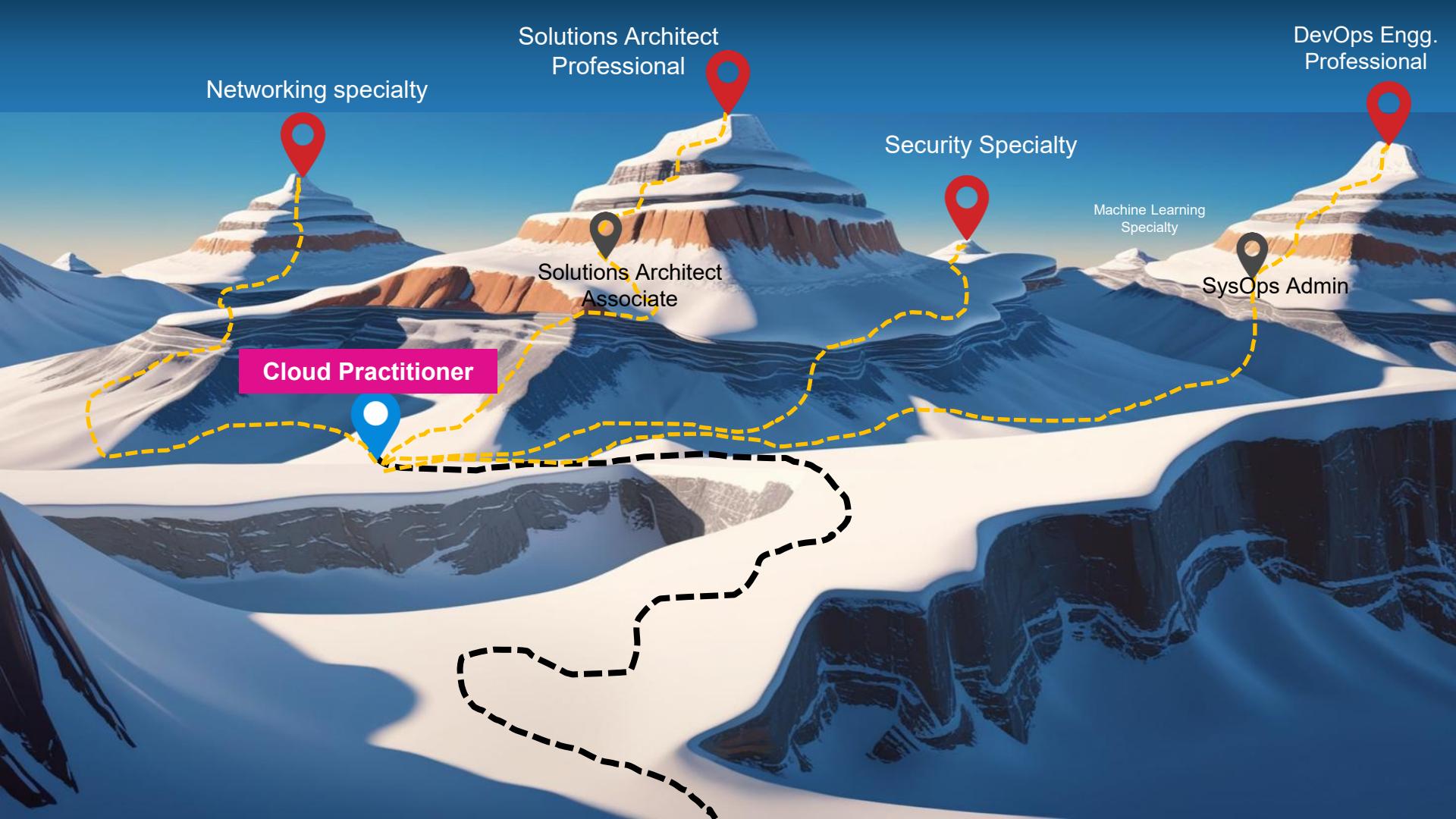


# Welcome to the course!

AWS Certified Cloud Practitioner – Foundational



Networking specialty

Solutions Architect  
Professional

DevOps Engg.  
Professional

Cloud Practitioner

Solutions Architect  
Associate

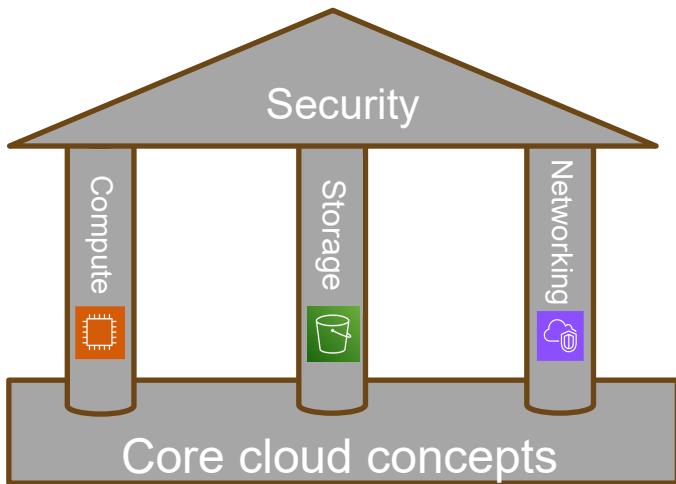
Security Specialty

Machine Learning  
Specialty

SysOps Admin

# The objectives of this course..

✓ To build a **very strong foundation** in AWS



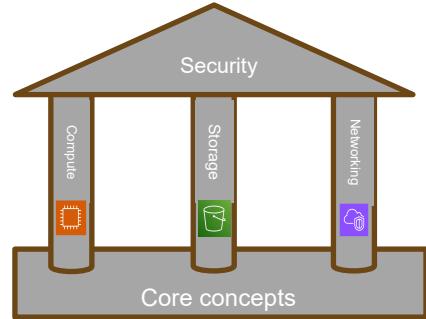
✓ To pass AWS Certified Cloud Practitioner Exam (CLF-C02)



Let's aim for... **100%**

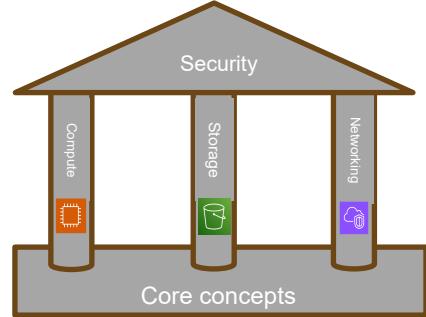
# After completing this course..

- ✓ You will acquire strong AWS foundation skills and practical knowledge about widely used AWS services
- ✓ 100% ready to take on your AWS Cloud Practitioner Certificate exam (CLF-C02)
- ✓ You will have hands-on experience to build simple solutions in AWS
- ✓ **You will be able to speak AWS language and participate confidently in AWS related discussions**



# Who should take this course?

- ✓ Anyone who wants to get AWS Cloud Practitioner Certified
- ✓ IT people who are looking to acquire cloud skills and shift their career in AWS
- ✓ Managers, senior executives to build foundational understanding of AWS cloud and AWS services that shall help them take right technical decisions
- ✓ Students who want to learn Cloud computing and who are preparing for their campus interviews



## Any pre-requisites with respect to background?

- Absolutely nothing ! Anyone interested to learn AWS can take this course and pass the AWS certification exam

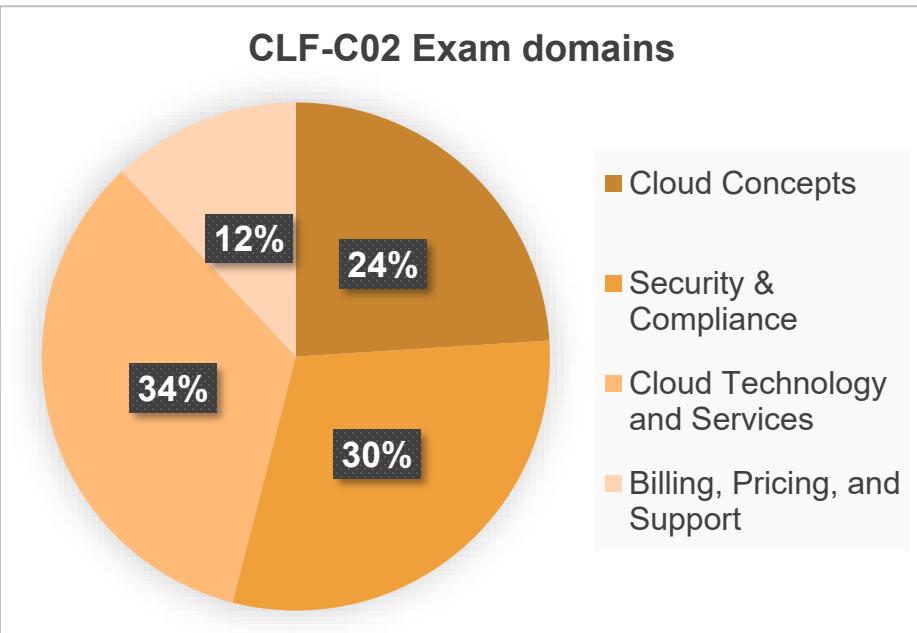


# What does this course contain?

- ✓ 20 hours of video lectures
- ✓ 25 sections, 200+ lectures
- ✓ Section summary (for quick recap) and Quizzes for every section
- ✓ 35+ Hands-on exercises with supporting assets (code base, instructions)
- ✓ 1 full Practice Test (65 questions)
- ✓ Full course slides can be downloaded (PDF document)

# Structure of this course

1. AWS Cloud – A Big picture (45 mins)
2. Course pre-requisites for exercises
3. Video lectures across all domains
  - AWS Cloud Concepts
  - Cloud Technology and Services
  - Security & Compliance
  - Billing, Pricing, and Support
4. Practice Test
5. Exam preparation



# Exam preparation

- AWS Sample exam questions walkthrough
- How to approach exam questions - Tips
- Time management and getting 30 mins extra time (as applicable)
- Getting 50% discount on your next certification exam

# How to approach this course?

- One or two sections in a day depending on the length of the section in the order.
- You should spend around 1.5-2hrs daily
- Try to complete this course in **2 weeks**

# How to connect?

1. Every lecture has a discussion panel where you can post your questions. I will try my best to answer all your questions there.
2. Reach via email at [awswithchetal@gmail.com](mailto:awswithchetal@gmail.com)
3. Provide your feedback for the course content as well as any new topics to be covered based on your exam experience.

# Know your instructor..

# Hello...I am Chetan !

- I have 20 years of IT industry experience
- I had been C/C++ developer in my early career, followed by DevOps Architect and Cloud Solutions Architect
- Currently, I work at AWS as a Senior Solutions Architect



<https://www.awswithchetan.com/>



<https://www.udemy.com/user/chetan-agrawal-4/>



<https://www.youtube.com/@AWSwithChetan>



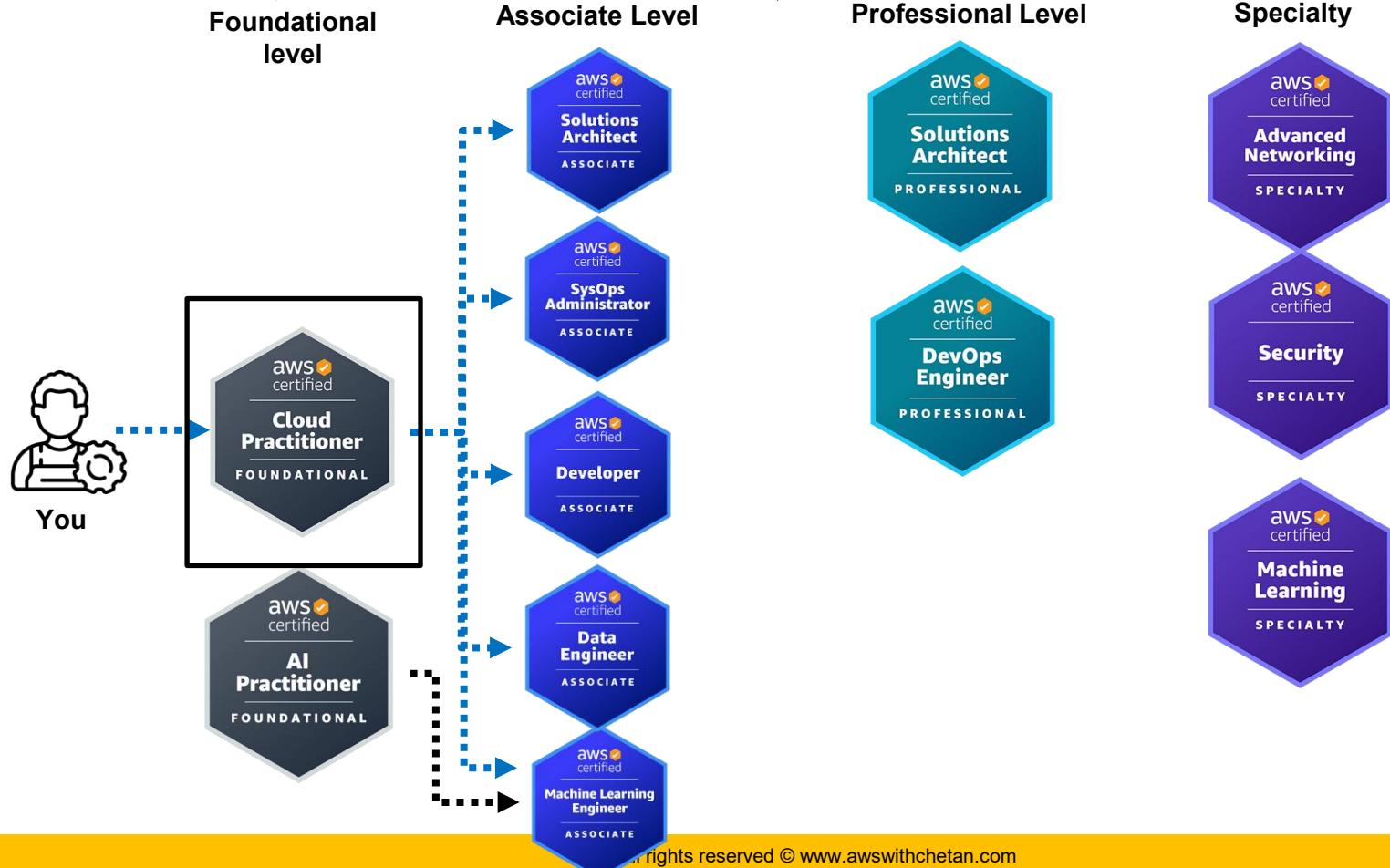
<https://www.linkedin.com/in/chetan-agrawal-30107310/>

200,000+ students, 4M+ views for youtube videos..





# Know your exam..



# Know your exam..

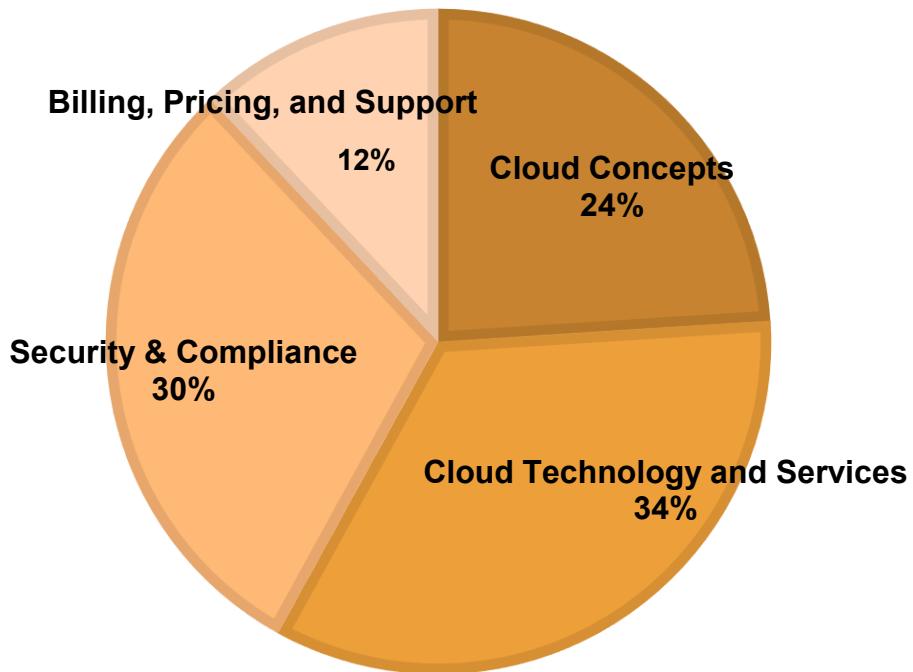
- Exam: **AWS Certified Cloud Practitioner**
- Exam code: CLF-C02
- Total 65 questions
- 50 questions are scored and 15 are unscored
- There are two types of questions on the exam:
  - Multiple choice: Has one correct response and three incorrect responses (distractors)
  - Multiple response: Has two or more correct responses out of five or more response options
- Minimum passing score: 700/1000

<b>Category</b>	Foundational
<b>Exam duration</b>	90 minutes
<b>Exam format</b>	65 questions; either multiple choice or multiple response
<b>Cost</b>	100 USD. Visit <a href="#">Exam pricing</a> for additional cost information, including foreign exchange rates
<b>Test in-person or online</b>	Pearson VUE testing center or online proctored exam
<b>Languages offered</b>	English, Japanese, Korean, Simplified Chinese, Traditional Chinese, Bahasa (Indonesian), Spanish (Spain), Spanish (Latin America), French (France), German, Italian, and Portuguese (Brazil)

<https://aws.amazon.com/certification/certified-cloud-practitioner/>

# Exam content Domains and weight

**CLF-C02 EXAM DOMAINS**



# Sample questions



## AWS Certified Cloud Practitioner Sample Exam Questions

**1) Why is AWS more economical than traditional data centers for applications with varying compute workloads?**

- A) Amazon EC2 costs are billed on a monthly basis.
- B) Users retain full administrative access to their Amazon EC2 instances.
- C) Amazon EC2 instances can be launched on demand when needed.
- D) Users can permanently run enough instances to handle peak workloads.

**2) Which AWS service would simplify the migration of a database to AWS?**

- A) AWS Storage Gateway
- B) AWS Database Migration Service (AWS DMS)
- C) Amazon EC2
- D) Amazon AppStream 2.0

**3) Which AWS offering enables users to find, buy, and immediately start using software solutions in their AWS environment?**

- A) AWS Config
- B) AWS OpsWorks
- C) AWS SDK
- D) AWS Marketplace

# Retake exam..

TRAINING AND CERTIFICATION

## Retake your AWS Foundational Certification exam for free

[Schedule your exam](#)

### Get ready. Get set. Get AWS Certified—and a free retake!

Ready to earn the [AWS Certified Cloud Practitioner](#) or the new [AWS Certified AI Practitioner](#)? Register for your exam using the code AWSRetake2025 between October 9, 2024 and February 15, 2025 to be eligible for a free exam retake, if you need it.\*

Follow the steps below to take advantage of this opportunity:

1. Register for the exam: Book your appointment for either the AWS Certified AI Practitioner or AWS Certified Cloud Practitioner and enter the code AWSRetake2025 at checkout.
2. Prepare for the exam: Go from start to certified. Access our easy-to-follow 4-step Exam Prep Plans for [AWS Certified Cloud Practitioner](#) and [AWS Certified AI Practitioner](#) on AWS Skill Builder, our online learning center, so you can approach exam day with confidence.
3. Take the first attempt of your exam between October 9, 2024 and February 15, 2025.
4. If you fail, register for a second attempt of the same exam before March 31, 2025 and your exam retake will automatically be free!

Register now for your exam with code: AWSRetake2025

[Schedule your exam](#)

# AWS Cloud – A Big picture

# Cloud computing..

On-demand delivery of electricity over the wire with pay-as-you-go pricing.

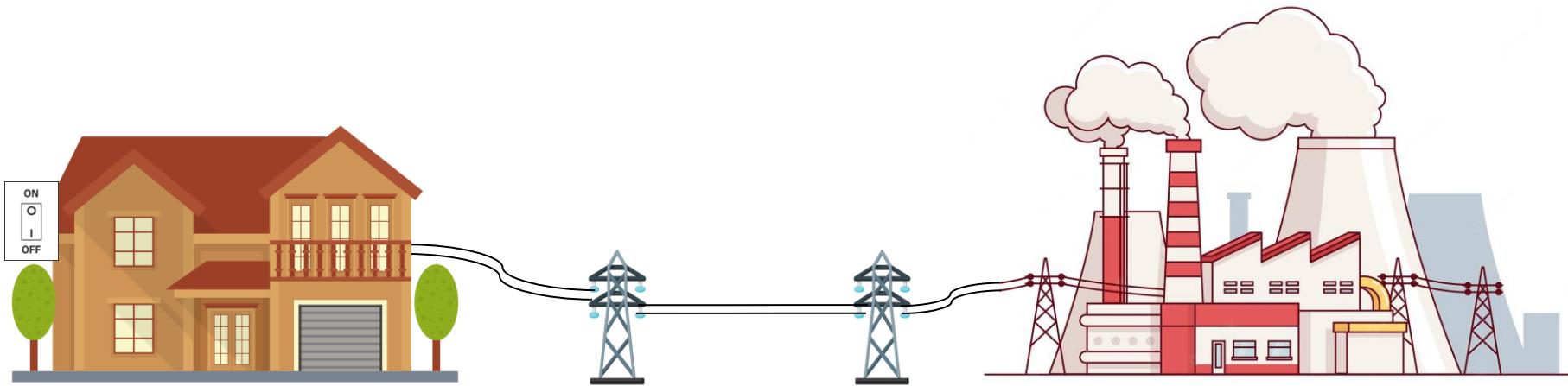
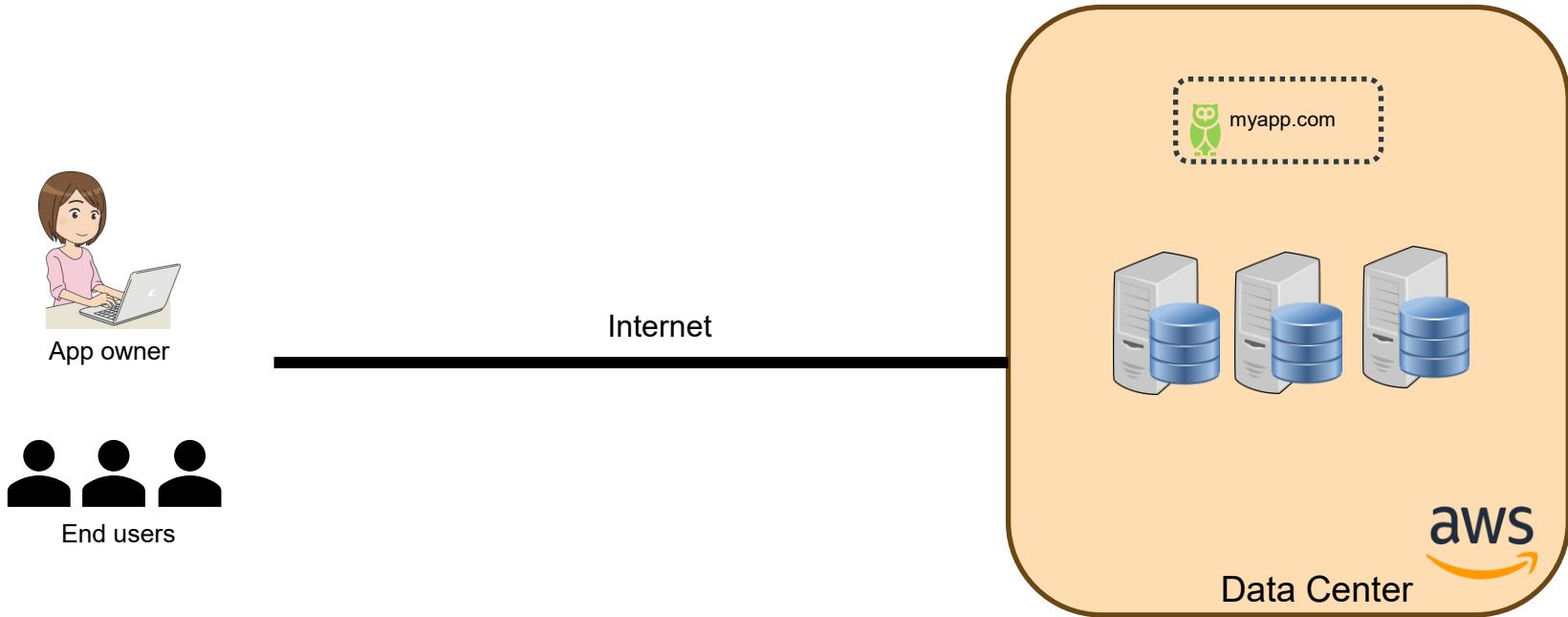


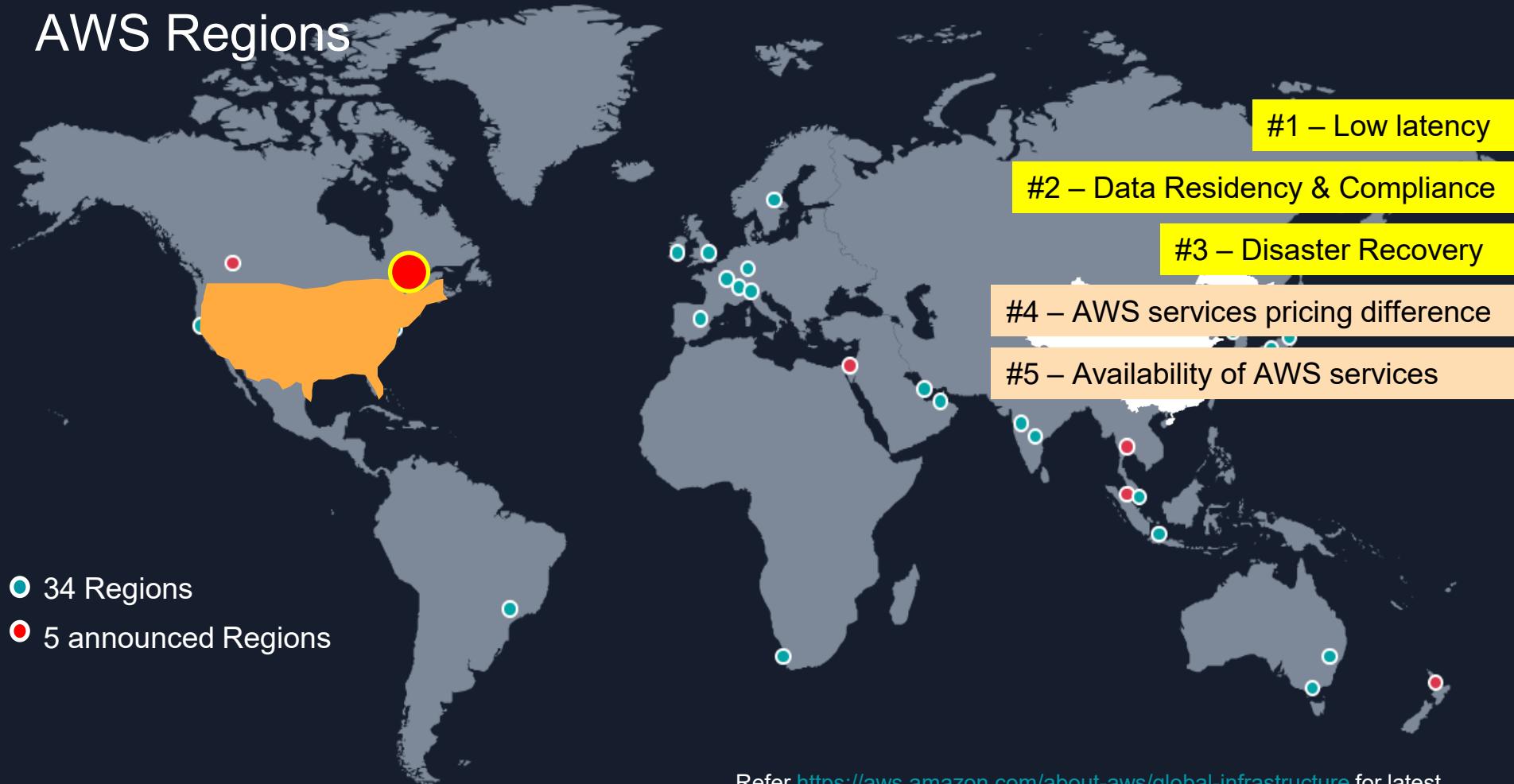
image: Freepik.com

# Cloud computing..

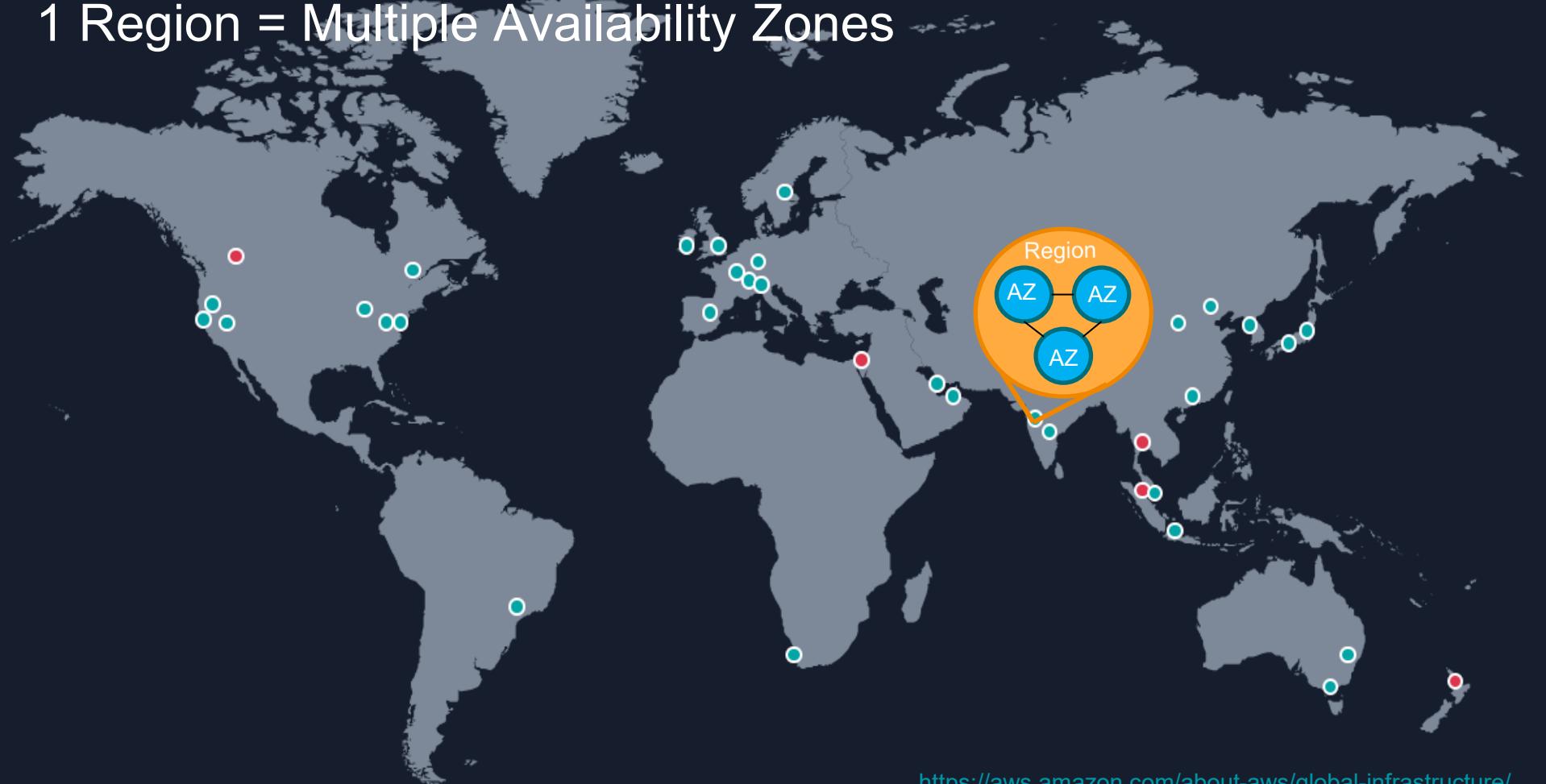
Cloud computing is the **on-demand** delivery of **IT resources**



# AWS Regions



# 1 Region = Multiple Availability Zones



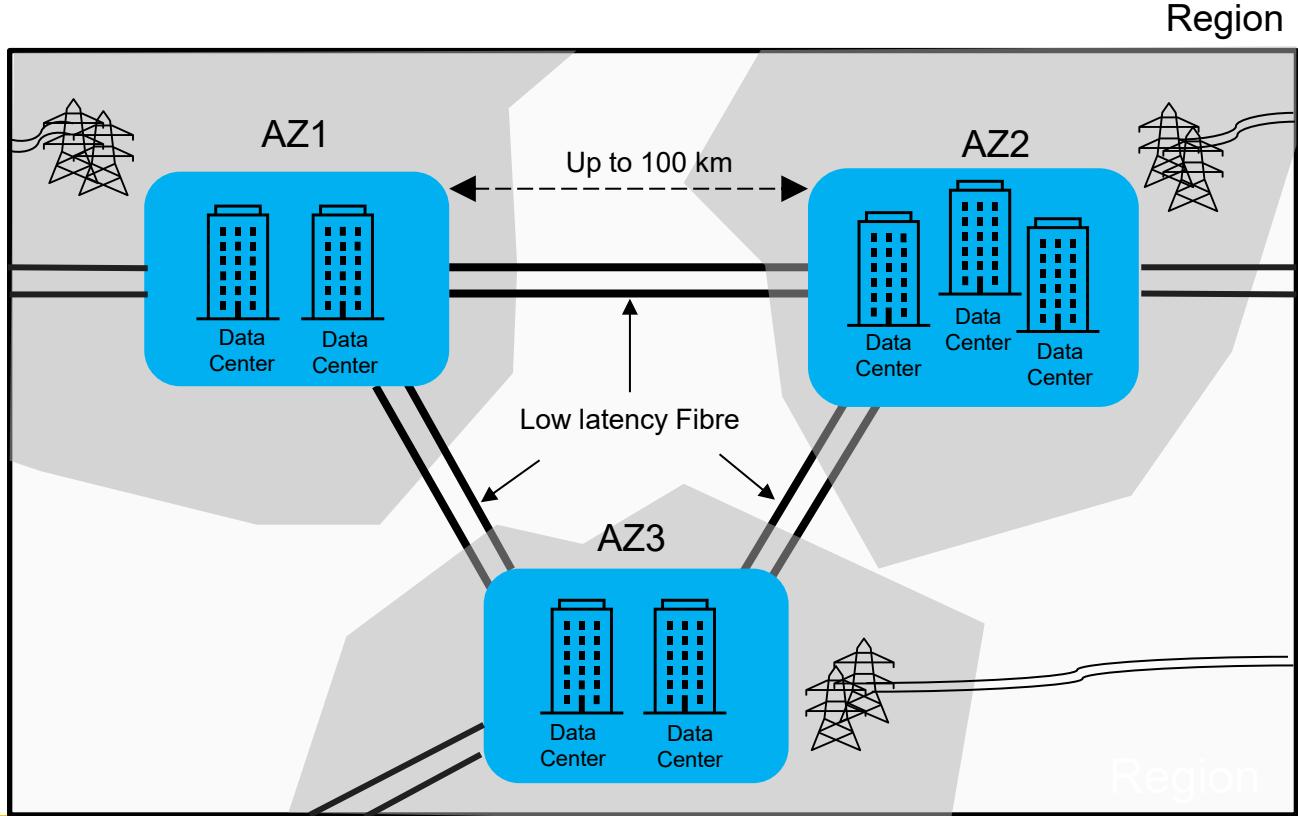
# 1 Region = Multiple AZs

## 1 AZ = Multiple data centers

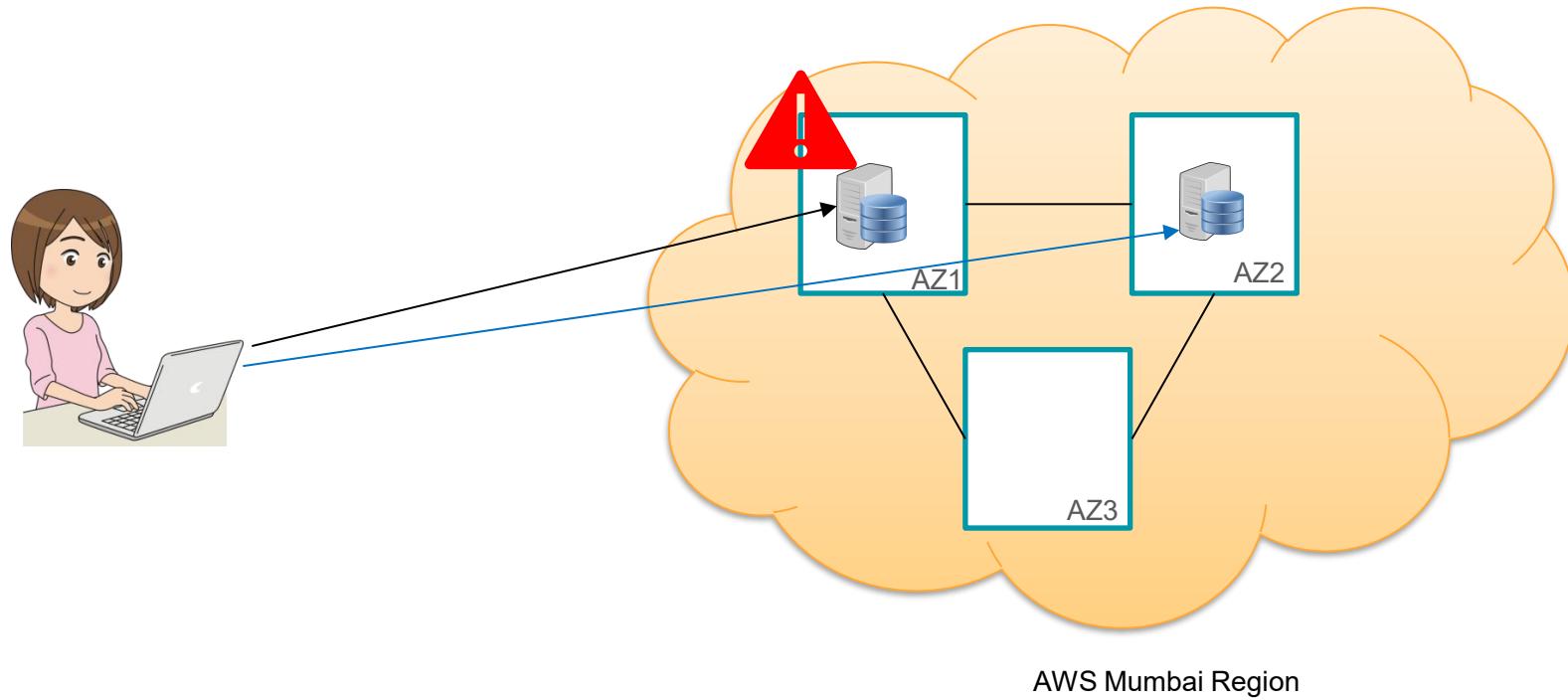
Different floodplains  
(in most cases)

Redundant Power Supply

Redundant Network Connectivity

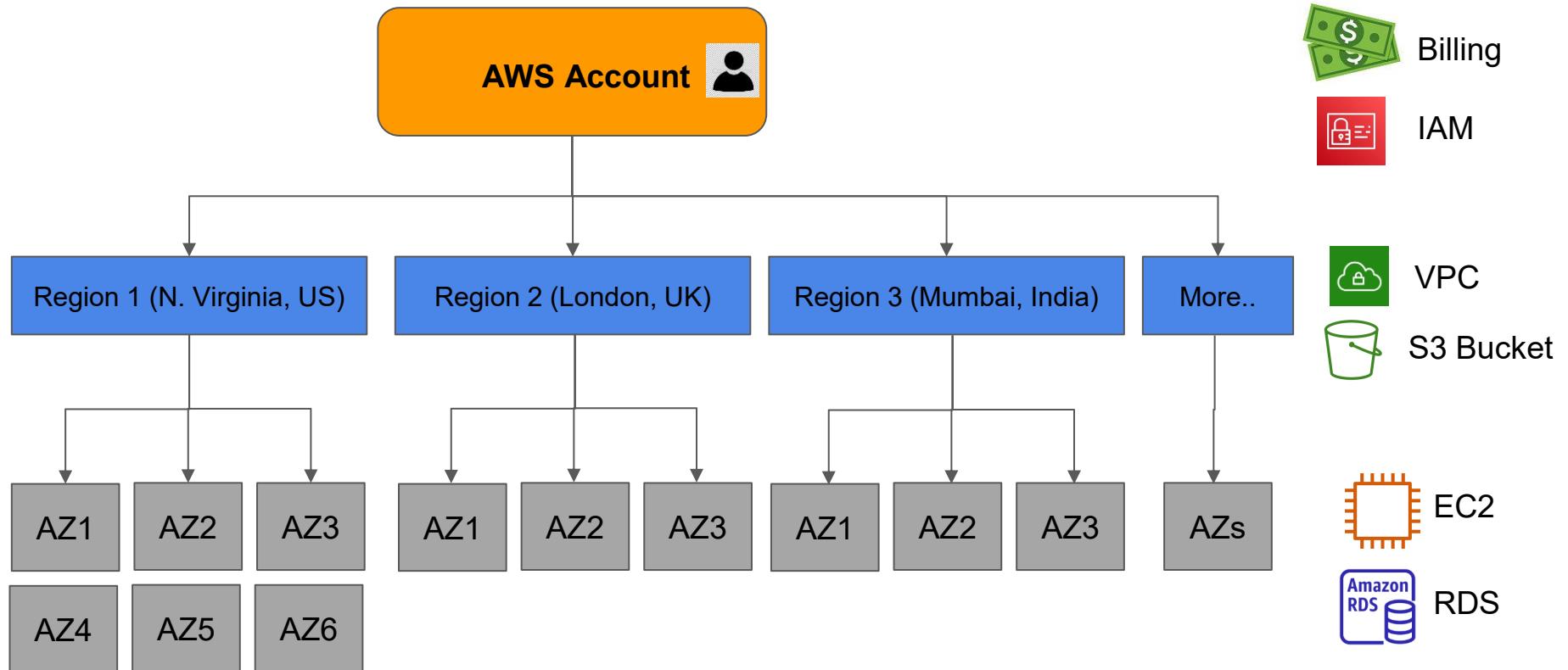


# What AZ means to you?

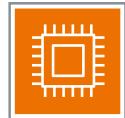


AWS Mumbai Region

# AWS Account



# AWS Services categories



Compute



Application  
Integration



Containers



IoT



Storage



Machine Learning  
and Artificial  
Intelligence



Databases



Management and  
Governance



Networking



Business  
Applications



Developer  
Tools



Media



Security



Cloud Financial  
Management



Mobile  
Applications



Migration



Analytics



Contact  
Center



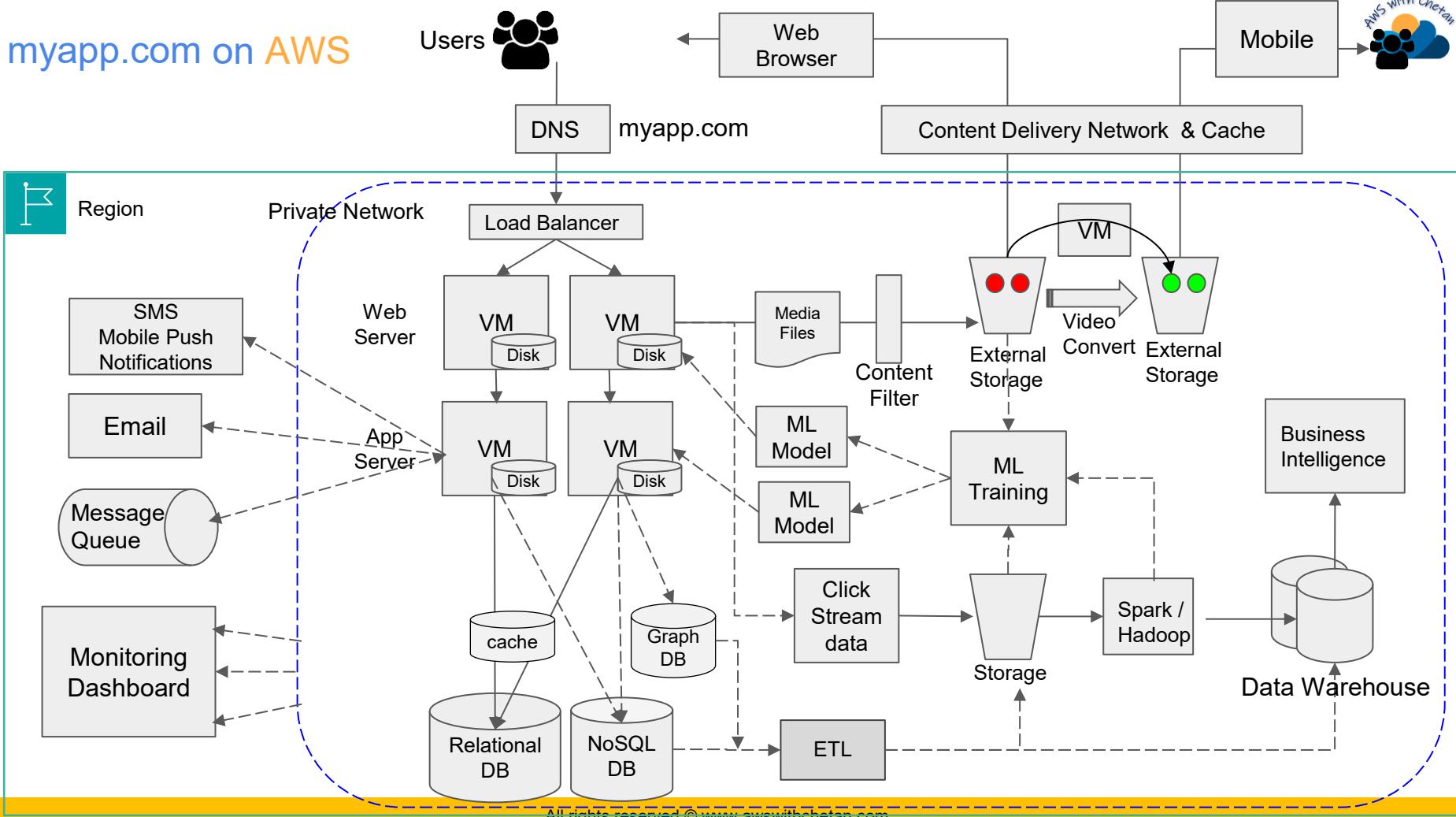
Games

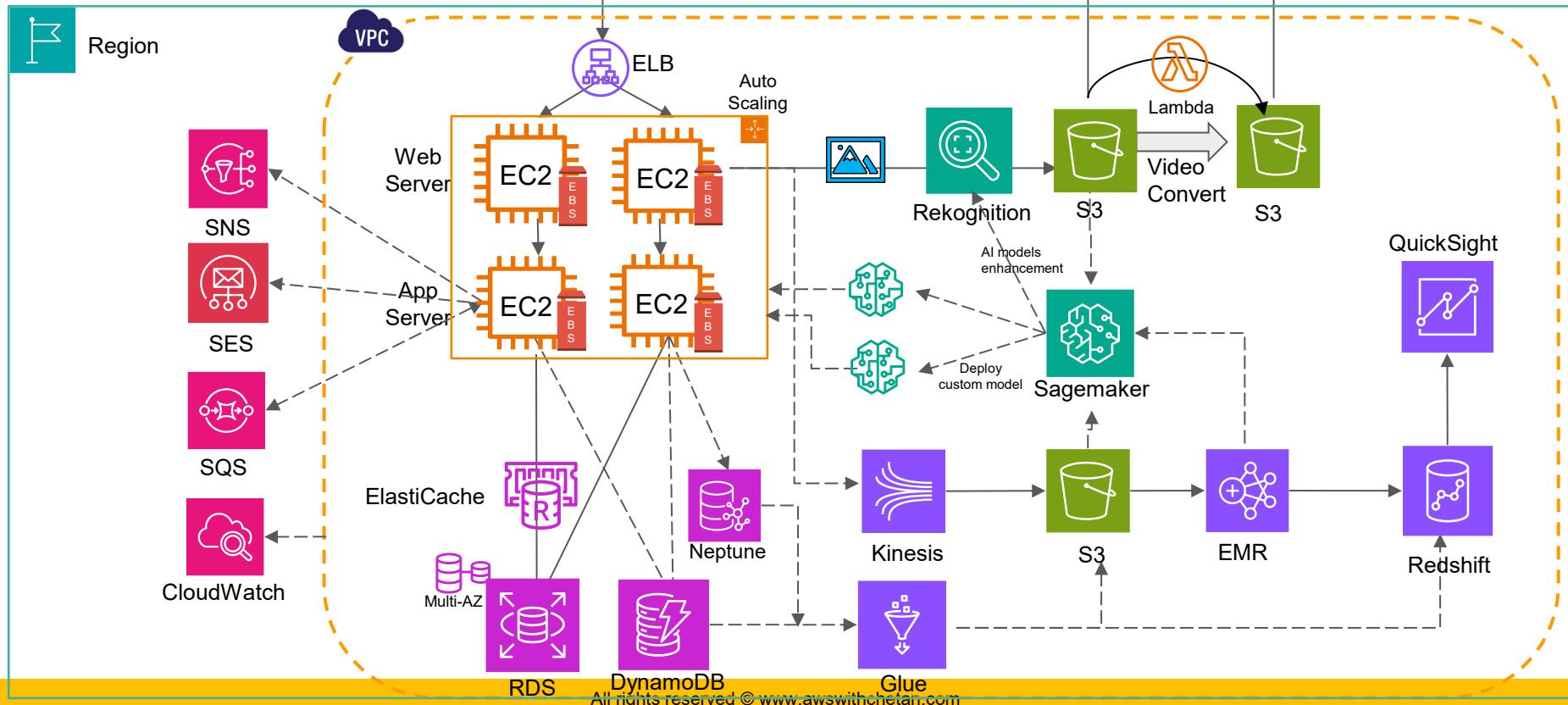
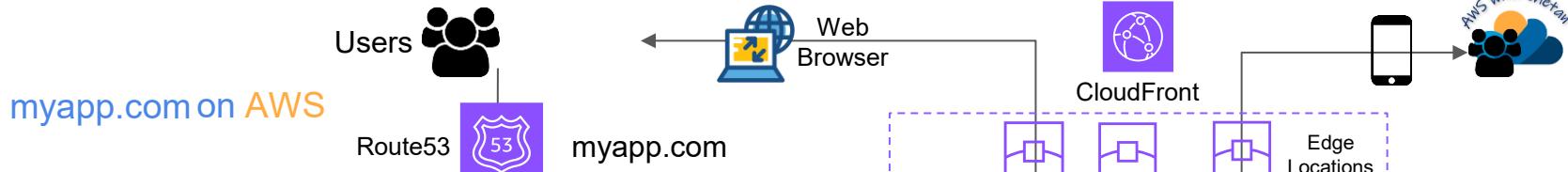


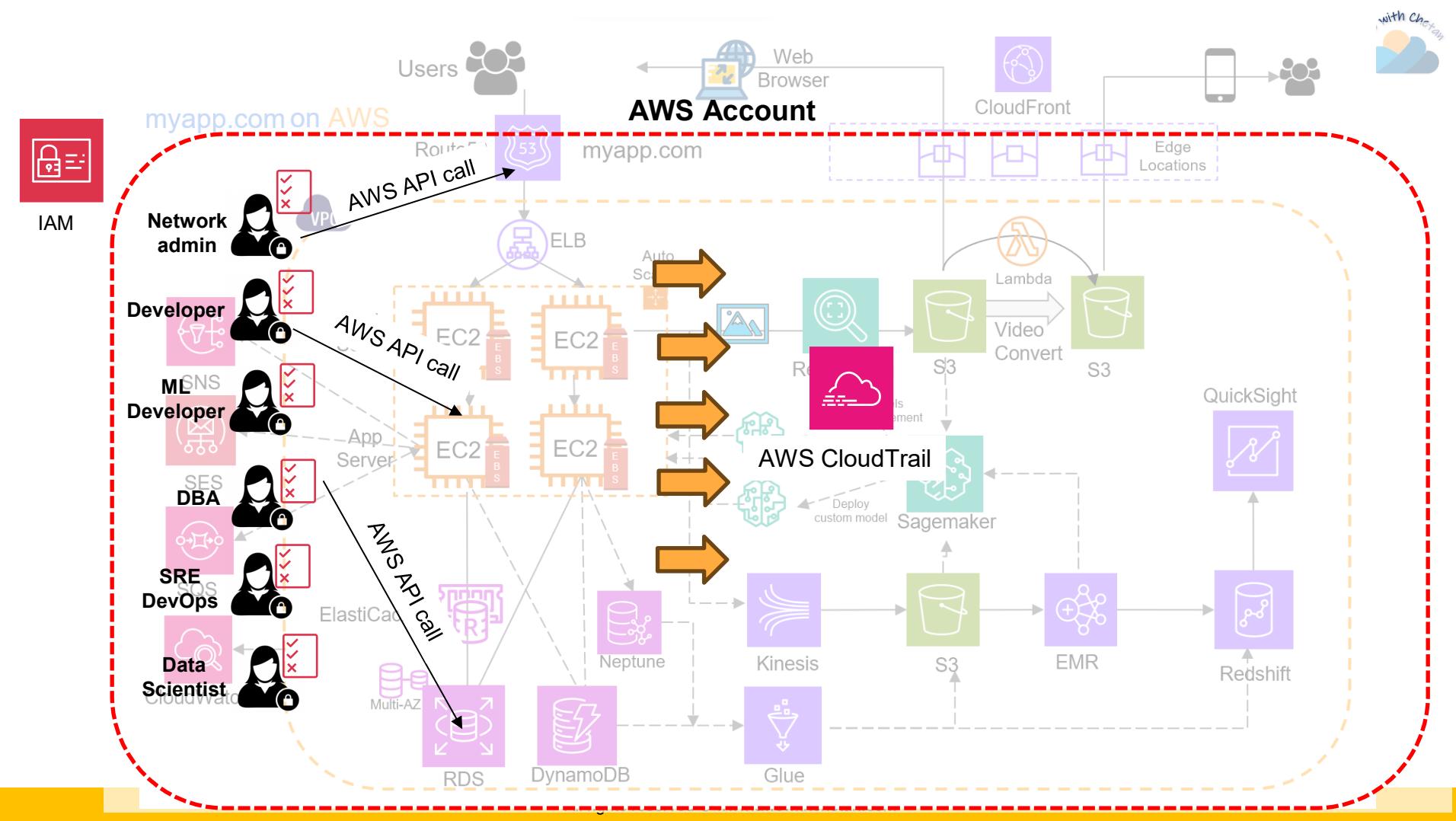
Satellite

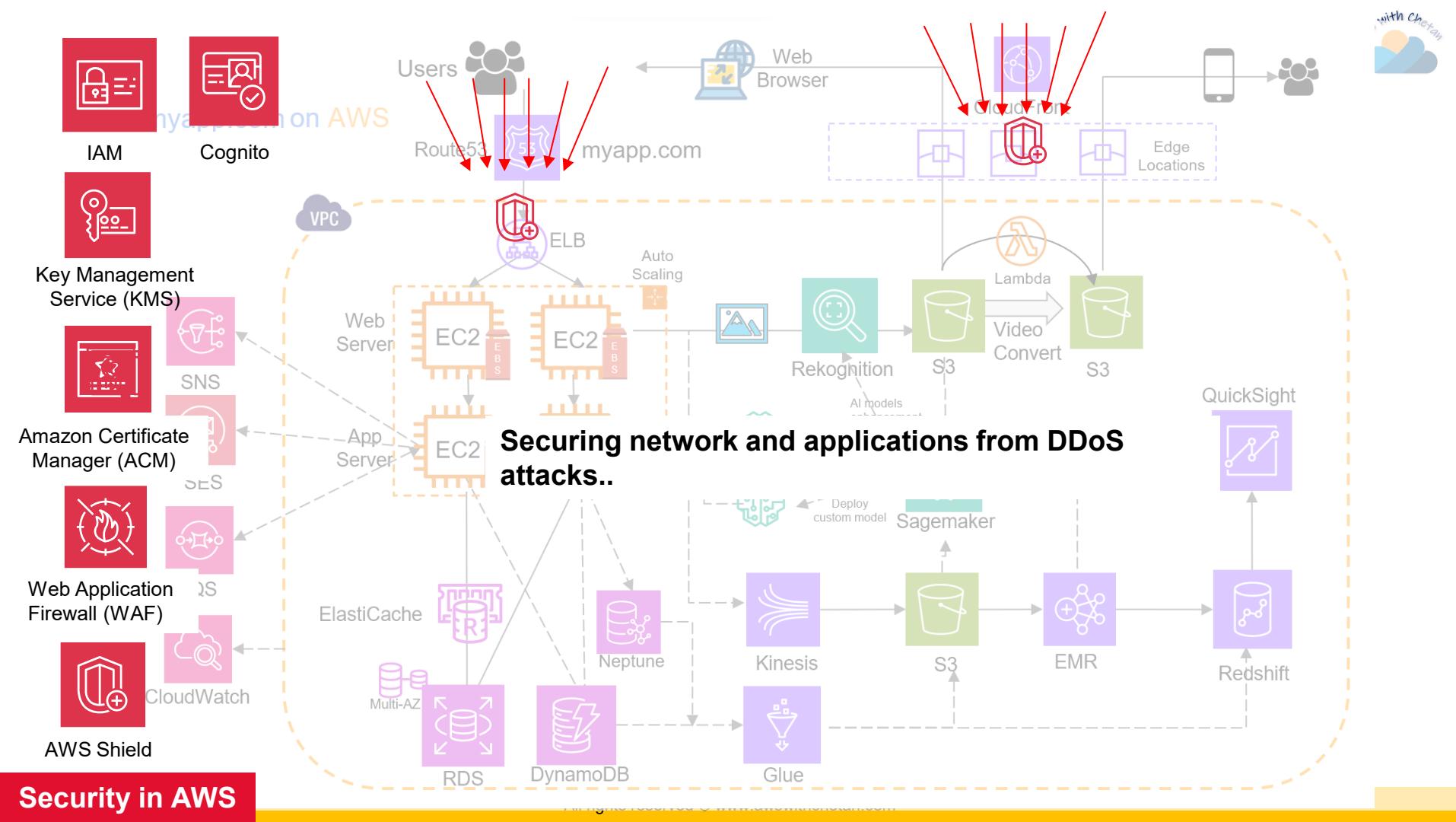
Let's try to build an architecture for simplified version  
of a social media application

# myapp.com on AWS

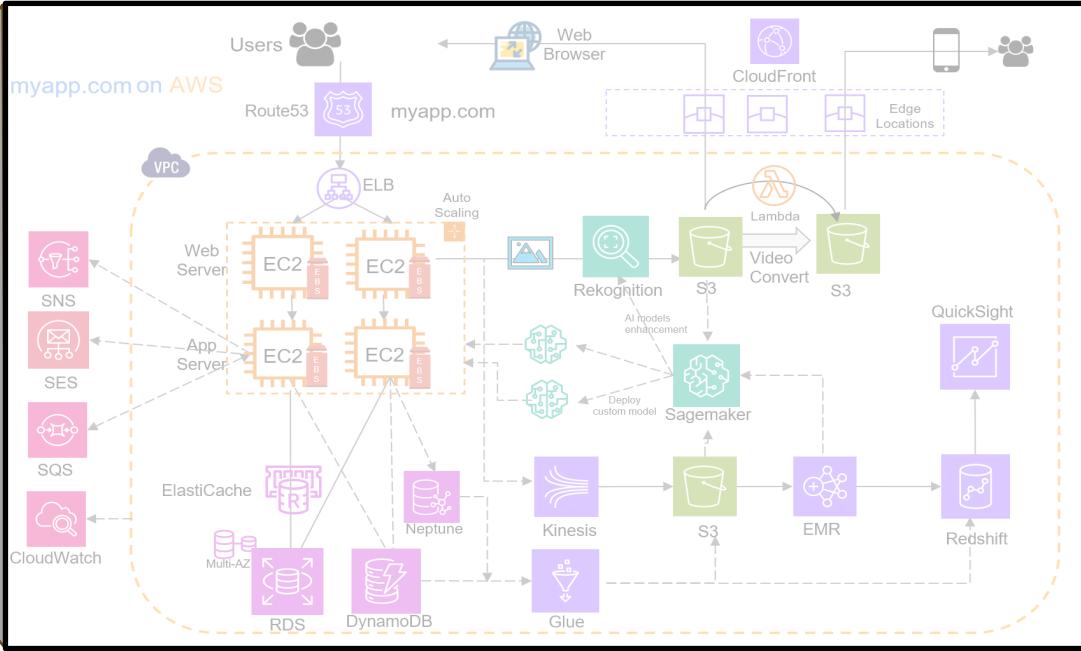
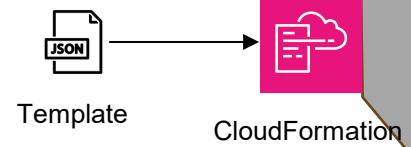








# Infrastructure as a Code



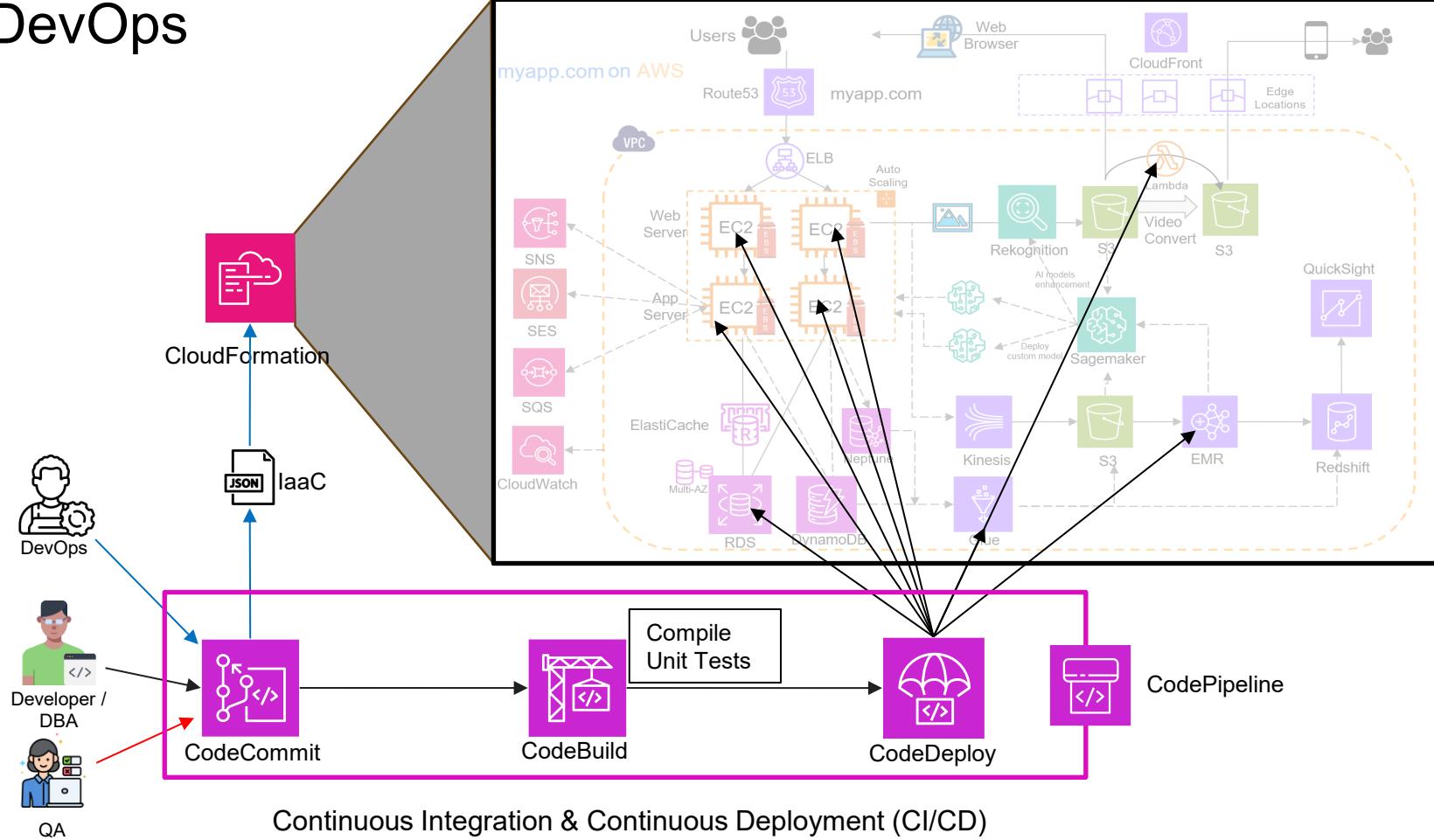
Repeatable

Consistent

Idempotent

As a stack

# DevOps



# Prerequisites for hands-on exercises

# Pre-requisites for the exercises..

1. Create a new AWS account, verify account activation and account limits
2. Set up AWS cost budget
3. Create an IAM user for yourself
4. Create SSH key pair for EC2 exercises
5. (Optional but recommended) Buy a public domain name and configure Route 53 DNS

# Create an AWS account

## For creating AWS account, you will need:

1. Email Address
2. Mobile Number
3. Billing Address (no proofs required)
4. Valid Debit Card or Credit Card

For new AWS accounts you get \$100 credits for 6 months  
+ additional \$100 (with some activities)

Visit <https://aws.amazon.com/free/>

## Free Plan

Experience AWS for up to 6 months without cost or commitment

- ✓ Receive up to \$200 USD in credits
- ✓ Includes free usage of select services
- ✓ No charges incurred unless you switch to the Paid Plan
- ✗ Workloads scale beyond credit thresholds
- ✗ Access to all AWS services and features

## Paid Plan

Develop production-ready workloads with access to over 150 AWS services

- ✓ Receive up to \$200 USD in credits
- ✓ Includes free usage of select services
- ✓ Pay beyond credit thresholds
- ✓ Workloads scale beyond credit thresholds
- ✓ Access to all AWS services and features

# Activities to get additional \$100 AWS credits

1. Launch an instance using Amazon EC2
2. Use a foundational model in the Amazon Bedrock playground
3. Set up a cost budget using AWS Budgets
4. Create a web app using AWS Lambda
5. Create an Amazon RDS database

# Verify account activation and account limits

1. Search for ec2 in the top search bar and go to ec2 console to verify that your account is fully active.

The screenshot shows the AWS EC2 Home page for the Asia Pacific (Mumbai) Region. The left sidebar includes links for EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area displays 'Account attributes' with supported platforms (VPC), default VPC (vpc-0f27cdafbf7e30a3), and various settings like EBS encryption and zones. Below this, the 'Resources' section shows zero instances, auto scaling groups, dedicated hosts, elastic IPs, instances, key pairs, load balancers, placement groups, security groups, snapshots, and volumes. A callout box suggests using the AWS Launch Wizard for Microsoft SQL Server Always On availability groups. The 'Launch instance' section allows launching an instance, and the 'Service health' section indicates the service is operating normally.

# Verify account limits

1. Login to AWS account -> Go to Service Quotas -> EC2 -> [Running On-Demand Standard \(A, C, D, H, I, M, R, T, Z\) instances](#)

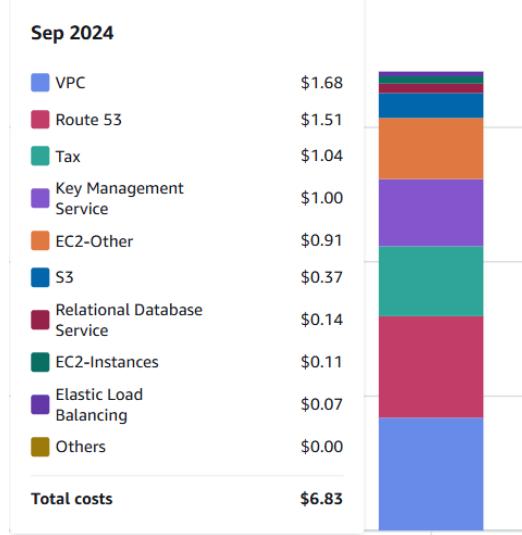
The screenshot shows the AWS Service Quotas console for the Amazon Elastic Compute Cloud (Amazon EC2). The left sidebar includes links for Dashboard, AWS services, Quota request history, Organization, and Quota request template. The main content area displays a table of service quotas for EC2. A search bar at the top of the table is set to 'on-demand'. The table has columns for Quota name, Applied quota value, AWS default quota value, and Adjustable status. One row, 'Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances', is highlighted with a red border.

Quota name	Applied quota value	AWS default quota value	Adjustable
Running On-Demand DL instances	0	0	Yes
Running On-Demand F instances	176	0	Yes
Running On-Demand G and VT instances	768	0	Yes
Running On-Demand High Memory instances	0	0	Yes
Running On-Demand HPC instances	0	0	Yes
Running On-Demand Inf instances	0	0	Yes
Running On-Demand P instances	460	0	Yes
Running On-Demand Standard (A, C, D, H, I, M, R, T, Z) instances	1,920	5	Yes

# Setup cost budget

# Cost for performing the exercises

- The estimated charge for all exercises can be up to **\$10**
- You will get \$100 default credits and additional \$100 as we complete all the required activities
- So effectively, you will have to pay 0 charges for all the exercises in this course



- Tip: Most of the exercises are independent and hence you should terminate/delete all the resources that you create during the exercise.

*[Note: Free tier credits are available only for first 6 months]*

# Activities to get additional \$100 AWS credits

1. Launch an instance using Amazon EC2
2. Use a foundational model in the Amazon Bedrock playground
3. Set up a cost budget using AWS Budgets ✓
4. Create a web app using AWS Lambda
5. Create an Amazon RDS database

# Set up AWS cost budget

Go to Billing and Cost Management -> Budgets

- a. Use a template -> Monthly cost budget
- b. Provide Budget name and budget value in USD (e.g. 5)
- c. Provide the email id to which you should receive an email when usage exceeds the budgeted value
- d. Create budget

AWS Budgets: 300 has exceeded your alert threshold [Inbox](#)

 budgets@costalerts.amazonaws.com  
to me ▾

Sat, 5 Oct, 06:11 (12 days ago) [Star](#) [Reply](#) [Forward](#) [More](#)



October 05, 2024

AWS Budget Notification  
AWS Account 387258180757

Dear AWS Customer,

You requested that we alert you when the **actual cost** associated with your 300 budget **exceeds \$4.00** for the current month. The month **actual cost** associated with this budget is **\$4.26**. You can find additional details below and by accessing the AWS Budgets dashboard.

Budget Name	Budget Type	Budgeted Amount	Alert Type	Alert Threshold	ACTUAL Amount
300	Cost	\$5.00	ACTUAL	> \$4.00	\$4.26

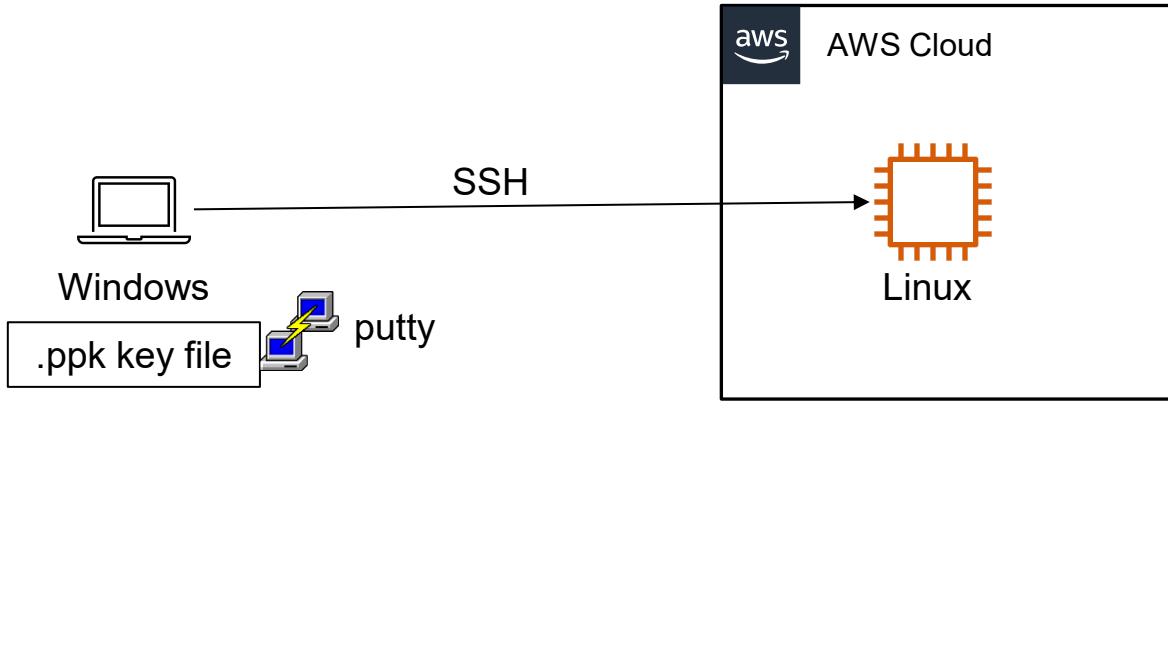
[Go to the AWS Budgets dashboard](#)

Email sent by AWS

# Create an IAM user

1. Login to your AWS account using Root user (email id/password)
2. Create IAM admin user. This user should be used for all the exercises. Do not use Root user.
  - i. IAM console -> Users -> Add user
  - ii. User name: admin (or whatever you prefer)
  - iii. Check box: Provide user access to AWS Management console
  - iv. Select: I want to create an IAM user
  - v. Console Password: Custom password -> Use strong password
  - vi. Uncheck: Users must create a new password at next sign-in -> Next
  - vii. Permissions Options -> Attach Policies directly -> Permissions policies -> Select "[AdministratorAccess](#)" -> Next
  - viii. Create User
3. Logout of your current session and log back in using IAM user
  - i. Visit Go to <https://aws.amazon.com> -> Sign In to the Console
  - ii. Enter IAM user name
  - iii. Enter IAM user password

# Connecting to EC2 instance



# Connecting to EC2 instance



# Create SSH key-pair

SSH key pairs have regional scope. Create SSH key pairs for the regions that you will be using for your exercises.

1. Go to **EC2 console** -> Left panel -> Network & Security -> Key pairs
2. Create key pair -> Name: **mumbai-key**, Key pair type: RSA
3. Private key file format: .pem if you use Linux/Mac workstation or .ppk if you use Windows workstation and use PuTTy to connect over the SSH -> Create key pair and save it on your machine.
4. Repeat the same process for your second AWS region.

EC2 > [Key pairs](#) > Create key pair

## Create key pair Info

**Key pair**  
A key pair, consisting of a private key and a public key, is a set of security credentials that you use to prove your identity when connecting to an instance.

Name  The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type Info  
 RSA  ED25519

Private key file format  
 .pem For use with OpenSSH  
 .ppk For use with PuTTY

Tags - *optional*  
No tags associated with the resource.  
[Add new tag](#)  
You can add up to 50 more tags.

[Cancel](#) [Create key pair](#)

# Using SSH key

## Windows Users:

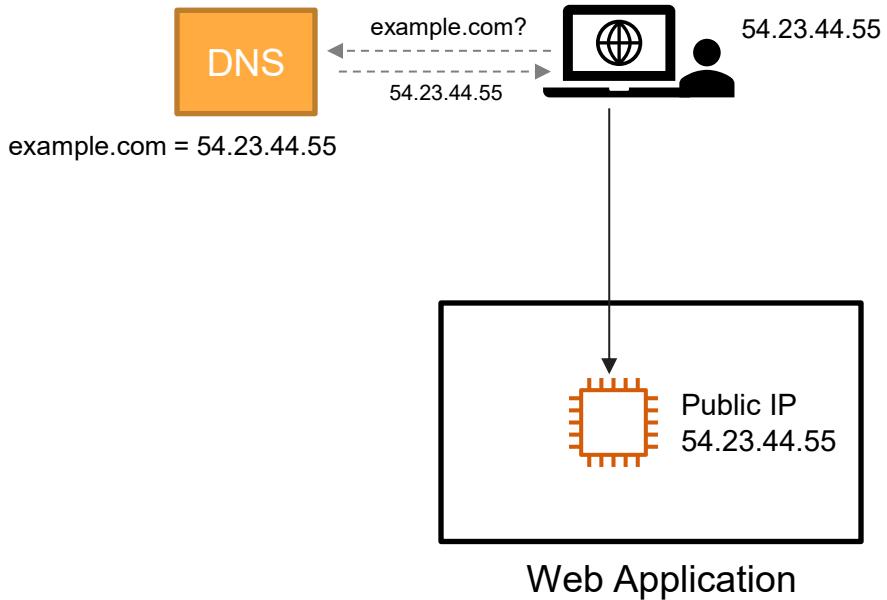
1. If you downloaded .ppk key file then you can directly use that key for connecting using PuTTy client
2. Download Putty software from <https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html>
3. If you downloaded .pem key file and you want to use PuTTy client then you need to first convert .pem format to .ppk
  - a. For this download the PuTTygen software from the website above
  - b. Open PuTTygen -> Load -> browse to .pem file (you may have to select All Files (\*.\*) -> Save private key (RSA) -> Save (without passphrase)

## Linux Users:

1. If you are having Linux/Ubuntu/Mac OS in your local workstation then you can use .pem key directly
2. Make sure you change the key permissions using command: **\$chmod 400 <key.pem>**
3. You can now use native terminal and .pem file for logging into the Linux EC2 instance:  
**\$ssh -i yourkey.pem ec2-user@publicip**

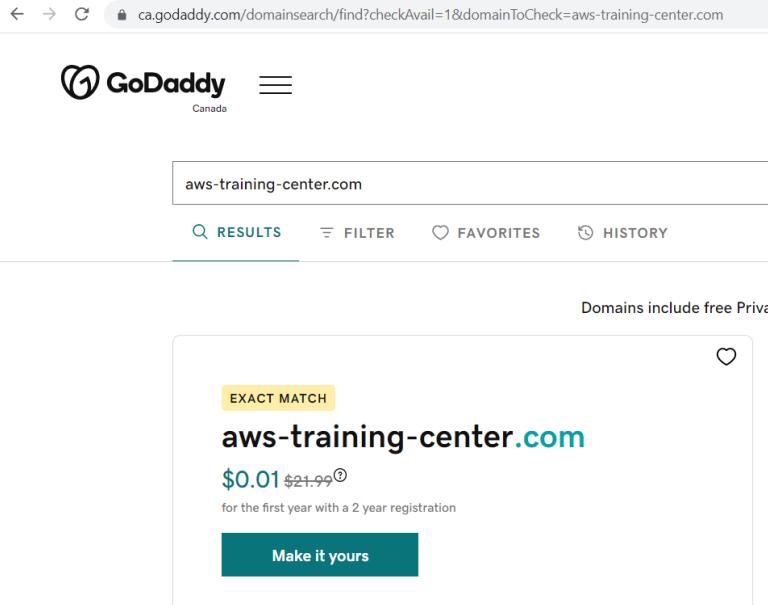
*\*ec2-user is a default username for Amazon Linux OS. For other OS its different e.g. ubuntu for Ubuntu OS, Administrator for Windows.*

# Why we need Public domain name?



# Buy a public domain name of your choice

1. Some of the exercises need a public domain name
2. It's recommended that you buy any public domain name of your choice and use it for those exercises
3. It may cost anywhere between \$5 to \$20 for 1 year depending on the domain name you choose.
4. You may want to purchase it from godaddy.com or any other public domain name service provider.

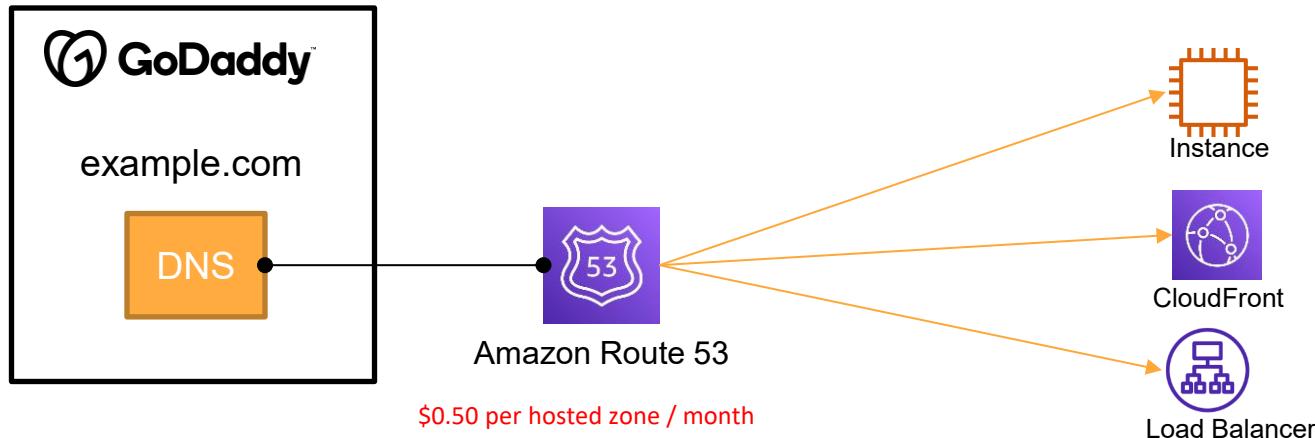


The screenshot shows a GoDaddy domain search interface. The URL in the address bar is ca.godaddy.com/domainsearch/find?checkAvail=1&domainToCheck=aws-training-center.com. The search term 'aws-training-center.com' is entered in the search bar. Below the search bar are buttons for 'RESULTS', 'FILTER', 'FAVORITES', and 'HISTORY'. A note at the bottom says 'Domains include free Priva'. The main result is displayed in a card with the heading 'EXACT MATCH' and the domain 'aws-training-center.com'. The price is listed as '\$0.01 \$21.99'. A note below the price says 'for the first year with a 2 year registration'. A large green button at the bottom right of the card says 'Make it yours'.

**Note:** www is a subdomain. Once you buy your domain name e.g. awswithchetan.com and then you can use any subdomain under it e.g. [www.awswithchetan.com](http://www.awswithchetan.com) or [web.awswithchetan.com](http://web.awswithchetan.com) likewise. Also, the Top Level Domain (TLD) can be anything e.g. .in or .live whatever. It need **not be** .com. Generally .com domains are little costlier than other top level domains.

# Let's setup a Route 53 DNS for your domain

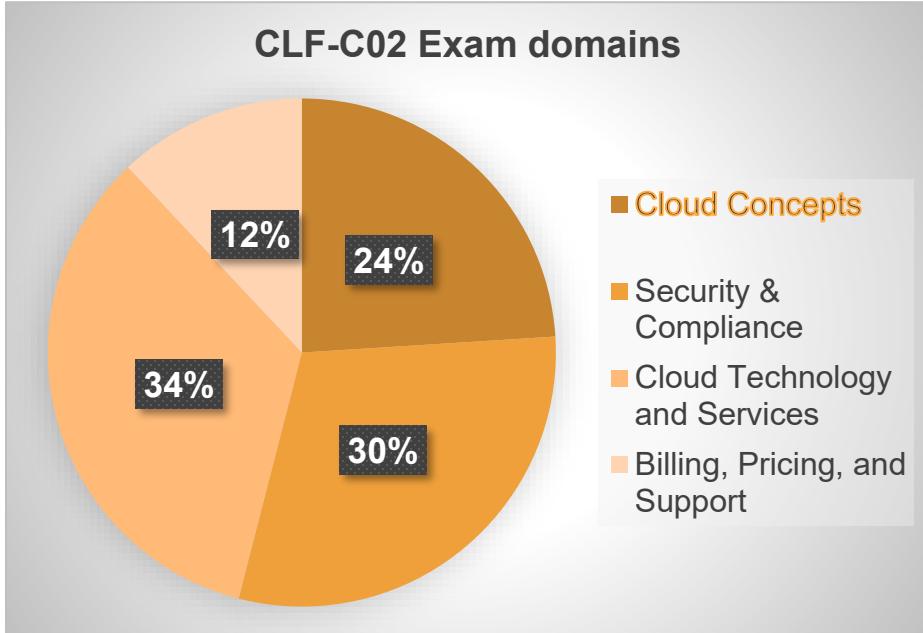
1. Go to AWS Route 53 console
2. Create a Public Hosted Zone (PHZ) with the same domain name that you purchased
3. From PHZ, note down the names of the 4 nameservers (NS records)
4. Go to your domain portal (e.g. godaddy dashboard) and go to DNS management
5. Replace the preconfigured nameservers with Route53 nameservers and save the changes.



# Cloud Concepts

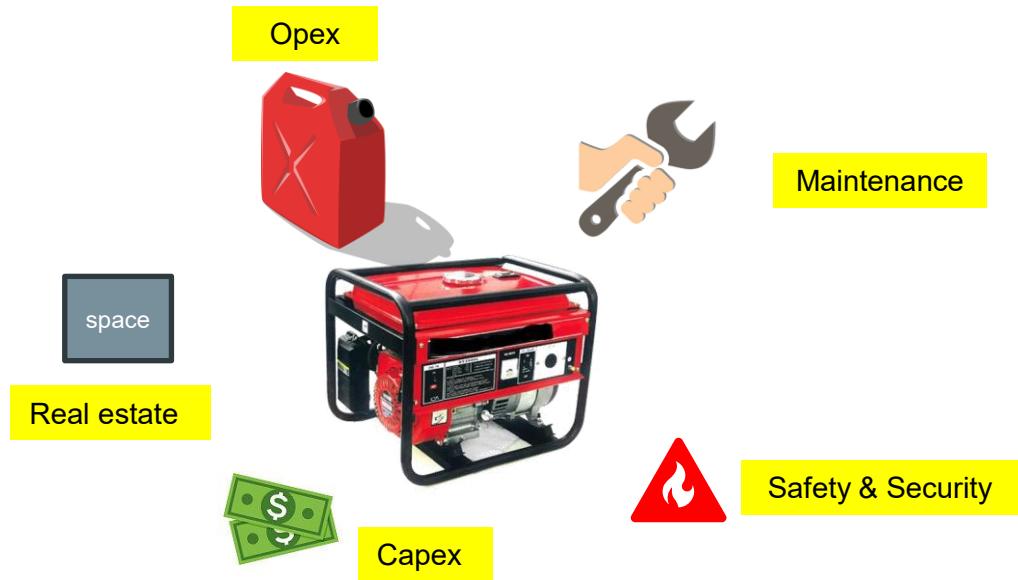
# In this section..

- **What is Cloud Computing?**
- Characteristics of the Cloud computing
- Advantages of the Cloud computing
- Deployment models of the Cloud
  - Public, Private & Hybrid
- Types of the Cloud computing
  - IaaS, PaaS, SaaS
- AWS Well Architected Framework
- AWS Cloud Adoption Framework

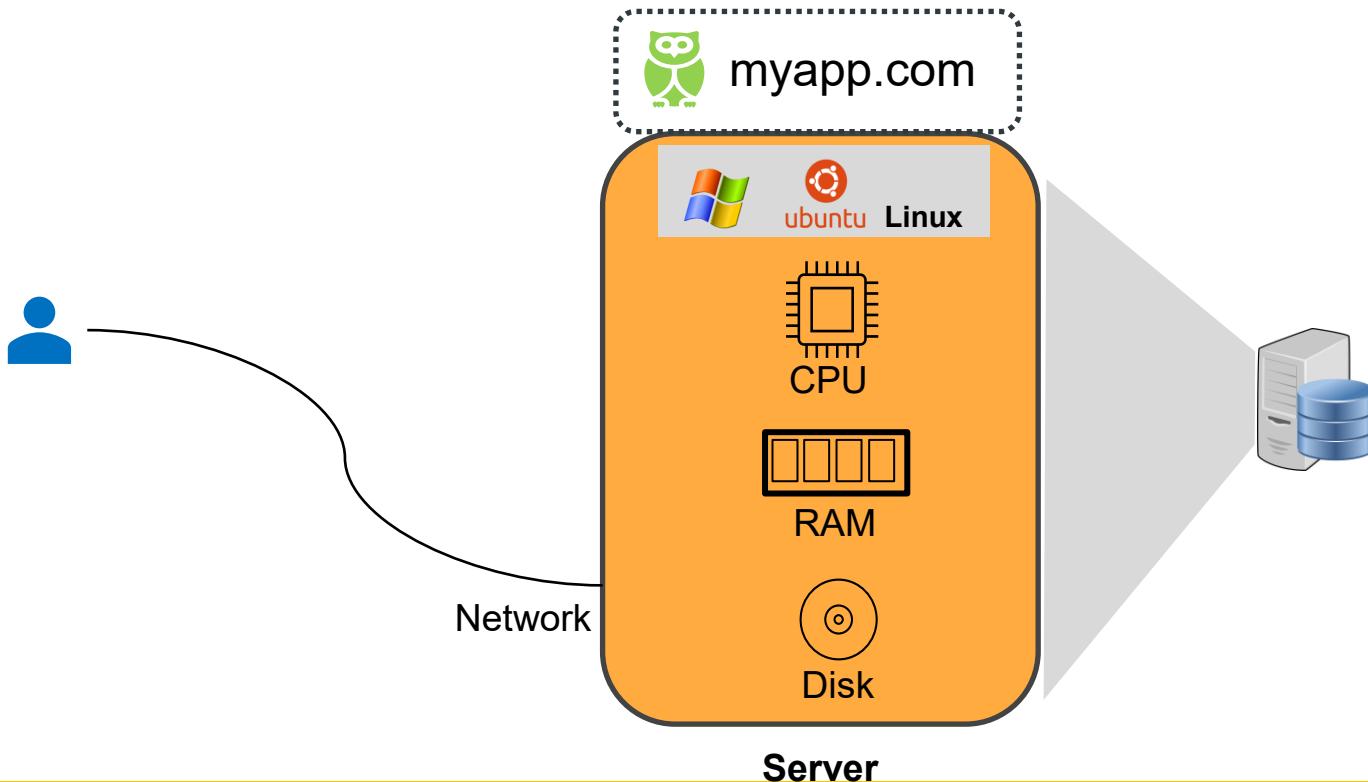


In the later sections

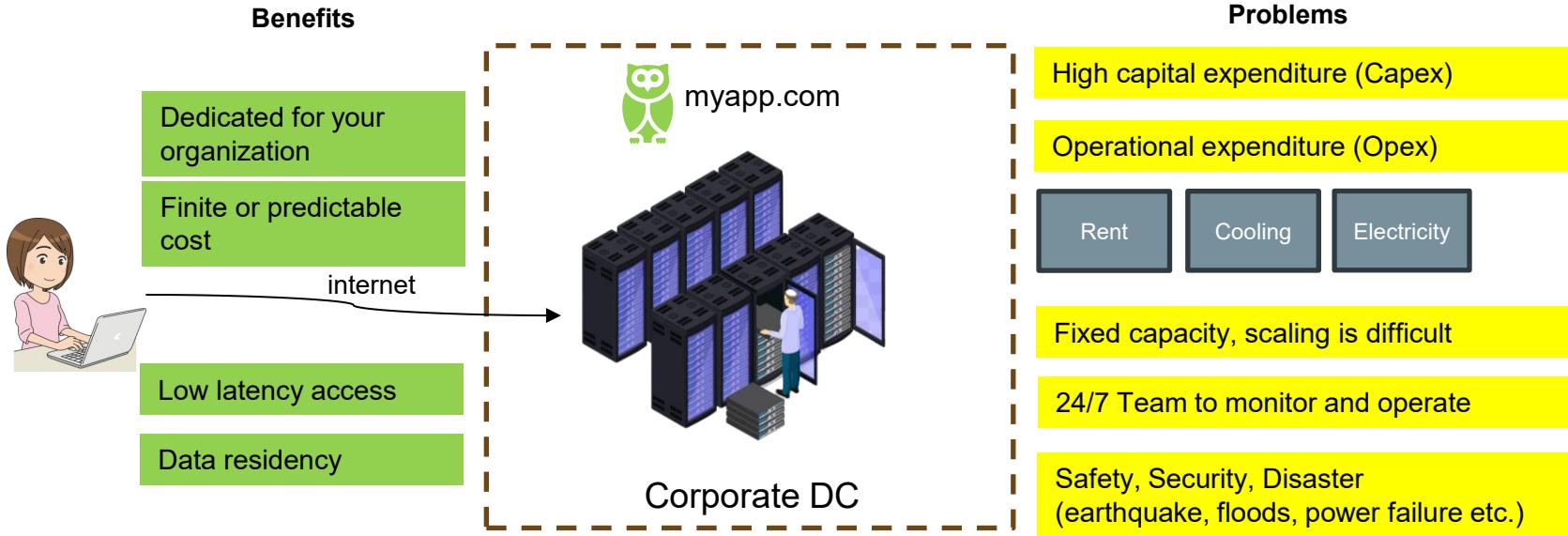
# What is Cloud computing?



# Application hosting and access..



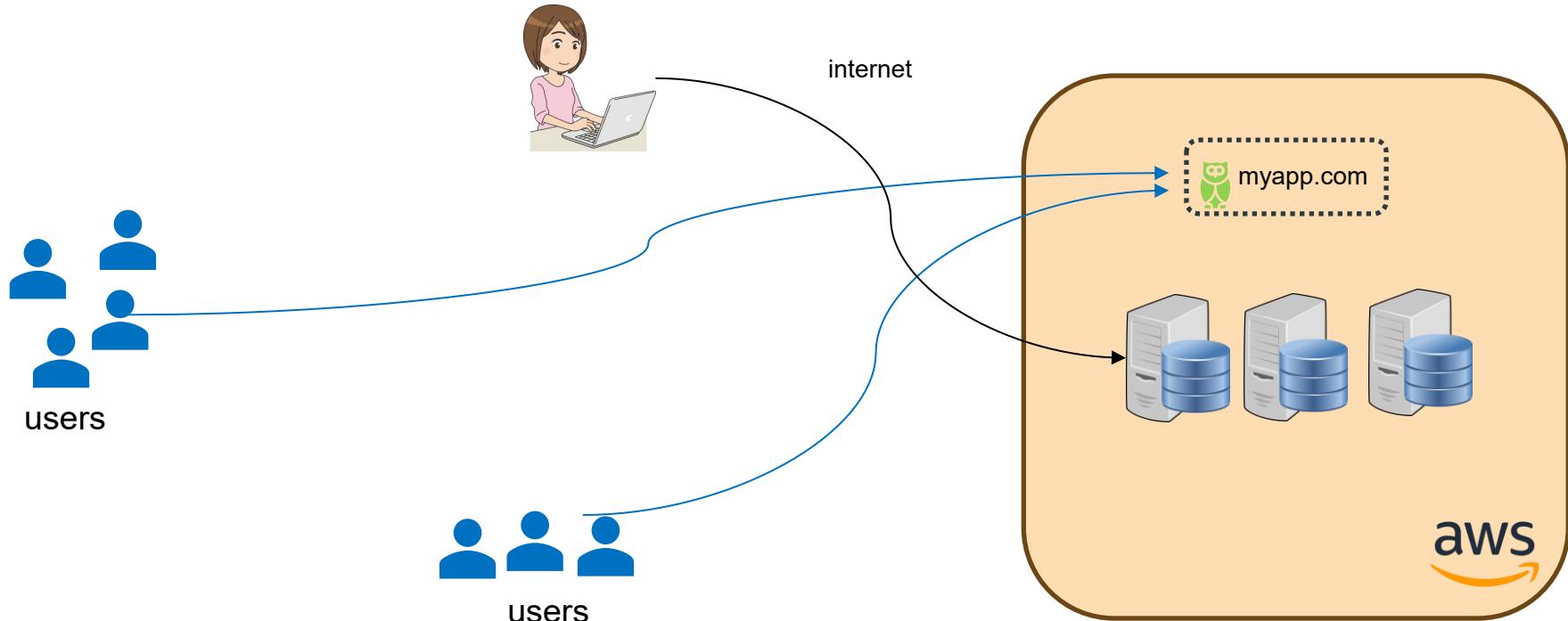
# What is Cloud computing?



# What is Cloud computing?

Cloud computing is the **on-demand** delivery of **IT resources** over the **Internet** with **pay-as-you-go pricing**.

(Compute, Memory, Disk)



# Five characteristics of Cloud computing

## 1. On-demand and Self-service

Users can provision resources and use them without human interaction from the service provider

## 2. Broad Network Access

Resources are accessible over the network thereby supporting heterogeneous client platforms like mobile and workstations

## 3. Multi-tenancy and resource Pooling

Service multiple customers from the same physical resources by securely separating the resources on logical level.

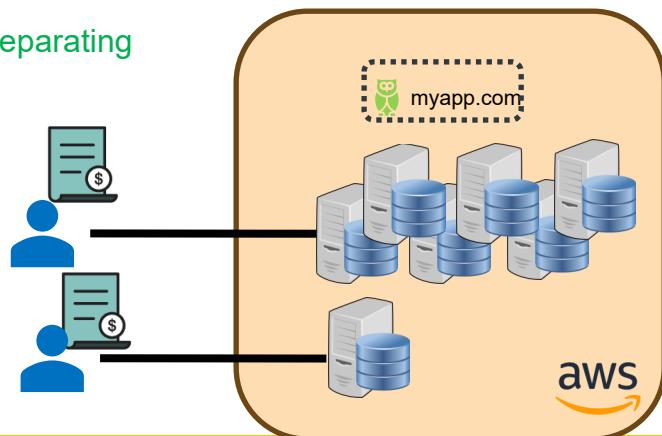
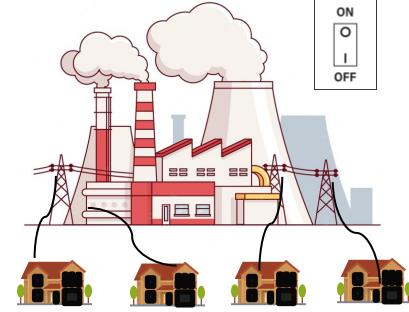
## 4. Rapid elasticity and scalability

Scale on demand

Automatically acquire and dispose resource when needed.

## 5. Measured services

Usage is monitored, measured and billed transparently based on utilization.



# Six advantages of Cloud Computing

## 1. Pay as you go

Pay only when you consume computing resources and how much you consume

## 2. Economy of scale

AWS is more efficient with respect to cost, power, environment due to it's large scale

## 3. Stop guessing capacity

Access as much or as little capacity as you need, and scale up and down as required

## 4. Increase speed and agility

Resources are only a few click away so it reduces the time to make the resources available from weeks to just minutes.

## 5. Stop spending money running and maintaining data centers

Reduced Total Cost of Ownership (TCO), Infrastructure refresh & Operational Expense (OPEX)

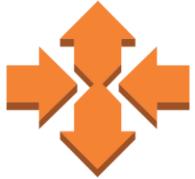
## 6. Go global in minutes

Choose from any of the available AWS locations across the globe – Lower latency, Data residency etc.

# Problems which are difficult to solve without Cloud

## Scalability

Add or remove infrastructure capacity as needed



## High Availability & Fault tolerance

Leverage multiple servers and cloud locations to build HA & fault tolerant applications



## Elasticity

Automatically or dynamically provision and deprovision resources based on demand



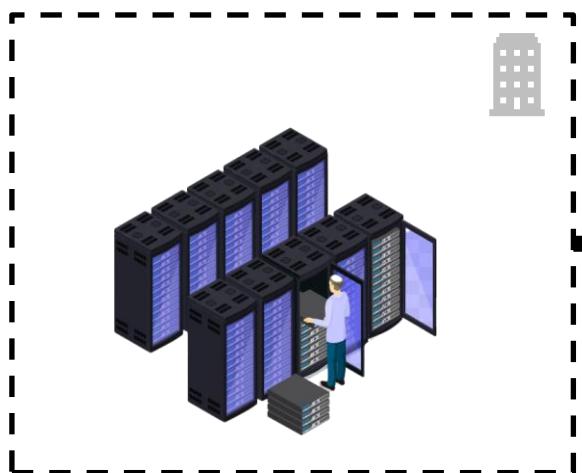
## Flexibility

Flexibility to choose and change resource location and resource types

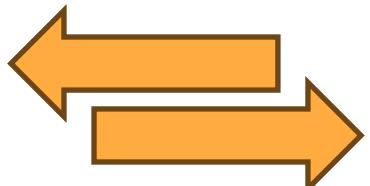


# Deployment models of Cloud

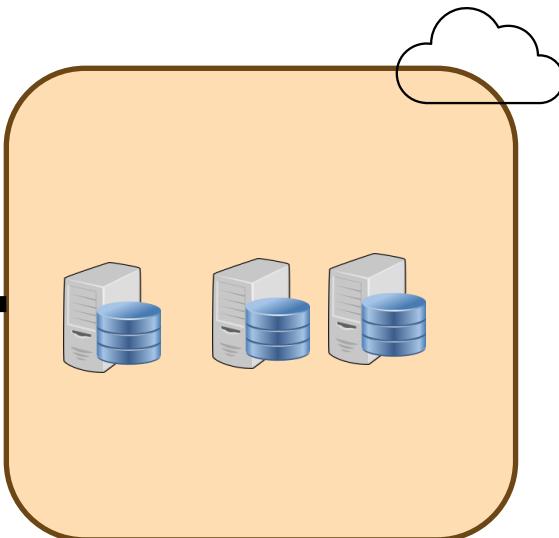
**Private Cloud**



**Hybrid Cloud**



**Public Cloud**



openstack

 **rackspace**  
the open cloud company

 **vmware**



# Deployment Models of Cloud

## Private Cloud

- Cloud services used by a single organization, not exposed to the public.
- Full control
- Low network latency
- Security measures as required by the business
- Meets specific business needs

## Hybrid Cloud

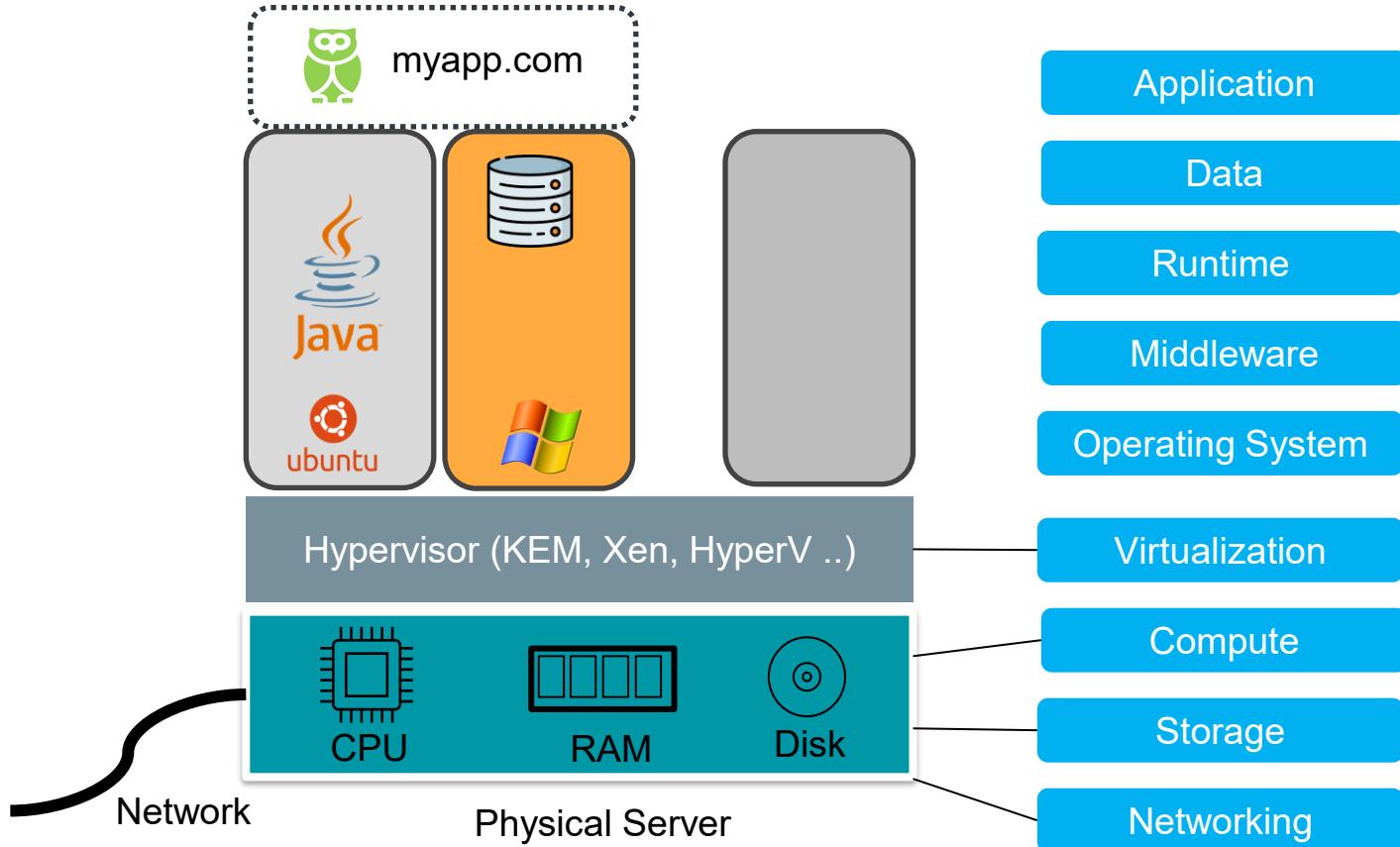
- Keep some servers on premises and extend some capabilities to the Cloud
- Flexibility and cost effectiveness of the public cloud and control of Private cloud.

## Public Cloud

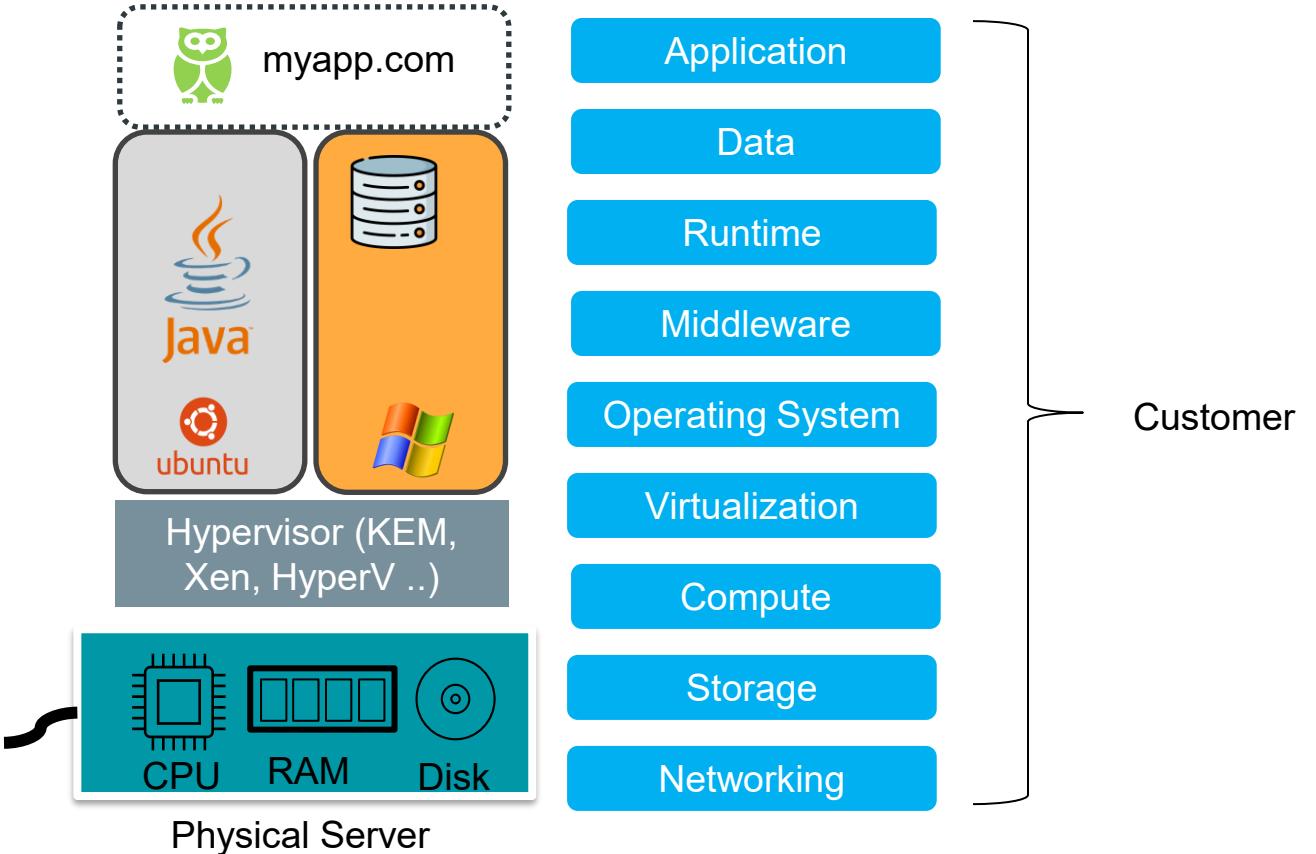
- Cloud resources owned and operated by a third-party cloud service provider delivered over the Internet.



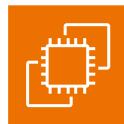
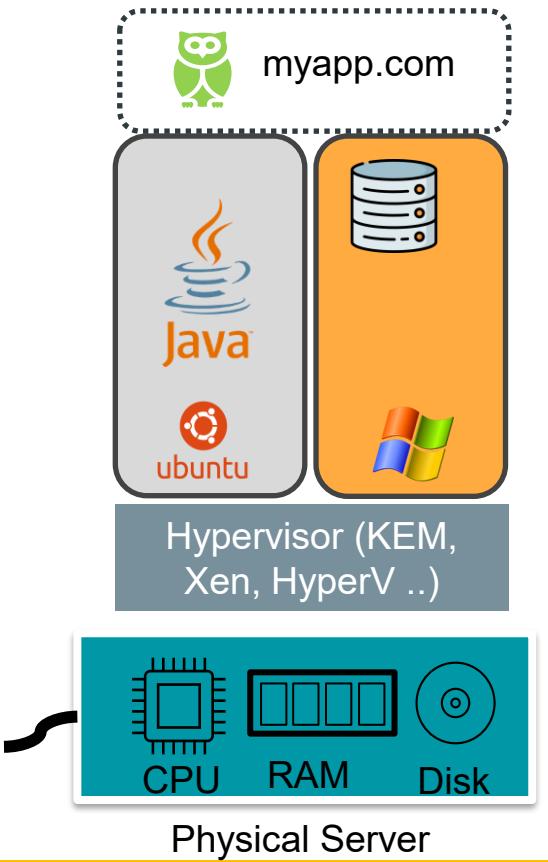
# Infrastructure & software stack



## On-premises Data Center

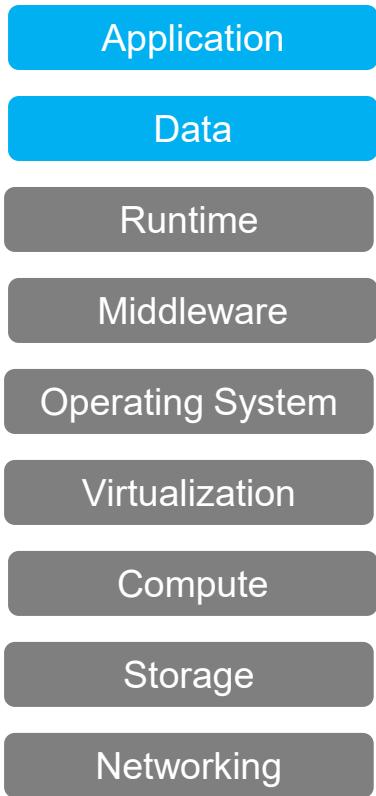
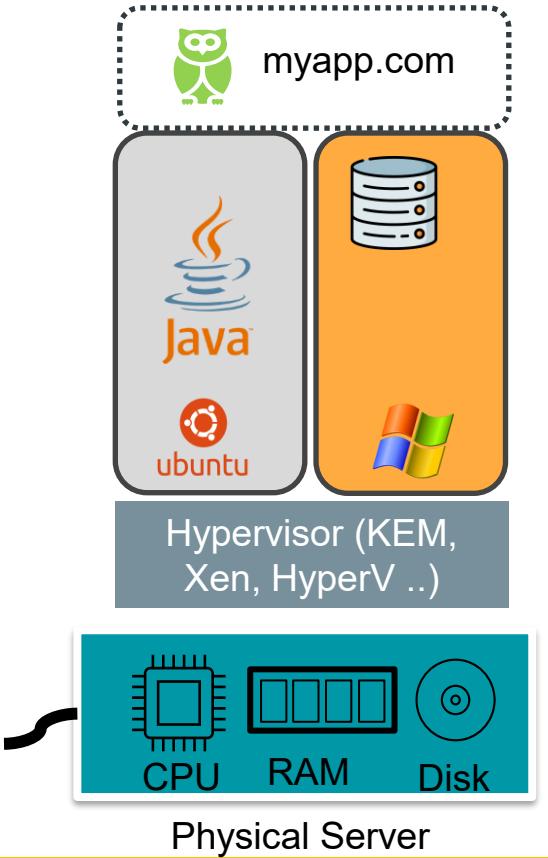


## Infrastructure as a Service (IaaS)



Elastic Compute Cloud (EC2)  
A virtual machine in cloud

## Platform as a Service (PaaS)



Elastic Beanstalk

Deploy & manage web applications

## Software as a Service (SaaS)

access



Application

Data

Runtime

Middleware

Operating System

Virtualization

Compute

Storage

Networking

Cloud Provider





## On-premises Data Center

Application
Data
Runtime
Middleware
Operating System
Virtualization
Compute
Storage
Networking

## Infrastructure as a Service **IaaS**

Application
Data
Runtime
Middleware
Operating System
Virtualization
Compute
Storage
Networking

## Platform as a Service **PaaS**

Application
Data
Runtime
Middleware
Operating System
Virtualization
Compute
Storage
Networking

## Software as a Service **SaaS**

Application
Data
Runtime
Middleware
Operating System
Virtualization
Compute
Storage
Networking



# Cloud Concepts section - summary

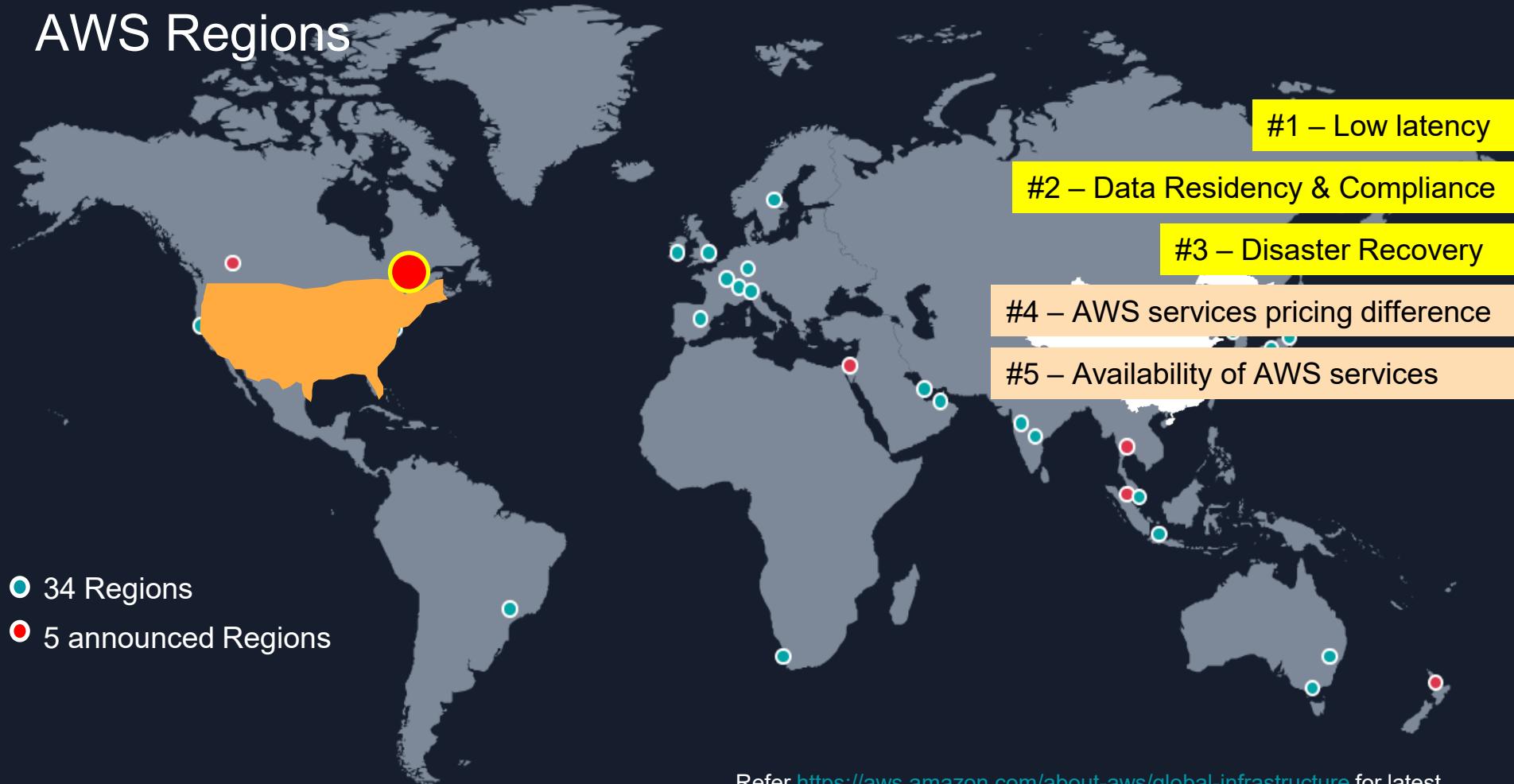
- Cloud computing is the on-demand delivery of IT resources over the internet with pay-as-you-go pricing
- There are five characteristics of the cloud – On-demand & self-service, Broad network access, Multi-tenancy & resource pooling, Rapid elasticity & scalability and Measured services
- There are six advantages of the cloud – Pay-as-you-go, Economy of scale, Stop guessing capacity, Increased speed & agility, Stop paying for running and maintaining data centers and Go global in minutes
- Problems which are difficult to solve without the cloud – Scalability, Elasticity, High Availability & Fault tolerance and Flexibility
- Deployment models of the cloud – Private Cloud, Public Cloud and Hybrid Cloud
- Types of Cloud computing – On-premises, Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS)

# AWS Global Infrastructure

# AWS Global Infrastructure

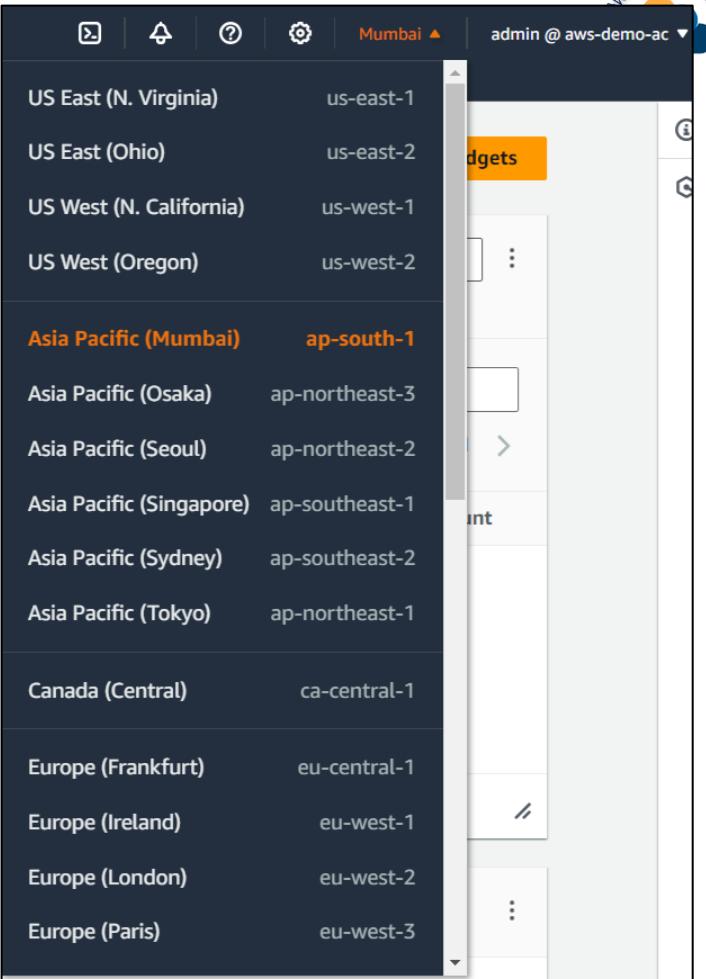
1. AWS Region
2. AWS Availability Zones
3. AWS Edge Locations (CloudFront)
4. AWS Local Zones
5. AWS Wavelength
6. AWS Outpost

# AWS Regions



# AWS Region naming

- Every AWS Region has a name and a code
  - N. Virginia = us-east-1
  - Mumbai = ap-south-1
  - London = eu-west-2



The screenshot shows the AWS CloudFormation console with the 'Regions' tab selected. The top navigation bar includes icons for back, forward, search, and settings, along with the text 'Mumbai' and 'admin @ aws-demo-ac'. The main area displays a table of regions with their names and codes. The 'Asia Pacific (Mumbai)' region is highlighted with an orange background.

US East (N. Virginia)	us-east-1
US East (Ohio)	us-east-2
US West (N. California)	us-west-1
US West (Oregon)	us-west-2
Asia Pacific (Mumbai)	ap-south-1
Asia Pacific (Osaka)	ap-northeast-3
Asia Pacific (Seoul)	ap-northeast-2
Asia Pacific (Singapore)	ap-southeast-1
Asia Pacific (Sydney)	ap-southeast-2
Asia Pacific (Tokyo)	ap-northeast-1
Canada (Central)	ca-central-1
Europe (Frankfurt)	eu-central-1
Europe (Ireland)	eu-west-1
Europe (London)	eu-west-2
Europe (Paris)	eu-west-3

# AWS Availability Zone

- There are minimum 3 AZs in each region
- Some AWS Regions have 6 AZs (N. Virginia)
- Availability Zone names
  - In N. Virginia Region:
    - AZ1 = us-east-1a
    - AZ2 = us-east-1b
    - AZ3 = us-east-1c
    - ...
  - In Mumbai Region:
    - AZ1 = ap-south-1a
    - AZ2 = ap-south-1b
    - AZ3 = ap-south-1c



# AWS Edge locations



\*Diagram just for illustration, not actual

# Without AWS edge network

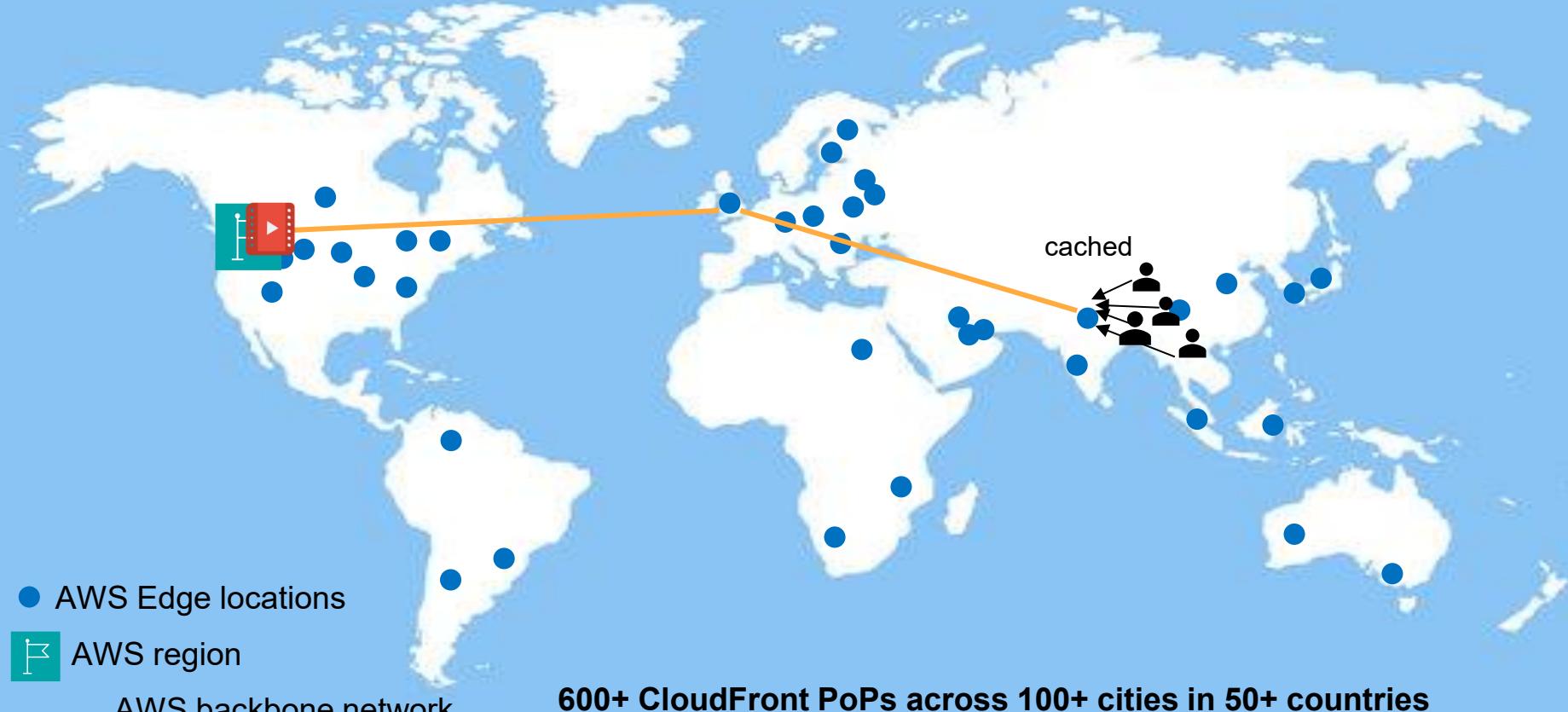


\*Diagram just for illustration, not actual

# With AWS edge network

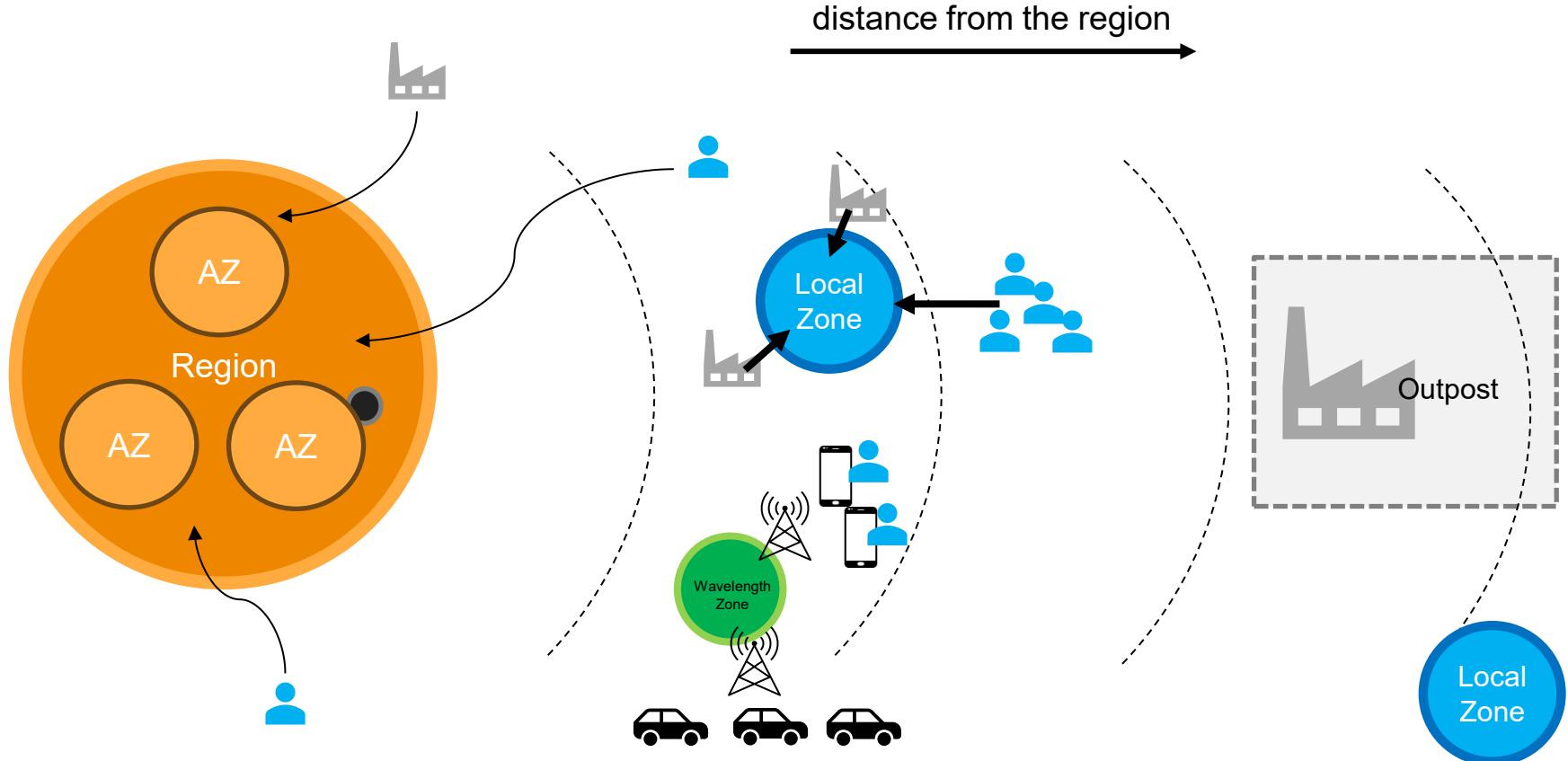


# Amazon CloudFront

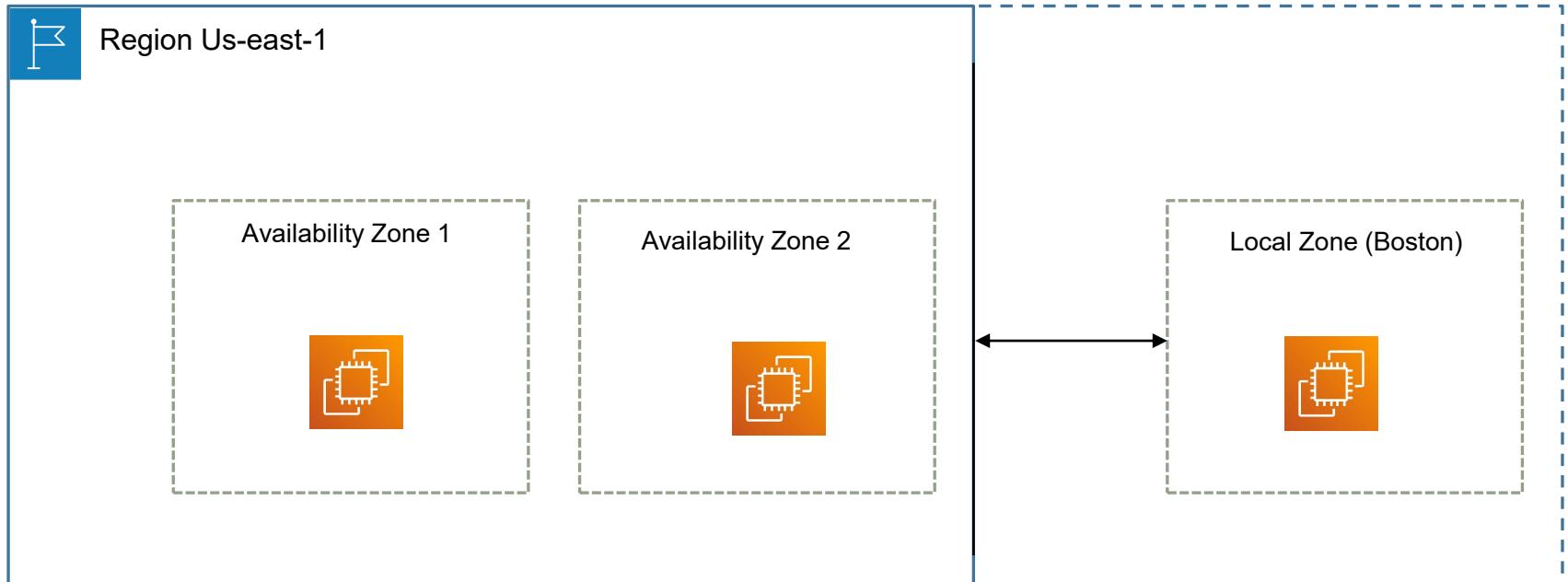


\*Diagram just for illustration, not actual

# Beyond the AWS Region..



# AWS Local Zones





# AWS Local Zones

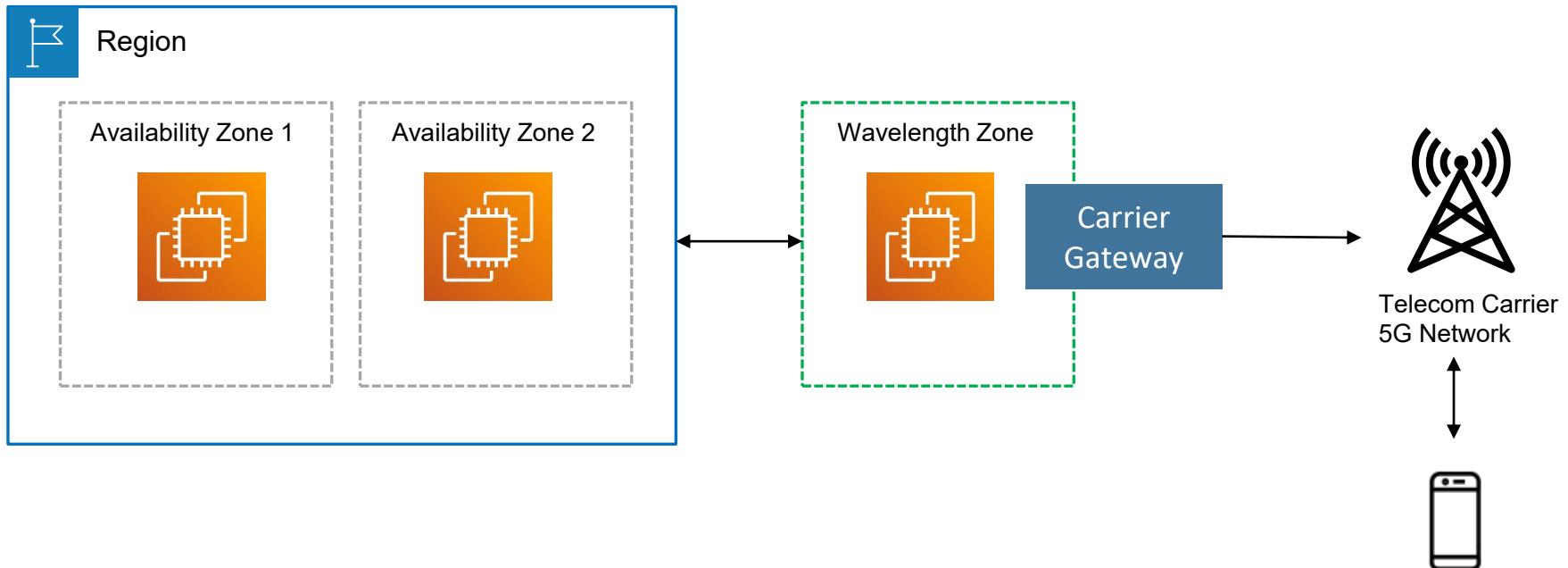
## AWS Local zone:

- Places compute, storage, database, and other select AWS resources close to end users to run latency-sensitive applications - “Extension of an AWS Region”.
- Can use subset of AWS services such as EC2, RDS, ECS, EBS etc.

## Use cases:

- Run low-latency applications at the edge close to end users – real-time gaming, live streaming, augmented and virtual reality (AR/VR), virtual workstations, and more.
- Meet stringent data residency requirements - Comply with state and local data residency requirements.

# AWS Wavelength





# AWS Wavelength

## AWS Wavelength:

AWS Wavelength are infrastructure deployments embedded within the telecommunications providers datacenters at the edge of the **5G networks**.

## Benefits:

- Ultra-low latency applications through 5G networks.
- Traffic doesn't leave the Communication Service Provider's (CSP) network.
- High-bandwidth and secure connection to the parent AWS Region.

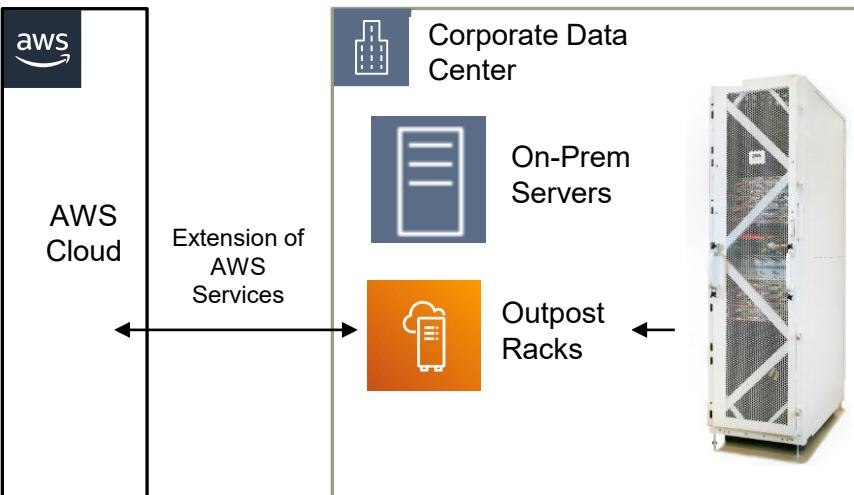
**Use Cases:** Smart Cities, ML-assisted diagnostics, Connected Vehicles, Interactive Live Video Streams, AR/VR, Real-time Gaming etc.

# AWS Outpost



AWS Outpost is a **fully managed service** that extends AWS infrastructure, services, APIs, and tools to **customer premises** just as in the cloud.

- AWS will setup and manage “**Outposts Racks**” within your on-premises infrastructure and you can start leveraging AWS services on-premises.
- You are responsible for the Outposts Rack physical security.



# AWS Outpost



## Benefits:

- Low-latency access to on-premises systems, Local data processing, Data residency

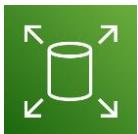
## Use cases:

- Manufacturing execution systems (MES), high-frequency trading, or medical diagnostics.

Some of the AWS services that can run on Outposts:



Amazon EC2



Amazon EBS



Amazon S3



Amazon EKS



Amazon ECS



Amazon RDS

# AWS Global Infrastructure – Summary

## Amazon Region & AZs:

- Amazon Regions consist of multiple Availability Zones (AZs) located in distinct geographic areas, offering redundancy and isolation from failures in other regions.
- Choose region based on end user latency, data residency, DR requirements, AWS Services availability & pricing

## AWS Edge Locations / CloudFront:

- Enables low-latency connectivity to AWS region through AWS backbone network.
- Can cache static content (e.g. video, images)
- **Can not** be used for hosting applications unlike Region, LZ, WLZ or Outpost.

# AWS Global Infrastructure – Summary

## AWS Local Zone:

- AWS Local Zones extend AWS infrastructure to specific geographic locations, enabling low-latency access to AWS services for applications that require single-digit millisecond latency.
- Choose AWS Local Zones when you need to place resources closer to end-users in **metropolitan** areas meeting low latency and compliance requirements.

## AWS Wavelength:

- Extends AWS infrastructure to **5G mobile network** (Limited AWS services)
- Choose AWS Wavelength when you want to deploy ultra-low latency applications at the edge of 5G networks (e.g. Gaming, Connected vehicles)

## AWS Outpost:

- Run AWS services **on-premises** (Limited AWS services)
- Use AWS Outposts when you need to run AWS infrastructure on-premises or in co-location facilities for low latency and local data processing.

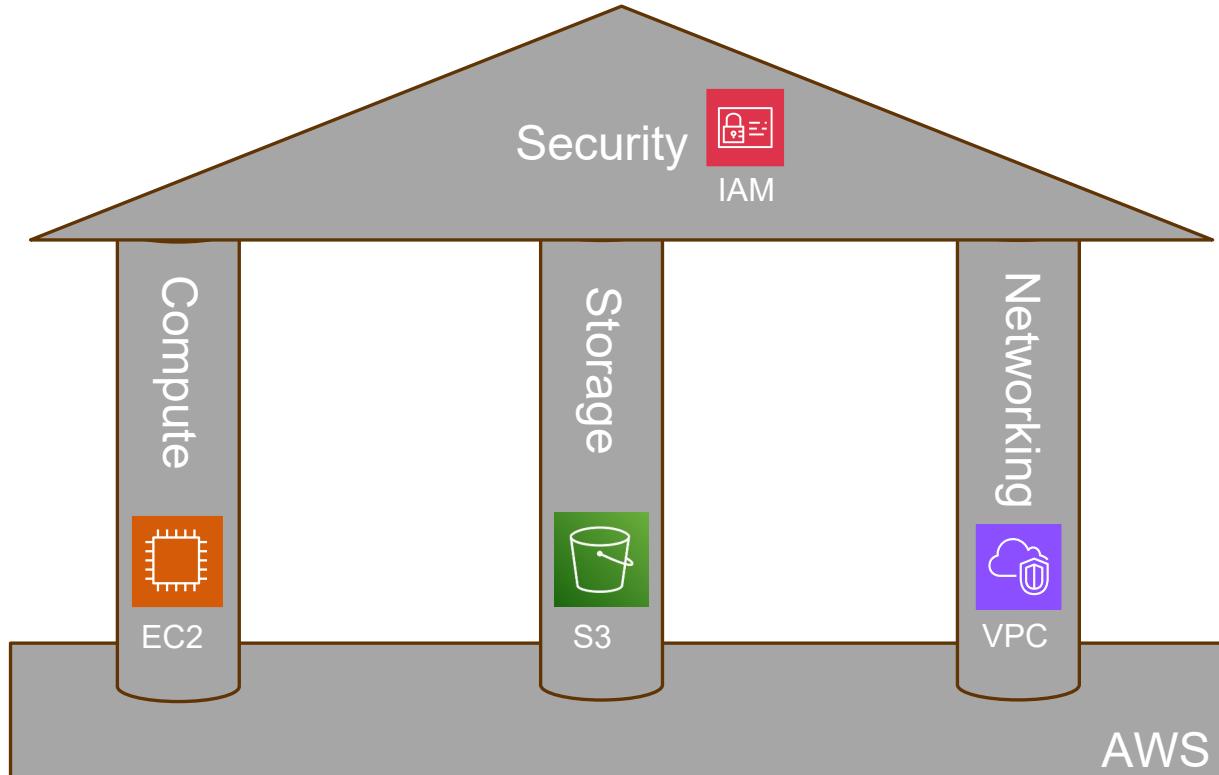


# AWS IAM

## Identity & Access Management

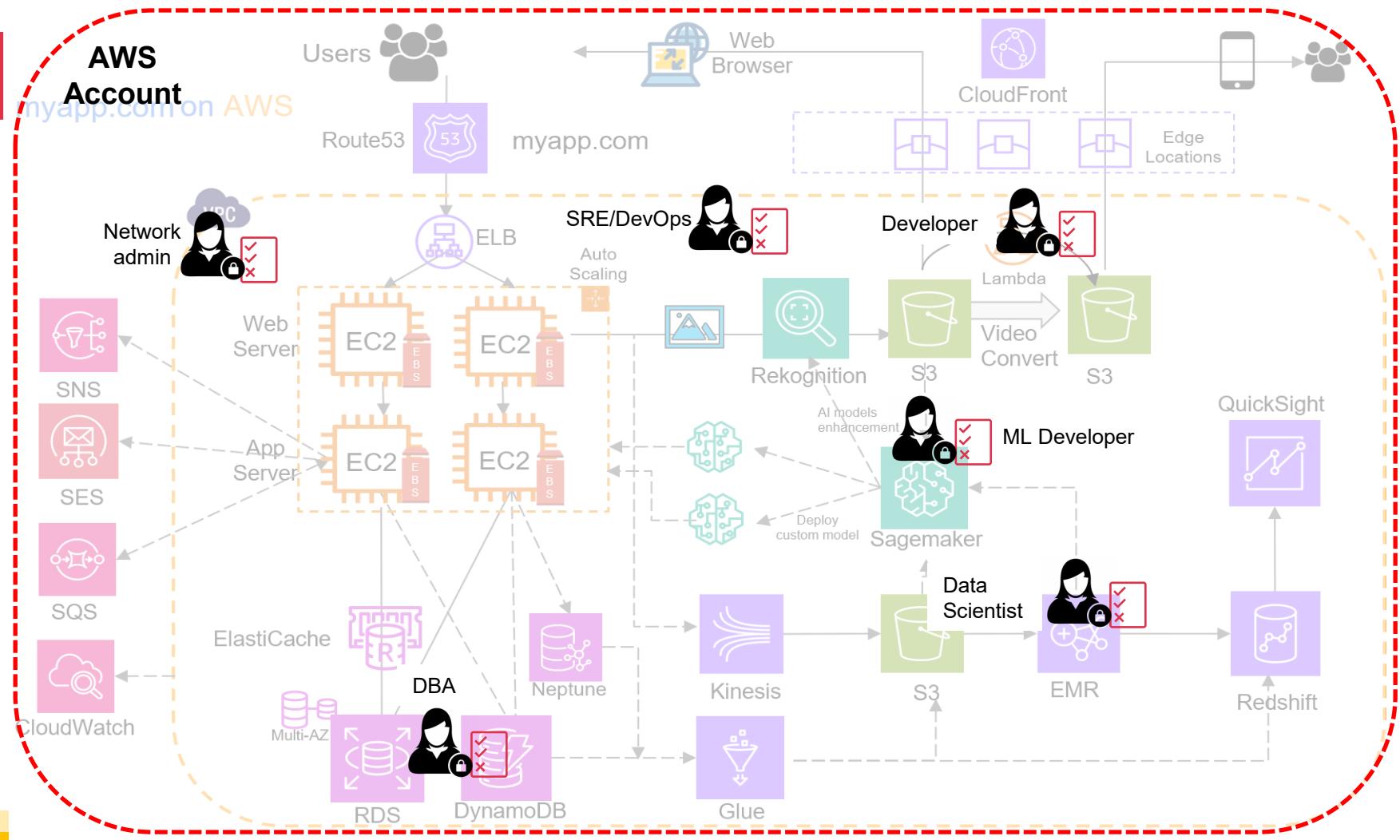
# AWS core services

*in my opinion..*

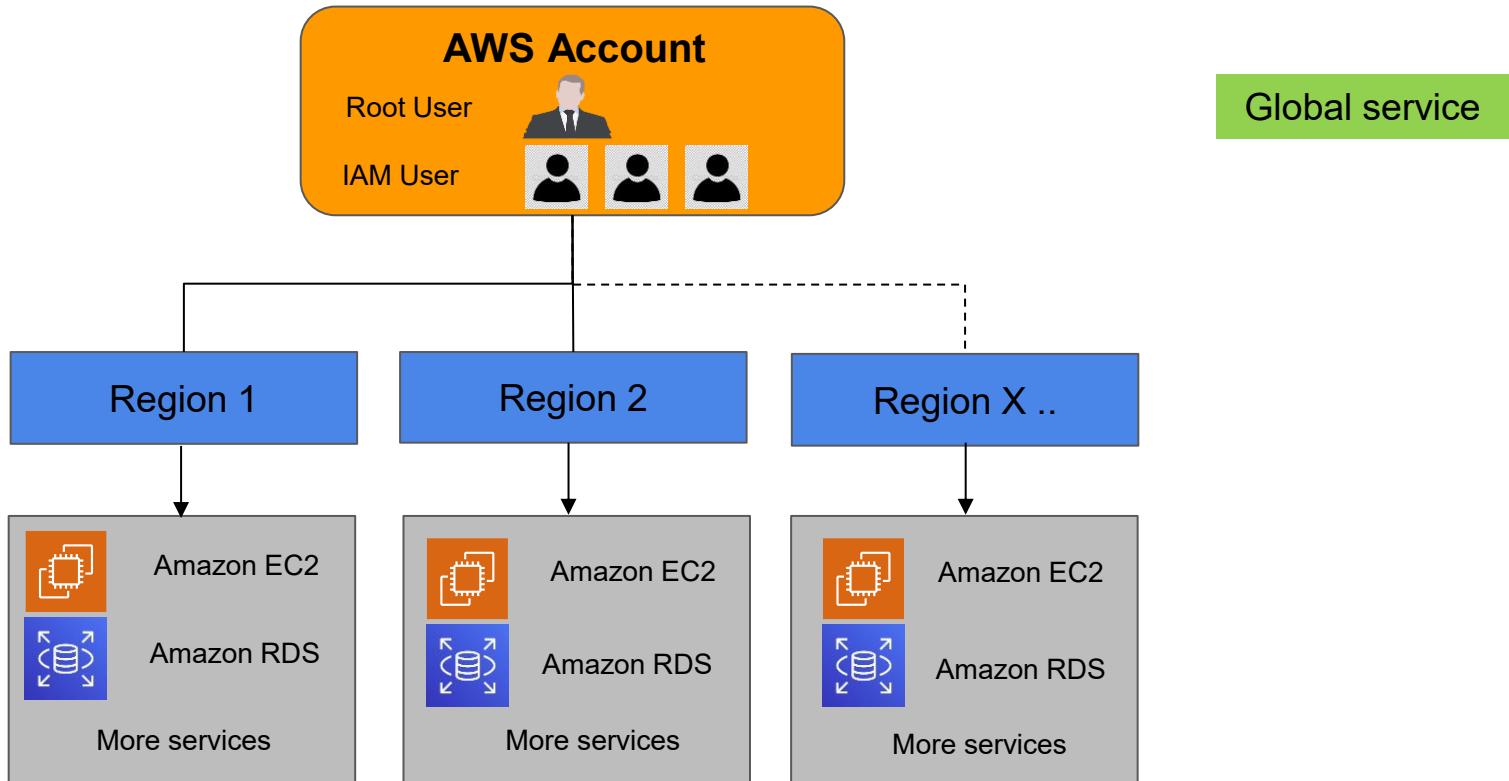




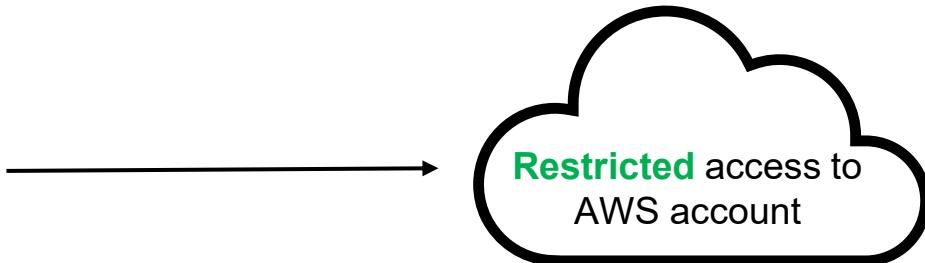
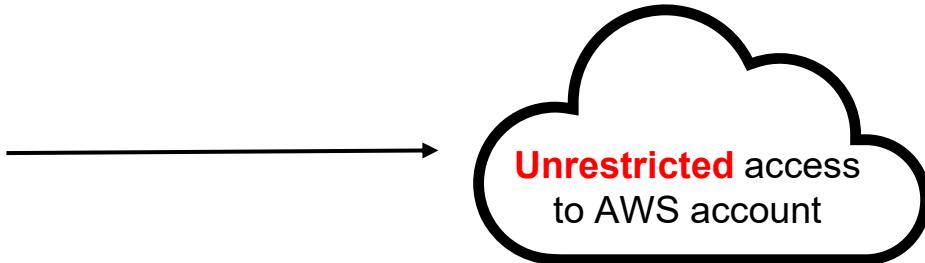
IAM



# AWS account and users



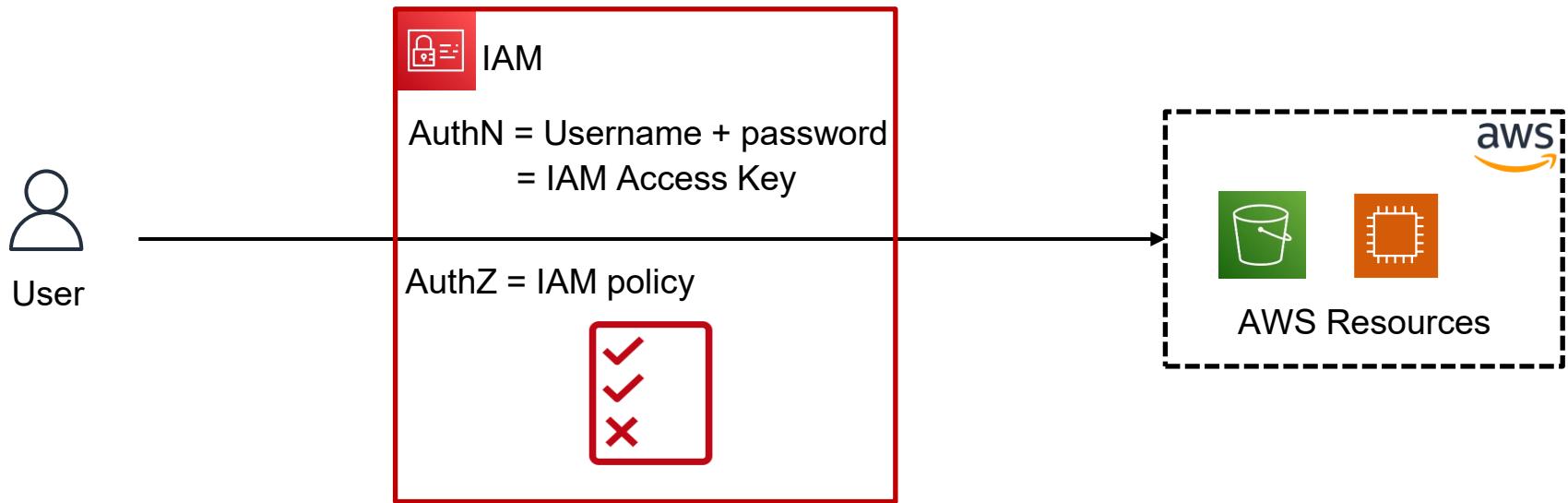
# AWS account user types



# User Authentication & Authorization using IAM

Authentication = Is this the legitimate user?

Authorization = Does user has permissions to access resources?



# AWS account user types



## Root User

- Owner of the AWS account
- Login with email id & password
- IAM policy is not applicable\* and hence no control over the permissions
- Use only when you need to perform special actions (account closure, billing access, payment etc.)
- Not recommended for day-to-day operations



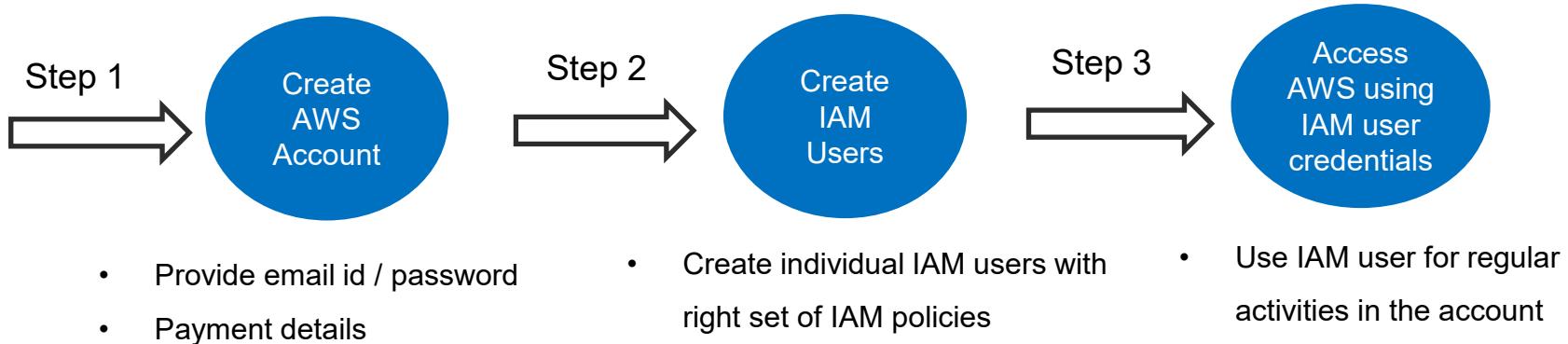
## IAM User

- Individual users having specific permissions
- Login with username & password
- IAM policy is applicable and can grant granular permissions
- Should be used to by an individual person or application with only required accessfor AWS services and resources to perform the job.
- Recommended for day-to-day operations

# Tasks that require Root User Access

- ✓ Change your account Settings
- ✓ Restore IAM User Permission
- ✓ Close your AWS Account
- ✓ Activate IAM access to the Billing and Cost Management Console
- ✓ Configure an Amazon S3 bucket to enable MFA
- ✓ Change AWS support plan

# AWS Account and Users



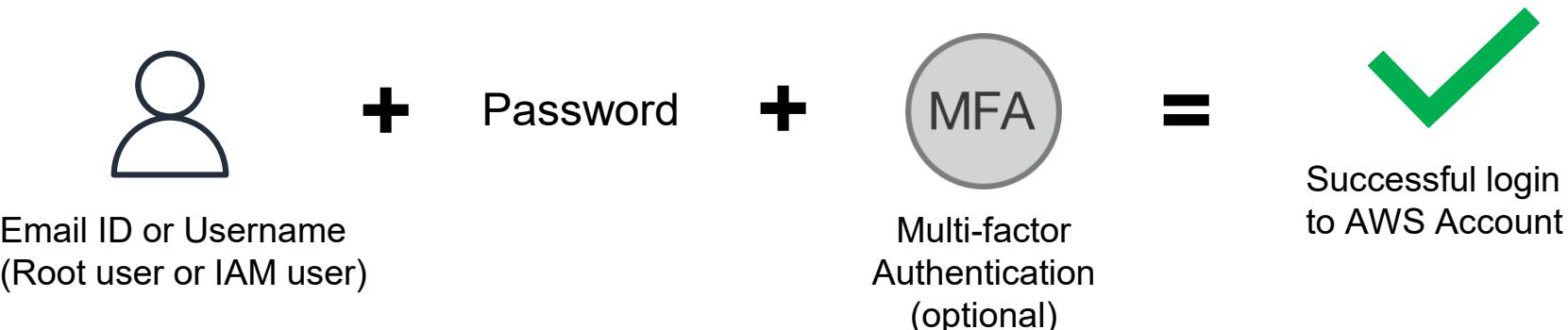
# Exercise : Create an IAM user for yourself

- 1 Login to AWS Management Console using Root user credentials (email/password) and navigate to IAM service
- 2 Go to Users -> Create user -> Provide user name -> Provide user access to AWS Management console -> I want to create IAM user -> Custom password -> Enter the password -> Uncheck users must create a new password at next sign in -> Next
- 3 On the permissions screen -> Attach policies directly -> Select **AdministratorAccess** policy -> Next -> Create user
- 4 Go to IAM Dashboard page and copy the account sign-in URL under AWS account details
- 5 Log out of the current session and open the sign-in URL you have just copied in step 4
- 6 This time provide IAM user name/password to login to AWS
- 7 Once logged in, you can check whether you can perform all AWS actions e.g. creating another IAM user

# Multi-factor authentication - MFA

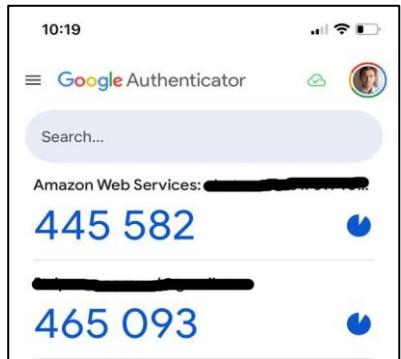
# User Authentication – Are you the right user?

- Primary authentication using Login/password
- Multi-Factor Authentication adds a second layer of security
- **Once enabled**, user will need to provide not only a password to authenticate but also temporary digital token sent through a preset device like a smartphone running an Authenticator app.



# Multi-factor Authentication (MFA)

## Virtual Authentication Apps



Google Authenticator  
Microsoft Authenticator  
More apps..

## FIDO security keys



By third-party providers such as Yubico, acs, Gotrust etc.

