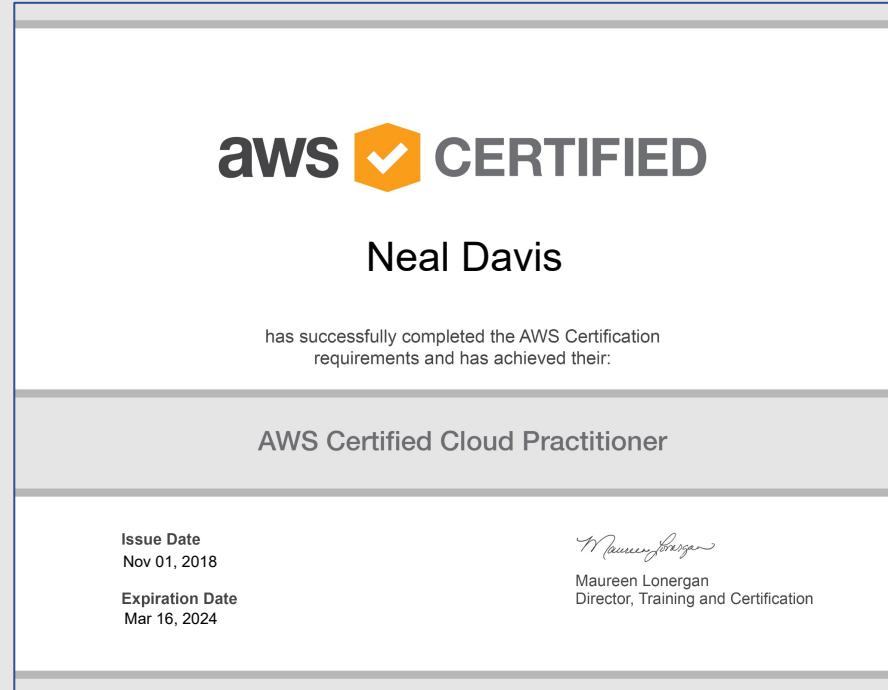


# SECTION 1

Let's Get Started!

# The CLF-C01 Exam





# The CLF-C01 Exam

---

---

	<b>Format</b> 65 questions; either multiple choice or multiple response
	<b>Type</b> Foundational
	<b>Delivery Method</b> Testing center or online proctored exam
	<b>Time</b> 90 minutes to complete the exam
	<b>Cost</b> 100 USD (Practice Exam: 20 USD)
	<b>Language</b> Available in English, Indonesian (Bahasa), Japanese, Korean, and Simplified Chinese



# The CLF-C01 Exam

---

---

## Recommended knowledge:

- At least 6 months experience with AWS Cloud in any role including technical, managerial, sales, purchasing or financial
- No experience necessary for this course!



# The CLF-C01 Exam

---

## Time and Length:

- 90 minutes
- 65 questions

## Scoring:

- Scaled score between 100 – 1000
- Minimum passing score of 700



# The CLF-C01 Exam

---

---

## Question format:

- Multiple-choice: Has one correct response and three incorrect responses
- Multiple-response: Has two or more correct responses out of five or more options



# The CLF-C01 Exam

---

---

## Domain 1: Cloud Concepts

- 1.1 Define the AWS Cloud and its value proposition
- 1.2 Identify aspects of AWS Cloud economics
- 1.3 List the different cloud architecture design principles

## Domain 2: Security

- 2.1 Define the AWS Shared Responsibility model
- 2.2 Define AWS Cloud security and compliance concepts
- 2.3 Identify AWS access management capabilities
- 2.4 Identify resources for security support



# The CLF-C01 Exam

---

---

## Domain 3: Technology

- 3.1 Define methods of deploying and operating in the AWS Cloud
- 3.2 Define the AWS global infrastructure
- 3.3 Identify the core AWS services
- 3.4 Identify resources for technology support

## Domain 4: Billing and Pricing

- 4.1 Compare and contrast the various pricing models for AWS
- 4.2 Recognize the various account structures in relation to AWS billing and pricing
- 4.3 Identify resources available for billing support



# The CLF-C01 Exam

---

---

Domain	% of Exam
Domain 1: Cloud Concepts	26%
Domain 2: Security and Compliance	25%
Domain 3: Technology	33%
Domain 4: Billing and Pricing	16%
<b>TOTAL</b>	<b>100%</b>

