# AWS NOTES
# Index

**1. Introduction to AWS**
**2. Amazon Simple Storage Service and Amazon Glacier  Storage**
**3. AWS Identity and Access Management (IAM)**
**4. Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Elastic Block**
**5. Amazon Virtual Private Cloud (Amazon VPC)**
**6. Elastic Load Balancing, Amazon CloudWatch, and  Auto Scaling**
**7. Databases and AWS**
**8. SQS, SWF, and SNS**
**9. Domain Name System (DNS) and Amazon Route 53**
**10. Amazon ElastiCache**
**11. Additional Key Services**
**12. Security on AWS**

# 1. Introduction

## Problems with traditional IT approach

- Pay for the rent for the data center
- Pay for power supply, cooling, maintenance
- Adding and replacing hardware takes time
- Scaling is limited
- Hire 24/7 team to monitor the infrastructure
- How to deal with disasters? (Earthquake, power shutdown, fire...)

- Can we externalize all this?



## What is Cloud Computing?

- Cloud computing is the on-demand delivery of compute power, database storage, applications, and other IT resources
- Through a cloud services platform with pay-as-you-go pricing
- You can provision exactly the right type and size of computing resources you need
- You can access as many resources as you need, almost instantly
- Simple way to access servers, storage, databases and a set of application services

- Amazon Web Services owns and maintains the network-connected hardware required for these application services, while you provision and use what you need via a web application.

# You've been using some Cloud services



**Gmail**

- E-mail cloud service
- Pay for ONLY your emails stored (no infrastructure, etc.)




**Dropbox**

- Cloud Storage Service
- Originally built on AWS



**Netflix**

- Built on AWS
- Video on Demand

# Six Advantages of Cloud Computing

- Trade capital expense (CAPEX) for operational expense (OPEX)
  - Pay On-Demand: don't own hardware
  - Reduced Total Cost of Ownership (TCO) & Operational Expense (OPEX)
- Benefit from massive economies of scale
  - Prices are reduced as AWS is more efficient due to large scale
- Stop guessing capacity
  - Scale based on actual measured usage
- Increase speed and agility
- Stop spending money running and maintaining data centers
- Go global in minutes: leverage the AWS global infrastructure

# Problems solved by the Cloud

- Flexibility: change resource types when needed
- Cost-Effectiveness: pay as you go, for what you use
- Scalability: accommodate larger loads by making hardware stronger or adding additional nodes
- Elasticity: ability to scale out and scale-in when needed
- High-availability and fault-tolerance: build across data centers
- Agility: rapidly develop, test and launch software applications

# The Deployment Models of the Cloud

## Private Cloud:

- Cloud services used by a single organization, not exposed to the public.
- Complete control
- Security for sensitive applications
- Meet specific business needs

## Public Cloud:

- Cloud resources owned and operated by a third-party cloud service provider delivered over the Internet.
- Six Advantages of Cloud Computing

## Hybrid Cloud:

- Keep some servers on premises and extend some capabilities to the Cloud
- Control over sensitive assets in your private infrastructure
- Flexibility and cost-effectiveness of the public cloud



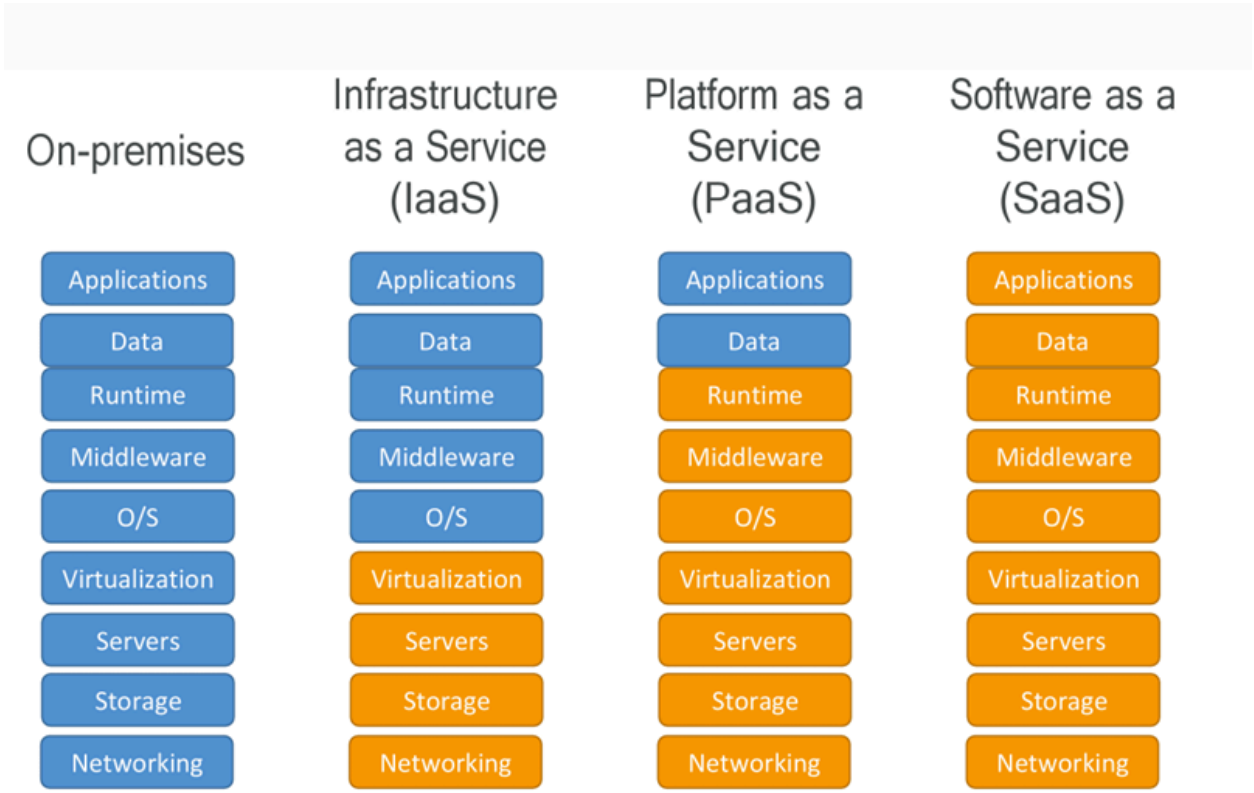






# Types of Cloud Computing

- Infrastructure as a Service (IaaS)
  - Provide building blocks for cloud IT
  - Provides networking, computers, data storage space
  - Highest level of flexibility
  - Easy parallel with traditional on-premises IT
- Platform as a Service (PaaS)
  - Removes the need for your organization to manage the underlying infrastructure
  - Focus on the deployment and management of your applications
- Software as a Service (SaaS)
  - Completed product that is run and managed by the service provider



# Example of Cloud Computing Types

- Infrastructure as a Service:
  - Amazon EC2 (on AWS)
  - GCP, Azure, Rackspace, Digital Ocean, Linode
- Platform as a Service:
  - Elastic Beanstalk (on AWS)
  - Heroku, Google App Engine (GCP), Windows Azure (Microsoft)
- Software as a Service:
  - Many AWS services (ex: Rekognition for Machine Learning)
  - Google Apps (Gmail), Dropbox, Zoom





# Pricing of the Cloud - Quick Overview

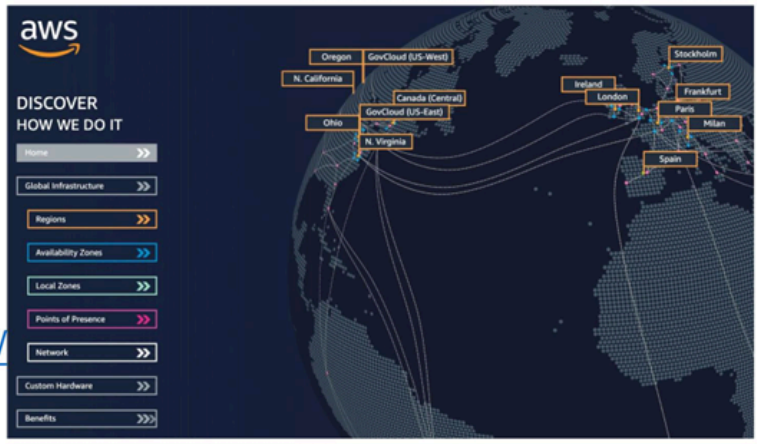
- AWS has 3 pricing fundamentals, following the pay-as-you-go pricing model
- Compute:
  - Pay for compute time
- Storage:
  - Pay for data stored in the Cloud
- Data transfer OUT of the Cloud:
  - Data transfer IN is free
- Solves the expensive issue of traditional IT



## AWS Global Infrastructure

- AWS Regions
- AWS Availability Zones
- AWS Data Centers
- AWS Edge Locations / Points of Presence

- https://infrastructure.aws/




# 2. Amazon S3

## Amazon S3 - Buckets

- Amazon S3 allows people to store objects (files) in "buckets" (directories)
- Buckets must have a globally unique name (across all regions all accounts)
- Buckets are defined at the region level
- S3 looks like a global service but buckets are created in a region
- Naming convention
  - No uppercase, No underscore
  - 3-63 characters long
  - Not an IP
  - Must start with lowercase letter or number
  - Must NOT start with the prefix xn--
  - Must NOT end with the suffix -s3alias



S3 Bucket

# Amazon S3 - Objects

Objects (files) have a Key

The key is the FULL path:

- s3://my-bucket/my_file.txt
- s3://my-bucket/my_folder1/another_folder/my_file.txt

- The key is composed of prefix + object name
  - s3://my-bucket/my_folder1/another_folder/my_file.txt
- There's no concept of "directories" within buckets (although the UI will trick you to think otherwise)
- Just keys with very long names that contain slashes ("/")



Object



S3 Bucket with Objects

# Amazon S3 - Objects (cont.)



- Object values are the content of the body:
  - Max. Object Size is 5TB (5000GB)
  - If uploading more than 5GB, must use "multi-part upload"

- Metadata (list of text key / value pairs - system or user metadata)
- Tags (Unicode key / value pair - up to 10) - useful for security / lifecycle
- Version ID (if versioning is enabled)

# S3 Bucket Policies

- JSON based policies
  - Resources: buckets and objects
  - Effect: Allow / Deny
  - Actions: Set of API to Allow or Deny
  - Principal:The account or user to apply the policy to

- Use S3 bucket for policy to:
  - Grant public access to the bucket
  - Force objects to be encrypted at upload
  - Grant access to another account (Cross Account)

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

# Amazon S3 - Static Website Hosting

- S3 can host static websites and have them accessible on the Internet

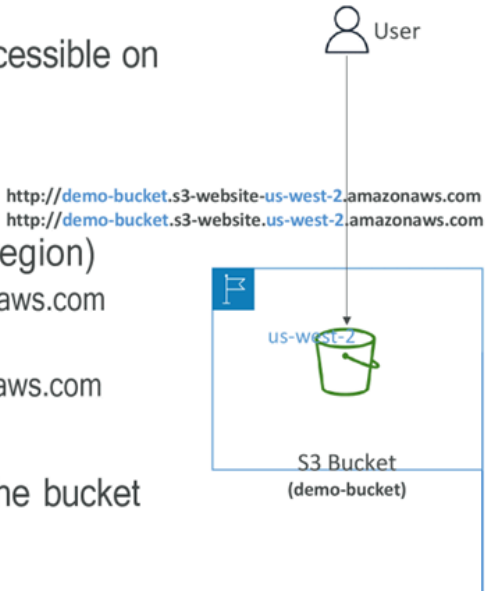
The website URL will be (depending on the region)

- http://*bucket-name*.s3-website-*aws-region*.amazonaws.com

OR

- http://*bucket-name*.s3-website.*aws-region*.amazonaws.com

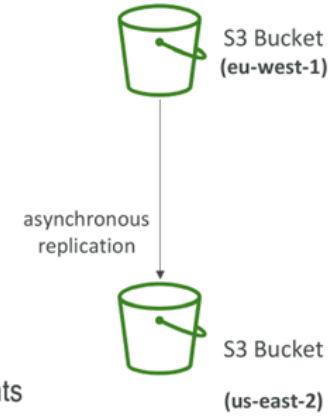
- If you get a 403 Forbidden error, make sure the bucket policy allows public reads!