## Hardware TOTP tokens



By Thales  
(3rd party provider)



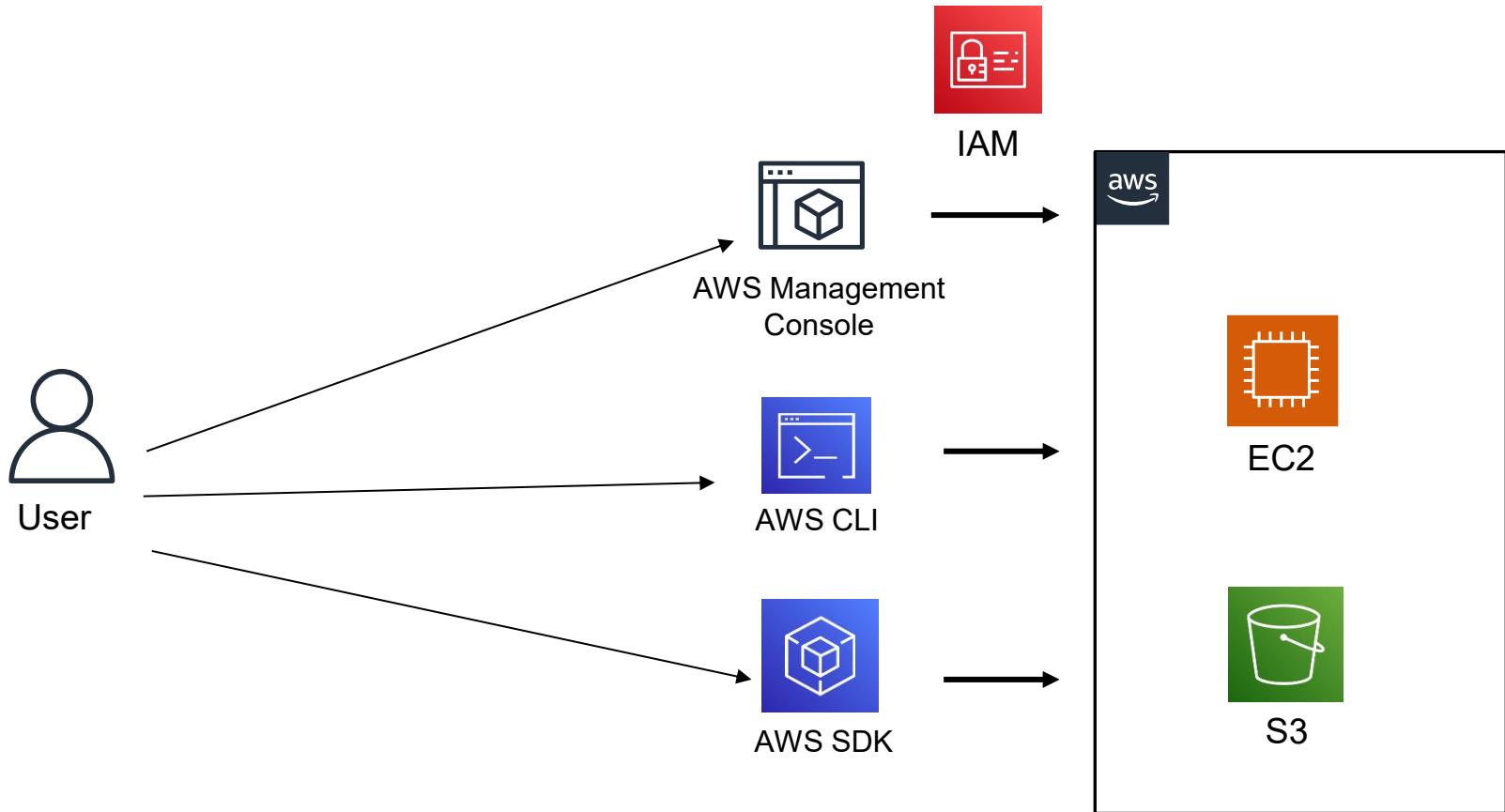
By Hypersecu  
(3rd party provider)

# Exercise : Set up MFA for IAM User

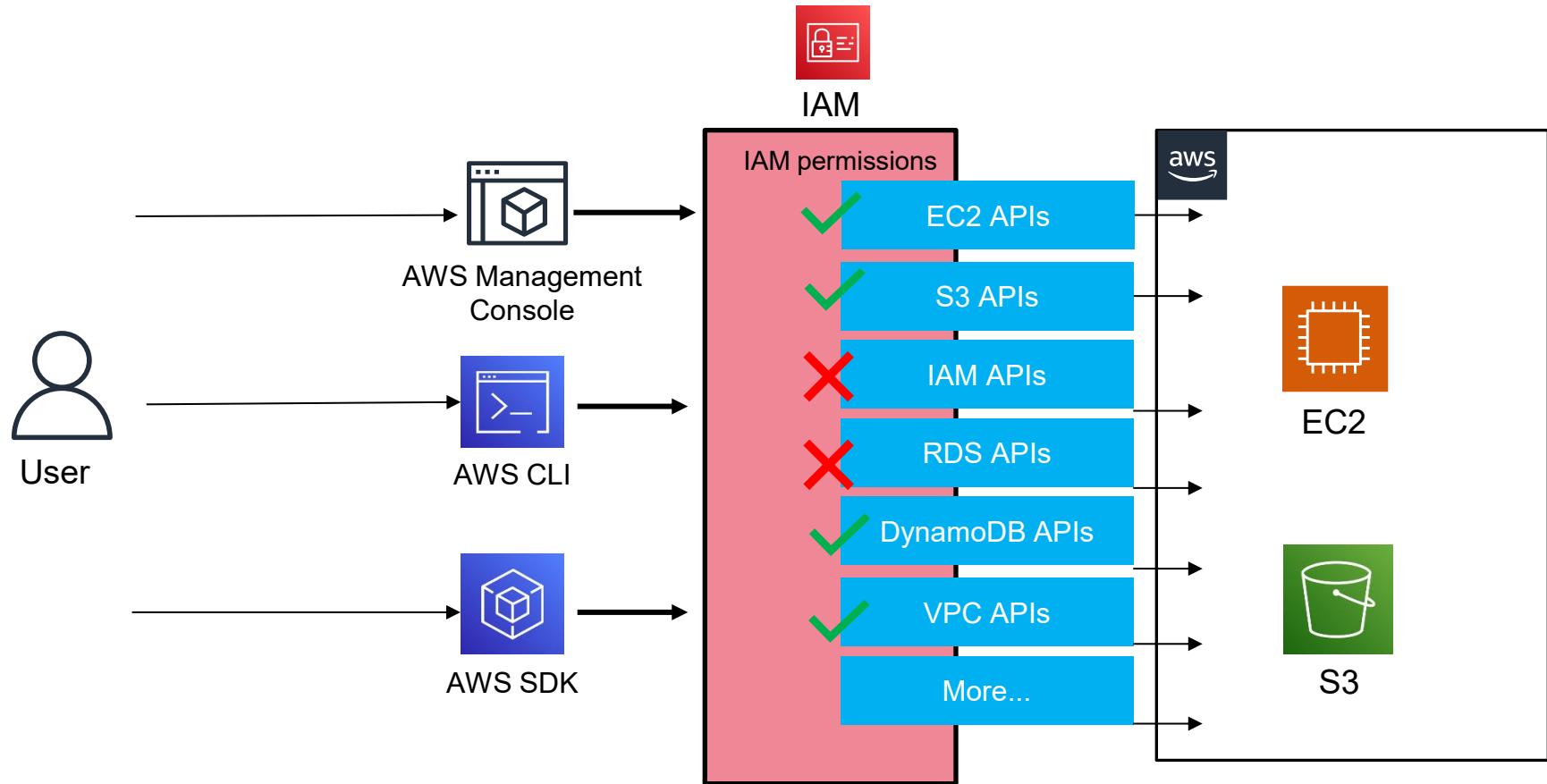
- 1 Download an AWS compatible Authenticator App e.g. Google Authenticator
- 2 Login to AWS Management Console using Root user or admin IAM user and navigate to IAM.
- 3 Click on your IAM User name for which you want to setup MFA
  - Click Users on the left navigation panel and click on the user for which you want to set up MFA
  - Click on Security credentials tab
  - Click on Manage in Assigned MFA device row
  - Select Virtual MFA device and click Continue
  - You will see a dialog window with instruction to setup MFA and a Show QR code button
  - Click on Show QR in the above dialog so that you can scan it using your app
- 4 Open Authenticator App installed in Step 1 and scan QR code.
  - App detects your account.
  - Click on Add ACCOUNT in Authenticator app to add your AWS account in the authenticator app.
  - Enter 2 Consecutive MFA codes from your Authenticator App.
- 5 Verify MFA Setup
  - Log out of your account and try to login again.
  - You will be prompted for an MFA code after you enter your username/password, provide MFA from your authenticator App.

# IAM policy

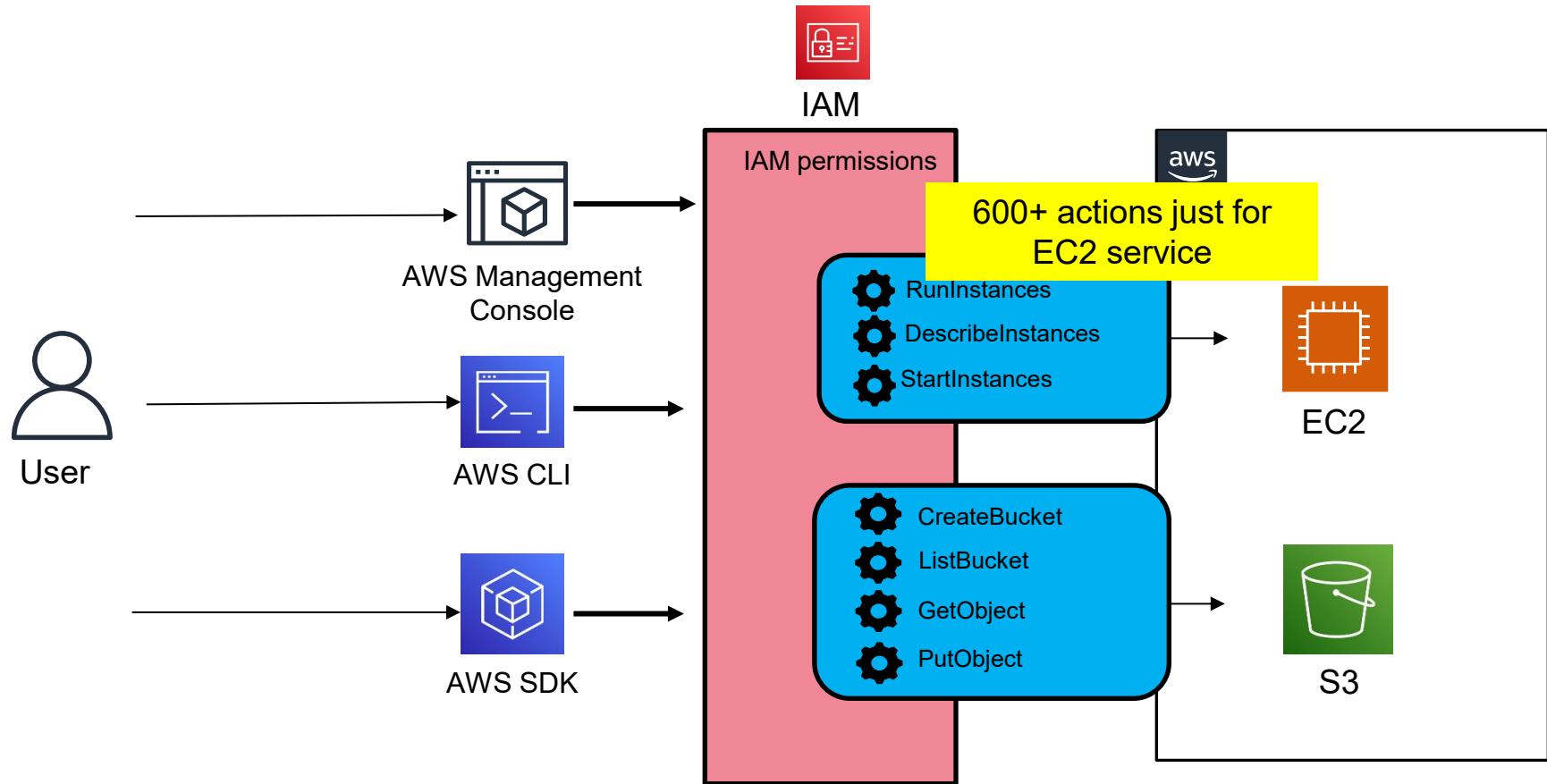
# Accessing AWS..



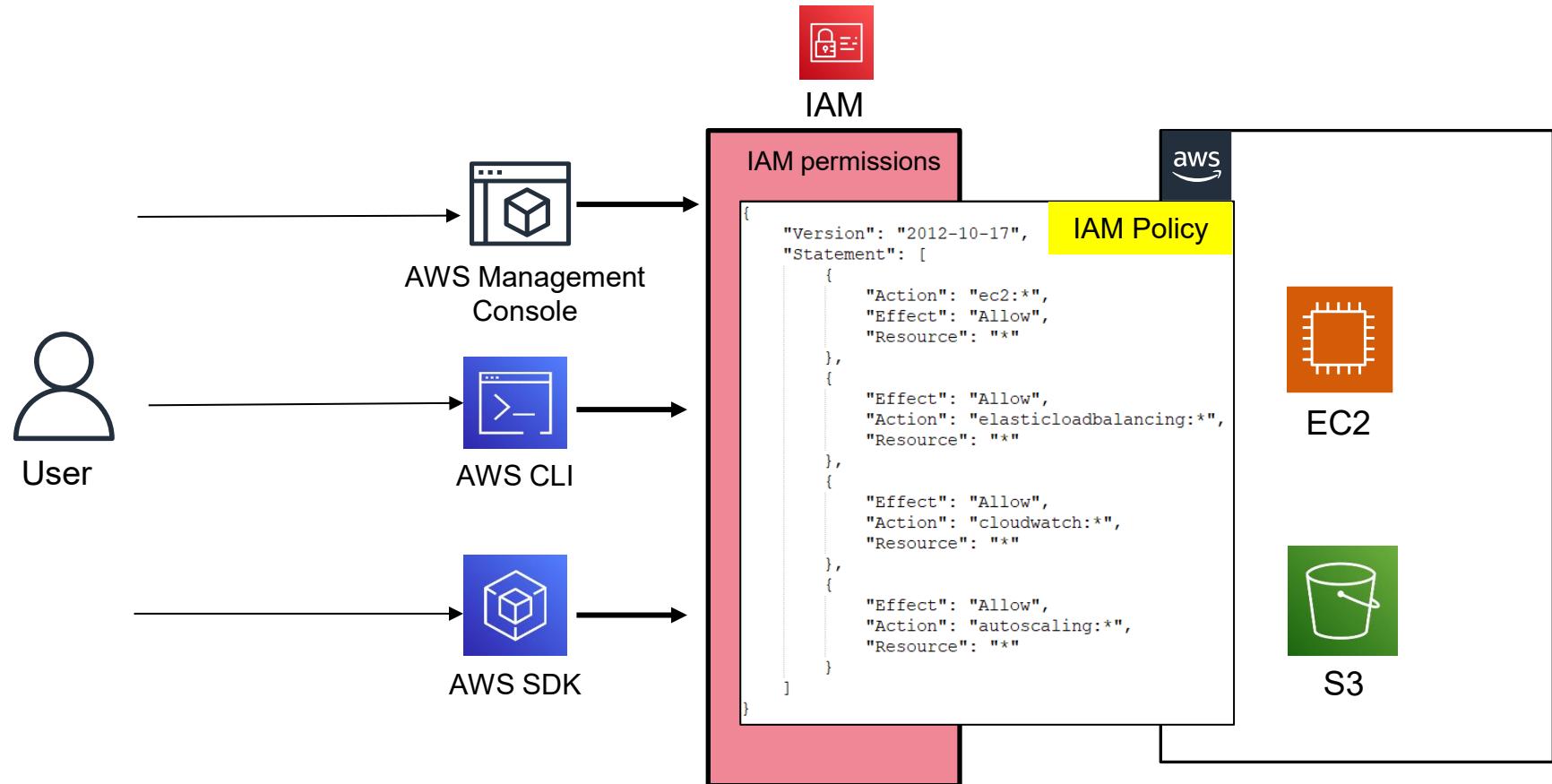
# Accessing AWS..



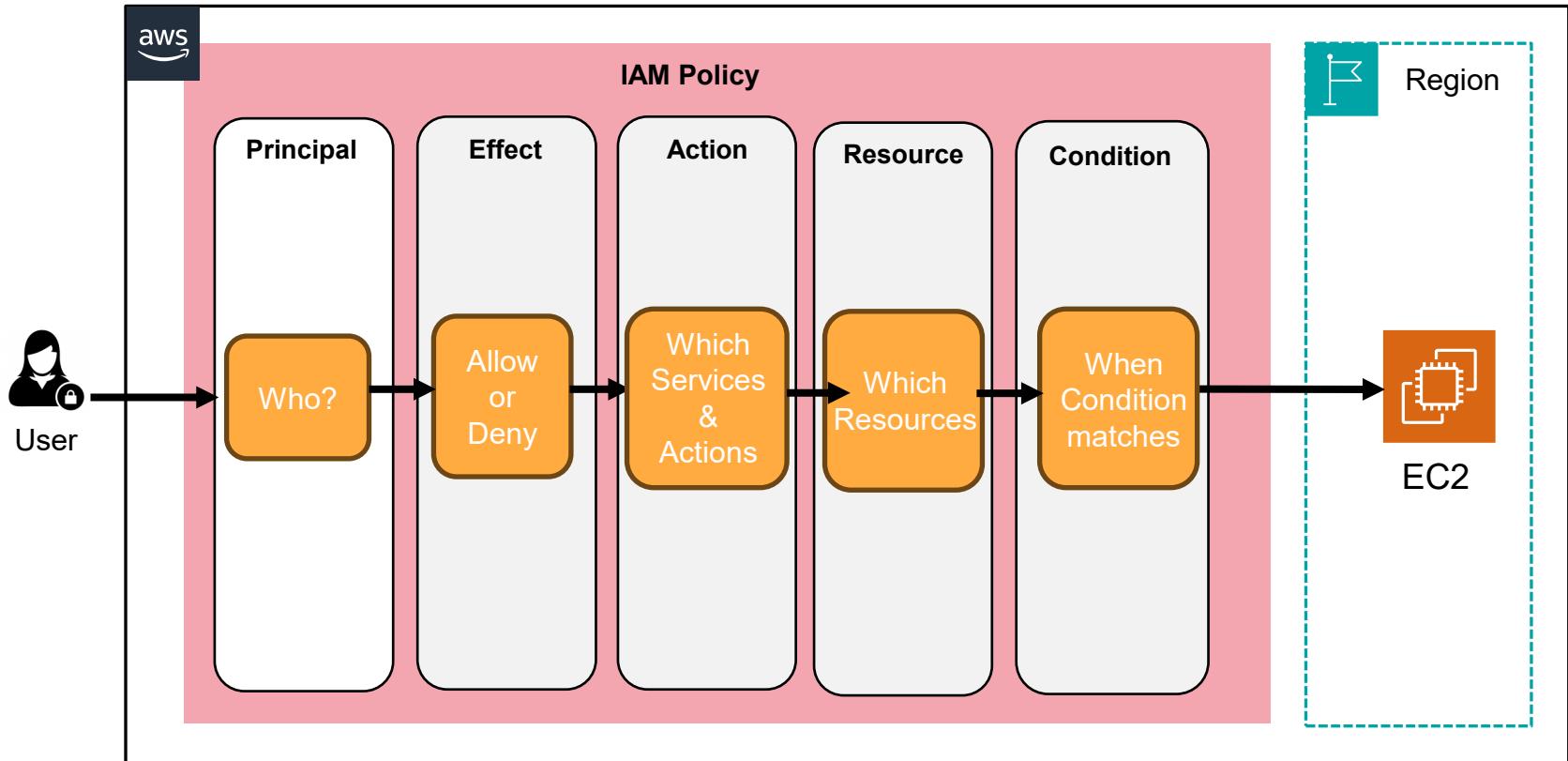
# Accessing AWS..



# IAM Policy



# IAM Policy components



# IAM Policy

- **Policy Version:** 2012-10-17
- **Id:** string-to-identify-policy (*optional*)
- **Statements:** one or more statements
- **Each statement has:**

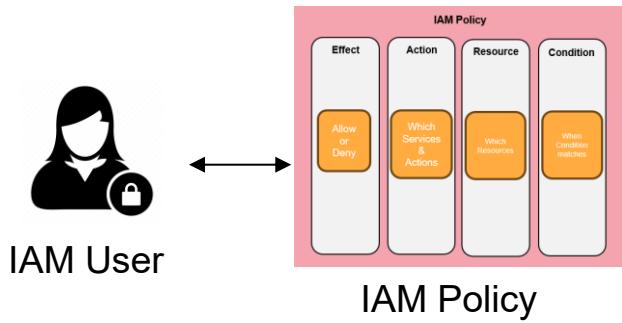
- **Sid:** Identifier for the statement (*optional*)
- **Effect:** Allow or Deny access
- **Principal:** AWS account or IAM user or IAM role to which this policy is applicable (*for resource-based policies*)
- **Action:** List of actions (*use \* for all*)
- **Resource:** Particular resources for which permissions are granted (*use \* for all*)
- **Condition:** Apply policy statement only when condition is valid (*optional*)

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "StartStopIfTags",  
      "Effect": "Allow",  
      "Action": [  
        "ec2:StartInstances",  
        "ec2:StopInstances"  
      ],  
      "Resource": "arn:aws:ec2:region:account-id:instance/*",  
      "Condition": {  
        "StringEquals": {  
          "aws:ResourceTag/Project": "DataAnalytics",  
          "aws:PrincipalTag/Department": "Data"  
        }  
      }  
    }  
  ]  
}
```

Example IAM Policy: Start or stop instances based on tags

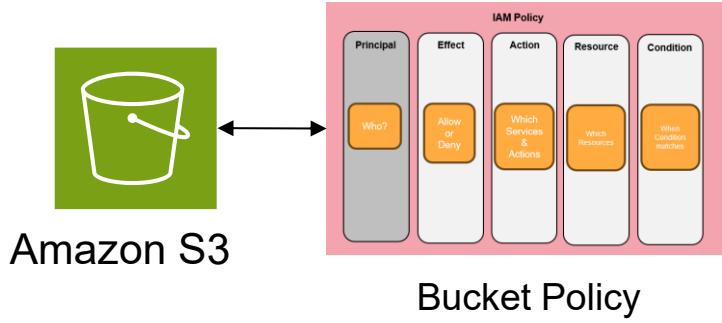
# IAM Policy types

**What permissions user has?**



**Identity-based policy**

**Who has permissions to access this resource?**

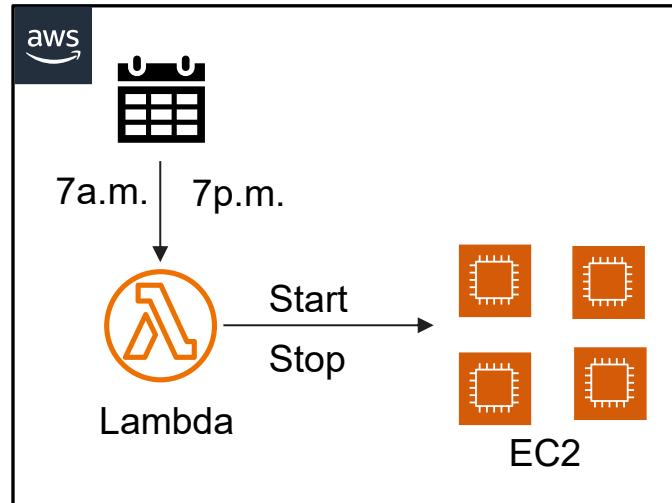


**Resource-based policy**

# Exercise : Create IAM policy

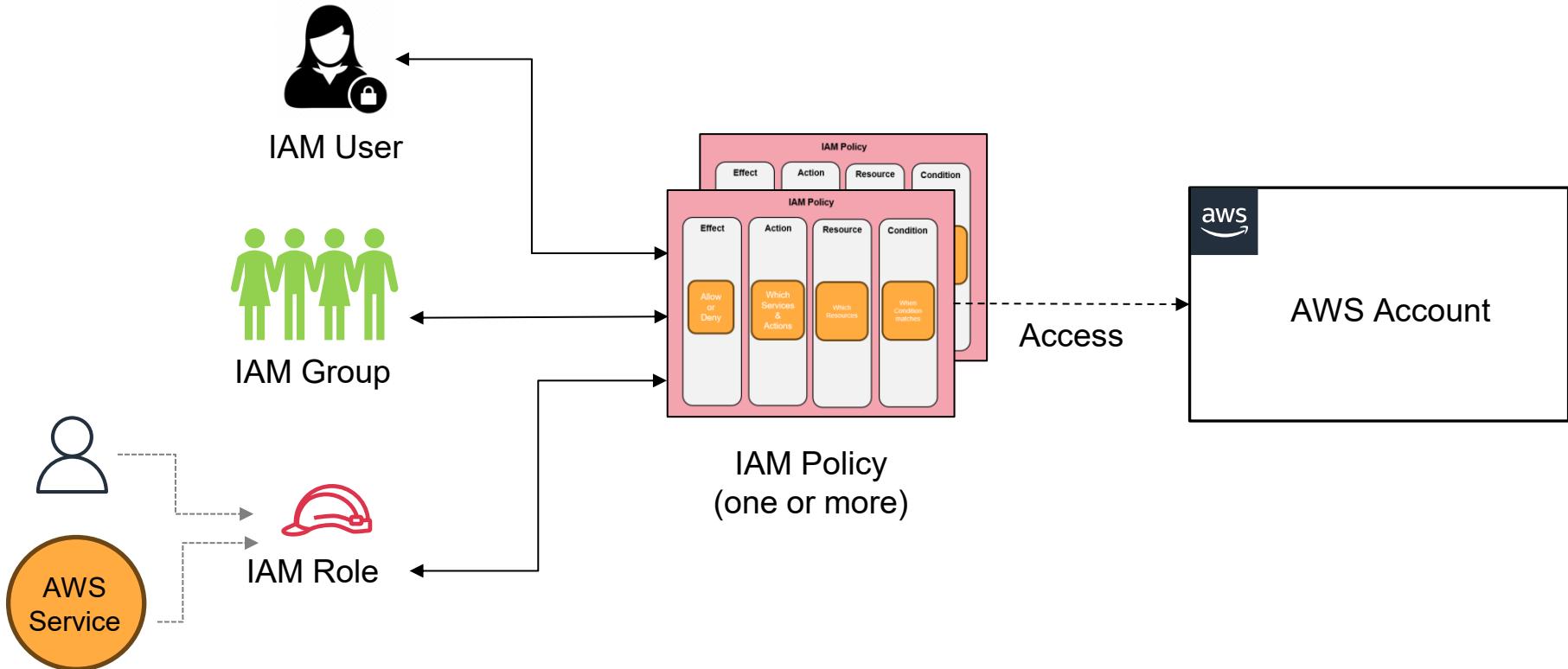
- 1 Create a new IAM policy with the name say “Start-Stop-EC2” having permissions to list, start and stop the EC2 instances

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Effect": "Allow",  
      "Action": [  
        "ec2:DescribeInstances",  
        "ec2:StartInstances",  
        "ec2:StopInstances"  
      ],  
      "Resource": "*"  
    }  
  ]  
}
```



- 2 Create a new IAM user with the name ‘automation’ and attach this policy to the user

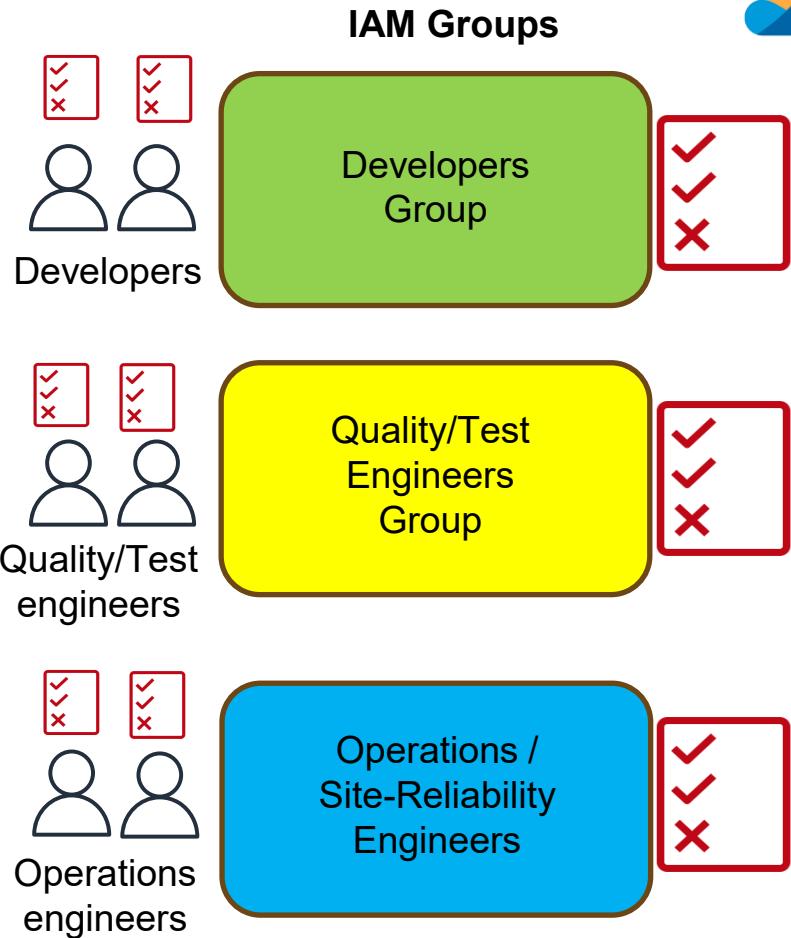
# IAM User, Group and Role



# IAM Groups



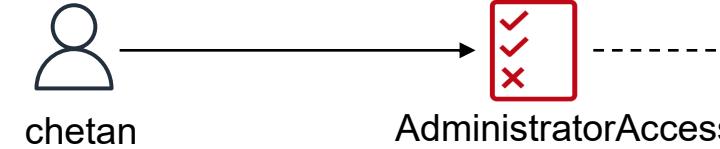
- Create IAM group and add IAM users to the group
- Instead of associating IAM policies to individual users, associate IAM policies to the group
- A group can contain multiple users, and user can belong to multiple groups.



# Exercise : Create IAM group

- 1 Create a new IAM group say 'Developers' and attach '**AmazonEC2FullAccess**' IAM policy to this group.
- 2 Create a new IAM user say 'Dave' and add it to Developers group. (Remember Dave's login credentials).
- 3 Login to AWS account as Dave (you can open Private/incognito window in the browser if you want to have 2 different AWS sessions).
- 4 Logged in as a 'Dave' user, try to create S3 bucket -> Access denied.
- 5 Try to launch a new EC2 instance -> Should be successful
- 6 Stop this EC2 instance

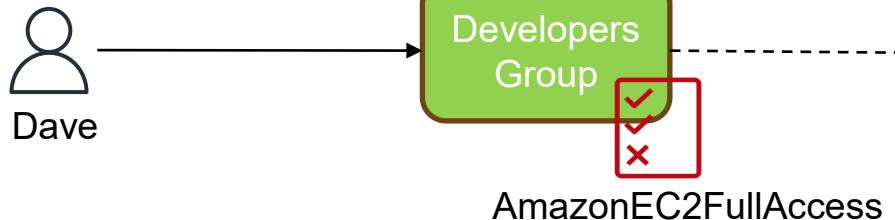
# IAM Users in our AWS account



Has all access to all AWS services (like a root user) but can't perform few actions e.g. closing AWS account etc.

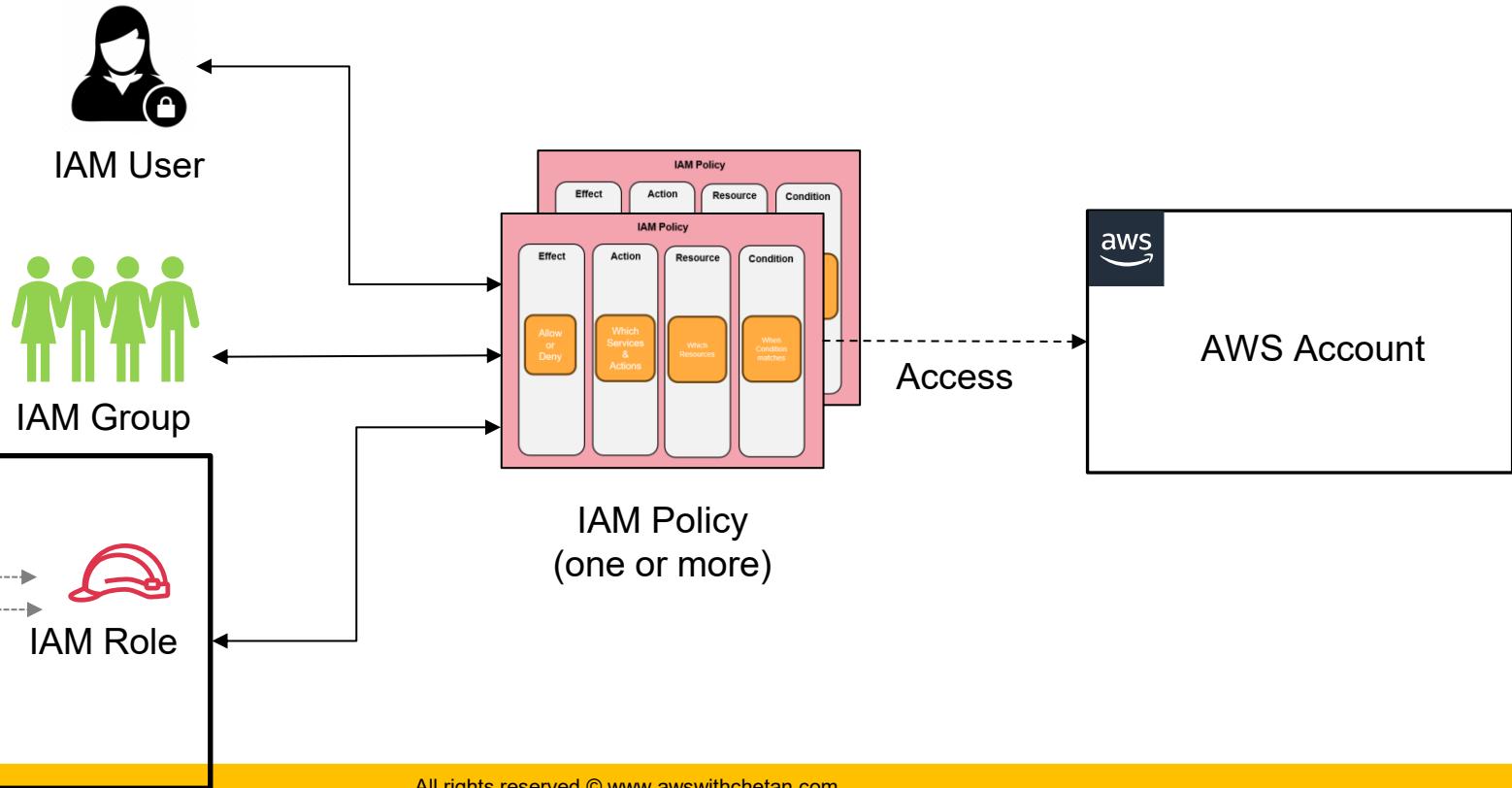


Has access to only list, start and stop EC2 instances. Do not have access to other EC2 actions and any other AWS service

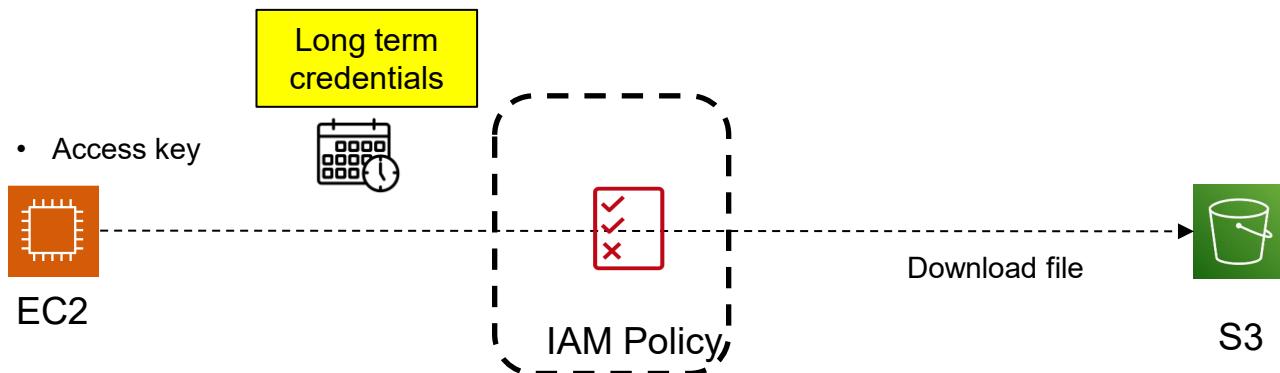
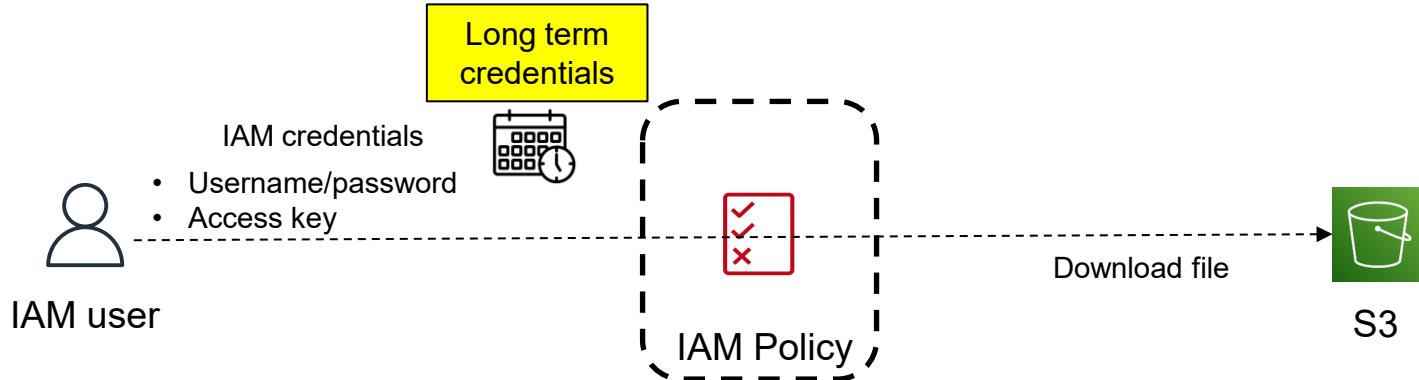


Has all access for EC2 service. Do not have access for any other AWS services.

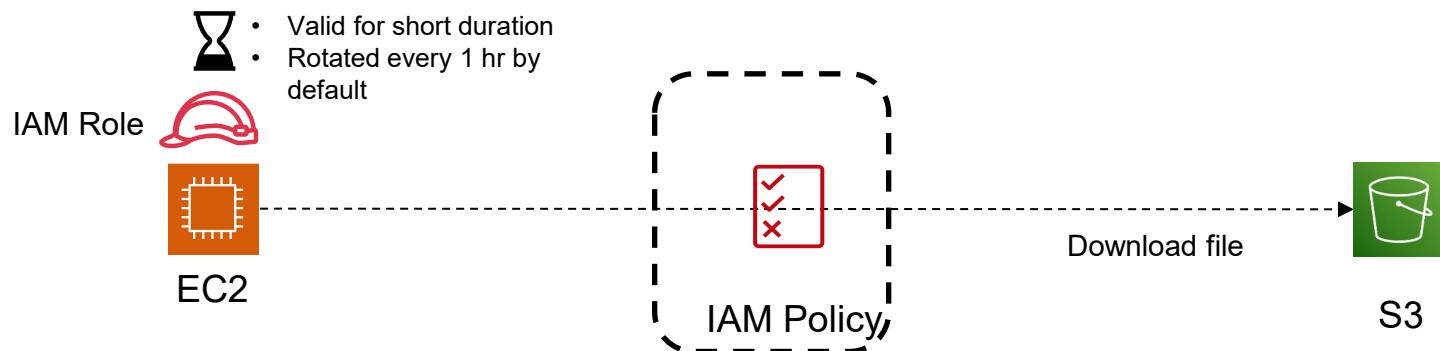
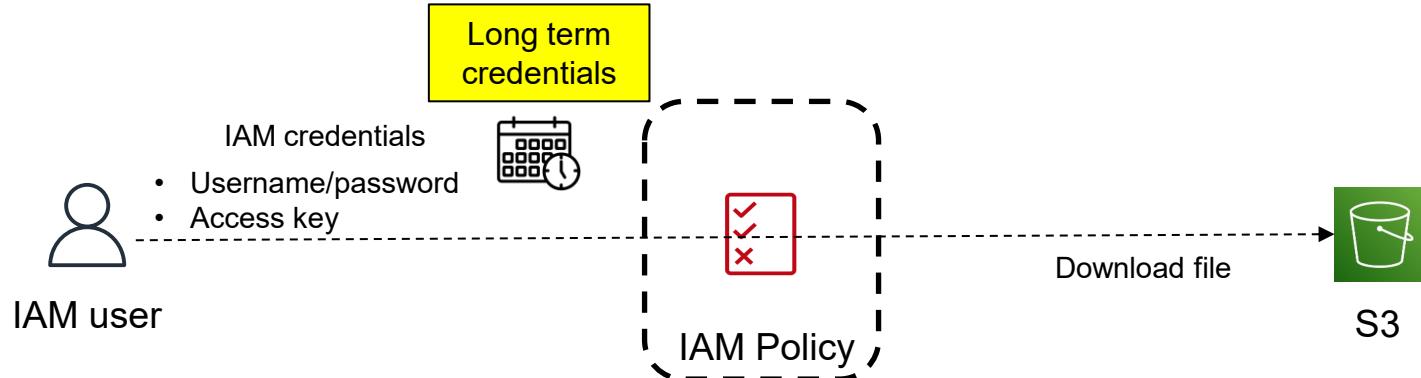
# IAM User, Group and Role



# IAM Roles

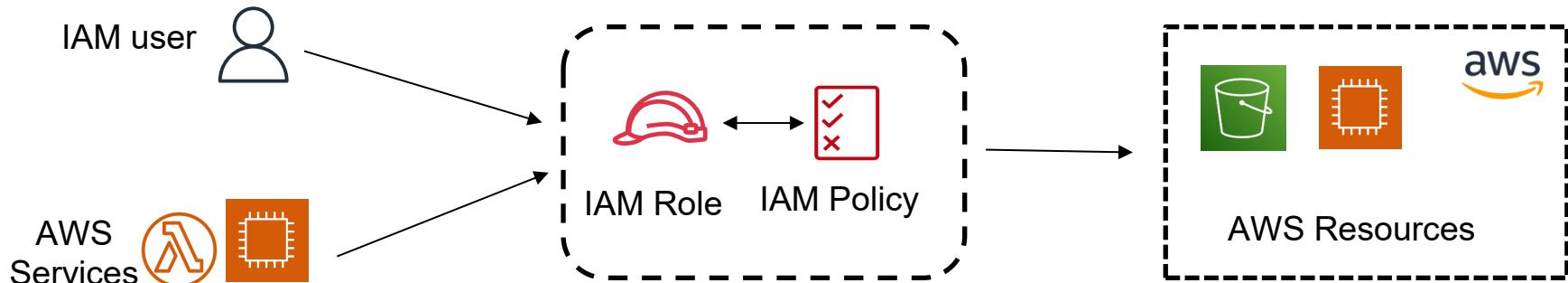


# IAM Roles

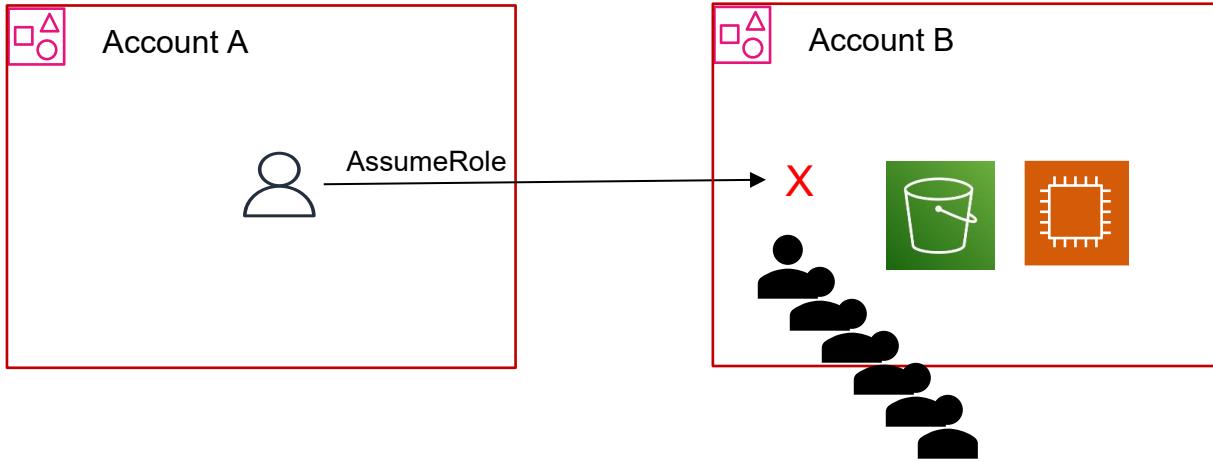


# IAM Roles

- IAM Role is similar to IAM user which has associated IAM policy to define permissions
- IAM Role is used by:
  - A web service offered by AWS such as Amazon EC2, Lambda
  - An IAM user in the same or a different AWS account (cross account access)

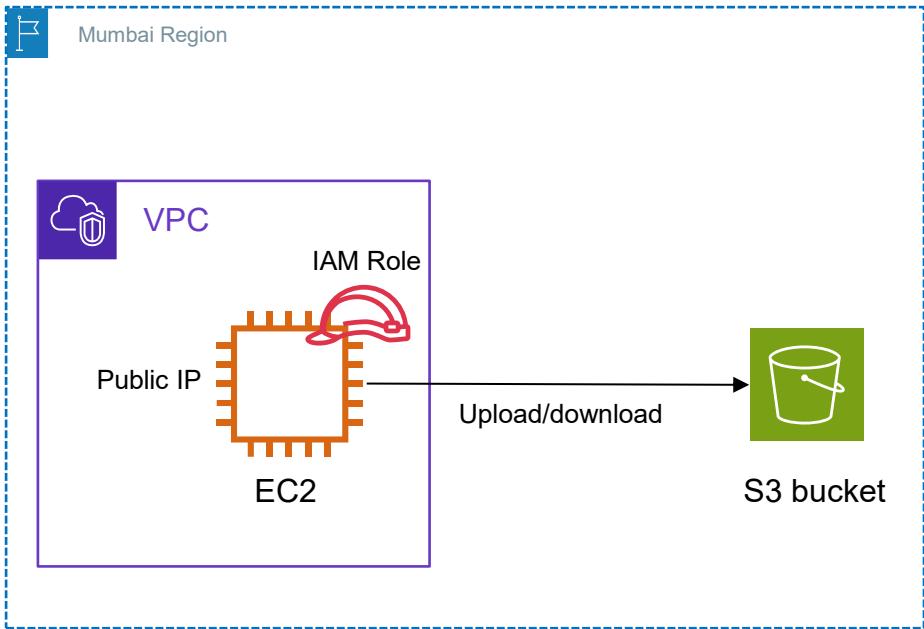


# IAM Role cross-account access



# Exercise: IAM Role

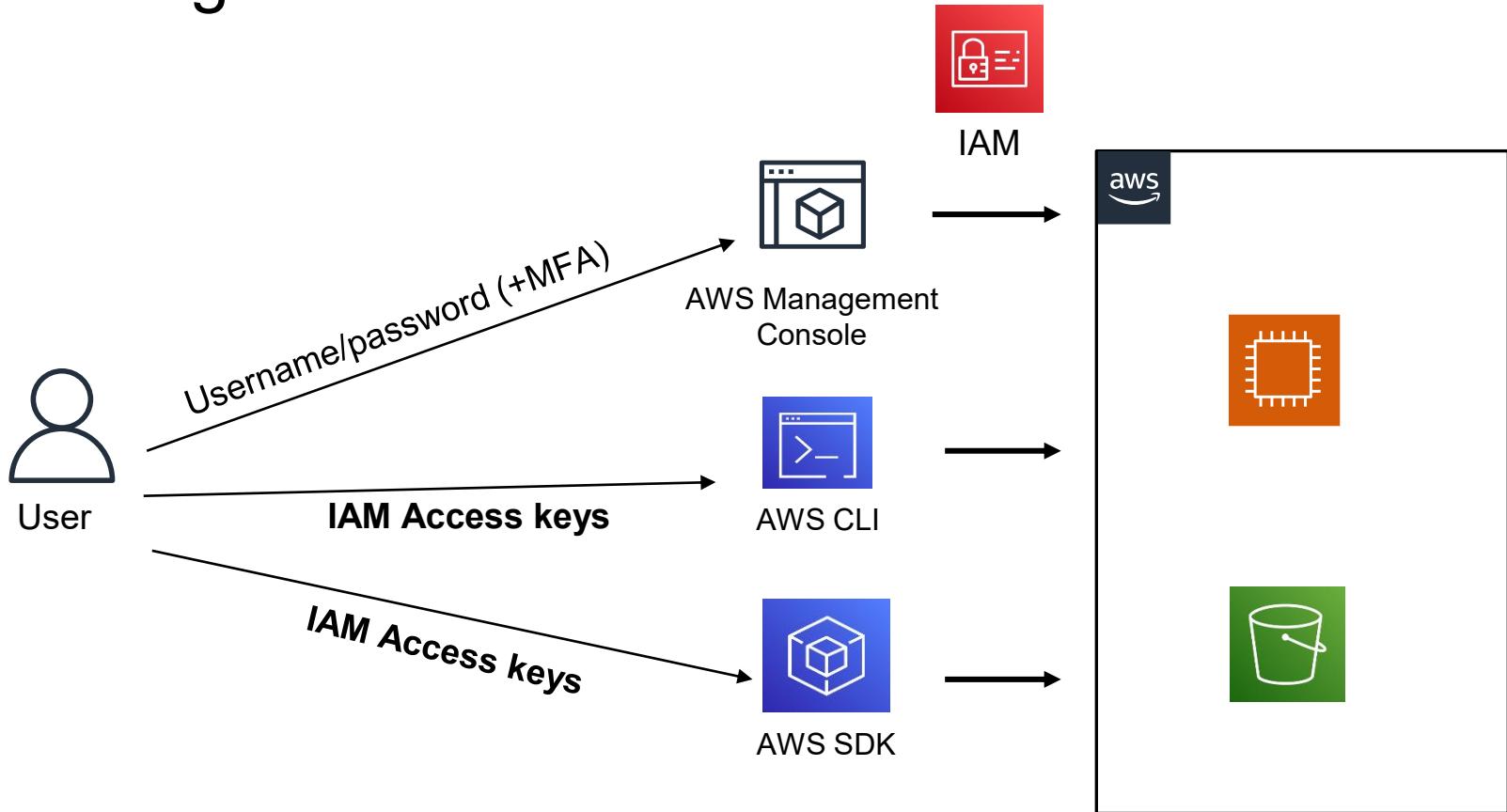
This exercise is a part of EC2 section



## Exercise steps

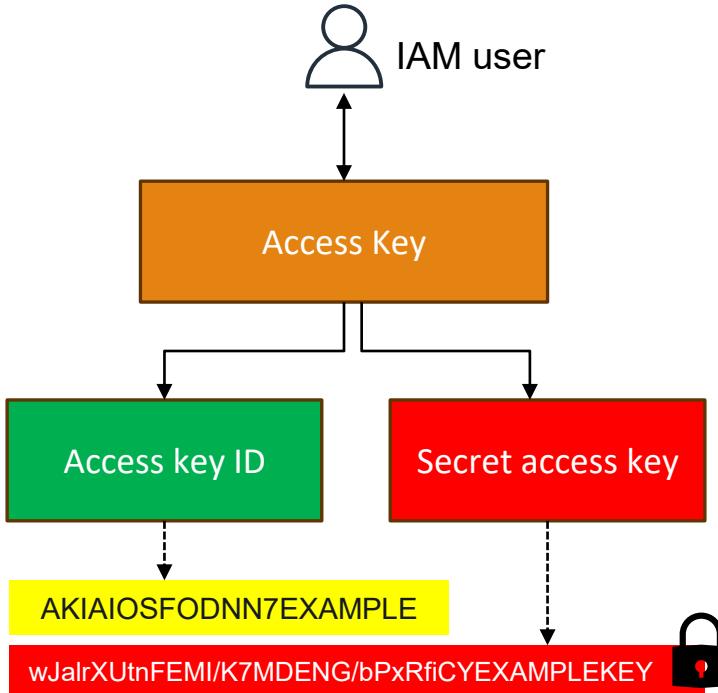
- 1 Launch EC2 Linux instance in a default VPC and connect to it over SSH.
- 2 Create S3 bucket in the same region and upload some sample text or image file.
- 3 From EC2 terminal, try to download file from S3 using AWS CLI command – Access denied.
- 4 Go to AWS IAM and create IAM role for EC2. Attach S3 full permissions policy to the role.
- 5 Associate this new role to EC2 instance
- 6 Try again to download file from S3 – should be successful.
- 7 Terminate EC2 instance. Optionally delete S3 bucket.

# Accessing AWS..



# IAM credentials – Access key

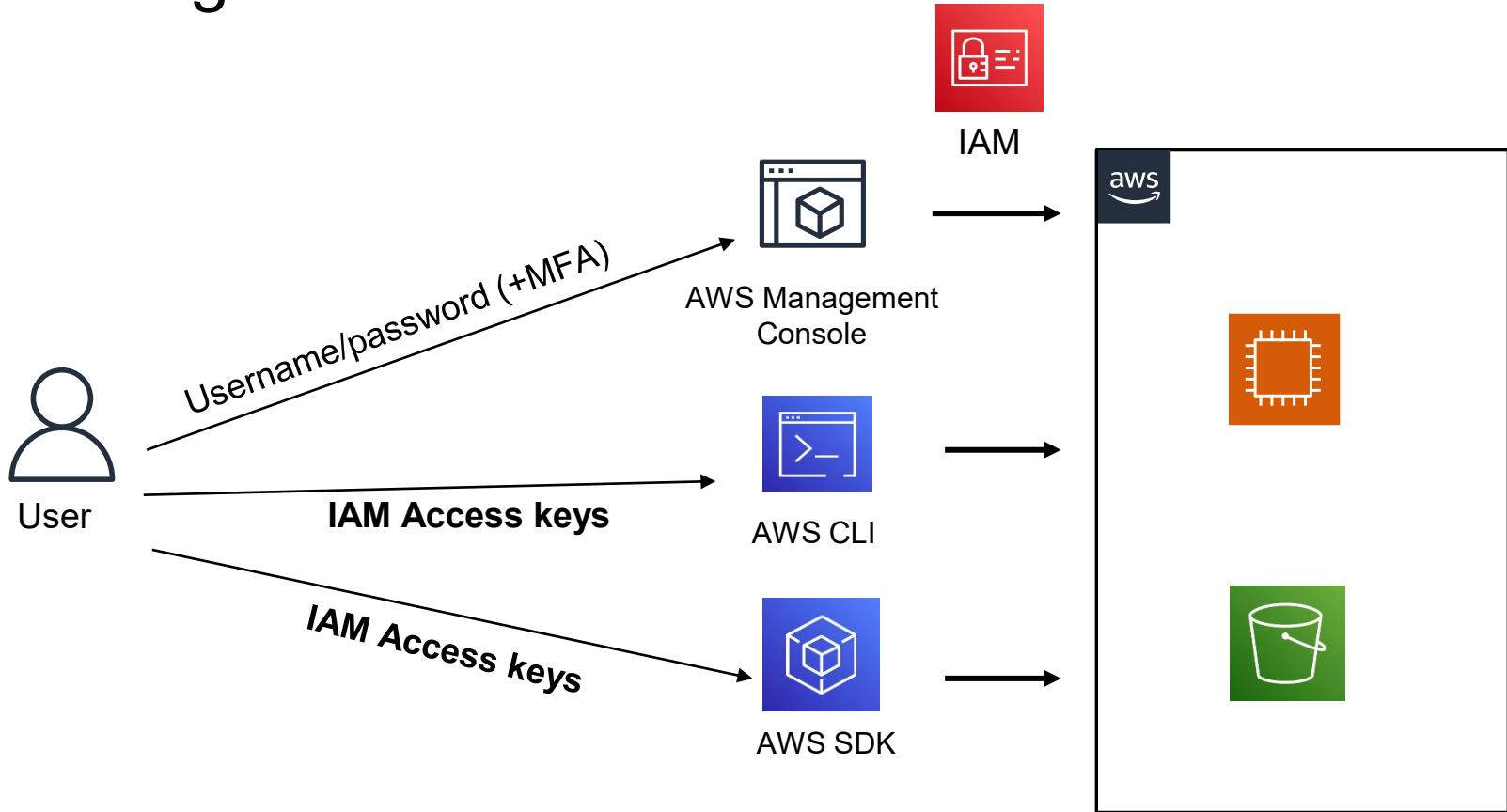
- Access keys are **long-term** credentials for an IAM user or the AWS account root user.
- Used to sign AWS API programmatic requests directly or via AWS CLI or AWS SDK
- Access keys consist of two parts:
  - **Access key ID** =~ **Identifies user**
  - **Secret access key** =~ **Like password**
- A user can have maximum 2 active Access keys at any time
- After generating Access key, store both Access key id and secret access key securely. It can not be re-generated.
- If lost, need to create a new Access key



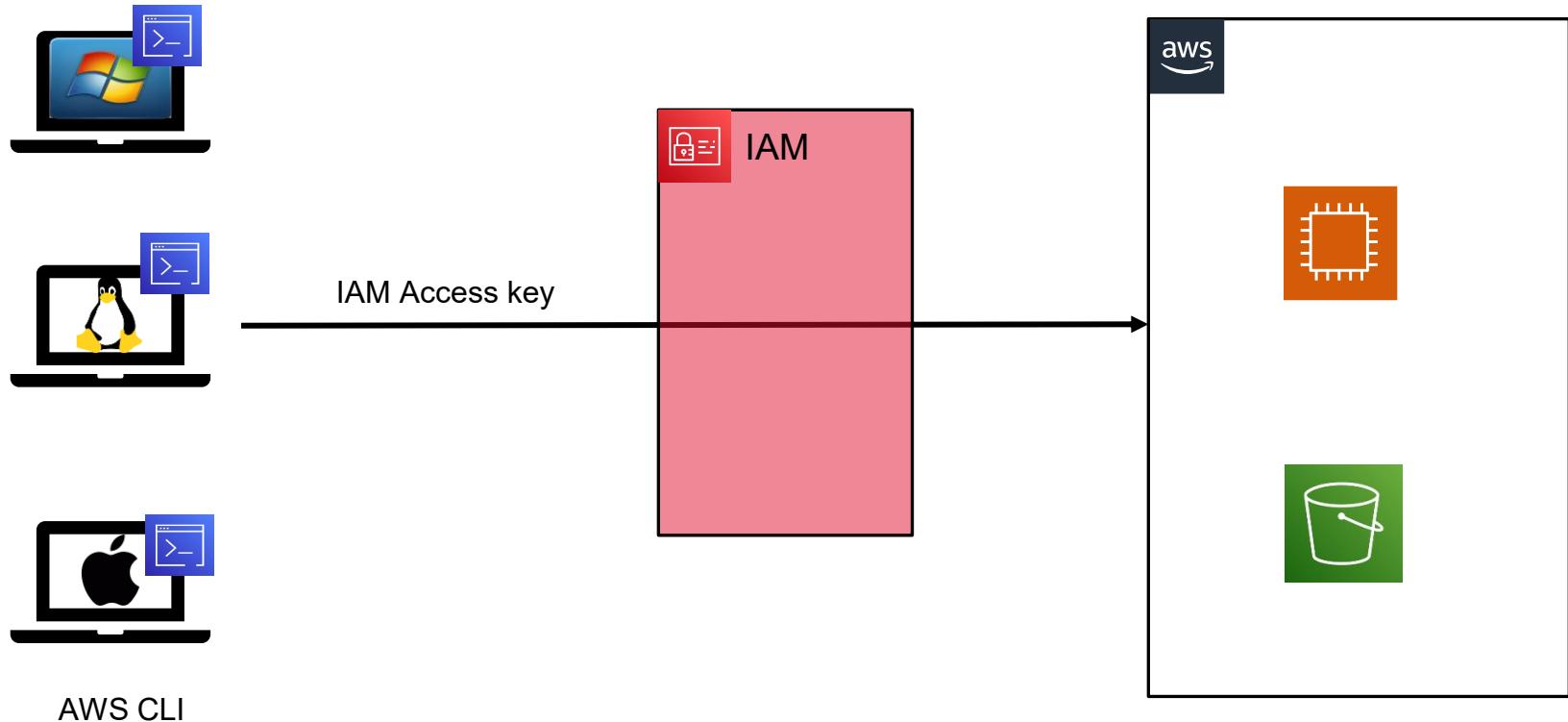
# Exercise : Create IAM access key for IAM user

- 1 Generate IAM access key for “automation” user. Store the generated keys securely.

# Accessing AWS..



# Accessing AWS using AWS CLI



# AWS CLI

- A Command Line Interface (CLI) to access AWS
- Configure AWS CLI by providing:
  - AWS region -> default region for CLI. Provide region code.
  - Access Key ID -> your access key id
  - Secret access key -> your secret access key
  - Output format -> text or json
- Access AWS using CLI commands:

```
[cloudshell-user@ip-10-134-40-230 ~]$ aws ec2 describe-regions
{
    "Regions": [
        {
            "Endpoint": "ec2.ap-south-1.amazonaws.com",
            "RegionName": "ap-south-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.eu-north-1.amazonaws.com",
            "RegionName": "eu-north-1",
            "OptInStatus": "opt-in-not-required"
        },
        {
            "Endpoint": "ec2.us-east-1.amazonaws.com",
            "RegionName": "us-east-1",
            "OptInStatus": "opt-in-not-required"
        }
    ]
}
```

```
$ aws configure
AWS Access Key ID [None]: AKIAIOSFODNN7EXAMPLE
AWS Secret Access Key [None]: wJalrXUtnFEMI/K7MDENG/bPxRfiCYEXAMPLEKEY
Default region name [None]: us-west-2
Default output format [None]: json
```

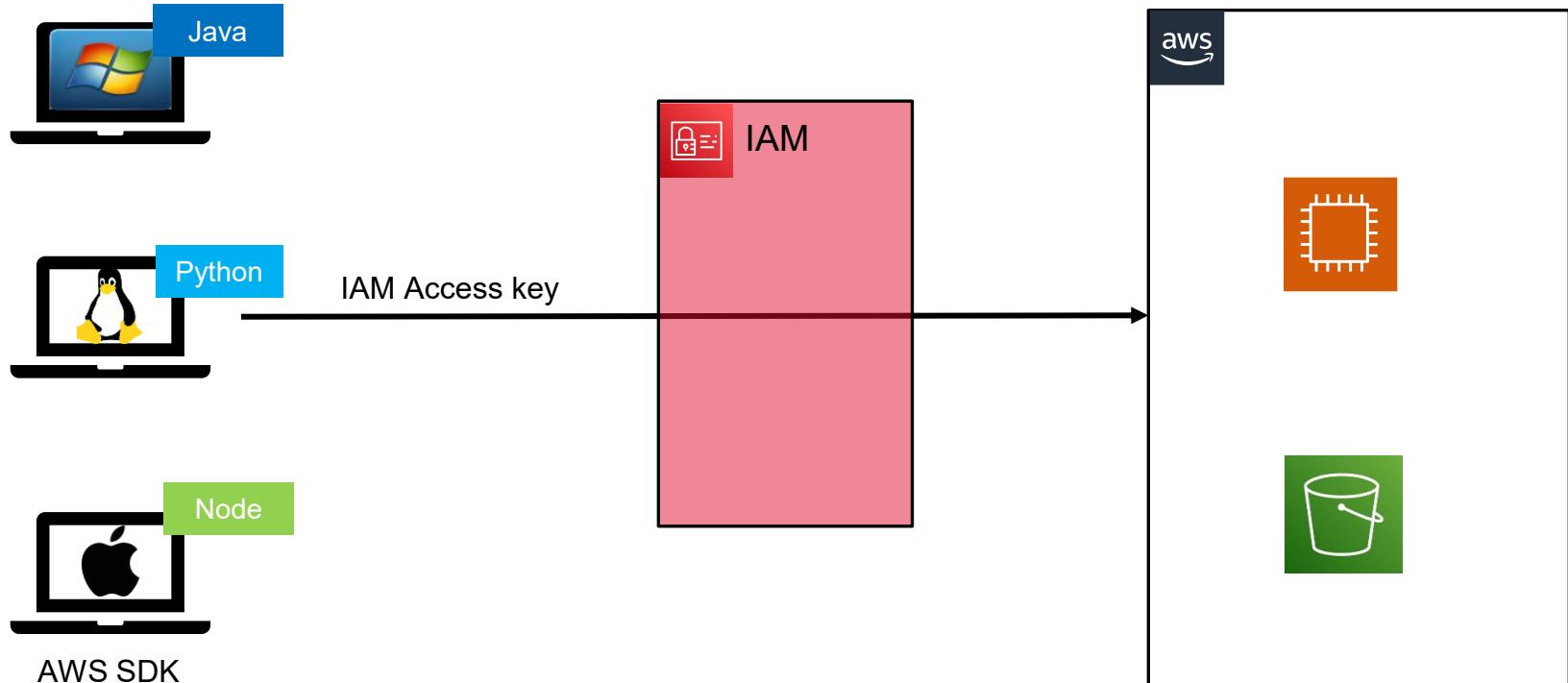
# Exercise : Use CLI to stop the EC2 instance

- 1 Install AWS CLI on your workstation
- 2 Configure AWS CLI – Provide the Access Key ID and Secret Access Key for “**automation**” user downloaded during the previous exercise
- 3 Run AWS CLI command (describe-instances) to list the ec2 instances in your account. Copy instance-id.
- 4 Run AWS CLI commands to start and stop the EC2 instance by providing your instance-id

```
$ aws ec2 describe-instances  
  
$ aws ec2 start-instances --instance-ids i-xxxxxxxxxxxxxx  
  
$ aws ec2 stop-instances --instance-ids i-xxxxxxxxxxxxxx
```

<https://docs.aws.amazon.com/cli/latest/userguide/getting-started-install.html#getting-started-install-instructions>

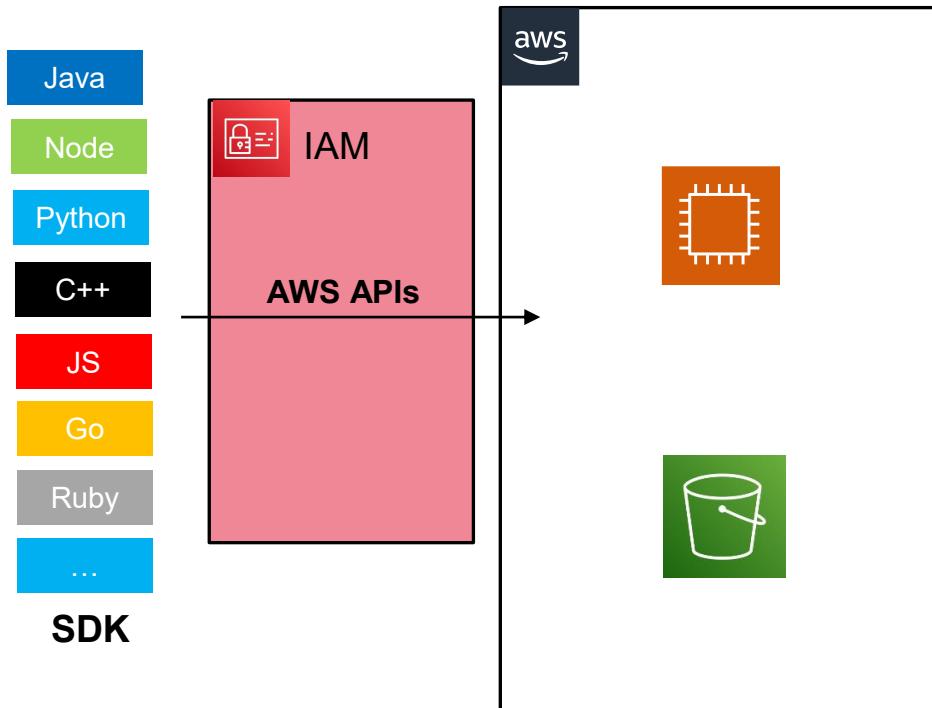
# Accessing AWS using AWS SDK



# AWS SDK

- AWS web services are exposed through REST & HTTP APIs
- SDK wraps these APIs to handle low level operations e.g. request signing, authentication, headers management
- SDKs are language specific
- Applications are typically built using corresponding language SDK

**Example: AWS CLI uses python SDK**



# Exercise : Use SDK to start/stop ec2 instance

- 1 Use your own Linux workstation or use AWS CloudShell
- 2 Install python boto3 <https://boto3.amazonaws.com/v1/documentation/api/latest/guide/quickstart.html>
- 3 Create a python file – code provided in the next slide. Also available on github.  
[https://github.com/awswithchetan/aws-cloud-practitioner/blob/main/iam/start\\_ec2\\_instance.py](https://github.com/awswithchetan/aws-cloud-practitioner/blob/main/iam/start_ec2_instance.py)  
[https://github.com/awswithchetan/aws-cloud-practitioner/blob/main/iam/stop\\_ec2\\_instance.py](https://github.com/awswithchetan/aws-cloud-practitioner/blob/main/iam/stop_ec2_instance.py)
- 4 Replace the access key id, secret access key, region and instance id with your values
- 5 Execute script: python your\_script.py

# Sample boto3 code

```
import boto3

# Replace these with your actual access key and secret access key
ACCESS_KEY = 'your-access-key-id'
SECRET_KEY = 'your-secret-access-key'

# Replace with your desired AWS region and instance ID
REGION = 'ap-south-1'
INSTANCE_ID = 'i-1234567890abcdef0'

# Create an EC2 client using the access key and secret key
ec2_client = boto3.client(
    'ec2',
    aws_access_key_id=ACCESS_KEY,
    aws_secret_access_key=SECRET_KEY,
    region_name=REGION
)

# Start the instance
response = ec2_client.start_instances(InstanceIds=[INSTANCE_ID])

# Print the response
print(response)
```

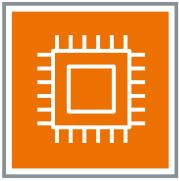
# IAM Access audit and governance

# AWS IAM Reports

- **IAM Credentials Report**
  - Generate and download a credential report that lists all users in your account and the status of their various credentials, including passwords, access keys, and MFA devices.
- **IAM Access Analyzer**
  - External access analyzer – Identify resources shared with external entities
  - Unused access analyzer – Identify unused access in the account
  - Validate IAM policy against your organization's security standards
  - Generate IAM policy based on the activities captured in AWS CloudTrail logs
- **IAM access advisor**
  - IAM Access advisor uses data analysis to help you set permission guardrails confidently by providing service last accessed information for your accounts.
  - You can use this information to finetune IAM policies in your account

# IAM summary

- Root user - Owner of the AWS account. Root user has unrestricted permissions to AWS account. Do not use unless really required to.
- IAM User - A regular user who can have restricted access to AWS account.
- Use MFA + Password Policy for securing AWS account console login for root and IAM users
- IAM User Group - Group of IAM users. Can attach IAM policies to the group.
- IAM Policy - JSON document that outlines permissions for IAM users or groups or role.
- IAM Role - Provides temporary IAM credentials for AWS services (e.g. EC2, Lambda) or cross-account IAM users or federated users.
- Access Keys – For programmatically accessing AWS account using the CLI or SDK. Consists of Access key ID + Secret access key. Must keep Access & secret keys secure.
- AWS CLI - Manage your AWS services using the command-line interface
- AWS SDK - Manage your AWS services using a programming language (python, java, node, C# and more)
- AWS CloudShell – A browser based pre-authenticated shell to access AWS CLI from AWS console
- IAM Audit and governance - IAM Credential Reports, IAM Access Analyzer and IAM access advisor



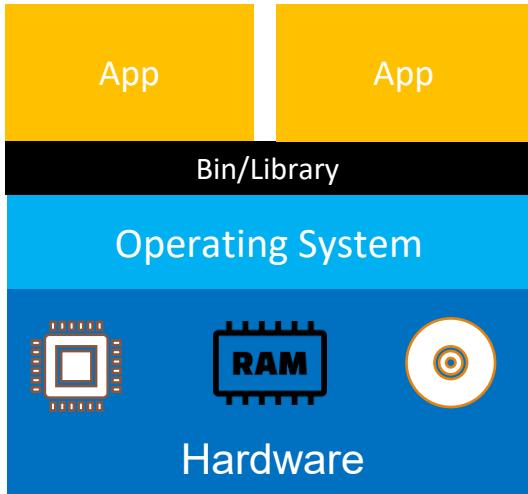
# AWS compute services

EC2, Containers, ECS, EKS, Fargate, Lambda

# Physical server vs Virtual Machines vs Containers

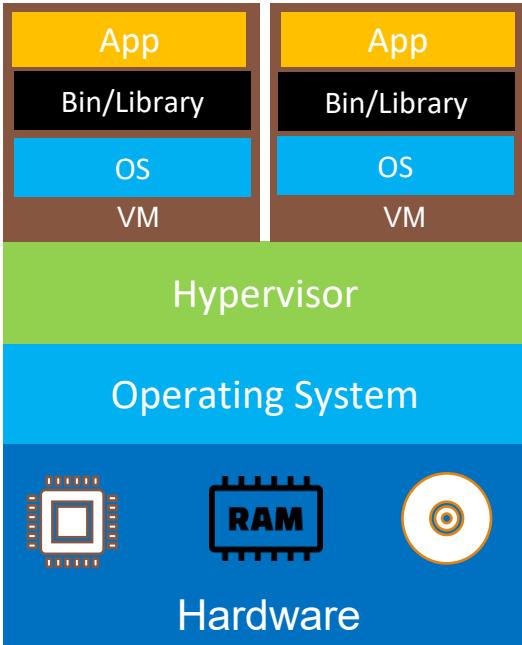


- ✓ High Performance
- ✓ Physical isolation / License compliance
- ✗ Resource contention for Apps
- ✗ Noisy neighbor problem



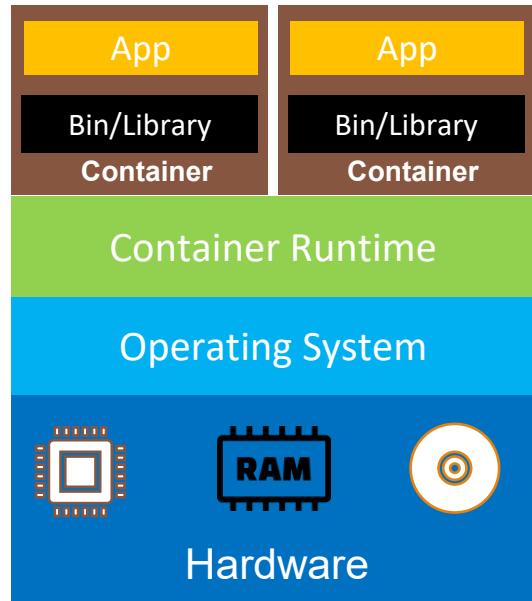
Physical Servers

- ✓ OS/Libraries flexibility
- ✓ VM level resource utilization flexibility



Virtualized deployment  
All rights reserved © www.awswithchetan.com

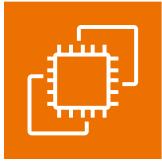
- ✓ Lightweight – shares underlying OS
- ✓ Faster to deploy
- ✓ Portability



Containerized deployment

# AWS compute choices

Need Virtual machine?



EC2

Need to run containers?



Elastic Container Service (ECS)



Elastic Kubernetes Service (EKS)



Need to run code or function without managing servers?



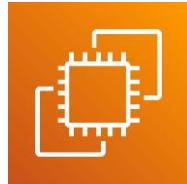
Lambda

Serverless

more

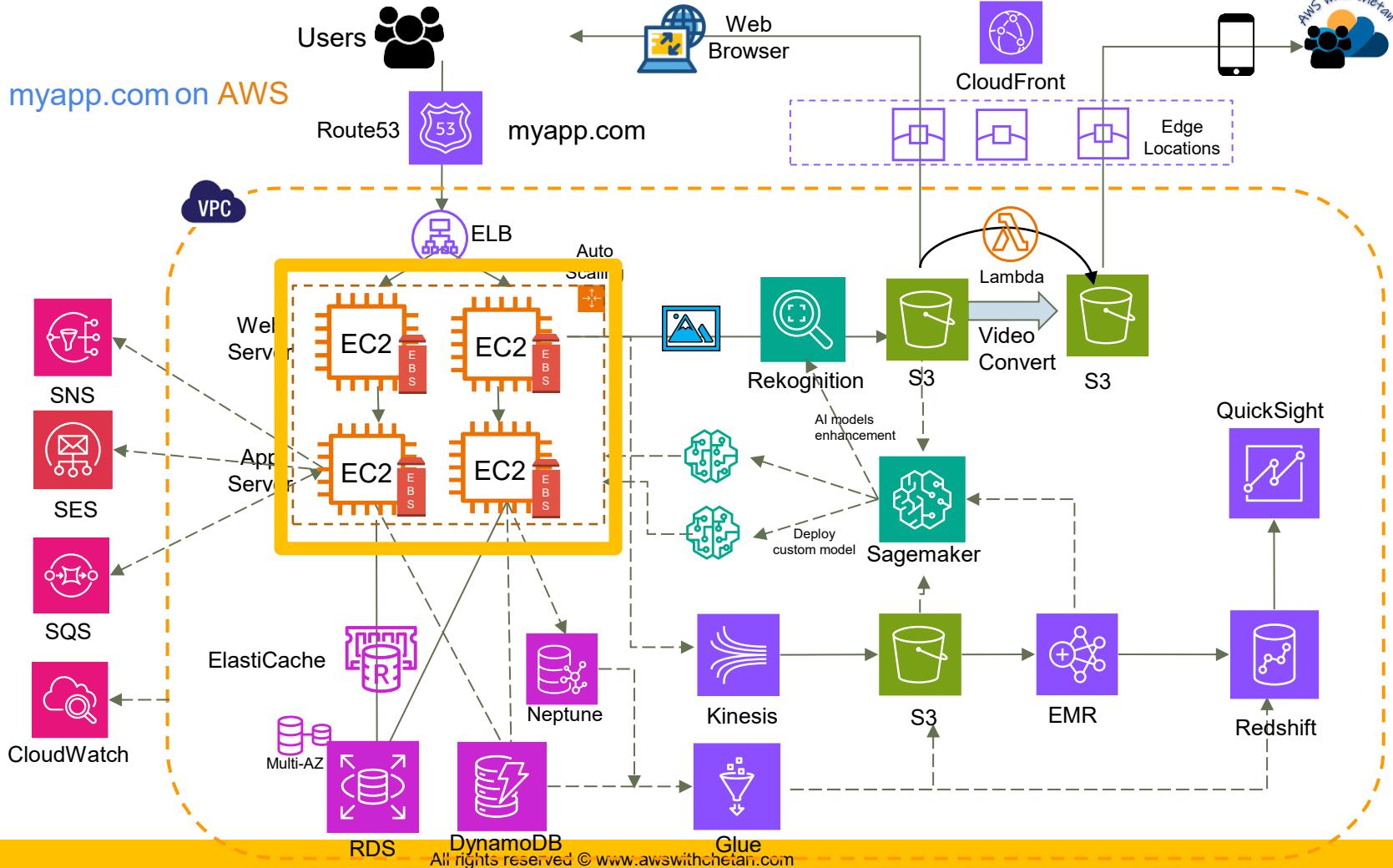
less

Control and Management by the customer



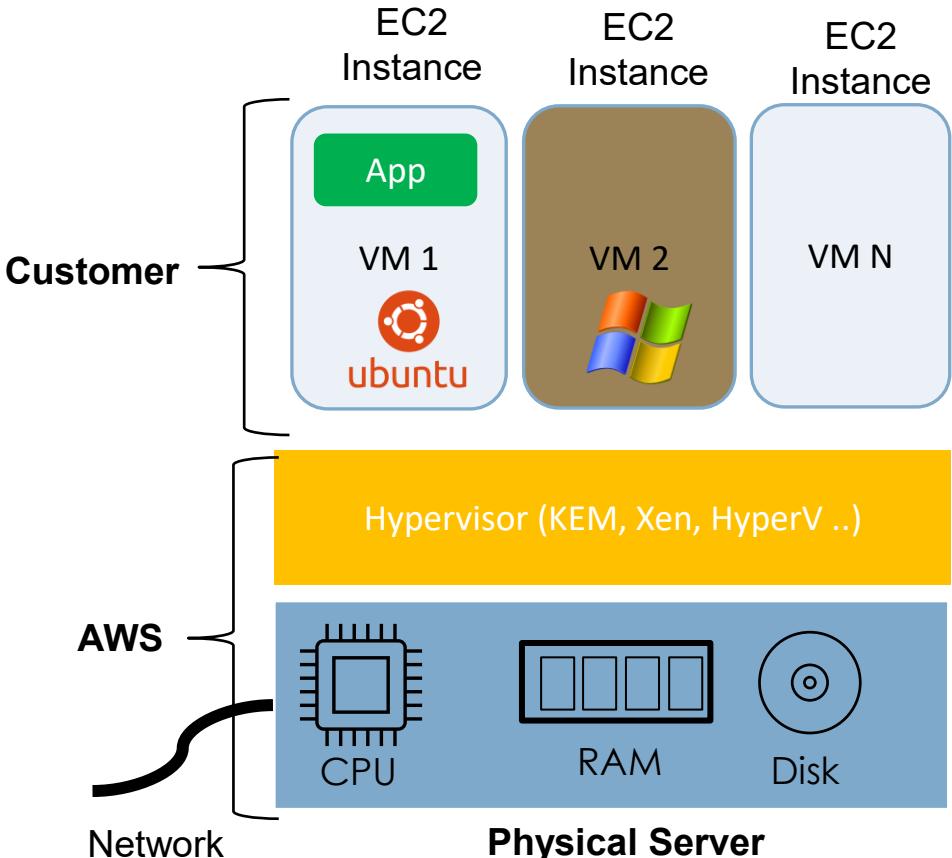
# Amazon EC2

## Elastic Compute Cloud



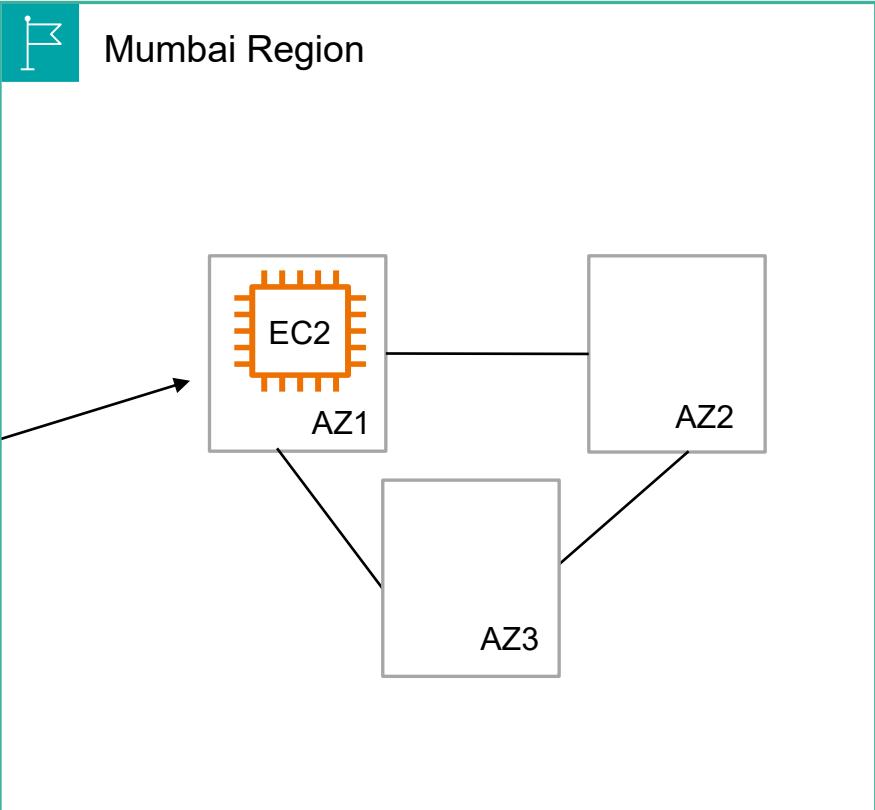
# What is EC2?

- **Elastic Compute Cloud**
- Virtual Machine in AWS Cloud
- AWS customers get access to EC2 instance
- Customer do not have access to underlying physical hardware

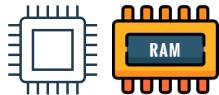


# Where EC2 is hosted?

AWS Region -> Availability Zone -> EC2



# Configuration options for EC2



Processor, CPU and Memory => **EC2 Instance Type/Size**



Storage => **Elastic Block Storage (EBS) or Instance Store**



Operating System => **Amazon Machine Image (AMI)**



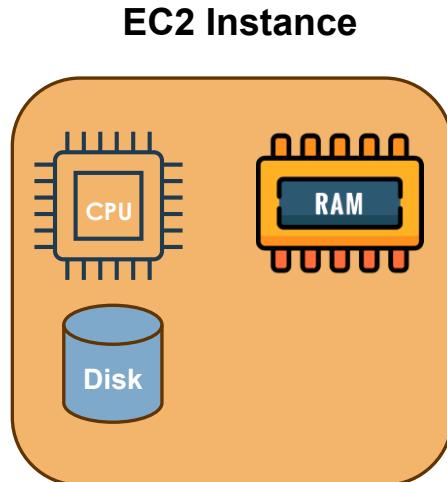
Network => **Virtual Private Cloud (VPC)**



Firewall => **Security Group**



Login Credentials => **SSH Key-pair**



# Additional configuration options for EC2



Initialization scripts => **EC2 Userdata**



Permissions to access other => **IAM Role**  
AWS Services



Purchasing Options => **On-demand, Spot, Reserved, Savings Plan**

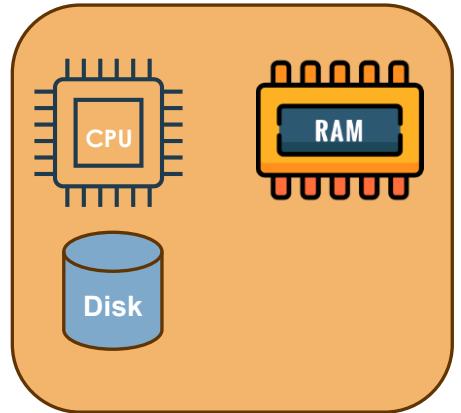


EC2 Tenancy Options => **Dedicated or Shared**

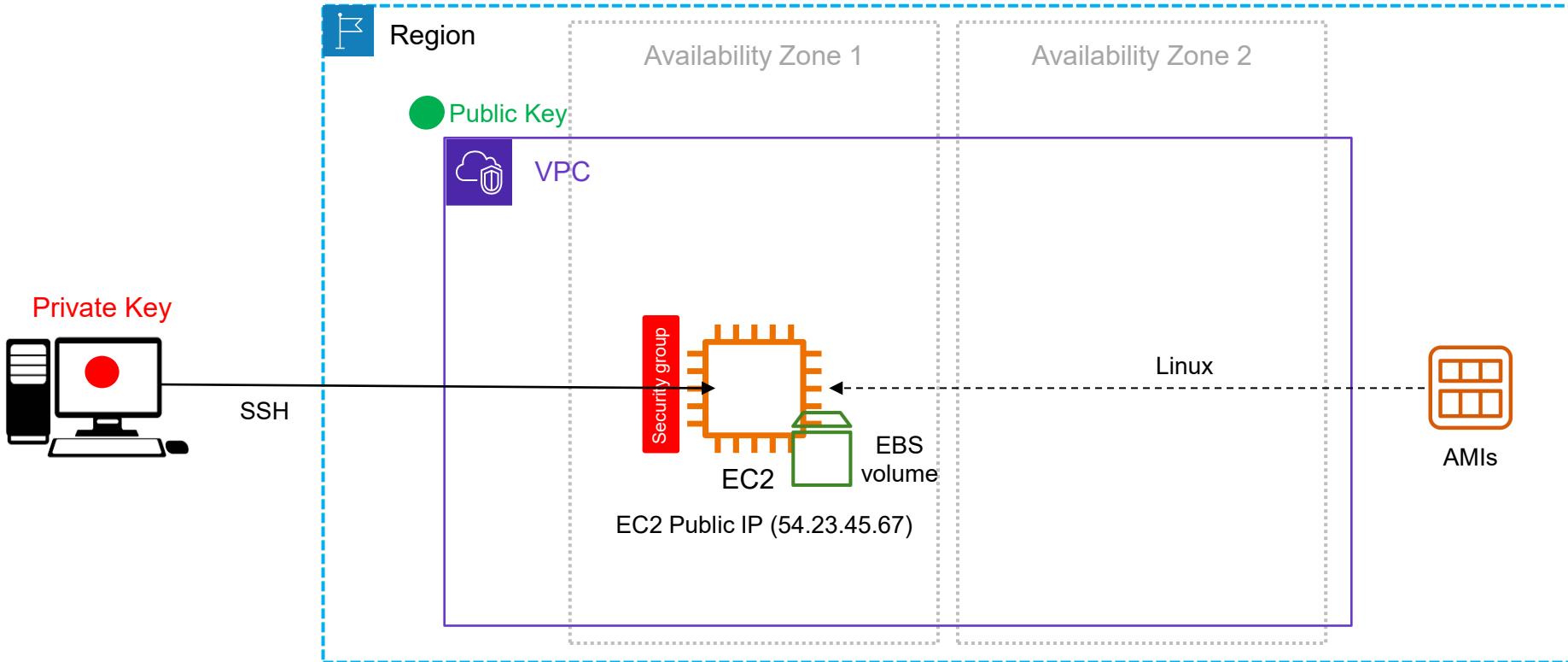


A label to identify EC2 => **EC2 Tags**

**EC2 Instance**



# Process to launch EC2 Instance



# Exercise : Launch EC2 instance (Linux)

1

## Launch EC2 (Linux) instance in Mumbai Region

- a) Go to EC2 Service -> EC2 Dashboard -> Launch Instances
- b) Name: MyEC2Linux
- c) Select Application and OS Images (Amazon Machine Image): Amazon Linux 2 (default)
- d) Select instance type: t2.micro (default)
- e) Select key pair : *Your key-pair that you had created earlier in pre-requisites*
- f) Network settings -> Default VPC
- g) Make sure Auto-Assign Public IP is enabled
- h) Firewall -> Create security group
  - a) Allow SSH traffic from -> Select My IP [Type-> SSH, port Range-> 22, source type -> My IP]
  - b) Allow HTTP traffic from -> Everywhere-IPv4
- a) Configure Storage -> 8GiB, gp3 (default)
- b) Launch Instance

2

## Connect over SSH

- a) If using windows workstation – Open Putty session, Load key file and use EC2 public IP or DNS for connection
- b) If using Mac or Linux workstation – Open terminal and use ssh command: `ssh -i <keyfile> <ec2 public IP>`
- c) Use login user as **ec2-user**

3

## Terminate the instance

- a) EC2 console -> Select your instance -> Instance State -> Terminate instance

# Exercise : Launch EC2 instance (Windows)

## 1 Launch EC2 (Linux) instance in Mumbai Region

- a) Go to EC2 Service -> EC2 Dashboard -> Launch Instances
- b) Name: MyEC2Windows
- c) Select Application and OS Images (Windows): Microsoft Windows Server 20XX Base
- d) Select instance type: t2.micro (default)
- e) Select key pair : *Your key-pair that you had created earlier in pre-requisites*
- f) Network settings -> Default VPC
- g) Make sure Auto-Assign Public IP is enabled
- h) Firewall -> Create security group
- i) Allow RDP traffic from -> Select My IP [Type-> RDP, port Range-> 3389, source type -> My IP]
- j) Configure Storage -> 8GiB, gp3 (default)
- k) Launch Instance

## 2 Retrieve Windows Administrator password

- a) Go to EC2 console -> Select your instance -> Actions -> Get Windows Password
- b) Paste your .pem key file content (or browse and load .pem file). AWS will provide you the decrypted Windows password

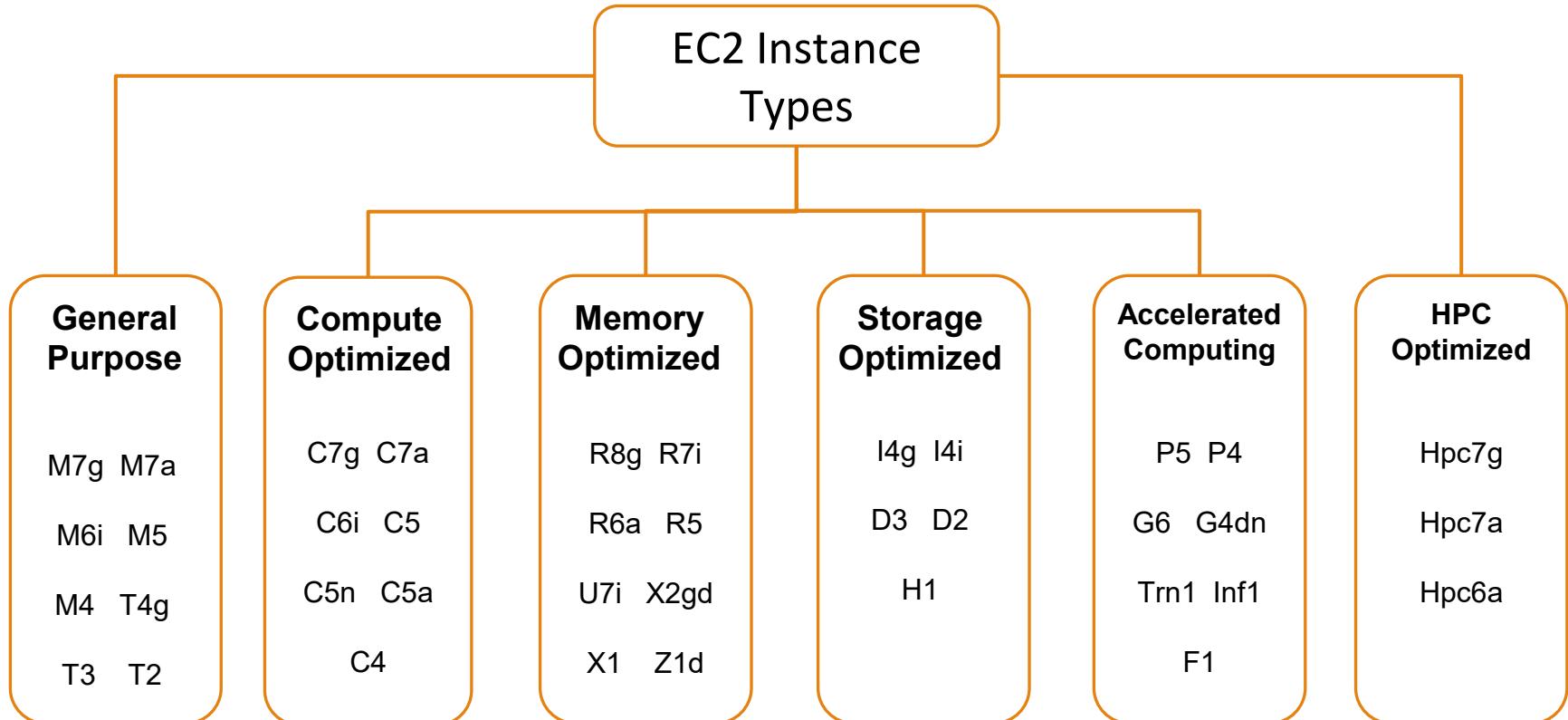
## 3 Connect over RDP

- a) Open RDP session using Remote Desktop Client to Public IP/DNS of EC2 Windows instance
- b) Provide Username= Administrator, Password= <decrypted password>

## 4 Terminate the instance

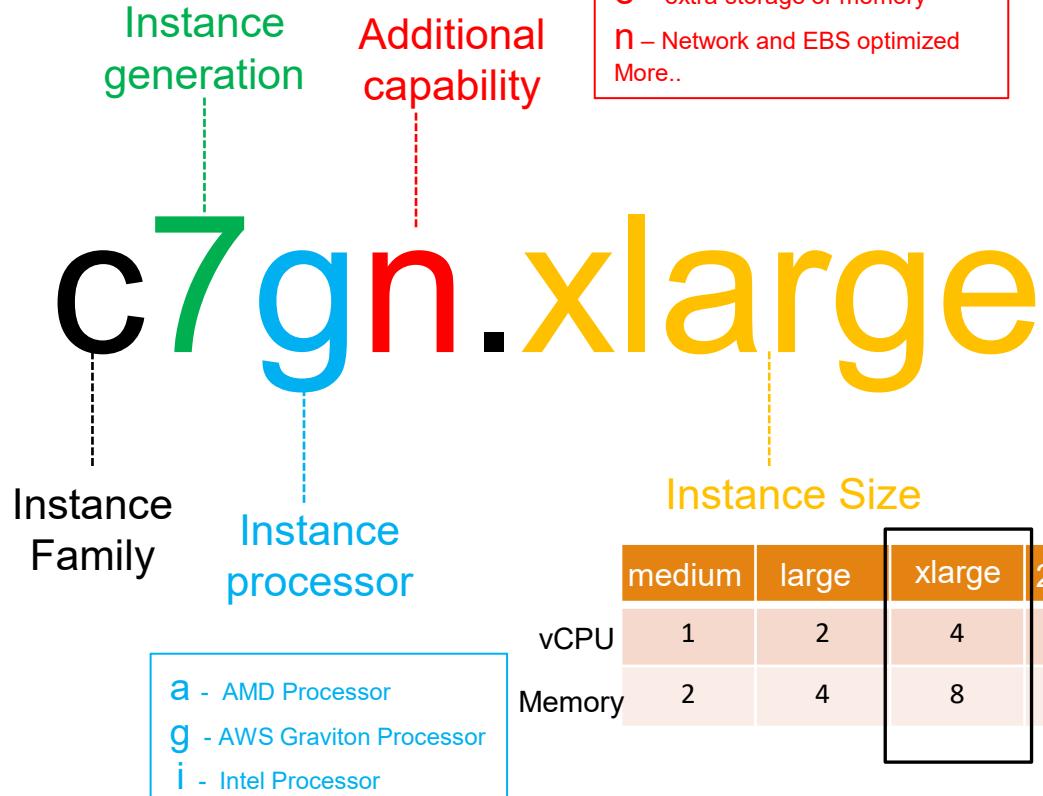
- a) EC2 console -> Select your instance -> Instance State -> Terminate instance

# EC2 Instance Types



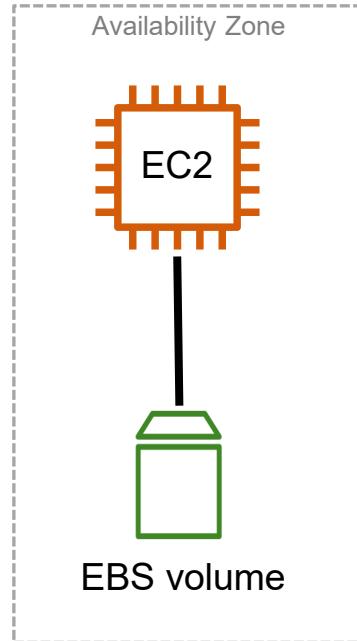
# EC2 Instance naming

C - Compute Optimized
D - Dense storage
F - FPGA
G - Graphics Intensive
I - Storage Optimized
Hpc - HPC optimized
P - GPU accelerated
M - General Purpose
T - Burstable Performance
U - High Memory
Vt - Video Transcoding
X - Memory intensive
More..



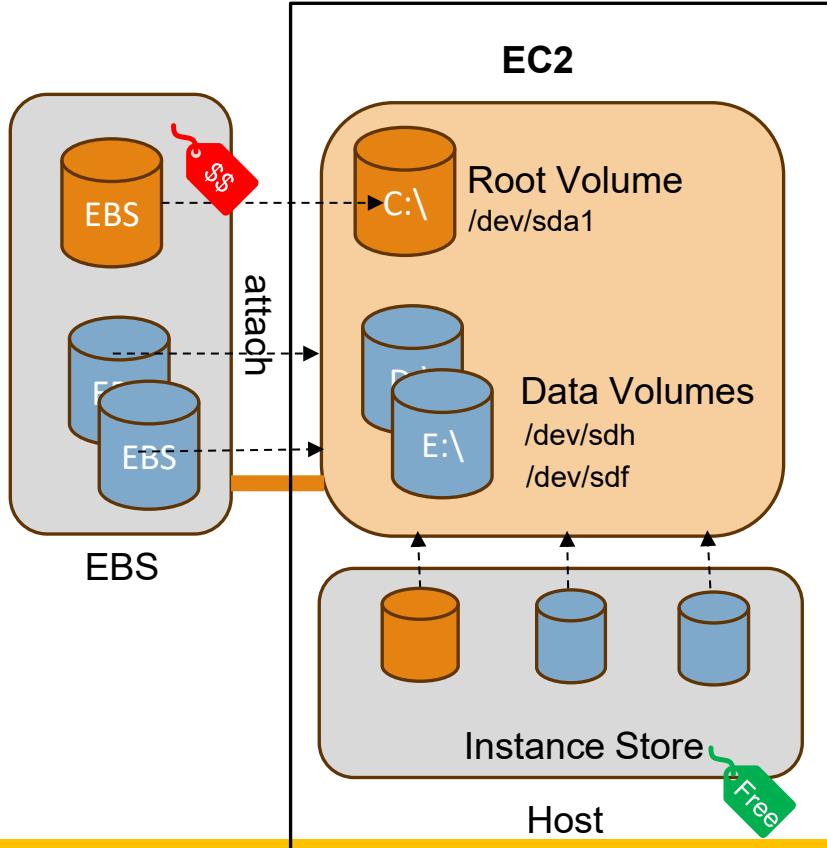
# Storage for EC2 = EBS

- EBS is **Elastic Block Storage** service which provides persistent block storage for EC2 instance
- We can create one or more EBS volumes (disk) and attach to EC2 instance
- EBS volumes are elastic – we can increase the size of EBS volumes as required
- EC2 has a root volumes (contains operating system) and optionally can have one or more data volumes
- EBS is a network attached storage (SAN) and hence volumes can be persisted irrespective of EC2 lifecycle
- EBS volume data can be backed up using Snapshots

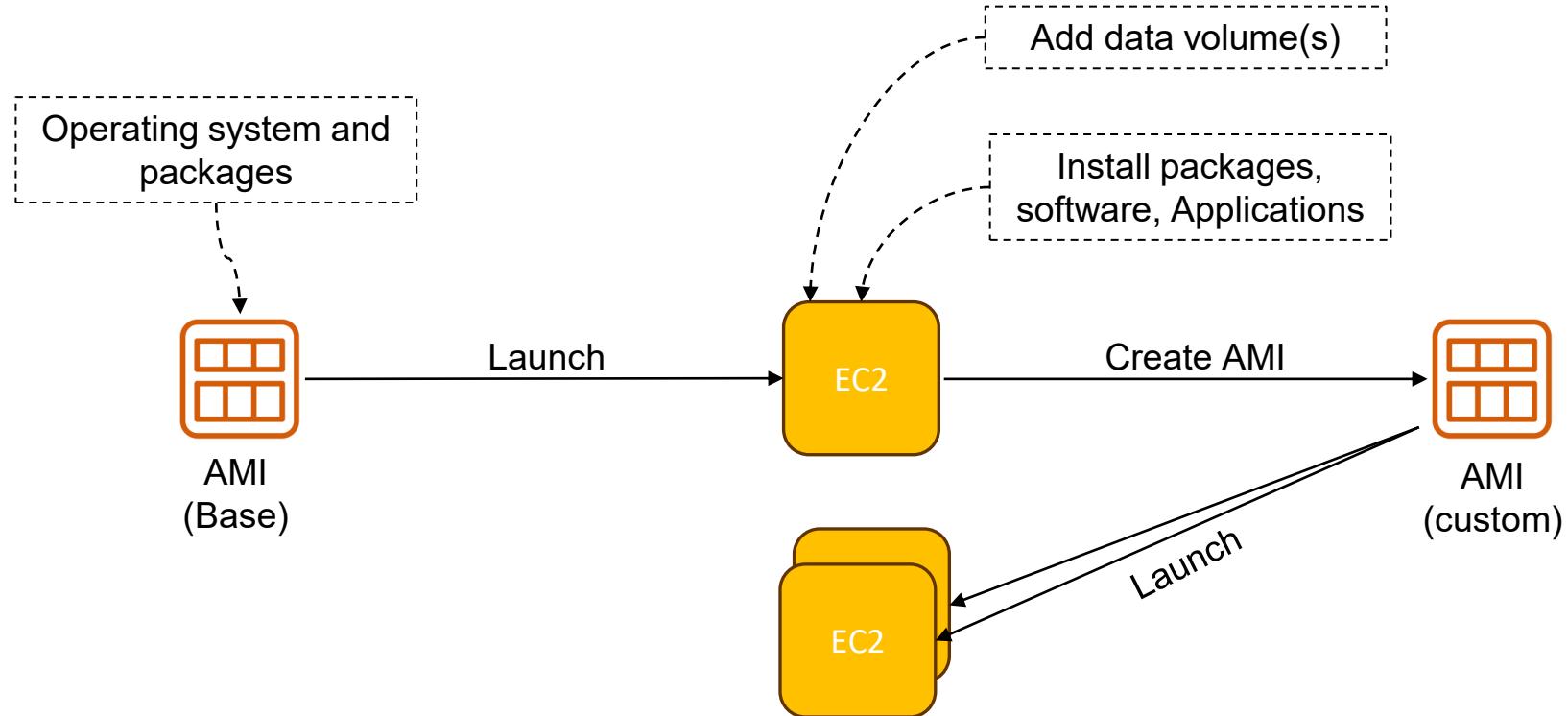


# EBS vs Instance store

- EC2 disks can be of 2 types:
  - Elastic Block Storage (external to EC2 host)
  - Instance Store – Local to EC2 host
- EBS volume data is persisted until volume is deleted
- Instance Store volume data is wiped out if EC2 is stopped and started.
- EBS volumes can be created and attached to EC2 instance. Instance store volume has size limit and comes with specific EC2 instance types only.
- EBS has per GB cost. Instance volumes are free.
- EBS is used for almost all purposes whereas Instance store can be used for temp directory, buffer or cache

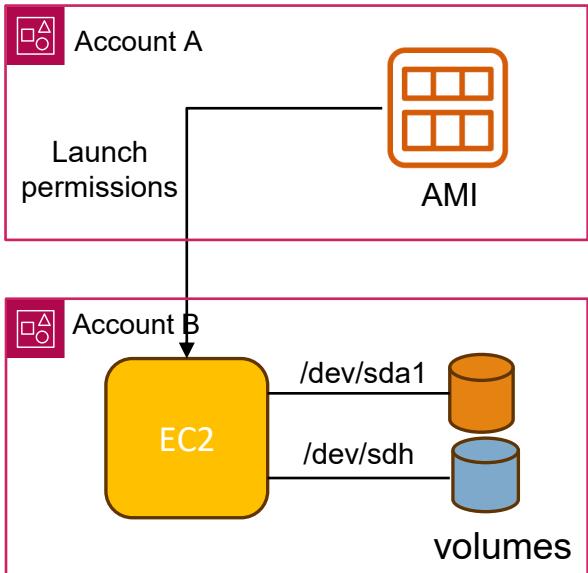


# Amazon Machine Image (AMI)



# Amazon Machine Image (AMI)

- An AMI provides information required to launch an instance.
- You can launch multiple instances from a single AMI when you require multiple instances with the same configuration.
- An AMI includes the below:
  - A template for the root volume for the instance
  - One or more Amazon EBS snapshots
- Launch permissions that control which AWS accounts can use AMI to launch instances
- You can use AMIs from AWS Community or AWS Marketplace AMIs or you can create your own AMIs.



# Amazon Machine Image (AMI)

Quick Start AMIs (45)      My AMIs (2)      AWS Marketplace AMIs (10621)      Community AMIs (500)

Commonly used AMIs      Created by me      AWS & trusted third-party AMIs      Published by anyone

---

**Refine results**

[Clear all filters](#)

Free tier only [Info](#)

OS category

All Linux/Unix

All products (45 filtered, 45 unfiltered)

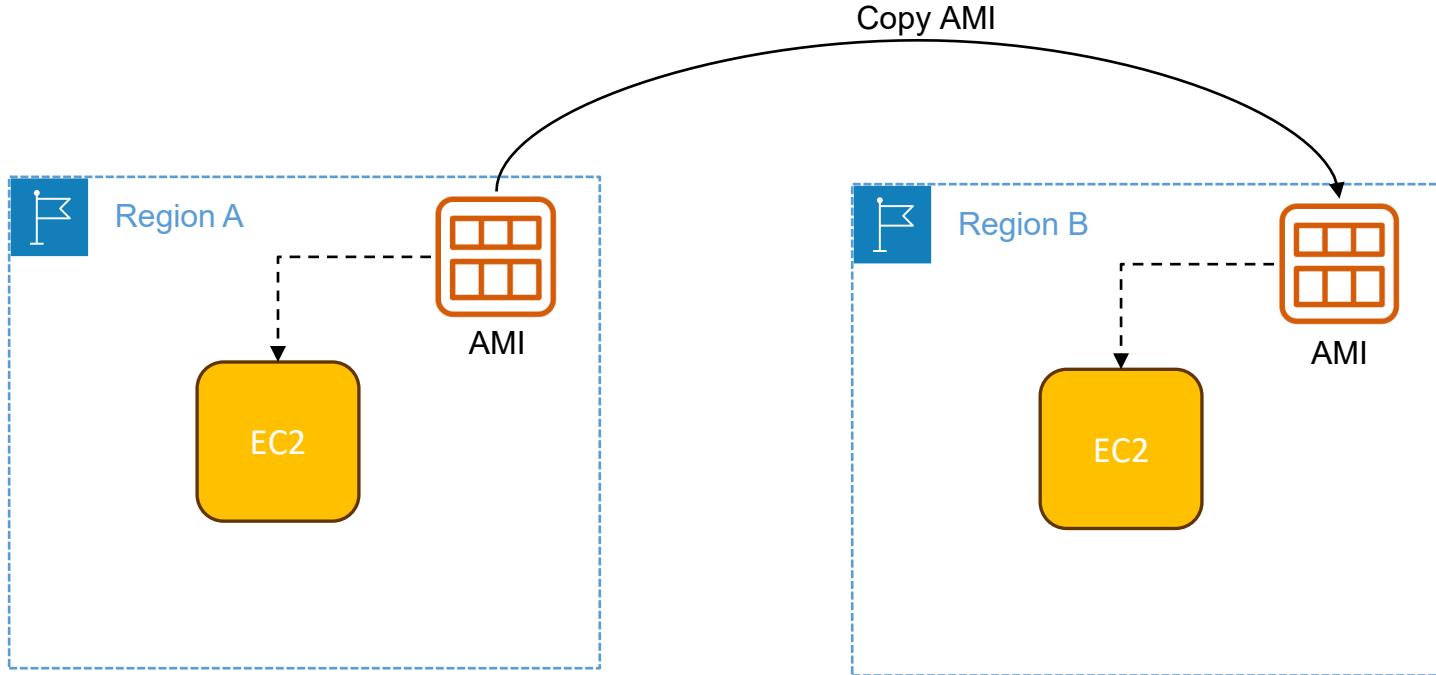


Amazon Linux 2023 AMI  
ami-0e53db6fd757e38c7 (64-bit (x86), uefi-preferred)  
Amazon Linux 2023 is a modern, general purpose Linux distribution designed for the AWS cloud environment to develop and run your cloud applications.  
Platform: amazon      Root device type: ebs

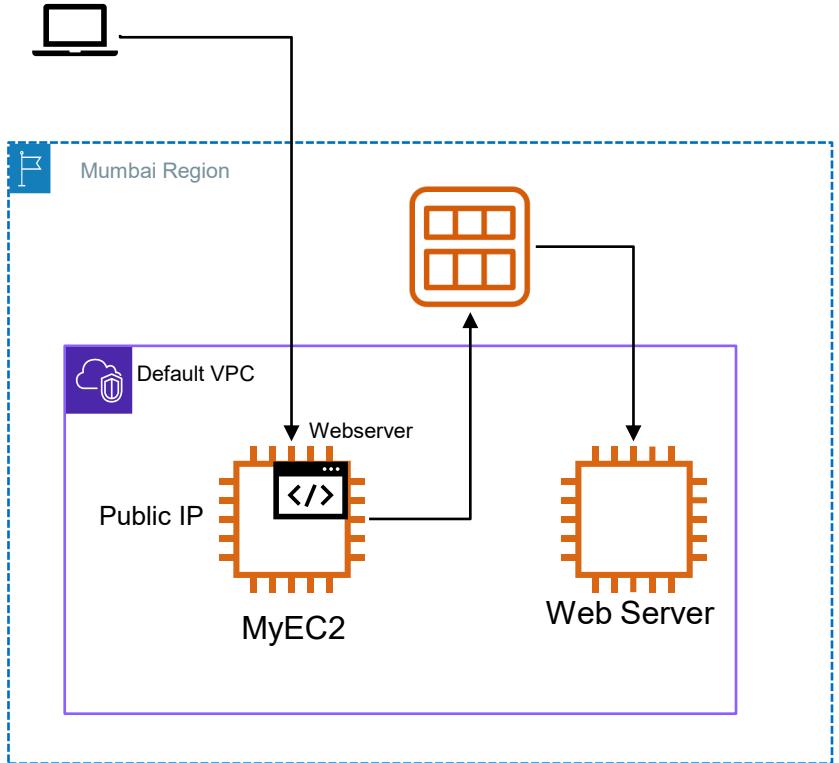
Free tier eligible

Verified provider

# AMIs are regional



# Exercise – EC2 AMI



## High level steps

- 1 Launch EC2 Linux instance in a default VPC. Allow SSH and HTTP port access.
- 2 Connect over SSH
- 3 Install a HTTP web server and create a web page. Verify that web page can be accessed.
- 4 Stop the instance
- 5 Create an AMI
- 6 Launch a new EC2 instance from the AMI you created
- 7 Verify that Web page can be accessed
- 8 Terminate both the instances

# Exercise : EC2 AMI

1

## Launch EC2 (Linux) instance in a default VPC (Mumbai Region)

- a) Go to AWS Console -> Select Mumbai region
- b) Go to EC2 Service -> EC2 Dashboard -> Launch Instances
- c) Name: EC2-A
- d) Select Application and OS Images (Amazon Machine Image): Amazon Linux (default)
- e) Select instance type: t2.micro (default)
- f) Select key pair : *Your key-pair that you had created earlier in pre-requisites*
- g) Network settings -> Default VPC
- h) Make sure Auto-Assign Public IP is enabled
- i) Firewall -> Create security group
- j) Allow SSH traffic from -> Select My IP [Type-> SSH, port Range-> 22, source type -> My IP]
- k) Allow HTTP traffic from the internet [Type-> HTTP, port Range-> 80, source type -> Anywhere (0.0.0.0/0)]
- l) Configure Storage -> 8GiB, gp3 (default)
- m) Launch Instance

2

SSH into EC2 instance over EC2 *Public IP* using Putty or terminal with user *ec2-user*

# Exercise : EC2 AMI

## 3 Install http web server and verify

- a) Run command: `sudo yum install httpd -y`
- b) Run command: `sudo systemctl start httpd.service`
- c) Run command: `sudo systemctl enable httpd.service`
- d) Create `/var/www/html/index.html` file (use vi editor) and add some HTML content e.g.<h1>Hello World</h1>
- e) Open browser from your workstation and access **EC2 Public IP**
- f) You should see a web page which displays “Hello World”. If there is connection issue, then check EC2 security group and check if port 80 is open for 0.0.0.0/0 (anywhere)

## 4 Stop the instance

- a) EC2 console -> Select your instance -> Instance State -> Stop instance

## 5 Create an AMI

- a) EC2 console -> Select your instance -> Actions -> Image and templates -> Create image
- b) Provide Image name and description -> Create image
- c) Wait for image to be created. Check in EC2 console -> Left menu -> AMIs -> Owned by me

## 6 Launch a new EC2 instance using your AMI

- a) Repeat Step 1 but this time select your own AMI instead of Amazon Linux. Application and OS images -> My AMIs -> Select AMI that your AMI
- b) Select existing key pair, default VPC, existing security group and launch the instance
- c) Wait for instance to be Running

# Exercise : EC2 AMI

7

## Verify web page for a new instance

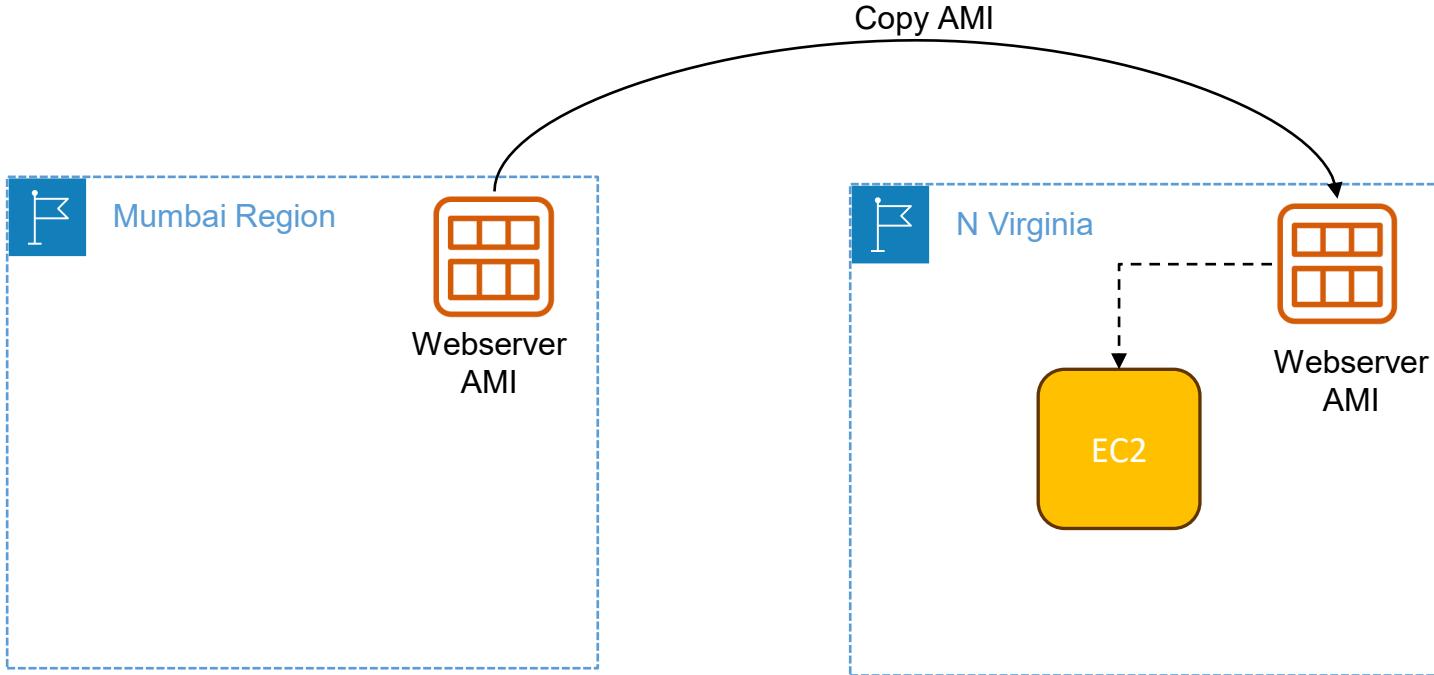
- a) Open browser from your workstation and access new EC2 **Public IP**
- b) You should see a web page which displays “Hello World”.

8

## Terminate both the instances

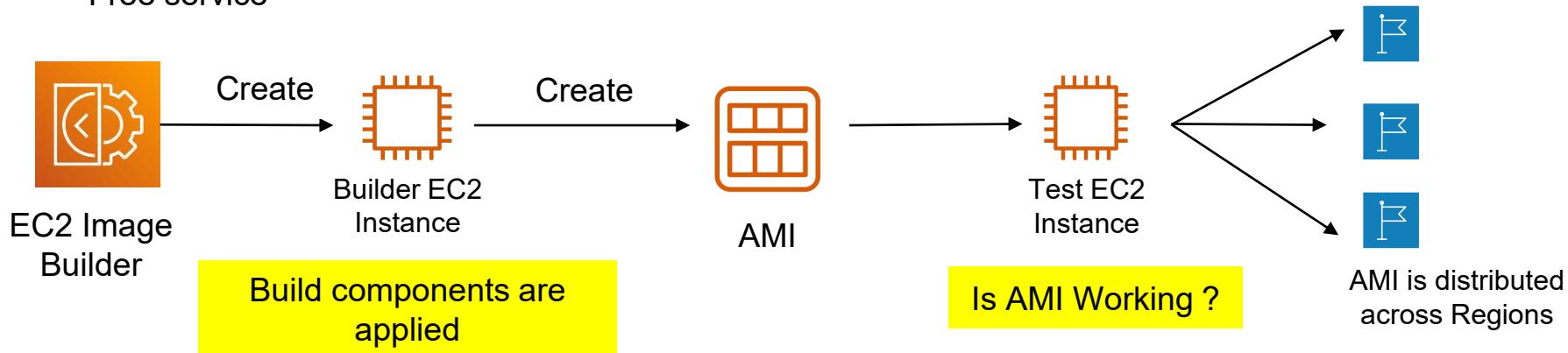
- a) EC2 console -> Select your instance -> Instance State -> Terminate instance

# Assignment – Copy AMI across region



# EC2 Image Builder

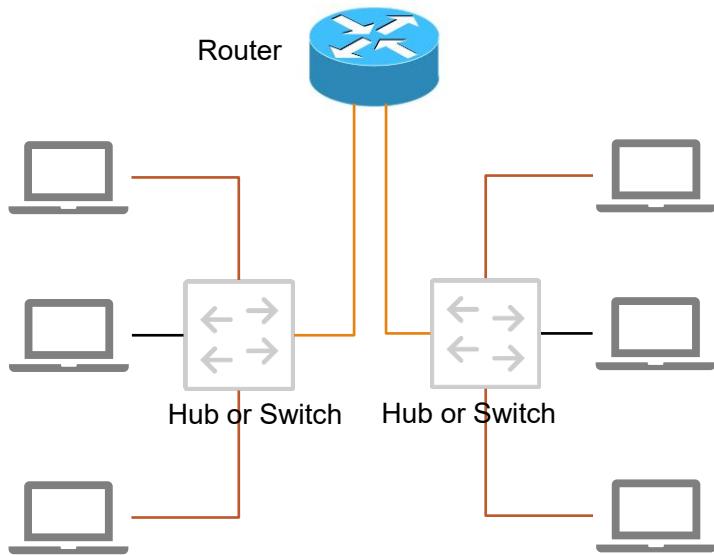
- Automate the creation, management, and deployment of AMIs
- Configure pipelines to automate updates and system patching for the images.
- Can be run on a schedule basis (weekly, whenever packages are updated).
- Works across AWS regions and AWS Accounts
- Free service



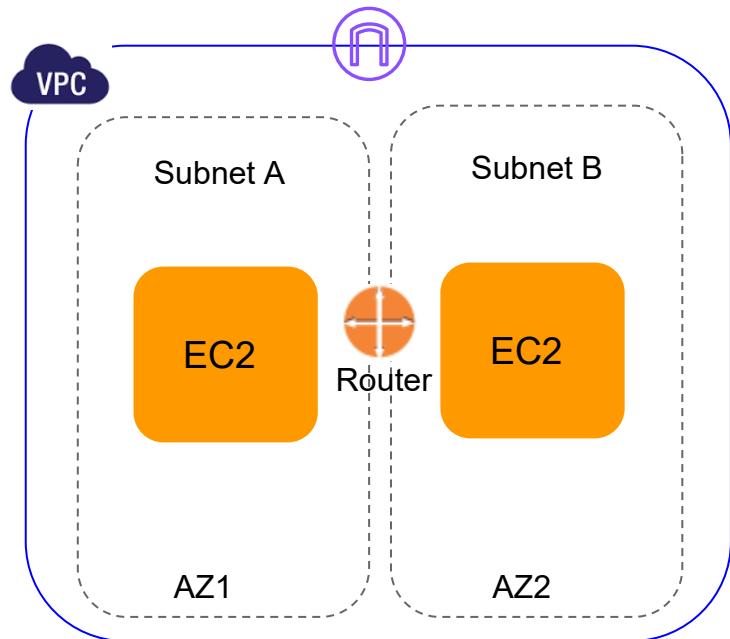
# Virtual Private Cloud (VPC)

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud.

**Physical Network**

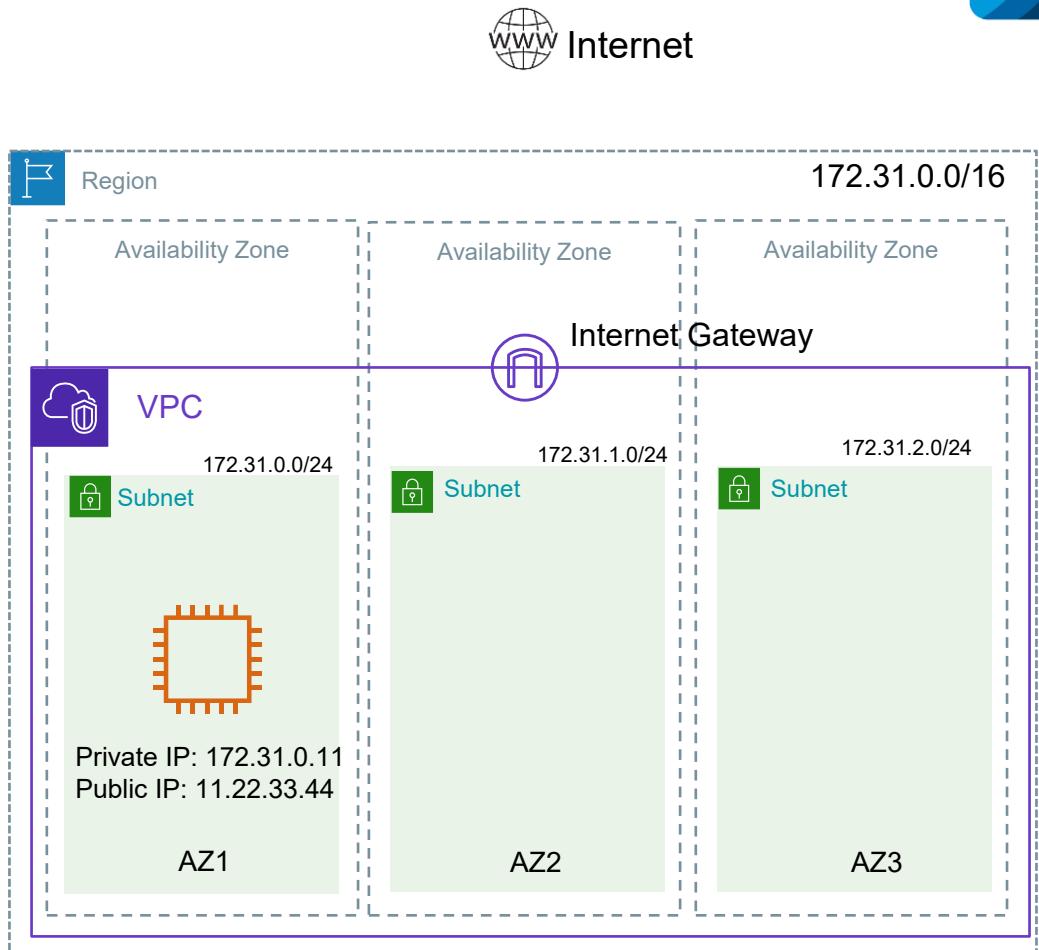


**AWS VPC**



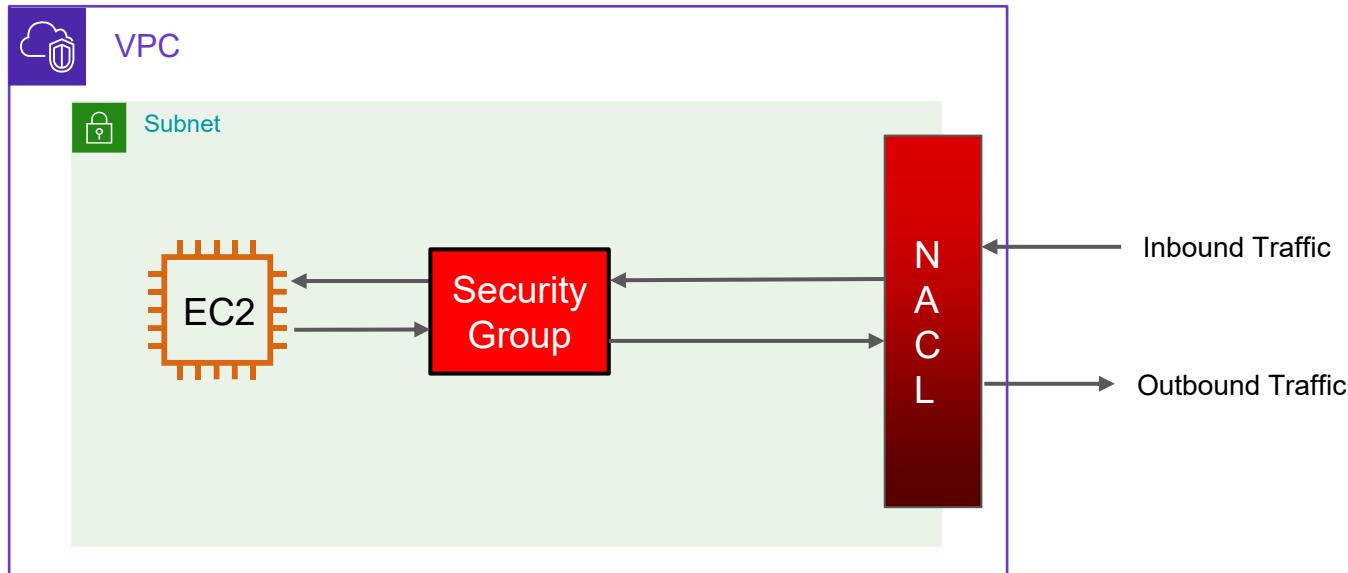
# Default VPC

- AWS Creates Default VPC in each AWS region
- If you don't select any VPC then this default VPC is used to launch an EC2 instance
- EC2 instance receives the Private IP from the subnet CIDR range and Public IP from the Amazon's pool of Public IPs



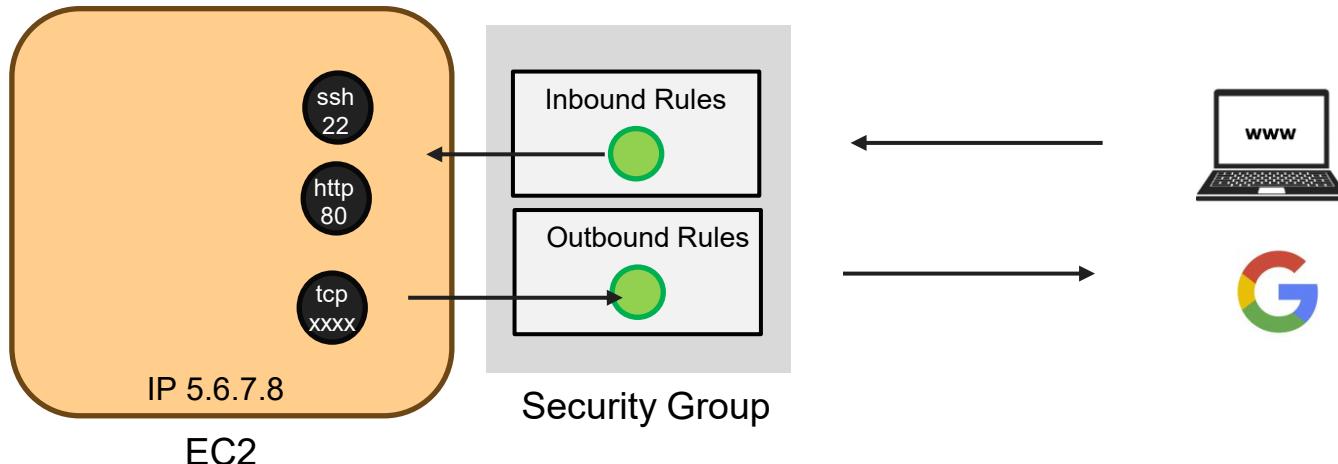
# Security Group - Firewall for EC2 instance

- Security Groups
- Network Access Control List (NACL)



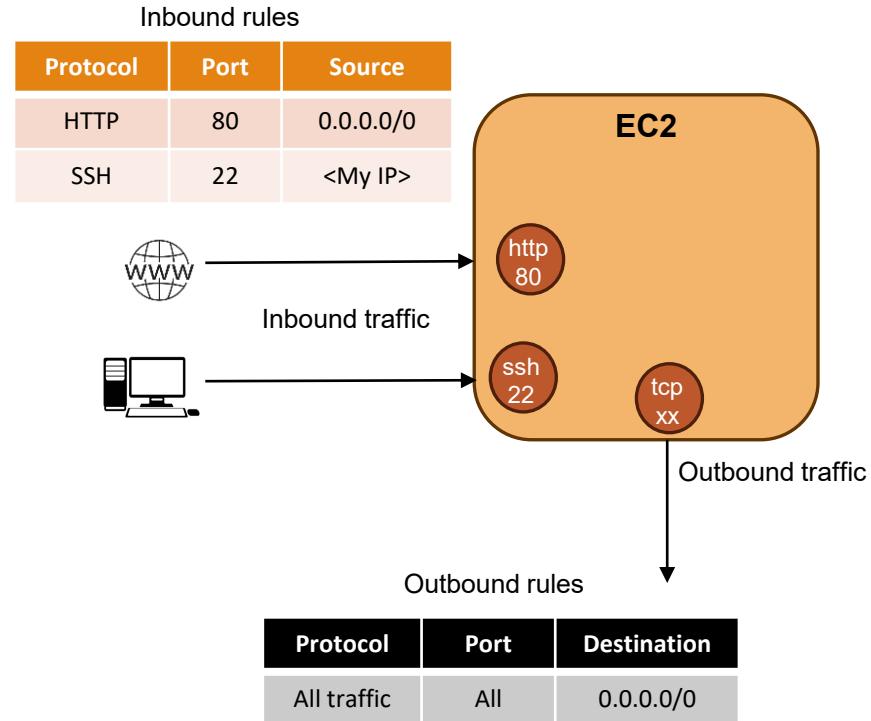
# Security Group

- Security Groups are most basic, native and important firewall for EC2 instances
- Security group has Inbound and Outbound rules
- Security group has only ALLOW rules. Does not support DENY/Block rules.
- Default Security group in each VPC
- Authorises traffic for both IPv4 and IPv6 traffic
- Security groups are stateful – return traffic is automatically allowed



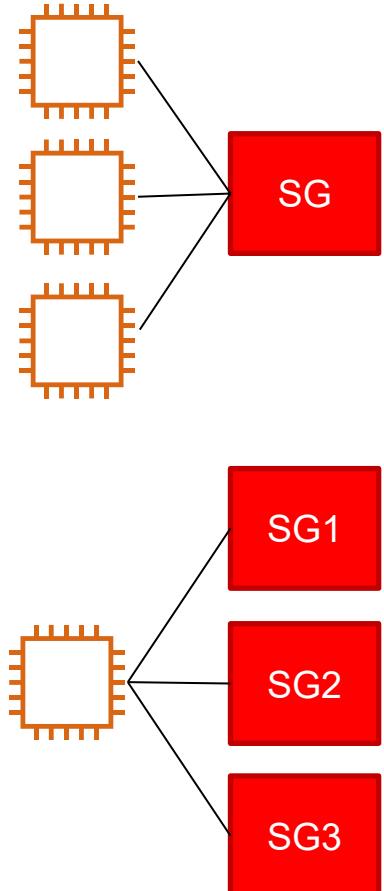
# Typical ports – good to know

- **22** = SSH (Secure Shell) for logging into a Linux server instance
- **3389** = RDP (Remote Desktop Protocol) for logging into a Windows server instance
- **21** = FTP (File Transfer Protocol) for uploading files to a file share
- **22** = SFTP (Secure File Transfer Protocol) for uploading files securely (over SSH)
- **80** = HTTP for accessing websites or web applications
- **443** = HTTPS for TLS secured websites



# Security Groups - Summary

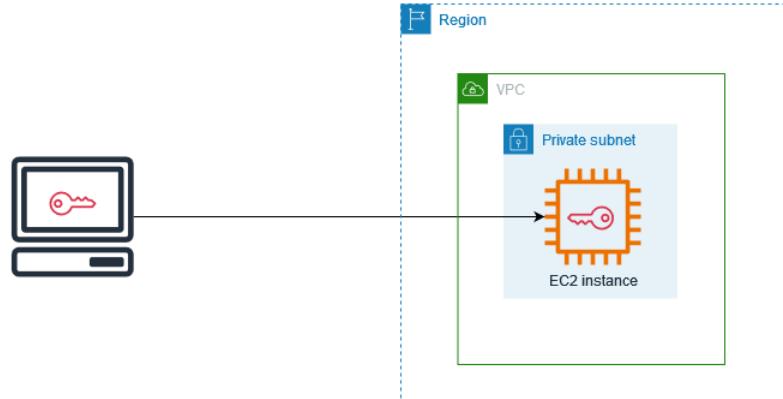
- Security group rules enable you to filter traffic based on protocols and port numbers.
- Single Security Group can be attached to multiple instances
- Single instance can have up to 16 Security groups with maximum 1000 rules\*
- All inbound traffic is blocked by default
- By default, security groups contain outbound rules that allow all outbound traffic.
- Authorises traffic for both IPv4 and IPv6 traffic
- Security groups are stateful - if you send a request from your instance, the response traffic for that request is allowed to flow in regardless of inbound security group rules.
- You can add and remove rules at any time. Your changes are automatically applied to the instances that are associated with the security group.



\*limits may change in the future

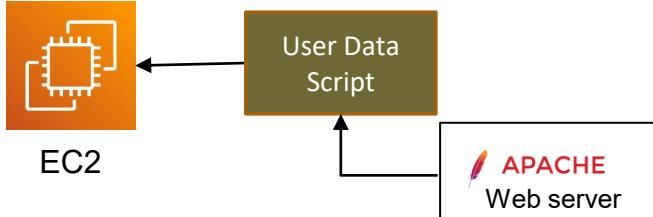
# SSH Key-pair

- A key pair consists of a public key and a private key
- For Linux instances, the private key allows you to securely SSH into your instance
- For Windows instances, the private key is required to decrypt the administrator password.
- Key pairs are regional
- We can use same key-pair to launch more than one instances. All those instances will have same security credentials.



# EC2 User data

- It is a **bootstrap** script to automatically configure the instance at the time of first launch
- EC2 User data script will run only **once** when instance first starts.
- It is used to automate boot tasks such as :
  - Install updates
  - Install software
  - Download common files from the Internet
- EC2 User data scripts run with a root user.



## Sample User data Script

```

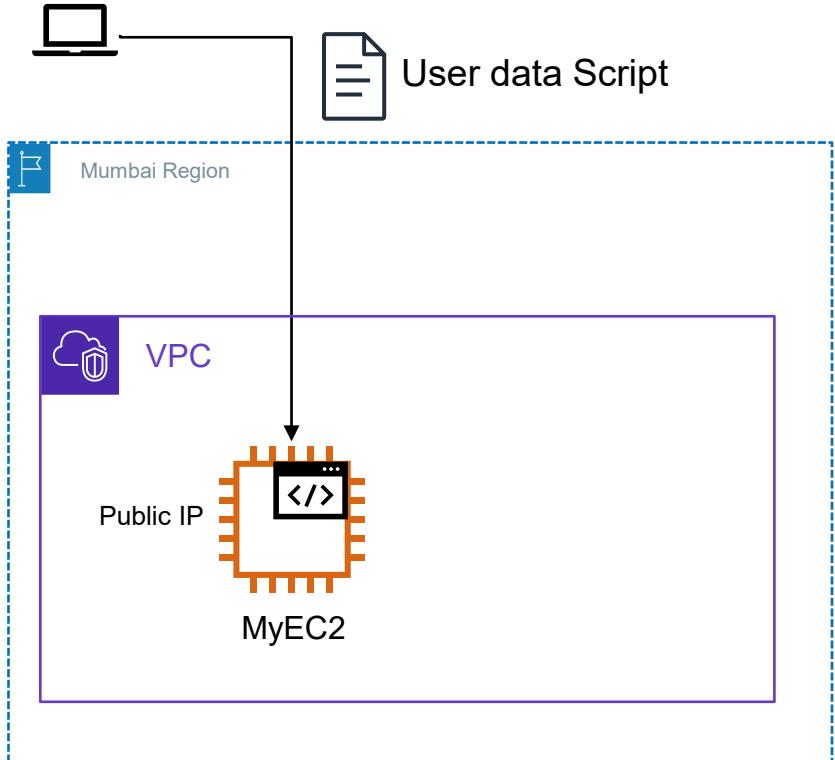
#!/bin/bash

yum update -y
yum install -y httpd

systemctl start httpd
systemctl enable httpd

echo "<h1>Hello World Hello
World - This time using User
data script.<h1>" >
/var/www/html/index.html
  
```

# Exercise – EC2 User data script



## High level steps

- 1 Launch EC2 Linux instance in a default VPC. Allow only HTTP port access in the Security Group. Use User data script to install and configure a web server.
- 2 Verify that Web page can be accessed
- 3 Terminate the instance

# EC2 User data script

```
#!/bin/bash

yum update -y
yum install -y httpd

systemctl start httpd
systemctl enable httpd

echo "<h1>Hello World Hello World - This time using User data
script.<h1>" > /var/www/html/index.html
```

# Exercise – EC2 User Data script

1

## Launch EC2 (Linux) instance in a default VPC (Mumbai Region)

- a) Go to AWS Console -> Select Mumbai region
- b) Go to EC2 Service -> EC2 Dashboard -> Launch Instances
- c) Name: EC2-A
- d) Select Application and OS Images (Amazon Machine Image): Amazon Linux (default)
- e) Select instance type: t2.micro (default)
- f) Select key pair : *Your key-pair that you had created earlier in pre-requisites*
- g) Network settings -> Default VPC
- h) Make sure Auto-Assign Public IP is enabled
- i) Firewall -> Create security group
- j) Allow HTTP traffic from the internet [Type-> HTTP, port Range-> 80, source type -> Anywhere (0.0.0.0/0)]
- k) Configure Storage -> 8GiB, gp3 (default)
- l) Advance Details -> Scroll down -> In the UserData script paste the script from the earlier page.
- m) Launch Instance

2

## Verify that Web page can be accessed

- a) Open browser from your workstation and access EC2 **Public IP**. Should see your web page.

3

## Terminate the instance

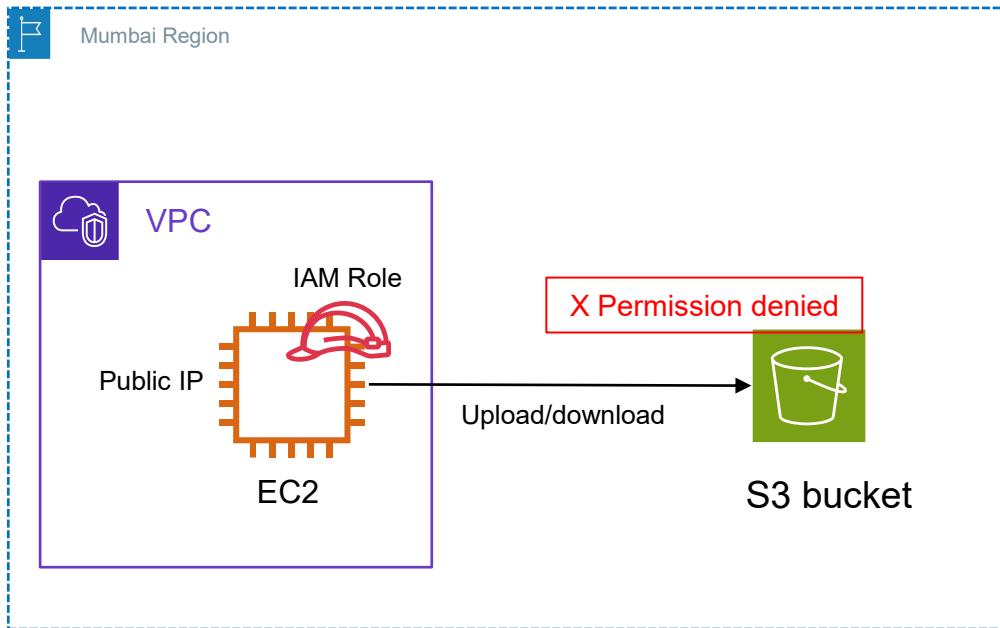
- a) EC2 console -> Select your instance -> Instance State -> Terminate instance

# EC2 Instance Connect

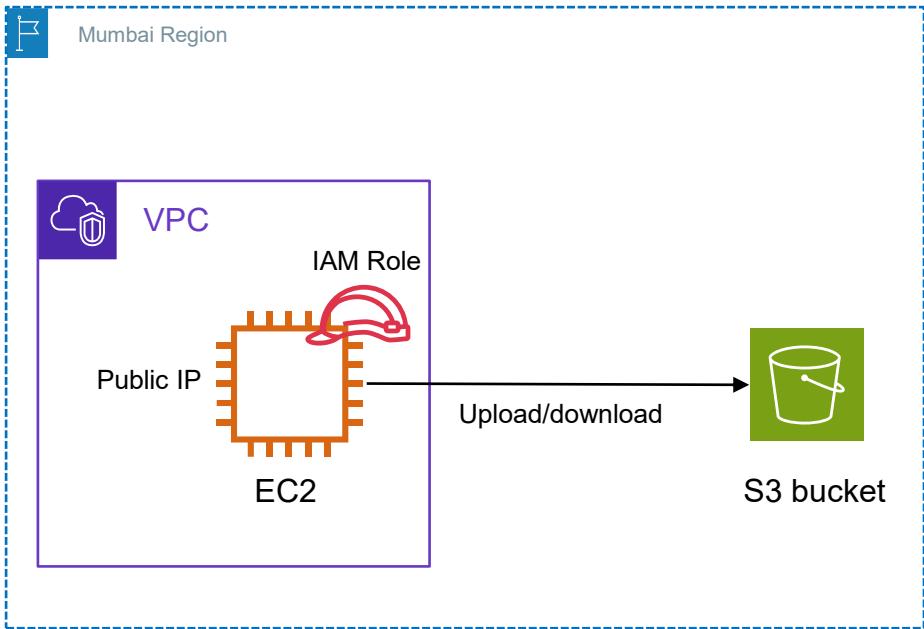
- Amazon EC2 Instance Connect enables system administrators to publish one-time use SSH public keys to EC2, providing users a simple and secure way to connect to their instances.
- Connect to your EC2 instance within your browser.
- No need to use your key file that was downloaded.
- The “magic” is that 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 in Security group.

# IAM Role for EC2

- IAM Role provides the IAM permissions to EC2 instance to access other AWS services
- We can change the IAM Role or its permissions as required
- It's a good practice to assign IAM role than using the long term IAM credentials
- By default, EC2 does not have any IAM Role



# Exercise: EC2 IAM Role



## Exercise steps

- 1 Launch EC2 Linux instance in a default VPC and connect to it over SSH.
- 2 Create S3 bucket in the same region and upload some sample text or image file.
- 3 From EC2 terminal, try to download file from S3 using AWS CLI command – Access denied.
- 4 Go to AWS IAM and create IAM role for EC2. Attach S3 full permissions policy to the role.
- 5 Associate this new role to EC2 instance
- 6 Try again to download file from S3 – should be successful.
- 7 Terminate EC2 instance. Optionally delete S3 bucket.

# Exercise: EC2 IAM Role

Command to download file from S3:

```
$aws s3 cp s3://bucket_name/file_path .
```

\*Note the dot at the end of the command. It's used to copy the file in the current directly of the EC2.

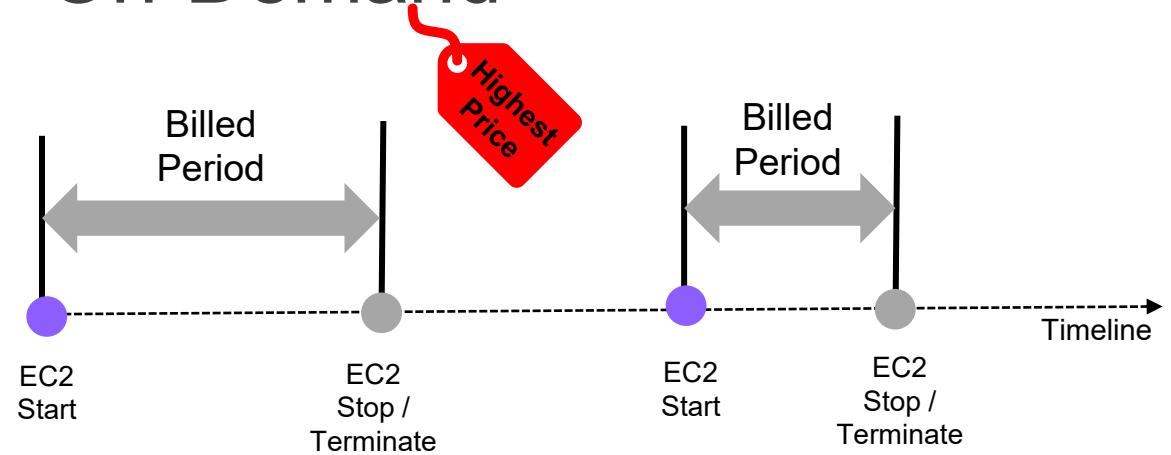
Example: \$aws s3 cp s3://aws-with-chetan/sample.txt .

# EC2 Purchasing options

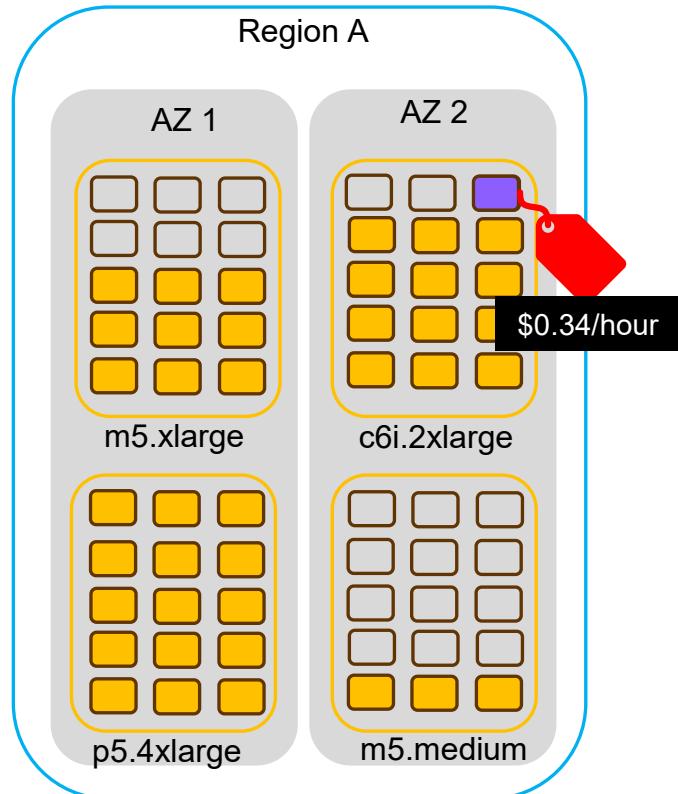
1. On-Demand
2. Spot
3. Reserved
4. Savings Plan



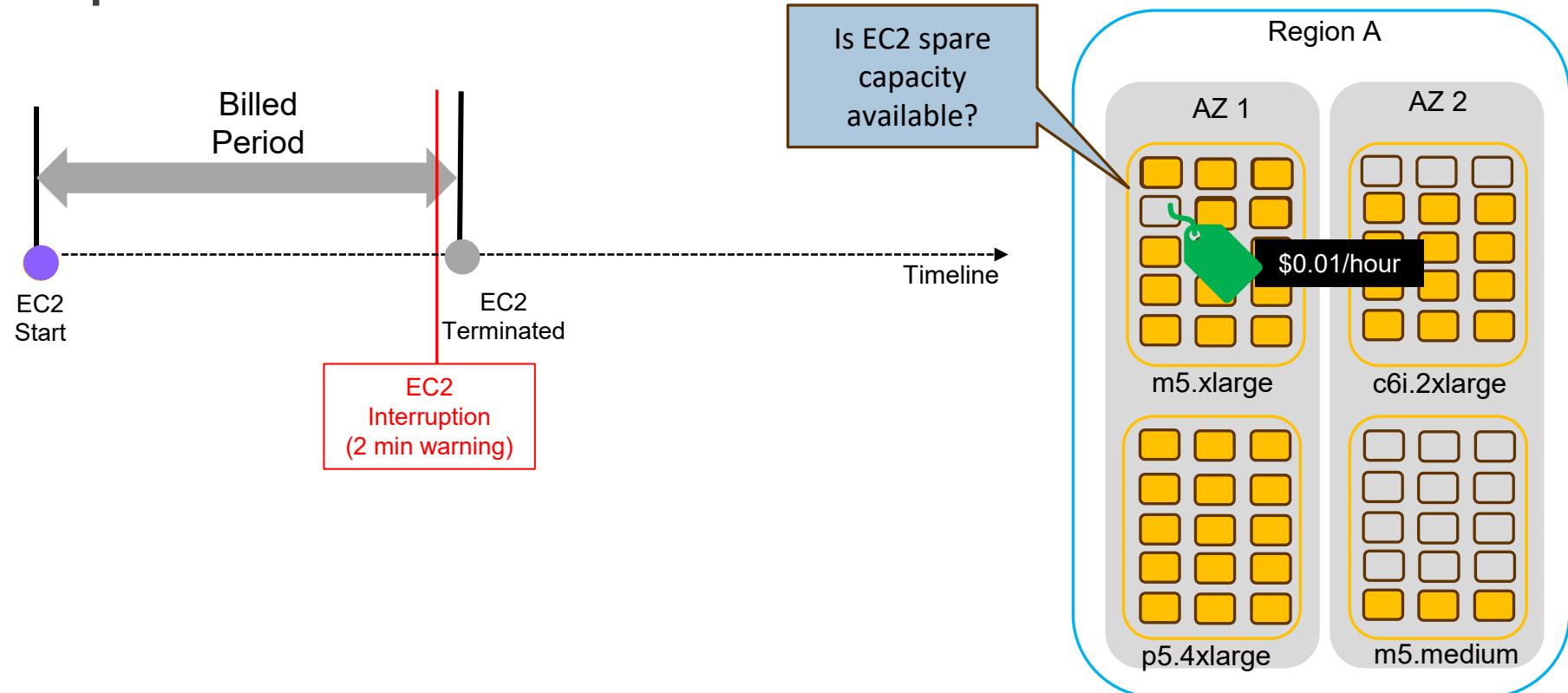
# On-Demand



- EC2 capacity becomes available instantly\*
- Start and Stop as required.
- No long-term commitment, no discounts
- Flexible
- Great for unpredictable, spiky, stateful workloads



# Spot



# Spot



Try different instance type, size, AZ, region

Billed Period



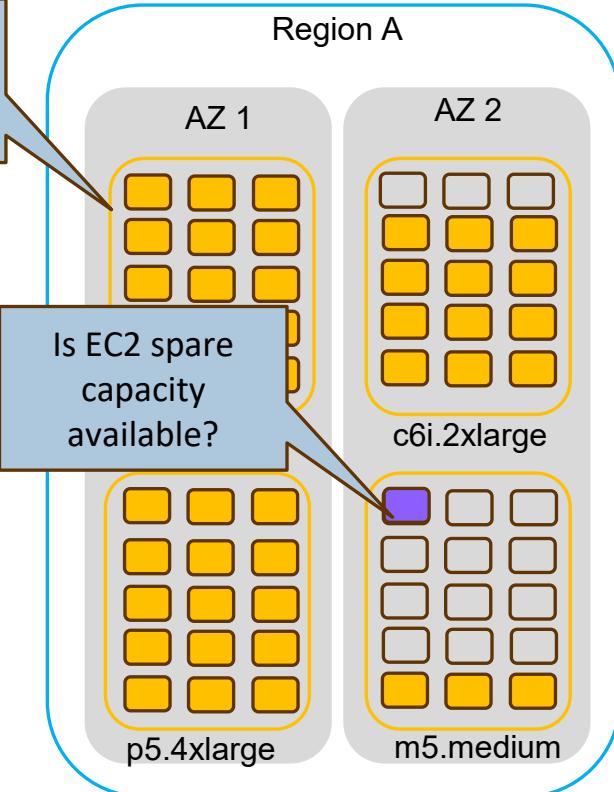
Is EC2 spare capacity available?

EC2 Start

EC2 Start

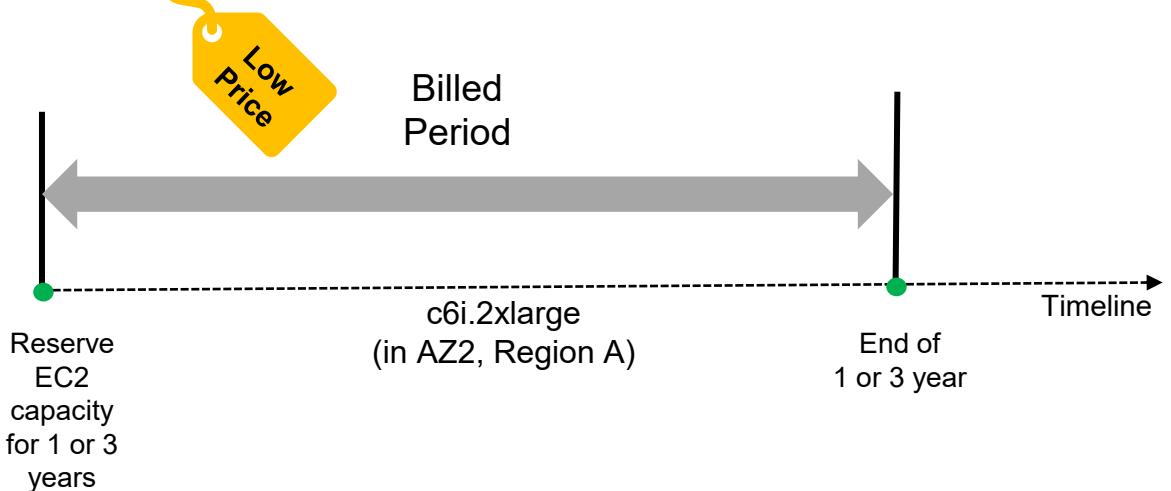
Timeline

Terminated by user or auto terminated after 2 min interruption warning

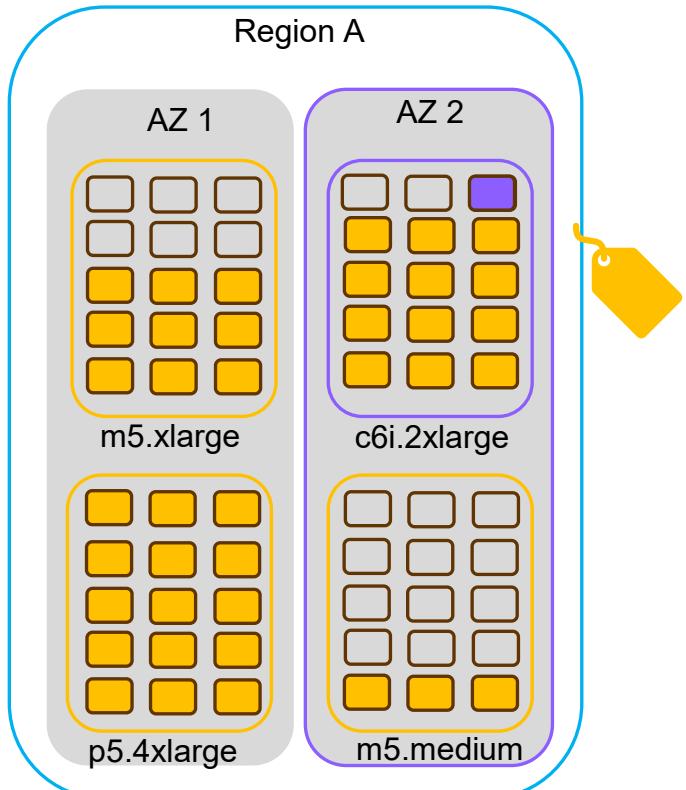


- EC2 becomes available if there is spare capacity
- AWS can claim the capacity back with 2 min warning
- Try changing instance type, size, AZ, Region
- Try at different time of the day
- Most cost effective (up to 90% cheaper than On-demand)
- Great for **fault tolerant, non time sensitive** and **stateless** workloads

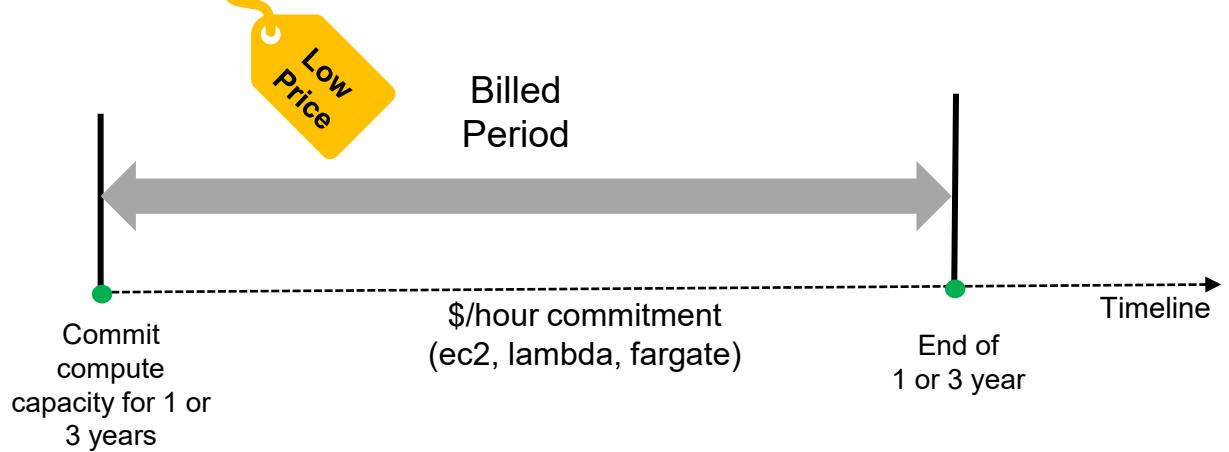
# Reserved Instances (RI)



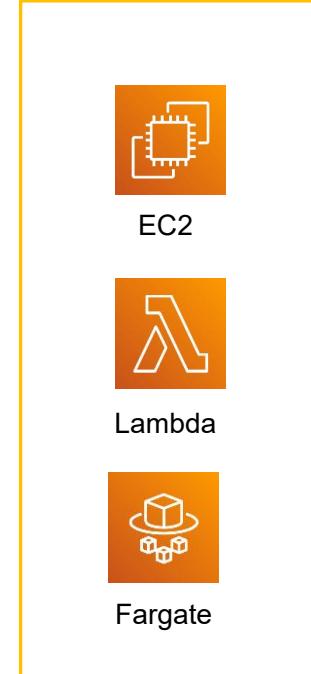
- EC2 capacity is reserved for 1 or 3 year term
- Pay whether you use the capacity or not
- Long-term commitment and discounts up to 72%
- **Standard RI** – Fixed instance type, AZ (max discount)
- **Convertible RI** – Can change instance type, AZ or Region
- Can pay **Full upfront, Partial upfront or No upfront**
- Great for steady state and predictable workloads



# Savings Plan (SP)

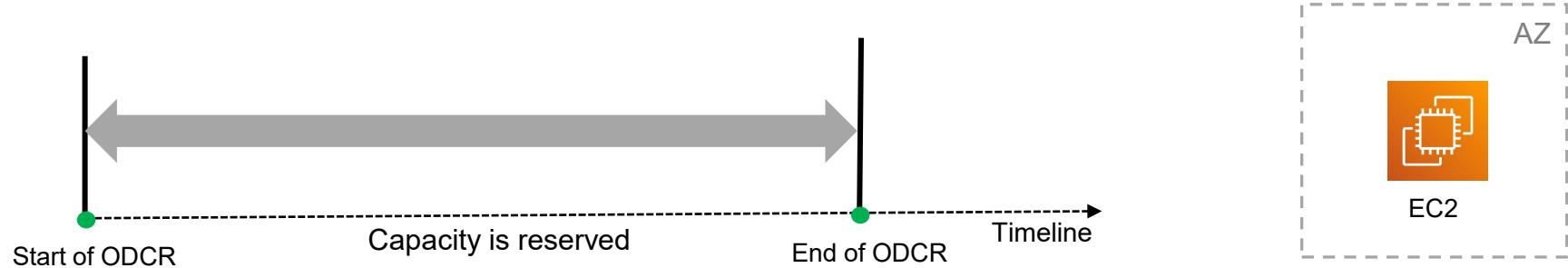


- Compute capacity (\$/hour) is reserved for 1 or 3 year term
- Pay whether you use the capacity or not
- Long-term commitment and discounts up to 72%
- **Compute SP** – Usage across any EC2, Lambda and Fargate in **any** region
- **EC2 SP** – Usage for EC2 instance family in selected Region (max discount)
- Great for steady state and predictable workloads



*AWS recommends to use Savings Plan over Reserved Instances*

# EC2 On-demand Capacity Reservation (ODCR)



- Reserve EC2 capacity for EC2 instances in a specific Availability Zone for any duration.
- Create ODCR anytime **without** entering a 1-year or 3-year term commitment.
- Cancel the ODCR anytime to release the capacity and stop incurring charges.
- For ODCR – specify AZ, number of instances with instance attributes – type, size, platform, tenancy.
- By default, charged at EC2 On-Demand rate. RI and SP discounts applies on matching capacity.
- Suitable for business-critical workloads that require a capacity assurance.