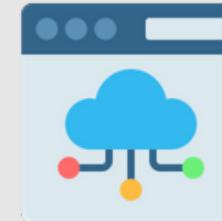
# SECTION 2

## Create AWS Free Tier Account

# SECTION 3

## Cloud Computing and AWS

# Traditional IT and Cloud Computing

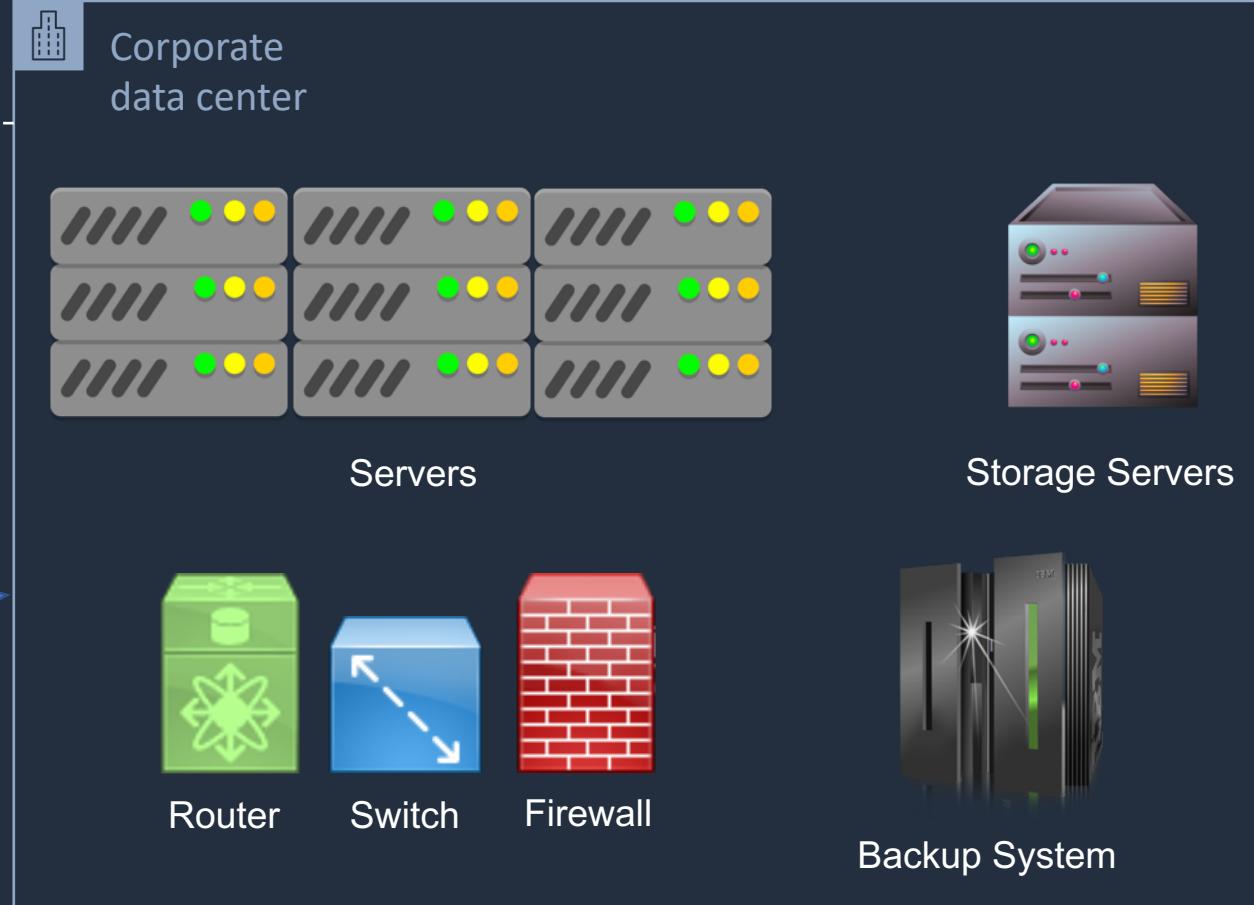




# Legacy IT / Traditional IT



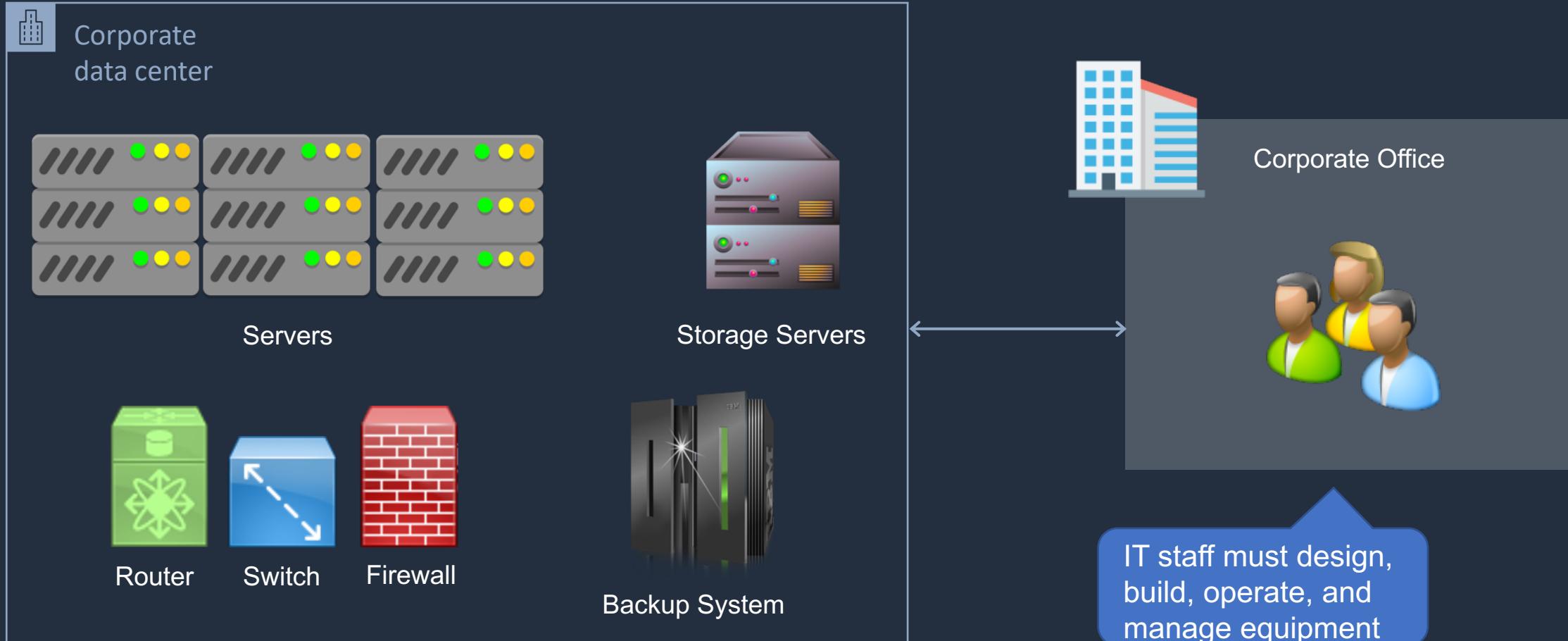
This model is very capital intensive



A company typically **leases** space in a data center, or may **own** the whole building

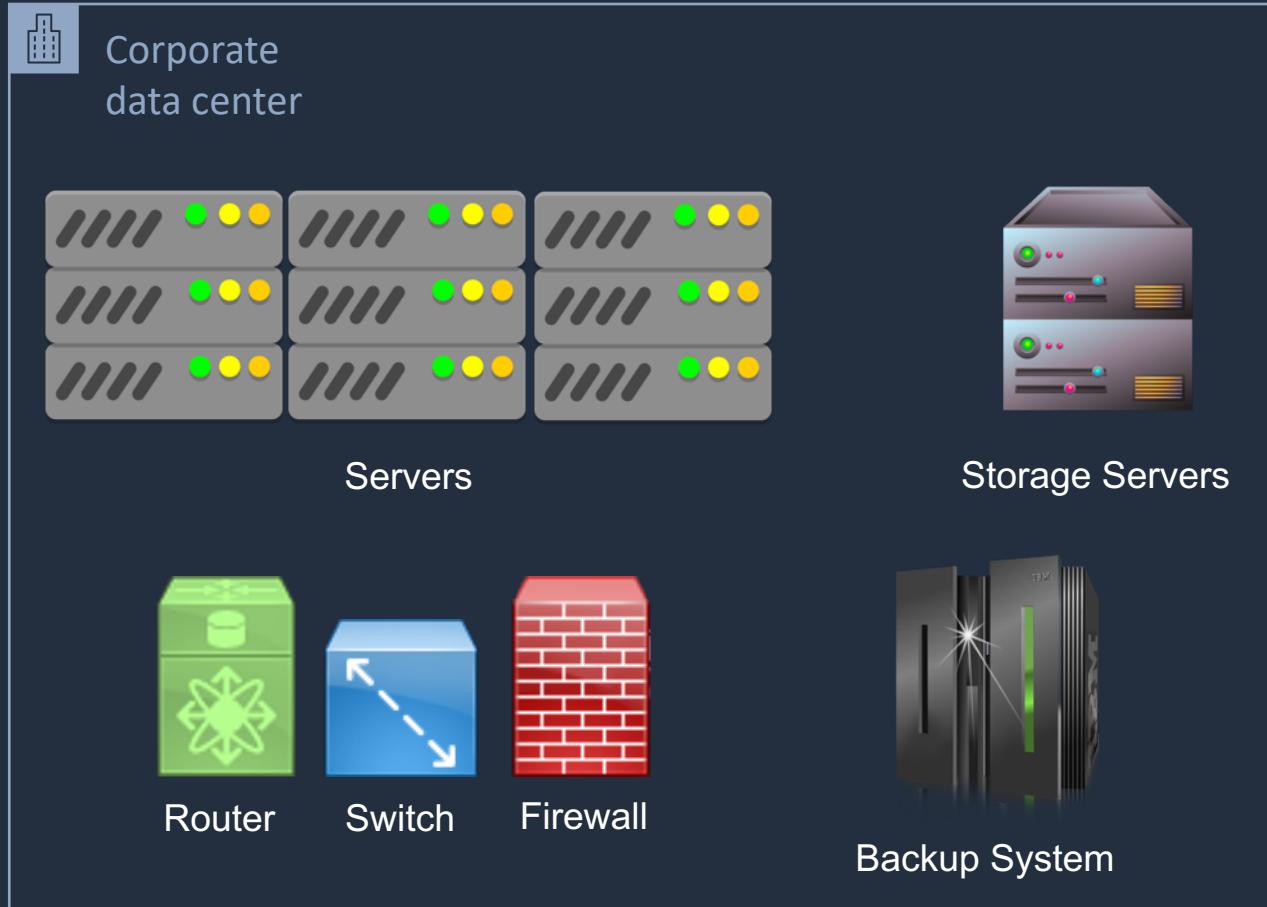


# Legacy IT / Traditional IT





# Legacy IT / Traditional IT



## Costs:

- Data center building
- Data center security
- Physical IT hardware
- Software licensing costs
- Maintenance contracts
- Power
- Internet connectivity
- Staff wages (design, build, operations, maintenance)



# Definition of Cloud Computing

On-demand,  
self-service

Broad network  
access

Resource  
pooling



Gmail



The Internet



Dropbox



Salesforce.com

Rapid  
elasticity

Measured  
service



Amazon Web Services



# Cloud vs Traditional IT

---

---

## Cloud Computing

On-demand, self-service

Broad network access

Resource pooling

Rapid elasticity

Measured service

## Traditional IT

Requires human involvement

Internal accessibility, limited public presence

Single-tenant, can be virtualized

Limited scalability

Usage is not typically measured

# Examples and Benefits of Cloud Computing





# Examples of Cloud Computing

## Non-Cloud Services:



Email Server



File Server



Customer Relationship  
Management (CRM)

## Cloud Services:



Gmail



Dropbox



Salesforce

You don't **own** or **manage** the infrastructure on which the service runs

Cloud services are offered on a **subscription / consumption** model

The service **scales** as **demand** changes



# Deploying a Website On-Premises

Assumes you don't have a private cloud, or don't have enough capacity

Activity:

Timeline:

1) Purchase hardware

4-12 weeks

2) Install and build

4-8 weeks

3) Acceptance testing

2-4 weeks

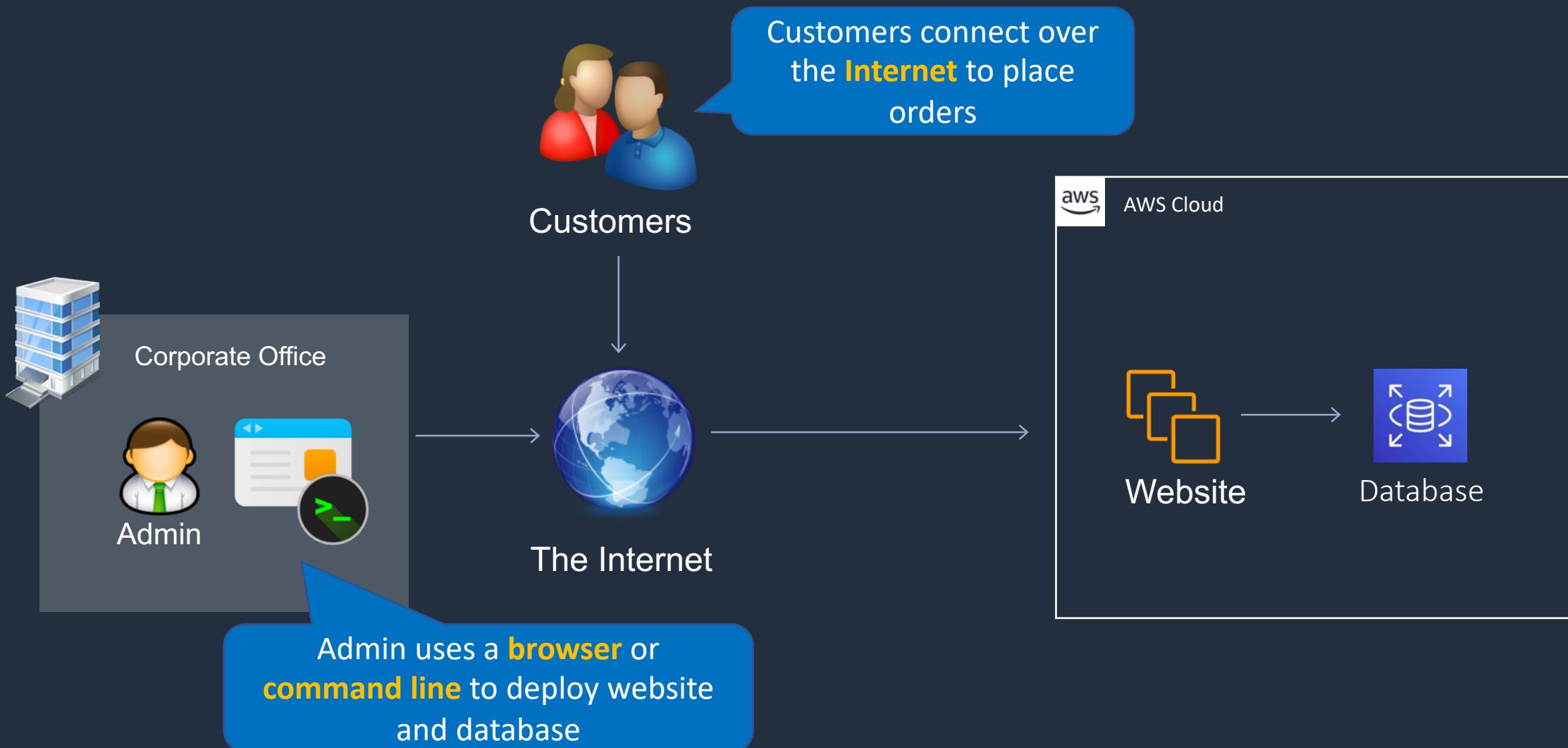
1) Handover to operations

1-2 weeks

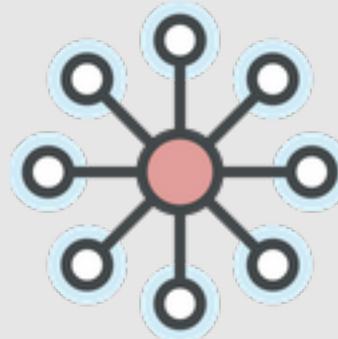
3-6 months !