# Amazon S3 – Replication (CRR & SRR)



- Must enable Versioning in source and destination buckets
- Cross-Region Replication (CRR)
- Same-Region Replication (SRR)
- Buckets can be in different AWS accounts
- Copying is asynchronous




- Must give proper IAM permissions to S3

- Use cases:
  - CRR - compliance, lower latency access, replication across accounts
  - SRR - log aggregation, live replication between production and test accounts

# S3 Storage Classes

- Amazon S3 Standard - General Purpose
- Amazon S3 Standard-Infrequent Access (IA)
- Amazon S3 One Zone-Infrequent Access
- Amazon S3 Glacier Instant Retrieval
- Amazon S3 Glacier Flexible Retrieval
- Amazon S3 Glacier Deep Archive
- Amazon S3 Intelligent Tiering

- Can move between classes manually or using S3 Lifecycle configurations

# S3 Durability and Availability

- Durability:
  - High durability (99.999999999%, 11 9's) of objects across multiple AZ
  - If you store 10,000,000 objects with Amazon S3, you can on average expect to incur a loss of a single object once every 10,000 years
  - Same for all storage classes

- Availability:
  - Measures how readily available a service is
  - Varies depending on storage class
  - Example: S3 standard has 99.99% availability = not available 53 minutes a year

# S3 Storage Classes – Infrequent Access

- For data that is less frequently accessed, but requires rapid access when needed
- Lower cost than S3 Standard

- Amazon S3 Standard-Infrequent Access (S3 Standard-IA)
  - 99.9% Availability
  - Use cases: Disaster Recovery, backups



- Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)
  - High durability (99.999999999%) in a single AZ; data lost when AZ is destroyed
  - 99.5% Availability
  - Use Cases: Storing secondary backup copies of on-premise data, or data you can recreate



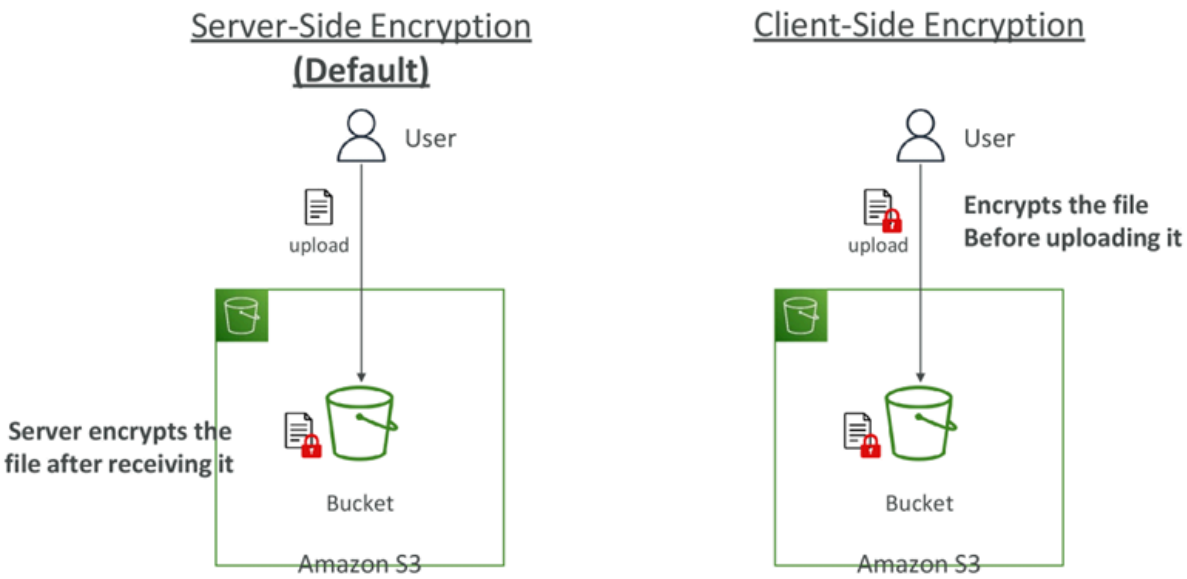
# Amazon S3 Glacier Storage Classes

- Low-cost object storage meant for archiving / backup
- Pricing: price for storage + object retrieval cost

- Amazon S3 Glacier Instant Retrieval
  - Millisecond retrieval, great for data accessed once a quarter
  - Minimum storage duration of 90 days
- Amazon S3 Glacier Flexible Retrieval (formerly Amazon S3 Glacier):
  - Expedited (1 to 5 minutes), Standard (3 to 5 hours), Bulk (5 to 12 hours) - free
  - Minimum storage duration of 90 days
- Amazon S3 Glacier Deep Archive - for long term storage:
  - Standard (12 hours), Bulk (48 hours)
  - Minimum storage duration of 180 days



# S3 Encryption



## Amazon S3 - Summary

- Buckets vs Objects: global unique name, tied to a region
- S3 security: IAM policy, S3 Bucket Policy (public access), S3 Encryption
- S3 Websites: host a static website on Amazon S3
- S3 Versioning: multiple versions for files, prevent accidental deletes
- S3 Replication: same-region or cross-region, must enable versioning
- S3 Storage Classes: Standard, IA, 1Z-IA, Intelligent, Glacier (Instant, Flexible, Deep)

# 3. IAM

## IAM: Users & Groups



- IAM = Identity and Access Management, Global service
- Root account created by default, shouldn't be used or shared
- Users are people within your organization, and can be grouped
- Groups only contain users, not other groups
- Users don't have to belong to a group, and user can belong to multiple groups



# IAM Policies Structure

- Consists of
  - Version: policy language version, always include "2012-10-17"
  - Id: an identifier for the policy (optional)
  - Statement: one or more individual statements (required)
- Statements consists of
  - Sid: an identifier for the statement (optional)
  - Effect: whether the statement allows or denies access (Allow, Deny)
  - Principal: account/user/role to which this policy applied to
  - Action: list of actions this policy allows or denies
  - Resource: list of resources to which the actions applied to
  - Condition: conditions for when this policy is in effect (optional)




# IAM – Password Policy

- Strong passwords = higher security for your account
- In AWS, you can setup a password policy:
  - Set a minimum password length
  - Require specific character types:
    - including uppercase letters
    - lowercase letters
    - numbers
    - non-alphanumeric characters
  - Allow all IAM users to change their own passwords
  - Require users to change their password after some time (password expiration)
  - Prevent password re-use

# Multi Factor Authentication - MFA




- Users have access to your account and can possibly change configurations or delete resources in your AWS account
- You want to protect your Root Accounts and IAM users
- MFA = password *you know* + security device *you own*



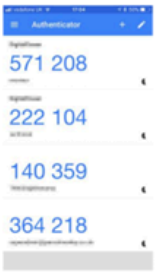
Alice

Password + MFA => Successful login

- Main benefit of MFA:
  if a password is stolen or hacked, the account is not compromised

# MFA devices options in AWS

**Virtual MFA device**




**Google Authenticator**
(phone only)




**Authy**
(multi-device)

Support for multiple tokens on a single device.

**Universal 2nd Factor (U2F) Security Key**



**YubiKey** by Yubico (3rd party)

Support for multiple root and IAM users using a single security key

# How can users access AWS ?



- To access AWS, you have three options:
  - AWS Management Console (protected by password + MFA)
  - AWS Command Line Interface (CLI): protected by access keys
  - AWS Software Developer Kit (SDK) - for code: protected by access keys
- Access Keys are generated through the AWS Console
- Users manage their own access keys
- Access Keys are secret, just like a password. Don't share them
- Access Key ID ~= username
- Secret Access Key ~= password

# What's the AWS CLI?

- A tool that enables you to interact with AWS services using commands in your command-line shell
- Direct access to the public APIs of AWS services
- You can develop scripts to manage your resources
- It's open-source https://github.com/aws/aws-cli
- Alternative to using AWS Management Console

```
➜  ~ aws s3 cp myfile.txt s3://ccp-mybucket/myfile.txt
upload: ./myfile.txt to s3://ccp-mybucket/myfile.txt
➜  ~ aws s3 ls s3://ccp-mybucket
2021-05-14 03:22:52          0 myfile.txt
➜  ~
```

# IAM Guidelines & Best Practices



- Don’t use the root account except for AWS account setup
- One physical user = One AWS user
- Assign users to groups and assign permissions to groups
- Create a strong password policy
- Use and enforce the use of Multi Factor Authentication (MFA)
- Create and use Roles for giving permissions to AWS services
- Use Access Keys for Programmatic Access (CLI / SDK)
- Audit permissions of your account using IAM Credentials Report & IAM Access Advisor

# IAM Section - Summary



- Users: mapped to a physical user, has a password for AWS Console
- Groups: contains users only
- Policies: JSON document that outlines permissions for users or groups
- Roles: for EC2 instances or AWS services
- Security: MFA + Password Policy
- AWS CLI: manage your AWS services using the command-line
- AWS SDK: manage your AWS services using a programming language
- Access Keys: access AWS using the CLI or SDK

# 4. EC2

## Amazon EC2

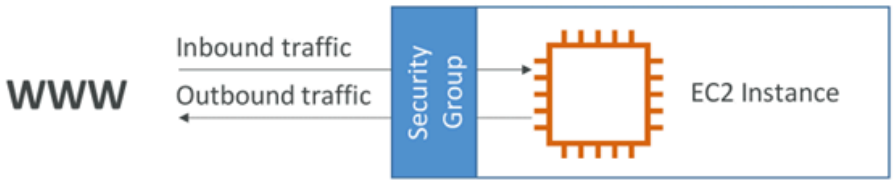


- EC2 is one of the most popular of AWS' offering
- EC2 = Elastic Compute Cloud = Infrastructure as a Service
- It mainly consists in the capability of :
  - Renting virtual machines (EC2)
  - Storing data on virtual drives (EBS)
  - Distributing load across machines (ELB)
  - Scaling the services using an auto-scaling group (ASG)

- Knowing EC2 is fundamental to understand how the Cloud works

## Introduction to Security Groups

- Security Groups are the fundamental of network security in AWS
- They control how traffic is allowed into or out of our EC2 Instances.




- Security groups only contain allow rules
- Security groups rules can reference by IP or by security group

## Classic Ports to know

- 22 = SSH (Secure Shell) - log into a Linux instance
- 21 = FTP (File Transfer Protocol) - upload files into a file share
- 22 = SFTP (Secure File Transfer Protocol) - upload files using SSH
- 80 = HTTP - access unsecured websites
- 443 = HTTPS - access secured websites
- 3389 = RDP (Remote Desktop Protocol) - log into a Windows instance

## EC2 Instance Connect

- Connect to your EC2 instance within your browser
- No need to use your key file that was downloaded
- The “magic” is that a temporary key is uploaded onto EC2 by AWS

- Works only out-of-the-box with Amazon Linux 2

- Need to make sure the port 22 is still opened!

