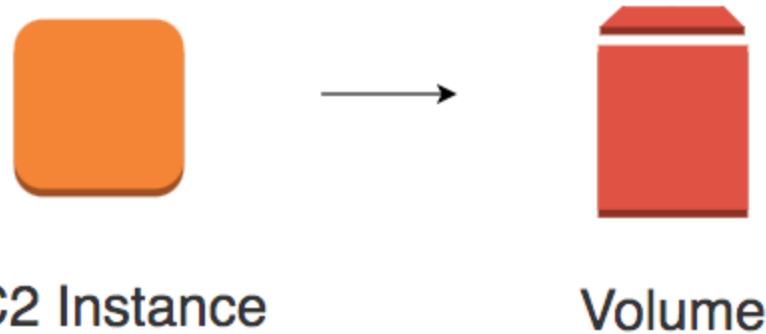
# Amazon EBS vs Instance Store

In 28  
Minutes

Feature	Elastic Block Store (EBS)	Instance Store
Attachment to EC2 instance	As a network drive	Physically attached
Lifecycle	Separate from EC2 instance	Tied with EC2 instance
Cost	Depends on provisioned size	Zero (Included in EC2 instance cost)
Flexibility	Increase size	Fixed size
I/O Speed	Lower (network latency)	2-100X of EBS
Snapshots	Supported	Not Supported
Use case	Permanent storage	Ephemeral storage
Boot up time	Low	High

# Elastic Block Store - Hands-on

In 28  
Minutes



- Create 3 EC2 instances
  - Instance A in AZ A - Root volume
  - Instance B in AZ A - Root volume and Secondary volume
  - Instance C in AZ B - Root volume

# Hard Disk Drive vs Solid State Drive

In 28  
Minutes

*Amazon EBS offers HDD and SSD options!  
How do you choose between them?*

Feature	HDD(Hard Disk Drive)	SSD(Solid State Drive)
Performance - IOPS	Low	High
Throughput	High	High
Great at	Large sequential I/O operations	Small, Random I/O operations & Sequential I/O
Recommended for	Large streaming or big data workloads	Transactional workloads
Cost	Low	Expensive
Boot Volumes	Not Recommended	Recommended

# Amazon Elastic Block Store (EBS) SSD Types

In 28  
Minutes

## General Purpose SSD (gp2) (\$\$\$)

- I/O performance **increases with size** - 3 IOPS/GB (min 100) upto 16,000 IOPS
- **Balance price & performance** for transactional workloads (Cost sensitive)
- **Use cases** : small/medium databases, dev/test environments, & boot volumes
- **Burst** up to 3,000 IOPS above the baseline

## Provisioned IOPS SSD (io1) (\$\$\$\$)

- Provision IOPS you need
- Designed for **low latency transactional** workloads
- Delivers consistent performance for **random and sequential** access
- **Use cases** : large relational or NoSQL databases

# Amazon Elastic Block Store (EBS) HDD Types

In 28  
Minutes



Amazon EBS

## Throughput Optimized HDD (st1) (\$\$)

- For frequently accessed, throughput-intensive sequential workloads
- Use cases : MapReduce, Kafka, log processing, data warehouse, and ETL

## Cold HDD (sc1) (\$)

- Lowest Cost
- Use cases : infrequent data access - very low transaction databases

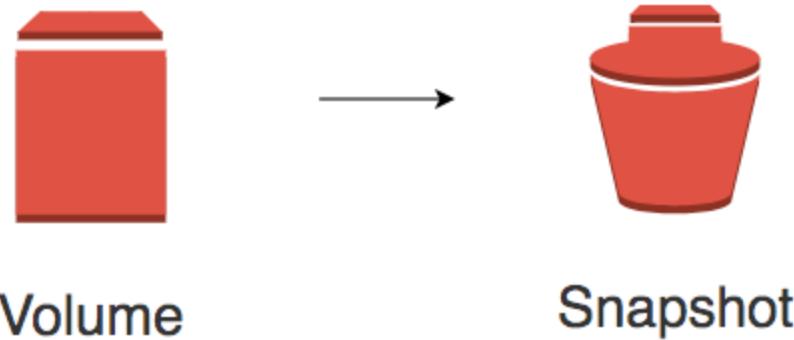
# Amazon Elastic Block Store (EBS) Types

In 28  
Minutes

	Provisioned IOPS SSD	General Purpose SSD	Throughput Optimized HDD	Cold HDD
Volume Size	4 GB - 16 TB	1 GB - 16 TB	500 GB - 16 TB	500 GB - 16 TB
Max IOPS/Volume	64,000	16,000	500	250
Max Throughput/Volume	1,000 MB/s	250 MB/s	500 MB/s	250 MB/s
Boot Volume	✓	✓	X	X

# Amazon Elastic Block Store (EBS)

In 28  
Minutes



- Supports **live changes to volumes** without service interruptions
  - Increase size
  - Change type
  - Increase IOPS capacity
- Take **point-in-time snapshots** of EBS volumes (stored in Amazon S3)
  - **Asynchronous process** - reduces performance but EBS volume is available
  - Snapshots cannot be accessed directly from S3
  - Use EC2 APIs to restore them to EBS volumes

# Amazon EBS Snapshots

In 28  
Minutes

- Snapshots are **incremental**
  - BUT you don't lose data by deleting older snapshots
  - Deleting a snapshot **only deletes data which is NOT needed** by other snapshots
  - Do not hesitate to delete unnecessary snapshots
  - All information needed to restore the active snapshots will be retained
- Can be **shared** with other AWS accounts
  - To share an encrypted snapshot, you would need to share (give permissions) to encryption keys also
- Constrained to the **created region**
  - To use in other regions, copy it
- **Fast Snapshot Restore** speeds up the process of creating volumes from snapshots
  - Eliminates need for pre-warming volumes created from snapshots

# Amazon EBS Encryption

In 28  
Minutes



- Encryption (AES-256) is done *transparently* using **master keys from KMS**
- Turning on Encryption **automatically encrypts**:
  - **Data at rest**
    - Data volumes, boot volumes
    - Snapshots
  - **Data in transit**
    - Between EC2 instances and EBS volume
    - Between EBS volume and EBS snapshots

# Faster I/O performance between EC2 and EBS

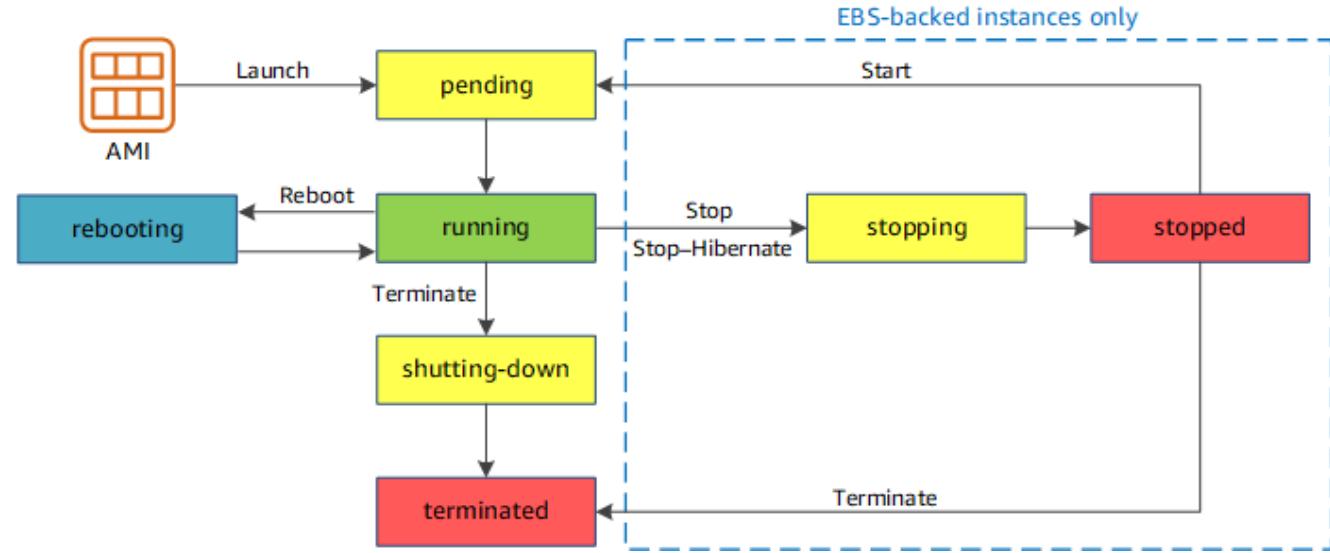
In 28  
Minutes

Option	Description	
Launch EC2 instances as EBS optimized Instances	Available on select instances Default and free for a few instance types Hourly fee for other instance types	
Enhanced networking through Elastic Network Adapter (ENA)	Increases throughput(PPS) Needs custom configuration	
Use Elastic Fabric Adapter (EFA)	Available on select instances NOT available on Windows EC2 instances EFA = ENA + OS-bypass <b>Ideal for High Performance Computing (HPC) applications like weather modeling</b>	 EC2 ↓ Amazon EBS

# EC2 Instance Lifecycle

In 28  
Minutes

- Only EBS backend instances can be **stopped or hibernated**
- When you terminate an EC2 instance, **everything** on root device (EBS or instance store) is lost
- Hibernating **preserves RAM memory** in root EBS volume
  - Provides **quick restarts** for use cases with either long running processes or slow boot up times
- Hibernating can be done for a **max of 60 days**

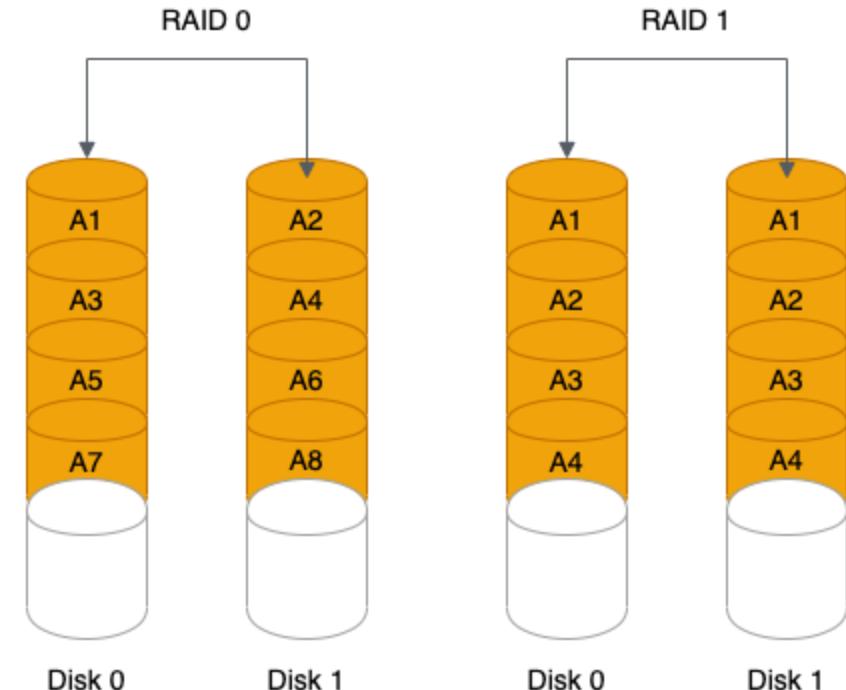


<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-lifecycle.html>

# RAID

In 28  
Minutes

- Need **higher durability** than one EBS volume?
  - Use **RAID 1** structure
  - Same performance and storage capacity BUT higher fault tolerance
- Need **higher IOPS or storage** than one EBS volume?
  - Use **RAID 0** structure
  - Double the storage, IOPS and throughput BUT data lost even if one disk fails
  - Use this when I/O performance is more important than fault tolerance. Ex: Replicated Database



# EBS Snapshots and AMIs

In 28  
Minutes



- You can create:
  - Snapshot from EBS volume and vice versa
  - AMI from EC2 instance and vice versa
  - AMI from root EBS volume snapshots

# Using an AMI from different AWS account or region

In 28  
Minutes



- Scenario : You want to use an AMI belonging to a different AWS account or a different region
  - REMEMBER : AMI are restricted to a region
  - Step I (Optional) : Owner of AMI provides read permission to the AMI
  - Step II(Optional) : For encrypted AMI, owner should share the encryption keys
  - Step III : Copy the AMI into your region
  - If you do not have permission to copy an AMI but you have permission to use an AMI:
    - Create an EC2 instances from AMI
    - Create a new AMI from EC2 instance

# Amazon EBS Scenarios - with EC2

In 28  
Minutes

Scenario	Solution
Can I attach an EBS volume in us-east-1a to EC2 instance in us-east-1b?	No. EBS volumes should be in the same AZ as your EC2 instance
Can I attach multiple EBS volumes to EC2 instance?	Yes
Can I attach an EBS volume with two EC2 instances?	No
Can I switch EBS volume from EC2 instance to another?	Yes
Will an EBS volume be immediately available when attached to an EC2 instance?	Yes. However, by default, data is lazily loaded
How do you ensure that an EBS volume is deleted when EC2 instance is terminated?	Enable <b>Delete on Termination</b> on EC2 instance
How do you retain EBS volume even if an EBS backed EC2 instance fails?	Remember : On termination of EC2 instance all data on root volume is lost (even if it is EBS backed) Detach the EBS volume before terminating the instance Recover data by connecting the EBS volume to another EC2 instance

# Amazon EBS Scenarios - Snapshots

In 28  
Minutes

Scenario	Solution
How do you create an EBS volume from an EBS volume in a different AZ?	Take a snapshot Create EBS volume from snapshot
How do you create an EBS volume from EBS volume in a different region?	Take a snapshot Copy the snapshot to second region Create EBS volume from snapshot in second region
What is the lowest cost option to maintain snapshots with EBS?	Store just the latest snapshot. Other snapshots can be deleted without a problem
How do you encrypt an unencrypted EBS volume?	Take a snapshot Encrypt the snapshot Create new encrypted volume from snapshot
How do you automate the complete lifecycle (creation, retention, and deletion) of Amazon EBS snapshots?	Use Amazon Data Lifecycle Manager Reduces costs and maintenance effort

# Amazon EBS - Summary

In 28  
Minutes

- Amazon EBS vs instance store
- **Features:**
  - Highly available and durable (within the same AZ)
  - Supports live changes to volumes without service interruptions
  - Transparent encryption integration with KMS
- **Types:**
  - **Cold HDD:** Infrequent access usecases (minimum cost)
  - **Throughput Optimized HDD:** Frequently accessed, large sequential operations with high throughput (cost sensitive)
  - **General Purpose SSD:** System boot volumes and transactional workloads
  - **Provisioned IOPS SSD:** Transactional workloads needing very high IOPS
- EBS volume <-> Snapshot -> AMI

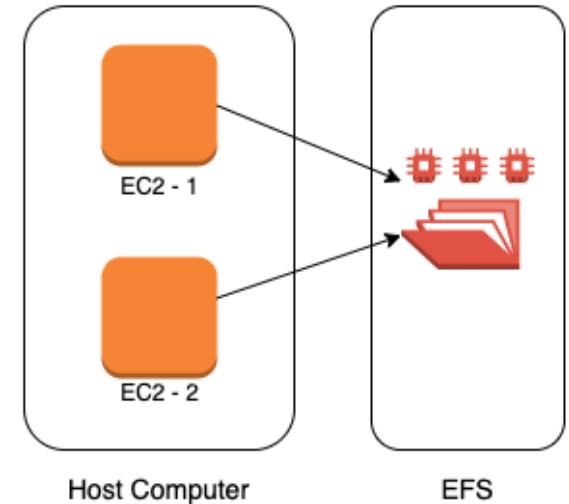


Amazon EBS

# Amazon EFS

In 28  
Minutes

- Petabyte scale, Auto scaling shared file storage
  - POSIX compliant
  - Supports NFS v4.0 and v4.1 protocols
- Pay for use
- High availability and durability across AZs in one region
- Compatible with Amazon EC2 Linux-based instances
  - Share with thousands of Amazon EC2 instances
  - Use Max I/O Mode for higher throughput (with a small increase in latency for all file operations)
- Use cases : home directories, file share, media workflows and content management



# Amazon FSx for Lustre

In 28  
Minutes

- File system **optimized for performance**
  - For high performance computing (HPC), machine learning, and media processing use cases
  - Sub-millisecond latencies, up to hundreds of gigabytes per second of throughput, and up to millions of IOPS
- Integrates with Amazon S3
  - Process data sets directly stored in S3
- POSIX-compliant
  - Connect Linux-based applications without having to make any changes
- File system data is automatically encrypted at-rest and in-transit

# Amazon FSx Windows File Servers

In 28  
Minutes

- Fully managed Windows file servers
- Uses Service Message Block (SMB) protocol
- Accessible from Windows, Linux and MacOS instances
- Integrates with Microsoft Active Directory (AD) to support Windows-based environments and enterprises.
- Offers single-AZ and multi-AZ deployment options, SSD and HDD storage options, and provides fully managed backups.
- File system data is automatically encrypted at rest and in transit.
- (Remember) All File Sharing options are accessible on AWS or on premises

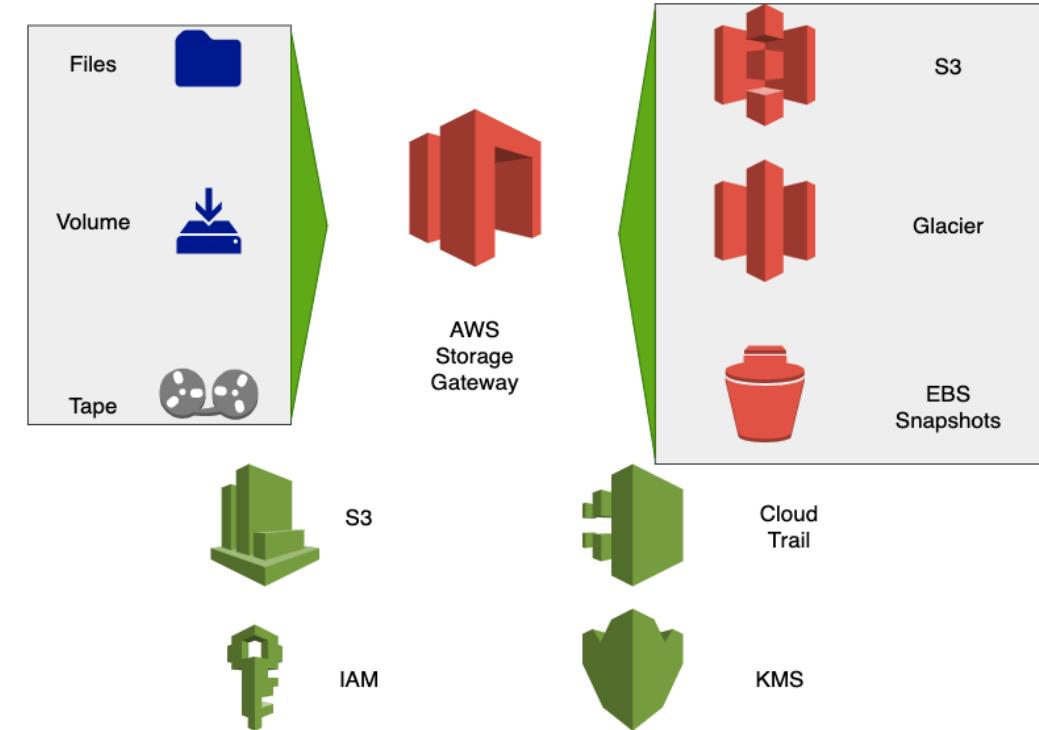
# Review of storage options

Type	Examples	Latency	Throughput	Shareable
Block	EBS, Instance Store	Lowest	Single	Attached to one instance at a time. Take snapshots to share.
File	EFS, FSx Windows, FSx for Lustre	Low	Multiple	Yes
Object	S3	Low	Web Scale	Yes
Archival	Glacier	Minutes to hours	High	No

# AWS Storage Gateway

In 28  
Minutes

- Hybrid storage (cloud + on premise)
- Unlimited cloud storage for on-premise software applications and users with good performance
- (Remember) Storage Gateway and S3 Glacier encrypt data by default
- **Three Options**
  - AWS Storage File Gateway
  - AWS Storage Tape Gateway
  - AWS Storage Volume Gateway
- VM image with AWS Storage Gateway software deployed on-premises



# AWS Storage File Gateway

In 28  
Minutes

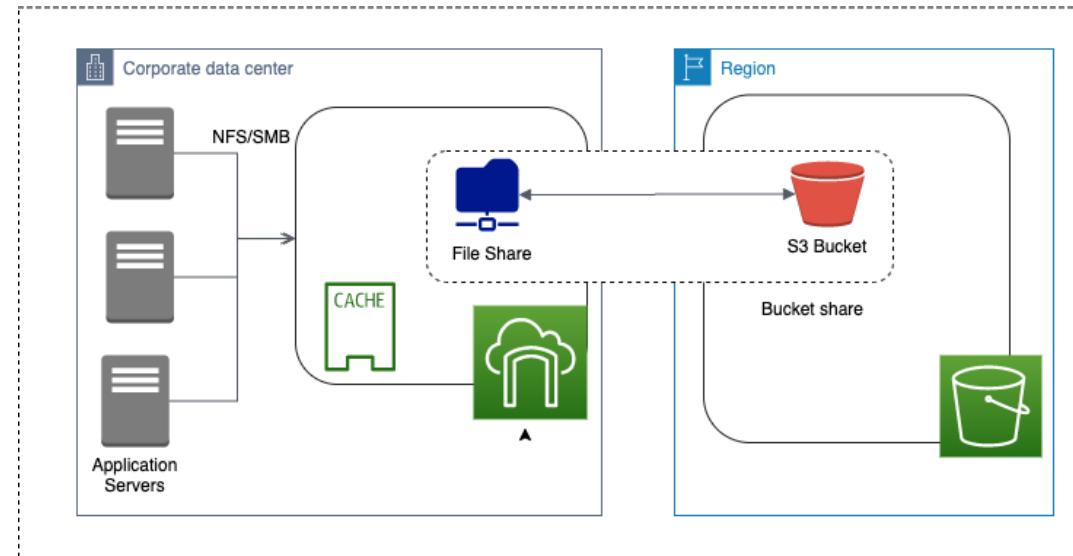
- **Problem Statement:** Large on-premise file share with terabytes of data
  - Users put files into file share and applications use the files
  - Managing it is becoming expensive
  - Move the file share to cloud without performance impact
- AWS Storage File Gateway provides cloud storage for your file shares
  - Files stored in Amazon S3 & Glacier
  - Supports Network File System (NFS) and Server Message Block (SMB)



# AWS Storage File Gateway

In 28  
Minutes

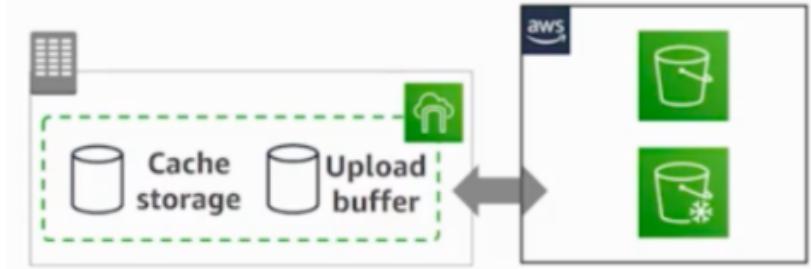
- File gateway deployed as **virtual machine on-premises**
  - Maintains a local cache with most recently used objects
- Benefits from **S3 features**
  - High durability, low-cost, lifecycle management and cross-region replication
- Benefits from **S3 integrations**
  - Data analytics and machine learning applications using Amazon EMR or Amazon Athena
- Each file gateway supports **up to 10 bucket shares**



# AWS Storage Tape Gateway

In 28  
Minutes

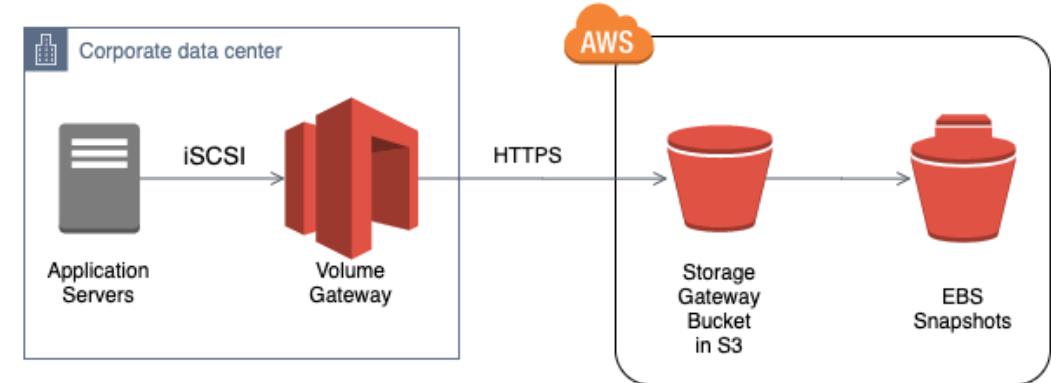
- Tape backups used in enterprises (archives)
  - Stored off-site - expensive, physical wear and tear
- **AWS Storage Tape Gateway** - Avoid physical tape backups
- No change needed for tape backup infrastructure
- Backup data to virtual tapes (actually, Amazon S3 & Glacier)
- Benefit from S3 features
  - encryption, high durability, low-cost, and cross-region replication
- Use **S3 lifecycle management**
  - move data to S3 Glacier and S3 Glacier Deep Archive



# AWS Storage Volume Gateway

In 28  
Minutes

- **Volume Gateway** : Move block storage to cloud
- Supports iSCSI protocol
- Reduce costs
- Automate backup and disaster recovery
- Use AWS Backup for backup and restore
- Use cases
  - Backup and disaster recovery
  - Migration of application data



# AWS Storage Volume Gateway - Cached and Stored

In 28  
Minutes

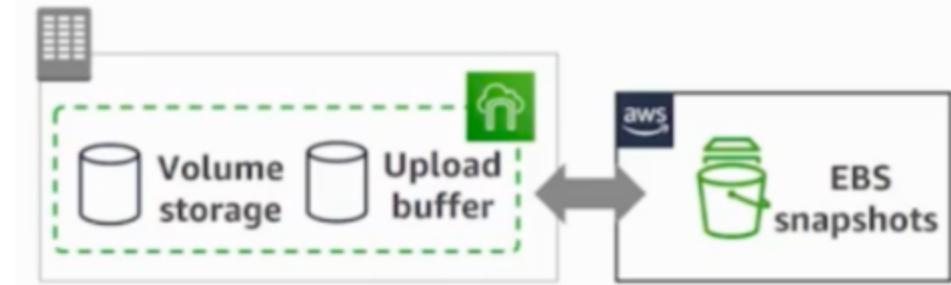
- **Cached (Gateway Cached Volumes):**

- Primary Data Store - AWS - Amazon S3
- On-premise cache stores frequently accessed data
- Data in S3 CANNOT be accessed directly
  - Take EBS snapshots from cached volumes



- **Stored (Gateway Stored Volumes):**

- Primary Data Store - On-Premises
- Asynchronous copy to AWS
- Stored as EBS snapshots
  - For disaster recovery, restore to EBS volumes



# AWS Storage Gateway - Summary

In 28  
Minutes

- Key to look for : **Hybrid storage** (cloud + on premise)
  - File share (NFS or SMB) + Looking for S3 features and integrations => **AWS Storage File Gateway**
  - Tapes on cloud => **AWS Storage Tape Gateway**
  - Volumes on cloud (Block Storage) => **AWS Storage Volume Gateway**
    - High performance => **Stored**
    - Otherwise => **Cached**
- Needs additional setup on-premises
  - VM image with AWS Storage Gateway **software** deployed on-premises or on EC2 instance

# Database Fundamentals

# Databases Primer

In 28  
Minutes

- Databases provide **organized** and **persistent** storage for your data
- To **choose between different database types**, we would need to understand:
  - Availability
  - Durability
  - RTO
  - RPO
  - Consistency
  - Transactions etc
- Let's get started on a **simple journey** to understand these

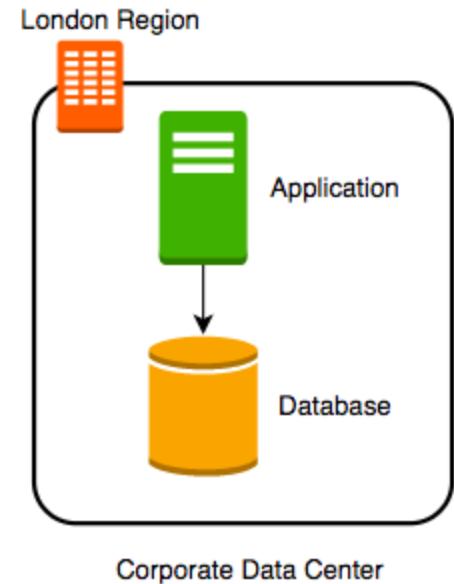


Database

# Database - Getting Started

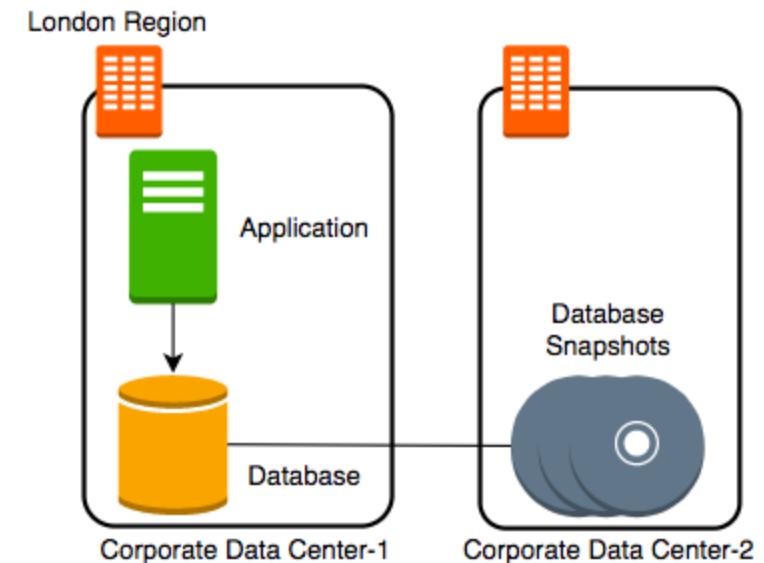
In 28  
Minutes

- Imagine a database deployed in a data center in London
- Let's consider some challenges:
  - Challenge 1: Your database will go down if the data center crashes or the server storage fails
  - Challenge 2: You will lose data if the database crashes



# Database - Snapshots

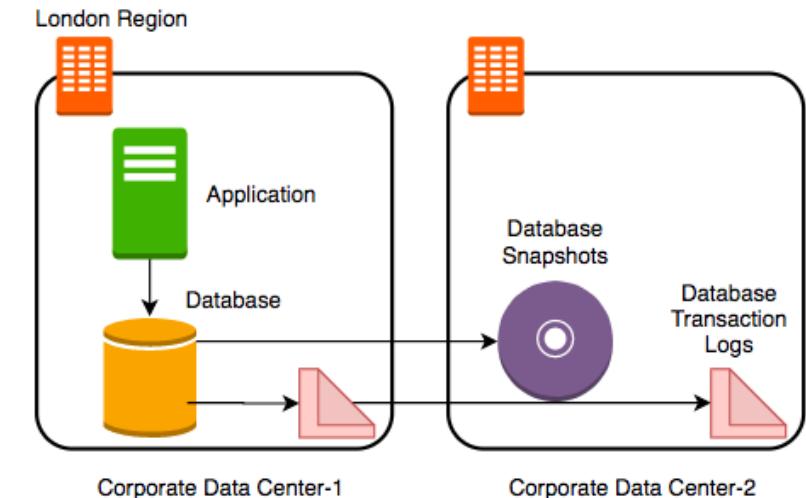
- Let's automate taking copy of the database (**take a snapshot**) every hour to another data center in London
- Let's consider some challenges:
  - **Challenge 1:** Your database will go down if the data center crashes
  - **Challenge 2 (PARTIALLY SOLVED):** You will lose data if the database crashes
    - You can setup database from latest snapshot. But depending on when failure occurs you can lose up to an hour of data
  - **Challenge 3(NEW):** Database will be slow when you take snapshots