# Deploying a Website in the Cloud

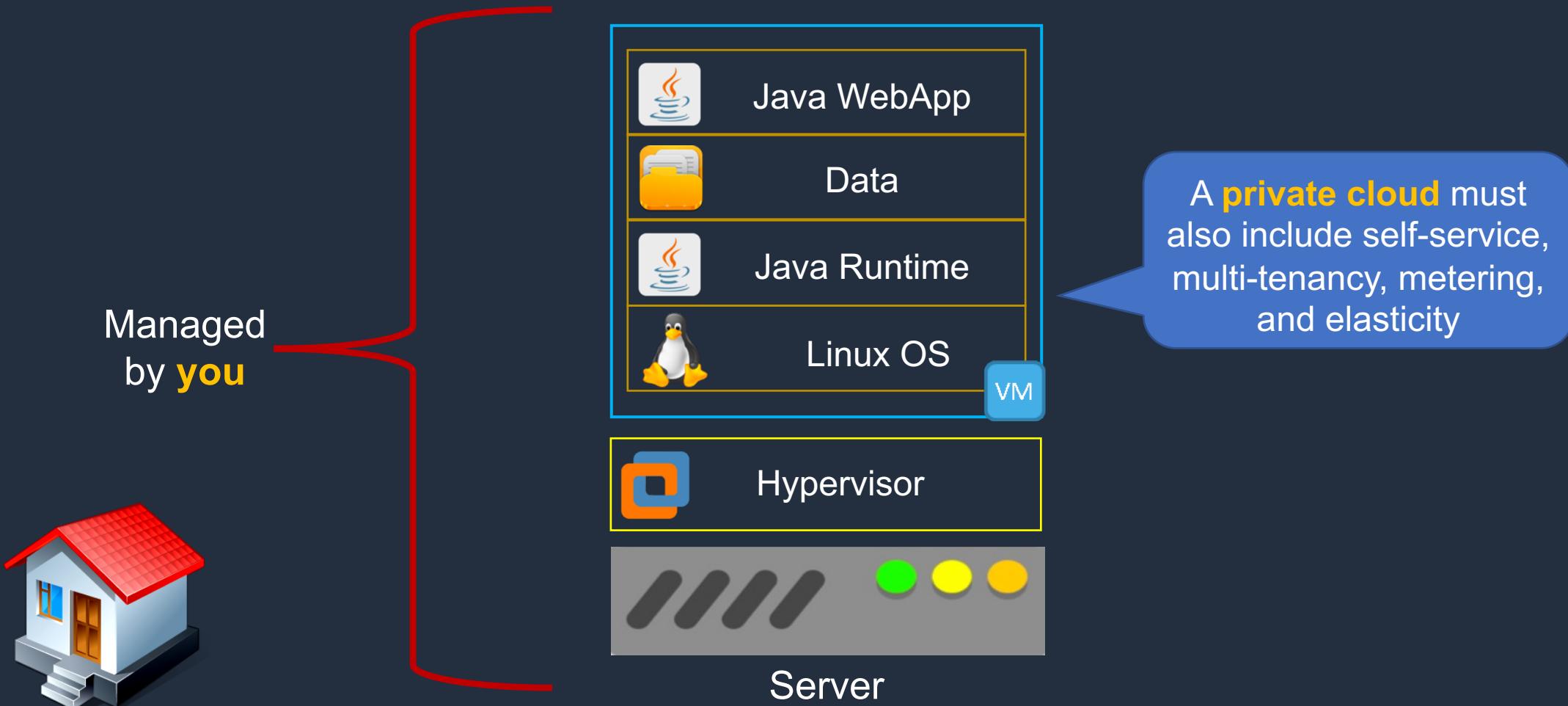


# Types of Cloud Service and Deployment





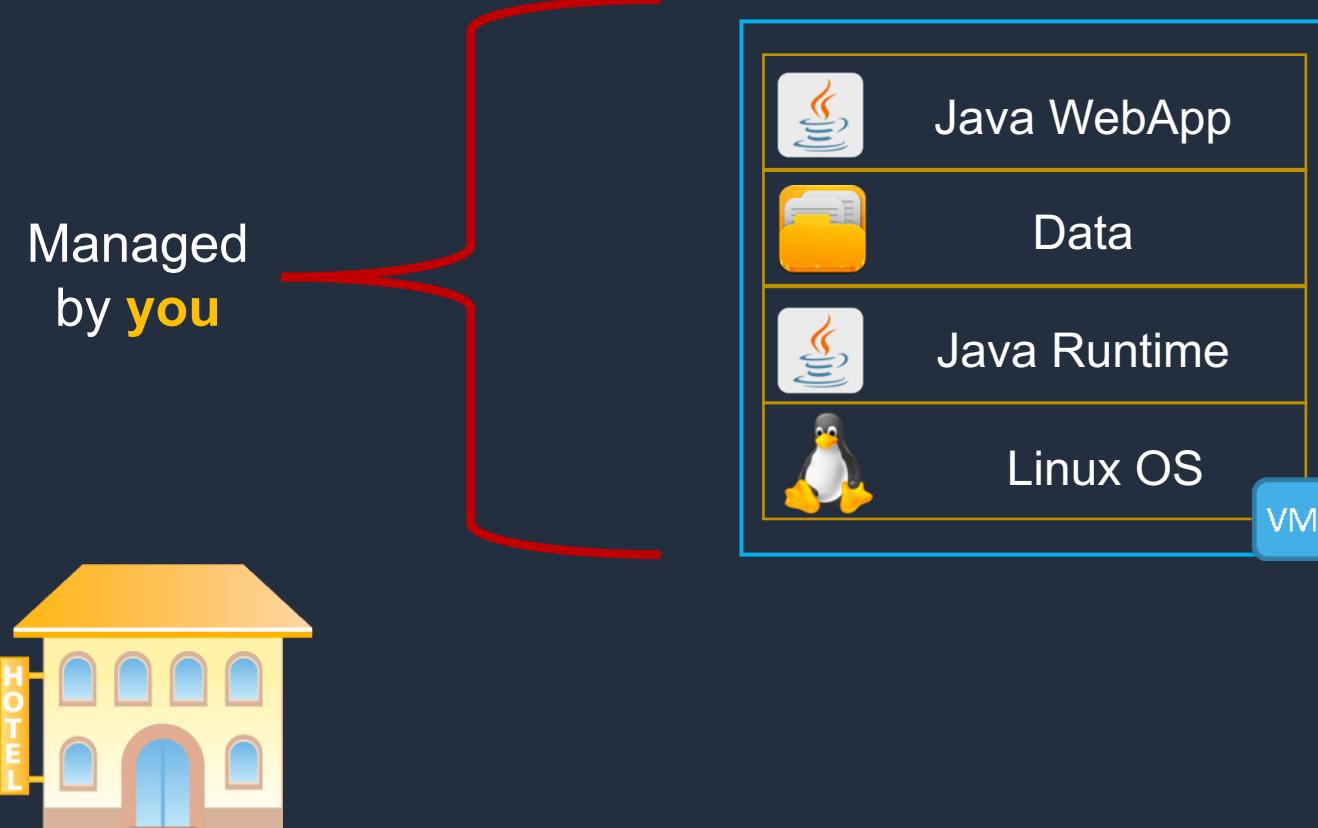
# Cloud Service Models: Private Cloud





# Cloud Service Models: Infrastructure as a Service (IaaS)

---



Examples:

- Amazon Elastic Compute Cloud (EC2)
- Azure Virtual Machines
- Google Compute Engine



# Cloud Service Models: Platform as a Service (PaaS)

---

---

Managed  
by **you**



Examples:

- AWS Elastic Beanstalk
- Azure WebApps
- Compute App Engine



# Cloud Service Models: Software as a Service (SaaS)

---

---

Managed  
by you



Java WebApp

Pure consumption  
model

Examples:

- Google Apps
- Salesforce.com
- Zoom



# Cloud Service Models: Comparison

## Private Cloud



## IaaS



## PaaS



## SaaS



You manage **everything**  
- greater responsibility +  
greater control

You manage from the  
**virtual server** upwards

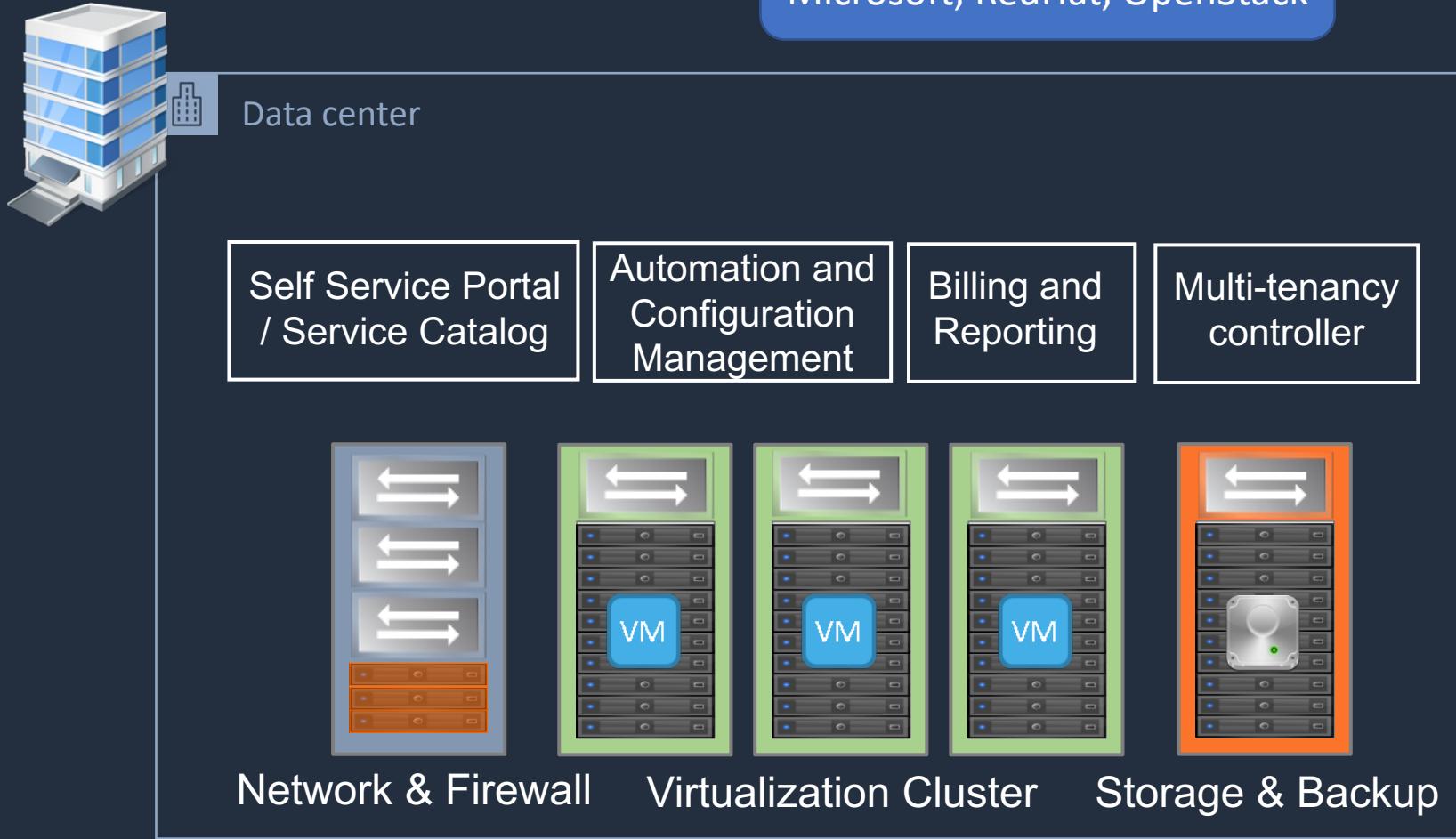
You simply upload your  
**code/data** to create your  
application

You simply **consume** the  
service - little  
responsibility + little  
control



# Private Cloud

Examples are VMware,  
Microsoft, RedHat, OpenStack



You **build** and **manage** the  
cloud deployment

- Benefits
- Complete control of the entire stack
  - Security – in a few cases, organizations may need to keep all or some of their applications and data in house

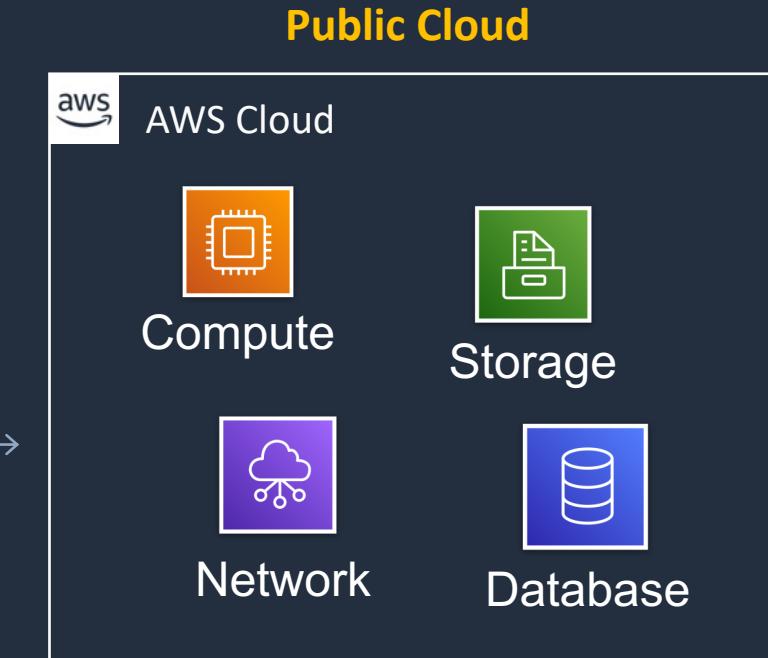


# Public Cloud

Examples are AWS, Microsoft Azure, Google Cloud Platform

## Benefits:

- Variable expense, instead of capital expense
- Economies of scale
- Massive elasticity