# Gib vs GB

one GB is defined as $1000^3$ (1,000,000,000) bytes and one GiB as $1024^3$ (1,073,741,824) bytes. That means one GB equals 0.93 GiB
GB=1000mb(gigabytes)
GIB= 1024(gibibytes)

# EC2 Instances Purchasing Options

- On-Demand Instances - short workload, predictable pricing, pay by second
- Reserved (1 & 3 years)
  - Reserved Instances - long workloads
  - Convertible Reserved Instances - long workloads with flexible instances
- Savings Plans (1 & 3 years) –commitment to an amount of usage, long workload
- Spot Instances - short workloads, cheap, can lose instances (less reliable)
- Dedicated Hosts - book an entire physical server, control instance placement
- Dedicated Instances - no other customers will share your hardware
- Capacity Reservations - reserve capacity in a specific AZ for any duration

# Which purchasing option is right for me?



- On demand: coming and staying in resort whenever we like, we pay the full price
- Reserved: like planning ahead and if we plan to stay for a long time, we may get a good discount.
- Savings Plans: pay a certain amount per hour for certain period and stay in any room type (e.g., King, Suite, Sea View, ...)
- Spot instances: the hotel allows people to bid for the empty rooms and the highest bidder keeps the rooms.You can get kicked out at any time
- Dedicated Hosts: We book an entire building of the resort
- Capacity Reservations: you book a room for a period with full price even you don't stay in it

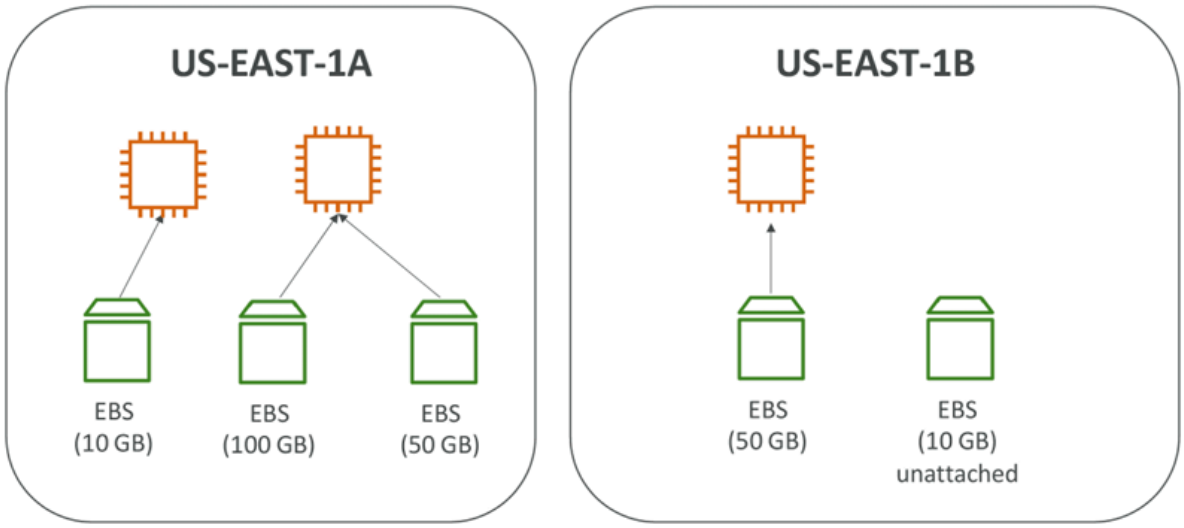
# What's an EBS Volume?



- An EBS (Elastic Block Store) Volume is a network drive you can attach to your instances while they run
- It allows your instances to persist data, even after their termination
- They can only be mounted to one instance at a time (at the CCP level)
- They are bound to a specific availability zone

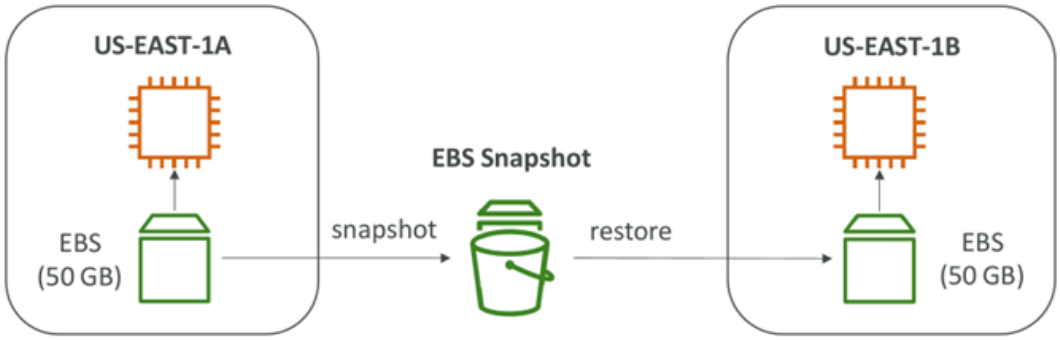
- Analogy:Think of them as a "network USB stick"
- Free tier: 30 GB of free EBS storage of type General Purpose (SSD) or Magnetic per month

# EBS Volume - Example



# EBS Snapshots

- Make a backup (snapshot) of your EBS volume at a point in time
- Not necessary to detach volume to do snapshot, but recommended
- Can copy snapshots across AZ or Region




# AMI Overview



- AMI = Amazon Machine Image
- AMI are a customization of an EC2 instance
  - You add your own software, configuration, operating system, monitoring...
  - Faster boot / configuration time because all your software is pre-packaged
- AMI are built for a specific region (and can be copied across regions)
- You can launch EC2 instances from:
  - A Public AMI: AWS provided
  - Your own AMI: you make and maintain them yourself
  - An AWS Marketplace AMI: an AMI someone else made (and potentially sells)

## EC2 Section - Summary



- EC2 Instance: AMI (OS) + Instance Size (CPU + RAM) + Storage + security groups + EC2 User Data
- Security Groups: Firewall attached to the EC2 instance
- EC2 User Data: Script launched at the first start of an instance
- SSH: start a terminal into our EC2 Instances (port 22)
- EC2 Instance Role: link to IAM roles
- Purchasing Options: On-Demand, Spot, Reserved (Standard + Convertible + Scheduled), Dedicated Host, Dedicated Instance

# 5. VPC

Before vpc we will see basic of networking

## IP Addresses in AWS

- IPv4 - Internet Protocol version 4 (4.3 Billion Addresses)
  - Public IPv4 - can be used on the Internet
  - EC2 instance gets a new a public IP address every time you stop then start it (default)
  - Private IPv4 - can be used on private networks (LAN) such as internal AWS networking (e.g., 192.168.1.1)
  - Private IPv4 is fixed for EC2 Instances even if you start/stop them
- Elastic IP - allows you to attach a fixed public IPv4 address to EC2 instance
  - Note: has ongoing cost if not attached to EC2 instance or if the EC2 instance is stopped
- IPv6 - Internet Protocol version 6 ($3.4 \times 10^{38}$ Addresses)
  - Every IP address is public (no private range)
  - Example: 2001:db8:3333:4444:cccc:dddd:eeee:ffff

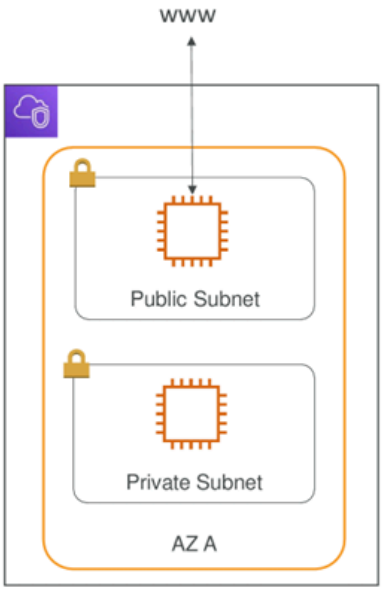
DHCP(dynamic host configuration protol)
Automatically assing ip address when they connect to internet

## Classes of public ip

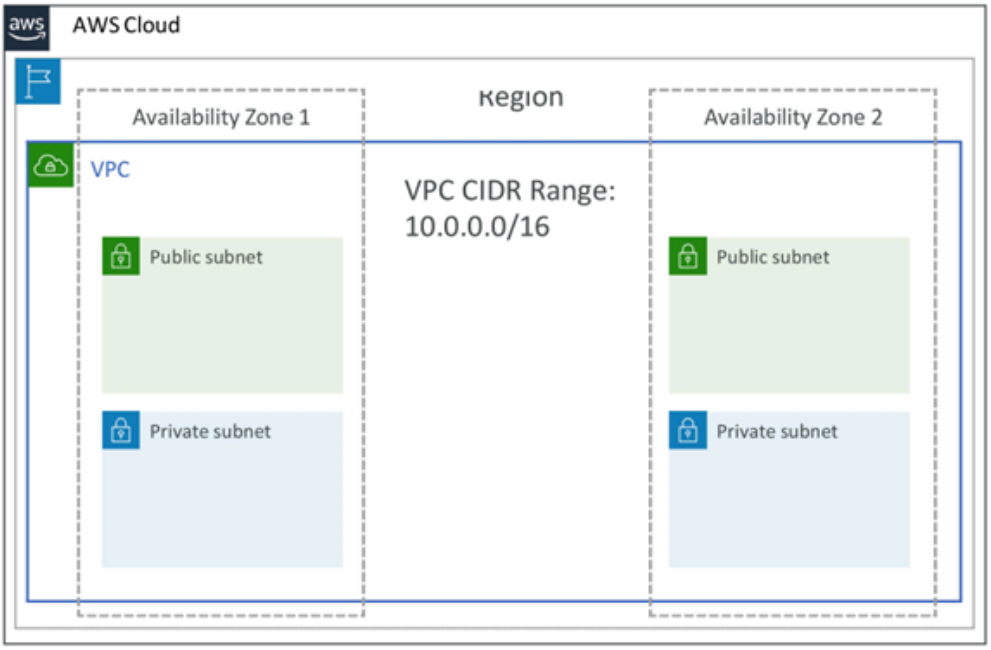
| Address Class | RANGE | Default Subnet Mask |
|---|---|---|
| A | 1.0.0.0 to 126.255.255.255 | 255.0.0.0 |
| B | 128.0.0.0 to 191.255.255.255 | 255.255.0.0 |
| C | 192.0.0.0 to 223.255.255.255 | 255.255.255.0 |
| D | 224.0.0.0 to 239.255.255.255 | Reserved for Multicasting |
| E | 240.0.0.0 to 254.255.255.255 | Experimental |
| Note: Class A addresses 127.0.0.0 to 127.255.255.255 cannot be used and is reserved for loopback testing. | | |

# VPC & Subnets Primer

- VPC - Virtual Private Cloud: private network to deploy your resources (regional resource)
- Subnets allow you to partition your network inside your VPC (Availability Zone resource)
- A public subnet is a subnet that is accessible from the internet
- A private subnet is a subnet that is not accessible from the internet
- To define access to the internet and between subnets, we use Route Tables.