# Database - Transaction Logs

In 28  
Minutes

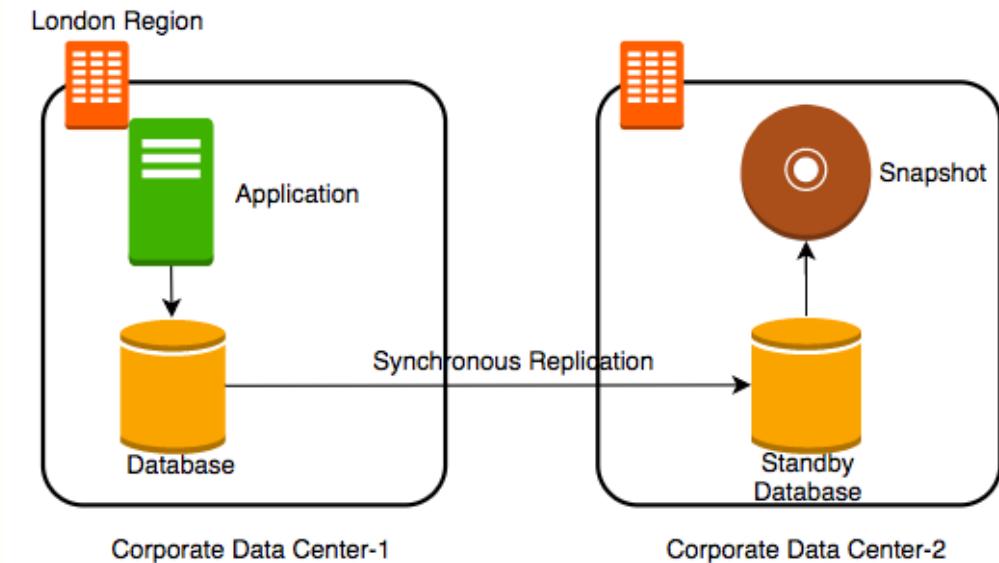
- Let's add transaction logs to database and create a process to copy it over to the second data center
- Let's consider some challenges:
  - **Challenge 1:** Your database will go down if the data center crashes
  - **Challenge 2 (SOLVED):** You will lose data if the database crashes
    - You can setup database from latest snapshot and apply transaction logs
  - **Challenge 3:** Database will be slow when you take snapshots



# Database - Add a Standby

In 28  
Minutes

- Let's add a **standby database** in the second data center with replication
- Let's consider some challenges:
  - **Challenge 1 (SOLVED):** Your database will go down if the data center crashes
    - You can switch to the standby database
  - **Challenge 2 (SOLVED):** You will lose data if the database crashes
  - **Challenge 3 (SOLVED):** Database will be slow when you take snapshots
    - Take snapshots from standby.
    - Applications connecting to master will get good performance always



# Availability and Durability

In 28  
Minutes

- **Availability**
  - Will I be able to access my data now and when I need it?
  - Percentage of time an application provides the operations expected of it
- **Durability**
  - Will my data be available after 10 or 100 or 1000 years?
- Examples of measuring availability and durability:
  - 4 9's - 99.99
  - 11 9's - 99.999999999
- Typically, an **availability of four 9's** is considered very good
- Typically, a **durability of eleven 9's** is considered very good

# Availability

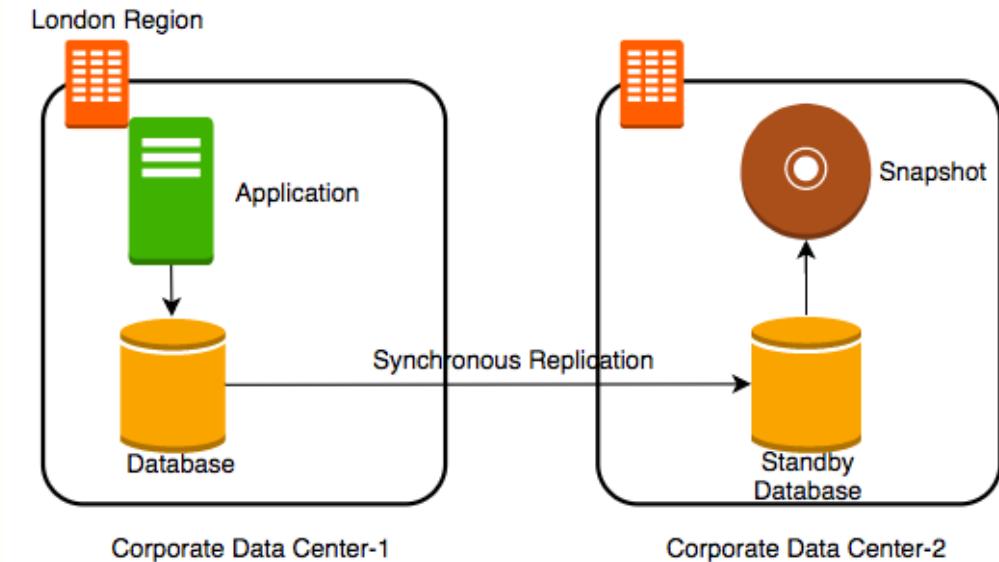
In 28  
Minutes

Availability	Downtime (in a month)	Comment
99.95%	22 minutes	
99.99% (4 9's)	4 and 1/2 minutes	Typically online apps aim for 99.99% (4 9's) availability
99.999% (5 9's)	26 seconds	Achieving 5 9's availability is tough

# Durability

In 28  
Minutes

- What does a durability of 11 9's mean?
  - If you **store one million files for ten million years**, you would expect to **lose one file**
- Why should durability be high?
  - Because we hate losing data
  - Once we lose data, it is gone



# Increasing Availability and Durability of Databases

In 28  
Minutes

- **Increasing Availability:**

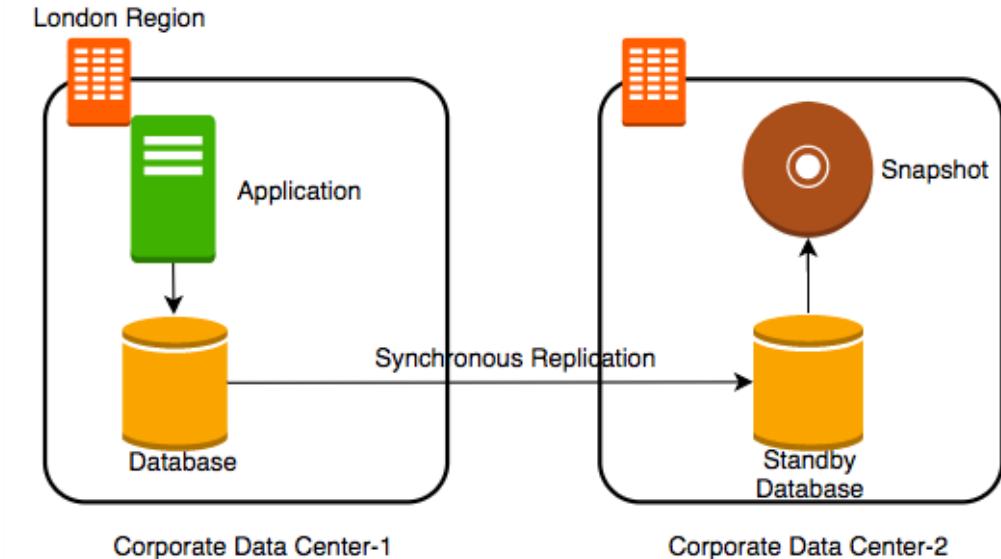
- Have multiple standbys available
  - in multiple AZs
  - in multiple Regions

- **Increasing Durability:**

- Multiple copies of data (standbys, snapshots, transaction logs and replicas)
  - in multiple AZs
  - in multiple Regions

- **Replicating data comes with its own challenges!**

- We will talk about them a little later



# Database Terminology : RTO and RPO

In 28  
Minutes

- Imagine a financial transaction being lost
- Imagine a trade being lost
- Imagine a stock exchange going down for an hour
- Typically businesses are fine with some downtime but they hate losing data
- Availability and Durability are technical measures
- How do we measure how quickly we can recover from failure?
  - RPO (Recovery Point Objective): Maximum acceptable period of data loss
  - RTO (Recovery Time Objective): Maximum acceptable downtime
- Achieving minimum RTO and RPO is expensive
- Trade-off based on the criticality of the data



Database

# Question - RTO and RPO

In 28  
Minutes

- You are running an EC2 instance storing its data on a EBS. You are taking EBS snapshots every 48 hours. If the EC2 instance crashes, you can manually bring it back up in 45 minutes from the EBS snapshot. What is your RTO and RPO?
  - RTO - 45 minutes
  - RPO - 48 hours

# Achieving RTO and RPO - Failover Examples

In 28  
Minutes

## Scenario

Very small data loss (RPO - 1 minute)

Very small downtime (RTO - 5 minutes)

Very small data loss (RPO - 1 minute)

BUT I can tolerate some downtimes (RTO - 15 minutes)

Data is critical (RPO - 1 minute) but I can tolerate downtime of a few hours (RTO - few hours)

Data can be lost without a problem (for example: cached data)

## Solution

Hot standby - Automatically synchronize data

Have a standby ready to pick up load

Use automatic failover from master to standby

Warm standby - Automatically synchronize data

Have a standby with minimum infrastructure

Scale it up when a failure happens

Create regular data **snapshots and transaction logs**

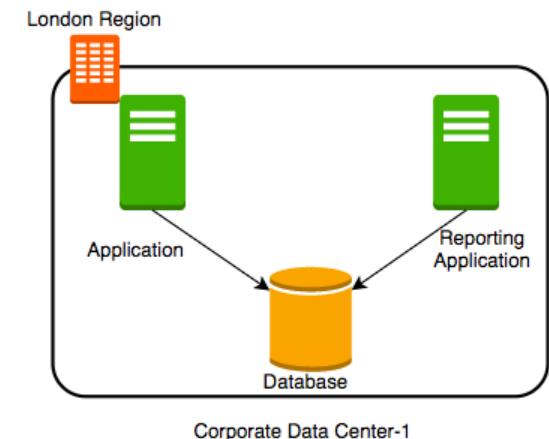
Create database from snapshots and transactions logs when a failure happens

Failover to a completely new server

# (New Scenario) Reporting and Analytics Applications

In 28  
Minutes

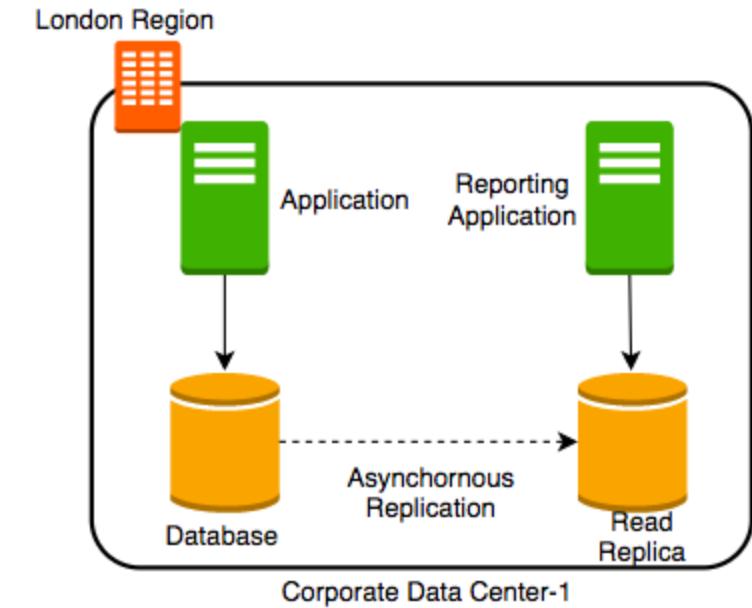
- New reporting and analytics applications are being launched using the same database
  - These applications will ONLY read data
- Within a few days you see that the database performance is impacted
- How can we fix the problem?
  - Vertically scale the database - increase CPU and memory
  - Create a database cluster - typically database clusters are expensive to setup
  - Create read replicas - Run read only applications against read replicas



# Database - Read Replicas

In 28  
Minutes

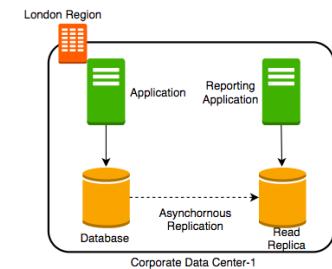
- Add read replica
- Connect reporting and analytics applications to read replica
- Reduces load on the master databases
- Upgrade read replica to master database (supported by some databases)
- Create read replicas in multiple regions
- Take snapshots from read replicas



# Consistency

In 28  
Minutes

- How do you ensure that data in multiple database instances (standbys and replicas) is updated simultaneously?
- **Strong consistency** - Synchronous replication to all replicas
  - Will be slow if you have multiple replicas or standbys
- **Eventual consistency** - Asynchronous replication. A little lag - few seconds - before the change is available in all replicas
  - In the intermediate period, different replicas might return different values
  - Used when scalability is more important than data integrity
  - Examples : Social Media Posts - Facebook status messages, Twitter tweets, LinkedIn posts etc
- **Read-after-Write consistency** - Inserts are immediately available. Updates and deletes are eventually consistent
  - Amazon S3 provides read-after-write consistency



# Database Categories

In 28  
Minutes

- There are **several categories** of databases:
  - Relational (OLTP and OLAP), Document, Key Value, Graph, In Memory among others
- **Choosing type of database** for your use case is not easy. A few factors:
  - Do you want a **fixed schema**?
    - Do you want flexibility in defining and changing your schema? (schemaless)
  - What level of **transaction properties** do you need? (atomicity and consistency)
  - What kind of **latency** do you want? (seconds, milliseconds or microseconds)
  - **How many transactions** do you expect? (hundreds or thousands or millions of transactions per second)
  - **How much data** will be stored? (MBs or GBs or TBs or PBs)
  - and a lot more...



Amazon RDS



ElasticCache



DynamoDB

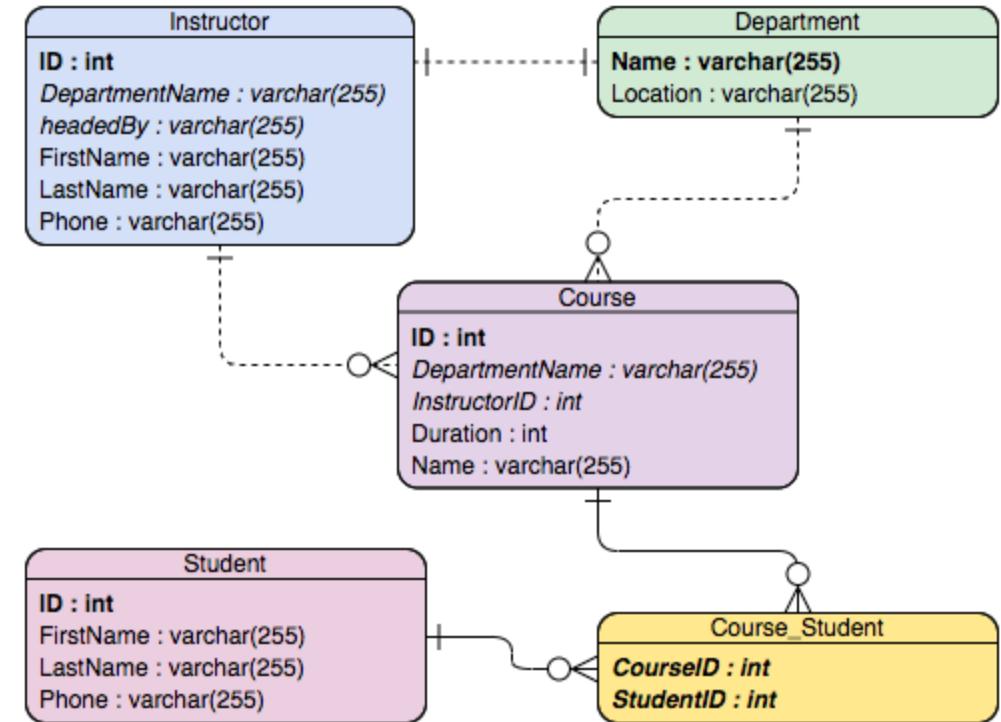


Redshift

# Relational Databases

In 28  
Minutes

- This was the **only** option until a decade back!
- Most **popular** (or unpopular) type of databases
- **Predefined schema** with tables and relationships
- Very **strong transactional** capabilities
- Used for
  - OLTP (Online Transaction Processing) use cases and
  - OLAP (Online Analytics Processing) use cases



# Relational Database - OLTP (Online Transaction Processing)

08  
Minutes

- Applications where **large number of users make large number of small transactions**
  - small data reads, updates and deletes
- **Use cases:**
  - Most traditional applications, ERP, CRM, e-commerce, banking applications
- **Popular databases:**
  - MySQL, Oracle, SQL Server etc
- Recommended AWS Managed Service:
  - **Amazon RDS**
  - Supports Amazon Aurora, PostgreSQL, MySQL, MariaDB (Enhanced MySQL), Oracle Database, and SQL Server



Amazon RDS

# Relational Database - OLAP (Online Analytics Processing)

In 28 Minutes

- Applications allowing users to **analyze petabytes of data**
  - Examples : Reporting applications, Data ware houses, Business intelligence applications, Analytics systems
  - Sample application : Decide insurance premiums analyzing data from last hundred years
  - Data is consolidated from multiple (transactional) databases
- Recommended AWS Managed Service
  - Amazon Redshift
  - Petabyte-scale distributed data ware house based on PostgreSQL

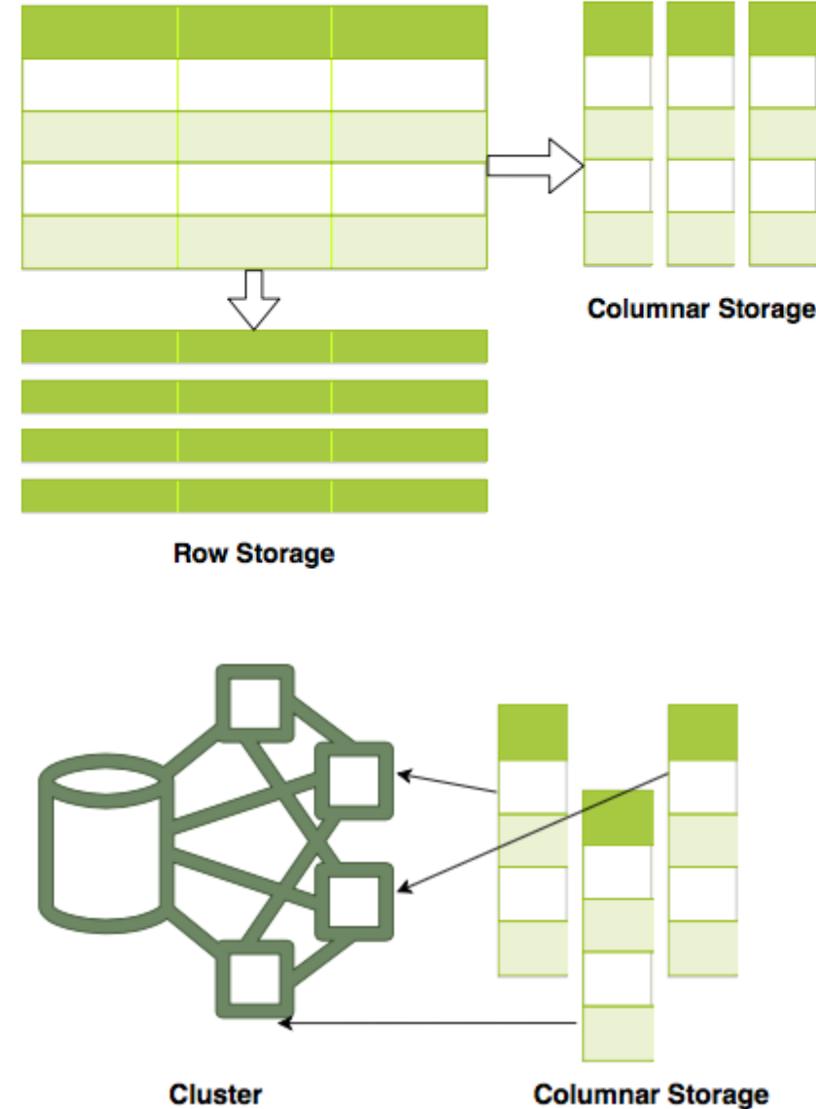


Redshift

# Relational Databases - OLAP vs OLTP

In 28  
Minutes

- OLAP and OLTP use **similar data structures**
- BUT **very different approach in how data is stored**
- **OLTP databases use row storage**
  - Each table row is stored together
  - Efficient for processing small transactions
- **OLAP databases use columnar storage**
  - Each table column is stored together
  - **High compression** - store petabytes of data efficiently
  - **Distribute data** - one table in multiple cluster nodes
  - **Execute single query across multiple nodes** - Complex queries can be executed efficiently

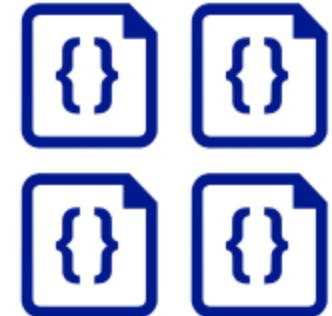




# Document Database

In 28  
Minutes

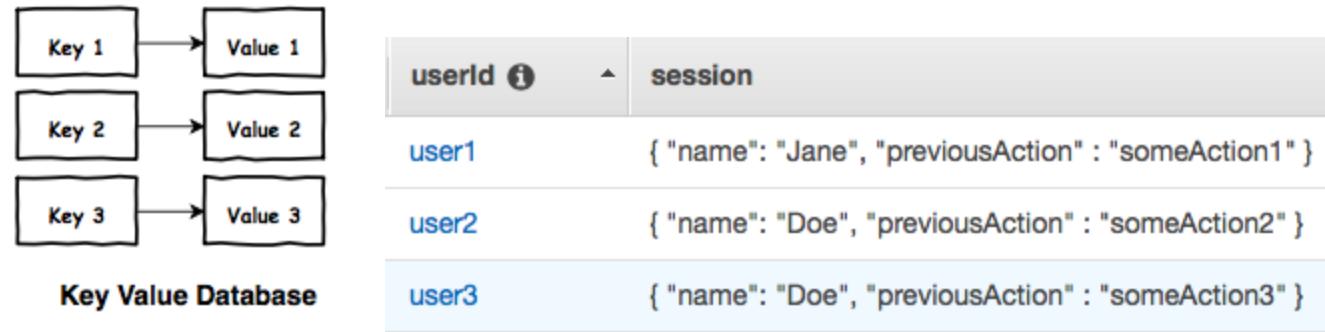
- Structure data the way your application needs it
- Create one table instead of dozens!
- Quickly evolving semi structured data (**schema-less**)
- Use cases : Content management, catalogs, user profiles
- Advantages : (Horizontally) Scalable to terabytes of data with millisecond responses upto millions of transactions per second
- Recommended AWS Managed Service
  - Amazon DynamoDB



Document Database

```
{  
  "id": 1,  
  "name": "Jane Doe",  
  "username": "abcdefgh",  
  "email": "someone@gmail.com",  
  "address": {  
    "street": "Some Street",  
    "suite": "Apt. 556",  
    "city": "Hyderabad",  
    "zipcode": "500018",  
    "geo": {  
      "lat": "-3.31",  
      "lng": "8.14"  
    },  
    "phone": "9-999-999-9999",  
    "website": "in28minutes.com",  
    "company": {  
      "name": "in28minutes"  
    }  
  }  
}
```





- Use a **simple key-value pair** to store data. Key is a unique identifier.
- Values can be objects, compound objects or simple data values
- **Advantages :** (Horizontally) Scalable to terabytes of data with millisecond responses upto millions of transactions per second
- Recommended AWS Managed Service - **Amazon DynamoDB** again
- **Use cases :** shopping carts, session stores, gaming applications and very high traffic web apps



- **Store and navigate** data with complex relationships
- **Use cases** : Social Networking Data (Twitter, Facebook), Fraud Detection
- Recommended AWS Managed Service - **Amazon Neptune**

# In-memory Databases

In 28  
Minutes

- Retrieving data from memory is much faster from retrieving data from disk
- In-memory databases like Redis deliver microsecond latency by storing **persistent data in memory**
- Recommended AWS Managed Service
  - **Amazon ElastiCache**
  - Supports Redis and Memcached
    - Redis is recommended for persistent data
    - Memcached is recommended for simple caches
- **Use cases :** Caching, session management, gaming leader boards, geospatial applications



ElastiCache

# Databases - Summary

In 28  
Minutes

Database Type	AWS Service	Description
Relational OLTP databases	Amazon RDS	Row storage Transactional usecases needing <b>predefined schema</b> and very <b>strong transactional</b> capabilities
Relational OLAP databases	Amazon Redshift	Columnar storage Reporting, analytics & intelligence apps needing <b>predefined schema</b>
Document & Key Databases	Amazon DynamoDB	Apps needing <b>quickly evolving</b> semi structured data ( <b>schema-less</b> ) <b>Scale to terabytes of data with millisecond responses upto millions of TPS</b> Content management, catalogs, user profiles, shopping carts, session stores and gaming applications

# Databases - Summary

In 28  
Minutes

Database Type	AWS Service	Description
Graph Databases	Amazon Neptune	Store and navigate data with <b>complex relationships</b> Social Networking Data (Twitter, Facebook), Fraud Detection
In memory databases/caches	Amazon ElastiCache	Applications needing <b>microsecond</b> responses <b>Redis</b> - persistent data <b>Memcached</b> - simple caches

# Databases - Questions

In 28  
Minutes

Scenario	Solution
A start up with quickly evolving tables	DynamoDB
Transaction application needing to process million transactions per second	DynamoDB
Very high consistency of data is needed while processing thousands of transactions per second	RDS
Cache data from database for a web application	Amazon ElastiCache
Relational database for analytics processing of petabytes of data	Amazon Redshift

# Amazon RDS (Relational Database Service)

In 28  
Minutes

- Do you want to manage the setup, backup, scaling, replication and patching of your relational databases?
  - Or do you want to use a managed service?
- Amazon RDS is a managed relational database service for OLTP use cases
- Supports:
  - Amazon Aurora
  - PostgreSQL
  - MySQL (InnoDB storage engine full supported)
  - MariaDB (Enhanced MySQL)
  - Oracle Database
  - Microsoft SQL Server



Amazon RDS

# Amazon RDS - Features

In 28  
Minutes

- Multi-AZ deployment (standby in another AZ)
- Read replicas:
  - Same AZ
  - Multi AZ (Availability+)
  - Cross Region(Availability++)
- Storage auto scaling (up to a configured limit)
- Automated backups (restore to point in time)
- Manual snapshots



Amazon RDS

# Amazon RDS - You vs AWS

In 28  
Minutes

- AWS is responsible for
  - Availability (according to your configuration)
  - Durability
  - Scaling (according to your configuration)
  - Maintenance (patches)
  - Backups
- You are responsible for
  - Managing database users
  - App optimization (tables, indexes etc)
- You CANNOT
  - SSH into database EC2 instances or setup custom software (NOT ALLOWED)
  - Install OS or DB patches. RDS takes care of them (NOT ALLOWED)

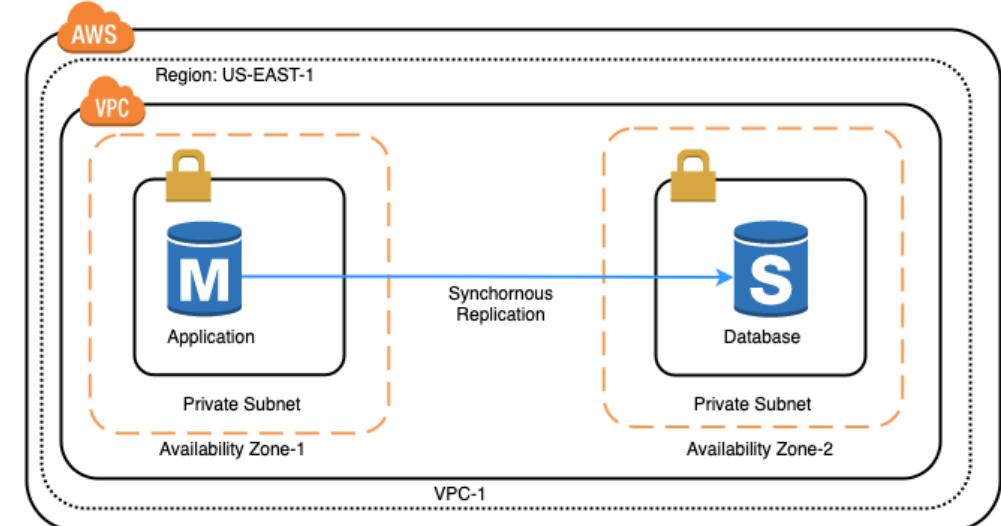


Amazon RDS

# Multi-AZ Deployments

In 28  
Minutes

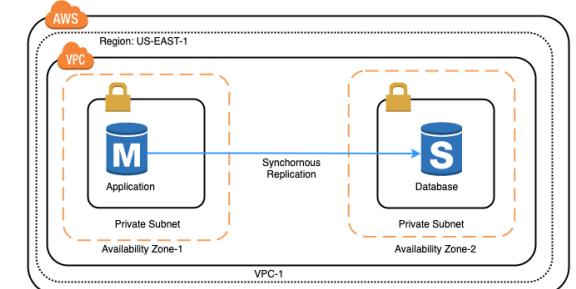
- Standby created in a different AZ
- **Synchronous replication** (strong consistency)
- Enhances durability, availability and fault tolerance of your database
- Multi-AZ makes **maintenance easy**
  - Perform maintenance (patches) on standby
  - Promote standby to primary
  - Perform maintenance on (old) primary
- **Avoid I/O suspension** when data is backed up (snapshots are taken from standby)



# Multi-AZ Deployments

In 28  
Minutes

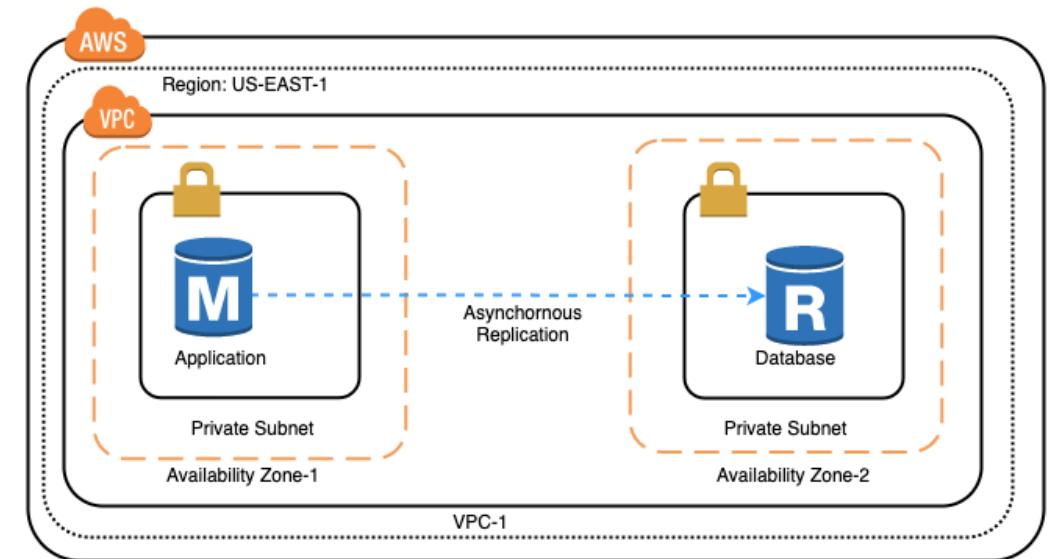
- No downtime when database is converted to Multi AZ
  - Increased latency until standby is ready
- Not allowed to connect to standby database directly
  - For example: Standby CANNOT be used to serve read traffic
  - Standby increases availability but does not improve scalability
- Automatic failover to standby if master has problems (compute, storage or networking)
  - CNAME record flipped to standby
  - Database performance issues (long running queries or deadlocks) will NOT cause a failover
- (Good Practice) Use DNS name of database in applications configuration



# Read Replicas

In 28  
Minutes

- Support **read-heavy database workloads** - reporting and data warehousing
- Can be in same or different AZ or different Region
- Your apps can connect to them
- Create read replica(s) of a read replica
- Uses **asynchronous replication**
  - Provides eventual consistency (from replica)
  - For higher consistency, read from master
- Need to be **explicitly deleted** (Not deleted when database is deleted)



# Read Replicas - Few Tips

In 28  
Minutes

- (Mandatory) Enable automatic backups before you can create read replicas
  - Set Backup Retention period to a value other than 0
- Reduce replication lag by using better compute and storage resources
- Maximum no of read replicas:
  - MySQL, MariaDB, PostgreSQL, and Oracle - 5
  - Aurora - 15
  - SQL Server does not support read replicas