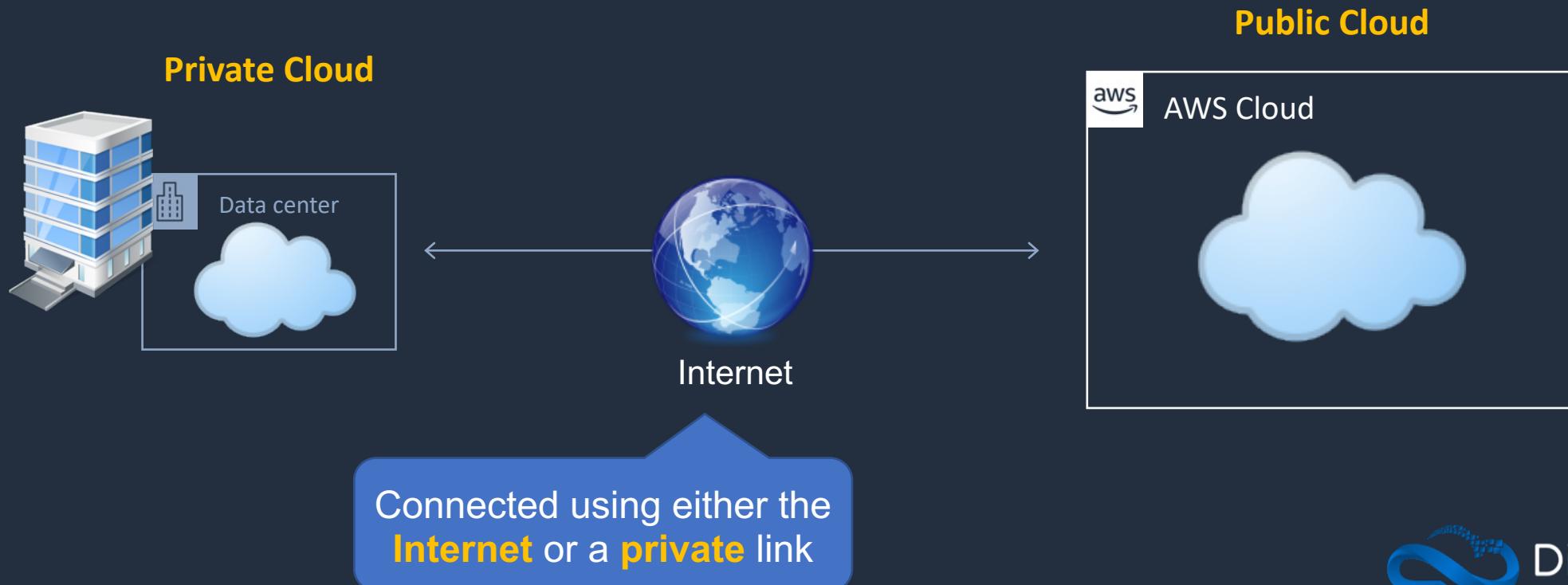
Connected using either the **Internet** or a **private link**



# Hybrid Cloud

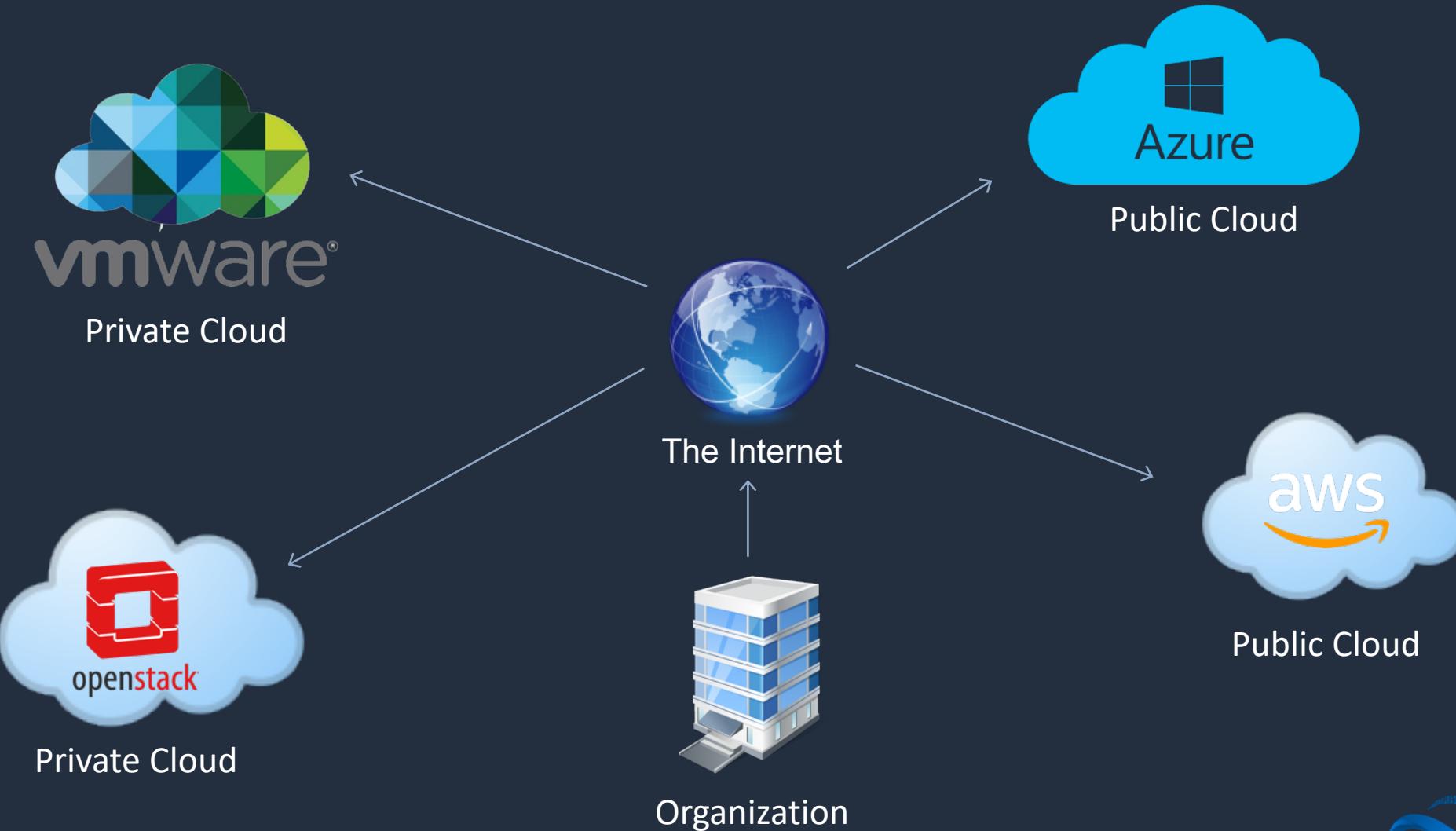
## Benefits:

- Allows companies to keep the critical applications and sensitive data in a traditional data center environment or private cloud
- Take advantage of public cloud resources like SaaS, for the latest applications, and IaaS, for elastic virtual resources
- Facilitates portability of data, apps and services and more choices for deployment models