# EC2 Pricing options comparison

On-Demand	Reserved Instances	Savings Plan	Spot
<ul style="list-style-type: none"> <li><b>No commitment</b></li> <li>Pay by second</li> <li>Start or Stop/Terminate at any time</li> <li>Billed only when instance is in Running state</li> <li><b>Highest cost</b></li> <li>Suitable for unpredictable, spiky, stateful workloads</li> </ul>	<ul style="list-style-type: none"> <li><b>1-year or 3-year commitment for EC2 instance type/hour</b></li> <li>Up to 72% discount</li> <li>3 payment options: Full upfront Partial Upfront No upfront</li> <li>Pay even if not using the capacity</li> <li>Standard RI</li> <li>Convertible RI</li> <li>Suitable for steady state and predictable workloads</li> </ul>	<ul style="list-style-type: none"> <li><b>1-year or 3-year commitment for \$/hour</b></li> <li>Up to 72% discount</li> <li>3 payment options: Full upfront Partial Upfront No upfront</li> <li>EC2 Savings Plan</li> <li>Compute Savings plan</li> <li>Suitable for steady state and predictable workloads</li> </ul>	<ul style="list-style-type: none"> <li>No commitment</li> <li>Up to 90% discount</li> <li>Instance can be terminated with 2-minute warning</li> <li><b>Lowest cost</b></li> <li>Suitable for fault tolerant, non-time sensitive and stateless workloads</li> </ul>

# Pricing example with different EC2 pricing models

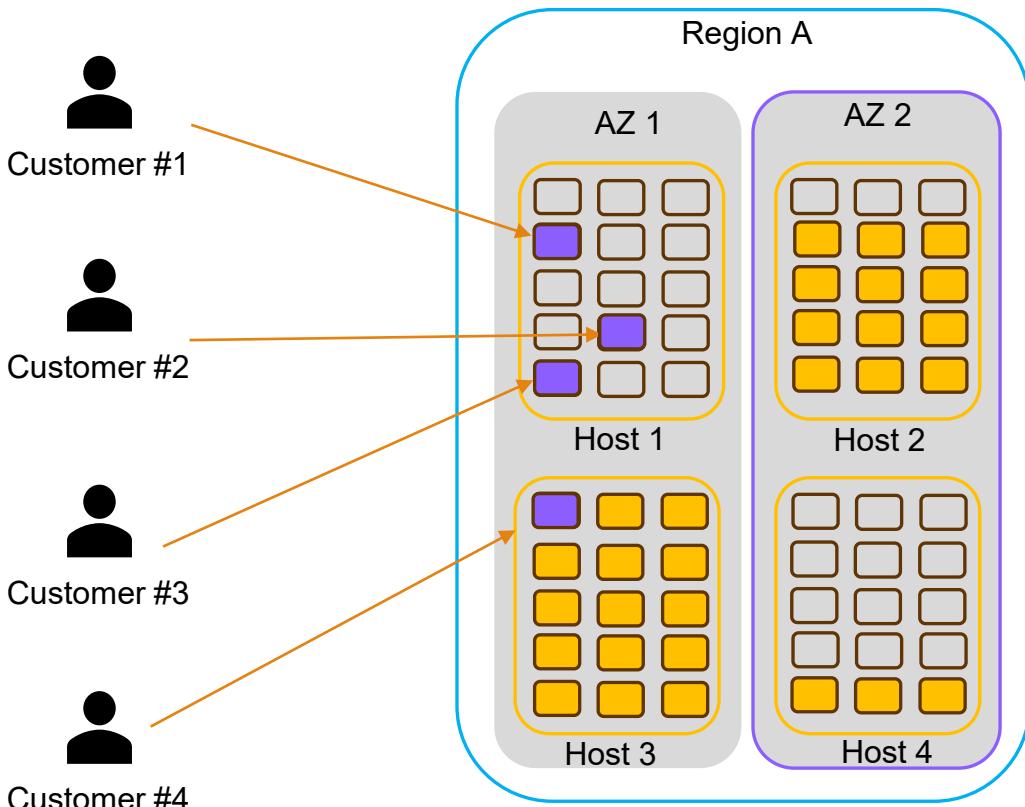
m5.large (2vcpu, 8GiB RAM) in N. Virginia region

Pricing model	Price (Per Hour)
On-Demand	<b>\$0.096</b>
Spot Instance (Spot Price)	\$0.038 - \$0.04 (up to 60% off as per current trend)
Reserved Instance (1 year)	\$0.060 (No Upfront) - \$0.057 (All Upfront)
Reserved Instance (3 years)	\$0.041 (No Upfront) - \$0.036 (All Upfront)
Reserved Convertible Instance (1 year)	\$0.071 (No Upfront) - \$0.066 (All Upfront)
EC2 Savings Plan (1 year)	\$0.060 (No Upfront) - \$0.056 (All Upfront)
EC2 Savings Plan (3 year)	\$0.041 (No Upfront) - \$0.036 (All Upfront)
Compute Savings Plan (1 year)	\$0.071 (No Upfront) - \$0.066 (All Upfront)
Compute Savings Plan (3 year)	\$0.049 (No Upfront) - \$0.044 (All Upfront)
Dedicated Instance (Instance/hr + \$2/hr/region)	\$0.102 + \$2/hr/region = \$2.102 (On-Demand Price)
Dedicated Host (m5 host) – (96 vCPU/48 core)	\$5.069 (On-Demand Price)
On-Demand Capacity Reservations (ODCR)	\$0.096 (On-Demand Price)

# EC2 Tenancy – Shared vs Dedicated

## Shared Tenancy

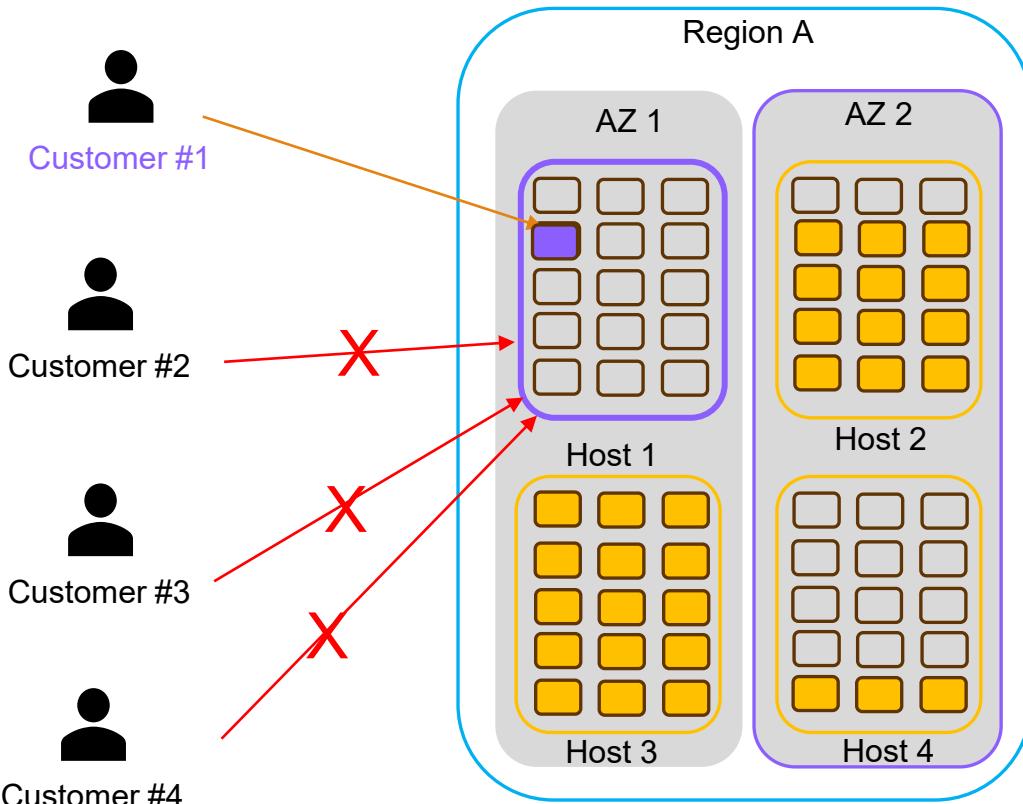
- Underlying physical host is shared across different AWS customers (AWS account)
- This is a default tenancy for VPC and EC2



# EC2 Tenancy – Shared vs Dedicated

## Dedicated Tenancy

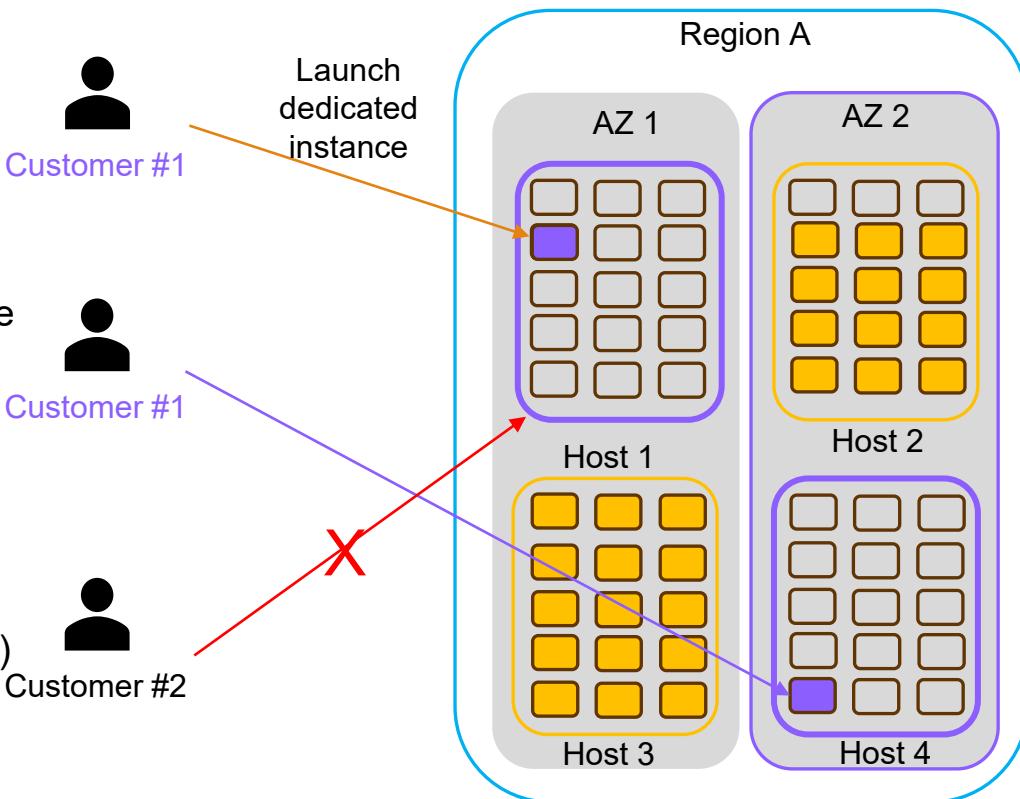
- Underlying physical host is dedicated for a single customer (AWS account)
- Used for Regulatory and Compliance requirements
- On-demand, Spot and Reserved pricing is available
- There are 2 options for Dedicated Tenancy:
  - **Dedicated Instances**
  - **Dedicated Hosts**



# EC2 Dedicated Instances

## Dedicated Instances

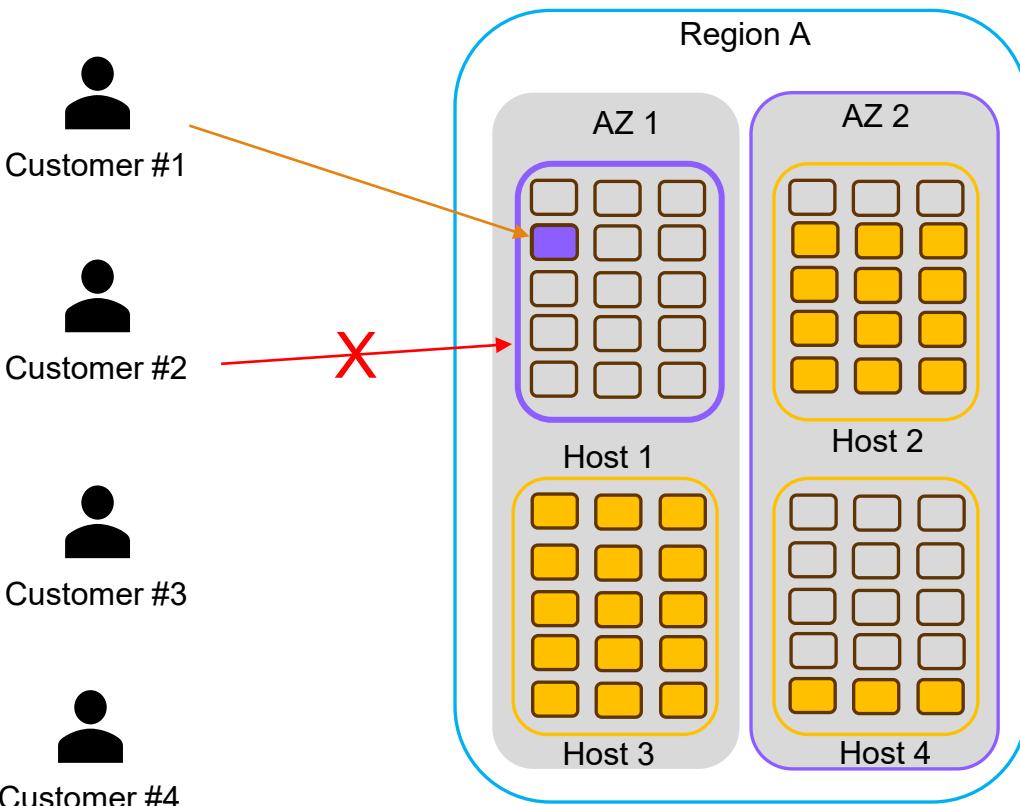
- Instance is launched on a host which is dedicated for the customer, but customer do not decide on which host to launch the instance
- Do not get access, placement choice or visibility of the underlying host
- There are 2 pricing elements:
  - EC2 per hour running charge
  - Dedicated host per hour charge (\$2/hour)
  - On-demand, Spot, Reserved



# EC2 Dedicated Host

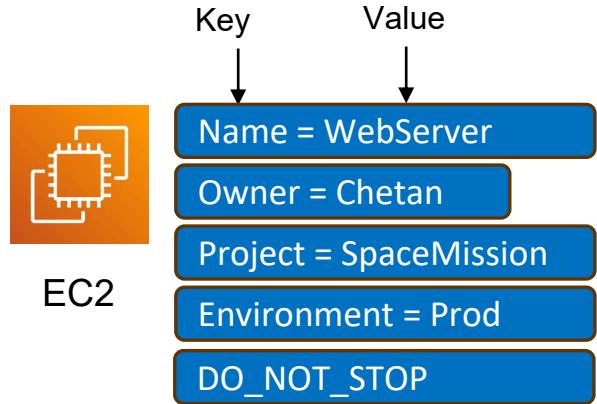
## Dedicated Host

- Allow to use existing per-socket, per-core, or software licenses tied to the physical host
- Host affinity where you can choose this host to launch EC2 instances
- Pricing
  - Per host instead of per instance billing
  - On-demand, Reserved, Savings Plan



# EC2 Instance Tags

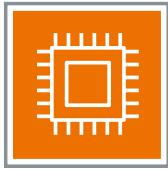
- Tags are in the form of **Key Value Pairs**.
- Can be useful to filter resources while queries AWS resources.
- Are useful when you use AWS deployment services like Code deploy, Code pipeline.
- Can use tags in IAM policies to restrict access to users for a particular instance(s).
- Very useful for cost allocation e.g. per project or environments etc.



Tags		Manage tags
Key	Value	< 1 >
Environmet	demo	
Name	Webserver	
Owner	Chetan	

# Amazon EC2 - Summary

- **EC2** - Virtual Machine in the AWS cloud hosted in an AZ within the Region
- **EC2 Configuration options** - Instance Type/Size (CPU + RAM) + AMI (OS) + Storage (EBS/Instance store) + VPC + Security Groups + SSH key-pair (+ IAM Role, User data, Tags)
- **EC2 Block Storage options** – Persistent storage - Elastic Block Storage (EBS), Non persistent storage - Instance Store
- **EC2 Security groups** – Contains inbound rules and outbound rules (Protocol, Port, IP range), Stateful
- **EC2 initialization** – EC2 User Data scripts
- **Login Options:** SSH, RDP (*there are more options to connect to EC2 & we will cover them later*)
- **Purchasing Options** - On-Demand, Spot, Reserved, Savings Plan
- **Capacity Reservation** (without commitment) - On-demand capacity reservation (ODCR)
- **Tenancy options** – Shared, Dedicated Instances, Dedicated Host
- **Tags** – Name-value key pairs, useful for filtering, deployment, User IAM permissions, cost allocation etc.

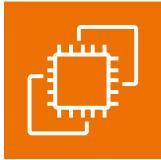


# AWS compute services

Running containers - ECS, EKS, Fargate  
Serverless - Lambda

# AWS compute choices

Amazon EC2



EC2

Other AWS Compute services

Containers



Elastic Container  
Service (ECS)



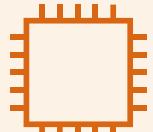
Elastic Kubernetes  
Service (EKS)

Serverless



Lambda

Data plane



EC2

Serverless



Fargate

# Why containers?

</> Code



Libs/dependencies



Configurations

Libc 5  
Glib 2.81.0  
OpenSSL 3.2.0  
+more

Libc 5  
Glib 2.81.0  
**OpenSSL 3.1.0**  
+more



Developer's  
workstation

Push



Server

# Why containers?

</> Code

 Libs/dependencies

 Configurations



Push



Developer's  
workstation

Server

# Containers

*Lightweight virtualization that allows to run applications and their dependencies in resource-isolated processes*

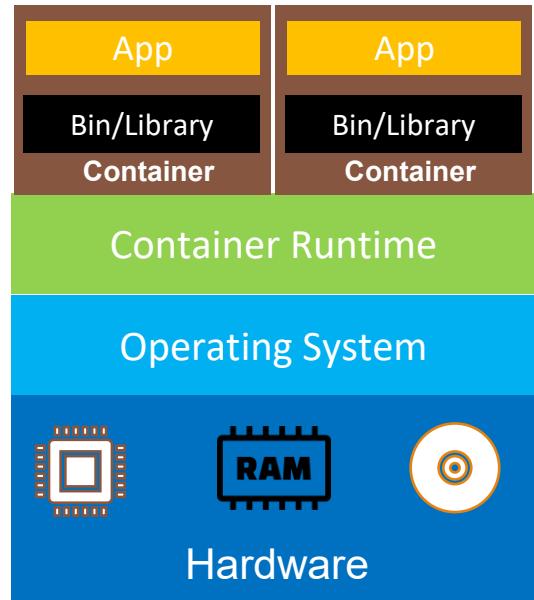
## Key benefits

**Isolated** - Isolated filesystem, process space, network, environment

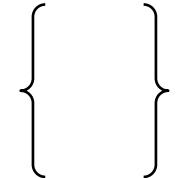
**Lightweight** - No full OS overhead, minimal kernel, fast

**Consistency** – Contains all dependencies, same everywhere

**Portability** – Can be easily packaged, shipped and run on any platform (local machine, server, cloud)



# What makes containers work?



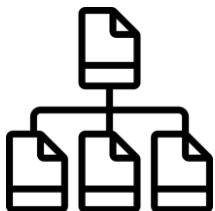
Namespaces

Provide isolation for running applications by limiting what they can see and access.



Control Groups  
(cgroups)

Limit and monitor the resource usage (CPU, memory, I/O) of containers.



Union Filesystem

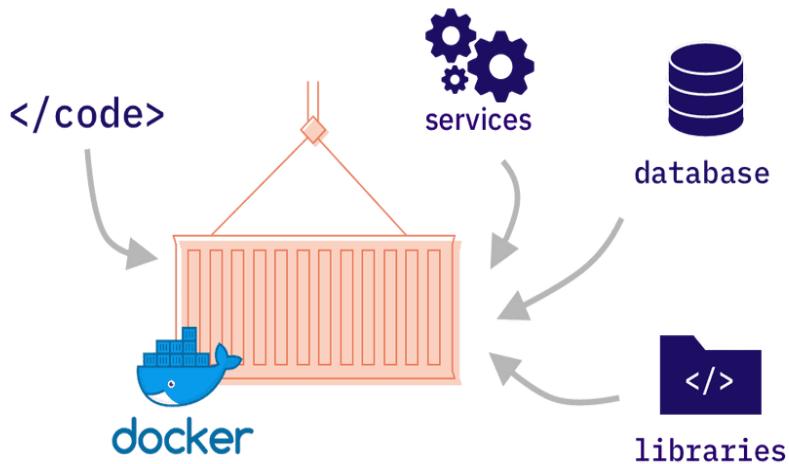
Enables the layering of filesystems, allowing containers to share common files and efficiently manage disk space.

## Container Engines

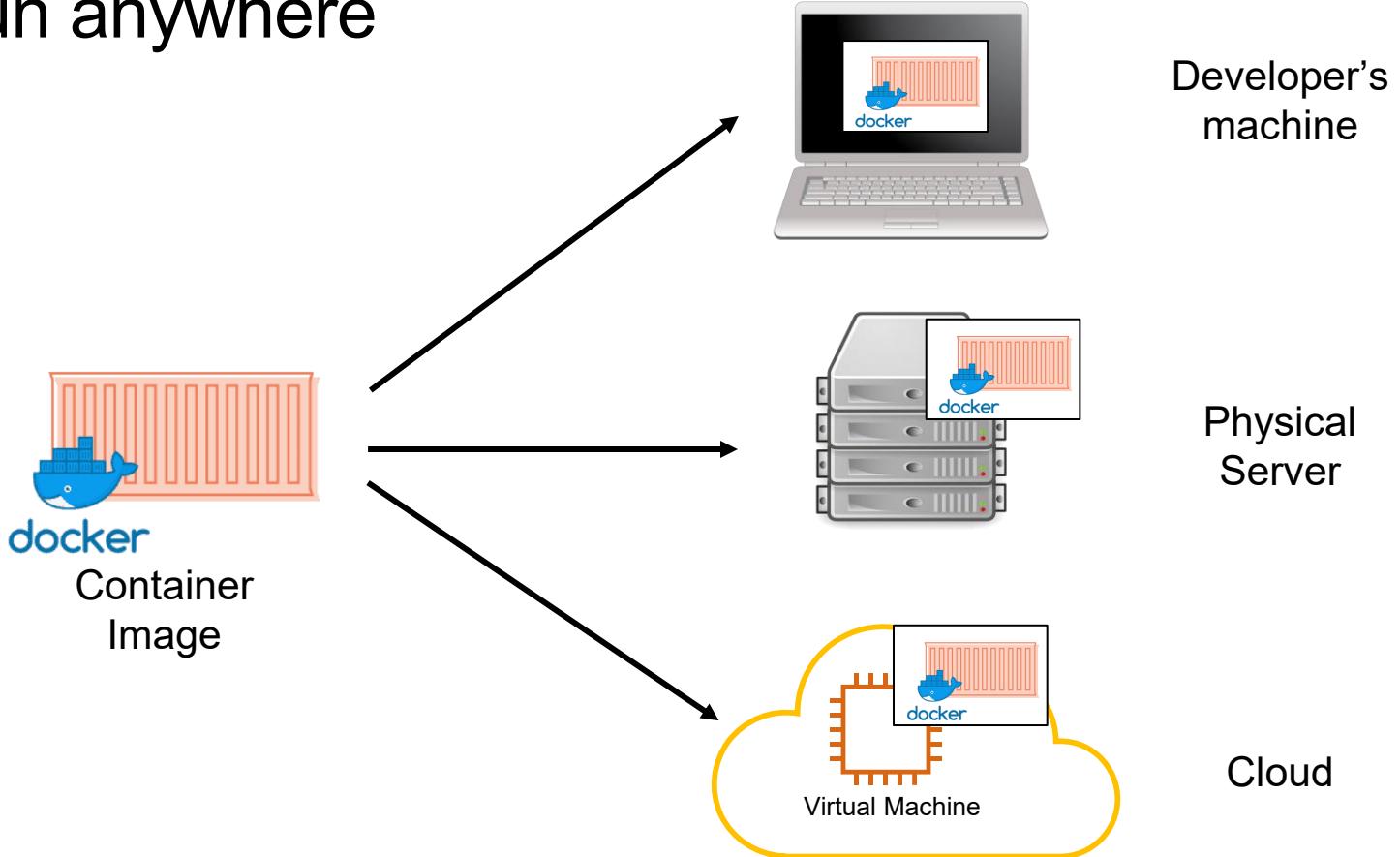


# What is docker?

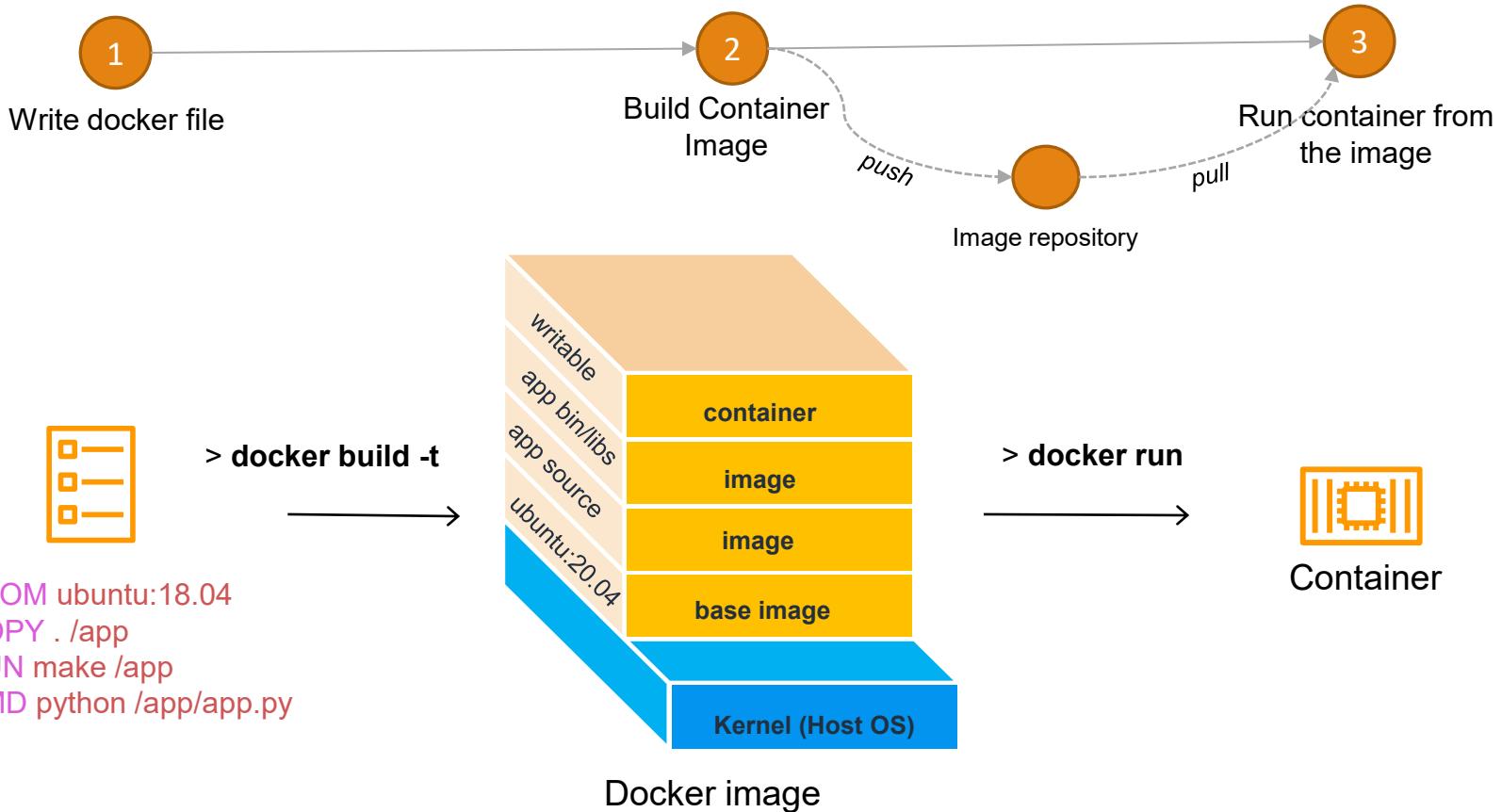
Docker is an open platform for developing, shipping, and running applications as containers



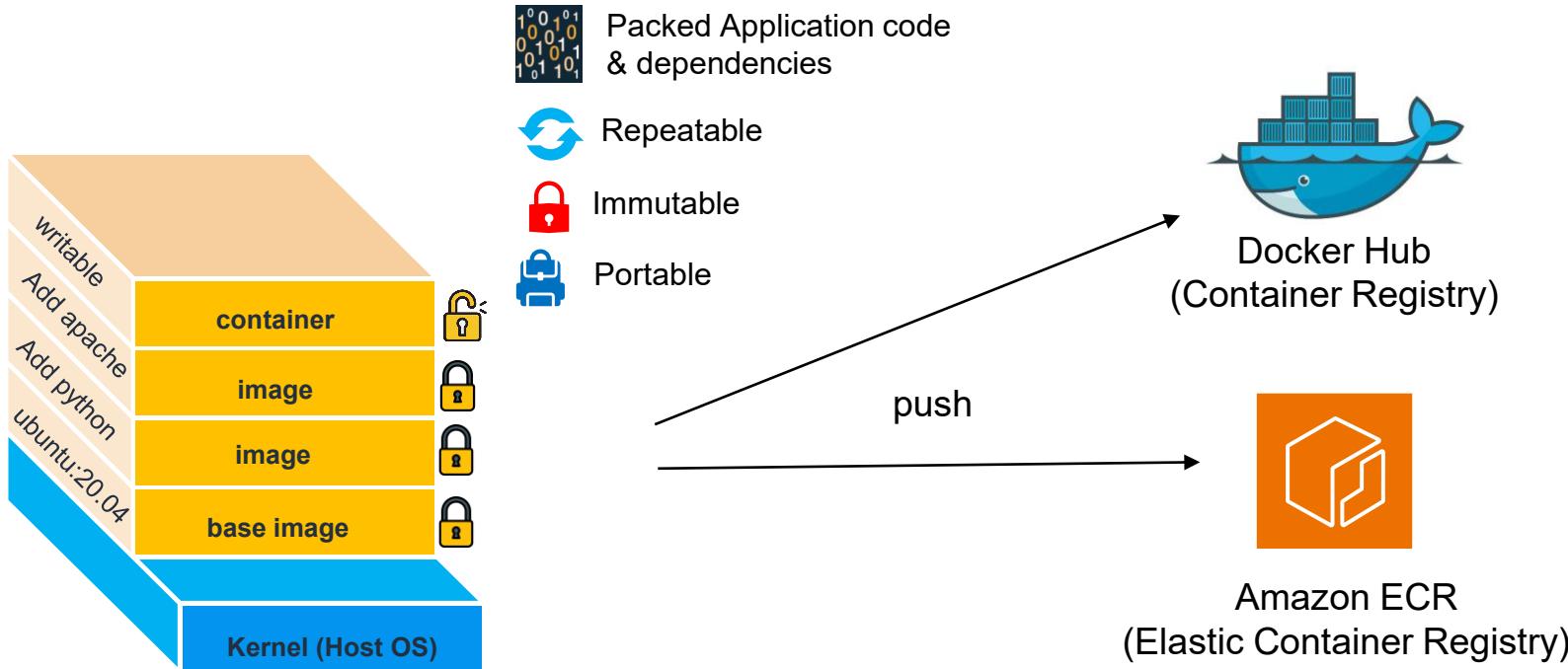
# Run anywhere



# How to use docker?



# Docker Image



# Exercise – Run docker container on EC2 instance

Run a docker container for a simple web server on an EC2 Linux instance

**1. Install Docker engine:**

```
sudo yum install docker -y
```

**2. Start Docker daemon**

```
sudo service docker start
```

**3. Create sample index.html file locally**

**4. Create Dockerfile:**

```
FROM httpd  
COPY index.html /usr/local/apache2/htdocs/
```

**5. Create image:**

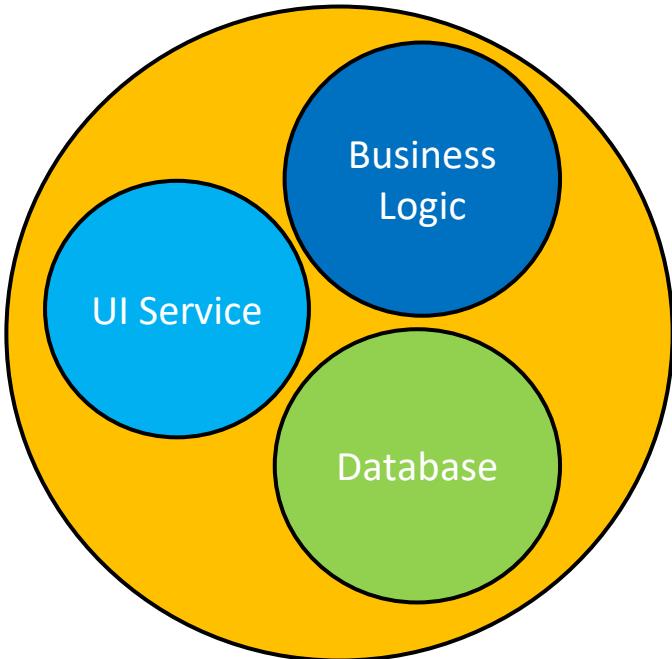
```
sudo docker build -t mywebserver .
```

**6. Run container:**

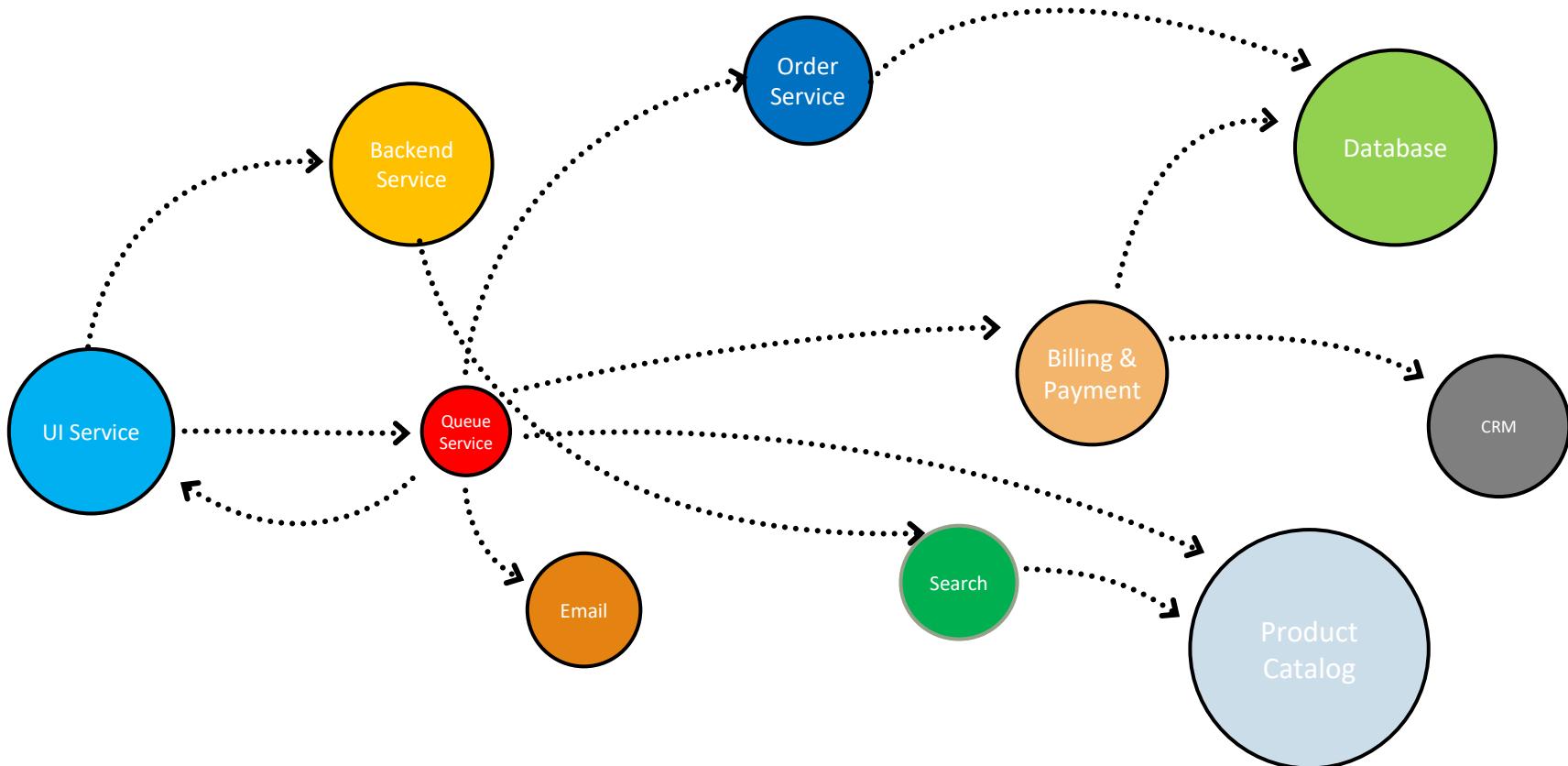
```
sudo docker run --name my-web-server -p 80:80 -d  
mywebserver
```

**7. Access webserver over EC2 Public IP**

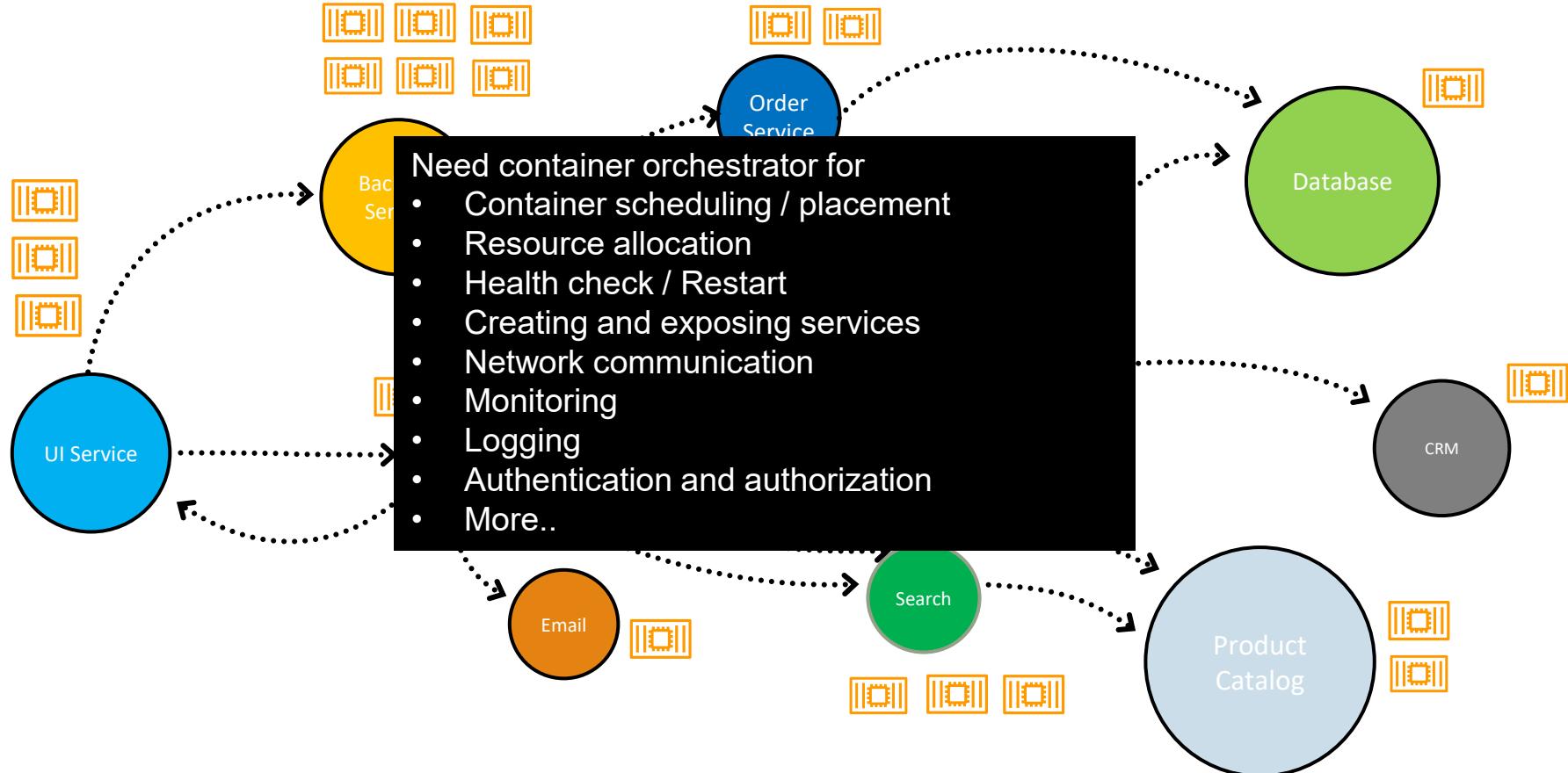
# Monolith application architecture



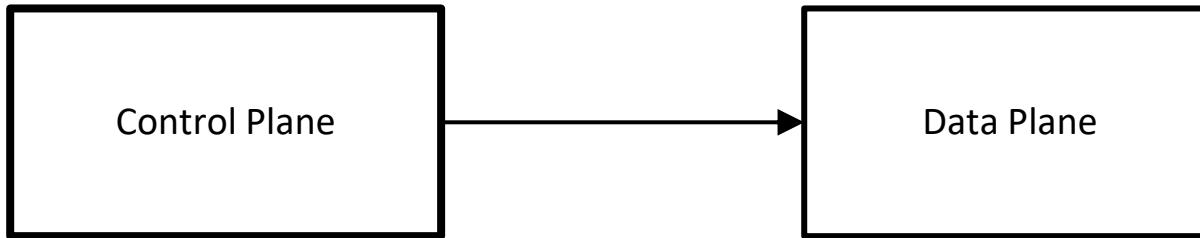
# Microservices application architecture



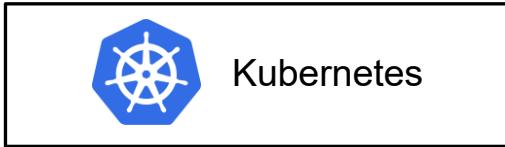
# Microservices application in real world



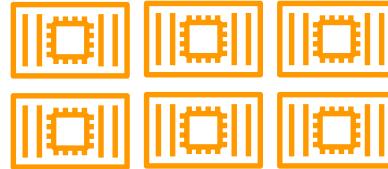
# What is orchestrator?



# Opensource Container Orchestration Tools

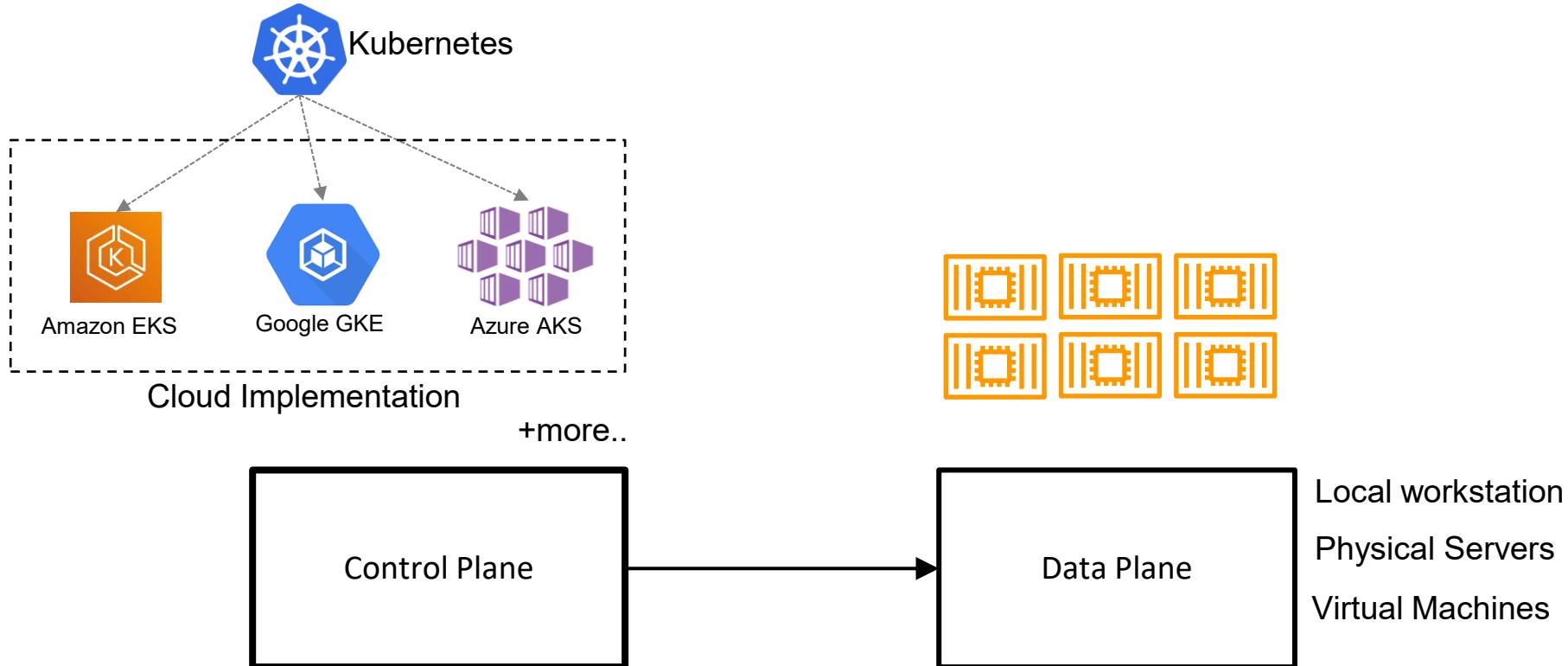


+more..

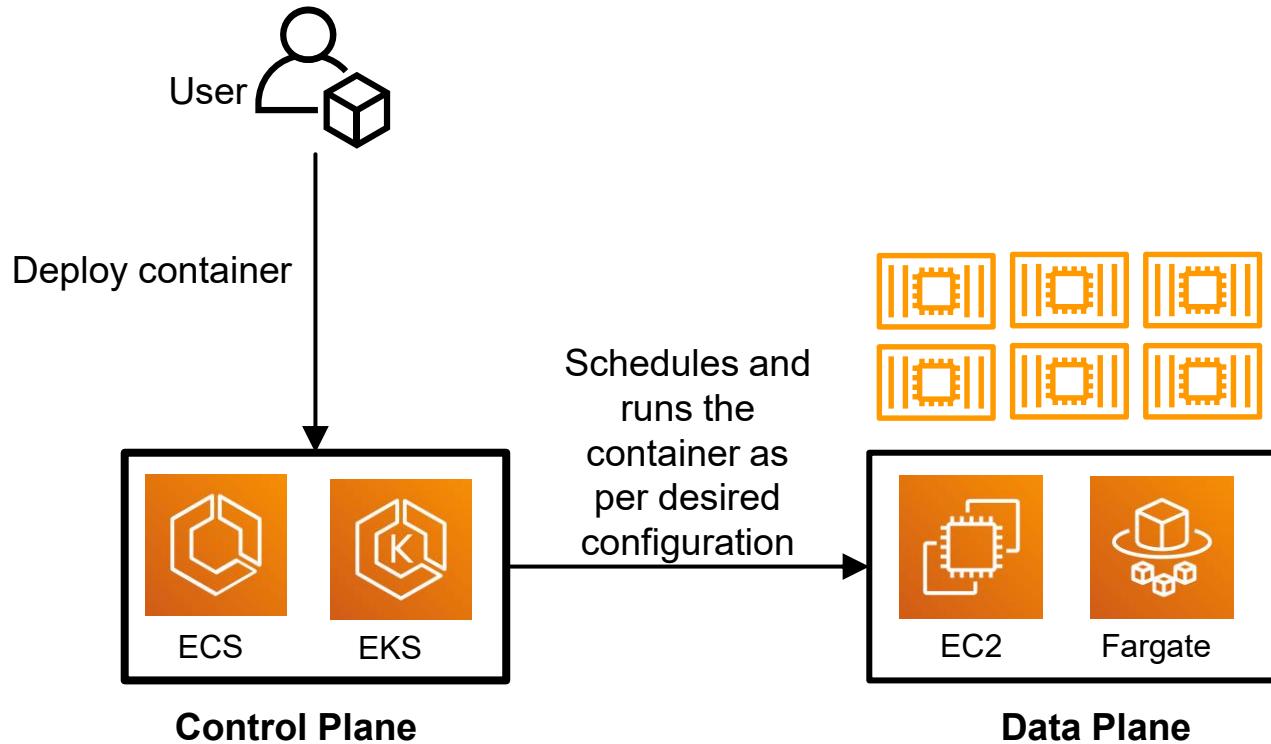


Local workstation  
Physical Servers  
Virtual Machines

# Opensource Container Orchestration Tools



# Container Orchestration services in AWS



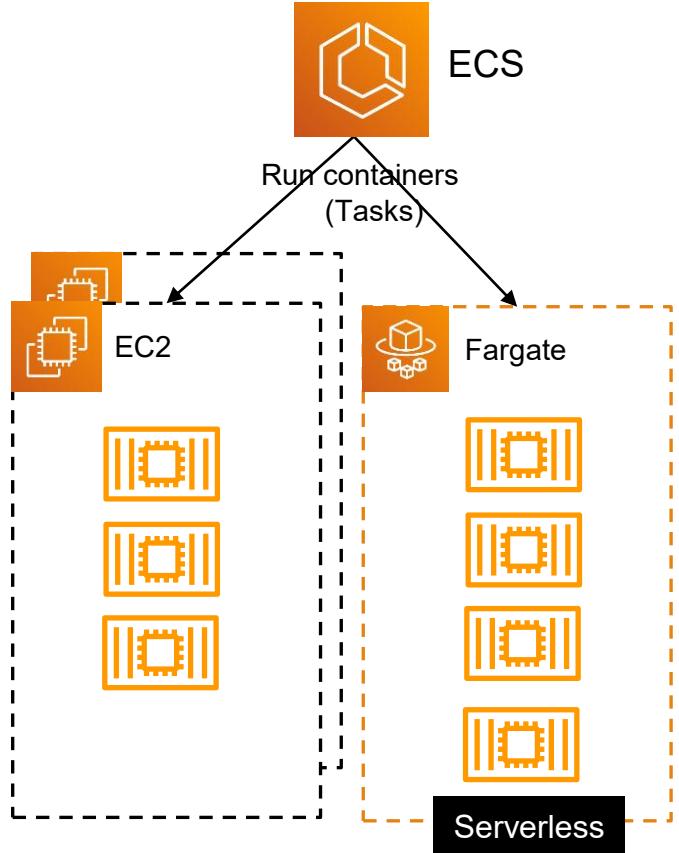


# Amazon ECS

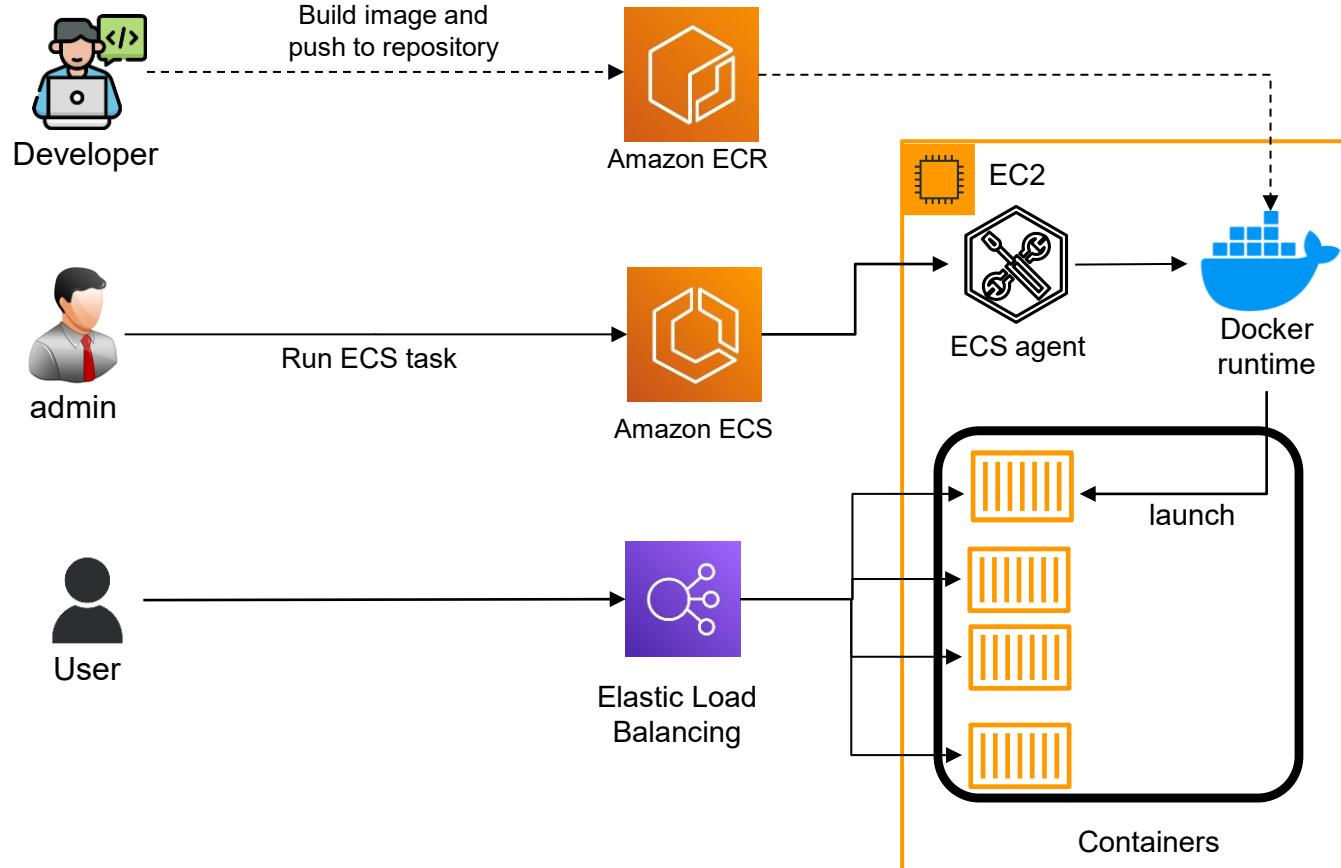
# Amazon ECS

## Elastic Container Service

- Run and manage docker containers at scale
- Best for hosting containers at scale without complexity
- ECS supports 2 different compute options for hosting the containers
  - EC2 – Customer to manage
  - Fargate – AWS manages the infrastructure
- Fargate is a serverless offering from AWS. It provisions resources based on ECS Task vCPU and RAM
- ECS integrates with many AWS services
  - Application Load Balancer
  - CloudWatch
  - Code Build / Code Deploy
  - API Gateway
  - More..



# Amazon ECS architecture



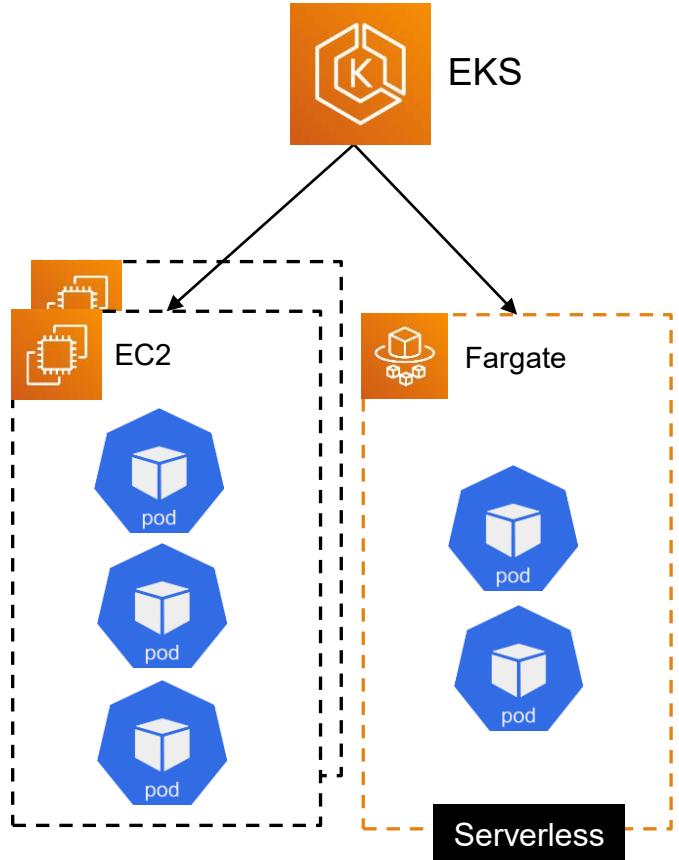


# Amazon EKS

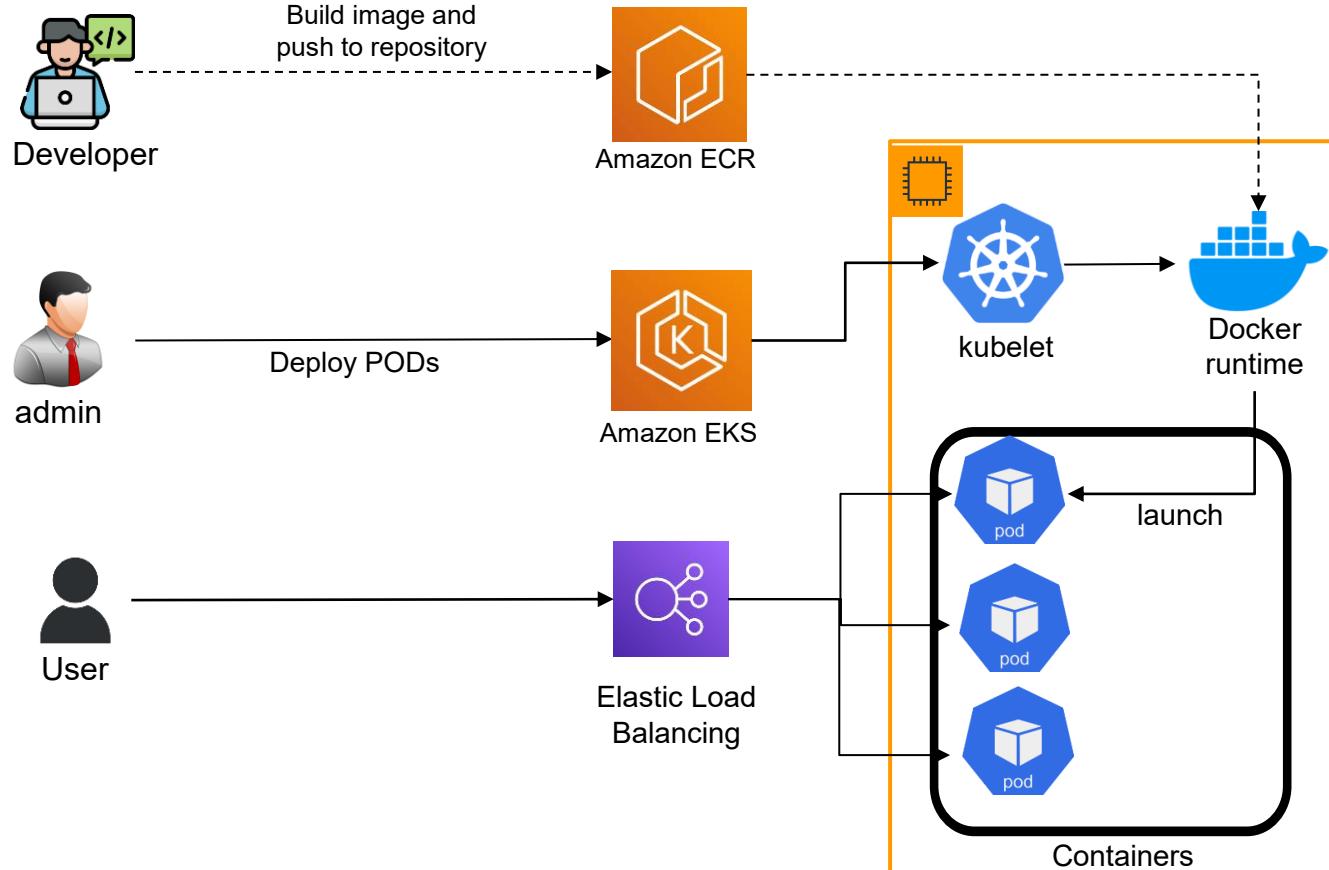
# Amazon EKS

## Elastic Kubernetes Service

- Kubernetes is most popular opensource and broadly used container orchestration platform
- Amazon EKS runs vanilla Kubernetes which is upstream certified conformant version of Kubernetes
- Amazon EKS supports 4 versions of Kubernetes
- Amazon EKS supports 2 different compute options for hosting the PODs (one or more containers)
  - EC2 – Managed node group by EKS
  - Fargate – AWS managed infrastructure
- EKS is a good choice if you already have good experience managing and running kubernetes cluster at scale in production environment



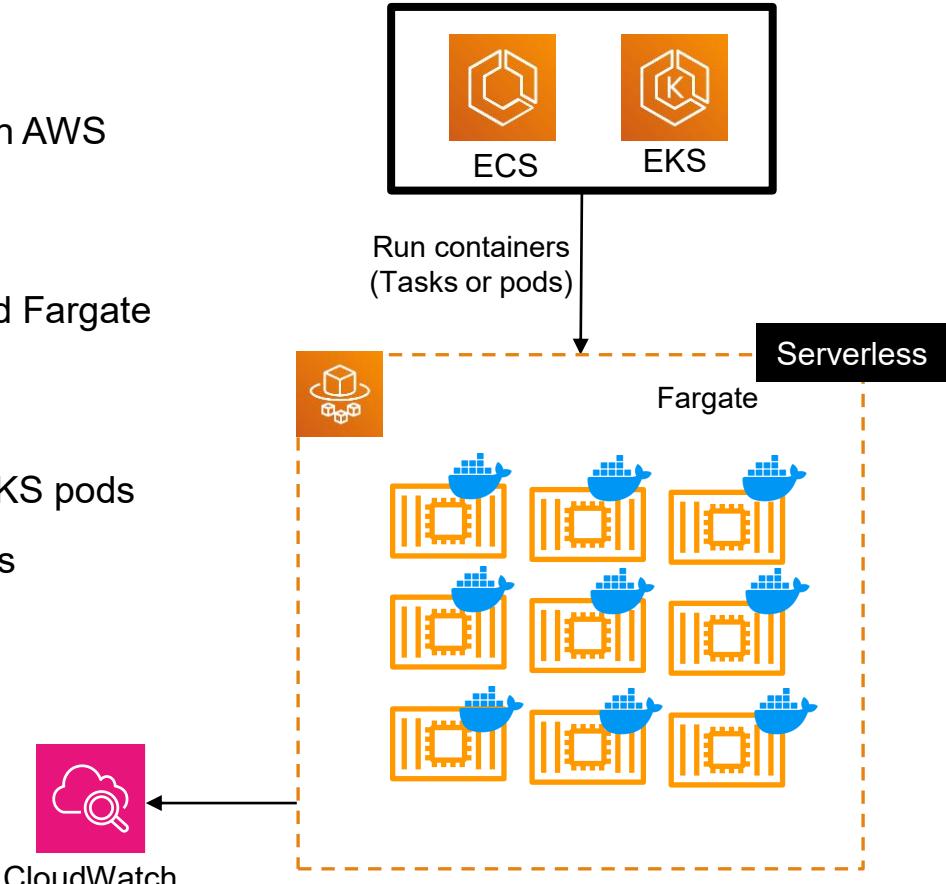
# Amazon EKS architecture



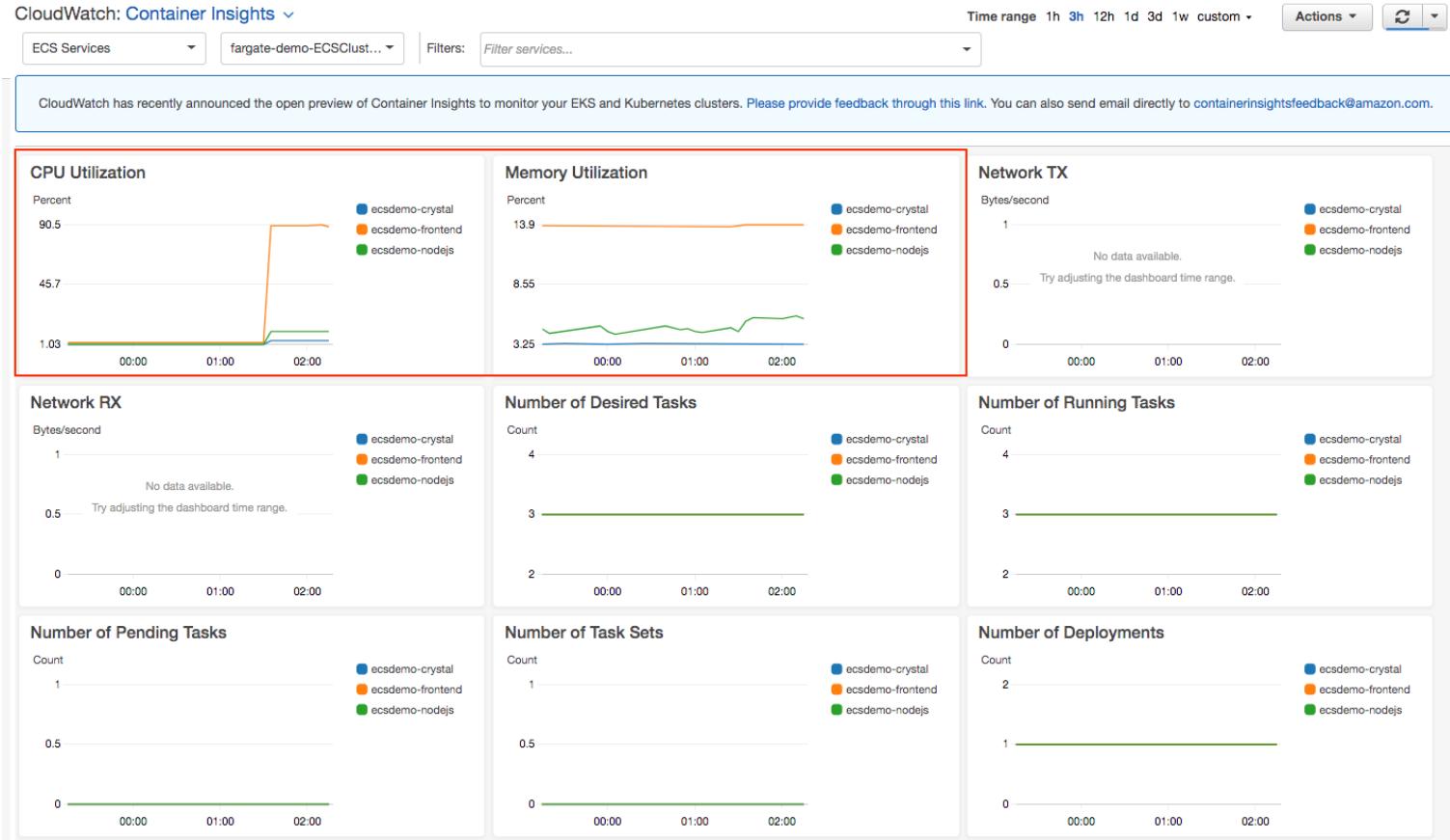
# Fargate



- A Serverless compute for running containers on AWS
- Runs docker containers on AWS managed infrastructure, No EC2 – no management !
- Just define container image, CPU, memory and Fargate will run the container
- Pay-as-you-go pricing
- Works with Amazon ECS tasks and Amazon EKS pods
- Monitoring using CloudWatch container insights

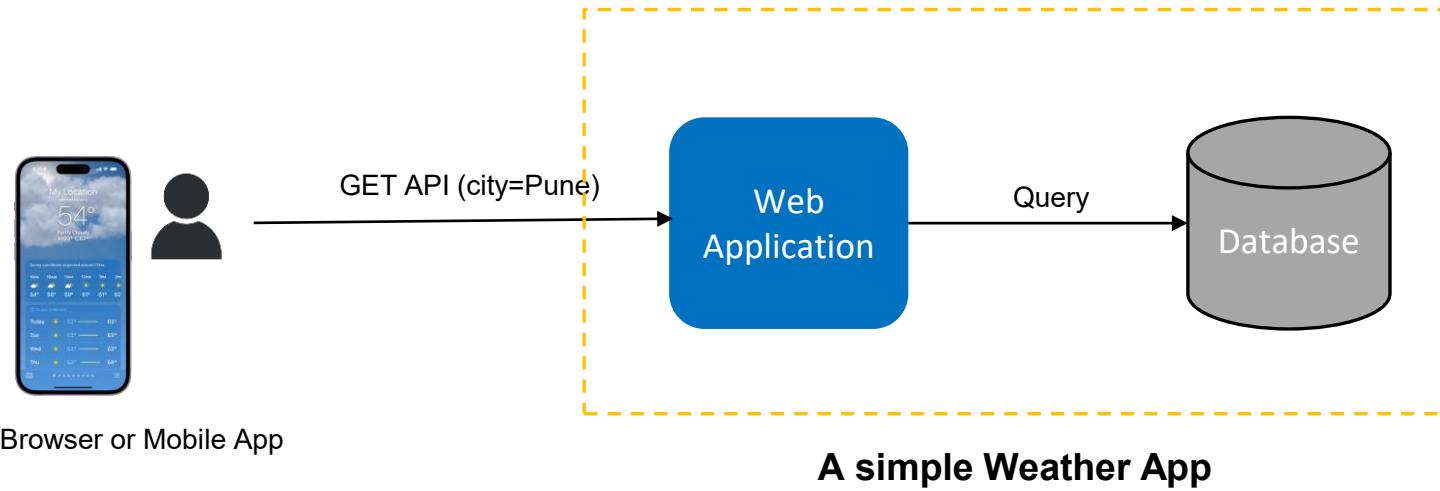


# CloudWatch container insights



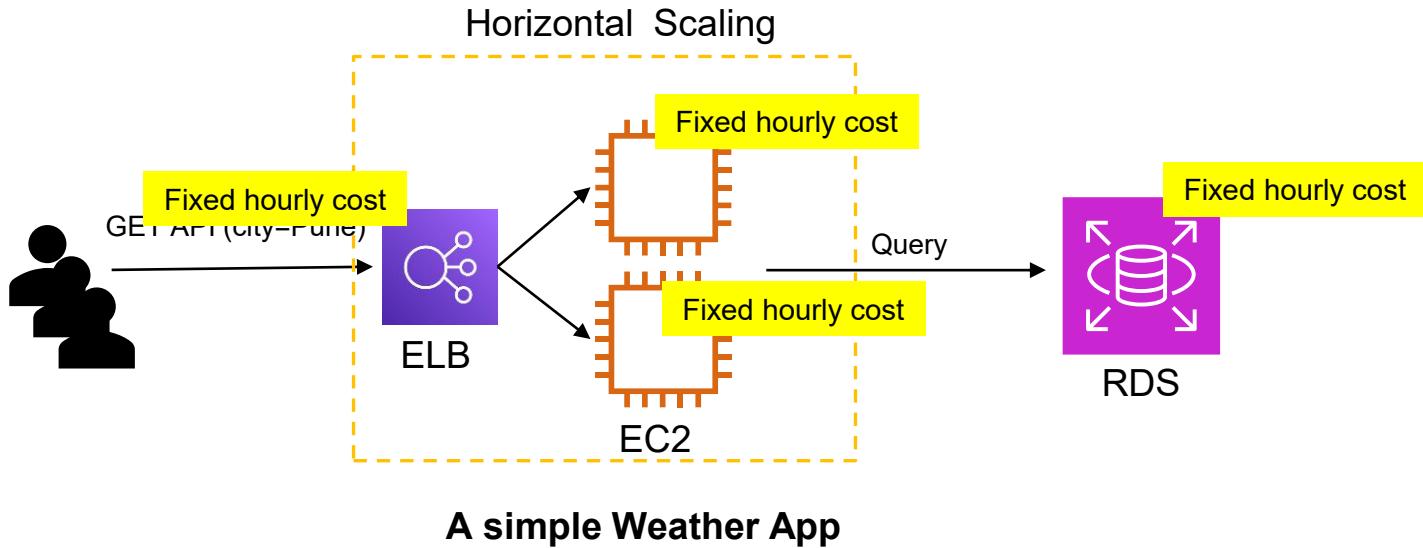
# AWS Serverless

# A simple weather app



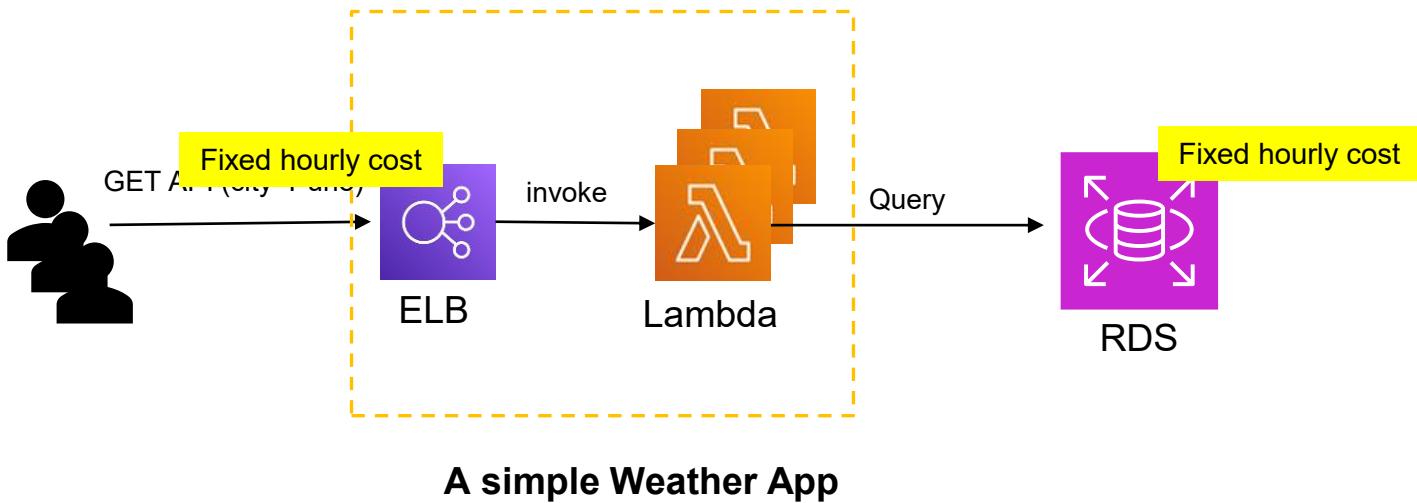
# Hosting application on EC2

What if there is no traffic?



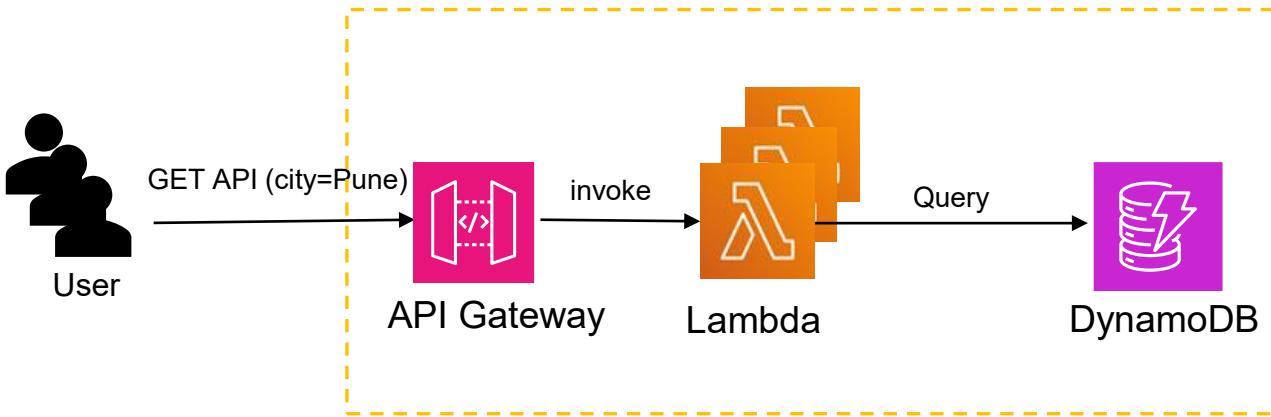
- ✓ Highly Available
- ✗ Infrastructure Management
- ✗ High cost even when no traffic

# Hosting weather app using AWS Lambda



- ✓ Highly Available, Scalable
- ✓ No infrastructure management
- ✓ No cost for Lambda when no traffic

# A serverless weather app



- ✓ Highly Available, Scalable
- ✓ No infrastructure management
- ✓ No infrastructure cost when no traffic
- ✓ Pay per unit

# AWS Serverless



Total monthly requests =  $10k \times 30 = 300k$

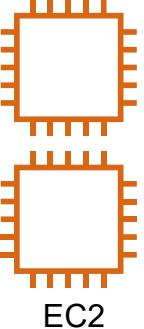
## Option 1



GET API (city=Pune)



ELB



EC2



RDS

Service	Cost/hr	Monthly
EC2 m5.large (2 vcpu, 8G Memory) x 2	\$0.101	730 hr x \$0.101 x 2 = \$147.46
ELB	\$0.0239	730 hr x \$0.0239 = \$17.44
<b>Total</b>		<b>\$164.9</b>

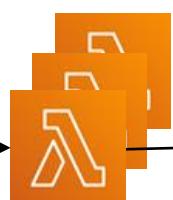
## Option 2



GET API (city=Pune)



API Gateway



Lambda



DynamoDB

Service	Cost/request	Monthly
Lambda (128MB) for every GB-second	\$0.0000166667	\$1.25
Lambda Requests	\$0.20 per 1M requests	\$0.08
API Gateway	\$3.50 per 1M	\$1.20
<b>Total</b>		<b>\$2.53</b>

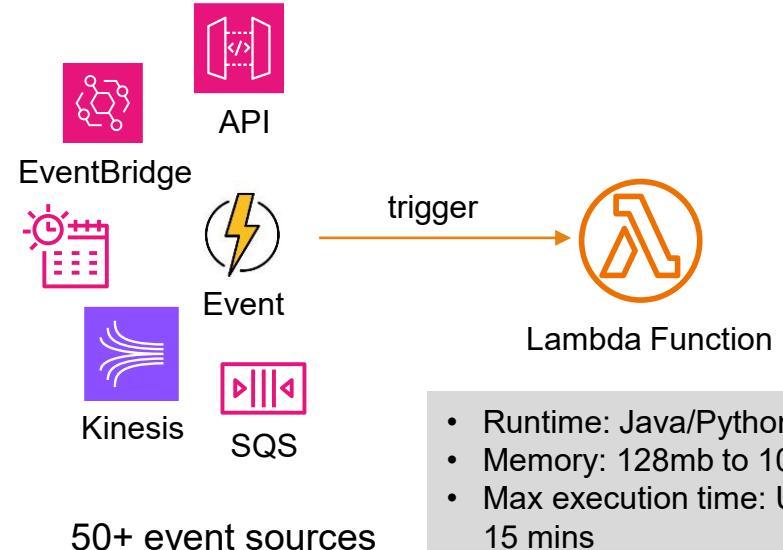
Assumption: 128mb memory, 2 sec execution time



# AWS Lambda

# AWS Lambda

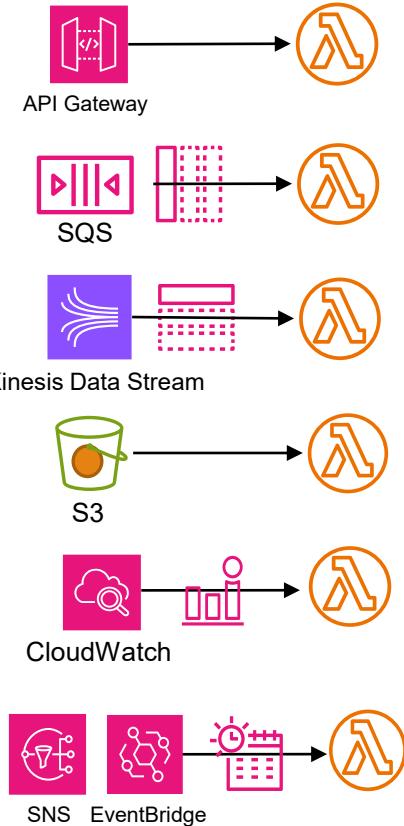
- Runs your code in response to events and automatically manages the underlying compute resources
- Ideal for **stateless** execution
- No administration
- Just write a function or upload a package with code and dependencies.
- Supports all languages (Python, Go, Node.js, Java, C#, Powershell, Ruby, Go and more)
- Scales automatically as per #requests, messages etc.
- Pay as per number of invocations and execution time
- Pricing: Per GB-second & # of requests
- Monitoring with Amazon CloudWatch



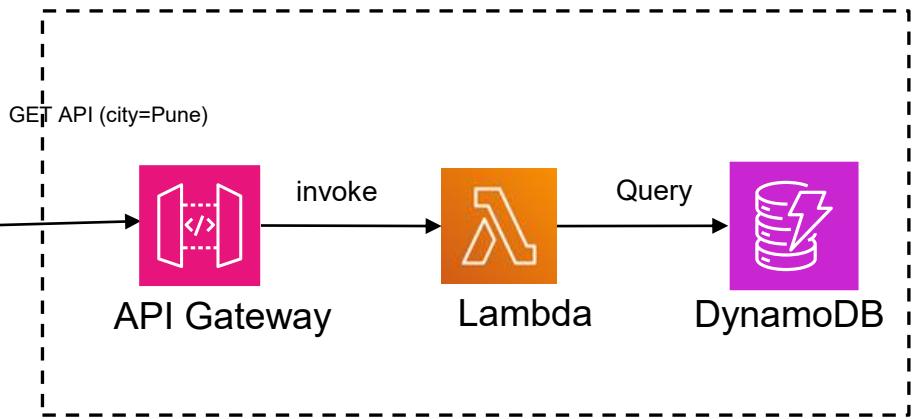
- Runtime: Java/Python etc.
- Memory: 128mb to 10gb
- Max execution time: Up to 15 mins
- IAM role
- More..

# AWS Lambda use cases

- Execute on each API request - Backend for **API gateway** or **Elastic load balancer**
- Order processing – Read **SQS** messages as they arrive in the queue
- Real-time data processing – Read data from **Kinesis** data stream and process in real-time
- File conversion as file is uploaded to **S3** e.g. create thumbnail of the uploaded image
- Take action on **CloudWatch** alarm e.g. Run a cleanup job when disk utilization > 80%
- Run scheduled job /cronjob using **EventBridge** or **SNS notifications** e.g. Send nightly report



# Exercise – A simple weather app with AWS Lambda



**A Weather App – Fully Serverless**

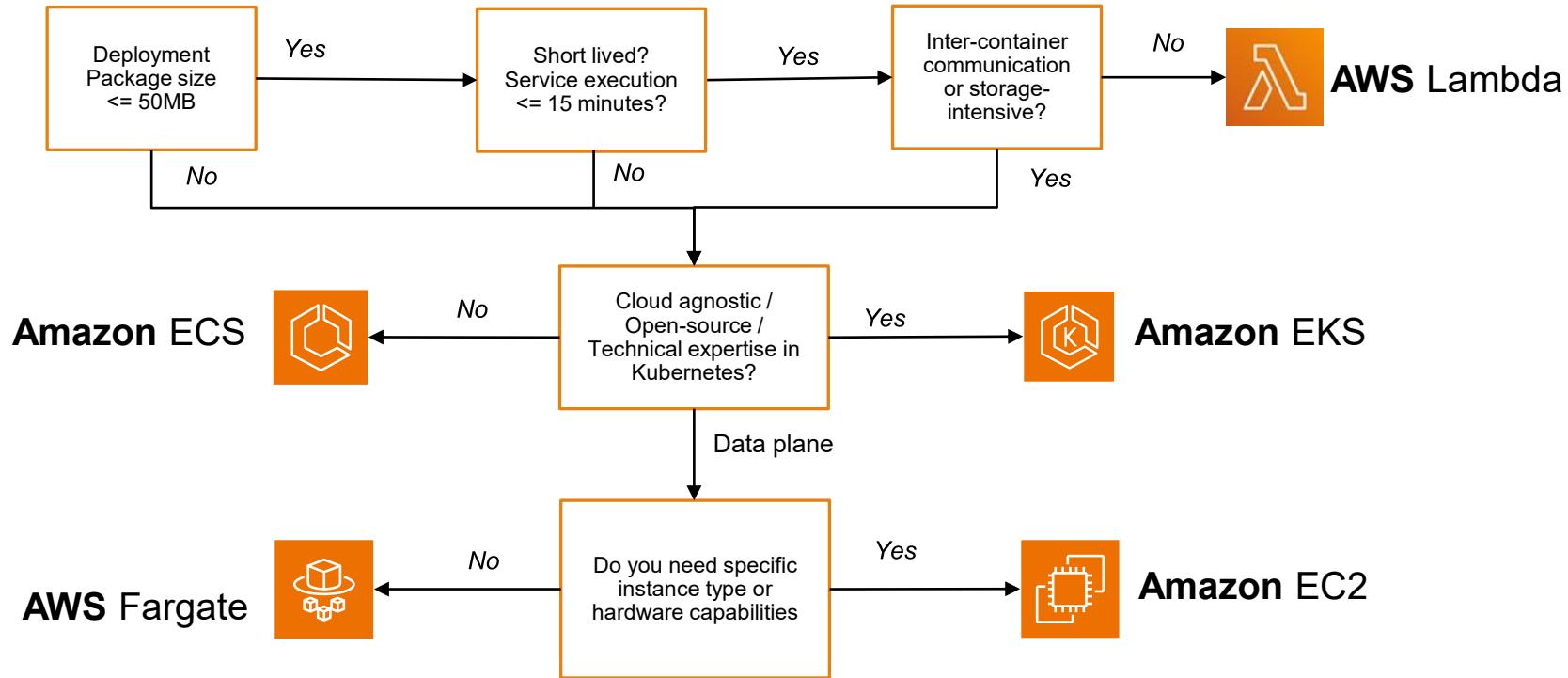
*Reference code files are provided with this lecture.*

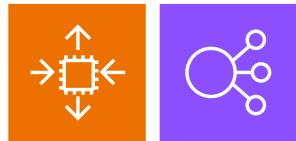
- 1 Create DynamoDB table and add sample items for cities and temperature e.g. Pune (String), 28 (Number)
- 2 Create Lambda function (Python 3.11) and add code provided with this lecture. Update dynamodb table name.
- 3 Create IAM role for Lambda to have Read-only permissions for DynamoDB. Configure Lambda function to use this role and change execution time limit to 1 min.
- 4 Create API Gateway REST API with Any method and invoke Lambda function with proxy integration. Deploy API (use any stage name)
- 5 Create sample HTML web page using the code provided and replace API endpoint with your endpoint.
- 6 Open HTML page and access the simple weather app.

# AWS compute services - summary

1. EC2 – Virtual machine in AWS. Supports almost all kinds of workloads.
2. Containers – Lightweight as compared to VMs. Repeatable, Consistent and Portable.
3. ECS – Elastic Container service built by AWS. Runs docker containers at scale. Simplifies container management.
4. ECS supports EC2 and Fargate as a compute platform for hosting containers.
5. Fargate – Serverless compute option for running containers where infrastructure is fully managed by AWS.
6. ECR – Elastic Container registry to store container images. Can be used by ECS or EKS.
7. EKS – Elastic Kubernetes Service. A managed kubernetes cluster.
8. Lambda – Serverless compute for event driven applications.
9. Lambda is ideal for stateless executions which needs to run for short amount of time (& vanish) e.g. API backend, Event processing, Data processing etc.

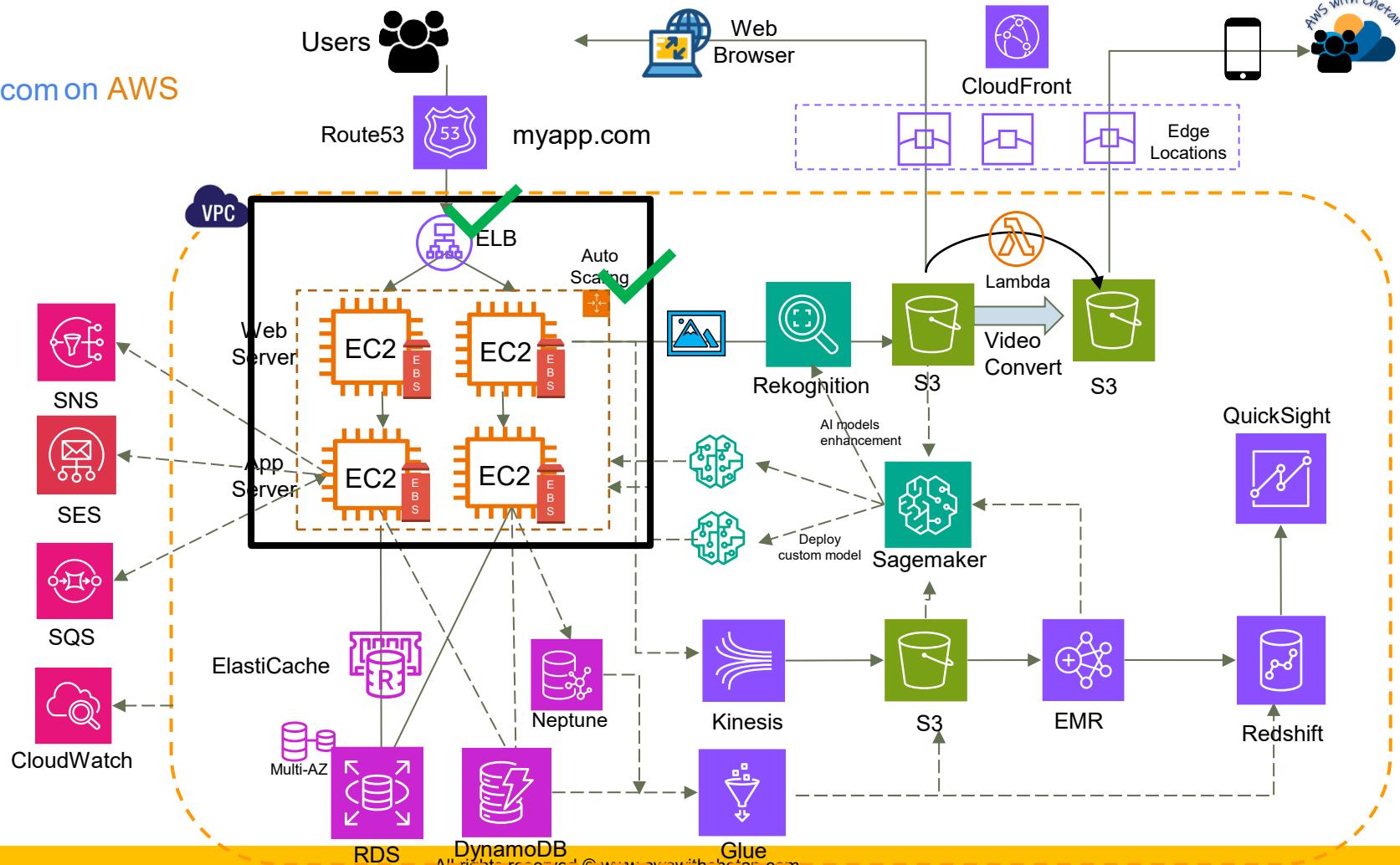
# Choosing between Containers and Lambda





# Load Balancing and Autoscaling

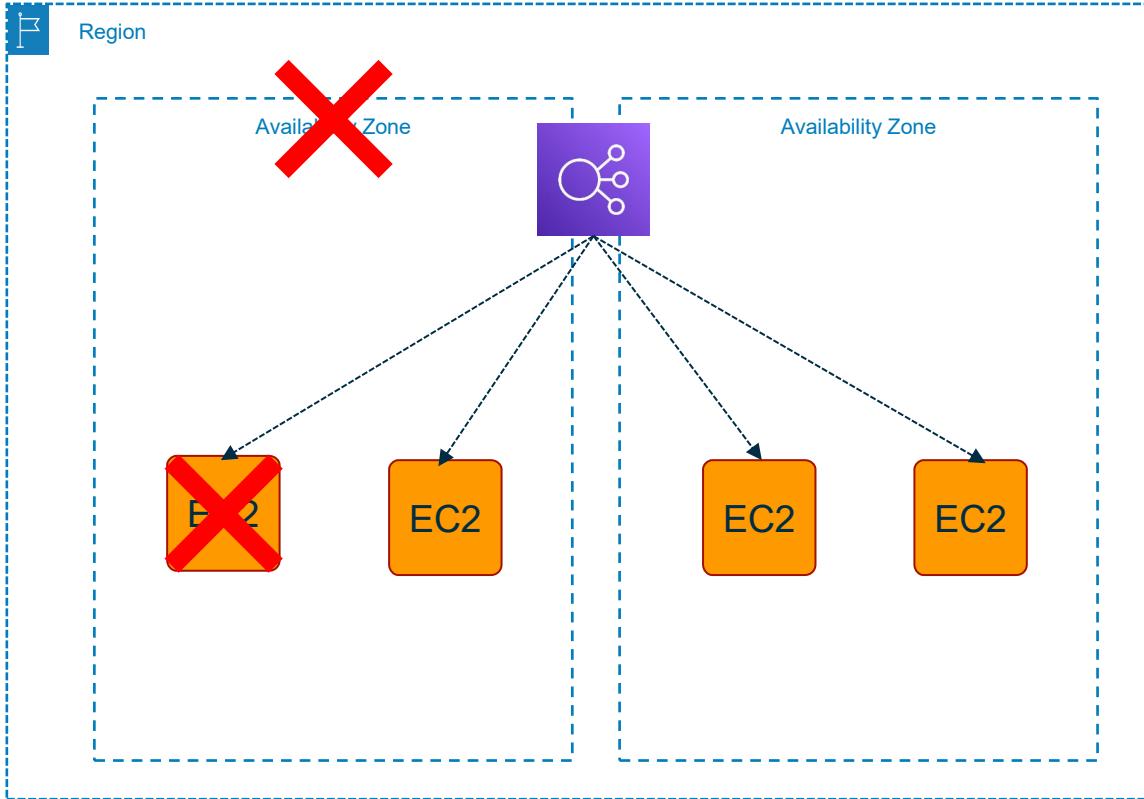
myapp.com on AWS



# High Availability and Scaling

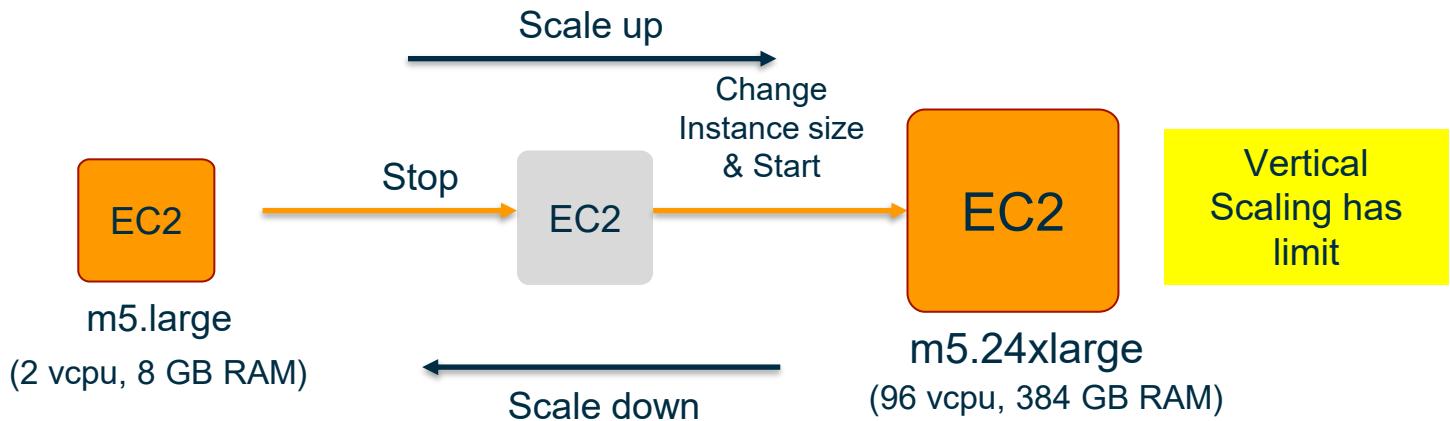
- High Availability means your application runs despite failure in some of the components and during increased traffic
- In AWS world High availability refers to using multiple AZs
- Scalability means ability to scale the capacity up or down as per demand
  - Horizontal scaling is used in distributed systems - Web applications, Big data processing
  - Vertical scaling is used in non-distributed systems - Relational databases
- High Availability and Scalability go hand-in-hand
- Use Elastic Load Balancer (ELB), Auto Scaling group (ASG) for achieving High Availability and Scalability

# High Availability



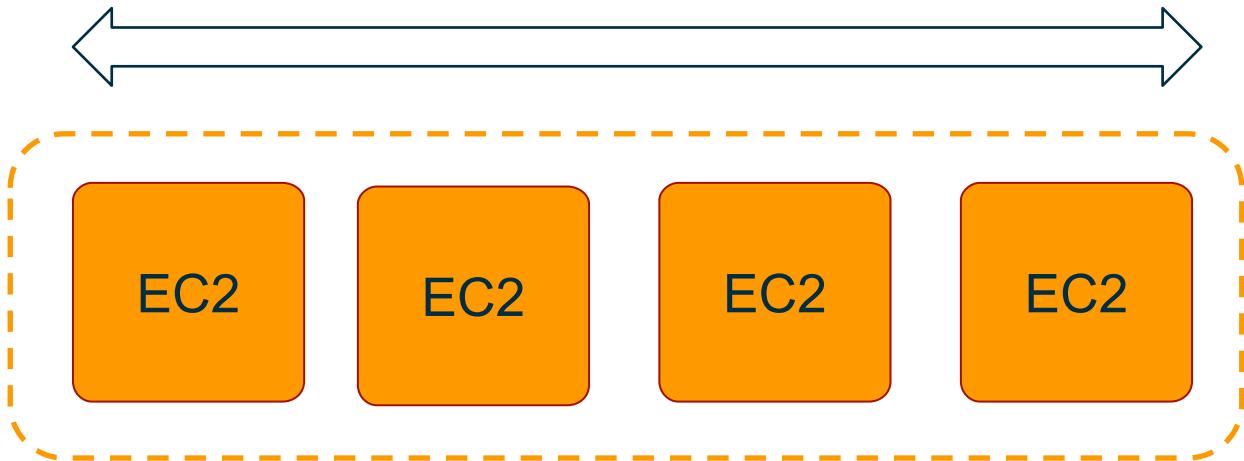
# Scaling

## Option 1: Vertical Scaling

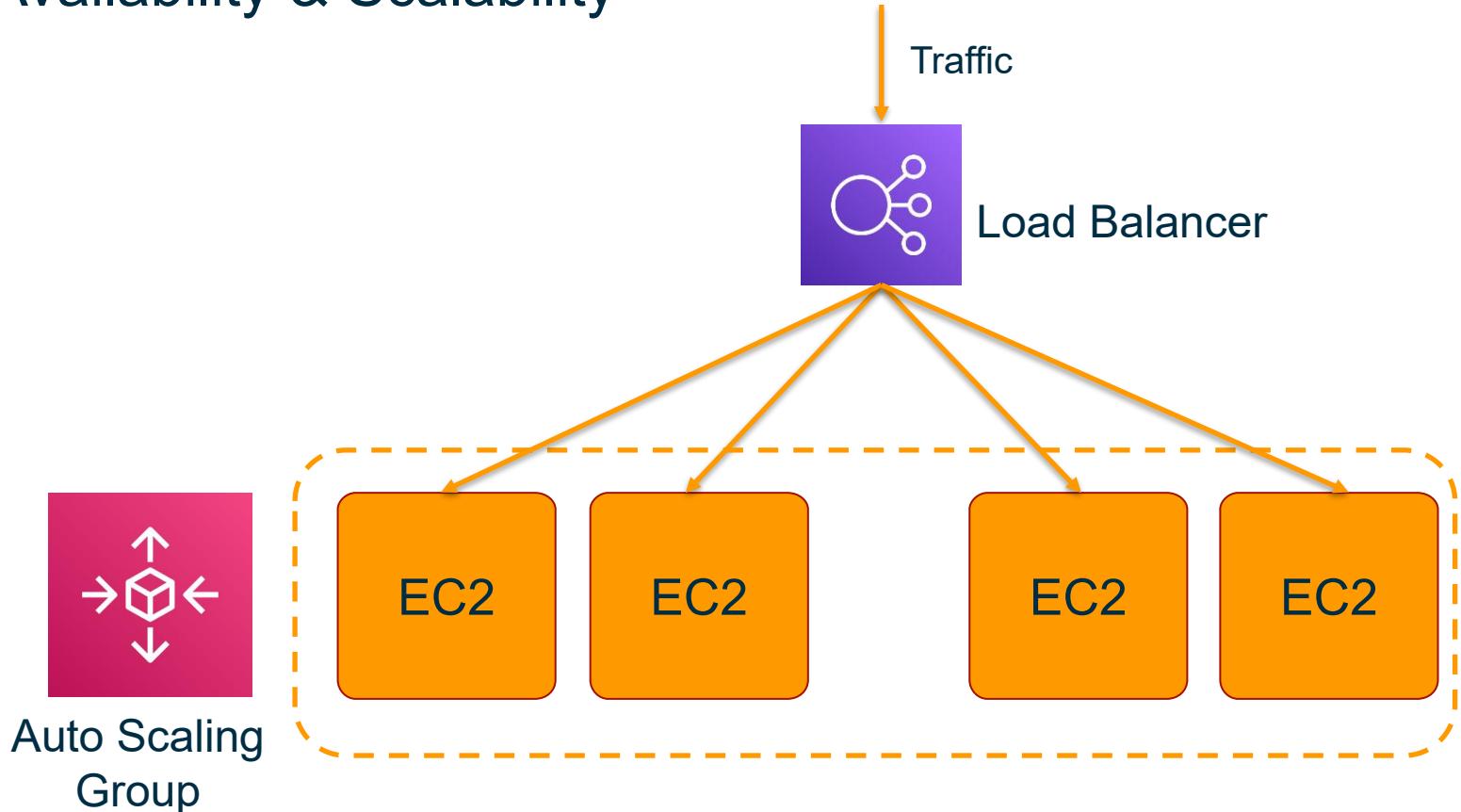


# Scaling

## Option 2: Horizontal Scaling

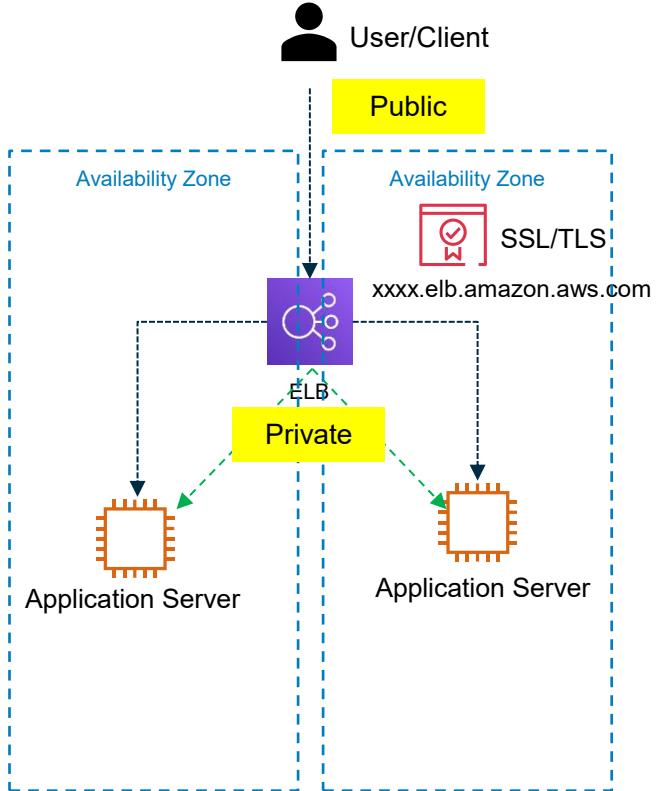


# High Availability & Scalability



# AWS Elastic Load Balancer

- Load Balancer distributes incoming traffic to multiple downstream servers (e.g. EC2 instances, Containers, IP addresses and Lambda functions)
- Supports protocols like HTTP, HTTPS, HTTP/2, TCP, UDP, gRPC
- Expose a single point of access (DNS) to your application
- Seamlessly handle failures of downstream instances by performing health checks to the instances
- Provide SSL/TLS termination for your websites and web applications
- High availability across Availability zones (AZs)
- Separate public traffic from private traffic
- Supports various routing mechanisms

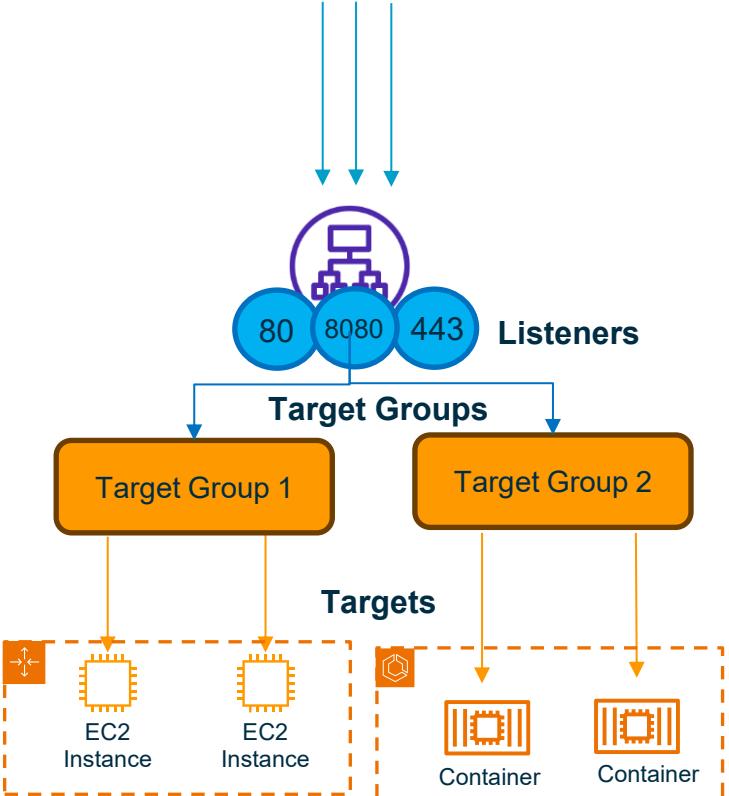


# Elastic Load Balancer components

- **Listener:** supports multiple listeners on different ports
- **Target Group:** group of targets
- **Target:** EC2 instance, Container, IP , Lambda functions

More features:

- Authentication
- Routing
- Health Checks
- Sticky sessions
- SSL/TLS certificate
- Load balancer algorithms

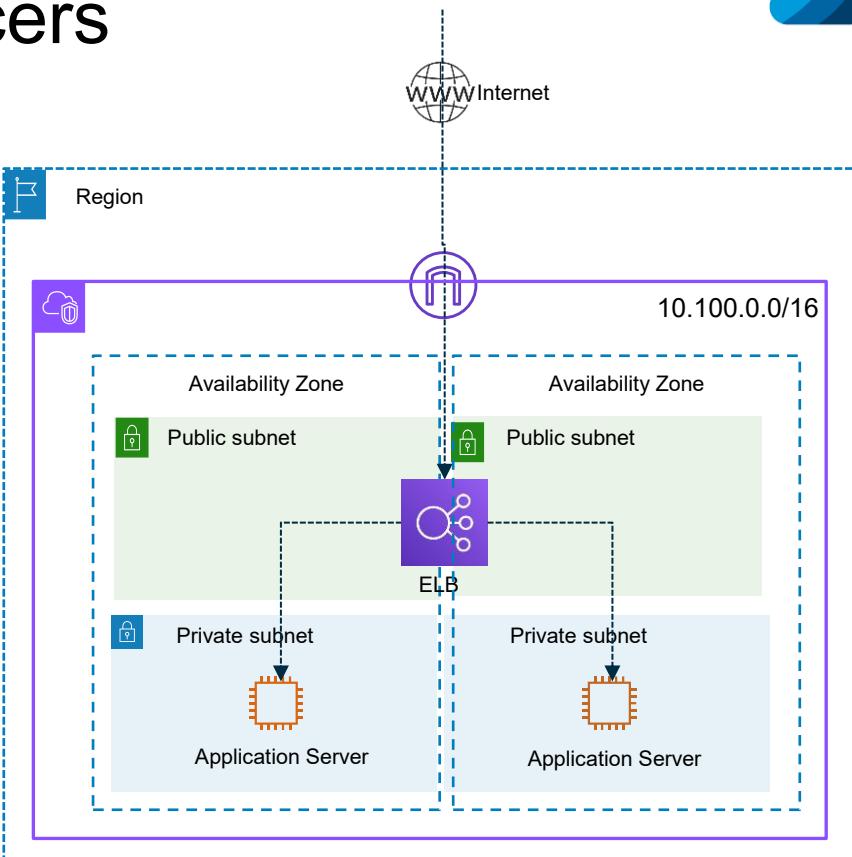


# Types of Elastic Load Balancers

AWS offers following types of Load Balancers

1. Application Load Balancer (ALB) – Layer 7  
[HTTP](#), [HTTPS](#), [WebSocket](#), [gRPC](#)
2. Network Load Balancer (NLB) – Layer 4  
[TCP](#), [TLS \(secure TCP\)](#), [UDP](#)
3. Gateway Load Balancer (GWLB) – Layer 3  
[GENEVE Protocol on IP](#)
4. Classic Load Balancer (CLB) – Layer 4 & 7  
[HTTP](#), [HTTPS](#), [TCP](#), [SSL/TLS \(secure TCP\)](#)

Legacy





## Application Load Balancer

- Operates at Layer 7
- Supported protocols HTTP, HTTPS, WebSocket, HTTP/2 and gRPC
- Gets static DNS
- Supports SSL/TLS termination
- Use: Web Application



## Network Load Balancer

- Operates at Layer 4
- Supported protocols TCP, UDP
- Provides Static IP so that client can whitelist (if required)
- Supports TLS termination
- Use: Ultra low latency requirement.
- Example: NTP server, MQTT brokers, Message passing applications



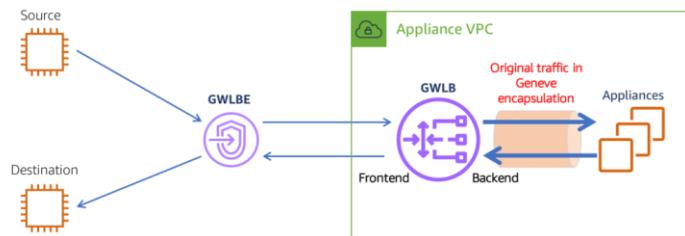
## Gateway Load Balancer

- Operates at Layer 3
- Supported GENEVE protocol
- Use: Traffic inspection where traffic is routed to backed Firewall or network Appliance instances

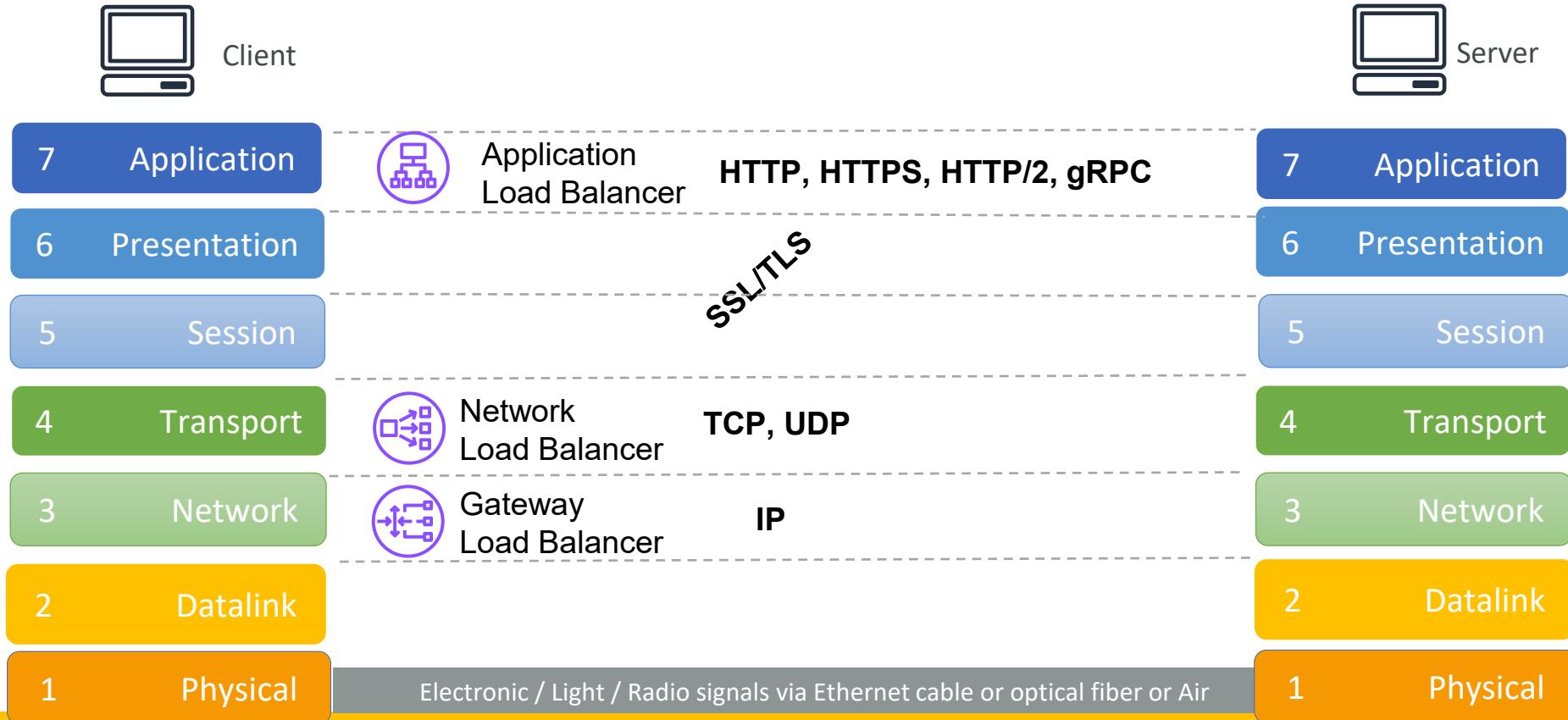


## Classic Load Balancer

- Operates at Layer 4 & 7
- Supported protocols HTTP, HTTPS, TCP, SSL
- Use: Web applications



# OSI Network Layers



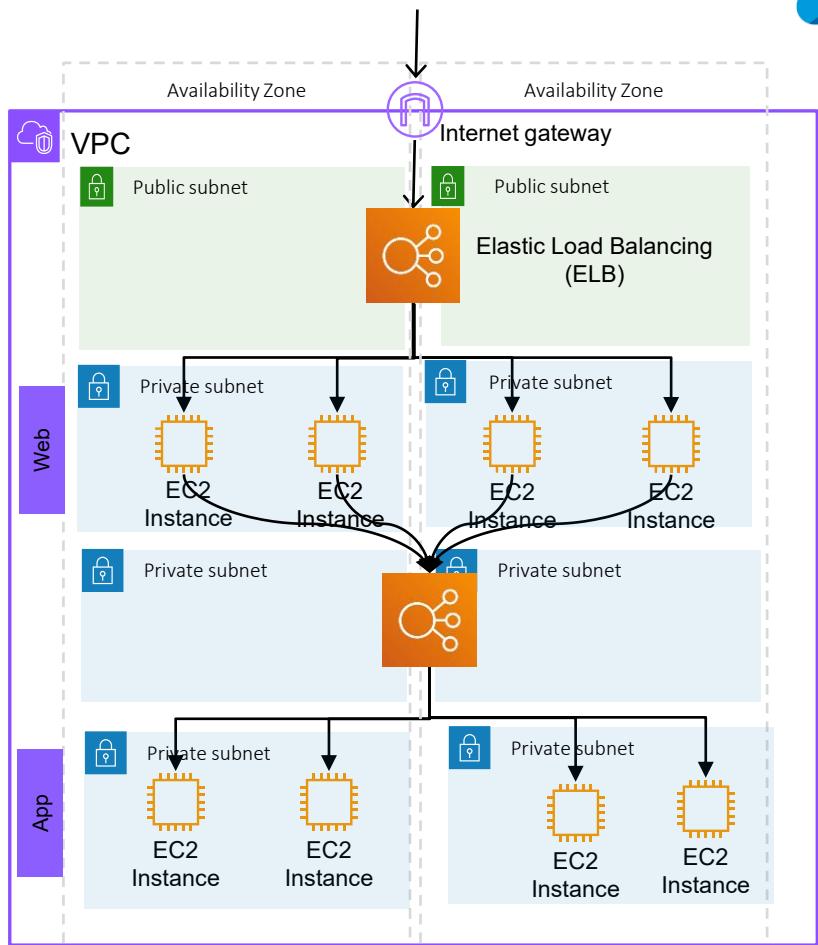
# Load Balancer network

## 1. External Load Balancer (Public)

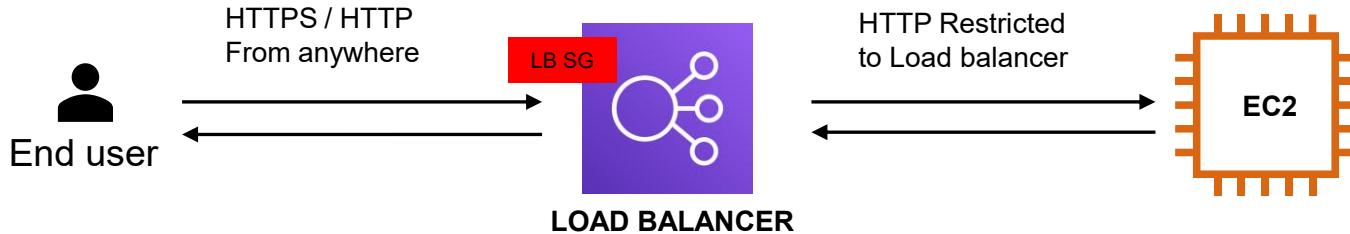
- To be launched in Public Subnet
- Receives traffic from the internet

## 2. Internal Load Balancer (Private)

- To be launched inside the Private subnet
- Receives traffic from within the VPC



# Load Balancer Security Groups



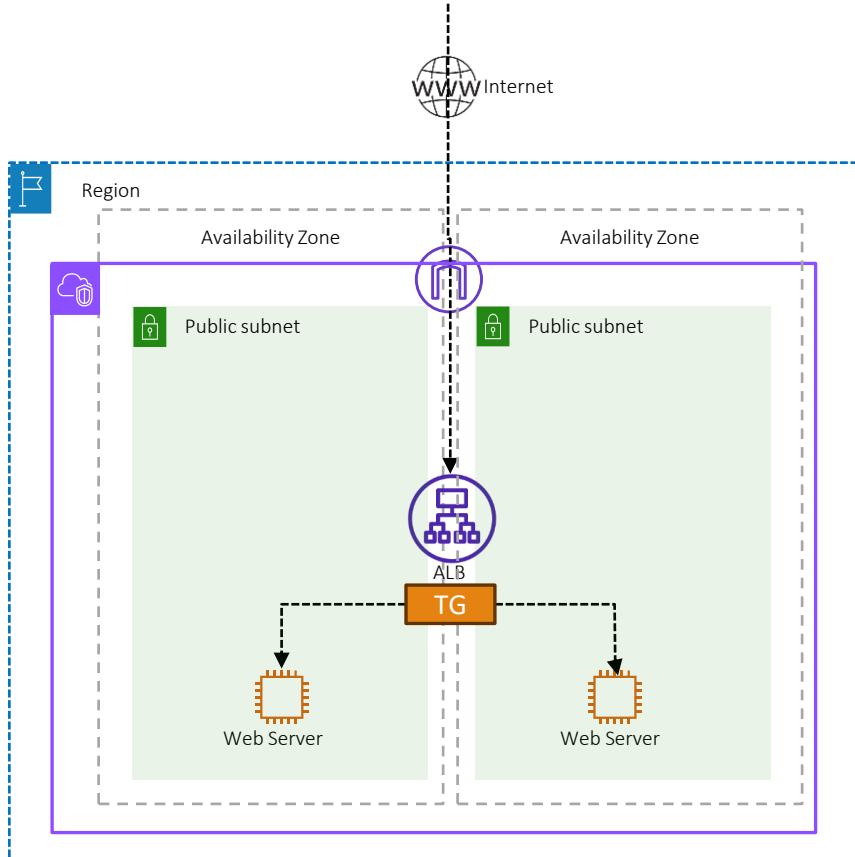
Load Balancer Security Group:

Type	Protocol	Port range	Source	Description
HTTP	TCP	80	0.0.0.0/0	Allow HTTP
HTTPS	TCP	443	0.0.0.0/0	Allow HTTPS

Application Security Group: Allow traffic only from Load Balancer

Type	Protocol	Port range	Source	Description
HTTP	TCP	80	sg-0bc494c71e6cf1896 / ALB-SG	Allow traffic from ALB

# Exercise – ALB with two backend EC2 instances



## High level steps

- 1 Using Default VPC
- 2 Launch Web server 1 and Web server 2 in respective in separate subnets (in different AZs) using user data script to install the web server.
- 3 Create ALB Target group and add both EC2 instances. Configure health check on port 80 with /index.html
- 4 Create ALB with HTTP listener and forward the traffic to Target group. Open ALB Security group for HTTP traffic.
- 5 Update EC2 security group to allow HTTP traffic from ALB security group
- 6 Access ALB over the browser using AWS provided ALB DNS

# EC2 User data script

## Web Server 1 User data script

```
#!/bin/bash
yum install httpd -y
systemctl start httpd.service
systemctl enable httpd.service
echo "<h1>This is Webserver 1</h1>" > /var/www/html/index.html
echo "Configured successfully"
```

## Web Server 2 User data script

```
#!/bin/bash
yum install httpd -y
systemctl start httpd.service
systemctl enable httpd.service
echo "<h1>This is Webserver 2</h1>" > /var/www/html/index.html
echo "Configured successfully"
```

# Cleanup – Very important

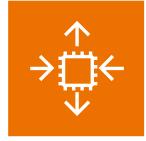
1. Terminate both the EC2 instances
2. Delete Load Balancer Target groups
3. Delete Load Balancer



# Autoscaling group

# Auto Scaling group (ASG)

- Auto Scaling groups automate the scaling of EC2
- The purpose of an Auto Scaling Group is to:
  - Scale out (add EC2 instances) to handle an increased load
  - Scale in (remove EC2 instances) to handle a decreased load
  - Ensure there is a desired number of instances running all the time with limit of minimum and a maximum number
- Auto Scaling helps optimize the cost by running only optimum number of EC2 instances as per demand
- ASG works perfectly with ELB and registers EC2 instances to Load Balancer automatically
- ASG also replaces EC2 instances after ELB detects them as unhealthy



Autoscaling Group



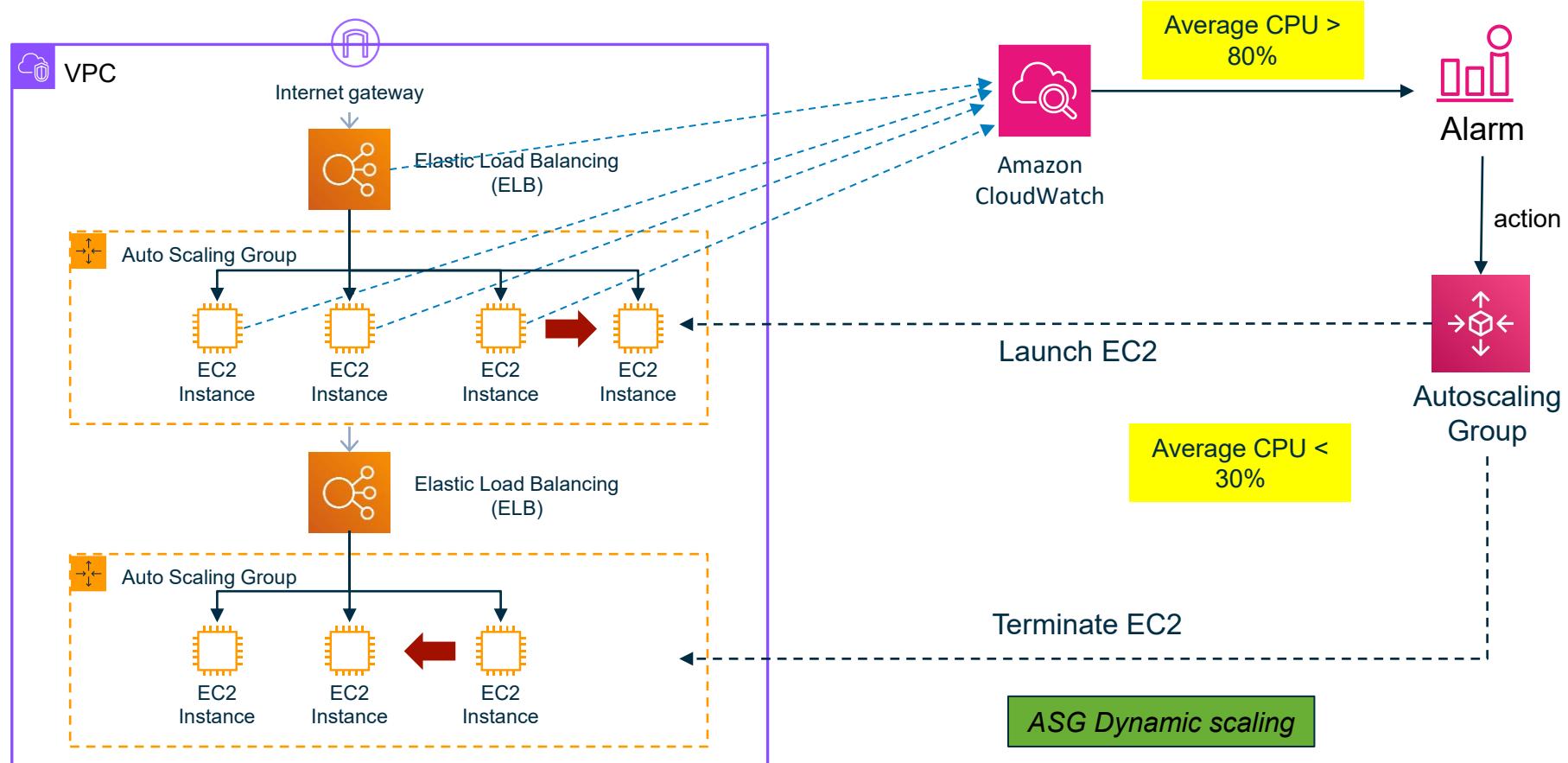
ELB

# Auto Scaling group scaling policies

- Manual Scaling
  - Set the required number of EC2 instances in the ASG configuration
- Dynamic Scaling
  - Step/Simple scaling
    - When avg. CPU > 80% add 1 EC2 instance
    - When avg. CPU < 40% remove 1 EC2 instance
  - Target Tracking
    - Maintain the avg. CPU utilization to ~40%
- Scheduled Scaling
  - Schedule the scaling for given period – Saturday evening 6-11pm
- Predictive Scaling
  - User Machine Learning to scale based on the historic utilization (CloudWatch metrics)



# ALB + Autoscaling Group



# Cleanup – Very important

1. Terminate both the EC2 instances
2. Deregister AMIs
3. Delete Load Balancer Target groups
4. Delete Load Balancer
5. Delete Route53 records
6. You can keep the VPC (there is no cost) and you can use it for your next assignment.

# Load Balancer and Auto scaling group - summary

- Cloud provides mechanism and services to implement High Availability and Scalability
- High Availability in AWS means deploying application across multiple Availability Zones (AZs)
- For EC2 based applications high availability and scalability, use Elastic Load Balancer and Auto Scaling group

## • **Elastic Load Balancer**

- For distributing incoming traffic across backend EC2 instances, containers, IP addresses or lambda function
- AWS has 4 types of load balancers: Application Load Balancer (Layer 7), Network Load balancer (Layer 4), Gateway Load Balancer (Layer 3), Classic Load Balancer (Layer 4 & 7)
- Load balancer performs health check of backend targets and send traffic to only healthy target

### **Load balancer troubleshooting tips:**

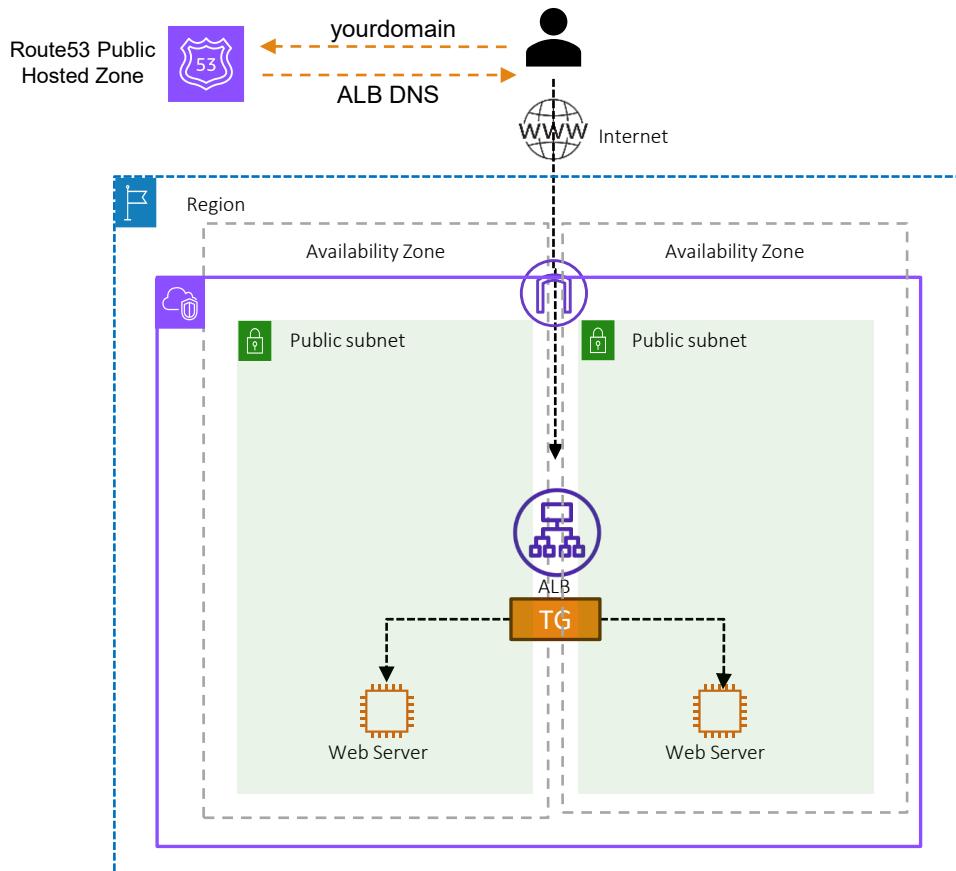
- If the Load Balancer can't connect to your application, check EC2 security groups.
- Load Balancer Errors 503 means no registered target

# Load Balancer and Auto scaling group - summary

- **Auto Scaling group**

- Scales the EC2 capacity up or down as per the auto scaling group policy
- Supports Manual scaling, Dynamic scaling (Simple, Target tracking), Scheduled scaling, Predictive scaling
- Works closely with ELB for dynamic scaling
- Provides cost benefits by running only required number of EC2 instances for applications serving unpredictable traffic/load

# Assignment 1 – ALB with custom domain name

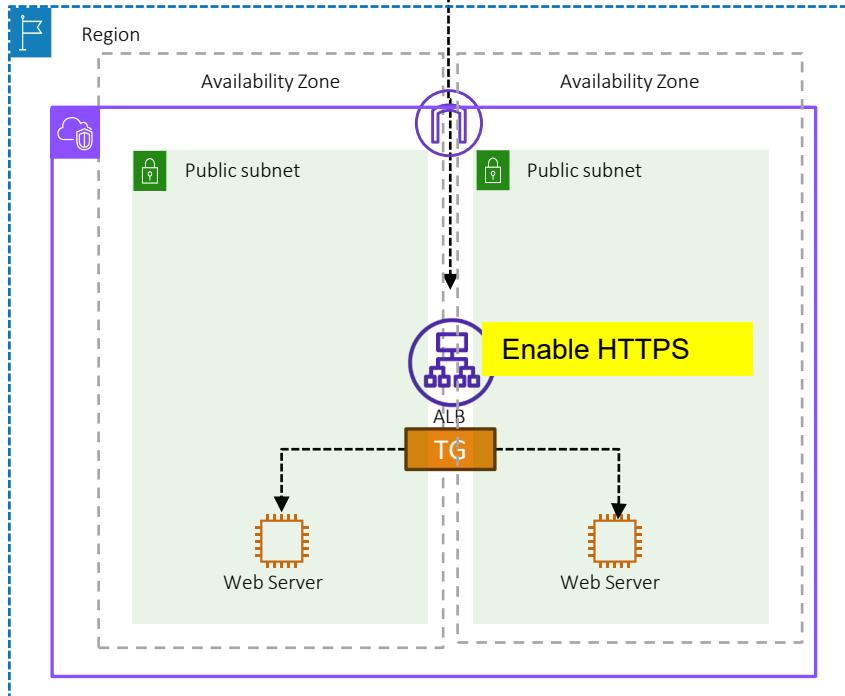
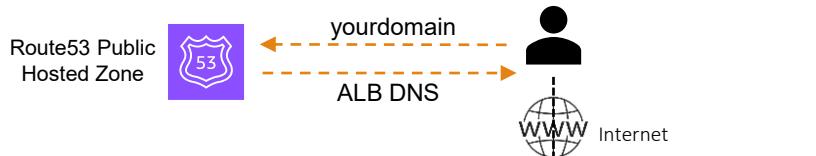


## Pre-requisite for this exercise:

- You should have your public domain name and DNS should be pointing to Route53 public hosted zone
- Refer Labs prerequisites section if you haven't done this earlier.

- 1 Exactly same setup as ALB exercise
- 2 In Route53 Public hosted zone create an A (Alias) record and point it to ALB DNS
- 2 Wait for some time and access application using your custom domain name

# Assignment 2 – Enable HTTPS

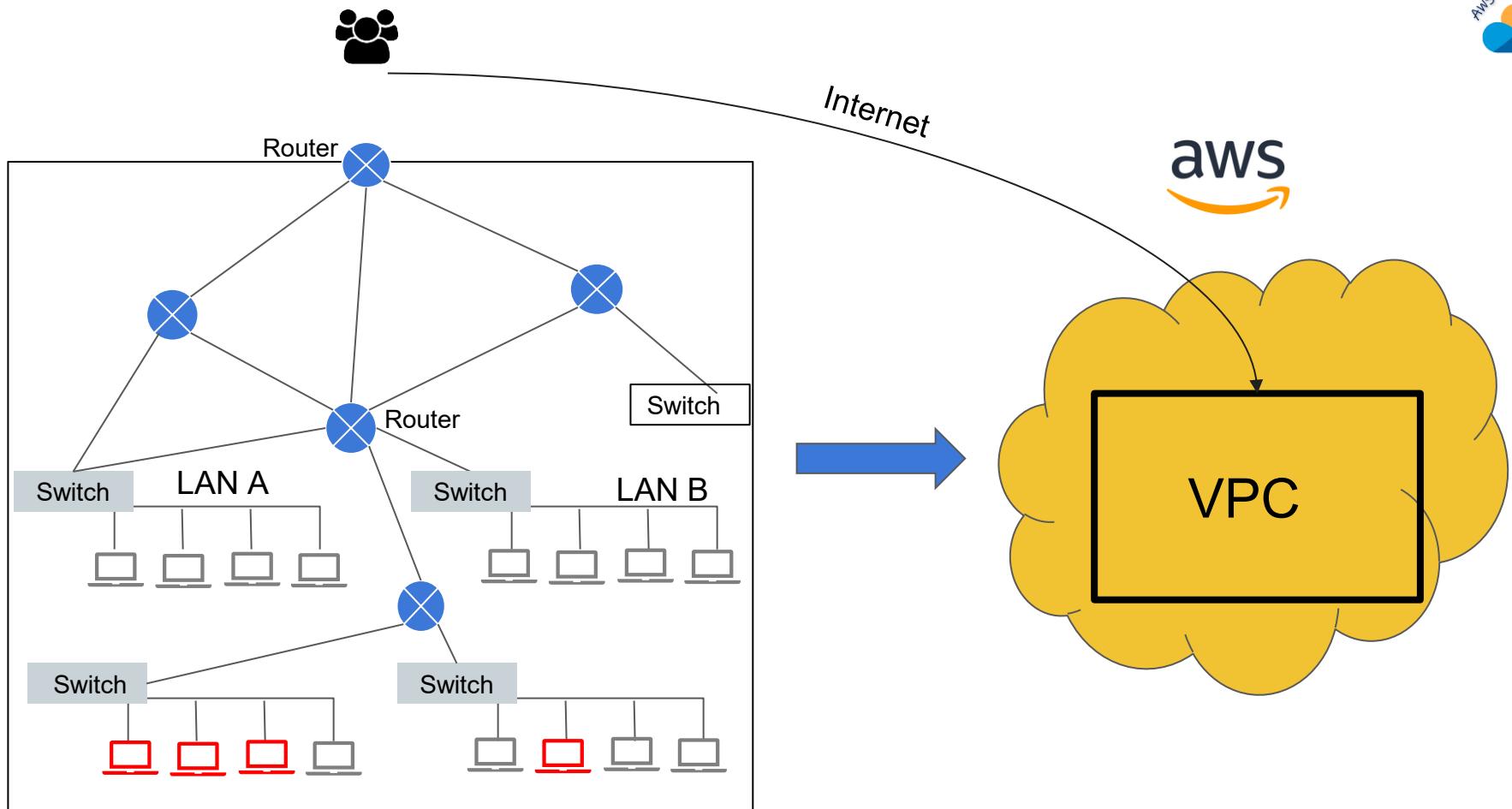


**Continuing with earlier setup**

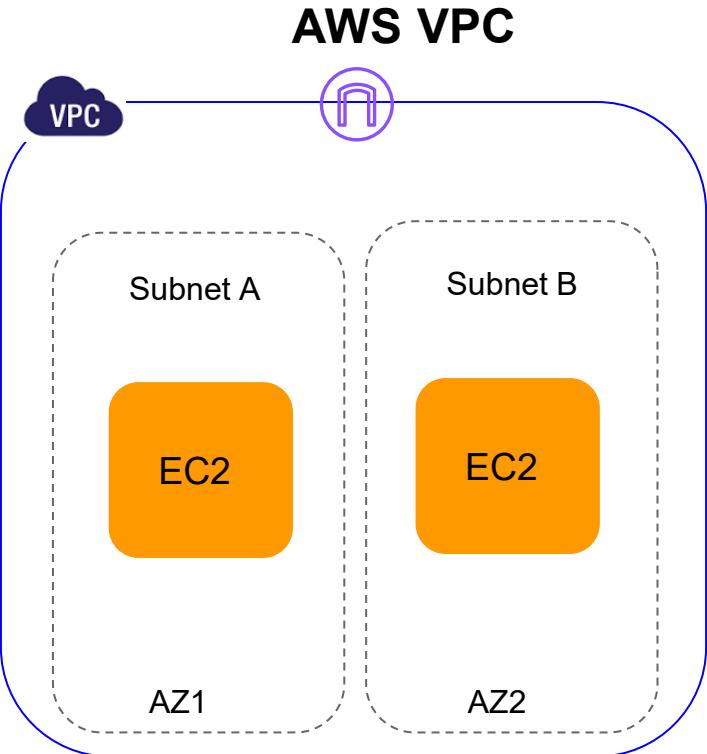
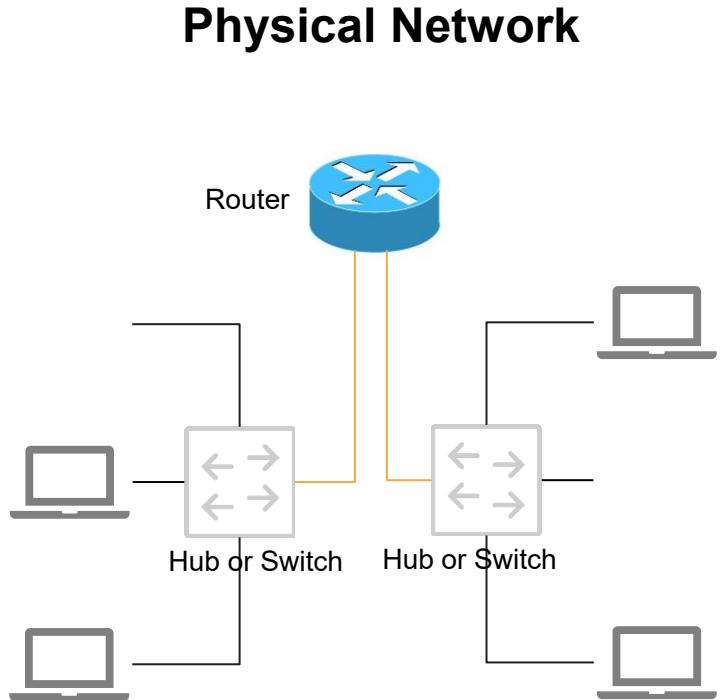
- 1 Get TLS certificate from Amazon ACM for your domain name. Validate the domain ownership through ACM portal.
- 2 Modify ALB listener to HTTPS (443) and associate TLS Certificate. Forward traffic to same target group as earlier.
- 3 Update ALB security group to allow HTTPS (443) traffic
- 4 Access website using <https://yourdomain>



# VPC and Networking



# Traditional IT network vs VPC

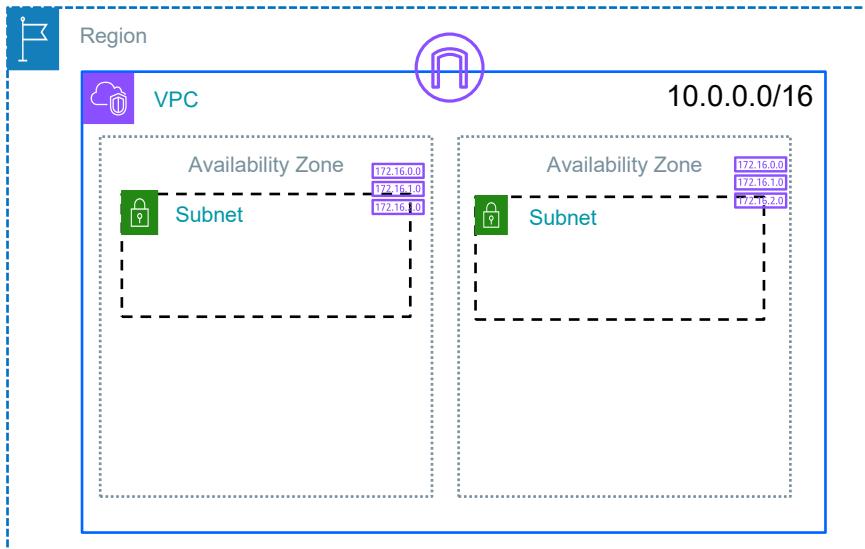


# Amazon Virtual Private Cloud (VPC)

- A logically isolated virtual network in the cloud which closely resembles the traditional IT network
- VPC is assigned a Private IP Address range (called CIDR) e.g. 10.0.0.0/16
- VPC can have both IPv4 (32 bit) or IPv6 (128 bit) IP addresses

## VPC building blocks:

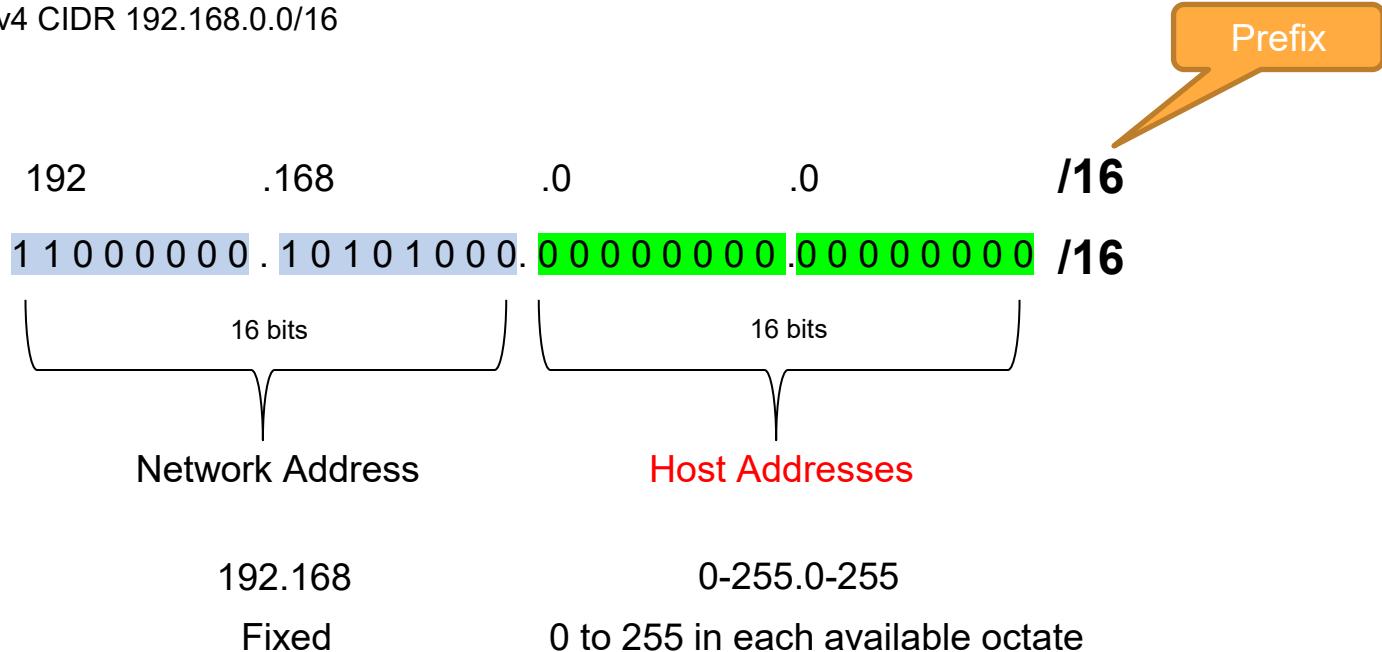
- VPC CIDR
- Subnets and Route table
- IP Addresses – IPv4 and IPv6
- Internet Gateway
- NAT Gateway
- VPC Firewalls - Security Group and Network ACL



# VPC Addressing - CIDR

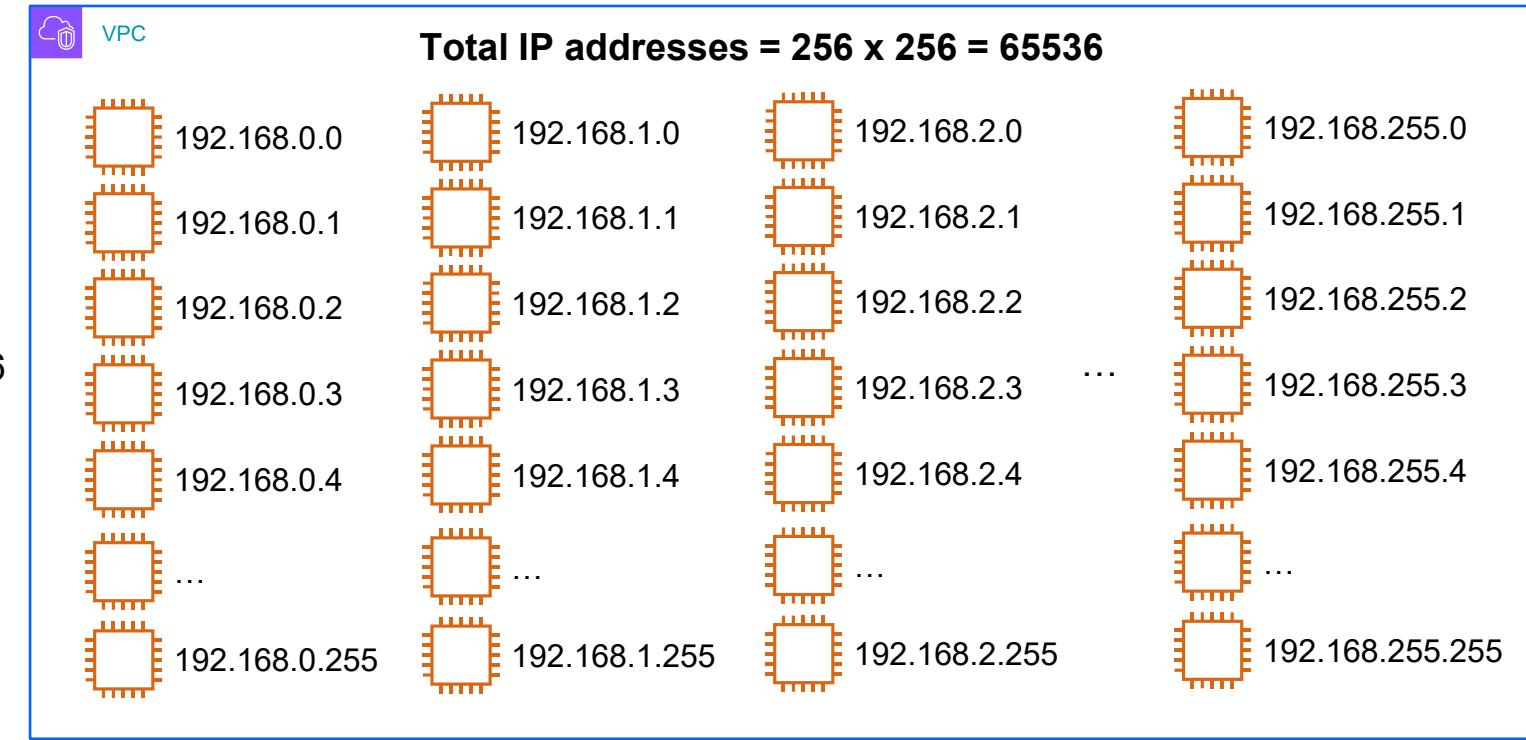
# CIDR – Classless Inter Domain Routing

- IP addressing scheme that replaces old address style of Class A, B, C
- Represented as an IP address and prefix
  - Example: IPv4 CIDR 192.168.0.0/16



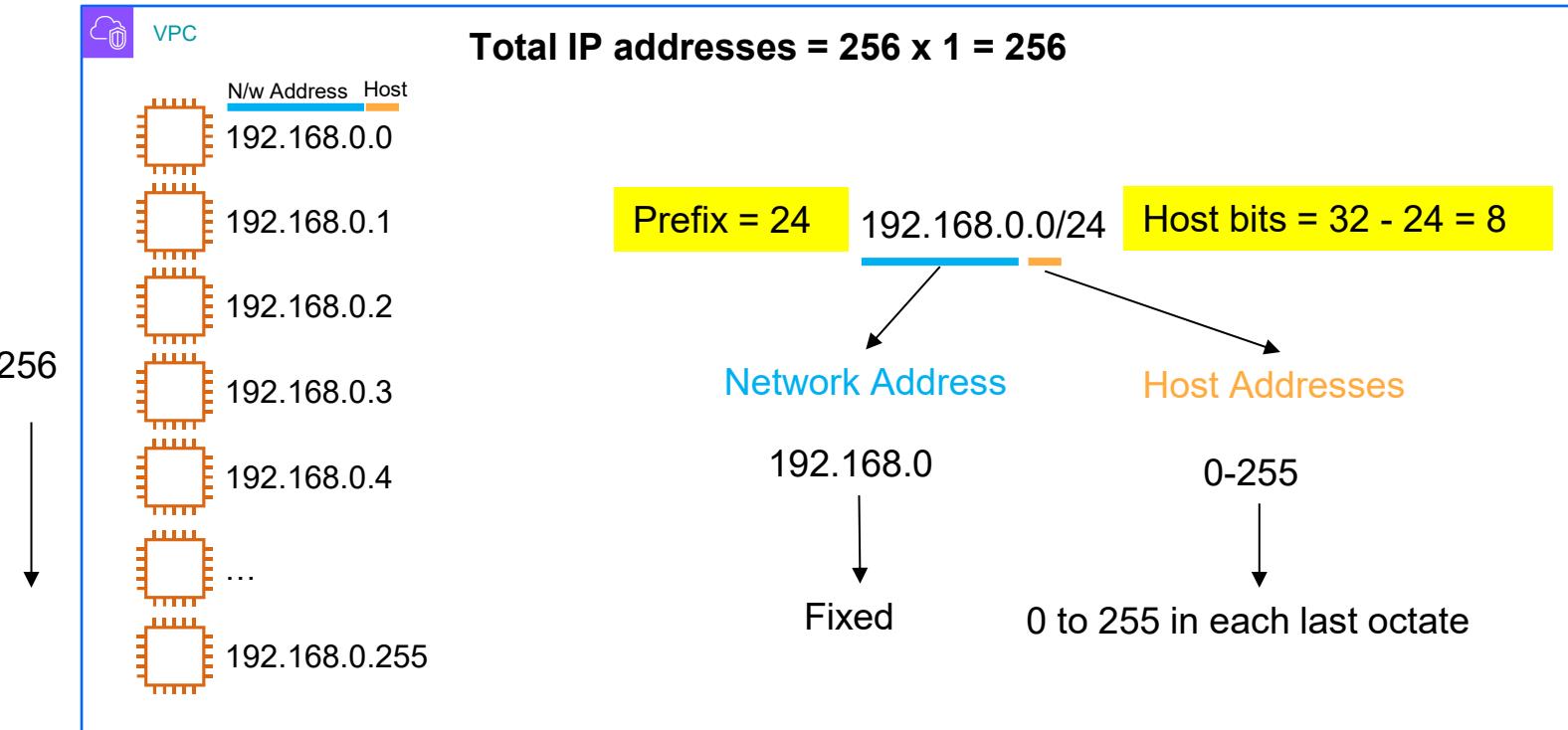
# CIDR – Classless Inter Domain Routing

192.168.0.0/16



# CIDR – Classless Inter Domain Routing

192.168.0.0/24

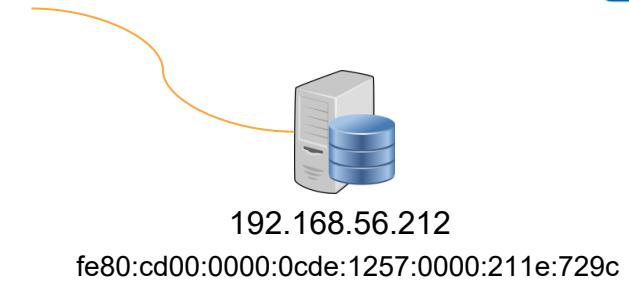


# VPC Addressing

- **AWS VPC CIDR (IPv4)**
  - VPC prefix between /16 (65536 IPs) and /28 (16 IPs)
  - RFC 1918 IP ranges for Private network and corresponding AWS recommended ranges
    - 10.0.0.0/8 => 10.0.0.0 – 10.255.255.255 => **AWS CIDR 10.X.0.0/16**
    - 172.16.0.0/12 => 172.16.0.0 - 172.31.255.255 => **AWS CIDR 172.16.0.0/16 to 172.31.0.0/16**
    - 192.168.0.0/16 => 192.168.0.0 - 192.168.255.255 => **AWS CIDR 192.168.0.0/16**
- **AWS VPC CIDR (IPv6)**
  - VPC CIDR with prefix /56 ( $2^{72}$  IPs)
  - IPv6 CIDR is allocated by AWS
  - IPv6 IP addresses are globally unique and publicly routable

# Public IP addresses

- AWS also assigns Public IP to EC2 instances (ENI)
- There are 2 types of IP addresses
  - IPv4 (32 bit)
  - IPv6 (128 bit)



**IPv4:** **192.168.56.212**

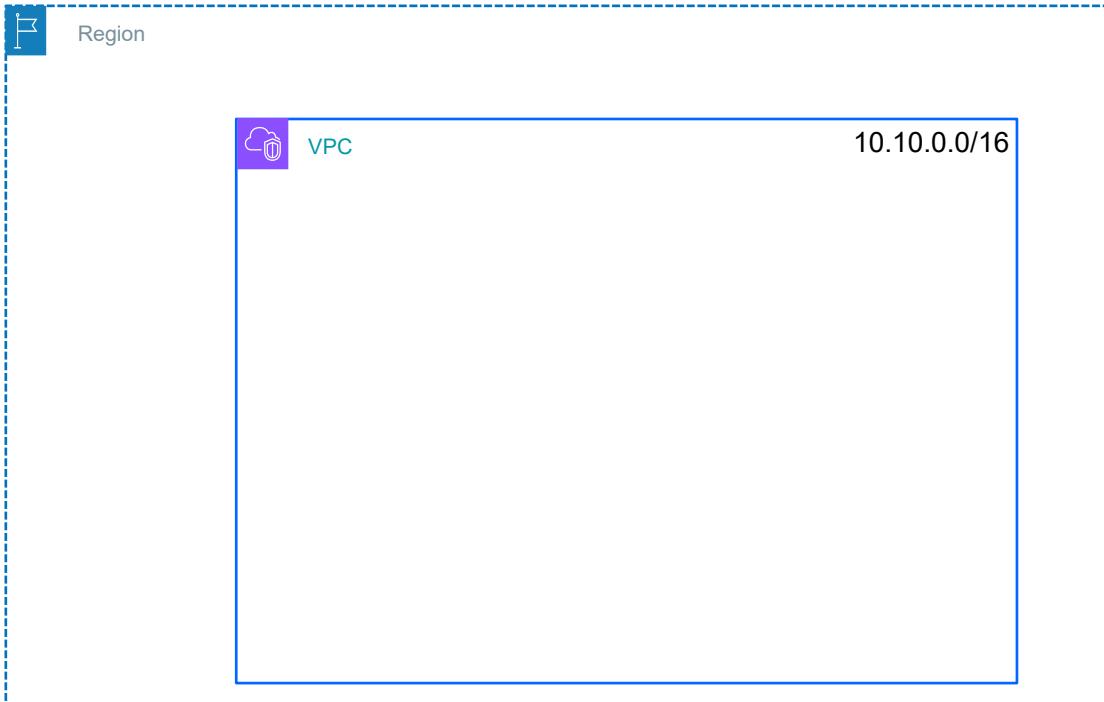
1 1 0 0 0 0 0 . 1 0 1 0 1 0 0 0 . 0 0 1 1 1 0 0 0 . 1 1 0 1 0 1 0 0  
8 bits            8 bits            8 bits            8 bits

**IPv6:**

**fe80:cd00:0000:0cde:1257:0000:211e:729c**

16 bits    16 bits  
128 bits

# Exercise – Create a VPC

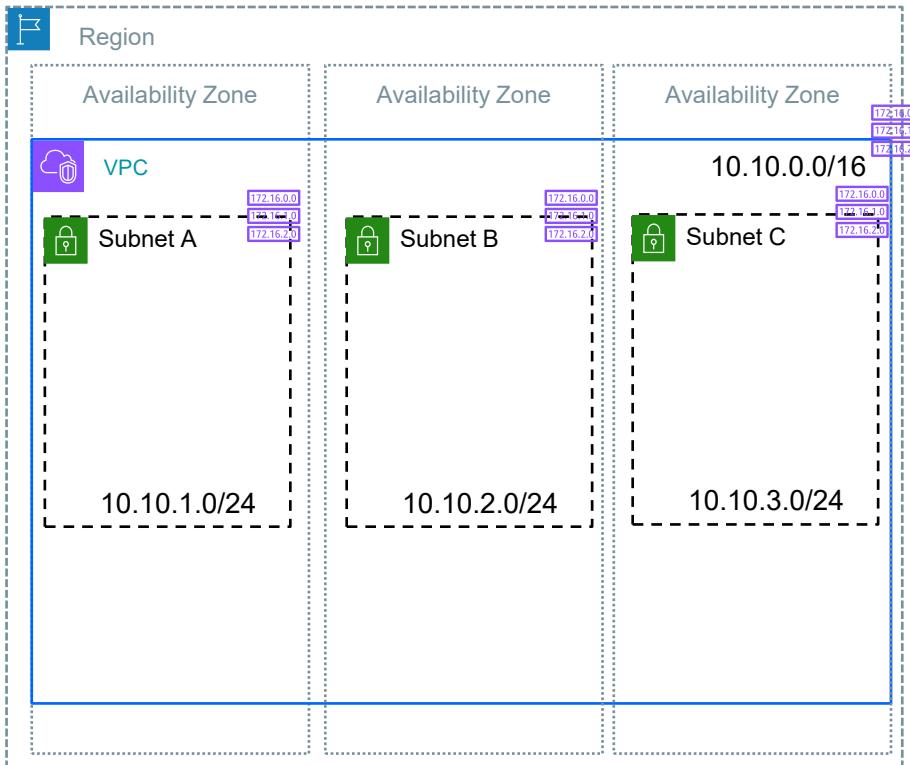


- Create a VPC with CIDR 10.10.0.0/16

# VPC Subnets, Route tables & Internet gateway

# VPC Subnets

- VPC network is partitioned into smaller networks called Subnets
- Subnets are created inside a specific Availability zone
- Route tables defines the routing logic for the subnets
- VPC has a main (default) route table which all subnets follow by default
- We can create subnet specific route tables.