# Multi-AZ vs Multi-Region vs Read replicas

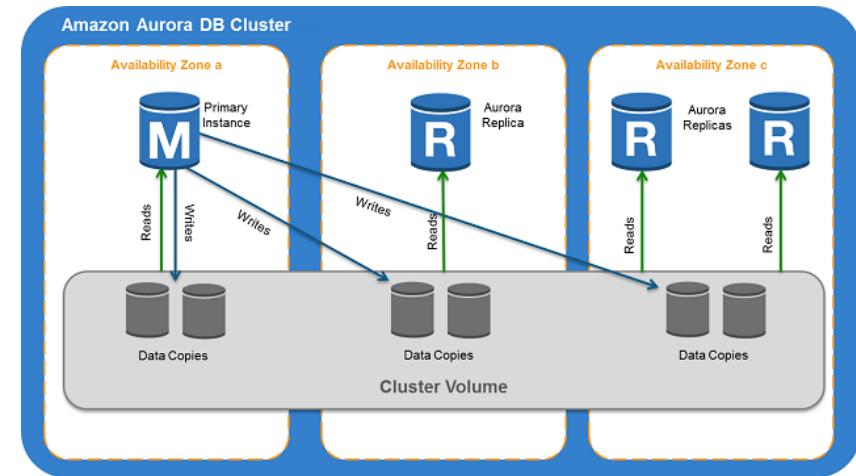
In 28  
Minutes

Feature	Multi-AZ deployments	Multi-Region Read Replicas	Multi-AZ Read replicas
Main purpose	High availability	Disaster recovery and local performance	Scalability
Replication	Synchronous (except for Aurora - Asynchronous)	Asynchronous	Asynchronous
Active	Only master (For Aurora - all)	All read replicas	All read replicas

# Amazon Aurora

In 28  
Minutes

- MySQL and PostgreSQL-compatible
- 2 copies of data each in a minimum of 3 AZ
- Up to 15 read replicas (Only 5 for MySQL)
- Provides "Global Database" option
  - Up to five read-only, secondary AWS Regions
    - Low latency for global reads
    - Safe from region-wide outages
  - Minimal lag time, typically less than 1 second
- Deployment Options
  - Single master (One writer and multiple readers)
  - Multi master deployment (multiple writers)
  - Serverless
- Uses cluster volume (multi AZ storage)



<https://docs.aws.amazon.com/AmazonRDS/latest>

# RDS - Scaling

In 28  
Minutes

- Vertical Scaling: Change DB instance type and scale storage
  - Storage and compute changes are typically applied during maintenance window
  - You can also choose to “apply-immediately”
  - RDS would take care of data migration
    - Takes a few minutes to complete
  - You can manually scale Aurora, MySQL, MariaDB, Oracle, and PostgreSQL engines to 64 TB
  - SQL Server can be scaled up to 16 TB
- Vertical Scaling: RDS also supports auto scaling storage
- Horizontal Scaling
  - Configure Read Replicas
  - For Aurora (Multi-master, Writer with multiple readers etc)



Amazon RDS

# RDS - Operations

In 28  
Minutes

- RDS console shows metrics upto a certain time period
- CloudWatch show historical data
- Configure CloudWatch alarms to alert when you near max capacity
- Enable Enhanced Monitoring to monitor slow queries
- Automatic backup during backup window (to Amazon S3)
  - Enables restore to point in time
  - Backups retained for 7 days by default (max - 35 days)
  - Elevated latency when snapshots are taken (except for Multi-AZ setup)
- Backup window used to apply patches
  - If you do not configure a 30 minute backup window, RDS chooses one randomly
- Achieve RPO of up to 5 minutes



Amazon RDS



Cloudwatch



AWS Config

# RDS - Security and Encryption

In 28  
Minutes

- Create in a VPC private subnet
- Use security groups to control access
- Option to use IAM Authentication with Aurora, MySQL and PostgreSQL
  - Use IAM roles and no need for passwords
- Enable encryption with keys from KMS
- When encryption is enabled
  - Data in the database, automated backups, read replicas and snapshots are all encrypted
- Data In-flight Encryption
  - Using SSL certificates



AWS KMS



Subnet



Amazon RDS



Security Group

# RDS - Costs - Key Elements

In 28  
Minutes

- DB instance hours - How many hours is the DB instance running?
- Storage (per GB per month) - How much storage have you provisioned for your DB instance?
- Provisioned IOPS per month - If you are using Amazon RDS Provisioned IOPS (SSD) Storage
- Backups and snapshot storage (across multi AZ) - More backups, More snapshots => More cost
- Data transfer costs

# Amazon RDS - When to use?

In 28  
Minutes

- Use Amazon RDS for transactional applications needing
  - Pre-defined schema
  - Strong transactional capabilities
  - Complex queries
- Amazon RDS is **NOT recommended** when
  - You need highly scalable massive read/write operations - for example millions of writes/second
    - Go for DynamoDB
  - When you want to upload files using simple GET/PUT REST API
    - Go for Amazon S3
  - When you need heavy customizations for your database or need access to underlying EC2 instances
    - Go for a custom database installation



Amazon RDS

Scenario	Solution
You want full control of OS or need elevated permissions	Consider going for a custom installation (EC2 + EBS)
You want to migrate data from an on-premise database to cloud database of the same type	Consider using AWS Database Migration Service
You want to migrate data from one database engine to another (Example : Microsoft SQL Server to Amazon Aurora)	Consider using AWS Schema Conversion Tool
What are retained when you delete a RDS database instance?	All automatic backups are deleted All manual snapshots are retained (until explicit deletion) (Optional) Take a final snapshot

# RDS - Scenarios

In 28  
Minutes

Scenario	Solution
How do you reduce global latency and improve disaster recovery?	Use multi region read replicas
How do you select the subnets a RDS instance is launched into?	Create DB Subnet groups
How can you add encryption to an unencrypted database instance?	Create a DB snapshot Encrypt the database snapshot using keys from KMS Create a database from the encrypted snapshot
Are you billed if you stop your DB instance?	You are billed for storage, IOPS, backups and snapshots. You are NOT billed for DB instance hours
I will need RDS for at least one year. How can I reduce costs?	Use Amazon RDS reserved instances.
Efficiently manage database connections	Use Amazon RDS Proxy Sits between client applications (including lambdas) and RDS

# Amazon DynamoDB

In 28  
Minutes

- Fast, scalable, distributed for any scale
- Flexible NoSQL Key-value & document database (schemaless)
- Single-digit millisecond responses for million of TPS
- Do not worry about scaling, availability or durability
  - Automatically partitions data as it grows
  - Maintains 3 replicas within the same region
- No need to provision a database
  - Create a table and configure read and write capacity (RCU and WCU)
  - Automatically scales to meet your RCU and WCU
- Provides an expensive serverless mode
- Use cases: User profiles, shopping carts, high volume read write applications



DynamoDB

# DynamoDB Tables

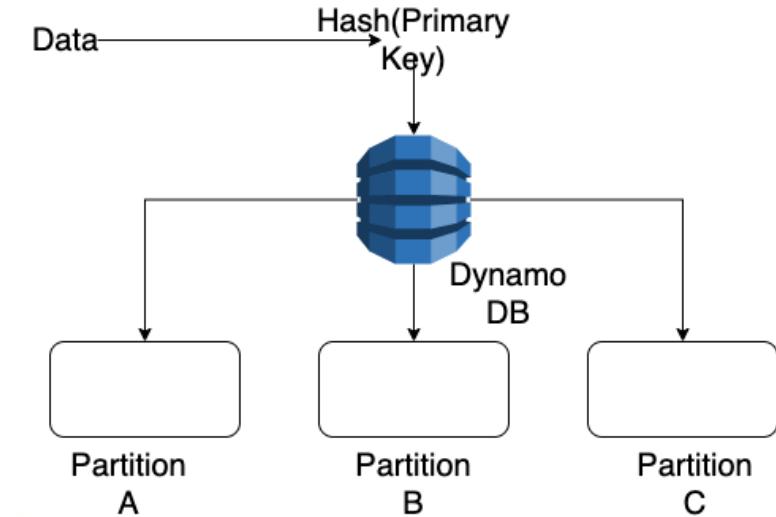
- Hierarchy : Table > item(s) > attribute (key value pair)
- Mandatory primary key
- Other than the primary key, tables are schemaless
  - No need to define the other attributes or types
  - Each item in a table can have distinct attributes
- Max 400 KB per item in table
  - Use S3 for large objects and DynamoDB for smaller objects

```
{  
    "id": 1,  
    "name": "Jane Doe",  
    "username": "abcdefgh",  
    "email": "someone@gmail.com",  
    "address": {  
        "street": "Some Street",  
        "suite": "Apt. 556",  
        "city": "Hyderabad",  
        "zipcode": "500018",  
        "geo": {  
            "lat": "-3.31",  
            "lng": "8.14"  
        }  
    },  
    "phone": "9-999-999-9999",  
    "website": "in28minutes.com",  
    "company": {  
        "name": "in28minutes"  
    }  
}
```

# DynamoDB - Keys

In 28  
Minutes

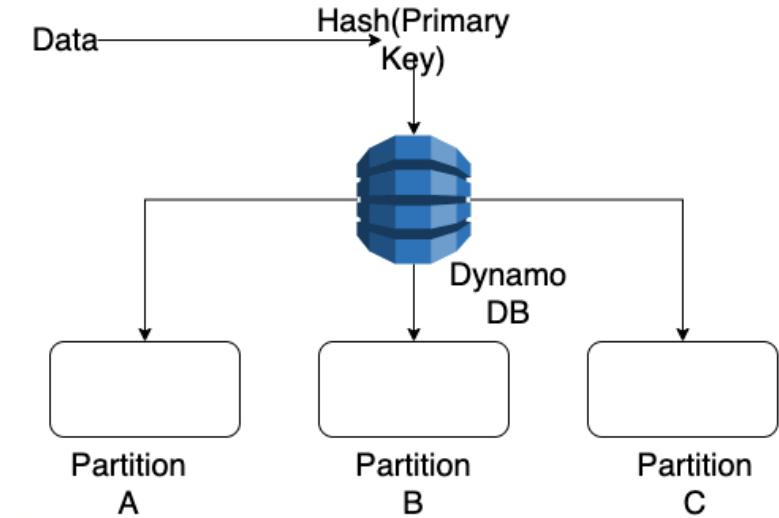
- Two parts:
  - (Mandatory) Partition key
  - (Optional) Sort key
- Primary key should be unique
- Partition key decides the partition (input to hash function)
- Same partition key items stored together (sorted by sort key)



# DynamoDB - Indexes

In 28  
Minutes

- (Optional) Secondary indexes to query on keys other than primary key
- Local secondary index
  - Same partition key as primary key but different sort key
  - Should be created at the table creation
- Global secondary index
  - Partition and sort key different from primary key
  - Can be added and removed at any point in time
  - Stored separately from the original table



# DynamoDB Query vs Scan

In 28  
Minutes

- **Query**
  - Search using a partition key attribute and a distinct value to search
  - Optional - sort key and filters
  - Results are sorted by primary key
  - Max 1 MB
- **Scan**
  - Reads every item in a table
  - Expensive compared to query
  - Returns all attributes by default
  - Supports paging above 1 MB
  - Filter items using expressions



DynamoDB

# DynamoDB Consistency Levels

In 28  
Minutes

- By default, eventually consistent (lag of about a second)
- Request for strongly consistent reads
  - Set `ConsistentRead` to true
  - Slow and more expensive
- Supports transactions
  - All-or-nothing changes to multiple items both within and across tables
  - Twice the cost



DynamoDB

# DynamoDB Read/Write Capacity Modes

In 28  
Minutes

- Provisioned
  - Provision read and write capacity
  - Dynamically adjustable
  - Unused capacity can be used in bursts
  - You are billed for the provisioned capacity irrespective of whether you make use of it or not
- On Demand
  - Truly serverless and expensive
  - For unknown workloads or traffic with huge spikes
  - Use On Demand only when your
    - Workloads are really spiky causing low utilization of Provisioned Capacity OR
    - Usage is very low (for example, in test environments) making manual adjustments expensive



DynamoDB

# DynamoDB Read/Write Capacity Used

In 28  
Minutes

- Capacity used depends on size of item, read consistency, transactions etc
- 1 capacity unit to read 4 KB or smaller (more for bigger items)
- 1 capacity unit to write 1 KB or smaller (more for bigger items)
- Twice the capacity for a strongly consistent or transactional requests
- On-demand RCU is almost 8 times the cost of Provisioned RCU
- Example: \$0.2500 per million vs \$0.0361 per million



DynamoDB

# DynamoDB - Operations

In 28  
Minutes

- Performance Monitoring - CloudWatch
- Alerts on RCU, WCU and Throttle Requests - CloudWatch Alarms
- Migrate data from RDS or MongoDB to DynamoDB - AWS Database Migration Service
- (Feature) Enable point-in-time recovery (max 35 days)
- Use Time to Live (TTL) to automatically expire items



DynamoDB

# DynamoDB - IAM and Encryption

In 28  
Minutes

- Server-side encryption in integration with keys from KMS
  - Always enabled
  - Automatically encrypts tables, DynamoDB streams, and backups
- Client-side encryption with DynamoDB Encryption Client
  - You can manage your keys with KMS or CloudHSM
- Use IAM roles to provide EC2 instances or AWS services access to DynamoDB tables
  - Predefined policies available for DynamoDB
    - AmazonDynamoDBReadOnlyAccess
    - AmazonDynamoDBFullAccess etc
  - Fine-grained control at the individual item level



DynamoDB

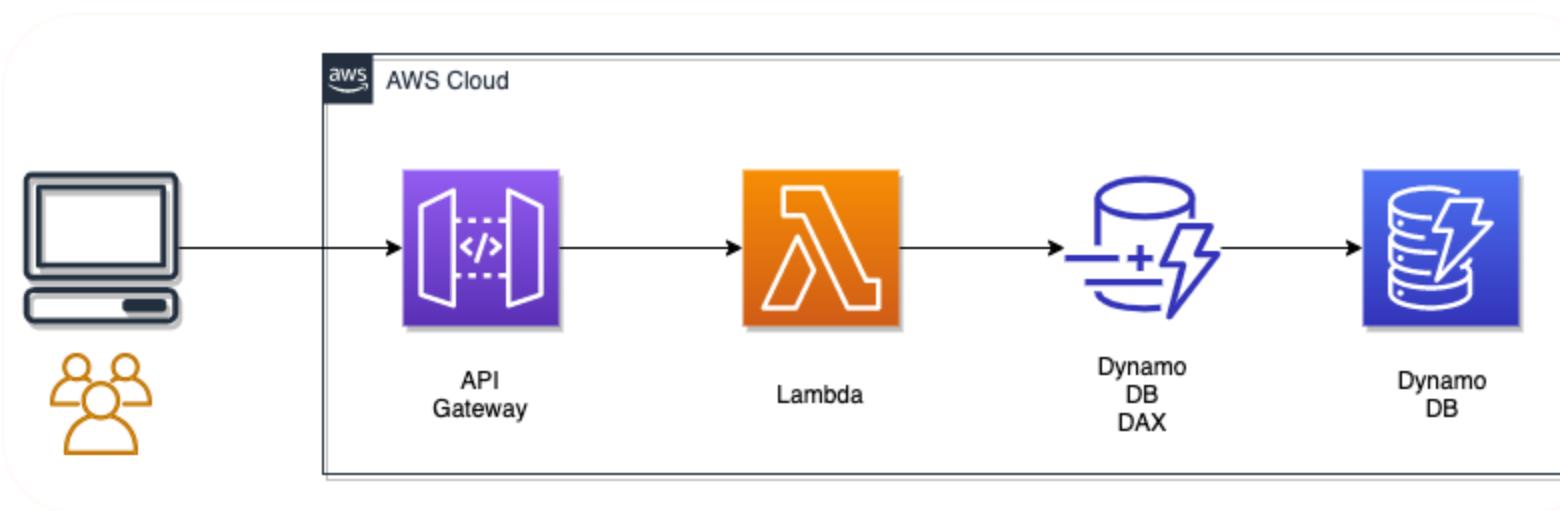
# DynamoDB vs RDS

In 28  
Minutes

Feature	DynamoDB	RDS
Scenario	Millisecond latency with millions of TPS	Stronger consistency (schema) and transactional capabilities
Schema	Schemaless (needs only a primary key - Great for use cases where your schema is evolving)	Well-defined schema with relationships
Data Access	Using REST API provided by AWS using AWS SDKs or AWS Management Console or AWS CLI	SQL queries
Complex Data Queries Involving Multiple Tables	Difficult to run	Run complex relational queries with multiple entities
Scaling	No upper limits	64 TB
Consistency	Typically lower consistency	Typically higher consistency

# DynamoDB Accelerator (DAX)

In 28  
Minutes



- In-memory caching for DynamoDB providing microsecond response times
  - Typical DynamoDB response times - single-digit milliseconds
- Very few changes needed to connect to DAX
  - Can reduce your costs by saving your read capacity units
- Not recommended
  - If you need strongly consistent reads or
  - Your application is write-intensive with very few reads

# Amazon ElastiCache

In 28  
Minutes

- Managed service providing highly scalable and low latency in-memory data store
- Used for distributed caching
- Two Options:
  - Redis
  - Memcached



ElastiCache

# Amazon ElastiCache for Redis

In 28  
Minutes

- Highly scalable and low latency in-memory data store
- Can be used as a cache, database or message broker
- Automatic failover with Multi-AZ deployments (if enabled)
- Supports backup and restore
- Supports encryption at-rest (KMS) and in-transit
- Use cases:
  - Caching
  - Session Store
  - Chat and Messaging
  - Gaming Leader boards
  - Geospatial Apps (Ride hailing, restaurant recommendations)
  - Queues



ElastiCache

# Amazon ElastiCache for Redis - Cluster

In 28  
Minutes

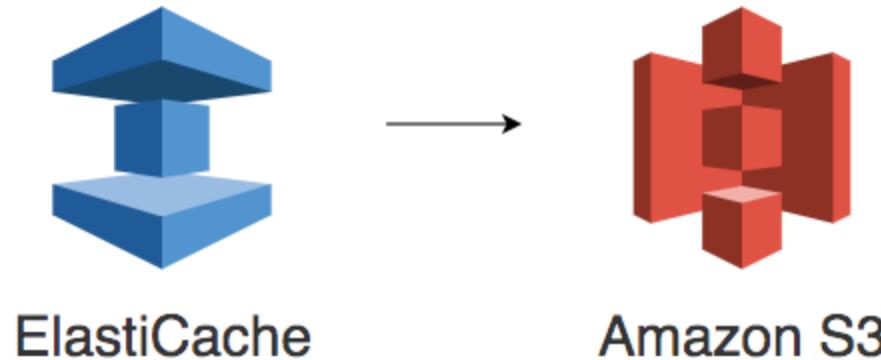
- Shard - collection of one or more nodes
- One node acts as read/write primary
- Other nodes act as read replicas (up to five read replicas)
- In case of failure:
  - Primary node is replaced
  - If Multi-AZ replication group is enabled, read replica is promoted to primary
  - DNS entry is updated



ElastiCache

# ElastiCache Redis - Backup and Snapshot

In 28  
Minutes



- Uses native backup feature of Redis (stored to S3)
  - Recommended to perform snapshot against read replicas
  - You can schedule snapshots
    - Configure backup window and
    - Days of backup you want to store
  - Manual snapshots are available until they are manually deleted

# Amazon ElastiCache for Memcached

In 28  
Minutes

- Simple caching layer intended for use in speeding up dynamic web applications
  - Pure cache
  - Non-persistent
  - Simple key-value storage
- Ideal front-end for data stores like RDS or DynamoDB
- Can be used as a transient session store
- Create upto 20 cache nodes
- Use Auto Discovery to discover cache nodes



ElastiCache

# Amazon ElastiCache for Memcached - Limitations

In 28  
Minutes

- Backup and restore NOT supported
- Does not support encryption or replication
- Does not support snapshots
  - When a node fails, all data in the node is lost
  - Reduce impact of failure by using large number of small nodes



ElastiCache

# ElastiCache Memcached vs Redis

In 28  
Minutes

- Use ElastiCache Memcached for
  - Low maintenance simple caching solution
  - Easy horizontal scaling with auto discovery
- Use ElastiCache Redis for
  - Persistence
  - Publish subscribe messaging
  - Read replicas and failover
  - Encryption



ElastiCache

# CloudTrail, Config & CloudWatch

- Track events, API calls, changes made to your AWS resources:
  - Who made the request?
  - What action was performed?
  - What are the parameters used?
  - What was the end result?
- (USE CASE) Compliance with regulatory standards
- (USE CASE) Troubleshooting. Locate a missing resource
- Delivers log files to S3 and/or Amazon cloud watch logs log group ( S3 is default )
- You can setup SNS notifications for log file delivery

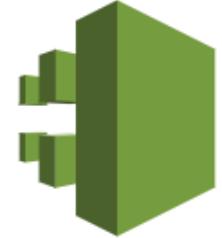


AWS CloudTrail

# AWS Cloud Trail Types

In 28  
Minutes

- Multi Region Trail
  - One trail of all AWS regions
  - Events from all regions can be sent to one CloudWatch logs log group
- Single Region Trail
  - Only events from one region
  - Destination S3 bucket can be in any region



AWS CloudTrail

# AWS Cloud Trail - Good to know

In 28  
Minutes

- Log files are automatically encrypted with Amazon S3 SSE
- You can configure S3 Lifecycle rules to archive or delete log files
- Supports log file integrity
  - You can prove that a log file has not been altered



AWS CloudTrail

- **Auditing**
  - Create a complete inventory of your AWS resources
- **Resource history and change tracking**
  - Find how a resource was configured at any point in time
  - Configuration of deleted resources would be maintained
  - Delivers history file to S3 bucket every 6 hours
  - Take configuration snapshots when needed
- **Governance**
  - Customize Config Rules for specific resources or for entire AWS account
  - Continuously evaluate compliance against desired configuration
  - Get a SNS notification for every configuration change
- **Consistent rules and compliance across AWS accounts:**
  - Group Config Rules and Remediation Actions into Conformance Packs



AWS Config

# Predefined Config Rule Examples (80+)

In 28  
Minutes

- **alb-http-to-https-redirection-check** - Checks whether HTTP to HTTPS redirection is configured on all HTTP listeners of Application Load Balancers
- **ebs-optimized-instance** - Checks whether EBS optimization is enabled for your EC2 instances that can be EBS-optimized
- **ec2-instance-no-public-ip** - Do EC2 instances have public IPs?
- **encrypted-volumes** - Are all EC2 instance attached EBS volumes encrypted?
- **eip-attached** - Are all Elastic IP addresses used?
- **restricted-ssh** - Checks whether security groups that are in use disallow unrestricted incoming SSH traffic



AWS Config

# AWS Config Rules

In 28  
Minutes

- (Feature) Create Lambda functions with your custom rules
- (Feature) You can setup auto remediation for each rule
  - Take immediate action on a non compliant resource
  - (Example) Stop EC2 instances without a specific tag!
- Enable AWS Config to use the rules
  - No Free Tier
  - More rules to check => More \$\$\$\$



AWS Config

# AWS Config + AWS CloudTrail

In 28  
Minutes



AWS Config



AWS CloudTrail

- AWS Config
  - What did my AWS resource look like?
- AWS CloudTrail
  - Who made an API call to modify this resource?

# Monitoring AWS with Amazon CloudWatch

In 28  
Minutes

- Monitoring and observability service
- Collects monitoring and operational data in the form of logs, metrics, and events
- Set alarms, visualize logs, take automated actions and troubleshoot issues
- Integrates with more than 70 AWS services:
  - Amazon EC2
  - Amazon DynamoDB
  - Amazon S3
  - Amazon ECS
  - AWS Lambda
  - and ....



Cloudwatch

# Amazon CloudWatch Logs

In 28  
Minutes

- Monitor and troubleshoot using system, application and custom log files
- Real time application and system monitoring
  - Monitor for patterns in your logs and trigger alerts based on them
  - Example : Errors in a specific interval exceed a certain threshold
- Long term log retention
  - Store logs in CloudWatch Logs for as long as you want (configurable - default:forever)
  - Or archive logs to S3 bucket (Typically involves a delay of 12 hours)
  - Or stream real time to Amazon Elasticsearch Service (Amazon ES) cluster using CloudWatch Logs subscription



Cloudwatch

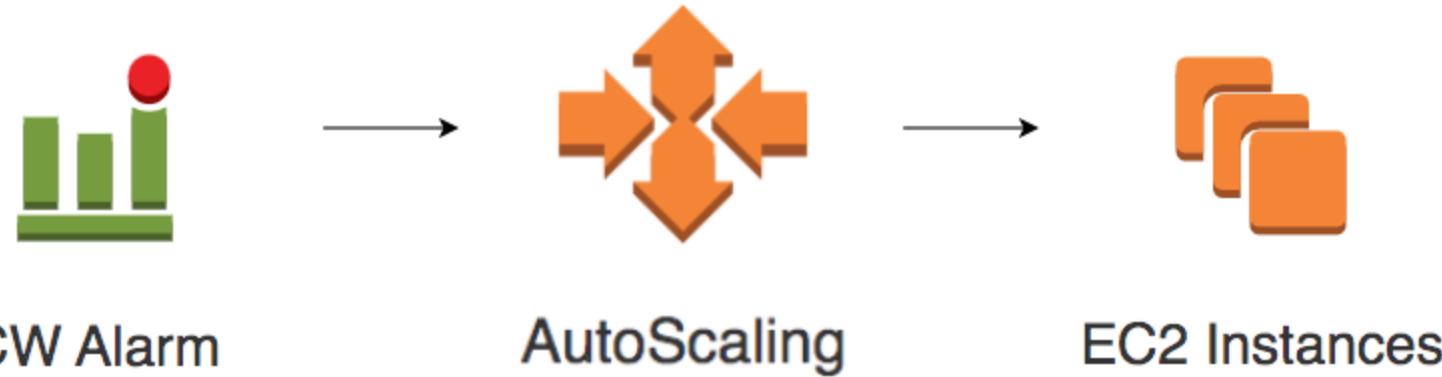
# Amazon CloudWatch Logs

In 28  
Minutes

- **CloudWatch Logs Agent**
  - Installed on ec2 instances to move logs from servers to CloudWatch logs
- **CloudWatch Logs Insights**
  - Write queries and get actionable insights from your logs
- **CloudWatch Container Insights**
  - Monitor, troubleshoot, and set alarms for your containerized applications running in EKS, ECS and Fargate

# Amazon CloudWatch Alarms

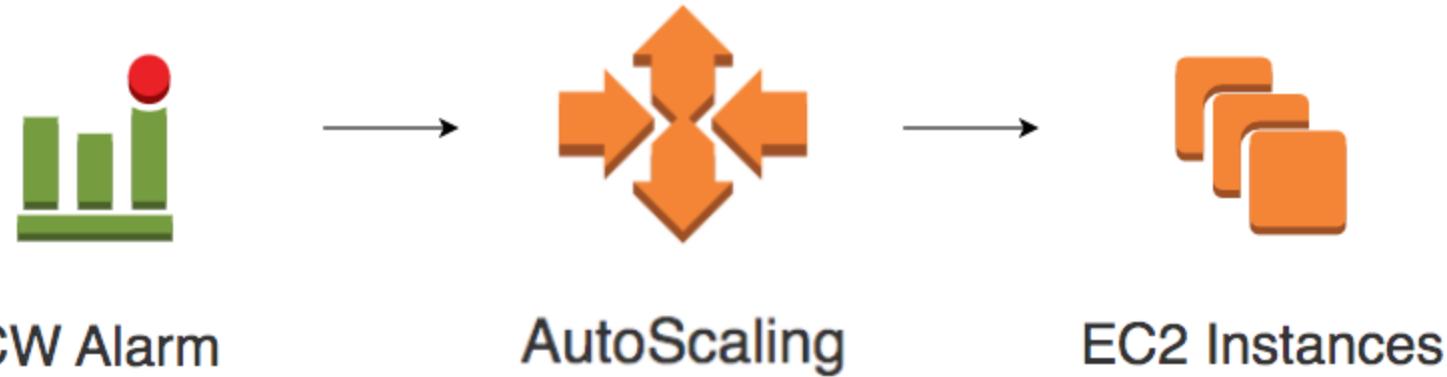
In 28  
Minutes



- Create alarms based on:
  - Amazon EC2 instance CPU utilization
  - Amazon SQS queue length
  - Amazon DynamoDB table throughput or
  - Your own custom metrics

# Amazon CloudWatch Alarms

In 28  
Minutes



- Take immediate action:
  - Send a SNS event notification
    - Send an email using SNS
  - Execute an Auto Scaling policy

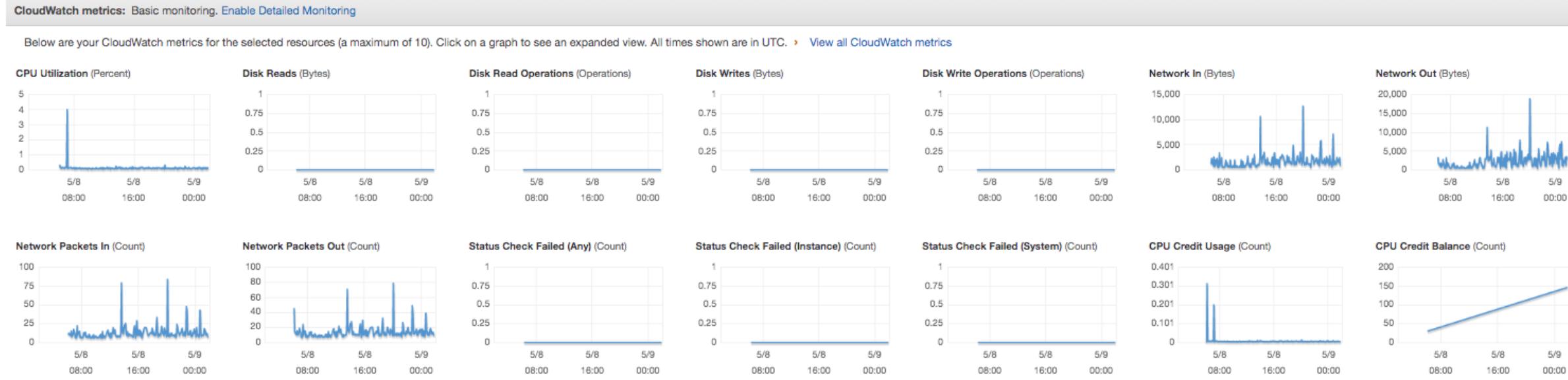
# Amazon CloudWatch Alarm - Example

In 28  
Minutes

- You set a CPU Utilization alarm on EC2 instance with a threshold of 80% over 3 periods of 10 minutes. If CPU utilization is 90% for 20 minutes, does the alarm get triggered?
  - No

# Amazon CloudWatch Dashboards

In 28  
Minutes



- Create auto refreshed graphs around all CloudWatch metrics
- Automatic Dashboards are available for most AWS services and resources
- Each Dashboard can have graphs from multiple regions

# Amazon CloudWatch Events

In 28  
Minutes

- Enable you to take immediate action based on events on AWS resources
  - Call a AWS Lambda function when an EC2 instance starts
  - Send event to an Amazon Kinesis stream when an Amazon EBS volume is created
  - Notify an Amazon SNS topic when an Auto Scaling event happens
- Schedule events - Use Unix cron syntax
  - Schedule a call to Lambda function every hour
  - Send a notification to Amazon SNS topic every 3 hours



Cloudwatch

# Decoupling Applications with SQS, SNS and MQ

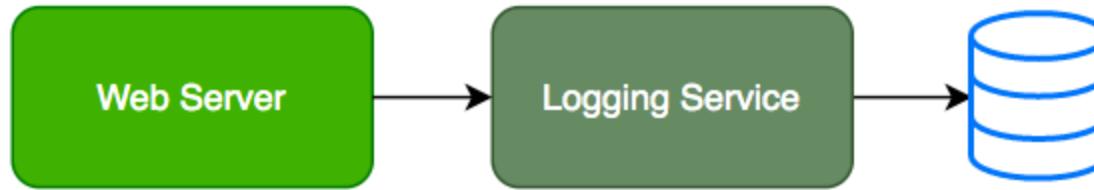
# Need for Asynchronous Communication

In 28  
Minutes

- Why do we need asynchronous communication?