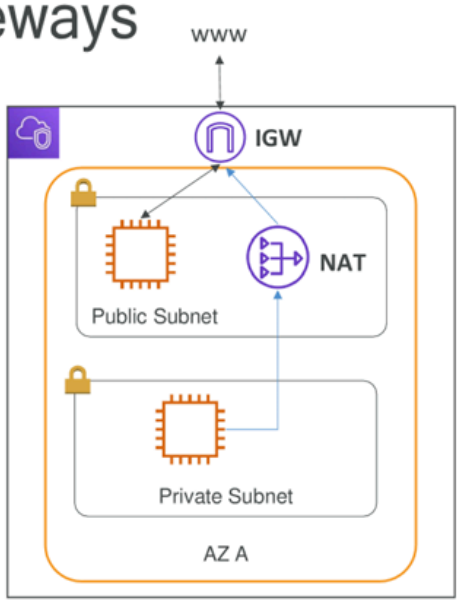
# VPC Diagram




# Internet Gateway & NAT Gateways

- Internet Gateways helps our VPC instances connect with the internet
- Public Subnets have a route to the internet gateway.

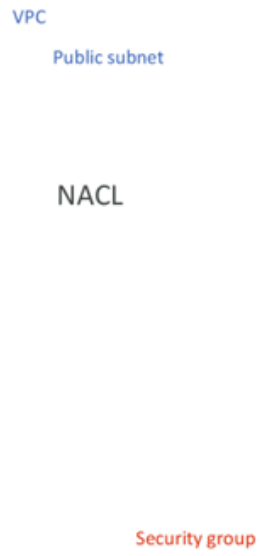
- NAT Gateways (AWS-managed) & NAT Instances (self-managed) allow your instances in your Private Subnets to access the internet while remaining private



# Network ACL & Security Groups

- NACL (Network ACL)
  - A firewall which controls traffic from and to subnet
  - Can have ALLOW and DENY rules
  - Are attached at the Subnet level
  - Rules only include IP addresses

- Security Groups
  - A firewall that controls traffic to and from an ENI / an EC2 Instance
  - Can have only ALLOW rules
  - Rules include IP addresses and other security groups

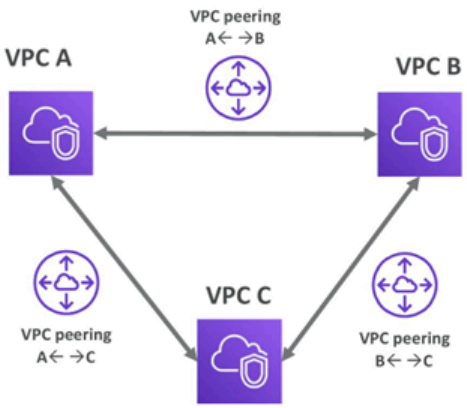



# Network ACLs vs Security Groups

| Security Group | Network ACL |
|---|---|
| Operates at the instance level | Operates at the subnet level |
| Supports allow rules only | Supports allow rules and deny rules |
| Is stateful: Return traffic is automatically allowed, regardless of any rules | Is stateless: Return traffic must be explicitly allowed by rules |
| We evaluate all rules before deciding whether to allow traffic | We process rules in number order when deciding whether to allow traffic |
| Applies to an instance only if someone specifies the security group when launching the instance, or associates the security group with the instance later on | Automatically applies to all instances in the subnets it's associated with (therefore, you don't have to rely on users to specify the security group) |

# VPC Peering

- Connect two VPC, privately using AWS' network
- Make them behave as if they were in the same network
- Must not have overlapping CIDR (IP address range)
- VPC Peering connection is not transitive (must be established for each VPC that need to communicate with one another)

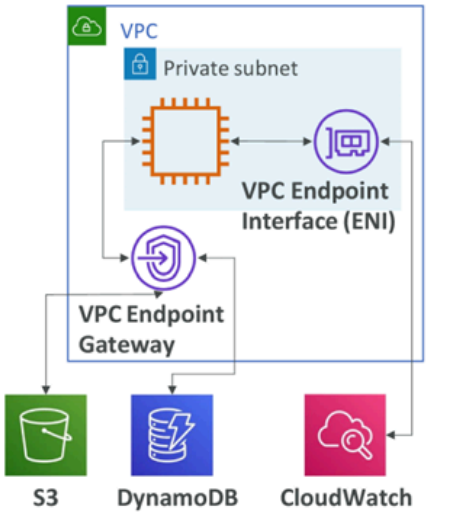



# VPC Endpoints



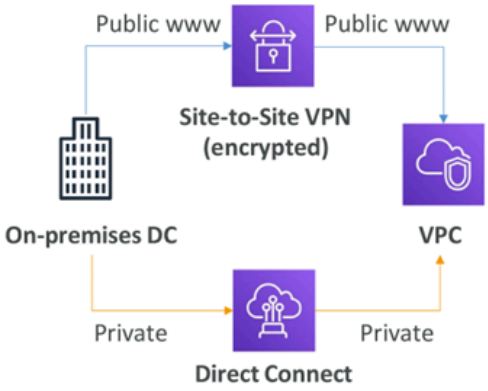
- Endpoints allow you to connect to AWS Services using a private network instead of the public www network
- This gives you enhanced security and lower latency to access AWS services

- VPC Endpoint Gateway: S3 & DynamoDB
- VPC Endpoint Interface: the rest



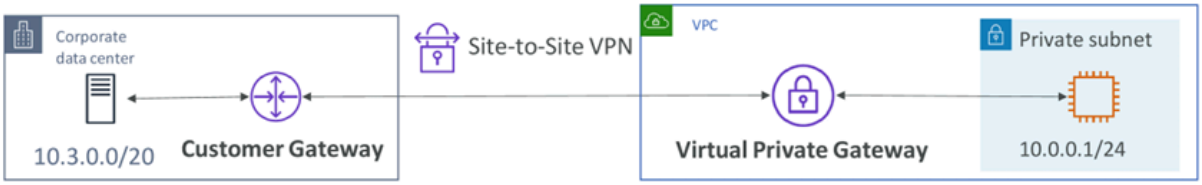
# Site to Site VPN & Direct Connect

- Site to Site VPN
  - Connect an on-premises VPN to AWS
  - The connection is automatically encrypted
  - Goes over the public internet
- Direct Connect (DX)
  - Establish a physical connection between on-premises and AWS
  - The connection is private, secure and fast
  - Goes over a private network
  - Takes at least a month to establish




# Site-to-Site VPN

- On-premises: must use a Customer Gateway (CGW)
- AWS: must use a Virtual Private Gateway (VGW)



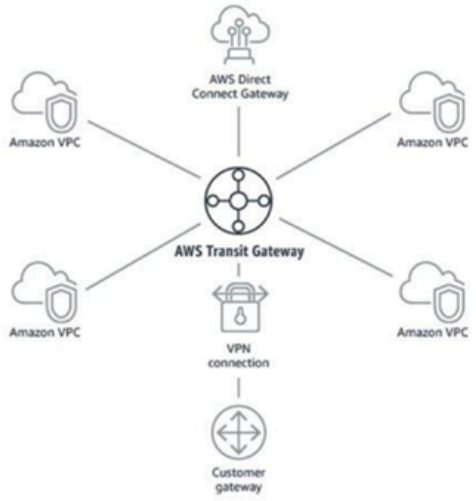
# AWS Client VPN

- Connect from your computer using OpenVPN to your private network in AWS and on-premises
- Allow you to connect to your EC2 instances over a private IP (just as if you were in the private VPC network)
- Goes over public Internet




# Transit Gateway

- For having transitive peering between thousands of VPC and on-premises, hub-and-spoke (star) connection
- One single Gateway to provide this functionality
- Works with Direct Connect Gateway, VPN connections



## VPC Closing Comments

- VPC - Virtual Private Cloud
- Subnets - Tied to an AZ, network partition of the VPC
- Internet Gateway - at the VPC level, provide Internet Access
- NAT Gateway / Instances - give internet access to private subnets
- NACL - Stateless, subnet rules for inbound and outbound
- Security Groups - Stateful, operate at the EC2 instance level or ENI
- VPC Peering - Connect two VPC with non overlapping IP ranges, nontransitive
- Elastic IP –fixed public IPv4, ongoing cost if not in-use

# 6. ELB, ASG and CloudWatch

## Scalability & High Availability

- Scalability means that an application / system can handle greater loads by adapting.
- There are two kinds of scalability:
  - Vertical Scalability
  - Horizontal Scalability (= elasticity)
- Scalability is linked but different to High Availability

- Let's deep dive into the distinction, using a call center as an example

# High Availability & Scalability For EC2

- Vertical Scaling: Increase instance size (= scale up / down)
  - From: t2.nano - 0.5G of RAM, 1 vCPU
  - To: u-12tb1.metal - 12.3 TB of RAM, 448 vCPUs

- Horizontal Scaling: Increase number of instances (= scale out / in)
  - Auto Scaling Group
  - Load Balancer

- High Availability: Run instances for the same application across multi AZ
  - Auto Scaling Group multi AZ
  - Load Balancer multi AZ

# Scalability vs Elasticity (vs Agility)

- Scalability: ability to accommodate a larger load by making the hardware stronger (scale up), or by adding nodes (scale out)

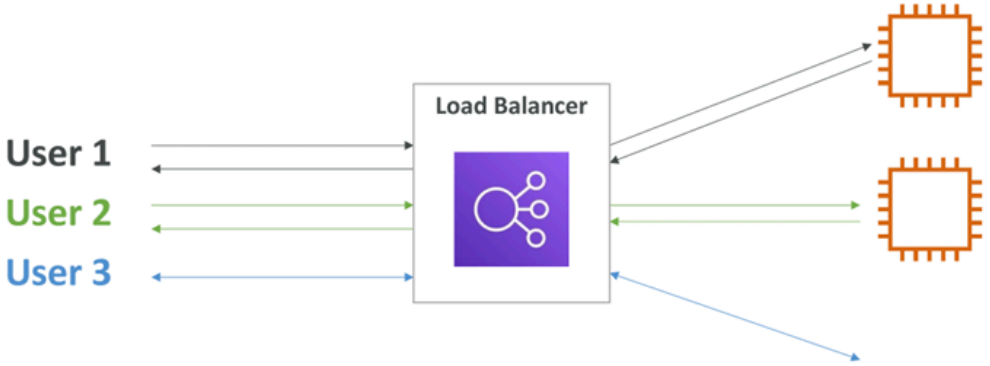
- Elasticity: once a system is scalable, elasticity means that there will be some “auto-scaling” so that the system can scale based on the load.This is “cloud-friendly”: pay-per-use, match demand, optimize costs

- Agility: (not related to scalability - distractor) new IT resources are only a click away, which means that you reduce the time to make those resources available to your developers from weeks to just minutes.

# What is load balancing?



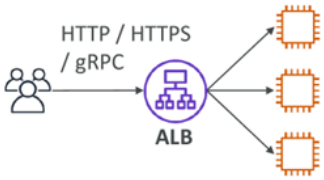
- Load balancers are servers that forward internet traffic to multiple servers (EC2 Instances) downstream.