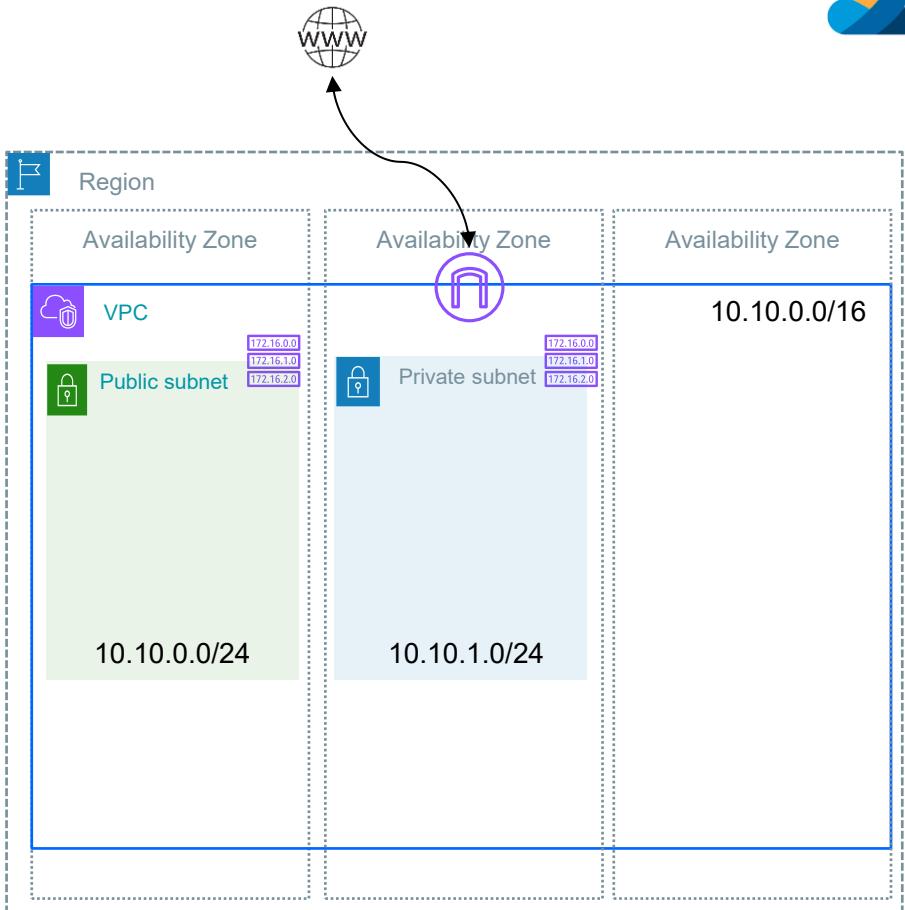
# Internet Gateway

- Internet gateway connects VPC to the internet
- If subnet route table has route to internet via the internet gateway, it's called Public subnet

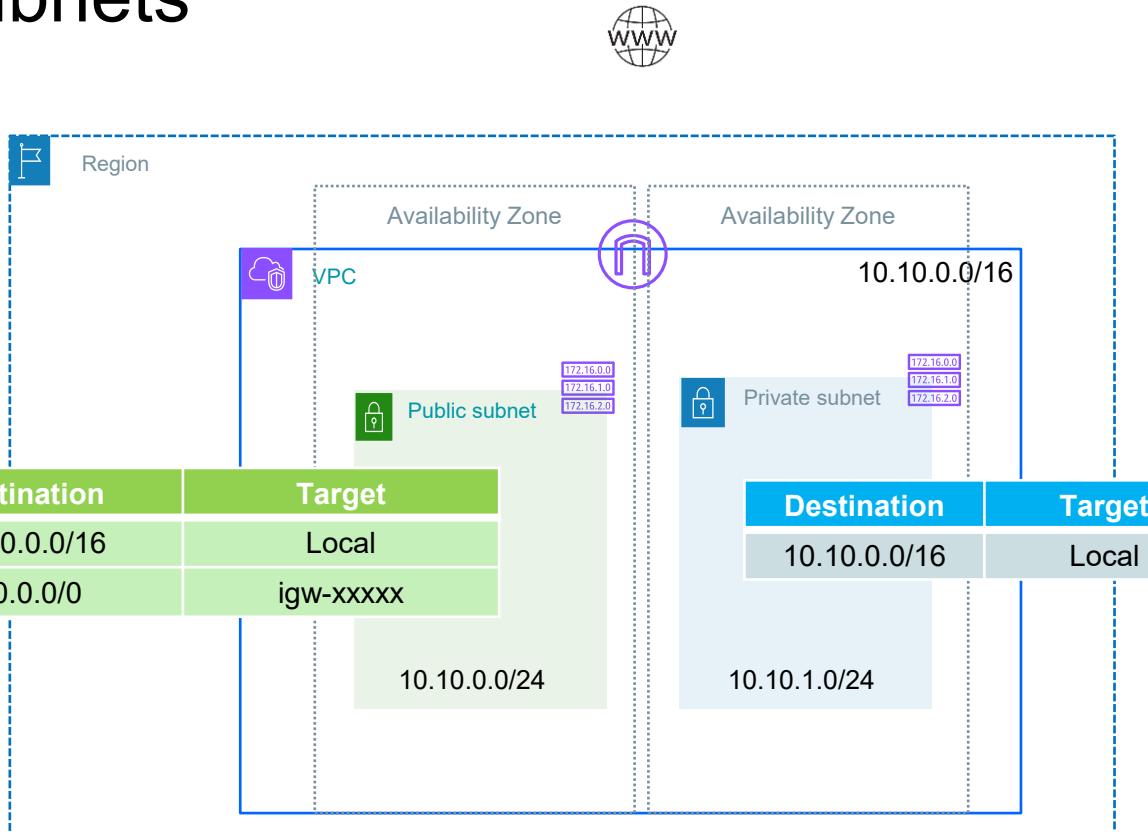
Destination	Target
10.10.0.0/16	Local
0.0.0.0/0	Internet Gateway (igw-x)
::/0	Internet Gateway (igw-x)

- If subnet route table does not have route to internet, it's called a Private subnet

Destination	Target
10.10.0.0/16	Local

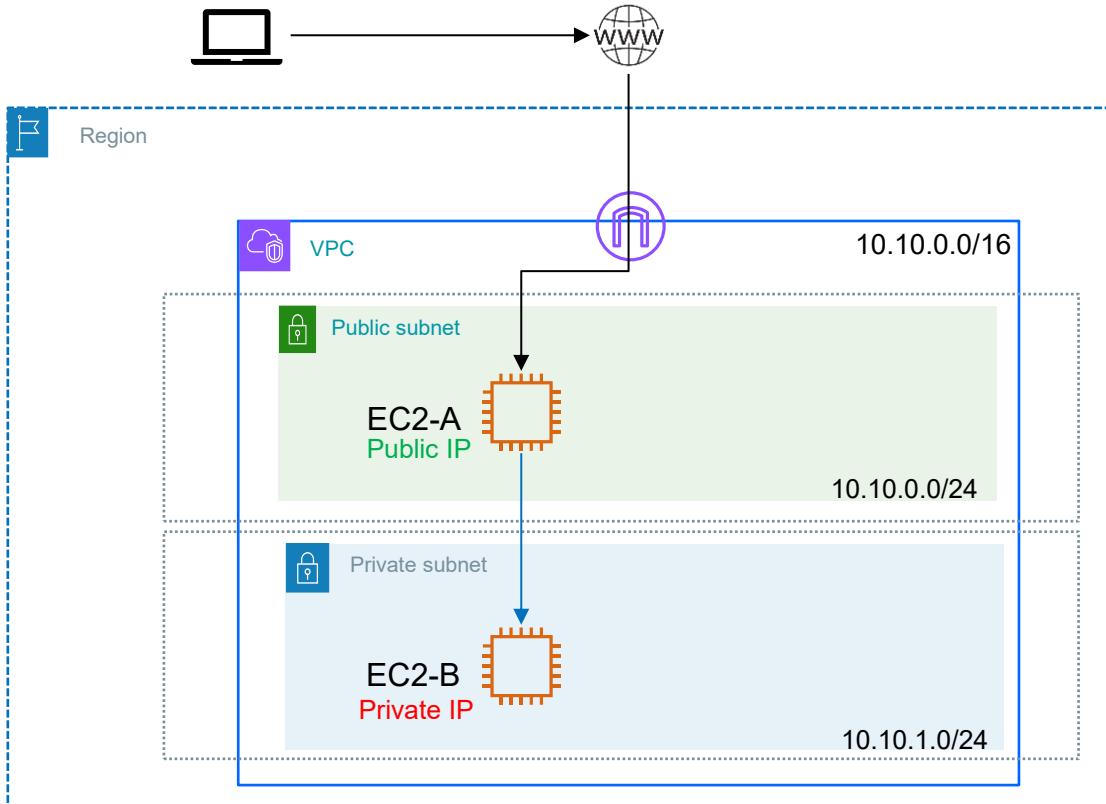


# Exercise – Add Internet gateway, Public/Private subnets



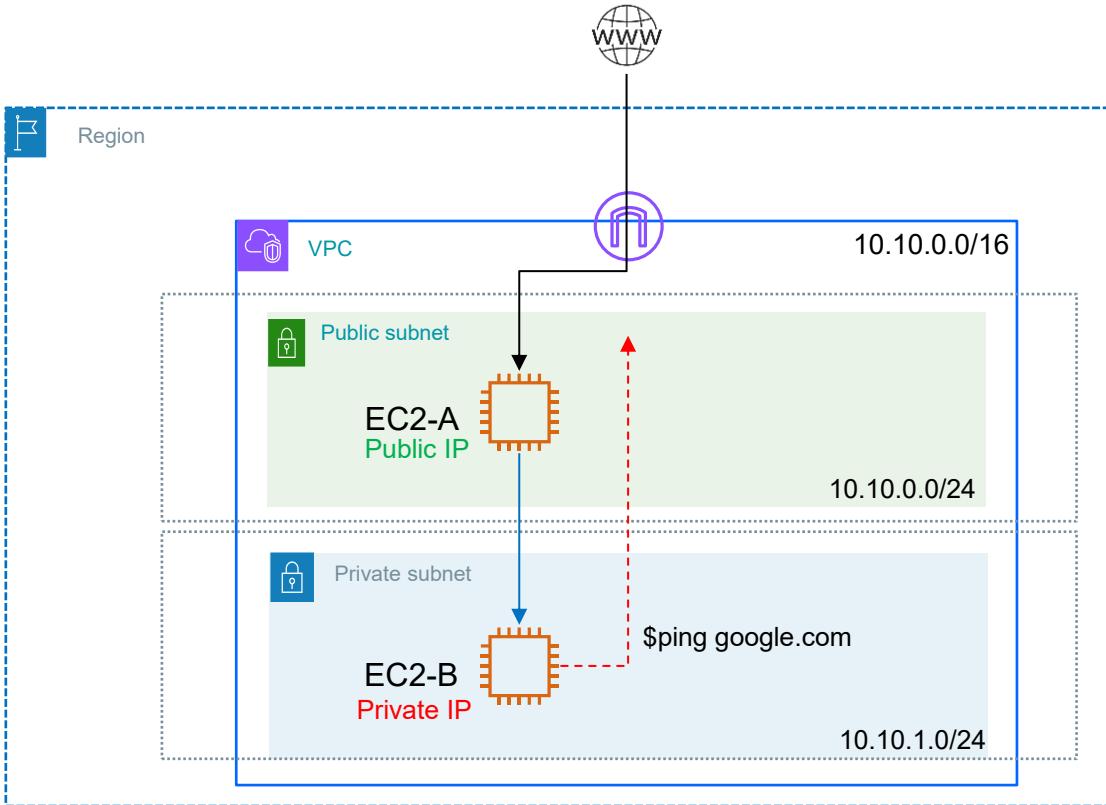
- Create a VPC with CIDR 10.10.0.0/16
- Create an Internet Gateway and associate with the VPC
- Create 2 subnets in 2 different availability zones
- Create a route table, add route entry for internet and associate with first subnet (Public subnet)
- Create another route table and associate with another subnet (Private subnet)

# Exercise – Launch EC2 instances



- Launch EC2-A in Public subnet (assign Public IP)
- Launch EC2-B in the Private subnet (no Public IP)
- Connect to EC2-A over SSH from your workstation.
- From EC2-A, SSH into EC2-B (for this you will need ssh private key on EC2-A instance)

# Exercise – Launch EC2 instances



- Launch EC2-A in Public subnet (assign Public IP)
- Launch EC2-B in the Private subnet (no Public IP)
- Connect to EC2-A over SSH from your workstation.
- From EC2-A, SSH into EC2-B (for this you will need ssh private key on EC2-A instance)
- After logged into EC2-B, try to ping to google.com

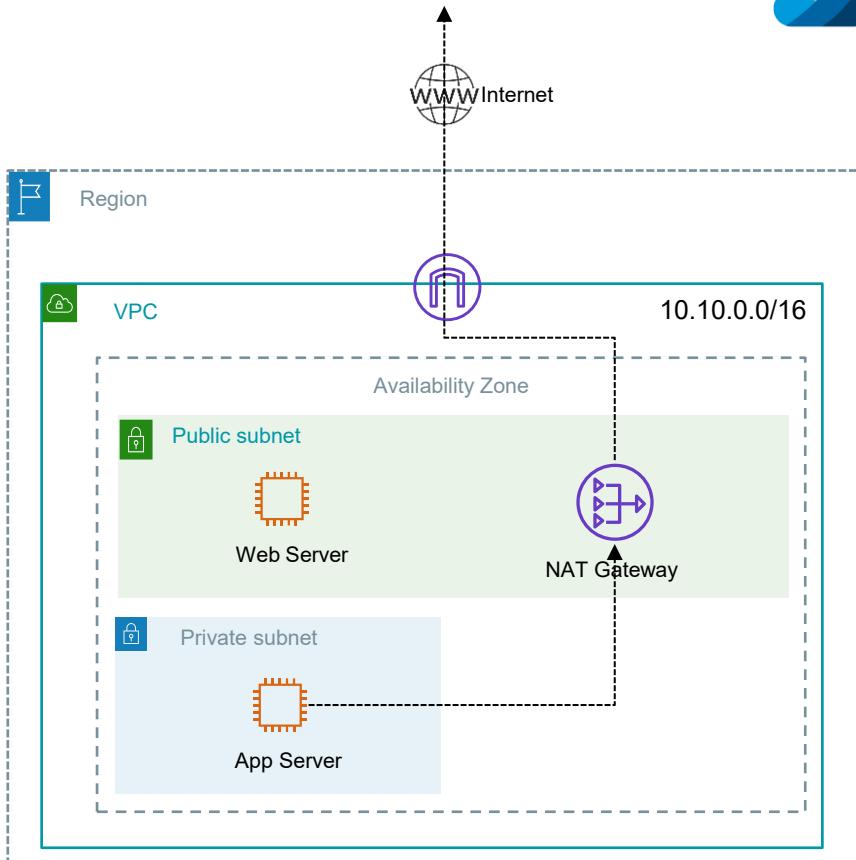
**Does it work?**

- EC2-B is not in a Public subnet
- EC2-B does not have a Public IP

# NAT Gateway

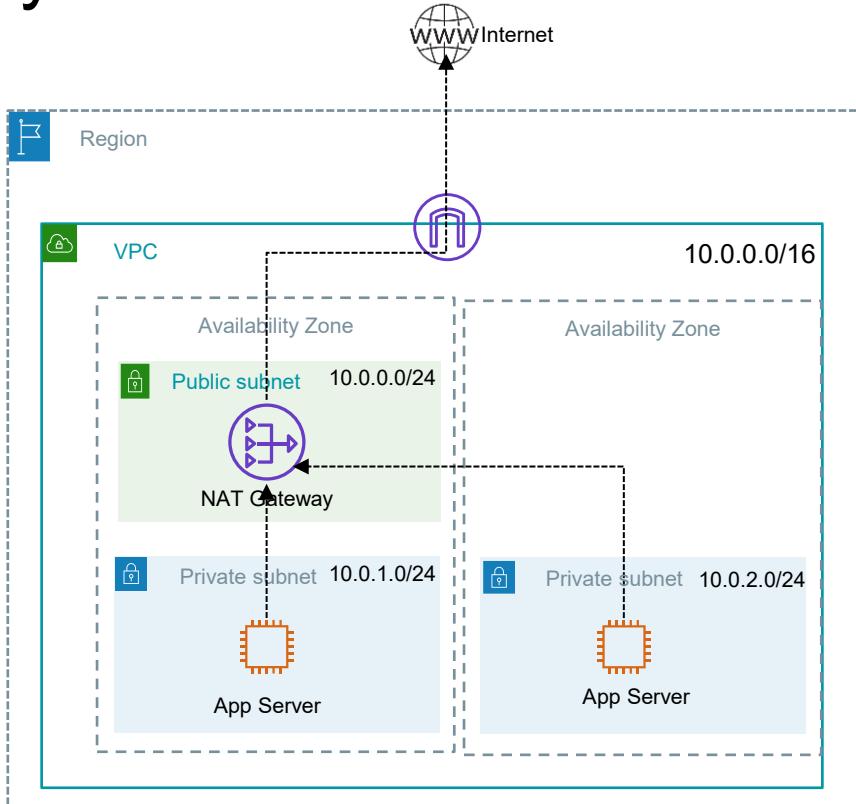
# NAT Gateway

- NAT Gateway allows instances in a private subnet to connect to services outside of the VPC, but external services cannot initiate a connection with those instances.
- It's AWS managed providing higher bandwidth, better availability, no administration
- Pay by the hour for usage and bandwidth
- 5 Gbps of bandwidth with automatic scaling up to 100 Gbps
- No security groups
- Supported protocols: TCP, UDP, and ICMP
- For outbound internet access, NAT Gateway should be created in Public Subnet and should be allocated an Elastic IP



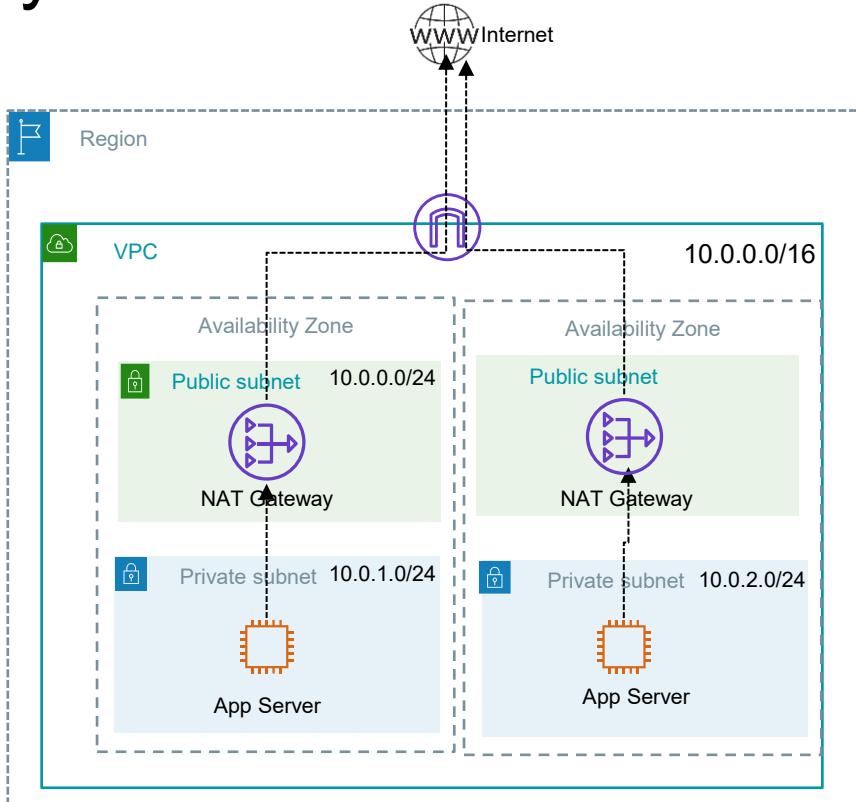
# NAT Gateway High availability

- NAT Gateways are highly available within a single AZ
- For HA across multiple AZs, multiple NAT gateways can be launched

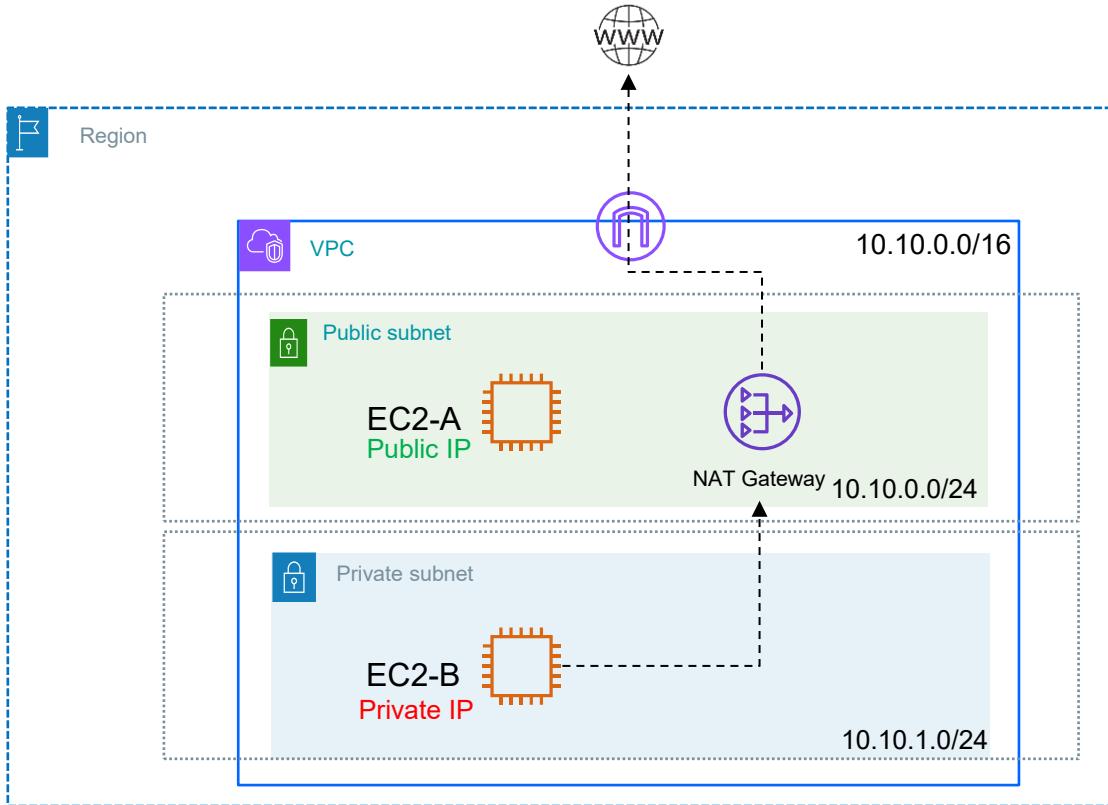


# NAT Gateway High availability

- NAT Gateways are highly available within a single AZ
- For HA across multiple AZs, multiple NAT gateways can be launched



# Exercise – NAT Gateway

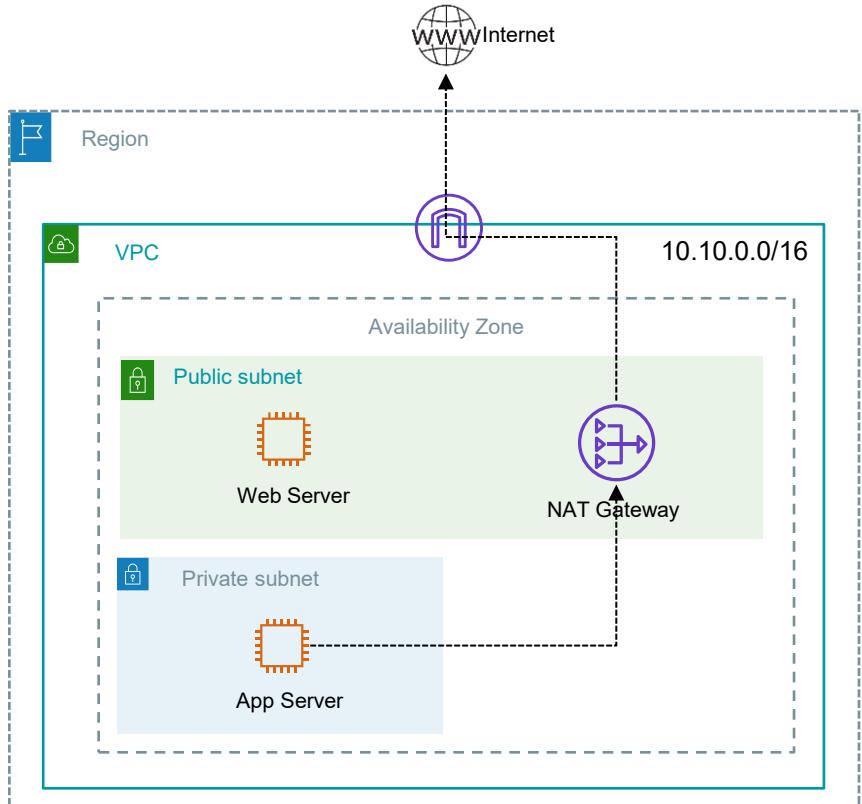


- Create a NAT gateway in the Public Subnet, assign Elastic IP
- Update Private subnet route table and add entry for the destination 0.0.0.0/0 with target as a NAT gateway
- Check the outbound internet connectivity from EC2-B e.g. ping google.com

## Does it work?

- The outbound internet traffic is sent through the NAT gateway

# Exercise - NAT Gateway



- Launch EC2-A in Public subnet (assign Public IP)
- Launch EC2-B in the Private subnet (no Public IP)
- Connect to EC2-A over SSH from your workstation.
  - From EC2-A
  - Create another Public subnet C
  - Create NAT gateway in this new subnet.
  - Update Private subnet route table and add entry for the destination 0.0.0.0/0 with target as NAT gateway.
  - Check the outbound internet connectivity from EC2-B

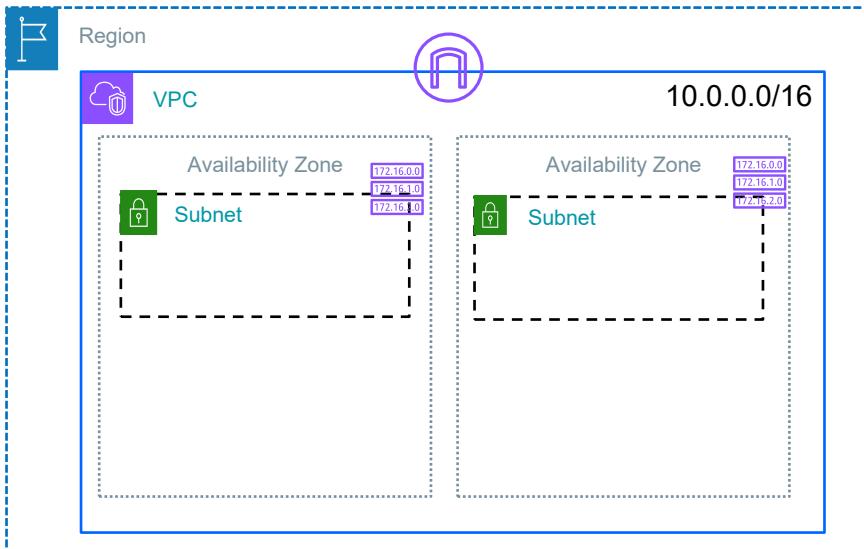
# VPC Firewall - Security group

# Amazon Virtual Private Cloud (VPC)

- A logically isolated virtual network in the cloud which closely resembles the traditional IT network
- VPC is assigned a Private IP Address range (called CIDR) e.g. 10.0.0.0/16
- VPC can have both IPv4 (32 bit) or IPv6 (128 bit) IP addresses

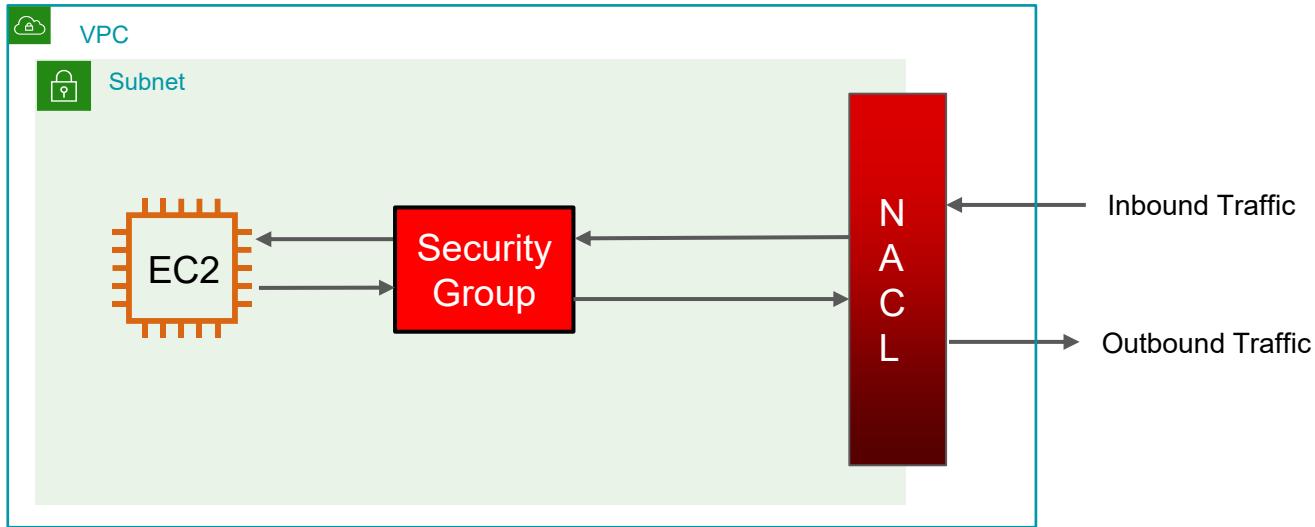
## VPC building blocks:

- VPC CIDR
- Subnets and Route table
- IP Addresses – IPv4 and IPv6
- Internet Gateway
- NAT Gateway
- **VPC Firewalls - Security Group and Network ACL**



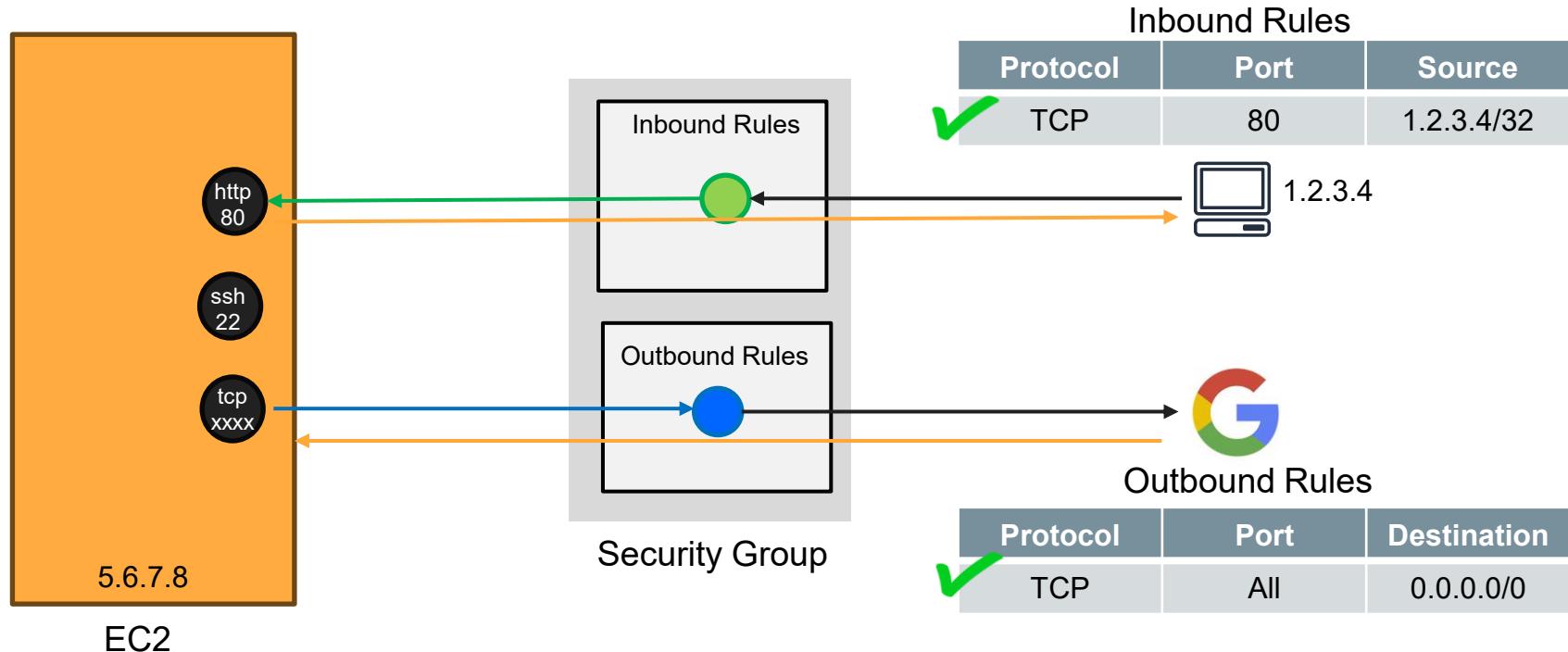
# Firewalls inside VPC

- Security Groups
- Network Access Control List (NACL)

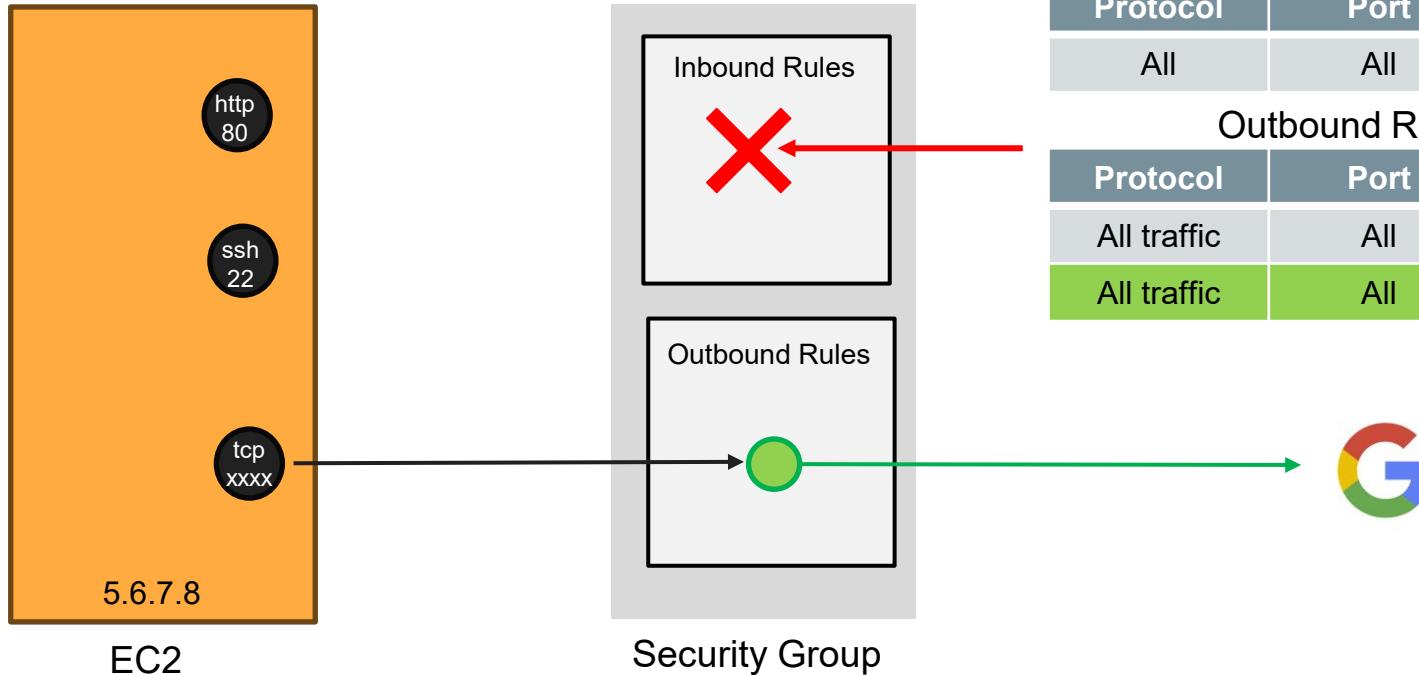


# Security Group

- Security Groups are most basic, native and important firewall for EC2 instances (ENIs)
- Security group has Inbound and Outbound rule



# Default Security Group



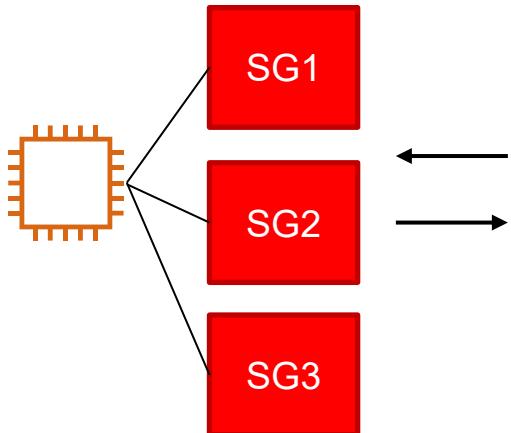
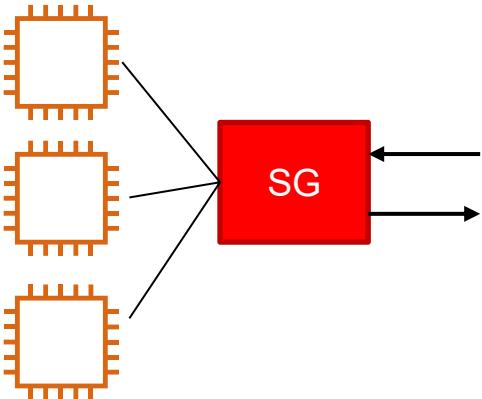
# Default Security Group

Inbound			
Source	Protocol	Port range	Description
sg- 1234567890abcdef0	All	All	Allows inbound traffic from all resources that are assigned to this security group. The source is the ID of this security group.

Outbound			
Destination	Protocol	Port range	Description
0.0.0.0/0	All	All	Allows all outbound IPv4 traffic.
::/0	All	All	Allows all outbound IPv6 traffic. This rule is added only if your VPC has an associated IPv6 CIDR block.

# Security Groups - Summary

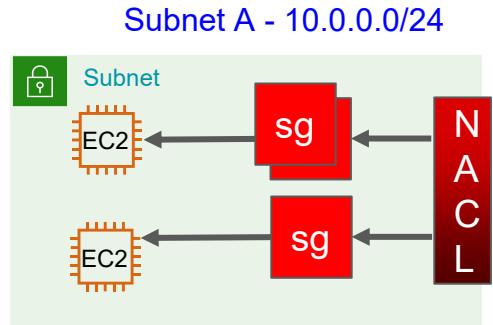
- Single Security Group can be attached to multiple instances
- Single Instance can have multiple Security groups
- All inbound traffic is blocked by default
- All outbound traffic is authorised by default
- Authorises traffic for both IPv4 and IPv6 traffic
- Security groups are stateful



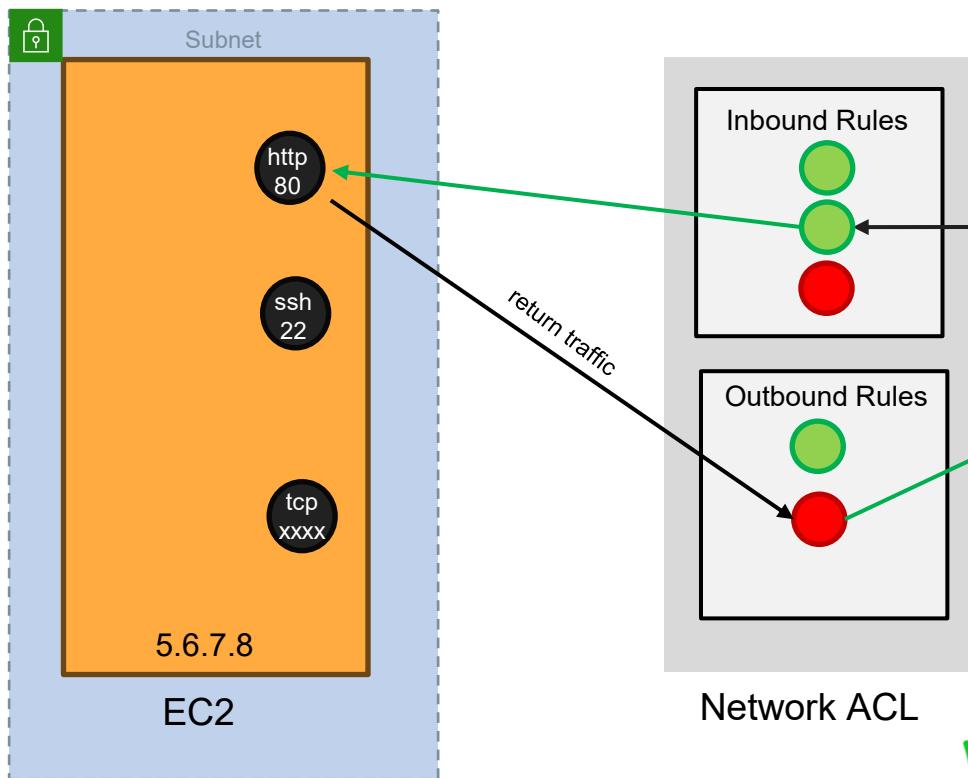
# VPC Firewall - Network ACLs

# Network Access Control List (NACL)

- Works at Subnet level – Hence automatically applied to all instances
- Contains both Allow and Deny rules. Rules are numbered.
- Rules are evaluated in the order of rule number (1 to 32766)
- Stateless – We need to explicitly open ports for return traffic



# Network ACL – Allow return traffic



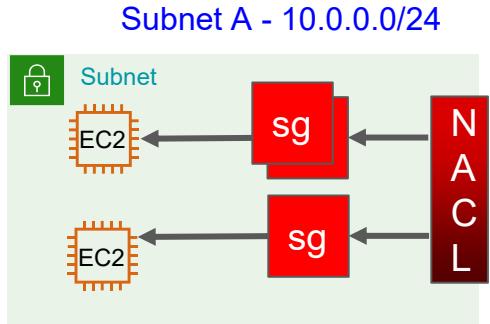
Inbound Rules				
Rule number	Protocol	Port	Source	Allow/Deny
100	TCP	80	1.2.3.4/32	Allow
200	TCP	80	9.10.11.12/32	Allow
*	All IPv4 traffic	All	0.0.0.0/0	Deny

9.10.11.12  
(port: XXXX)

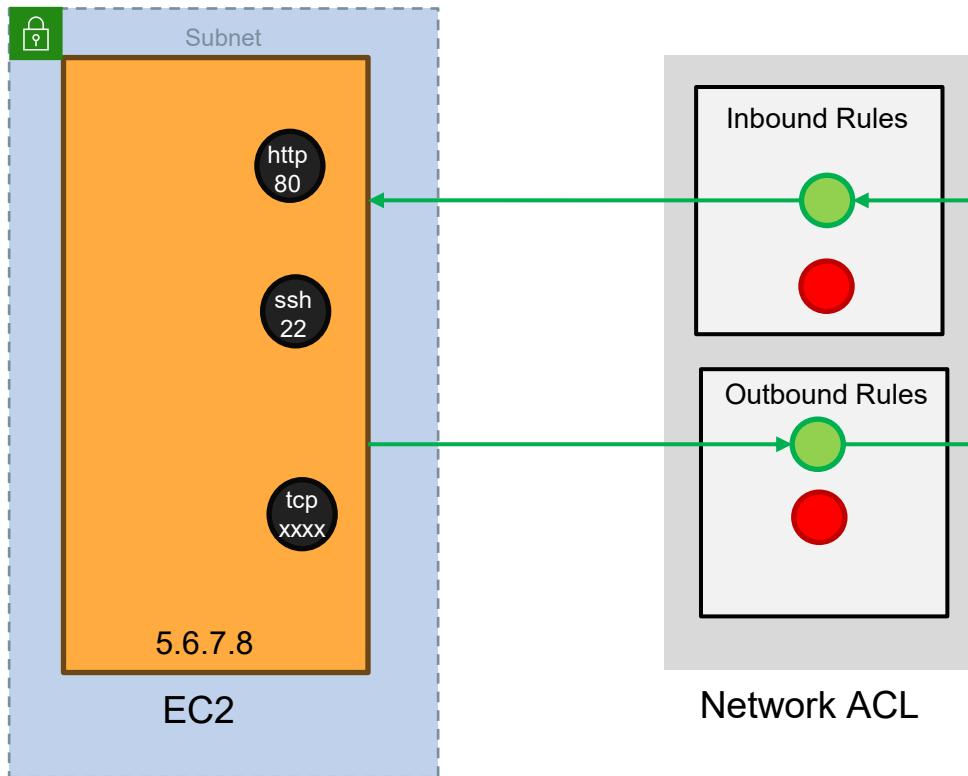
Outbound Rules				
Rule number	Protocol	Port	Destination	Allow/Deny
100	TCP	XXXX	1.2.3.4/32	Allow
200	TCP	XXXX	9.10.11.12/32	Allow
*	All IPv4 traffic	All	0.0.0.0/0	Deny

# Network Access Control List (NACL)

- Works at Subnet level – Hence automatically applied to all instances
- Contains both Allow and Deny rules. Rules are numbered.
- Rules are evaluated in the order of rule number (1 to 32766)
- Stateless – We need to explicitly open ports for return traffic
- Default NACL allows all inbound and outbound traffic



# Default Network ACL

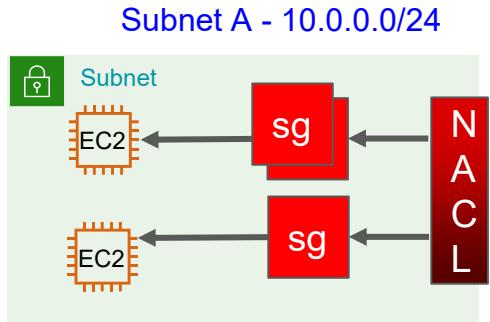


Inbound Rules				
Rule number	Protocol	Port	Source	Allow/Deny
100	All IPv4 traffic	All	0.0.0.0/0	Allow
*	All IPv4 traffic	All	0.0.0.0/0	Deny

Outbound Rules				
Rule number	Protocol	Port	Destination	Allow/Deny
100	All IPv4 traffic	All	0.0.0.0/0	Allow
*	All IPv4 traffic	All	0.0.0.0/0	Deny

# Network Access Control List (NACL)

- Works at Subnet level – Hence automatically applied to all instances
- Contains both Allow and Deny rules. Rules are numbered.
- Rules are evaluated in the order of rule number (1 to 32766)
- Stateless – We need to explicitly open ports for return traffic
- Default NACL allows all inbound and outbound traffic
- **NACL are a great way of blocking a specific IP at the subnet level**



Network ACL inbound rules

#Rule	Type	Protocol	Port	Source	Allow/Deny	
100	All IPv4 traffic	All	All	180.151.138.43/32	DENY	
101	HTTPS	TCP	443	0.0.0.0/0	ALLOW	
*	All IPv4 traffic	All	All	0.0.0.0/0	DENY	

# Security Groups vs Network ACL

## Security Group

Operates at EC2 instance

Supports only Allow rules

Stateful – Return traffic is allowed

All rules are evaluated before making a decision

## Network ACL

Operates at Subnet level

Supports both Allow and Deny rules

Stateless – Return traffic needs to be authorized in Outbound rules

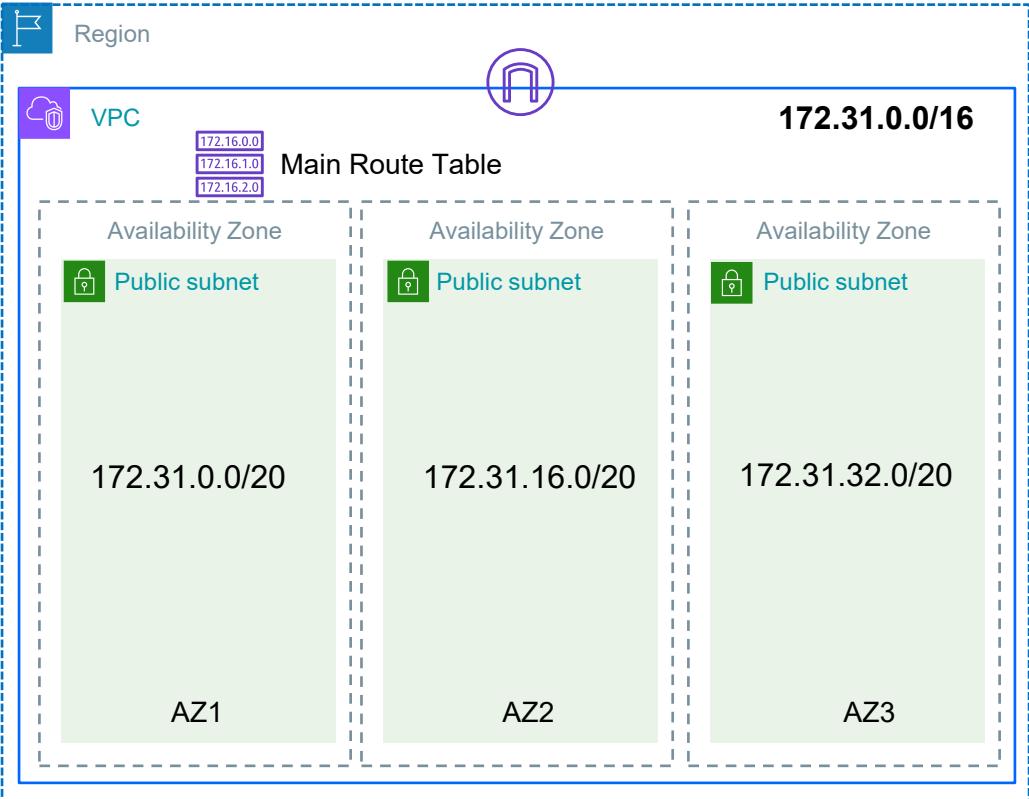
Rules are evaluated in the order (lower to higher) and first matching rule is applied

# Default VPC

# Default VPC

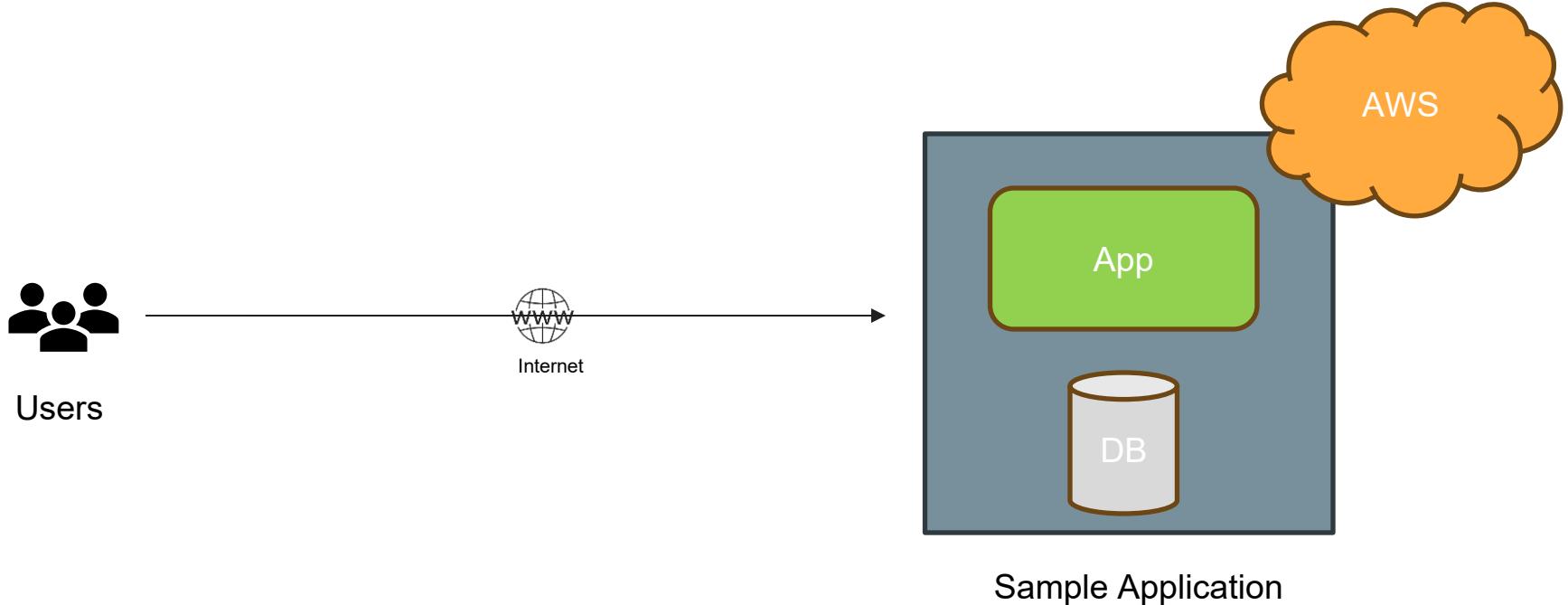
- AWS creates Default VPC in each AWS region
- Creates VPC with CIDR - 172.31.0.0/16
- Creates 1 subnet in every AZ
- Creates Internet Gateway
- All subnets are public subnets
- If deleted, you can recreate default VPC

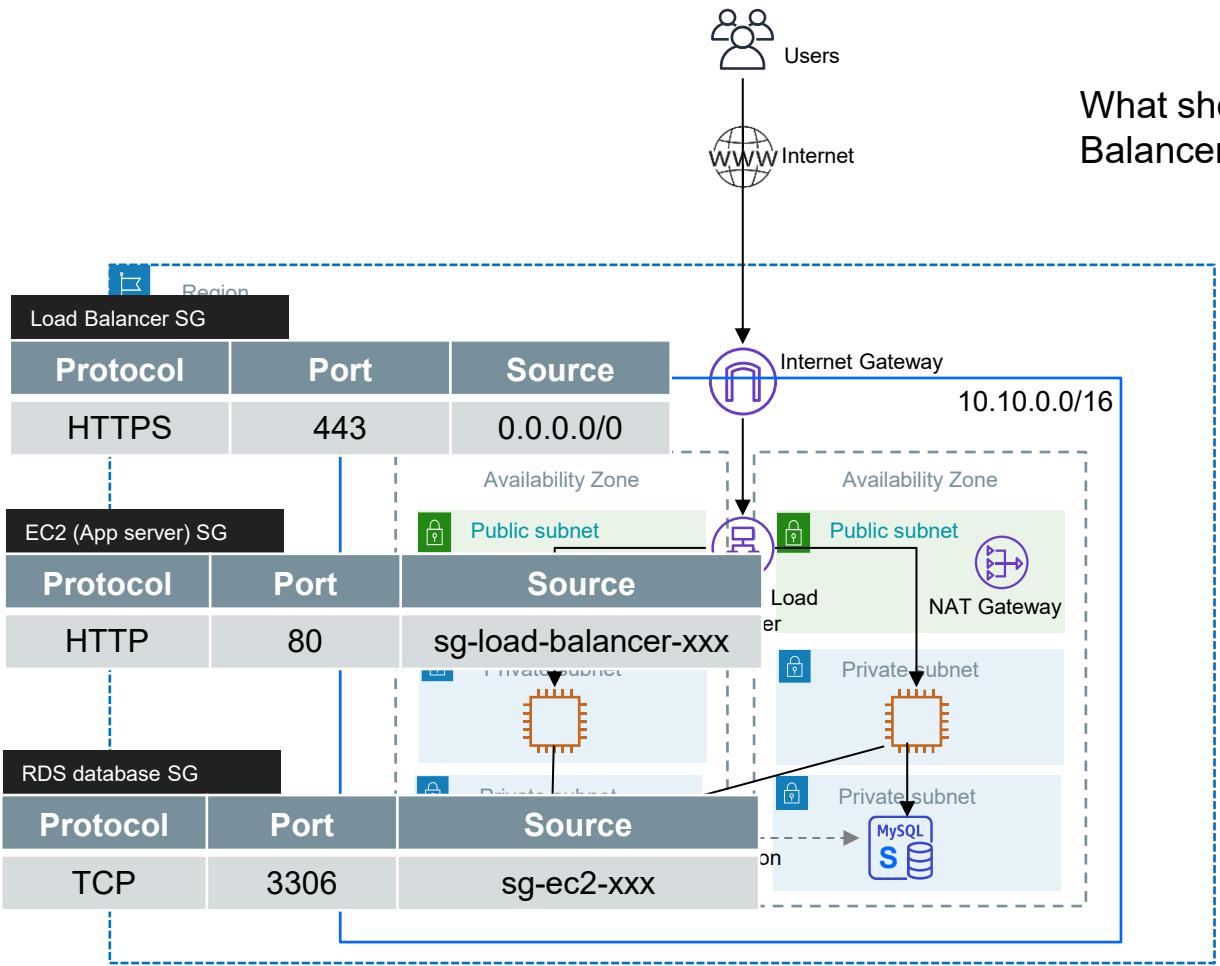
Used as a default network for launching EC2 or other resources if you didn't explicitly create your own VPC



# VPC Design for hosting a web application

# We want to deploy this application in AWS





What should be security groups for Load Balancer, EC2 (app servers) and Database?

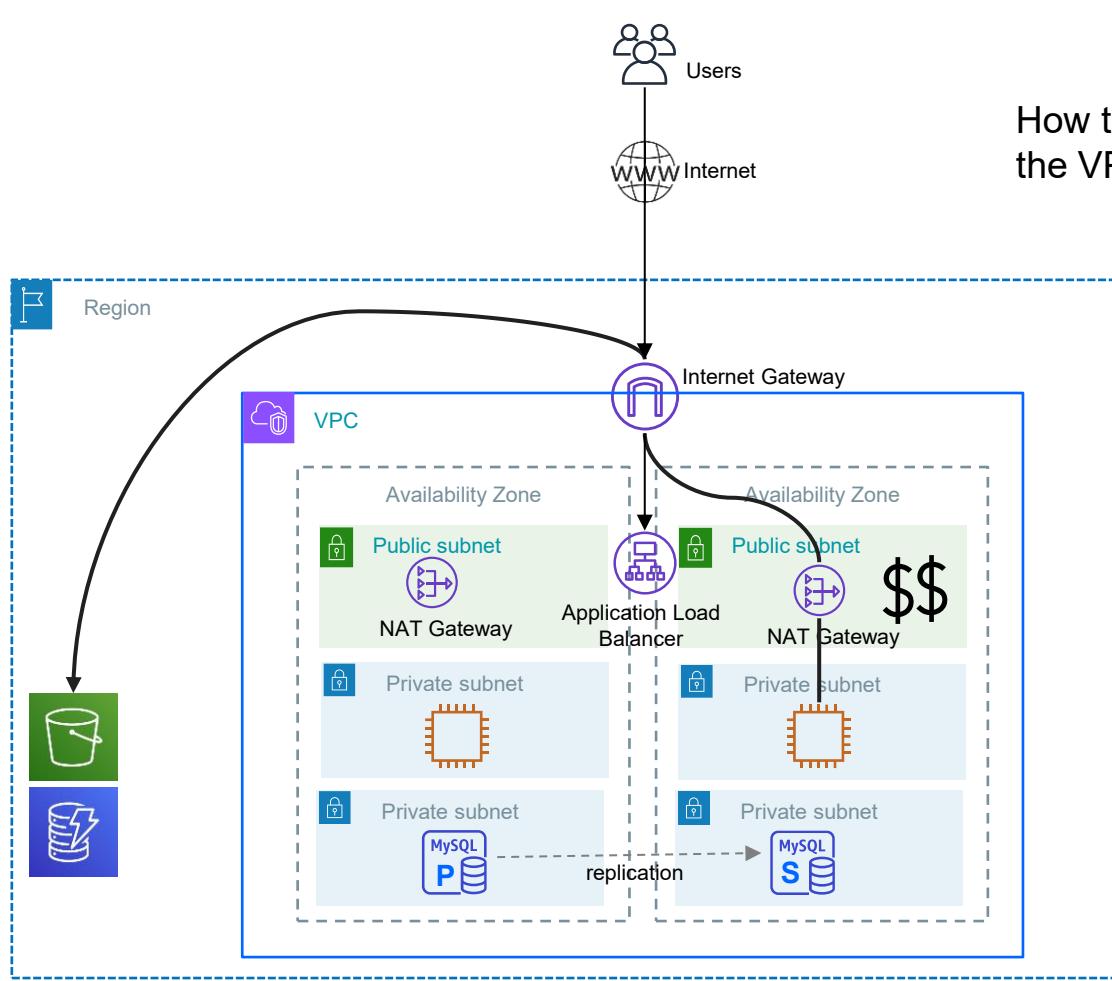
- Attach **Internet Gateway** for connecting VPC to internet
- **Public IPs** for the ALB
- A **Public subnet** can communicate with the internet directly using Internet gateway
- Two subnets in two different Availability Zones for High Availability
- **Private subnet** cannot directly communicate with the internet
- **Private IPs** for the instances
- A **NAT gateway** in the **Public subnet**

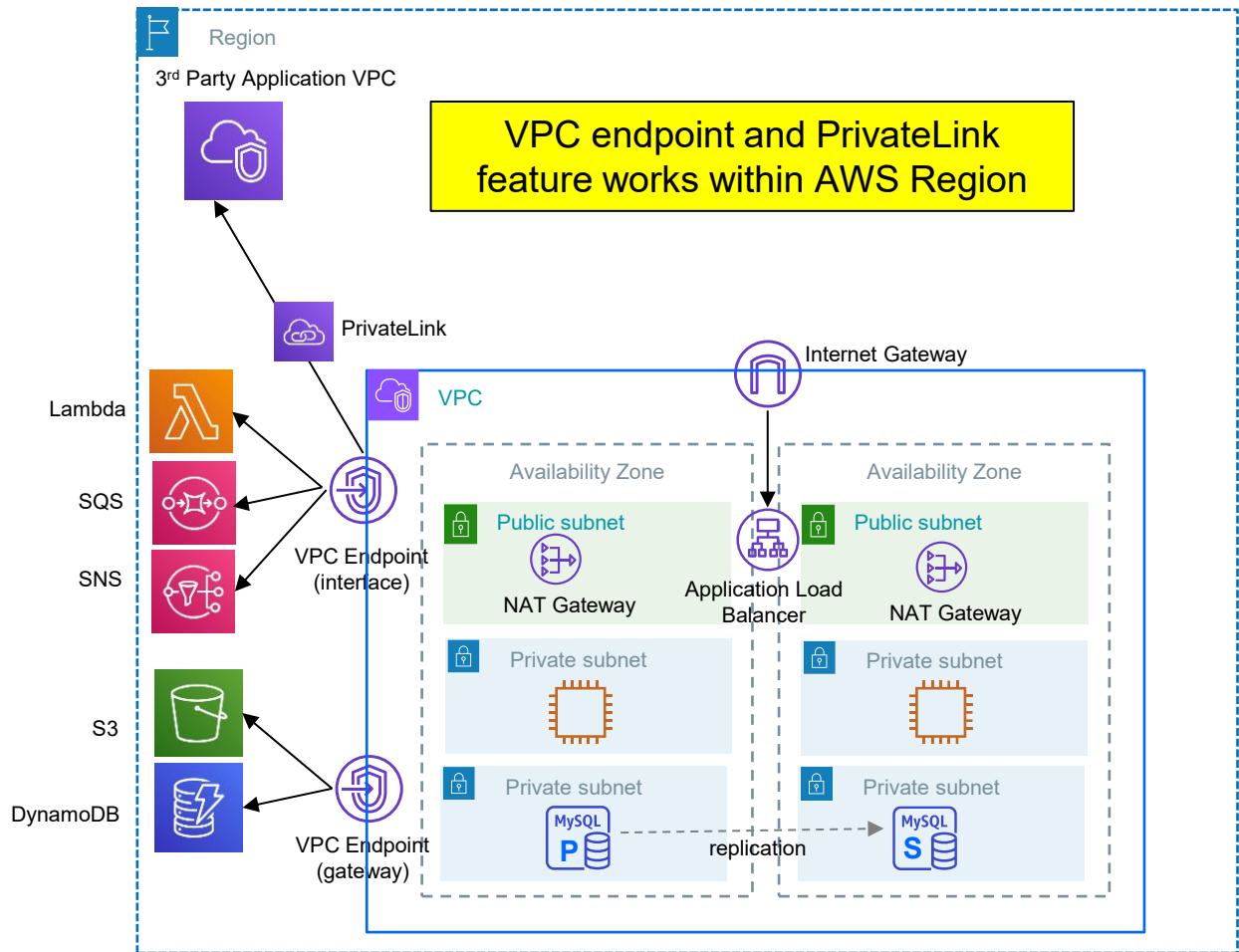
# VPC Private Connectivity options

# VPC connectivity options

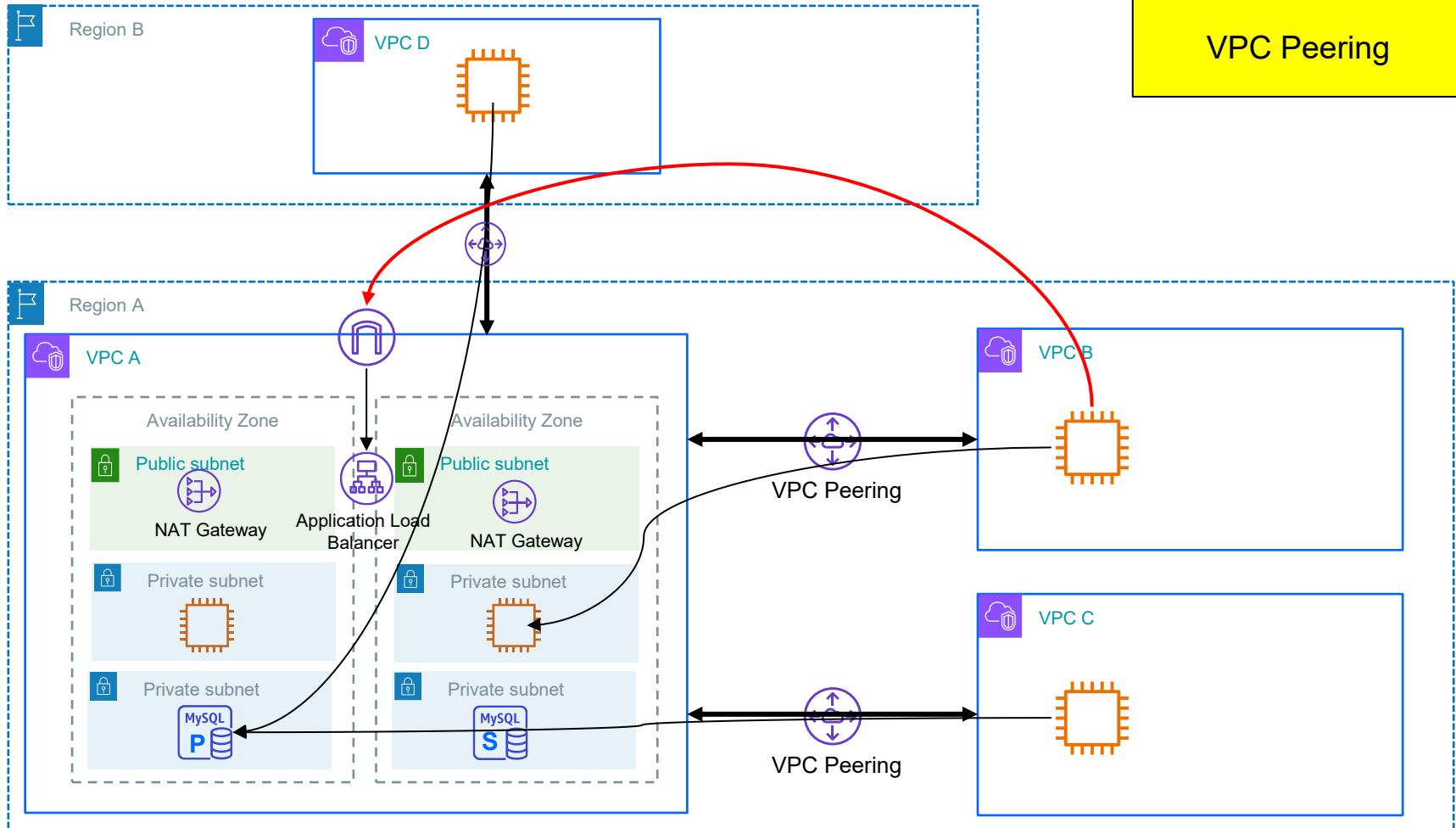
- VPC Endpoints & PrivateLink
- VPC Peering connection
- AWS Transit Gateway
- AWS Site-2-Site VPN
- AWS Client VPN
- AWS Direct Connect

## How to access AWS services from within the VPC?



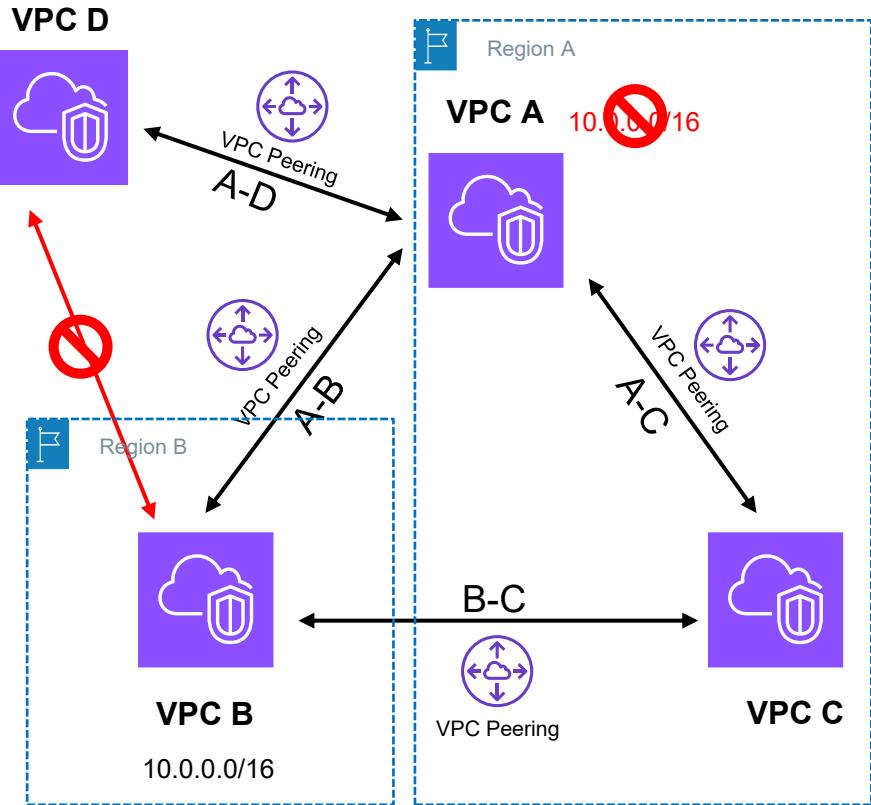


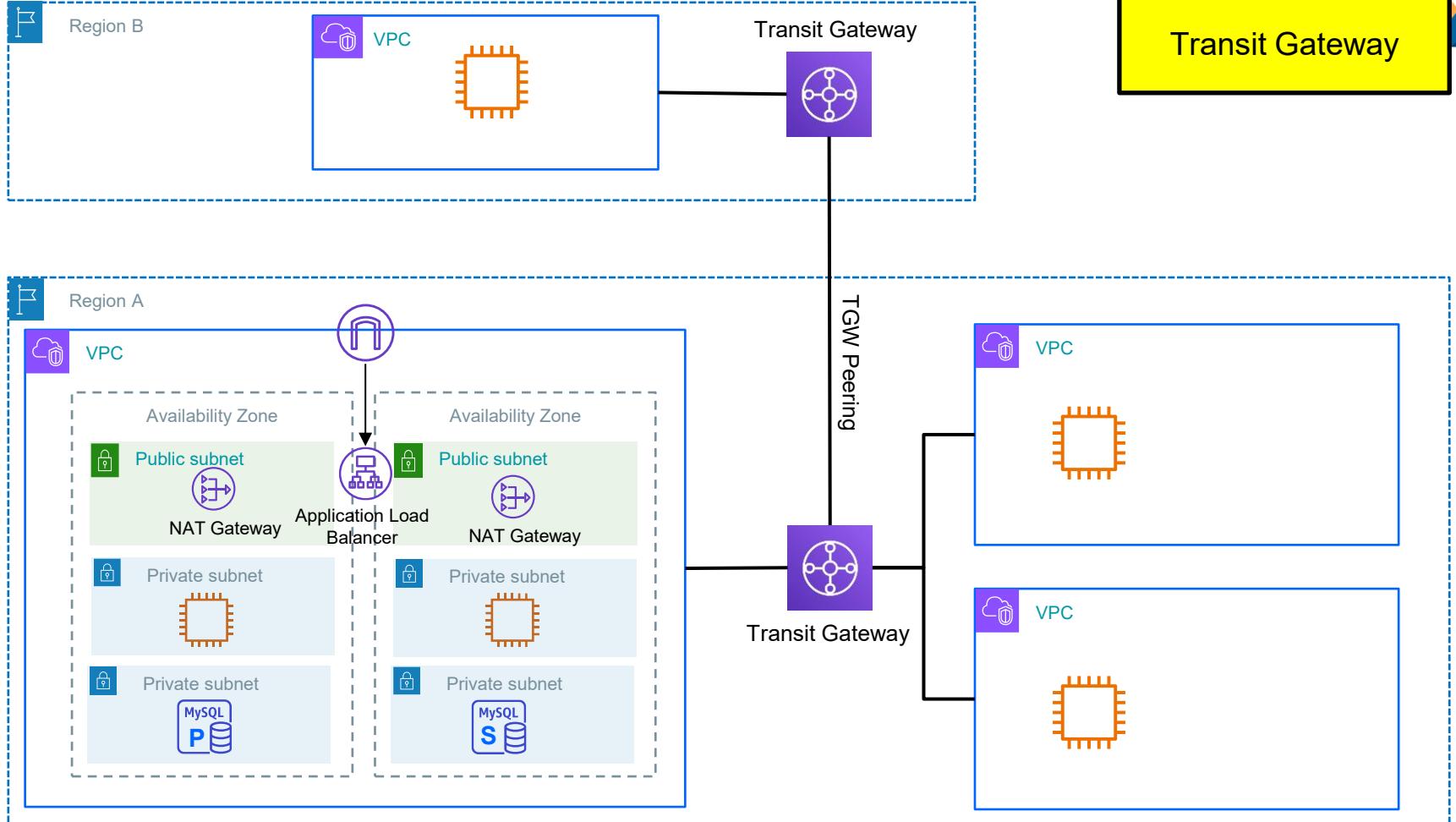
- **VPC endpoint** for accessing AWS services privately within the region



# VPC Peering

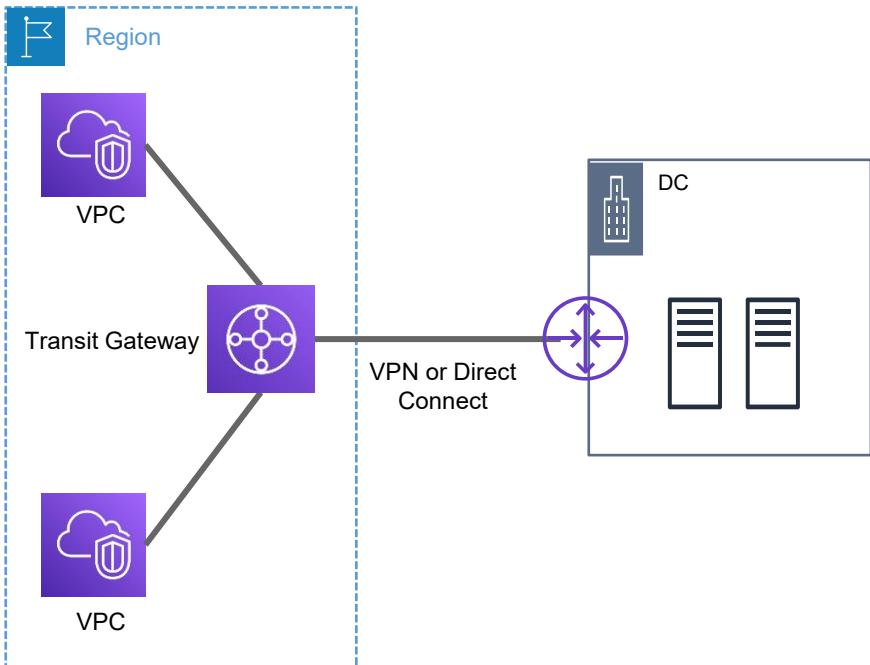
- Simplest way to connect two VPCs privately
- It's One-to-One connection
- No need to have internet gateways
- VPCs should not have overlapping CIDRs
- Does not support transitive routing
- Support intra region and inter region peering



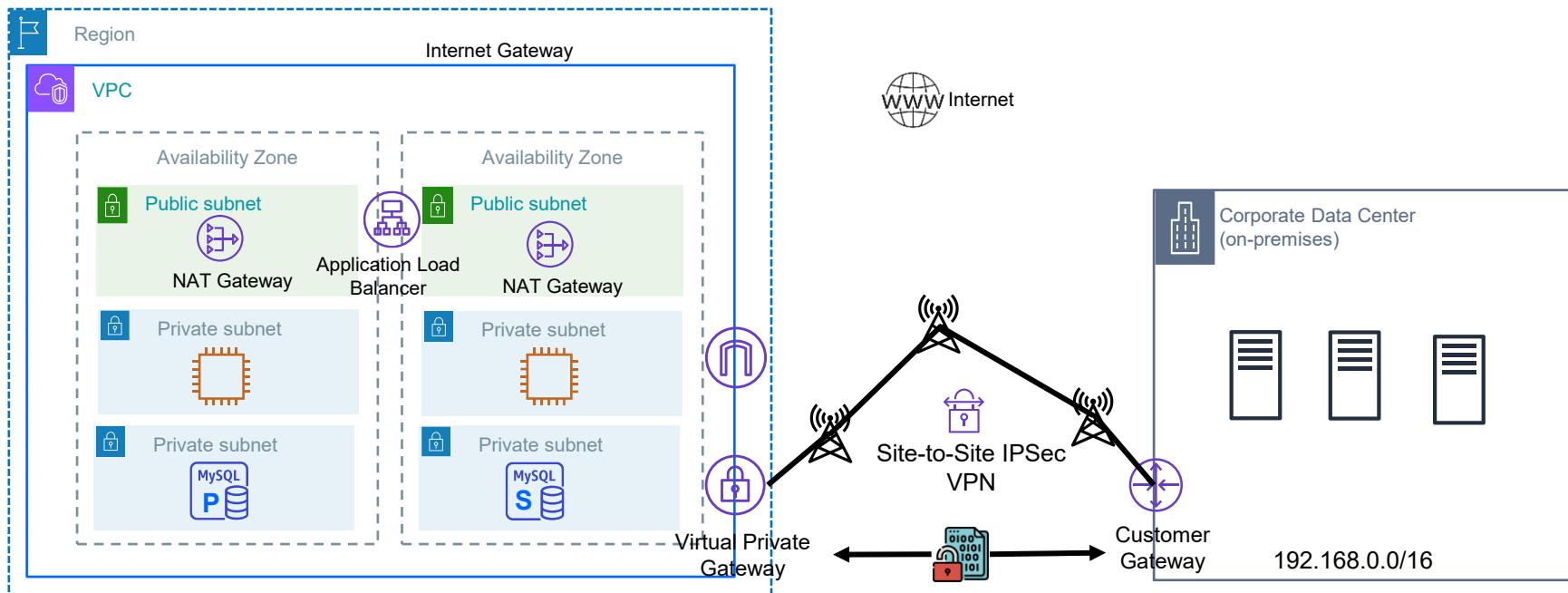


# Transit Gateway

- Allows customers to interconnect **thousands of VPCs and on-premises networks.**
- It's a regional router
- Hub and spoke architecture where we can connect
  - VPCs
  - Another Transit Gateway
  - VPN connection
  - Direct Connect Gateway
  - A Connect SD-WAN/third-party network appliance
- Ideal for centralized traffic inspection and Hybrid network setup etc.



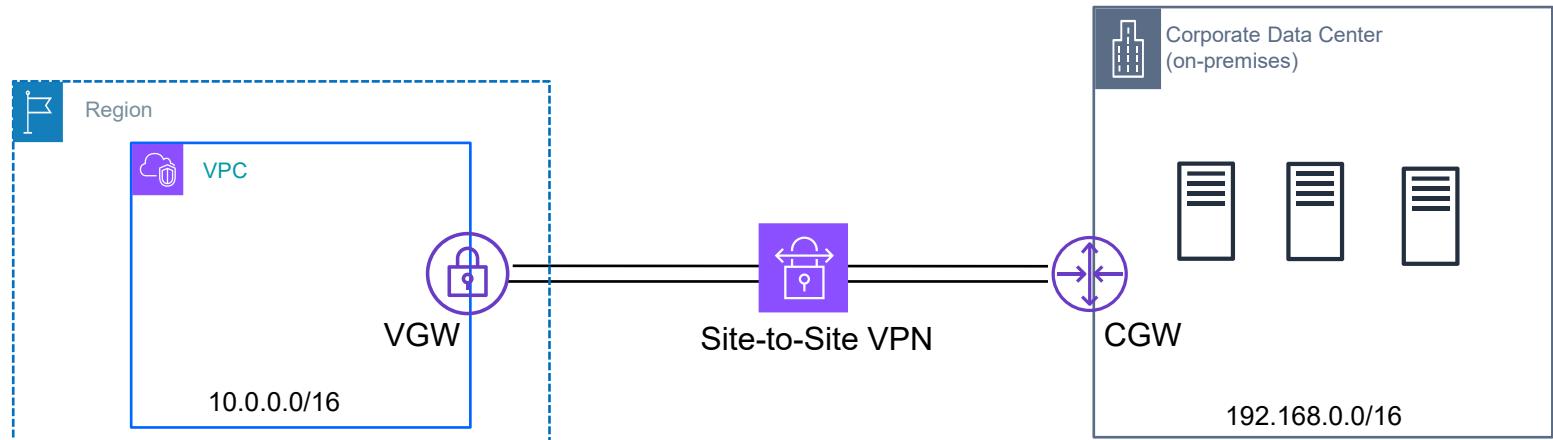
## Site-to-Site VPN





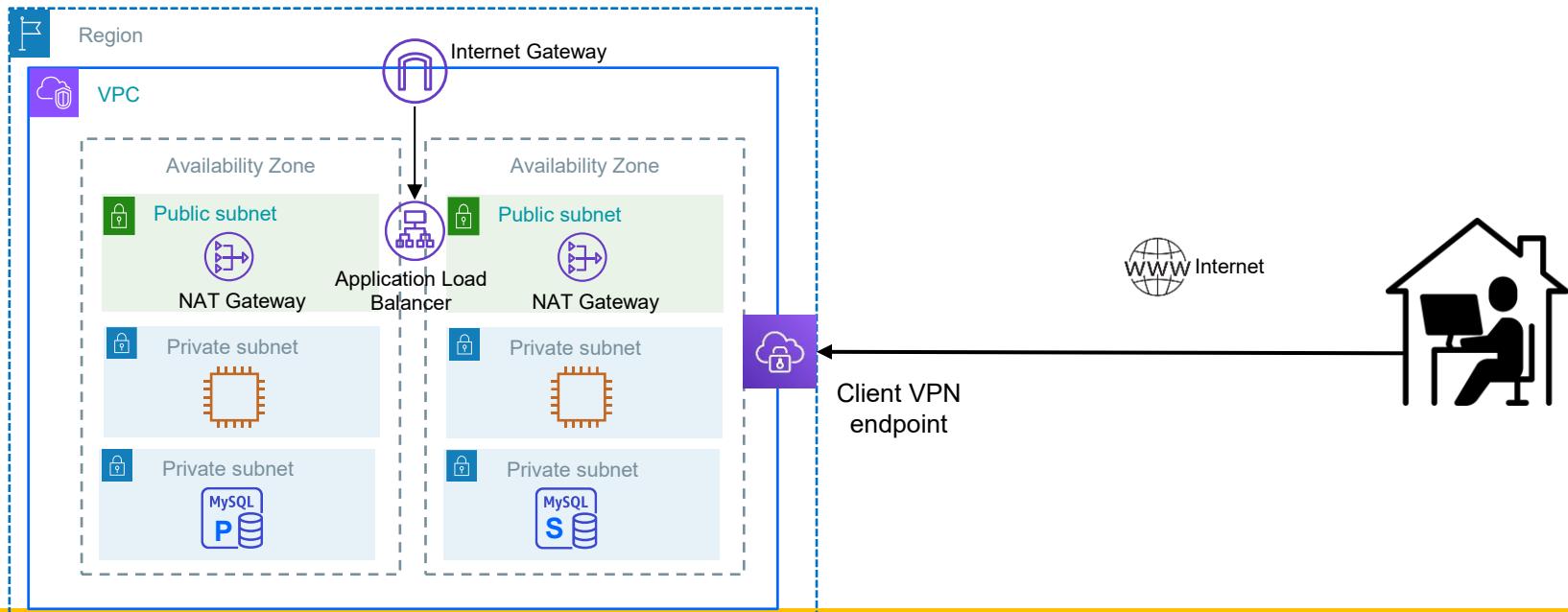
# AWS Site-to-Site VPN

- A Managed IPSec VPN connection between on-premises router and AWS VPC
- Customer Gateway (CGW) on Customer side and Virtual Private Gateway (VGW) on AWS side
- Traffic flows over the internet but **encrypted at Layer 3**
- 2 VPN Tunnels for High Availability

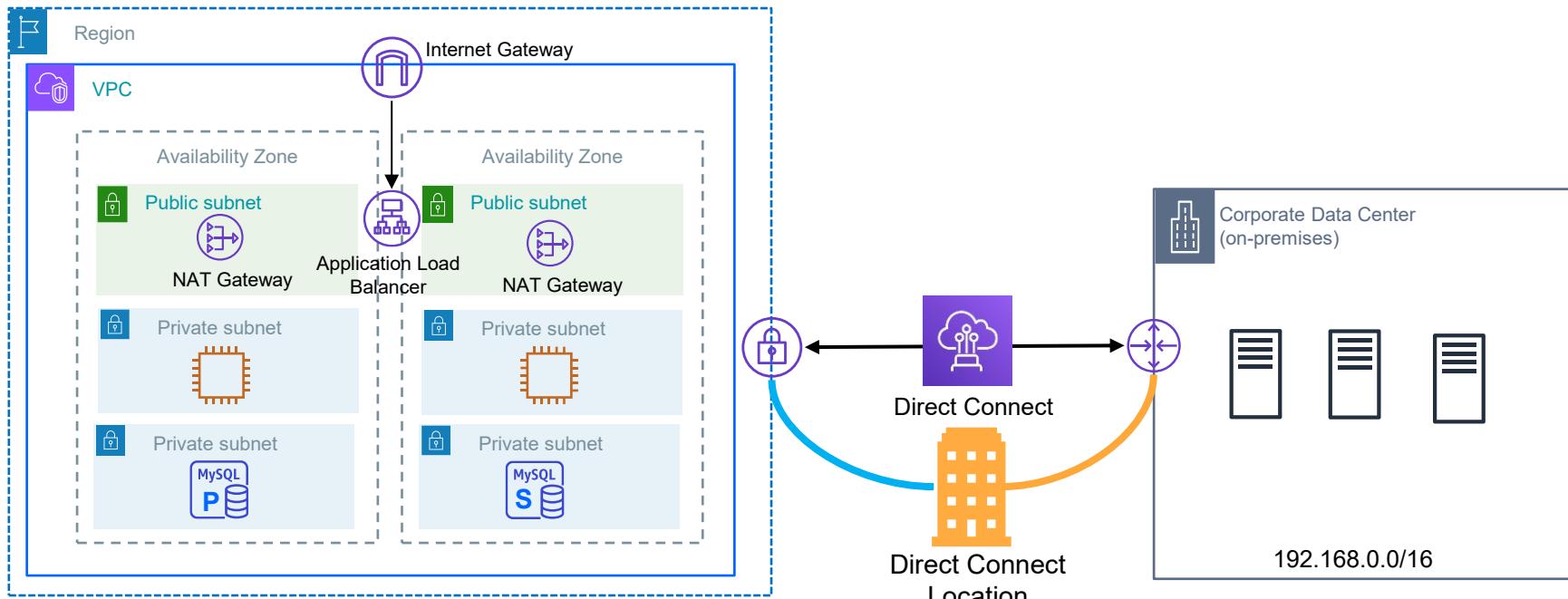


# AWS Client VPN

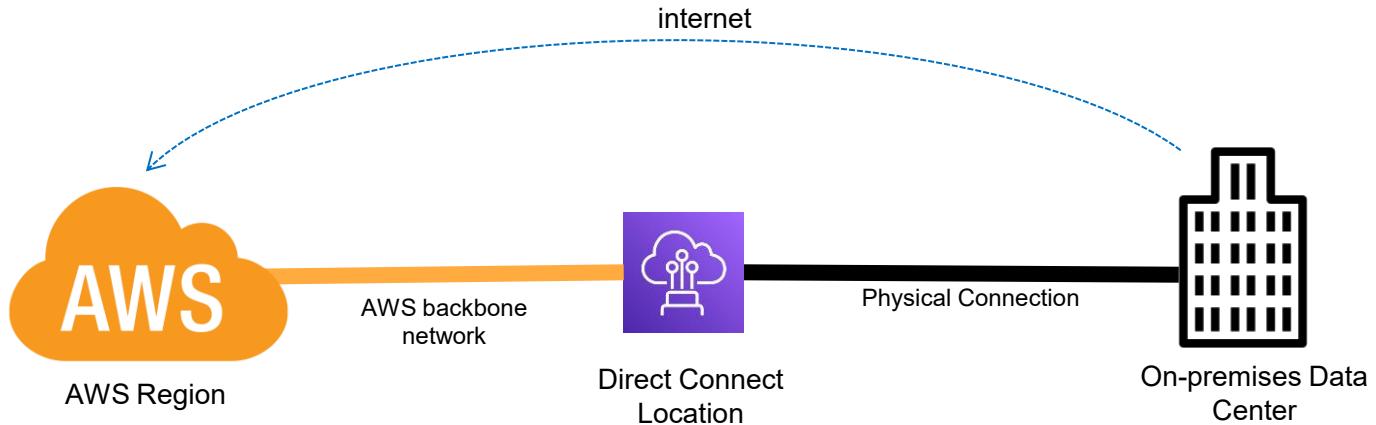
- Connect from your workstation privately to VPC using OpenVPN
- Encrypted traffic goes over the internet
- Access VPC resources over the Private IPs as if you are part of the same network



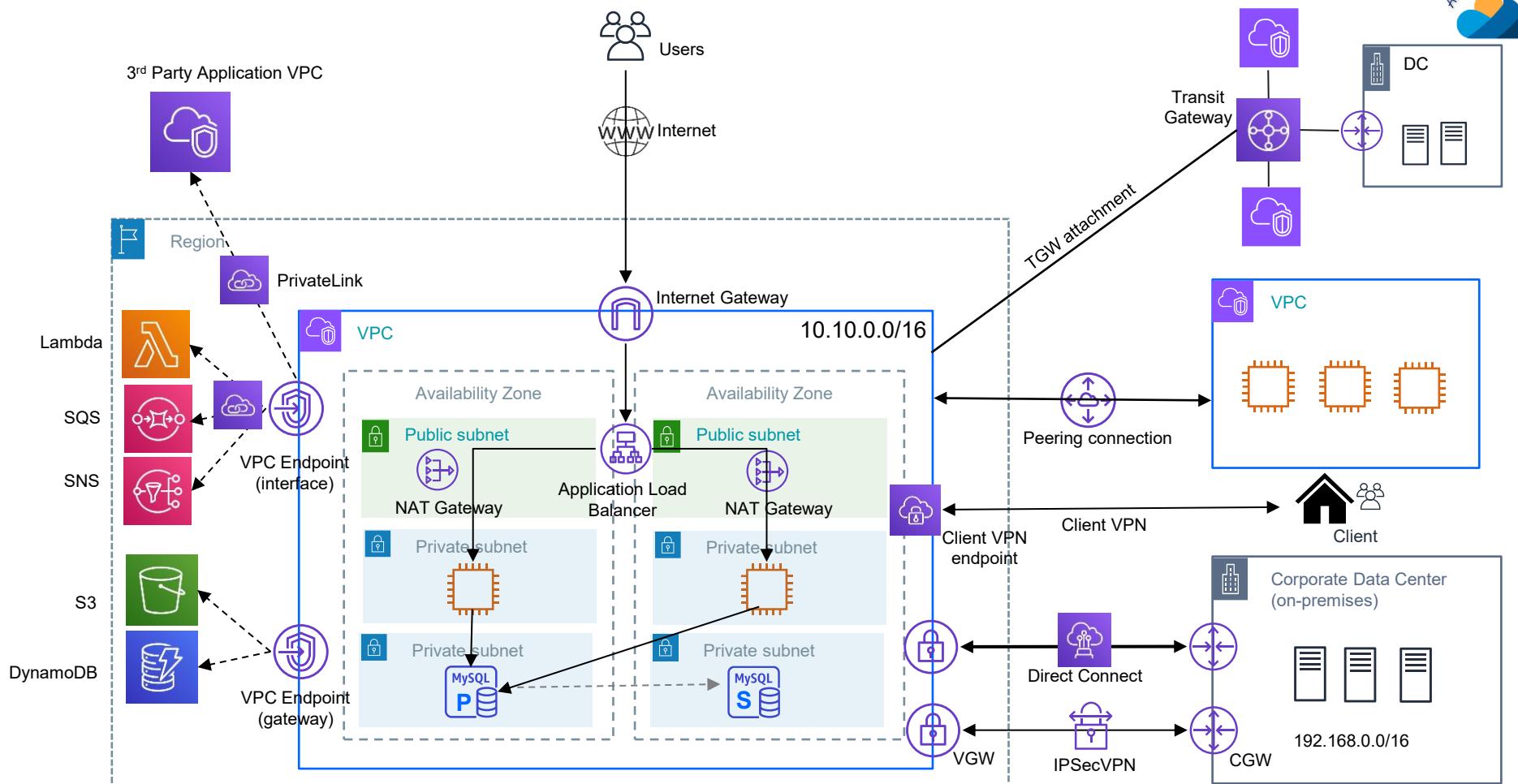
# AWS Direct Connect



# Direct Connect (DX)



- Provides dedicated and consistent network
- Provides network bandwidth from 50 Mbps up to 100 Gbps over a single connection
- Low data transfer cost
- **May take up to 1-3 months to establish end-to-end connectivity.**

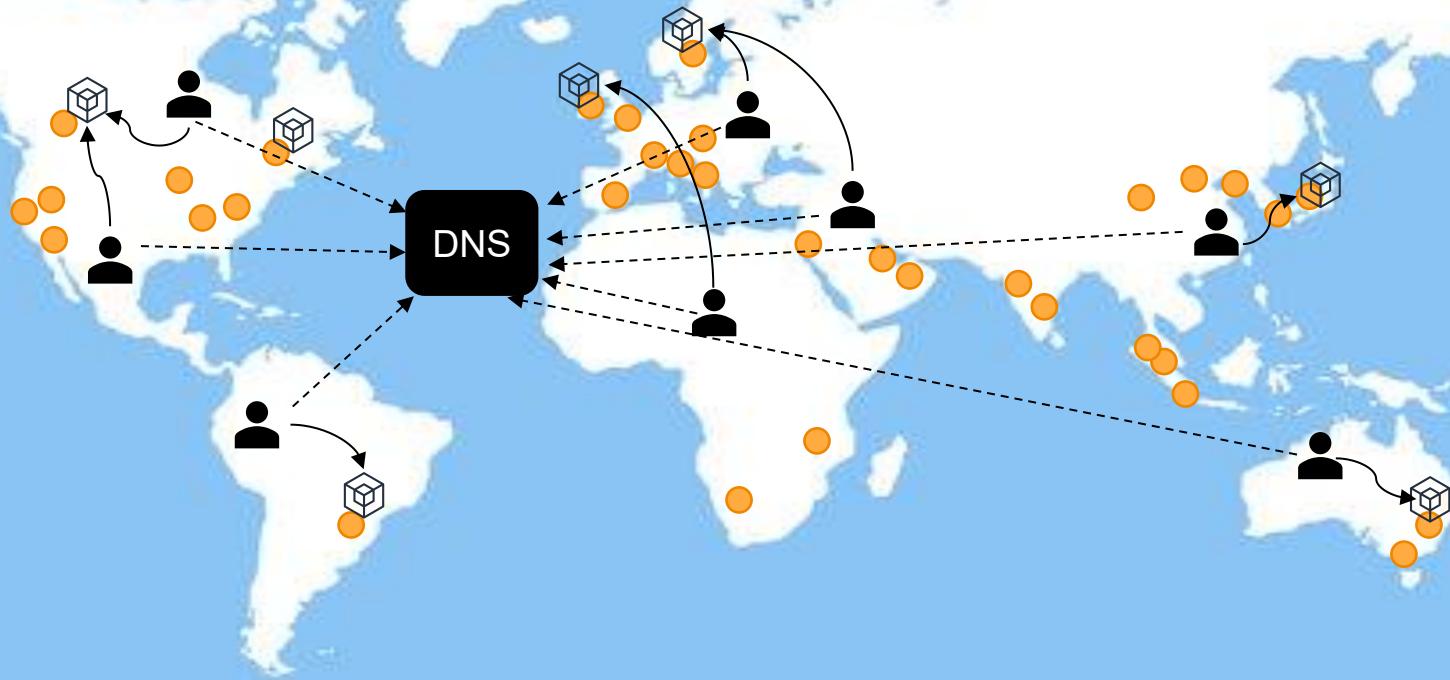


# VPC and Networking - Summary

- VPC is a logically isolated private network in AWS
- VPC has a private address range called CIDR (IPv4 and IPv6)
- VPC supports Public/Elastic IPs for public connectivity and Private IPs for private connectivity.
- Internet Gateway connects VPC to the internet
- VPC has subnets – Public subnet and Private Subnet
- NAT gateway provides outbound internet access to instances in the Private subnet
- VPC offers 2 types of firewalls – Security group and Network ACL
- AWS creates default VPC in each region
- VPC endpoint provides private connectivity to other AWS services from within the VPC
- VPC peering connects two VPCs, non-transitive, non-overlapping addresses
- AWS Transit Gateway connects thousands of VPCs, On-premises network
- AWS Site-to-Site VPN for encrypted traffic from on-premises to AWS
- AWS Client VPN provides for encrypted traffic from your workstation to AWS VPC
- AWS Direct connect for physical connection from on-premises to AWS

# DNS and Edge networking

# In this section..

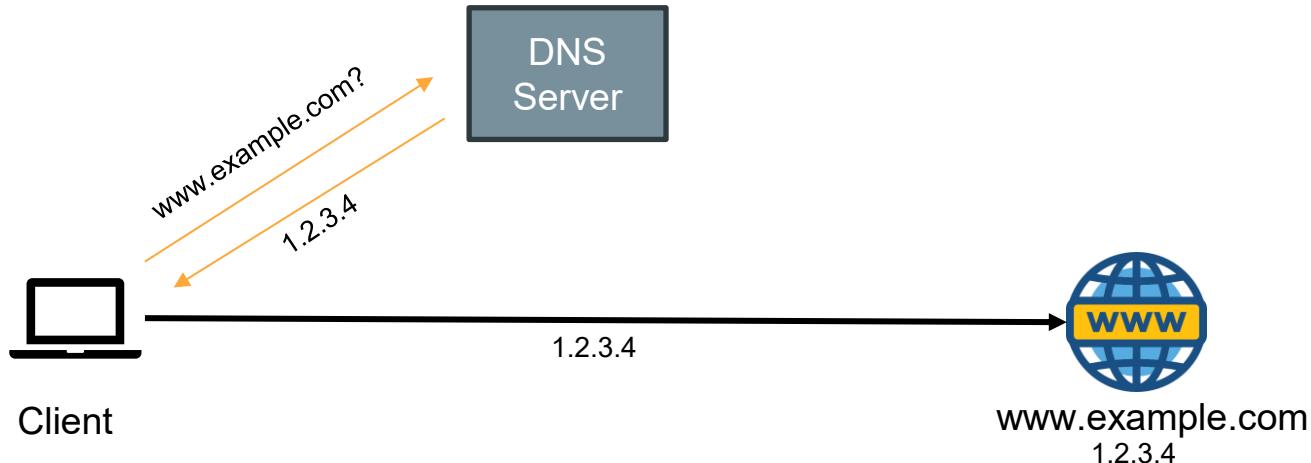


● AWS Regions

\*Diagram just for illustration, not actual

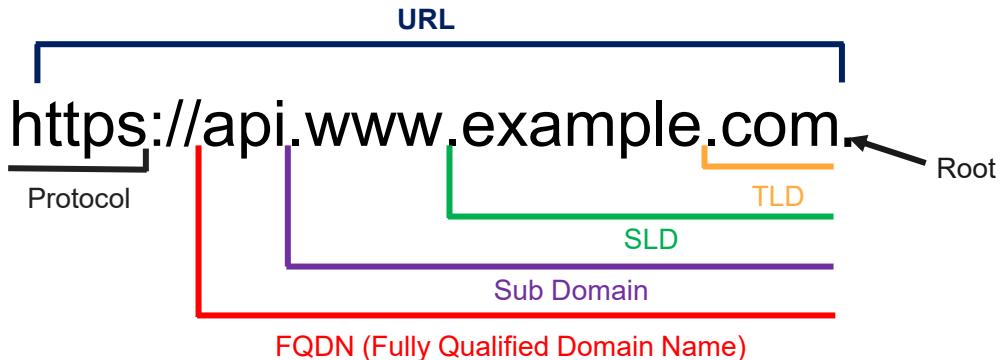
# What is DNS?

- Domain Name System which translates the human friendly hostnames into the machine IP addresses such as www.google.com => 172.217.18.36
- DNS is the backbone of the Internet

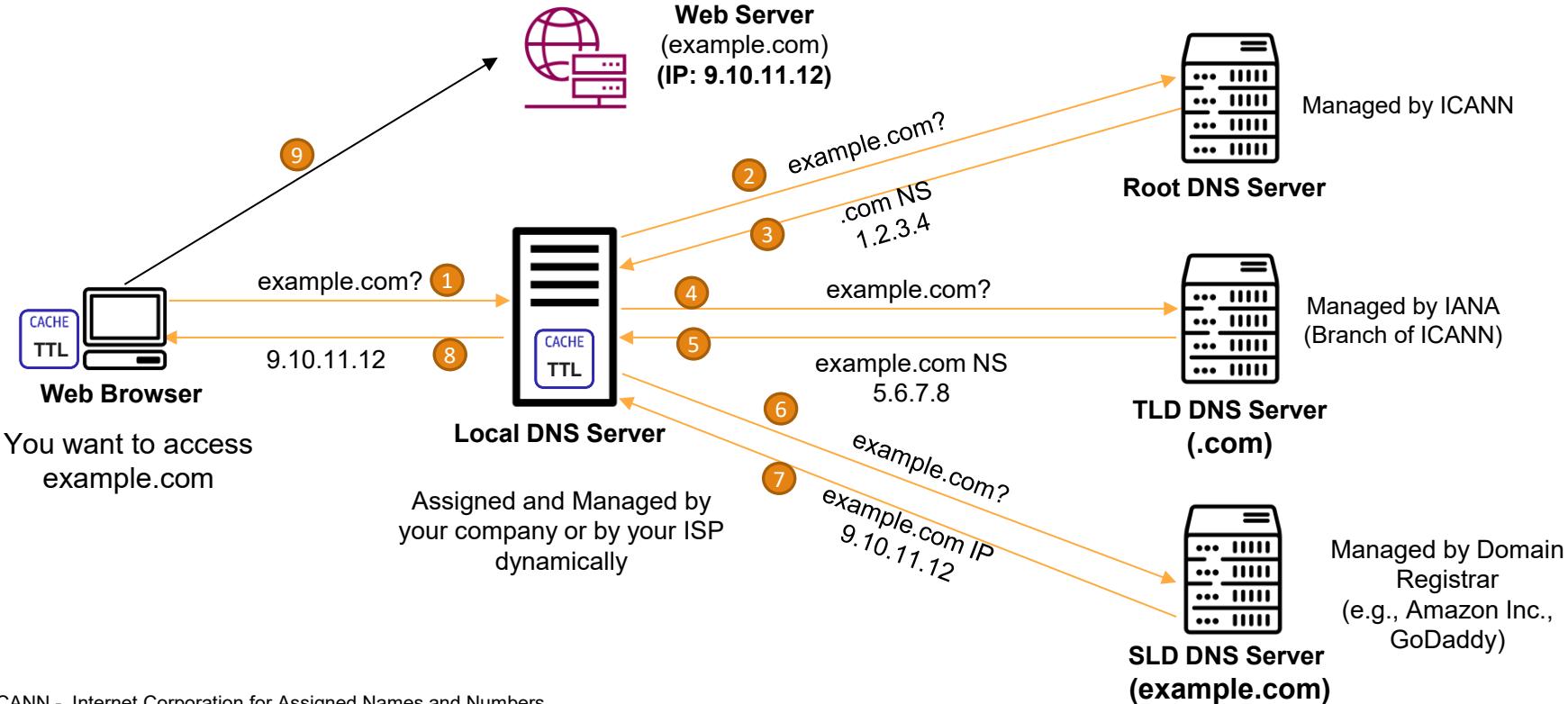


# DNS Terminologies

- Domain Registrar: Amazon Route 53, GoDaddy etc.
- DNS Records: A, AAAA, CNAME, NS
- Zone File: contains DNS records
- Name Server: Resolves DNS queries
- Top Level Domain (TLD): .com, .us, .in, .gov, .org, ...
- Second Level Domain (SLD): amazon.com, google.com etc.



# DNS resolution under the hood



ICANN - Internet Corporation for Assigned Names and Numbers

IANA - Internet Assigned Numbers Authority



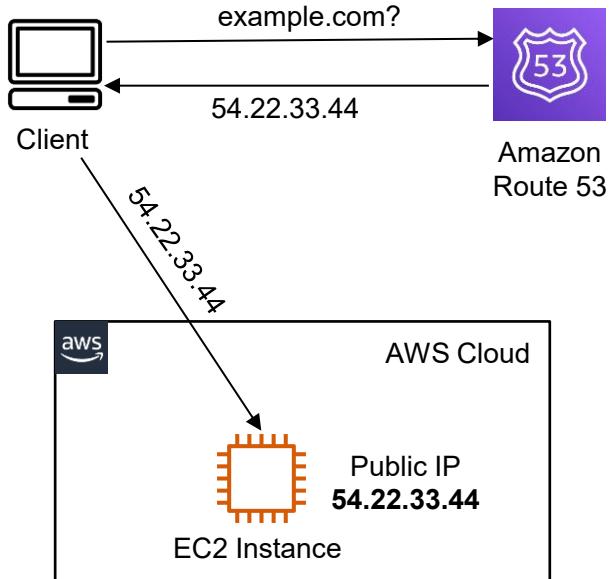
# AWS Route 53

# Amazon Route 53

- A highly available, scalable, fully managed global DNS service by AWS
- Route 53 is also a Domain Registrar
- Ability to check the health of your resources
- The only AWS service which provides 100% availability SLA

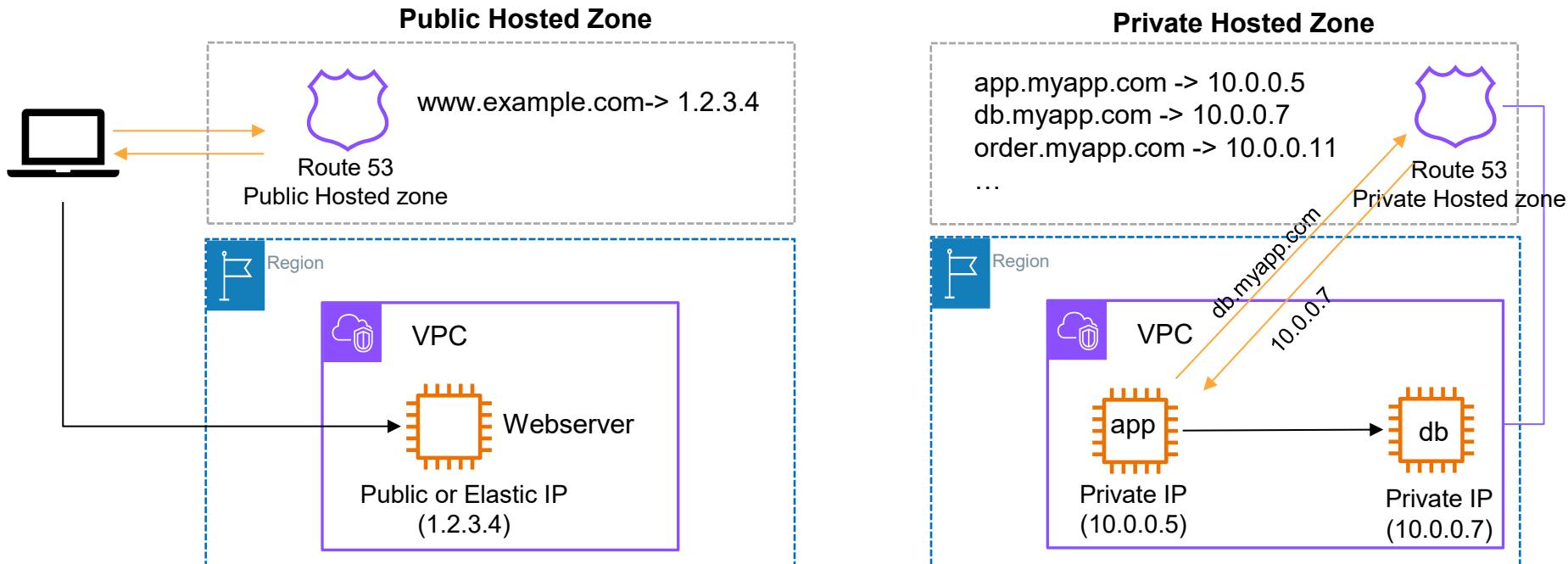


53 is a traditional  
DNS port



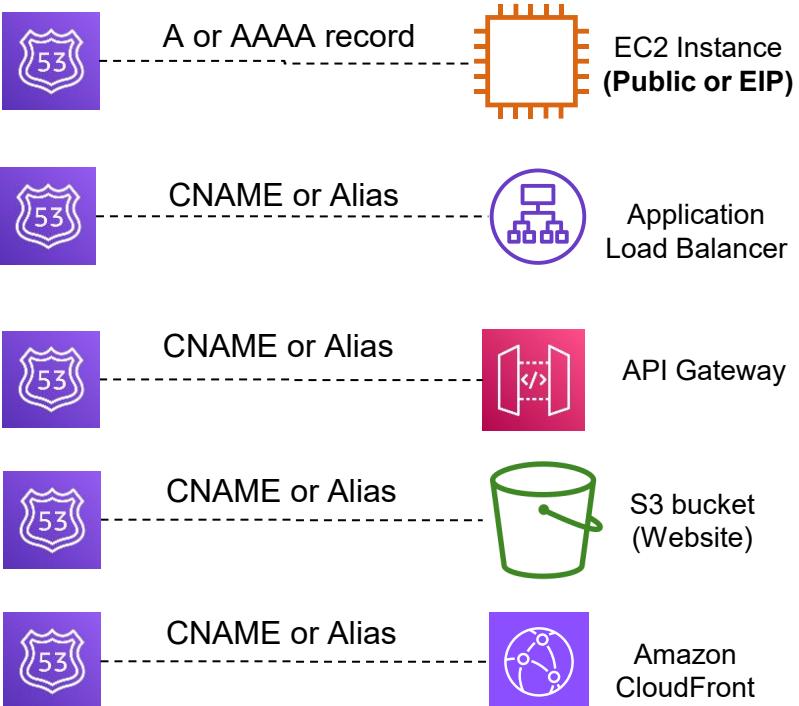
# Route 53 – Hosted Zones

- Public Hosted Zone – For public domain names. DNS resolution over the internet.
- Private Hosted Zone – For private domain names. DNS resolution within the VPC
- Zone file contains records to point domain name to target IP address or to another domain name.

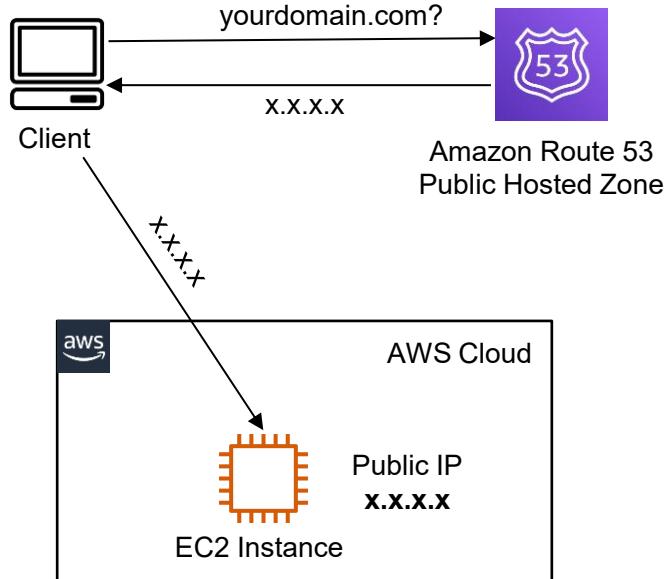


# Route53 DNS record types

- For pointing domain name to IP addresses, use A (IPv4) or AAAA (IPv6) records
- For pointing SLD e.g. example.com to AWS provided DNS, use Alias record
- For pointing subdomain e.g. api.example.com to AWS provided DNS, use CNAME record



# Exercise – Host a website on EC2 with public domain name



## Pre-requisites:

- Must own Public domain name and configure Route 53 as a DNS provider
- If not done already, refer pre-requisites section of this course.



1 Launch an EC2 instance in default VPC and install a web server (httpd). Create a sample webpage. Allow access to HTTP port 80 for everyone.



2 Verify that you are able to access webserver over EC2 Public IP



3 Assuming you already have Route 53 Public hosted zone, create a new A records for your bare domain name and www domain name pointing to EC2 Public IP.



4 Wait for 1-2 minutes and try accessing webserver using your domain name. It should work.



5 Stop EC2 instance and delete the A records. We will continue with this setup for Amazon CloudFront exercise.

*Note: There is a charge of \$0.5 USD per public hosted zone per month*

# Sample website

You can also download sample website on EC2:

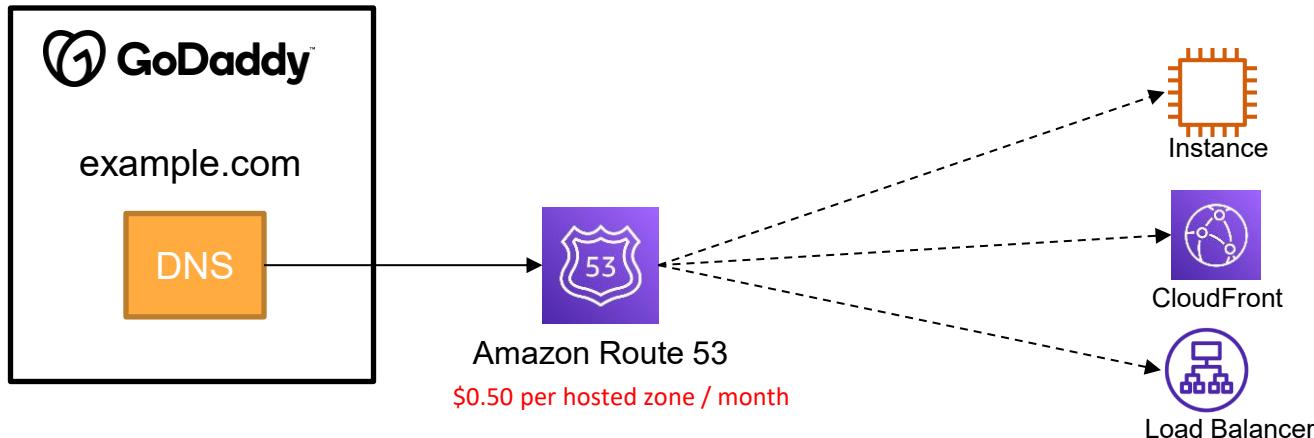
```
sudo yum install httpd -y  
cd /var/www/html  
wget https://s3.ap-south-1.amazonaws.com/www.awswithchetan.com/CloudPractitioner/certprotech.zip  
unzip certprotech.zip  
sudo systemctl start httpd.service  
sudo systemctl enable httpd.service
```

# You can also use following EC2 Userdata while launching an instance

```
#!/bin/bash
yum install httpd -y
cd /var/www/html && wget https://s3.ap-south-
1.amazonaws.com/www.awswithchetan.com/CloudPractitioner/certprotech.zip && unzip certprotech.zip
systemctl start httpd.service
systemctl enable httpd.service
```

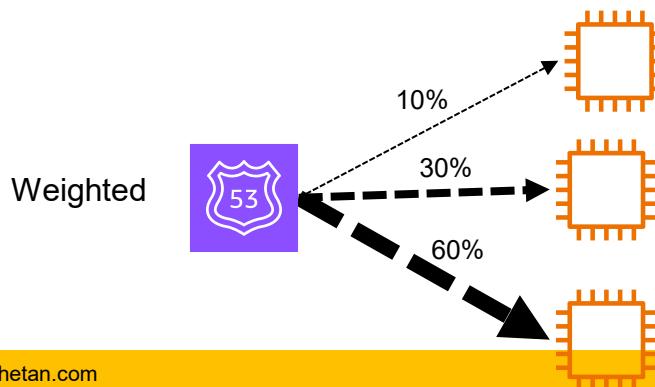
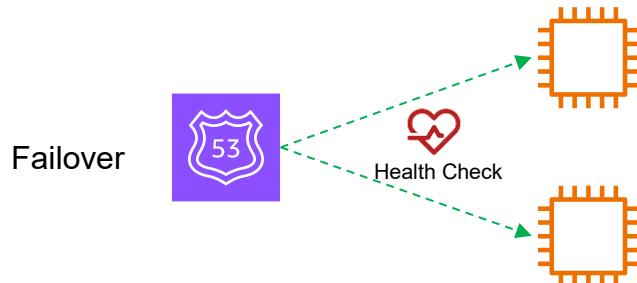
# Setup a Route 53 DNS for your domain

1. Go to AWS Route 53 console
2. Create a Public Hosted Zone (PHZ) with the same domain name that you purchased
3. From PHZ, note down the names of the 4 nameservers (NS records)
4. Go to your domain portal (e.g. godaddy dashboard) and go to DNS management
5. Replace the preconfigured nameservers with Route53 nameservers and save the changes. Note: If you are using GoDaddy then you need to remove the trailing dot from the NS names as Godaddy doesn't accept it.



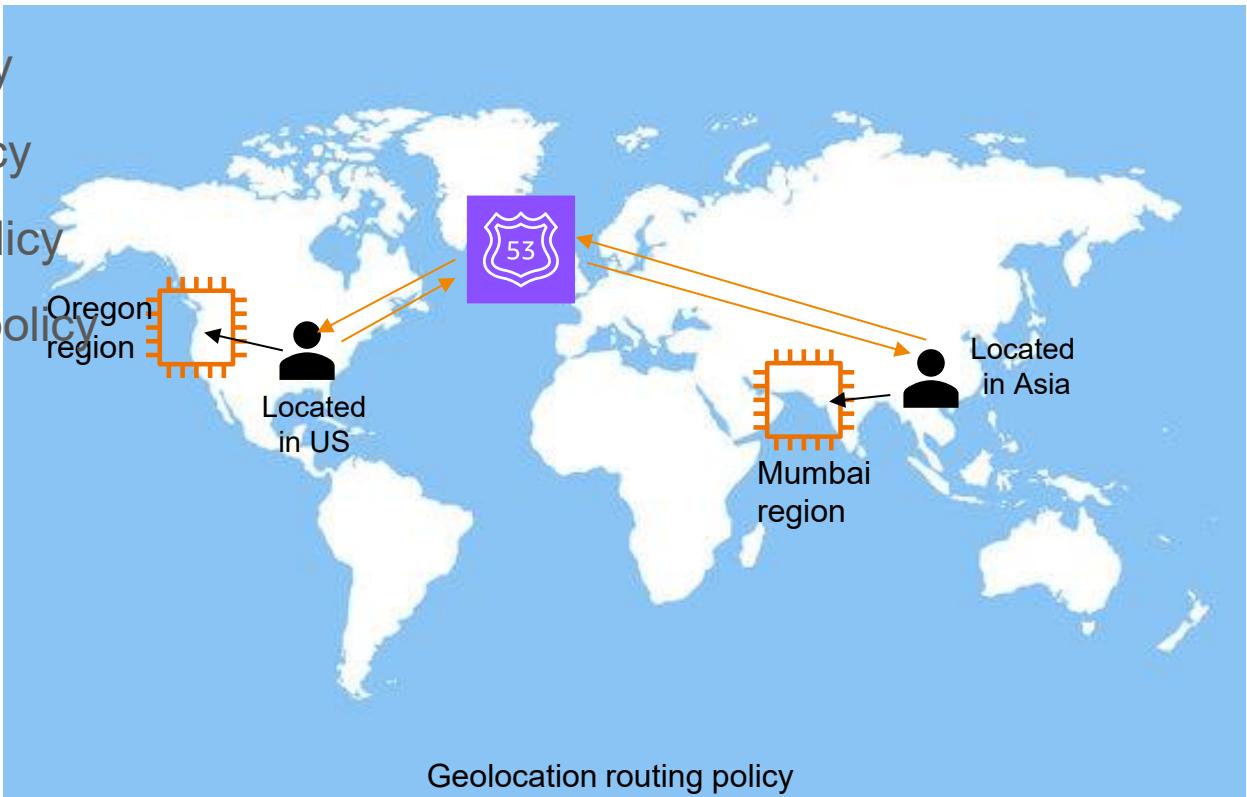
# Route 53 routing policies

- Simple Routing Policy
- Failover Routing policy
- Weighted Routing policy



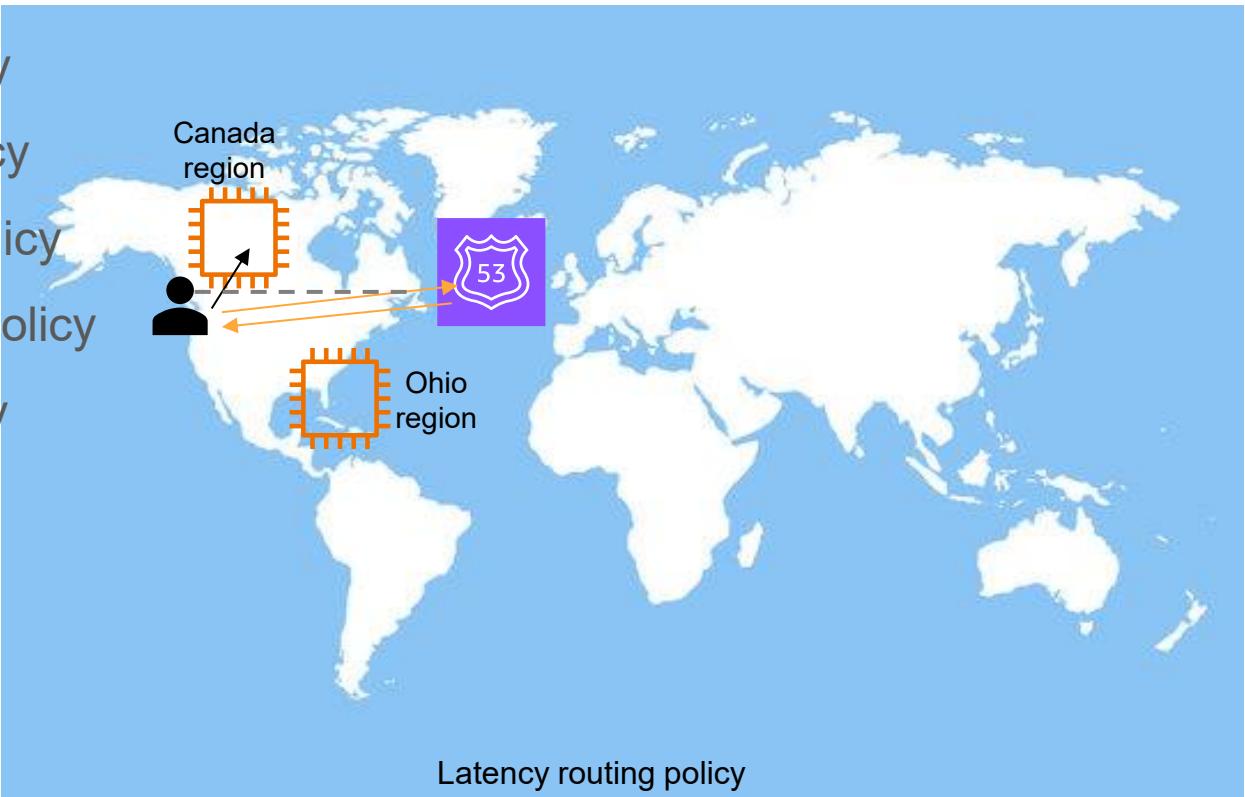
# Route 53 routing policies

- Simple Routing Policy
- Failover Routing policy
- Weighted Routing policy
- Geolocation routing policy



# Route 53 routing policies

- Simple Routing Policy
- Failover Routing policy
- Weighted Routing policy
- Geolocation routing policy
- Latency routing policy
- More..



# AWS edge networking

Access applications with lowest latency across the globe

# AWS Edge locations



\*Diagram just for illustration, not actual

# Without AWS edge network



\*Diagram just for illustration, not actual

# With AWS edge network

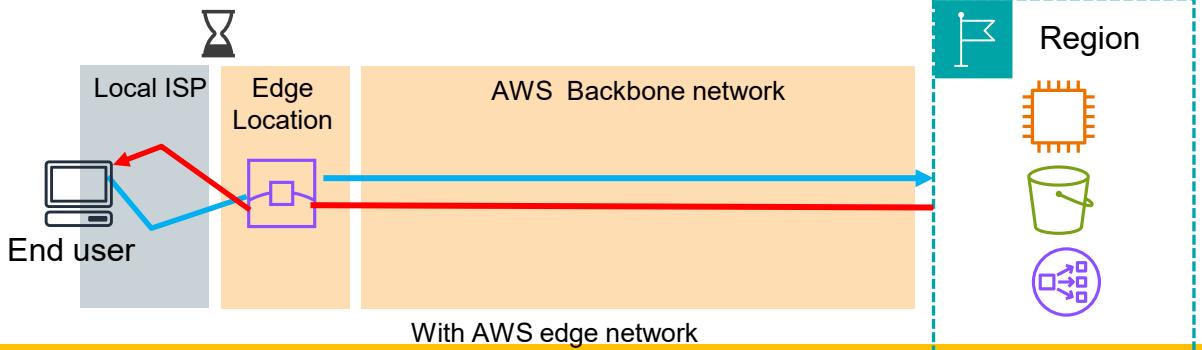
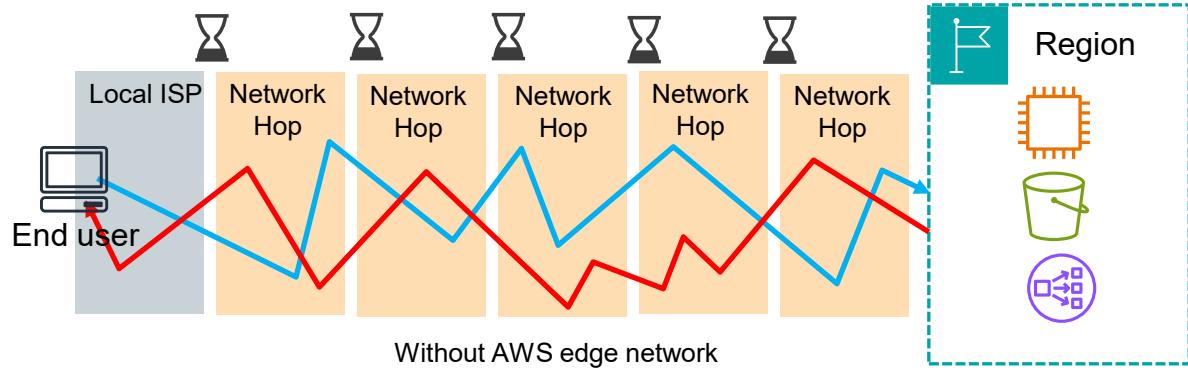


\*Diagram just for illustration, not actual

# AWS edge network and services

AWS Edge locations are used by different AWS services to lower the network latency for end users

- Following AWS services uses AWS edge locations to optimize the network path



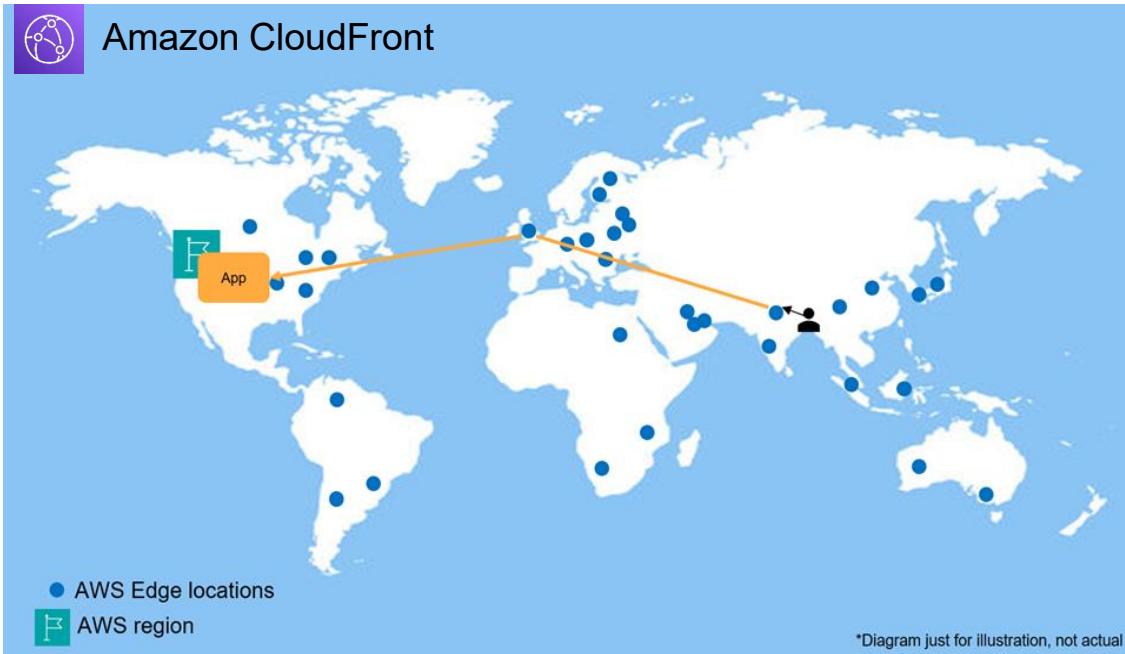
**Accelerator**



# Amazon CloudFront

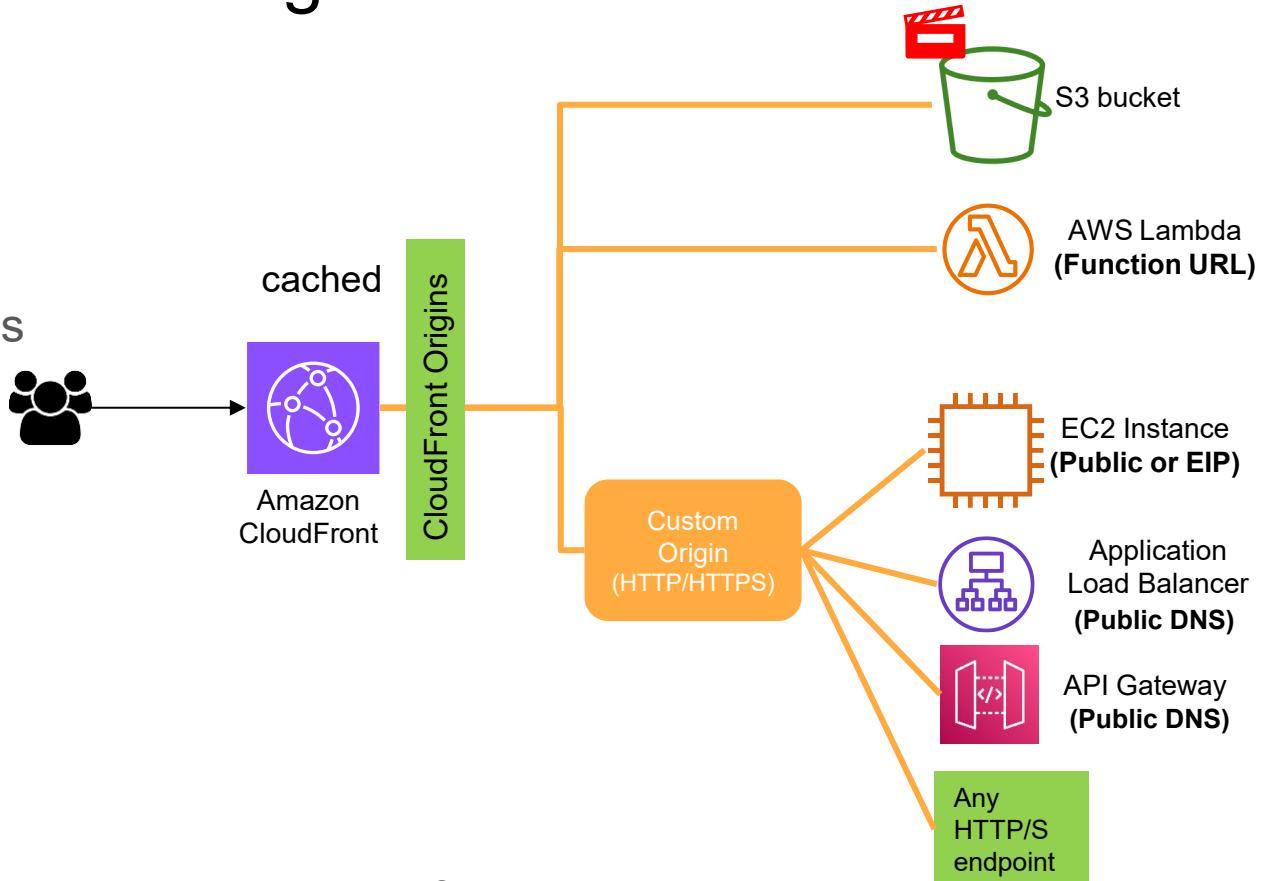
# Amazon CloudFront

- Amazon CloudFront is a **Content Delivery Network (CDN)** service.
- CloudFront delivers content through a worldwide network of AWS data centers called edge locations.
- CloudFront caches the static contents e.g. images, videos etc. at the PoP locations
- 119+ Point of Presence locations and 600+ embedded PoPs, 13 Regional Edge Caches (RECs) globally
- Built-in DDoS protection with Amazon Shield



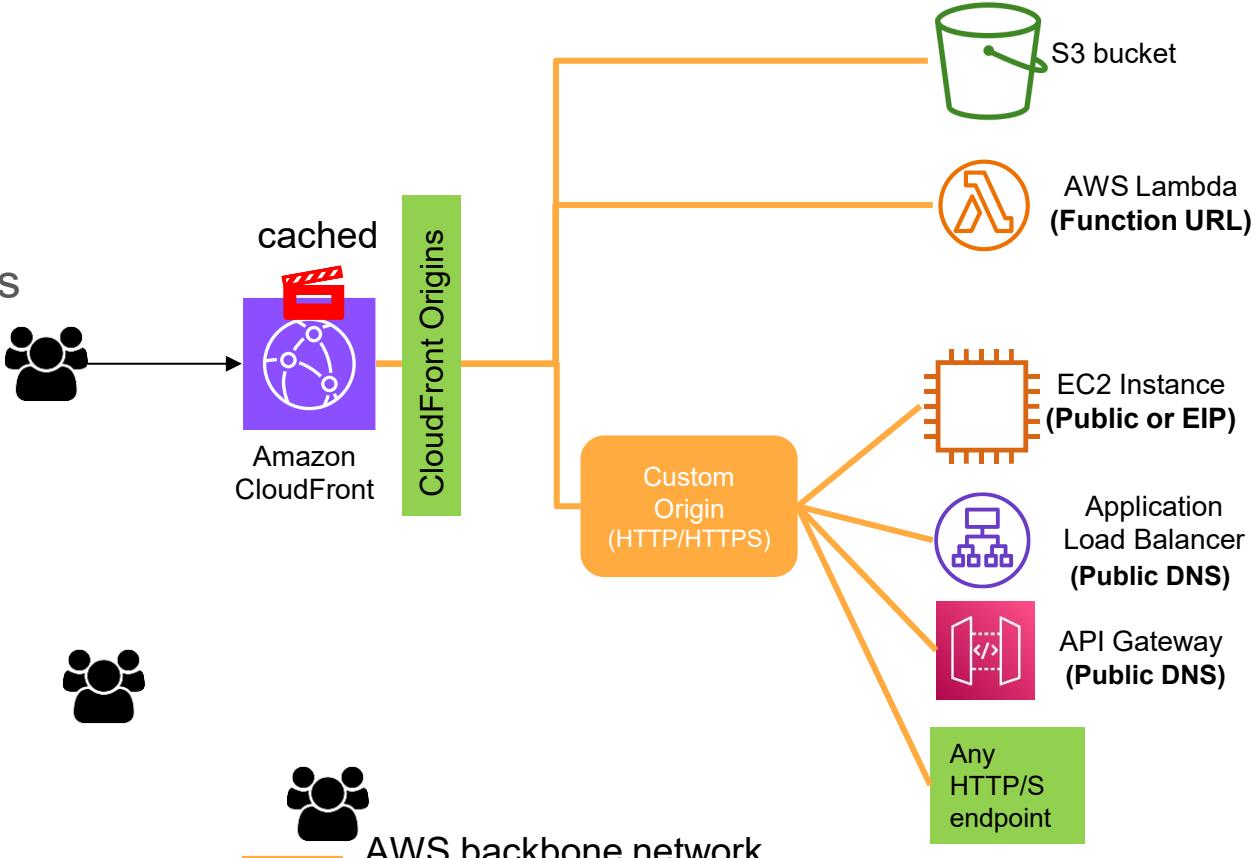
# Amazon CloudFront Origins

- S3 Bucket
- Lambda Function
- Custom HTTP Origins
  - EC2
  - ALB
  - API gateway
  - Any HTTP endpoint



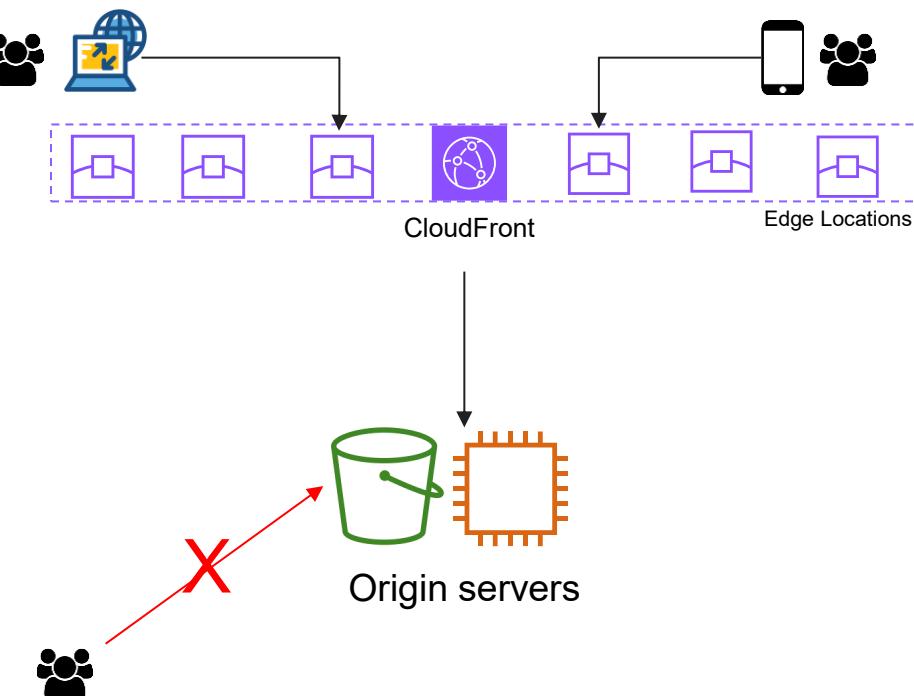
# Amazon CloudFront Origins

- S3 Bucket
- Lambda Function
- Custom HTTP Origins
  - EC2
  - ALB
  - API gateway
  - Any HTTP endpoint



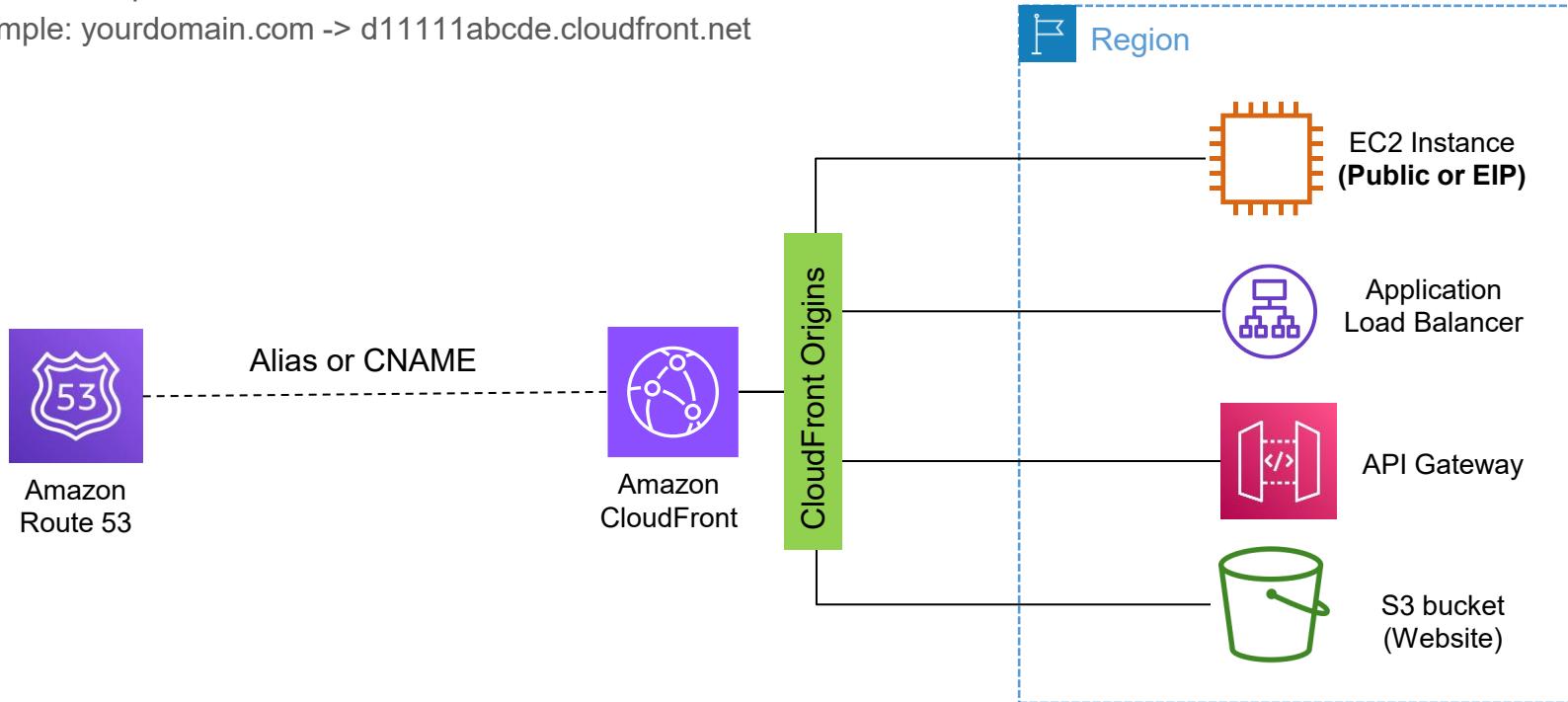
# Amazon CloudFront benefits

- Delivers the static and dynamic contents at lowest latency
- CloudFront caches the static contents which helps deliver the content faster and it reduces the load on the origin server
- Reduced Data transfer internet rate and 1TB free data transfer out per month
- Improves security by encrypting traffic and uses Amazon Shield for DDoS protection
- Can restrict direct access to Origins, for example: Supports Origin Access Control (OAC) and origin access identity (OAI) for S3



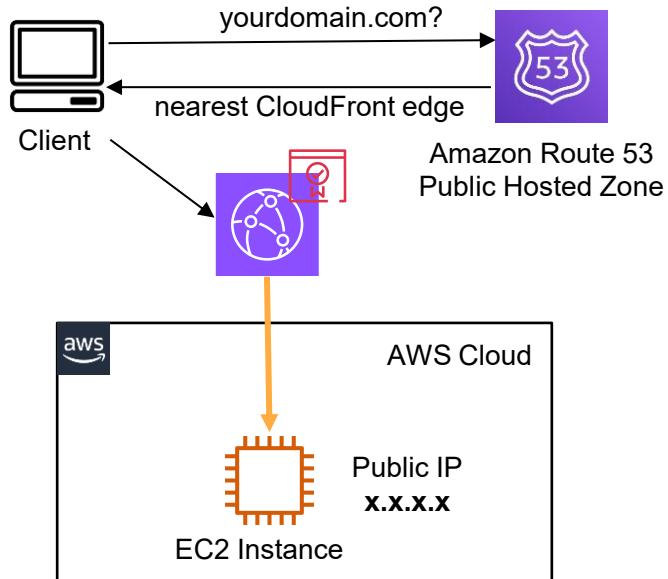
# Using Amazon CloudFront with Route53

- You can also point Domain name to Amazon CloudFront distribution DNS
- Example: yourdomain.com -> d11111abcde.cloudfront.net



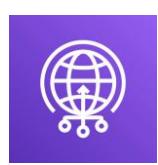
# Exercise – Setup https website with CloudFront

Continuing with Route53 exercise setup..



- 1 Using ACM, create TLS certificate for your domain name. Verify domain to issue the certificate.
- 2 Create a CloudFront distribution and configure EC2 instance Public DNS as origin.
- 3 Create an Alias record in Route53 public hosted zone to point it to CloudFront distribution
- 4 Wait for 2-3 minutes and try accessing website using your domain name. It should work.
- 5 Terminate EC2 instance. Disable and delete CloudFront distribution. Delete Route53 records and Route53 Public hosted zone.

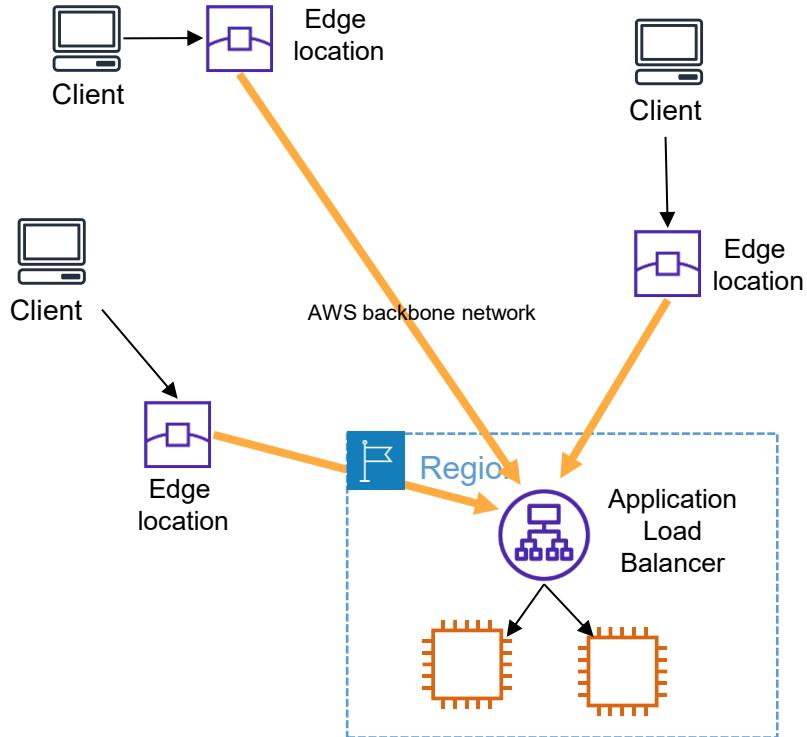
Note: There is a charge of \$0.5 USD per public hosted zone per month



# AWS Global Accelerator

# AWS Global Accelerator

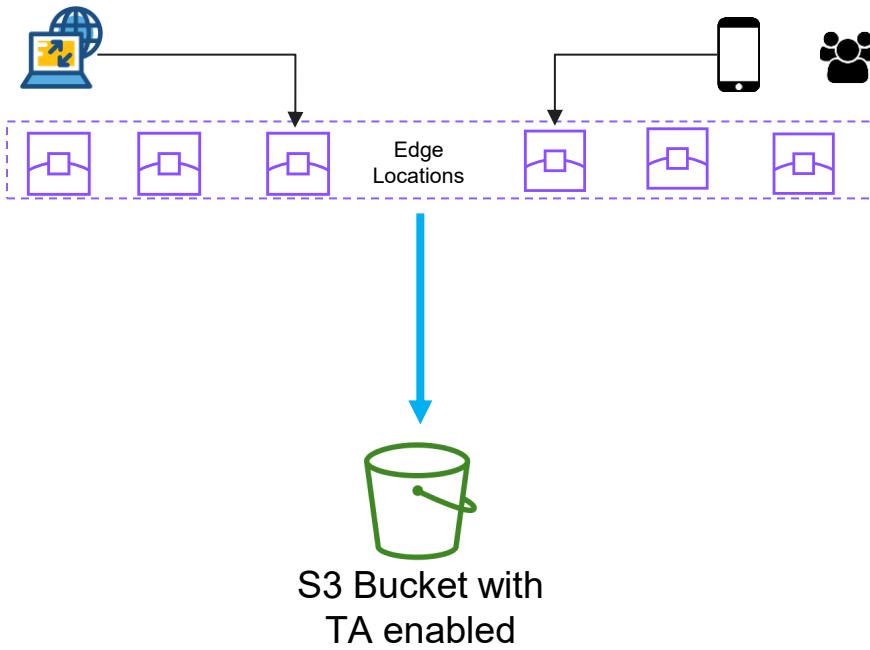
- Similar to Amazon CloudFront, AWS Global Accelerator also uses AWS edge locations
- Used with Application Load Balancer, Network Load Balancer or EC2 instance
- **Allocates 2 Anycast IPs**
- Anycast IPs send traffic directly to Edge Locations
- It's possible to whitelist these IP addresses on client side



# S3 Transfer Accelerator

# S3 Transfer Accelerator (S3TA)

- Speeds up content transfer to and from S3 bucket by 50-500% for long-distance transfer of larger objects.
- Routes traffic through Amazon CloudFront's globally distributed Edge Locations and over AWS backbone networks
- Additionally, uses network protocol optimizations
- We can enable/disable S3 Acceleration for S3 bucket
- Provides new s3 bucket endpoint when accelerator is enabled e.g. *bucket-name.s3-accelerate.amazonaws.com*
- Speed comparison tool: <https://s3-accelerate-speedtest.s3-accelerate.amazonaws.com/en/accelerate-speed-comparison.html>
- So Why not use Amazon CloudFront with S3 origin?



# CloudFront vs S3 Transfer Accelerator (S3TA)

Feature	CloudFront with S3 Origin	S3 Transfer Acceleration
Primary Function	Content delivery and caching	Fast uploads to S3
Optimized For	Reducing latency for end-users accessing content	Improving upload speeds
Typical Use Cases	Websites, APIs, streaming	Large, geographically distant uploads
Cache Benefits	Caches data in edge locations	No caching, direct upload
Costs	CloudFront data transfer + S3 storage fees	Transfer Acceleration fees + S3 transfer fees

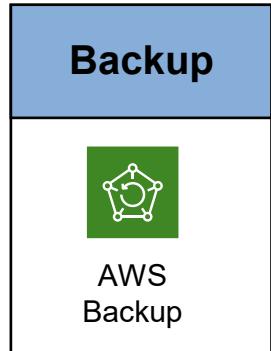
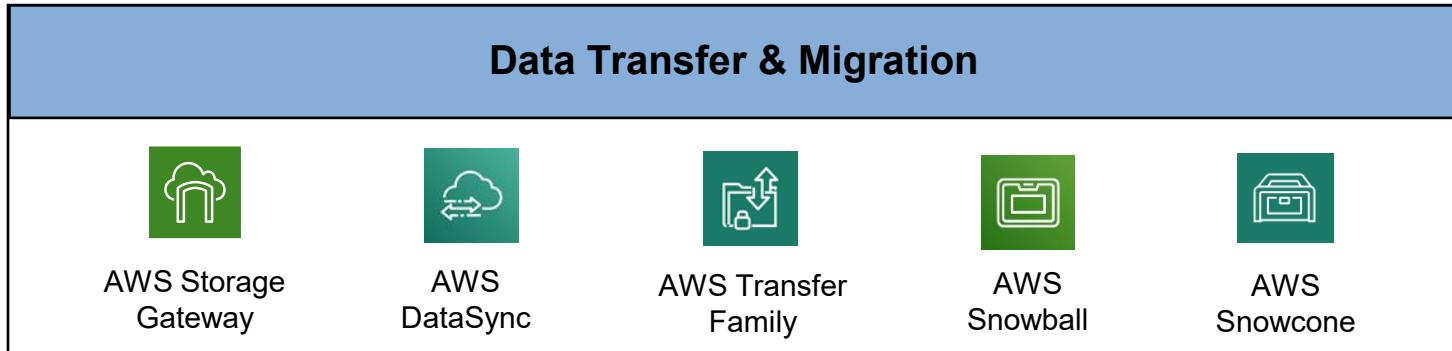
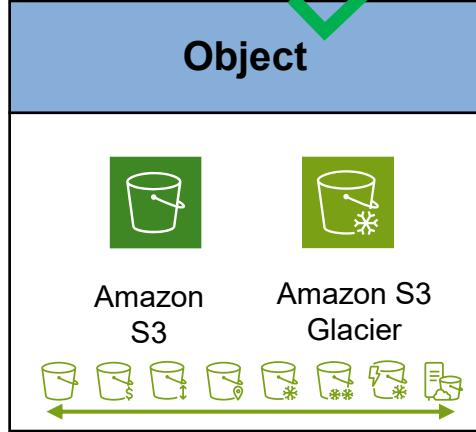
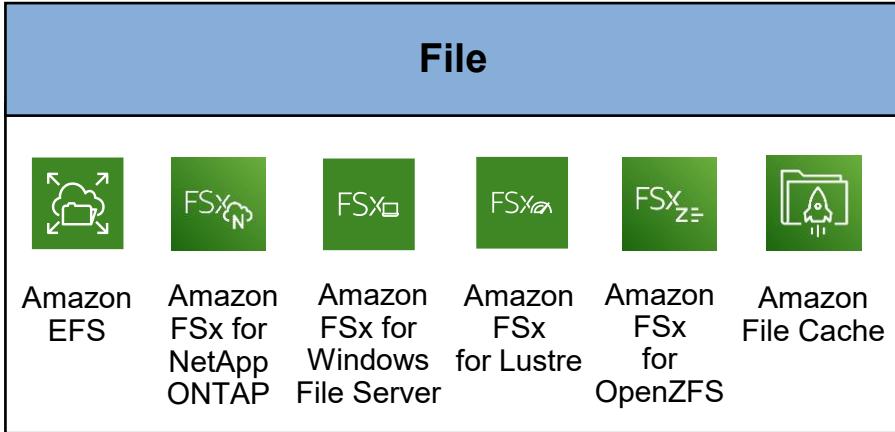
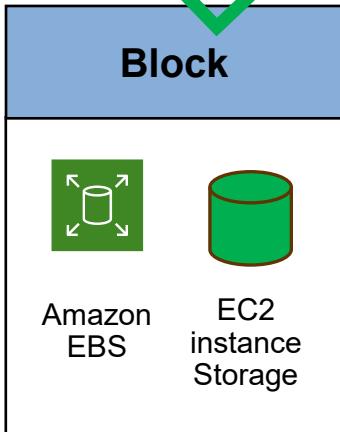
# DNS and Edge networking - summary

- Amazon Route 53 is Global DNS service and Domain Registrar service of AWS
- Create Public hosted zone for Public DNS and Private hosted zone for Private DNS within the VPC
- Route53 supports different DNS records e.g. A (IPv4), AAAA (IPv6), CNAME, Alias and more
- There are different routing policies e.g. Simple, Failover, Weighted, Geolocation, Latency based.
- AWS has hundred plus edge locations across 90+ cities
- These edge locations are used by Amazon CloudFront, AWS Global Accelerator and AWS S3 Transfer accelerator to optimize network path from end user to the AWS region where applications are hosted.
- Amazon CloudFront is a Content Delivery Network (CDN) service of AWS
- Amazon CloudFront uses AWS Edge locations to optimize network path and cache the static contents e.g. static html files, images, videos etc.
- AWS Global Accelerator provides 2 static IPs which clients can use to connect to Global accelerator endpoint
- AWS Global Accelerator supports Application Load Balancer, Network Load Balancer and EC2
- AWS S3 Transfer Acceleration (S3TA) is a feature of S3 bucket which enables edge locations to optimize network path and uses network optimization protocols to speed up the data transfer to s3. Ideal for large file uploads to s3.

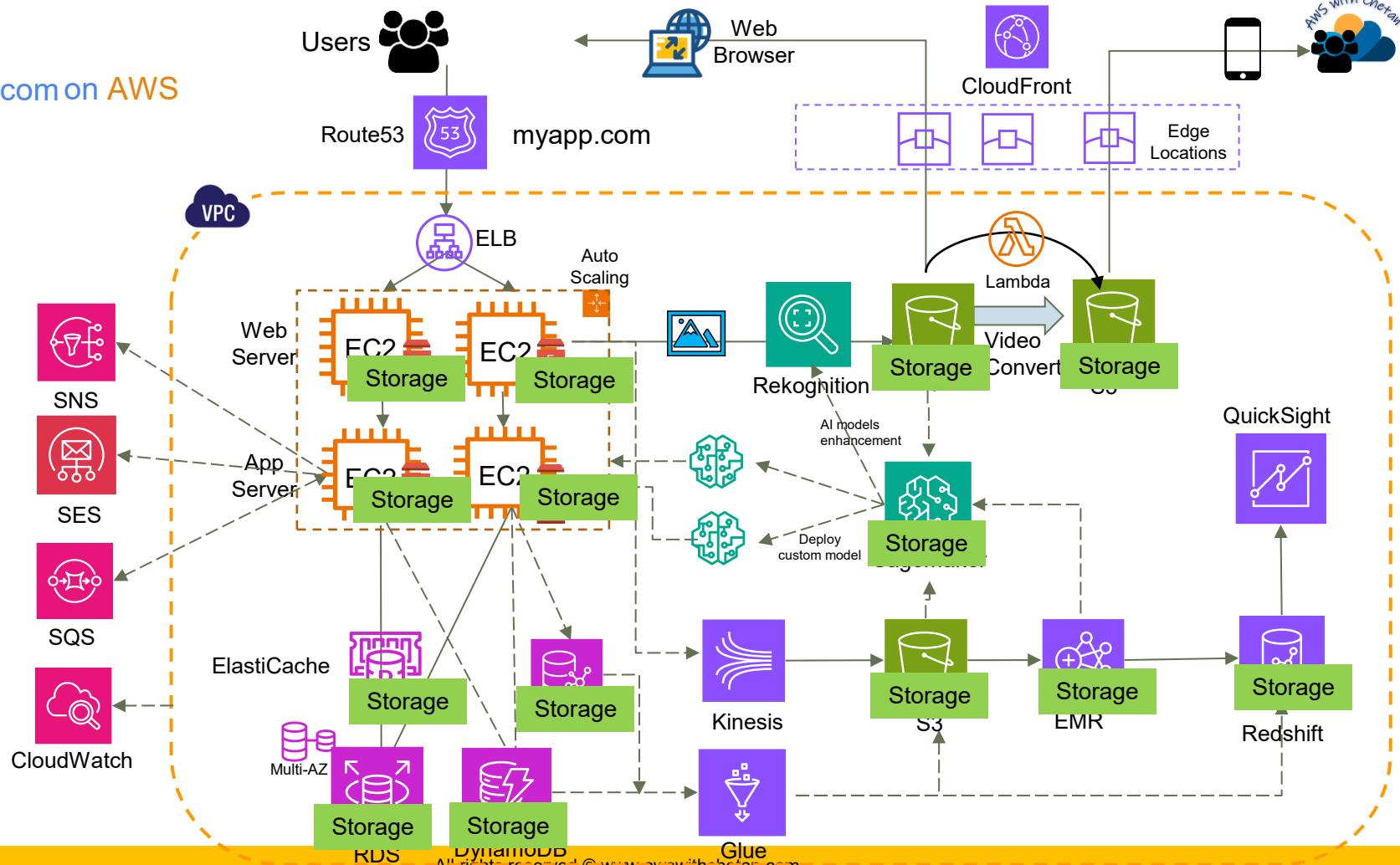


# AWS Storage services

# AWS Storage services



myapp.com on AWS

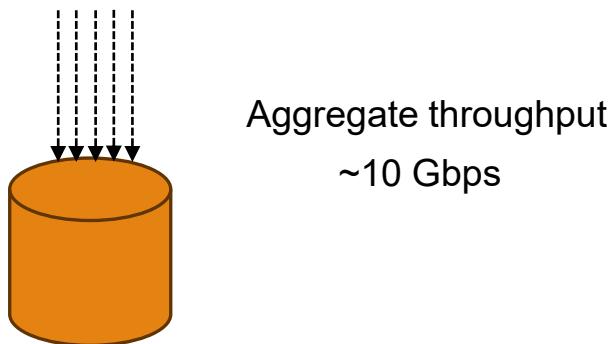
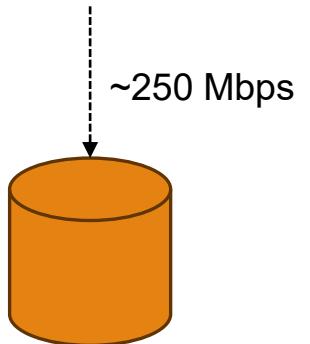


# Block storage / File storage / Object Storage

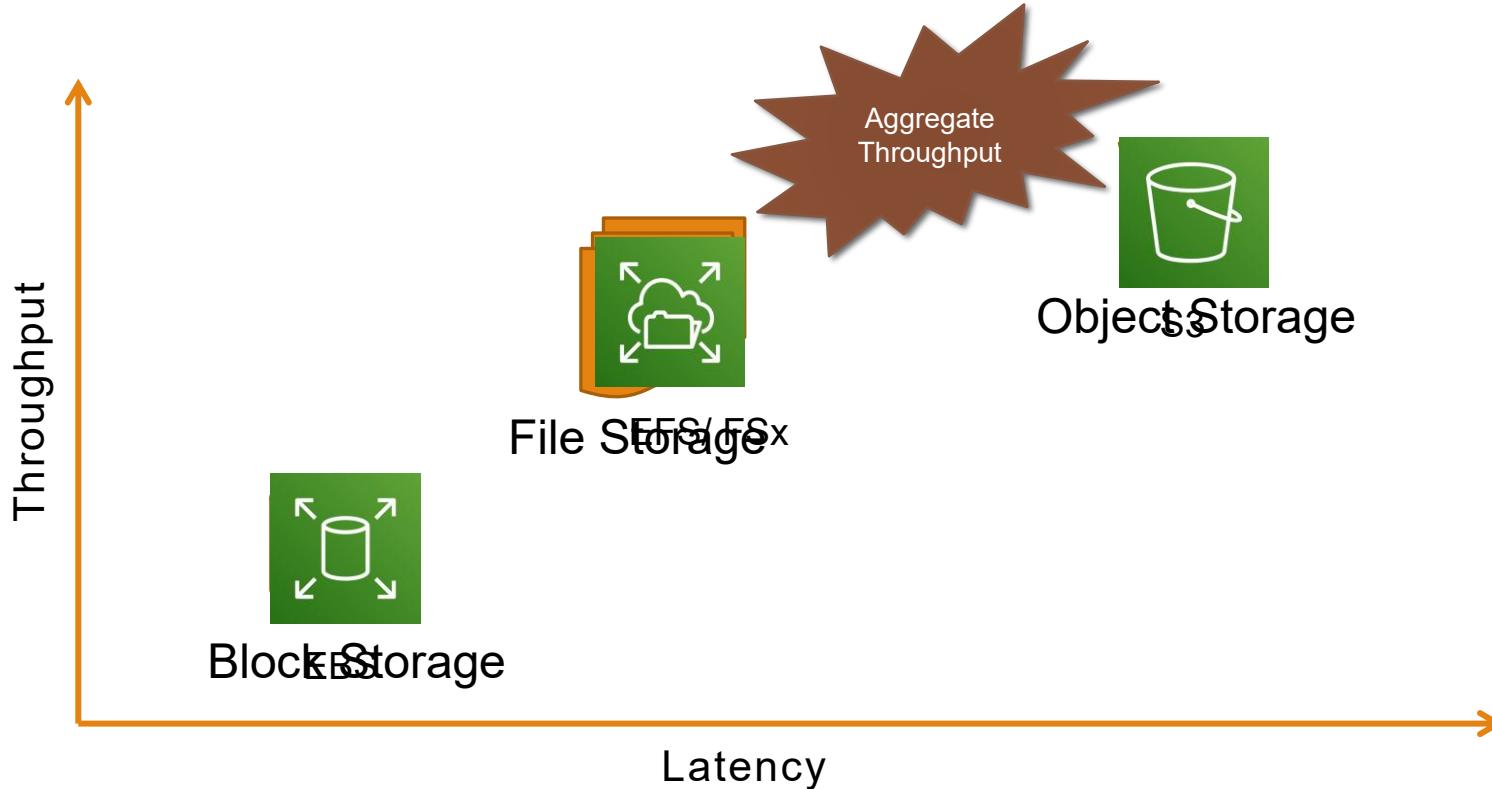
# Quick look at storage performance parameters

## Storage Performance parameters – IOPS, Throughput, Latency

- What is IOPS? – Input Output operations per second e.g. 3000 IOPS
- What is Throughput? – How fast storage can read/write data e.g. 10 MB/s
- What is Latency? – Time delay between request of data and response of data



# Performance comparison of storage types

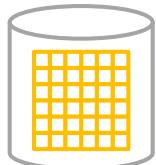


# AWS Storage services



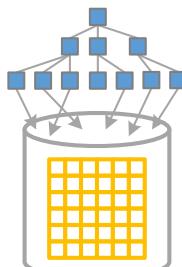
EBS

- **Block Storage** – Data is stored into unique blocks
- Host File System places data on Disk using protocols like iSCSI
- Must be attached to EC2
- High performance, high IOPS, low latency



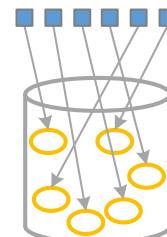
EFS/FSx

- **Shared file system** – Hierarchical structure
- There is a serving File system
- Should be mounted on EC2 or on-premises servers
- High throughput, moderate performance



S3

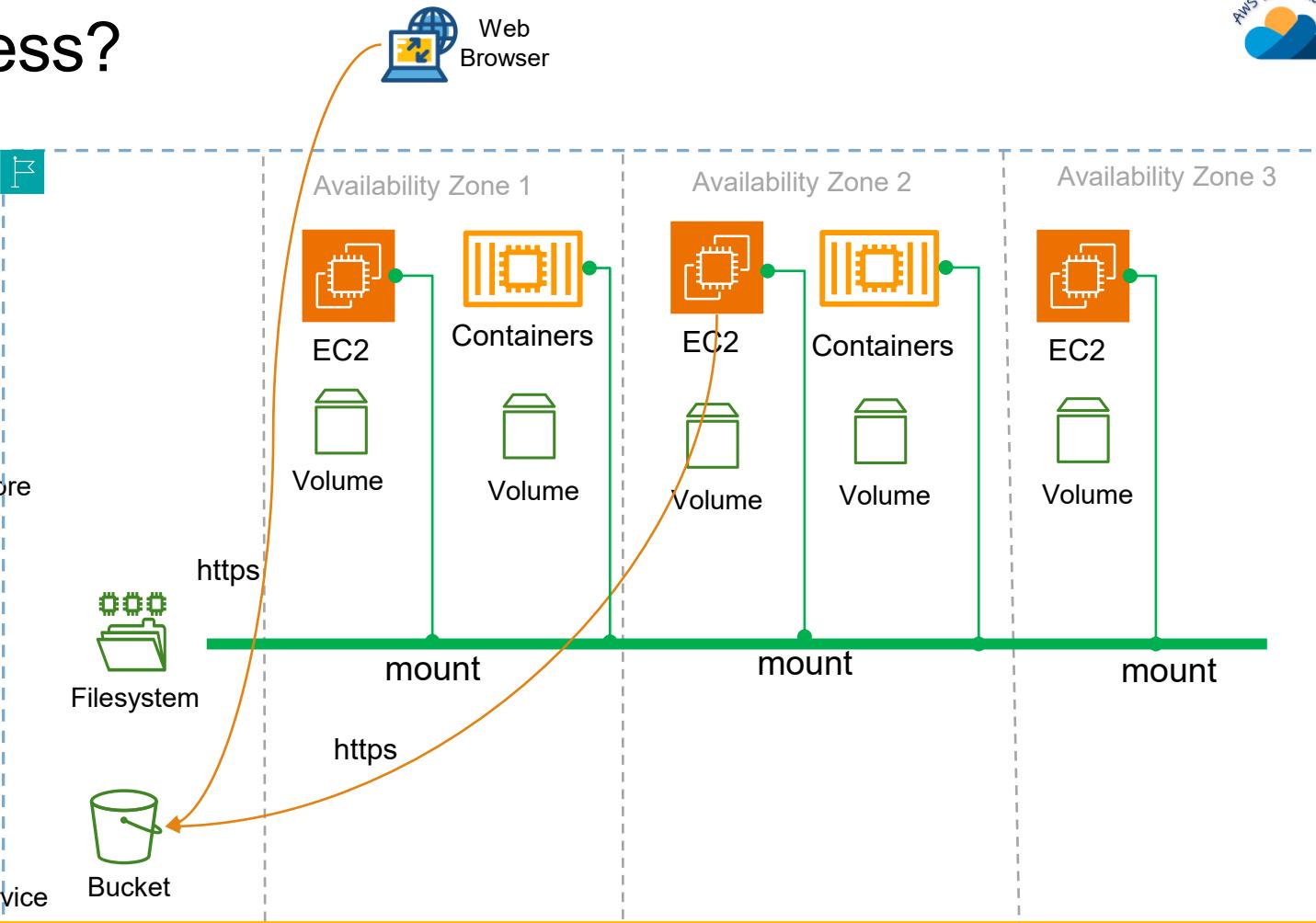
- **Object storage** – Flat structure
- API access to data (HTTPS)
- Metadata driven (Attributes, Policy)
- High Throughput, unlimited storage, moderate performance

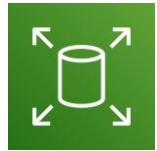


# How to access?

Object Storage | File Storage | Block Storage

- Elastic Block Store (EBS)
- Elastic File System (EFS)
- FSx for Windows
- Simple Storage Service (S3)

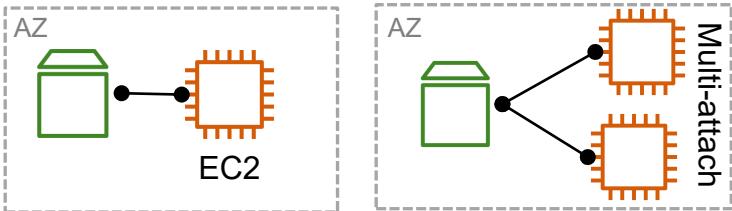




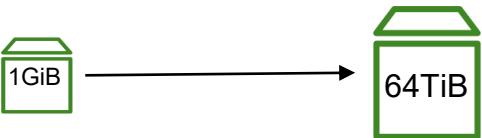
# Elastic Block Storage

# Elastic Block Storage (EBS)

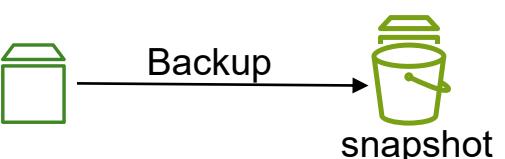
- An Amazon EBS is a block-level storage device that you can attach to your EC2 instance in a given AZ.
- EBS provides high availability, reliability and durability for the data stored in EBS volumes
- Dynamically increase size, modify the provisioned IOPS capacity, and change volume type for existing volumes.
- Allows encryption using encryption keys from AWS Key Management System (KMS).
- EBS volumes data can be backed up using point-in-time snapshots. Snapshots are stored in S3.