# Synchronous Communication

In 28  
Minutes



- Applications on your web server make synchronous calls to the logging service
- What if your logging service goes down?
  - Will your applications go down too?
- What if all of sudden, there is high load and there are lots of logs coming in?
  - Log Service is not able to handle the load and goes down very often

# Asynchronous Communication - Decoupled

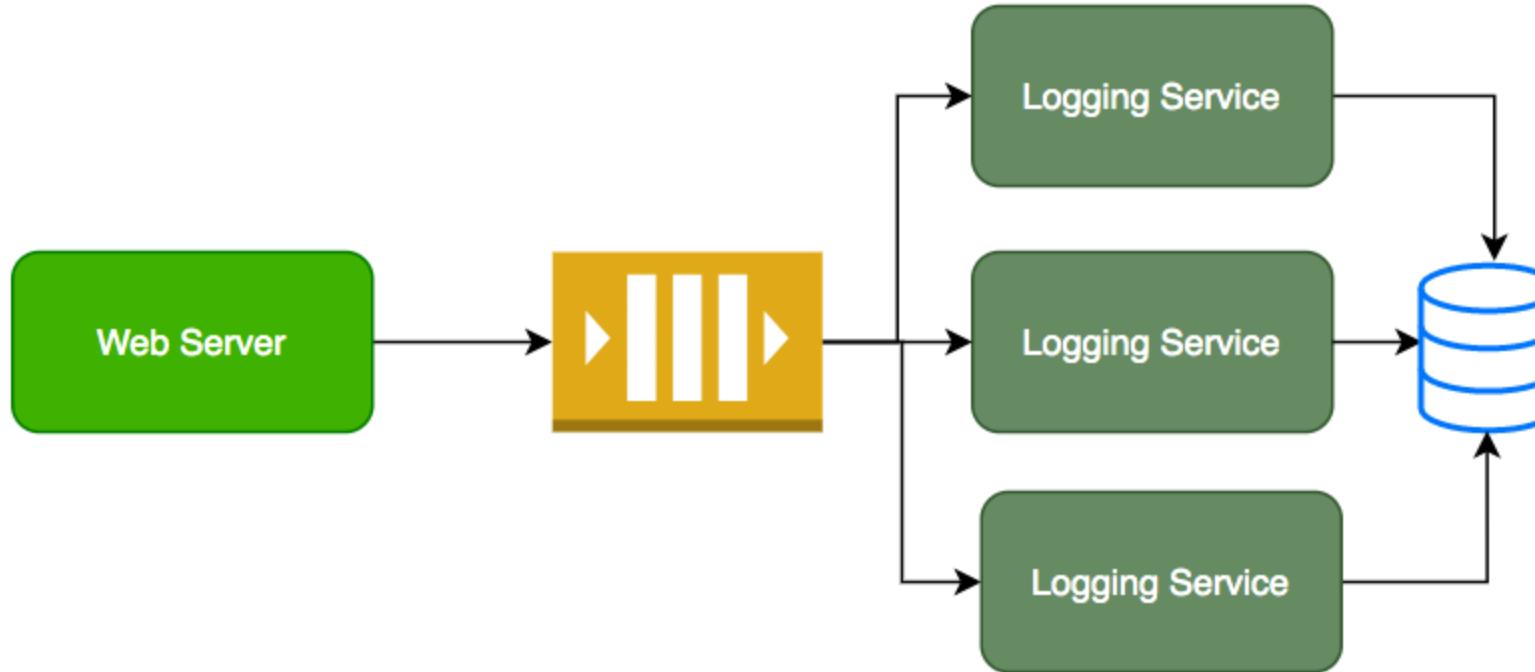
In 28  
Minutes



- Create a queue or a topic
- Your applications put the logs on the queue
- They would be picked up when the logging service is ready
- Good example of decoupling!

# Asynchronous Communication - Scale up

In 28  
Minutes

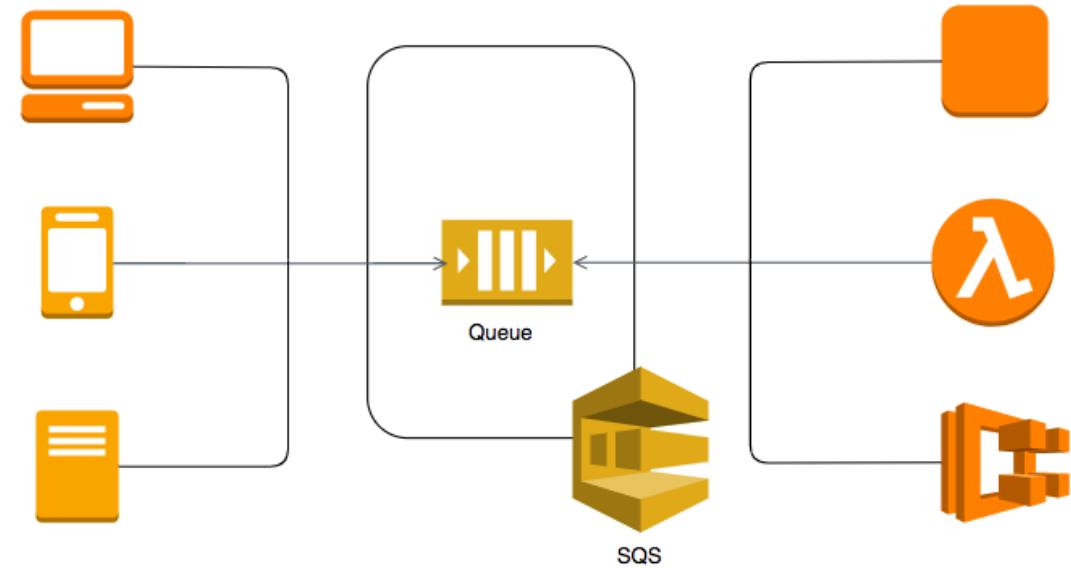


- You can have multiple logging service instances reading from the queue!

# Asynchronous Communication - Pull Model - SQS

In 28  
Minutes

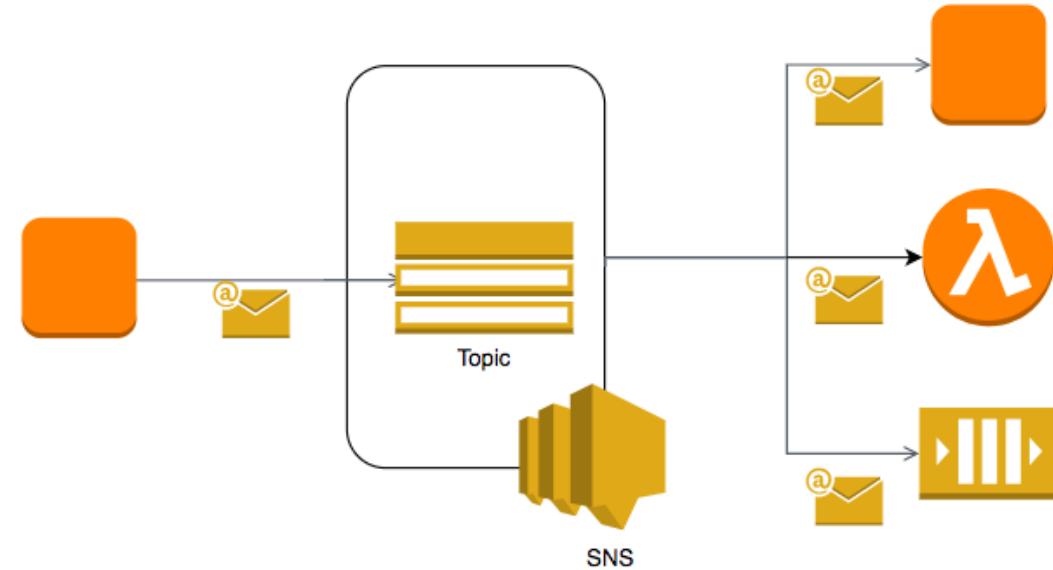
- Producers put messages on the queue
- Consumers poll on the queue
  - Only one of the consumers will successfully process a given message
- Scalability
  - Scale consumer instances under high load
- Availability
  - Producer up even if a consumer is down
- Reliability
  - Work is not lost due to insufficient resources
- Decoupling
  - Make changes to consumers without effect on producers worrying about them



# Asynchronous Communication - Push Model - SNS

In 28  
Minutes

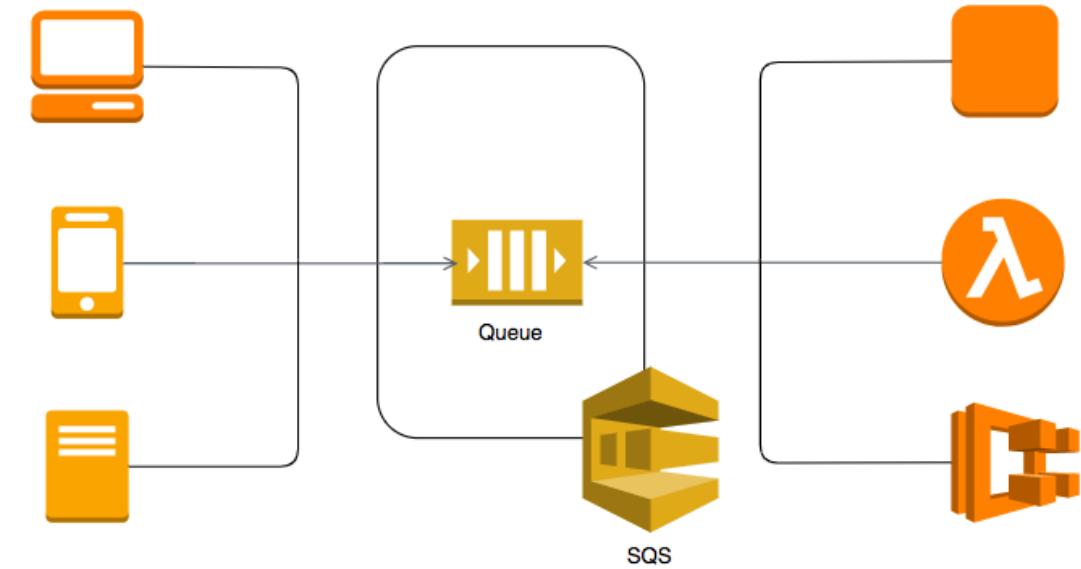
- Subscribers subscribe to a topic
- Producers send notifications to a topic
  - Notification sent out to all subscribers
- Decoupling
  - Producers don't care about who is listening
- Availability
  - Producer up even if a subscriber is down



# Simple Queuing Service

In 28  
Minutes

- Reliable, scalable, fully-managed message queuing service
- High availability
- Unlimited scaling
  - Auto scale to process billions of messages per day
- Low cost (Pay for use)



# Standard and FIFO Queues

In 28  
Minutes

- Standard Queue
  - Unlimited throughput
  - BUT NO guarantee of ordering (Best-Effort Ordering)
  - and NO guarantee of exactly-once processing
    - Guarantees at-least-once delivery (some messages can be processed twice)
- FIFO (first-in-first-out) Queue
  - First-In-First-out Delivery
  - Exactly-Once Processing
  - BUT throughput is lower
    - Up to 300 messages per second (300 send, receive, or delete operations per second)
    - If you batch 10 messages per operation (maximum), up to 3,000 messages per second
- Choose
  - Standard SQS queue if throughput is important
  - FIFO Queue if order of events is important

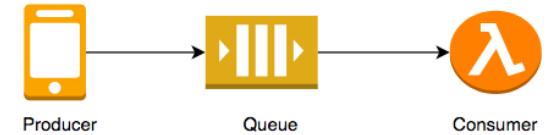


Amazon SQS

# Sending and receiving a SQS Message - Best case scenario

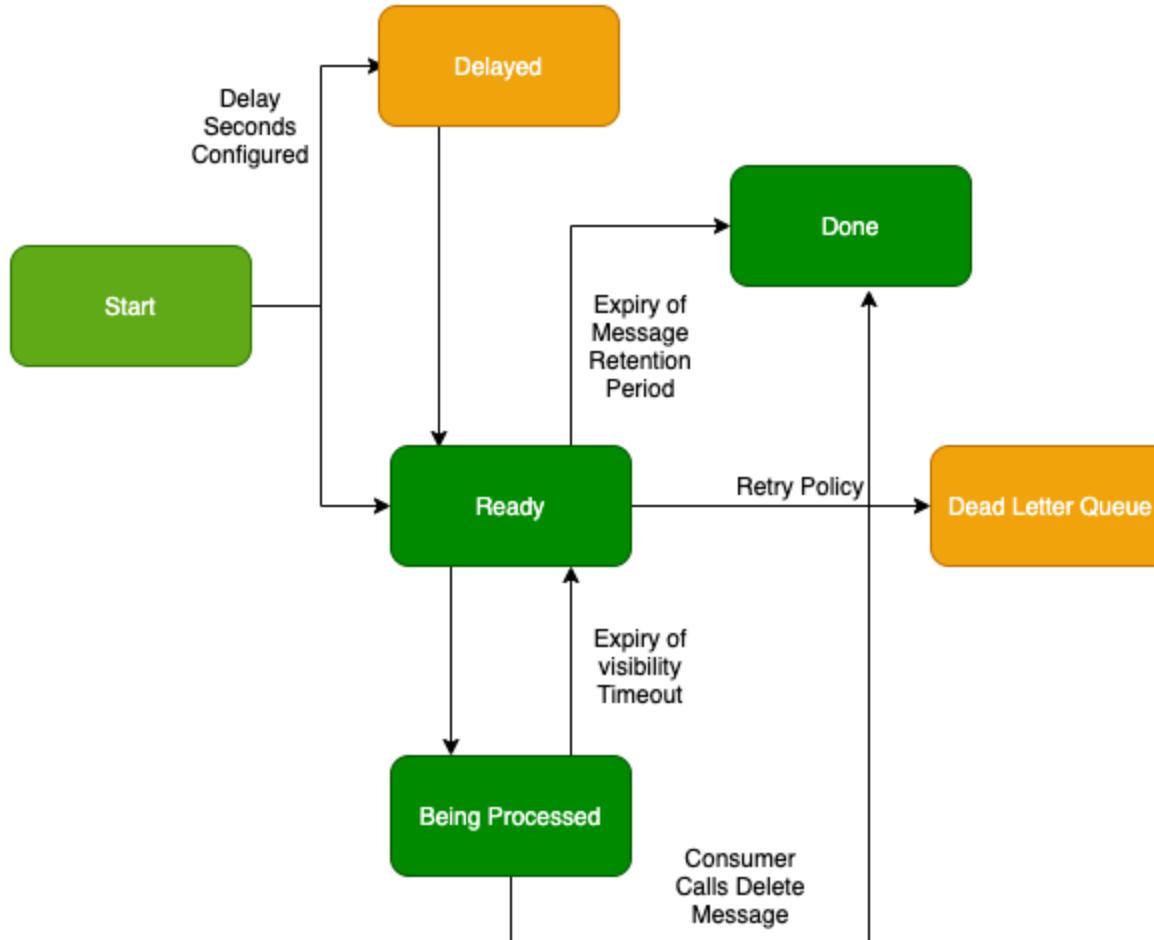
28  
Minutes

- Producer places message on queue
  - Receives globally unique message ID ABCDEFGHIJ (used to track the message)
- Consumer polls for messages
  - Receives the message ABCDEFGHIJ along with a receipt handle XYZ
- Message remains in the queue while the consumer processes the message
  - Other consumers will not receive ABCDEFGHIJ even if they poll for messages
- Consumer processes the message successfully
  - Calls delete message (using receipt handle XYZ)
  - Message is removed from the queue



# Simple Queuing Service Lifecycle of a message

In 28  
Minutes



# SQS - Auto Scaling

In 28  
Minutes



- Use target tracking scaling policy
- Use a SQS metric like ApproximateNumberOfMessages

# SQS Queue - Important configuration

In 28  
Minutes

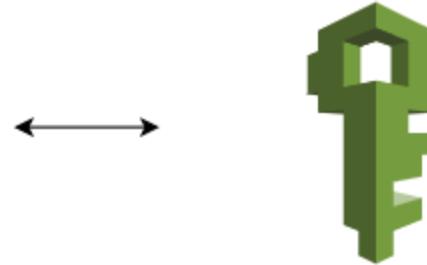
Configuration	Description
<b>Visibility timeout</b>	Other consumers will not receive a message being processed for the configured time period (default - 30 seconds, min - 0, max - 12 hours) Consumer processing a message can call ChangeMessageVisibility to increase visibility timeout of a message (before visibility timeout)
<b>DelaySeconds</b>	Time period before a new message is visible on the queue Delay Queue = Create Queue + Delay Seconds default - 0, max - 15 minutes Can be set at Queue creation or updated using SetQueueAttributes Use message timers to configure a message specific DelaySeconds value
<b>Message retention period</b>	Maximum period a message can be on the queue Default - 4 days, Min - 60 seconds, Max - 14 days
<b>MaxReceiveCount</b>	Maximum number of failures in processing a message

# Simple Queuing Service Security

In 28  
Minutes



Amazon SQS



AWS IAM

- You can provide access to other AWS resources to access SQS using IAM roles (EC2 -> SQS)
- By default only the queue owner is allowed to use the queue
  - Configure SQS Queue Access Policy to provide access to other AWS accounts

# SQS - Scenarios

In 28  
Minutes

Scenario	Result
Consumer takes more than visibility timeout to process the message	Message is visible on queue after visibility timeout and another consumer might receive the message
Consumer calls <code>ChangeMessageVisibility</code> before visibility timeout	Visibility timeout is extended to requested time
DelaySeconds is configured on the queue	Message is delayed for DelaySeconds before it is available
Receiver wants to decide how to handle the message without looking at message body	Configure Message Attributes

# SQS - Scenarios

In 28  
Minutes

## Scenario

How to reduce number of API calls to SQS?

Your receive messages and start processing them after a week. You see that some messages are not processed at all!

Give high priority to premium customers

## Result

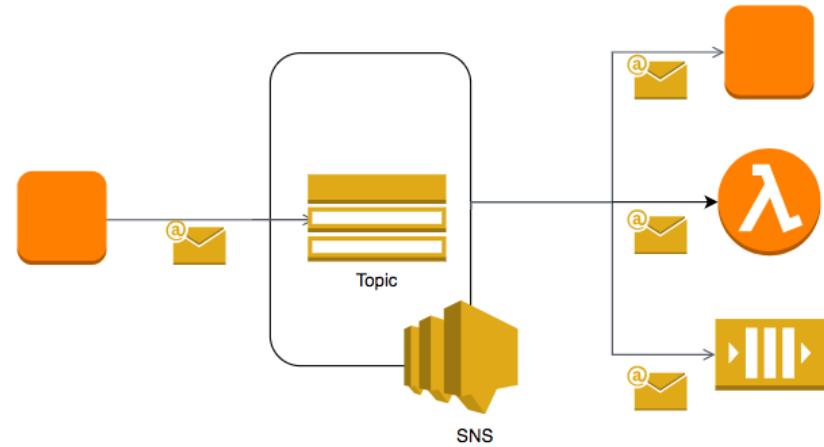
Use Long Polling - When looking for messages, you can specify a WaitTimeSeconds upto 20 seconds

Exceeded message retention period. Default message retention period is 4 days. Max 14 days.

Create separate queues for free and premium customers

# Amazon Simple Notification Service(SNS)

- Publish-Subscribe (pub-sub) paradigm
- Broadcast asynchronous event notifications
- Simple process
  - Create an SNS Topic
  - Subscribers can register for a Topic
  - When an SNS Topic receives an event notification (from publisher), it is broadcast to all Subscribers
- Use Cases : Monitoring Apps, workflow systems, mobile apps



# Amazon Simple Notification Service(SNS)

In 28  
Minutes

- Provides mobile and enterprise messaging web services
  - Push notifications to Apple, Android, FireOS, Windows devices
  - Send SMS to mobile users
  - Send Emails
- REMEMBER : SNS does not need SQS or a Queue
- You can allow access to other AWS accounts using AWS SNS generated policy



Amazon SNS

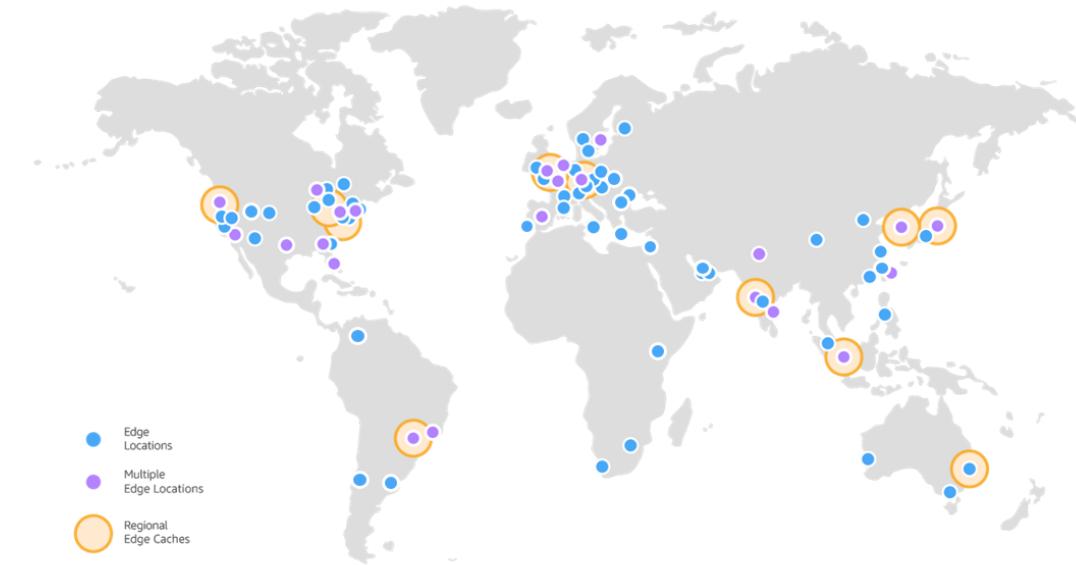
- Managed message broker service for Apache ActiveMQ
- (Functionally) Amazon MQ = Amazon SQS (Queues) + Amazon SNS (Topics)
  - BUT with restricted scalability
- Supports traditional APIs (JMS) and protocols (AMQP, MQTT, OpenWire, and STOMP)
  - Easy to migrate on-premise applications using traditional message brokers
  - Start with Amazon MQ as first step and slowly re-design apps to use Amazon SQS and/or SNS
- Scenario: An enterprise uses AMQP (standard message broker protocol). They want to migrate to AWS without making code changes
  - Recommend Amazon MQ

# Routing and Content Delivery

# Content Delivery Network

In 28  
Minutes

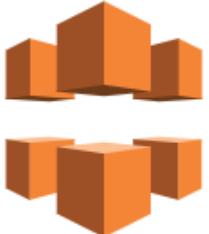
- You want to deliver content to your global audience
- Content Delivery Networks distribute content to multiple edge locations around the world
- AWS provides 200+ edge locations around the world
- Provides high availability and performance



# Amazon CloudFront

In 28  
Minutes

- How do you enable serving content directly from AWS edge locations?
  - Amazon CloudFront (one of the options)
- Serve users from nearest edge location (based on user location)
- Source content can be from S3, EC2, ELB and External Websites
- If content is not available at the edge location, it is retrieved from the origin server and cached
- No minimum usage commitment
- Provides features to protect your private content

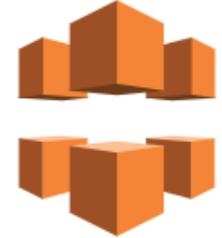


CloudFront

# Amazon CloudFront

In 28  
Minutes

- Use Cases
  - Static web apps. Audio, video and software downloads. Dynamic web apps
  - Support media streaming with HTTP and RTMP
- Integrates with
  - AWS Shield to protect from DDoS attacks
  - AWS Web Application Firewall (WAF) to protect from SQL injection, cross-site scripting, etc
- Cost Benefits
  - Zero cost for data transfer between S3 and CloudFront
  - Reduce compute workload for your EC2 instances

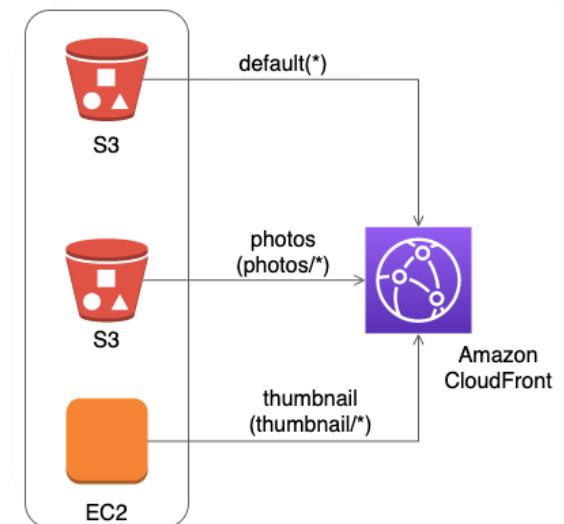


CloudFront

# Amazon CloudFront Distribution

In 28  
Minutes

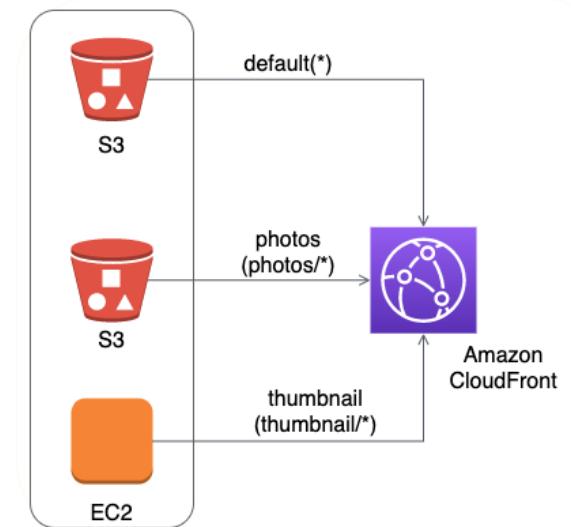
- Create a CloudFront Distribution to distribute your content to edge locations
  - DNS domain name - example abc.cloudfront.com
  - Origins - Where do you get content from? S3, EC2, ELB, External Website
  - Cache-Control
    - By default objects expire after 24 hours
    - Customize min, max, default TTL in CloudFront distribution
    - (For file level customization) Use Cache-Control max-age and Expires headers in origin server
- You can configure CloudFront to only use HTTPS (or) use HTTPS for certain objects
  - Default is to support both HTTP and HTTPS
  - You can configure CloudFront to redirect HTTP to HTTPS



# Amazon CloudFront - Cache Behaviors

In 28  
Minutes

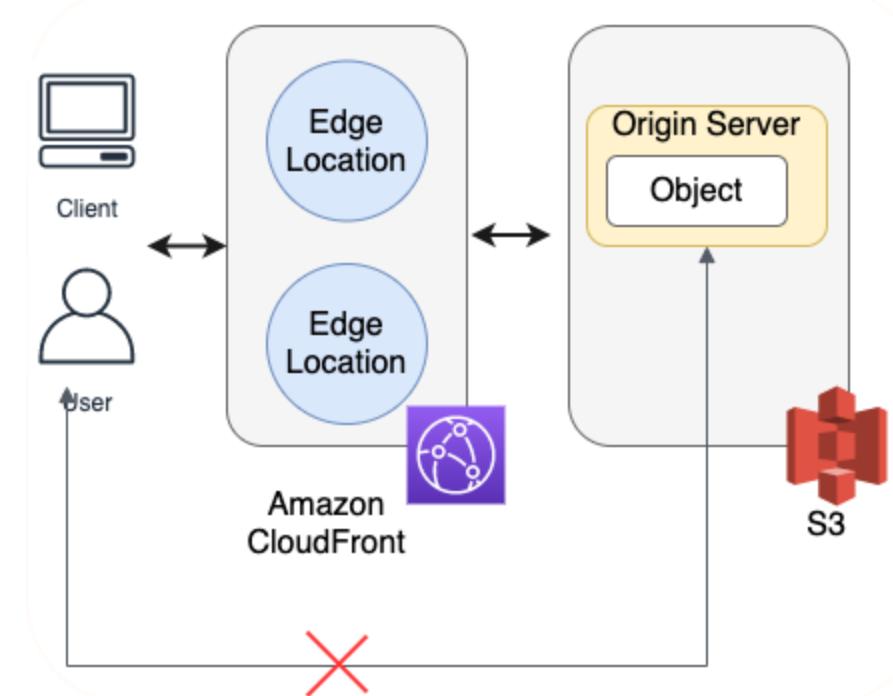
- Configure different CloudFront behavior for different URL path patterns from same origin
  - Path pattern(can use wild cards - \*.php, \*.jsp),
  - Do you want to forward query strings?
  - Should we use https?
  - TTL



# Amazon CloudFront - Private content

In 28  
Minutes

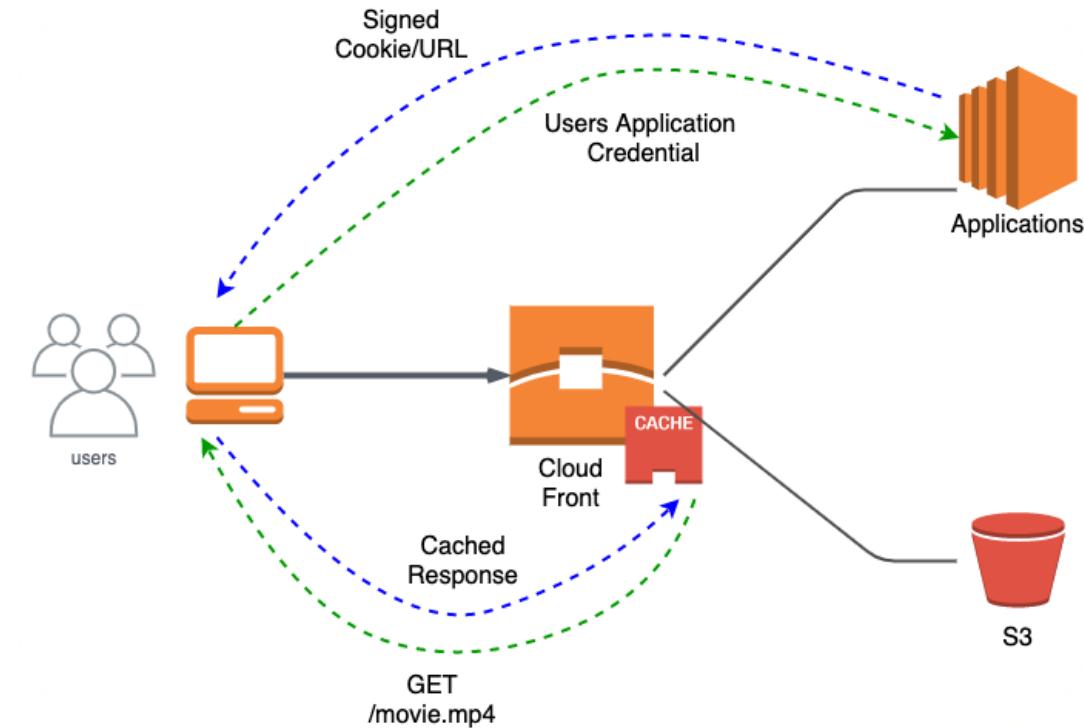
- Signed URLs
- Signed cookies using key pairs
- Origin Access Identities(OAI)
  - Ensures that only CloudFront can access S3
  - Allow access to S3 only to a special CloudFront user



# Amazon CloudFront - Signed URLs and Cookies

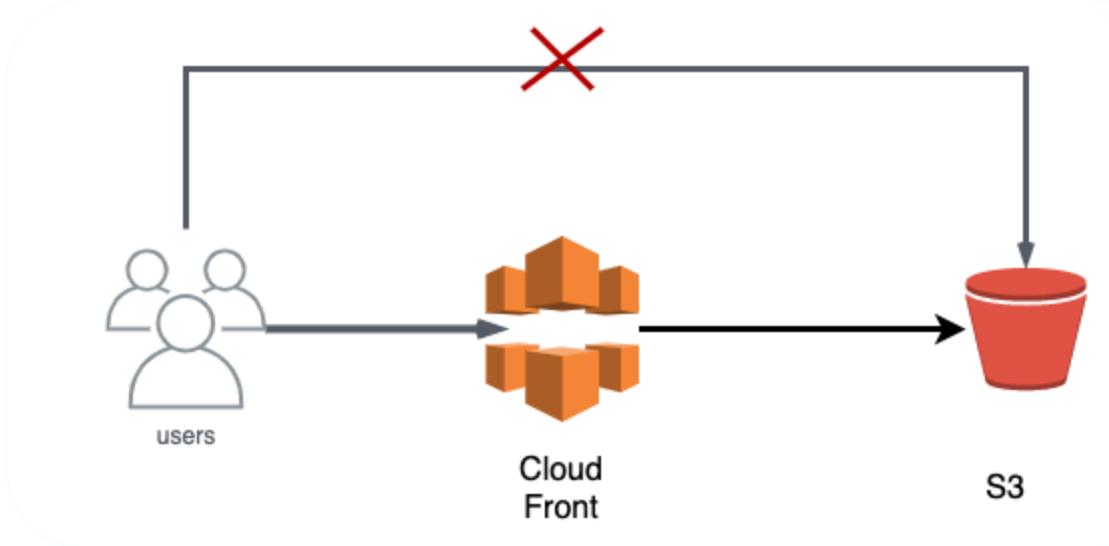
In 28  
Minutes

- Signed URLs
  - RTMP distribution
  - Application downloads (individual files) and
  - Situations where cookies are not supported
- Signed Cookies
  - Multiple files (You have a subscriber website)
  - Does not need any change in application URLs



# Amazon CloudFront - Origin Access Identities(OAI)

In 28  
Minutes



- Only CloudFront can access S3
- Create a Special CloudFront user - Origin Access Identities(OAI)
- Associate OAI with CloudFront distribution
- Create a S3 Bucket Policy allowing access to OAI

# Bucket Policy - S3 ONLY through Cloud Front

In 28  
Minutes

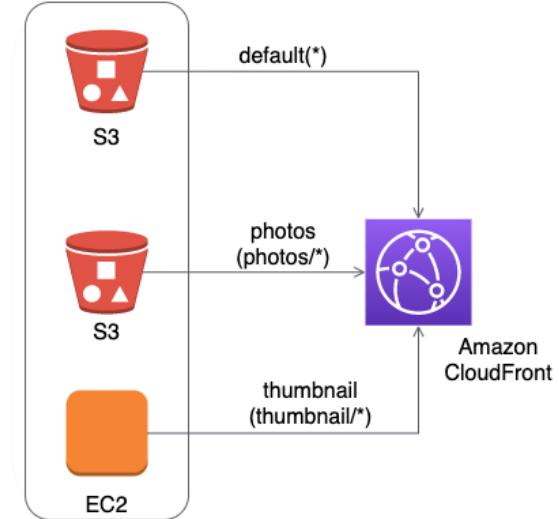


```
{  
    "Version": "2012-10-17",  
    "Id": "PolicyForCloudFrontPrivateContent",  
    "Statement": [  
        {  
            "Effect": "Allow",  
            "Principal": {  
                "AWS":  
                    "arn:aws:iam::cloudfront:user/CloudFront Origin Access Identity YOUR_IDENTITY_NAME"},  
            "Action": "s3:GetObject",  
            "Resource": "arn:aws:s3:::mybucket/*"  
        }  
    ]  
}
```

# Amazon CloudFront - Remember

22

- Old content automatically expires from CloudFront
- Invalidation API - remove object from cache
  - REMEMBER : Designed for use in emergencies
- Best Practice - Use versioning in object path name
  - Example : /images/profile.png?version=1
  - Prevents the need to invalidated content
- Do not use CloudFront for
  - all requests from single location
  - all requests from corporate VPN
- Scenario: Restrict content to users in certain countries
  - Enable CloudFront Geo restriction
  - Configure White list(countries to be allowed) and Blacklist(countries to be blocked)



- What would be the steps in setting up a website with a domain name (for example, in28minutes.com)?
  - Step I : Buy the domain name in28minutes.com (Domain Registrar)
  - Step II : Setup your website content (Website Hosting)
  - Step III : Route requests to in28minutes.com to the my website host server (DNS)
- Route 53 = Domain Registrar + DNS
  - Buy your domain name
  - Setup your DNS routing for in28minutes.com



# Route 53 - DNS (Domain Name Server)

In 28  
Minutes

*How should traffic be routed for in28minutes.com?*

- Configure Records:

- Route api.in28minutes.com to the IP address of api server
- Route static.in28minutes.com to the IP address of http server
- Route email (ranga@in28minutes.com) to the mail server(mail.in28minutes.com)
- Each record is associated with a TTL (Time To Live) - How long is your mapping cached at the routers and the client?