## Application Load Balancer



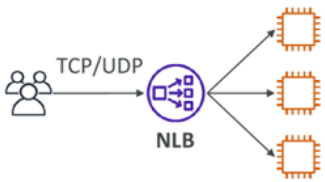
- HTTP / HTTPS / gRPC protocols (Layer 7)
- HTTP Routing features
- Static DNS (URL)




## Network Load Balancer



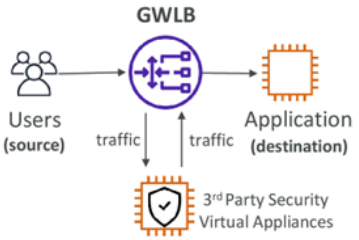
- TCP / UDP protocols (Layer 4)
- High Performance: millions of request per seconds
- Static IP through Elastic IP




## Gateway Load Balancer



- GENEVE Protocol on IP Packets (Layer 3)
- Route Traffic to Firewalls that you manage on EC2 Instances
- Intrusion detection

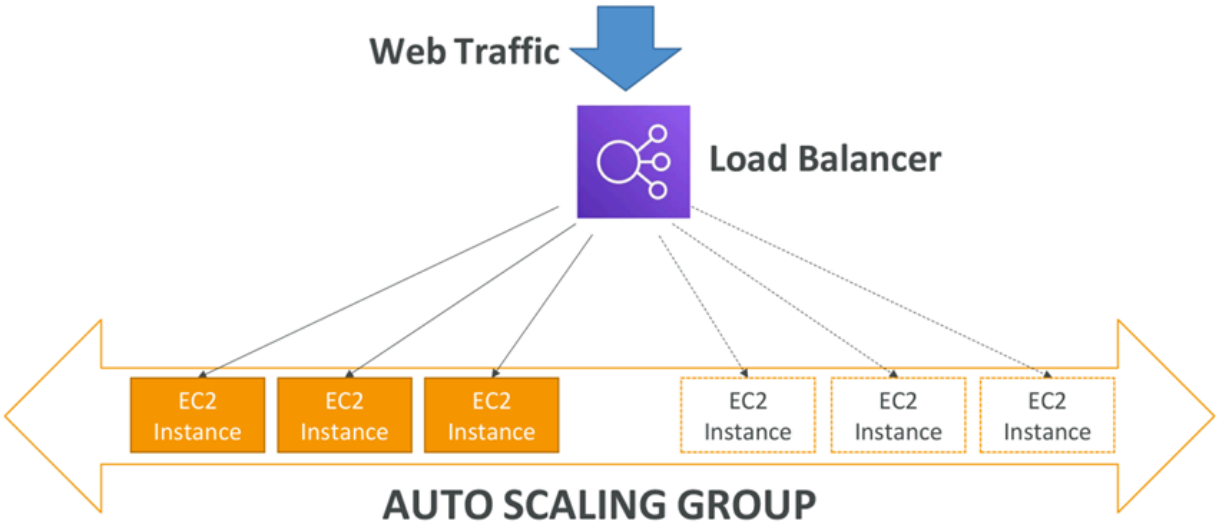


# What's an Auto Scaling Group?



- In real-life, the load on your websites and application can change
- In the cloud, you can create and get rid of servers very quickly
- The goal of an Auto Scaling Group (ASG) is to:
  - Scale out (add EC2 instances) to match an increased load
  - Scale in (remove EC2 instances) to match a decreased load
  - Ensure we have a minimum and a maximum number of machines running
  - Automatically register new instances to a load balancer
  - Replace unhealthy instances
- Cost Savings: only run at an optimal capacity (principle of the cloud)

## Auto Scaling Group in AWS With Load Balancer



# Amazon CloudWatch Metrics



- CloudWatch provides metrics for *every* services in AWS
- Metric is a variable to monitor (CPUUtilization, NetworkIn...)
- Metrics have timestamps
- Can create CloudWatch dashboards of metrics

# Important Metrics

- EC2 instances: CPU Utilization, Status Checks, Network (not RAM)
  - Default metrics every 5 minutes
  - Option for Detailed Monitoring ($$$): metrics every 1 minute
- EBS volumes: Disk Read/Writes
- S3 buckets: BucketSizeBytes, NumberOfObjects, AllRequests
- Billing: Total Estimated Charge (only in us-east-1)
- Service Limits: how much you've been using a service API
- Custom metrics: push your own metrics

# Amazon CloudWatch Alarms



- Alarms are used to trigger notifications for any metric
- Alarms actions...
  - Auto Scaling: increase or decrease EC2 instances "desired" count
  - EC2 Actions: stop, terminate, reboot or recover an EC2 instance
  - SNS notifications: send a notification into an SNS topic
- Various options (sampling, %, max, min, etc...)
- Can choose the period on which to evaluate an alarm
- Example: create a billing alarm on the CloudWatch Billing metric
- Alarm States: OK. INSUFFICIENT_DATA, ALARM

# ELB & ASG - Summary

- High Availability vs Scalability (vertical and horizontal) vs Elasticity vs Agility in the Cloud
- Elastic Load Balancers (ELB)
  - Distribute traffic across backend EC2 instances, can be Multi-AZ
  - Supports health checks
  - 4 types: Classic (old), Application (HTTP - L7), Network (TCP - L4), Gateway (L3)
- Auto Scaling Groups (ASG)
  - Implement Elasticity for your application, across multiple AZ
  - Scale EC2 instances based on the demand on your system, replace unhealthy
  - Integrated with the ELB

- Cloud watch :help to monitor aws services

# 7. Database

## Databases Intro



- Storing data on disk (EFS, EBS, EC2 Instance Store, S3) can have its limits
- Sometimes, you want to store data in a database...
- You can structure the data
- You build indexes to efficiently query / search through the data
- You define relationships between your datasets

- Databases are optimized for a purpose and come with different features, shapes and constraints

## Relational Databases

- Looks just like Excel spreadsheets, with links between them!
- Can use the SQL language to perform queries / lookups

Students

| Student ID | Dept ID | Name | Email |
|---|---|---|---|
| 1 | M01 | Joe Miller | joe@abc.com |
| 2 | B01 | Sarah T | sarah@abc.com |

Subjects

| Student ID | Subject |
|---|---|
| 1 | Physics |
| 1 | Chemistry |
| 1 | Math |
| 2 | History |
| 2 | Geography |
| 2 | Economics |

Departments

| Dept ID | SPOC | Email | Phone |
|---|---|---|---|
| M01 | Kelly Jones | kelly@abc.com | +1234567890 |
| B01 | Satish Kumar | satish@abc.com | +1234567891 |

# NoSQL Databases

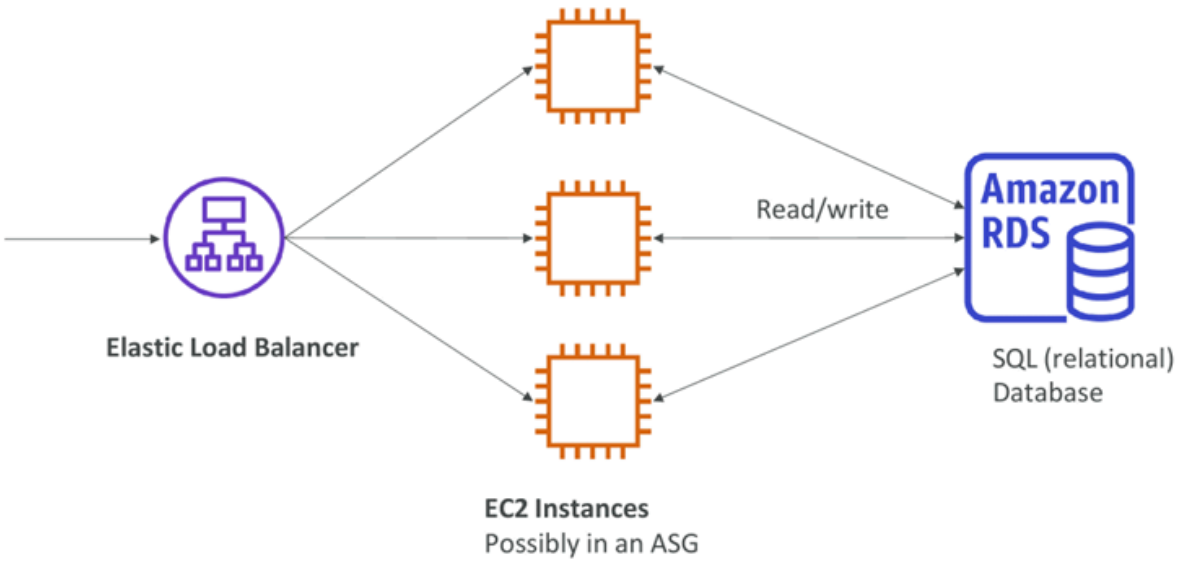
- NoSQL = non-SQL = non relational databases
- NoSQL databases are purpose built for specific data models and have flexible schemas for building modern applications.
- Benefits:
  - Flexibility: easy to evolve data model
  - Scalability: designed to scale-out by using distributed clusters
  - High-performance: optimized for a specific data model
  - Highly functional: types optimized for the data model
- Examples: Key-value, document, graph, in-memory, search databases

# AWS RDS Overview



- RDS stands for Relational Database Service
- It's a managed DB service for DB use SQL as a query language.
- It allows you to create databases in the cloud that are managed by AWS
  - Postgres
  - MySQL
  - MariaDB
  - Oracle
  - Microsoft SQL Server
  - Aurora (AWS Proprietary database)

# RDS Solution Architecture

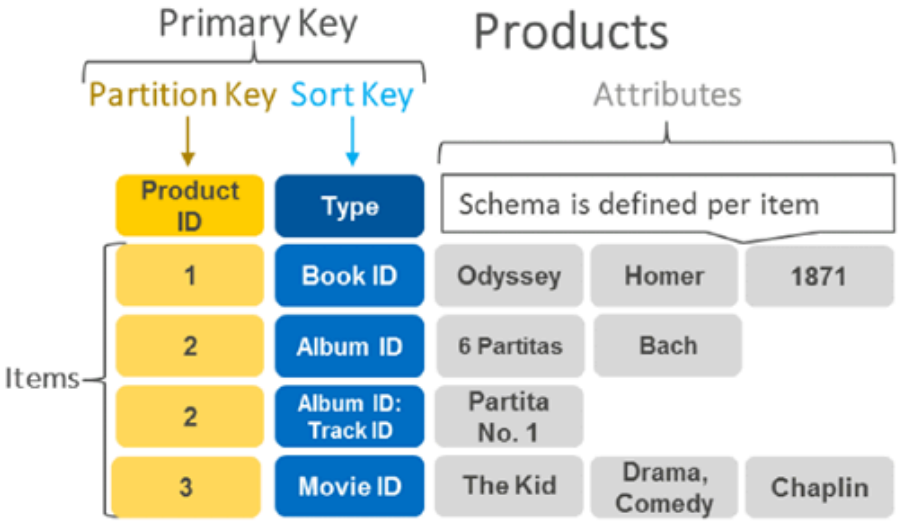



# DynamoDB

- Fully Managed Highly available with replication across 3 AZ
- NoSQL database - not a relational database
- Scales to massive workloads, distributed “serverless” database
- Millions of requests per seconds, trillions of row, 100s of TB of storage
- Fast and consistent in performance
- Single-digit millisecond latency - low latency retrieval
- Integrated with IAM for security, authorization and administration
- Low cost and auto scaling capabilities
- Standard & Infrequent Access (IA) Table Class