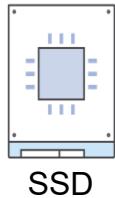
99.9% Availability (within AZ)  
99.999% Durability



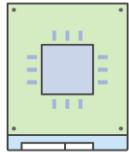
Data at rest encryption



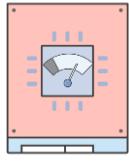
# Amazon EBS use cases



SSD



gp2, gp3



io1, io2



HDD

General Purpose  
SSD

- Operating system
- Relational databases: MySQL, SQL Server, PostgreSQL, SAP, Oracle

Provisioned IOPS  
SSD

- NoSQL Databases
- Cassandra, MongoDB, CouchDB

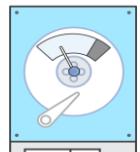
---

Throughput Optimized  
HDD

- Big Data , Analytics
- Kafka, Splunk, Hadoop, Data Warehousing



st1



sc1

Cold HDD

- File / Media CIFS/NFS
- Transcoding, Encoding, Rendering

# Exercise - Attach new EBS Volume to EC2 Instance

- 1 Launch an EC2 instance (Amazon Linux, t2.micro)
- 2 Create an EBS volume of 10 GB in same AZ as EC2 (EC2 -> Volumes) -> Attach a volume to an instance
- 3 SSH into the instance and mount the volume

```
$ lsblk
```

```
$ sudo mkfs -t ext4 /dev/xvdf [This could be different for you say /dev/xvdb]
```

```
$ sudo mkdir /disk1
```

```
$ sudo mount /dev/xvdf /disk1
```

```
$ df [check that you see your disk in df output]
```

```
$ sudo vim /disk1/file.txt [write something in the file]
```

# Exercise - Create EBS volume snapshot

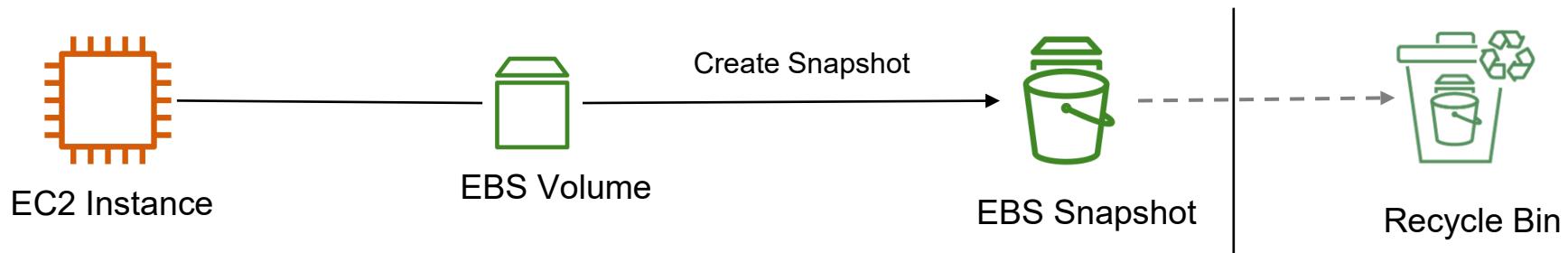
- 1 Launch an EC2 instance (Amazon Linux, t2.micro)
- 2 Create an EBS volume of 10 GB in same AZ as EC2 (EC2 -> Volumes) -> Attach a volume to an instance
- 3 SSH into the instance and mount the volume
  - lsblk
  - sudo mkfs -t ext4 /dev/xvdf (This could be different for you say /dev/xvdb)
  - sudo mkdir /disk1
  - sudo mount /dev/xvdf /disk1
  - df (check that you see your disk in df output)
  - sudo vim /disk1/file.txt (write something in the file)
- 4 Unmount and detach the volume
  - sudo umount /disk1
- 5 Delete the volume



# EBS Snapshots

# EBS Snapshots

- Point in time backup of EBS Volume.
- Snapshot are stored in S3 and are highly durable within a region.
- Snapshots are incremental – **one full snapshot + changed data only**
- Can be shared with other AWS Accounts
- Can be copied to other AWS Regions
- You can delete snapshots and retain it in Recycle Bin for a desired retention period



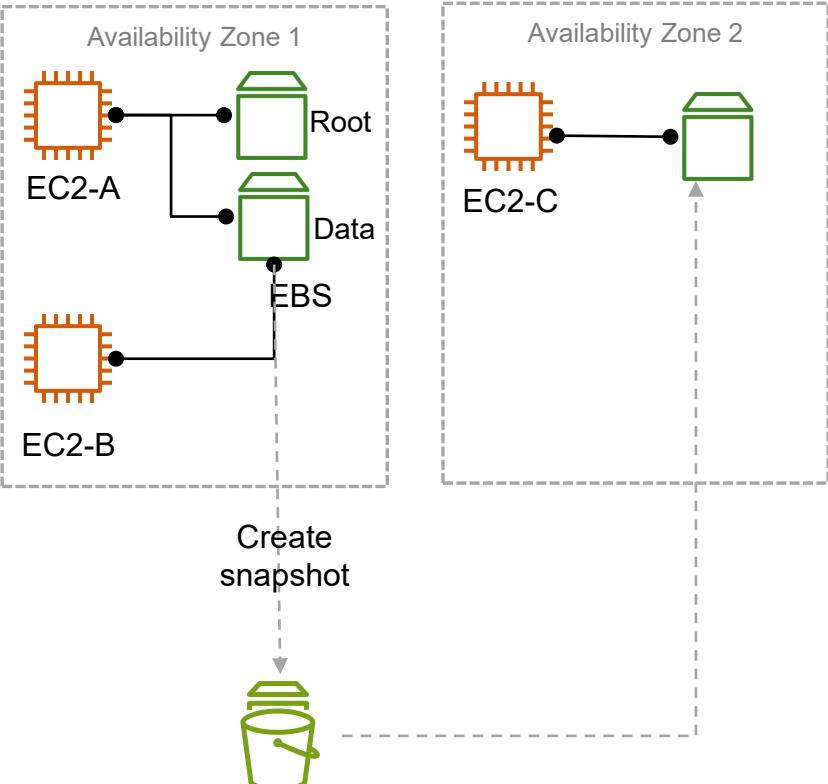
# Exercise – Create EBS snapshot

- 1 Go to EBS volume -> Select volume -> Actions -> Create snapshot
- 2 Terminate EC2 instance and Delete EBS volume

# EBS – Good to know

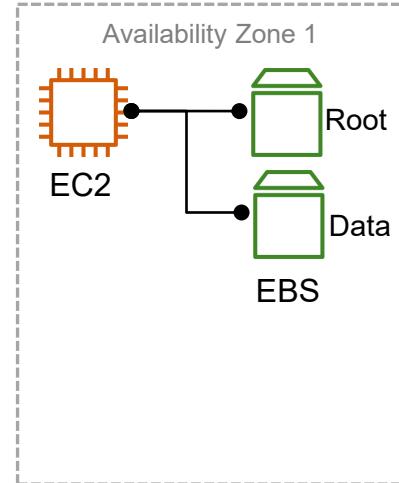
# EBS volume AZ considerations

- EBS volumes resides in given AZ
- EBS volume can be detached and attached to another instance in the same AZ
- For moving EBS volume across the AZs, create a snapshot of volume and then create a new volume from the snapshot in another AZ



# Delete on Termination Attribute

- EBS volume attribute **DeleteOnTermination** defines whether volume is deleted or retained when EC2 instance terminates
- By default, the root EBS volume is deleted (`DeleteOnTermination = true`)
- By default, any other (non-root) EBS volume is not deleted (`DeleteOnTermination = false`)
- This attribute value can be controlled by the user (AWS console / CLI)



	Volume ID	Device name	Volume size (GiB)	Attachment status	Attachment time	Encrypted	KMS key ID	Delete on termination
<input checked="" type="checkbox"/>	vol-[REDACTED]	/dev/xvda	8	<span>Attached</span>	2024/07/27 21:58 GMT+5:30	No	-	Yes
<input type="checkbox"/>	vol-[REDACTED]	/dev/sdb	20	<span>Attached</span>	2024/07/27 21:58 GMT+5:30	No	-	No

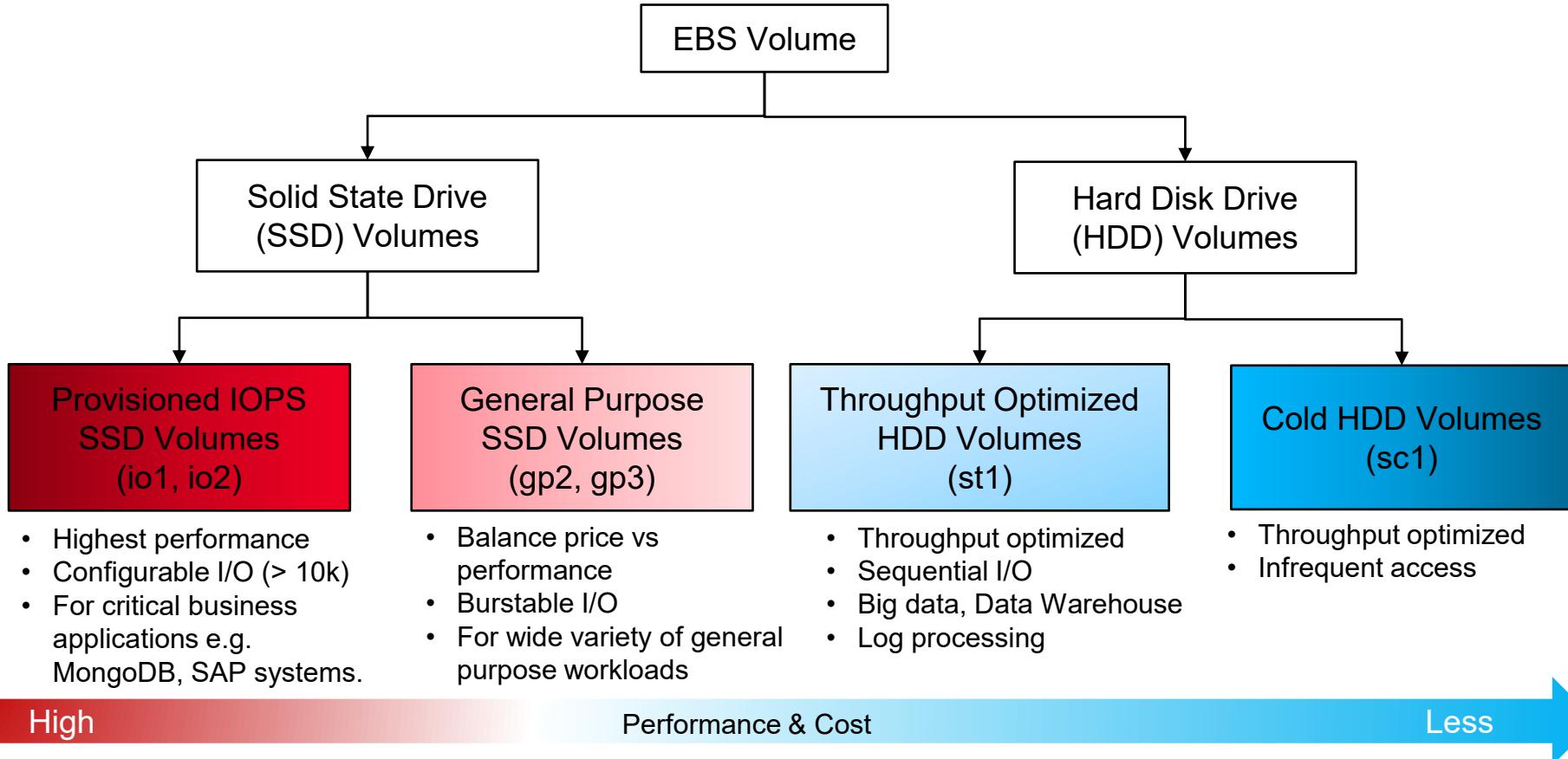
# Exercise – Delete EBS volume

- 1 Terminate EC2 instance – Check what happens to EBS volume that we had created
- 2 Wait for EC2 instance to be terminated. Go to EBS volumes -> Select Volume -> Actions -> Delete volume
- 3 Verify that EBS snapshot is still there, and you can create a new EBS volume from this snapshot

# Amazon EBS - summary

1. EBS is network drive providing block storage to the EC2 instance
2. Generally, EBS volume is attached to a single EC2 instance but depending on EC2 instance types (based on nitro system), specific EBS volumes can be multi-attach.
3. EBS volume types – General purpose (gp2, gp3), Provisioned IOPS (io1, io2), Throughput optimized HDD (st1), Cold HDD (sc1)
4. EBS volumes are in single AZ
5. EBS volumes can be backed up using EBS snapshot
6. EBS snapshots are incremental
7. For moving EBS volume across AZ, create a snapshot and then create a volume in other AZ
8. EBS volume has an attribute called ‘DeleteOnTermination’
9. By default, for Root volume this flag is True and for non-root volumes it is False
10. For long term retention of EBS data, create a snapshot and delete the volume

# EBS volume types

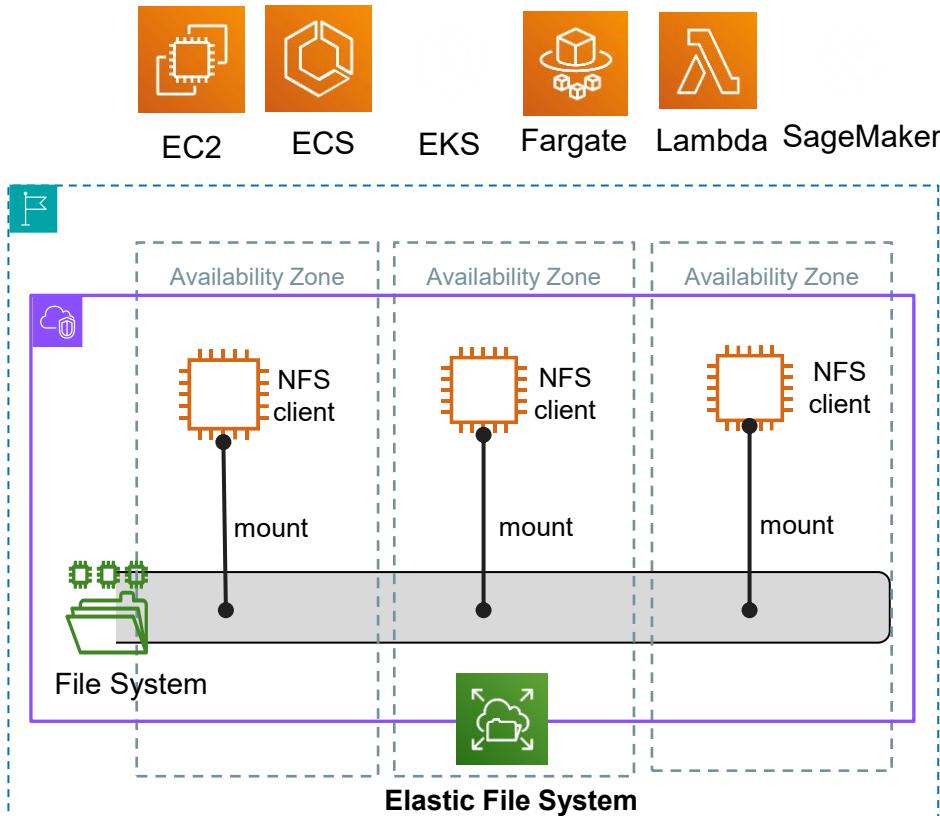




# Elastic File System

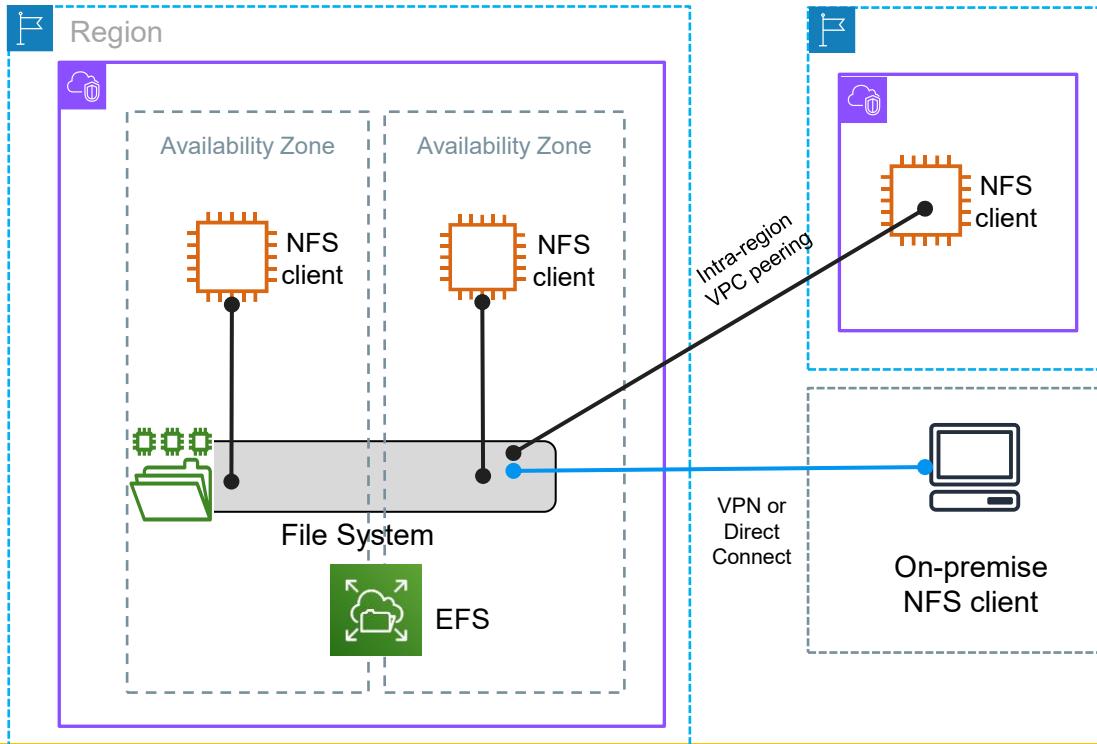
# Amazon Elastic File System (EFS)

- A managed **Network File System (NFS 4)**
- Can be shared by hundreds of EC2 Linux instances
- Works with most of the AWS compute services
- Serverless - no need to manage infrastructure
- Elastic – no need to pre-provision the capacity, pay as per the storage used
- Use cases:
  - Containerized and serverless applications
  - Machine learning training
  - Web serving and content management
  - User home directories



# Amazon Elastic File System (EFS)

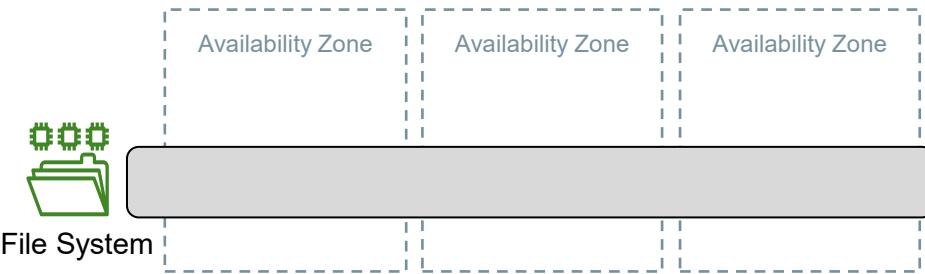
- EFS filesystem can be accessed by instances in a different VPC within or across AWS regions over VPC peering connection and from On-premises host if it's connected to the VPC over VPN or Direct connect



# EFS Filesystem storage options – Regional vs OneZone

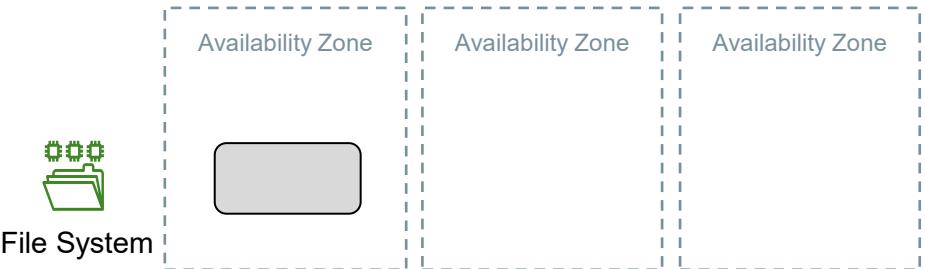
## Standard Storage (Regional)

- Stores data redundantly across multiple Availability Zones in the region
- Provides 11 9's of durability



## OneZone Storage

- Stores data within in a single Availability Zone
- Ideal for non-critical / reproducible data
- 50% cheaper than standard regional store filesystem



# EFS Storage Classes – Based on access pattern

## EFS Standard

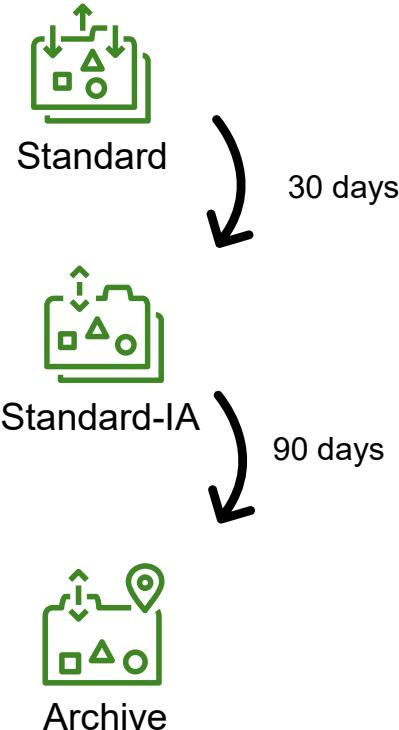
- SSD storage designed to deliver sub-millisecond latency for active data

## EFS Standard-Infrequent Access (EFS-IA)

- Cost-optimized for data accessed only a few times a quarter which doesn't need the sub-millisecond latencies of EFS Standard.
- EFS Lifecycle policy "**Transition into IA**" automatically moves the files from Standard storage to Standard-IA if files are not accessed for 30 days

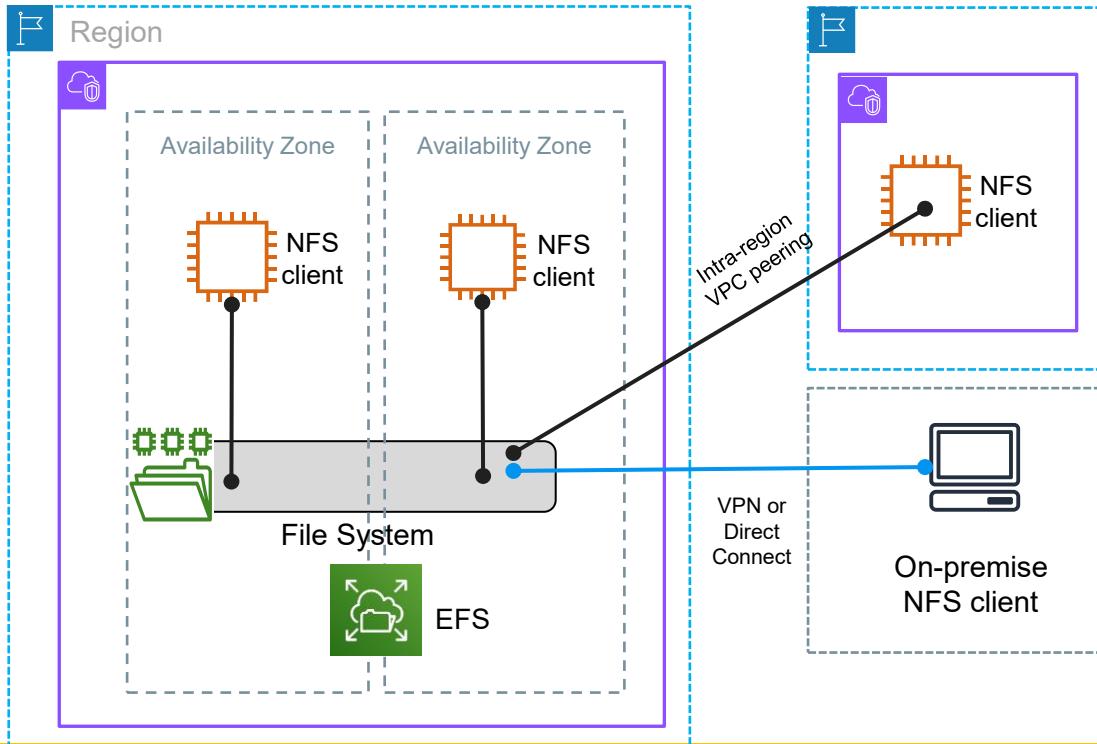
## EFS Archive

- Cost-optimized for long-lived data accessed a few times a year or less and offering similar performance to EFS IA
- EFS Lifecycle policy "**TransitionToArchive**" automatically moves the files to EFS archive storage if files are not accessed for 90 days
- EFS Lifecycle policy "**Transition to Standard**" is set to None by default.



# Amazon Elastic File System (EFS)

- EFS filesystem can be accessed by instances in a different VPC within or across AWS regions over VPC peering connection and from On-premises host if it's connected to the VPC over VPN or Direct connect

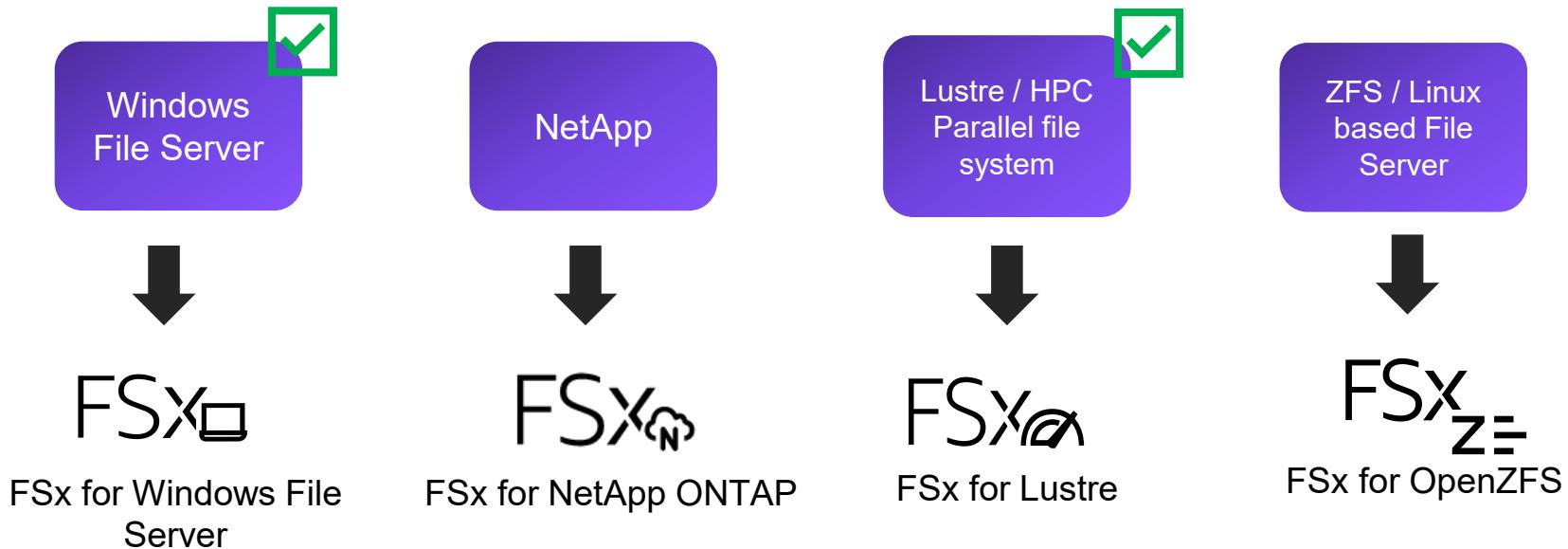




# Amazon FSx

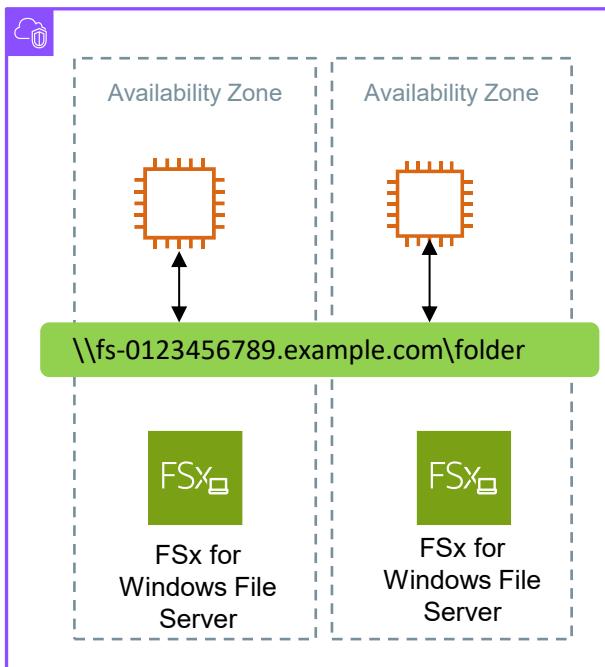
# Amazon FSx

- Like-to-Like 3<sup>rd</sup> party fully managed file systems on AWS



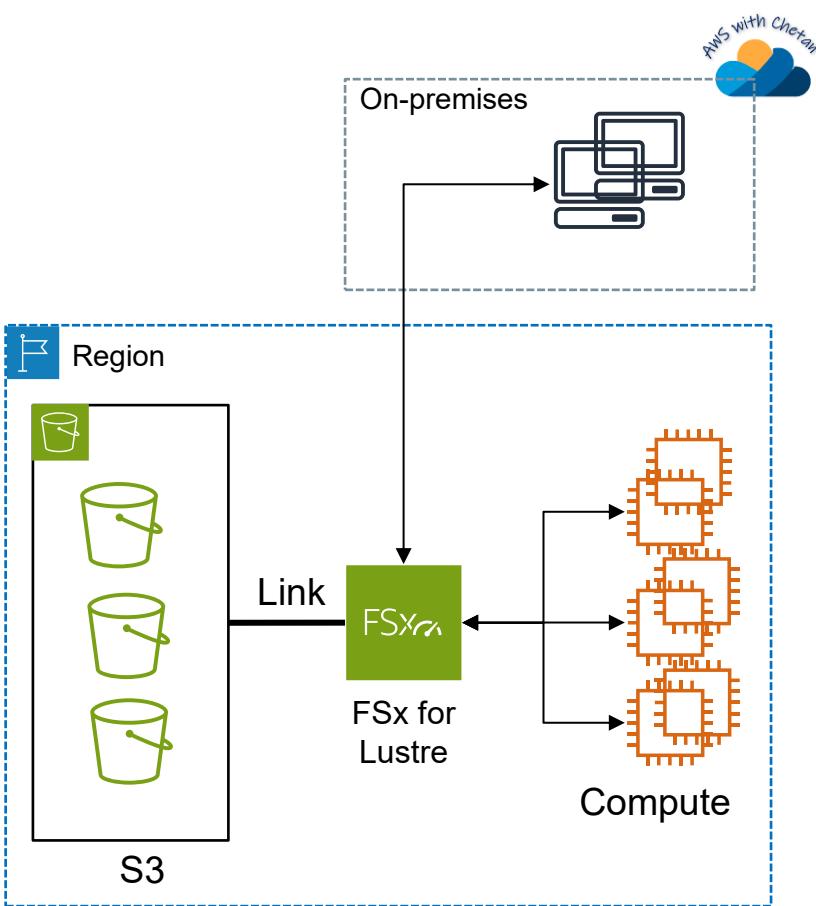
# FSx for Windows File Server

- A fully managed, reliable and scalable **Windows shared file system**
- Uses Windows File Server
- Supports Windows **NTFS** and **SMB** protocols
- Accessible across VPCs, AWS account and regions
- Accessible from within AWS or on-premises
- Integration with Active Directory



# FSx for Lustre

- High performance File System for HPC workloads shared by thousands of compute machines
- FSx for Lustre provides a high-performance file interface for Amazon S3 objects
- Data stored in Amazon S3 is loaded to FSx for processing
- Output of data processing is sent to Amazon S3 for retention
- **Use cases:** Access ML training data in S3, CFD simulations, VFX, rendering, transcoding





# Amazon S3

# Amazon S3

- Amazon S3 is a web-based object storage service which provides virtually unlimited space
- One of the early and most powerful service launched by AWS (in 2006)
- Designed for 99.999999999% durability.
- Unlike block storage (EBS) or file storage (EFS/FSx), S3 uses object storage, which is ideal for storing vast amounts of unstructured data.
- Customers of all sizes and industries use S3 for range of use cases.



Amazon S3

# Amazon S3 use cases & customers

- Datalake
- Media store
- Backup and Restore
- Archive
- Enterprise Applications
- IoT devices
- Big data Analytics
- Static websites
- Content Storage and Distribution



# S3 Bucket and objects

- Data is organized into 'Bucket'.
- Buckets are created and located in the AWS region and has globally unique name
- Bucket contain Objects (single object maximum size = 5TB, infinite number of Objects)
- Objects are identified with unique path called object **Key**
- Bucket/ + key = Object full path



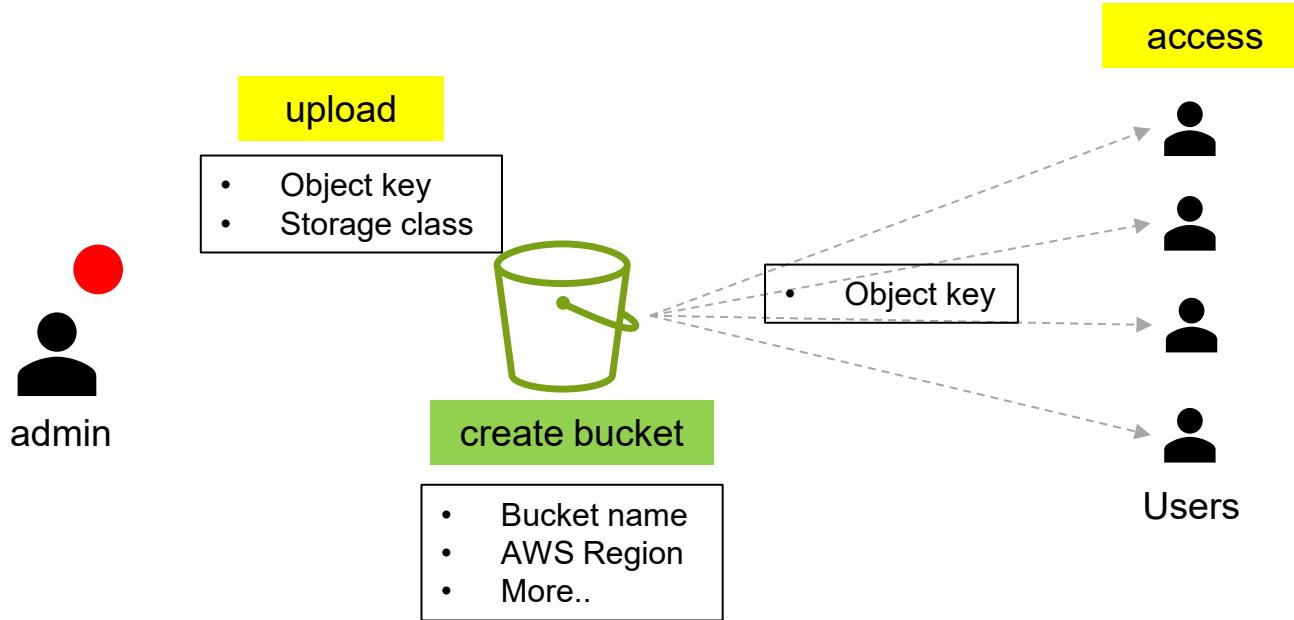
**Bucket:** awswithchetaN

**Object:** section5/s3-overview.pdf

**URI:** s3://awswithchetaN/section5/s3-overview.pdf

**Object URL:** <https://awswithchetaN.s3.ap-south-1.amazonaws.com/section5/s3-overview.pdf>

# How to use S3?



- *Admin must have permissions to create bucket and upload object*
- *Users must have permissions to read objects from this bucket*

# Exercise - Create bucket and upload file



- 1 Create a S3 bucket in region of your choice
- 2 Upload an object (image/video/pdf anything)

# Amazon S3 features

- **Scalable:** Can store virtually unlimited data
- **Durable:** Designed for 99.999999999% durability
- **Highly available:** Offers 99.99% availability
- **Secure:** Web accessible through Command line, secure API or HTTPS
- **Accessible:** Data can be accessed from anywhere over the internet
- Multiple storage classes for different use cases
- Data Lifecycle Rules for moving data across storage classes
- Can be used for Static (http/s) Website Hosting
- Cost effective e.g. \$0.023/GB/month for standard storage & \$0.004/GB/month for archive



# S3 Storage Classes

## Frequently Accessed Objects

- S3 Standard (General Purpose)
- 11 9's (99.99999999%) of durability

## Infrequently Accessed Objects

- Standard-IA
- S3 One Zone-IA

## Automatically moving Data

- S3 Intelligent-Tiering

## Archiving Objects – S3 Glacier

- S3 Glacier Instant Retrieval
- S3 Glacier Flexible Retrieval
- S3 Glacier Deep Archive



S3 Standard



S3 Standard-IA



S3 One Zone-IA



S3 Intelligent-Tiering



S3 Glacier Instant Retrieval



S3 Glacier Flexible Retrieval



S3 Glacier Deep Archive

# S3 Storage Classes



S3  
Standard



S3 Intelligent-  
Tiering



S3 Standard-IA



S3 One Zone-IA



S3 Glacier  
Instant  
Retrieval



S3 Glacier  
Flexible Retrieval

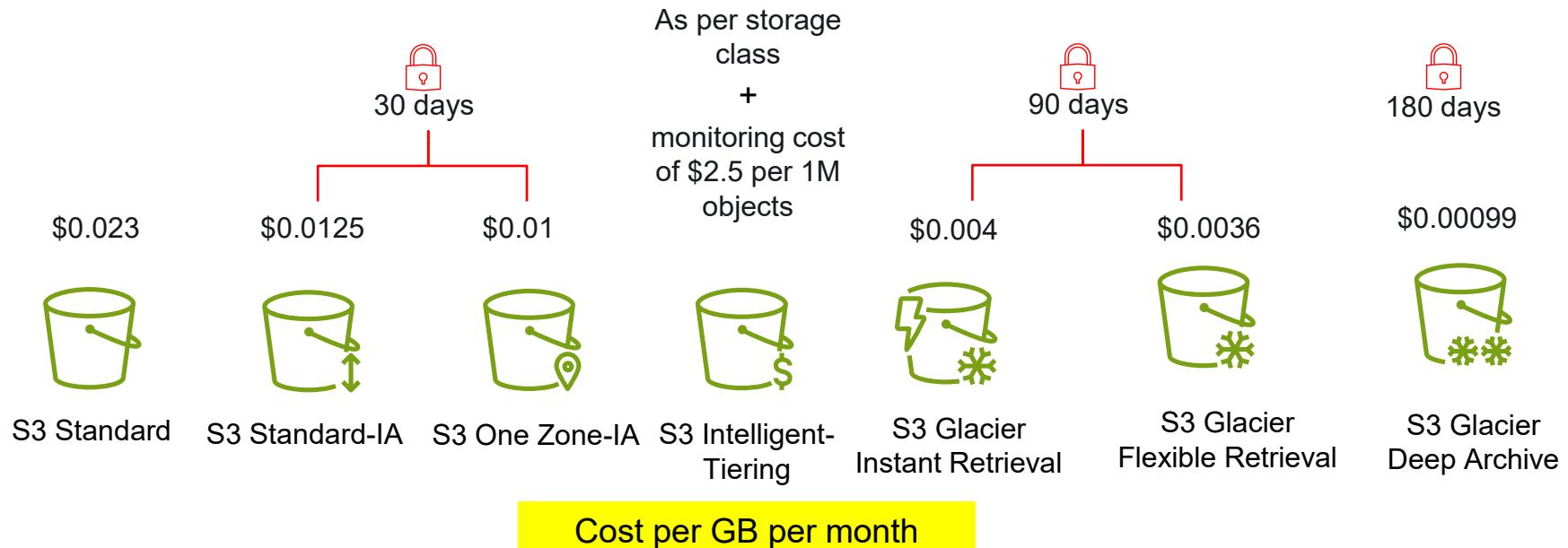


S3 Glacier  
Deep Archive

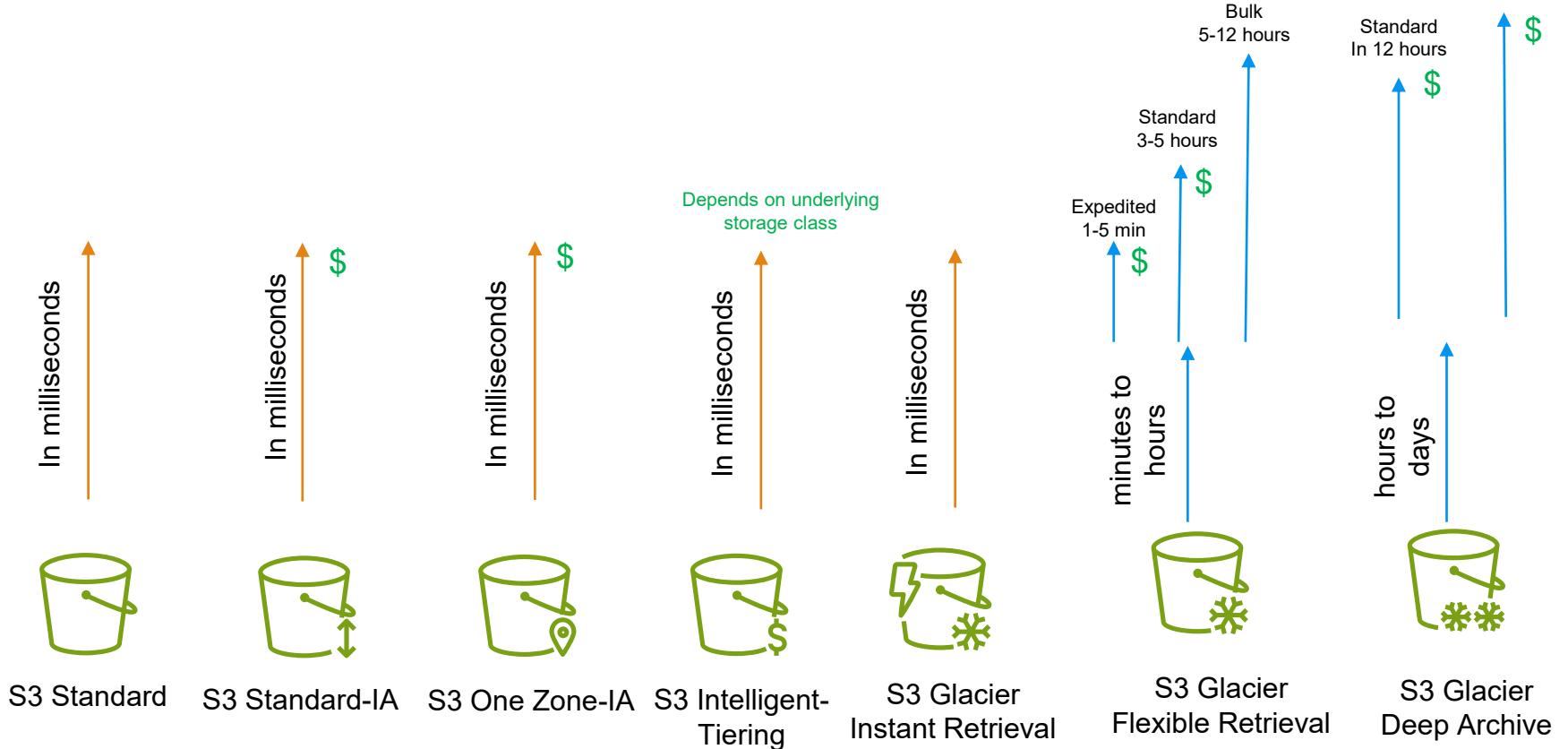
	Low	< single-digit millisecond	> minutes to hours	High
Access Frequency	High	once a quarter	once or twice in a year	< once a year
Storage cost	High			Low
Data Retrieval cost		Low		High
Number of requests cost	Low			High

# Data storage cost

\*N. Virginia region, initial tier



# Data Retrieval time/cost



# S3 Storage Classes – Use cases



## S3 Standard

- Big Data analytics
- Mobile and Gaming applications
- Media Content & distribution



## S3 Intelligent-Tiering

- For unknown, changing or unpredictable data access pattern
- Data lakes, data analytics



## S3 Standard-IA

- Hot backups
- Data for disaster recovery
- Older media contents, logs



## S3 One Zone-IA

- Storing secondary backup copies of on-premise data
- Storing data that you can recreate

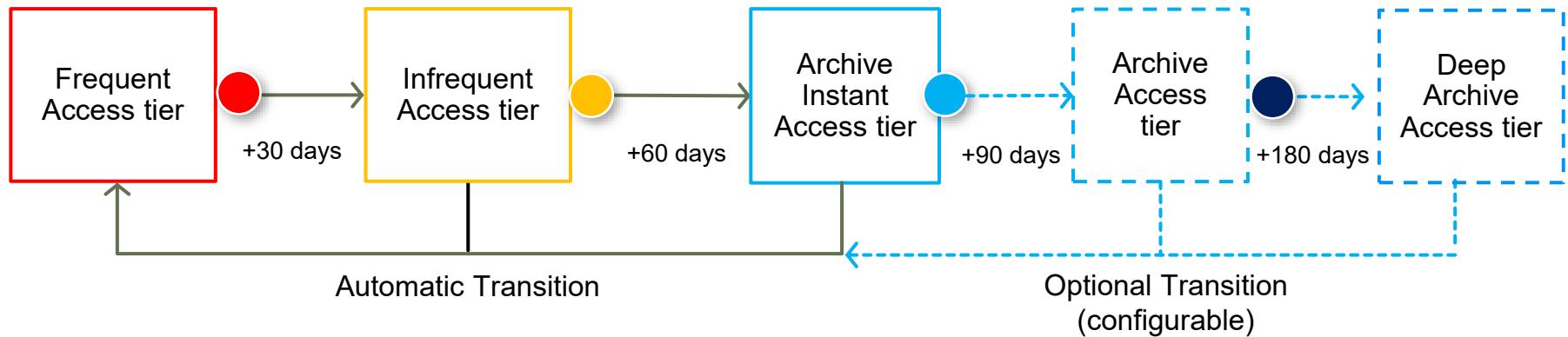


## S3 Glacier

- Backup and Archive
- Long term storage for regulatory and compliance

# Amazon S3 Intelligent-Tiering

For data with unknown or changing access patterns



Moves objects between three access tiers for a small monthly monitoring and automation fee

# Amazon S3 Security

## 1. Access / permissions

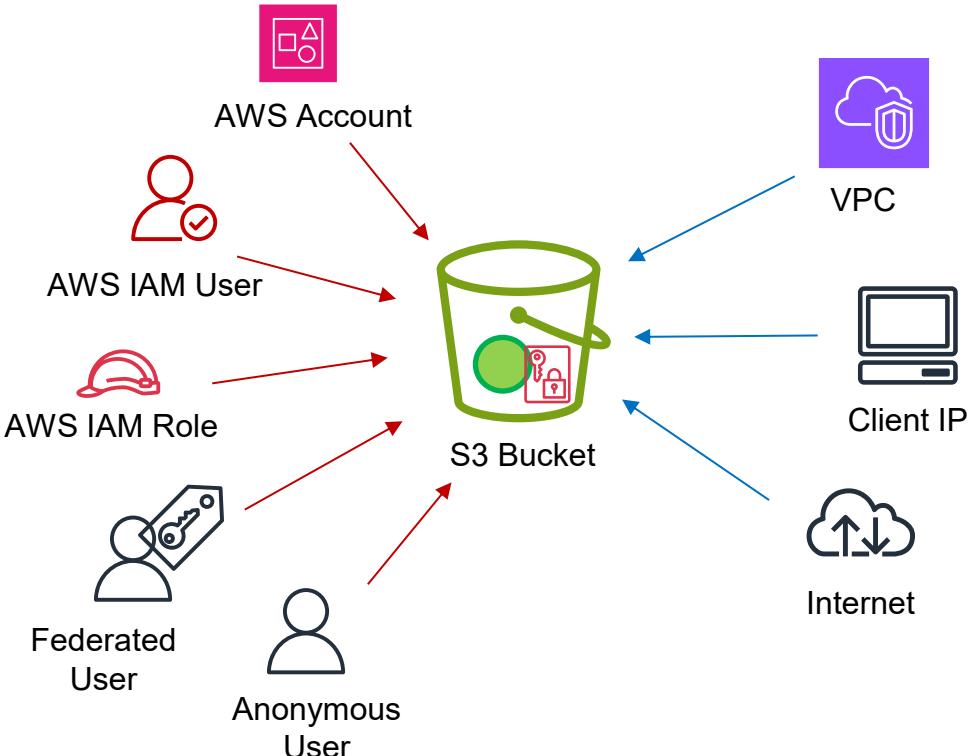
- Block Public Access
- Access Control List (ACL)
- IAM User or Role policy
- Bucket Policy ★

## 2. Network

- Restrict access from VPC or specific IP address

## 3. Data

- Server-side encryption
- Client-side encryption



# Amazon S3 Security - Access

## Block Public Access

- To prevent company data leaks
- Set at the account level or bucket level
- Best practice: Leave these ON unless bucket needs to be publicly accessible

### Block all public access

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

#### Block public access to buckets and objects granted through **new** access control lists (ACLs)

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for permissions that allow public access to S3 resources using ACLs.

#### Block public access to buckets and objects granted through **any** access control lists (ACLs)

S3 will ignore all ACLs that grant public access to buckets and objects.

#### Block public access to buckets and objects granted through **new** public bucket or access point policies

S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies.

#### Block public and cross-account access to buckets and objects through **any** public bucket or access point policies

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

# Amazon S3 Security - Access

## Access Control List (ACL)

- Each bucket and object has an ACL attached to it
- ACL controls which AWS accounts or groups are granted access and the type of access
- Bucket owner can enable/disable ACLs
- ACLs are disabled by default



Bucket Level ACL



Object Level ACL

**Access control list (ACL)**  
Grant basic read/write permissions to other AWS accounts. [Learn more](#)

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account) Canonical ID: [REDACTED]	<input checked="" type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Everyone (public access) Group: <a href="http://acs.amazonaws.com/groups/global/AllUsers">http://acs.amazonaws.com/groups/global/AllUsers</a>	<input type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Authenticated users group (anyone with an AWS account) Group: <a href="http://acs.amazonaws.com/groups/global/AuthenticatedUsers">http://acs.amazonaws.com/groups/global/AuthenticatedUsers</a>	<input type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input type="checkbox"/> Read <input checked="" type="checkbox"/> Write
S3 log delivery group Group: <a href="http://acs.amazonaws.com/groups/s3/LogDelivery">http://acs.amazonaws.com/groups/s3/LogDelivery</a>	<input type="checkbox"/> List <input type="checkbox"/> Write	<input type="checkbox"/> Read <input type="checkbox"/> Write

A majority of modern use cases in Amazon S3 **no longer** require the use of ACLs.

# Amazon S3 Security – Access

## IAM User or Role policy



Does user have permission?

## S3 Bucket Policy



Does bucket allow users to access?



Anonymous user on the internet

# Amazon S3 Security – Access

## S3 Bucket Policy

It's a JSON-based policy

- **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
- **Condition:** Apply policy when condition is true

Examples of bucket policy:

- Grant public read access to the bucket e.g. website
- Objects must be encrypted at upload
- Cross Account access to bucket and objects

```
{  
    "Version": "2012-10-17",  
    "Statement": [  
        {"Principal": {  
            "AWS": [  
                "arn:aws:iam::111122223333:user/JohnDoe"  
            ]  
        },  
        "Effect": "Allow",  
        "Action": [  
            "s3:PutObject"  
        ],  
        "Resource": [  
            "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
        ],  
        "Condition": {  
            "StringEquals": {  
                "s3:RequestObjectTag/Department": "Finance"  
            }  
        }  
    ]  
}
```

# Example: How to make S3 bucket public?

## 1. Block Public Access should be disabled

### **Block all public access**

Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

#### **Block public access to buckets and objects granted through new access control lists (ACLs)**

S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for permissions that allow public access to S3 resources using ACLs.

#### **Block public access to buckets and objects granted through any access control lists (ACLs)**

S3 will ignore all ACLs that grant public access to buckets and objects.

#### **Block public access to buckets and objects granted through new public bucket or access point policies**

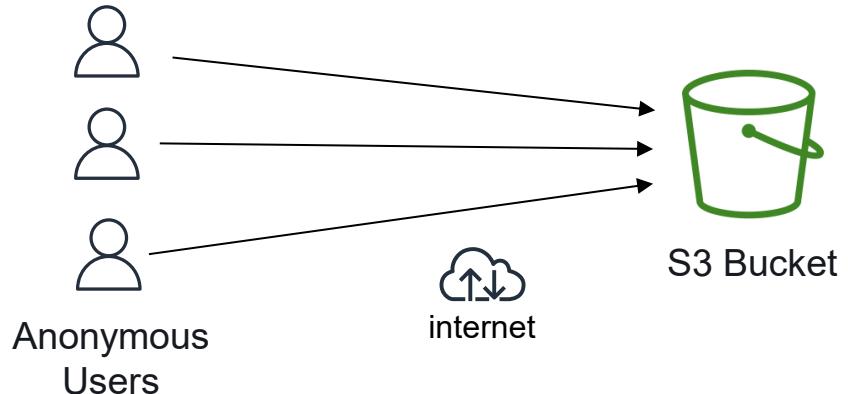
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies.

#### **Block public and cross-account access to buckets and objects through any public bucket or access point policies**

S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

# Exercise: Public access to bucket

1. Block Public Access should be disabled
2. Bucket policy to allow access

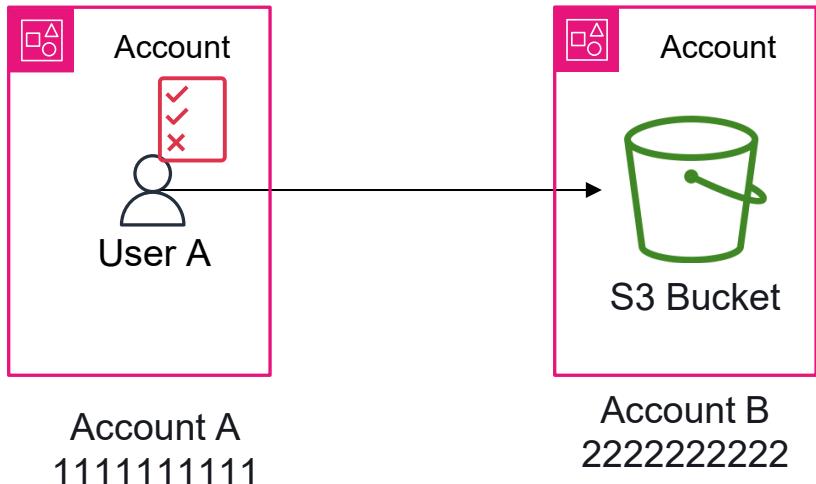


## Bucket Policy

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

# Example: S3 bucket cross-account access

1. Bucket policy in Account B to allow access to Account A
2. IAM user/role in Account A to allow UserA to access Bucket in Account B

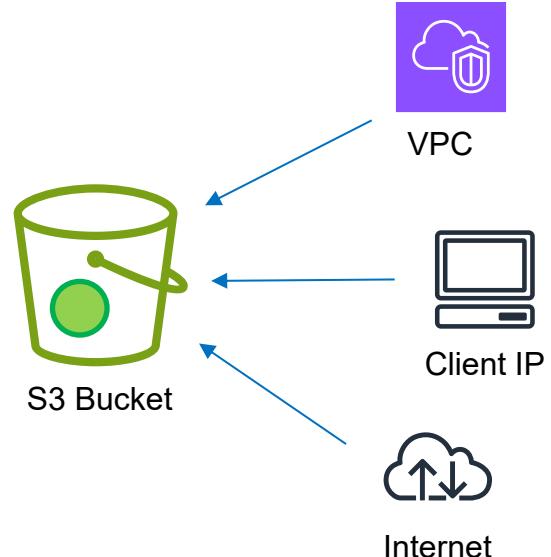


```
Bucket Policy

{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "arn:aws:iam::111111111111:root",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::EXAMPLE-BUCKET/*"
    }
  ]
}
```

# Amazon S3 Security – Network

- Restrict access from specific VPC
- Restrict access from specific VPC endpoint
- Restrict access from VPC source IP ranges
- Restrict access from specific external IP



# Amazon S3 Security – Network

- **Restrict access from specific VPC**
- Restrict access from specific VPC endpoint
- Restrict access from VPC source IP ranges
- Restrict access from specific external IP

```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PutObjectIfNotVPCID",  
      "Effect": "Deny",  
      "Action": "s3:PutObject",  
      "Resource": "arn:aws:s3:::amzn-s3-demo-bucket3/*",  
      "Condition": {  
        "StringNotEqualsIfExists": {  
          "aws:SourceVpc": "vpc-1234567890abcdef0"  
        },  
        "Bool": {  
          "aws:ViaAWSService": "false"  
        }  
      }  
    }  
  ]  
}
```

# Amazon S3 Security – Network

- Restrict access from specific VPC
- **Restrict access from specific VPC endpoint**
- Restrict access from VPC source IP ranges
- Restrict access from specific external IP

```
{  
  "Id": "VPCE",  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "VPCE",  
      "Action": "s3:*",  
      "Effect": "Deny",  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
      ],  
      "Condition": {  
        "StringNotEquals": {  
          "aws:SourceVpce": [  
            "vpce-1111111",  
            "vpce-2222222"  
          ]  
        }  
      },  
      "Principal": "*"  
    }  
  ]  
}
```

# Amazon S3 Security – Network

- Restrict access from specific VPC
- Restrict access from specific VPC endpoint
- **Restrict access from VPC source IP ranges**
- Restrict access from specific external IP

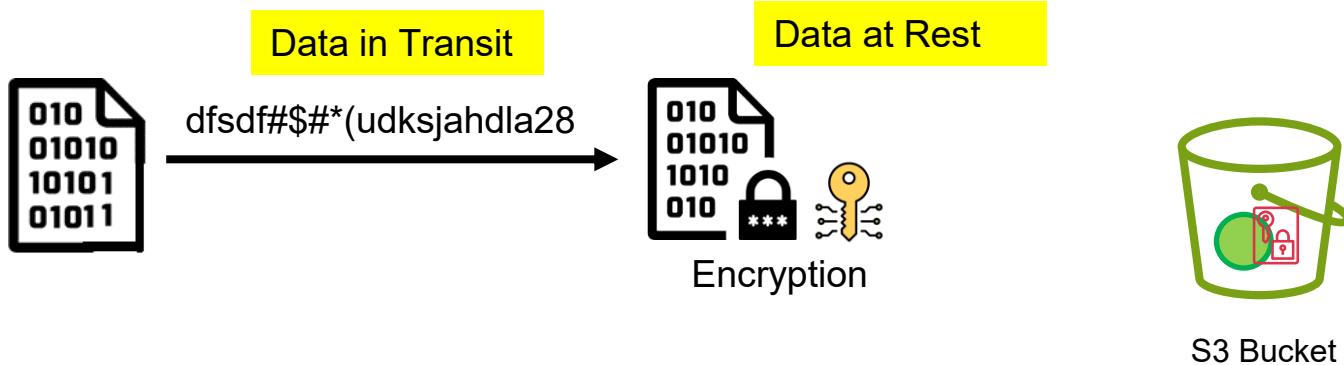
```
{  
  "Id": "VpcSourceIp",  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "VpcSourceIp",  
      "Action": "s3:*",  
      "Effect": "Deny",  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
      ],  
      "Condition": {  
        "NotIpAddress": {  
          "aws:VpcSourceIp": [  
            "10.1.1.1/32",  
            "172.1.1.1/32"  
          ]  
        }  
      },  
      "Principal": "*"  
    }  
  ]  
}
```

# Amazon S3 Security – Network

- Restrict access from specific VPC
- Restrict access from specific VPC endpoint
- Restrict access from VPC source IP ranges
- **Restrict access from specific external IP**

```
{  
  "Id": "SourceIP",  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "SourceIP",  
      "Action": "s3:*",  
      "Effect": "Deny",  
      "Resource": [  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET",  
        "arn:aws:s3:::DOC-EXAMPLE-BUCKET/*"  
      ],  
      "Condition": {  
        "NotIpAddress": {  
          "aws:SourceIp": [  
            "11.11.11.11/32",  
            "22.22.22.22/32"  
          ]  
        }  
      },  
      "Principal": "*"  
    }  
  ]  
}
```

# Amazon S3 Security – Data



# Amazon S3 Security – Data

**Data in Transit → HTTPS/TLS**

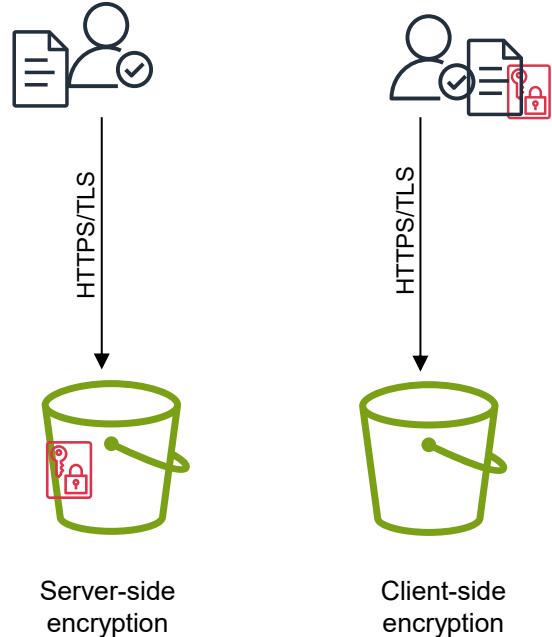
**Data at Rest →      Encryption**

## Server-side encryption

- S3 service encrypts object while saving it on disk and decrypts it when you access the object
- Supports different Encryption keys SSE-S3, SSE-KMS, SSE-C

## Client-side encryption

- Data is encrypted at the client side and then uploaded to S3 in the encrypted format.
- S3 doesn't know about the encryption and can not decrypt objects

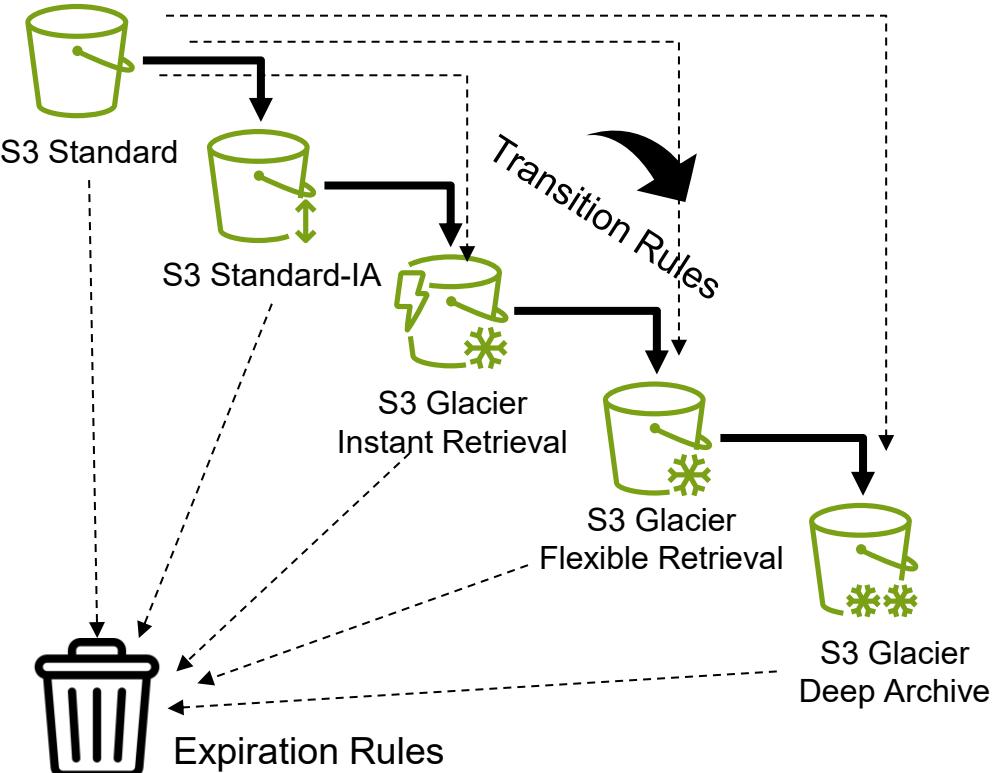


# Amazon S3 features

1. S3 Lifecycle
2. Static Website Hosting
3. Bucket Versioning
4. Bucket Replication
5. IAM Access Analyzer

# S3 Lifecycle

- Set of rules that applies on the group of objects in the S3 bucket
- Transition Action
  - Transitions objects to different storage class
  - Example: Move from Standard class to Standard IA class after 30 days
- Expiration Action
  - When objects expire, the expired objects are deleted by S3
  - Example: Expire objects after 365 days
- S3 lifecycle rules apply to both existing and new objects.



# Static Website Hosting

- Because S3 can be accessed over the Web, we can host Static (**http**) website on S3
- For Setting up Static Website on S3:
  1. Upload static HTML files to S3 bucket
  2. Make S3 bucket Public (Disable Block Public Access & set Bucket Policy)
  3. Enable Static website hosting for the bucket
- Depending on your Region, your Amazon S3 website endpoint follows one of these two formats.
  - s3-website dash (-) Region - **http://bucket-name.s3-website-Region.amazonaws.com**
  - s3-website dot (.) Region - **http://bucket-name.s3-website.Region.amazonaws.com**



# Exercise: Static Website Hosting

- 1 Create S3 Bucket in the region of your choice.
- 2 Disable “Block Public Access” setting for the bucket
- 2 Download sample static website template and extract locally on your machine.
- 3 Upload static website content to S3 bucket:
  - Drag and drop all files and folders directly in the bucket.
  - Make sure index.html file is directly inside bucket and not inside folder.
- 4 Enable Static Website for the bucket:
  - Select Bucket -> Properties -> Static Website Hosting -> Use this bucket to host a website.
  - Provide index document as “index.html” -> Save
- 5 Try to access website using HTTP endpoint. You should get Permissions denied error.  
`<bucket-name>.s3-website.<AWS-region>.amazonaws.com`

# Exercise: Static Website Hosting

- 6 Go to bucket permissions and add following Bucket Policy. This allows public READ access to all objects in your bucket:

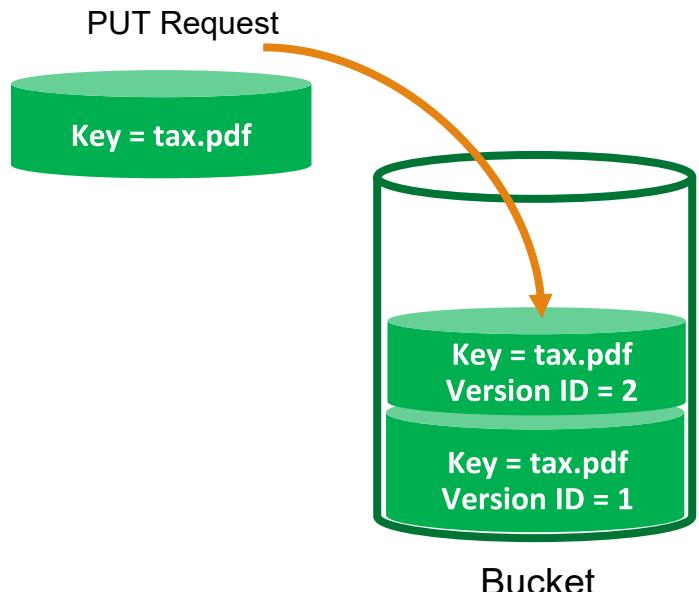
```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "PublicReadGetObject",  
      "Effect": "Allow",  
      "Principal": "*",  
      "Action": "s3:GetObject",  
      "Resource": "arn:aws:s3:::<your bucket name>/*"  
    }  
  ]  
}
```

- 7 Now access website using following URL. You should be able to access it.

<bucket-name>.s3-website.<AWS-region>.amazonaws.com

# Amazon S3 bucket versioning

- Creates a new version with every upload of the object.
- Previous versions are **not overwritten**.
- Should specify version ID for deleting the object version.
- Delete requests without a version ID removes access to objects (error 404) but keeps the data. There is a **delete marker** added as a new version.
- Transition or expire non-current versions with S3 lifecycle rules
- There will be cost for storing all the versions.
- Must enable bucket versioning for using Bucket replication and Object lock features



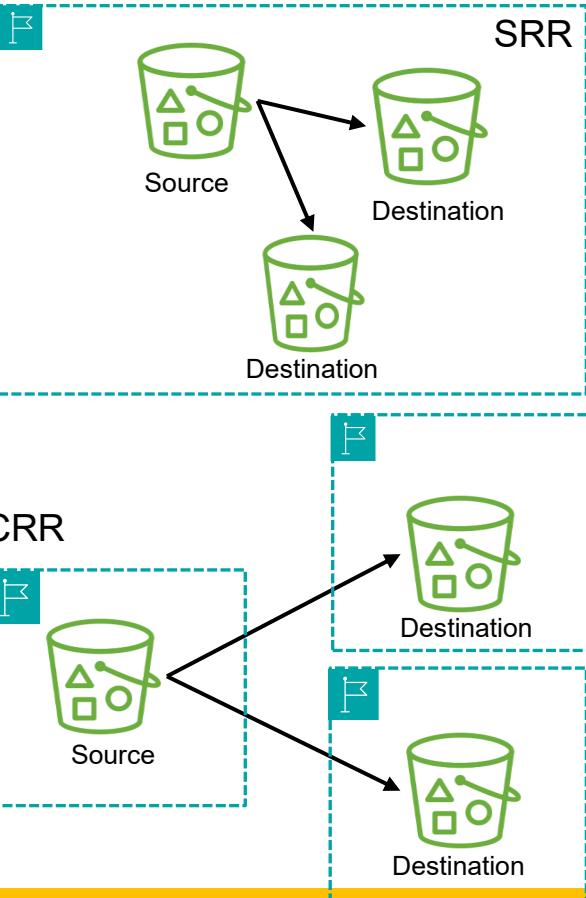
Use versioning to protect your data from accidental deletion or for recovering / rollback to the older versions of the objects

# Exercise: Enable S3 Versioning

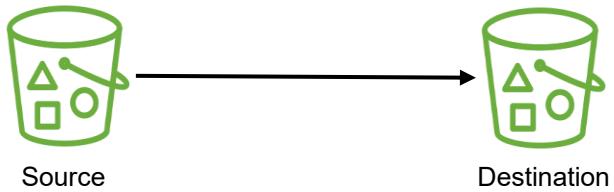
- 1 Create S3 Bucket.
- 2 Go to Bucket -> Properties -> Versioning -> Enable
- 3 Upload sample text file to S3 Bucket.
- 4 Upload modified text file with same name multiple times by doing small changes every time.
- 5 Check Versions. On S3 Console -> Bucket -> Versions -> Show. How many versions do you see?
- 6 Try deleting a particular Version of the uploaded file.
- 7 Click Hide Version. S3 Console -> Bucket -> Versions -> Hide Versions (You should see only one file)
- 8 Delete a File. Deleted?
- 9 Click Show Version. S3 console -> Versions -> Show versions. Restore file using desired version.

# Amazon S3 Replication

- Automatic and asynchronous replication of objects between S3 buckets
- For replication, Bucket versioning must be enabled on both source and destination bucket(s)
- Supports - Same Region Replication (SRR) & Cross Region Replication (CRR)
- Supports - Live Replication & On-demand replication
- Source and Destination buckets can be owned by same AWS account or different Accounts
- S3 Replication use cases:**
  - Aggregate logs into centralized bucket (SRR)
  - Live replication from Production to Test environment (SRR)
  - Data Sovereignty – copies of data in different accounts
  - Compliance – store copy of data at distance (CRR)
  - Minimize latency by bringing data closer to the end user (CRR)



# Exercise - Amazon S3 Replication

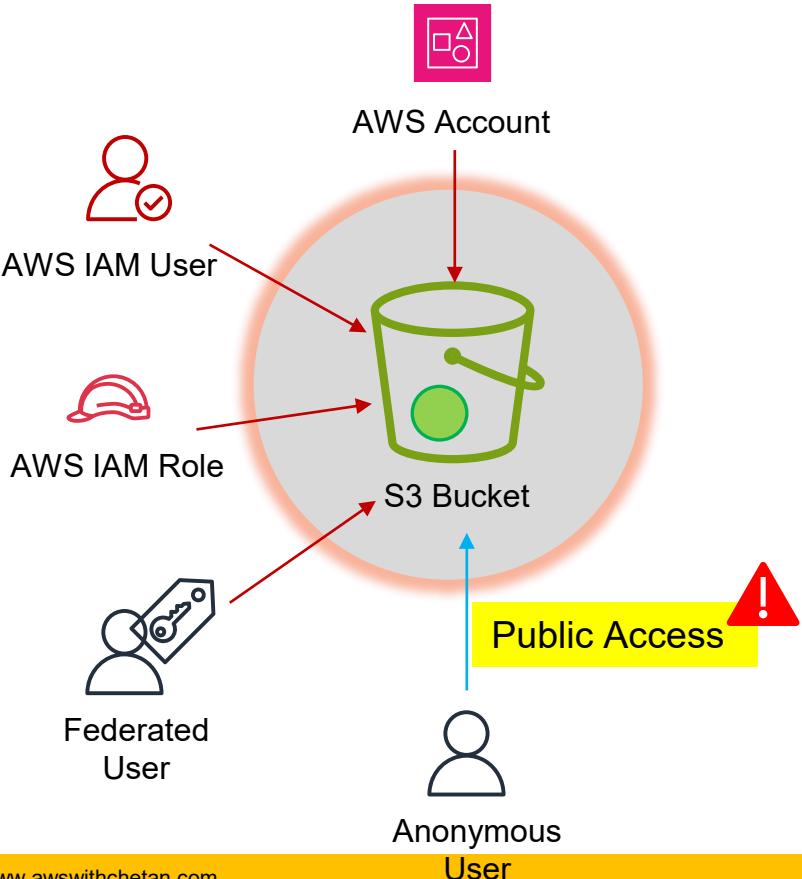


- 1 Create source and destination buckets
- 2 Enable versioning for both the buckets
- 3 In source bucket, go to Management and create Replication rule
- 4 Upload an object into source bucket and wait for few minutes
- 5 Verify in the target bucket if you see the same object with same object version

# IAM Access Analyzer for S3



- Analyses the S3 ACL and Bucket policies to ensure that only required entities have access to S3 buckets
- Identifies Publicly accessible S3 buckets
- Identifies buckets which are shared with other AWS accounts



# IAM Access Analyzer for S3

Screenshot of the AWS IAM Access Analyzer for S3 interface. The left sidebar shows navigation links for Amazon S3, including Buckets, Access Grants, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, and IAM Access Analyzer for S3. A section for Block Public Access settings for this account is also present.

The main content area displays findings under the heading "IAM Access Analyzer for S3". It includes a warning message: "9 buckets are configured to allow access to anyone on the internet or any other AWS users. Review this risky configuration immediately." Below this, there are two tables:

- Buckets with public access (9)**

Bucket name	Discovered by Access ...	Shared through	Status	Access level
<a href="#">3dstreetart.in</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">kvrikish.com</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">venice-with-chetan</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">www.3dstreetart.in</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">www.3dstreetartindia.com</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">www.awscloudtraining.com</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">www.awstrainingcenter.com</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">www.awswithchetan.com</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">www.spacemission.in</a>	3 minutes ago	Bucket policy	Active	Read
- Buckets with access from other AWS accounts - including third party AWS accounts (2)**

Bucket name	Discovered by Access ...	Shared through	Status	Access level
<a href="#">cloudbitz.in</a>	3 minutes ago	Bucket policy	Active	Read
<a href="#">www.cloudbitz.in</a>	3 minutes ago	Bucket policy	Active	Read

User

Anonymous User

# Amazon S3 summary

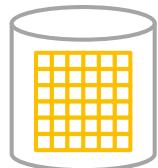
1. S3 is an object storage which is accessible directly over the web (http/https)
2. S3 bucket contains Objects (max size 5TB), unlimited number of objects in the bucket.
3. S3 supports different storage classes for different storage requirements based on latency, frequency of access and durability of the data.
4. S3 storage classes: S3 Standard, S3 Standard-IA, S3 One Zone-IA, S3 Intelligent-Tiering, S3-Glacier\*
5. S3 security - Block Public Access, IAM user policy, Bucket Policy, Bucket/Object ACL,
6. S3 Data encryption using Server-side encryption (SSE-S3, SSE-KMS, SSE-C) and Client-side encryption
7. S3 lifecycle configurations to transition or expire objects after specific duration
8. S3 can host a static website (not required to have server-side processing e.g. PHP, python)
9. S3 supports versioning where multiple versions for objects are stored. Helps to recover objects in case of accidental deletion.
10. S3 Replication - SRR (same-region) or CRR (cross-region). For replication versioning must be enabled.
11. IAM access analyzer can identify public s3 buckets and buckets accessible by other AWS accounts

# AWS Storage services - Summary



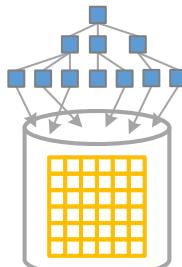
EBS

- **Block Storage** – Data is stored into unique blocks
- Host File System places data on Disk using protocols like iSCSI
- Must be attached to EC2
- High performance, high IOPS, low latency



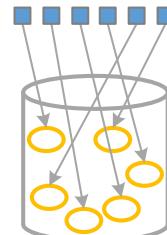
EFS/FSx

- **Shared file system** – Hierarchical structure
- Should be mounted on EC2 or on-premises servers over NFS, SMB protocols
- High throughput, moderate performance

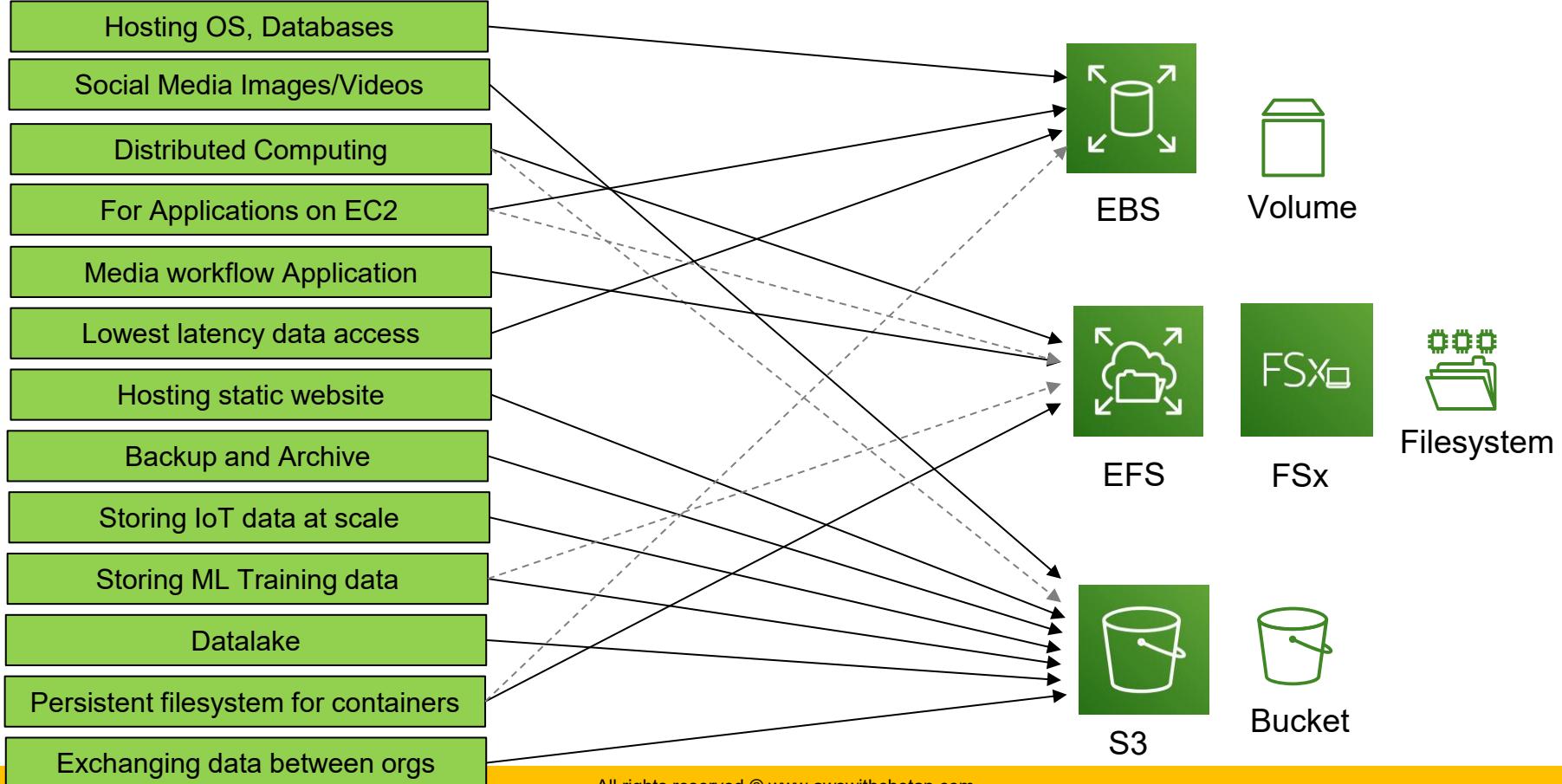


S3

- **Object storage** – Flat structure
- API access to data (HTTPS)
- Metadata driven (Attributes, Policy)
- High Throughput, unlimited storage, moderate performance



# AWS Storage services - When to use what?

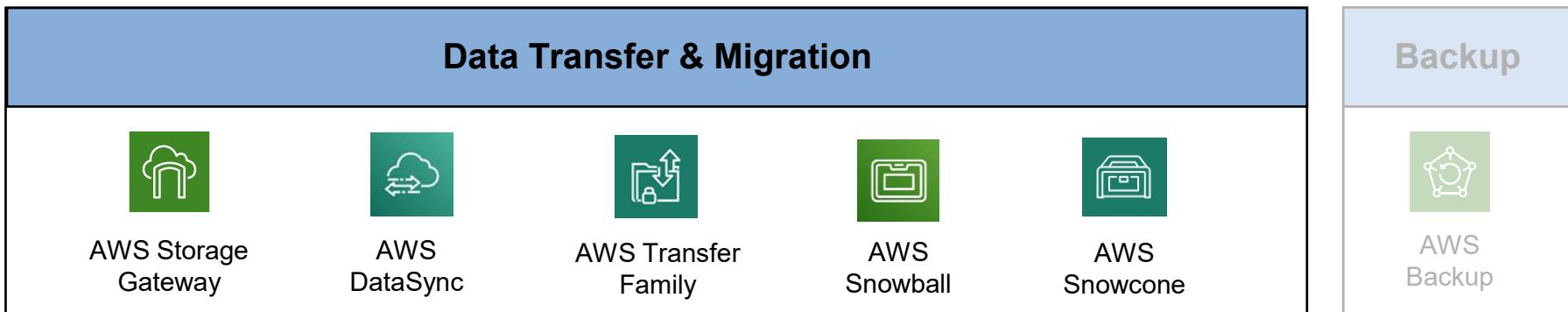
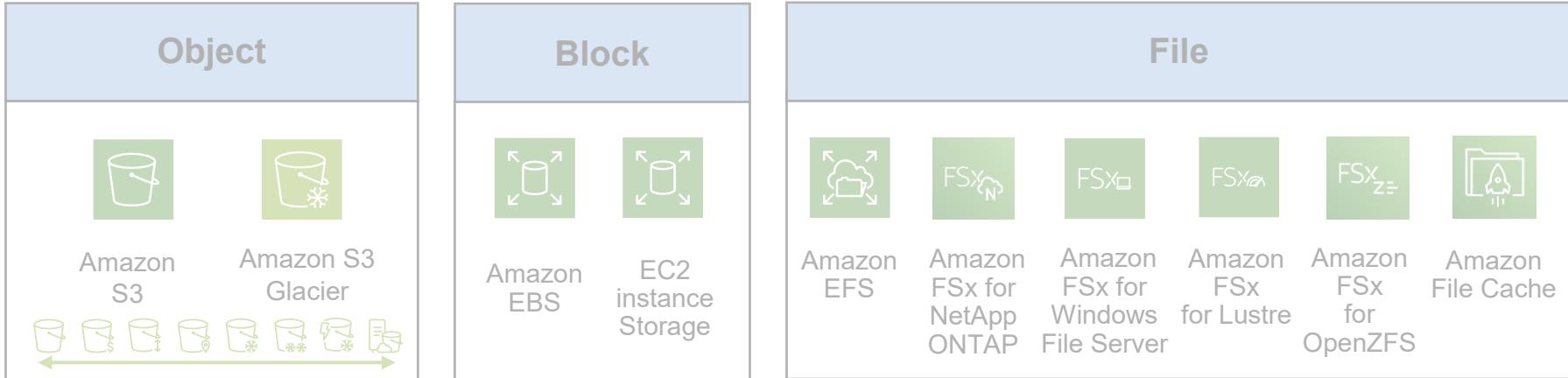




# Data Migration and Hybrid storage

AWS Snow Family, AWS DataSync and AWS Storage Gateway

# AWS Storage services



# Migrating data to AWS



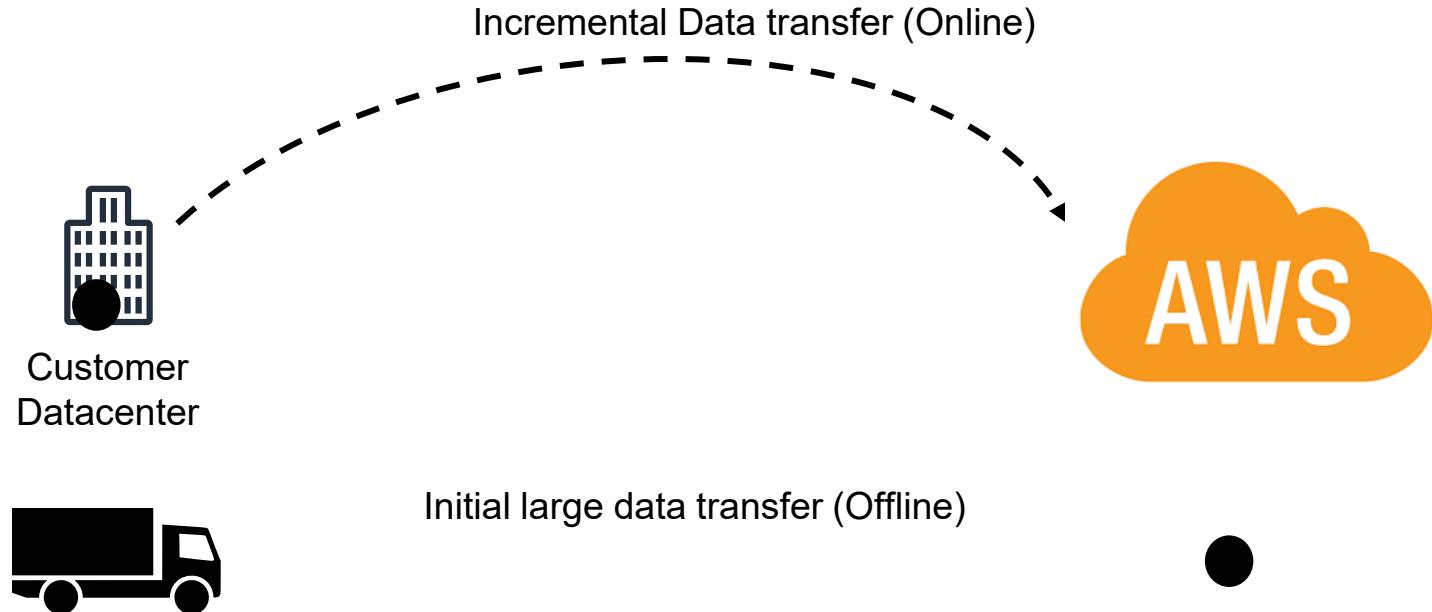
# How much time does it take to move data online?

Bandwidth ->

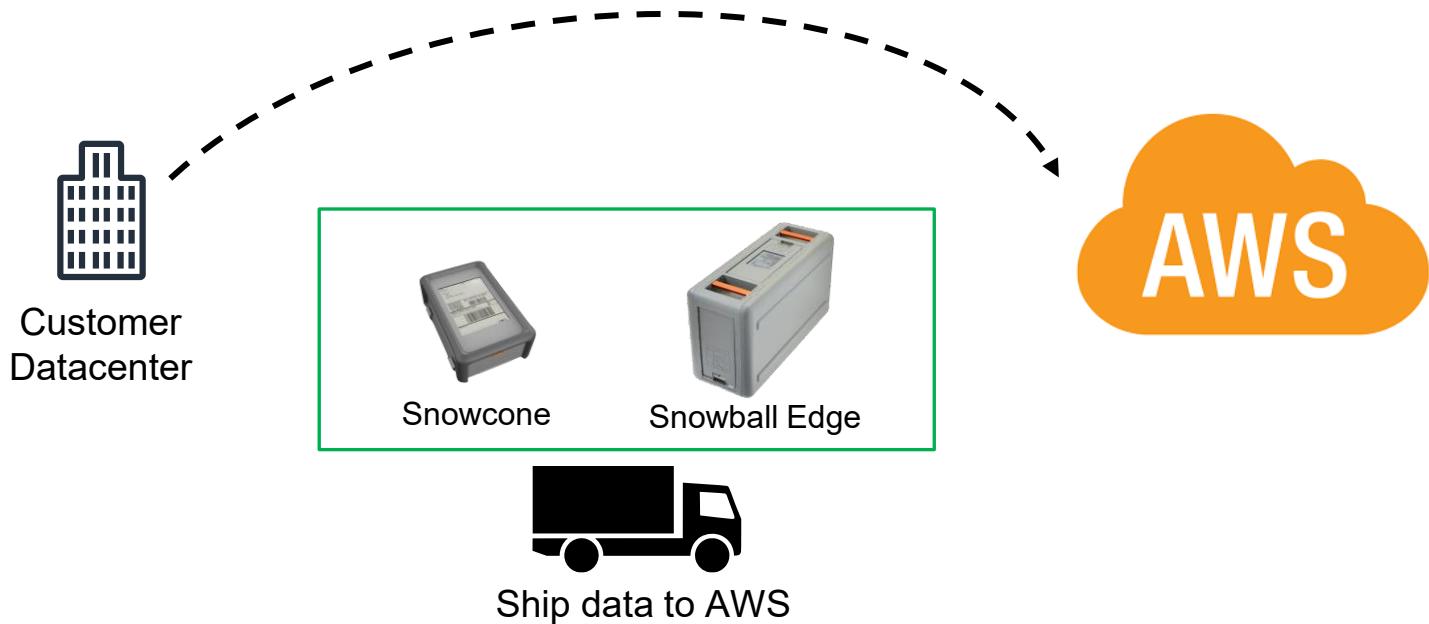
Data size ->

	1 Gbps	2 Gbps	5 Gbps	10 Gbps
500 TB	58 days	29 days	12 days	6 days
5 PB	2 years	1 year	116 days	58 days
10 PB	4 years	2 years	232 days	116 days

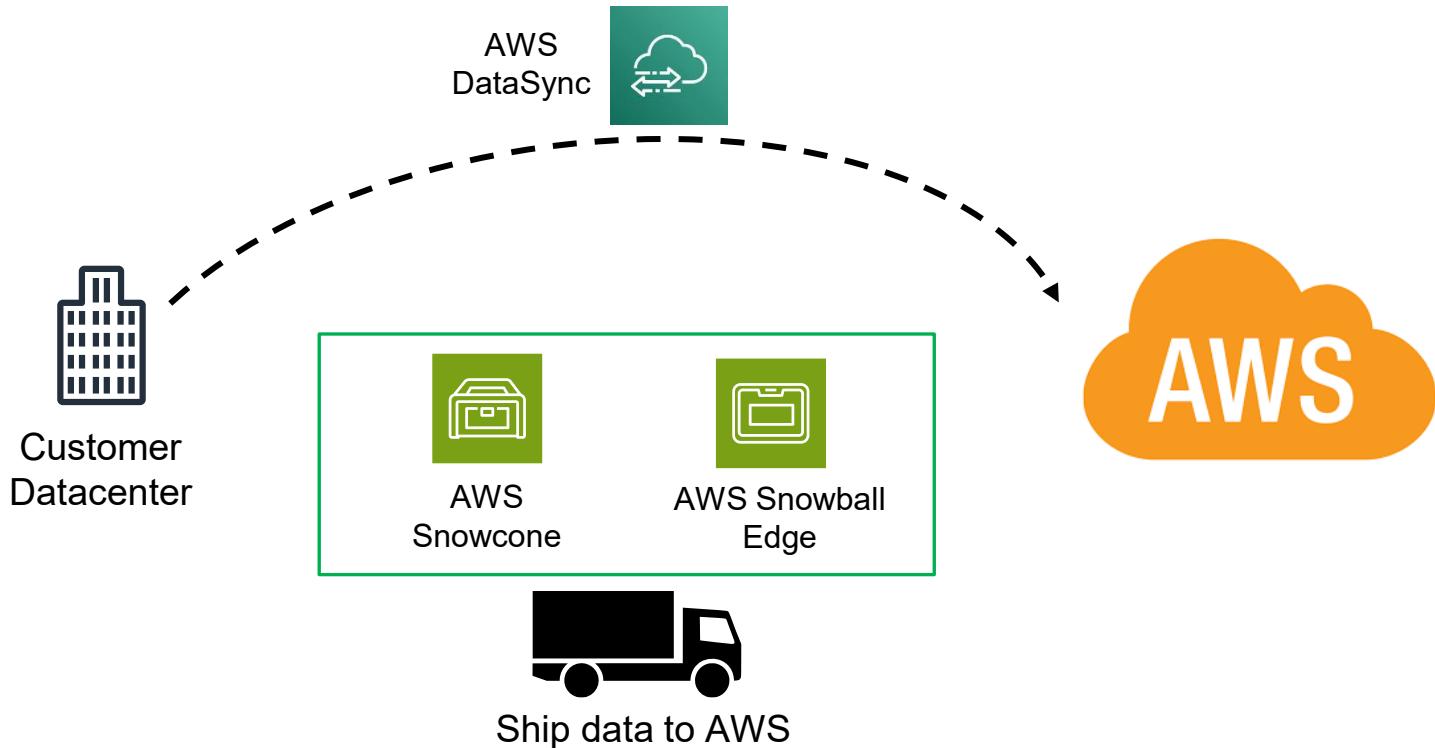
# Migrating data to AWS



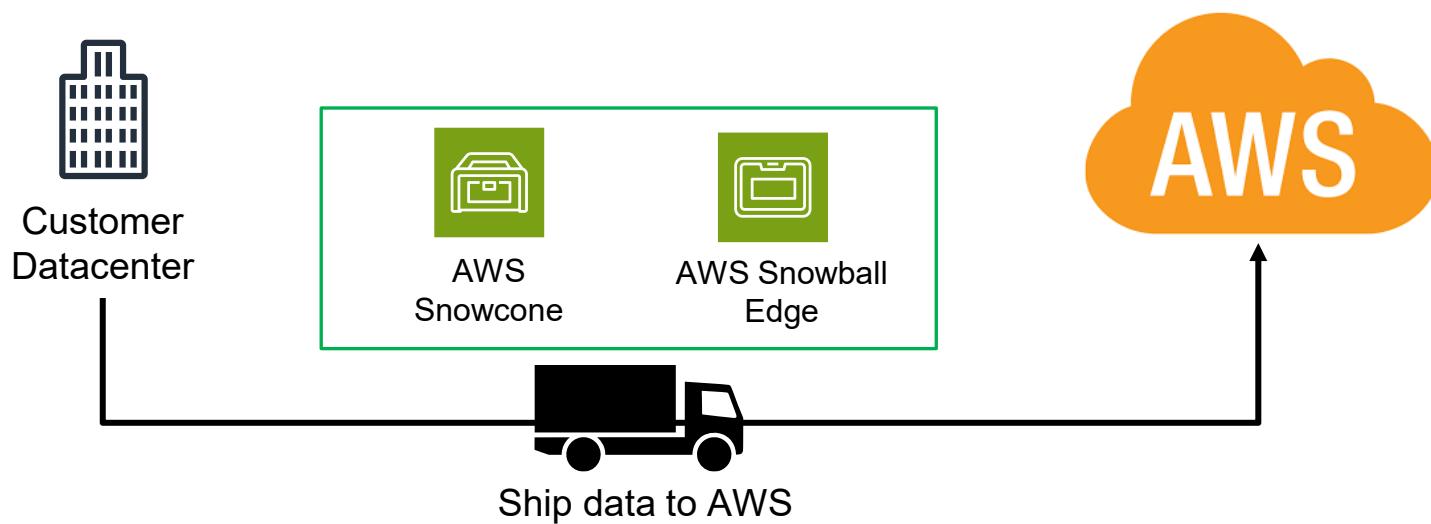
# Migrating data to AWS



# Migrating data to AWS



# Offline data transfer with AWS Snow devices





# AWS Snowcone

Small, portable, rugged, and secure edge computing and data transfer device

- Encryption, tamper-evident
- 4.5 pounds (2.1 kg)
- Portable
- Withstands harsh environments
- 8 TB HDD or 14 TB SSD storage
- 2 CPUs, 4 GB of memory
- Wi-Fi or Ethernet data copy
- AWS DataSync agent pre-installed for online data transfer



## Use cases

Industrial IoT – sensor or machine data in a factory, Content distribution and aggregation, data migration



# AWS Snowball Edge

Snowball device with on-board storage and compute power

- 49.7 pounds (22.54 Kg)
- 2 x 10 Gbit, 1 x 40 Gbit, 1 x 100 G-bit Network Interfaces
- Device Options - Storage optimized / Compute optimized
- Up to 104 vCPU, 416GB RAM
- Up to 80TB HDD and 210TB SSD



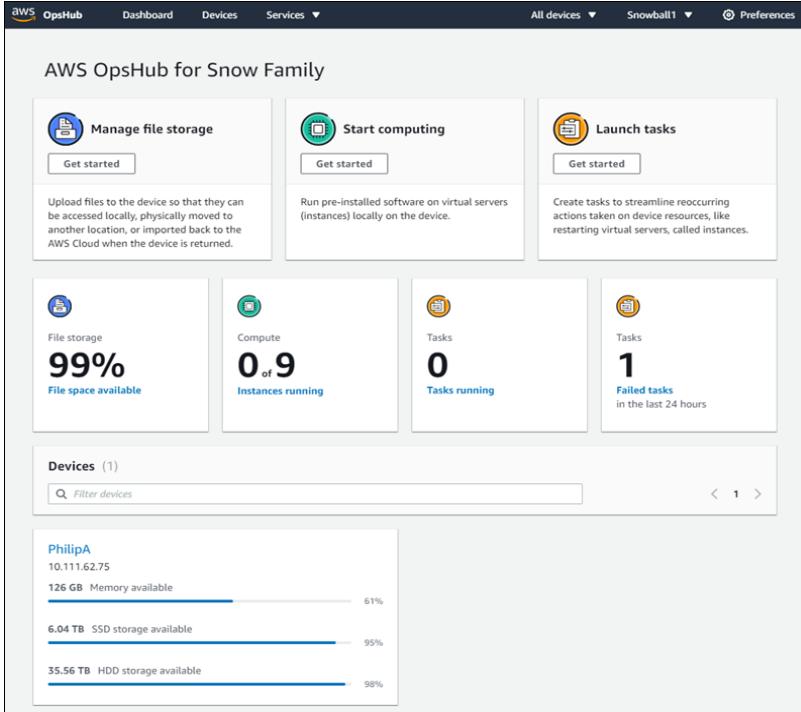
## Use cases

- Storage optimized – Large scale data migration
- Compute optimized – Edge (on-premises) data processing, machine learning, full motion video analytics,

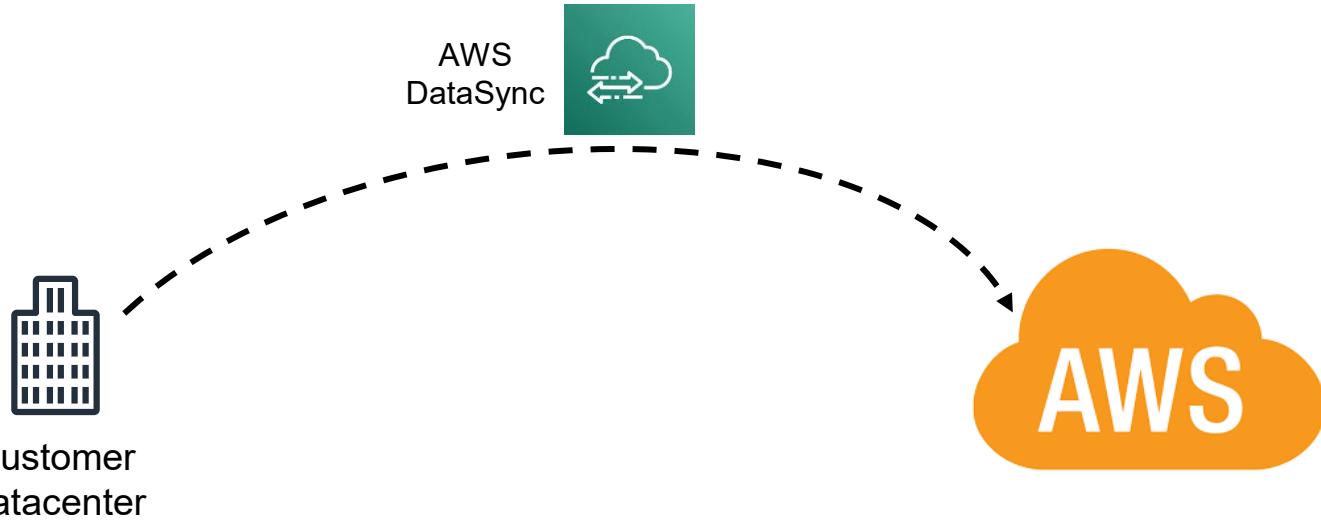
# AWS OpsHub

A new user-friendly tool to manage the AWS Snow family devices.

- GUI based, just download and install AWS OpsHub on your local machine
- Unlocking the devices
- Transferring files
- Launching EC2 instances on snow devices
- Dashboard that summarizes key metrics such as storage capacity and active instances on the snow device

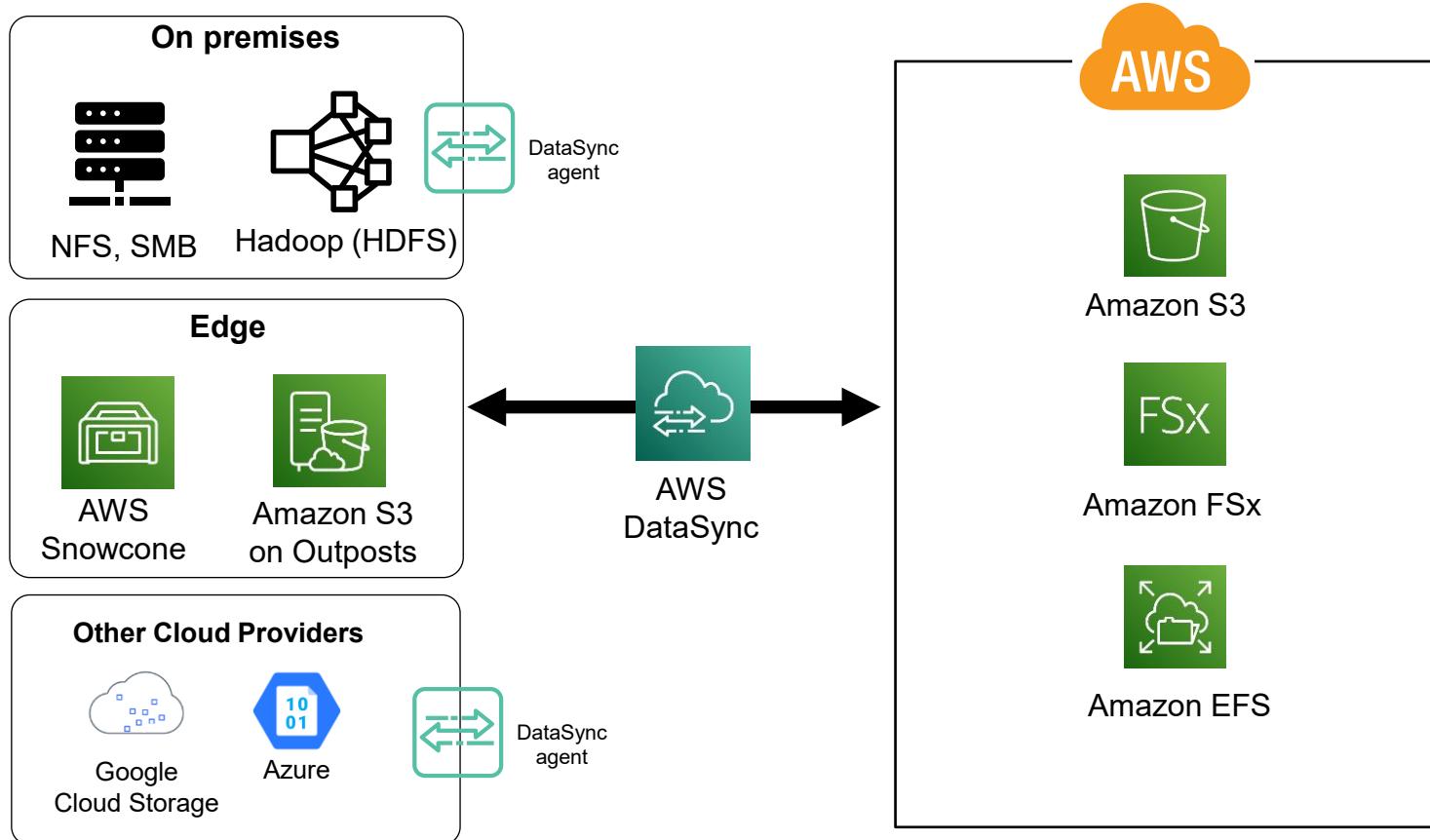


# Online data transfer with AWS DataSync



Customer  
Datacenter

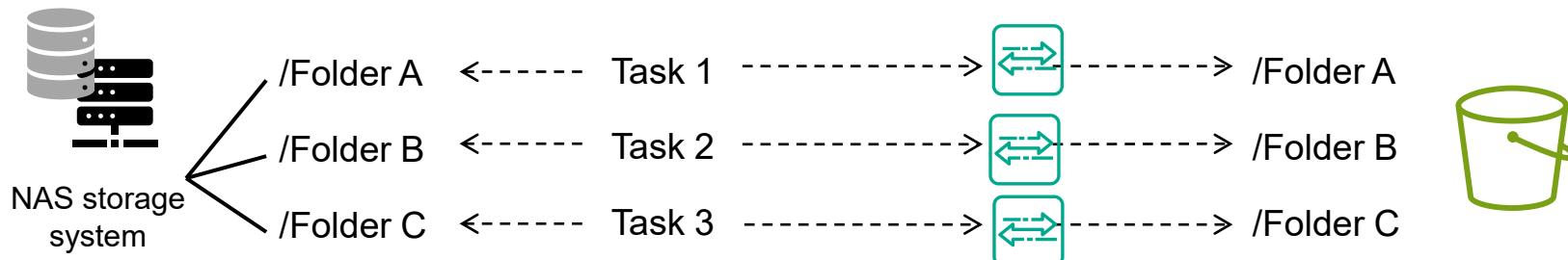
# AWS DataSync





# AWS DataSync

- Moves data between on premises and AWS Storage services
- Copies data between Network File System share (NFS), Server Message Block share (SMB), Hadoop distributed file system (HDFS), Self-managed object storage, AWS Snowcone and AWS storage services such as S3, EFS and FSx.
- Data transfer tasks can be scheduled – Hourly, daily, weekly etc.
- Data is copied incrementally.
- Handles failure (if any) during the transmission.
- You can set limit on the bandwidth usage by DataSync. Single task can utilize up to 10 Gbps.
- Supports TLS encryption.
- Can run multiple tasks in parallel

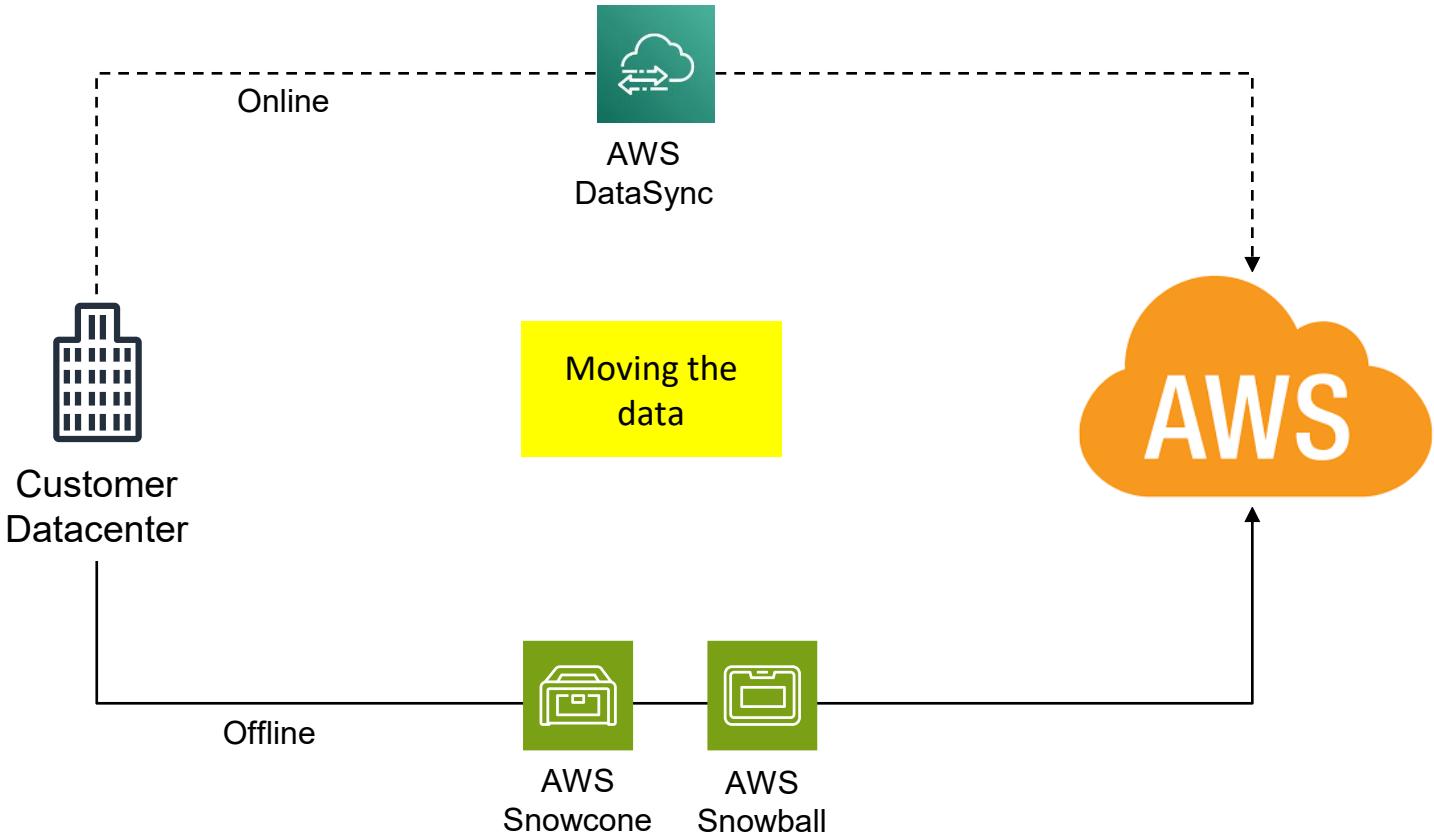




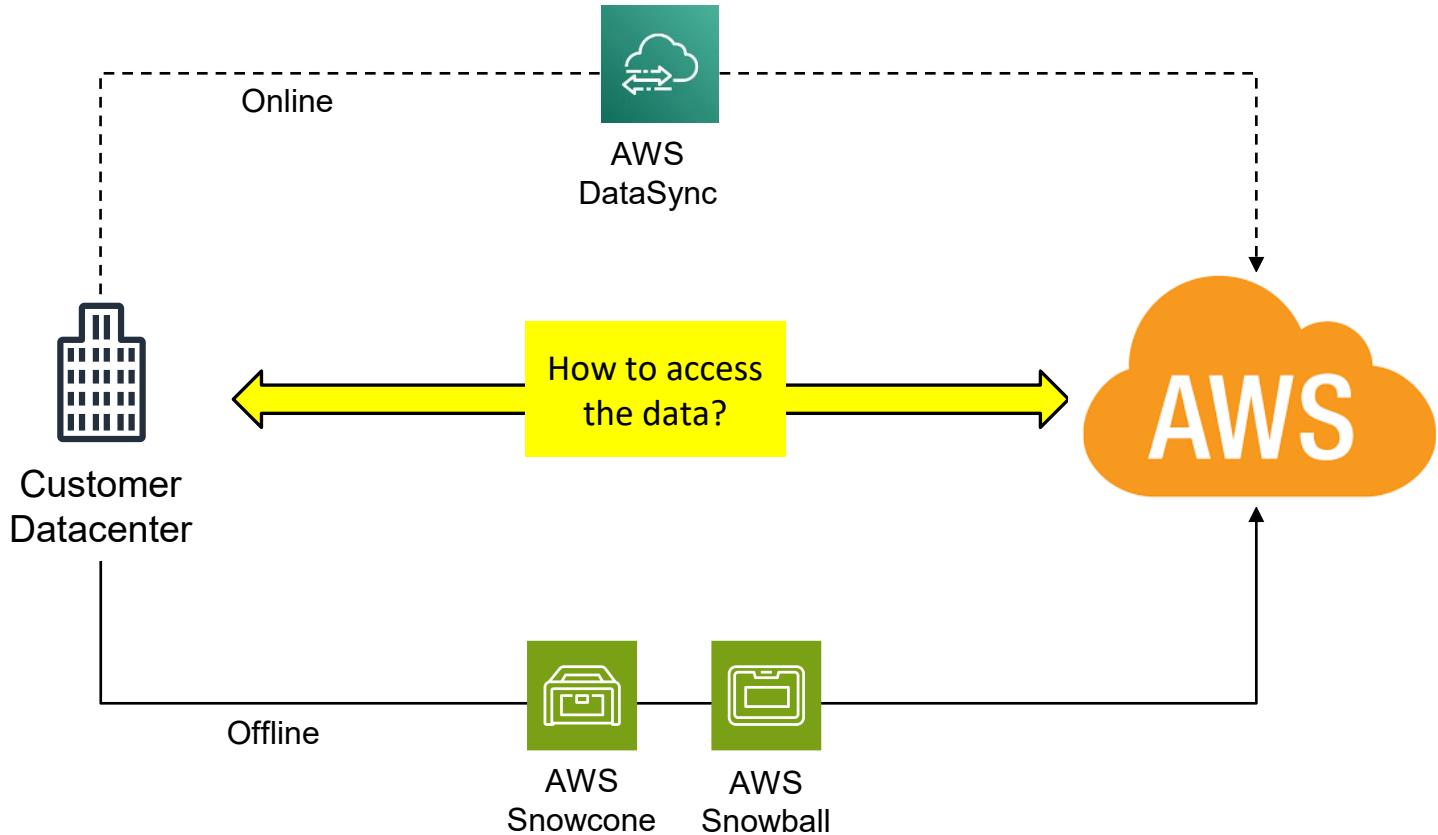
# AWS Storage Gateway

For Hybrid storage

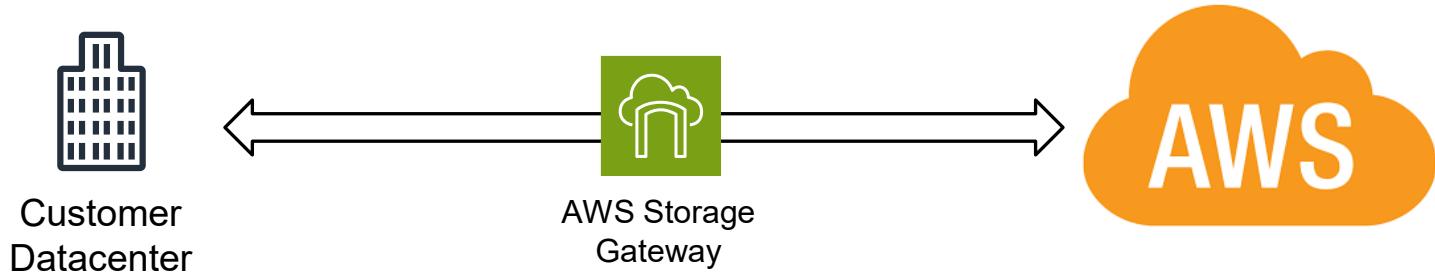
# Access data in AWS from on-premises



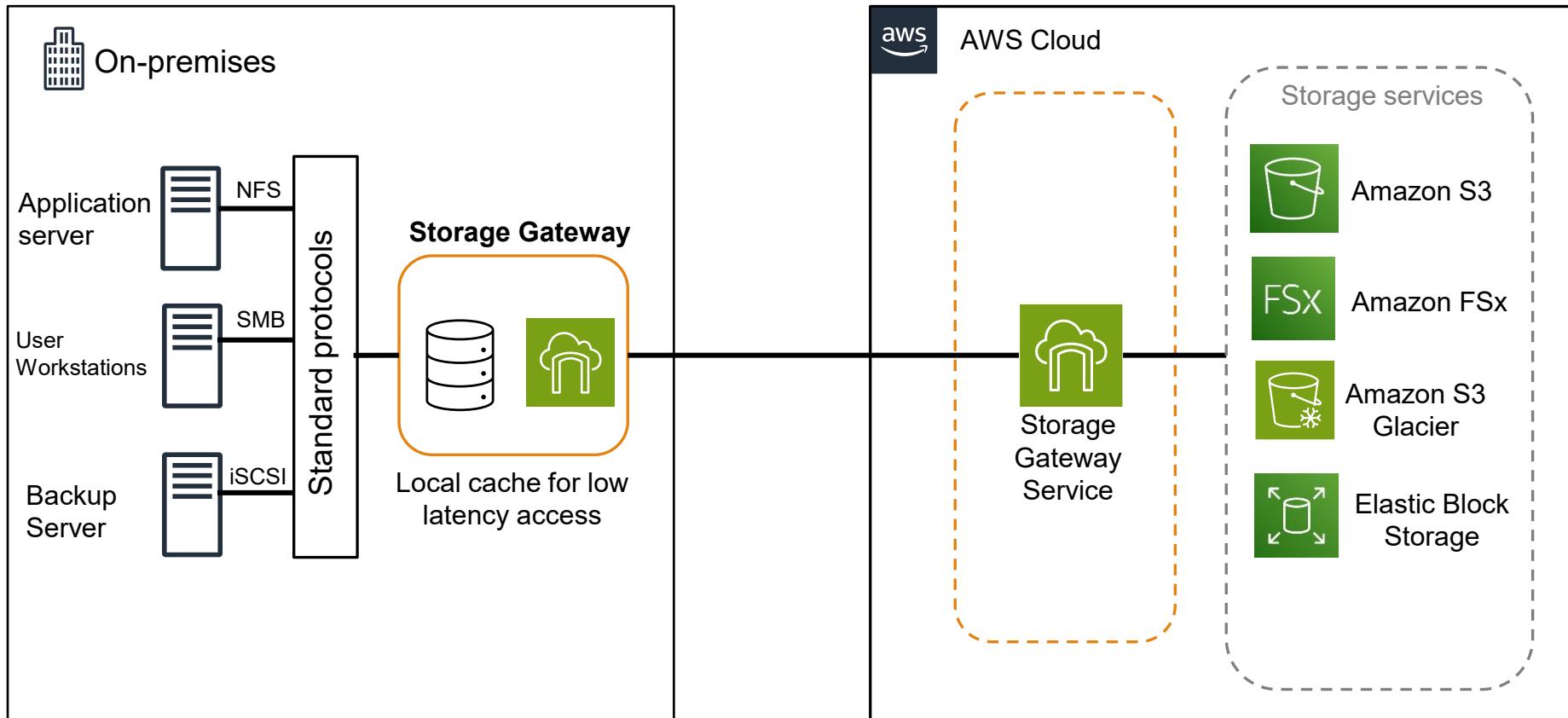
# Access data in AWS from on-premises



# Access data in AWS from on-premises



# Access AWS storage from on-premises





# AWS Storage Gateway

- AWS Storage Gateway connects on-premises storage to AWS cloud storage services, enabling hybrid cloud storage for backup, archiving, and disaster recovery.
- It provides low-latency performance by caching frequently accessed data on premises
- AWS Storage gateway appliance can be deployed on-premises on Virtual Machines or on EC2 instance in AWS
- Supports storage protocols like NFS, SMB, iSCSI
- There are following different storage gateway types:
  - **Amazon S3 File Gateway** - Access to S3 (Data lake, archive images/videos, database backups to S3)
  - **Amazon FSx File Gateway** – Access to FSx for Windows Server (File backups)
  - **Tape Gateway** - Replace physical tapes with virtual tapes in AWS (Tape backup)
  - **Volume Gateway** – Access to EBS volumes (On-premises cache, backup)



S3 File Gateway



FSx File Gateway



Tape Gateway



Volume Gateway

# Data migration and Hybrid storage - summary

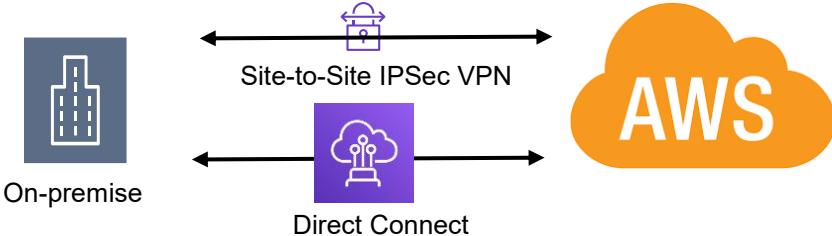
- AWS Snow family for **offline data migration** to AWS and Edge computing.
- AWS Snowcone is a small, portable, rugged, and secure edge computing and data transfer device
- AWS Snowball edge comes with on-board storage and compute power which makes it suitable for both data transfer (Storage optimized) and edge computing/analytics (Compute optimized)
- AWS DataSync for **online data migration** to AWS.
- AWS DataSync can be used with on-premises NFS file share, AWS Snowcone and other cloud vendor storage to transfer data to Amazon S3, EFS/FSx.
- AWS Storage gateway for **hybrid storage** where on-premises applications **can access and backup data in AWS**

# AWS services for hybrid cloud

## Hybrid Networking

- AWS Site-to-Site VPN
- AWS Direct Connect

Additionally - AWS Route53, Transit Gateway etc.



## Hybrid Compute

- AWS Outpost
- AWS Snow devices

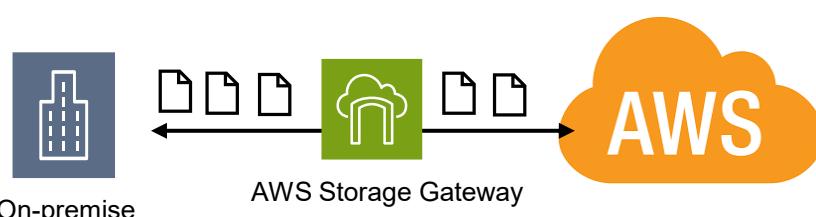
Additionally - EKS Anywhere, ECS Anywhere



## Hybrid Storage

- AWS Storage Gateway

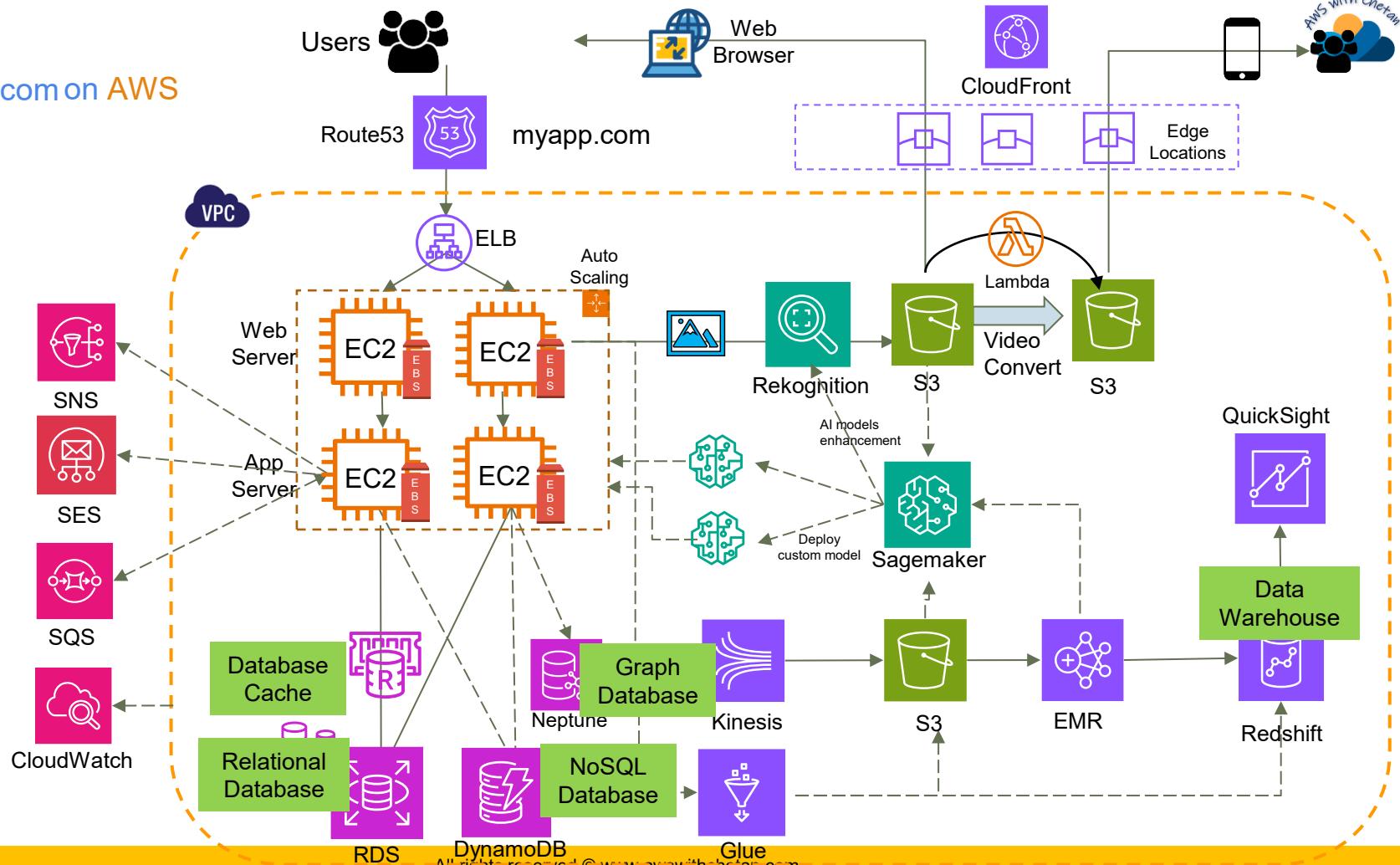
Additionally – DMS for database migration, Snow devices for offline data migration, DataSync for online data migration





# AWS Database services

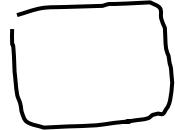
# myapp.com on AWS



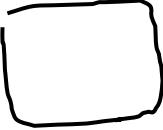
# Evolution of applications and databases



Mainframes



Client-Server



3-Tier



Microservices



# Today's need

What we used traditionally?



What is needed today?

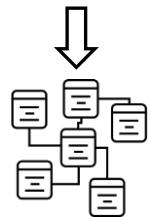


A Relational Database

Purpose built databases

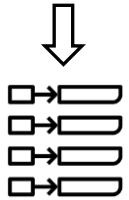
# AWS Database services

Banking,  
Finance,  
Bookings,  
ERP, CRM



Relational Database

Shopping cart,  
Product catalog,  
Customer  
attributes



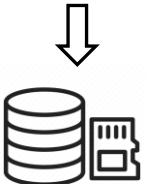
Key-value or  
NoSQL  
Database

Content  
management,  
personalization,  
mobile



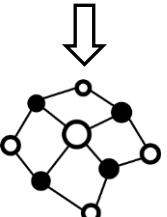
Document  
Database

Leaderboards,  
real-time  
analytics,  
caching



In-memory  
Database

Fraud detection,  
social  
networking,  
recommendation  
engine



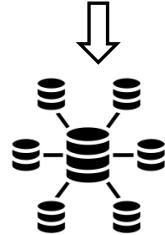
Graph  
Database

IoT  
applications,  
event tracking



Timeseries  
Database

Analytics,  
Data Marts



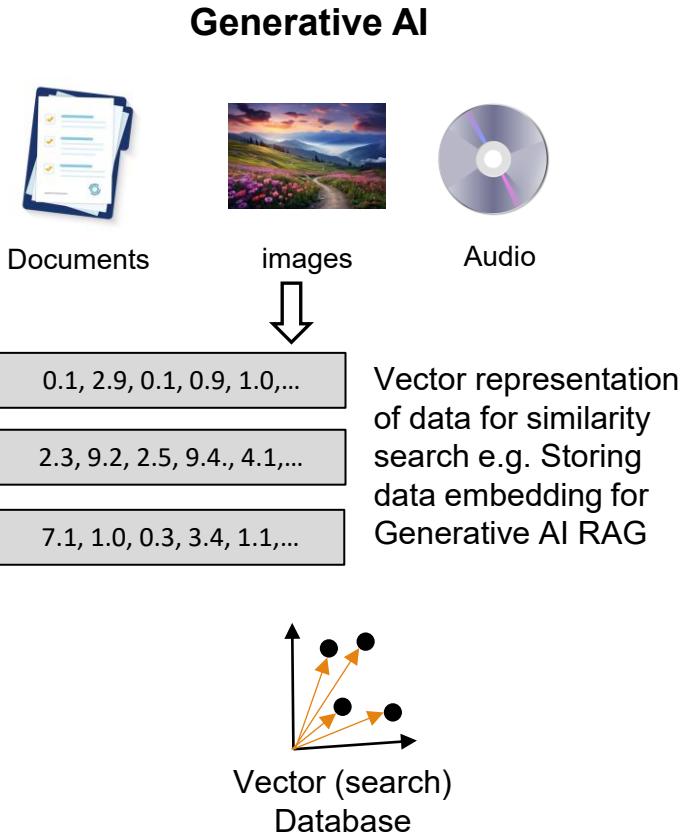
Data  
Warehouse

# AWS Database services

Systems of record,  
Supply chain,  
health care,  
financial

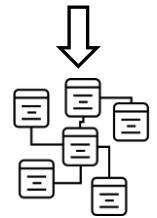


Ledger  
Database



# AWS Database services

Banking,  
Finance,  
Bookings,  
ERP, CRM

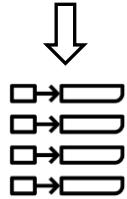


Relational  
Database



Amazon  
RDS      Amazon  
Aurora

Shopping cart,  
Product catalog,  
Customer  
attributes



Key-value or  
NoSQL  
Database



Amazon  
DynamoDB

Content  
management,  
personalization,  
mobile

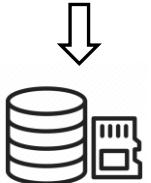


Document  
Database



Amazon  
DocumentDB

Leaderboards,  
real-time  
analytics,  
caching

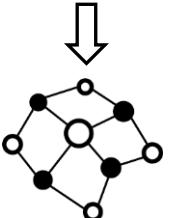


In-memory  
Database



Amazon  
ElastiCache

Fraud detection,  
social  
networking,  
recommendation  
engine



Graph  
Database



Amazon  
Neptune

IoT  
applications,  
event tracking

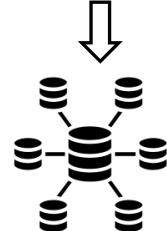


Timeseries  
Database



Amazon  
Timestream

Analytics,  
Data Marts



Data  
Warehouse



Amazon  
Redshift

# AWS Database services

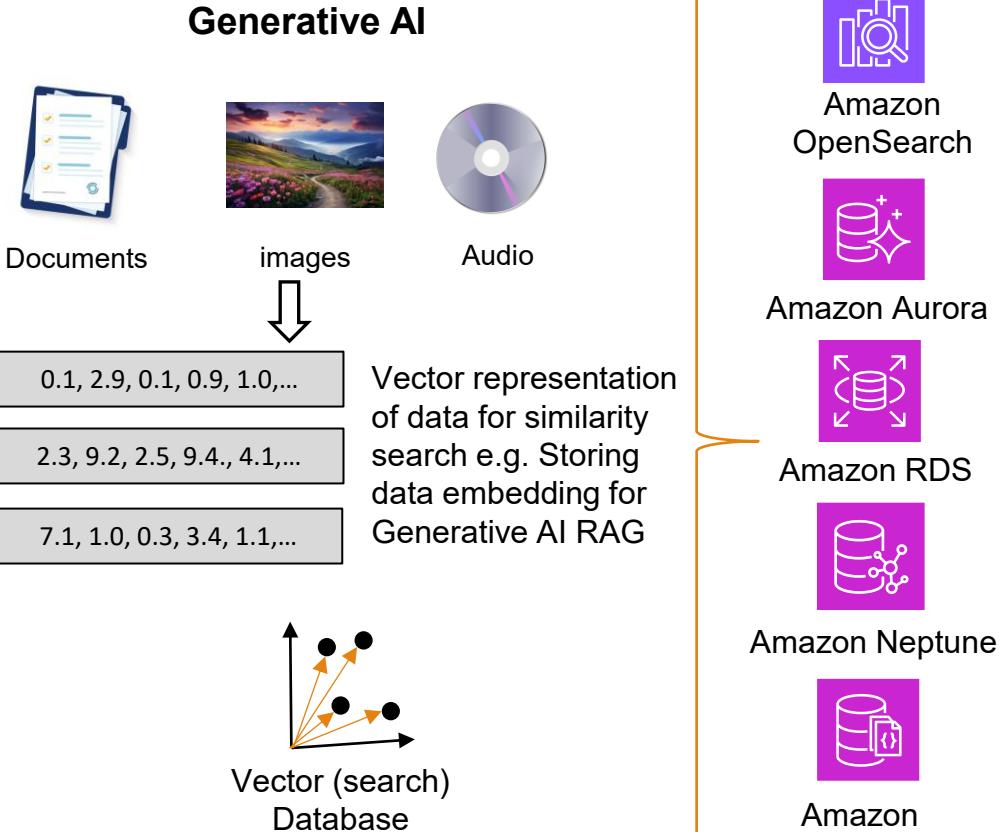
Systems of record,  
Supply chain,  
health care,  
financial



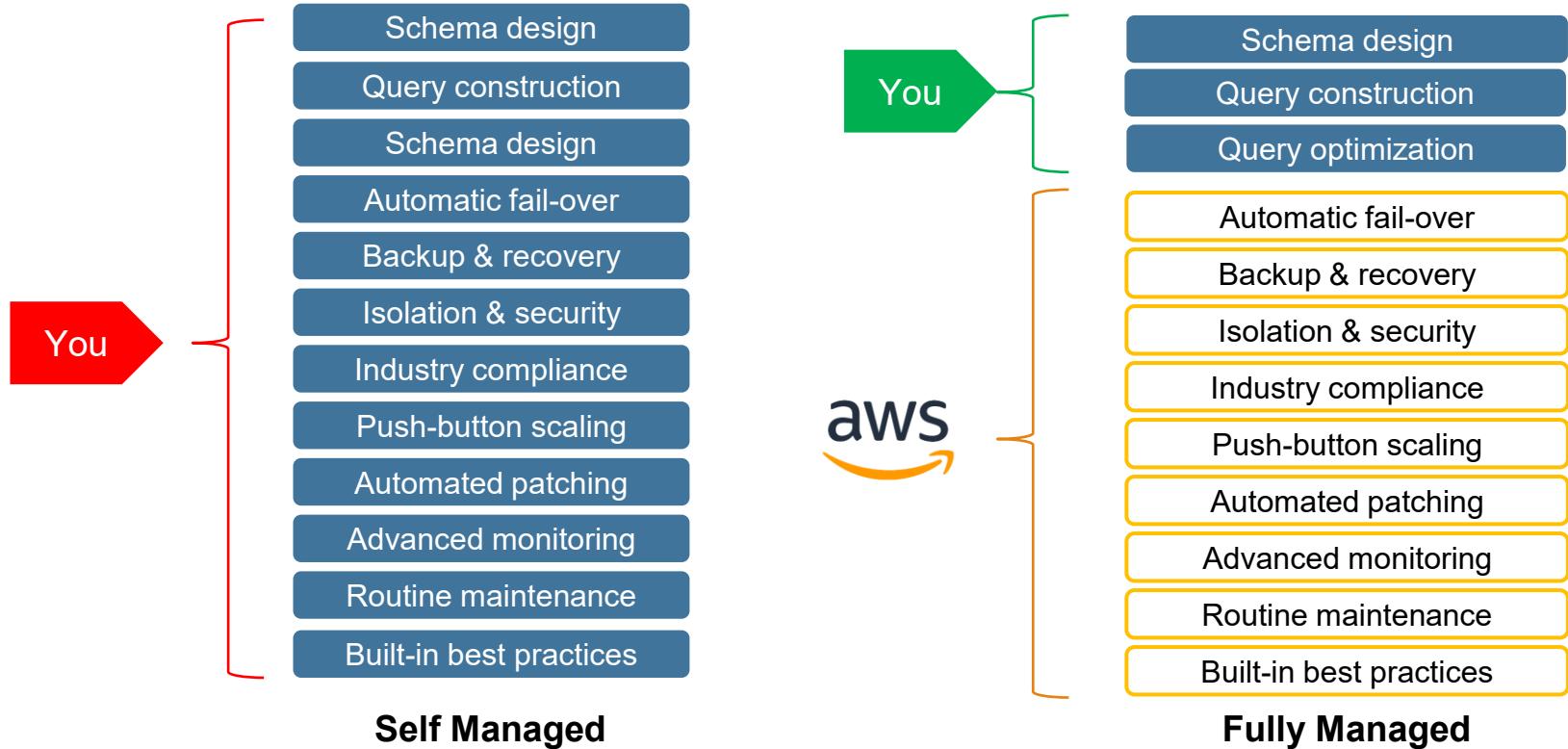
Ledger  
Database



Amazon Quantum  
Ledger Database  
(Amazon QLDB)



# Why managed database service?

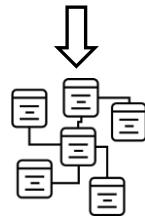




# Amazon RDS

# AWS Database services

Banking,  
Finance,  
Bookings,  
ERP, CRM

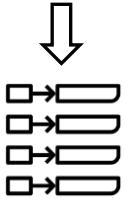


Relational  
Database



Amazon  
RDS      Amazon  
Aurora

Shopping cart,  
Product catalog,  
Customer  
attributes



Key-value or  
NoSQL  
Database



Amazon  
DynamoDB

Content  
management,  
personalization,  
mobile

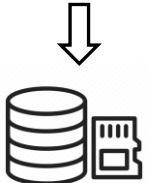


Document  
Database



Amazon  
DocumentDB

Leaderboards,  
real-time  
analytics,  
caching

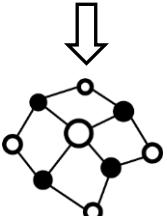


In-memory  
Database



Amazon  
ElastiCache

Fraud detection,  
social  
networking,  
recommendation  
engine



Graph  
Database



Amazon  
Neptune

IoT  
applications,  
event tracking

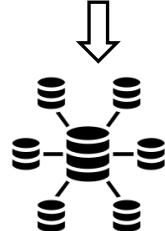


Timeseries  
Database



Amazon  
Timestream

Analytics,  
Data Marts



Data  
Warehouse



Amazon  
Redshift

# Amazon RDS

- Amazon **Relational Database Service**
- A managed SQL database service where AWS handles provisioning, patching, upgrades, backup, recovery, repair, monitoring etc.
- RDS supports following database engines:

AWS native DB



Amazon Aurora

Open-source DBs



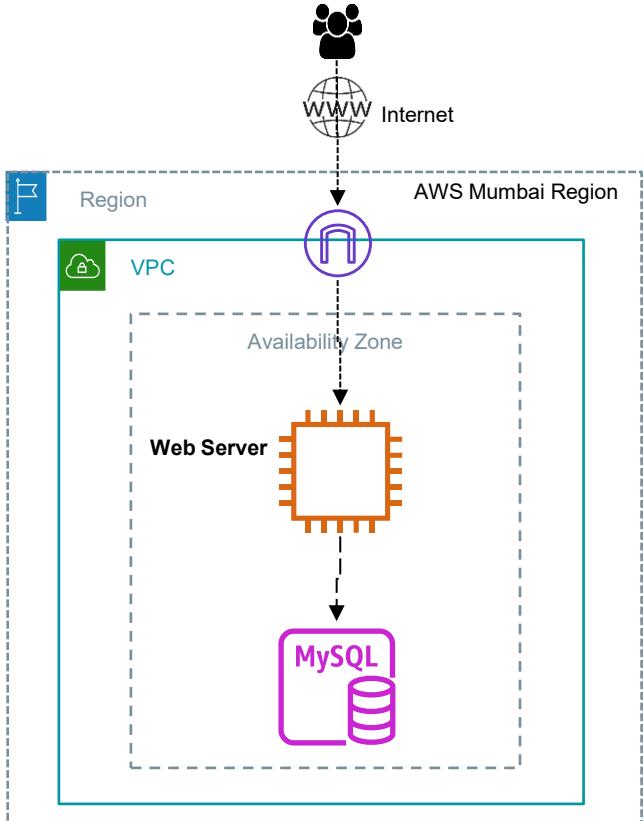
PostgreSQL



Commercial DB



# Demo: Launch RDS DB and connect



## High level steps

- 1 Launch an EC2 instance in the default VPC and connect over SSH
- 2 Create MySQL RDS database in the default VPC (No public access). Security group to allow traffic from EC2.
- 3 Install mysql client on EC2
- 4 Connect to RDS DB using database endpoint

# How to connect to Mysql RDS?

1. Install mysql client in the ec2 instance

```
# install pip (Amazon Linux 2023 does not have one by default)
$dnf install -y pip

# install dependencies
$dnf install -y mariadb105-devel gcc python3-devel

# install mysqlclient
$pip install mariadb105
```

2. Connect to Database and query the data

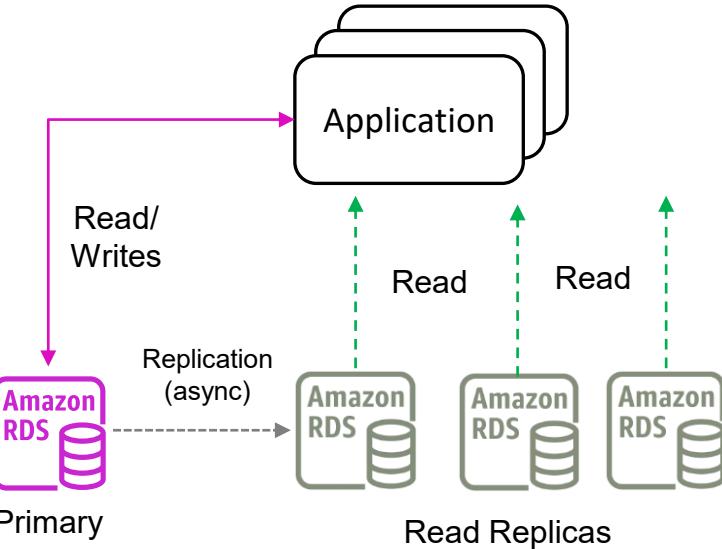
```
$mysql -h <database endpoint> -u admin -p
$MySQL [(none)]> create database awswithchetan;
$MySQL [(none)]> use awswithchetan;
```

3. Create table and add data.

```
MySQL [corp]> create table students (emp_id int, name varchar(64), department varchar(32), location varchar(64));
MySQL [corp]> insert into students values (1001, 'Chetan Agrawal', 'IT', 'Pune');
Query OK, 1 row affected (0.003 sec)
```

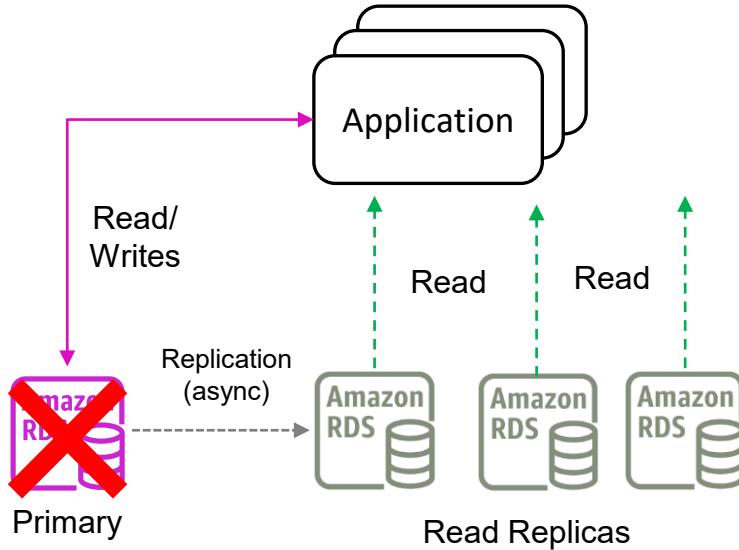
# RDS read replicas

- Create Read Replicas to scale the Read queries thereby relieving pressure on Primary DB
- Data is only written to the Primary DB and can be read from any of the read replicas
- Create up to 15 Read Replicas in the **same or different AWS region**
- Data is replicated asynchronously
- Promote Read replica to standalone DB in case of failure of Primary DB instance.



# RDS read replicas

- Create Read Replicas to scale the Read queries thereby relieving pressure on Primary DB
- Data is only written to the Primary DB and can be read from any of the read replicas
- Create up to 15 Read Replicas in the **same or different AWS region**
- Data is replicated asynchronously
- Promote Read replica to standalone DB in case of failure of Primary DB instance.

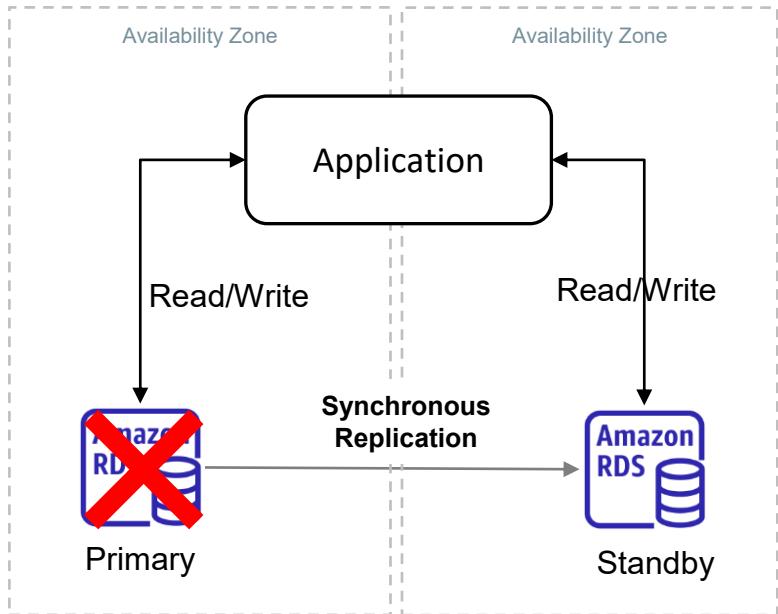


## Use cases:

1. Business reporting and data warehousing where queries can run against a read replica, instead of production DB instance.
2. Implementing disaster recovery.
3. Read data with low latency from the local region even if Primary DB is in another region.

# RDS Multi-AZ deployment

- Primary DB instance in one AZ and Standby DB instance(s) in another AZ
- Application reads from and writes to only Primary DB.
- Data is synchronously replicated from Primary DB to Standby DB.
- In case of Primary DB failure or AZ failure, Standby DB is made Primary and the DB endpoint points to Standby.
- Applications automatically gets redirected to standby for read/write queries.



## Use cases:

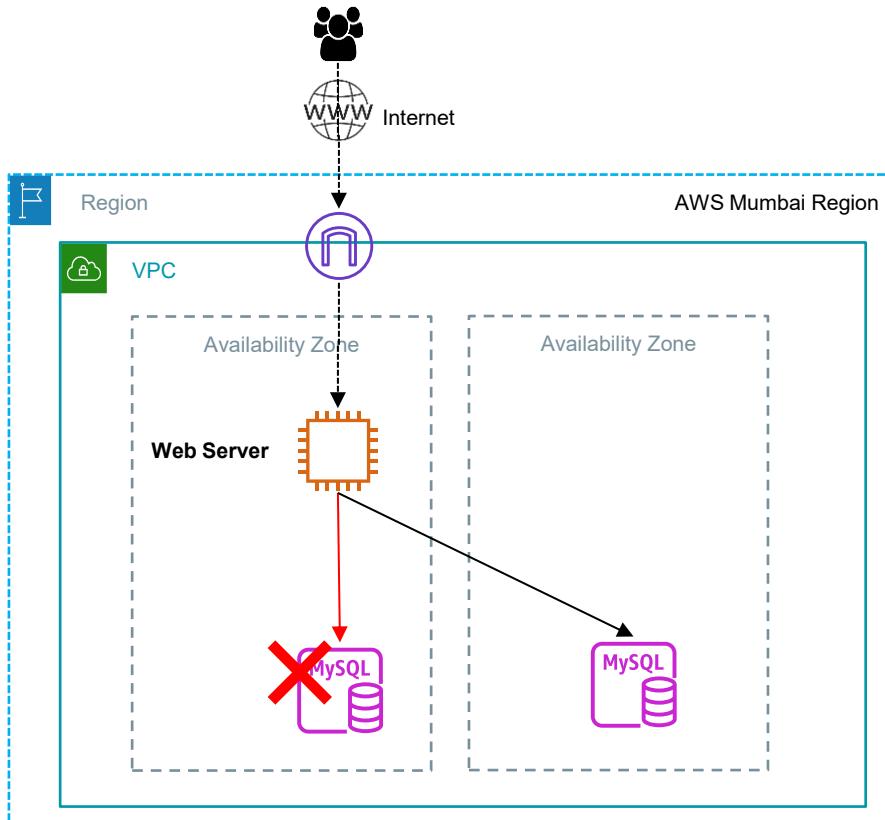
- High availability / automatic failover

# Amazon RDS – important to know

- RDS supports Automatic continuous backups and on-demand or scheduled snapshots.
- With continuous backups we can do Point in time restore (specific time with 1 sec granularity up to maximum 35 days)
- RDS snapshot are stored in S3. We can share the snapshots across AWS accounts or copy across AWS regions to create a new database.
- Provides monitoring dashboards to monitor database performance metrics such as Database connections, CPU Utilization, Free storage space, disk I/O etc.
- There is a 30 mins maintenance windows for upgrades which can be chosen by the customer.

Remember – You do not get access to underlying EC2 instances which host RDS

# Demo: RDS Multi-AZ failover



## High level steps

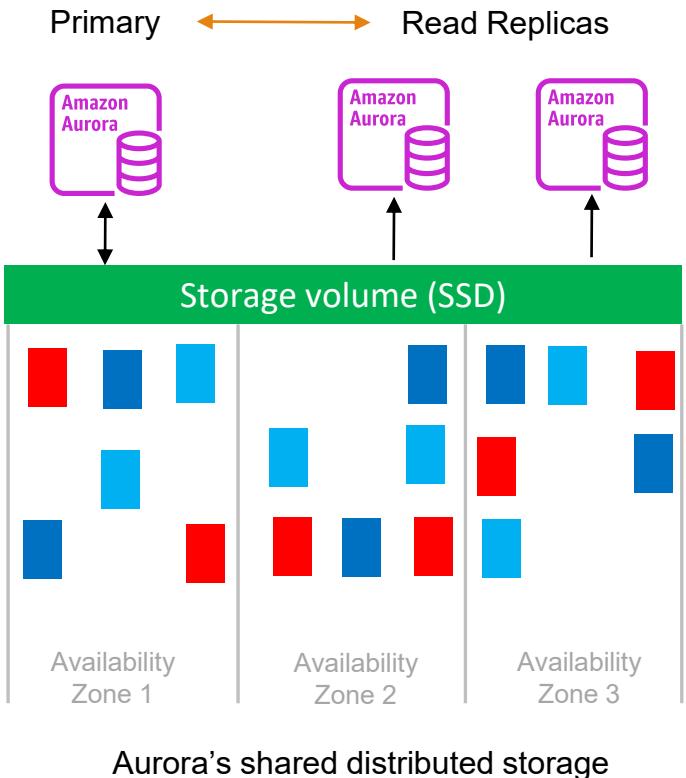
- 1 Launch an EC2 instance in the default VPC and connect over SSH
- 2 Create MySQL Multi-AZ RDS database in the default VPC.
- 3 Install mysql client on EC2
- 4 Connect to RDS DB using database endpoint
- 5 Simulate failure in Primary DB



# Amazon Aurora

# Amazon Aurora

- Aurora is **PostgreSQL** and **MySQL** compatible relational database engines built by AWS.
- Aurora claims to provide 5x the throughput of MySQL and 3x of PostgreSQL DB and 1/10<sup>th</sup> cost of commercial databases.
- Aurora maintains 6 copies of data across 3 Availability Zones.
- Aurora supports massive read scaling with up to 15 read replicas.
- Amazon Aurora key features:
  - Aurora Global Database
  - Aurora Serverless



# Amazon Aurora Global Database

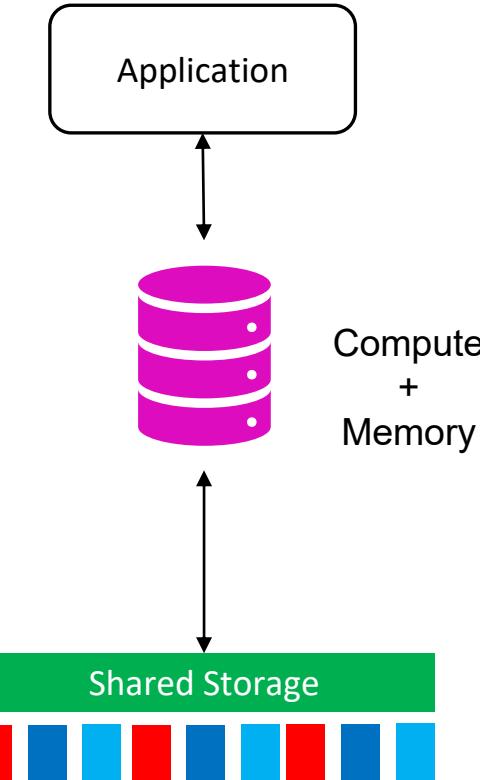
- Aurora global database spans across multiple regions
- Data is written to the primary DB in the primary region and is replicated to other regions (within 1 sec)

## Use cases:

- Bring data close to your customer's applications in **different regions**
- Promote read-replica to primary for faster recovery **in the event of disaster**

# Aurora Serverless

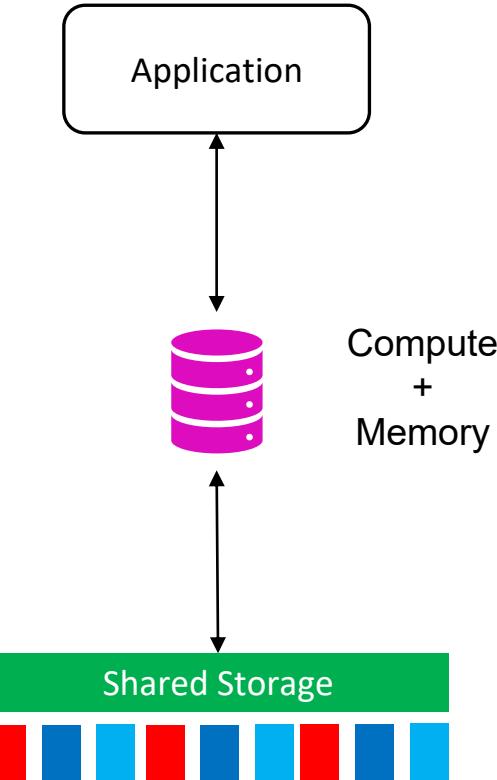
- It's hard to imagine having Relational Database as Serverless.
- No need to provision database for peak loads.
- With Aurora Serverless, you create a database, specify the desired database capacity range, and connect your applications.
- It automatically starts up, shuts down, and scales capacity up or down based on your application's needs.



# Aurora Serverless

- It's hard to imagine having Relational Database as Serverless.
- No need to provision database for peak loads.
- With Aurora Serverless, you create a database, specify the desired database capacity range, and connect your applications.
- It automatically starts up, shuts down, and scales capacity up or down based on your application's needs.
- Pay on a per-second basis for the database capacity that you use.

**Use case:** Development and test environments, websites, applications having infrequent, intermittent, or unpredictable load, business critical applications that require high scale



# SQL vs NoSQL

- Need to run ad-hoc queries?
- Predictable traffic?
- Schema is mostly fixed?

Example: Flight booking system, Banking



SQL Database

Oracle, MSSQL, Amazon Aurora,  
PostgreSQL, MySQL

- 90-95% of queries are pre-defined?
- Need consistent performance at any scale?
- Need flexible schema?