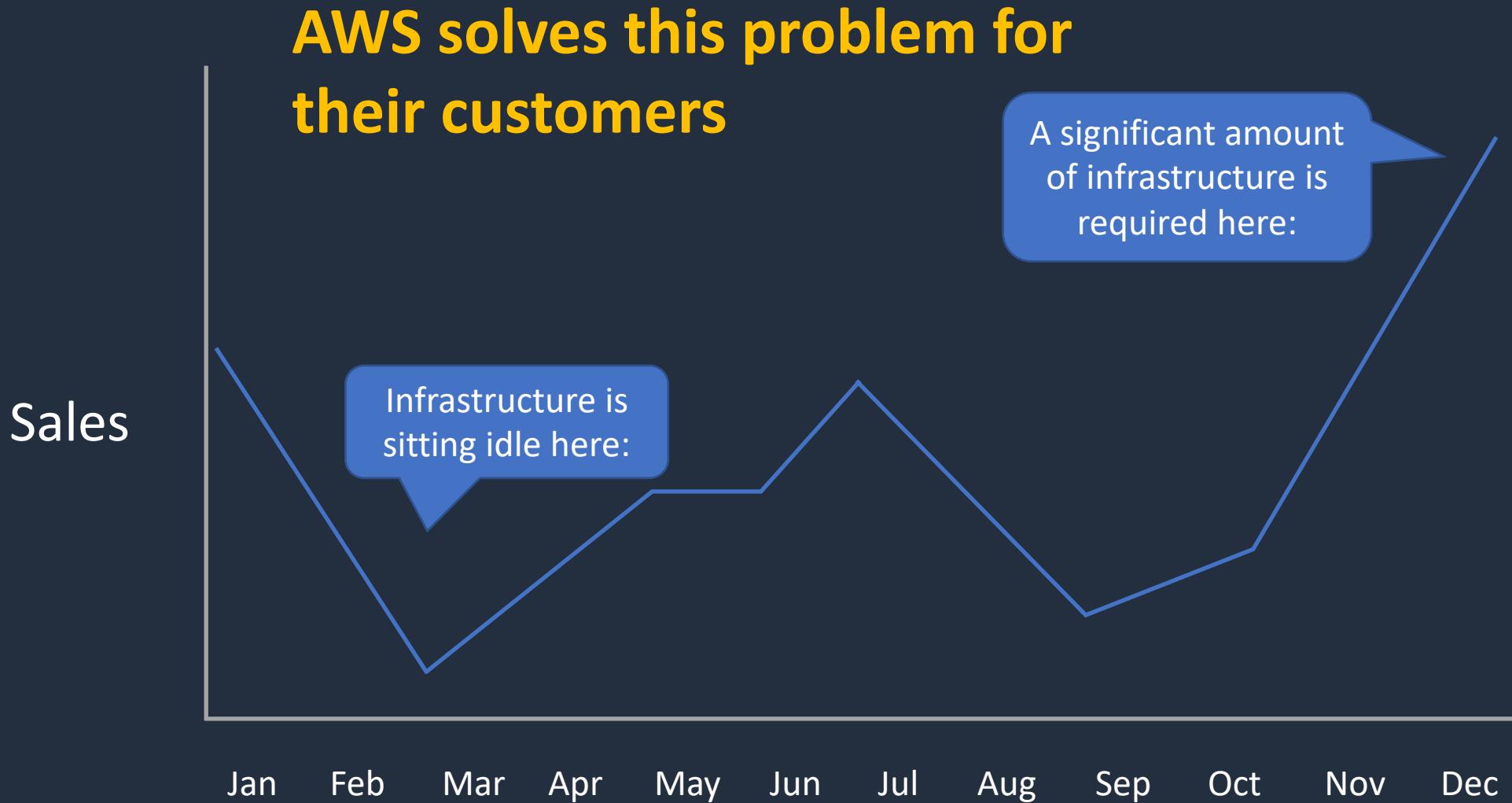


# Multicloud



# Overview of Amazon Web Services (AWS)





Charge for  
services based  
on usage

25 Regions  
around the  
world

Subsidiary  
of Amazon



Hyperscale  
Public Cloud  
Provider

Services are  
offered on-  
demand

# AWS Service Categories (a few examples)



Compute



Machine Learning



Storage



Database



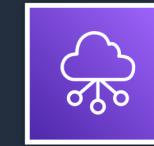
Analytics



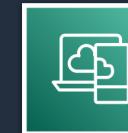
Many more categories  
and over **200** services!



Media Services



Networking



End User Computing



Internet of Things

# AWS Pricing Fundamentals

## Compute



Amount of resources such as CPU and RAM and duration

## Storage



Quantity of data stored

## Outbound Data Transfer



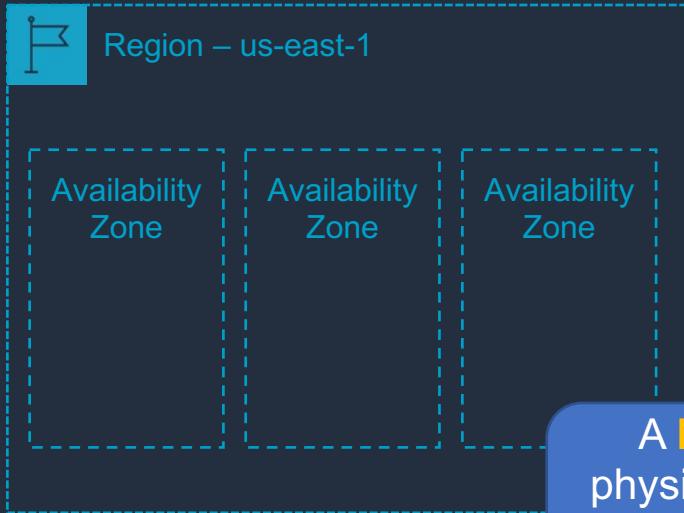
Quantity of data that is transferred out from all services