# DynamoDB - type of data

- DynamoDB is a key/value database



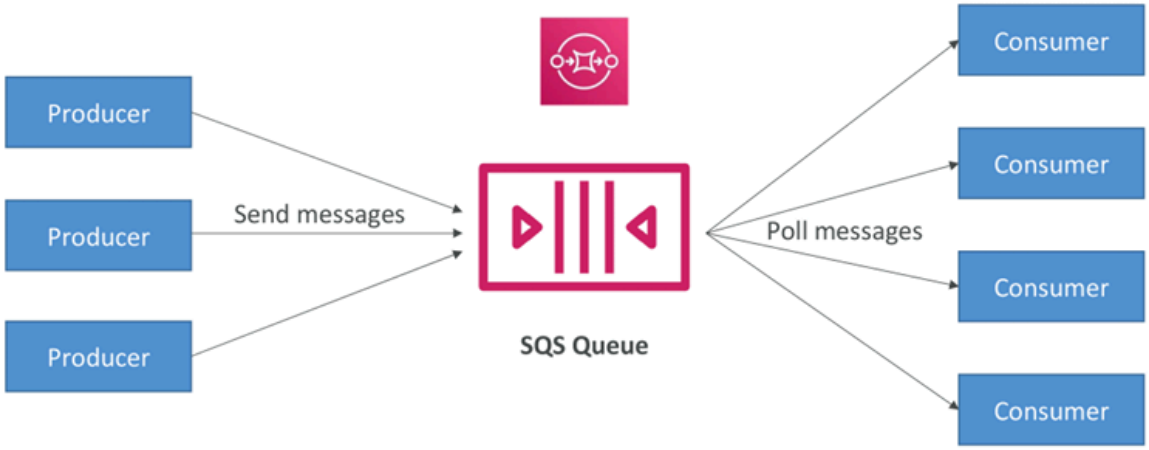

# Redshift Overview



- Redshift is based on PostgreSQL, but it's not used for OLTP
- It's OLAP - online analytical processing (analytics and data warehousing)
- Load data once every hour, not every second
- 10x better performance than other data warehouses, scale to PBs of data
- Columnar storage of data (instead of row based)
- Massively Parallel Query Execution (MPP), highly available
- Pay as you go based on the instances provisioned
- Has a SQL interface for performing the queries
- BI tools such as AWS Quicksight or Tableau integrate with it

# 8. SQS, SWF, SNS

## Amazon SQS - Simple Queue Service
## What's a queue?

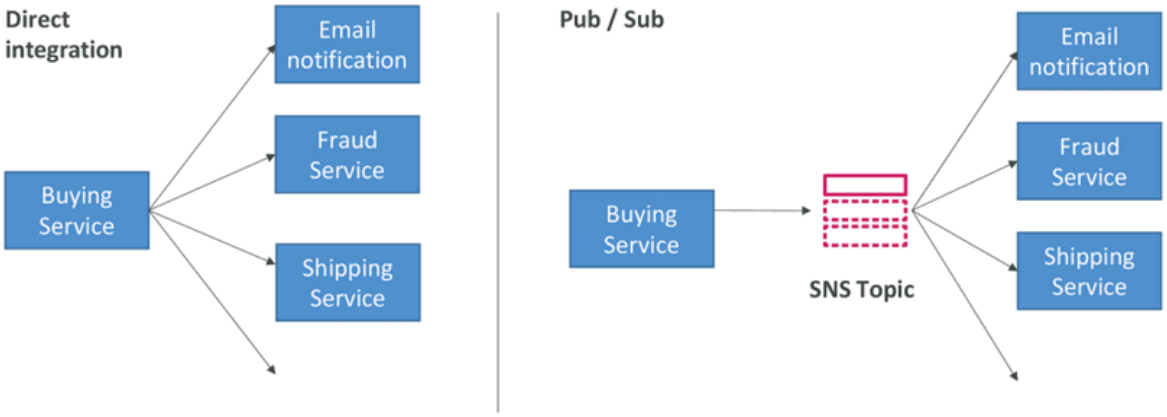



## Amazon SQS - Standard Queue

- Oldest AWS offering (over 10 years old)
- Fully managed service (~serverless), use to decouple applications
- Scales from 1 message per second to 10,000s per second
- Default retention of messages: 4 days, maximum of 14 days
- No limit to how many messages can be in the queue
- Messages are deleted after they're read by consumers
- Low latency (<10 ms on publish and receive)
- Consumers share the work to read messages & scale horizontally

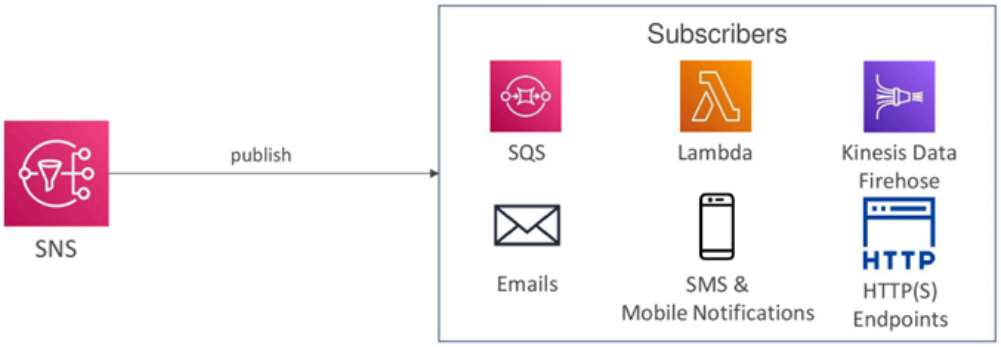
# Amazon SNS

- What if you want to send one message to many receivers?

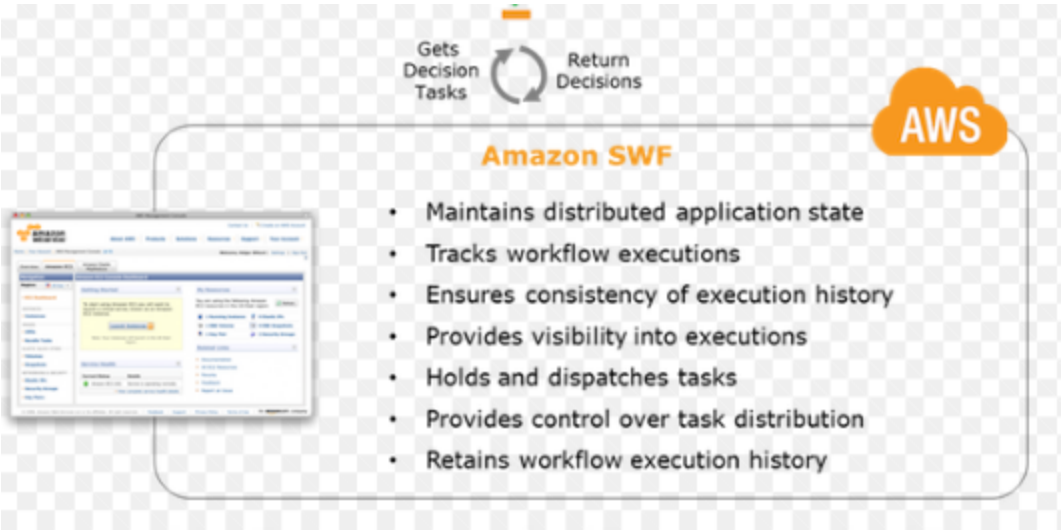



# Amazon SNS

- The "event publishers" only sends message to one SNS topic
- As many "event subscribers" as we want to listen to the SNS topic notifications
- Each subscriber to the topic will get all the messages
- Up to 12,500,000 subscriptions per topic, 100,000 topics limit




# Amazon SWF




# 9. DNS and Route 53

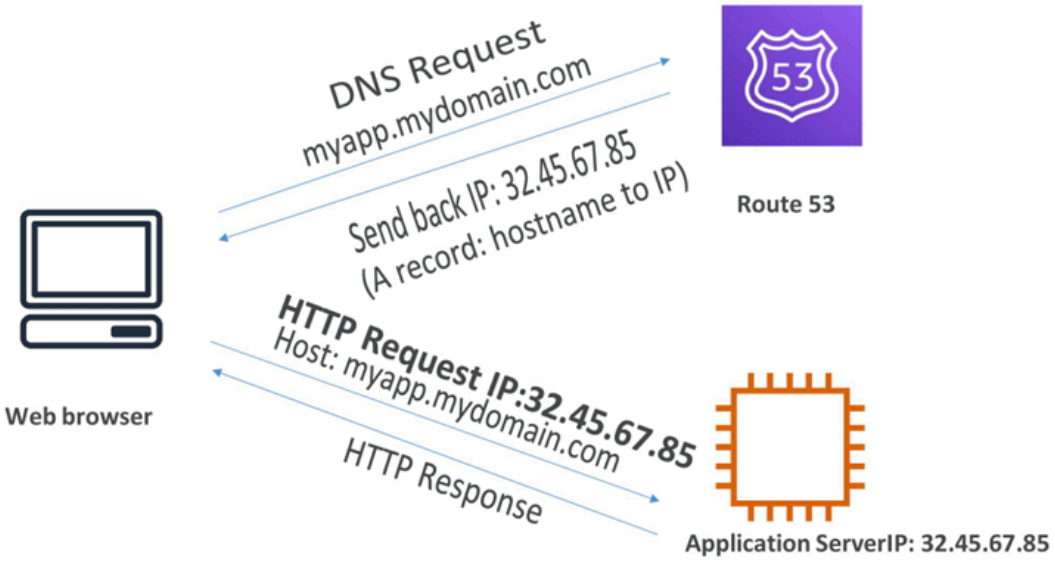
## Amazon Route 53 Overview




- Route53 is a Managed DNS (Domain Name System)
- DNS is a collection of rules and records which helps clients understand how to reach a server through URLs.

- In AWS, the most common records are:
  - www.google.com => 12.34.56.78 == A record (IPv4)
  - www.google.com => 2001:0db8:85a3:0000:0000:8a2e:0370:7334 == AAAA IPv6
  - search.google.com => www.google.com == CNAME: hostname to hostname
  - example.com => AWS resource == Alias (ex: ELB, CloudFront, S3, RDS, etc...)