Example: eCommerce product catalog, multiplayer games

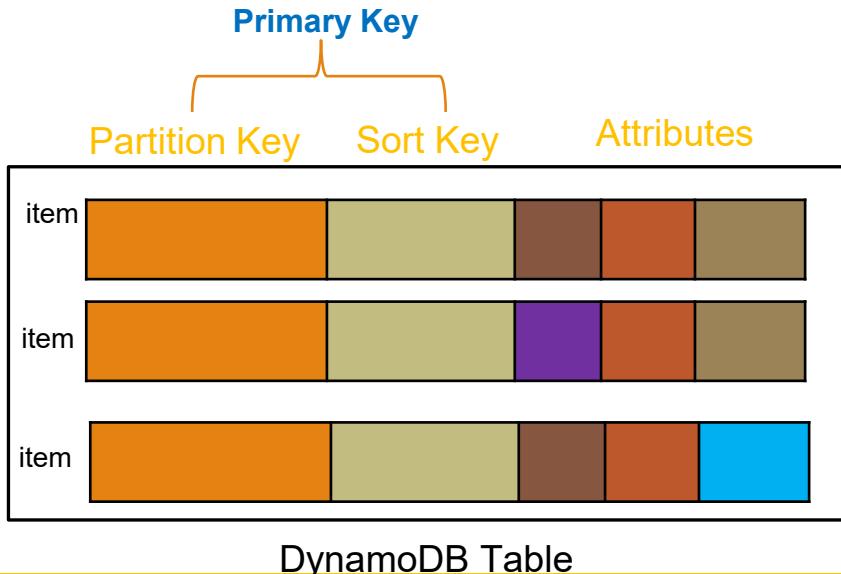


NoSQL Database

MongoDB, Cassandra, Amazon DynamoDB,  
Apache CouchDB, Azure CosmosDB

# Amazon DynamoDB

- Amazon DynamoDB is a fully managed, serverless **NoSQL** (key-value) database service
- Consistent, single-digit millisecond read and write performance at any scale – millions of requests/seconds, trillions of row, 100s of TB of storage
- Serverless: No provisioning or capacity management, Encryption for data at rest, 99.999% availability
- DynamoDB has Table as basic entity to store data
  - Table contains items (like rows)
  - Items contains Partition Key, Sort Key, and attributes (key-value pairs).
  - Partition Key + Sort Key = Primary Key
- Fetch data using Query and Scan operations
  - Query – fetches specific items using Partition Key
  - Scan – fetches every item of the table

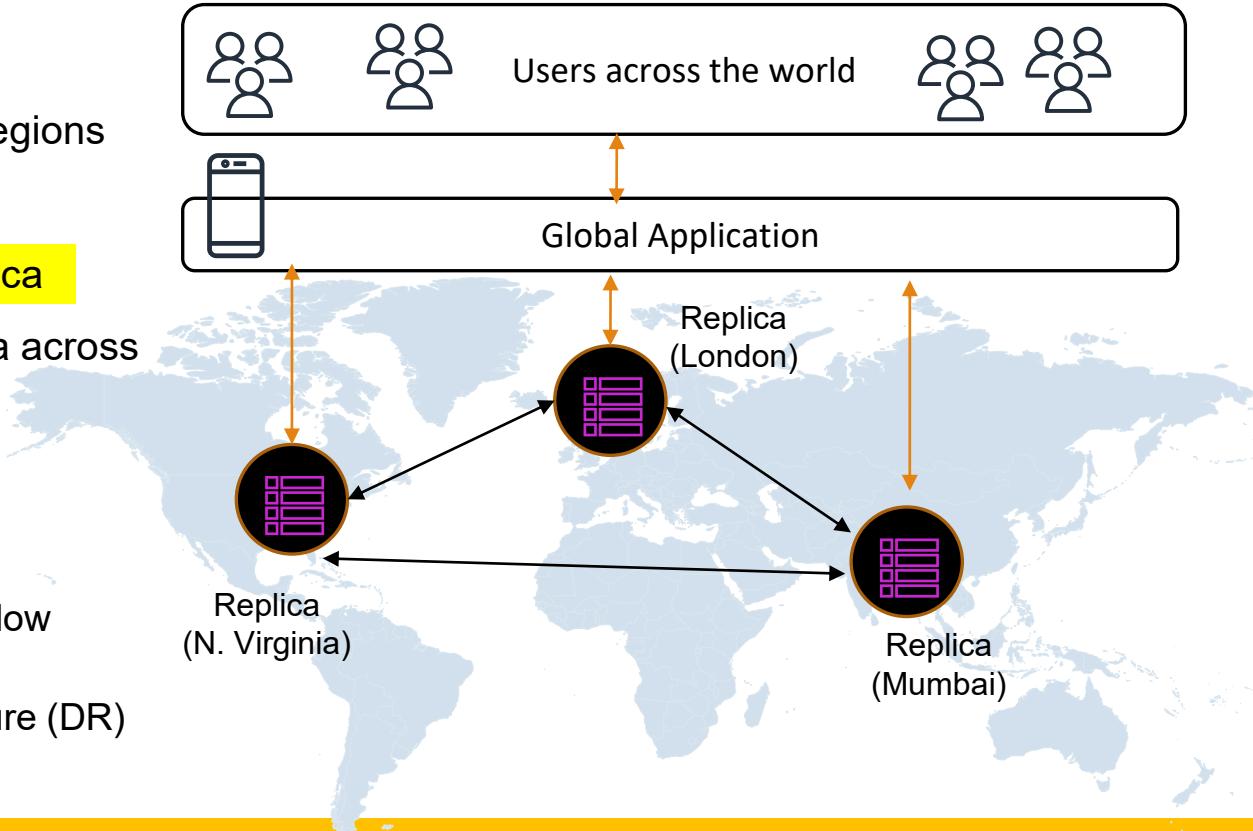


# DynamoDB Global table

- Multi-region, multi-active, serverless tables across regions
- 99.999% availability
- Can read/write to any replica
- DynamoDB replicates data across all replicas

## Use cases:

- Global application requiring low latency access for users
- Can handle region level failure (DR)



# DynamoDB Accelerator - DAX

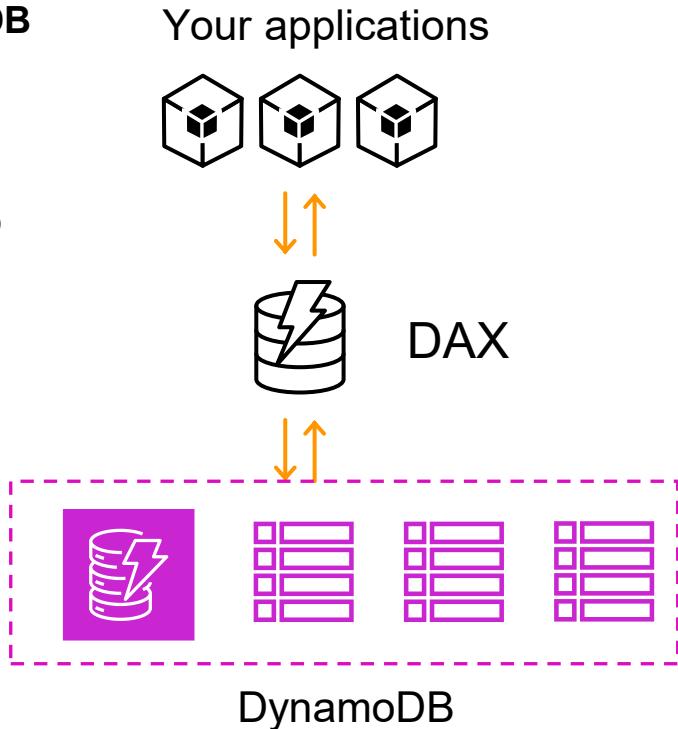
- Fully managed highly available **in-memory cache for DynamoDB**
- 10x performance improvement with single digit millisecond to microsecond level latency
- API-compatible with DynamoDB. Only client & endpoint needs to change to use with an existing application.
- DAX provides access to **eventually consistent** data from DynamoDB tables

## Use cases:

- Real-time bidding, social gaming, and trading applications

## Anti-pattern:

- Strongly consistent read, Write intensive, not having repeated reads

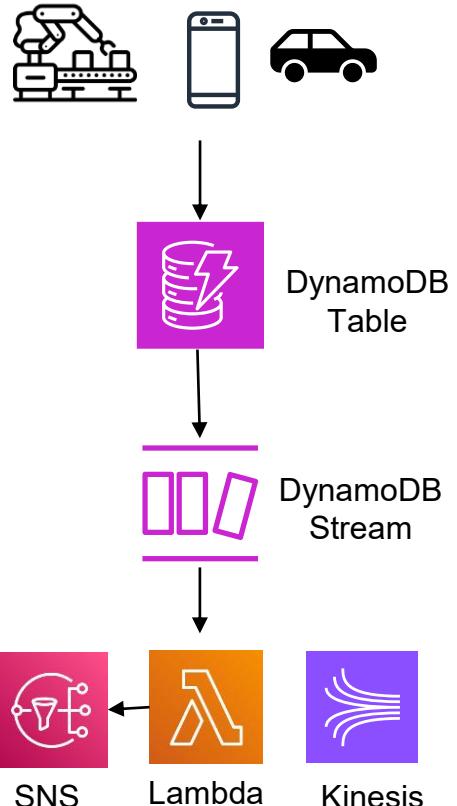


# DynamoDB Streams

- Captures the item level modifications in time-ordered sequence
- Stores the changes in the logs exactly once and strictly ordered for 24 hours
- You can enable/disable a stream on a new or existing table
- DynamoDB Streams operates asynchronously, so there is no performance impact on a table if you enable a stream.

## Use cases:

- Real-time monitoring e.g. connected vehicles, sensor data, Notifying everyone on an activity e.g. friend creates a post on social media
- Backup/Change data capture of DynamoDB table





# Quick recap

- Amazon DynamoDB is a fully managed, serverless cloud-native **NoSQL** database service
- Provides millisecond latency at any scale
- Has Partition key, Sort key (optional) and attributes. Partition key + Sort key = Primary Key
- Support Query and Scan operations – Query fetches specific item, Scan fetches all items in the table.
- Supports Global tables (active-active) for global applications.
- Support DynamoDB accelerator (DAX) providing microsecond latency
- Supports DynamoDB streams to act on table level modifications

# What is document?

- A JSON-like schema with nested key-value structure

## Use cases:

- User profile information (like on LinkedIn)
- Product details, reviews (eCommerce)

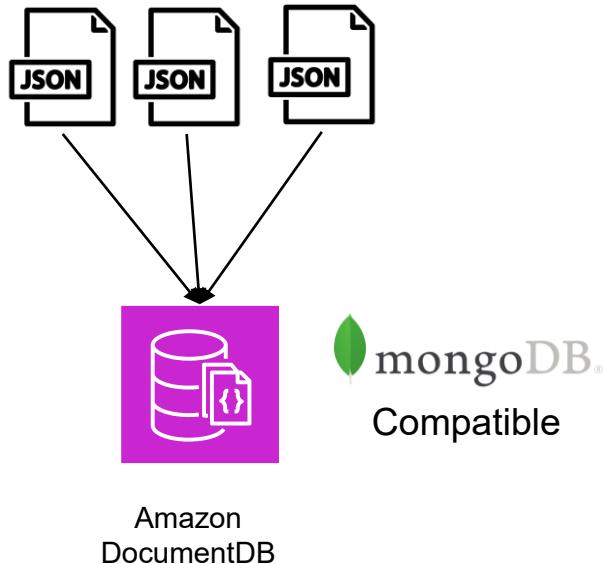
```
{  
    "ProductID": "12345",  
    "Category": "Mobile Phones",  
    "Brand": "ExampleBrand",  
    "Model": "ExampleModel X1",  
    "Specifications": {  
        "ScreenSize": "6.5 inches",  
        "Resolution": "1080x2400 pixels",  
        "Processor": "Octa-core 2.3 GHz",  
        "RAM": "8 GB",  
        "Storage": "128 GB",  
        "Battery": "4000 mAh",  
        "OS": "ExampleOS 12",  
        "Camera": {  
            "Rear": "48 MP + 8 MP + 5 MP",  
            "Front": "16 MP"  
        },  
        "Dimensions": {  
            "Height": "160 mm",  
            "Width": "74 mm",  
            "Depth": "8 mm",  
            "Weight": "180 g"  
        },  
        "Connectivity": ["Wi-Fi", "Bluetooth 5.0", "NFC", "4G LTE"],  
        "Ports": ["USB Type-C"]  
    },  
    "Price": 699.99  
}
```

# Amazon DocumentDB

- Amazon DocumentDB (with **MongoDB** compatibility) is a fast, reliable, and fully managed database.
- Used to store, query, and index JSON-like documents
- DocumentDB storage automatically grows in increments of 10GB, up to 64 TB.
- Automatically scales to workloads with millions of requests per seconds.

## Use cases:

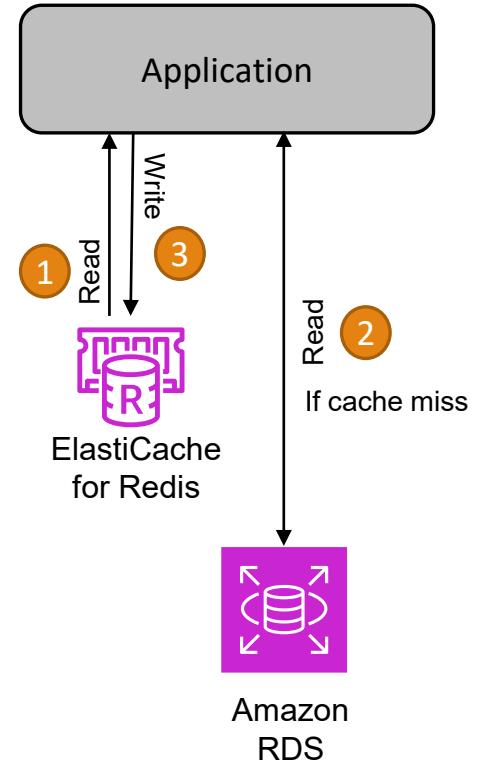
- User profile
- Content management like blogs, video metadata
- E-commerce product catalog (similar to DynamoDB)



# Amazon ElastiCache

- Database Caches are in-memory databases with high performance and extremely low latency to read/write data
- Different caching strategies: Lazy load, Write-through, TTL
- Helps reduce load off database I/O for read intensive workloads.
- Serverless, Highly available, Multi-AZ with cross region replication
- Amazon ElastiCache is compatible
  - Redis OSS 
  - Memcached 
- Memcached is simple to use whereas Redis offers more features such as advanced data structures, transactions, replication, snapshots, pub/sub etc.

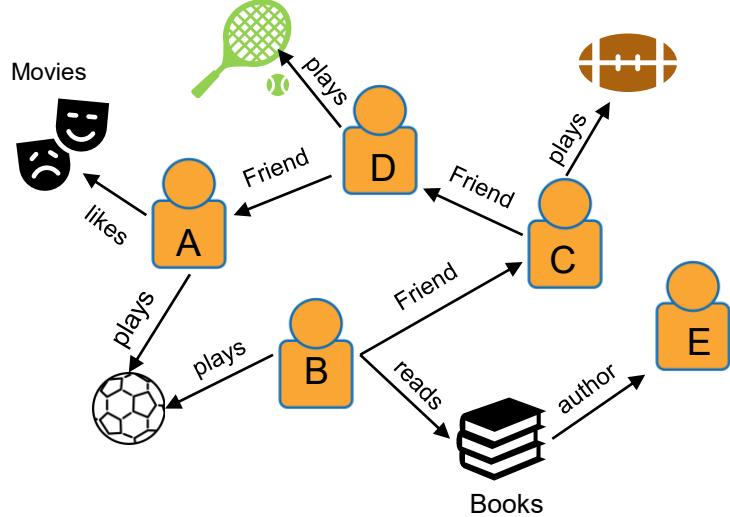
**Use cases:** Session stores, Gaming leader boards, online shopping cart



# Amazon Neptune

- Graph databases store data as a network of entities and relationships.
- Graph consists of Node and Edges where Node represents the object (e.g. person, game) and edge represents relationship between the nodes (e.g. is friend of, plays).
- **Amazon Neptune is Serverless Graph database** (with all the features such as HA, replication etc.)
- Can analyse graph datasets with tens of billions of relationships within seconds using built-in algorithms
- Can perform similarity searches on vectors stored along with your graph for gen AI apps.

**Use cases:** Social networking, Fraud detection, Recommendation engines, Route optimization, Knowledge graphs (e.g. Wikipedia)



Amazon  
Neptune

# Amazon Timestream, Amazon QLDB



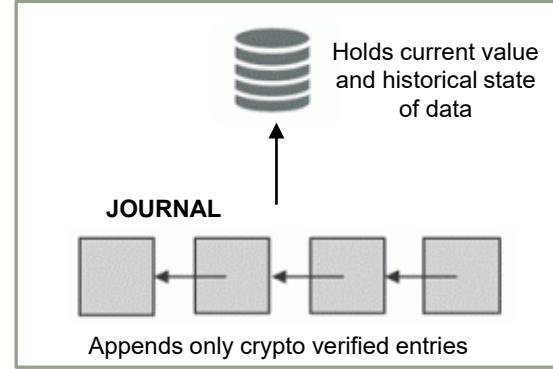
## Amazon Timestream

- Time series Database
- Up to 1000 times faster and 1/10th of cost of Relational databases
- InfluxDB compatible



## Amazon QLDB

- Quantum Ledger Database
- Immutable chain of records
- Cryptographically verifiable log of data changes

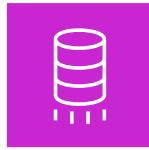


### Use cases:

- Live analytics
- Real time IoT data ingestion and analytics
- Real time Web traffic, Operational metrics

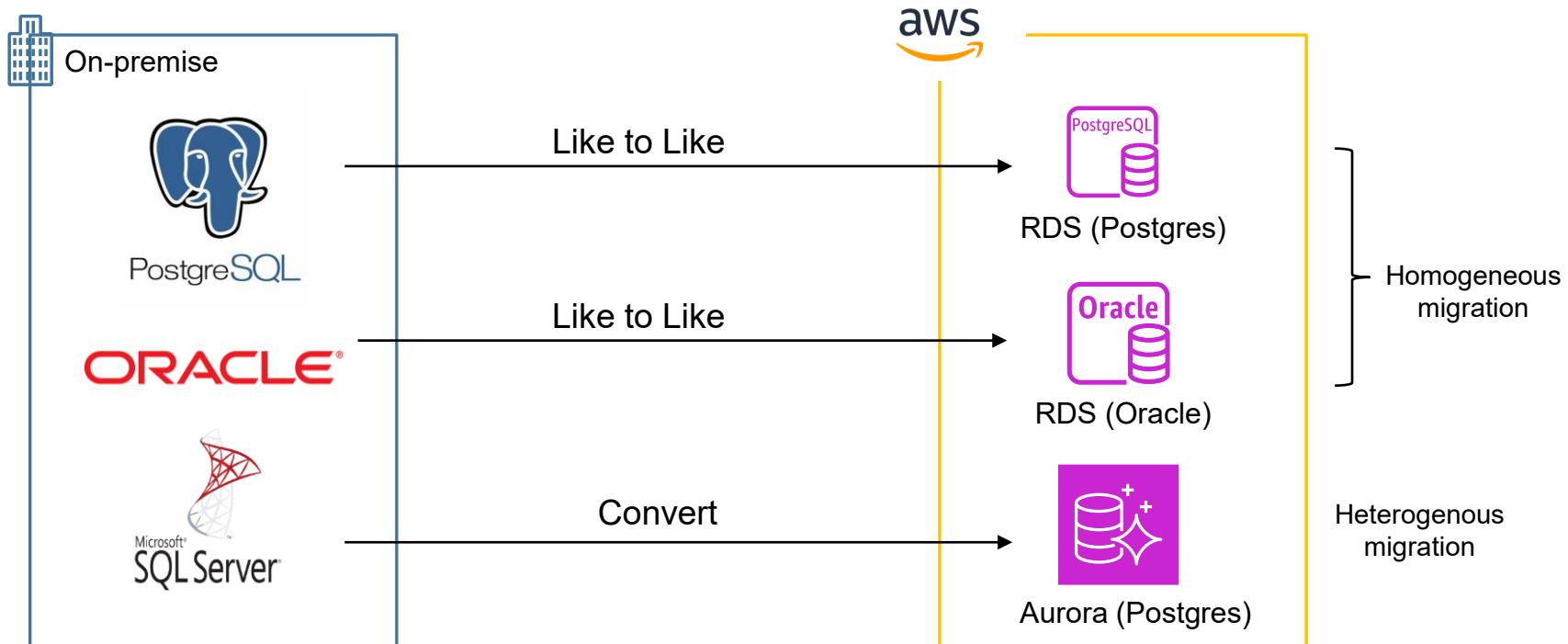
### Use cases:

- Financial records
- Supply chain system
- Claim history
- Trace and Tracking systems  
e.g. spare parts/inventory movement



# Amazon Database Migration Service

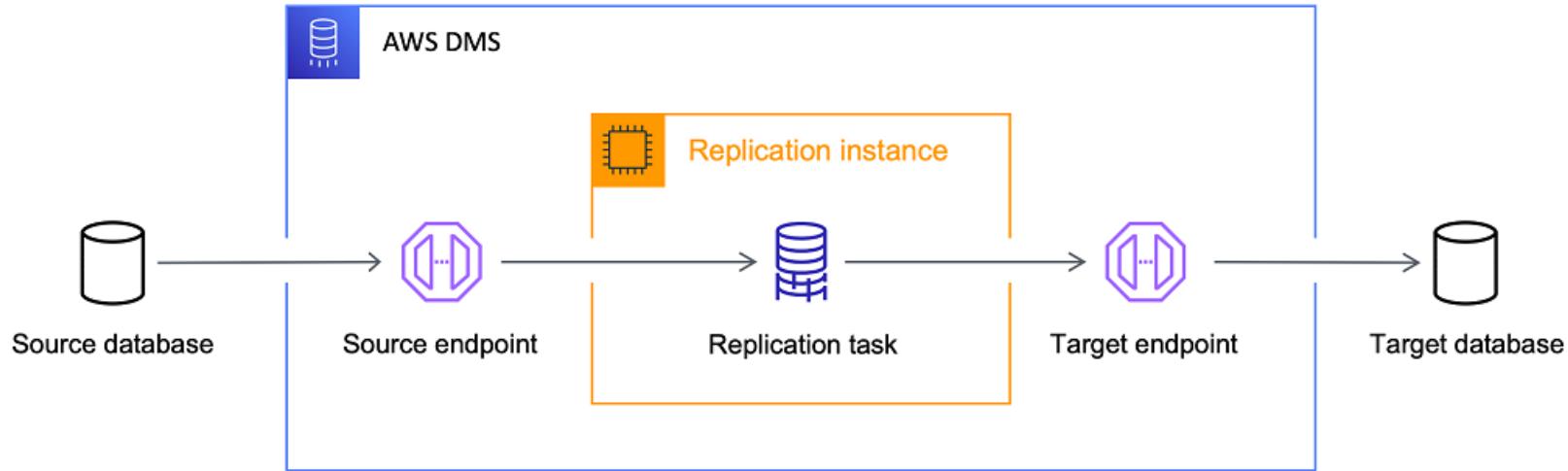
# How to migrate existing database to AWS?



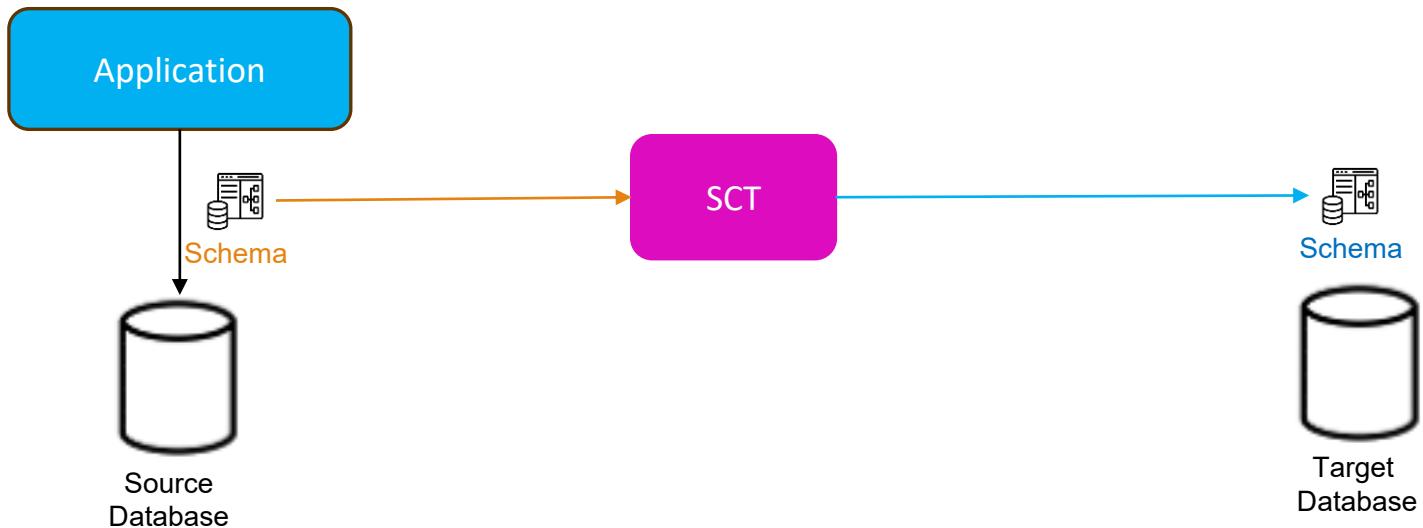
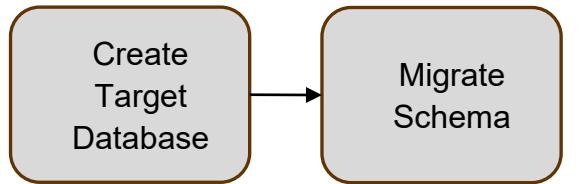
# Amazon Database Migration Service - DMS

- Quickly and securely migrate databases to AWS with minimal downtime and zero data loss.
- The source database remains available during the migration.
- Supports:
  - Homogeneous Migrations: Oracle to Oracle
  - Heterogeneous Migrations: Microsoft SQL Server to Aurora
- **AWS Schema Conversion Tool** (SCT) converts the source schema and a majority of custom code, views, stored procedures to a format as per target database

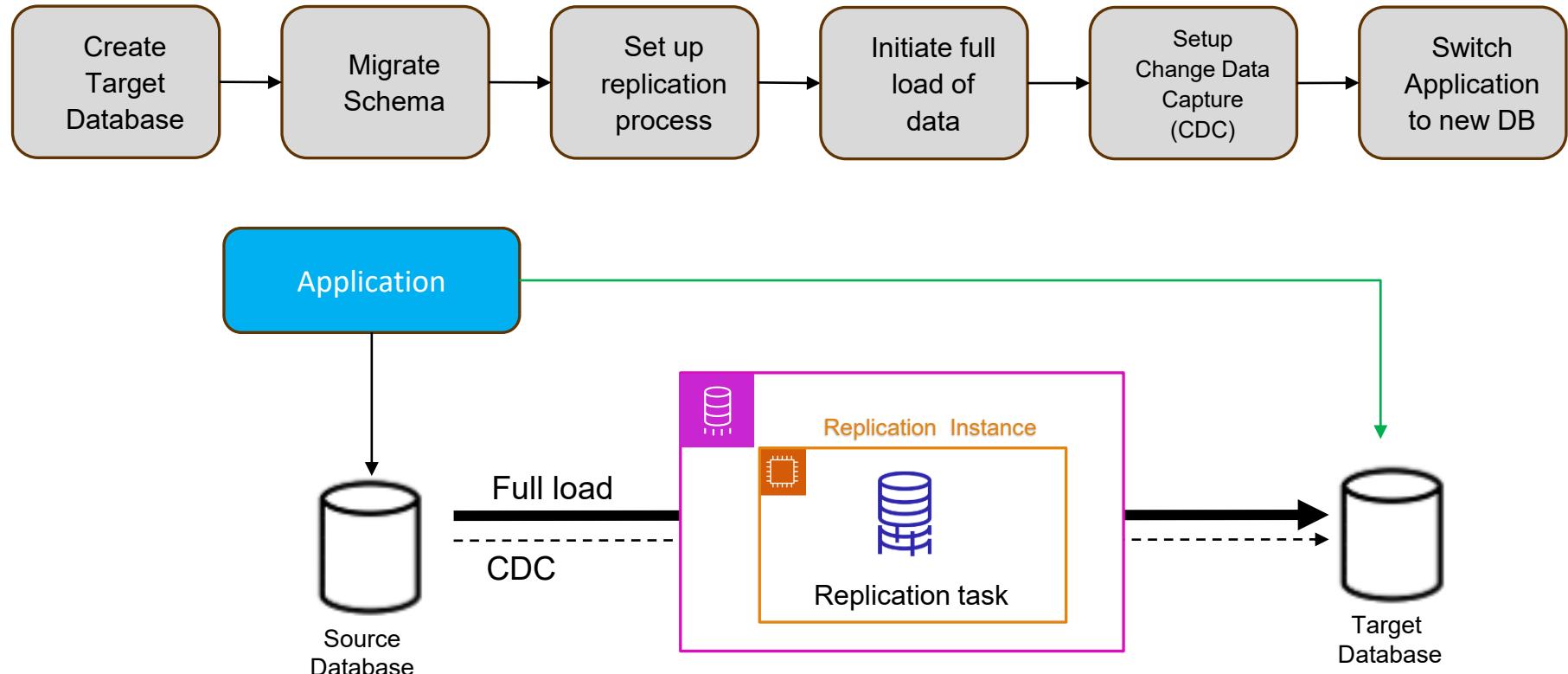
# Amazon Database Migration Service - DMS



# Database migration steps

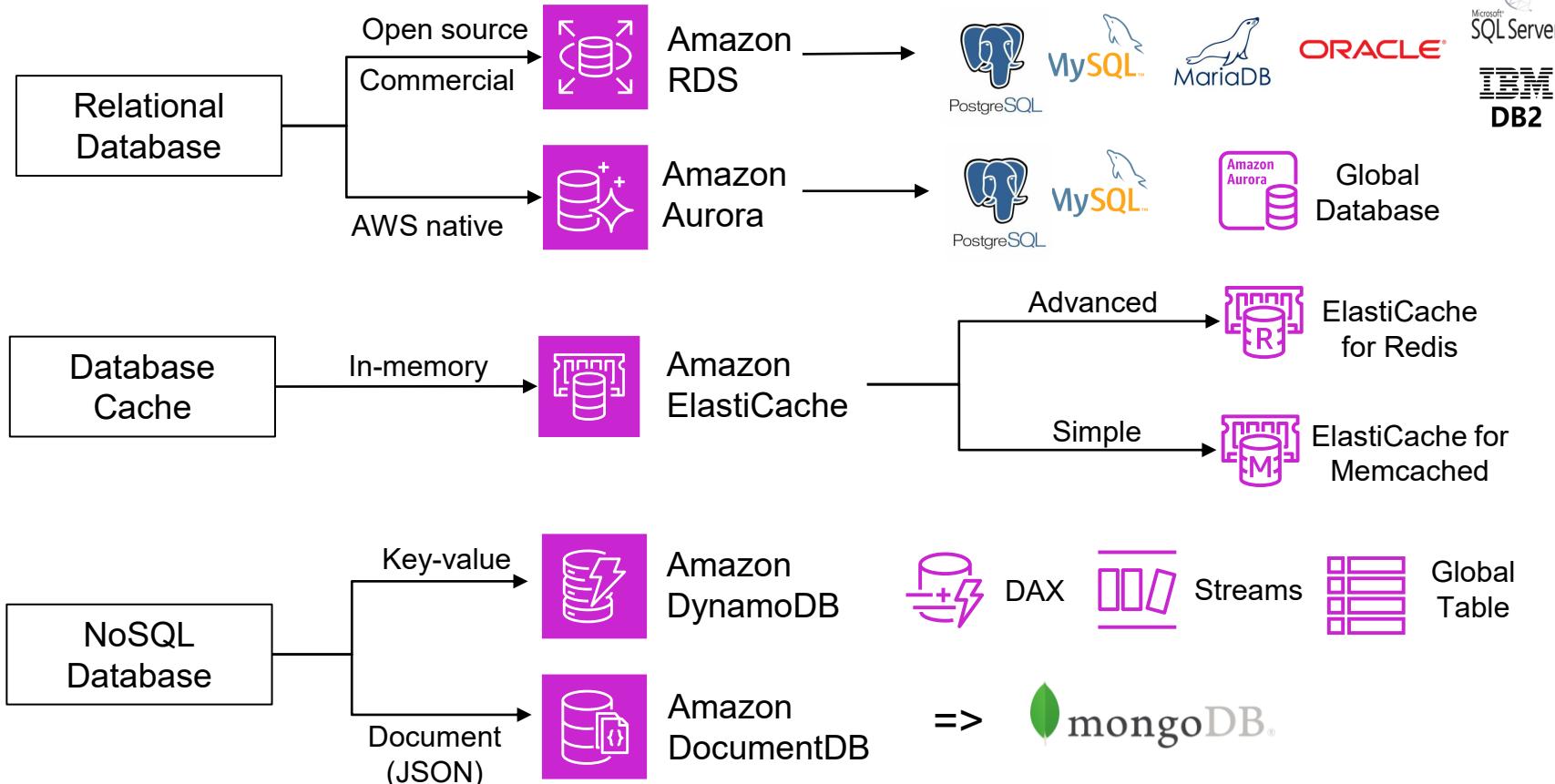


# Database migration steps

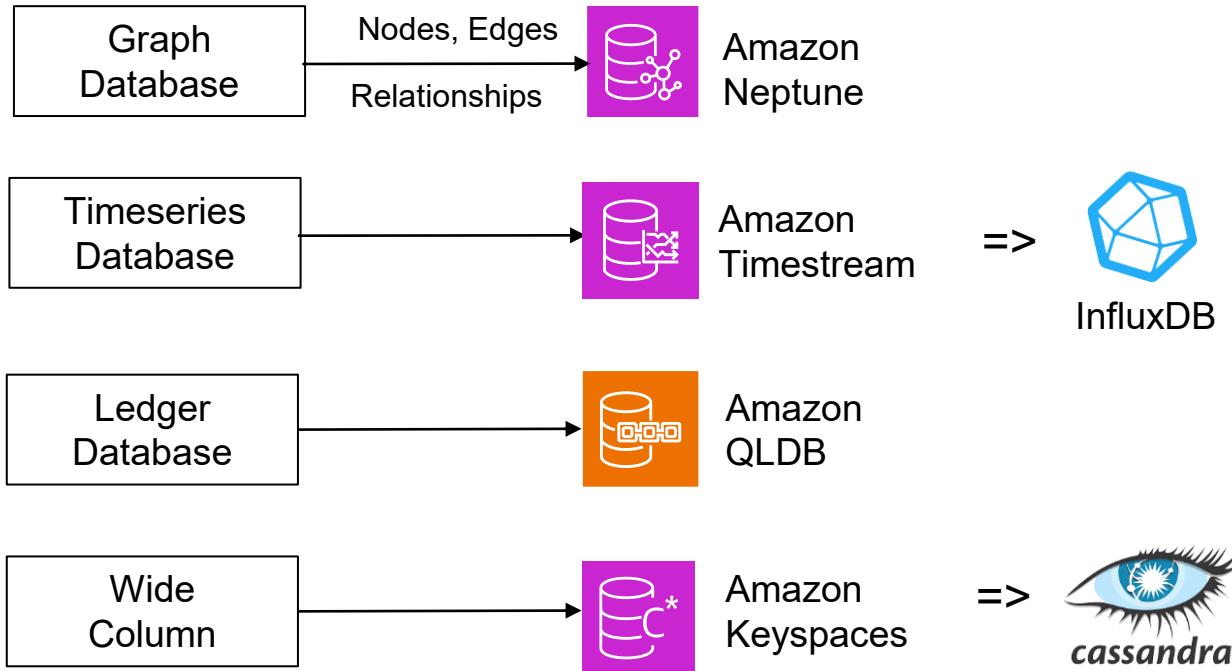


# Amazon Databases - Summary

# Amazon Database services summary



# Amazon Database services summary



# Amazon Database services summary

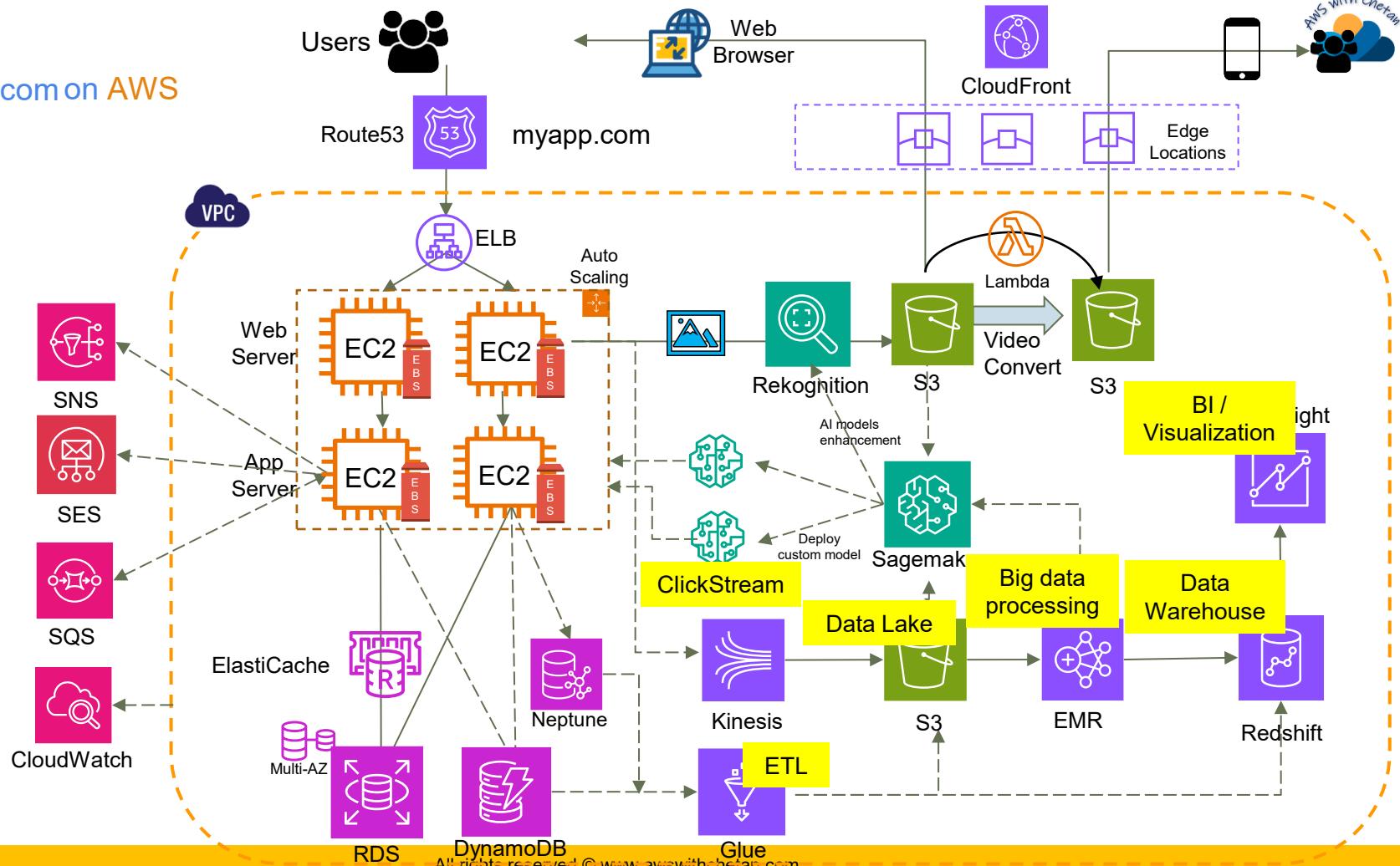
1. RDS is a fully managed relational database service where provisioning, patching, upgrades etc. is handled by AWS.
2. RDS supports multi-az deployment for high availability. Real time replication across AZs.
3. RDS supports Read replicas for scaling the read traffic. Read replica can be promoted to primary DB in case of failure of primary DB instance.
4. Amazon Aurora is PostgreSQL and MySQL compatible AWS proprietary relational database.
5. DynamoDB is a NoSQL key-value database which provides single digital millisecond latency at scale.
6. DynamoDB Global Table has active-active setup across region. Data can be written to any replica which is replicated to other replicas asynchronously.
7. DynamoDB accelerator (DAX) is a cache for DynamoDB. It provides sub-milliseconds to microsecond latency for read queries.
8. DocumentDB stores JSON-like documents and useful for content management, profiles, product catalog etc.
9. Neptune DB is graph database where nodes are the objects and relations are the edges.
10. Neptune DB can be used for social media connections, fraud detection, route optimization, knowledge graphs etc.

# Important RDS features to remember

- **Multi-AZ deployment**
  - DB instances across 2 or more Availability Zones for High Availability
  - Automatic failover to standby in case of failure in Primary DB or Availability Zone
  - Synchronous Replication thereby no loss of data
- **Read Replicas**
  - **Primary purpose:** Scale the read throughput using Read replicas (across Regions if required)
  - In case of failure in Primary database, promote Read Replica to standalone database

Let's understand basics & terminologies of  
**Big data and Analytics**

myapp.com on AWS

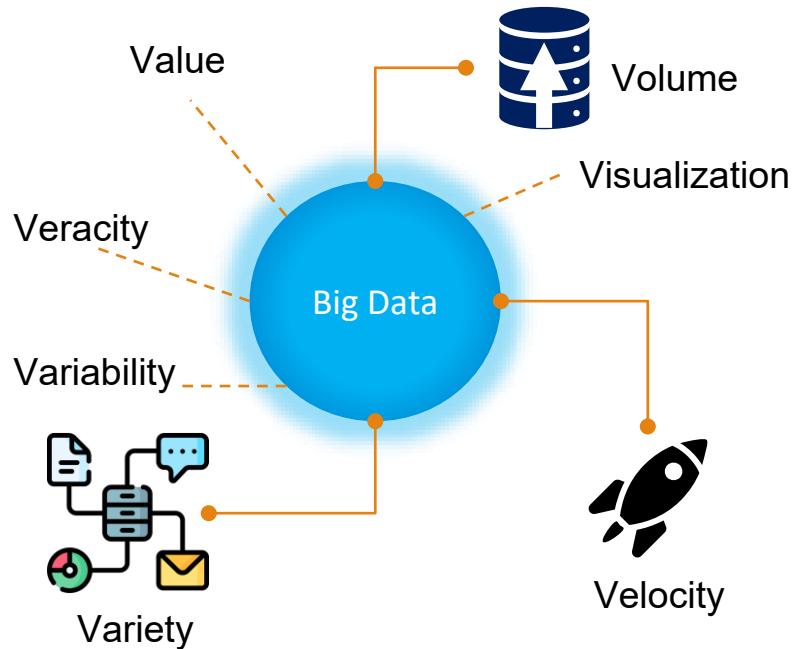


# What is Big Data?

**Big Data** refers to extremely large datasets that cannot be easily managed, processed, or analysed using traditional data processing techniques.

**Big data is characterized by Three V's:**

- **Volume:** The amount of data generated every second from various sources like social media, sensors, transactions, etc.
- **Velocity:** The speed at which new data is generated and needs to be processed. This includes real-time data feeds and streaming data.
- **Variety:** The different types of data, such as structured data (databases), semi-structured data (XML, JSON), and unstructured data (text, images, videos).



# Big data frameworks..

Big data frameworks are instruments that simplify the processing of big data.

## Popular Big Data frameworks

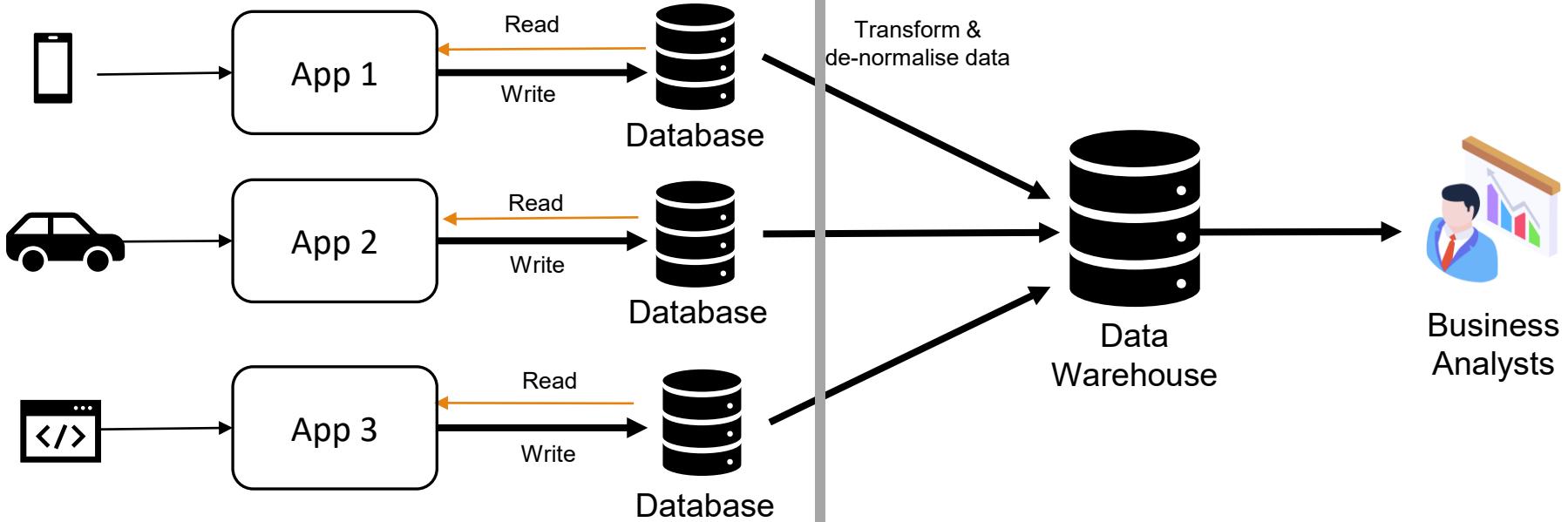
- **Hadoop:** Open-source batch-processing framework for the distributed storage and processing of big data sets. Hadoop operates by splitting files into large blocks of data and then distributing those datasets across the nodes in a cluster. Uses HDFS, MapReduce, YARN.
- **Apache Hive:** For data summarization, query, and analysis. Queries data from HDFS.
- **Apache Hbase:** A distributed big data store that supports structured data storage for large tables
- **Apache Spark:** In-memory real-time processing, batch processing
- **Presto:** Distributed SQL query engine optimized for running interactive analytic queries against data sources of all sizes ranging from gigabytes to petabytes.
- **Apache Pig, Apache Sqoop, Apache Flume, Apache Mahout** and more..

# OLTP

Online transaction Processing

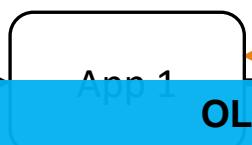
# OLAP

Online Analytical Processing



# OLTP

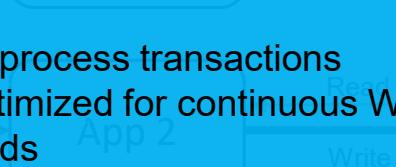
Online transaction Processing



- To process transactions
- Optimized for continuous Writes & small reads
- Data is highly normalized
- Distributed databases
- ACID properties
- Data volume – in order of GBs
- Example: Order management, Ticket booking, ATM, Banking transactions

# OLAP

Online Analytical Processing



Transform &  
de-normalise data

# OLAP

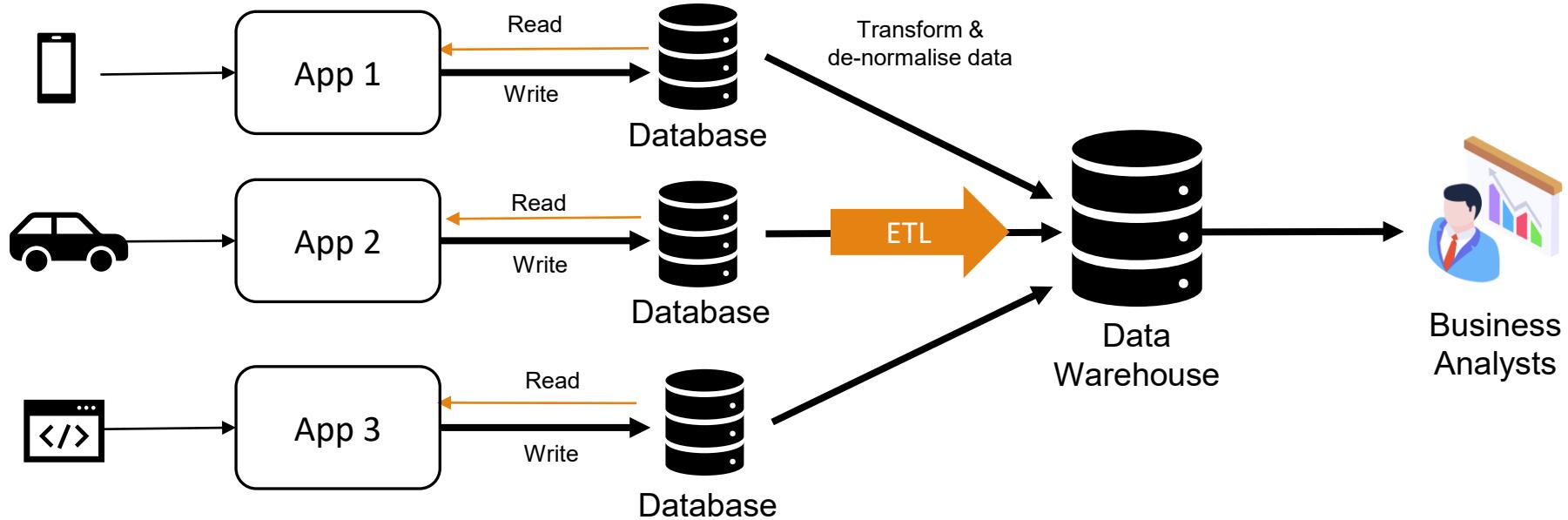
- To analyse aggregated data
- Optimized for Batch Writes & high volume read
- Data is denormalized
- Centralized database
- Data volume – in order of TBs/ PBs
- Example: Data warehouse, analyse market trends, Predict customer behaviour, sales reports



Business  
Analysts

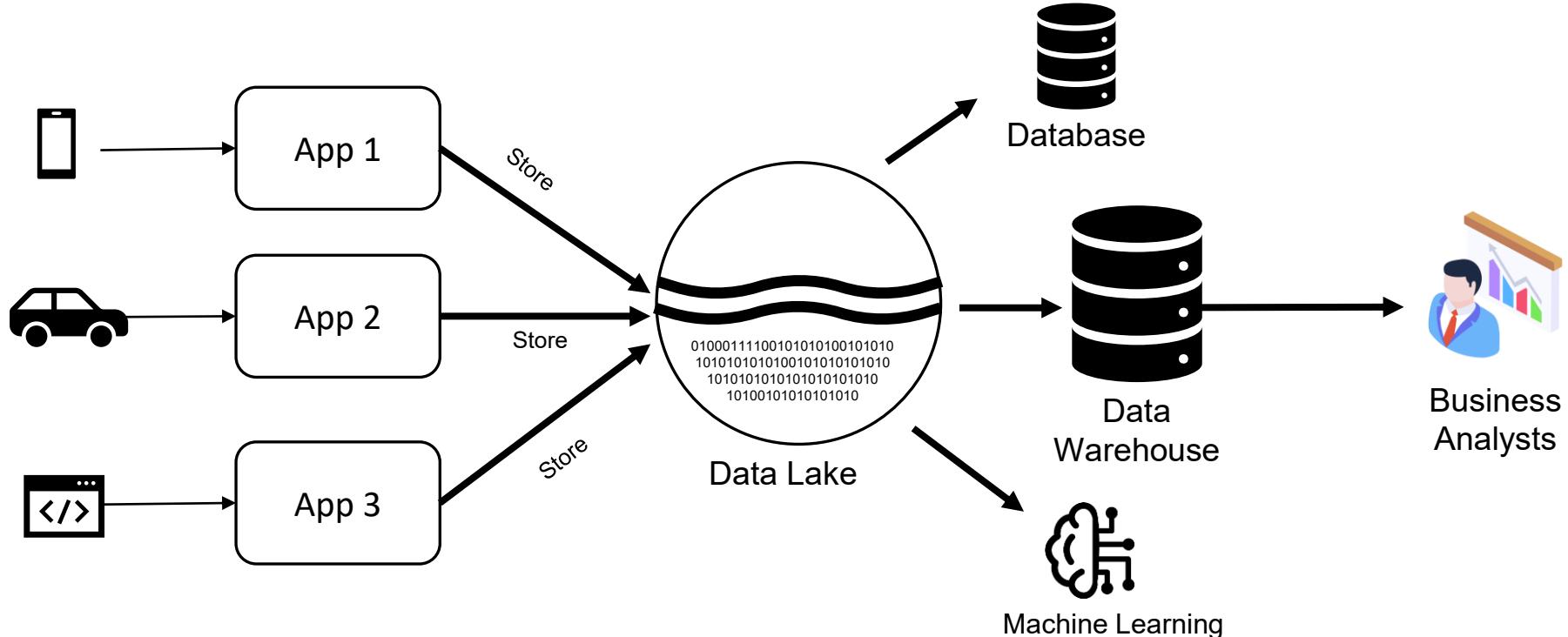
# Data Warehouse

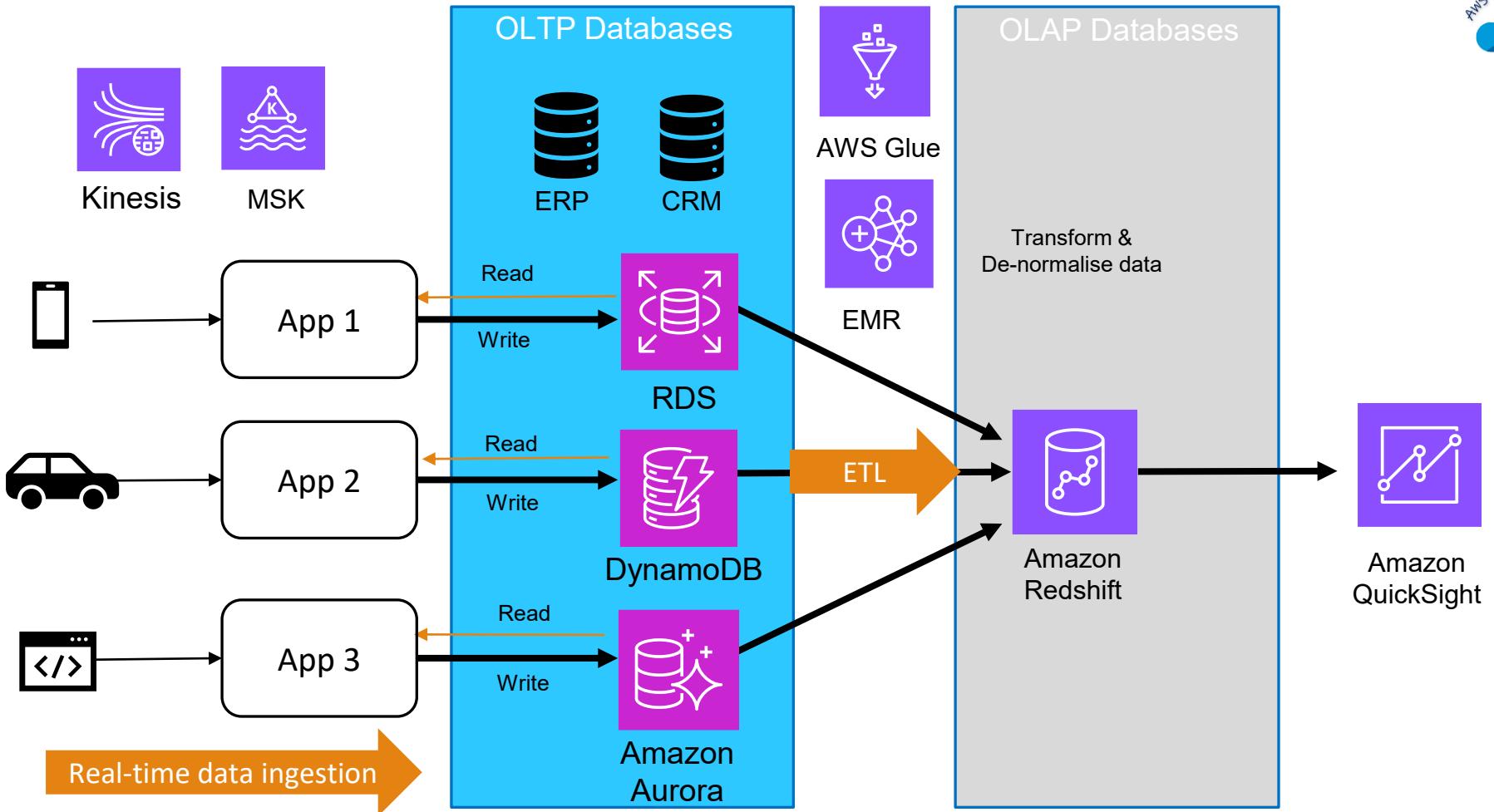
A data warehouse is a system that stores and organizes data from multiple sources for reporting and analysis

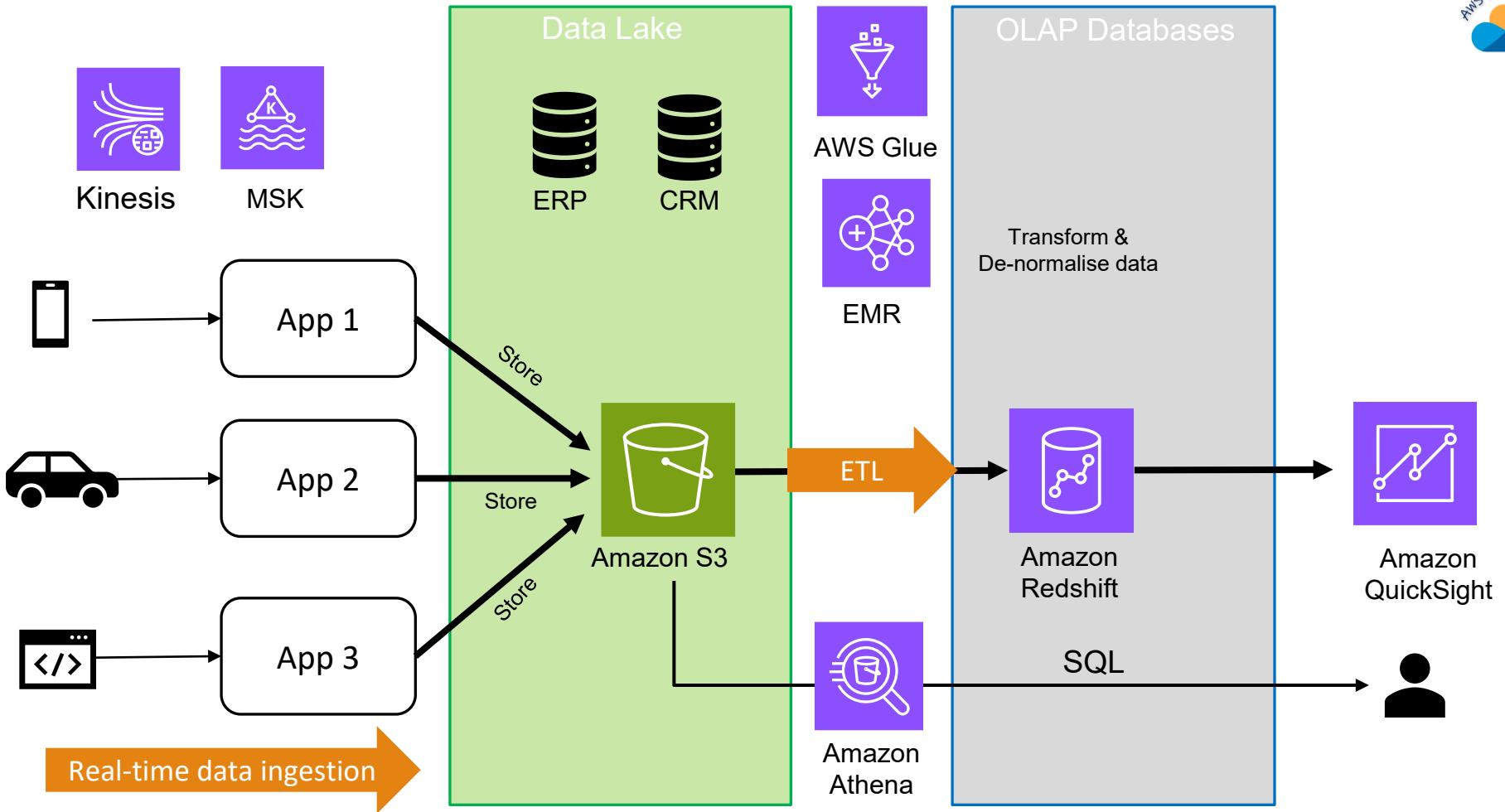


# Data Lake

A data lake is a centralized repository for storing, processing, and securing large amounts of data



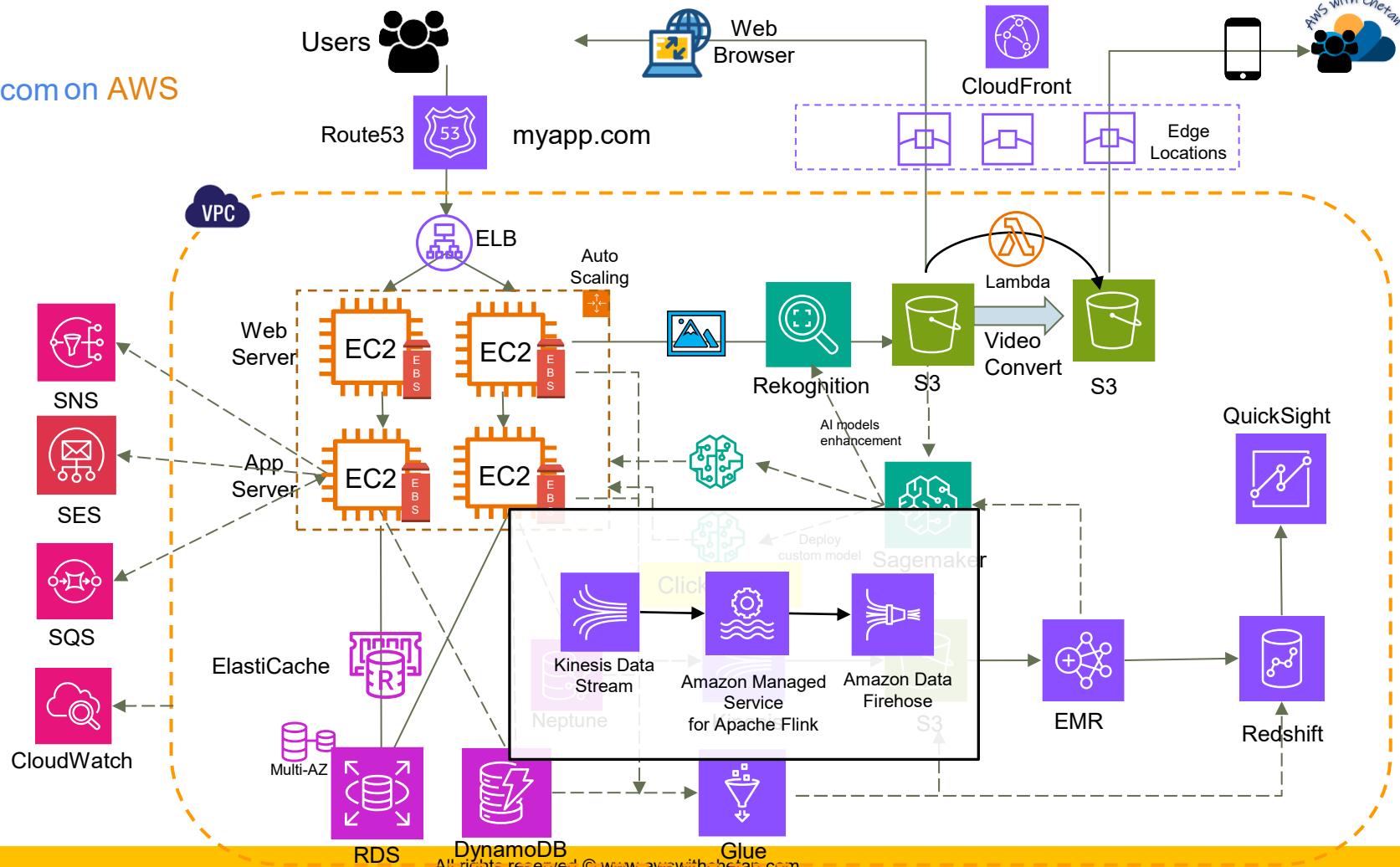






# Amazon Kinesis

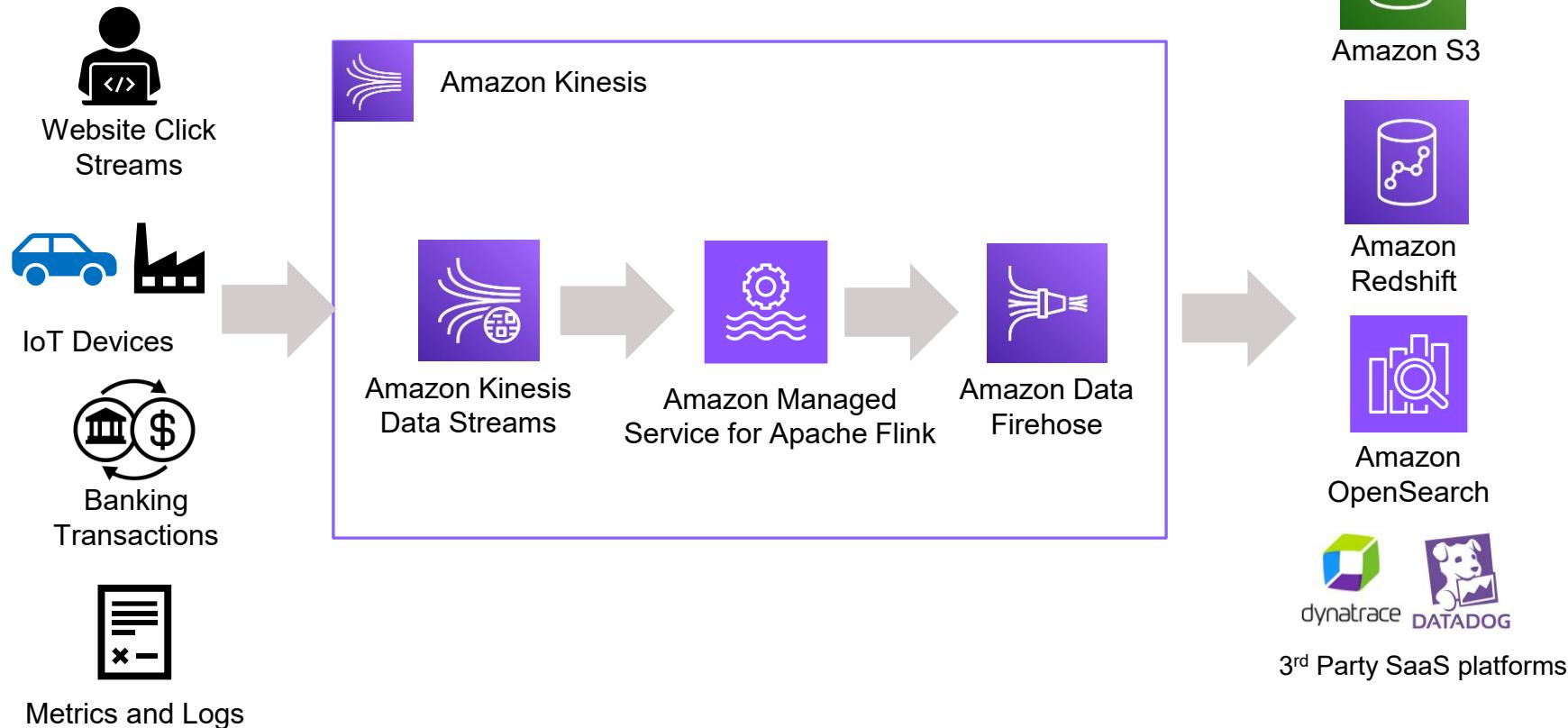
# myapp.com on AWS



# Amazon Kinesis

- AWS managed service to collect, process and analyzes real-time, streaming data so you can get timely insights and react quickly to new information.
- Ingest real-time data such as video, audio, application logs, website clickstreams and other applications.
- Kinesis offers following services:
  - **Amazon Kinesis Data Streams** : Low latency streaming to ingest data at scale from hundreds of thousands of sources.
  - **Amazon Kinesis Video Streams** : Monitor real-time video streams for analytics or ML.
  - **Amazon Managed Service for Apache Flink** : Perform real-time analytics on streams using SQL.
  - **Amazon Data Firehose** : Load streams into S3, Redshift, OpenSearch etc. Optionally perform transformation of the records using Lambda.

# Amazon Kinesis family of services



# Amazon Kinesis – Use cases

- Fraud detection in Financial services
- Analyzing customer behavior in real-time – personalized recommendations, offers, discount codes
- Logs monitoring and alerting
- Read time Ad targeting
- IoT Data processing – To detect malfunctions or anomalies
- Social Media – Trending topics, user engagement, content recommendations

- Home security and Smart cameras – Motion detection, face detection
- Connected vehicles – Dizzy driver, Drive Behavior
- Industrial IoT and remote monitoring – Equipment health, Oil rigs in powerplant, security breaches



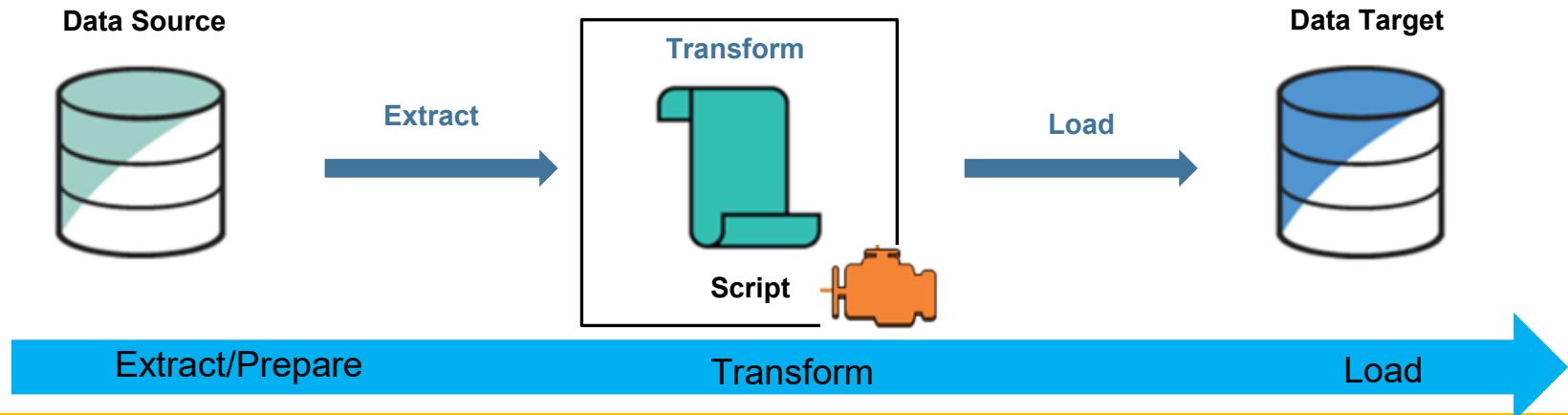
Amazon Kinesis  
Video Streams



# AWS Glue

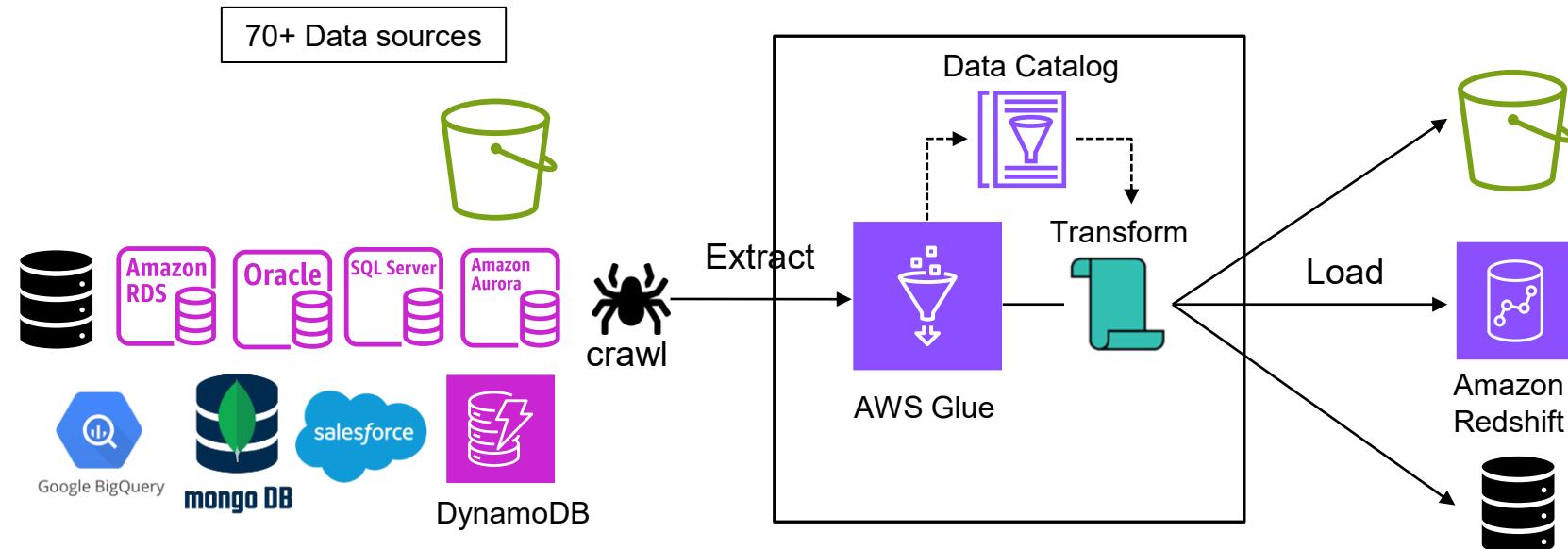
# AWS Glue

- Managed Extract, Transform, and Load (ETL) service.
- Automatic **Schema discovery** and **Data cataloging** with Glue Crawler.
- Supports Apache Spark / Python shell / Ray engines for Data transformation.



# AWS Glue

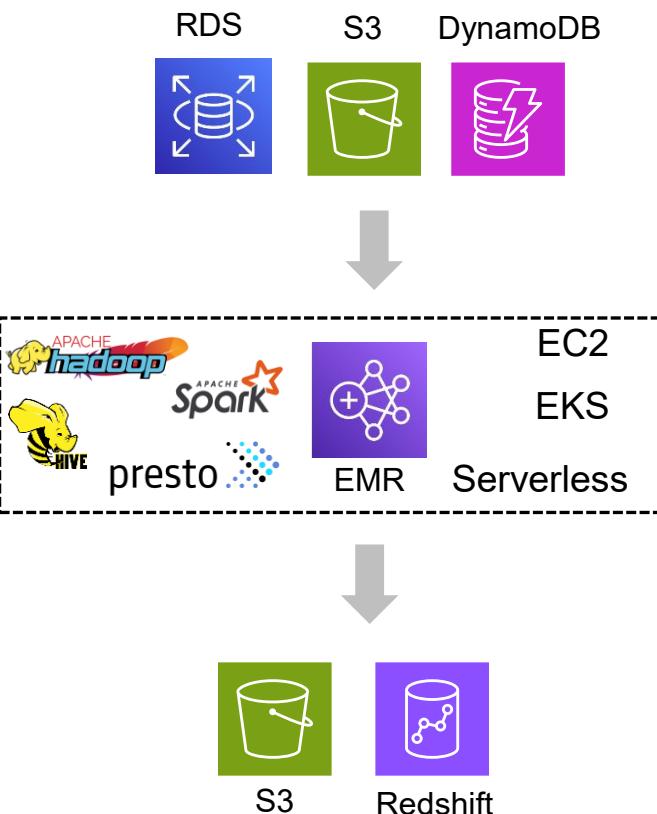
- Managed Extract, Transform, and Load (ETL) service.
- Automatic **Schema discovery** and **Data cataloging** with Glue Crawler.
- Supports Apache Spark / Python shell / Ray engines for Data transformation.





# Amazon EMR

- Amazon EMR (**Elastic MapReduce**) is an AWS service used for big data processing and analysis.
- Supports Hadoop Map Reduce and Apache Spark, Hive, Presto frameworks
- Distribute your data and processing across resizable clusters of Amazon EC2, EKS
- Uses HDFS and EMRFS (S3) for storage.
- Auto-scaling and integrated with EC2 Spot instances.
- **Use Cases:** Log analysis, Web indexing, Machine learning training data, Financial data analysis, Market trends, Customer preferences.



# AWS Glue vs Amazon EMR



AWS Glue

- **Simplicity** – Low code/no code
- Primarily for running ETL jobs
- Suitable for ad-hoc & small batch jobs
- Fully managed / no code
- When to use AWS Glue?
  - Built-in capabilities - connectors, transformations, incremental load, job monitoring, orchestration.
  - Visual and low code ETL development tools
  - Migration from ETL providers such as Informatica, Talend, Matillion

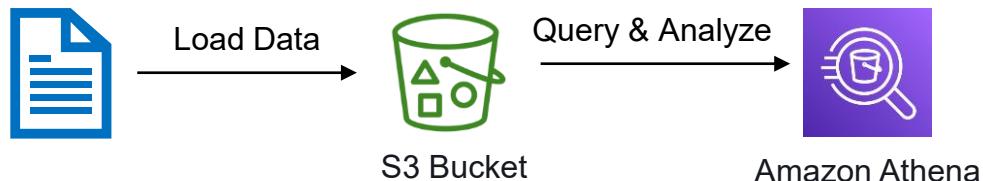


Amazon EMR

- **Flexibility**
- Primarily for Big data processing and ETL
- Suitable for large scale distributed data processing with consistent usage
- Some level of infrastructure management
- When to use Amazon EMR?
  - Hadoop Migration from on-prem
  - Have expertise beyond just Spark, for ex. Hive, Presto
  - Customer is skilled in loading their own data source connector libraries for their jobs.

# Amazon Athena

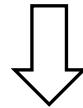
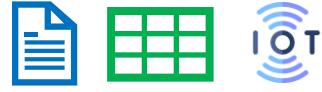
- Provides the easiest way to run ad hoc **interactive SQL queries** for data in Amazon S3 and other data sources without the need to setup or manage any servers.
- Athena is serverless, built on open-source Trino and Presto engines
- Pay only for the queries you run. Pricing: \$5.00 per TB of data scanned.
- Supports CSV, JSON, or columnar data formats such as Apache Parquet and Apache ORC.
- Athena integrates with Amazon **QuickSight** for easy data visualization.
- **Use cases:** Ad-hoc SQL queries for Business intelligence and analytics, query AWS logs e.g. VPC Flow Logs, ELB Logs, CloudTrail logs



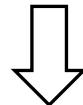
# Amazon Redshift

- Amazon Redshift is a fully managed, **petabyte-scale data warehouse** service
- Amazon Redshift uses massively Parallel Processing (MPP) architecture and has a SQL interface for performing the queries.
- Can ingest data across data lakes, databases, streaming data - with no code/low code **zero-ETL** approach
- No complex infrastructure management.
- Amazon Redshift data sharing feature allow sharing data across AWS regions, teams, and third-party data warehouses without data movement or data copying.
- BI tools such as AWS QuickSight, Tableau work seamlessly with Redshift.
- **Amazon Redshift Serverless** - Automatically provisions and scales data warehouse underlying capacity

Data



Amazon  
Redshift

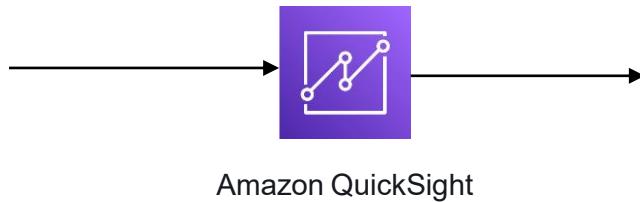


Insights

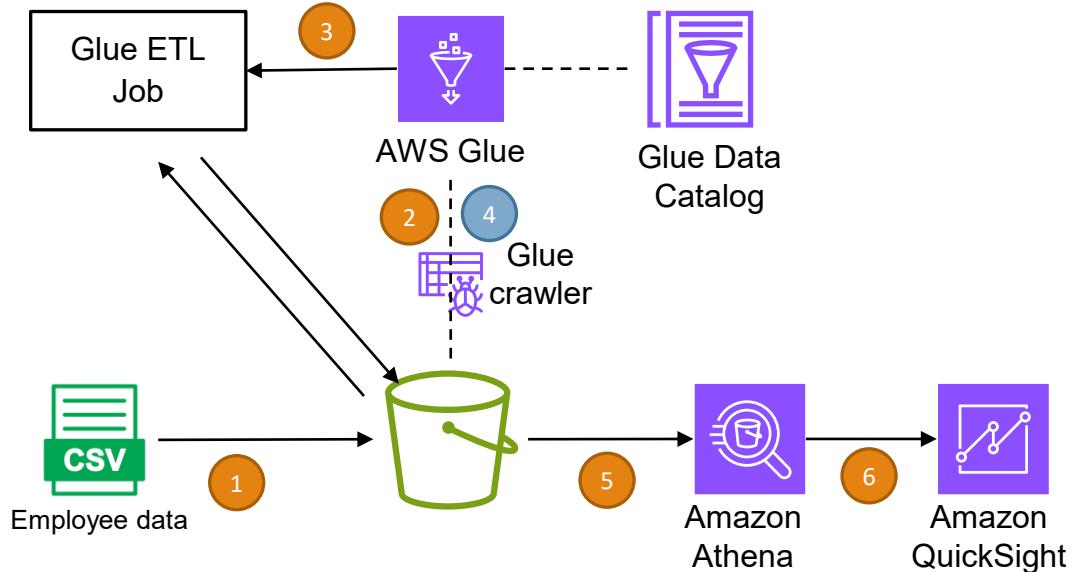
# Amazon QuickSight

- Serverless machine learning-powered **Business Intelligence** service to create interactive dashboards.
- Fast, automatically scalable, embeddable, with per-session pricing.
- Integrates with Amazon RDS, Aurora, Athena, S3, OpenSearch, Redshift and allows uploading files (CSV, XLS, JSON etc.) using file data source

**Use Cases:** Build Visualizations, Perform ad-hoc analysis



# Demo – Data pipeline using S3, Glue, Athena

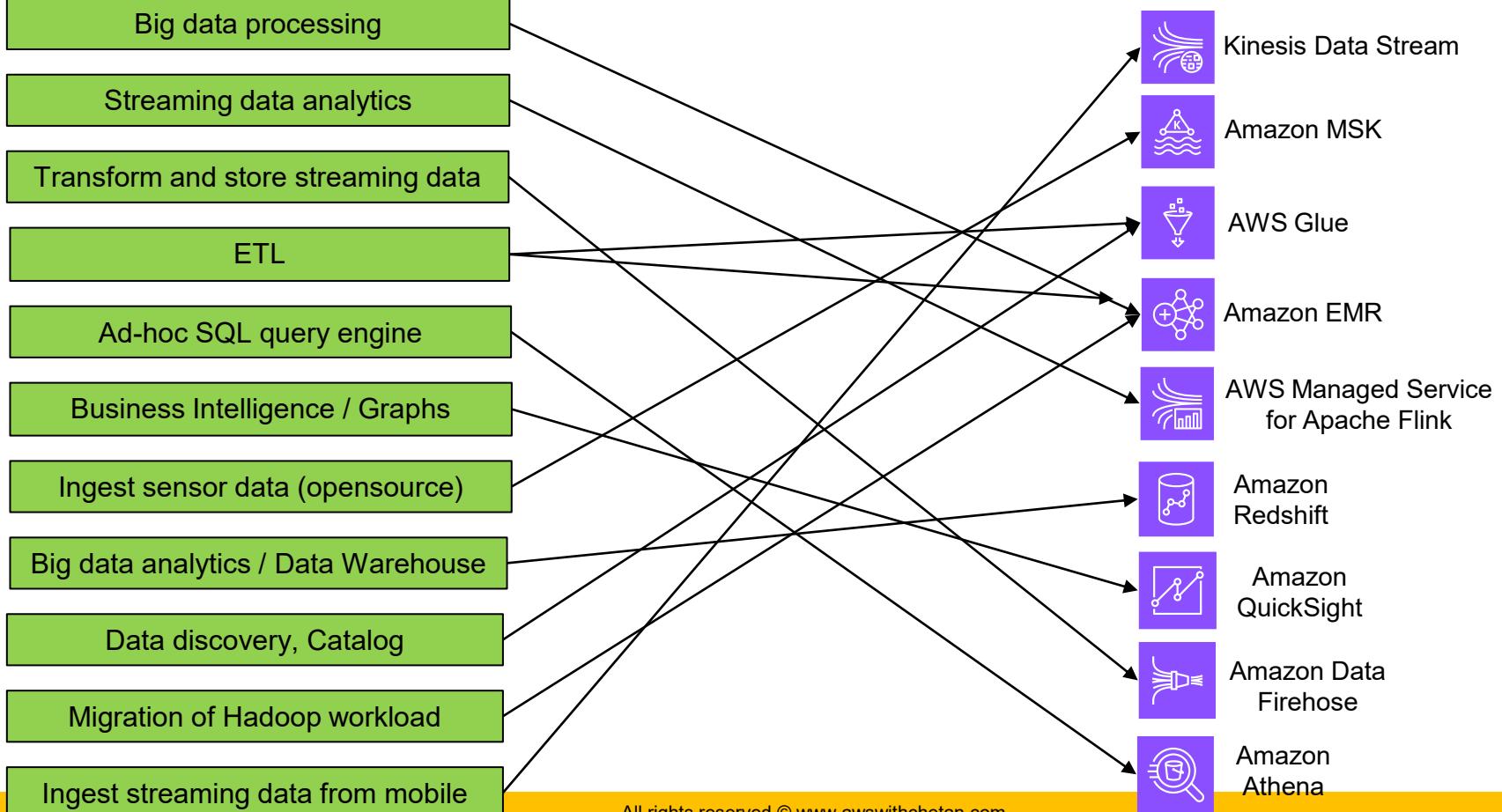


## High level steps

- 1 Upload sample CSV file to S3
- 2 Create AWS Glue crawler and run it. This should create a data catalog with Glue database and table
- 3 Create Glue ETL (python) job to modify one of the column in the CSV and create a new CSV file in S3
- 4 Rerun the Glue crawler again. This should create another table in the database
- 5 Using Amazon Athena query the Glue database and table created above
- 6 (Optional) Create analysis and dashboard in Amazon QuickSight using Athena as a data source

# AWS Analytics services - Summary

# AWS Analytics services - When to use what?



# AWS Analytics services - summary

- OLTP – Online Transaction Processing – Amazon RDS, Aurora and NoSQL databases
- OLAP – Online Analytical Processing – Amazon Redshift
- Data Warehouse - System that stores and organizes data from multiple sources for reporting and analysis (think Amazon Redshift)
- Data Lake – Centralized repository to store large amount of data (think Amazon S3)
- Amazon Kinesis Data Streams – Real-time data ingestion (Connected cars, IoT, machine sensor, click stream etc.)
- Amazon Kinesis Video Streams – Real-time video ingestion (Smart cameras, Home security, Driver behavior etc.)
- Amazon Managed Service for Apache Flink – Real-time data analysis (Fraud detection, Ad, recommendations etc.)
- Amazon Data Firehose – Store streaming data into Amazon S3, Redshift or OpenSearch
- AWS Glue – Managed ETL service with Schema discovery and Data Cataloging
- Amazon EMR – Hadoop platform to run big data frameworks for ETL and big data processing
- Amazon Athena – Interactive, serverless SQL query engine to query data stored in S3 and other data sources
- Amazon Redshift – Petabyte scale Data warehouse service (similar to Snowflake, Google BigQuery)
- Amazon QuickSight – Machine learning powered Business Intelligence (BI) service

# Application Integration services



## Amazon API Gateway

Managed HTTPS, REST & Websocket API service



## AWS AppSync

Managed GraphQL and Pub/Sub API service



## Simple Queue Service (SQS)

Managed Queue service



## Simple Notification Service (SNS)

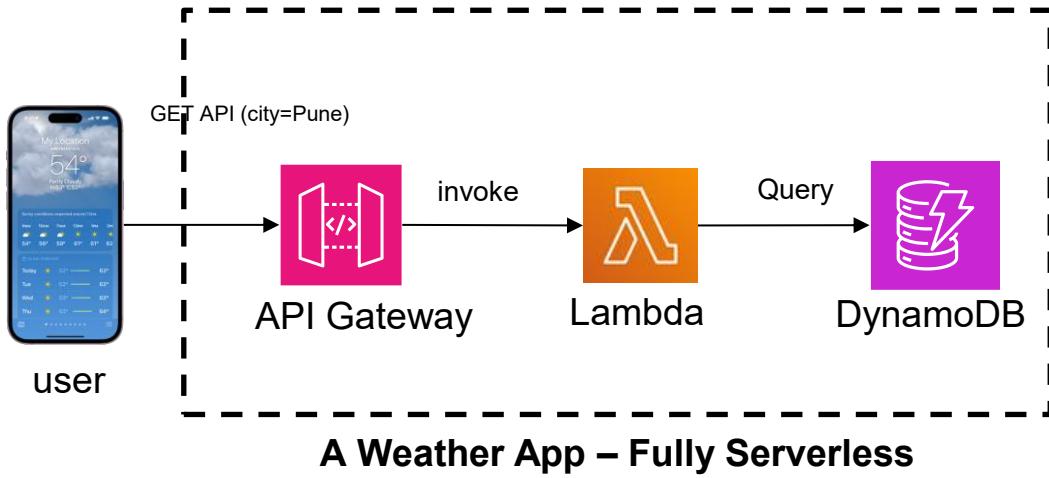
Notification service which delivers SMS, Email, Mobile Push notifications

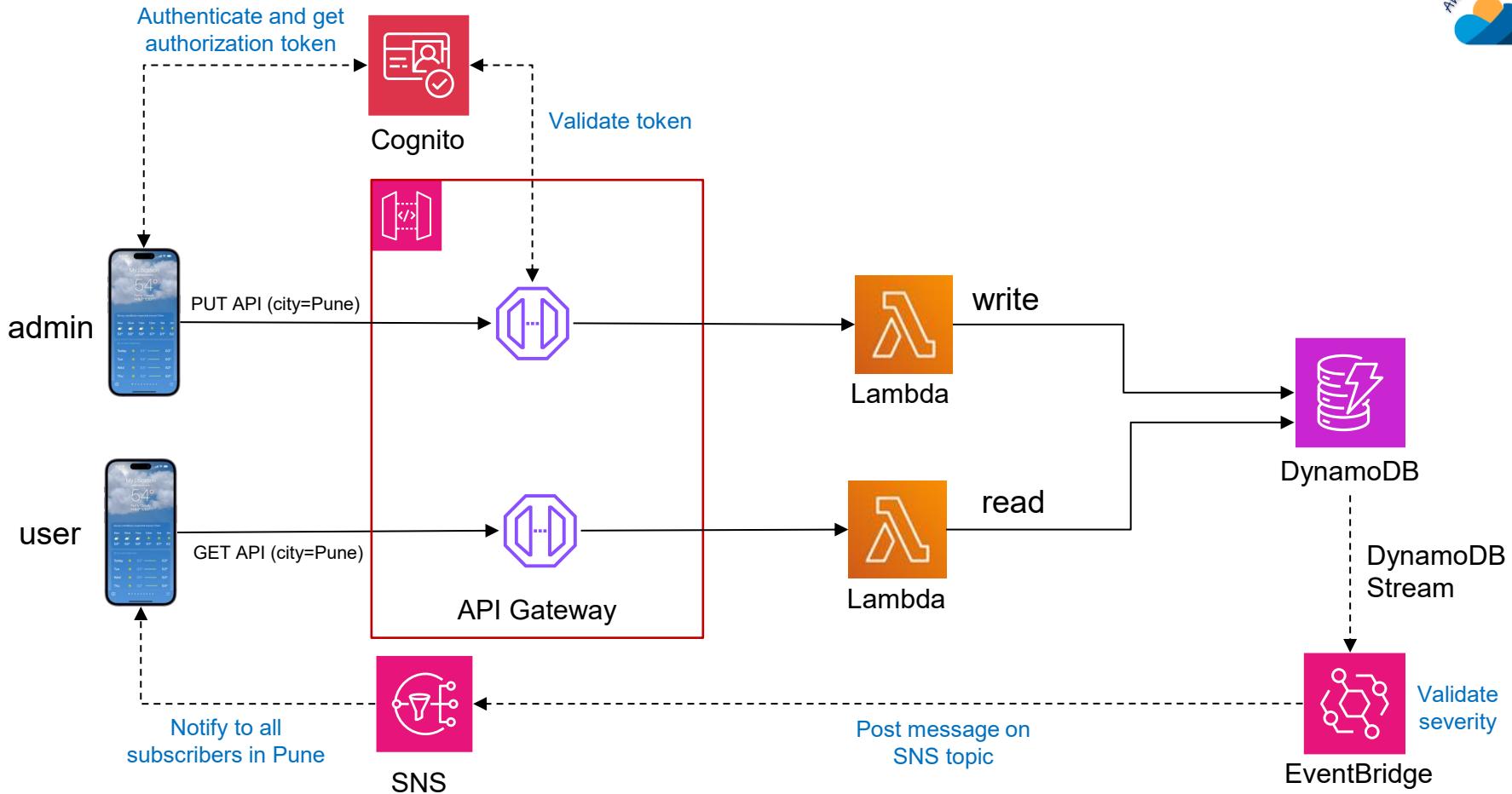


## Amazon EventBridge

Event bus enabling integration between AWS services, SaaS and your applications

# Remember this?



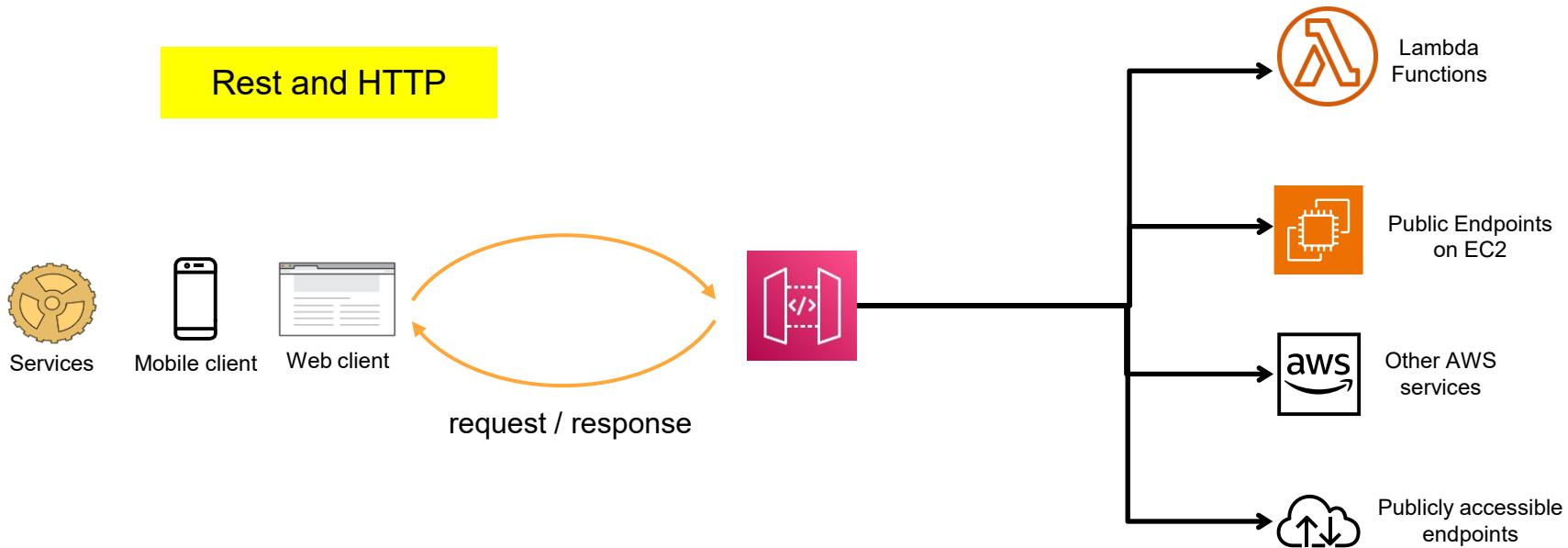


# Amazon API Gateway

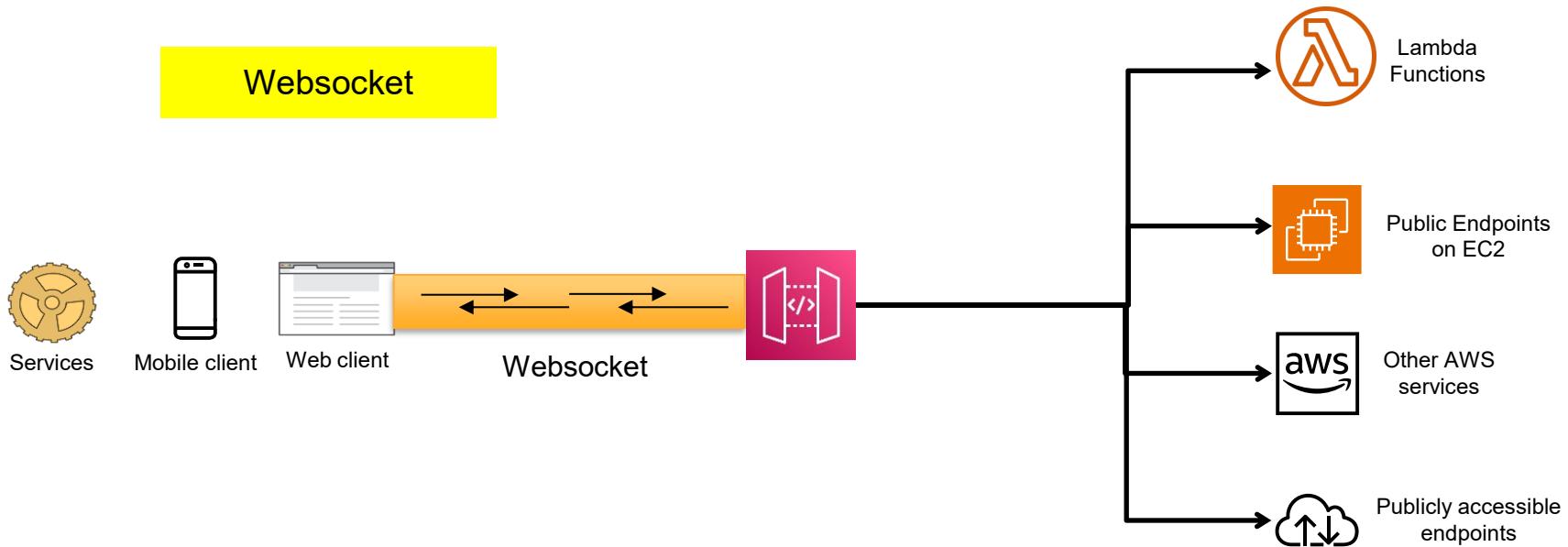
- API Gateway is a fully managed API service
- API gateway provides:
  - RESTful and HTTPS APIs (application programming interfaces)
  - Simple integration with AWS Lambda, Amazon ECS, EKS, Elastic BeanStalk etc. for hosting backend services.
  - Out of the box monitoring and logging that is integrated with Amazon CloudWatch
  - Support for tracking the cost of calls made to your APIs – For charge back to the clients
  - Pricing based on calls made to your APIs
  - Support for security, user authentication, API throttling, API keysetc.



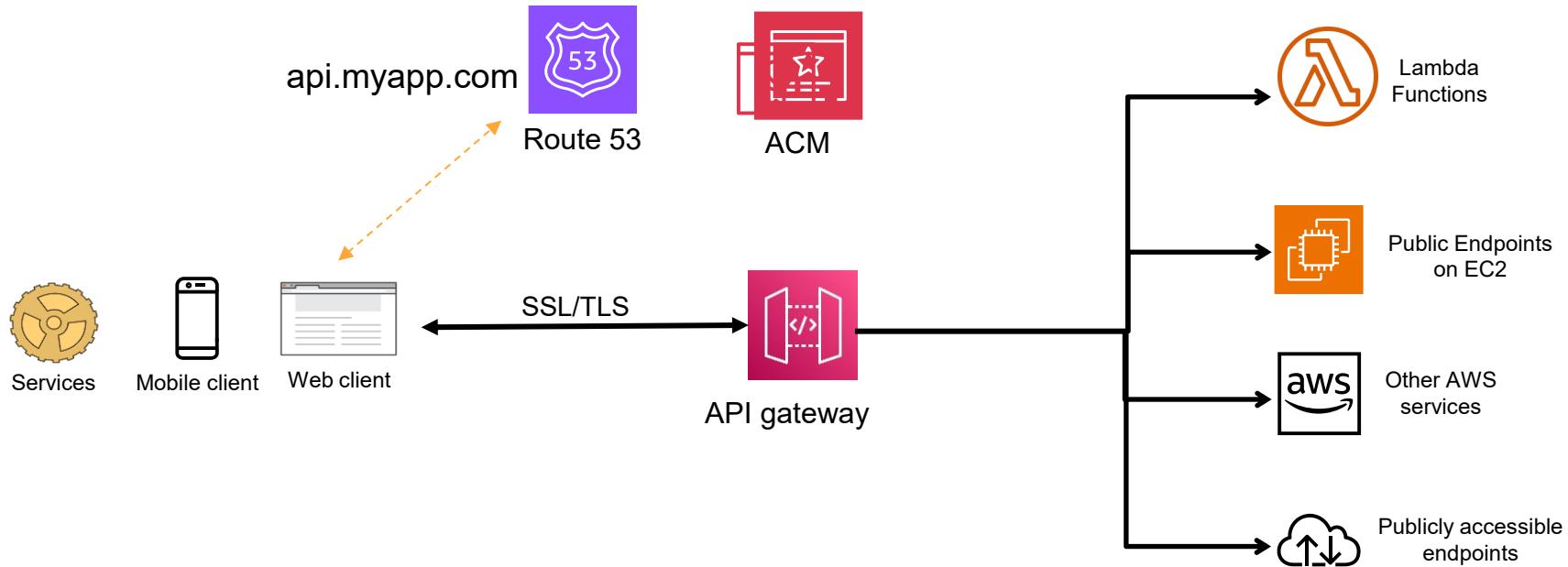
# API Gateway protocols



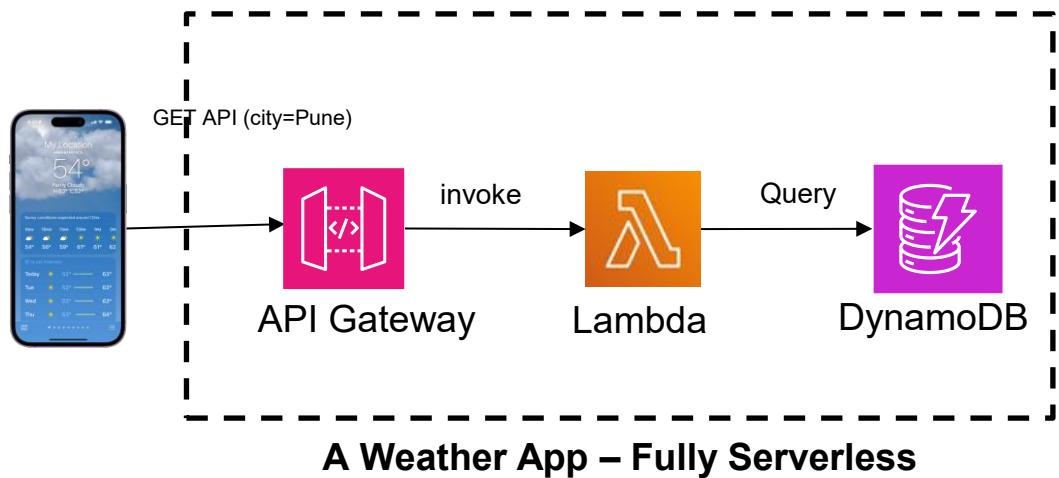
# API Gateway protocols



# API Gateway - Custom domain name



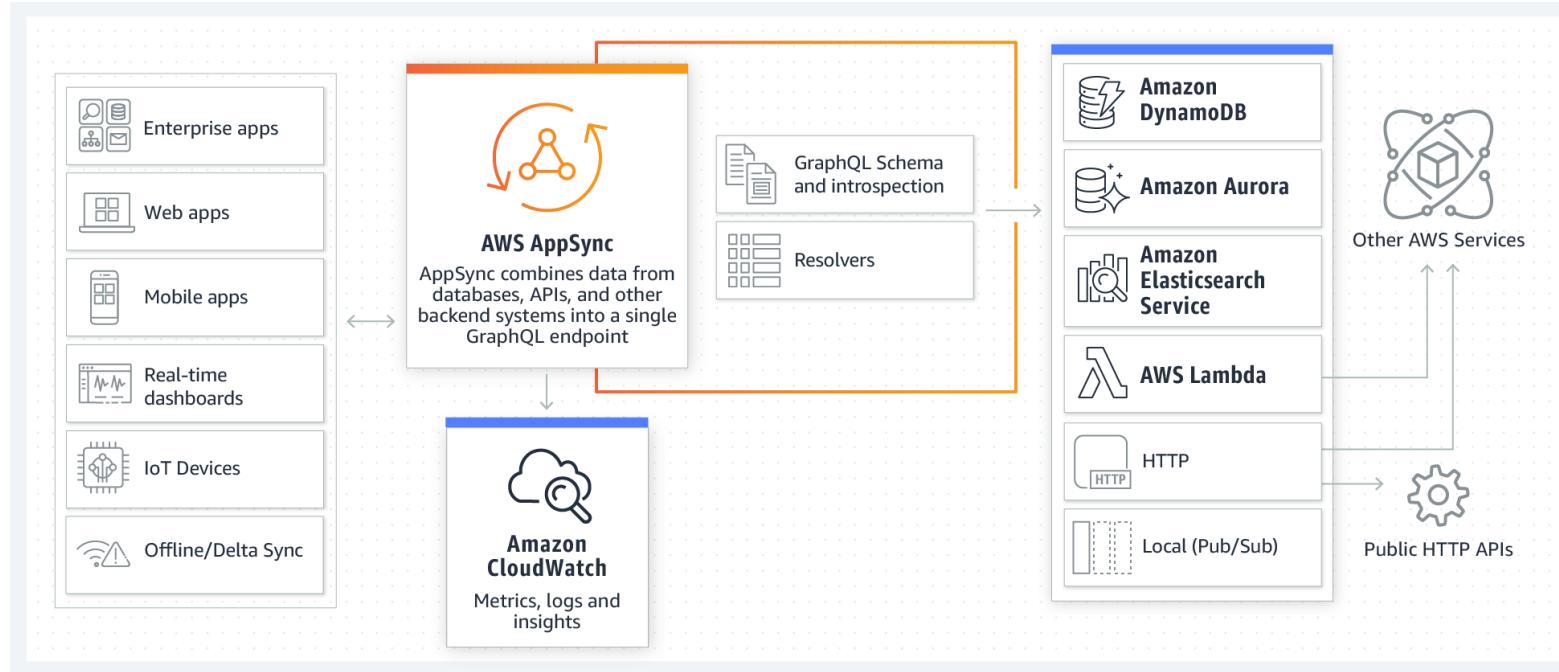
# Exercise – A simple weather app with AWS Lambda



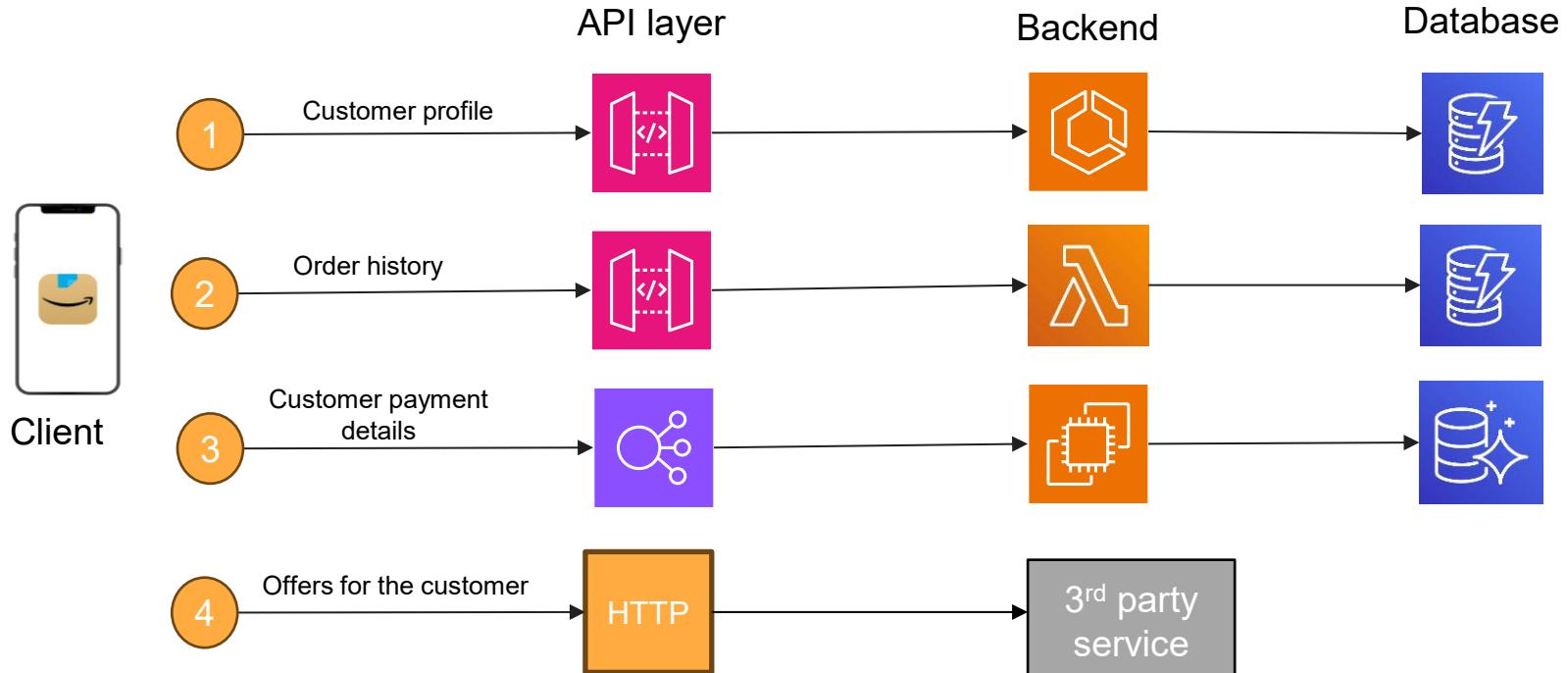
- 1 Create DynamoDB table and add sample items for cities and temperature e.g. Pune (String), 28 (Number)
- 2 Create Lambda function (Python 3.11) and add code provided with this lecture. Update dynamodb table name.
- 3 Create IAM role for Lambda to have Read-only permissions for DynamoDB. Configure Lambda function to use this role and change execution time limit to 1 min.
- 4 Create API Gateway REST API with Any method and invoke Lambda function with proxy integration. Deploy API (use any stage name)
- 5 Create sample HTML web page using the code provided and replace API endpoint with your endpoint.
- 6 Open HTML page and access the simple weather app.

# AWS AppSync

- AWS AppSync connects apps to data with secure, serverless, and performant **GraphQL** and Pub/Sub APIs



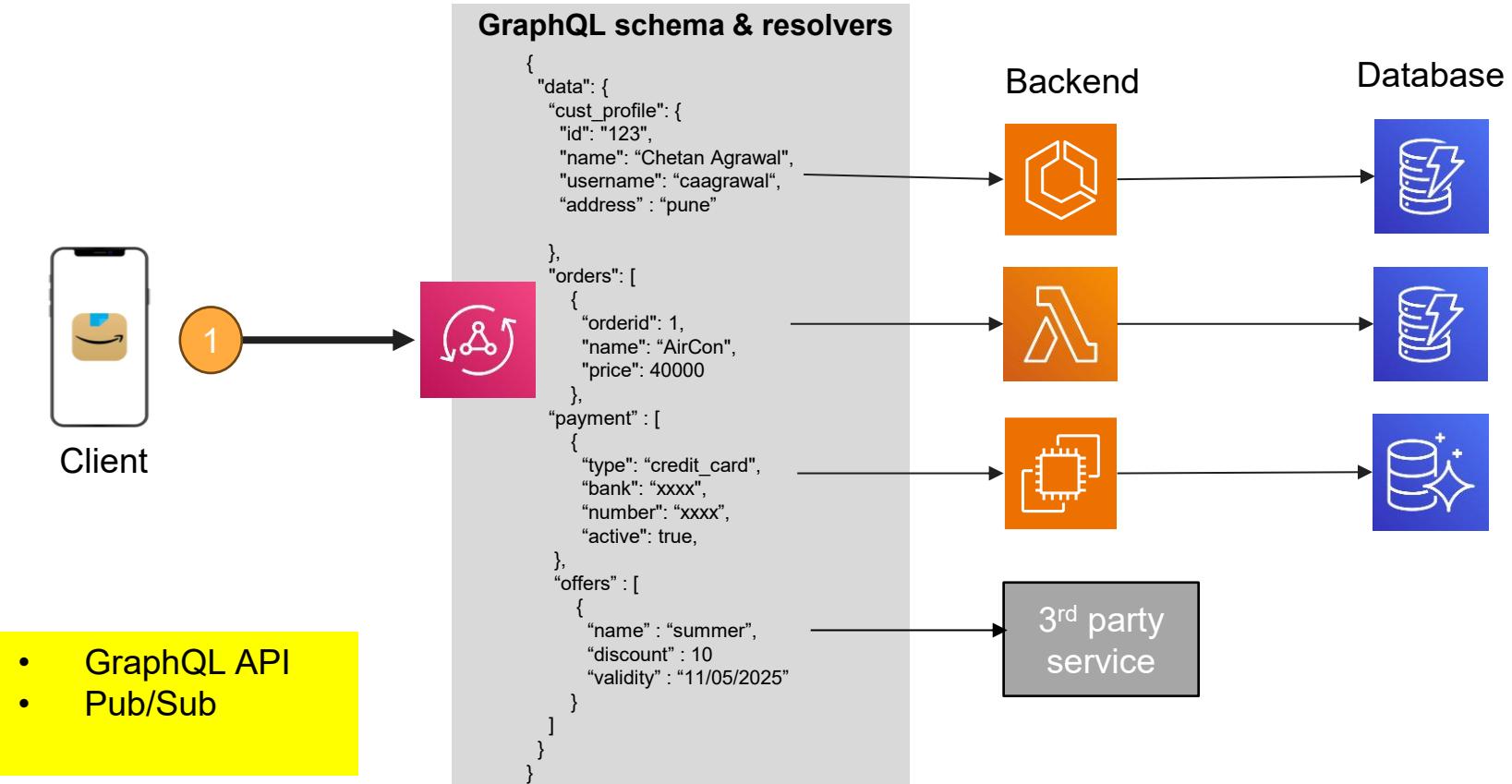
# APIs for sample eCommerce application



✖ Multiple API calls

✖ Client receives full response (un-necessary data)

# AWS AppSync

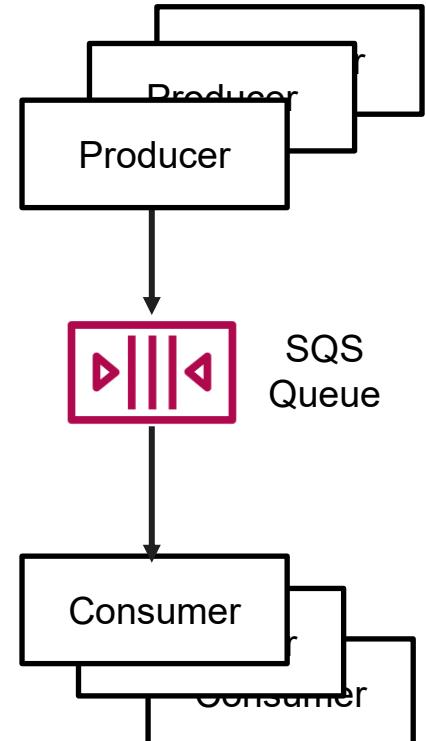


- GraphQL API
- Pub/Sub

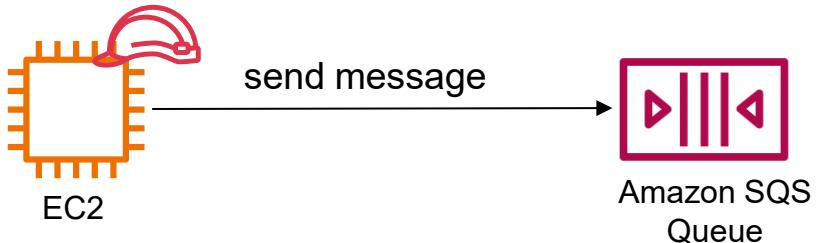


# Amazon SQS - Simple Queue Service

- Amazon SQS is a Highly available distributed message queue system.
- SQS supports two type of queues:
  - Standard (Default)
  - FIFO (First-In-First-Out)
- SQS supports multiple readers and writers for the same queue.
- Locks the messages during the processing using visibility timeout.
- Delay queues to introduce short delays in message delivery.
- Use cases:
  - Decouple components or microservices to increase reliability e.g. Order and Shipping
  - Scale backend system components depending on queue depth
  - Dead-letter queue for storing failed events or transactions



# Exercise – Send message to SQS queue

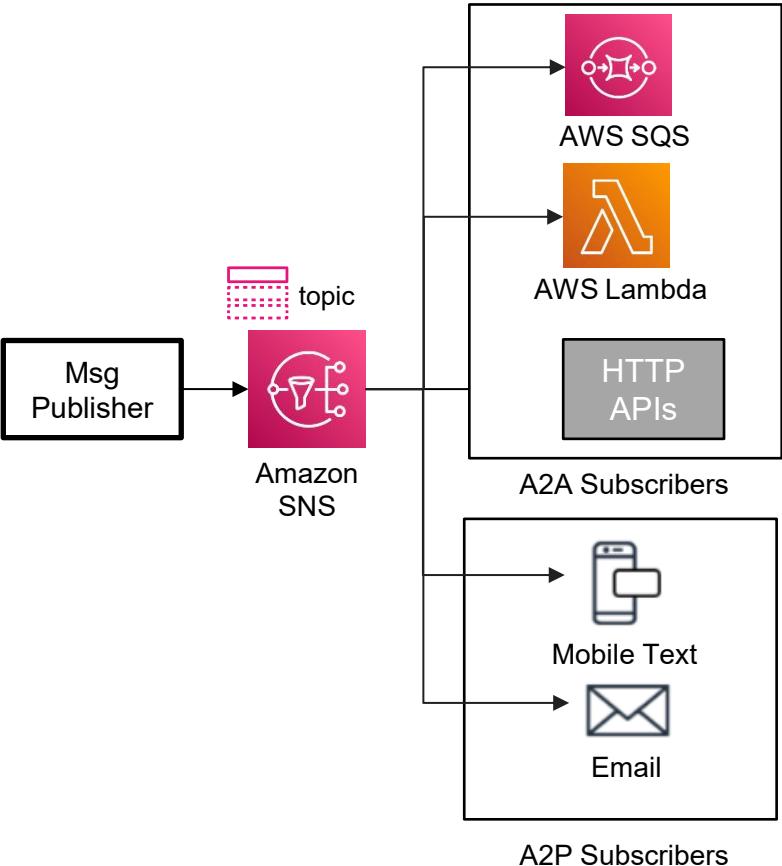


- 1 Create SQS queue (standard queue)
- 2 Create IAM role for EC2 and add IAM policy having sqs:SendMessage permission
- 3 Launch EC2 instance and attach IAM role
- 4 SSH into EC2 instance and run CLI command to send message to the queue
- 5 From SQS console, verify if you see the messages

```
$aws sqs send-message --queue-url <QUEUE_URL> --message-body "Hello, this is test message 1"
```

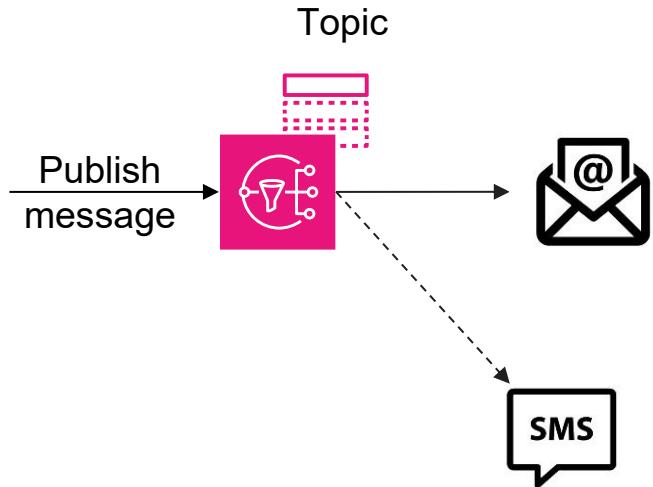
# Amazon SNS - Simple Notification Service

- Amazon Simple Notification Service (SNS) sends notifications two ways:
  - Application to Application (A2A)** – To AWS services such as SQS, AWS Lambda, and other HTTP/S endpoints.
  - Application to Person (A2P)** - Send SMS or Email, mobile push notifications
- Use cases:
  - Send alert notification when Cloudwatch alarms trigger
  - Mobile app push notification
  - Sending same event to multiple downstream applications e.g. As new order is placed, send order details to shipping, billing and inventory systems





# Exercise – Send email/SMS using SNS

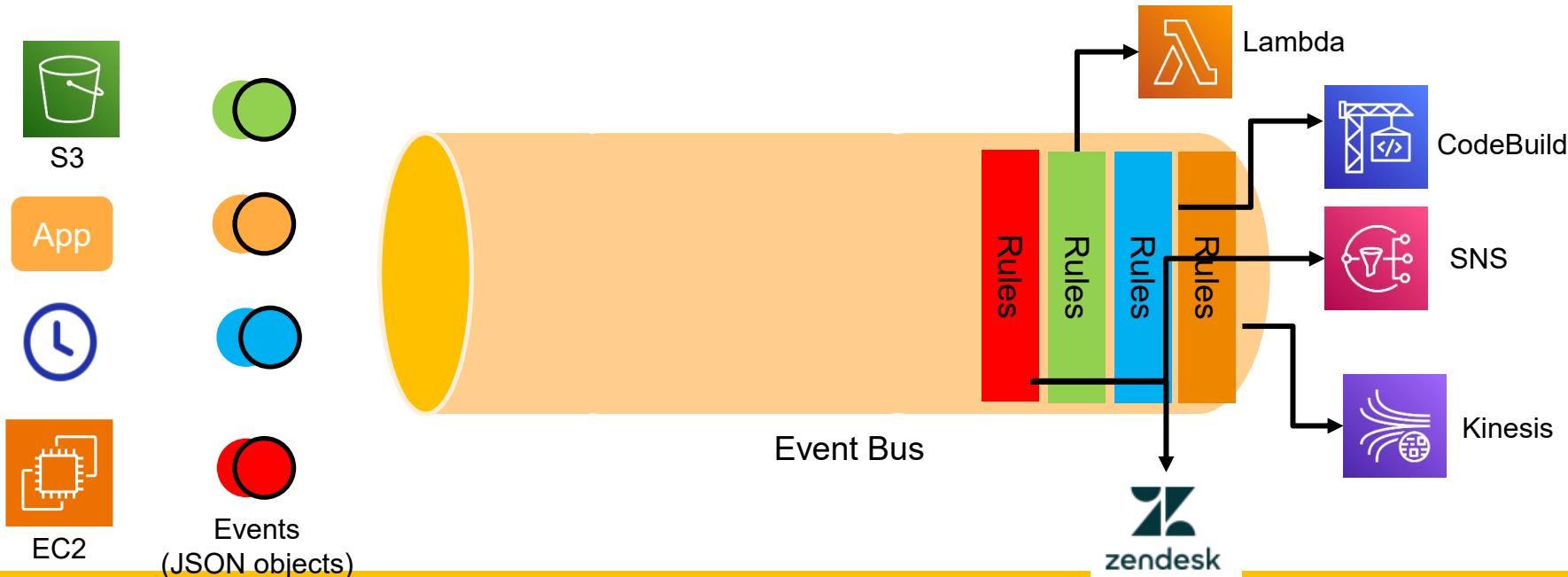


- 1 Create a SNS topic
- 2 Add email subscription for the topic. You will receive confirmation email. Click the link in the email to confirm.
- 3 Add SMS subscription by adding your phone number. You will receive text message with code. Enter code to confirm your subscription.
- 4 Using SNS console, publish a message onto the topic
- 5 Verify if you receive the message over an email

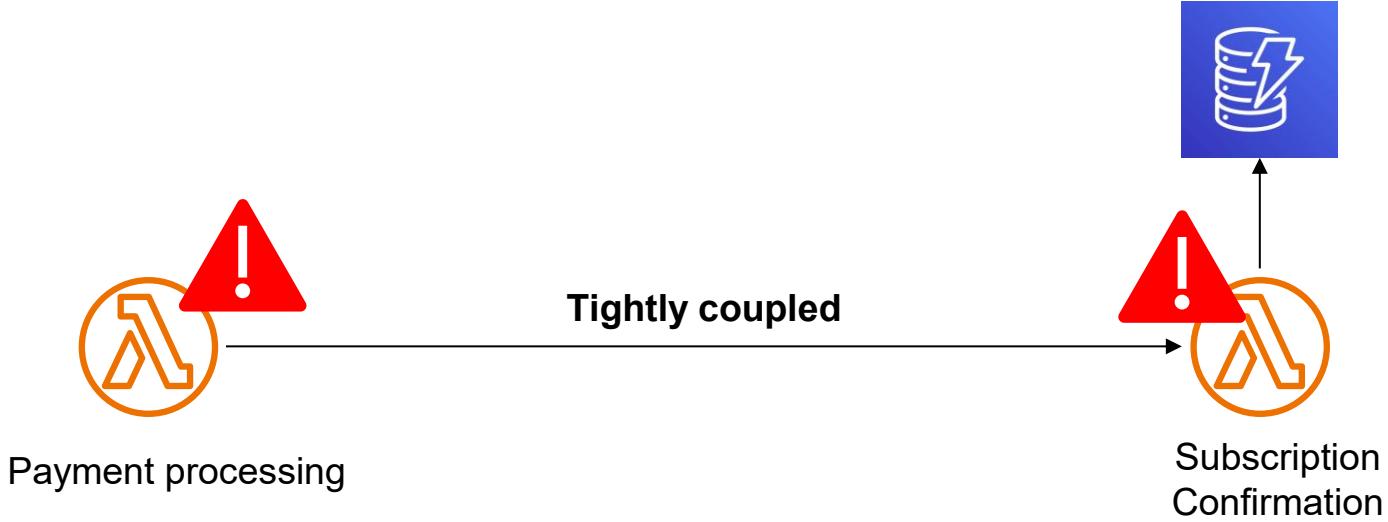
Note: In some countries there are restrictions on sending SMS like SMS are delivered only during specific time of the day or promotional messages are blocked etc.

# Amazon EventBridge

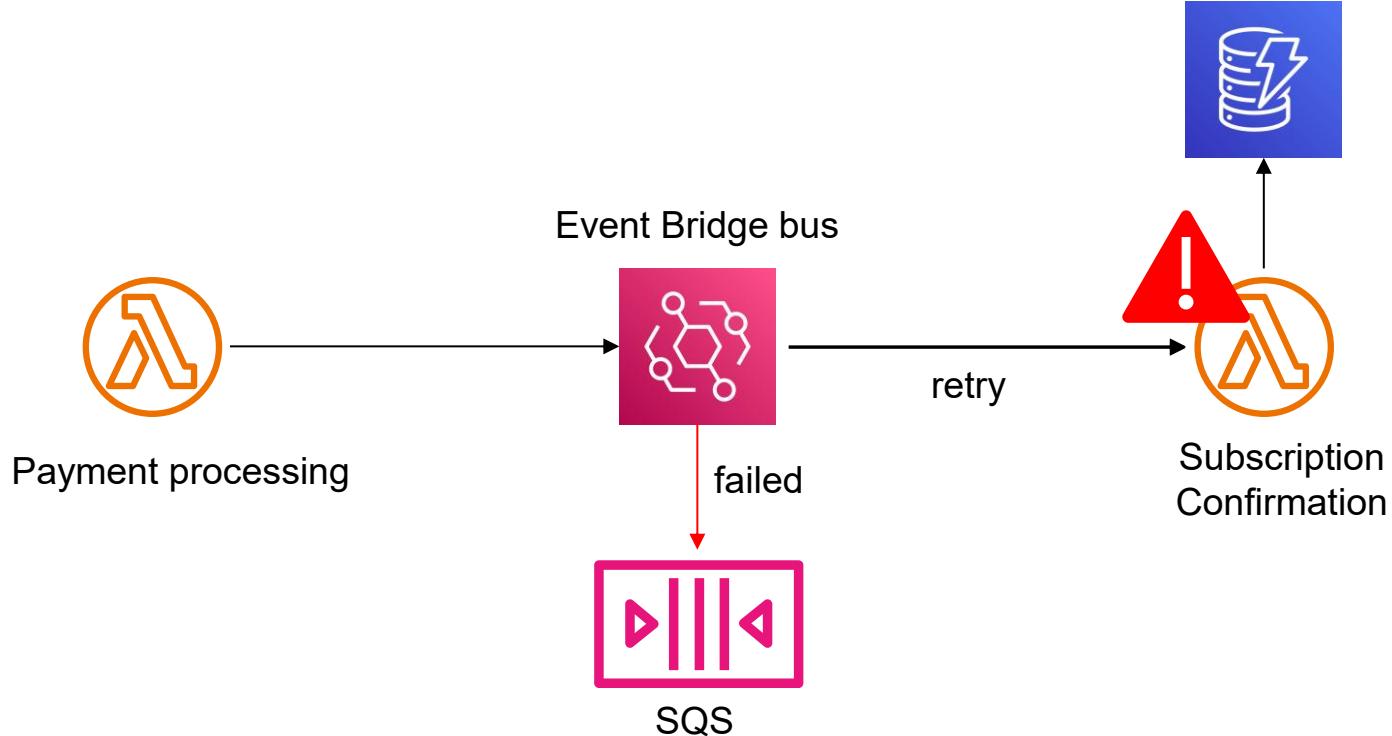
- A Serverless event bus for building event-driven applications enabling loose coupling
- Connects your Custom applications, AWS services and 3<sup>rd</sup> party SaaS applications



# Decoupling services

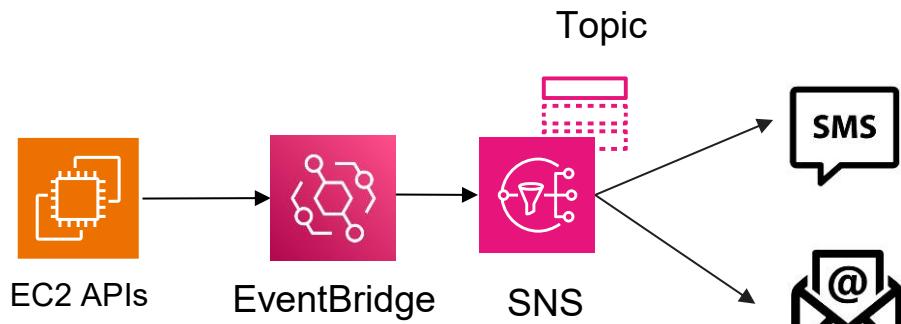


# Decoupling services





# Exercise – Send notification when EC2 instance is launched



- 1 Create EventBridge Rule for EC2-state-change event using default event bus
- 2 Target should be the SNS topic that you created earlier
- 3 Launch a test EC2 instance. Verify if you receive the message over an email or SMS.

Note: In some countries there are restrictions on sending SMS like SMS are delivered only during specific time of the day or promotional messages are blocked etc.



# Amazon MQ

- Amazon MQ is a managed message broker service that supports open-source message broker engines like **ActiveMQ** and **RabbitMQ**
- Enables decoupling for reliable communication between distributed applications and services.
- Amazon MQ is ideal for organizations needing a fully managed message broker that supports existing applications and standard messaging protocols such as MQTT, AMQP, STOMP etc.
- In AWS, the first choice for queue service should be SQS but if you don't want to change application code then you can use Amazon MQ.



# Application Integration Section – Summary

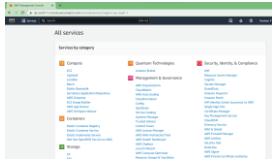
- **Amazon API gateway** - RESTful and HTTPS APIs (application programming interfaces)
  - Supports REST, HTTP and Websocket protocols
  - Supports features like caching, authentication, authorization, API keys, Throttling, Custom domain name etc.
- **AWS AppSync** - Serverless and high-performing GraphQL and Pub/Sub API service
  - Based on a predefined GraphQL schema that defines the data structure and operations.
  - Connects to DynamoDB, Aurora, Elasticsearch, HTTP endpoints, and Lambda functions.

# Application Integration Section – Summary

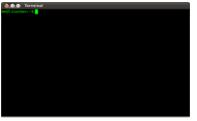
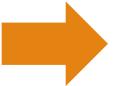
- **Amazon SQS** - Queue service in AWS, supports Standard Queue and FIFO queue
  - Used to decouple applications in AWS
  - Messages can be retained for up to 14 days
- **Amazon SNS** - Notification service in AWS
  - Topic and Subscribers: Email, Lambda, SQS, HTTP, Mobile phone numbers (SMS), Push notifications
  - No message retention
- **Amazon EventBridge** – Event bus for event-driven applications
  - Allow to connect applications based on the events
  - Event filters and rules to process an event before sending it to the target
- **Amazon MQ** - Managed message broker for ActiveMQ and RabbitMQ in the cloud

# AWS Infrastructure and Application deployment

# Ways to deploy AWS infrastructure



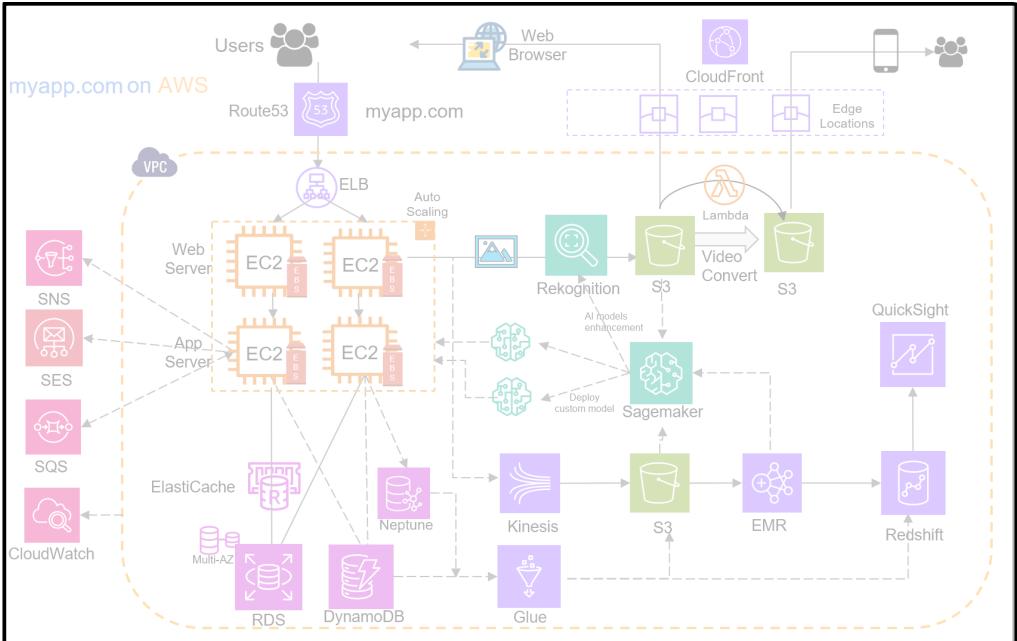
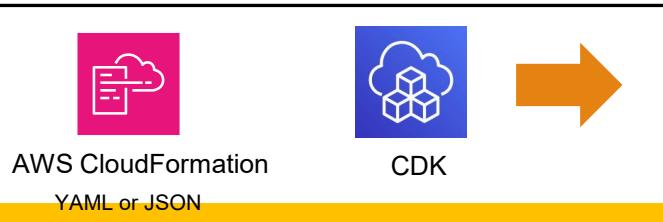
AWS Console



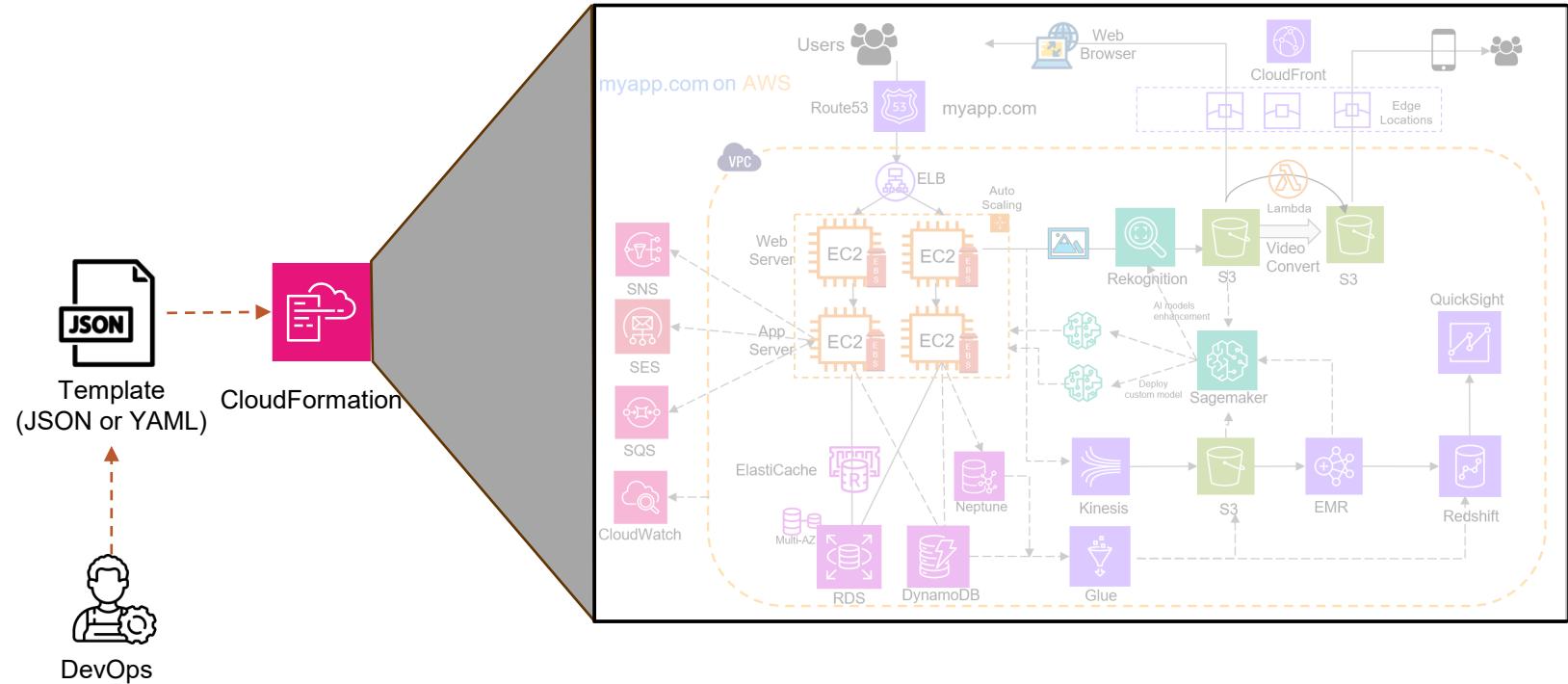
AWS CLI



C++, go, java, Java script, Kotlin, .Net, Node.js, PHP, python, ruby, rust, swift



# Using AWS CloudFormation

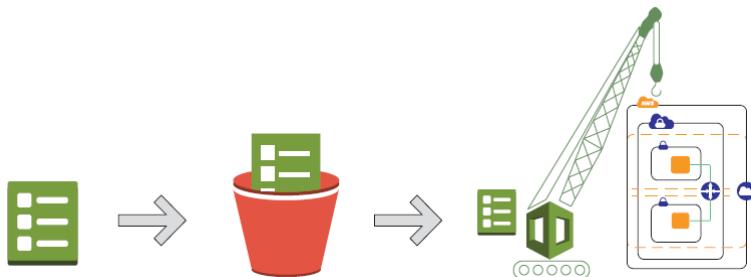


# AWS CloudFormation

AWS CloudFormation is a declarative way of outlining your AWS Infrastructure for any resources.

For example, within a CloudFormation template, you say:

- ✓ I want a VPC and Subnets
- ✓ I want an internet gateway and attach it to the VPC
- ✓ I want a security group
- ✓ I want two EC2 machines using this security group in the subnet just created



Then CloudFormation creates those for you, in the **right order**, possibly **in parallel** and with the **exact configuration** that you specify.

# Benefits of AWS CloudFormation

- AWS resources are created & deleted automatically, hence no manual errors
- Resources are created and deleted as a group (called Stack), hence no ghost resources on stack deletion
- AWS resources which are not dependent are created in parallel which speeds up infrastructure deployment significantly.
- Same CloudFormation template can be deployed in different AWS regions or accounts with minimal or no changes. Templates can be easily shared among teams.
- The template can be **version controlled** (using git, svn etc.) so it's easy to go back to previous deployment
- Supports many features including conditions e.g. use t2.micro for development environment but m5.large for production environment

# CloudFormation template examples

```

AWSTemplateFormatVersion: '2010-09-09'
Description: Launch an EC2 instance with SSH access

Resources:
  EC2Instance:
    Type: 'AWS::EC2::Instance'
    Properties:
      InstanceType: t2.micro
      KeyName: !Ref KeyName
      ImageId: ami-0c55b159cbfafe1f0 # Replace with your AMI ID
      SecurityGroups:
        - !Ref InstanceSecurityGroup

  InstanceSecurityGroup:
    Type: 'AWS::EC2::SecurityGroup'
    Properties:
      GroupDescription: Allow SSH
      SecurityGroupIngress:
        - IpProtocol: tcp
          FromPort: 22
          ToPort: 22
          CidrIp: 0.0.0.0/0

Parameters:
  KeyName:
    Description: Name of an existing EC2 KeyPair
    Type: String

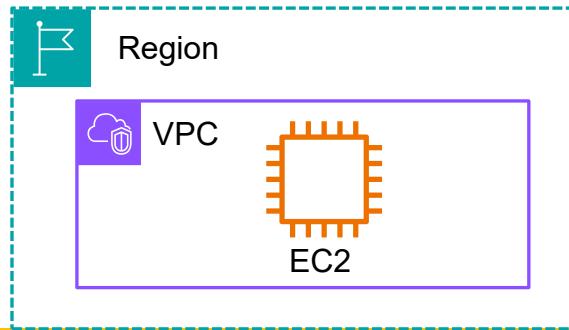
Outputs:
  InstanceId:
    Value: !Ref EC2Instance
  PublicIP:
    Value: !GetAtt EC2Instance.PublicIp
  
```



Upload to S3 bucket

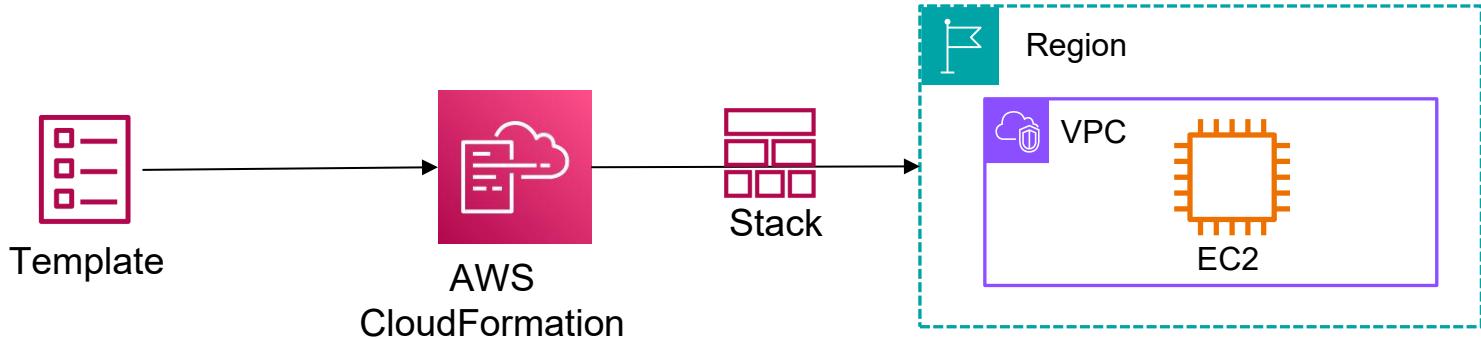


Create Stack



# Exercise - Launch EC2 instance using CloudFormation

1. Download a CloudFormation template: <https://github.com/awswithchetan/aws-cloud-practitioner/blob/main/cloudformation/ec2.yaml>
2. Replace the values in the templates such as Region, ssh-key-pair, AMI ID
3. Use updated template to create a CloudFormation stack



# AWS Cloud Development Kit (CDK)

An open-source software development framework to define your cloud application resources using familiar programming languages.



cdk init

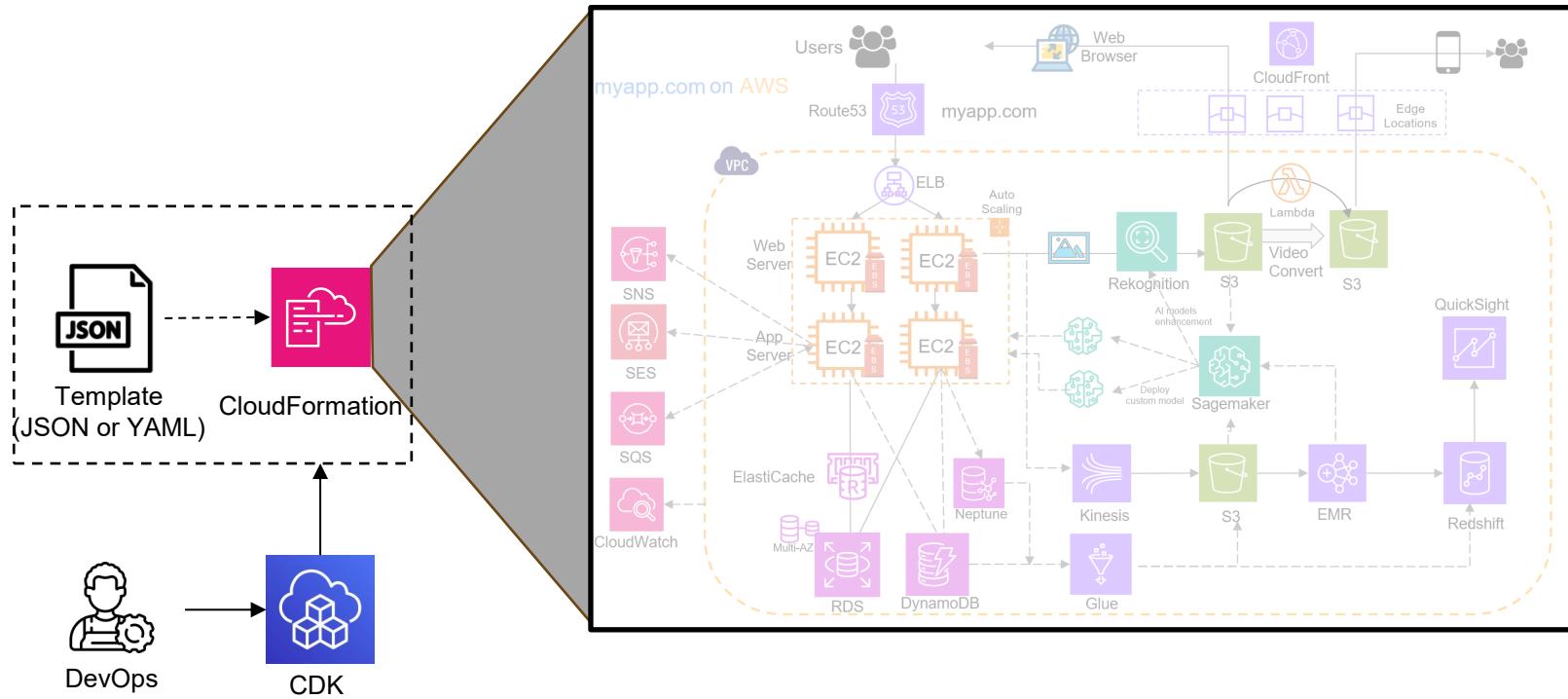
npm run build

cdk synth

cdk diff

cdk deploy

# Using AWS CDK



TS

node

Write code in  
language of choice

# CDK code example

```
from aws_cdk import (
    aws_ec2 as ec2,
    core
)

class EC2InstanceStack(core.Stack):
    def __init__(self, scope: core.Construct, id: str, **kwargs) -> None:
        super().__init__(scope, id, **kwargs)

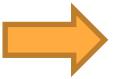
        # VPC where the instance will be launched
        vpc = ec2.Vpc.from_lookup(self, "VPC", is_default=True)

        # Security Group to allow SSH access
        security_group = ec2.SecurityGroup(self, "InstanceSG",
            vpc=vpc,
            description="Allow SSH access",
            allow_all_outbound=True
        )
        security_group.add_ingress_rule(
            peer=ec2.Peer.any_ipv4(),
            connection=ec2.Port.tcp(22),
            description="Allow SSH access from anywhere"
        )
        instance = ec2.Instance(self, "EC2Instance",
            instance_type=ec2.InstanceType("t2.micro"),
            machine_image=ec2.MachineImage.latest_amazon_linux(),
            vpc=vpc,
            security_group=security_group,
            key_name="your-key-pair-name" # Replace with your key pair name
        )

        # Output the instance ID and public IP
        core.CfnOutput(self, "InstanceId", value=instance.instance_id)
        core.CfnOutput(self, "InstancePublicIP", value=instance.instance_public_ip)

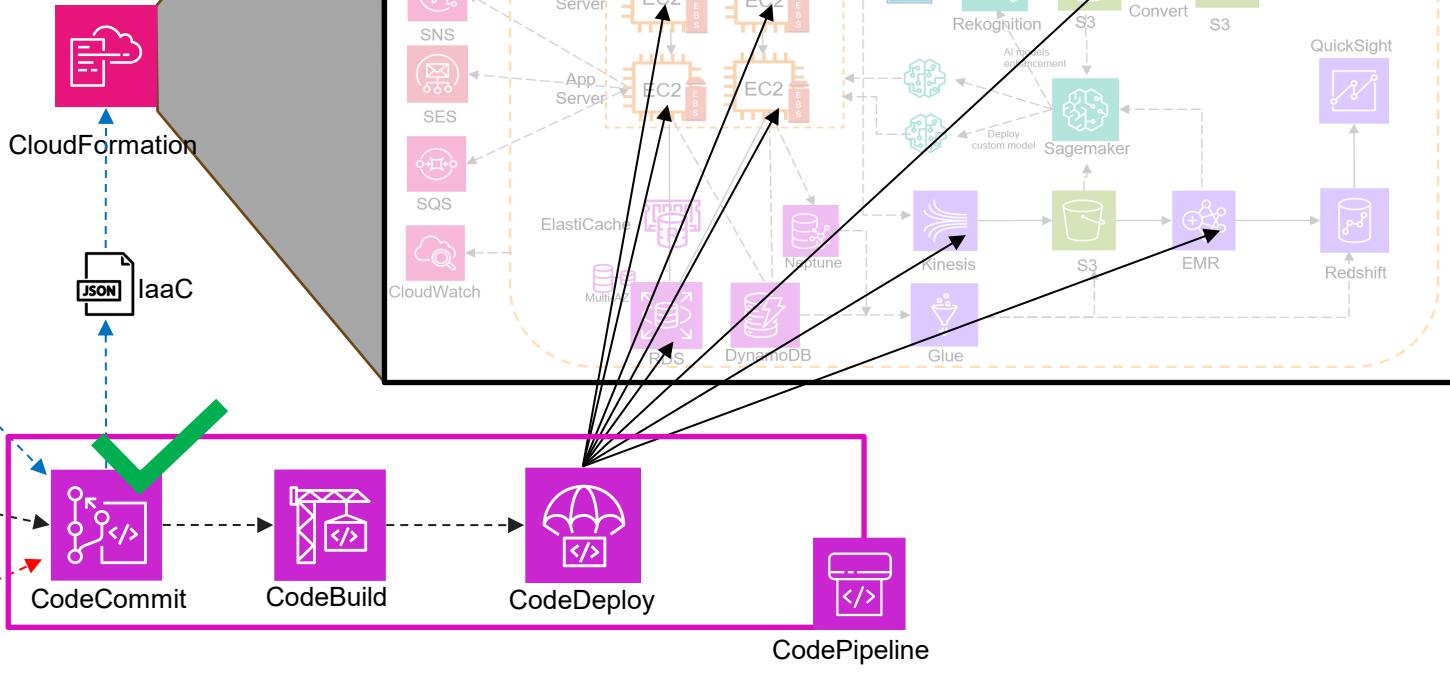
app = core.App()
EC2InstanceStack(app, "EC2InstanceStack")
app.synth()
```

ec2-instance-cdk.py



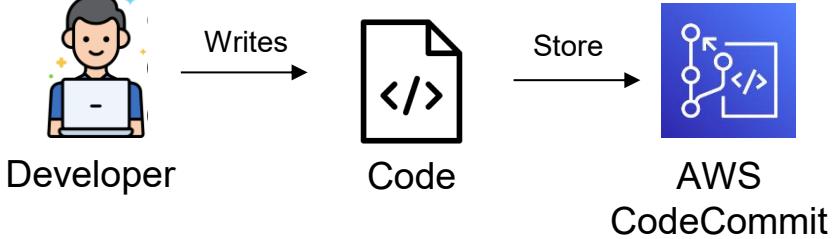
```
$npm install -g aws-cdk
$mkdir ec2-instance-cdk
$cd ec2-instance-cdk
$cdk init app --language python
$pip install aws-cdk.aws-ec2
$cdk deploy
```

# AWS DevOps Services



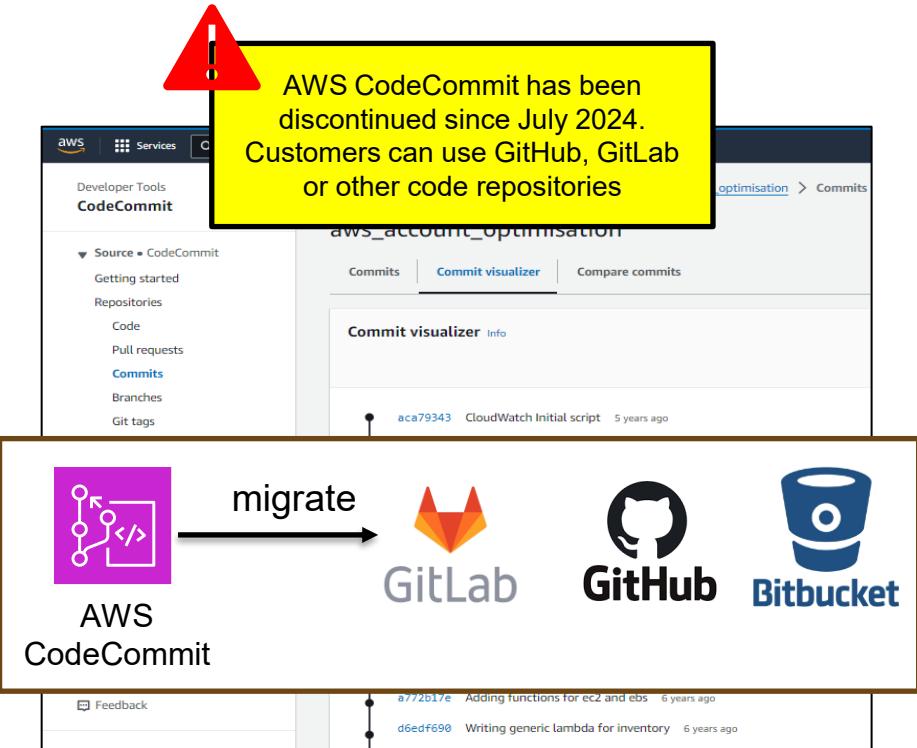
# AWS CodeCommit

- Before pushing the application code to servers, it needs to be stored somewhere.
- Developers usually store code in a repository, using the Git technology.
- **CodeCommit**



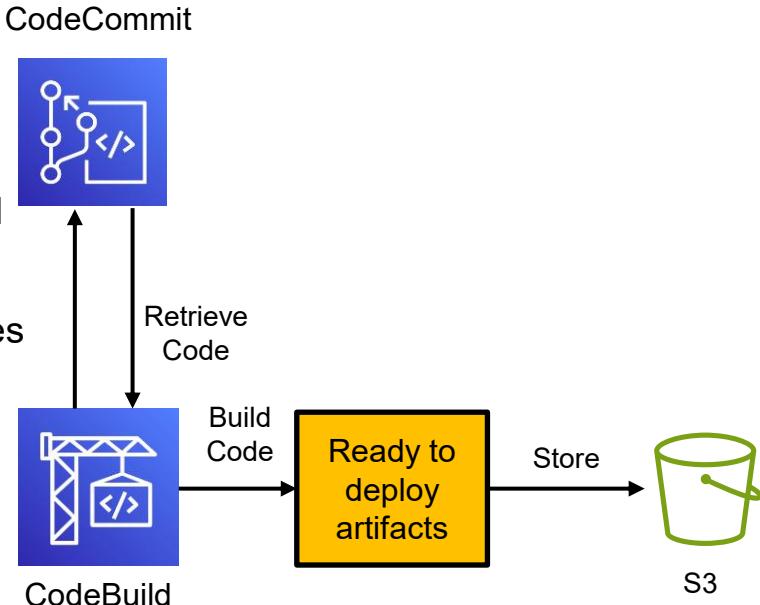
# AWS CodeCommit

- Before pushing the application code to servers, it needs to be stored somewhere.
- Developers usually store code in a repository, using the Git technology.
- **CodeCommit**
  - Securely host highly scalable private Git repositories.
  - Makes it easy to collaborate with others on code.
  - Serverless version control service.



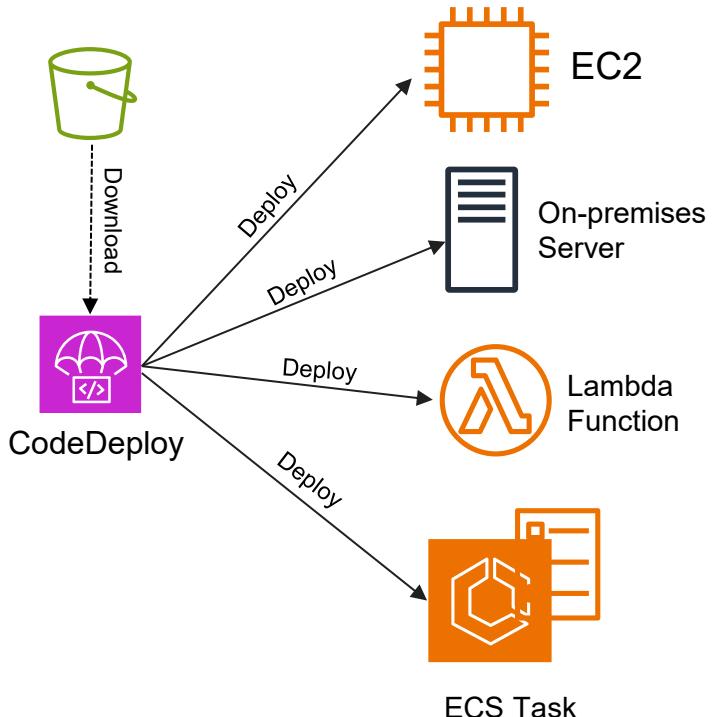
# AWS CodeBuild

- Code Building service in AWS
- Provides pre-configured Build environment that contains Operating System, programming language runtime and build tools such as Apache maven, gradle, npm etc.
- Compiles source code, run unit tests, and produces packages that are ready to deploy.
- Benefits:
  - Fully managed, serverless
  - Continuously scalable & highly available
  - Pay-as-you-go pricing – only pay for the build time



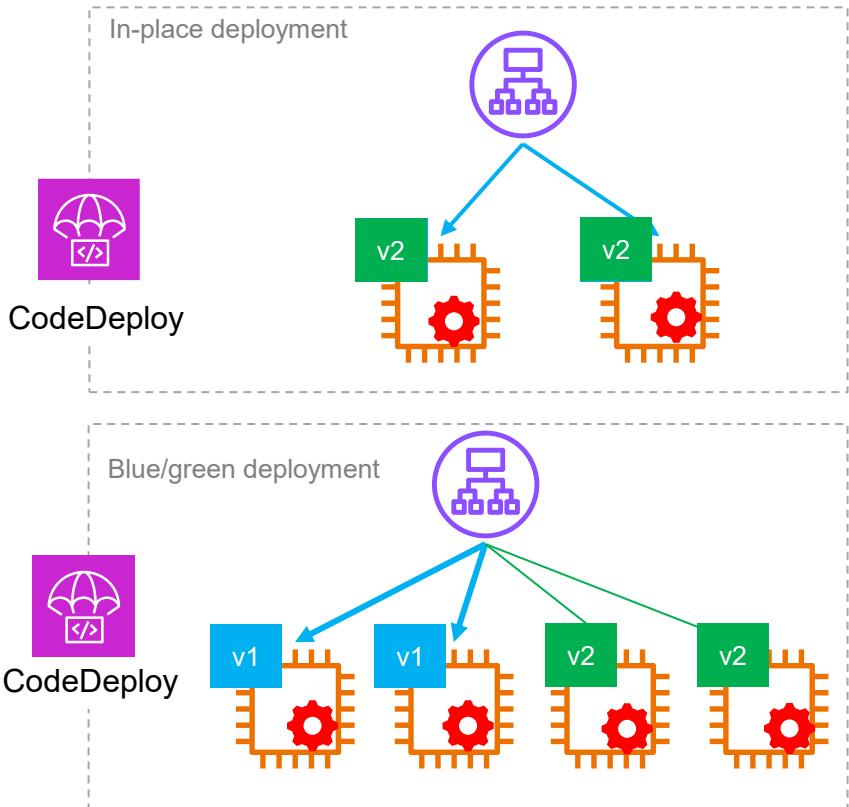
# AWS CodeDeploy

- Automates application deployments.
- You can deploy variety of application content:
  - Code
  - Web and configuration files
  - Executables
  - Packages and Scripts
- Works with EC2 Instances, On-Premises Servers, AWS Lambda and Amazon ECS



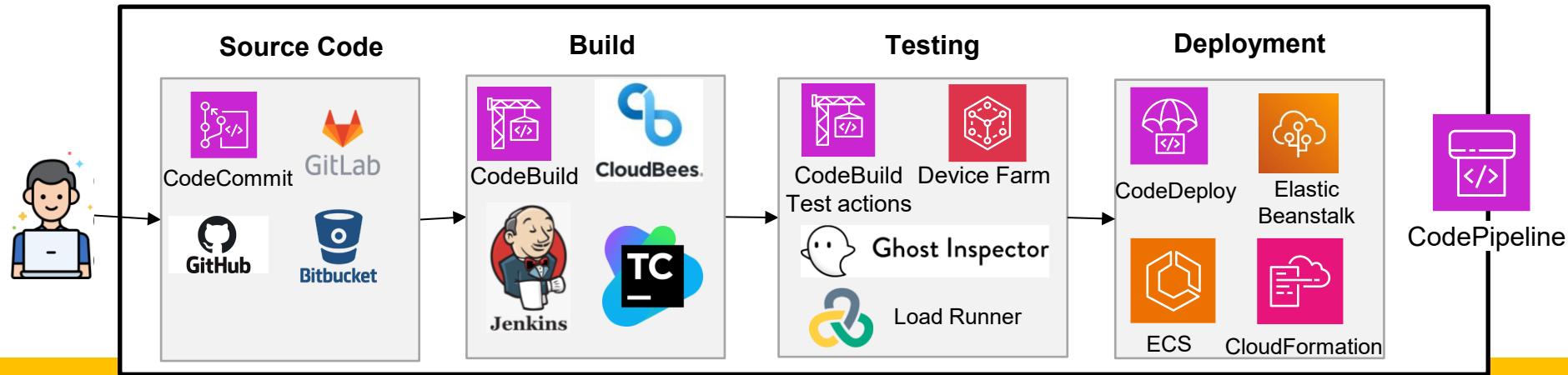
# AWS CodeDeploy

- Automates application deployments.
- You can deploy variety of application content:
  - Code
  - Web and configuration files
  - Executables
  - Packages and Scripts
- Works with EC2 Instances, On-Premises Servers, AWS Lambda and Amazon ECS
- Supports **in-place** deployment and **Blue/green** deployment with traffic routing options like Canary, linear or all-at-once
- Must have **CodeDeploy agent** installed on EC2 or on-premises servers



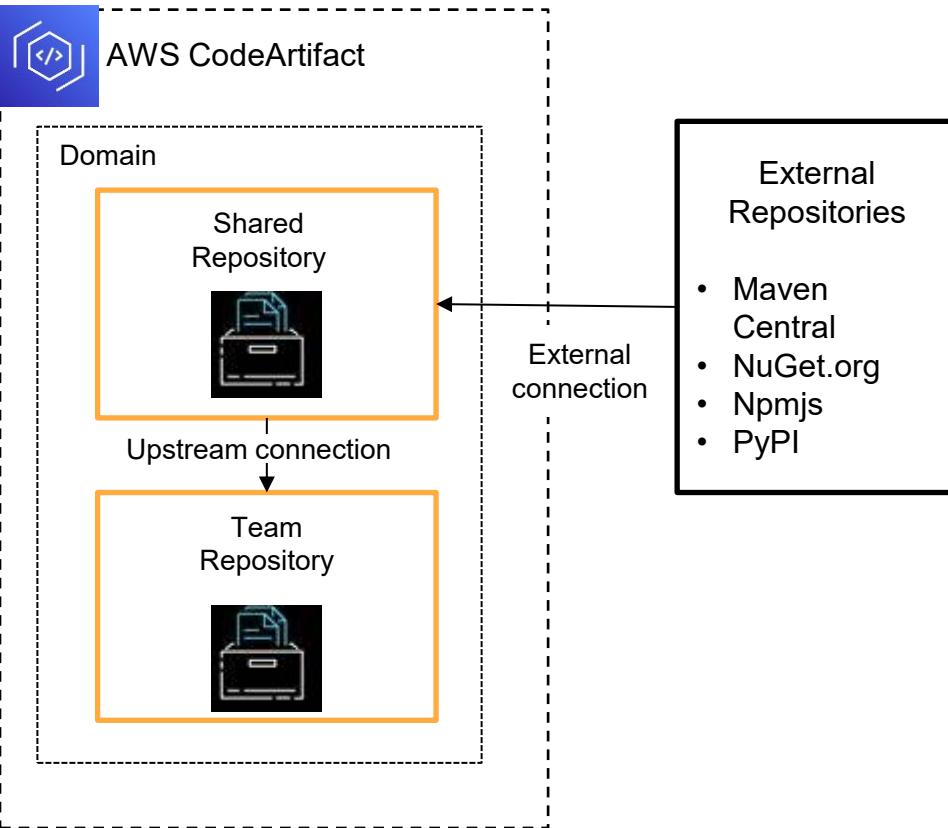
# AWS CodePipeline

- Continuous Integration and Delivery (CI/CD) service for setup, visualize, and automate the steps required for **software release**.
- Connects different stages of CI/CD to push changes to Production automatically
  - Code => Build => Test => Approval => Deploy
- Works with different AWS services (CodeCommit, CodeBuild, CodeDeploy, Elastic Beanstalk, CloudFormation, Amazon S3) and 3<sup>rd</sup> party tools for the deployment to non-AWS platforms.