# The AWS Global Infrastructure





# AWS Global Infrastructure

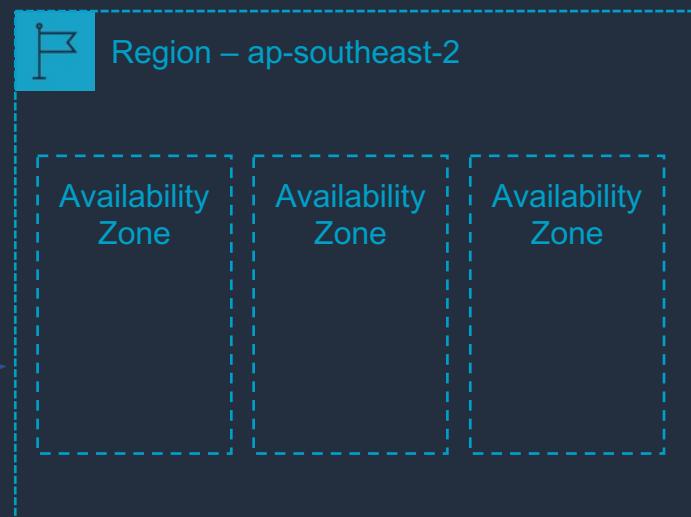


Every region is **connected** via a high bandwidth, fully redundant network

There are 25 **regions** around the world

Each region consists of **two or more** Availability Zones

An Availability Zone is composed of **one or more** data centers

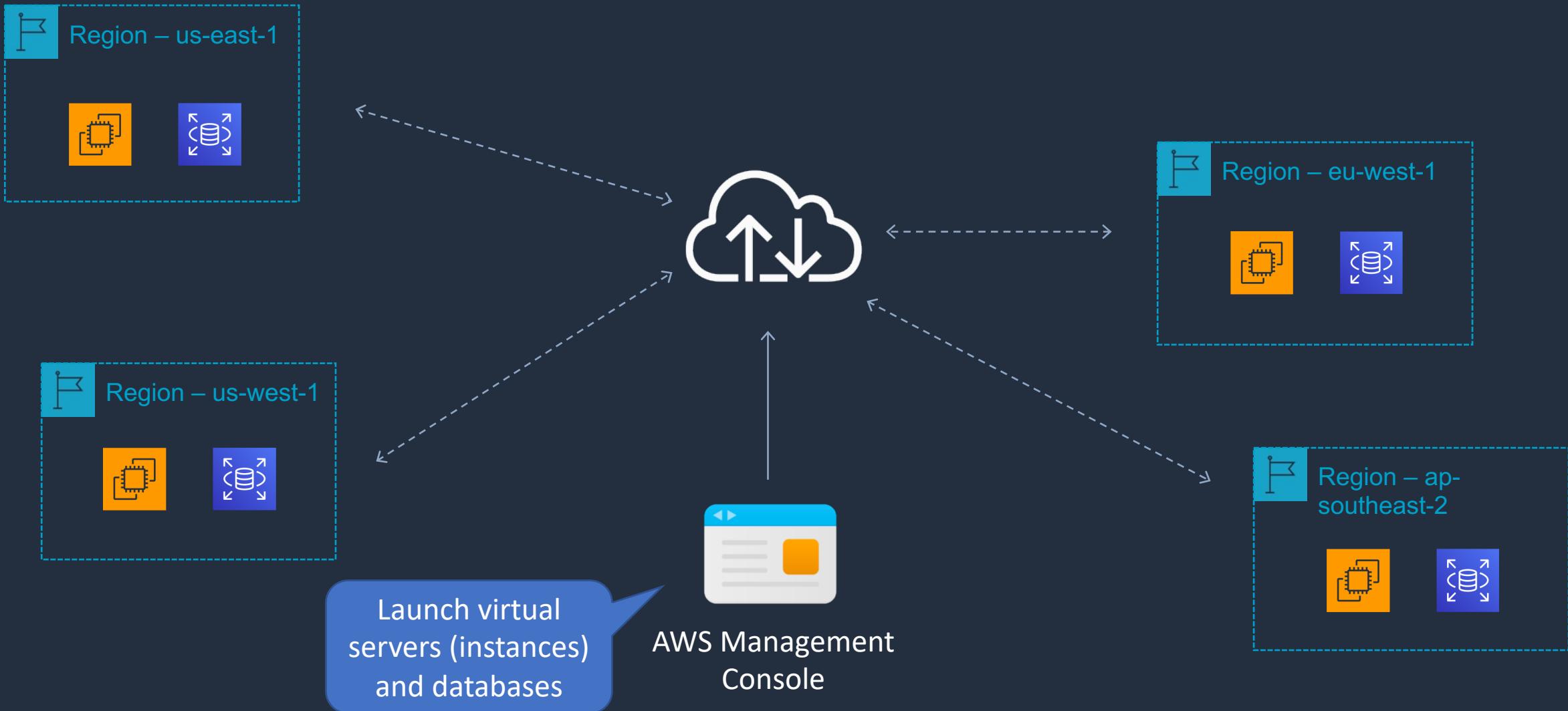


**Local Zones** extend regions closer to end-users





# Deploying Services Globally

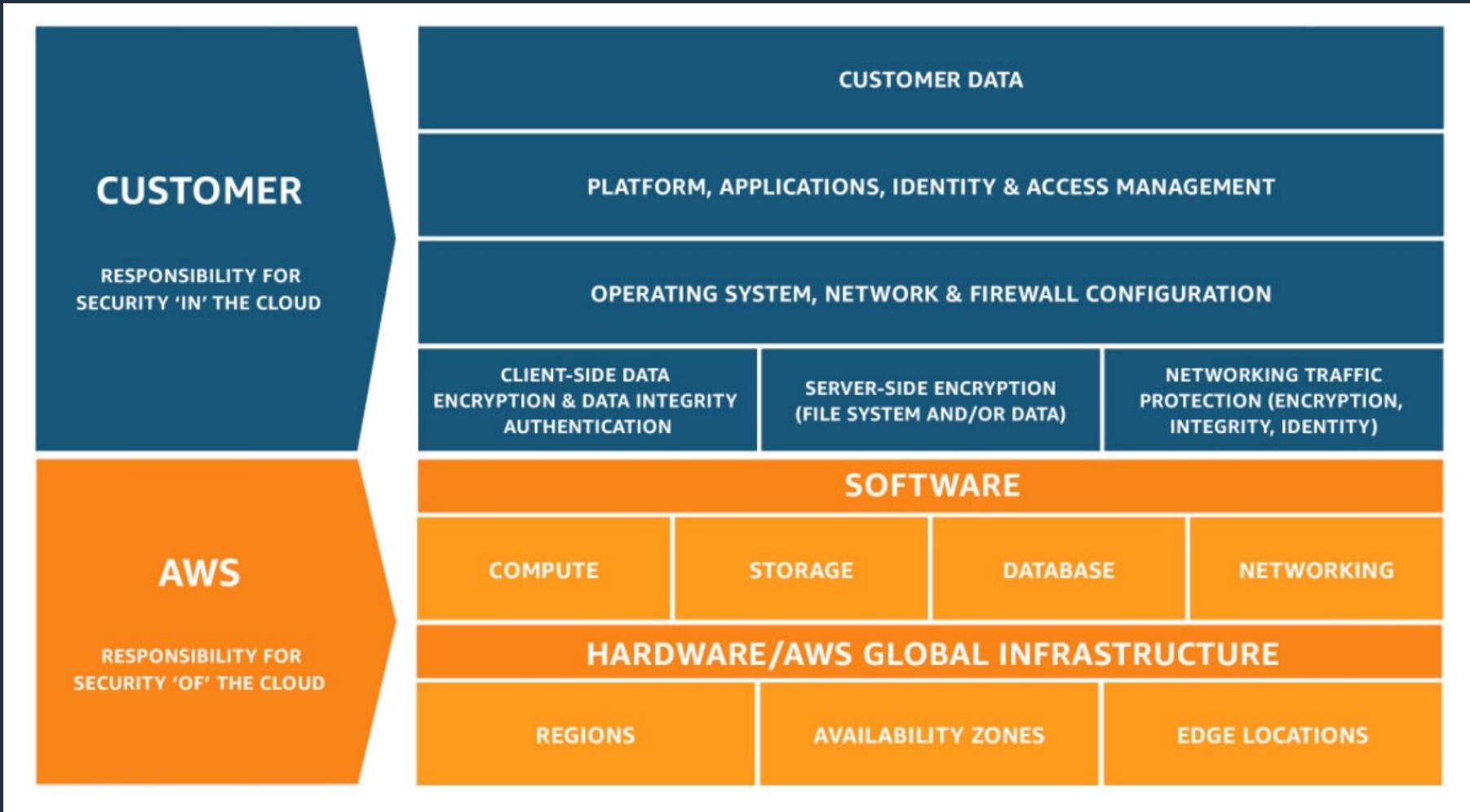


# The AWS Shared Responsibility Model





# The AWS Shared Responsibility Model





# The AWS Shared Responsibility Model

## CUSTOMER RESPONSIBILITY



Bucket with objects



Role



Multi-Factor Authentication



Security Group



Patch management



Staff training



Data encryption



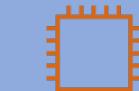
IAM User



Network ACL



SSL encryption



EC2 Instance



Auto Scaling



Elastic load balancer

## AWS RESPONSIBILITY



Data center security



Network router



Network switch



Server



Storage



Database Server

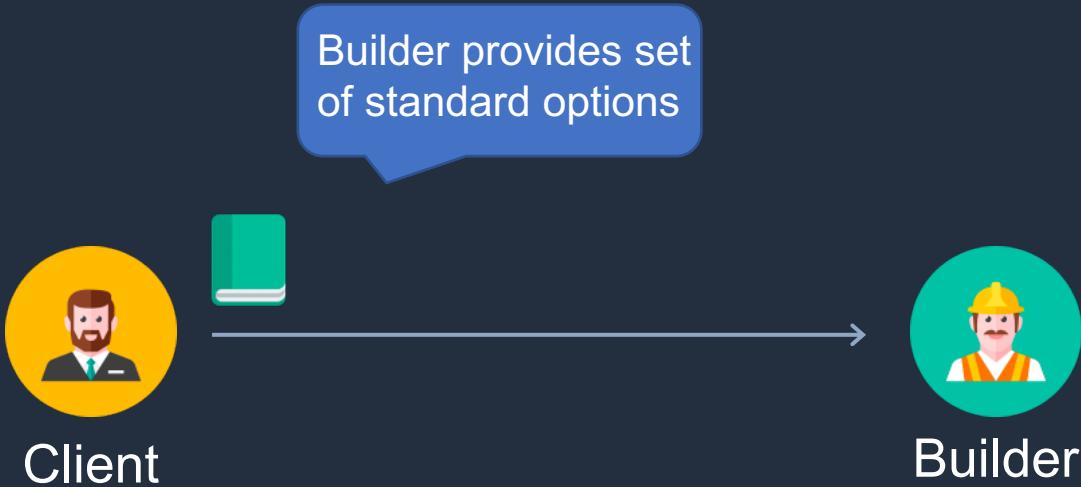


Disk drive

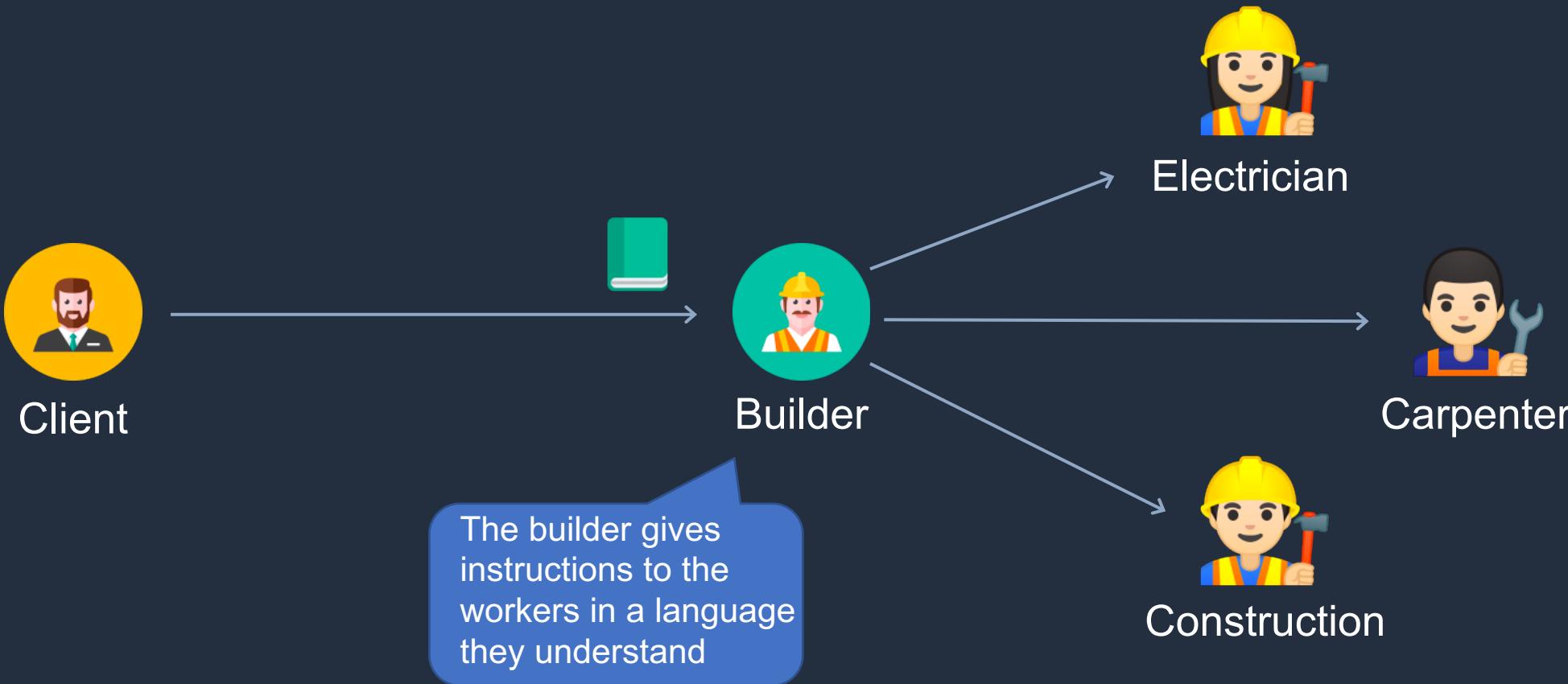
# Application Programming Interfaces (APIs)