## Route 53 - Diagram for A Record




# 10. Elastic cache

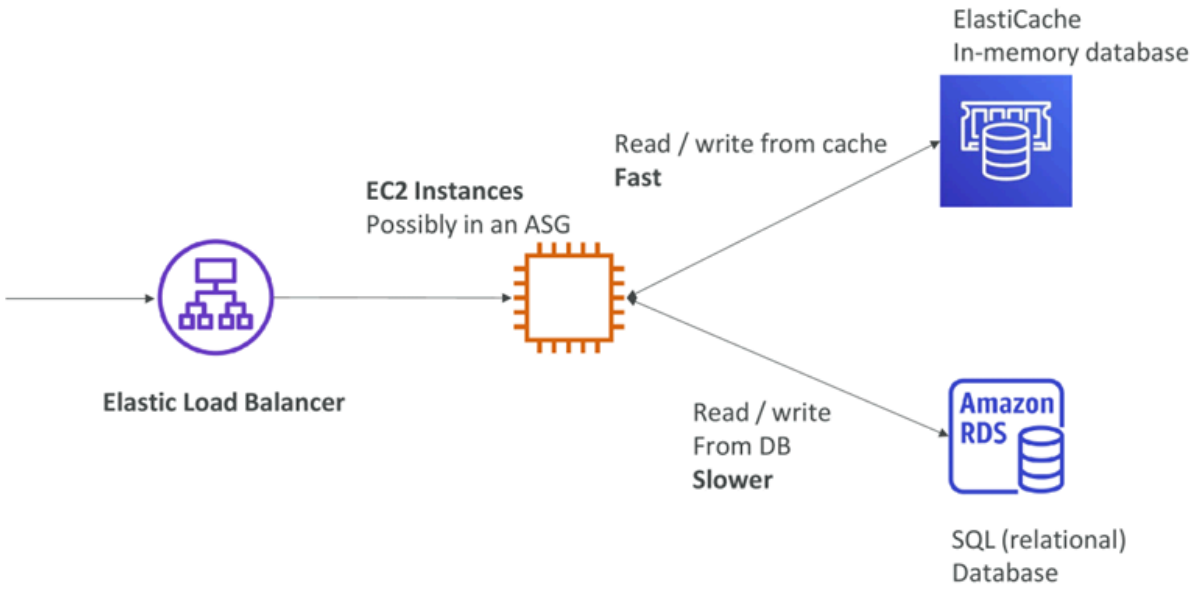
## Amazon ElastiCache Overview



- The same way RDS is to get managed Relational Databases...
- ElastiCache is to get managed Redis or Memcached
- Caches are in-memory databases with high performance, low latency
- Helps reduce load off databases for read intensive workloads

- AWS takes care of OS maintenance / patching, optimizations, setup, configuration, monitoring, failure recovery and backups

## ElastiCache
## Solution Architecture - Cache




# 11. Key Services(server less)

## Why AWS Lambda

Amazon EC2

- Virtual Servers in the Cloud
- Limited by RAM and CPU
- Continuously running
- Scaling means intervention to add / remove servers

Amazon Lambda

- Virtual functions - no servers to manage!
- Limited by time - short executions
- Run on-demand
- Scaling is automated!

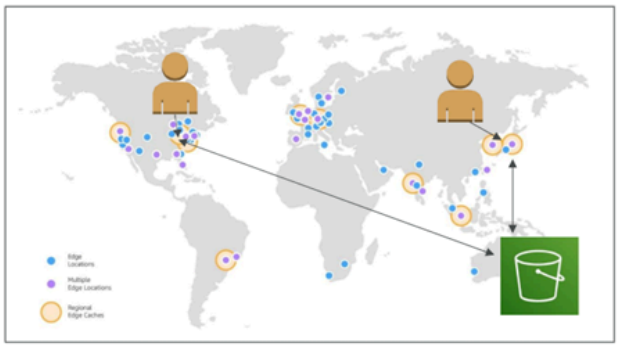
# Benefits of AWS Lambda

- Easy Pricing:
  - Pay per request and compute time
  - Free tier of 1,000,000 AWS Lambda requests and 400,000 GBs of compute time
- Integrated with the whole AWS suite of services
- Event-Driven: functions get invoked by AWS when needed
- Integrated with many programming languages
- Easy monitoring through AWS CloudWatch
- Easy to get more resources per functions (up to 10GB of RAM!)
- Increasing RAM will also improve CPU and network!

# Amazon CloudFront



- Content Delivery Network (CDN)
- Improves read performance, content is cached at the edge
- Improves users experience
- 216 Point of Presence globally (edge locations)
- DDoS protection (because worldwide), integration with Shield, AWS Web Application Firewall



Source: https://aws.amazon.com/cloudfront/features/?nc=sn&loc=2

# CloudFront – Origins

- S3 bucket
  - For distributing files and caching them at the edge
  - Enhanced security with CloudFront Origin Access Control (OAC)
  - OAC is replacing Origin Access Identity (OAI)
  - CloudFront can be used as an ingress (to upload files to S3)
- Custom Origin (HTTP)
  - Application Load Balancer
  - EC2 instance
  - S3 website (must first enable the bucket as a static S3 website)
  - Any HTTP backend you want

# Redshift Overview



- Redshift is based on PostgreSQL, but it's not used for OLTP
- It's OLAP - online analytical processing (analytics and data warehousing)
- Load data once every hour, not every second
- 10x better performance than other data warehouses, scale to PBs of data
- Columnar storage of data (instead of row based)
- Massively Parallel Query Execution (MPP), highly available
- Pay as you go based on the instances provisioned
- Has a SQL interface for performing the queries
- BI tools such as AWS Quicksight or Tableau integrate with it

# Amazon Kinesis

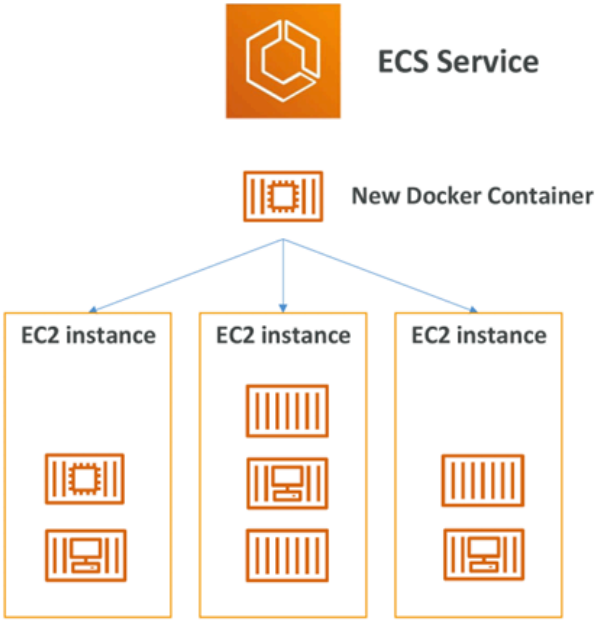


- For the exam: Kinesis = real-time big data streaming
- Managed service to collect, process, and analyze real-time streaming data at any scale

- Too detailed for the Cloud Practitioner exam but good to know:
  - Kinesis Data Streams: low latency streaming to ingest data at scale from hundreds of thousands of sources
  - Kinesis Data Firehose: load streams into S3, Redshift, ElasticSearch, etc...
  - Kinesis Data Analytics: perform real-time analytics on streams using SQL
  - Kinesis Video Streams: monitor real-time video streams for analytics or ML

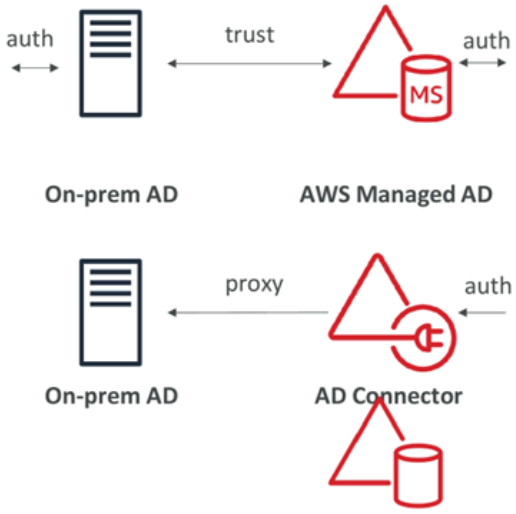
# ECS

- ECS = Elastic Container Service
- Launch Docker containers on AWS
- You must provision & maintain the infrastructure (the EC2 instances)
- AWS takes care of starting / stopping containers
- Has integrations with the Application Load Balancer



# AWS Directory Services

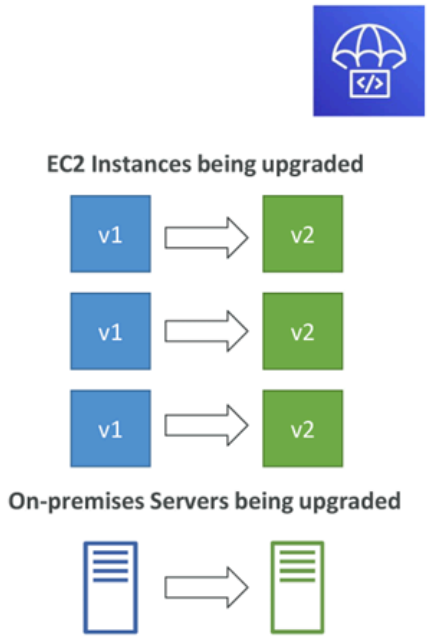
- AWS Managed Microsoft AD
  - Create your own AD in AWS, manage users locally, supports MFA
  - Establish "trust" connections with your on-premise AD
- AD Connector
  - Directory Gateway (proxy) to redirect to on-premise AD, supports MFA
  - Users are managed on the on-premise AD
- Simple AD
  - AD-compatible managed directory on AWS
  - Cannot be joined with on-premise AD




Devops(CI/CD)

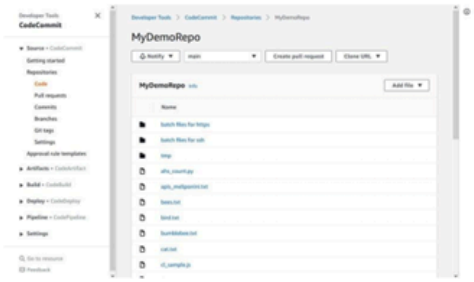
# AWS CodeDeploy

- We want to deploy our application automatically
- Works with EC2 Instances
- Works with On-Premises Servers
- Hybrid service
- Servers / Instances must be provisioned and configured ahead of time with the CodeDeploy Agent



# AWS CodeCommit

- Before pushing the application code to servers, it needs to be stored somewhere
- Developers usually store code in a repository, using the Git technology
- A famous public offering is GitHub, AWS' competing product is CodeCommit
- CodeCommit:
  - Source-control service that hosts Git-based repositories
  - Makes it easy to collaborate with others on code
  - The code changes are automatically versioned
- Benefits:
  - Fully managed
  - Scalable & highly available
  - Private, Secured, Integrated with AWS




# AWS CodeBuild

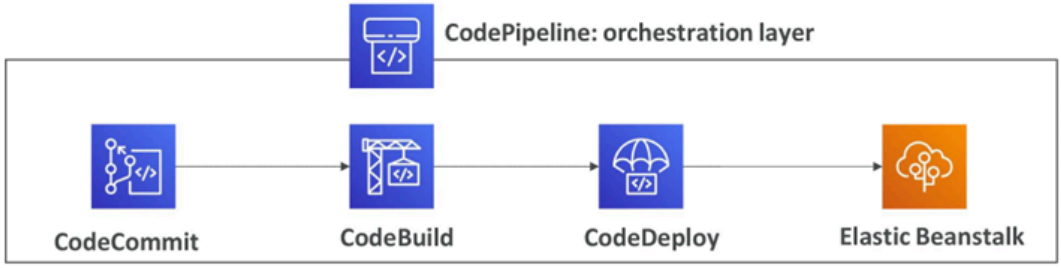
- Code building service in the cloud (name is obvious)
- Compiles source code, run tests, and produces packages that are ready to be deployed (by CodeDeploy for example)




- Benefits:
  - Fully managed, serverless
  - Continuously scalable & highly available
  - Secure
  - Pay-as-you-go pricing - only pay for the build time

## AWS CodePipeline

- Orchestrate the different steps to have the code automatically pushed to production
  - Code => Build => Test => Provision => Deploy
  - Basis for CICD (Continuous Integration & Continuous Delivery)
- Benefits:
  - Fully managed, compatible with CodeCommit, CodeBuild, CodeDeploy, Elastic Beanstalk, CloudFormation, GitHub, 3rd-party services (GitHub...) & custom plugins...
  - Fast delivery & rapid updates