Route53

# Route 53 Hosted Zone

In 28  
Minutes

- Container for records containing DNS records routing traffic for a specific domain
- I want to use Route 53 to manage the records (Name Server) for [in28minutes.com](http://in28minutes.com)
  - Create a hosted zone for [in28minutes.com](http://in28minutes.com) in Route 53
- Hosted zones can be
  - private - routing within VPCs
  - public - routing on internet
- Manage the DNS records in a Hosted Zone



Route53

# Standard DNS Records

In 28  
Minutes

- A - Name to IPV4 address(es)
- AAAA - Name to IPV6 address(es )
- NS - Name Server containing DNS records
  - I bought in28minutes.com from GoDaddy (Domain Registrar)
  - BUT I can use Route 53 as DNS
    - Create NS records on GoDaddy
    - Redirect to Route 53 Name Servers
- MX - Mail Exchange
- CNAME - Name1 to Name2

	Name	Type	Value
	api.in28minutes.com.	A	192.0.2.235
	static.in28minutes.com.	AAAA	2001:0db8:85a3:0:0:8a2e:0370:7334
	dummy.in28minutes.com.	CNAME	www.example.com
	in28minutes.com.	MX	10 mailserver.in28minutes.com
	in28minutes.com.	NS	ns-1423.awsdns-49.org. ns-146.awsdns-18.com. ns-981.awsdns-58.net. ns-1997.awsdns-57.co.uk.
	in28minutes.com.	SOA	ns-1423.awsdns-49.org. awsdns-hostmaster.amazon.com

# Route 53 Specific Extension - Alias records

In 28  
Minutes

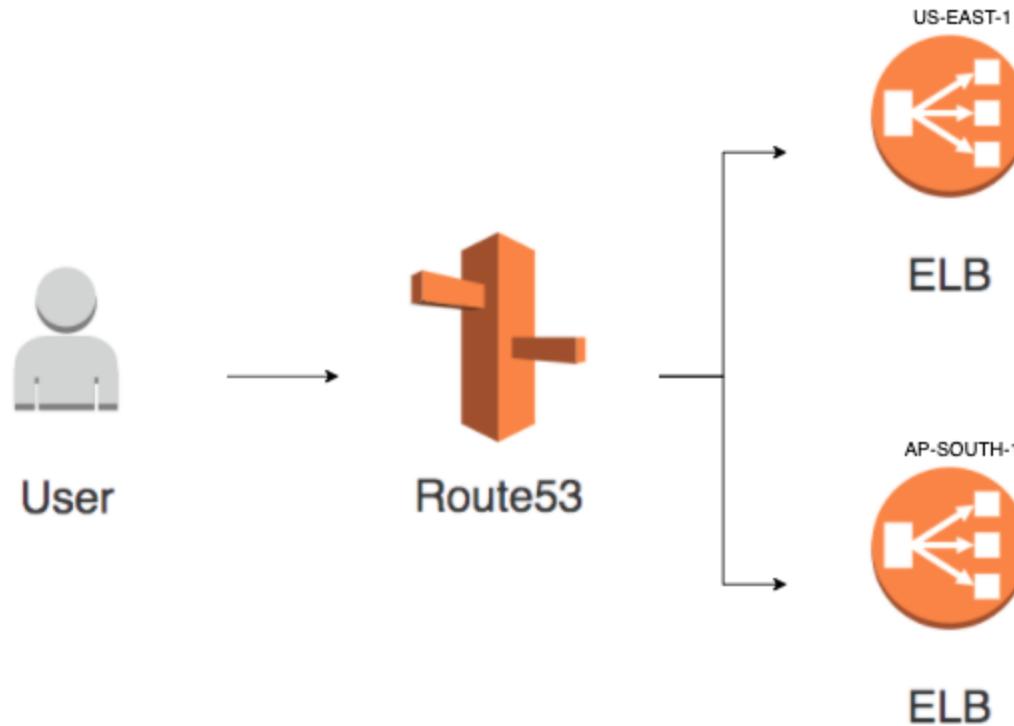
- Route traffic to selected AWS resources
  - Elastic Beanstalk environment
  - ELB load balancer
  - Amazon S3 bucket
  - CloudFront distribution
- Alias records can be created for
  - root(in28minutes.com) and
  - non root domains(api.in28minutes.com)
- COMPARED to CNAME records which can only be created for
  - non root domains (api.in28minutes.com)



Route53

# Route 53 - Routing

In 28  
Minutes



- Route 53 can route across Regions
  - Create ALBs in multiple regions and route to them!
  - Offers multiple routing policies

# Route 53 Routing Policies

In 28  
Minutes

Policy	Description
Simple	Maps a domain name to (one or more) IP Addresses
Weighted	Maps a single DNS name to multiple weighted resources 10% to A, 30% to B, 60% to C (useful for canary deployments)
Latency	Choose the option with minimum latency Latency between hosts on the internet can change over time
Failover	Active-passive failover. Primary Health check fails (optional cloud Watch alarm) => DR site is used
Geoproximity	Choose the nearest resource (geographic distance) to your user. Configure a bias.
Multivalue answer	Return multiple healthy records (upto 8) at random You can configure an (optional) health check against every record
Geolocation	Choose based on the location of the user

# Route 53 Routing Policies - Geolocation

In 28  
Minutes

- Choose based on the location of the user
  - continent, country or a (state in USA)
  - Send traffic from Asia to A
  - Send traffic from Europe to B etc.
- Record set for smallest geographic region has priority
- Use case
  - Restrict distribution of content to specific areas where you have distribution rights
- (RECOMMENDED) Configure a default policy (used if none of the location records match)
  - Otherwise, Route 53 returns a "no answer" if none of the location records match

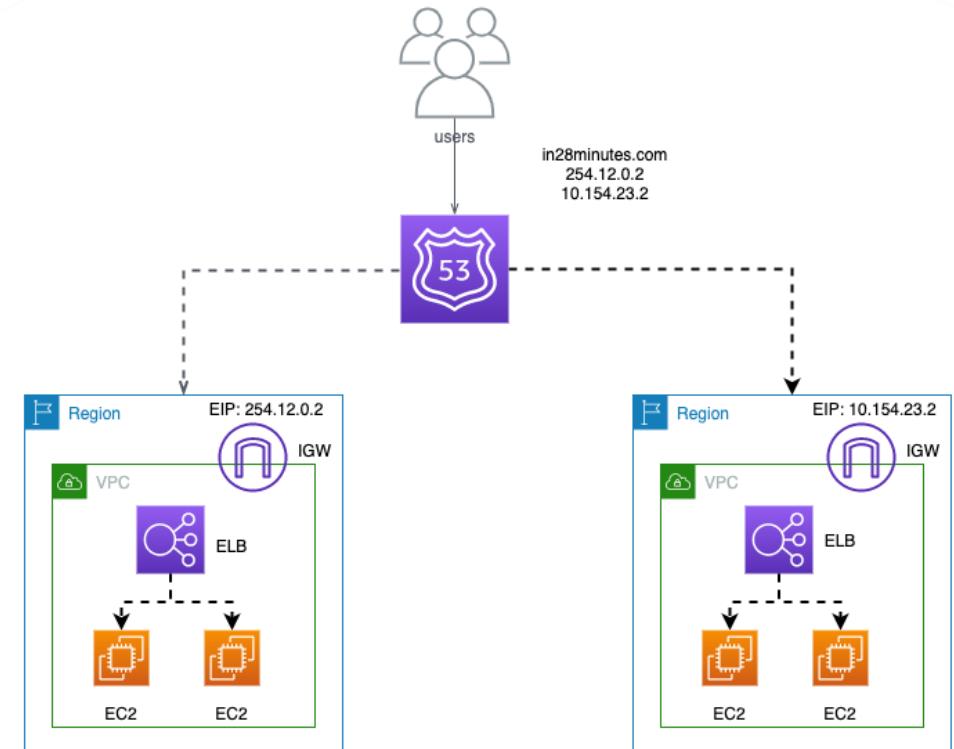


Route53

# Need for AWS Global Accelerator

In 28  
Minutes

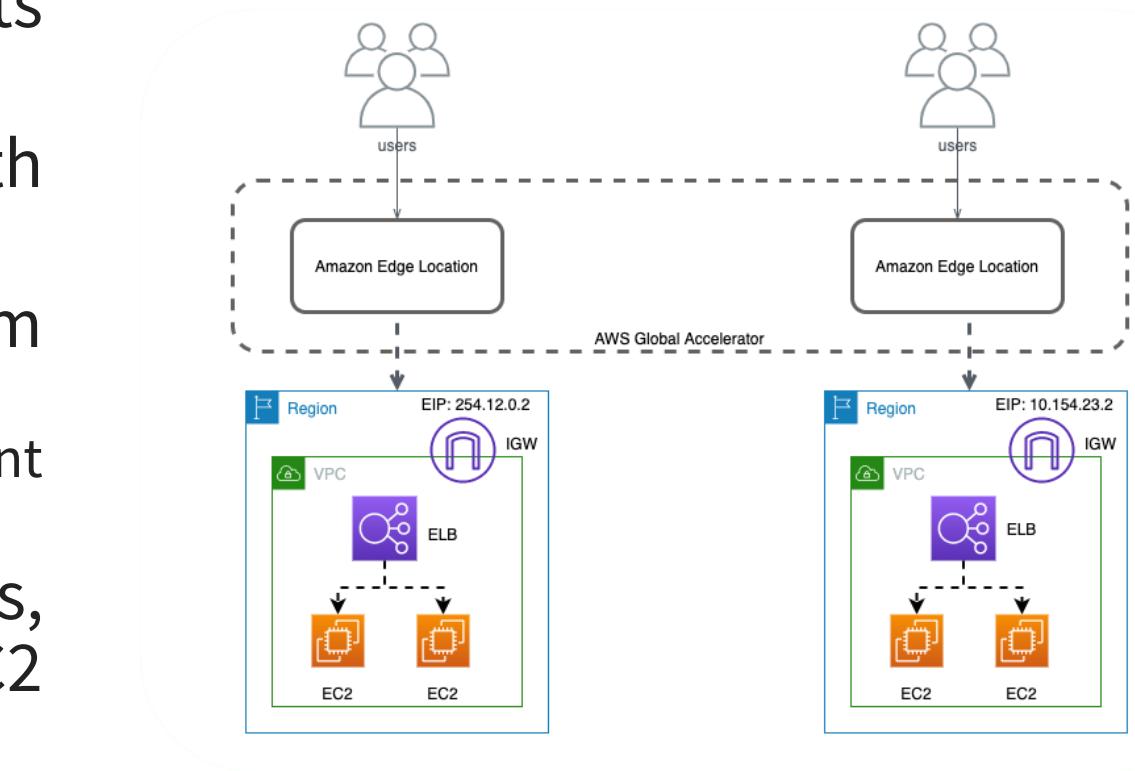
- Cached DNS answers
  - clients might cache DNS answers causing a delay in propagation of configuration updates
- High latency
  - users connect to the region over the internet



# AWS Global Accelerator

In 28  
Minutes

- Directs traffic to optimal endpoints over the AWS global network
- Global Accelerator provides you with two static IP addresses
- Static IP addresses are anycast from the AWS edge network
  - Distribute traffic across multiple endpoint resources in multiple AWS Regions
- Works with Network Load Balancers, Application Load Balancers, EC2 Instances, and Elastic IP addresses



# ETL & Big Data

# Redshift and EMR

# Amazon Redshift

In 28  
Minutes

- Redshift is a relational database ( tables and relationships)
- What is the need for another relational database?
  - RDS is optimized for online transaction processing
  - It is optimized to provide a balance between both reads and write operations
- OLAP workloads have exponentially larger reads on the databases compared to writes:
  - Can we use a different approach to design the database?
  - How about creating a cluster and splitting the execution of the same query across several nodes?
- Redshift is a **petabyte-scale distributed data ware house** based on PostgreSQL



Redshift

# Amazon Redshift

In 28  
Minutes

- Three important characteristics of Redshift:
  - Massively parallel processing (MPP) - storage and processing can be split across multiple nodes
  - Columnar data storage
  - High data compression
- As a result
  - A single row of data might be stored across multiple nodes
  - A query to Redshift leader node is distributed to multiple compute nodes for execution
- Start with a single node configuration and scale to multi node configuration
- You can dynamically add and remove nodes



Redshift

# Amazon Redshift

In 28  
Minutes

- Used for traditional ETL(Extract, Transform, Load), OLAP and Business Intelligence (BI) use cases
  - Optimized for high-performance analysis and reporting of very large datasets
- Supports standard SQL
- Integration with data loading, reporting, mining and analytics tools
- Provides high availability and durability:
  - Automatic replication (maintains 3 copies of your data)
  - Automated backups (to S3. Default retention - 1 day. Max - 35 days)
  - Automatic recovery from any node failures

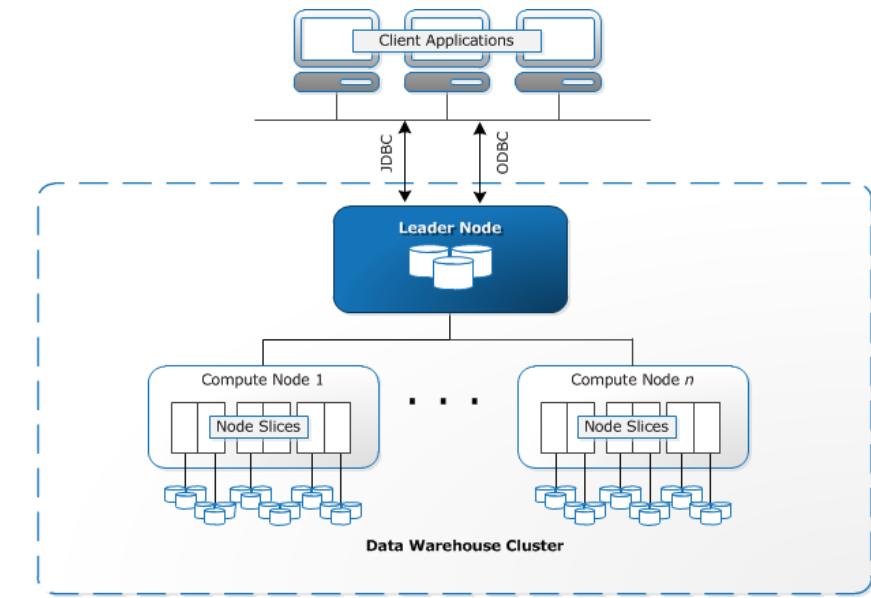


Redshift

# Redshift Cluster

In 28  
Minutes

- One leader node and multiple compute nodes
  - Add compute nodes for more performance
  - Create a cluster subnet group to use a VPC
- One or more databases in a cluster
- Clients communicate with leader node
  - Leader node divides the query execution between compute nodes
  - No direct access to compute nodes



<https://docs.aws.amazon.com/redshift/latest/>

*NodeRelationships.png*

# Redshift - Designing Tables

In 28  
Minutes

- Compression Encoding (optional)
  - Let Redshift choose or configure for each column
    - Examples : Raw, Bytedict, LZO, Runlength, Text255, Text32K
  - Find the right compression encoding by running tests
- Sort Keys (optional)
  - Data is stored in sorted order (using sort key)
  - Increase efficiency of your queries
  - Example 1 : Columns used frequently in range (year > 1995 and year < 2005) or equal (year = 2015) conditions
  - Example 2 : Join columns with other tables
  - Example 3 : Timestamp columns if you use the most recent data frequently



Redshift

# Redshift - Designing Tables - Distribution Strategy

In 28  
Minutes

- How are the rows of the table distributed across compute nodes?
  - Aim to distribute data equally across nodes and minimize data movement during query execution
- EVEN (default) - data is uniformly distributed
- KEY - based on values of one column
  - Matching values are stored close together
  - Use join columns as KEY if you want matching columns to be co-located
- ALL - entire table on all nodes
  - Used for lookup tables



# Loading Data into Amazon Redshift

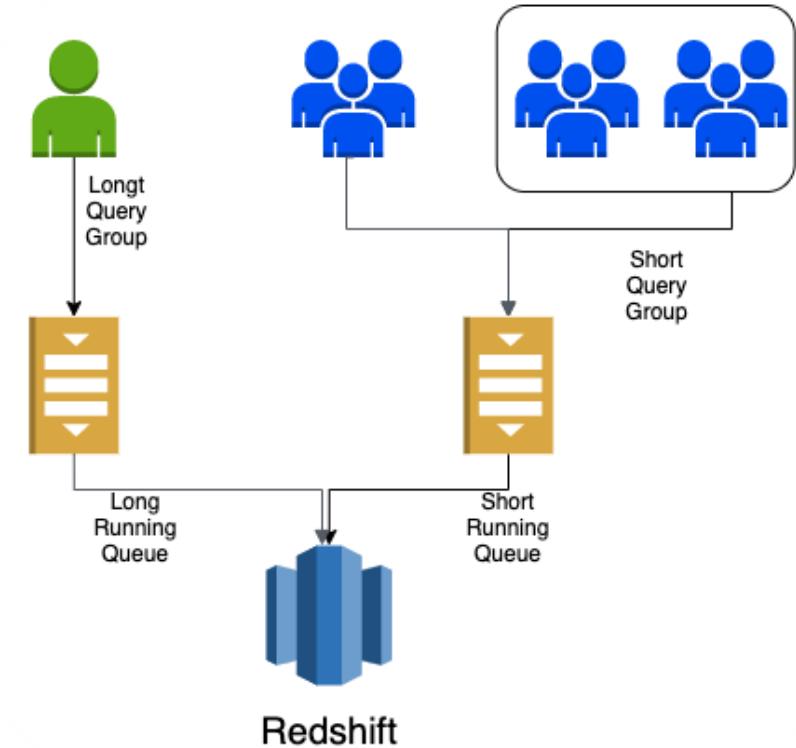
In 28  
Minutes

Scenario	Solution
Simple	Use SQL insert queries using ODBC or JDBC
Efficient	Use Amazon Redshift COPY command to load data from Amazon S3, Amazon DynamoDB, Amazon EMR etc
Data Pipelines	Load using AWS Data Pipeline
On-premises data	Use Storage Gateway or Import/Export to import data into S3. COPY data from S3
Other databases	AWS Database Migration Service : RDS, DynamoDB or another Amazon Redshift Database
Recommendation	Prefer COPY over INSERT for bulk operations as COPY is done in parallel
Recommendation	Prefer COPY from multiple files. Split large files into multiple small input files

# Redshift Workload Management

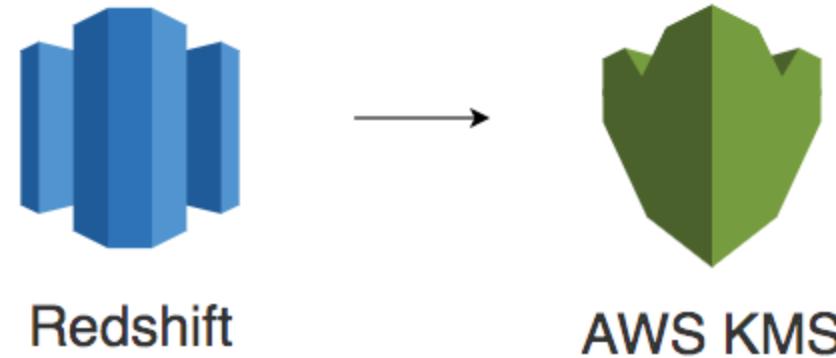
In 28  
Minutes

- WLM can be configured to prioritize queues
- Create multiple queues with different concurrency level for different purposes
- One queue for long running queries with low concurrency
- One queue for short running queries with high concurrency (upto 50 concurrent queries)



# Redshift Security

In 28  
Minutes



- Uses 4-tier, key-based architecture for encryption
  - master key (chosen from keys in KMS)
  - a cluster encryption key (CEK)
  - a database encryption key (DEK)
  - and data encryption keys
- Manage keys using AWS KMS or AWS Cloud HSM
- IAM to manage user permissions for cluster operations
  - Grant permissions on a per cluster basis instead of per table basis

# Redshift Operations

In 28  
Minutes



Redshift

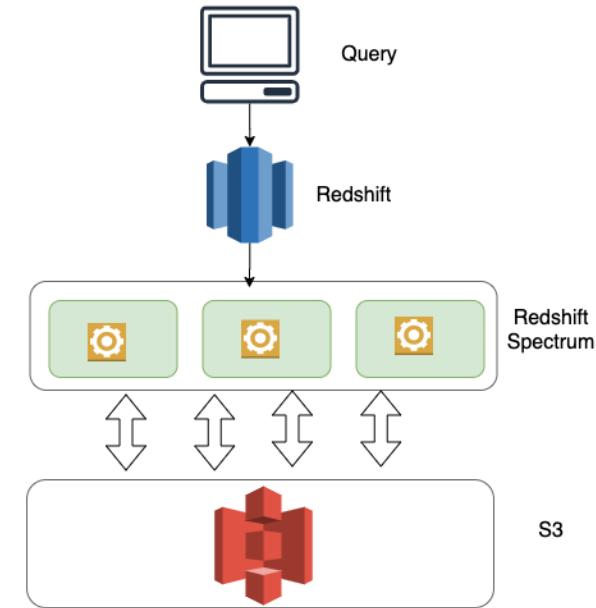


Cloudwatch

- Add new columns by using ALTER TABLE
  - Existing columns cannot be modified
- SQL operations are logged
  - Use SQL queries to query against system tables or download to S3
- Monitor performance & queries with Cloud Watch and Redshift web console
- When deleting a Redshift cluster, take a final snapshot to Amazon S3

# Amazon Redshift Spectrum

- Run SQL queries against datasets in Amazon S3
  - Does not need any intermediate data stores
- Auto scales based on your queries
- Scale storage and compute independently
- Metadata defined in Amazon Redshift
  - Avro, CSV, Ion, JSON, ORC, Parquet formats supported
- Eliminate expensive data transfers from S3 to data warehousing solutions (Cost effective)
- Integrates with Amazon Athena
- Query against Amazon EMR (as well)



# Amazon EMR - Elastic MapReduce

In 28  
Minutes

- Managed Hadoop service with high availability and durability
- EMR gives access to underlying OS => You can SSH into it
- Important tools in Hadoop eco system are natively supported:
  - Examples: Pig, Hive, Spark or Presto
- Install others using bootstrap actions
- Use cases
  - Log processing for insights
  - Click stream analysis for advertisers
  - Genomic and life science dataset processing

# Amazon EMR - Storage Types

In 28  
Minutes

Feature	Hadoop Distributed File System (HDFS)	EMR File System (EMRFS)
Standard for Hadoop	✓	X
Data Storage	EBS or instance storage	S3
Data Survival on cluster shutdown	Yes for EBS. No for Instance Storage	Yes
Persistent Clusters running 24 X 7 analysis	✓ (low latency on instance storage)	
Transient Clusters running Infrequent big data jobs		✓(Run MapReduce jobs against S3 bucket)

# Amazon Redshift and EMR Alternatives

In 28  
Minutes

Alternative	Scenario
Amazon EMR	For big data frameworks like Apache Spark, Hadoop, Presto, or Hbase to do large scale data processing that needs high customization For example: machine learning, graph analytics etc
Amazon Redshift	Run complex queries against data warehouse - housing structured and unstructured data pulled in from a variety of sources
Amazon Redshift Spectrum	Run queries directly against S3 without worrying about loading entire data from S3 into a data warehouse
Amazon Athena	Quick ad-hoc queries without worrying about provisioning a compute cluster (serverless) Amazon Redshift Spectrum is recommended if you are executing queries frequently against structured data

# Handling Workflows

# Amazon Simple Workflow Service (SWF)

In 28  
Minutes

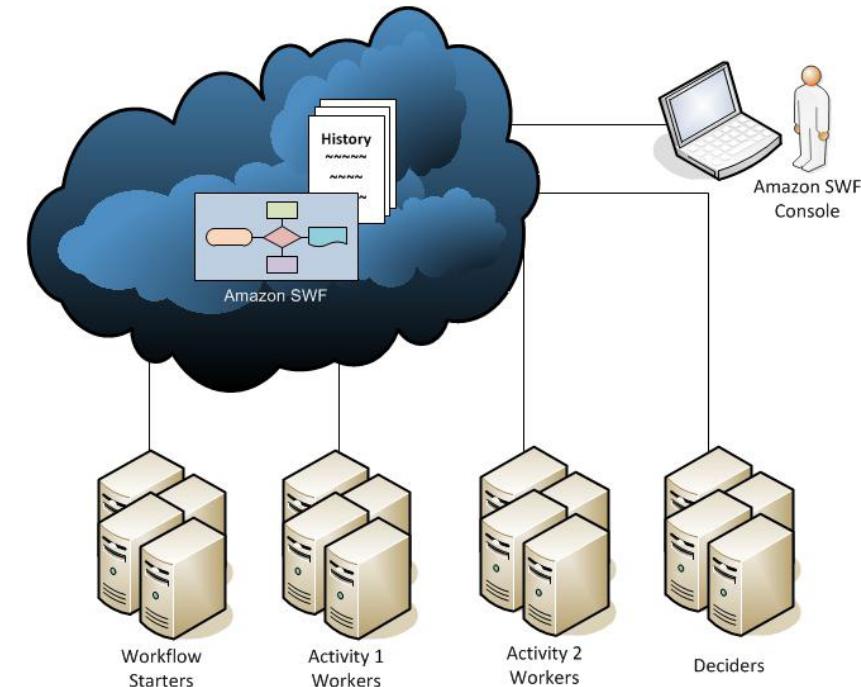
- Build and run background jobs with
  - parallel or sequential steps
  - synchronously or asynchronously
  - with human inputs (can indefinitely wait for human inputs)
- (Use cases) Order processing and video encoding workflows
- A workflow can start when receiving an order, receiving a request for a taxi
- Workflows can run upto 1 year
- Deciders and activity workers can use long polling



# Amazon SWF - Order Process

In 28  
Minutes

- Key Actors : Workflow starter, Decider and Activity worker
- Workflow starter calls SWF action to start workflow
  - Example: when an order is received
- SWF receives request and schedules a decider
  - Decider receives the task and returns decision to SWF:
    - For example, schedule an activity "Activity 1"
  - SWF schedules "Activity 1"
  - Activity worker performs "Activity 1". Returns result to SWF.
  - SWF updates workflow history. Schedules another decision task.
  - Loop continues until decider returns decision to close workflow
- SWF archives history and closes workflow



<https://docs.aws.amazon.com/amazonswf/latest/dev-actors.html>

# Handling Data Streams

# Streaming Data

In 28  
Minutes

- Imagine implementing analytics for a website:
  - You have a continuous stream of data (page views, link clicks etc)
- Characteristics of streaming data:
  - Continuously generated
  - Small pieces of data
  - Sequenced - mostly associated with time
- How do you process continuous streaming data originating from application logs, social media applications?