# AWS CodeArtifact

- CodeArtifact is a secure, scalable, and cost-effective artifact management service for software development.
- Works with common dependency management tools such as Maven, Gradle, npm, yarn, twine, pip, and NuGet.
- Developers and CodeBuild can then retrieve dependencies straight from CodeArtifact.



# AWS Infrastructure deployment & DevOps - Summary

## Infrastructure deployment automation

- CloudFormation – Infrastructure as a code (YAML or JSON templates)
- CDK – Cloud Development Kit to create, share, deploy CloudFormation template programmatically



DevOps

## DevOps CI/CD services

---

- CodeCommit – Git compatible source code depository *[discontinued from July'2024]*
- CodeBuild – Compile source code and run tests
- CodeDeploy – Deploy the code builds onto the compute servers (EC2, on-premises, ECS, Lambda)
- CodePipeline – End to end CI/CD pipeline automation
- CodeArtifact – Software repository / package management for software development and build



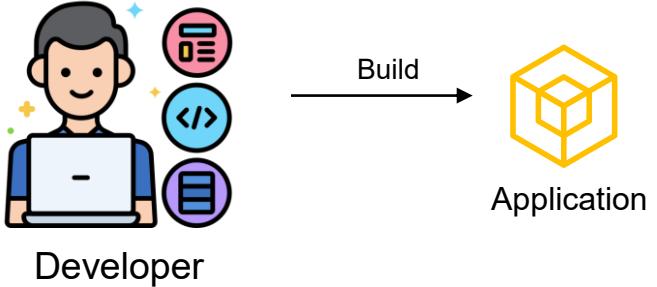
Developer



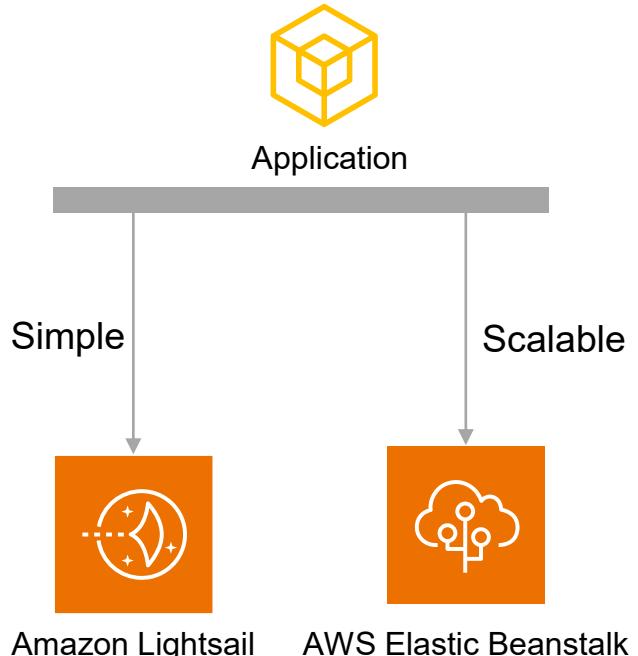
DevOps

# Application Deployment

# What developers want?



# What developers want?



# Amazon Lightsail

- For everything you need to jumpstart your project on AWS.
- Quickly deploy popular applications
  - WordPress, LAMP, Nginx, MEAN, Node.js etc.
- Create preconfigured Virtual Private Server (VPS) or Database or Container service
- Use Cases:
  - Simple web applications
  - Websites (templates for WordPress, Magento, Plesk, Joomla)
  - Dev / Test environment
- Has high availability but no auto-scaling
- Fixed price as per selected bundle

The screenshot shows the Amazon Lightsail blueprint selection interface. At the top, there's a large orange icon with a speaker symbol. Below it, two main sections are visible: "Select a platform" and "Select a blueprint".

**Select a platform:**

- Linux/Unix  
26 blueprints
- Microsoft Windows  
6 blueprints

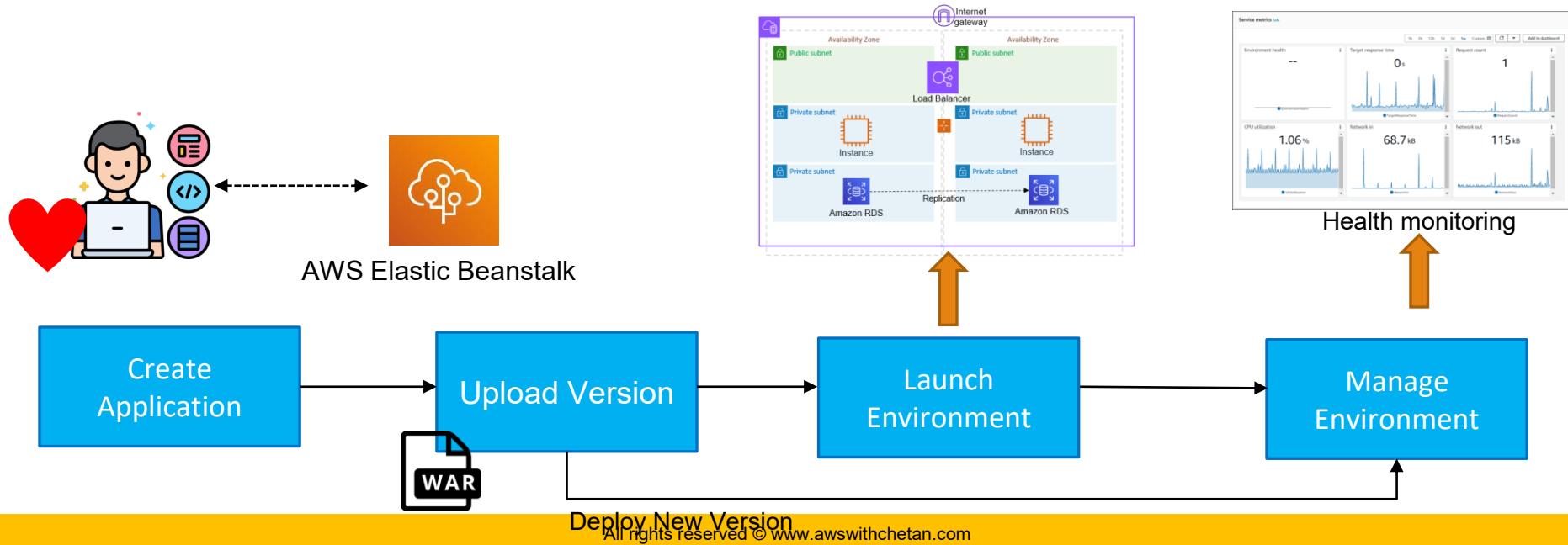
**Select a blueprint:**

Under the "Apps + OS" tab, the following blueprints are listed:

- WordPress 6.6.1-0
- WordPress Multisite 6.6.1-0
- LAMP (PHP 8) 8.3.9-2
- Node.js 18.20.4-0
- Joomla 5.1.2-1
- Magento 2.4.7-4
- MEAN 7.0.12-3
- Drupal 10.3.1-2
- GitLab CE 16.11.5-ce.0-0
- Redmine 5.1.3-1
- Nginx 1.25.5-2
- Ghost 5.88.3-0
- Django 4.2.14-2
- PrestaShop 8.1.7-1
- Plesk Hosting Stack on Ubuntu (BYOL) 18.0.62
- cPanel & WHM for AlmaLinux RELEASE Tier

# AWS Elastic Beanstalk

- A Platform-as-a-service (PaaS) to easily deploy, scale and monitor 3-tier web applications
- Underlying infrastructure is provisioned and managed Elastic Beanstalk (EB)
- Developer friendly 😊



# AWS Elastic Beanstalk - Platform Support

Application frameworks	Container platforms	AWS services
<ul style="list-style-type: none"><li>■ Go</li><li>■ Java SE</li><li>■ Java with Tomcat</li><li>■ .NET on Windows Server with IIS</li><li>■ Node.js</li><li>■ PHP</li><li>■ Python</li><li>■ Ruby</li></ul>	<ul style="list-style-type: none"><li>■ Docker</li><li>■ ECS</li></ul>	<ul style="list-style-type: none"><li>■ EC2</li><li>■ Elastic Load Balancer</li><li>■ Auto-scaling Group</li><li>■ CloudWatch</li><li>■ RDS</li><li>■ Route 53</li><li>■ CloudFront</li><li>■ More..</li></ul>

# Infra, Application and DevOps - summary

Application Deployment



AWS  
Elastic Beanstalk



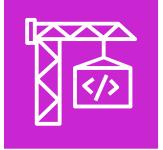
Amazon Lightsail

---

DevOps – CI/CD



CodeCommit



CodeBuild



CodeDeploy



CodePipeline

---

Infrastructure Deployment



AWS CloudFormation



AWS CDK

# Infra, Application and DevOps - summary

## Infrastructure deployment services

- CloudFormation – Infrastructure as a code (YAML or JSON templates)
- CDK – Cloud Development Kit to create CloudFormation templates programmatically

## DevOps - CI/CD services

- CodeCommit – Git compatible source code depository
- CodeBuild – Compile source code and run unit tests
- CodeDeploy – Deploy the code builds onto the compute servers (EC2, ECS, on-premises)
- CodePipeline – End to end CI/CD orchestration pipeline
- CodeArtifact – Package repository for software development and build

## Application deployment services

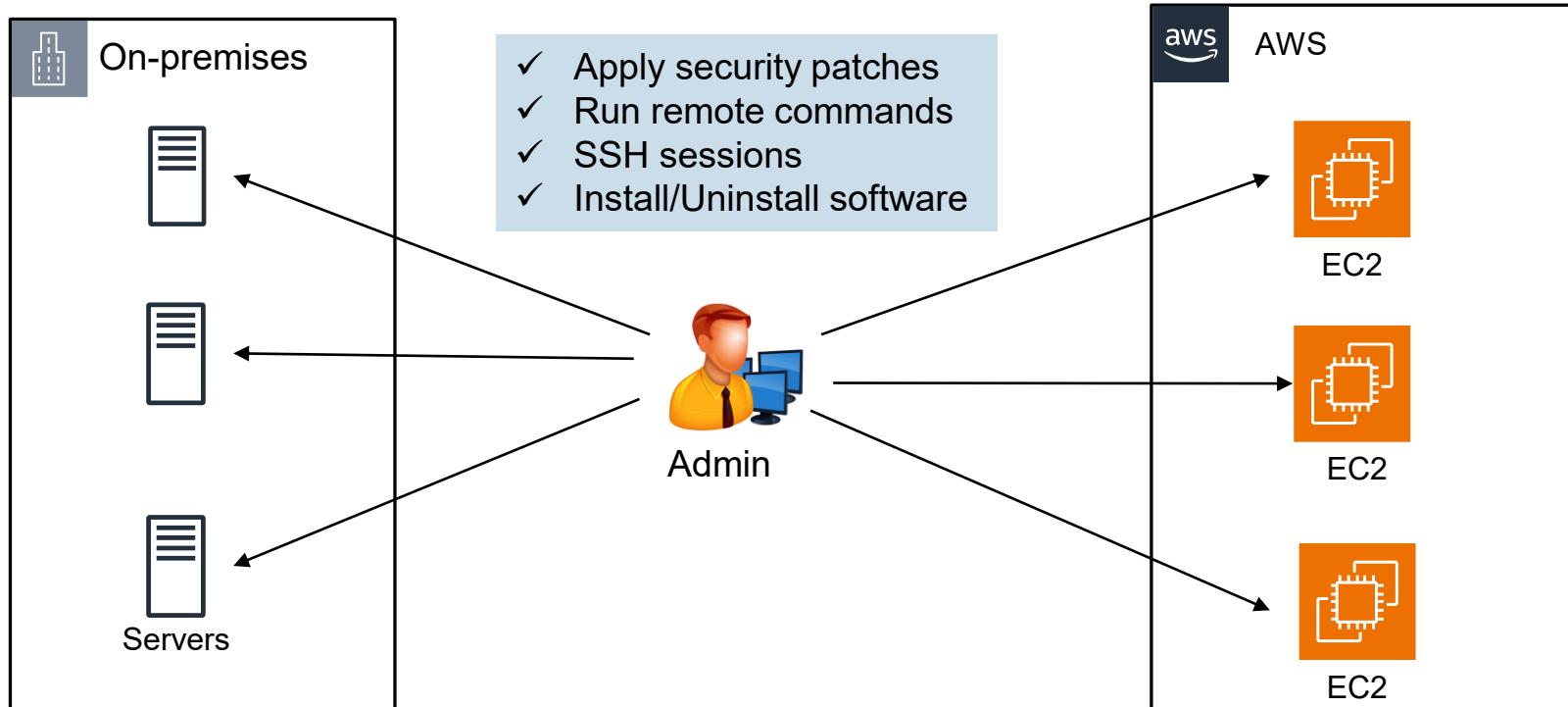
- Elastic Beanstalk – Fully managed platform-as-a-service (PaaS) for 3-tier web applications (ALB, ASG, EC2). Developer's friendly where no-infrastructure management needs to be done.
- Lightsail – Preconfigured servers with fix capacity and price for simple applications, databases and containers.

# Infrastructure Management and Compliance



# AWS Systems Manager

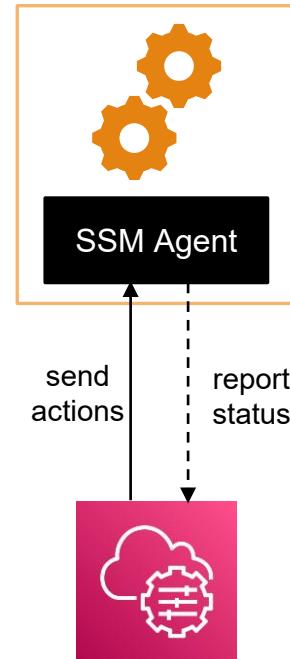
# Managing servers



# AWS Systems Manager (SSM)

- **Automate common and repetitive IT operations and management tasks**
- Works for Linux, Windows, MacOS, and Raspberry Pi OS (Raspbian)
- Run Command - Author and execute runbooks across fleet of EC2 or on-premises servers such as:
  - Attach IAM role to EC2 instance
  - Install Windows updates
  - Execute Shell script or Powershell script
  - Run Ansible playbooks or Chef recipes
- Fleet Manager - Get operational insights about the state of your infrastructure
- Patch Manager - Select and install OS and software security patches
- More features..

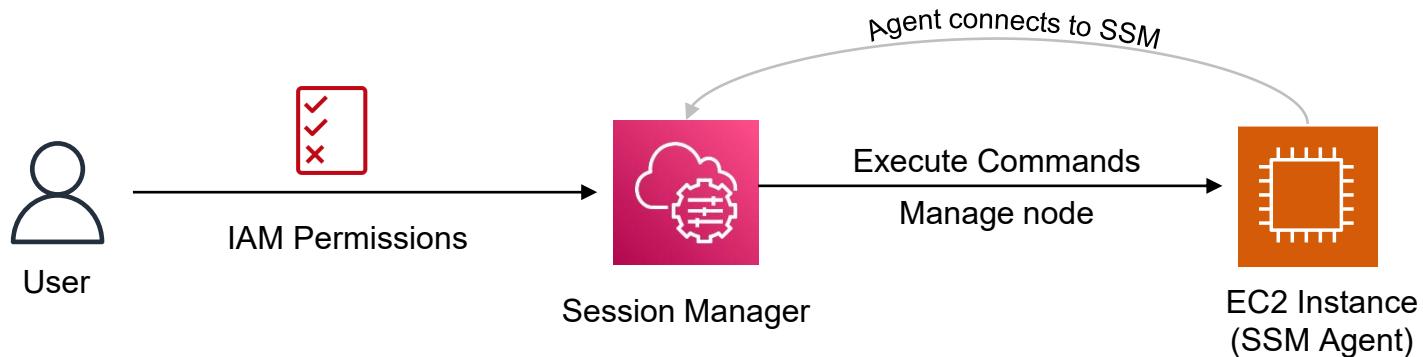
EC2 or Server



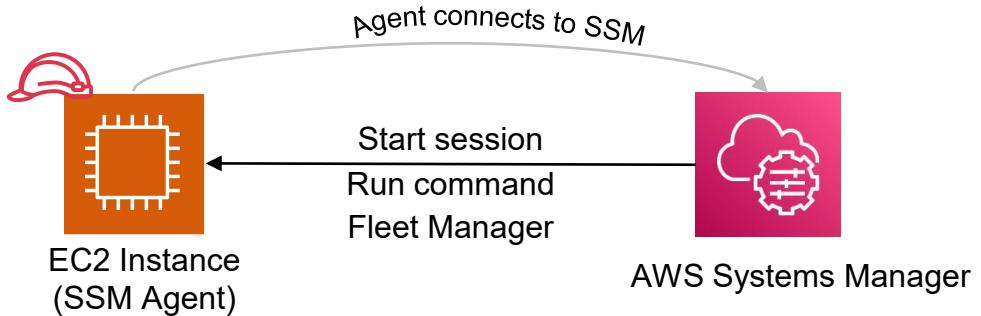
AWS Systems Manager

# Systems Manager - Session Manager

- Allows you to start a SSH session to your EC2 instance and on-premises servers.
- **No bastion hosts or SSH keys or SSH port (22) needed.**
- Access is granted through AWS IAM role/policies.
- Supports Linux, macOS, and Windows.



# Exercise – Systems manager for connecting and managing EC2 instances



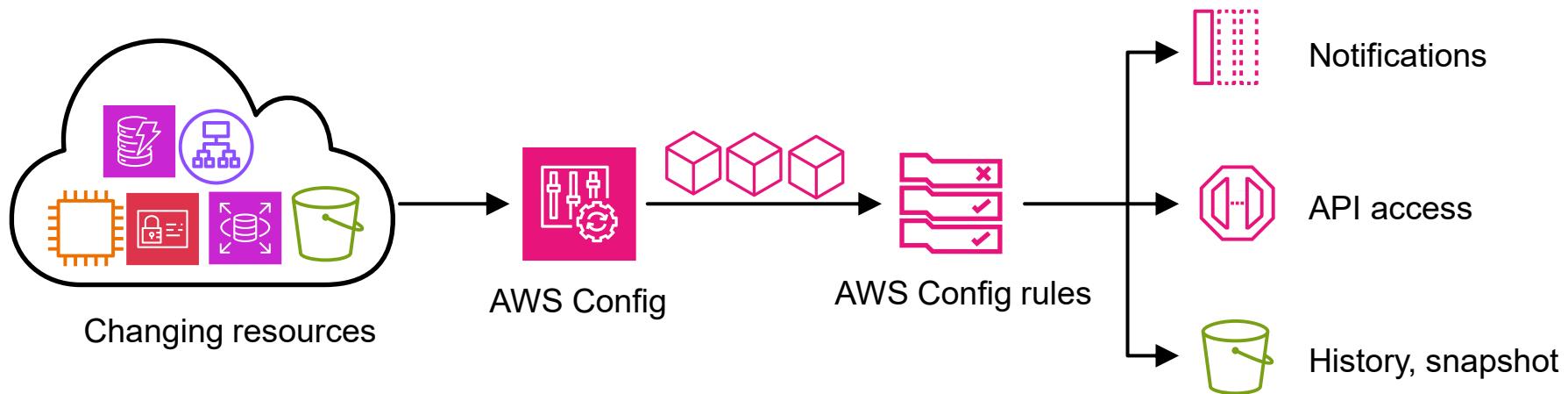
- 1 Create IAM Role for EC2 with permissions AmazonEC2RoleforSSM
- 2 Launch EC2 instance (Amazon Linux) and associate this IAM role while launching.
  - Amazon Linux comes with SSM agent pre-installed
  - For security group - No need to have any inbound rule.
- 3 Go to AWS Systems Manager and Session Manager -> Start Session with EC2 instance
- 4 Go to Systems Manager -> Fleet Manager and explore options to manage EC2 instance
- 5 Go to Systems Manager -> Run command and execute simple Shell command on EC2
- 6 Go to EC2 console -> Select Instance -> Connect -> Session Manager
- 7 After the exercise, terminate EC2 instance

# Troubleshooting – If instance does not appear in Systems Manager

- 1 Check if Systems Manager agent is running on EC2 instance
  - a) Open SSH port and connect to instance from your local workstation
  - b) Check if systems manager agent is running by running the command:  
`$sudo systemctl status amazon-ssm-agent`
- 2 Check if correct IAM role is associated with EC2 instance
  - a) For EC2 instance check the IAM role name associated with the instance
  - b) Go to IAM role and verify the IAM policy attached to the role. It should be: AmazonEC2RoleforSSM
- 3 Check if EC2 security group allows Outbound traffic
  - a) For EC2 instance to connect to Systems Manager endpoint, it needs outbound HTTPS connection
  - b) Check if EC2 security group allows outbound traffic (by default all outbound traffic is allowed)
- 4 Restart SSM agent on EC2 instance
  - a) If you modified the IAM role after EC2 instance is launched, you may have to restart the SSM agent.
  - b) Connect to EC2 instance over SSH from your workstation and run the command:  
`$sudo systemctl restart amazon-ssm-agent`

# AWS Config

- Record and Track AWS resources configuration changes
- AWS config is regional service but can aggregate the data from across AWS regions and accounts
- Evaluate resource configurations against the compliance rules
- On detecting any configuration deviation -> Send SNS notification or invoke API call to remediate the changes.
- Resource configuration data can be stored in S3 and queried by Athena



# AWS Config

## Questions that can be answered by AWS Config:

- Is there a security group which has port 22 (SSH) open for the world (0.0.0.0/0)?
- Are there any S3 buckets in my account which are Public (can be read by anyone)?
- Is there any DynamoDB table having provisioned capacity above 100 WCU?
- Is there a RDS database which is not multi-AZ?
- Are there any EBS volumes not protected by a backup plan?

AWS Config > Advanced queries

### Advanced queries

Use one of the sample queries or write your own query by referring to the configuration schema of the AWS resource.

Advanced queries (58)			
	Name	Description	Creator
<input type="radio"/>	Count EC2 Instances	Count EC2 instances, group by instance type	AWS
<input type="radio"/>	Count Non-compliant	Count number of non-compliant resources, group by resource type, sort by count in descending order	AWS
<input type="radio"/>	Count by compliant	Count number of resources, group by Config rule compliance status	AWS
<input type="radio"/>	Count of compliant and non-compliant rules of a conformance pack	Count of compliant and non-compliant rules for conformance pack "conformance-pack-12345"	AWS
<input type="radio"/>	Active DynamoDB tables	List all active DynamoDB tables	AWS
<input type="radio"/>	Availability zones for Load Balancer	List all availability zones of a Load Balancer "arn:aws:elasticloadbalancing:12345"	AWS
<input type="radio"/>	DynamoDB tables with disabled SSE	List all DynamoDB tables where server-side encryption is disabled	AWS
<input type="radio"/>	Unused EBS	List all EBS volumes that are not in use	AWS

# Infrastructure Management and Compliance - Summary

## AWS Systems Manager

- Centrally manage EC2 instances and on-premises servers

## AWS Config

- Record and track AWS resources configuration changes
- Run compliance checks against ideal configurations and take remediation action

# AWS Monitoring and Logging services

# AWS Monitoring and logging services

- Amazon CloudWatch
- AWS CloudTrail
- AWS X-Ray
- AWS Health Dashboard

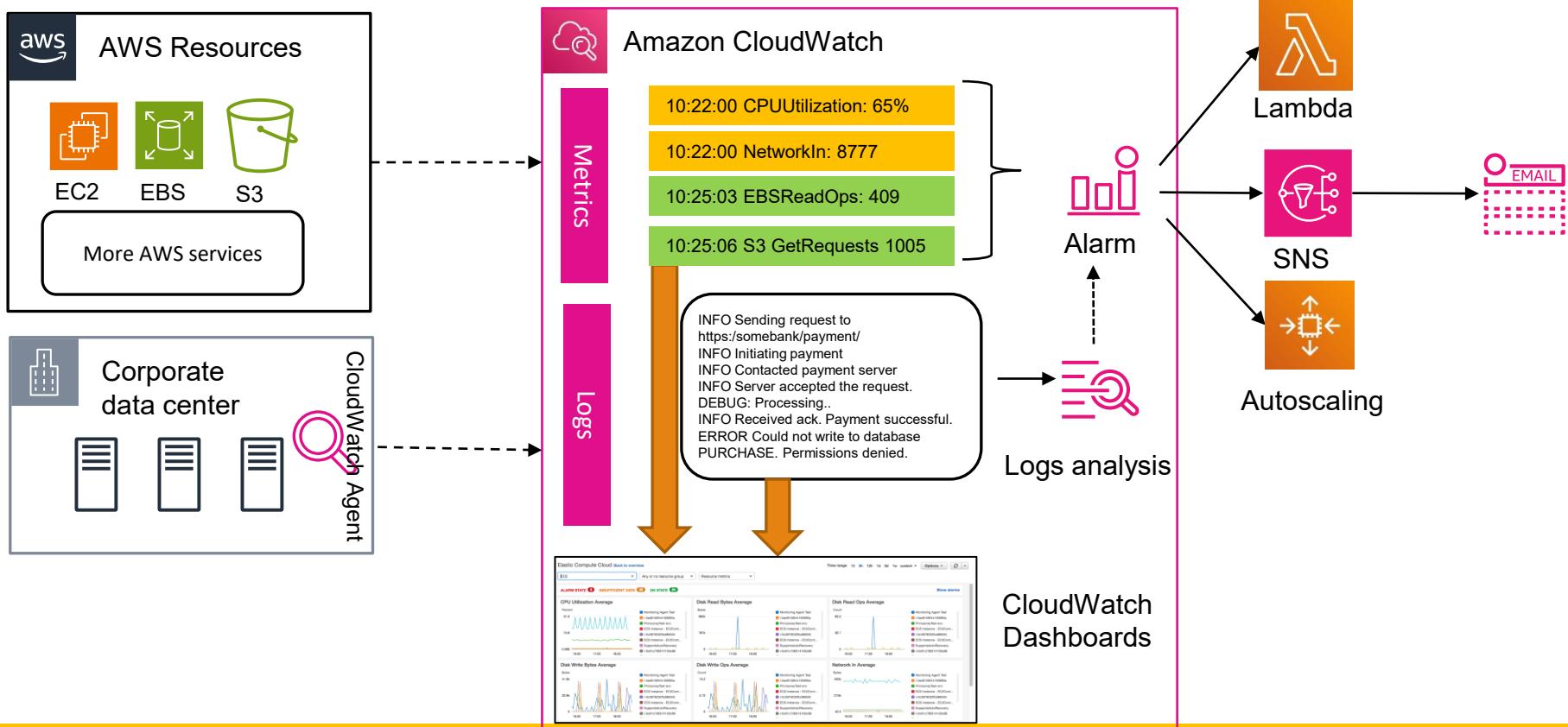
# Amazon CloudWatch

- Collects and tracks metrics, logs, and events for AWS resources and applications.
- Create custom dashboards for a visual representation of metrics and logs, aiding in monitoring and decision-making.
- Provides real-time and historical data
- Set alarms to automatically trigger actions (e.g., scaling, notifications) based on defined thresholds.
- Seamlessly integrates with majority of the AWS services (e.g., EC2, RDS, Lambda) and also supports third-party apps via API.



Amazon  
CloudWatch

# How CloudWatch works?

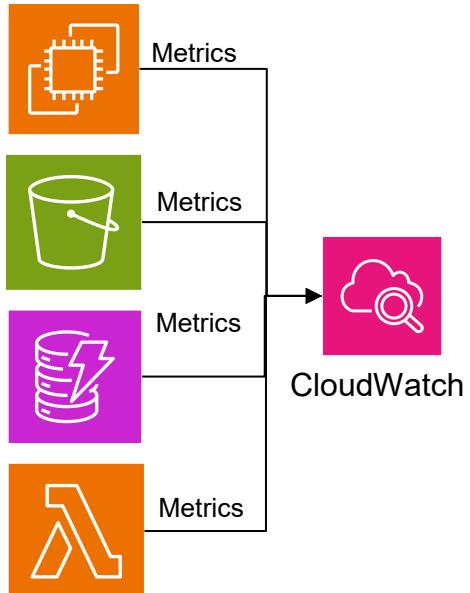


# Amazon CloudWatch

- CloudWatch Metrics
- CloudWatch Alarms
- CloudWatch Logs
- CloudWatch Event -> Now Amazon EventBridge
- CloudWatch Synthetics
- CloudWatch ServiceLens
- CloudWatch Anomaly detection
- CloudWatch RUM
- More..

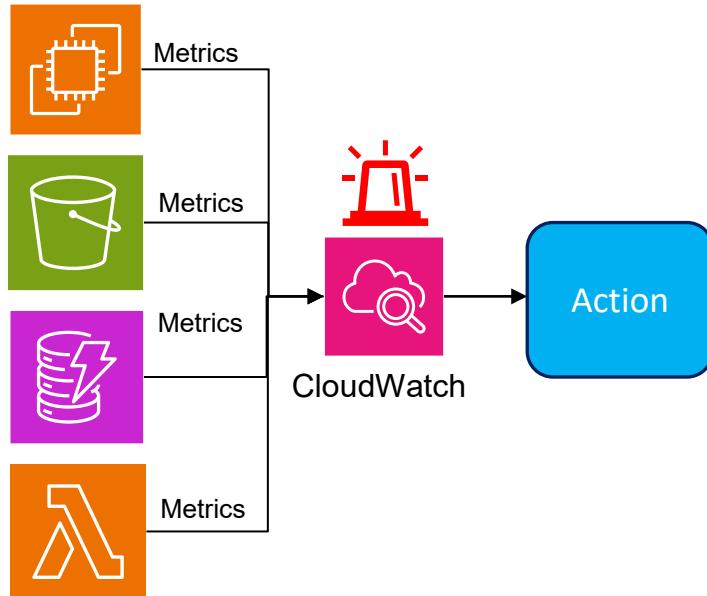
# CloudWatch Metrics

- CloudWatch Metric is the unit of measurement for system performance at a specific time. Example:
  - CPUUtilization
  - NetworkIn and NetworkOUT Bytes
  - EBS DiskReads and DiskWrites
  - S3 NumberOfRequests
- Metrics data is stored as a timeseries to be able to analyse
  - Act based on metrics values e.g. Raise an alarm if CPUUtilization is greater than 90% for last 5 mins
- CloudWatch provides Basic monitoring and Detailed monitoring.
  - Example: EC2 basic monitoring publishes metrics every 5-minutes whereas detailed monitoring publishes same metrics at every 1-minute
- We can also create custom CloudWatch Metrics



# Amazon CloudWatch Alarms

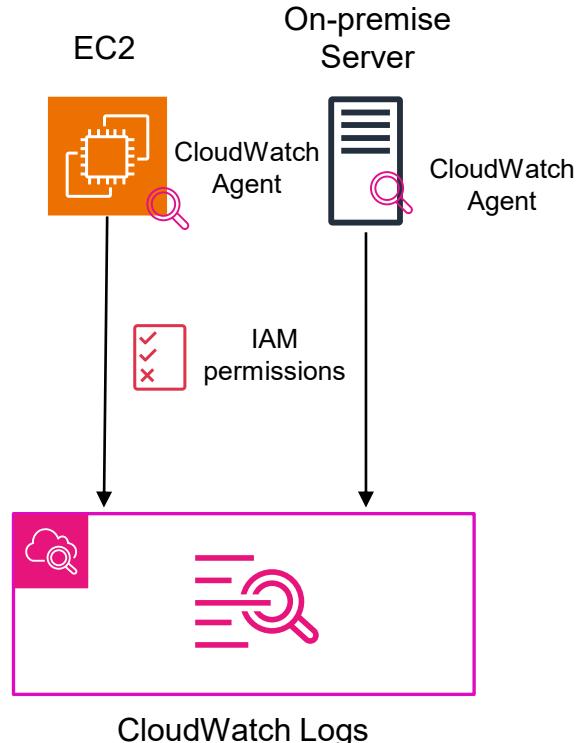
- CloudWatch Alarms allows you to take action when the metrics fall outside of the levels (high or low thresholds)
- A CloudWatch Alarm is always in one of three states: OK, ALARM, INSUFFICIENT\_DATA
- Alarms actions
  - **Auto Scaling:** Increase or decrease EC2 instances “desired” count.
  - **EC2 Actions:** Stop, terminate, reboot or recover an EC2 instance.
  - **SNS notifications:** Send a notification into an SNS topic.



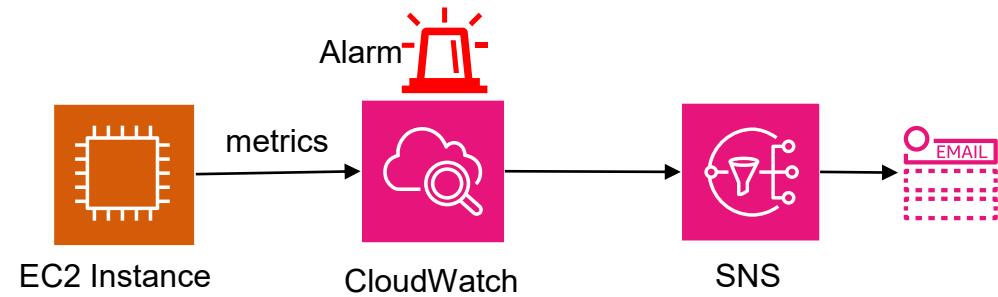
# Amazon CloudWatch Logs

- Collect, store, monitor and analyse logs from AWS services and applications such as EC2, ECS, Elastic Beanstalk, CloudTrail, Route53 DNS
- By default, no logs from your EC2 instance will go to CloudWatch
- Need to install CloudWatch Agent to push logs to CloudWatch. Agent needs IAM permissions to push the logs.
- By default, logs are retained forever however we can change the logs retention duration as required.
- Logs can be stored in Standard storage class or Infrequent-Access (Logs-IA) at lower cost.
- CloudWatch Logs insights: Write interactive queries on the logs

```
filter @message like /ERROR/ | limit 10
```



# Exercise - CloudWatch Metrics and Alarm



- 1 Create a SNS notification topic and create an email subscription. Confirm subscription by clicking the link received in the email.
- 2 Launch an EC2 instance
- 3 Go to AWS CloudWatch -> Alarms -> Create new alarms for your EC2 instance when CPUUtilization > 50 percent
- 4 Login into EC2 instance over SSH and load the CPU by using *stress* command
- 5 Wait up to 5 mins to see EC2 CPU utilization going high. This should trigger an Alarm.
- 6 Leave the EC2 instance running if you will be continuing with CloudWatch Logs lecture and corresponding exercise. Otherwise terminate the instance.

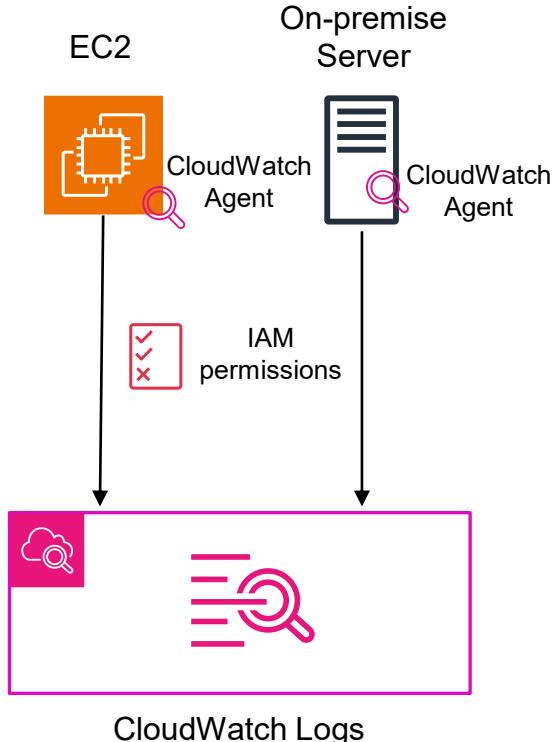
Commands to install and run stress to load the EC2 CPU:

```
sudo yum install -y stress  
stress --cpu 1 --timeout 600
```

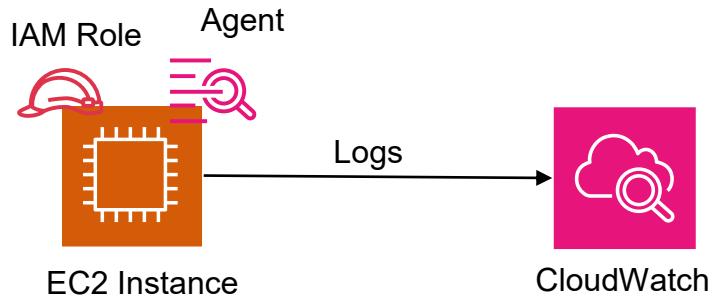
# Amazon CloudWatch Logs

- Collect, store, monitor and analyse logs from AWS services and applications such as EC2, ECS, Elastic Beanstalk, CloudTrail, Route53 DNS
- By default, no logs from your EC2 instance will go to CloudWatch
- Need to install CloudWatch Agent to push logs to CloudWatch. Agent needs IAM permissions to push the logs.
- By default, logs are retained forever however we can change the logs retention duration as required
- CloudWatch Logs insights: Write interactive queries on the logs

```
filter @message like /ERROR/ | limit 10
```



# Exercise - CloudWatch Logs



- 1 Create an IAM role for EC2 with IAM policy CloudWatchAgentServerPolicy
- 2 Attach this role to an EC2 instance
- 3 Login to EC2 over SSH and install AWS CloudWatch agent
- 4 Create CloudWatch agent configuration file for setting up logs that you want to send to CloudWatch service
- 5 Start CloudWatch agent service
- 6 Create a dummy log file in the directory as configured in the agent config file
- 7 Go to CloudWatch Logs. You should see a new CloudWatch Logs group and Log stream. Check if you see the log entries.
- 8 (Optional) Go to CloudWatch Logs insights and run few queries to filter the logs by some keywords e.g. ERROR
- 9 After the exercise, terminate EC2 instance and delete CloudWatch Logs group

# Exercise - Useful commands

1. Perform following action using root user. Run this command to change user to root: ***sudo su***
2. Install AWS CloudWatch agent: ***yum install amazon-cloudwatch-agent***
3. Create CloudWatch agent configuration file using vi or vim editor. Sample file config.json provided for download.
4. Start CloudWatch agent: ***/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -a fetch-config -m ec2 -c file:<path to the /config file that you created> -s***
5. Check the status of CloudWatch agent: ***/opt/aws/amazon-cloudwatch-agent/bin/amazon-cloudwatch-agent-ctl -m ec2 -a status***
6. Now create dummy log file in the ***/var/log/*** directory. Sample application.log provided for download.

# Exercise - Sample files

## 1. config.json

```
{  
  "logs": {  
    "logs_collected": {  
      "files": {  
        "collect_list": [  
          {  
            "file_path": "/var/log/application.log",  
            "log_group_name": "/myapplication",  
            "log_stream_name": "{instance_id}/application_logs",  
            "timestamp_format": "%b %d %H:%M:%S"  
          }  
        ]  
      }  
    }  
  }  
}
```

## 2. application.log

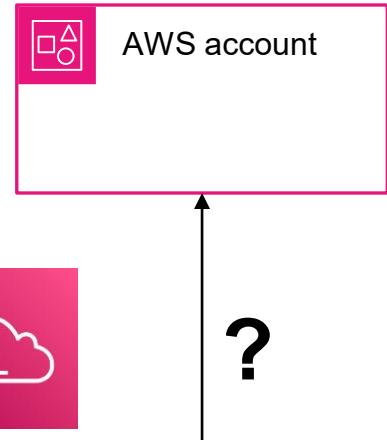
```
INFO MainApp - Application started successfully.  
DEBUG ServiceA - Initializing Service A with config: {timeout: 5000ms, retries: 3}  
INFO ServiceB - Fetching data from external API...  
ERROR ServiceB - Failed to fetch data from API: TimeoutException  
DEBUG DatabaseConnection - Attempting to connect to database at 10.0.0.1:5432  
INFO DatabaseConnection - Connection to database established successfully.  
ERROR UserService - User not found for ID: 12345  
INFO Authentication - User login successful for username: john doe  
DEBUG ServiceA - Processing user input: {userId: 12345, action: 'login'}  
INFO ServiceA - User session initialized for userID: 12345  
ERROR PaymentService - Payment failed: Insufficient funds for userID: 12345  
DEBUG PaymentService - Payment request payload: {amount: 100.00, currency: 'USD'}  
INFO NotificationService - Sending email notification to userID: 12345  
ERROR NotificationService - Failed to send email to userID: 12345:  
SMTPServerException  
DEBUG CacheManager - Cache miss for key: userDetails_12345  
INFO ServiceC - Scheduled job 'DataSync' completed successfully.  
DEBUG ServiceC - DataSync job processed 150 records.  
ERROR ServiceC - DataSync job failed: NullPointerException encountered in  
processRecords()  
INFO MainApp - Shutting down application...  
DEBUG MainApp - Application shutdown completed. Cleanup performed.
```



# AWS CloudTrail

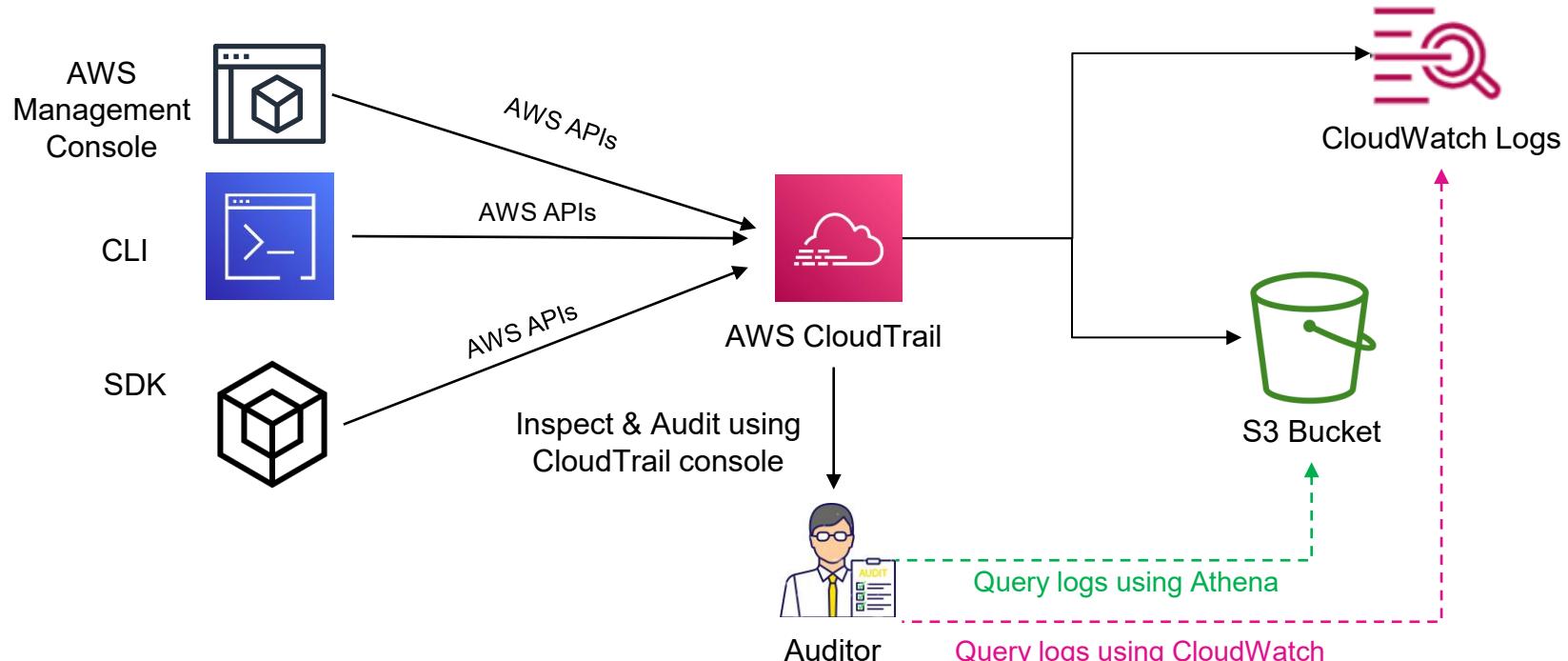
# AWS CloudTrail

- AWS CloudTrail help you answer the question of "**Who did what and when in your AWS account?**"
- You are able to audit your AWS account for any suspicious activity.
- Actions taken by a IAM user, IAM role, or an AWS service are recorded as events in CloudTrail.
- Get an history of events / API calls made within your AWS Account by:
  - AWS Management Console
  - AWS Command Line Interface
  - AWS SDKs and APIs
- CloudTrail is enabled on your AWS account by default when you create it.
- A trail can be applied to All AWS Regions (default) or a single Region.



# AWS CloudTrail - Ways to access the logs

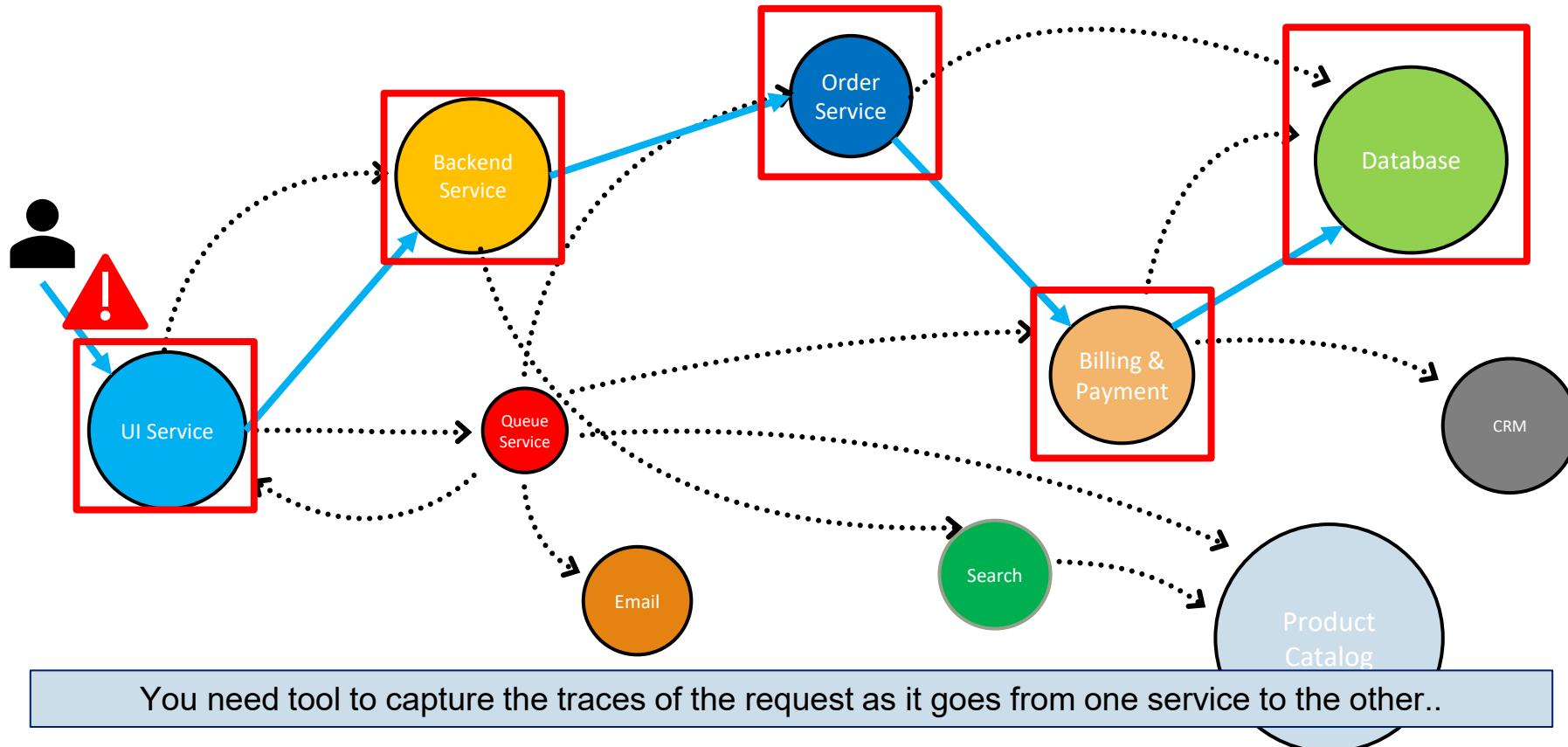
- We can send logs from CloudTrail into CloudWatch Logs or Amazon S3 and then query using CloudWatch logs insights or Amazon Athena





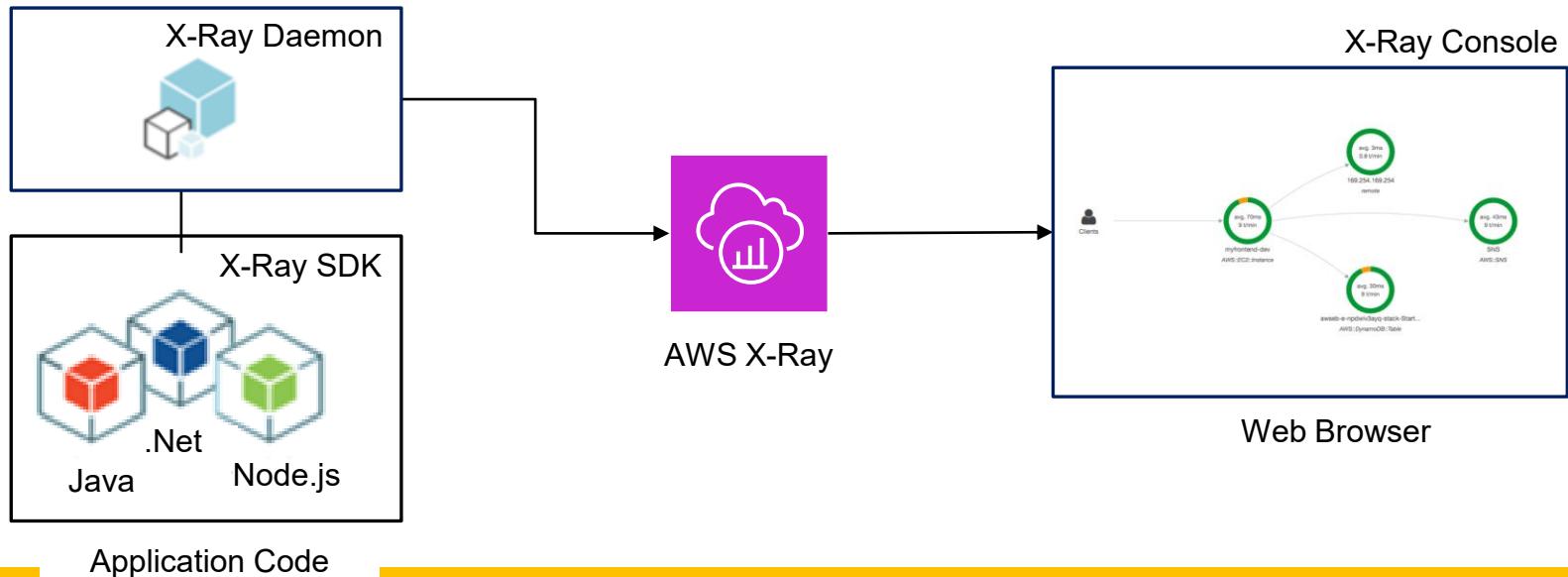
# AWS X-Ray

# Microservices application in real world

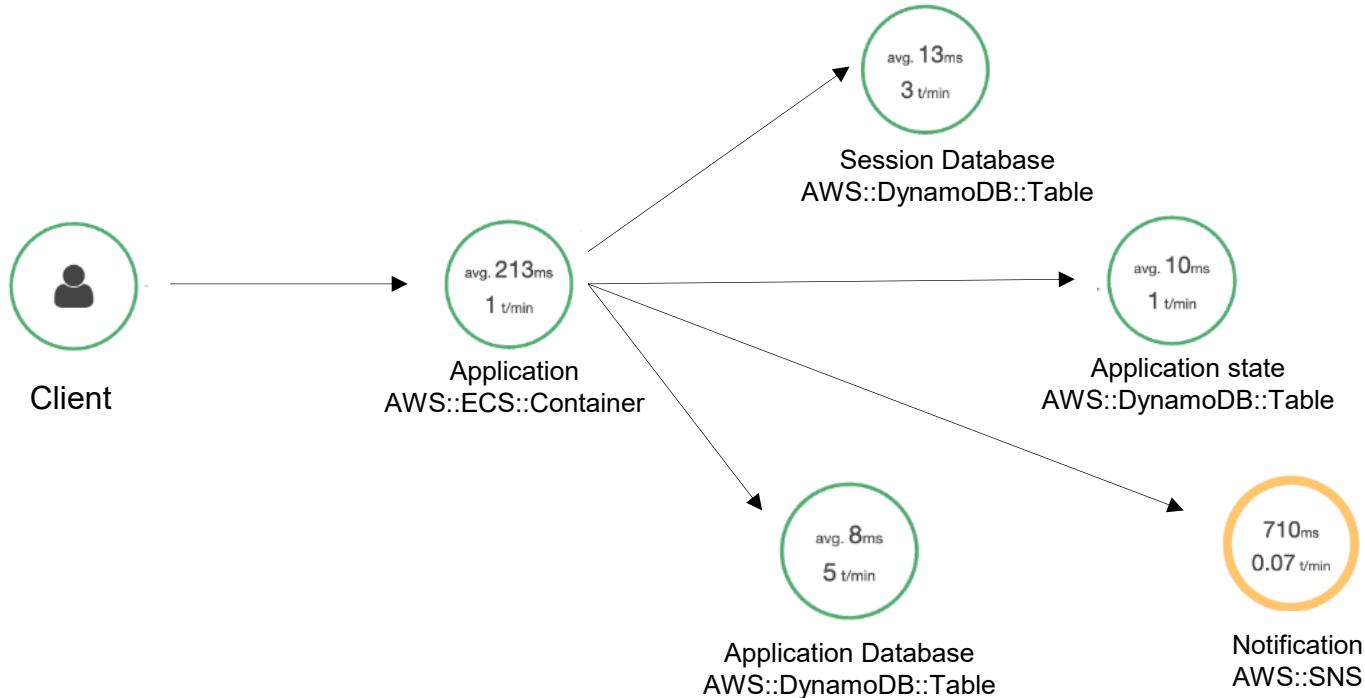


# AWS X-Ray

- Service that collects data for requests that your application serves.
- Provides tools that you can use to view, filter, and gain insights into that data to identify issues and opportunities for optimization.



# AWS X-Ray : Visual Analysis of Applications



# Use AWS X-Ray for..

- Debugging slow responses to the user requests
- Troubleshooting increased error rates
- Tracing down the user request path -> Traces
- Are requests getting completed in SLA time?
- Which service is the bottleneck in the application?
- Which users are impacted by the specific service disruption?
- More..



AWS X-Ray



# AWS Health Dashboard



# AWS Health Dashboard - Service health

- Shows AWS services disruption events

AWS Health Dashboard Updated less than 1 minute ago Contact us

## Service health

View the current and historical status of all AWS services.

[Open and recent issues \(1\)](#) [Service history](#)

**▼ Operational issue - Multiple services (Ohio)**

Service	Severity	Action
Multiple services	Resolved	<a href="#">RSS</a>

**[RESOLVED] Increased API Error Rates**

**Oct 07 7:38 PM PDT** We wanted to provide some additional information. Beginning at 12:27 PM, we experienced increased error rates for S3 GET/PUT requests made in the US-EAST-2 Region. This issue also impacted other AWS Services who use these APIs as part of service operations. The S3 issue was resolved at 12:52 PM, when S3 error rates returned to normal operations. Since then, we have been working to fully understand the root cause.

This issue was caused by a latent software issue in a subsystem of S3 that is responsible for assessing metadata (such as versioning) during S3 PUT and GET operations. We have already implemented mitigations, which include removing the software issue in the S3 request path, and have identified new testing to detect these types of software issues in the future.

**Oct 07 1:19 PM PDT** Between 12:27 PM and 12:52 PM PDT we experienced increased API error rates for PUT and GET requests to S3 in the US-EAST-2 Region. Other AWS Services that use PUT and GET S3 APIs were also affected. The error rates have recovered and all services are operating normally. Engineers were automatically engaged immediately and began mitigating the impact and investigating the root cause in parallel. We will post a final update as we validate root cause.

**Oct 07 1:06 PM PDT** Between 12:27 PM and 12:52 PM PDT we experienced increased API error rates for PUT and GET requests to S3 in the US-EAST-2 Region. Other AWS Services that use PUT and GET S3 APIs were also affected. The error rates have recovered, and we continue to investigate root cause.

**Oct 07 12:46 PM PDT** We are investigating Increased API Error Rates for Multiple services in the US-EAST-2 Region.

**View your account health**

Get a personalized view of events that affect your AWS account or organization.

[Open your account health](#)

# AWS Health Dashboard - Service history

- Shows AWS services disruption events
- Shows current and historical status of the AWS services health

Service history

The following table is a running log of AWS service interruptions for the past 12 months. Choose a status icon to see status updates for that service. All dates and times are reported (PDT). To update your time zone, see [Time zone settings](#).

Find an AWS service or Region  2024/08/27

North America	South America	Europe	Africa	Asia Pacific	Middle East	All locales	<	1	2	3	4	5
Service	RSS	«	Today	26 Aug	25 Aug	24 Aug	23 Aug	22 Aug				
Amazon API Gateway (Calgary)	RSS	«	Today	26 Aug	25 Aug	24 Aug	23 Aug	22 Aug				
Amazon API Gateway (Canada-Central)	RSS	«	Today	26 Aug	25 Aug	24 Aug	23 Aug	22 Aug				
Amazon API Gateway (N. California)	RSS	«	Today	26 Aug	25 Aug	24 Aug	23 Aug	22 Aug				
Amazon API Gateway (N. Virginia)	RSS	«	Today	26 Aug	25 Aug	24 Aug	23 Aug	22 Aug				
Amazon API Gateway (Ohio)	RSS	«	Today	26 Aug	25 Aug	24 Aug	23 Aug	22 Aug				
Amazon API Gateway (Oregon)	RSS	«	Today	26 Aug	25 Aug	24 Aug	23 Aug	22 Aug				

# AWS Health Dashboard – Your account

- Provides alerts and remediation guidance when AWS is experiencing events that may impact you.
- Proactive notification to help you plan for scheduled activities.

## Your account health

Stay informed of important events affecting your AWS resources.

[Open and recent issues \(16\)](#) | [Scheduled changes \(0\)](#) | [Notifications \(3\)](#) | [Event log](#)

[Go to EventBridge](#)

### Open and recent issues (16)

View events that might affect your AWS infrastructure. [35 issues](#) were resolved in the past 24 hours.

Add filter | Service: Elastic Compute Cloud X | Clear filter | < 1 >

Event summary	
<a href="#">Operational issue - EC2 (Ohio)</a> Last update: February 20, 2022 at 11:16:34 PM UTC-8 us-east-2	
<a href="#">Operational issue - EC2 (Ohio)</a> Last update: February 17, 2022 at 11:56:09 PM UTC-8 us-east-2	
<a href="#">Operational issue - EC2 (N. Virginia)</a> Last update: February 16, 2022 at 1:36:29 AM UTC-8 us-east-1	

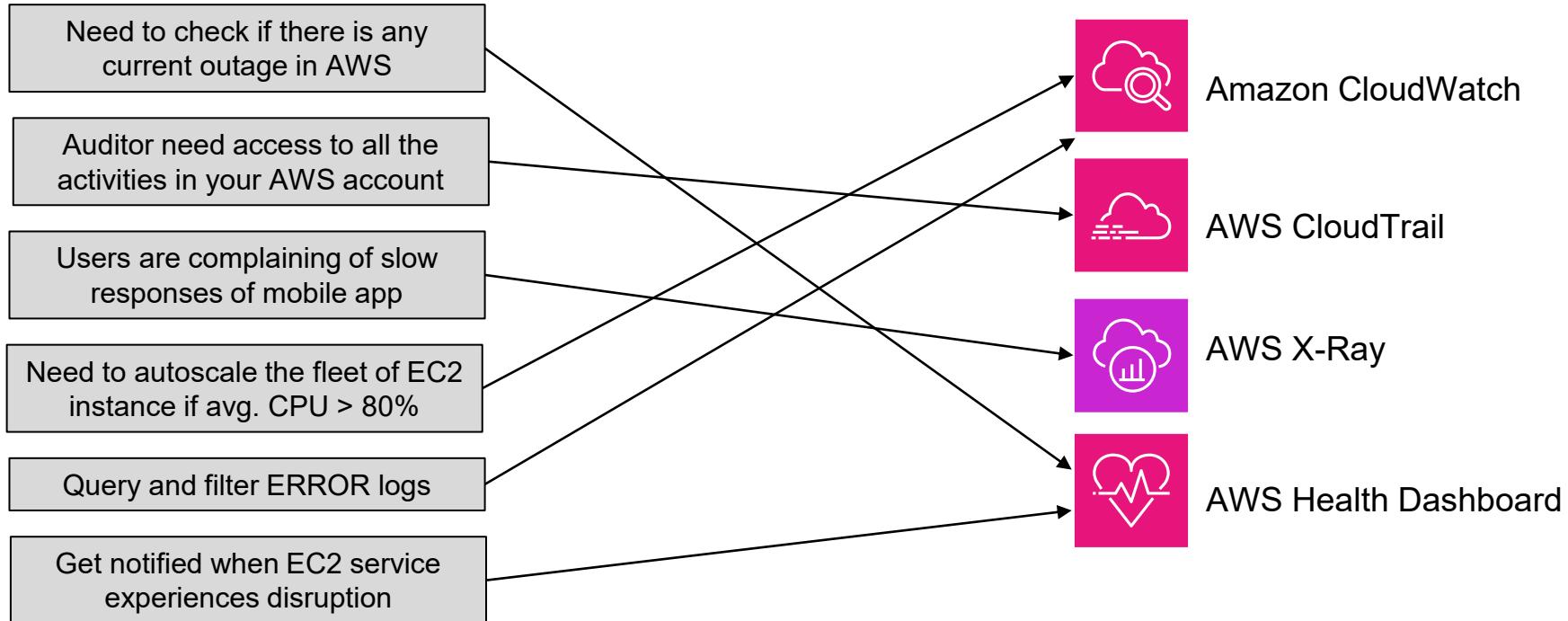
### Operational issue - EC2 (Ohio)

[Details](#) | [Affected resources](#)

#### Event data

Service	Start time
EC2	February 20, 2022 at 11:16:24 PM UTC-8
Status	End time
Open	-
Region / Availability Zone	Category
us-east-1	Issue
Account specific	Affected resources
No	1
<b>Description</b>	
[04:35 AM PST] We are investigating increased EC2 launch failures and networking connectivity issues for some instances in a single Availability Zone (USE1-AZ4) in the US-EAST-1 Region. Other Availability Zones within the US-EAST-1 Region are not affected by this issue.	

# AWS Monitoring & Logging services - When to use what?

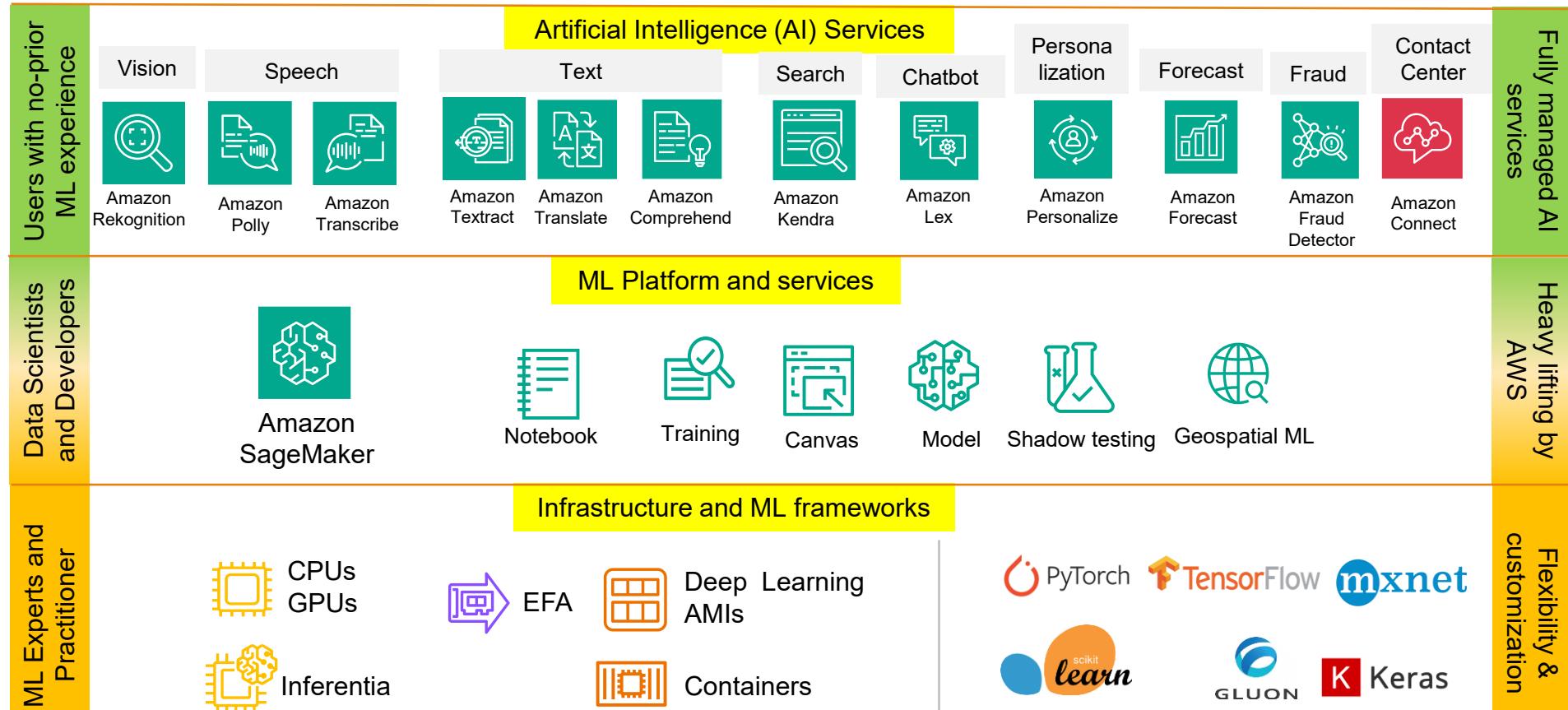


# AWS Monitoring & Logging services - summary

- Amazon CloudWatch
  - Metrics: Unit of measurement for performance of AWS services
  - Alarms: Take action if metrics falls outside of desired value/limit
  - Logs: Collect log files from EC2 instances, on-premises servers, Lambda functions etc.
- AWS CloudTrail – Captures AWS API calls made within your AWS account
- AWS X-Ray - Trace requests made through your distributed applications
- AWS Health Dashboard - Status of all AWS services across all regions
- AWS Account Health Dashboard - AWS events having impact on your AWS services or resource in your AWS account

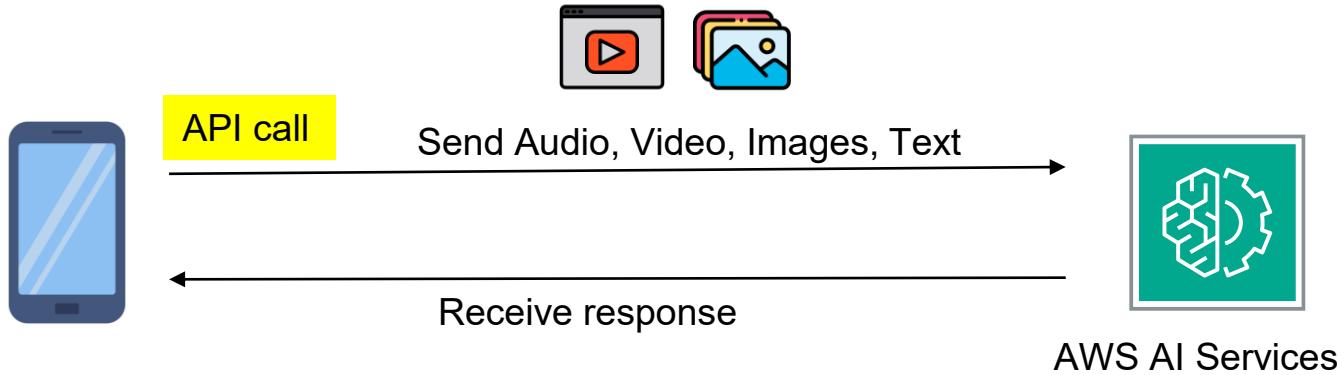
# AWS Machine Learning & AI

# AWS' three layered approach to AI/ML



# AWS AI services

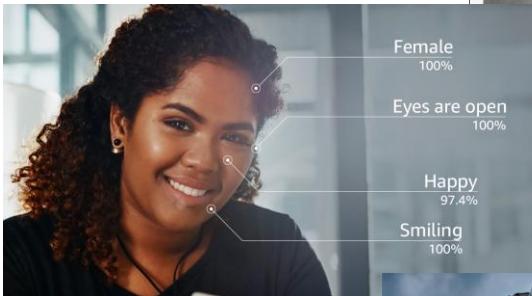
# Using AWS AI services





# Amazon Rekognition

- Computer Vision based AI service to find people, texts, objects, scene in images and videos.
- Common use cases:
  - Celebrity recognition
  - Face compare and search
  - Face detection and analysis
  - Content moderation
  - Detect objects, Brand logos, Texts
  - Video segment detection (blank frames etc.)
  - Easy filtering of video for explicit and suggestive content



Celebrity recognition

Rekognition automatically recognizes celebrities in images and provides confidence scores (Your images aren't stored.)

Done with the demo? [Download SDKs](#)

Results

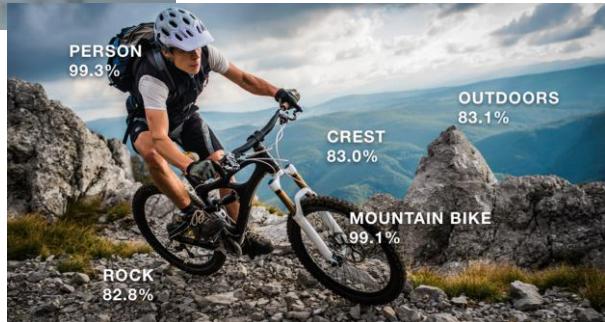
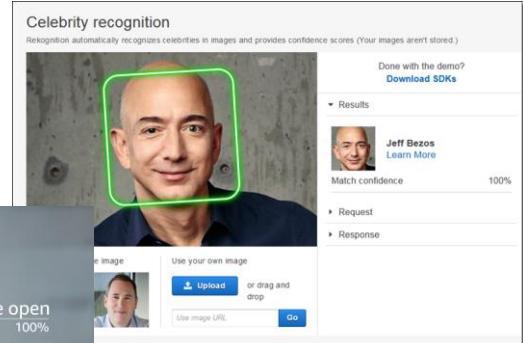
Jeff Bezos [Learn More](#)

Match confidence 100%

Request Response

Upload or drag and drop

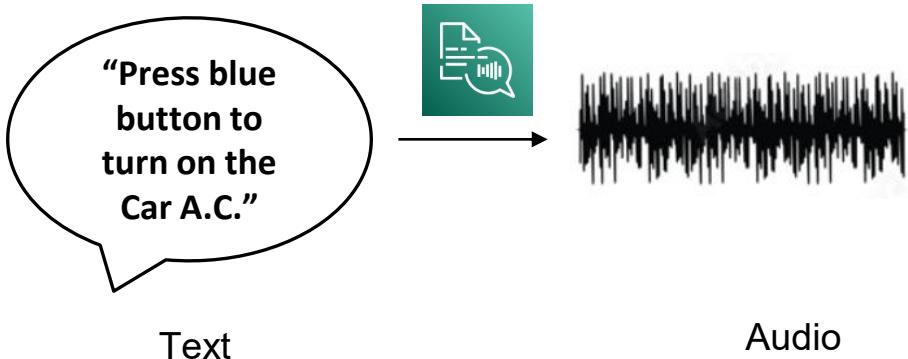
Use image URL: [Go](#)



# Amazon Polly



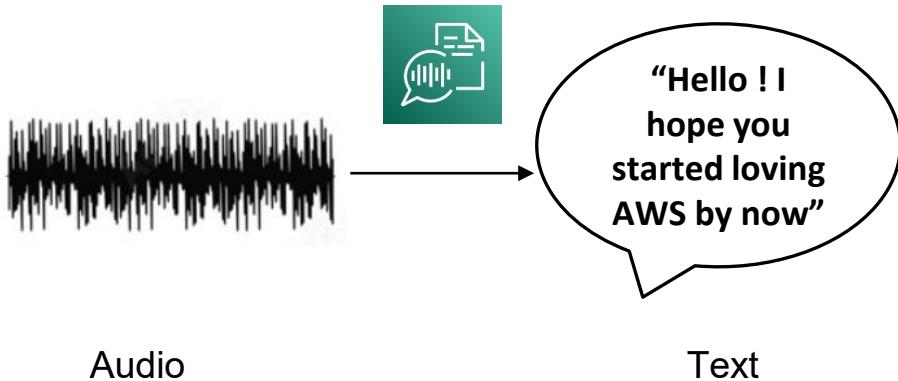
- Converts text to speech
- Create applications that talk to increase engagement and accessibility
- Supports Speech synthesis - phrases, punctuations, pauses
- Common use cases:
  - Add speech for global audience RSS feeds, websites, blogpost
  - Automated voice response system ("Dear customer, your account balance is zero ☺")
  - Supports variety of lifelike voices that you can choose from to better serve your customers as per their location



# Amazon Transcribe



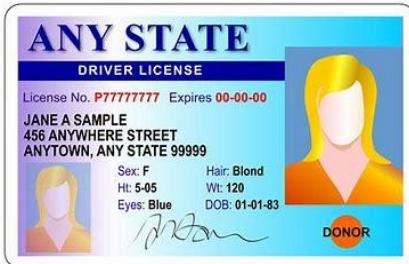
- Automatic speech recognition (ASR) service to convert **audio** to **text**.
- Supports Automatic Language Identification
- Supports over 100+ languages (English, Chinese, Arabic, French, Korean, Hindi, German etc.)
- Automatically removes Personally Identifiable Information (PII) using redaction.
- Common use cases:
  - Transcribe customer service calls
  - Automate closed captioning and subtitling
  - Generate metadata for media assets to create a fully searchable archive
  - Detect toxic content in the audio (social media)
  - Convert clinical conversation into health records (\*Amazon Transcribe for medical)



# Amazon Textract



- Automatically extract printed text, handwriting, layout elements, and data from scanned documents.
- Much more powerful than simple OCR (optical character recognition) where it understands the document layout, forms, tables, images and extracts relevant and related data
- Use Cases:
  - Financial Services (e.g., Invoices, Financial reports)
  - Healthcare (e.g., Medical records, Insurance claims)
  - Public Sector (e.g., Tax forms, ID documents, Passports)



```
{  
  "Document ID": "P77777777777777",  
  "Name": "JANE A SAMPLE",  
  "SEX": "F",  
  "DOB": "01-01-83",  
  ...  
}
```

# Amazon Translate



- Fluent and accurate language translation.
- Supports translating text between 75 languages (& growing)
- Use cases: Translate user manuals, books, documents, websites etc.

Translation

Text      Document

Source language: English (en)      Target language: Hindi (hi)

Enter text:

Congratulation on completing your AWS certification

Translated text:

आपके AWS प्रमाणन को पूरा करने पर बधार्ह

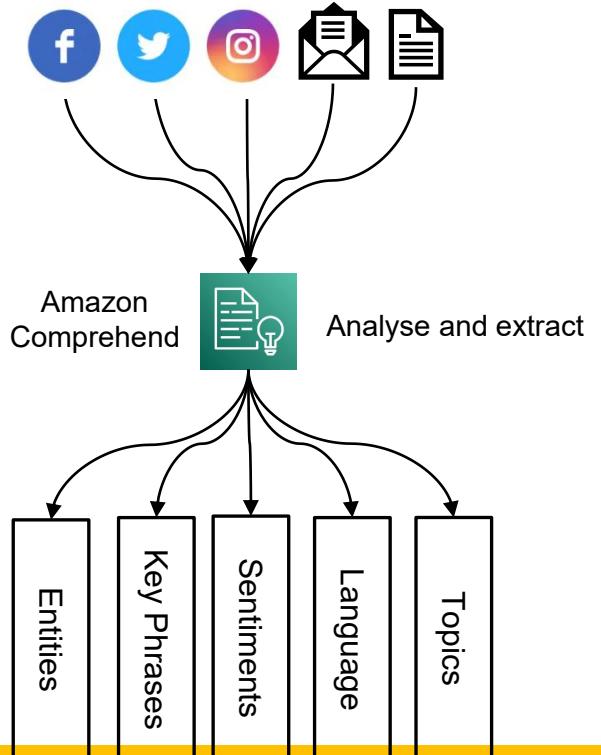
51 characters, 51 of 10000 bytes used. [Info](#)

► Additional settings

# Amazon Comprehend

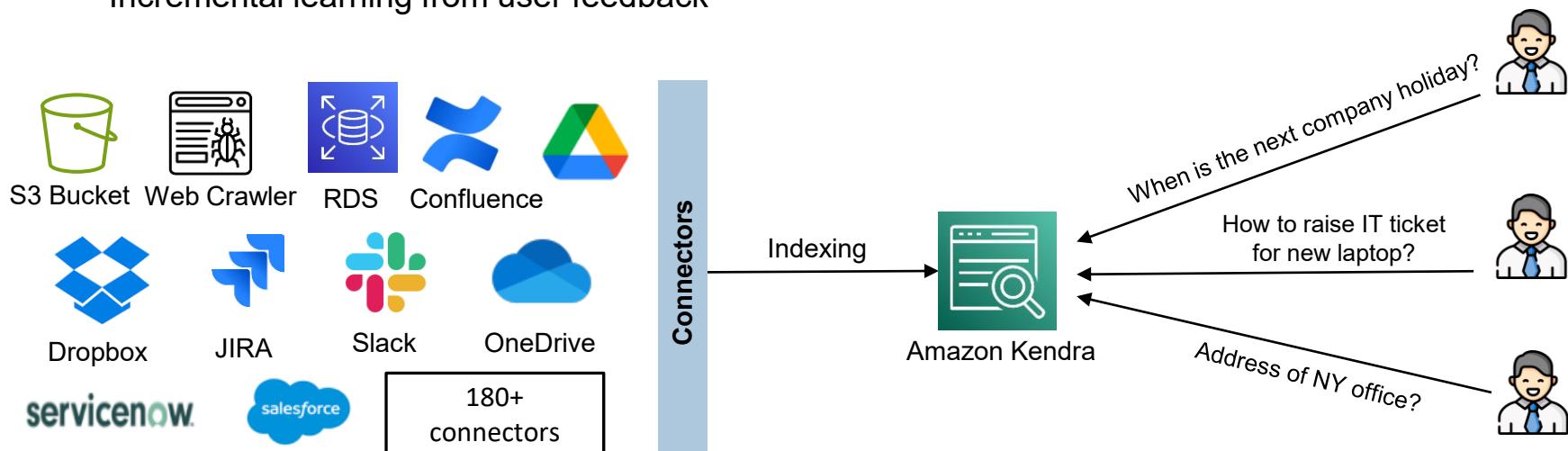


- Uses **Natural Language Processing (NLP)** to extract insights about the content of documents.
- Finds insights and relationships in text such as:
  - Language of the text
  - Extracts key phrases, places, people, brands, or events
  - Understands how positive or negative sentiments are
  - Automatically organizes a collection of text files by topic
- Use Cases:
  - Analyze customer interactions to find what leads to a positive or negative experience
  - Create and group articles by topics



# Amazon Kendra

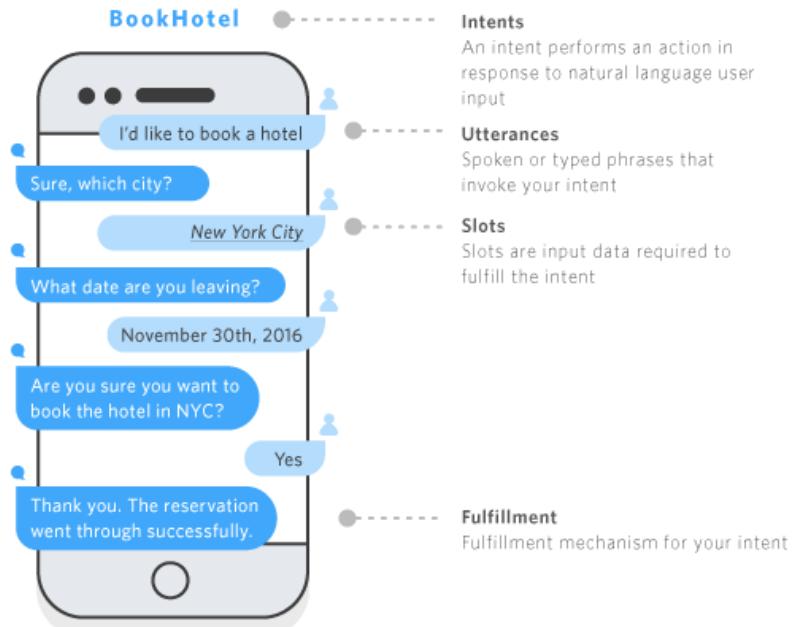
- An intelligent ML powered enterprise search service
- Employees and customers can find the content they're looking for from multiple locations and content repositories within your organization.
- Indexes variety of documents such as Websites/HTML, Share point, PPT, MS Word and provides accurate answers based on Natural language search capabilities.
- Incremental learning from user feedback





# Amazon Lex

- Advanced Natural Language model to design, build, test and deploy **conversational chatbot**.
- Lex integrates with AWS Lambda, used to easily trigger functions for execution of your back-end business logic for data retrieval and updates.
- Use cases:
  - Contact centers
  - Virtual contact center agent
  - FAQs for technical support, HR, finance
  - Automate CRM activities across digital channels





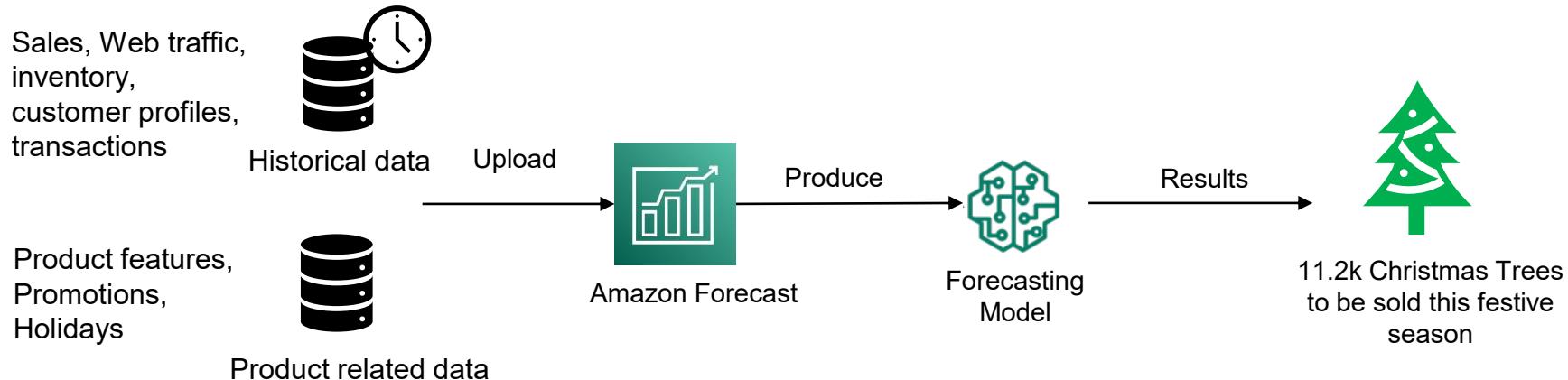
# Amazon Personalize

- A fully managed Machine Learning (ML) service that uses your data to generate item recommendations for your users.
- Use the same ML models and technology that is used by Amazon.com
- **Examples:**
  - User personalization - Used for Home or landing page per user
  - Related items recommendations - As customer look for a particular item, it shows additional items that can be purchased together
  - Personalized ranking - Ranks the items to be displayed on the page in different order as per user's current interest
- **Use Cases:** eCommerce, retail stores, movies / media and entertainment

# Amazon Forecast

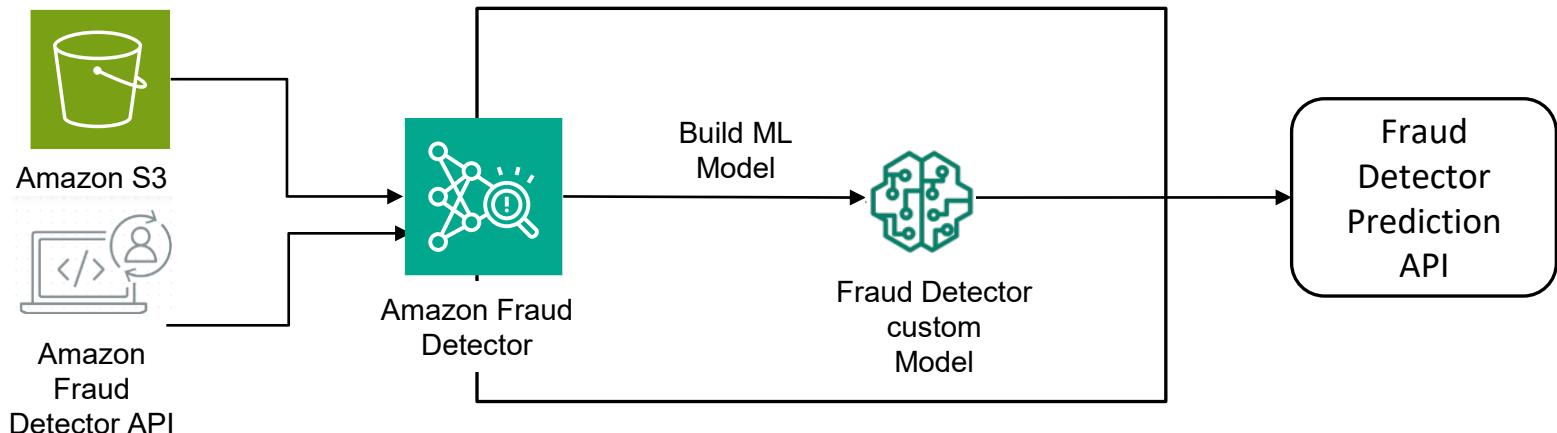
DISCONTINUED

- Use the same forecasting ML models that are used by Amazon.com
- Optimize inventory and reduce waste with accurate forecast
- Reduce forecasting time from months to hours.
- Use cases: Product Demand Planning, Financial Planning, Resource Planning and more ...



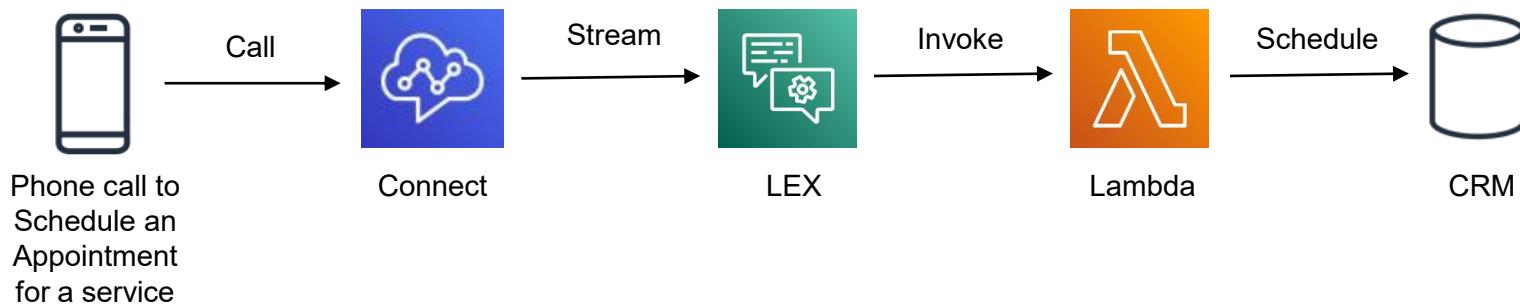
# Amazon Fraud Detector

- Detect online fraud faster with machine learning
- Build, deploy, and manage fraud detection models without previous machine learning (ML) experience
- Leverage 20+ years of Amazon's experience detecting online frauds
- **Use cases:** Identify suspicious online payment, Detect new account fraud, Prevent loyalty program abuse, compromised accounts etc.



# Amazon Connect

- AI-powered cloud contact center
- Automatically detects customer issues and provides agents with contextual customer information and suggested responses and actions for faster resolution of issues.
- Easy to create flows and integration with other CRM systems or AWS

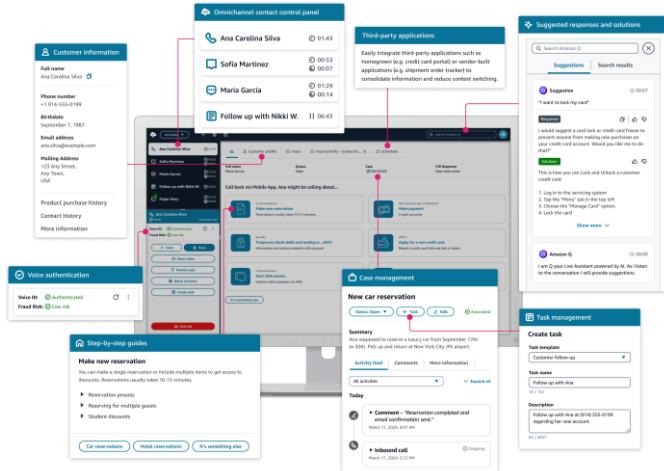


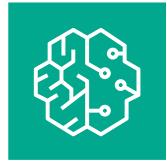
# Amazon Connect

- AI-powered cloud contact center
- Automatically detects customer issues and provides agents with contextual customer information and suggested responses and actions for faster resolution of issues.
- Easy to create flows and integration with other CRM systems or AWS

## Agent Workspace:

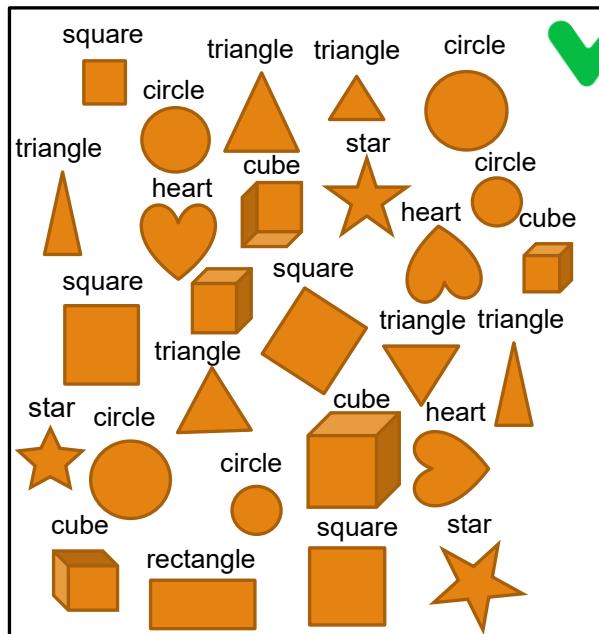
- When an agent accepts a call, chat, or task, they receive necessary information about the case and customer and real-time recommendations.



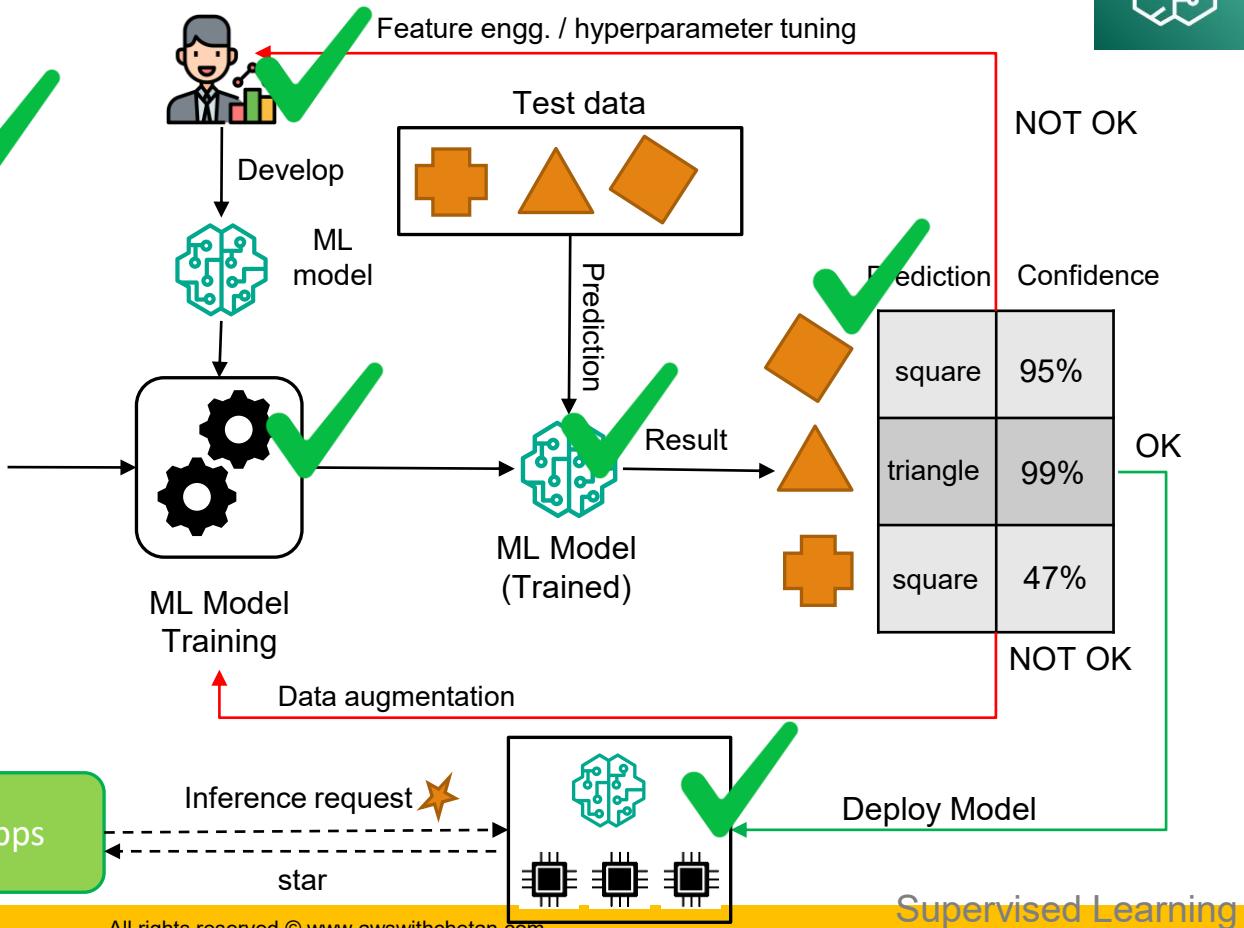


# Amazon SageMaker

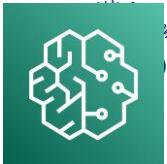
# Machine Learning with Amazon SageMaker



Labelled Data

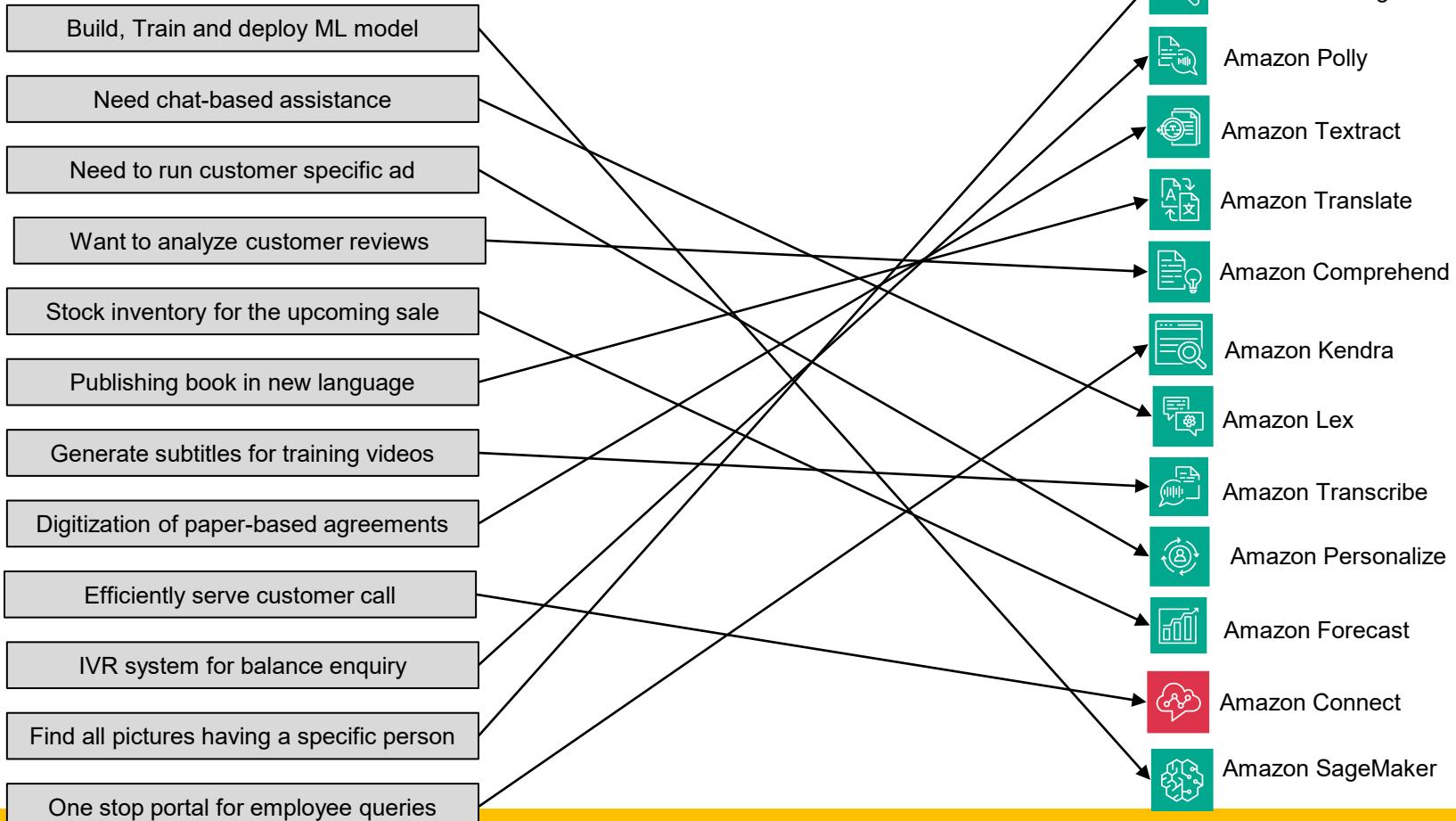


# Amazon SageMaker



- Fully managed service for developers / data scientists to build, train and deploy ML models at scale
- Amazon SageMaker Features:
  - **SageMaker Studio** – A web-based IDE interface for building, training and deploying models.
  - **AutoML** (SageMaker Autopilot) - Automatically explores and creates the best ML models
  - **Built-in Algorithms** – A wide range of built-in machine learning algorithms optimized for performance and scalability
  - **Notebook Instances** - Managed Jupyter notebooks with pre-installed libraries, scalable compute, and data storage, making it easier to explore data and develop models.
  - **Training & Tuning** – Manages ML training infrastructure, Distributed training across multiple GPUs.
  - **Model deployment** – Deploy models to production in One click by creating endpoints, multi-model endpoints
  - **SageMaker GroundTruth** – A data labelling service (Text, images, video labelling)
  - **SageMaker Model Monitor** - Continuously monitors the quality of your models in production, detecting drift in model performance.
  - **SageMaker Pipelines**: Facilitates the automation and orchestration of machine learning workflows, supporting the implementation of MLOps best practices.

# AWS AI and ML services - When to use what?

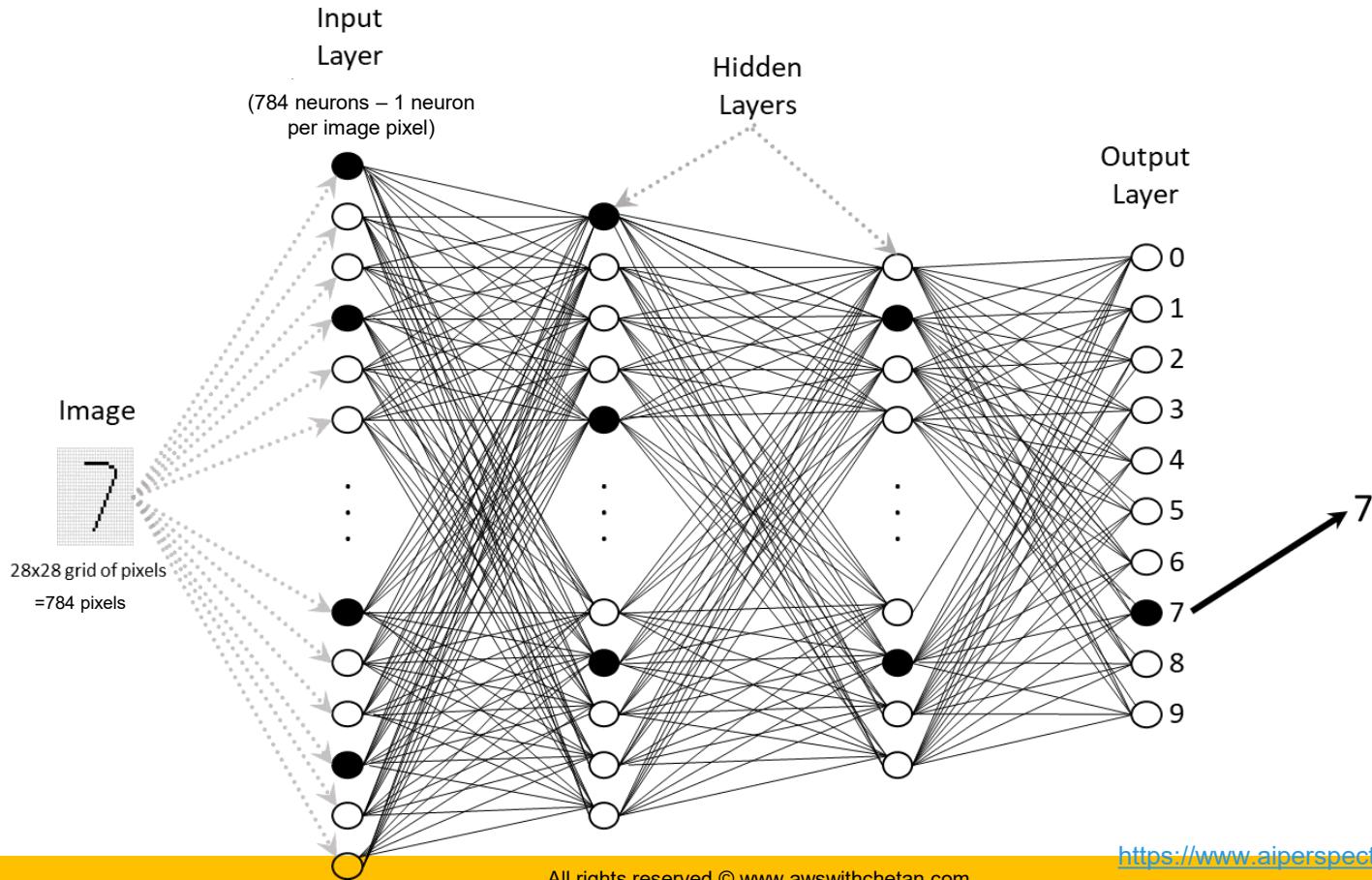


# AWS AI/ML- Summary

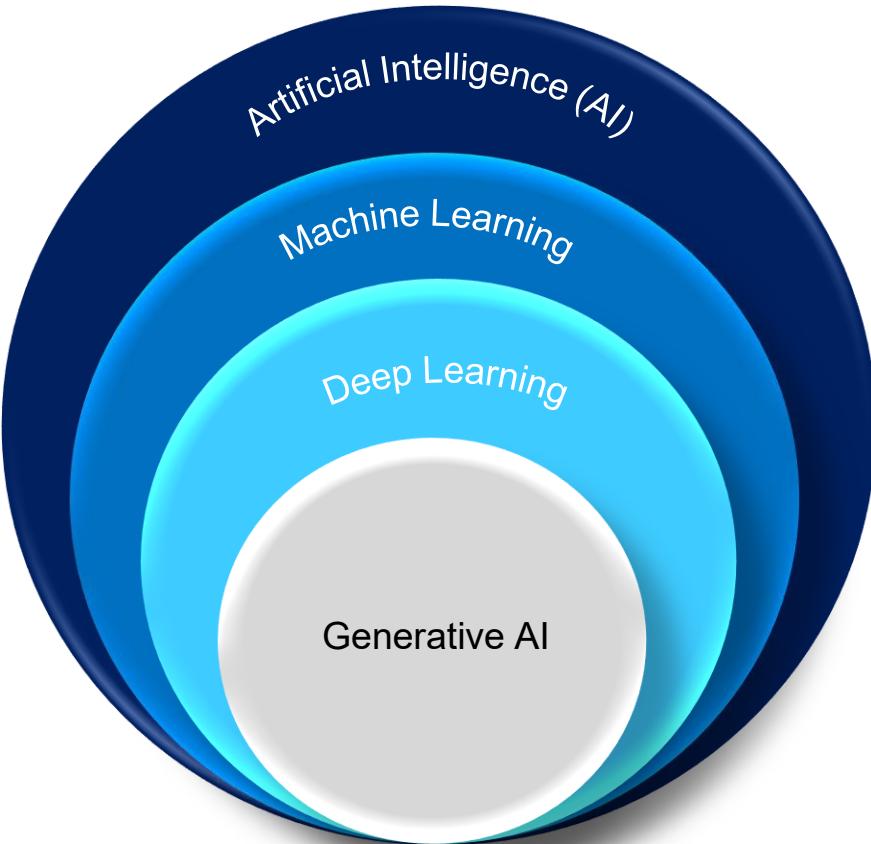
- **Amazon's three-layered approach to AI/ML** – Infrastructure, ML platform, AI services
- **Amazon Rekognition** – Computer vision-based object detection (Face, celebrity, text, scene) in Images and Videos
- **Amazon Transcribe** - Speech to text
- **Amazon Polly** - Text to Speech
- **Amazon Textract** – Extract text from scanned documents
- **Amazon Translate** – Translation from one language to other
- **Amazon Comprehend** – Entity extraction, sentiment analysis
- **Amazon Kendra** – Enterprise search engine for employees and customers
- **Amazon Lex** - Build conversational chatbots
- **Amazon Connect** - Cloud contact center
- **Amazon Personalize** – Personalized recommendation engine
- **Amazon Forecast** - Build highly accurate forecasts using ML models used by Amazon.com
- **Amazon Fraud Detector** – Detect potential fraud in transactions, payments
- **Amazon SageMaker** - Machine learning platform for developers and data scientists

# AI/ML, Deep Learning and Generative AI

# Deep Learning example..

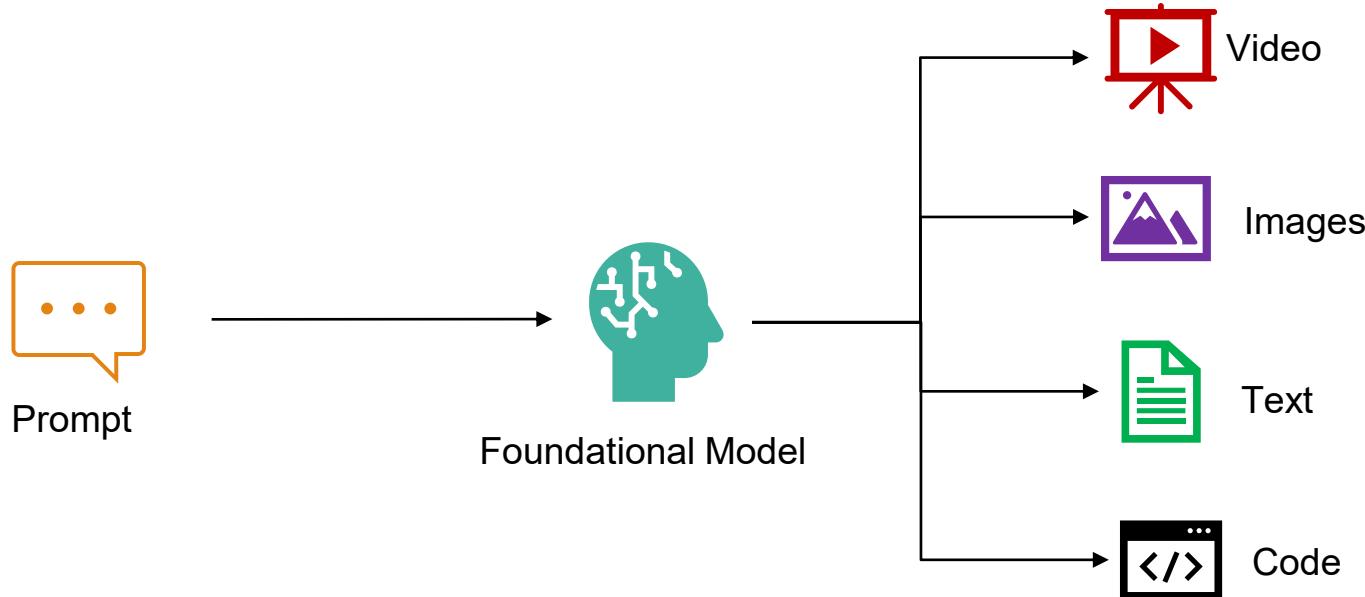


# AI, ML and GenAI

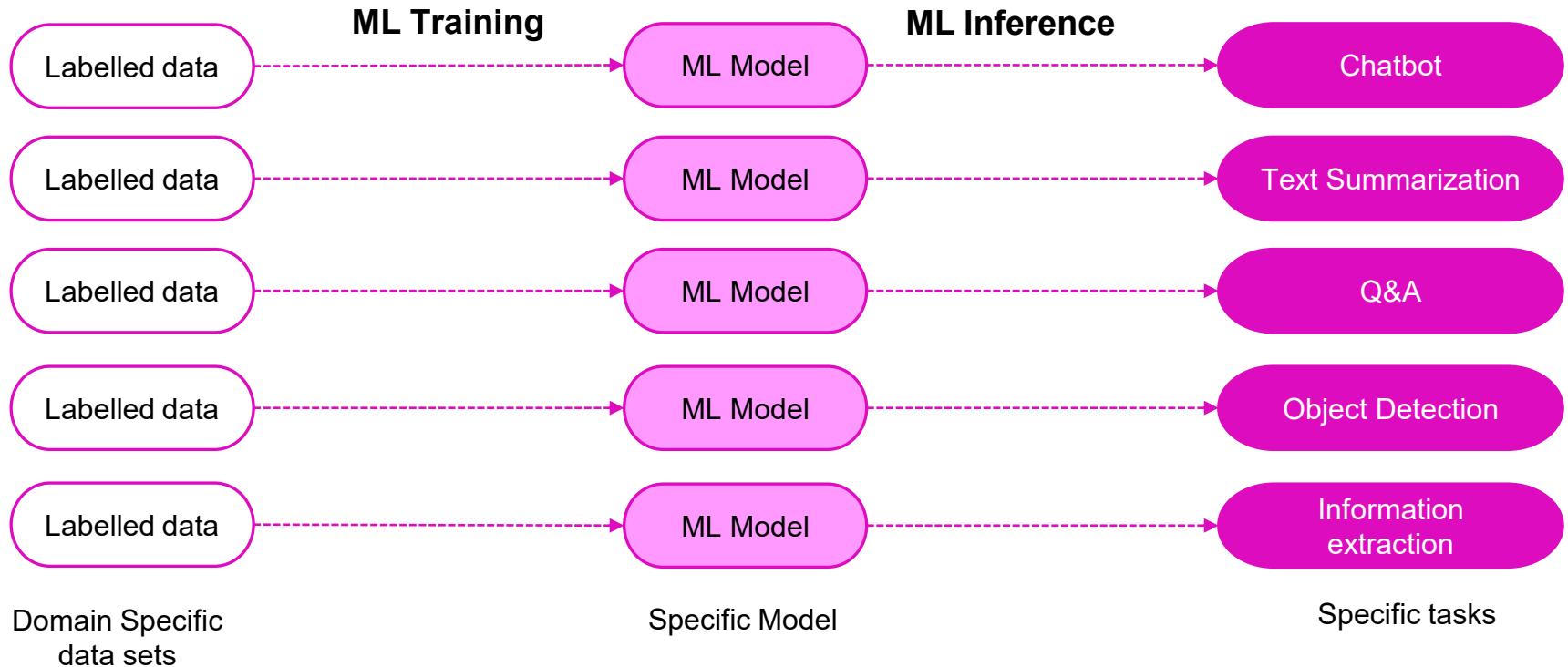


# What is Generative AI?

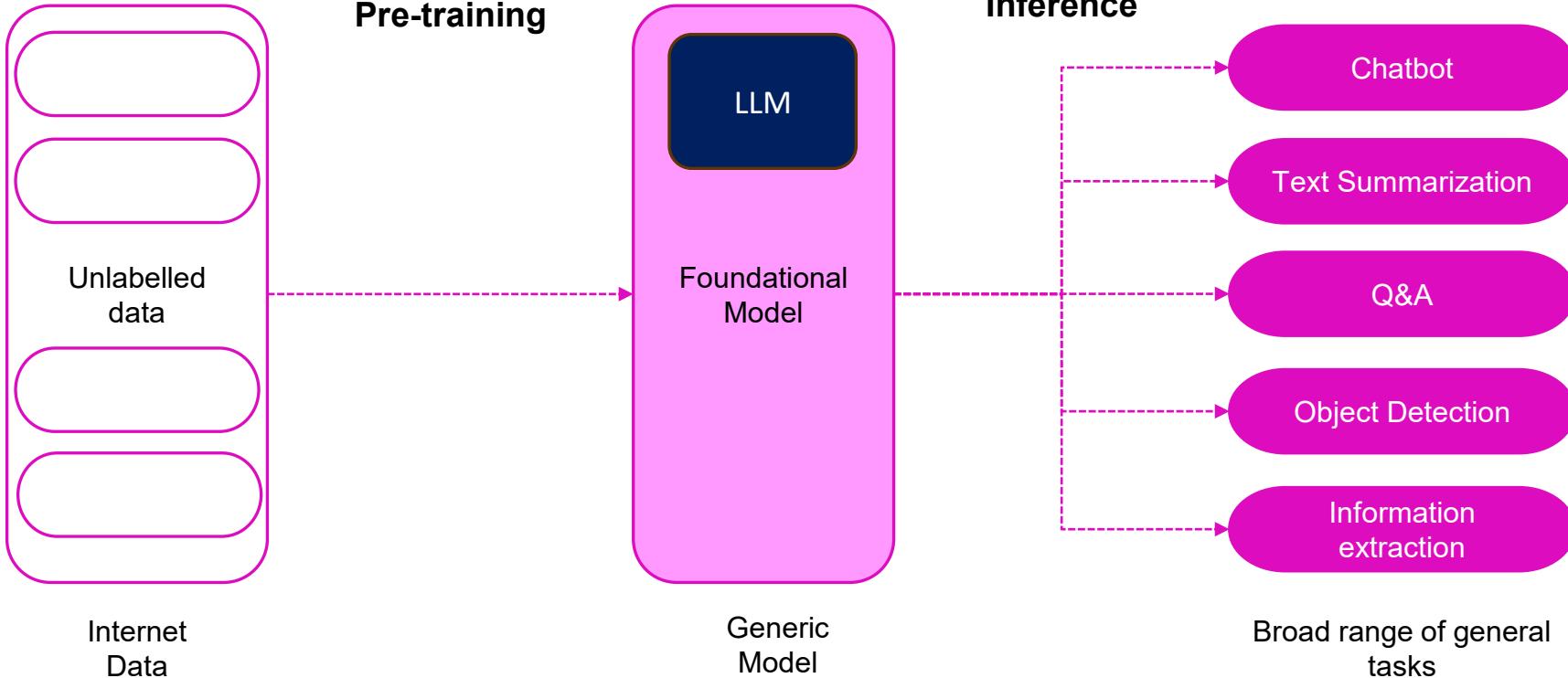
Generative AI is a type of artificial intelligence that can create original content such as text, images, video, audio or software code in response to a user's prompt or request.



# Traditional ML models

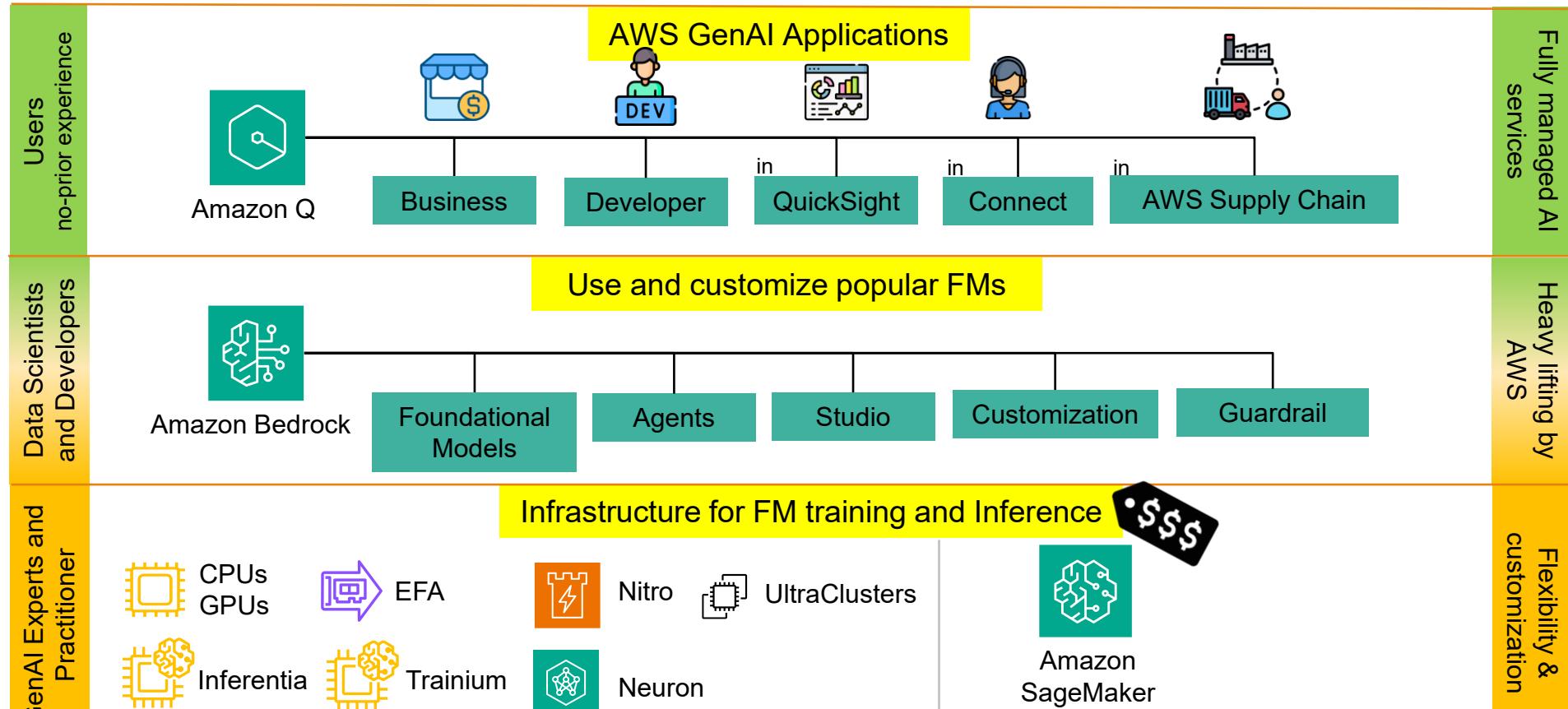


# Foundational models



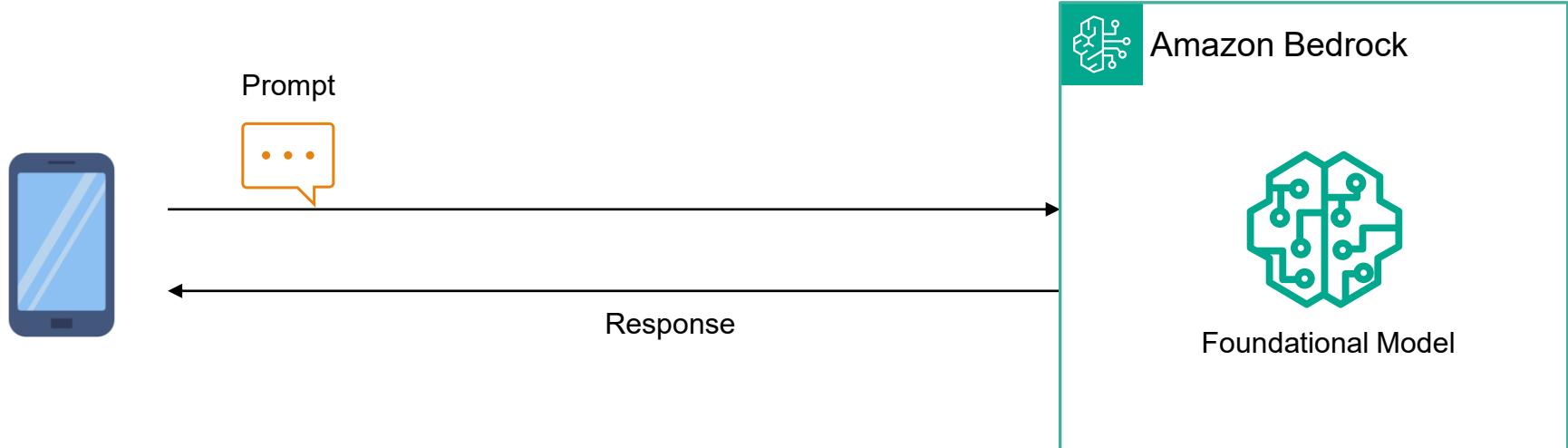
# AWS Generative AI Services

# AWS' three layered approach to GenAI



# Amazon Bedrock

Build and Scale generative AI applications with Foundational Models (FMs)





# Amazon Bedrock

Build and Scale generative AI applications with Foundational Models (FMs)

## stability.ai



## ANTHROPIC



## AI21 labs



### Choice of leading FMs

Block undesirable topics

Filter harmful content

Redact sensitive (PII) information

Detect hallucination



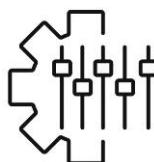
### Responsible AI

Prompt Engineering

Retrieval Augmented Generation (RAG)

Fine tuning

Continued pretraining



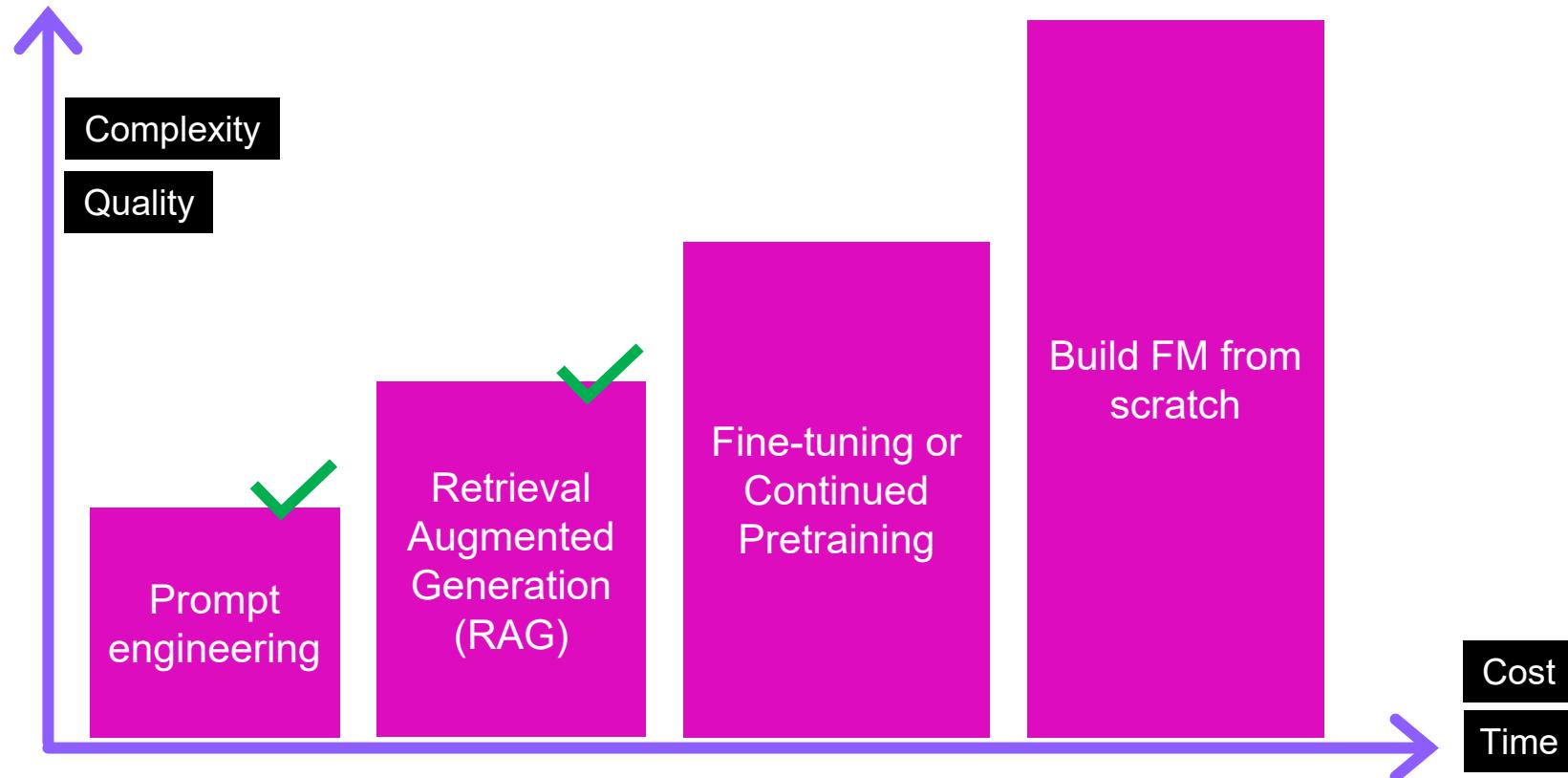
### Model Customization



### Agents

- Access FMs from leading providers over a single API
- Data Privacy & Security Guardrails
- Provide FMs domain specific context and information
- Access company systems and data sources
- Take actions

# Approaches to customize FMs



# Prompt Engineering

Prompts are a specific set of inputs provided by you, the user, that guide LLMs on Amazon Bedrock to generate an appropriate response or output for a given task or instruction.

## Few-shot prompting

**User prompt:** Tell me the sentiment of the following headline and categorize it as either positive, negative or neutral.

Here are some examples:

*Research firm fends off allegations of impropriety over new technology.*

Answer: Negative

*Offshore windfarms continue to thrive as vocal minority in opposition dwindles.*

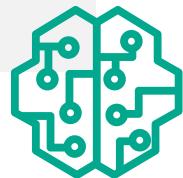
Answer: Positive

*Manufacturing plant is the latest target in investigation by state officials.*

Answer:



User

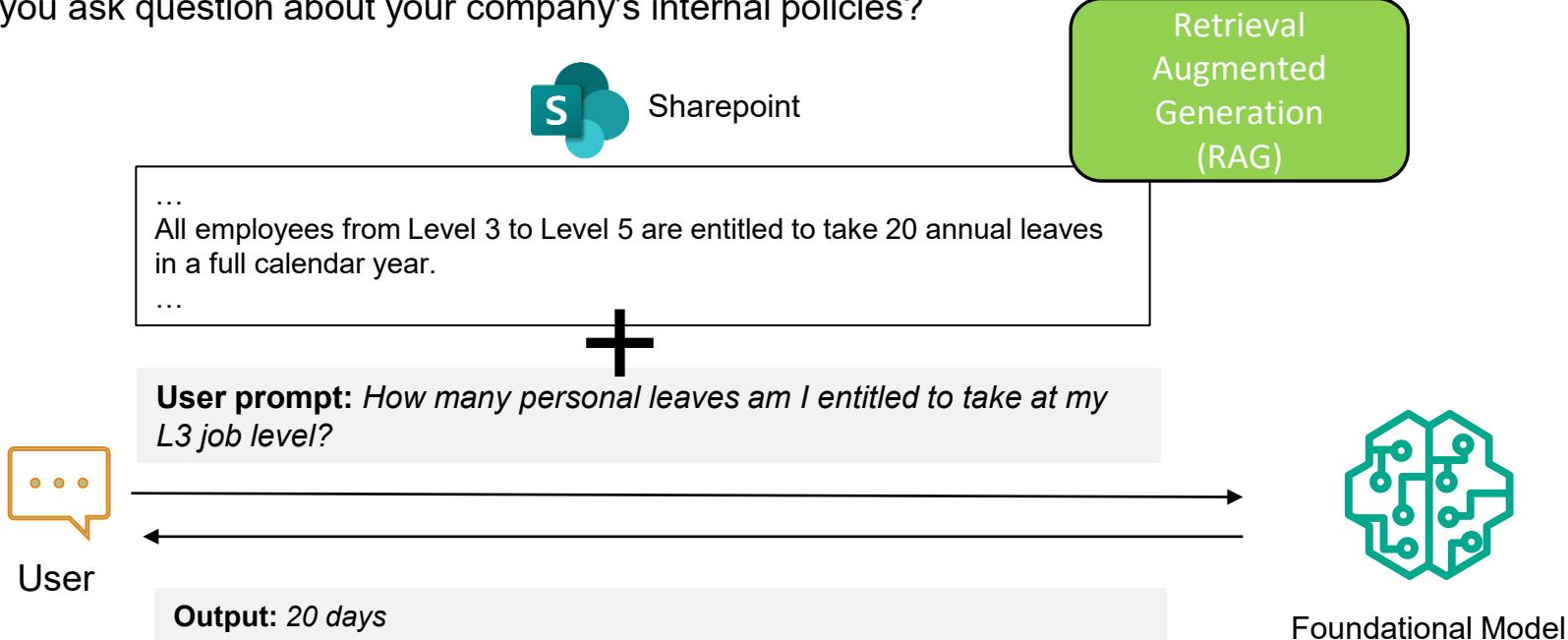


Foundational Model

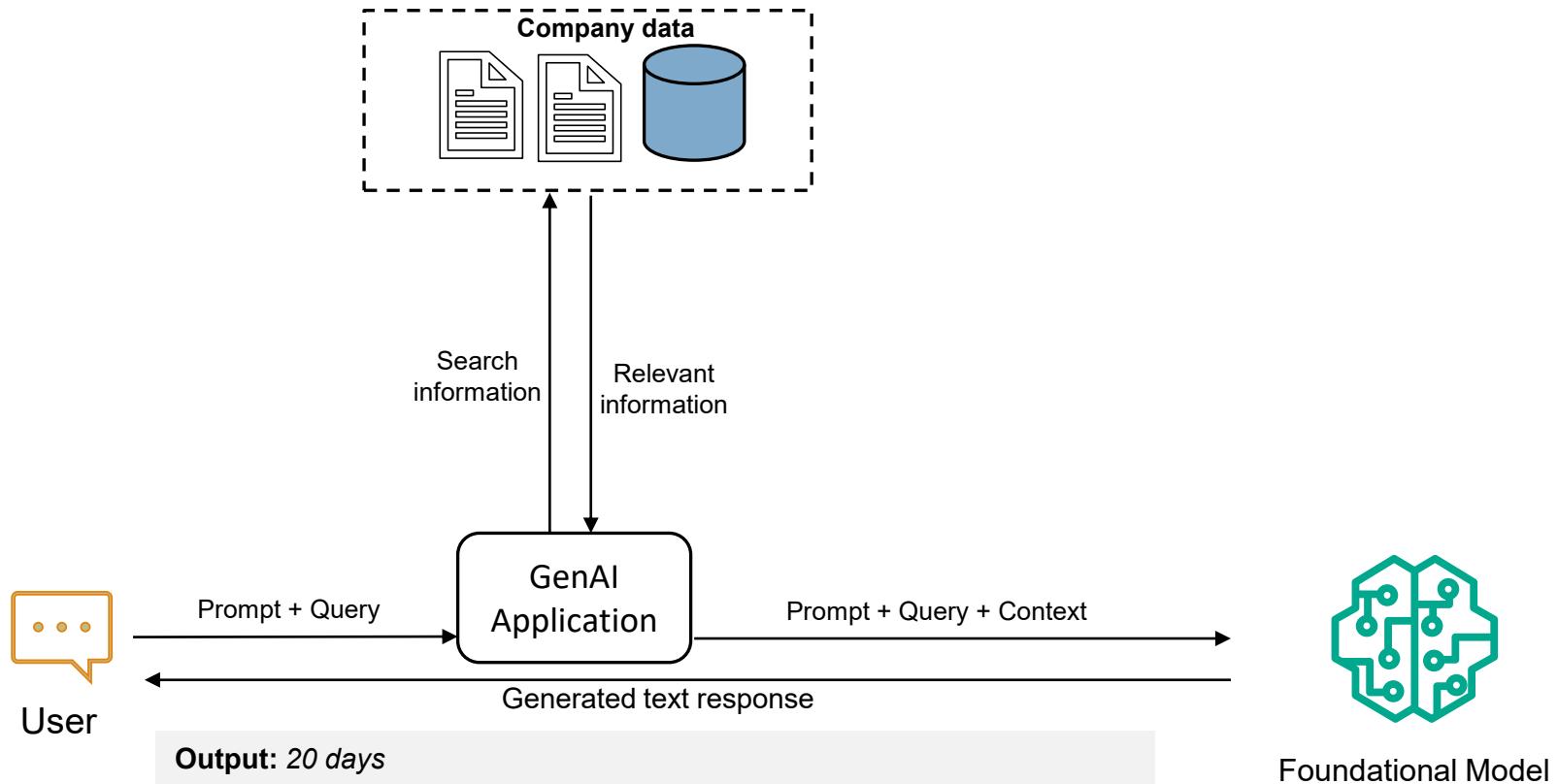
**Output: Negative**

# Retrieval Augmented Generation (RAG)

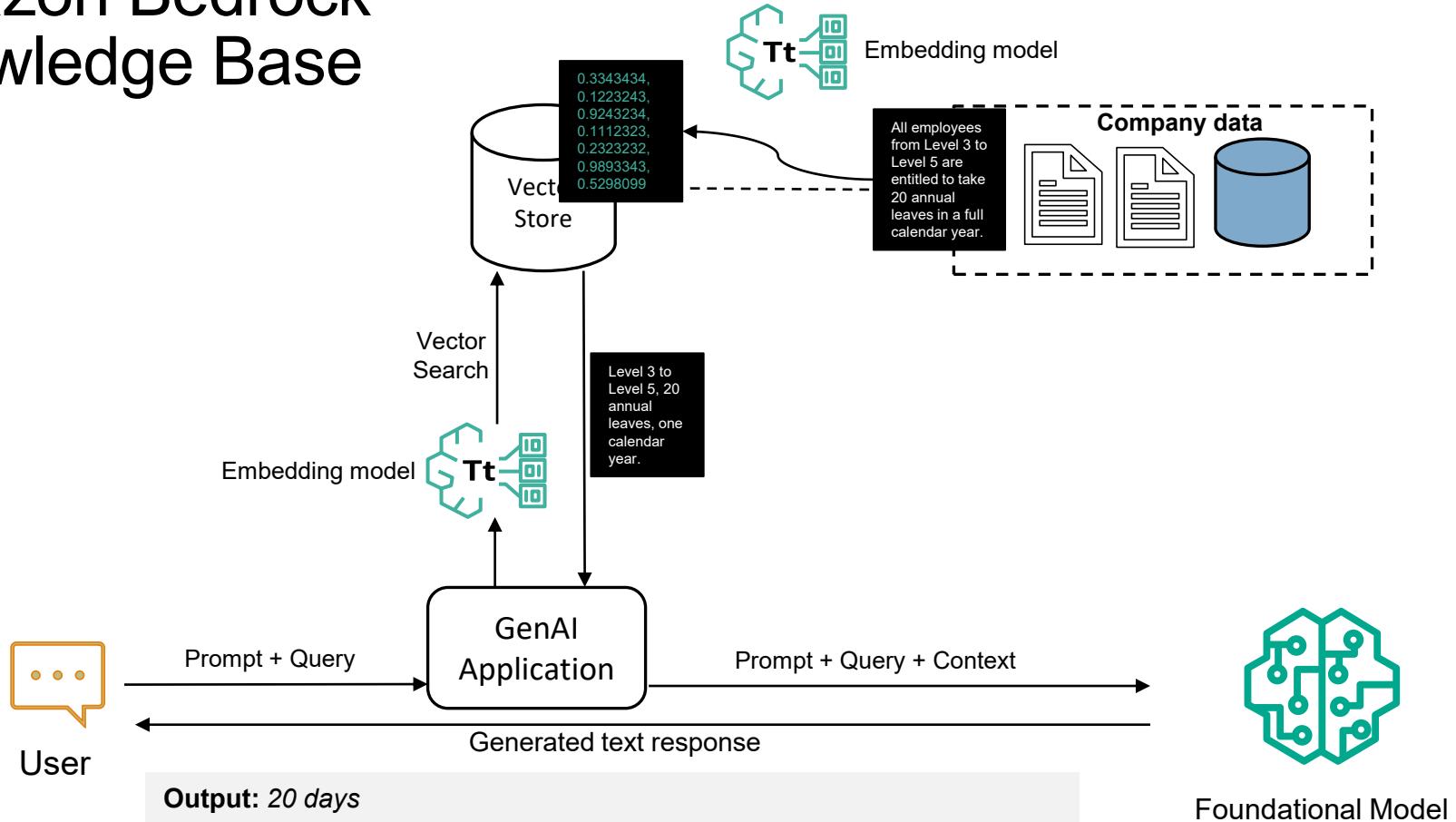
What if you ask question about your company's internal policies?



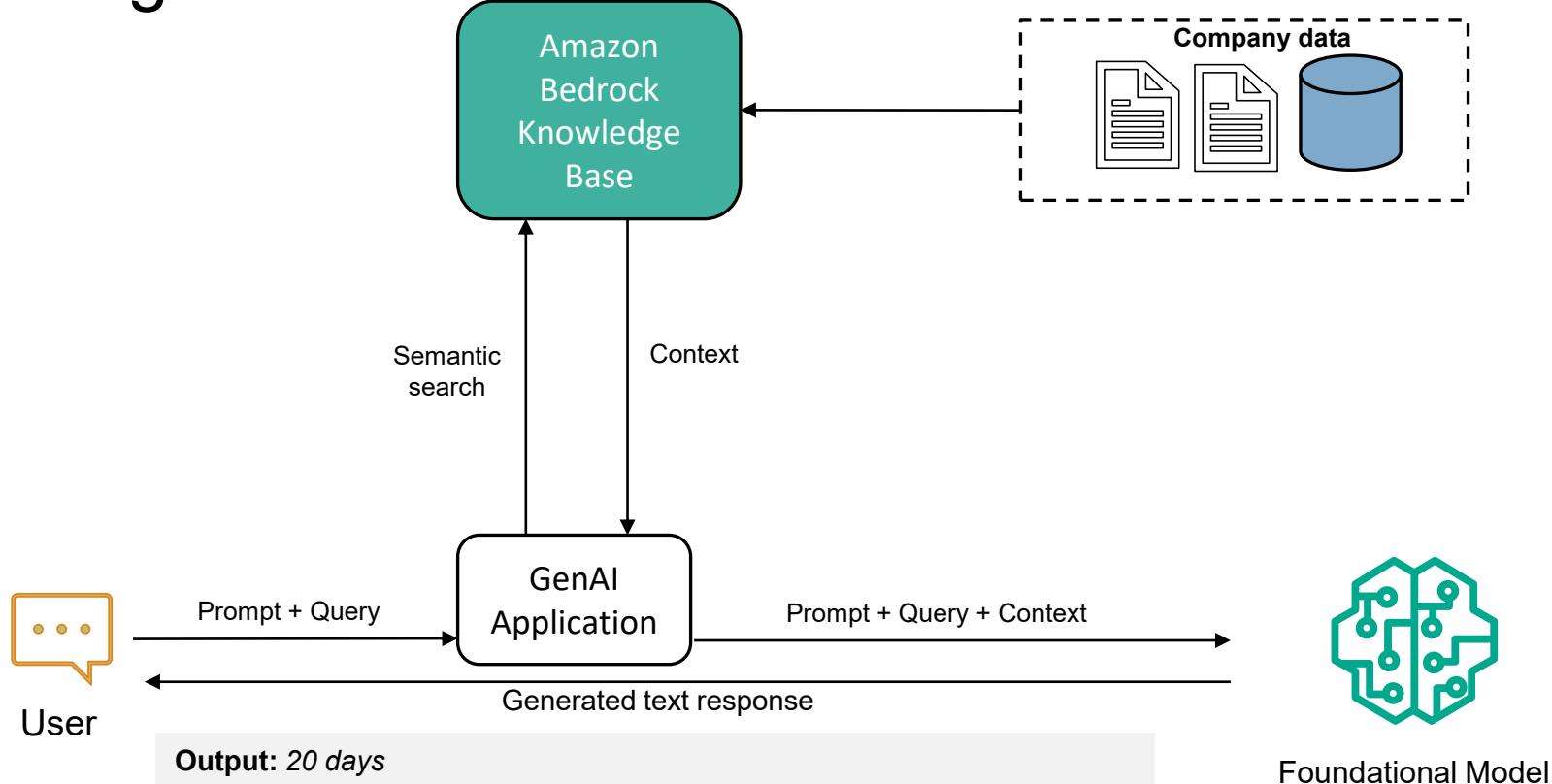
# How RAG works?



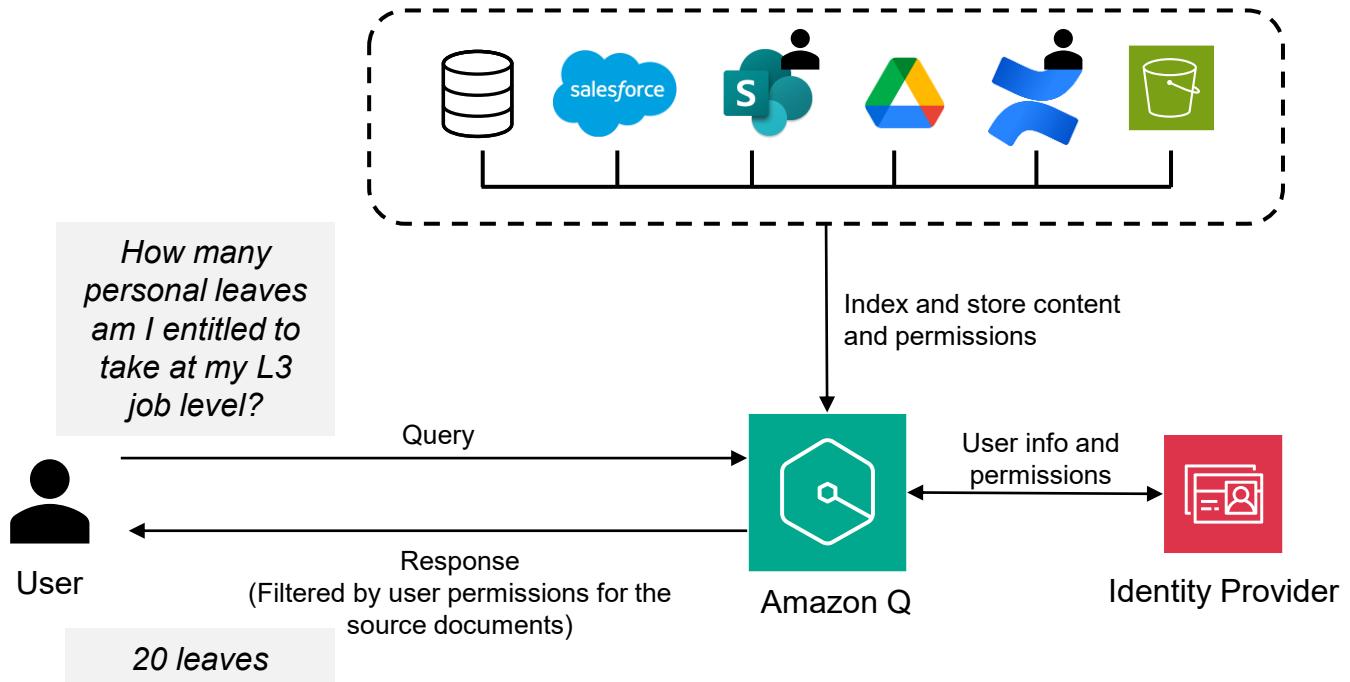
# Amazon Bedrock Knowledge Base



# Amazon Bedrock Knowledge Base



# Amazon Q Business





# Amazon Q Developer

AI-powered assistance for software development

Get Assistance

How can I build 3-tier serverless web application?

Modernize legacy code

/transform

Input JDK version

JDK8

Target JDK Version

JDK17

....

30 files modified

Code faster

```
import boto3
import logging
#function to upload local file to s3
def upload_file_to_s3(file_name, bucket, object_name=None):
    """Upload a file to an S3 bucket"""
    # Create S3 client
    s3 = boto3.client('s3')

    # If S3 object_name was not specified, use file_name
    if object_name is None:
        object_name = file_name

    # Upload the file
    s3.upload_file(file_name, bucket, object_name)

    print(f"File {file_name} uploaded to {bucket}/{object_name}")
```

# Amazon Q Developer

Explain existing code

```
class DataSyncStack(cdk.Stack):  
  
    def __init__(self, scope: Construct, id: str, **kwargs) -> None:  
        super().__init__(scope, id, **kwargs)  
  
        bucket_access_role = iam.Role(scope, 'chetan-gt-s3-role',  
                                       assumed_by=iam.ServicePrincipal('datasync.amazonaws.com')  
                                       )  
        policy = iam.Policy(  
            scope,  
            'BucketAccessPolicy',  
            statements=[  
                iam.PolicyStatement(  
                    actions=['s3:GetObject', "s3>ListObjectsV2"],  
                    resources=["arn:aws:s3:::chetan-gt-poc-s3/*"]  
                ),  
                iam.PolicyStatement(  
                    actions=['s3>ListBucket'],  
                    resources=["arn:aws:s3:::chetan-gt-poc-s3"],  
                )  
            ]  
        )  
        bucket_access_role.attach_inline_policy(policy)
```



...

...

Overall, this code sets up the necessary resources and configurations for an AWS DataSync task that synchronizes data from an S3 bucket to an Amazon EFS file system. It creates the required IAM role with the necessary permissions, defines the source and destination locations, and configures the DataSync task with specific options.

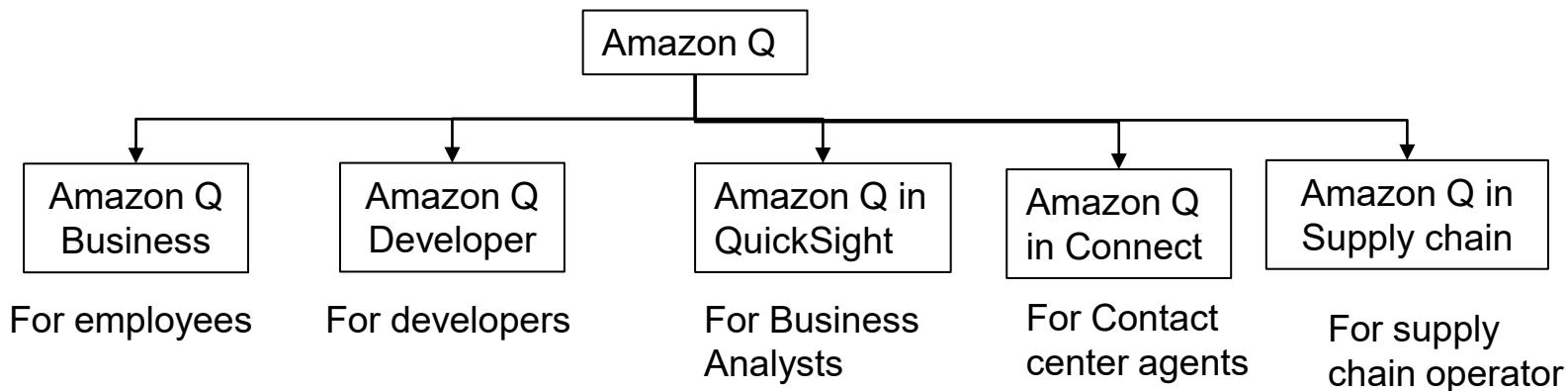
# AWS GenAI Summary

## Amazon Bedrock

- Amazon Bedrock is a fully managed Generative AI platform
- Offers a choice of high-performing foundation models (FMs) from leading AI companies like AI21 Labs, Anthropic, Cohere, Meta, Mistral AI, Stability AI, and Amazon through a single API

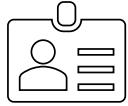
## Amazon Q

- Generative AI-powered assistant for accelerating software development and leveraging companies' internal data



# AWS Security services

# AWS Security services



## Identity and access management

 AWS IAM

AWS IAM Identity Center

AWS Directory Service

Amazon Cognito



## Data protection

AWS Key Management Service (KMS)

AWS CloudHSM

AWS Certificate Manager

AWS Secrets Manager

Amazon Macie



## Infrastructure protection

 VPC Security Groups and NACL

AWS WAF

AWS Shield

AWS Firewall Manager

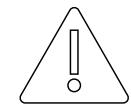


## Detective controls

Amazon GuardDuty

Amazon Inspector

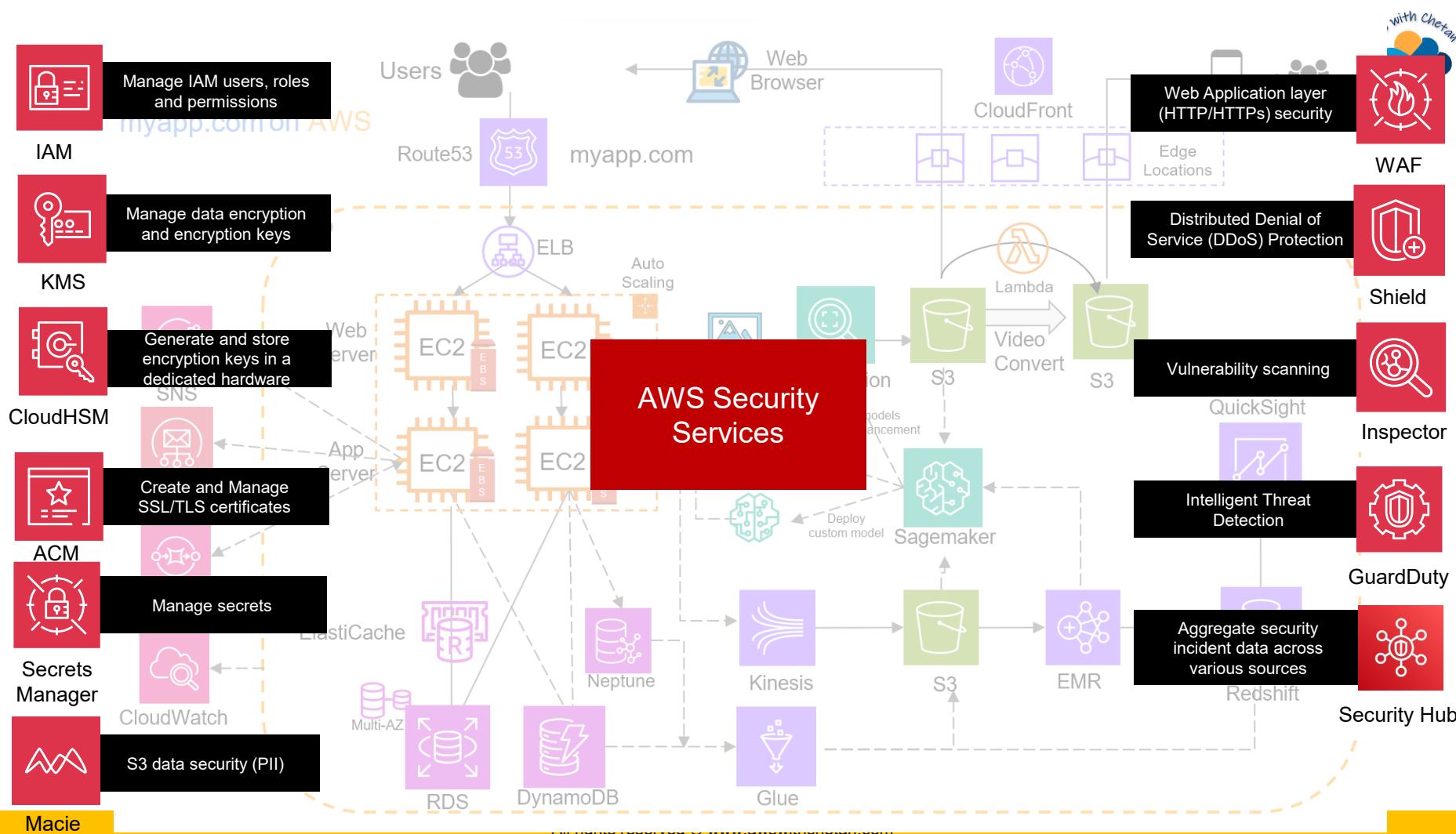
AWS Network Firewall



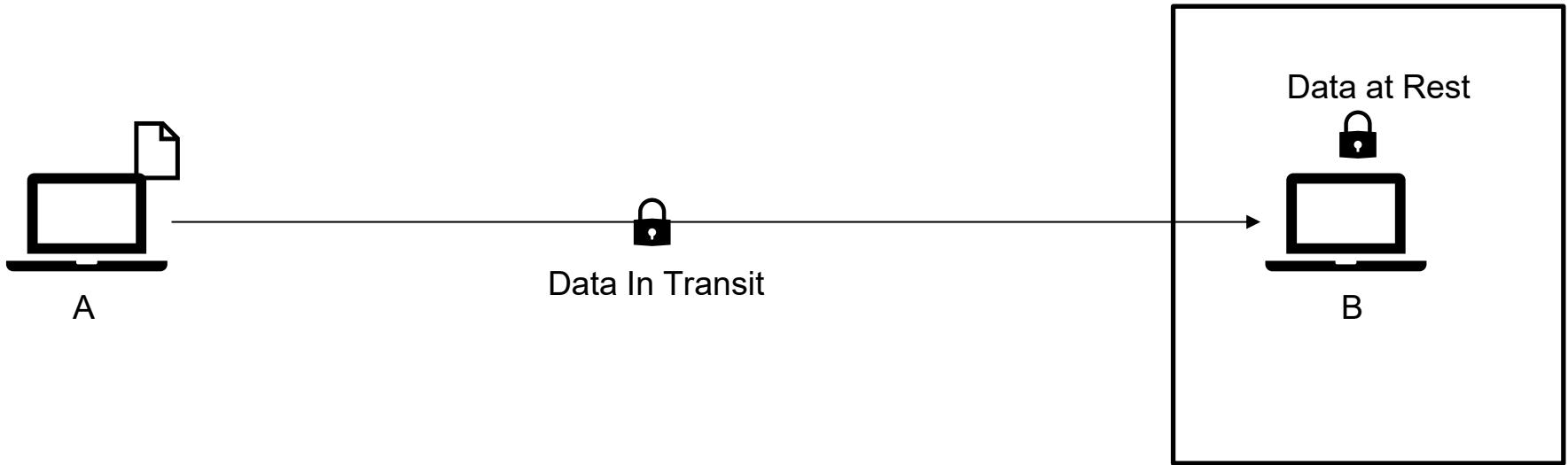
## Incidence Response

Amazon Detective

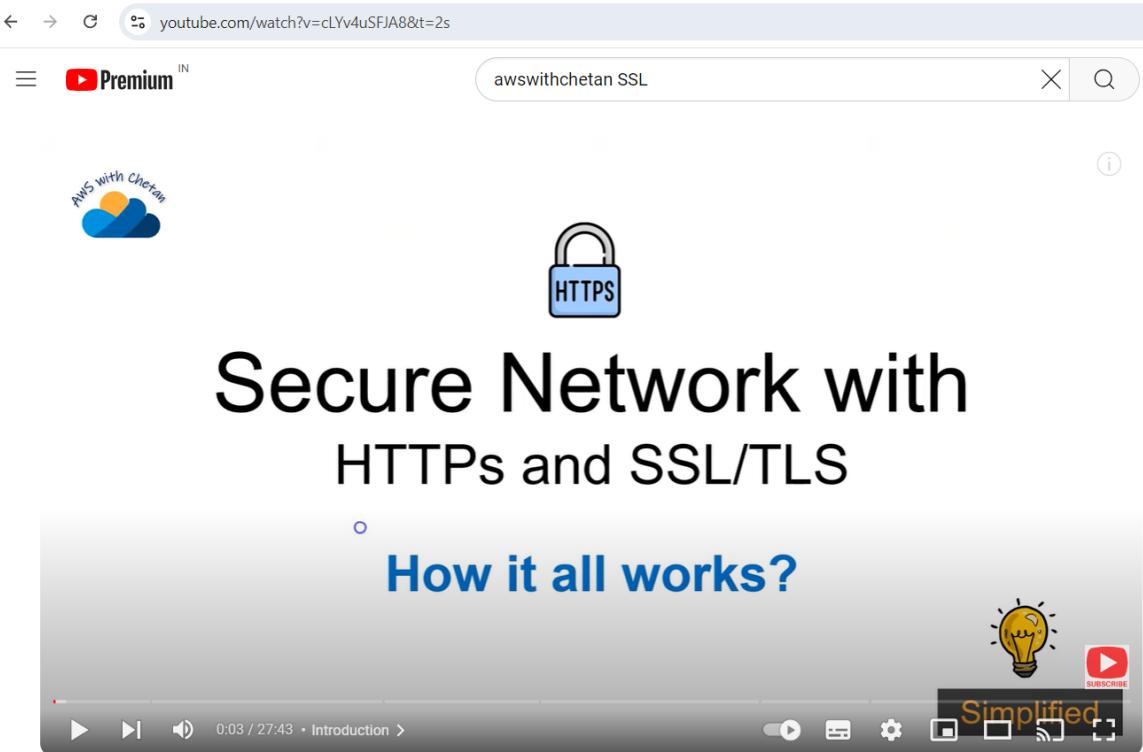
AWS Security Hub



# Data Security

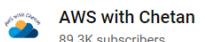


# Securing data in transit with SSL/TLS/HTTPS



The screenshot shows a YouTube video player with the URL [youtube.com/watch?v=cLYv4uSFJA8&t=2s](https://youtube.com/watch?v=cLYv4uSFJA8&t=2s) in the address bar. The search bar contains "awswithchetan SSL". The video thumbnail features the "AWS with Chetan" logo. The main video frame displays a large blue padlock icon with the word "HTTPS" next to it. Below the padlock, the title "Secure Network with HTTPS and SSL/TLS" is centered. A blue circular progress bar indicates the video is at 0:03 of 27:43. The video player interface includes standard controls like play/pause, volume, and a progress bar. In the bottom right corner of the video frame, there is a lightbulb icon with "Simplified" text and a red "SUBSCRIBE" button.

How HTTPS, SSL/TLS actually work?