# 12. Security on aws

## AWS Shared Responsibility Model

- AWS responsibility - Security of the Cloud
  - Protecting infrastructure (hardware, software, facilities, and networking) that runs all the AWS services
  - Managed services like S3, DynamoDB, RDS, etc.
- Customer responsibility - Security in the Cloud
  - For EC2 instance, customer is responsible for management of the guest OS (including security patches and updates), firewall & network configuration, IAM
  - Encrypting application data
- Shared controls:
  - Patch Management, Configuration Management, Awareness & Training

# Example, for S3



- AWS responsibility:
  - Guarantee you get unlimited storage
  - Guarantee you get encryption
  - Ensure separation of the data between different customers
  - Ensure AWS employees can't access your data
- Your responsibility:
  - Bucket configuration
  - Bucket policy / public setting
  - IAM user and roles
  - Enabling encryption

# Example, for RDS

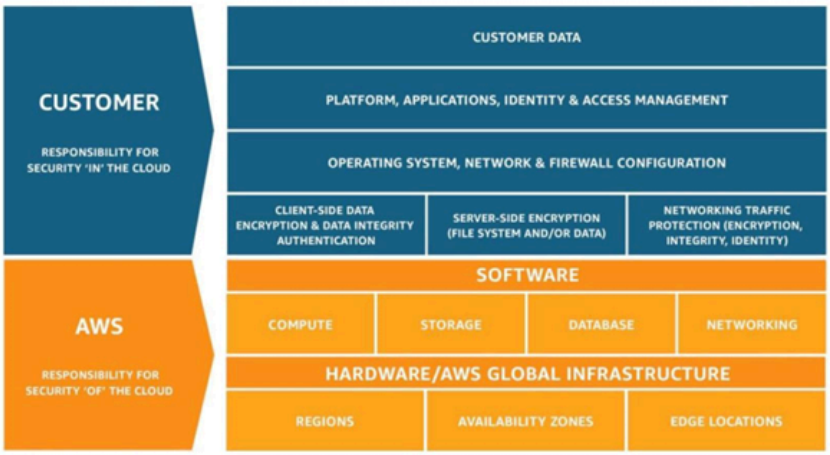


- AWS responsibility:
  - Manage the underlying EC2 instance, disable SSH access
  - Automated DB patching
  - Automated OS patching
  - Audit the underlying instance and disks & guarantee it functions
- Your responsibility:
  - Check the ports / IP / security group inbound rules in DB's SG
  - In-database user creation and permissions
  - Creating a database with or without public access
  - Ensure parameter groups or DB is configured to only allow SSL connections
  - Database encryption setting

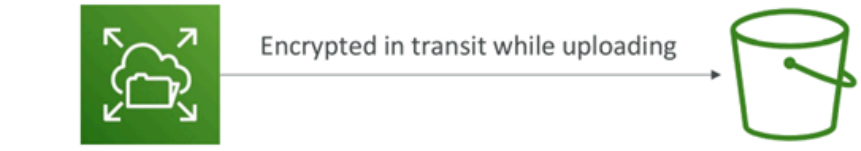
# Shared Responsibility Model diagram

CUSTOMER = RESPONSIBILITY FOR THE SECURITY **IN** THE CLOUD

AWS = RESPONSIBILITY FOR THE SECURITY **OF** THE CLOUD




# Data at rest vs. Data in transit




Encrypted at rest on EFS

Encrypted at rest on S3

- At rest: data stored or archived on a device
  - On a hard disk, on a RDS instance, in S3 Glacier Deep Archive, etc.
- In transit (in motion): data being moved from one location to another
  - Transfer from on-premises to AWS, EC2 to DynamoDB, etc.
  - Means data transferred on the network
- We want to encrypt data in both states to protect it!
- For this we leverage encryption keys

## Root user privileges




- Root user = Account Owner (created when the account is created)
- Has complete access to all AWS services and resources
- Lock away your AWS account root user access keys!
- Do not use the root account for everyday tasks, even administrative tasks
- Actions that can be performed only by the root user:
  - Change account settings (account name, email address, root user password, root user access keys)
  - View certain tax invoices
  - Close your AWS account
  - Restore IAM user permissions
  - Change or cancel your AWS Support plan
  - Register as a seller in the Reserved Instance Marketplace
  - Configure an Amazon S3 bucket to enable MFA
  - Edit or delete an Amazon S3 bucket policy that includes an invalid VPC ID or VPC endpoint ID
  - Sign up for GovCloud

# Additional services

## Amazon Rekognition



- Find objects, people, text, scenes in images and videos using ML
- Facial analysis and facial search to do user verification, people counting
- Create a database of "familiar faces" or compare against celebrities
- Use cases:
  - Labeling
  - Content Moderation
  - Text Detection
  - Face Detection and Analysis (gender, age range, emotions...)
  - Face Search and Verification
  - Celebrity Recognition
  - Pathing (ex: for sports game analysis)

# Amazon QuickSight



- Serverless machine learning-powered business intelligence service to create interactive dashboards
- Fast, automatically scalable, embeddable, with per-session pricing
- Use cases:
  - Business analytics
  - Building visualizations
  - Perform ad-hoc analysis
  - Get business insights using data
- Integrated with RDS, Aurora, Athena, Redshift, S3...

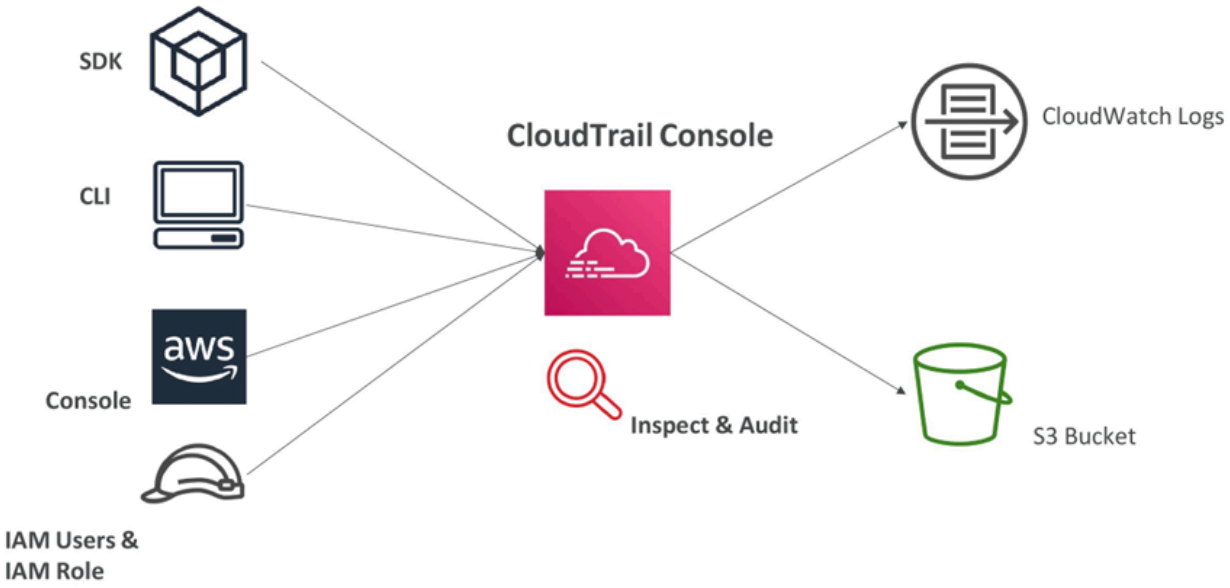
https://aws.amazon.com/quicksight/

# AWS CloudTrail



- Provides governance, compliance and audit for your AWS Account
- CloudTrail is enabled by default!
- Get an history of events / API calls made within your AWS Account by:
  - Console
  - SDK
  - CLI
  - AWS Services
- Can put logs from CloudTrail into CloudWatch Logs or S3
- A trail can be applied to All Regions (default) or a single Region.
- If a resource is deleted in AWS, investigate CloudTrail first!

# CloudTrail Diagram

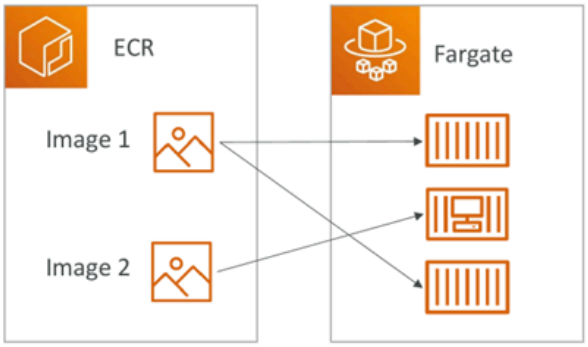



# Amazon Lightsail

- Virtual servers, storage, databases, and networking
- Low & predictable pricing
- Simpler alternative to using EC2, RDS, ELB, EBS, Route 53...
- Great for people with little cloud experience!
- Can setup notifications and monitoring of your Lightsail resources
- Use cases:
  - Simple web applications (has templates for LAMP, Nginx, MEAN, Node.js...)
  - Websites (templates for WordPress, Magento, Plesk, Joomla)
  - Dev / Test environment
- Has high availability but no auto-scaling, limited AWS integrations

# ECR

- Elastic Container Registry
- Private Docker Registry on AWS
- This is where you store your Docker images so they can be run by ECS or Fargate




# AWS Marketplace

- Digital catalog with thousands of software listings from independent software vendors (3rd party)
- Example:
  - Custom AMI (custom OS, firewalls, technical solutions...)
  - CloudFormation templates
  - Software as a Service
  - Containers
- If you buy through the AWS Marketplace, it goes into your AWS bill
- You can sell your own solutions on the AWS Marketplace

# AWS Support Plans Pricing



- Basic Support: free

| Developer | Business | Enterprise On-Ramp | Enterprise |
|---|---|---|---|
| Greater of $29.00<br>- or -<br>3% of monthly AWS charges | Greater of $100.00<br>- or -<br>10% of monthly AWS charges for the first $0--$10K<br>7% of monthly AWS charges from $10K--$80K<br>5% of monthly AWS charges from $80K--$250K<br>3% of monthly AWS charges over $250K | Greater of $5,500.00<br>- or -<br>10% of monthly AWS charges | Greater of $15,000.00<br>- or -<br>10% of monthly AWS charges for the first $0--$150K<br>7% of monthly AWS charges from $150K--$500K<br>5% of monthly AWS charges from $500K--$1M<br>3% of monthly AWS charges over $1M |

# AWS Ecosystem - AWS Support

| | |
|---|---|
| DEVELOPER | • Business hours email access to Cloud Support Associates<br>• General guidance: < 24 business hours<br>• System impaired: < 12 business hours |
| BUSINESS | • 24x7 phone, email, and chat access to Cloud Support Engineers<br>• Production system impaired: < 4 hours<br>• Production system down: < 1 hour |
| ENTERPRISE | • Access to a Technical Account Manager (TAM)<br>• Concierge Support Team (for billing and account best practices)<br>• Business-critical system down: < 15 minutes |

# Your AWS Certification journey

### FOUNDATIONAL

**Six months** of fundamental AWS Cloud and industry knowledge




### PROFESSIONAL

**Two years** of experience designing, operating, and troubleshooting solutions using the AWS Cloud




### ASSOCIATE

**One year** of experience solving problems and implementing solutions using the AWS Cloud




### SPECIALTY

Technical AWS Cloud experience in the Specialty domain as specified in the exam guide