# S3 Notifications

In 28  
Minutes

- Send notifications to SNS, SQS, trigger lambda functions on
  - creation, deletion or update of an S3 object
- Setup at bucket level
  - You can use prefix and suffix to configure
- Cost efficient for simple use cases
  - S3 notification -> Lambda
  - Almost negligible cost (storage for file + invocation)



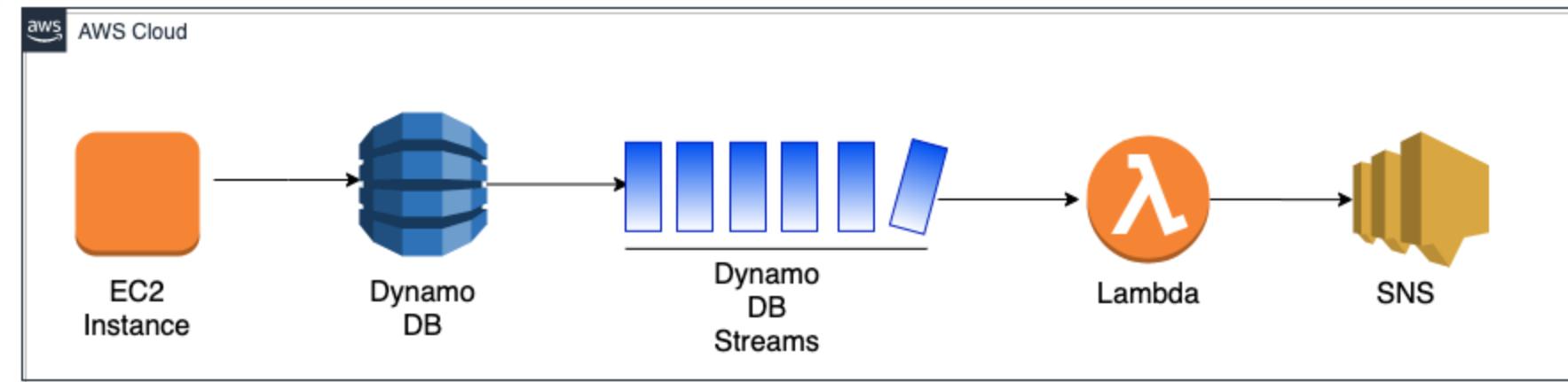
S3 Bucket



Lambda Fn

# DynamoDB Streams

In 28  
Minutes



- Each event from DynamoDB (in time sequenced order) is buffered in a stream near real-time
- Can be enabled or disabled
- Use case - Send email when user registers
  - Tie a Lambda function to DynamoDB Streams
- Stream allow iteration through records (**last 24 hours**)

# Amazon Kinesis

In 28  
Minutes

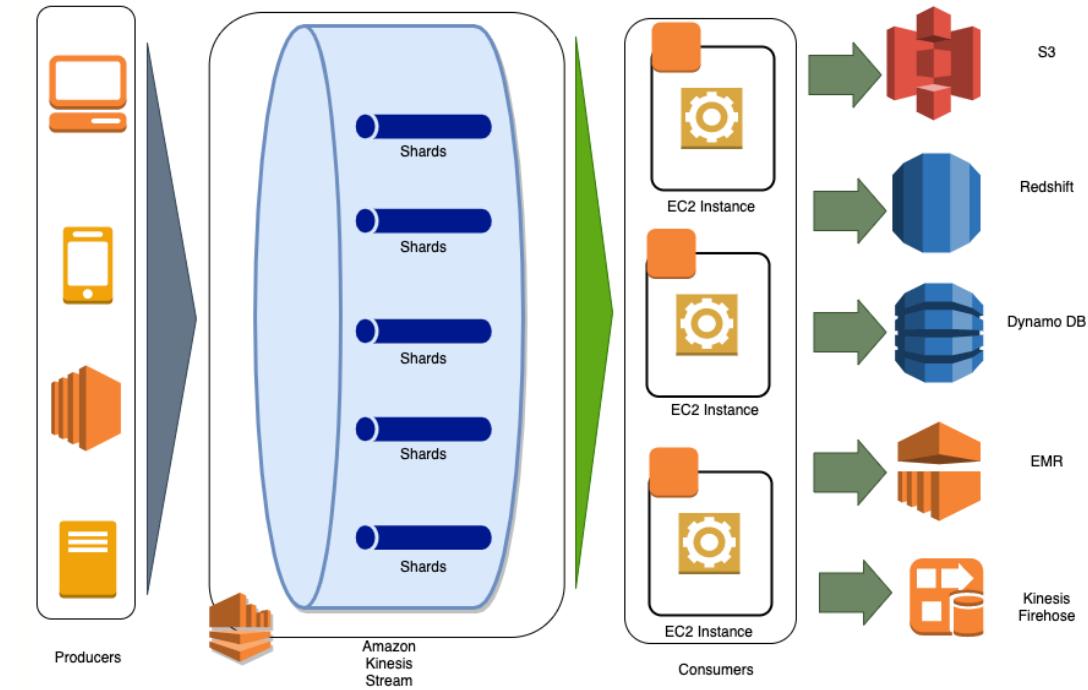
- Handle streaming data
  - NOT recommended for ETL Batch Jobs
- Amazon Kinesis Data Streams
  - Process Data Streams
- Amazon Kinesis Firehose
  - Data ingestion for streaming data : S3, Elasticsearch etc
- Amazon Kinesis Analytics
  - Run queries against streaming data
- Amazon Kinesis Video Streams
  - Monitor video streams



# Amazon Kinesis Data Streams

In 28  
Minutes

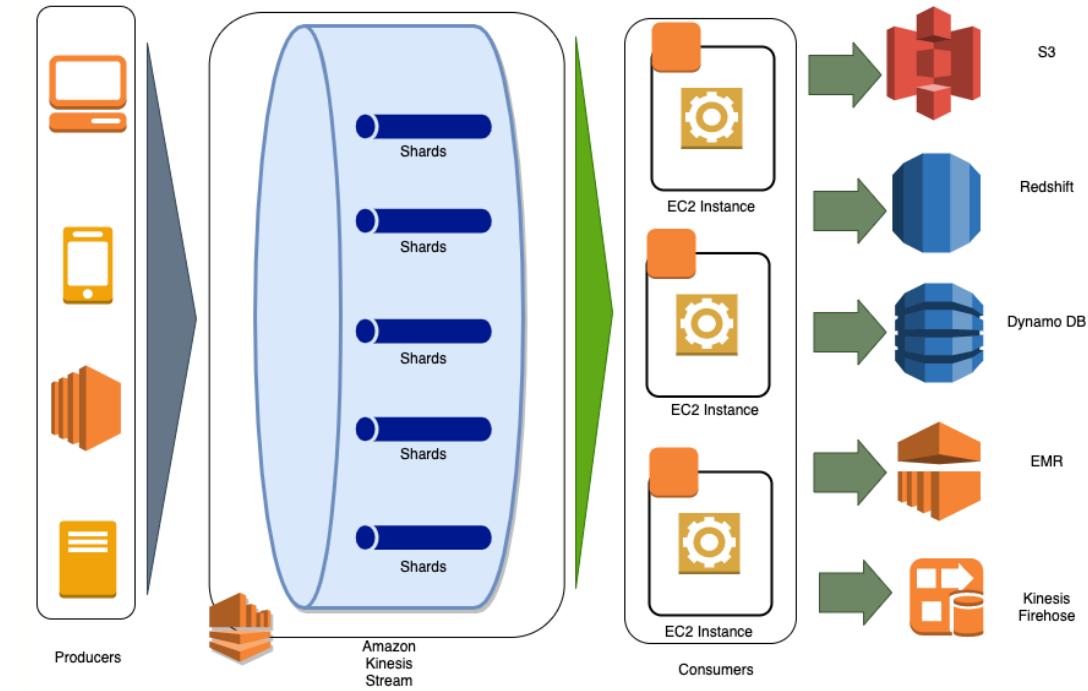
- Limitless Real time stream processing
  - Sub second processing latency
- Alternative for Kafka
- Supports multiple clients
  - Each client can track their stream position
- Retain and replay data (max 7 days & default 1 day)



# Amazon Kinesis Data Streams - Integrations

In 28  
Minutes

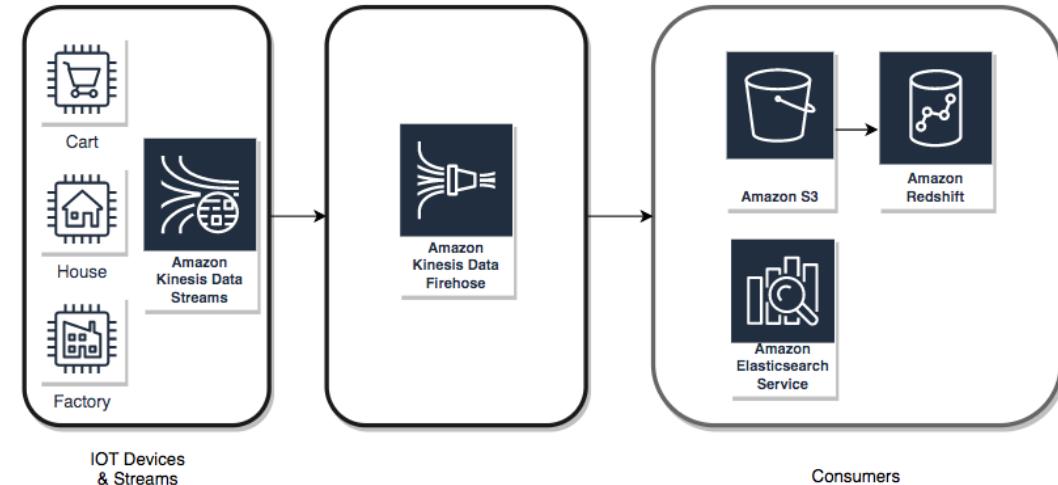
- Use application integrations to generate streams
  - Toolkits : AWS SDK, AWS Mobile SDK, Kinesis Agent
  - Service Integrations : AWS IOT, CloudWatch Events and Logs
- Process streams using Kinesis Stream Applications
  - Run on EC2 instances
  - Written using Kinesis Data Streams APIs



# Amazon Kinesis Data Firehose

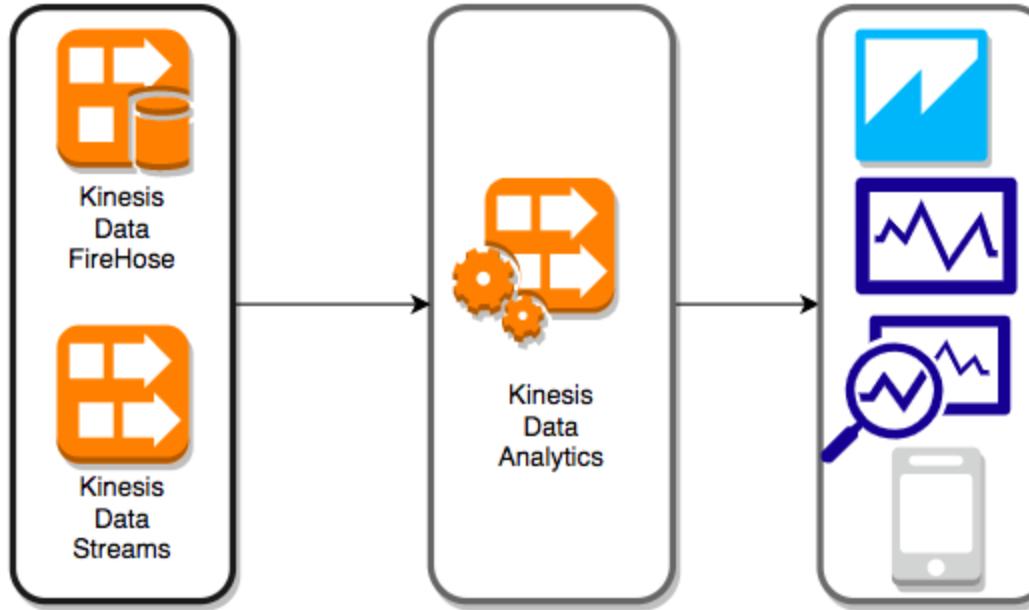
In 28  
Minutes

- Data ingestion for streaming data
  - Receive
  - Process ( transform - Lambda, compress, encrypt )
  - Store stream data to S3, Elasticsearch, Redshift and Splunk
- Use existing analytics tools based on S3, Redshift and Elasticsearch
- Pay for volume of data ingested (Serverless)



# Amazon Kinesis Analytics

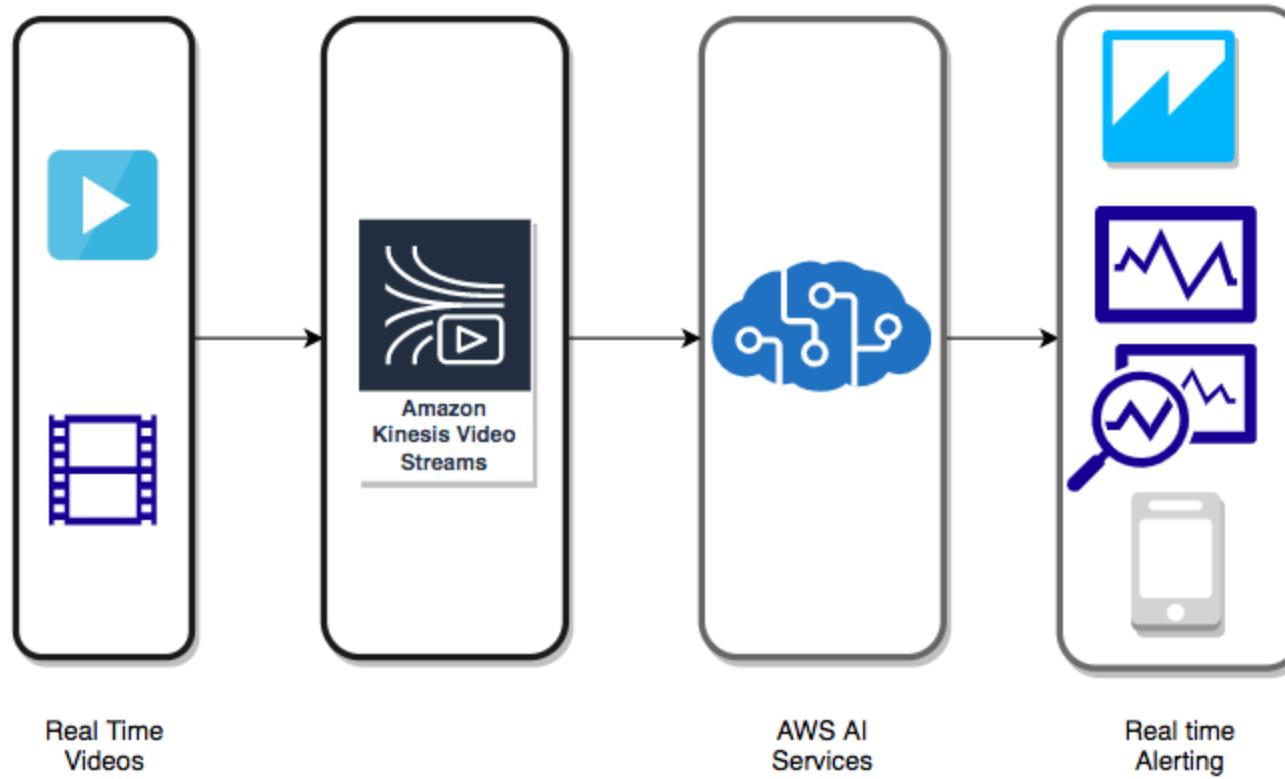
In 28  
Minutes



- You want to continuously find active number of users on a website in the last 5 minutes based on streaming website data
- With Amazon Kinesis Analytics, you can write SQL queries and build Java applications to continuously analyze your streaming data

# Amazon Kinesis Video Streams

In 28  
Minutes



- Monitor video streams from web-cams
- Examples: traffic lights, shopping malls, homes etc
- Integrate with machine learning frameworks to get intelligence

# AWS Data Lakes

# AWS Data Lakes - Simplified Big Data Solutions

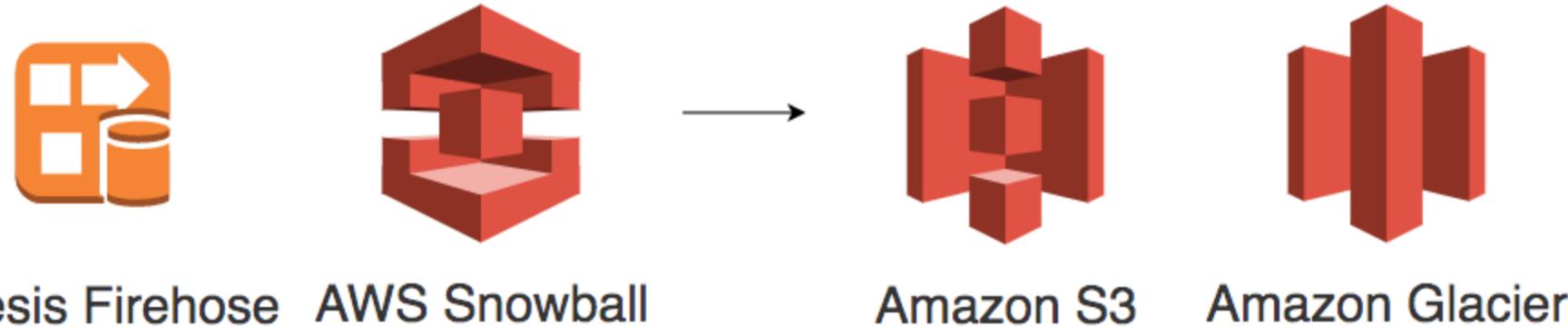
In 28  
Minutes

- Usual big data solutions are complex
- How can we make collecting, analyzing (reporting, analytics, machine learning) and visualizing huge data sets easy?
- How to design solutions that scale?
- How to build flexibility while saving cost?
- Data Lake
  - Single platform with combination of solutions for data storage, data management and data analytics



# AWS Data Lakes - Storage and Ingestion

In 28  
Minutes



- **Storage**
  - Amazon S3 and S3 Glacier provide an ideal storage solution for data lakes
- **Data Ingestion**
  - Streaming data - Amazon Kinesis Firehose
    - Transform and store to Amazon S3
    - Transformation operations - compress, encrypt, concatenate multiple records into one (to reduce S3 transactions cost) and execute lambda functions
  - Bulk data from on-premises - AWS Snowball
  - Integrate on-premises platforms with Amazon S3 - AWS Storage Gateway

# Amazon S3 Query in Place

In 28  
Minutes

- Run your analytics directly from Amazon S3 and S3 Glacier
- S3 Select and Glacier Select
  - SQL queries to retrieve subset of data
    - Supports CSV, JSON, Apache Parquet formats
  - Build serverless apps connecting S3 Select with AWS Lambda
  - Integrate into big data workflows
    - Enable Presto, Apache Hive and Apache Spark frameworks to scan and filter data
- Amazon Athena
  - Direct ad-hoc SQL querying on data stored in S3
  - Uses Presto and supports CSV, JSON, Apache Parquet and Avro
- Amazon Redshift Spectrum
  - Run queries directly against S3 without loading complete data from S3 into a data warehouse
  - Recommended if you are executing queries frequently against structured data



Athena



Redshift



Amazon S3



Amazon Glacier

# Amazon S3 Query in Place - Recommendations

In 28  
Minutes

- You want to get quick insights from your cold data stored in S3 Glacier. You want to run queries against archives stored in S3 Glacier without restoring the archives.
  - Use S3 Glacier Select to perform filtering and basic querying using SQL queries
  - Stores results in S3 bucket
  - No need to temporarily stage data and then run queries
- Recommendations:
  - Store data in Amazon S3 in Parquet format
    - Reduce storage (upto 85%) and improve querying (upto 99%) compared to formats like CSV, JSON, or TXT
  - Multiple compression standards are supported BUT GZIP is recommended
    - Supported by Amazon Athena, Amazon EMR and Amazon Redshift

# AWS Data Lakes - Analytics with data in S3 Data Lake

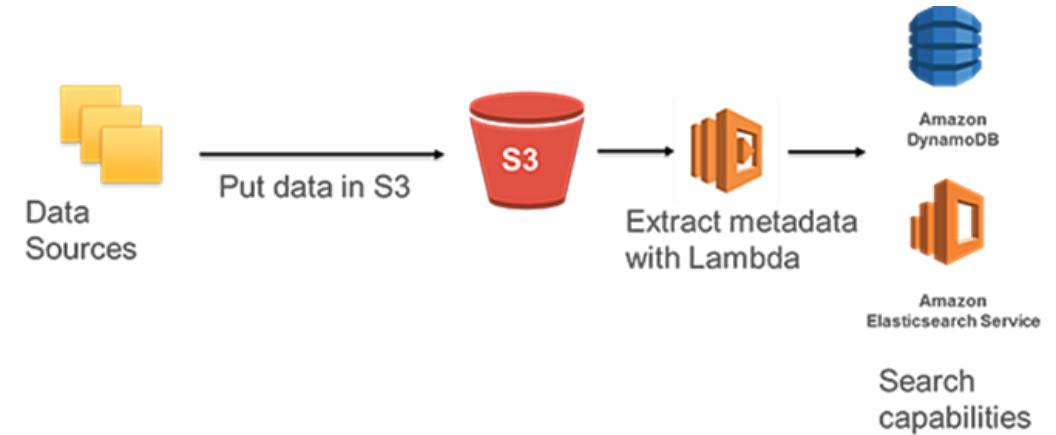
In 28  
Minutes

Service	Description
Amazon EMR	EMR integrates well with Amazon S3 - Use big data frameworks like Apache Spark, Hadoop, Presto, or Hbase. For example: machine learning, graph analytics etc
Amazon Machine Learning (ML)	Create and run models for predictive analytics and machine learning (using data from Amazon S3, Amazon Redshift, or Amazon RDS)
Amazon QuickSight	For visualizations (using data from Amazon Redshift, Amazon RDS, Amazon Athena, and Amazon S3)
Amazon Rekognition	Build image recognition capabilities around images stored in Amazon S3. Example use case : Face based verification

# AWS Data Lakes - Data Cataloging

In 28  
Minutes

- 1 : What data (or assets) is stored?
- 2 : What is the format of data?
- 3 : How is the data structured?
- Question 1 - Stored in comprehensive data catalog
- Questions 2 and 3 - Stored using a Hive Meta store Catalog (HCatalog)
- AWS Glue also supports storing HCatalog



<https://docs.aws.amazon.com/whitepapers/latest/building-data-lakes/data-cataloging.html>

- Fully managed extract, transform, and load (ETL) service
- Simplify data preparation (capturing metadata) for analytics:
  - Connect AWS Glue to your data on AWS (Aurora, RDS, Redshift, S3 etc)
  - AWS Glue creates a AWS Glue Data Catalog with metadata abstracted from your data
  - Your data is ready for searching and querying
- Run your ETL jobs using Apache Spark
- Metadata from AWS Glue Data Catalog can be used from:
  - Amazon Athena
  - Amazon EMR
  - Amazon Redshift Spectrum



AWS Glue

# More Serverless

# Serverless Options - Compute

In 28  
Minutes



AWS Lambda

- AWS Lambda
  - Run code without provisioning servers!
  - Also called FAAS (Function as a Service)
- Lambda@Edge
  - Run lambda functions at AWS Edge Locations
  - Integrated with CloudFront
- AWS Fargate
  - Container Orchestration without worrying about ec2 instances

# Serverless Options - Storage

In 28  
Minutes



Amazon S3



Amazon EFS

- Amazon S3
  - Highly scalable object storage
  - We've talking sufficiently about it already!
- Amazon Elastic File System
  - Elastic file storage for UNIX compatible systems

# Serverless Options - Databases

20



DynamoDB

- Amazon DynamoDB
  - Fast, scalable, distributed and flexible non-relational (NoSQL) database service for any scale
  - Need to configure read and write capacity for tables
    - NOT truly serverless BUT don't tell that to AWS
    - Truly serverless mode is expensive
- Amazon Aurora Serverless
  - Use Amazon RDS with Aurora in serverless mode
  - WARNING : I would still consider this early stage
- Amazon RDS Proxy
  - Sits between client applications (including lambdas) and your RDS database
  - Efficient management of short lived database connections (by pooling database connections)

# Serverless Options - API Proxy and Orchestration

In 28  
Minutes



API Gateway



Step Functions

- **Amazon API Gateway**
  - API Management platform helping you create, publish, maintain, monitor and secure your APIs
  - Provides authorization, rate limiting and versioning
- **AWS Step Functions**
  - Setup workflows involving services like AWS Lambda and AWS Fargate
  - Orchestration and state management

# Serverless Options - Application Integration and Analytics

28  
Minutes

- Amazon SNS
  - Follows “publish-subscribe” (pub-sub) messaging paradigm to broadcast asynchronous event notifications - SMS, e-mails, push notifications etc
- Amazon SQS
  - Fully managed queuing service
  - Helps you decouple your applications
- Amazon Kinesis
  - Multiple solutions to process streaming data
- Amazon Athena
  - Query using SQL on data in Amazon S3
  - Pay only for queries!



Amazon SNS



Amazon SQS



Kinesis



Athena

# Serverless Options - Others

In 28  
Minutes

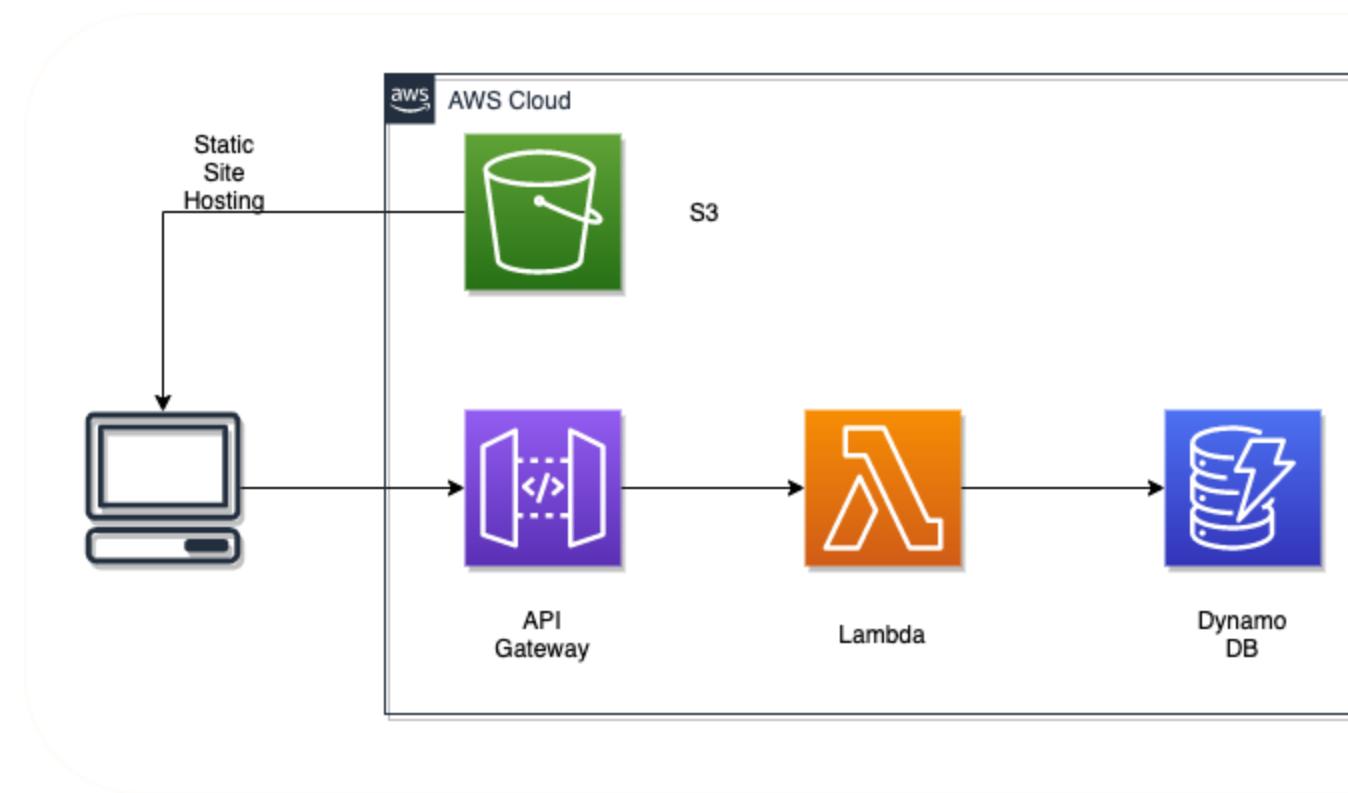


Amazon Cognito

- **Amazon Cognito**
  - Fully managed solution providing authorization and authentication solutions for web/mobile apps
- **AWS Serverless Application Model**
  - Open source framework for building serverless applications

# Serverless Use case 1 - Full Stack Web Application

In 28  
Minutes

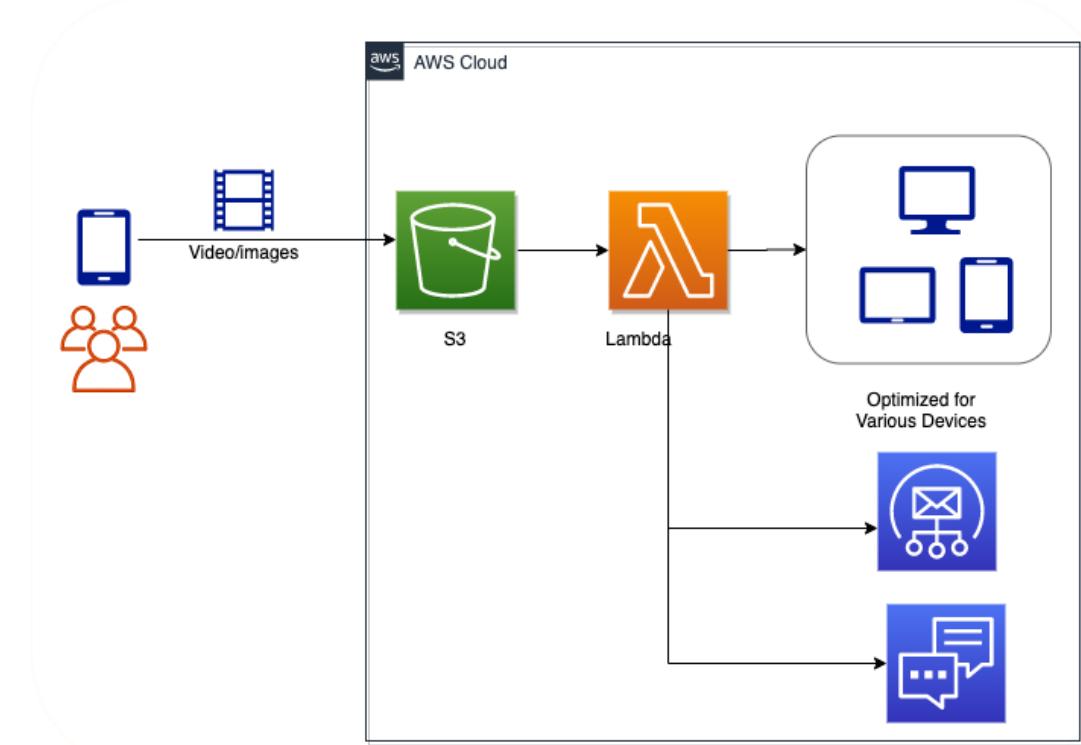


- Static content stored in S3
- API Gateway and Lambda are used for the REST API
- DynamoDB is used to store your data

# Serverless Use case 2 - Real time event processing

In 28  
Minutes

- User uploads videos to S3
- S3 notifications are used to invoke Lambda functions to optimize videos for different devices.