Analytics

Edit video

263

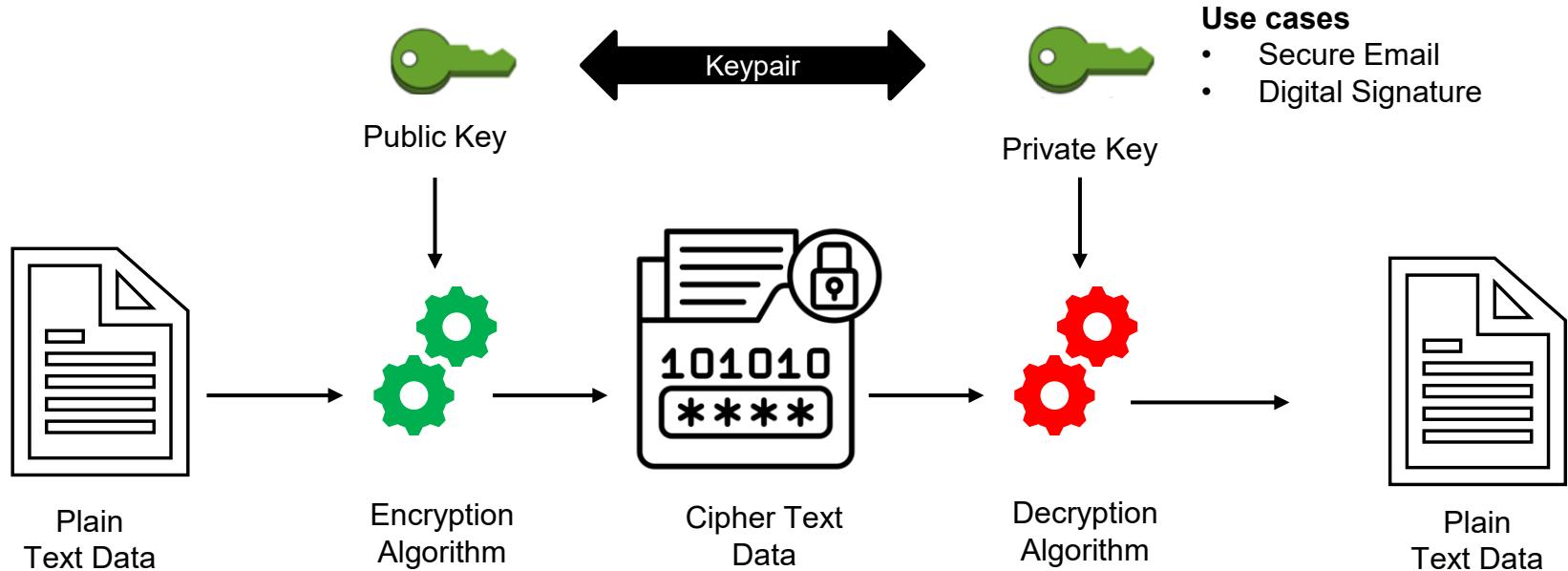


Share

Promote

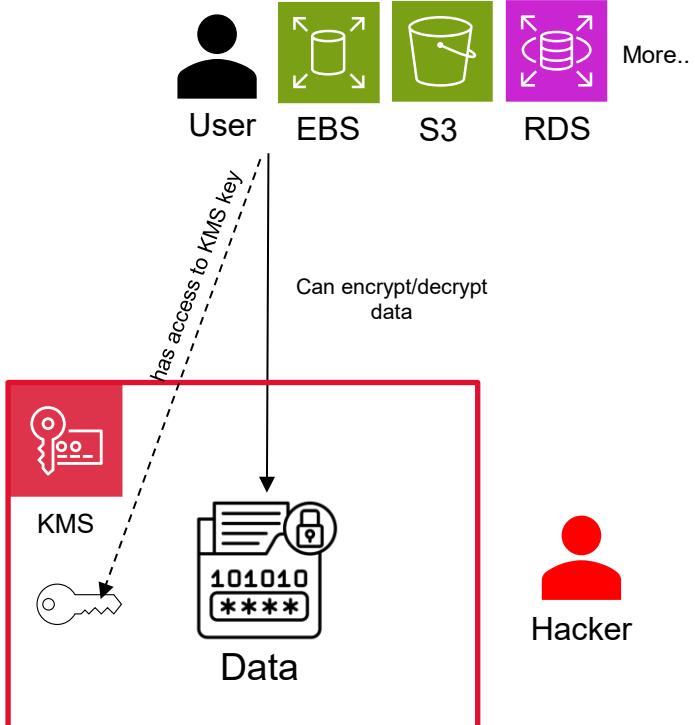
...

# Data at rest security – Asymmetric Encryption



# AWS Key Management Service (KMS)

- AWS KMS is a data encryption and key management service
- Supports both **Symmetric** and **Asymmetric** keys
- Supports various encryption algorithms (e.g. AES, ECC etc.) and keys (RSA-2048, ECC NIST P-256 etc.)
- Most of the AWS services natively integrates with KMS to provide built in encryption: **EBS**, **S3**, **Redshift**, **EFS**, **RDS** etc. For services like CloudTrail Logs, S3 Glacier storage, CloudWatch logs encryption is automatically enabled.
- User or AWS service needs IAM permissions to use KMS service and encryption keys.
- KMS Keys can be shared with other AWS accounts
- KMS Keys are regional but you can also create Multi-region KMS keys to replicate keys across regions for DR, backup etc.



# AWS CloudHSM

- Single tenant (dedicated) **Hardware Security Module** to generate and use encryption keys
- Provides secure key storage in a **tamper-resistant hardware device FIPS 140 -2 Level 3 compliance.**
- Helps you meet corporate, contractual, and regulatory compliance requirements for data security.
- Unlike KMS you control and manage your own encryption keys



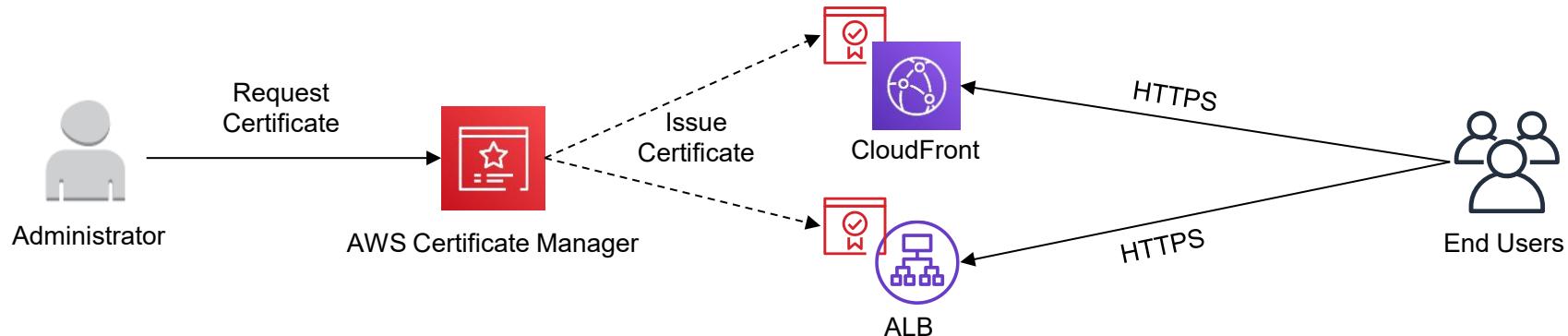
AWS CloudHSM



Sample HSM device

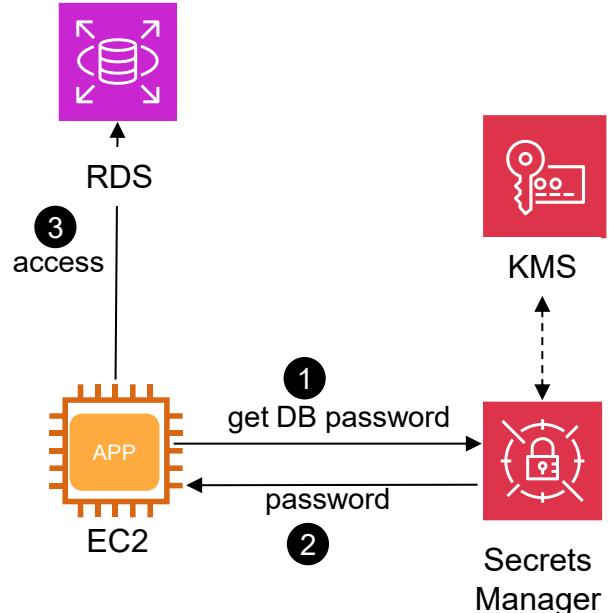
# Amazon Certificate Manager (ACM)

- Provides Public and Private SSL/TLS (X.509 standard) certificates for secure communication
- Request Public SSL/TLS certificate for your web applications or Private certificates for internal communication.
- For Public certificates ACM validates domain ownership (through email or DNS) and issues certificate which you can then associate with AWS resource such as ALB, CloudFront, ElasticBeanstalk, API gateway etc.
- You can also request wildcard certificate e.g. \*.example.com which can be used for subdomains
- Public certificates are FREE and Private certificate has cost per Certificate Authority (CA).



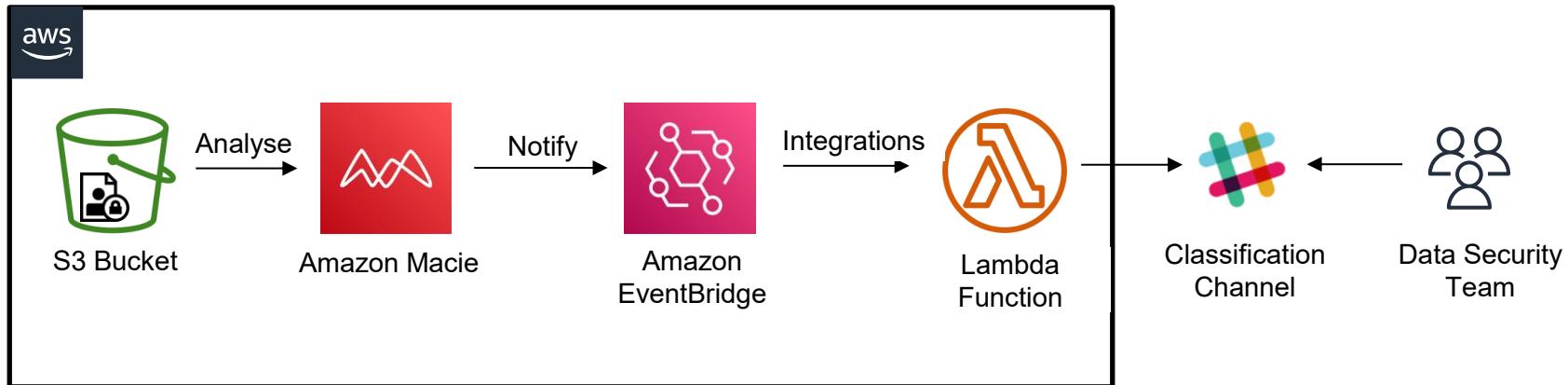
# AWS Secrets Manager

- Helps you to securely store, retrieve and rotate credentials for your databases and other services.
- You can store passwords, API keys, Tokens or any other login credentials.
- No need to hardcode credentials in the application code.
- Rotate credentials automatically for integrated services (e.g. RDS - MySQL, PostgreSQL, Aurora etc. ) or invoke Lambda function to rotate the credentials.
- Secrets are encrypted using KMS
- Access to secrets is controlled using IAM permissions

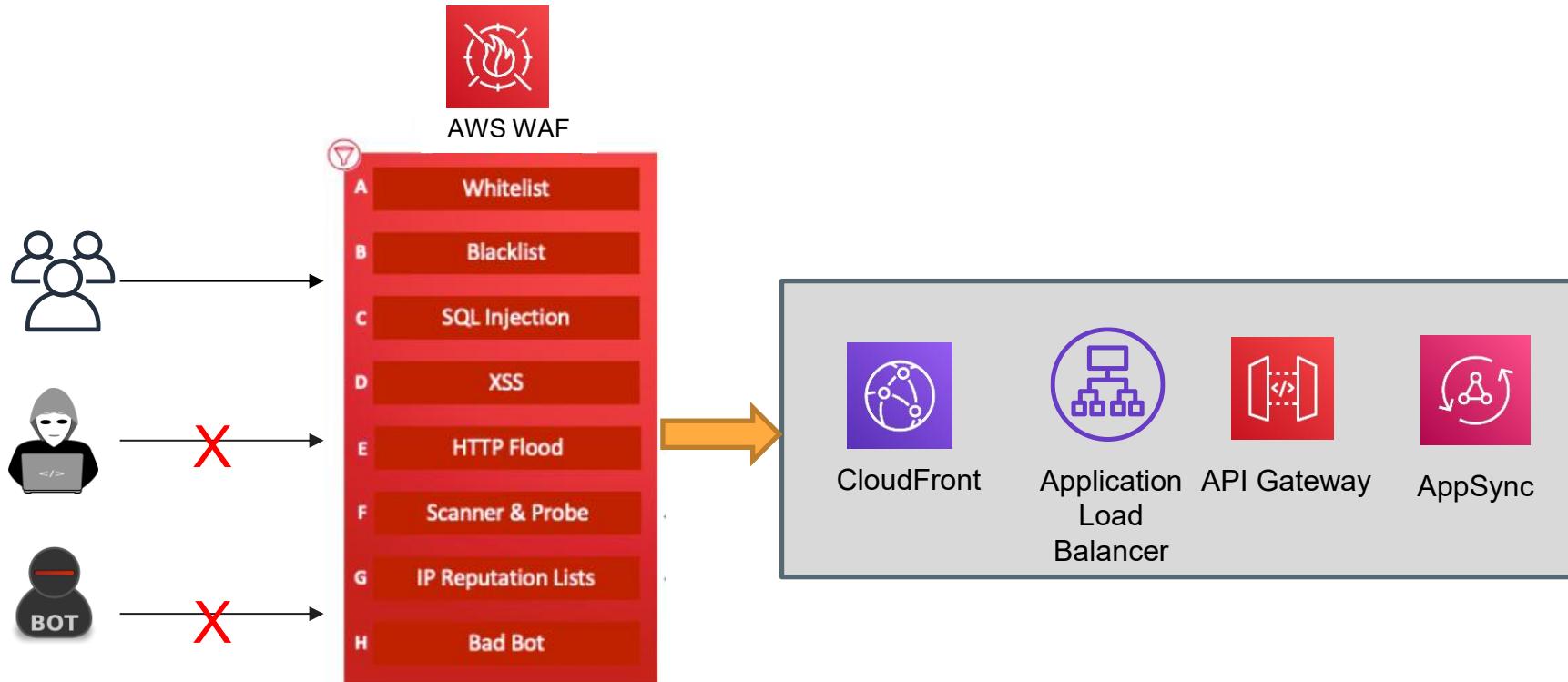


# Amazon Macie

- Fully managed S3 data security service
- Can discover **Personally Identifiable Information (PII)** data
- Uses machine learning and pattern matching to discover, monitor, and protect your data in S3
- Triggers alert when sensitive data is identified



# AWS Web Application Firewall (WAF)



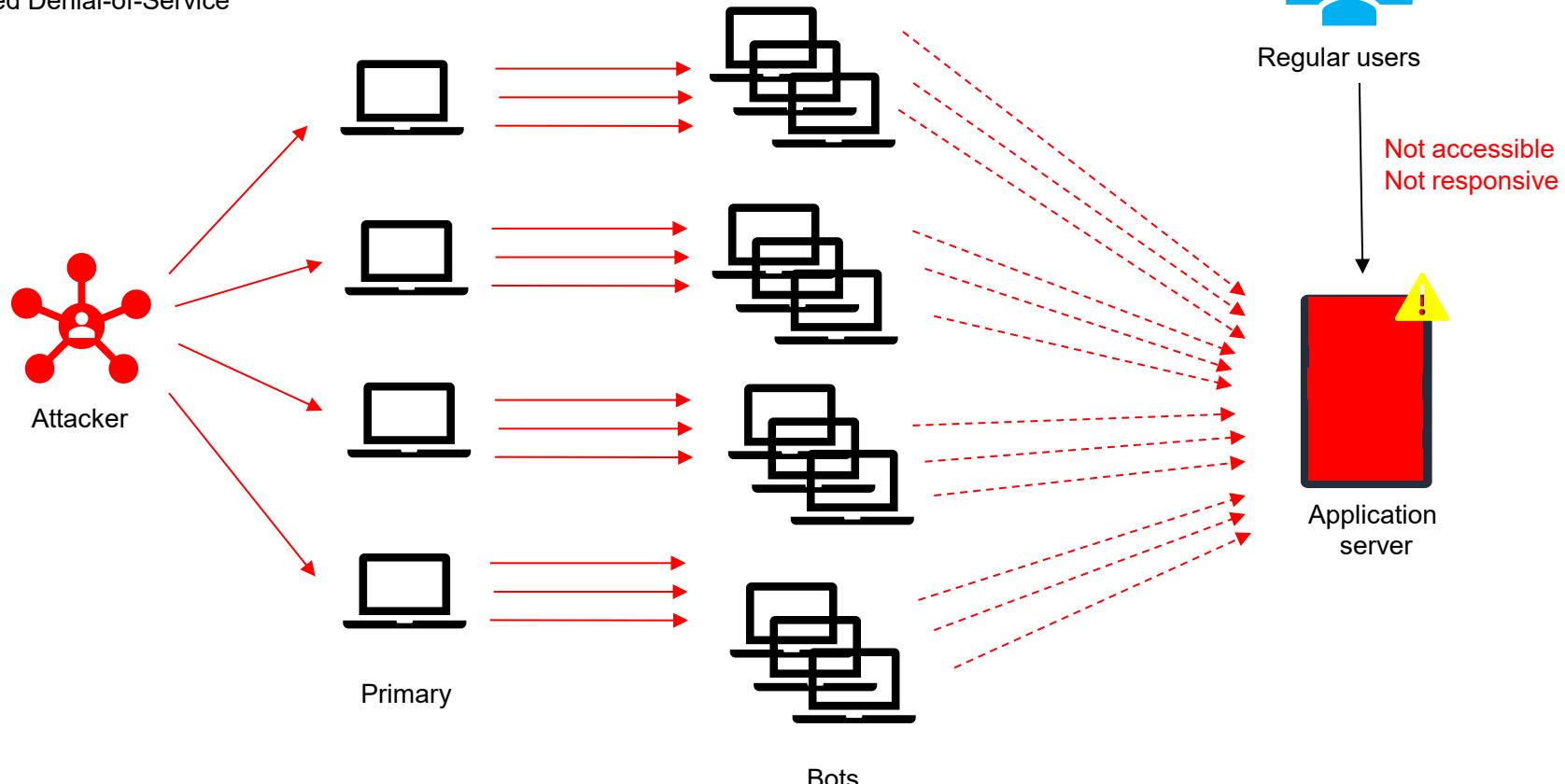
# AWS WAF – Web Application Firewall



- Protects web applications from common web exploits (**Layer 7**) – **OWASP Top 10**
- Deploy on **Application Load Balancer**, **API Gateway** or **AppSync** GraphQL APIs and **CloudFront**
- Define Web ACL (Web Access Control List):
  - Block IP addresses
  - Filter traffic based on HTTP headers, HTTP body, or URI strings
  - Detect and block common attacks - SQL injection and Cross-Site Scripting (XSS)
  - Geo-match – allow or block countries or regions
  - Rate-based rules (to count occurrences of events) – for DDoS protection
- Can present CAPTCHA or challenge to prevent bot attacks
- On blocking the malicious traffic WAF returns HTTP 403 status code (Forbidden)

# What's a DDOS attack?

Distributed Denial-of-Service

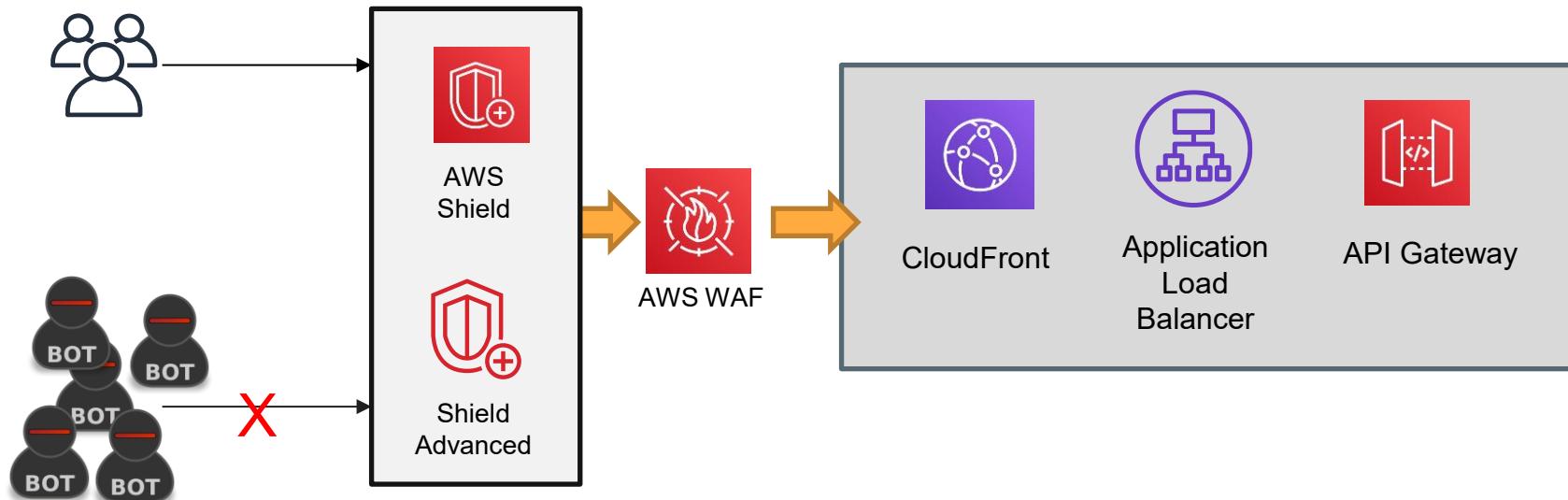


# Common DDoS attacks

- SYN Flood attack: Too many half open TCP connections
- UDP Flood attack: Too many UDP requests
- UDP Reflection attack: Spoof the victim server IP as a source for UDP packet. Victim server receives the unexpected responses.
- DNS Flood attack: Overwhelm the DNS so legitimate users can't find the site
- Slow Loris attack (Layer7): A lot of HTTP connections are opened and maintained
- Cache Busting attack: Request un-cached data from CDN

# AWS Shield

Provides protection against DDoS attack at Network Layer (layer3) and Transport Layer (layer4)



# AWS Shield

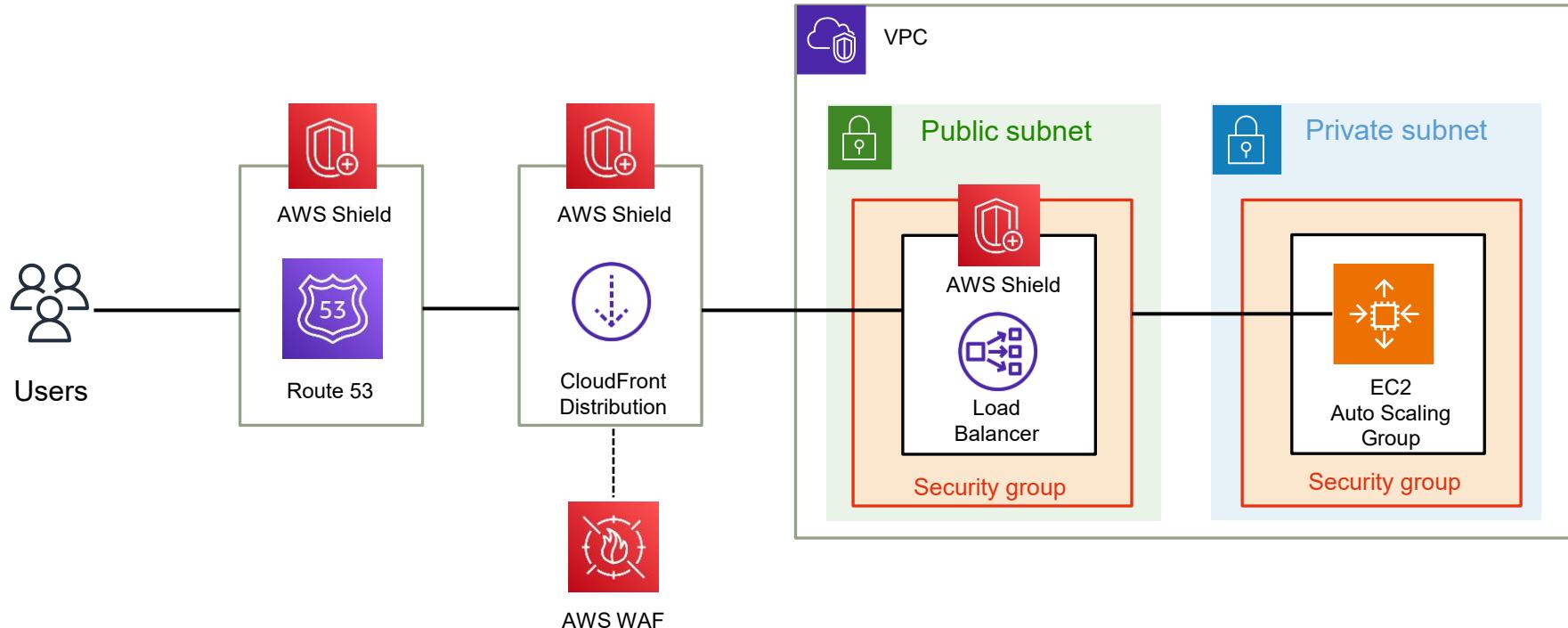
## AWS Shield Standard:

- Free service that is activated for every AWS customer

## AWS Shield Advanced:

- Optional DDoS mitigation service (\$3,000 per month per organization)
- Protect against more sophisticated attack on Amazon EC2, Elastic Load Balancing (ELB), Amazon CloudFront, AWS Global Accelerator, and Route 53
- **24/7 access to AWS Shield Response Team (SRT)**
- Protect against higher fees during usage spikes due to DDoS

# Reference Architecture for DDoS Protection

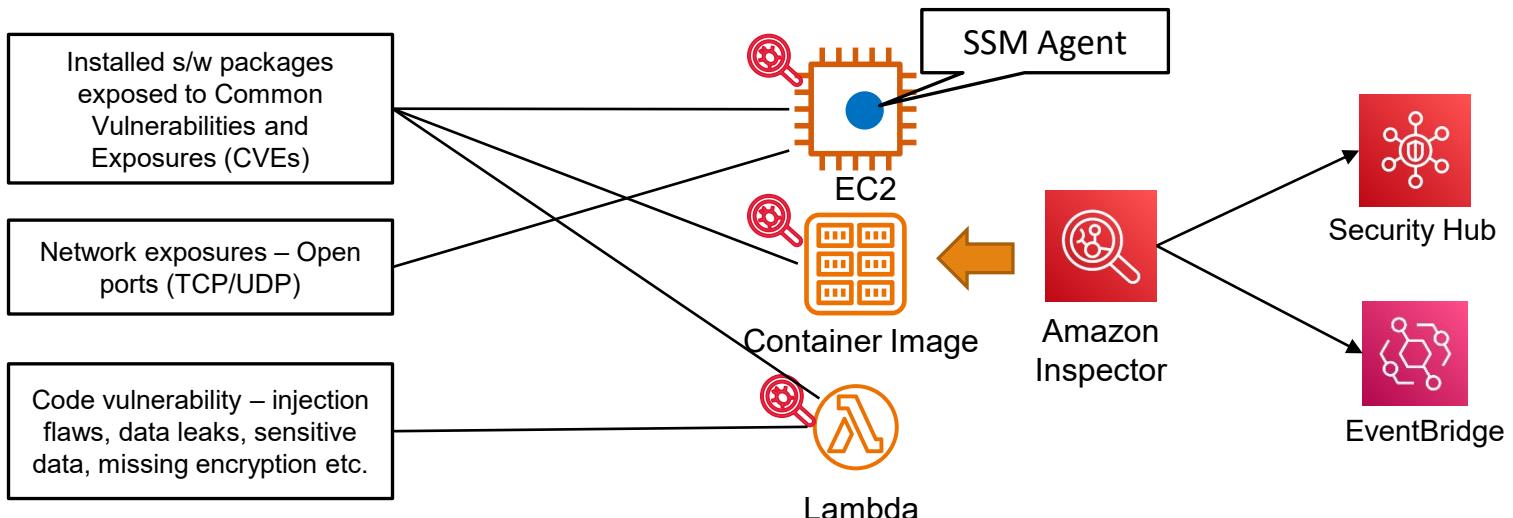


# Amazon Inspector

S with Chetan



- Scans EC2 instances, Container Images & Lambda functions for vulnerabilities (CVEs)
- Publishes findings to Amazon EventBridge and AWS Security Hub for reporting and action



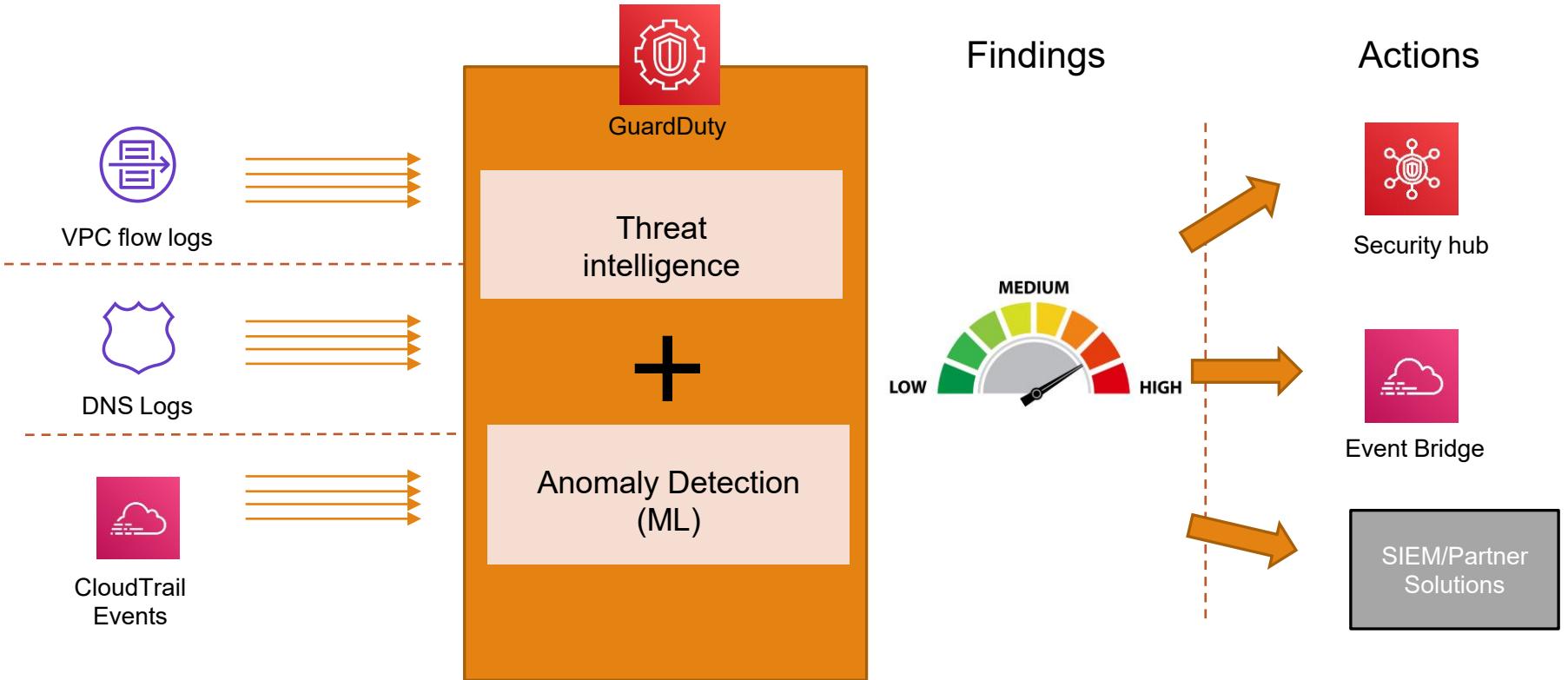
# Amazon GuardDuty

- Intelligent threat detection service - Enable with just “One click”
- Uses Machine learning to detect threats more accurately
- Can detect sophisticated attacks e.g. EC2 instance being used for crypto currency or Bitcoin mining, data exfiltration, unusual access to malicious IP etc.
- Analyses tens of billions of events across multiple AWS data sources such as:
  - **AWS CloudTrail logs:** Unusual API calls, unauthorized deployments.
  - **VPC Flow Logs:** Unusual internal traffic, unusual IP address.
  - **DNS query logs:** Compromised EC2 instances sending encoded data within DNS queries.
- Notifies the security findings:
  - In the GuardDuty console
  - Amazon EventBridge
  - AWS Security Hub



Amazon GuardDuty

# Amazon GuardDuty



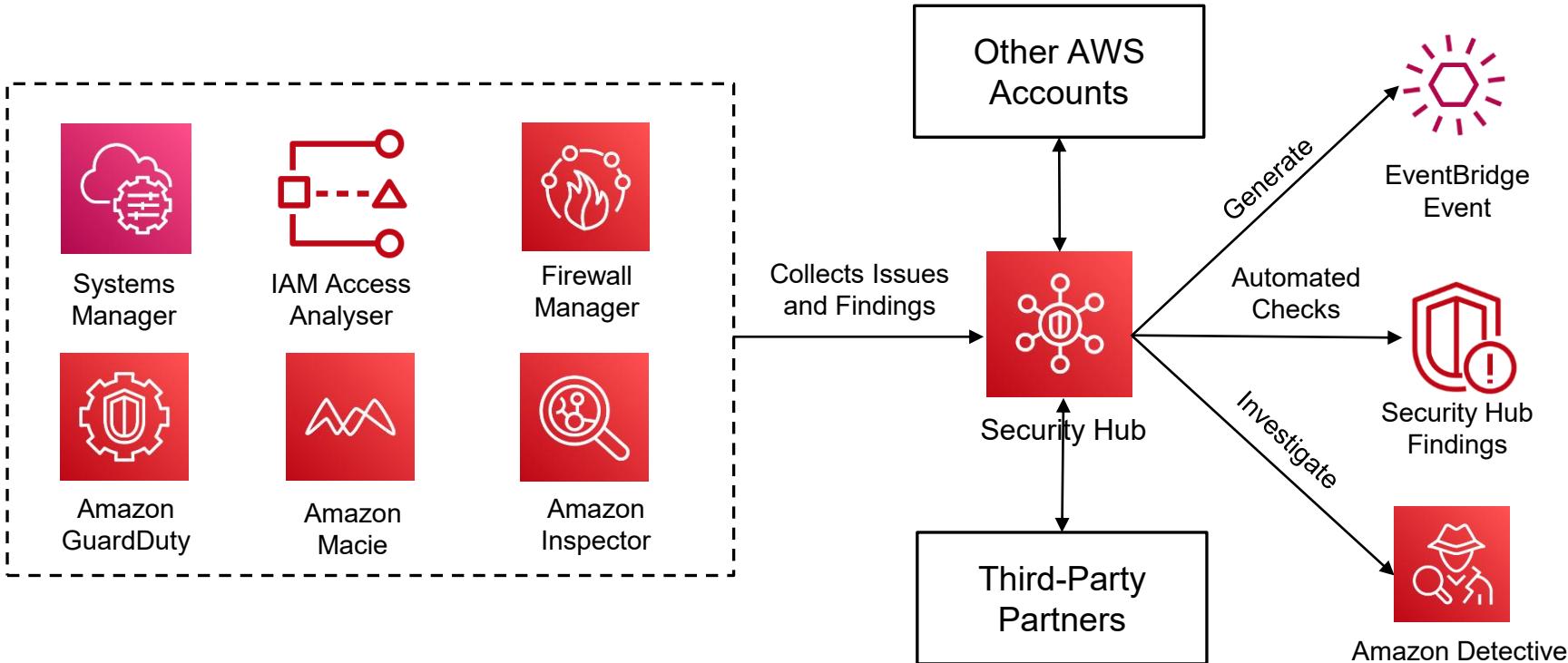
# AWS Security Hub



- Centrally collects security data from across AWS accounts and services.
- Helps analyze security trends to identify and prioritize the security issues across your AWS landscape.
- Automatically aggregates alerts from various AWS services and AWS partner tools such as:
  - Amazon GuardDuty
  - Amazon Inspector
  - Amazon Macie
  - IAM Access Analyzer
  - AWS Systems Manager
  - AWS Firewall Manager
  - AWS Partner Network Solutions

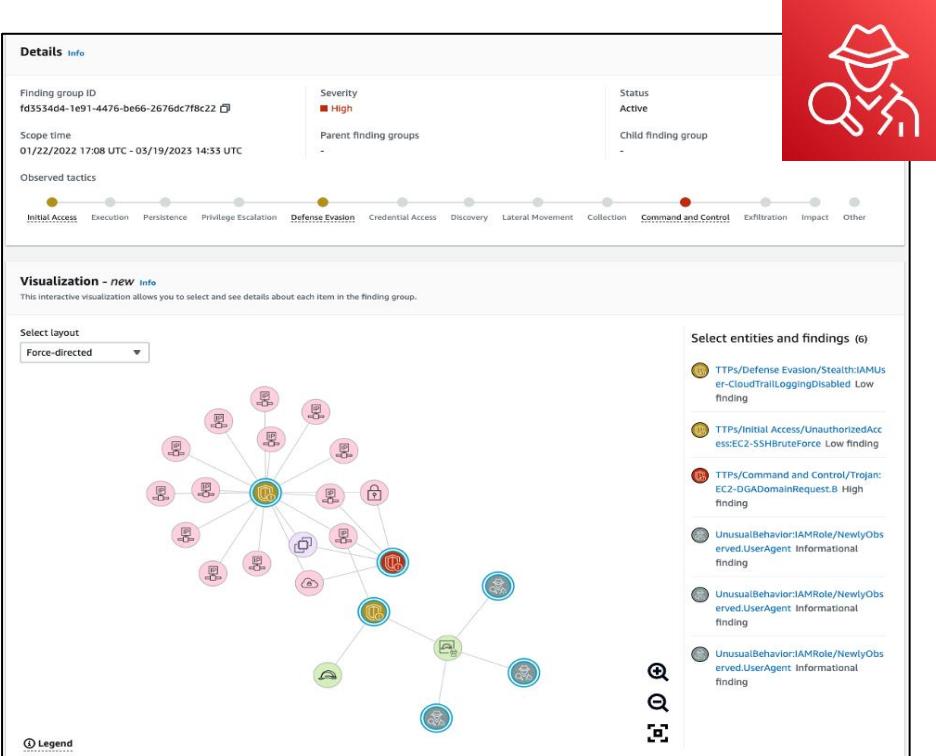
Must enable AWS Config service so that AWS resource state can be captured and analysed

# AWS Security Hub



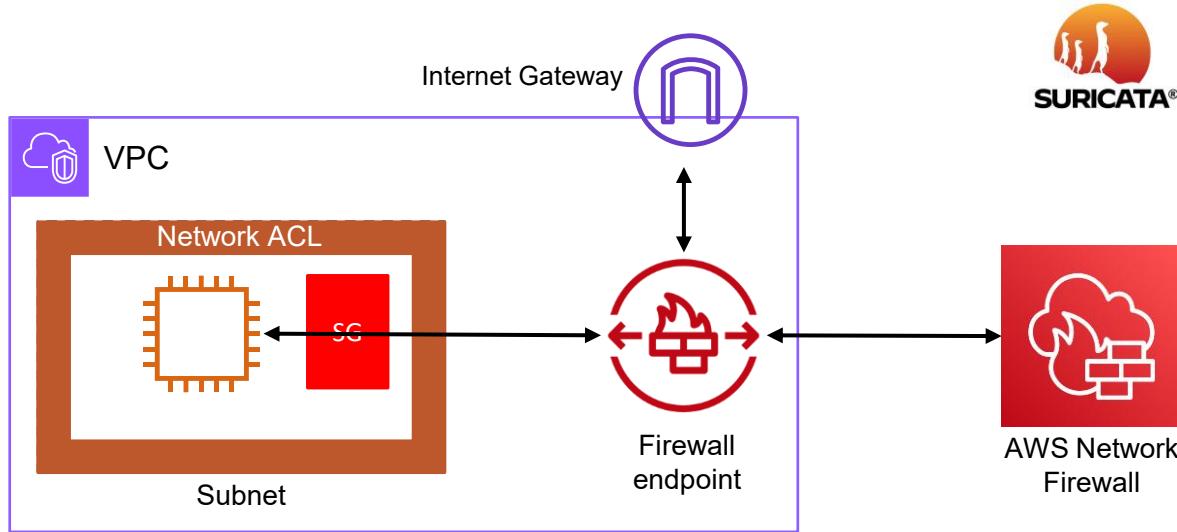
# Amazon Detective

- Uses Machine learning and Graph technology to analyze, investigate, and identify the **root cause** of security issues
- Uses network traffic data, AWS account activity events and security findings from Amazon Macie, Amazon GuardDuty, Security Hub etc.



# AWS Network Firewall

A stateful network firewall and **intrusion detection and prevention** service for VPC



# AWS Firewall Manager

- Centrally configure and manage Firewall rules across AWS accounts
- Manages the rules for AWS WAF, AWS Shield Advanced, Amazon VPC security groups, AWS Network Firewall, and Amazon Route 53 Resolver DNS Firewall
- Integrates with AWS Organization:
  - Provides centralized monitoring of DDoS attacks across AWS organization
  - New accounts added under the AWS Organizations are automatically protected



# AWS Artifact

A web portal to access and download security and compliance reports and agreements from AWS and 3<sup>rd</sup> party ISVs who sell their products on AWS Marketplace.

- ✓ Artifact Reports: ISO, SOC, PCI reports for AWS compliance
- ✓ Artifact Agreements: Review, accept, and manage agreements for your AWS account or organization e.g. Business Associate Addendum (BAA) agreement for HIPAA compliance if you deal with Personal Health Information (PHI)

**ISO 27018**



**ISO 50001**



**HITRUST CSF Certified**



**NIST**



**SOC 1**



**SOC 2**



**FedRAMP**



**FIPS 140-2  
CRYPTOGRAPHY**



**FISMA**



**PINAKES**

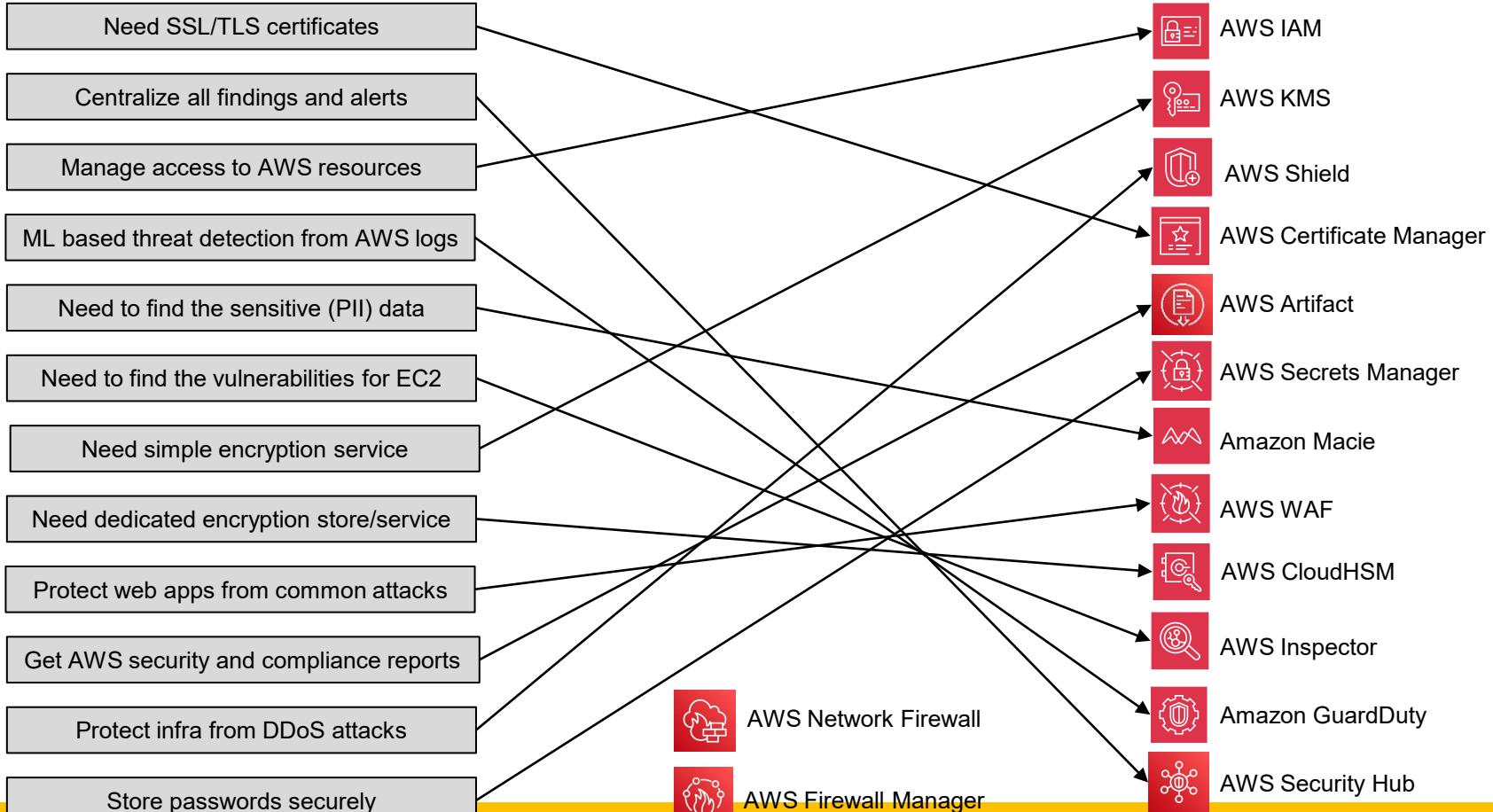


**TISAX**

# Penetration Testing

- AWS allows customers to carry out security assessment and penetration tests on their AWS infrastructure
- There is a list of “Permitted Services” for which customers do not need to take any prior approval.
  - ✓ Amazon EC2, NAT, ELB, RDS, CloudFront, Aurora, API Gateway, Lambda, Lightsail, Elastic Beanstalk
- For some specific types of testing e.g. Command & Control (C2), customers need to get AWS approval.
- **Activities like DDoS, port flooding, DNS hijacking are not permitted.**
- If AWS detects an abuse activities related to your security testing, AWS will send you the report and you need to acknowledge and provide the additional details as requested by AWS.
- For any other simulated events, you should contact [aws-security-simulatedevent@amazon.com](mailto:aws-security-simulatedevent@amazon.com)

# AWS security services - When to use what?



# AWS Security services - summary

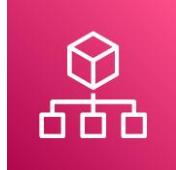
- **AWS IAM** - To manage users, roles and their permissions to access AWS account and services
- **Amazon KMS** - Create and control keys used to encrypt or sign your data
- **AWS CloudHSM** - Manage single-tenant hardware security modules (HSMs) on AWS
- **AWS Certificate Manager** - Provision and manage SSL/TLS certificates with AWS services and connected resources
- **AWS Secrets Manager** - Centrally manage the lifecycle of secrets
- **AWS Macie** - Discover and protect your sensitive data stored in S3.
- **AWS WAF** - Protects your web applications from common exploits.
- **AWS Shield** - Maximize application availability and responsiveness with managed DDoS protection.
- **Amazon Inspector** - Find software vulnerabilities in EC2, ECR Images, and Lambda functions
- **Amazon GuardDuty** - Protect AWS accounts with intelligent threat detection by analyzing AWS services logs
- **AWS Security Hub** - Centrally gather security findings and security alerts from multiple AWS accounts.
- **Amazon Detective** - Find the root cause of security issues or suspicious activities

# AWS Security services - summary

- **AWS Network Firewall** - A stateful intrusion detection and prevention service
- **AWS Firewall Manager** - Centrally configure and manager Firewall rules across AWS accounts
- **AWS Artifact** - A portal to access and download security and compliance reports and agreements from AWS

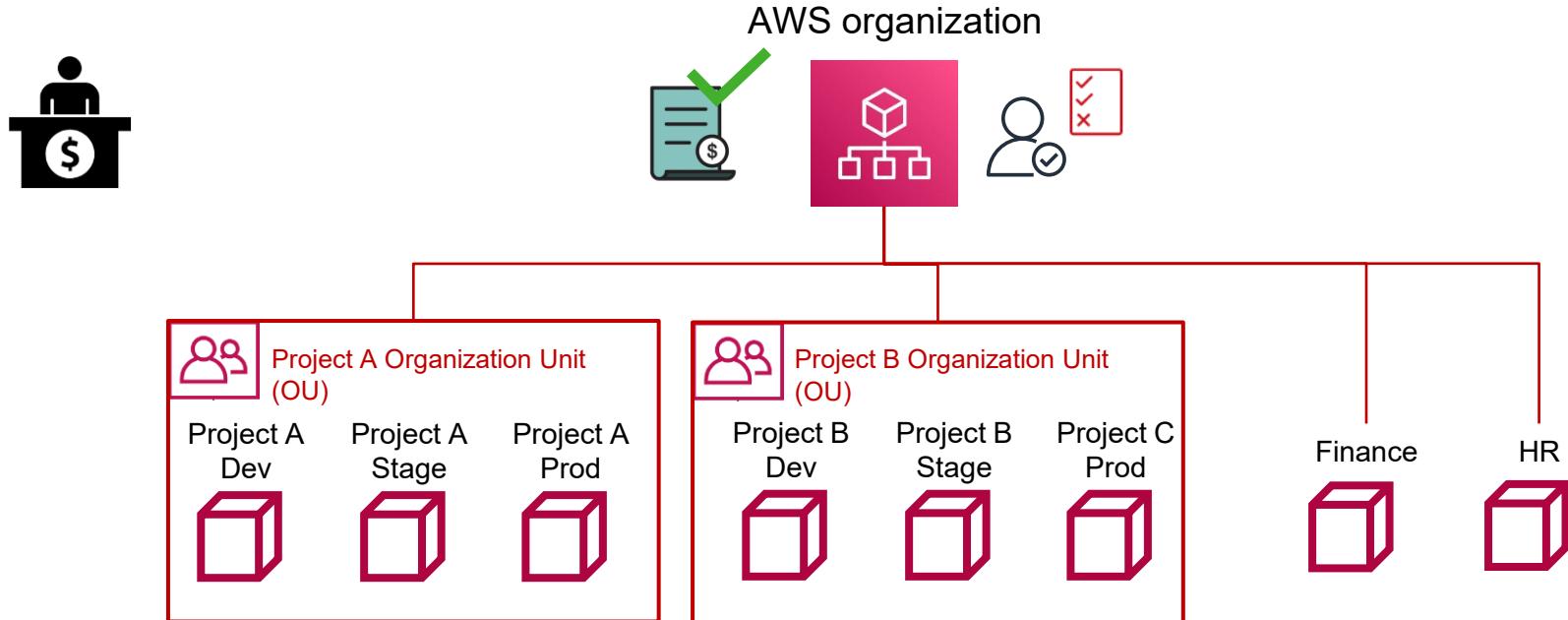
# AWS Account Management

AWS Organizations, Control Tower and Resource Access Manager (RAM)



# AWS Organizations

# AWS Organizations

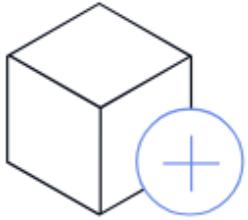




# AWS Organizations

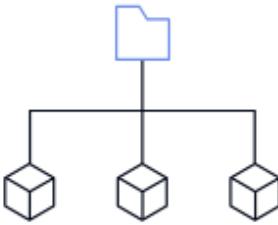
- Consolidate and centrally manage multiple AWS Accounts
- AWS organization consists of a management account and zero or more member accounts
- Member accounts can be organized into a group called **Organization Units**.
- AWS account creation and deletion can be automated
- Restrict account privileges using **Service Control Policies (SCP)**
- AWS Organization features:
  - Consolidated Billing - Single bill for all member accounts
  - Centralized policies - To control access to AWS accounts/APIs/AI services/tags/backups etc.
  - Organization Units (OUs) - Hierarchical grouping of accounts into Organization Units (OUs) for budgetary, security and compliance needs. OUs support five level of nesting.

# AWS Organizations



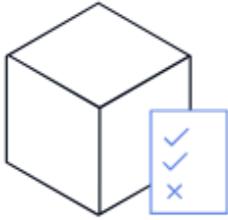
## Add accounts

Create new accounts or invite existing accounts to your organization



## Group accounts

Group accounts into organizational units (OUs) by use-case or workstream



## Apply policies

Apply policies to accounts or OUs, such as service control policies (SCPs) which create permission boundaries



## Enable AWS services

Enable AWS services integrated with Organizations

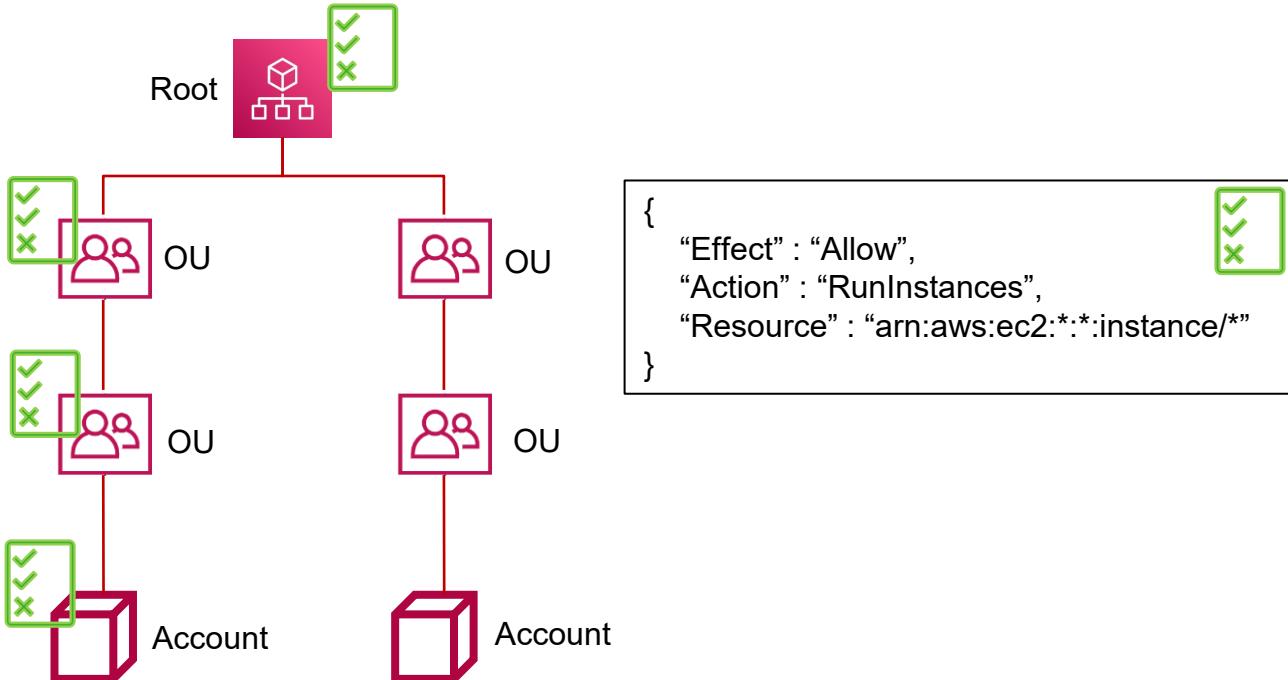
[https://docs.aws.amazon.com/organizations/latest/userguide/orgs\\_introduction.html](https://docs.aws.amazon.com/organizations/latest/userguide/orgs_introduction.html)

# Service Control Policies (SCP)

- AWS Organization control for managing IAM permissions across Accounts and OUs.
- SCP governs the maximum available permissions for the IAM users and roles in the AWS organization
- SCPs do not grant permissions to the IAM users and IAM roles.
- Applied at the OU level or Account level.
- SCPs don't affect users or roles in the management account. They affect only the member accounts in your organization.
- Use cases:
  - Restrict access to certain services which are banned in your organization (for example: AI services)
  - Restrict access to certain AWS regions where you do not have any workloads
  - Restrict access to certain instance types and size in the lower environments

# SCP evaluation

- For a permission to be allowed for a specific account, there must be an explicit Allow statement at every level from the root through each OU in the direct path to the account (including the target account itself).



# Sample SCP policies..

Restrict all users to launch only t2.micro instance type

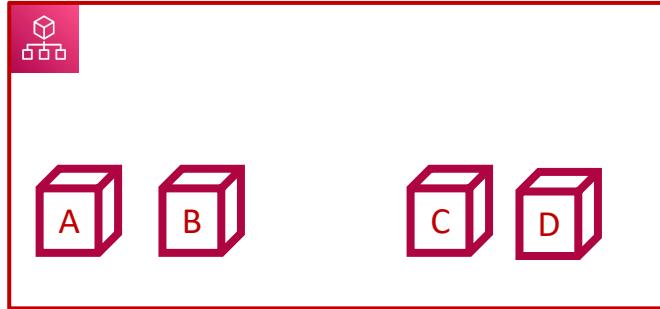
```
{  
  "Version": "2012-10-17",  
  "Statement": [  
    {  
      "Sid": "RequireMicroInstanceType",  
      "Effect": "Deny",  
      "Action": "ec2:RunInstances",  
      "Resource": [  
        "arn:aws:ec2:*:*:instance/*"  
      ],  
      "Condition": {  
        "StringNotEquals": {  
          "ec2:InstanceType": "t2.micro"  
        }  
      }  
    }  
  ]  
}
```

Restricts all users from uploading unencrypted objects to S3 buckets.

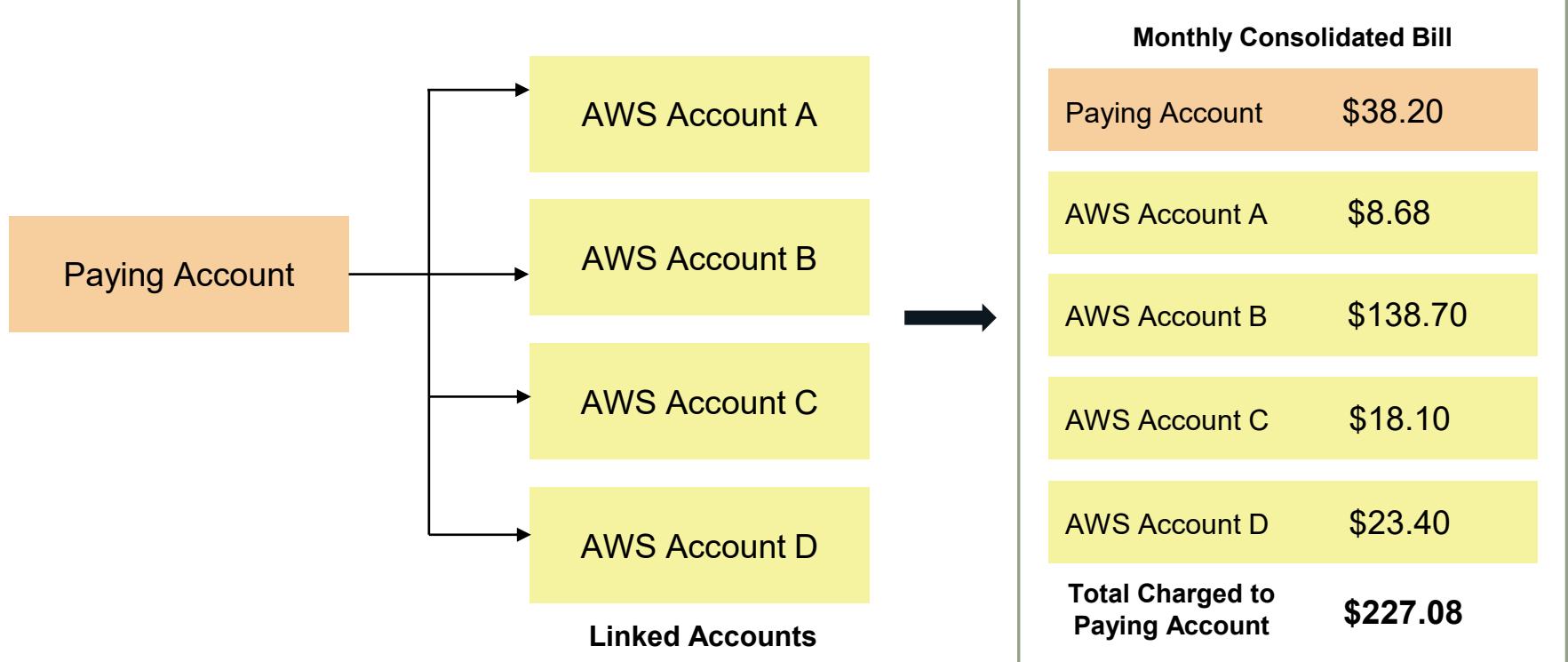
```
{  
  "Effect": "Deny",  
  "Action": "s3:PutObject",  
  "Resource": "*",  
  "Condition": {  
    "Null": {  
      "s3:x-amz-server-side-encryption": "true"  
    }  
  }  
}
```

# AWS Organizations – Consolidated Billing

- Management account to pay bill for entire AWS organization (including all member accounts)
- Benefits:
  - **One Bill** - Get one bill for multiple accounts.

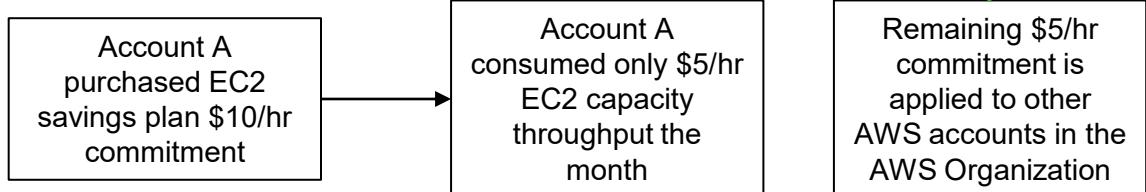
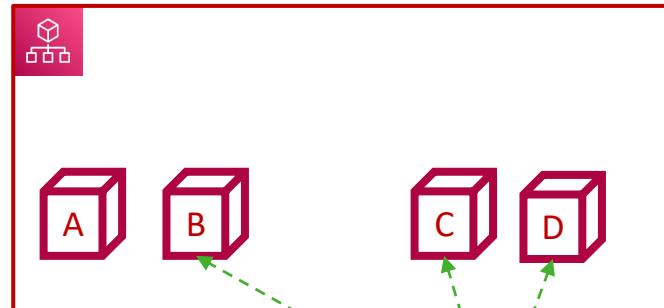


# AWS Organizations – Consolidated Billing



# AWS Organizations – Consolidated Billing

- Management account to pay bill for entire AWS organization (including all member accounts)
- Benefits:
  - One Bill - Get one bill for multiple accounts.
  - Combined Usage - Combine the usage across all accounts in the organization to share the volume pricing discounts, Reserved Instance discounts, and Savings Plans. This can result in a lower charge for your company.
  - No extra Fee - Consolidated billing is offered at no additional cost.



# Exercise : Create new member account and apply SCP

-  Management account steps
-  Member account steps

- 1 Login into AWS Management account and go to AWS Organizations
- 2 Under Root OU -> Create a new OU (say Development)
- 3 Under Development OU -> Create a new AWS account (provide email id)
- 4 After account created successfully, login to new account in new browser private window with email id. You would have to choose forgot password option for first time login.
- 5 In the management account, go to SCP and create a new SCP policy which restricts launching only EC2 t2.micro instances. Refer to the policy document in the next slide.
- 6 Attach this policy to the Development OU that you created earlier. It should be automatically applied to the new AWS account that you created.
- 7 In the new AWS account, try to launch t2.large EC2 instance. Access should be denied. But now try launching t2.micro EC2 instance, it should be launched successfully.
- 8 If you do not need a new account that you created, then Close the account from management account screen.

## Sample SCP policy:

```
{  
    "Version": "2012-10-17",  
    "Statement": {  
        "Effect": "Deny",  
        "Action": "ec2:RunInstances",  
        "Resource": "arn:aws:ec2:*:*:instance/*",  
        "Condition": {  
            "StringNotEquals": {  
                "ec2:InstanceType": "t2.micro"  
            }  
        }  
    }  
}
```

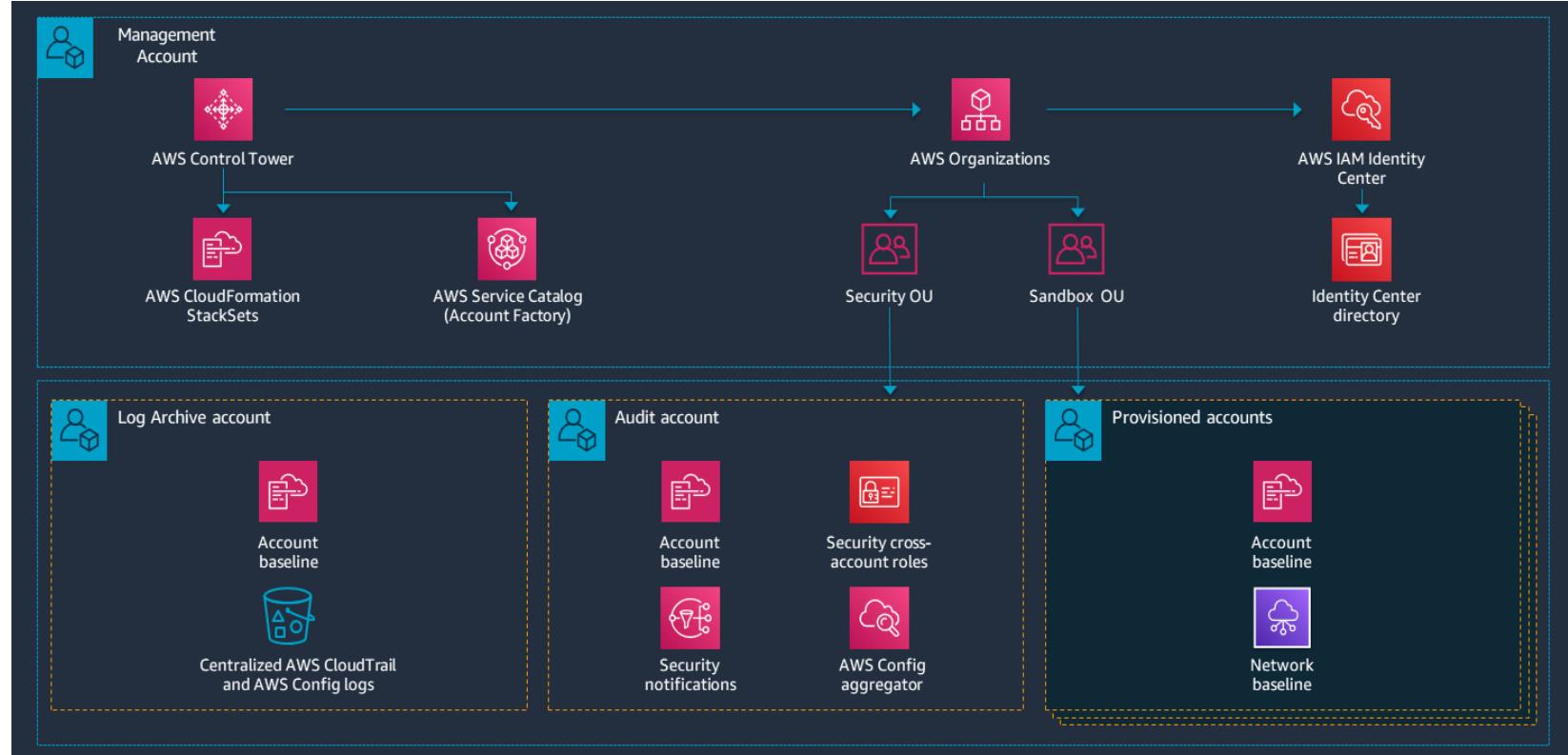


# AWS Control Tower

# AWS Control Tower

- Deploy a multi-account AWS environment with pre-configured best practices.
- Works with AWS Organizations, AWS Identity Center, and AWS Service Catalog
- **Landing Zone:** A well-architected, multi account-environment that's based on security and compliance best practices.
  - Controls (or Guardrails): Pre-configured, customizable rules to meet security and compliance.
  - Mandatory guardrails: Disallow root access, disallow changes to encryption, enforce MFA, disallow public access to S3 buckets etc.
  - Elective guardrails: Disallow SSH/RDP from internet, Enforce tagging, Disallow RDS snapshot public sharing etc.
- **Account Factory:** A template for creating new accounts with pre-configured resources and controls e.g. VPC with 2 subnets
- **Dashboard:** Centralized view of entire Landing zone, compliance policies, non-compliant resources organized by accounts and OUs

# AWS Landing zone provisioned by AWS Control Tower

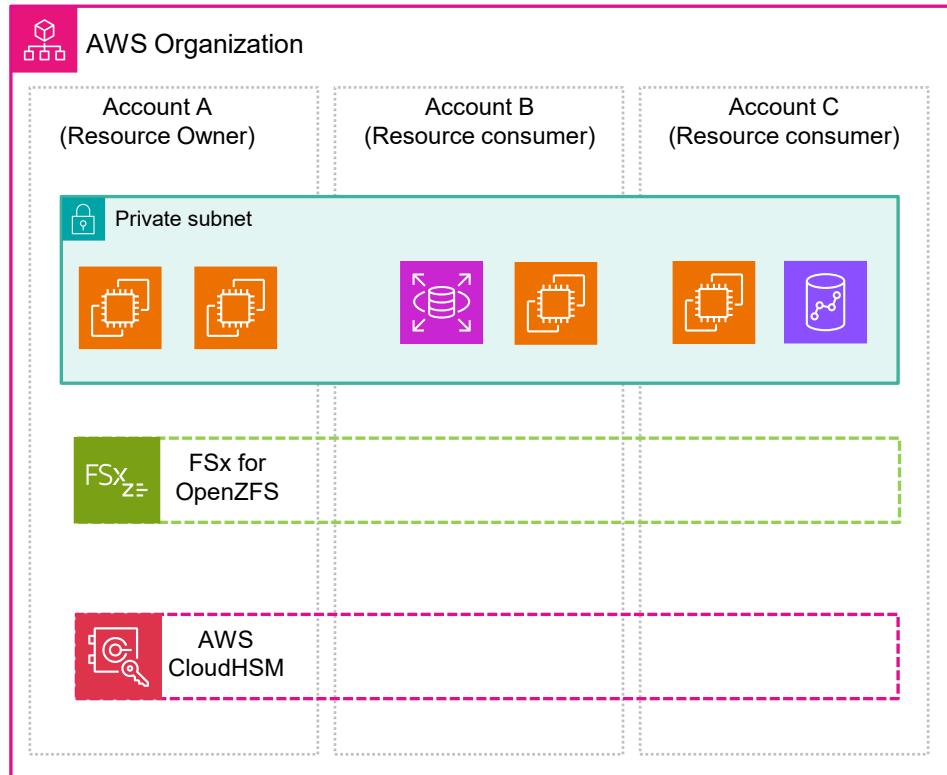




# AWS Resource Access Manager

# AWS Resource Access Manager (RAM)

- Share AWS resources with other AWS Accounts
- In AWS organization, resource can be shared with a single account or OU or all accounts
- Resource can also be shared with AWS account outside of AWS organization
- Supports AWS services and resources like VPC subnets, Transit gateway, Route53, EC2 dedicated host, CloudHSM and more..



# AWS Account Management - summary

- **AWS Organization:**
  - Operate and manage multiple AWS accounts
  - Use Organization Units (OU) to group accounts
  - Use Service Control Policies (SCP) to restrict access to AWS services, resource types, or regions etc.
  - SCP can be applied at Root level or OU level or member account level
- **AWS Control Tower**
  - Setup well-architected Landing Zone by following AWS best practices
- **AWS Resource Access Manager (RAM)**
  - Share AWS resources with other AWS accounts using AWS Resource Access Manager (RAM)

# AWS Billing and Cost management

# How to estimate, view and analyze AWS cost?

- How to estimate cost of AWS services for your architecture?



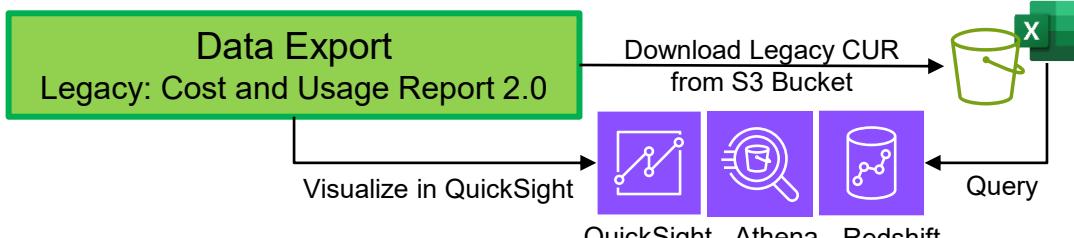
- View overall AWS monthly Bills



- Analyze the cost per service, region etc. and also see forecasted usage



- Granular level report of cost breakdowns. Use your BI tools to analyze cost.



- How to track the free tier usage?



# AWS Bills

Screenshot of the AWS Billing and Cost Management service showing the charges for Amazon Web Services India Private Limited.

The main panel displays the following information:

- Total active services: 34
- Total pre-tax service charges in USD: **USD 11.76**
- Filter by service name or region name:  (Placeholder: Filter by service name or region name)
- Number of pages: 1 of 2

Description	Usage Quantity	Amount in USD
QuickSight		USD 5.95
Virtual Private Cloud		USD 1.97
Route 53		USD 1.50
Key Management Service		USD 0.81
Textract		USD 0.49
Elastic Compute Cloud		USD 0.37
Glue		USD 0.24
Simple Storage Service		USD 0.20
Bedrock		USD 0.14

# AWS Cost explorer

Billing and Cost Management > Cost Explorer > New cost and usage report

## New cost and usage report

Recent reports ▾

Save to report library

### Cost and usage graph Info

Total cost

\$28.26

Average monthly cost

\$4.71

Service count

29

Costs (\$)

1.8

1.2

0.6

0

-0.6

Apr 2024

May 2024

Jun 2024

Jul 2024

Aug 2024

Sep 2024



■ Route 53 ■ Key Management Service ■ EC2-Other ■ Tax ■ S3 ■ VPC ■ Relational Database Service ■ EC2-Instances ■ Elastic Load Balancing ■ Others

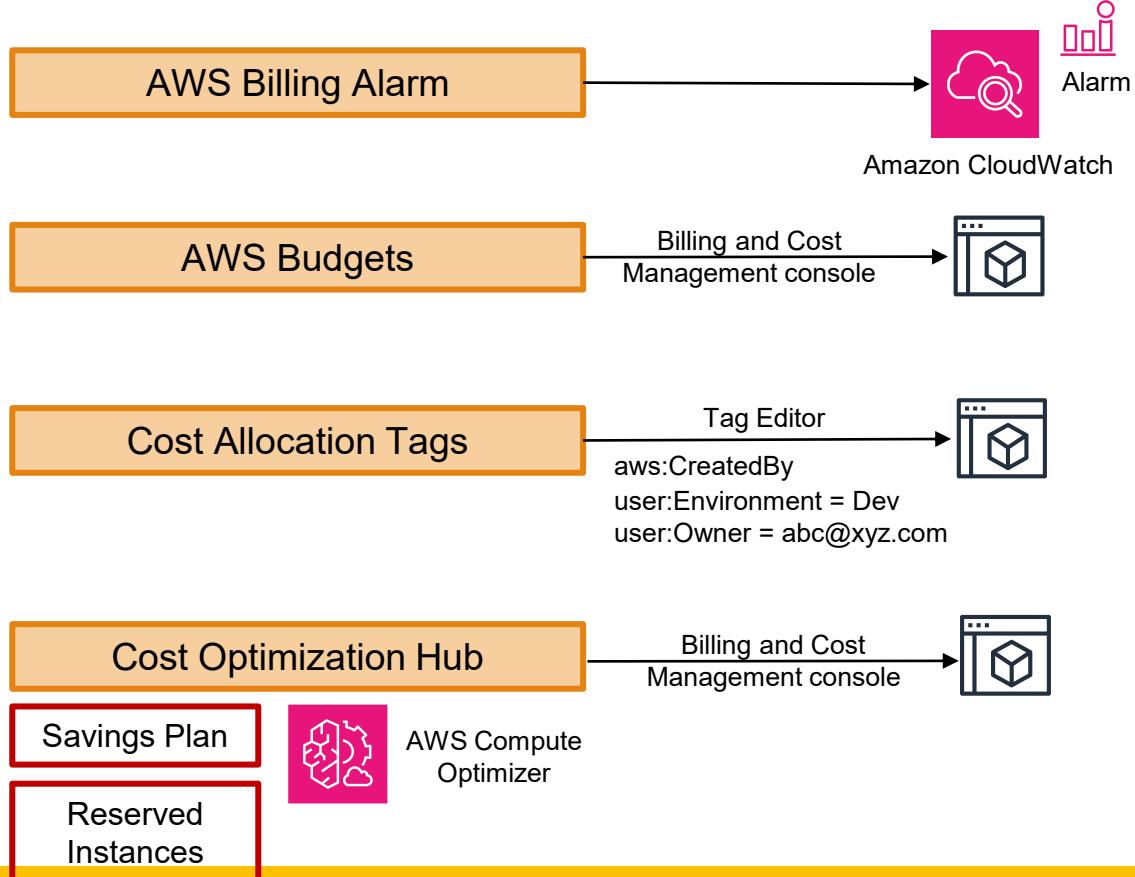
# Free tier usage dashboard

AWS Billing and Cost Management console -> Cost Analysis -> Free tier

Free Tier offers in use (39)						
Service	AWS Free Tier usage limit	Current usage	Forecasted usage	MTD actual usage %	MTD forecasted %	Action
Amazon Redshift	750.0 Hrs for free per month during a short-term trial as part of AWS Free Usage Tier (Global-Node:dc2.large)	664 Hrs	709 Hrs	<div style="width: 88.47%; background-color: red;"></div> 88.47%	<div style="width: 88.47%; background-color: red;"></div>	
Amazon DynamoDB	18600.0 ReadCapacityUnit-Hrs are always free per month as part of AWS Free Usage Tier (APS3-ReadCapacityUnit-Hrs)	13,300 ReadCapacityUnit-Hrs	14,217 ReadCapacityUnit-Hrs	<div style="width: 71.51%; background-color: blue;"></div> 71.51%	<div style="width: 71.51%; background-color: blue;"></div>	
Amazon DynamoDB	18600.0 WriteCapacityUnit-Hrs are always free per month as part of AWS Free Usage Tier (APS3-WriteCapacityUnit-Hrs)	10,640 WriteCapacityUnit-Hrs	11,374 WriteCapacityUnit-Hrs	<div style="width: 57.20%; background-color: blue;"></div> 57.20%	<div style="width: 57.20%; background-color: blue;"></div>	
Amazon DynamoDB	18600.0 ReadCapacityUnit-Hrs are always free per month as part of AWS Free Usage Tier (USW2-ReadCapacityUnit-Hrs)	9,975 ReadCapacityUnit-Hrs	10,663 ReadCapacityUnit-Hrs	<div style="width: 53.63%; background-color: blue;"></div> 53.63%	<div style="width: 53.63%; background-color: blue;"></div>	
Amazon DynamoDB	18600.0 WriteCapacityUnit-Hrs are always free per month as part of AWS Free Usage Tier (USW2-WriteCapacityUnit-Hrs)	9,975 WriteCapacityUnit-Hrs	10,663 WriteCapacityUnit-Hrs	<div style="width: 53.63%; background-color: blue;"></div> 53.63%	<div style="width: 53.63%; background-color: blue;"></div>	

# How to track and control cost?

- Notify when total usage exceeds certain threshold e.g. > \$100 for current month
- Set cost budgets for overall usage or AWS Services e.g. \$1000 for EC2, \$500 for S3 and get notified when exceeded
- Want to track cost by environments, projects, services, users or based on different dimensions?
- Want to get recommendations for purchasing Savings plan, RIs, right sizing and configuration of AWS resource (e.g. EC2, EBS etc.) ?



# Exercise: AWS Billing Alarm

DO THIS IN N.VIRGINIA (us-east-1) REGION

Set a Billing alert so that you get notified when your AWS usage bill exceeds some threshold (say > \$5/month)

- 1) Billing and Cost Management -> Billing Preferences -> Alert preferences -> Edit -> Receive CloudWatch Billing Alerts -> Save
- 2) In Amazon SNS -> Create a new topic and subscribe with your email id
- 3) In Amazon CloudWatch -> Go to -> Alarms -> Billing -> Create alarm

Metric Name: EstimatedCharges

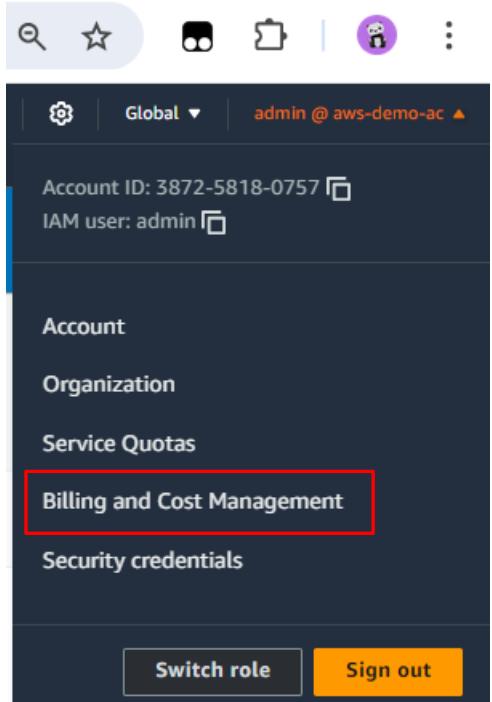
Currency: USD

Statistic: Maximum

Period: 6 hours

Threshold Type: Static

Whenever EstimatedCharges is: **Greater**, than -> **5**



# Exercise: AWS Billing Alarm

4) Alarm State trigger: In alarm -> Send a notification to the following SNS topic -> Select an existing SNS topic -> Select the topic you created earlier -> next -> Alarm name: BillingAlarm5USD -> next -> Create alarm

# AWS Budgets

**AWS Budgets**

Filter by budget name

Download CSV Create budget

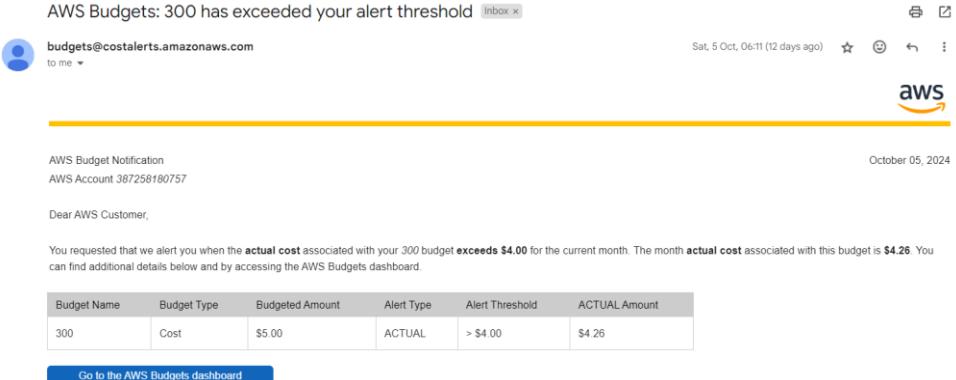
All budgets (8)	Cost budgets (6)	Usage budgets (2)	Reservation budgets (0)			
Budget name	Budget type	Current	Budgeted	Forecasted	Current vs. budgeted	Forecasted vs. budgeted
Project Nemo Cost Budget	Cost	\$42.48	\$40.00	\$42.48	<div style="width: 106.2%; background-color: #f08080;"></div> 106.2%	<div style="width: 106.2%; background-color: #f08080;"></div> 106.2%
Eastern US Regional Budget	Cost	\$82.95	\$100.00	\$93.32	<div style="width: 82.95%; background-color: #0072bc;"></div> 82.95%	<div style="width: 93.32%; background-color: #0072bc;"></div> 93.32%
Total Monthly Cost Budget	Cost	\$138.80	\$200.00	\$163.76	<div style="width: 69.4%; background-color: #0072bc;"></div> 69.4%	<div style="width: 81.88%; background-color: #0072bc;"></div> 81.88%
Total EC2 Cost Budget	Cost	\$135.34	\$200.00	\$159.05	<div style="width: 67.67%; background-color: #0072bc;"></div> 67.67%	<div style="width: 79.53%; background-color: #0072bc;"></div> 79.53%
Loft	Cost	\$40.29	\$65.00	\$50.53	<div style="width: 61.99%; background-color: #0072bc;"></div> 61.99%	<div style="width: 77.74%; background-color: #0072bc;"></div> 77.74%
Monthly DataTransfer Usage Budget	Usage	2.19 GB	4 GB	2.75 GB	<div style="width: 54.65%; background-color: #0072bc;"></div> 54.65%	<div style="width: 68.85%; background-color: #0072bc;"></div> 68.85%
S3 Usage Budget	Usage	2,608 Requests	5,500 Requests	3,309.22 Requests	<div style="width: 47.42%; background-color: #0072bc;"></div> 47.42%	<div style="width: 60.17%; background-color: #0072bc;"></div> 60.17%
Quarterly Budget	Cost	\$131.41	\$550.00	\$473.95	<div style="width: 23.89%; background-color: #0072bc;"></div> 23.89%	<div style="width: 86.17%; background-color: #0072bc;"></div> 86.17%

<https://aws.amazon.com/blogs/aws-cloud-financial-management/beginners-guide-to-aws-cost-management/>

# Exercise: AWS Budget

Go to Billing and Cost Management -> Budgets

- a. Use a template -> Monthly cost budget
- b. Provide Budget name and budget value in USD (e.g. 5)
- c. Provide the email id to which you should receive an email when usage exceeds the budgeted value
- d. Create budget



The screenshot shows an email from AWS Budgets. The subject line is "AWS Budgets: 300 has exceeded your alert threshold". The email is from "budgets@costalerts.amazonaws.com" and was sent on "Sat, 5 Oct, 06:11 (12 days ago)". The body of the email starts with "AWS Budget Notification" and "AWS Account 387258180757". It addresses "Dear AWS Customer," and informs them that their budget for "300" has exceeded the alert threshold of "\$4.00" for the current month, with an actual cost of "\$4.26". A table provides details about the budget:

Budget Name	Budget Type	Budgeted Amount	Alert Type	Alert Threshold	ACTUAL Amount
300	Cost	\$5.00	ACTUAL	> \$4.00	\$4.26

[Go to the AWS Budgets dashboard](#)

Email sent by AWS



# AWS Cost Optimization Hub – Savings plan

Billing and Cost Management > Savings Plans > Recommendations

Settings ? ▾

Home Cost Explorer Saved reports Budgets Recommendations Savings Plans Overview Inventory Recommendations Purchase Savings Plans Utilization Report Coverage Report Reservations Overview Recommendations Utilization Report Coverage Report My Billing Dashboard

Save with Savings Plans

Savings Plans offer discounts in exchange for an hourly commitment to compute usage. Savings Plans discounts

Recommendation options

Savings Plans type: Compute (selected) EC2 Instance

Savings Plans term: 1-year (selected) 3-year

Payment option: All upfront (selected) Partial upfront No upfront

Based on the past: 30 days (selected) 7 days 60 days

Recommendation: Purchase a Compute Savings Plan at a commitment of \$0.30/hour

You could save an estimated \$48 monthly by purchasing the recommended Compute Savings Plan.

Based on your past 30 days of usage, we recommend purchasing a Savings Plan with a commitment of \$0.30/hour for a 1-year term. With this commitment, we project that you could save an average of \$0.07/hour - representing a 14% savings compared to On-Demand. To account for variable usage patterns, this recommendation maximizes your savings by leaving an average \$0.09/hour of On-Demand spend.

Before recommended purchase	After recommended purchase (based on your past 30 days of usage)	
Monthly On-Demand spend ⓘ	Estimated monthly spend ⓘ	Estimated monthly savings ⓘ
\$329 (\$0.45/hour) Your estimated On-Demand spend based on your usage over the past 30 days (including all active Savings Plans)	\$281 (\$0.39/hour) Your recommended \$0.30/hour Savings Plans commitment + an average \$0.09/hour of On-Demand spend	\$48 (\$0.07/hour) 14% monthly savings over On-Demand \$329 - \$281 = \$48

This recommendation examines your usage over the past 30 days (including your existing Savings Plans and EC2 Reserved Instances) and calculates what your costs would have been had you purchased the recommended Savings Plans. See applicable rates for Savings Plans here. To generate this recommendation, AWS simulates your bill for different commitment amounts and recommends the commitment amount that provides the greatest estimated savings. Learn more

Recommended Compute Savings Plans

Download CSV Add selected Savings Plan(s) to cart

* Term	Payment option	Recommended commitment	Estimated hourly savings ⚡
1-year	All upfront	\$0.30/hour	\$0.07 (14%)

<https://aws.amazon.com/blogs/aws-cloud-financial-management/beginners-guide-to-aws-cost-management/>



# AWS Cost Optimization Hub - Reservation

Billing and Cost Management > Reservations > Recommendations

Settings ?

**\$3,020** Estimated Annual Savings\*

**28%** Savings vs. On-Demand

**4** Purchase Recommendations

Based on your past 60 days of EC2 usage, we have identified 4 one-year, partial-upfront, standard RI purchase recommendations to save an estimated \$3,020 annually, representing a savings of 28% versus on-demand costs. You can take action on these recommendations in the [EC2 Reservation Purchase Console](#).

Generate recommendations based on:

All accounts Individual accounts

Sort by:

Monthly Estimated Savings ▾ Download CSV

**Purchase Recommendations (4)**

	Details
<b>Buy 1 g3s.xlarge reserved instance</b> US West (Oregon)   Windows (Amazon VPC)   Shared Based on your past 60 days of on-demand usage, we recommend purchasing 1 g3s.xlarge reserved instance. <a href="#">View Associated EC2 Usage</a>	<b>\$163.92 monthly savings</b> Upfront Cost: \$3,105.00 Recurring Monthly Cost: \$259.15 Expected RI Utilization: 100%
<b>Buy 3 m5.large reserved instances</b> Size flexible** US East (N. Virginia)   Linux/UNIX   Shared Based on your past 60 days of on-demand usage, we recommend purchasing 3 m5.large reserved instances to cover 12 normalized units per hour of m5 family usage to maximize savings. <a href="#">View Associated EC2 Usage</a>	<b>\$82.73 monthly savings</b> Upfront Cost: \$768.00 Recurring Monthly Cost: \$63.51 Expected RI Utilization: 100%
<b>Buy 2 t2.nano reserved instances</b> Size flexible** US East (N. Virginia)   Linux/UNIX   Shared Based on your past 60 days of on-demand usage, we recommend purchasing 2 t2.nano reserved instances to cover 0.5 normalized units per hour of t2 family usage to maximize savings. <a href="#">View Associated EC2 Usage</a>	<b>\$3.49 monthly savings</b> Upfront Cost: \$30.00 Recurring Monthly Cost: \$2.48 Expected RI Utilization: 100%

Select a service

Elastic Compute Cloud (EC2)

RI Recommendation Parameters ⓘ

RI term

1 year  
 3 years

Offering Class

Standard  
 Convertible

Payment option

All upfront  
 Partial upfront  
 No upfront

Based on the past

7 days  
 30 days  
 60 days

Additional Filters

Linked Account [Include all ▾](#)

<https://aws.amazon.com/blogs/aws-cloud-financial-management/beginners-guide-to-aws-cost-management/>

# AWS Billing and Cost management - summary

- AWS Pricing calculator – Create cost estimates and share with others (free tool)
- AWS Bills – Overall monthly bills with service level breakup
- Cost Explorer – Detailed cost breakup and analysis and also cost forecast
- Cost Anomaly detector – Detect unusual spend using historic usage data and machine learning
- Data Export – Raw dataset for created your own analysis and dashboard (Formerly CUR)
- AWS Billing Alert – CloudWatch alarm in us-east-1 (N. Virginia) region to alert when AWS usage exceeds
- AWS Budgets – Create budgets for overall AWS usage or service usage or RI/Savings plan usage
- Cost Allocation Tags – Group the AWS resources by resource tags and create cost categories
- Cost Optimization Hub – Recommendation for purchasing Savings Plan, Reserved Instances
- AWS Cost Optimizer – Recommendations to optimize cost by right sizing resources (e.g. EC2 size)
- AWS Free tier dashboard – 12 months free services, Free trials and Always free services



# AWS Support



# AWS Support

- AWS Support offers a range of plans that provide access to tools and expertise to support the operational health of your AWS solutions.
- All support plans provide 24/7 access to customer service, AWS documentation, technical papers, and support forums
- **AWS Support offers five support plans:** Basic, Developer, Business, Enterprise On-Ramp, Enterprise
- **AWS Trusted Advisor** – Recommendations to save money, close security gaps and improve system reliability and performance
- **AWS Service Quotas** - Request to change quota (or limit) for # AWS resources or capacity in AWS account

# AWS Support plans

## Basic Support

Free

- ✓ Applied by default
- ✓ 24x7 access to customer service, documentation, whitepapers, and AWS re:Post
- ✓ Trusted Advisor recommendations: Only service quotas and core security checks
- ✓ Access to AWS Health Dashboard

# AWS Support plans

## Developer Support

Minimum \$29

- ✓ Basic Support +
- ✓ Technical support during business hours
- ✓ Unlimited support cases
- ✓ Response time: < 24 hrs for general, < 12 hrs for system impaired
- ✓ Contact over web

## Business Support

Minimum \$100

- ✓ Developer Support +
- ✓ 24x7 Technical support
- ✓ Response time: Production system impaired: < 4 hours, Production system down: < 1 hour
- ✓ Trusted Advisor full set of checks
- ✓ 24x7 phone call support

## Enterprise On-ramp support

Minimum \$5500

- ✓ Business Support +
- ✓ AWS Trusted Advisor Priority
- ✓ Response time: Business-critical system down: < 30 minutes
- ✓ Annual consultative review
- ✓ Access to online self-paced labs
- ✓ Pool of Technical Account Manager
- ✓ many other benefits..

## Enterprise support

Minimum \$15000

- ✓ Business Support +
- ✓ AWS Trusted Advisor Priority
- ✓ Response time: Business-critical system down: < 15 mins
- ✓ Consultative review and guidance based on your applications
- ✓ Access to online self-paced labs
- ✓ Dedicated Technical Account Manager
- ✓ many other benefits..

- Support fee in \$ per month
- Generally, its either 3-10% of monthly bill (depending on the total bill) or minimum fees as per support tier, whichever is higher.

# AWS Support plans

## Developer Support

Minimum \$29

- ✓ Recommended if you are experimenting or testing in AWS
- ✓ Contact over web

## Business Support

Minimum \$100

- ✓
- ✓ Minimum recommended tier if you have production workloads in AWS
- ✓ Phone call support

✓ 24x7 phone call support

## Enterprise On-ramp support

Minimum \$5500

- ✓ Recommended if you have production and/or business critical workloads in AWS
- ✓ Pool of Technical Account Manager
- ✓ many other benefits..

## Enterprise support

Minimum \$15000

- ✓ Recommended if you have business and/or mission critical workloads in AWS

- ✓ Access to online self-paced labs
- ✓ Dedicated Technical Account Manager
- ✓ many other benefits..

- Support fee in \$ per month
- Generally, its either 3-10% of monthly bill (depending on the total bill) or minimum fees as per support tier, whichever is higher.



# AWS Trusted Advisor

# Trusted Advisor

- Trusted Advisor inspects your AWS environment, and provides over 450+ recommendations across following 6 categories:

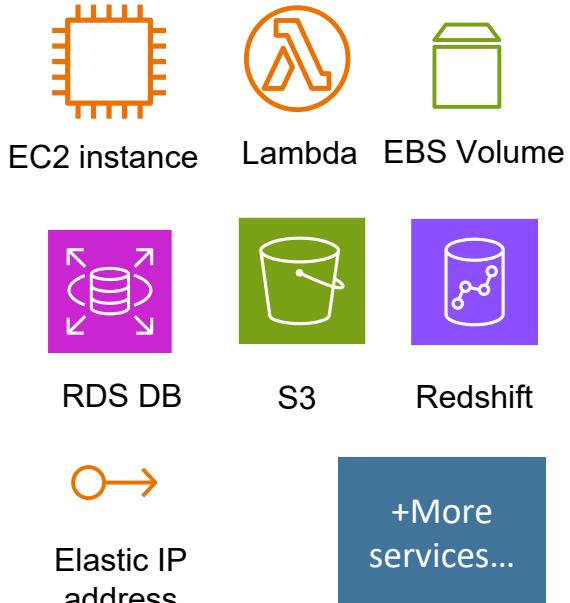
1. **Cost Optimization** - These checks highlight unused, underutilized or idle resources
2. **Performance** - To improve the speed and responsiveness of your applications
3. **Security** - Close security gaps
4. **Fault tolerance** - Highlight redundancy shortfalls and overused resources
5. **Service Limits** - whether your account approaches or exceeds the service limits (quotas).
6. **Operational Excellence** - AWS recommended configurations for logging, monitoring etc.

Check	Date
✓ <a href="#">Stopped EC2 instances should be removed after a specified time period</a>	08/01/2024
✓ <a href="#">EC2 instances should be managed by AWS Systems Manager</a>	07/31/2024
✓ <a href="#">Amazon RDS DB instances not using Multi-AZ deployment</a>	07/31/2024
✓ <a href="#">EC2 instances should use Instance Metadata Service Version 2 (IMDSv2)</a>	07/31/2024
✓ <a href="#">Amazon RDS Performance Insights is turned off</a>	07/31/2024
✓ <a href="#">Amazon RDS DB instances have storage autoscaling turned off</a>	07/31/2024
✓ <a href="#">Amazon RDS DB clusters have one DB instance</a>	07/31/2024
✓ <a href="#">Amazon RDS Aurora storage encryption is turned off</a>	07/31/2024
✓ <a href="#">Amazon RDS storage encryption is turned off</a>	07/31/2024
✓ <a href="#">Amazon RDS Enhanced Monitoring is turned off</a>	07/31/2024

# Trusted Advisor

## Cost Optimization checks examples

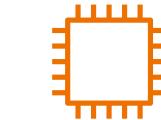
- Over provisioned resources – EC2 instances, Lambda functions
- Underutilized resources – EC2 instances, EBS volumes, Redshift Clusters etc.
- Idle resources – RDS DB instance, Load balancer
- Inactive resources - NAT Gateway, VPC interface endpoints, Network firewall etc.
- Reserve instances lease expiration
- Unassociated Elastic IPs
- Configurations – S3 version enabled bucket without lifecycle policy, AWS account not part of AWS Organization



# Trusted Advisor

## Performance checks examples

- High CPU utilization for EC2 instances
- RDS DB configurations for optimum performance
- EBS volume throughput, IOPS and size optimization
- Enabling autoscaling for EC2, DynamoDB
- Lambda functions without concurrency
- Amazon EFS burst throughput mode



EC2 instance



Lambda



RDS DB



S3



Redshift



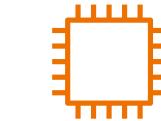
Elastic IP address

+More services...

# Trusted Advisor

## Security Checks examples

- EC2 instances reaching End of Support e.g. Microsoft SQL server, Microsoft Windows server etc.
- Encryption is turned off – EFS, RDS
- Security group – Unrestricted / Public Access
- S3 bucket public access
- CloudTrail logs should be enabled
- IAM – Access keys age, password policy
- SSL certificates – expired or nearing expiration



EC2 instance



Lambda



RDS DB



S3



Redshift



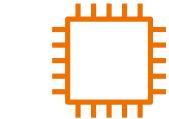
Elastic IP address

+More services...

# Trusted Advisor

## Fault Tolerance examples

- **Multi-AZ deployment** – Application Load Balancer, Autoscaling group, RDS databases, ECS, ElastiCache, MSK cluster etc.
- AWS Backup – Include resources such as EBS volume, EFS filesystem, DynamoDB table, in the backup plan
- Automated Backup/snapshot - For RDS, Redshift cluster
- Snapshot age – EBS volumes
- Monitoring – Detailed monitoring for EC2, RDS etc.
- Amazon Route53 – Health checks, failover records & more..



EC2 instance



Lambda



RDS DB



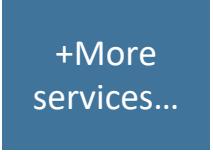
S3



Redshift



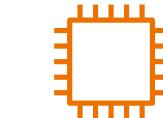
Elastic IP address



# Trusted Advisor

## Service Limits

- Some limits are soft limits and can be adjusted through Service Quota console or by opening a support case
  - ✓ Number of S3 buckets = 100, can be increased
  - ✗ S3 Lifecycle rules = 1000, can not be increased
- Yellow and Red alerts on reaching 80% and 100% limit



EC2 instance



EBS Volume



RDS DB



S3



Elastic IP address



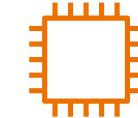
Redshift

+More services...

# Trusted Advisor

## Service limit checks examples

- DynamoDB read usage is 80% or more than provisioned throughput
- Autoscaling is operating over 80% of the capacity
- EBS IOPS is more than 80% of the provisioned capacity
- Number of resources used is more than 80% of the current limit for the AWS account – EC2 instances, VPCs, IAM users, role, EBS volumes etc.



EC2 instance



EBS Volume



RDS DB



S3



Elastic IP address



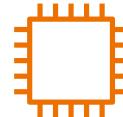
Redshift

+More services...

# Trusted Advisor

## Operational excellence checks examples

- Logs not enabled – API gateway, S3 access logs, VPC flow logs, ELB access logs etc.
- EC2 instance is not managed by Systems Manager
- Deletion protection – RDS DB instance, ELB etc.
- Version upgrade – AWS Fargate platform, RDS DB instance etc.



EC2 instance



EBS Volume



RDS DB



S3



Elastic IP  
address



Redshift

+More services...

# Trusted Advisor checks with AWS Support Plans

## Basic & Developer Support Plan

50+ checks

- S3 Bucket Permissions
- Security Groups – Specific Ports Unrestricted
- MFA on Root Account
- EBS Public Snapshots
- RDS Public Snapshots
- Service Limits

## Business & Enterprise Support Plan

430+ checks

- Checks included in Basic/Development support plan
- Full checks across 6 categories of Trusted Advisor
- Ability to set CloudWatch alarms when reaching limits
- Programmatic Access using AWS Support API

# AWS Service Quotas

- AWS account comes with default quotas, formerly referred to as **limits**, for each AWS service.
- Manage your quotas for many AWS services through AWS Service Quotas.
- Notifies using *CloudWatch Alarm* when you're close to a service quota value threshold.
- AWS Support might approve, deny, or partially approve your requests.

The screenshot shows the AWS Service Quotas dashboard. On the left, a sidebar lists 'Service Quotas' with options for 'Dashboard', 'AWS services', 'Quota request history', and 'Organization'. Under 'Organization', there are links for 'Quota request template' and 'Modify dashboard cards'. The main area is titled 'Service Quotas > Dashboard' and shows a grid of service quotas. Each card includes the service name, a small icon, total quotas, and a link to the service's quota details. The services listed are Amazon Athena (Total quotas: 6), Amazon DynamoDB (Total quotas: 13), Amazon Elastic Block Store (Amazon EBS) (Total quotas: 40), Amazon Elastic Compute Cloud (Amazon EC2) (Total quotas: 174), Amazon Relational Database Service (Amazon RDS) (Total quotas: 29), Amazon Virtual Private Cloud (Amazon VPC) (Total quotas: 27), AWS CloudFormation (Total quotas: 27), and AWS Key Management Service (AWS KMS) (Total quotas: 55).

Service	Total Quotas
Amazon Athena	6
Amazon DynamoDB	13
Amazon Elastic Block Store (Amazon EBS)	40
Amazon Elastic Compute Cloud (Amazon EC2)	174
Amazon Relational Database Service (Amazon RDS)	29
Amazon Virtual Private Cloud (Amazon VPC)	27
AWS CloudFormation	27
AWS Key Management Service (AWS KMS)	55



# AWS Support - summary

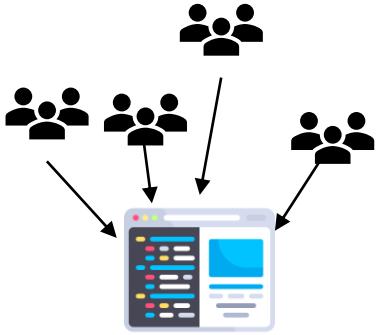
- All support plans provide 24/7 access to customer service, AWS documentation, technical papers, and support forums
- **AWS Support offers five support plans:** Basic (Free), Developer, Business, Enterprise On-Ramp and Enterprise
  - If you are running Production workload in AWS then at minimum it's recommended to subscribe to AWS Business support
  - With AWS Basic and Developer support plan you get core TA checks but with Business and enterprise support plans you get all the checks
- **AWS Trusted Advisor** – Recommendations to save money, close security gaps and improve system reliability and performance
- **AWS Service Quotas** - Request to change quota (or limit) for # AWS resources or capacity in AWS account
  - There are soft of adjustable limits and there are hard limits

# AWS Well Architected and Cloud Adoption Framework

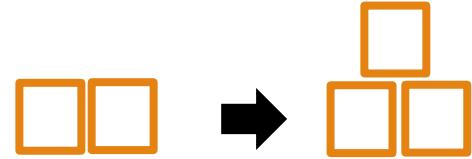
# AWS General Design Principles



Stop guessing capacity



Test system at Production scale



Automate with architectural experimentation in mind



Consider evolutionary architectures



Drive architectures using data



Improve through game days

# AWS Well-Architected

1. AWS Well-Architected helps cloud architects build secure, high-performing, resilient, and efficient infrastructure for a variety of applications and workloads.
2. Built around 6 pillars - Operational excellence, Security, Reliability, Performance efficiency, Cost optimization, and Sustainability
3. AWS Well-Architected Framework includes domain specific lenses (e.g. IoT, ML, Serverless etc.)
4. AWS Well-Architected Tool to document and review your architecture
5. AWS Customers can also work with AWS Well-Architected Partners to evaluate their workloads

<https://docs.aws.amazon.com/wellarchitected/latest/framework/welcome.html>

# 6 pillars of AWS Well-architected framework

1. **Operational excellence** - The ability to support development and run workloads effectively, gain insight into their operations, and to continuously improve.
2. **Security** - The ability to protect information, systems, and assets while delivering business value
3. **Reliability** - The ability of a system to recover from infrastructure or service failures, dynamically acquire computing resources to meet demand, and mitigate disruptions
4. **Performance efficiency** - The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve.
5. **Cost Optimization** - The ability to avoid or eliminate unneeded cost or suboptimal resources
6. **Sustainability** - Guidance on how AWS can help you reduce your carbon footprint, and best practices you can use to improve the sustainability of your workloads.

# 1. Operational Excellence

- Focus on **People, Process and Technology** to deliver business value
- Implement **observability** to get actionable insights
- **Automate** where possible and keep **experimenting** with ability to revert the change automatically
- Make frequent, small, reversible changes to reduce blast radius
- Refine operations procedures frequently and effectively communicate with your teams
- **Anticipate failure** – everything fails, all the time (Dr. Werner Vogels, CTO)
- Learn from past operational events and metrics
- Use managed services to reduce operational burden



## 2. Security

- Implement principle of **least privilege** and separation of duties
- Maintain **traceability** – monitor, audit and alert
- Apply **security at all layers** – Data, Network, Code, Application, Operating system , Identity & access permissions
- Automate security best practices
- Protect data in transit and at rest
- Keep people and data separate with no direct access to data
- Prepare for **security events** - Build architectures and automation for quick detection, investigation and recovery from security events



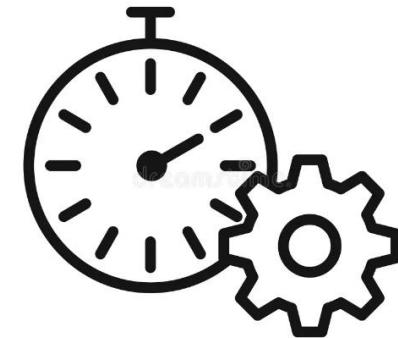
# 3. Reliability

- Build automation to **detect the failures** and to **recover** from failure automatically
- Test recovery procedures e.g. Chaos engineering
- **Scale horizontally** with multiple small resources to distribute the risk of failure in single large system
- **Stop guessing capacity** by leveraging Cloud's elasticity and scalability
- Manage change through automation, even the smallest change



# 4. Performance Efficiency

- Adapt the **latest** and greatest of **technologies** early by using services offered by Cloud provider
- **Go global in minutes** by deploying your application in any of the AWS regions providing lower latency and meeting regulatory and compliance requirements
- Use **serverless** architectures to scale on-demand as per the number of requests or events.
- Evaluate multiple options and choose the best for your needs



# 5. Cost Optimization

- Build **Cloud Financial Management** to manage your cloud spend just like you have security and operational teams
- Adopt a **consumption** model – Only use what and when you need
- Stop spending money on **undifferentiated heavy-lifting** and focus on building great products and delivering value to the customer. Leave the rest to AWS like racking, stacking and powering servers !
- Analyze expenditure for your applications/systems with the cost mechanisms that AWS provides (e.g. Cost explorer, Cost allocation tags etc.). This helps measure your ROI.



# 6. Sustainability

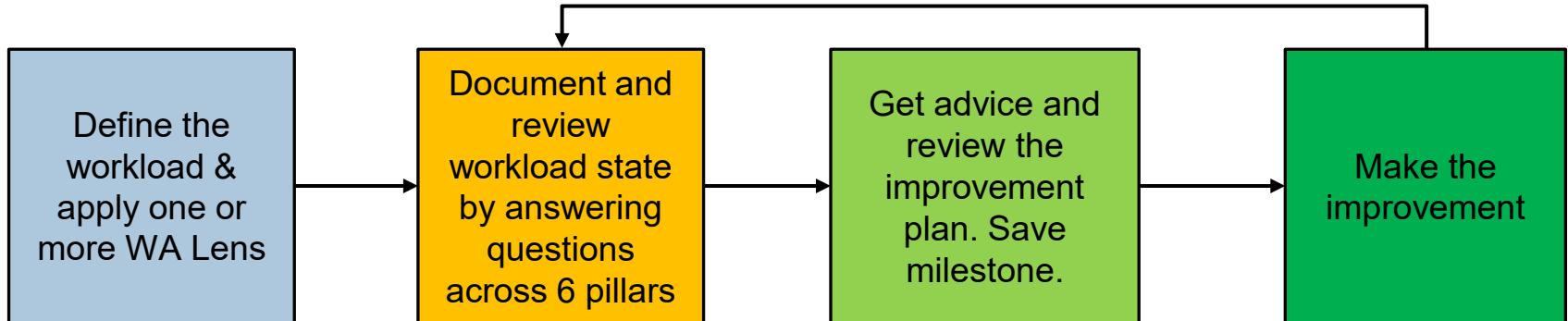
- Measure the Sustainability impact of your workloads – AWS **Customer Carbon Footprint** tool
- Set sustainability goals for your workloads by tracking the impact and making improvements with recommended actions
- Maximize **resources utilization**
- Anticipate and adopt new, more efficient hardware and software offerings
- Reduce the impact of your cloud workloads by using managed and shared services



# AWS Well-architected Tool



- Free tool to review your architectures against the 6 pillars Well-Architected
- Tool available at: <https://console.aws.amazon.com/wellarchitected>
- Can apply one or more Well-architected **Lens** for industry or domain specific workload e.g. IoT, Connected Mobility, SaaS, Serverless, DevOps, Migration and more..





# AWS Cloud Adoption Framework

# AWS Cloud Adoption Framework - CAF



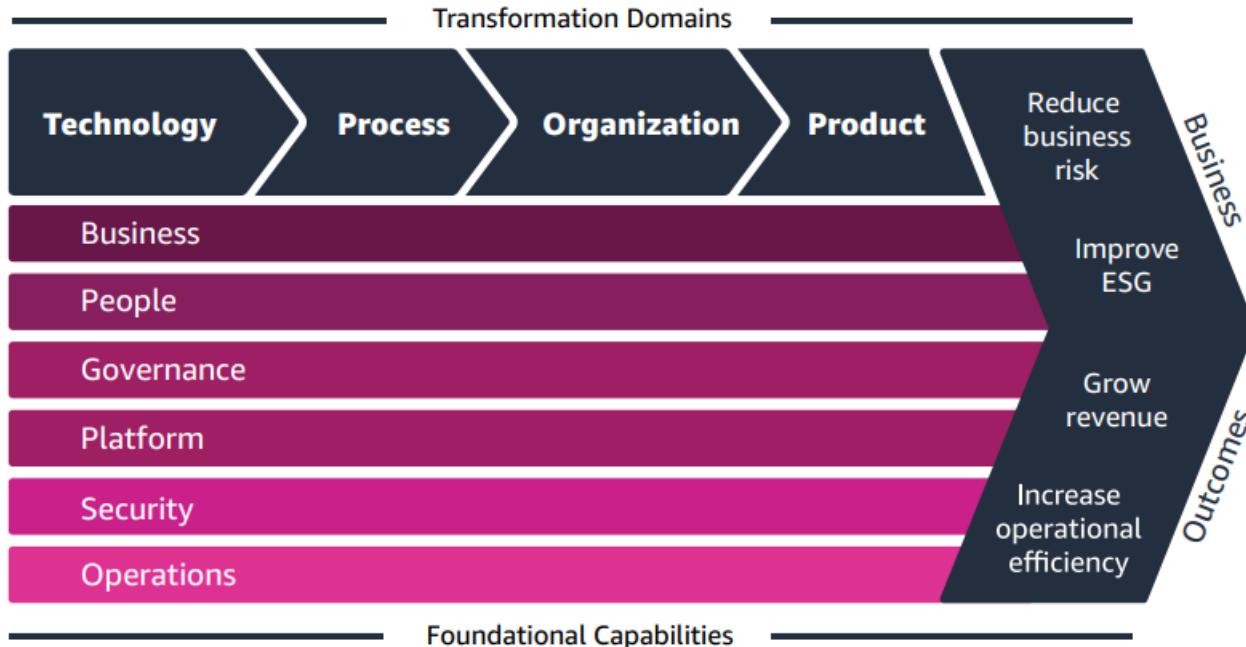
- The AWS Cloud Adoption Framework (AWS CAF) leverages AWS experience and best practices to help customers digitally transform and accelerate their business outcomes through innovative use of AWS.
- AWS CAF improves Cloud Readiness
- For successful digital transformation organization needs to work across four transformation domains and six foundational capabilities:
  - ✓ Technology, Process, Organization, Products
  - ✓ Business, People, Governance, Platform, Security, and Operations



<https://d1.awsstatic.com/whitepapers/aws-caf-ebook.pdf>

# AWS CAF – Four Transformation domains

- Technology, Process, Organization and Products



# AWS CAF – Six foundational capabilities

## Business

- Strategy Management
- Portfolio Management
- Innovation Management
- Product Management
- Strategic Partnership
- Data Monetization
- Business Insight
- Data Science

## People

- Culture Evolution
- Transformational Leadership
- Cloud Fluency
- Workforce Transformation
- Change Acceleration
- Organization Design
- Organizational Alignment

## Governance

- Program and Project Management
- Benefits Management
- Risk Management
- Cloud Financial Management
- Application Portfolio Management
- Data Governance
- Data Curation

# AWS CAF – Six foundational capabilities

## Platform

Platform Architecture  
Data Architecture  
Platform Engineering  
Data Engineering  
Provisioning and Orchestration  
Modern Application Development  
Continuous Integration and Continuous Delivery

## Security

Security Governance  
Security Assurance  
Identity and Access Management  
Threat Detection  
Vulnerability Management  
Infrastructure Protection  
Data Protection  
Application Security  
Incident Response

## Operations

Observability  
Event Management (AIOps)  
Incident and Problem Management  
Change and Release Management  
Performance and Capacity Management  
Configuration Management  
Patch Management  
Availability and Continuity Management  
Application Management

# AWS CAF – Six perspectives



Business Perspective

Business Perspective helps ensure that **your cloud investments** accelerate your digital transformation ambitions and **business outcomes**. Common stakeholders include chief executive officer (CEO), chief financial officer (CFO), chief operations officer (COO), chief information officer (CIO), and chief technology officer (CTO).



People Perspective

People Perspective serves as a bridge between technology and business, accelerating the cloud journey **with focus on culture, organizational structure, leadership, and workforce**. Common stakeholders include CIO, COO, CTO, cloud director, and cross-functional and enterprise-wide leaders.



Governance Perspective

Governance Perspective helps you orchestrate your cloud initiatives while maximizing organizational benefits and **minimizing transformation-related risks**. Common stakeholders include chief transformation officer, CIO, CTO, CFO, chief data officer (CDO), and chief risk officer (CRO).

# AWS CAF – Six perspectives



## Platform Perspective

Platform Perspective helps you build an **enterprise-grade, scalable, hybrid cloud platform**; modernize existing workloads; and implement new cloud native solutions. Common stakeholders include CTO, technology leaders, architects, and engineers.



## Security Perspective

Security Perspective helps you achieve the **confidentiality, integrity, and availability of your data and cloud workloads**. Common stakeholders include chief information security officer (CISO), chief compliance officer (CCO), internal audit leaders, and security architects and engineers.

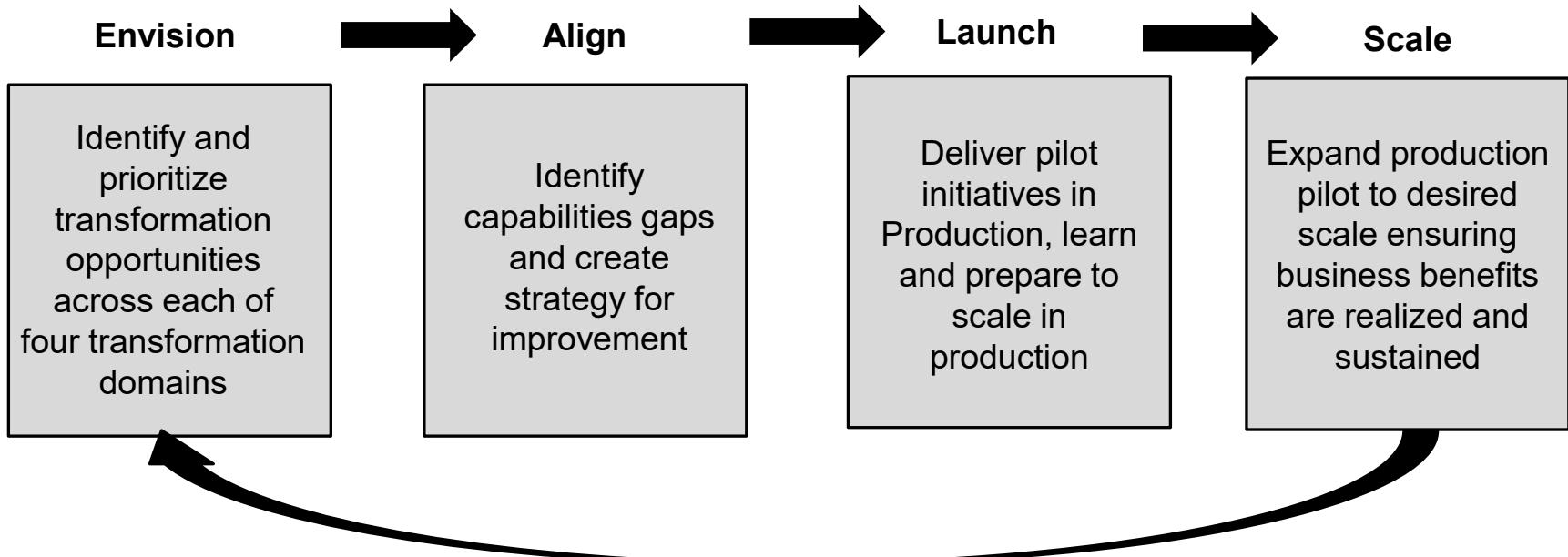


## Operations Perspective

Operations Perspective helps ensure that your **cloud services are delivered** at a level that meets the **needs of your business**. Common stakeholders include infrastructure and operations leaders, site reliability engineers, and information technology service managers.

# AWS CAF – Four cloud transformation phases

- The AWS CAF recommends four iterative and incremental cloud transformation phases shown in the following figure.



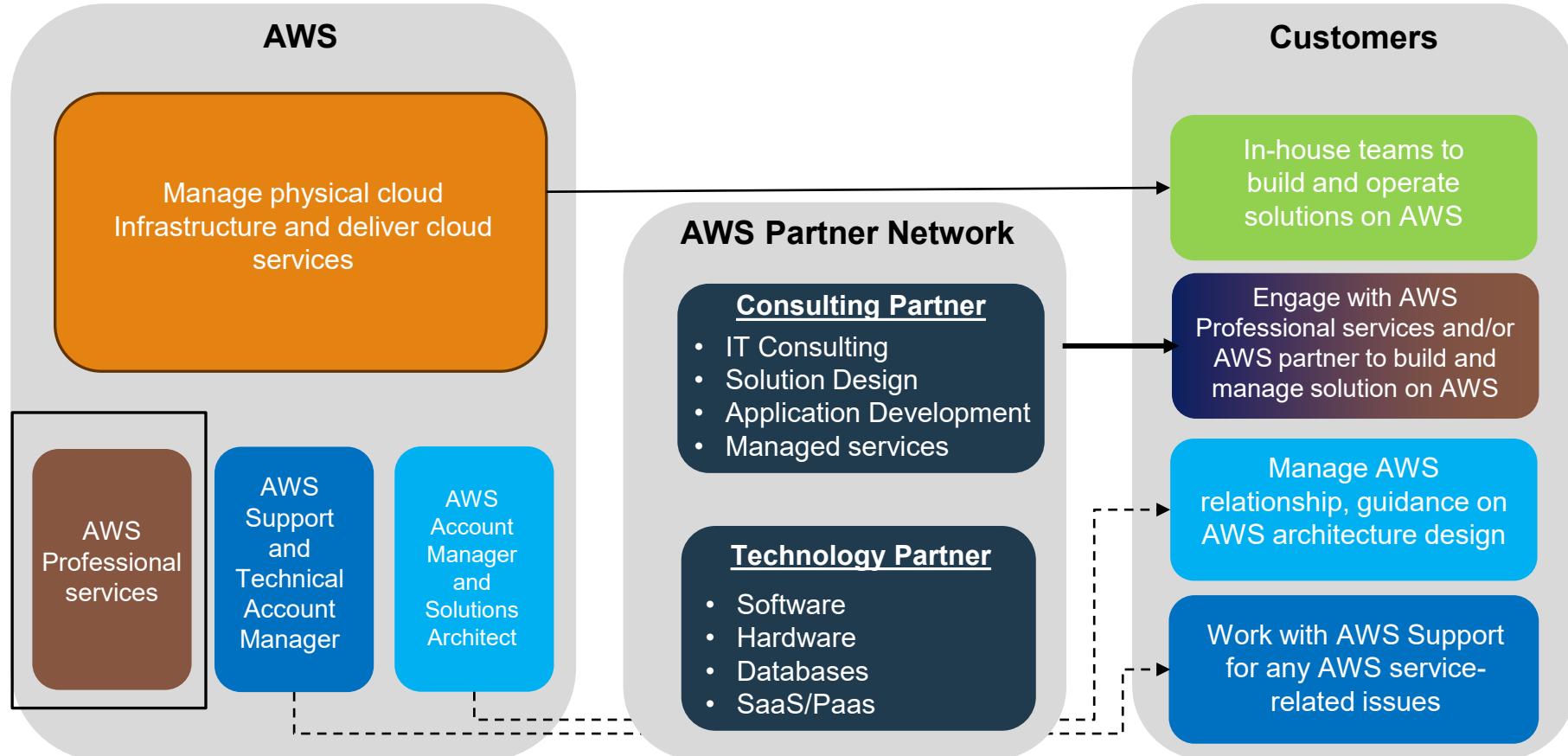
# AWS Well-architected and CAF - summary

1. AWS Well-Architected helps to build secure, high-performing, resilient, and efficient infrastructure
2. Six pillars of AWS Well-Architected Framework - Operational excellence, Security, Reliability, Performance efficiency, Cost optimization, and Sustainability
3. AWS Well-Architected Tool to review your application architecture against six pillars and optionally for industry or technology specific Lens and receive improvement recommendations.
4. AWS Cloud Adoption Framework (CAF) helps digitally transform the businesses across four transformation domains – Technology, People, Process and Products
5. CAF defines foundational capabilities and perspectives across six areas – Business , People, Governance, Platform, Security and Operations
6. Understand the different areas and responsibilities under foundational capabilities and perspectives
7. CAF defines four iterative stages of transformation – Envision, Align, Launch and Scale

# AWS Partners, Professionals, Community & More

# AWS engagement models

# AWS engagement model



# AWS Professional services



## AWS Professional Services

- Global team of AWS experts available for short-term contracting for building new solutions on AWS
- They are AWS employees



## AWS IQ

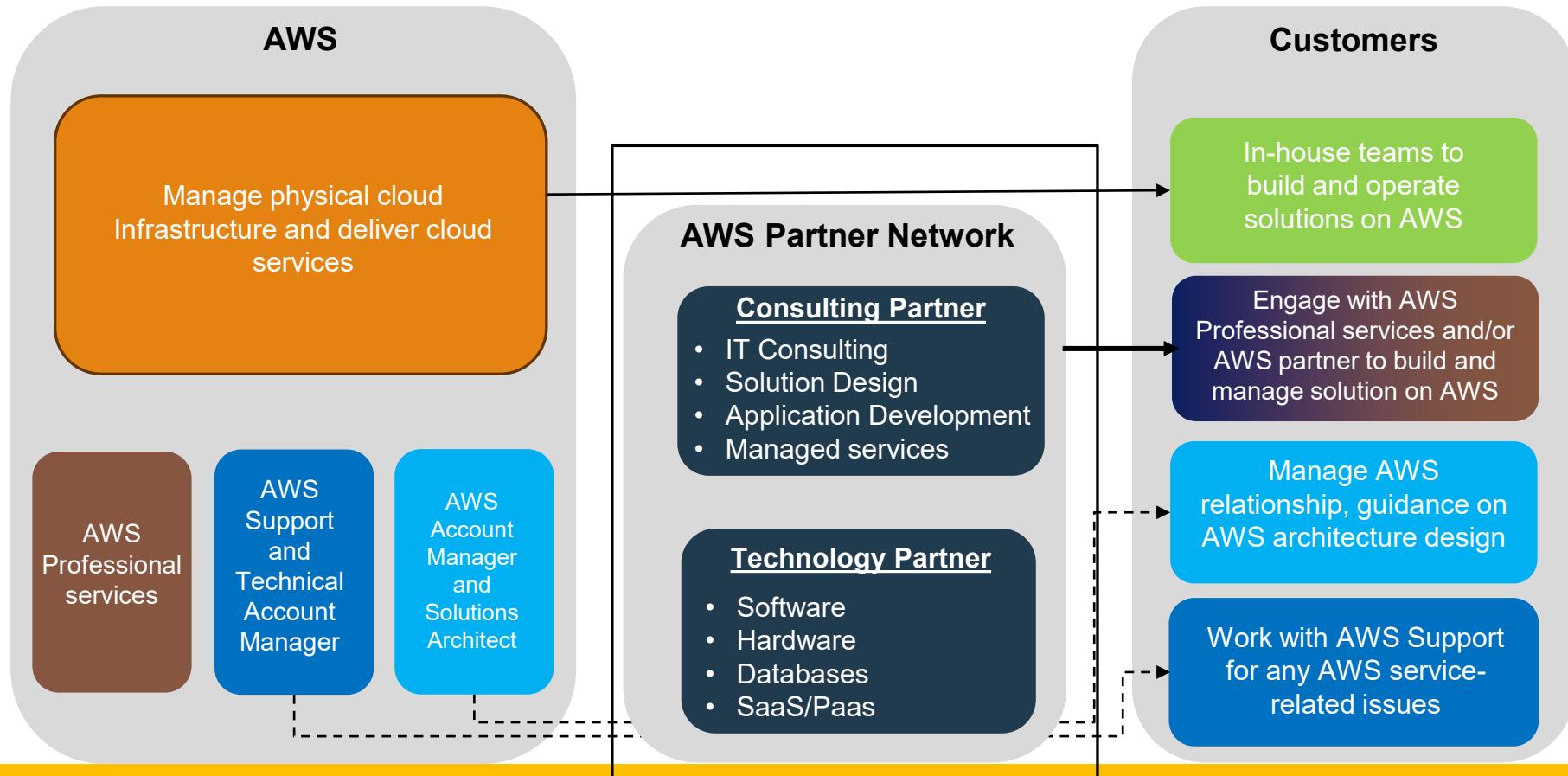
- A service that connects customers with AWS Certified third-party experts for on-demand project work



## AWS Managed Services

- AMS offers a tools and services to adopt AWS at scale and operate your AWS infrastructure and services securely

# AWS Partners



# AWS Partner Network

- While customers can build with and use Amazon Web Services (AWS) services directly from AWS, they can also work with AWS Partner Network
- There are different types of the partners e.g. Consulting Partners, Technology Partners and Training Partners
- There are many AWS partner programs such as AWS Competency Programs under which proficient APN partners are granted specific competencies e.g. Automotive, Healthcare, Financial services etc.

## Consulting Partners

- IT Consulting
- Solution Design
- Application Development
- Managed services

## Technology Partners

- Software
- Hardware
- Databases
- SaaS/PaaS

## Training Partners

- Classroom Training
- Digital Training

Partnership  
Tiers

Registered

Select

Advanced

Premier

# AWS Training & Certification



- AWS Digital Training – Provides access to 600+ free digital courses and 1000+ labs
- AWS Classroom Training – In person training where you can interact with the instructor
- AWS Skill Builder – Subscription based learning platform (similar to Udemy)
- AWS Educate – For new-cloud learners (as young as 13 years of age)
- AWS Academy - At higher educational institute to prepare students for their career in the Cloud



# AWS re:post - Ask questions, connect with AWS community



Expert-led AWS community with curated answers, articles, and access to the AWS Knowledge Center



AWS re:post



AWS re:post private

The screenshot shows the AWS re:Post interface. At the top, there's a navigation bar with the AWS logo, a search bar, language and resources dropdowns, and a sign-in button. Below the navigation is a main menu with Home, Questions (which is highlighted), Knowledge Center, Articles, Selections, Tags, Topics, and More... On the right side of the menu is a yellow 'Ask question' button. The main content area displays six posts in a grid:

- Amazon QuickSight access error**  
Hello team, I am trying to access amazonbi account on Amazon QuickSight, i have requested access and it seems i should have permissions to view the required dashboard, however when i try to login, the...  
  
ACCEPTED ANSWER | Amazon QuickSight  
1 answers 0 votes 13 views | posted by repost-user-9568839 an hour ago
- Lambda URL with x-amz-content-sha256 header**  
Hey guys, I want to ask can we (from client) create x-amz-content-sha256 then add it to header each time we request to AWS URL? How to implement that solution. Thank you!!!  
  
Lambda@Edge | Amazon CloudFront | AWS Lambda  
1 answers 0 votes 20 views | posted by kaii an hour ago
- Are Cognito custom authentication with adaptive MFA functionality compatible?**  
Hi all I'm looking for some advice on to what extent Cognito's Custom Authentication (define-, create- and verify auth challenge lambdas) can integrate with 'Advanced Security' and adaptive...  
  
Security | Multi-factor Authentication | ...  
0 answers 0 votes 12 views | posted by reyes 2 hours ago
- Client VPN with VPC peering vs Client VPN withTransit gateway**  
A company has introduced a new policy that allows employees to work remotely from their homes if they connect by using a VPN. The company is hosting internal applications with VPCs in multiple AWS...  
  
Amazon VPC | AWS Transit Gateway | AWS Client VPN  
0 answers 0 votes 0 views | posted by [user] an hour ago
- 1/3 checks passed for Ec2**  
Hi Team, My Ec2 has 1/3 checks passed which is new and I want to know what it is for?  
  
Support Case | High Performance Compute | ...  
0 answers 0 votes 0 views | posted by [user] an hour ago
- Getting error "The request signature we calculated does not match the signature yo...**  
Hi I am working on Mendix application, there I am trying to call \*\*Invoke Model\*\* API and getting ("message":"The request signature we calculated does not match the signature you provided. Check your...  
  
Amazon Bedrock  
0 answers 0 votes 0 views | posted by [user] an hour ago

<https://repost.aws/>



# AWS re:post Knowledge center

AWS Official Knowledge Center articles and videos covering the most frequent questions and requests that AWS receives from AWS customers.

Screenshot of the AWS re:Post Knowledge Center website:

The screenshot shows the AWS re:Post Knowledge Center interface. At the top, there's a navigation bar with links for Home, Questions, Knowledge Center (which is highlighted), Articles, Selections, Tags, Topics, Community Groups, AWS Support Official, and Sign In. There's also a search bar and language/resource dropdowns.

The main content area has sections for "Knowledge Center" and "Featured".

**Knowledge Center:** A note says "AWS re:Post includes AWS Official Knowledge Center articles and videos covering the most frequent questions and requests that we receive from AWS customers." It includes a link to "Browse all Knowledge Center content".

**Featured:** Three cards are shown:

- Why aren't my Amazon S3 objects replicating when I set up replication between my buckets?** Updated a year ago. It discusses cross-Region replication (CRR) or same-Region replication (SRR) between Amazon Simple Storage Service (Amazon S3) buckets. Objects aren't replicating to the destination bucket.  
Tags: KNOWLEDGE CENTER, Amazon Simple Storage Service.
- Why can't I connect to my EC2 instance?** Updated a year ago. It discusses connecting to an Amazon Elastic Compute Cloud (Amazon EC2) instance, mentioning an error when trying to connect.  
Tags: KNOWLEDGE CENTER, Amazon EC2, Linux, Windows.
- How can I reactivate my suspended AWS account?** Updated 6 months ago. It discusses regaining access to a suspended AWS account and services.  
Tags: KNOWLEDGE CENTER, AWS Account Management.

**Newly created:** Three cards are shown:

- What AWS Services support reserved nodes size flexibility or Reserved Instances size flexibility?** Updated a day ago. It discusses what AWS Services support reserved nodes size flexibility or Reserved Instances size flexibility.  
Tags: KNOWLEDGE CENTER, AWS Account Management.
- How do I deactivate single tunnel notifications for my AWS Site-to-Site VPN?** Updated 14 days ago. It discusses turning off "AWS\_VPN\_SINGLE\_TUNNEL\_NOTIFICATION" messages for Site-to-Site Virtual Private Network (VPN) connections.  
Tags: KNOWLEDGE CENTER, AWS Virtual Private Network (VPN).
- How do I configure an AWS Site-to-Site VPN?** Updated 10 days ago. It discusses configuring an AWS Site-to-Site Virtual Private Network (VPN).  
Tags: KNOWLEDGE CENTER, AWS Virtual Private Network (VPN).

<https://repost.aws/knowledge-center>

# AWS Partners, Professionals & Community - summary

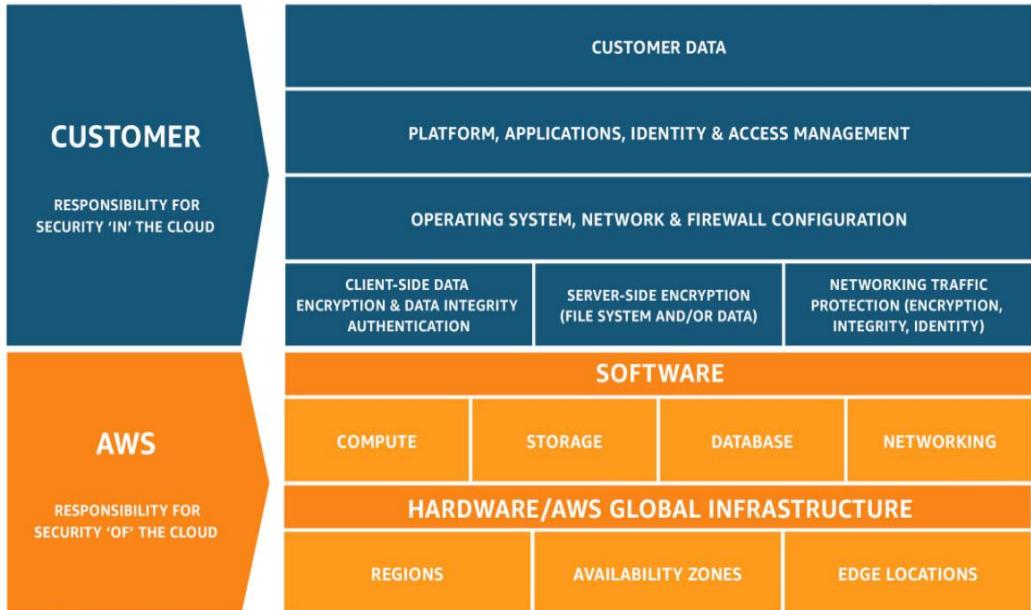
- For consulting work you can either work with AWS Professional services or AWS consulting partners from APN
- While AWS Professional services is a team of AWS experts employed by AWS, there is also AWS IQ which is the community of AWS experts you can connect to outsource your work
- AWS also offers AWS Managed Services (AMS) to help organizations adopt AWS at scale and operate securely in the cloud
- APN partners are of different types namely Consulting partners, Technology partners and Training partners and at different levels of tier as per their maturity
- AWS offers different modes of Training in the form of Online, Classroom, instructor led trainings for organizations and educational institutes
- AWS re:post is a public forum to discuss AWS issues. There is also private version of re:post which you can use inside your organization to collaborate internally.

# AWS Shared Responsibility Model

# AWS shared Responsibility Model

- **Security and Compliance is a shared responsibility between AWS and the customer.**
- Shared responsibility model enable customers to manage risk effectively and efficiently.
- When it comes to managing security and compliance in the AWS Cloud, each party has distinct responsibilities.
- **AWS** operates, manages and controls the components from the host operating system and virtualization layer down to the physical security of the facilities in which the service operates.
- The customer assumes responsibility and management of the guest operating system , other associated application software as well as the configuration of the AWS provided security group firewall.

# AWS Shared Responsibility model



## "Security in the Cloud" - Customer responsibility

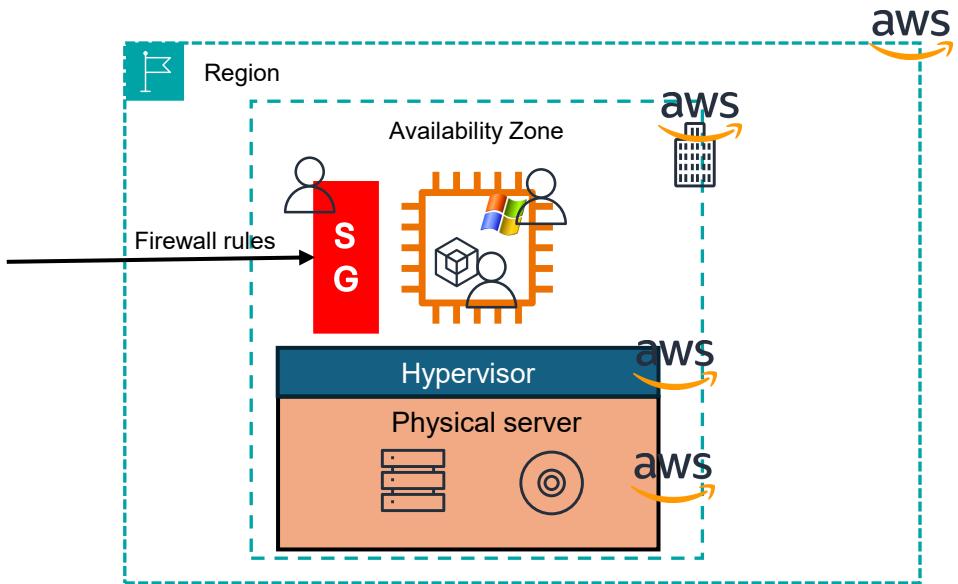
- Depends on AWS service
- Typically protecting data, access, firewall configurations is customer's responsibility



## "Security of the Cloud" - AWS responsibility

- Protecting the infrastructure that runs all of the services offered in the AWS Cloud.
- Physical security of the data center facilities
- Hardware, software, networking for fully managed services like DynamoDB, S3

# AWS Shared responsibility model



## "Security in the Cloud" - Customer responsibility

- EC2 guest operating system patching, updates
- Security group rules
- Software installed in the EC2
- SSH access to EC2
- IAM role for EC2



## "Security of the Cloud" - AWS responsibility

- Physical security of the data center facilities (Regions and AZ)
- Physical hardware racks, storage disks
- Hypervisor security, patching, updates

# Shared responsibility for Compute



- Securing the physical infrastructure, including data centers, servers, networking hardware, and foundational networking.
- Managing the hypervisor layer, ensuring no vulnerabilities allow access between instances on the same physical server.
- For AWS Lambda, ECS, Fargate etc. - Patching the runtime environments and maintaining the managed execution platform
- Configuring security at the OS level, including patching, firewall settings, and vulnerability scanning.
- Managing access controls (e.g., IAM roles and policies) and key management.
- Encrypting data at rest or in transit
- Managing application-level security, including installing and maintaining antivirus or intrusion detection systems.
- Writing secure code and implementing error handling.

# Shared responsibility for Storage



- Underlying infrastructure for AWS storage services
- Direct physical access to the disk
- Vulnerability and compliance of the physical disks
- Maintaining multiple copies of data for durability as per SLA e.g. copies across AZs for S3 standard, multiple copies within AZ for EBS
- Access and control of the data
- Taking backups (snapshots)
- Enable S3 versioning
- S3 bucket policy & permissions
- Enabling logging and monitoring
- Using appropriate storage class
- Enabling data encryption

# Shared responsibility for **Databases**



- Provision and manage the underlying EC2 instance hosting RDS database
- Automated backups (feature of RDS)
- Underlying OS and DB patching
- Database maintenance
- Public or Private network access to the database
- Security group inbound rules for database instance (e.g. 3306 port to be opened for MySQL DB)
- Enabling encryption for data at rest
- Database user creation and permissions
- Manual backups (on-demand)

# AWS Acceptable Use Policy

- The AWS Acceptable Use Policy <https://aws.amazon.com/aup/> governs your use of the services offered by Amazon Web Services, Inc.
- You may not use, or facilitate or allow others to use, the Services or the AWS website:
  - For any illegal or fraudulent activity;
  - To violate the rights of others;
  - To threaten, incite, promote, or actively encourage violence, terrorism, or other serious harm;
  - For any content or activity that promotes child sexual exploitation or abuse;
  - To distribute, publish, send, or facilitate the sending of unsolicited mass email or “spam” messages.

For reporting any violations, you can visit <https://support.aws.amazon.com/#/contacts/report-abuse>

# Other AWS Services

Good to know for your exam

# Few more AWS services

- AWS Migration services
- AWS Marketplace
- AWS License Manager
- Other AWS services..

# AWS Migration services

# AWS Migration Services



AWS Application  
Discovery Service  
(ADS)

- Capture system and applications inventory and dependencies
- Agent based and Agentless tools
- Connect and collect details from on-premises IT systems, CMDB etc.



AWS Migration Hub

- Consolidate discovery data across regions
- Visualize discovery data in Quicksight
- Plan for right sized EC2 instances
- Track the entire migration centrally



AWS Application  
Migration Service (MGN)

- Replicate application and databases to AWS EC2
- Minimum downtime for source application during cut over time
- Example apps: Oracle, peoplesoft, SAP CRM, Apache, Active directory and more
- Example DBs: SQL server, SAP HANA, MySQL and more



Agentless Collector



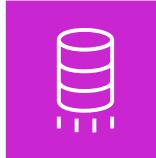
Discovery Agent

# AWS Migration Services



AWS Application  
Migration Service (MGN)

vs



AWS Database Migration  
Service (AWS DMS)

- Replicate application and databases to **AWS EC2**
- It's a server based like to like migration

- Migrates on-premises databases to Amazon RDS, Aurora, Amazon Redshift or DynamoDB databases
- Supports both homogeneous database migration (like to like) and heterogeneous migration (e.g. SQL server to Aurora Postgres)

# AWS Marketplace

- Independent Software Vendors (ISVs) sell their products, solutions and data through Digital Marketplace
- You can buy or say subscribe to required product and start consuming it by launching EC2 instances, Containers or CloudFormation stacks or using vendor's Software-as-a-Service platform
- Product license cost is automatically included in your AWS Bill
- Examples:
  - OpenVPN access server (EC2 AMI based)
  - Antivirus for Amazon S3 (Container based)
  - MongoDB Atlas (SaaS)
  - CrowdStrike's Falcom endpoint protection (SaaS)
  - 20 years of Stock data by Alpha Vantage (Data)
- You can sell your software on AWS Marketplace by going through AWS Seller Registration process.



<https://aws.amazon.com/marketplace>

# AWS License Manager

- AWS License Manager is a service that manages software licenses in **AWS** and **on-premises** environments
- It supports Bring-Your-Own-License (BYOL) for third-party workloads such as Microsoft Windows Server, SQL Server etc.
- It enables administrators to create licensing rules which can stop the EC2 instances from launching or by notifying administrators about the license infringement (violation of a law).
- Central visibility of all the licenses using AWS License Manager dashboard.
- Integrates with AWS Systems Manager to manage licenses on physical or virtual servers hosted outside of AWS using AWS License Manager.
- Supports AWS Organization enabling centralized license management



AWS License Manager

# Other AWS services

Just in case you see these in your exam..

# AWS End User Computing services



## Amazon WorkSpaces

- Cloud desktops
- Full desktop app and peripheral support
- **Persistent desktop experience**
- Active Directory integration for users to use their existing credentials
- PCoIP and WSP (Built on DCV)
- HIPAA eligible, PCI, FedRamp, ITAR, SOC 1,2,3, ISO



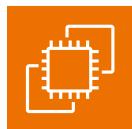
## Amazon AppStream 2.0

- Streamed applications
- Build a curated app catalog
- **Non-persistent experience** - Instance replacement after each session
- Active Directory is Optional
- SAML 2.0 federation and built-in identity management
- Accelerated remote 3D graphics based on Amazon DCV
- HIPAA eligible, FedRamp PCI DSS, SOC 1,2,3, ISO

# AWS Backup



- Centralizes and automates backups for data across AWS services and on-premises.
- Uses Backup plans to configure Frequency of the backup, backup time window, AWS resources to backup, storage tier for backup etc.
- Supports lifecycle management policies to move backups into low cost (cost) storage
- Supports Cross-region backups where backups are stored in another region to recover from region level failure
- Supports Cross-account backups
- Backups are incremental (only first backup is a fully copy) for the supported resources
- Integrates with most of the AWS services such as -



EC2



EBS



S3



DynamoDB



RDS



EFS



FSx

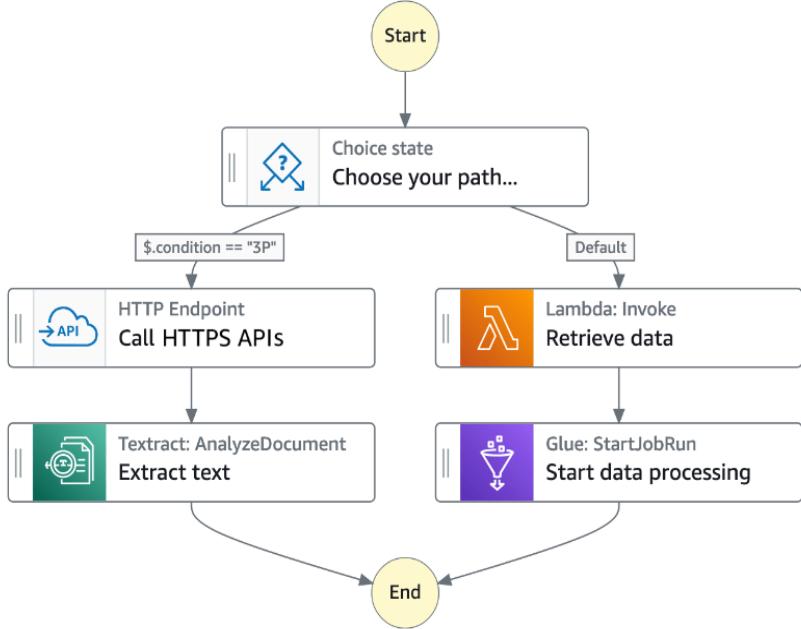
Storage  
Gateway

On EC2

and more..

# AWS Step functions

- With AWS Step Functions, you can create workflows called State machines
- Used to build distributed applications, automate processes, orchestrate microservices, create data and machine learning pipelines.
- Supports different types of tasks such as Orchestration task, Choice task, Retry task, Add human in the loop, process data in parallel and process data with the map
- Step function integrates with most of the AWS services including AWS Lambda, Amazon API Gateway, AWS Batch, AWS CodeBuild, Amazon DynamoDB, Amazon ECS/Fargate, AWS Elemental Media Convert, AWS Glue, Amazon EventBridge, Amazon SQS, Amazon SNS and many more..



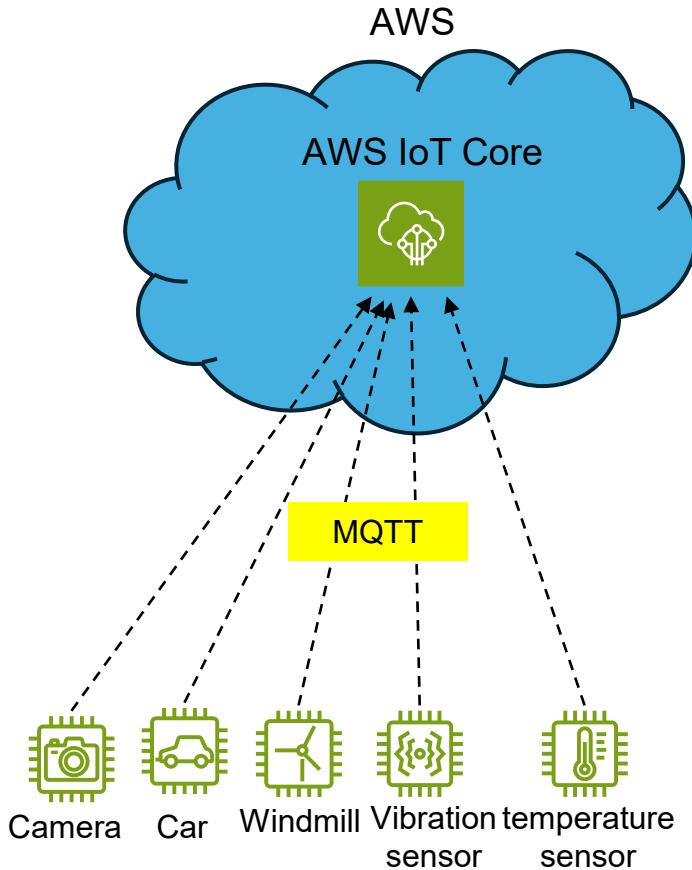
# Other AWS services – Section summary

- **AWS Migration services**
  - AWS Application Discovery service (ADS) – To discover on-premises application & server inventory
  - AWS Migration Hub – Plan, manage and track the entire migration
  - AWS Application Migration service (MGN) – Migrate Applications and Servers to AWS
- **AWS Marketplace** – A marketplace for ISVs to sell their products on AWS
- **AWS License Manager** – Centrally manage the software licenses across AWS and on-premises
- AWS End User Computing Services
  - **Amazon Workspaces** – Managed VDI service
  - **Amazon Appstream** – Managed Application streaming service
- **AWS Backup** – Automate backups across AWS and on-premises
- **AWS Step function** – Build workflows for distributed applications and microservices

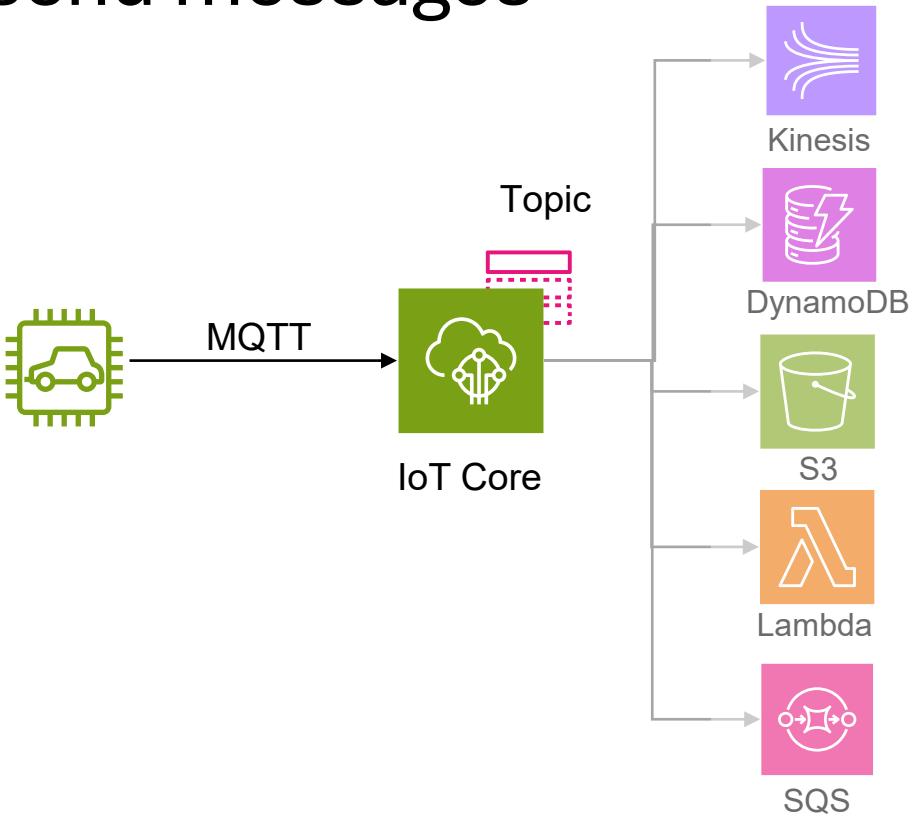
# AWS IoT

# AWS IoT Core

- IoT is "Internet of Things"
- Used to build IoT applications where data is continuously collected from field devices & sensors and sent to cloud where data is processed, analyzed and stored.
- Connect IoT devices to the AWS Cloud over **MQTT5 protocol**
- It's a serverless platform and can connect billions of devices and process trillions of messages per day
- **Use cases:** Connected Cars, Connecting and monitoring of Industrial machines, Smart cities, Smart devices, Anomaly detection



# Process to connect IoT thing to IoT Core and send messages

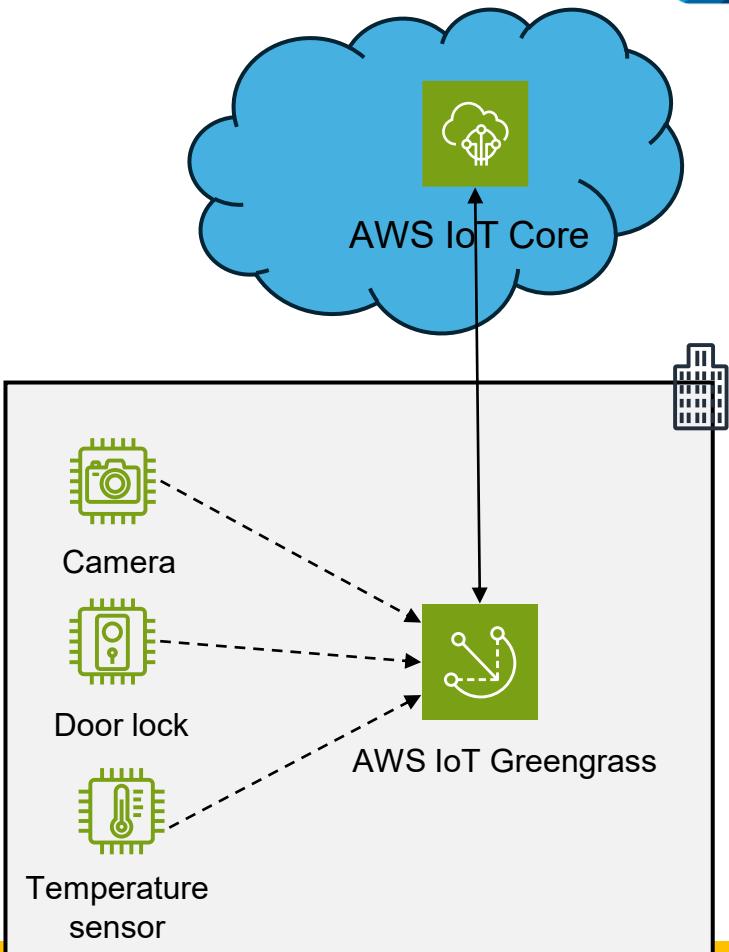


- 1 Create new IoT Thing
- 2 Download the certificates and SDK and install on your client device
- 3 Install SDK onto the Client device and configure the device to use IoT Thing certificate for registration of the device and secure connection. Run the client device application to send messages to IoT Core endpoint.
- 4 Run the device program to connect and send messages
- 5 Configure IoT Core Rules to process the messages e.g. send it to Kinesis data stream or SQS queue or store in DynamoDB or S3

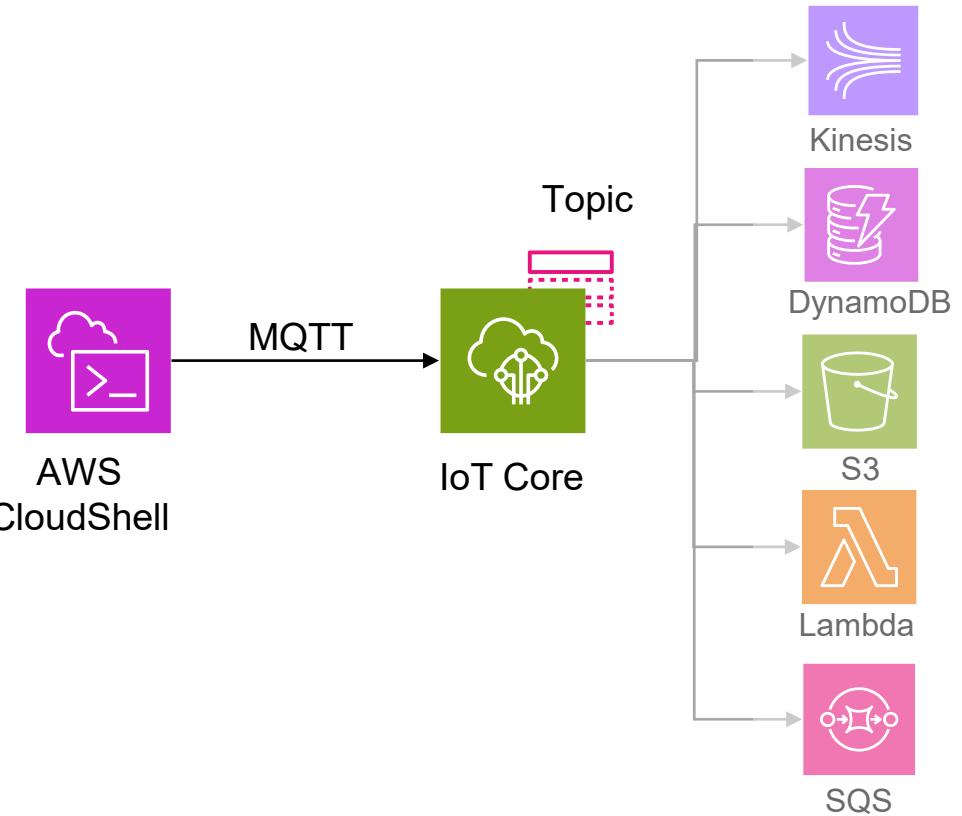
# AWS IoT Greengrass



- AWS IoT Greengrass enables your devices to collect and analyze data closer to where that data is generated, react autonomously to local events, and communicate securely with other devices on the local network.
- Facilitates offline operation and periodic synchronization with AWS Cloud.
- Enables remote device management and device software updates
- **Use cases:** Safe homes, industrial predictive maintenance, autonomous vehicle etc.



# Exercise – Connect IoT thing to IoT core



- 1 Create new IoT Thing
- 2 Download the Thing certificates and python SDK zip file from IoT core
- 3 Run the AWS CloudShell and upload the zip file. Unzip zip file. There should be start.sh file.
- 4 Change the start.sh file permissions to 755 and run the script. Script should connect to IoT Core endpoint and start sending the messages to the topic 'sdk/test/python'
- 5 Verify messages from the IoT Console

# Exam preparation

- Know your exam
- AWS sample exam questions
- How to approach exam questions – Tips
- Practice Test
- Getting 30 mins extra time for the exam (as applicable)
- Scheduling the exam
- Last minute revision – Just read through all section summary slides
- Getting 50% discount on your next AWS certification exam

# Know your exam..

- Exam: **AWS Certified Cloud Practitioner**
- Exam code: CLF-C02
- Total 65 questions
- 50 questions are scored and 15 are unscored
- There are two types of questions on the exam:
  - Multiple choice: Has one correct response and three incorrect responses (distractors)
  - Multiple response: Has two or more correct responses out of five or more response options
- Minimum passing score: 700/1000

<b>Category</b>	Foundational
<b>Exam duration</b>	90 minutes
<b>Exam format</b>	65 questions; either multiple choice or multiple response
<b>Cost</b>	100 USD. Visit <a href="#">Exam pricing</a> for additional cost information, including foreign exchange rates
<b>Test in-person or online</b>	Pearson VUE testing center or online proctored exam
<b>Languages offered</b>	English, Japanese, Korean, Simplified Chinese, Traditional Chinese, Bahasa (Indonesian), Spanish (Spain), Spanish (Latin America), French (France), German, Italian, and Portuguese (Brazil)



<https://aws.amazon.com/certification/certified-cloud-practitioner/>

# Exam sample questions



[https://d1.awsstatic.com/training-and-certification/docs-cloud-practitioner/AWS-Certified-Cloud-Practitioner\\_Sample-Questions.pdf](https://d1.awsstatic.com/training-and-certification/docs-cloud-practitioner/AWS-Certified-Cloud-Practitioner_Sample-Questions.pdf)

# How to approach exam questions?

## Multiple choice

A company wants to save costs by purchasing EC2 instances for predictable workloads that run continuously for one year. Which purchasing option is best for this use case?

- On-Demand
- Reserved Instances
- Spot Instances
- Dedicated Hosts

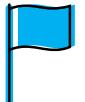
## Multiple responses

How does AWS charge for AWS Lambda usage once the free tier has been exceeded? (Select TWO.)

- By the time it takes for the lambda function to run
- By number of the versions of the lambda function
- By the number of the requests made for a given lambda function
- By the programming language
- By the total number of Lambda functions in the account

# How to approach exam questions?

- ✓ Don't spend more than a minute for a question.
- ✓ If in doubt, select the answer which came first to your mind and Mark that question for review
- ✓ Even if you do not have any idea about the question, just answer it and Mark the question for review
- ✓ Try to finish the first pass over all 65 questions in around 60 mins or less
- ✓ In last 15-30 mins – Go over all the Marked for review questions and double check your answers



Mark for review

# Practice Test

- 65 Questions
- Supporting explanation for correct and incorrect answers
- Retake as many times

# Getting 30 mins extra time for the exam

- For non-native English speakers AWS provides 30 mins extra time
- You can request it through AWS certification portal -> Exam accommodations -> ELS + 30 accommodation
- Once opted, it will be there for all your future AWS certification exams

The screenshot shows the AWS Certification Portal interface. On the left, there's a sidebar with links: HOME, PROFILE, EXAM REGISTRATION (which is expanded, showing 'Schedule an exam' and 'Exam accommodations'), EXAM HISTORY, CERTIFICATIONS, BENEFITS, DIGITAL BADGES, and SUPPORT AND FAQS. The 'Exam accommodations' link is highlighted with a blue border. To the right, the main content area has a header 'Exam accommodations'. Below it, a section titled 'Requesting new accommodations' contains a bulleted list:

- Extra 30 minutes for non-native English speakers: You must request the
- NOTE If you already scheduled the exam and realized you forgot to add ESL + 30 minutes, you can request a refund.
- All other accommodations: [Request Pearson VUE exam accommodations](#)

Below this, there's a search bar and a table with columns 'SORT BY', 'Name', 'Expiration Date', and 'Review State'. A row in the table is highlighted in green and labeled 'Approved'. The details for this row are: 'ESL +30 MINUTES', 'EXPIRATION DATE Never'.

# Congratulations !



- ✓ Kindly leave the review for this course



- ✓ Share your Certification achievement on LinkedIn



Chetan Agrawal - 200,000+ students, 4M+ views for youtube videos..

# Let's connect



Subscribe to my YouTube channel



<https://www.youtube.com/@AWSwithChetan>



Connect with me on LinkedIn



<https://www.linkedin.com/in/chetan-agrawal-30107310/>



Chetan Agrawal - 200,000+ students, 4M+ views for youtube videos..

# Thank you !!