# Amazon API Gateway Features

In 28  
Minutes

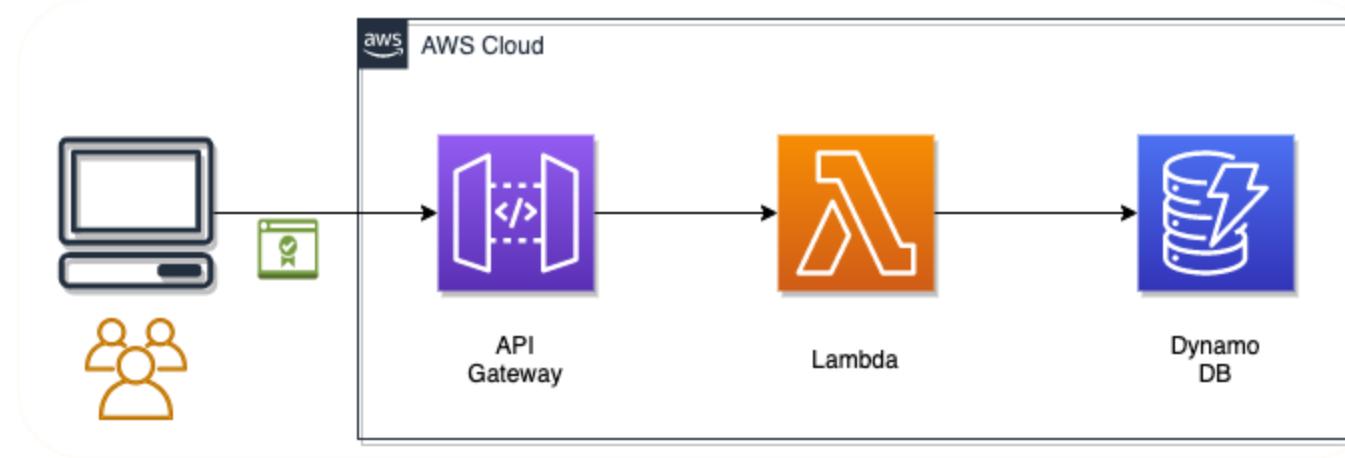
- Lifecycle management for REST APIs
- Versioning and multiple environments
- API keys - Generate API keys to monitor usage
  - Implement plans and quota limits for external applications (or developer)
  - WARNING - Do NOT use API keys for Authorization
- Enable caching for API calls with TTL
- Protect backends by throttling requests
- Integrates with
  - Amazon CloudWatch - Performance metrics, API calls, latency data and error rates
  - Amazon CloudWatch Logs - Debug logging
  - AWS CloudTrail - Complete history of changes to your REST API



API Gateway

# Amazon API Gateway - Authentication and Authorization

In 28  
Minutes



- How do you authenticate a REST API call?
  - Attach a signature or token with your API call

# Amazon API Gateway - Authentication and Authorization Approaches

In 28  
Minutes

- AWS Signature Version 4
  - Create a signature using your AWS secret access key and send it with your API request
  - For API consumers belonging to your AWS account
- Lambda authorizers
  - Implement a Lambda function to authenticate (JWT, OAuth etc) the token and return IAM policies.
  - Integrate with any custom user directory
- Amazon Cognito
  - We will look at authentication with Cognito next

# Amazon Cognito

In 28  
Minutes

- Want to quickly add a sign-up page and authentication for your mobile and web apps?
- Want to integrate with web identity providers (example: Google, Facebook, Amazon) and provide a social sign-in?
- Do you want security features such as multi-factor authentication (MFA), phone and email verification?
- Want to create your own user database without worrying about scaling or operations?
- Let's go : Amazon Cognito
- Support for SAML



Amazon Cognito

# Amazon Cognito - User Pools

In 28  
Minutes

- Do you want to create your own secure and scalable user directory?
- Do you want to create sign-up pages?
- Do you want a built-in, customizable web UI to sign in users (with option to social sign-in )?
- Create a user pool



Amazon Cognito

# Amazon Cognito - Identity pools

In 28  
Minutes

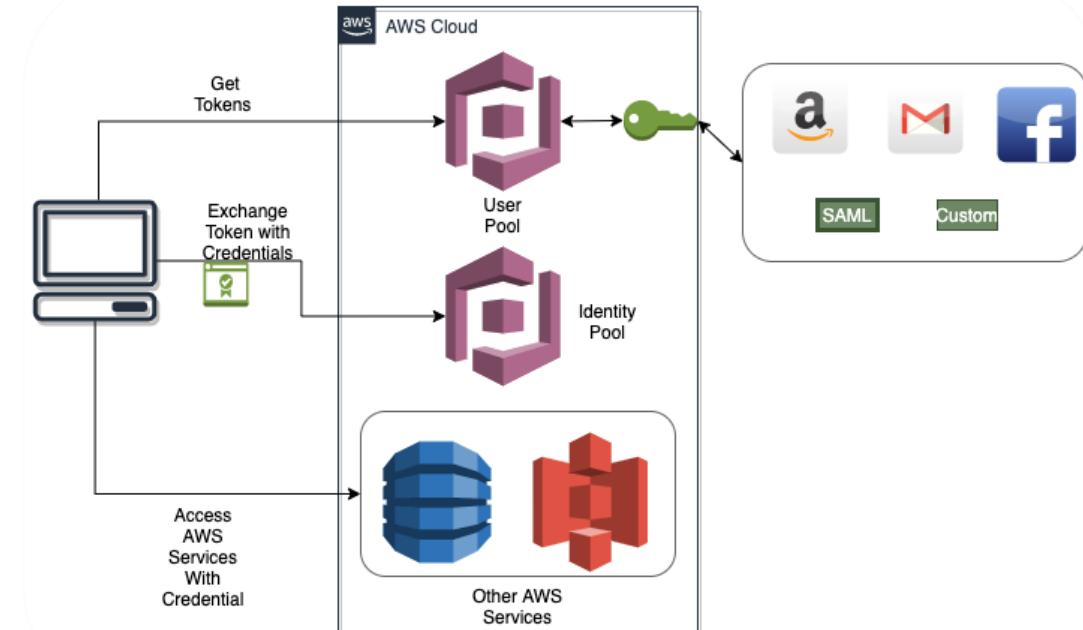


- Identity pools provide AWS credentials to grant your users access to other AWS services
- Connect identity pools with authentication (identity) providers
  - Your own user pool OR
  - Amazon, Apple, Facebook, Google+, Twitter OR
  - OpenID Connect provider OR
  - SAML identity providers (SAML 2.0)
- Configure multiple authentication (identity) providers for each identity pool
- Federated Identity
  - An external authentication (identity) provider
  - ex: Amazon, Apple, Facebook, OpenID or SAML identity providers

# Amazon Cognito - How does it work?

In 28  
Minutes

- Application sends user credentials to identity provider
  - (If authenticated) Identity provider sends a token to application
- Application sends the token to Identity Pool
  - (If valid token) Identity Pool creates temporary credentials (access key, secret key, and session token) using STS
- App sends a request with the credentials to the AWS service



# Lambda@Edge

In 28  
Minutes

- Run lambda functions at AWS Edge Locations
  - Lowest network latency for end users
- Use cases : Search Engine Optimization, A/B Testing, Dynamically routing to different origins
- Can be triggered on these Amazon CloudFront events:
  - Viewer Request - when request arrives at edge location
  - Origin Request - Just before sending request to origin (when object is not in cache)
  - Origin Response - After the edge location receives response from origin
  - Viewer Response - Just before a response is sent back from edge location
- LIMITATION : Supports ONLY Node.js and Python programming languages
- LIMITATION : No free tier and more expensive than Lambda

# Serverless Application Model

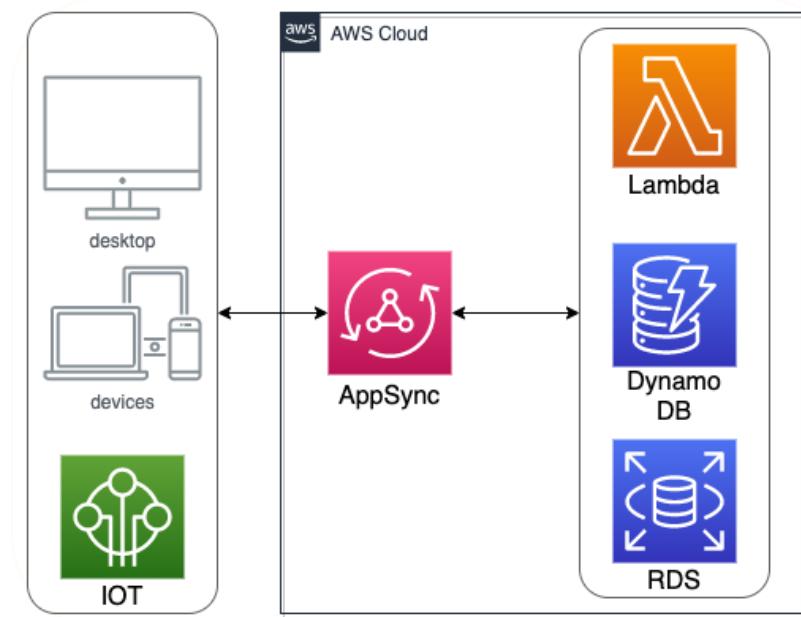
In 28  
Minutes

- 1000s of Lambda functions to manage, versioning, deployment etc
- Serverless projects can become a maintenance headache
- How to test serverless projects with Lambda, API Gateway and DynamoDB in your local?
- How to ensure that your serverless projects are adhering to best practices?
  - Tracing (X-Ray), CI/CD(CodeBuild, CodeDeploy, CodePipeline) etc
- Welcome SAM - Serverless Application Model
  - Open source framework for building serverless applications
  - Define a YAML with all the serverless resources you want:
    - Functions, APIs, Databases etc
  - BEHIND THE SCENES : Your configuration is used to create a AWS CloudFormation syntax to deploy your application

# AWS AppSync

In 28  
Minutes

- We are in multi device world
  - Want to synchronize app data across devices?
  - Want to create apps which work in off-line state?
  - Want to automatically sync data once user is back online?
- Welcome AWS AppSync
- Based on GraphQL
- App data can be accessed from anywhere
  - NoSQL data stores, RDS or Lambda
- (Alternative) Cognito Sync is limited to storing simple key-value pairs
  - AppSync recommended for almost all use cases



# AWS Step Functions

In 28  
Minutes

- Create a serverless workflow in 10 Minutes using a visual approach
- Orchestrate multiple AWS services into serverless workflows:
  - Invoke an AWS Lambda function
  - Run an Amazon Elastic Container Service or AWS Fargate task
  - Get an existing item from an Amazon DynamoDB table or put a new item into a DynamoDB table
  - Publish a message to an Amazon SNS topic
  - Send a message to an Amazon SQS queue
- Build workflows as a series of steps:
  - Output of one step flows as input into next step
  - Retry a step multiple times until it succeeds
  - Maximum duration of 1 year



Step Functions

# AWS Step Functions

In 28  
Minutes

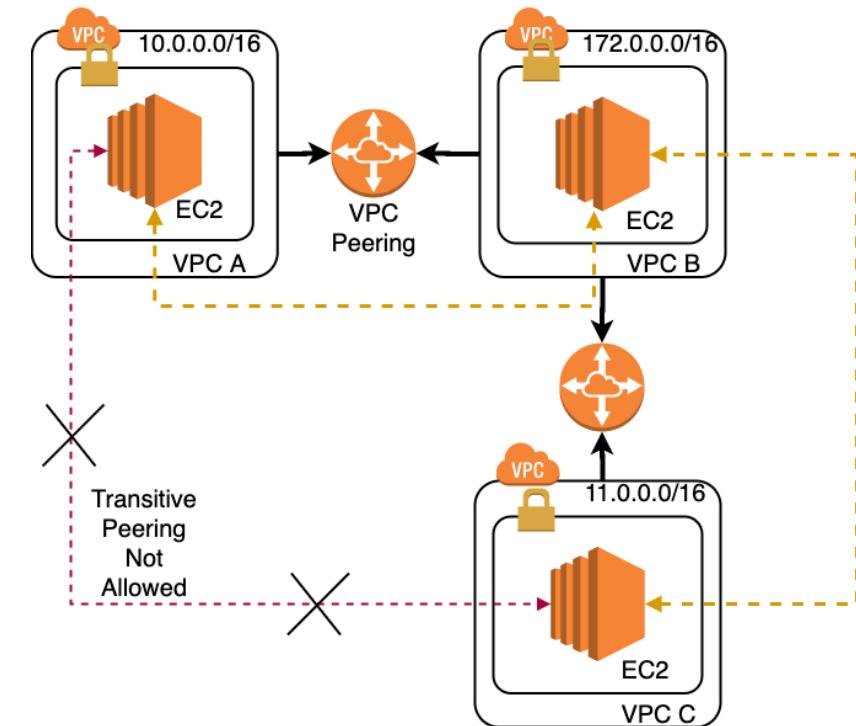
- Integrates with Amazon API Gateway
  - Expose API around Step Functions
  - Include human approvals into workflows
- (Use case) Long-running workflows
  - Machine learning model training, report generation, and IT automation
- (Use case) Short duration workflows
  - IoT data ingestion, and streaming data processing
- (Benefits) Visual workflows with easy updates and less code
- (Alternative) Amazon Simple Workflow Service (SWF)
  - Complex orchestration code (external signals, launch child processes)
- Step Functions is recommended for all new workflows  
UNLESS you need to write complex code for orchestration



# Extend and Secure Your VPCs - In AWS and To On-Premises

# VPC Peering

- Connect VPCs belonging to same or different AWS accounts irrespective of the region of the VPCs
- Allows private communication between the connected VPCs
- Peering uses a request/accept protocol
  - Owner of requesting VPC sends a request
  - Owner of the peer VPC has one week to accept
- Remember : Peering is not transitive
- Remember : Peer VPCs cannot have overlapping address ranges



# VPC Endpoint

In 28  
Minutes

- Securely connect your VPC to another service
- **Gateway endpoint**
  - Securely connect to Amazon S3 and DynamoDB
  - Endpoint serves as a target in your route table for traffic
  - Provide access to endpoint (endpoint, identity and resource policies)
- **Interface endpoint**
  - Securely connect to AWS services EXCEPT FOR Amazon S3 and DynamoDB
  - Powered by PrivateLink (keeps network traffic within AWS network)
  - Needs a elastic network interface (ENI) (entry point for traffic)
- (Avoid DDoS & MTM attacks) Traffic does NOT go thru internet
- (Simple) Does NOT need Internet Gateway, VPN or NAT



VPC Endpoint

# VPC Flow Logs

In 28  
Minutes

- Monitor network traffic
- Troubleshoot connectivity issues (NACL and/or security groups misconfiguration)
- Capture traffic going in and out of your VPC (network interfaces)
- Can be created for
  - a VPC
  - a subnet
  - or a network interface (connecting to ELB, RDS, ElastiCache, Redshift etc)
- Publish logs to Amazon CloudWatch Logs or Amazon S3
- Flow log records contain ACCEPT or REJECT
  - Is traffic is permitted by security groups or network ACLs?



VPC Flow Logs

# Troubleshoot using VPC Flow Logs

In 28  
Minutes

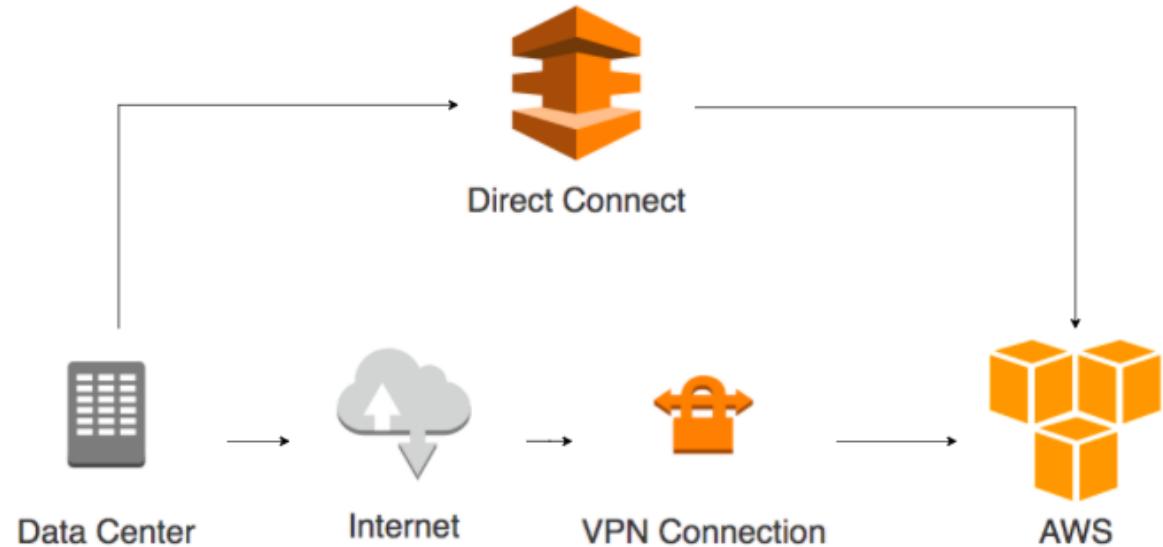


- Inbound traffic rules - NACL IN, SG IN, NACL OUT (SG OUT NOT checked)
  - If inbound request is rejected, SG or NACL could be mis-configured
  - If outbound response is rejected, NACL is mis-configured
- Outbound traffic rules - SG OUT, NACL OUT, NACL IN (SG IN NOT checked)
  - If outbound request is rejected, SG or NACL could be mis-configured
  - If inbound response is rejected, NACL is mis-configured
- Problem with response => Problem with NACL
- Problem with request could be problems with NACL or SG

# AWS and On-Premises - Overview

In 28  
Minutes

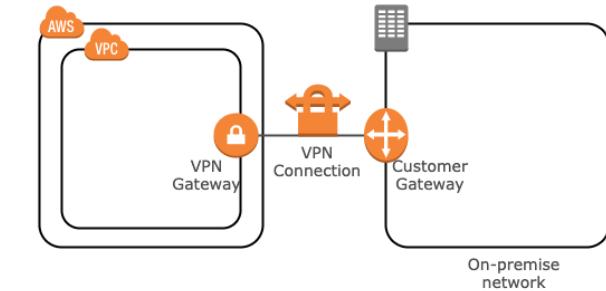
- AWS Managed VPN
  - IPsec VPN tunnels from VPC to customer network
- AWS Direct Connect (DX)
  - Private dedicated network connection from on-premises to AWS



# AWS Managed VPN

In 28  
Minutes

- IPsec VPN tunnels from VPC to customer network
- Traffic over internet - encrypted using IPsec protocol
- VPN gateway to connect one VPC to customer network
- Customer gateway installed in customer network
  - You need a Internet-routable IP address of customer gateway



# AWS Direct Connect (DC)

In 28  
Minutes

- Private dedicated network connection from on-premises to AWS
- Advantages:
  - Private network
  - Reduce your (ISP) bandwidth costs
  - Consistent Network performance because of private network
- Connection options:
  - Dedicated: Dedicated 1 Gbps or 10 Gbps network connections
  - Hosted: Shared 50Mbps to 10 Gbps network connections
- (REMEMBER) Establishing DC connection can take more than a month
- (REMEMBER) Establish a redundant DC for maximum reliability
- (REMEMBER) Direct Connect DOES NOT encrypt data (Private Connection ONLY)



Data Center



Direct Connect

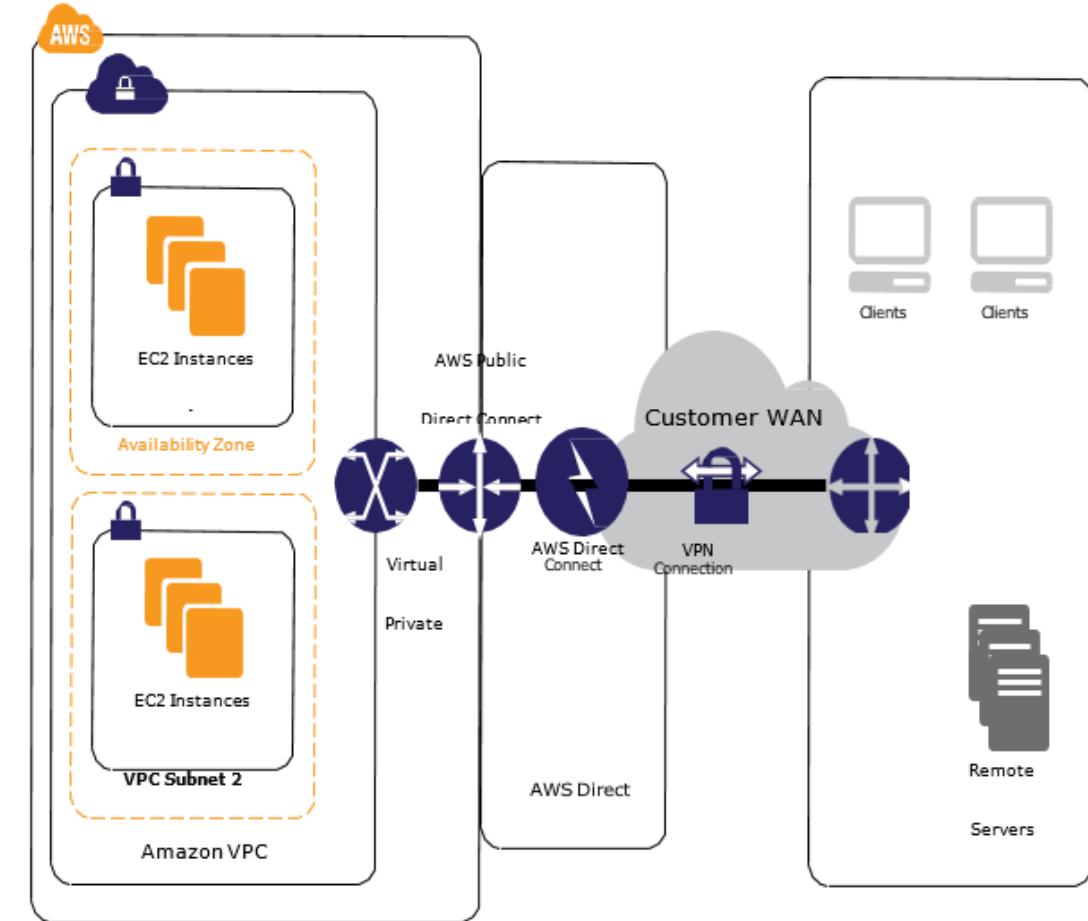


AWS

# AWS Direct Connect Plus VPN

In 28  
Minutes

- IPsec Site-to-Site VPN tunnel from an direct connect location to customer network
- Traffic is encrypted using IPsec protocol



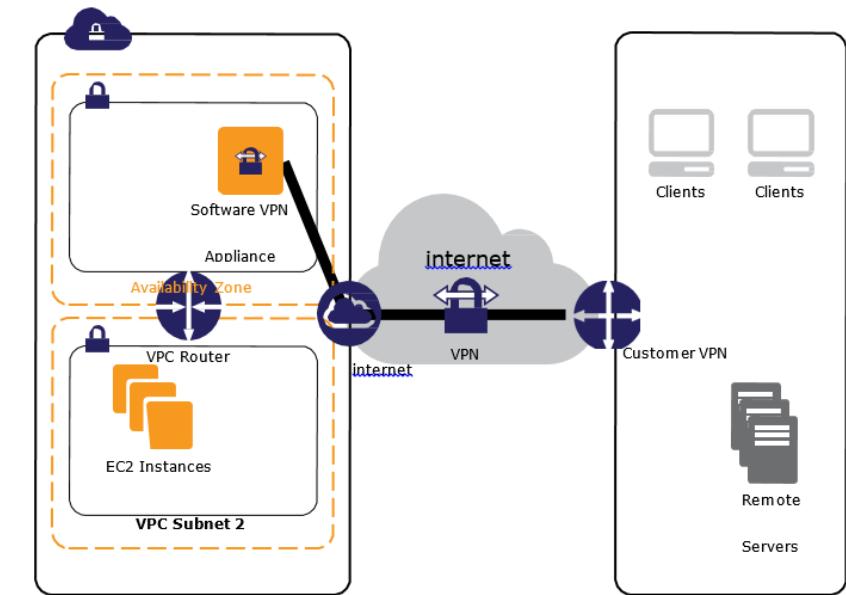
<https://docs.aws.amazon.com/whitepapers/latest/aws-vpc/>



# Software VPN

In 28  
Minutes

- Provides flexibility to fully manage both sides of your Amazon VPC connectivity
- Run software VPN appliance in your VPC
- Recommended for compliance - You need to manage both sides of connection
- Recommended when you use gateway devices which are not supported by Amazon VPN solution
- You are responsible for patches and updates to Software VPN appliance
- Software VPN appliance becomes a Single Point of Failure

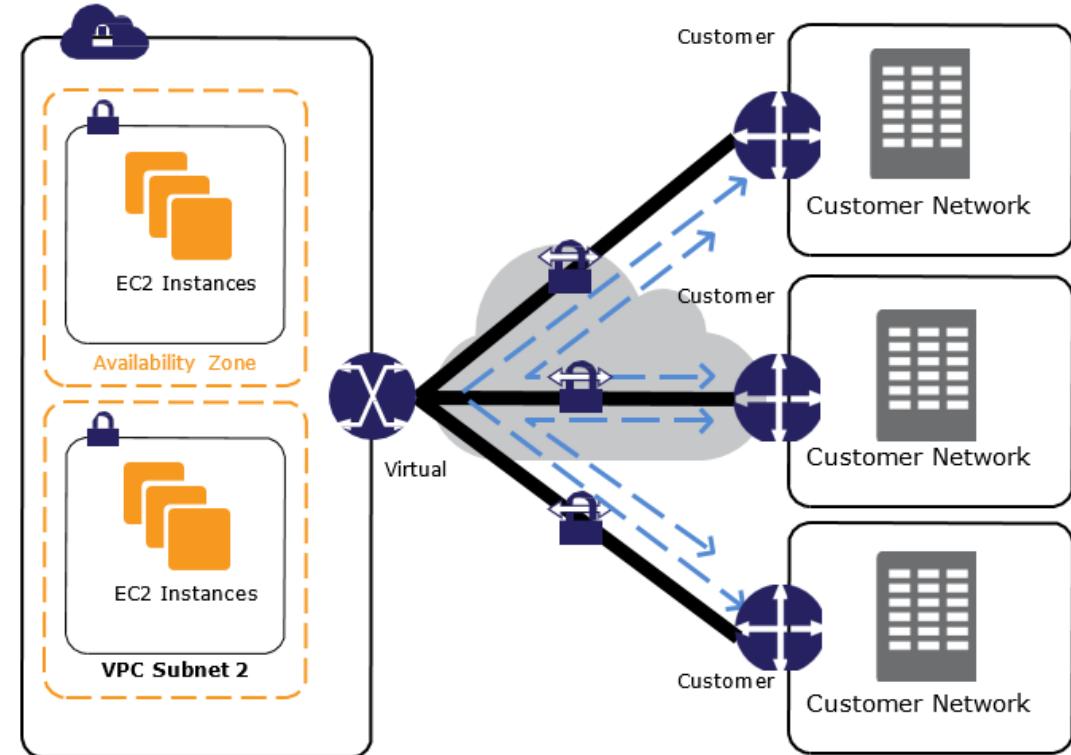


<https://docs.aws.amazon.com/whitepapers/latest/vpc-connectivity-options/aws-vpn-cloudhub-network-to-amazon.html>

# AWS VPN CloudHub

In 28  
Minutes

- Use either VPN or AWS Direct Connect to setup connectivity between multiple branch offices
- Operates on a simple hub-and-spoke model
- Uses Amazon VPC virtual private gateway with multiple gateways



<https://docs.aws.amazon.com/whitepapers/latest/aws-vpc-connectivity-options/aws-vpn-cloudhub-network-to-amazon.html>

# VPC Connections - Review

In 28  
Minutes

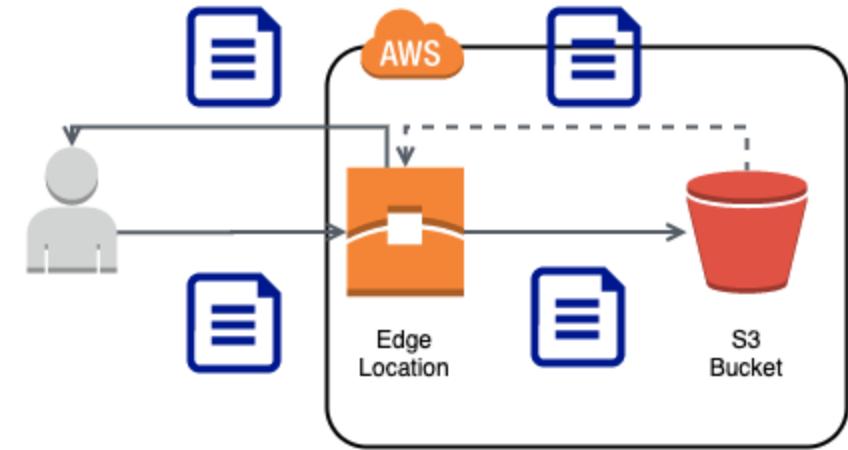
- VPC peering: Connect to other VPCs
- NAT gateways: Allow internet traffic from private subnets
- Internet gateway: Connect to internet
- AWS Direct Connect: Private pipe to on-premises
- AWS VPN: Encrypted (IPsec) tunnel over internet to on-premises

# Moving Data between AWS and On-premises

# Amazon S3 Transfer Acceleration

In 28  
Minutes

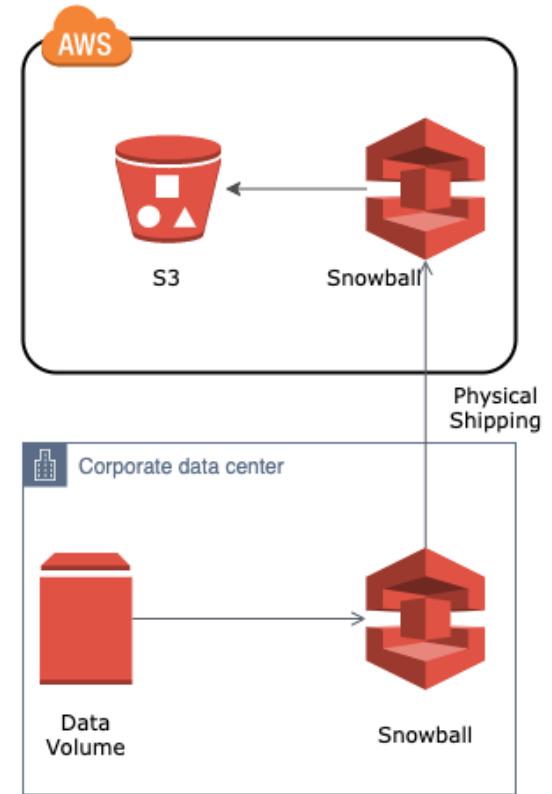
- Most basic option when you are transferring less data (upto a few terabytes) into S3
- Uses Amazon CloudFront's Edge Locations to enable fast transfer of files to/from your clients
- Enable S3 Transfer Acceleration and use endpoints
  - [s3-accelerate.amazonaws.com](https://s3-accelerate.amazonaws.com) or
  - [.s3-accelerate.dualstack.amazonaws.com](https://.s3-accelerate.dualstack.amazonaws.com)



# AWS Snowball

In 28  
Minutes

- Transfer dozens of terabytes to petabytes of data from on-premises to AWS
- 100TB (80 TB usable) per appliance
  - If needed, request multiple appliances
- Involves physical shipping
- Simple Process
  - Request for Snowball
  - Copy data
  - Ship it back
- Manage jobs with AWS Snowball console
- Data is automatically encrypted with KMS (AES-256)



# AWS Snowball

In 28  
Minutes

- Current versions of AWS Snowball use Snowball Edge devices
  - Provide both compute and storage
  - Pre-process data (using Lambda functions)
- Choose between
  - Storage optimized (24 vCPUs, 32 GiB RAM)
  - Compute optimized(52 vCPUs, 208 GiB RAM)
  - Compute optimized with GPU
- Choose Snowball if direct transfer takes over a week
  - 5TB can be transferred on 100Mbps line in a week at 80% utilization



AWS Snowball

# AWS Snowmobile

In 28  
Minutes



- How do I transfer dozens of petabytes to exabytes of data from on-premises to AWS for cloud migration?
- 100PB storage per truck
- If needed, use multiple trucks in parallel
- Data is automatically encrypted with KMS (AES-256)

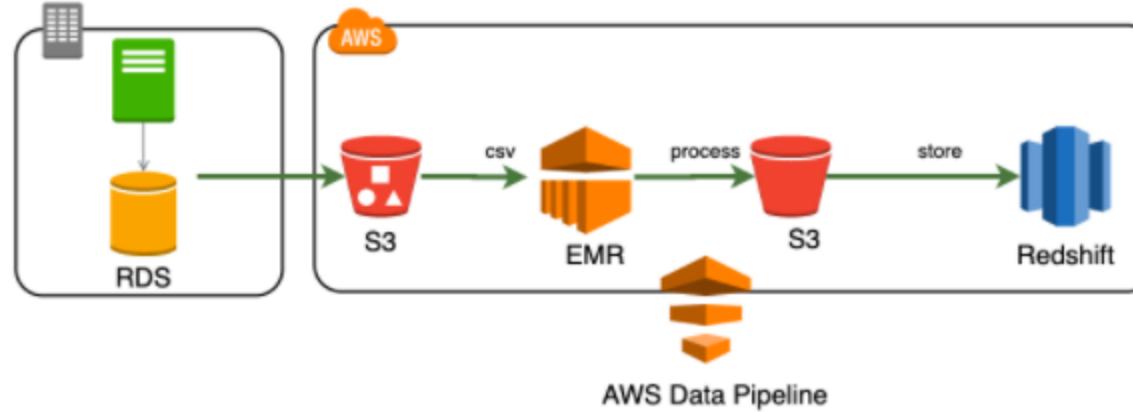
# AWS DataSync - Transfer File Storage to Cloud

In 28  
Minutes

- Secure and 10x faster (100s of TB) data transfers from/to AWS over internet or AWS Direct Connect
- Transfer from onpremise file storage (NFS, SMB) to S3, EFS or FSx for Windows
- Monitor progress using Amazon CloudWatch
- (Use cases) Data Migration, Data replication and Cold data archival
- (Alternative) Use AWS Snowball if you are bandwidth constrained or transferring data from remote, or disconnected
- (Alternative) Use S3 Transfer Acceleration when your applications are integrated with S3 API. If not, prefer AWS DataSync(Supports multiple destinations, built-in retry)
- (Integration) Migrate data using DataSync and use AWS Storage Gateway for ongoing updates from on-premises applications

# AWS Data Pipeline

In 28  
Minutes



- Process and move data (ETL) between S3, RDS, DynamoDB, EMR, On-premise data sources
- Create complex data processing workloads that are fault tolerant, repeatable, and highly available
- Launches required resources and tear them down after execution
- REMEMBER : NOT for streaming data!

# AWS Database Migration Service

In 28  
Minutes

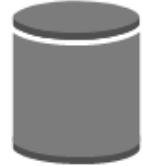


- Migrate databases to AWS while keeping source databases operational
  - Homogeneous Migrations (ex: Oracle to Oracle)
  - Heterogeneous Migrations (ex: Oracle to Amazon Aurora, MySQL to Amazon Aurora)
- Free for first 6 months when migrating to Aurora, Redshift or DynamoDB
- (AFTER MIGRATION) Keep databases in sync and pick right moment to switch
- (Use case) Consolidate multiple databases into a single target database
- (Use case) Continuous Data Replication can be used for Disaster Recovery

# AWS Schema Conversion Tool

In 28  
Minutes

- Migrate data from commercial databases and data warehouses to open source or AWS services
  - Preferred option for migrating data warehouse data to Amazon Redshift
- Migrate database schema (views, stored procedures, and functions) to compatible targets
- Features:
  - SCT assessment report
    - Analyze a database to determine the conversion complexity
  - Update source code (update embedded SQL in code)
  - Fan-in (multiple sources - single target)
  - Fan-out (single source - multiple targets)



Database

# Database Migration Service VS Schema Conversion Tool

In 28  
Minutes



- (Remember) SCT is part of DMS service
- DMS is preferred for homogeneous migrations
- SCT is preferred when schema conversion are involved
- DMS is for smaller workloads (less than 10 TB)
- SCT preferred for large data warehouse workloads
  - Prefer SCT for migrations to Amazon Redshift
- Only DMS provides continuous data replication after migration

# DevOps

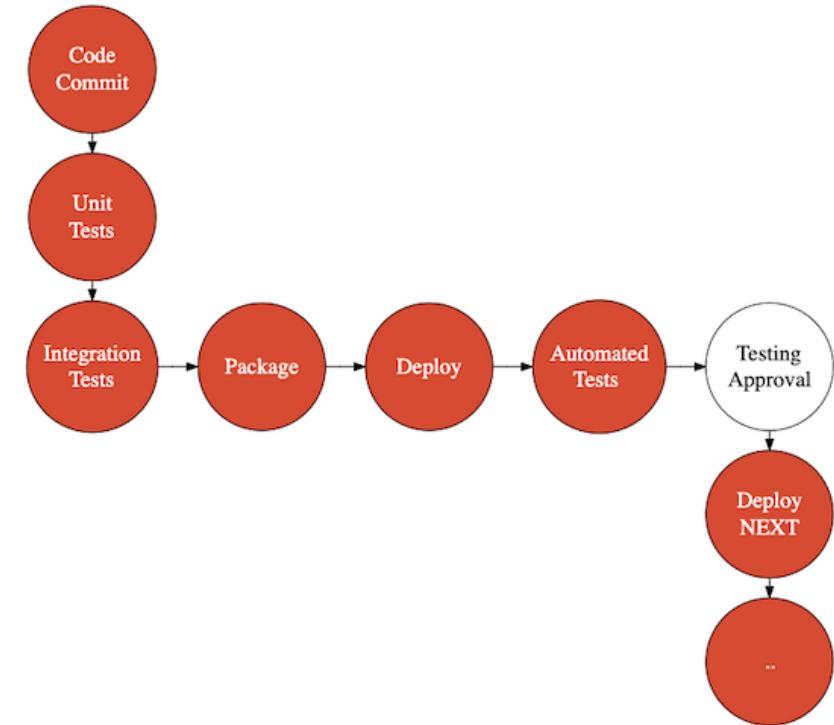


- Getting Better at "**Three Elements of Great Software Teams**"
  - Communication - Get teams together
  - Feedback - Earlier you find a problem, easier it is to fix
  - Automation - Automate testing, infrastructure provisioning, deployment, and monitoring

# DevOps - CI, CD

In 28  
Minutes

- Continuous Integration
  - Continuously run your tests and packaging
- Continuous Deployment
  - Continuously deploy to test environments
- Continuous Delivery
  - Continuously deploy to production



# DevOps - CI, CD Tools

In 28  
Minutes



Codecommit



Codepipeline



Codebuild



Codedeploy

- AWS CodeCommit - Private source control (Git)
- AWS CodePipeline - Orchestrate CI/CD pipelines
- AWS CodeBuild - Build and Test Code (application packages and containers)
- AWS CodeDeploy - Automate Deployment (EC2, ECS, Elastic Beanstalk, EKS, Lambda etc)



- Treat infrastructure the same way as application code
- Track your infrastructure changes over time (version control)
- Bring repeatability into your infrastructure
- Two Key Parts
  - Infrastructure Provisioning
    - Provisioning compute, database, storage and networking
    - Open source cloud neutral - Terraform
    - AWS Service - CloudFormation
  - Configuration Management
    - Install right software and tools on the provisioned resources
    - Open Source Tools - Chef, Puppet, Ansible
    - AWS Service - OpsWorks

# AWS CloudFormation - Introduction

In 28  
Minutes

- Lets consider an example:
  - I would want to create a new VPC and a subnet
  - I want to provision a ELB, ASG with 5 EC2 instances and an RDS database in the subnet
  - I would want to setup the right security groups
- AND I would want to create 4 environments
  - Dev, QA, Stage and Production!
- CloudFormation can help you do all these with a simple (actually NOT so simple) script!



CloudFormation

# AWS CloudFormation - Advantages

In 28  
Minutes

- Automate deployment and modification of AWS resources in a controlled, predictable way
- Avoid configuration drift
- Avoid mistakes with manual configuration
- Think of it as version control for your environments



CloudFormation

# AWS CloudFormation

20



CloudFormation

- All configuration is defined in a simple text file - JSON or YAML
  - I want a VPC, a subnet, a database and ...
- CloudFormation understands dependencies
  - Creates VPCs first, then subnets and then the database
- (Default) Automatic rollbacks on errors (Easier to retry)
  - If creation of database fails, it would automatic delete the subnet and VPC
- Version control your configuration file and make changes to it over time
- Free to use - Pay only for the resources provisioned
  - Get an automated estimate for your configuration

# AWS CloudFormation - Example 1 - JSON

In 28  
Minutes

```
{  
  "Resources" : {  
    "MyBucket" : {  
      "Type" : "AWS::S3::Bucket"  
      "Properties" : {  
        "AccessControl" : "PublicRead"  
      }  
    }  
  }  
}
```

# AWS CloudFormation - Example 2 - YAML

In 28  
Minutes

```
Resources:  
  MyBucket:  
    Type: AWS::S3::Bucket  
    Properties:  
      AccessControl: PublicRead
```

# AWS CloudFormation - Example 3

In 28  
Minutes

## Resources:

### Ec2Instance:

Type: 'AWS::EC2::Instance'

#### Properties:

ImageId: "ami-0ff8a91507f77f867"

InstanceType: t2.micro

#### SecurityGroups:

- !Ref InstanceSecurityGroup

### InstanceSecurityGroup:

Type: 'AWS::EC2::SecurityGroup'

#### Properties:

GroupDescription: Enable SSH access via port 22

#### SecurityGroupIngress:

- IpProtocol: tcp

FromPort: '22'

ToPort: '22'

CidrIp: 0.0.0.0/0

# AWS CloudFormation - Terminology

In 28  
Minutes

- **Template**
  - A CloudFormation JSON or YAML defining multiple resources
- **Stack**
  - A group of resources that are created from a CloudFormation template
  - In the earlier example, the stack contains an EC2 instance and a security group
- **Change Sets**
  - To make changes to stack, update the template
  - Change set shows what would change if you execute
  - Allows you to verify the changes and then execute

# AWS CloudFormation - Important template elements

In 28  
Minutes

```
{  
    "AWSTemplateFormatVersion" : "version date",  
    "Description" : "JSON string",  
    "Metadata" : {},  
    "Parameters" : {},  
    "Mappings" : {},  
    "Resources" : {},  
    "Outputs" : {}  
}
```

- Resources - What do you want to create?
  - One and only mandatory element
- Parameters - Values to pass to your template at runtime
  - Which EC2 instance to create? - ("t2.micro", "m1.small", "m1.large")
- Mappings - Key value pairs
  - Example: Configure different values for different regions
- Outputs - Return values from execution
  - See them on console and use in automation

# AWS CloudFormation - Mappings Example

In 28  
Minutes

```
"Mappings" : {  
    "RegionMap" : {  
        "us-east-1"      : { "AMI" : "AMI-A" },  
        "us-west-1"      : { "AMI" : "ami-B" },  
        "eu-west-1"      : { "AMI" : "ami-C" },  
        "ap-southeast-1" : { "AMI" : "ami-D" },  
        "ap-northeast-1" : { "AMI" : "ami-E" }  
    }  
}
```

# AWS CloudFormation - Remember

In 28  
Minutes

- Deleting a stack deletes all the associated resources
  - EXCEPT for resources with DeletionPolicy attribute set to "Retain"
  - You can enable termination protection for the entire stack
- Templates are stored in S3
- Use CloudFormation Designer to visually design templates
- AWS CloudFormation StackSets
  - Create, update, or delete stacks across multiple accounts and regions with a single operation



CloudFormation

# CloudFormation vs AWS Elastic Beanstalk

In 28  
Minutes



- (Do you know?) You can create an Elastic Beanstalk environment using CloudFormation!
- Think of Elastic Beanstalk as a pre-packaged CloudFormation template with a User Interface
  - You choose what you want
  - (Background) A Cloud Formation template is created and executed
  - The environment is ready!

# AWS OpsWorks - Configuration Management

In 28  
Minutes

- OpsWorks is used for Configuration Management
  - How do you ensure that 100 servers have the same configuration?
  - How can I make a change across 100 servers?
- Managed service based on Chef & Puppet
- One service for deployment and operations in cloud and on-premise environments
- Configuration - Chef recipes or cookbooks, Puppet manifests
- All metrics are sent to Amazon CloudWatch
- (IMPORTANT) All configuration management tools can also do infrastructure provisioning
  - However, I would recommend NOT doing that as they are not good at infrastructure provisioning



AWS Opsworks

# AWS Certification - FAQ

# High Availability

- High Availability - 99.99% or 99.9% - You can fail a few times
- Consider this problem:
  - You have an application deployed on EC2 instances (Load distribution using an ALB)
- How do you design for high availability in a single region (survive a loss of AZ) while being cost effective?
  - Need : 2 EC2 instances running all the time
    - 2 instances in AZ1 and 2 instances in AZ2
  - Need : 4 EC2 instances running all the time
    - 2 instances in AZ1 and 2 instances in AZ2 and 2 instances in AZ3

# High Availability vs Fault Tolerance

In 28  
Minutes

- Fault Tolerant - Zero chance of failure
- If you want fault tolerance, you need to take additional precautions
  - 2 EC2 instances running all the time
    - 2 instances in AZ1 and 2 instances in AZ2 and 2 instances in AZ3

# Data Transfer Costs

In 28  
Minutes

- Using Public IP addresses for communication between EC2 instances can get expensive.
  - Use Private IP Addresses
- Here are some of the relaxations that AWS provides:
  - Same Availability Zone - FREE - Data transfer between
    - Amazon EC2, Amazon RDS, Amazon Redshift, Amazon ElastiCache instances and Elastic Network Interfaces
  - Same Region - FREE - Data transfer between your EC2 instances and
    - Amazon S3, Amazon Glacier, Amazon DynamoDB
    - Amazon SNS, Amazon SQS, Amazon Kinesis
- (Best Practice) Maximize traffic that stays with an AZ (at least with a Region)



# More AWS Services



AWS Shield



Route53



CloudFront



EC2



ELB

- Shields from Distributed Denial of Service (DDoS) attacks
  - Disrupt normal traffic of a server by overwhelming it with a flood of Internet traffic
- Protect
  - Amazon Route 53
  - Amazon CloudFront
  - AWS Global Accelerator
  - Amazon Elastic Compute Cloud (EC2) instances
  - Elastic Load Balancers (ELB)

# AWS Shield - Standard and Advanced

In 28  
Minutes

- AWS Shield Standard
  - Zero Cost. Automatically enabled.
  - Protection against common infrastructure (layer 3 and 4) DDoS attacks
- AWS Shield Advanced
  - Paid service
  - Enhanced protection for Amazon EC2, Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator, and Amazon Route 53
  - 24x7 access to the AWS DDoS Response Team (DRT)
  - Protects your AWS bill from usage spikes as a result of a DDoS attack
- Protect any web application (from Amazon S3 or external) from DDoS by putting Amazon CloudFront enabled with AWS Shield in front of it



AWS Shield



CloudFront

# AWS WAF - Web Application Firewall

In 28  
Minutes

- AWS WAF protect your web applications from OWASP Top 10 exploits, CVE and a lot more!
  - OWASP (Open Web Application Security Project) Top 10
    - List of broadly agreed "most critical security risks to web applications"
    - Examples : SQL injection, cross-site scripting etc
  - Common Vulnerabilities and Exposures (CVE) is a list of information-security vulnerabilities and exposures
- Can be deployed on Amazon CloudFront, Application Load Balancer, Amazon API Gateway
- Customize rules & trigger realtime alerts (CloudWatch Alarms)
- Web traffic filtering : block attacks
  - Filter traffic based on IP addresses, geo locations, HTTP headers and body (block attacks from specific user-agents, bad bots, or content scrapers)



AWS WAF

# AWS Organizations

20



Organizations

- Organizations typically have multiple AWS accounts
  - Different business units
  - Different environments
- How do you centralize your management (billing, access control, compliance and security) across multiple AWS accounts?
- Welcome AWS Organizations!
- Organize accounts into Organizational Units (OU)
- Provides API to automate creation of new accounts

# AWS Organizations - Features

In 28  
Minutes

- One consolidated bill for all AWS accounts
- Centralized compliance management for AWS Config Rules
- Send AWS CloudTrail data to one S3 bucket (across accounts)
- AWS Firewall Manager to manage firewall rules (across accounts)
  - AWS WAF, AWS Shield Advanced protections and Security Groups
- Use Service control policies (SCPs) to define restrictions for actions (across accounts):
  - Prevent users from disabling AWS Config or changing its rules
  - Require Amazon EC2 instances to use a specific type
  - Require MFA to stop an Amazon EC2 instance
  - Require a tag upon resource creation



Organizations

# AWS Resource Access Manager

In 28  
Minutes

- Share AWS resources with any AWS account or within your AWS Organization
  - AWS Transit Gateways
  - Subnets
  - AWS License Manager configurations
  - Amazon Route 53 Resolver rules
- Reduce Operational Overhead
- Optimize Costs

# AWS Trusted Advisor

In 28  
Minutes

- Recommendations for cost optimization, performance, security and fault tolerance
  - Red - Action recommended Yellow - investigate and Green - Good to go
- All AWS customers get 4 checks for free:
  - Service limits (usage > 80%)
  - Security groups having unrestricted access (0.0.0.0/0)
  - Proper use of IAM
  - MFA on Root Account
- Business or Enterprise AWS support plan provides over 50 checks
  - Disable those you are not interested in
  - How much will you save by using Reserved Instances?
  - How does your resource utilization look like? Are you right sized?



Trusted Advisor

# AWS Trusted Advisor Recommendations

In 28  
Minutes

- Cost Optimization
  - Highlight unused resources
  - Opportunities to reduce your costs
- Security
  - Settings that can make your AWS solution more secure
- Fault Tolerance
  - Increase resiliency of your AWS solution
  - Redundancy improvements, over-utilized resources
- Performance
  - Improve speed and responsiveness of your AWS solutions
- Service Limits
  - Identify if your service usage is more than 80% of service limits



Trusted Advisor

# AWS Service Quotas

In 28  
Minutes

- AWS account has Region-specific default quotas or limits for each service
  - You don't need to remember all of them :)
- Service Quotas allows you to manage your quotas for over 100 AWS services, from one location

# AWS Directory Service

In 28  
Minutes

- Provide AWS access to on-premise users without IAM users
- Managed service deployed across multiple AZs
- Option 1 : AWS Directory Service for Microsoft AD
  - More than 5000 Users
  - Trust relationship needed between AWS and on-premise directory
- Option 2 : Simple AD
  - Less than 5000 users
  - Powered by Samba4 and compatible with Microsoft AD
  - Does not support trust relationships with other AD domains
- Option 3 : AD Connector
  - Use your existing on-premise directory with AWS cloud services
  - Your users use existing credentials to access AWS resources



Directory Service

# AWS Workspaces

In 28  
Minutes

- Desktop-as-a-Service (DaaS)
- Provision Windows or Linux desktops in minutes
- Eliminate traditional desktop management - Virtual Desktop Infrastructure (VDI)

# AWS Systems Manager Parameter Store

In 28  
Minutes

- Manage application environment configuration and secrets
  - database connections, password etc
- Supports hierarchical structure
- Store configuration at one place
  - multiple applications
  - multiple environments
- Maintains history of configuration over a period of time
- Integrates with KMS, IAM, CloudWatch and SNS

# AWS Secrets Manager

In 28  
Minutes

- Rotate, Manage and retrieve database credentials, API keys, and other secrets for your applications
- Integrates with KMS(encryption), Amazon RDS, Amazon Redshift, and Amazon DocumentDB
- (KEY FEATURE) Rotate secrets automatically without impacting applications
- (KEY FEATURE) Service dedicated to secrets management
- Recommended for workloads needing HIPAA, PCI-DSS compliance

# AWS Elemental MediaConvert

In 28  
Minutes

- New video transcoding service
- Create high-quality video processing workflows
- Optimize video files for playback on virtually any device
- Convert between multiple media formats (MPEG-2, AVC, Apple ProRes, and HEVC)
- (Alternative) AWS Elastic Transcoder
  - Use AWS Elastic Transcoder to create WebM video, MP3 audio, or animated GIF files
  - For all other video processing use cases, AWS Elemental MediaConvert is recommended
- (Alternative) For live video, use AWS Elemental MediaLive

# Amazon Macie

In 28  
Minutes

- Fully managed data security and data privacy service
- Automatically discover, classify, and protect sensitive data in Amazon S3 buckets
- When migrating data to AWS use S3 for staging
  - Run Macie to discover secure data
- Uses machine learning
- Recognizes sensitive data
  - Example: personally identifiable information (PII) or intellectual property
- Provides you with dashboards and alerts
  - Gives visibility into how data is being accessed or moved

# AWS Single Sign On

In 28  
Minutes

- Cloud-based single sign-on (SSO) service
- Centrally manage SSO access to all of your AWS accounts
- Integrates with Microsoft AD (Supports using your existing corporate accounts)
- Supports Security Assertion Markup Language (SAML) 2.0
- Deep integration with AWS Organizations (Centrally manage access to multiple AWS accounts)
- One place auditing in AWS CloudTrail

# AWS Elasticsearch

In 28  
Minutes

- AWS Managed Service around Elasticsearch
- Supports the popular ELK stack
  - Elasticsearch for search and analytics
  - Logstash to ingest data from multiple sources
  - Kibana for visualization
- Use cases
  - Search (Provide fast search for websites)
  - Application monitoring (Get intelligence from your application logs)
  - Infrastructure monitoring (Get intelligence from your server logs)

# Architecture and Best Practices

# Well Architected Framework

In 28  
Minutes

- Helps cloud architects build application infrastructure which is:
  - Secure
  - High-performing
  - Resilient and
  - Efficient
- Five Pillars
  - Operational Excellence
  - Security
  - Reliability
  - Performance Efficiency
  - Cost Optimization



# Operational Excellence

In 28  
Minutes



AWS Lambda



CloudFormation



Codepipeline



AWS Config



Cloudwatch

- Avoid/Minimize effort and problems with
  - Provisioning servers
  - Deployment
  - Monitoring
  - Support

# Operational Excellence - Solutions and AWS services

In 28  
Minutes

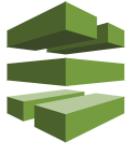
- Use Managed Services
  - You do not need to worry about managing servers, availability, durability etc
- Go serverless
  - Prefer Lambda to EC2!
- Automate with Cloud Formation
  - Use Infrastructure As Code
- Implement CI/CD to find problems early
  - CodePipeline
  - CodeBuild
  - CodeDeploy
- Perform frequent, small reversible changes



AWS Lambda



CloudFormation



Codepipeline



Codebuild



Codedeploy

# Operational Excellence - Solutions and AWS services

In 28  
Minutes

- Prepare: for failure
  - Game days
  - Disaster recovery exercises
  - Implement standards with AWS Config rules
- Operate: Gather Data and Metrics
  - CloudWatch (Logs agent), Config, Config Rules, CloudTrail, VPC Flow Logs and X-Ray (tracing)
- Evolve: Get intelligence
  - Use Amazon Elasticsearch to analyze your logs



AWS Config



Cloudwatch



AWS CloudTrail



AWS X-Ray



Amazon ES

# Security Pillar

In 28  
Minutes



AWS IAM



AWS Shield



AWS WAF



AWS KMS



Cloud HSM

- Principle of least privilege for least time
- Security in Depth - Apply security in all layers
- Protect Data in Transit and at rest
- Actively monitor for security issues
- Centralize security policies for multiple AWS accounts

# Security Pillar - Principle of least privilege for least time

In 28  
Minutes



AWS IAM

- Use temporary credentials when possible (IAM roles, Instance profiles)
- Use IAM Groups to simplify IAM management
- Enforce strong password practices
- Enforce MFA
- Rotate credentials regularly

# Security Pillar - Security in Depth

In 28  
Minutes

- VPCs and Private Subnets
  - Security Groups
  - Network Access Control List (NACL)
- Use hardened EC2 AMIs (golden image)
  - Automate patches for OS, Software etc
- Use CloudFront with AWS Shield for DDoS mitigation
- Use WAF with CloudFront and ALB
  - Protect web applications from CSS, SQL injection etc
- Use CloudFormation
  - Automate provisioning infrastructure that adheres to security policies



VPC



EC2 AMI



AWS Shield



AWS WAF



CloudFormation

# Security Pillar - Protecting Data at Rest

In 28  
Minutes

- Enable Versioning (when available)
- Enable encryption - KMS and Cloud HSM
  - Rotate encryption keys
- Amazon S3
  - SSE-C, SSE-S3, SSE-KMS
- Amazon DynamoDB
  - Encryption Client, SSE-KMS
- Amazon Redshift
  - AWS KMS and AWS CloudHSM
- Amazon EBS, Amazon SQS and Amazon SNS
  - AWS KMS
- Amazon RDS
  - AWS KMS, TDE



AWS KMS



Cloud HSM

# Security Pillar - Protecting Data in Transit

In 28  
Minutes



Certificate Manager

- Data coming in and going out of AWS
- By default, all AWS API use HTTPS/SSL
- You can also choose to perform client side encryption for additional security
- Ensure that your data goes through AWS network as much as possible
  - VPC Endpoints and AWS PrivateLink

# Security Pillar - Detect Threats

In 28  
Minutes



Cloudwatch



Organizations

- Actively monitor for security issues:
  - Monitor CloudWatch Logs
  - Use Amazon GuardDuty to detect threats and continuously monitor for malicious behavior
- Use AWS Organization to centralize security policies for multiple AWS accounts

# Reliability

In 28  
Minutes



AWS Lambda



Amazon SQS



Amazon SNS



API Gateway

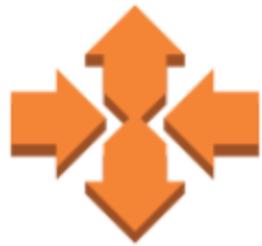


AutoScaling

- Ability to
  - Recover from infrastructure and application issues
  - Adapt to changing demands in load

# Reliability - Best Practices

In 28  
Minutes



AutoScaling



CW Alarm



Cloudwatch

- Automate recovery from failure
  - Health checks and Auto scaling
  - Managed services like RDS can automatically switch to standby
- Scale horizontally
  - Reduces impact of single failure
- Maintain Redundancy
  - Multiple Direct Connect connections
  - Multiple Regions and Availability Zones

# Reliability - Best Practices

In 28  
Minutes



AWS Lambda



Amazon SQS



Amazon SNS



API Gateway

- Prefer serverless architectures
- Prefer loosely coupled architectures
  - SQS, SNS
- **Distributed System Best Practices**
  - Use Amazon API Gateway for throttling requests
  - AWS SDK provides retry with exponential backoff

# Loosely coupled architectures

In 28  
Minutes

- ELB
  - Works in tandem with AWS auto scaling
- Amazon SQS
  - Polling mechanism
- Amazon SNS
  - Publish subscribe pattern
  - Bulk notifications and Mobile push support
- Amazon Kinesis
  - Handle event streams
  - Multiple clients
  - Each client can track their stream position



ELB



Amazon SNS



Amazon SQS



Kinesis

# Troubleshooting on AWS - Quick Review

In 28  
Minutes

Option	Details	When to Use
Amazon S3 Server Access Logs	S3 data request details - request type, the resources requested, and the date and time of request	Troubleshoot bucket access issues and data requests
Amazon ELB Access Logs	Client's IP address, latencies, and server responses	Analyze traffic patterns and troubleshoot network issues
Amazon VPC Flow Logs	Monitor network traffic	Troubleshoot network connectivity and security issues

# Troubleshooting on AWS - Quick Review

In 28  
Minutes

Option	Details	When to Use
Amazon CloudWatch	Monitor metrics from AWS resources	Monitoring
Amazon CloudWatch Logs	Store and Analyze log data from Amazon EC2 instances and on-premises servers	Debugging application issues and Monitoring
AWS Config	AWS resource inventory. History. Rules.	Inventory and History
Amazon CloudTrail	History of AWS API calls made via AWS Management Console, AWS CLI, AWS SDKs etc.	Auditing and troubleshooting. Determine who did what, when, and from where.

# Performance Efficiency

In 28  
Minutes

- Meet needs with minimum resources (efficiency)
- Continue being efficient as demand and technology evolves

# Performance Efficiency - Best Practices

In 28  
Minutes



AWS Lambda



API Gateway



Cloudwatch



Amazon SQS

- Use Managed Services
  - Focus on your business instead of focusing on resource provisioning and management
- Go Serverless
  - Lower transactional costs and less operational burden
- Experiment
  - Cloud makes it easy to experiment
- Monitor Performance
  - Trigger CloudWatch alarms and perform actions through Amazon SQS and Lambda

# Performance Efficiency - Choose the right solution

In 28  
Minutes

- Compute
  - EC2 instances vs Lambda vs Containers
- Storage
  - Block, File, Object
- Database
  - RDS vs DynamoDB vs RedShift ..
- Caching
  - ElastiCache vs CloudFront vs DAX vs Read Replicas
- Network
  - CloudFront, Global Accelerator, Route 53, Placement Groups, VPC endpoints, Direct Connect
- Use product specific features
  - Enhanced Networking, S3 Transfer Acceleration, EBS optimized instances



AWS Lambda



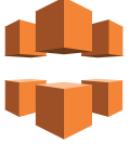
Amazon S3



DynamoDB



ElastiCache



CloudFront

# Cost Optimization

In 28  
Minutes

- Run systems at lowest cost



AutoScaling



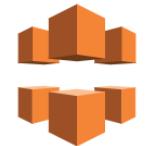
AWS Lambda



Trusted Advisor



Cloudwatch



CloudFront

# Cost Optimization - Best Practices

In 28  
Minutes

- Match supply and demand
  - Implement Auto Scaling
  - Stop Dev/Test resources when you don't need them
  - Go Serverless
- Track your expenditure
  - Cost Explorer to track and analyze your spend
  - AWS Budgets to trigger alerts
  - Use tags on resources



AutoScaling



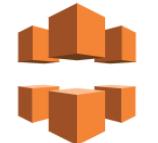
AWS Lambda



Trusted Advisor



Cloudwatch



CloudFront

# Cost Optimization - Choose Cost-Effective Solutions

In 28  
Minutes

- Right-Sizing : Analyze 5 large servers vs 10 small servers
  - Use CloudWatch (monitoring) and Trusted Advisor (recommendations) to right size your resources
- Email server vs Managed email service (charged per email)
- On-Demand vs Reserved vs Spot instances
- Avoid expensive software : MySQL vs Aurora vs Oracle
- Optimize data transfer costs using AWS Direct Connect and Amazon CloudFront



AutoScaling



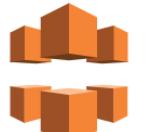
AWS Lambda



Trusted Advisor



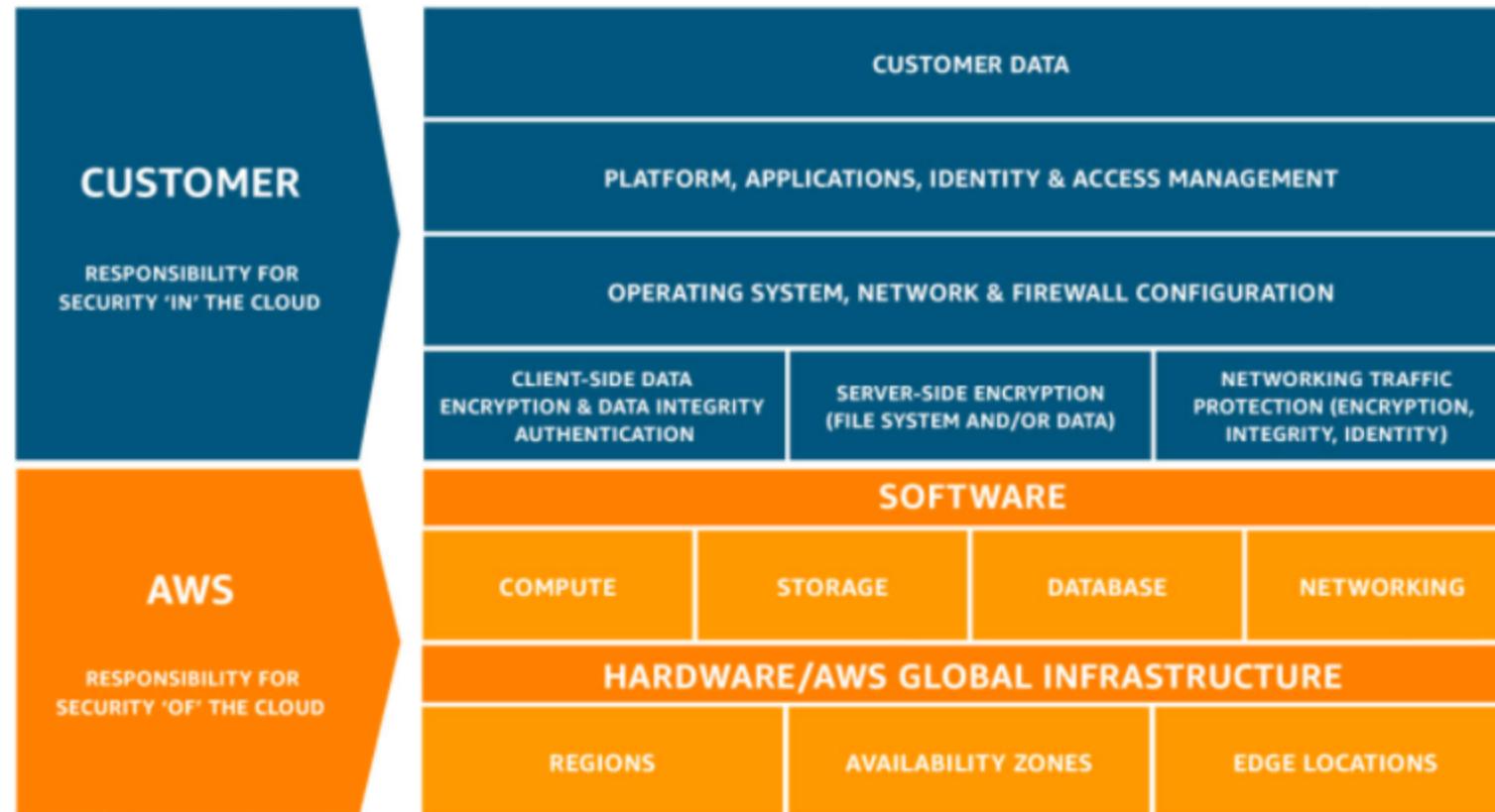
Cloudwatch



CloudFront

# Shared Responsibility Model

In 28  
Minutes



<https://aws.amazon.com/compliance/shared-responsibility-model/>

Security & Compliance is shared responsibility between AWS and customer

# Shared Responsibility Model - Amazon EC2

In 28  
Minutes



EC2



Security Group



EC2 AMI

- Amazon EC2 instances is Infrastructure as a Service (IaaS)
- You are responsible for
  - Guest OS (incl. security patches)
  - Application software installed
  - Configuring Security Groups (or firewalls)
- AWS is responsible for infrastructure layer only

# Shared Responsibility Model - Managed Services

In 28  
Minutes

- Amazon S3 & DynamoDB are managed services
- AWS manages infrastructure layer, OS, and platform
- You are responsible for
  - Managing your data
  - Managing security of data at rest(encryption)
  - Managing security of data in transit
    - Mandating SSL/HTTPS
    - Using the right network - AWS global network or dedicated private network when possible
  - Managing access to the service
    - Configure right permissions (IAM users/roles/user policies/resource policies)
    - (FOR AWS RDS) Managing in database users
    - Configuring the right security groups (control inbound and outbound traffic)
    - Disabling external access (public vs private)



Amazon S3



DynamoDB

# Get Ready

# Certification Resources

In 28  
Minutes

Title	Link
Certification - Home Page	<a href="https://aws.amazon.com/certification/certified-solutions-architect-associate/">https://aws.amazon.com/certification/certified-solutions-architect-associate/</a>
AWS Architecture Home Page	<a href="https://aws.amazon.com/architecture/">https://aws.amazon.com/architecture/</a>
AWS FAQs	<a href="https://aws.amazon.com/faqs/">https://aws.amazon.com/faqs/</a> (EC2, S3, VPC, RDS, SQS etc)

# Certification Exam

In 28  
Minutes

- Multiple Choice Questions
  - Type 1 : Single Answer - 4 options and 1 right answer
  - Type 2 : Multiple Answer - 5 options and 2 right answers
- No penalty for wrong answers
  - Feel free to guess if you do not know the answer
- 65 questions and 130 minutes
  - Ask for 30 extra minutes BEFORE registering if you are non native English speaker
- Result immediately shown after exam completion
- Email with detailed scores (a couple of days later)

# Certification Exam - My Recommendations

In 28  
Minutes

- Read the entire question
- Read all answers at least once
- Identify and write down the key parts of the question:
  - Features: serverless, key-value, relational, auto scaling
  - Qualities: cost-effective, highly available, fault tolerant
- If you do NOT know the answer, eliminate wrong answers first
- Mark questions for future consideration and review them before final submission

# Registering for Exam

In 28  
Minutes

- Certification - Home Page - <https://aws.amazon.com/certification/certified-solutions-architect-associate/> |

# You are all set!

# Let's clap for you!

In 28  
Minutes

- You have a lot of patience! Congratulations
- You have put your best foot forward to be an AWS Solution Architect
- Make sure you prepare well and
- Good Luck!

# Do Not Forget!

In 28  
Minutes

- Recommend the course to your friends!
  - Do not forget to review!
- Your Success = My Success
  - Share your success story with me on LinkedIn (Ranga Karanam)
  - Share your success story and lessons learnt in Q&A with other learners!