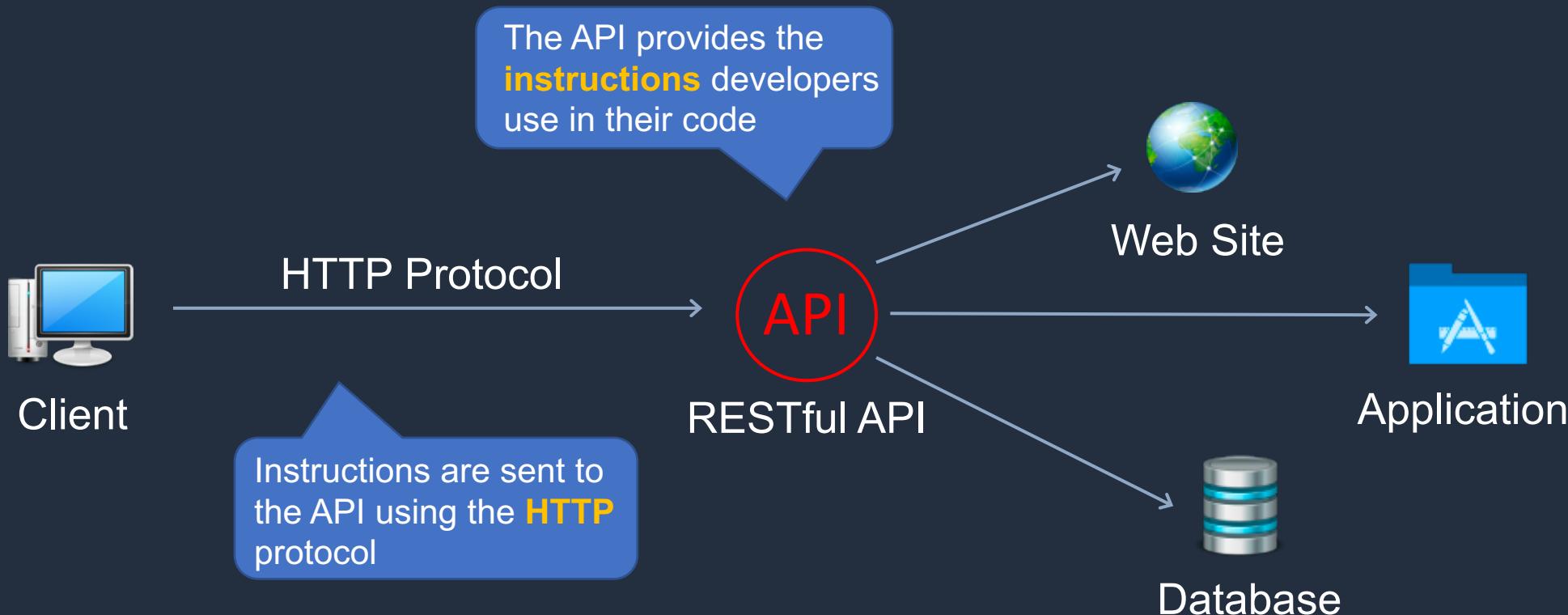
# Application Programming Interfaces (APIs) – Building a house analogy



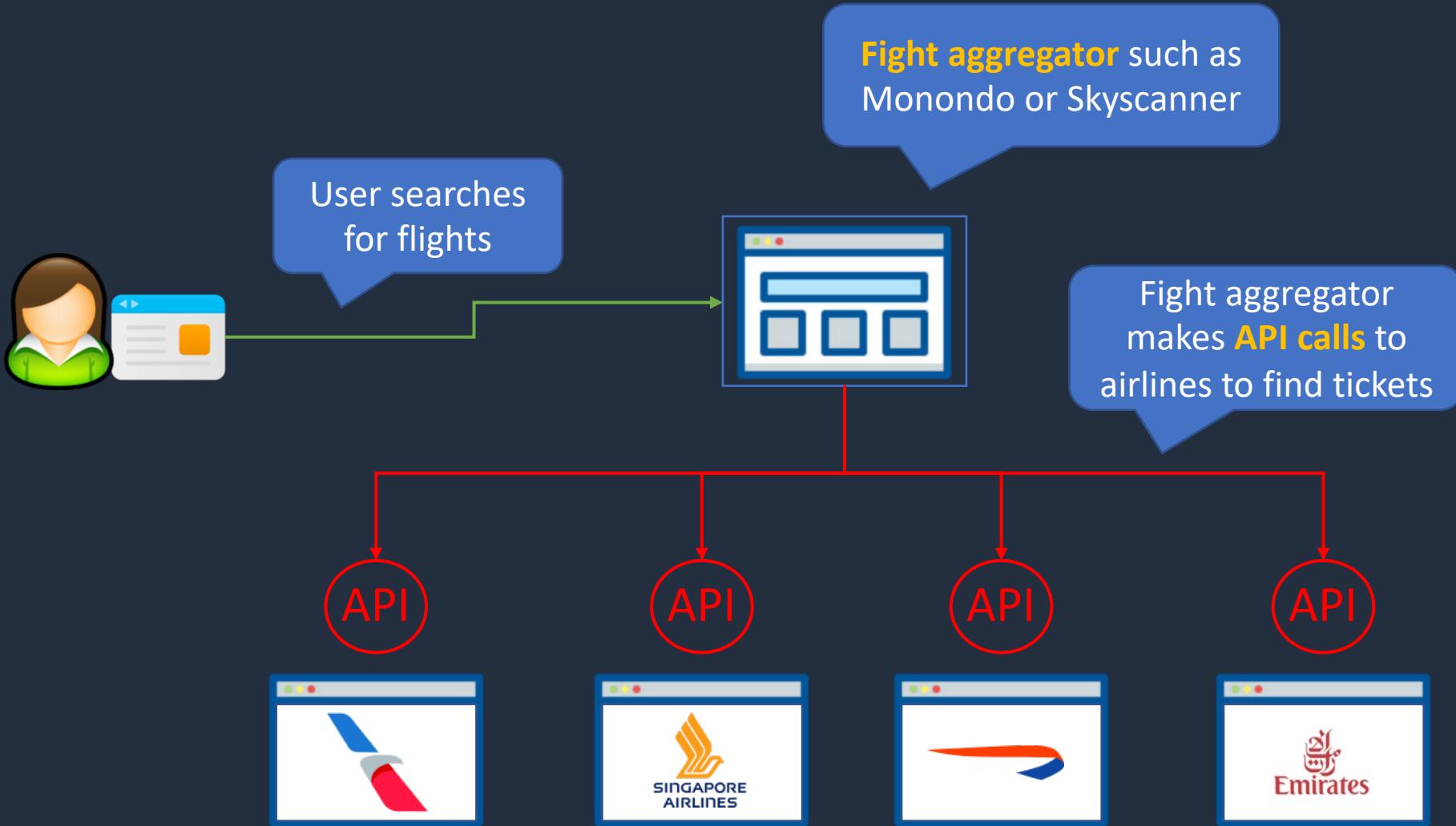
# Application Programming Interfaces (APIs) – Building a house analogy



# Application Programming Interfaces (APIs)



# Flight Aggregator Example



# Launching Cloud Services





# Launching Cloud Services: Management Console

## AWS Management Console

The screenshot shows the AWS Management Console navigation menu. It is organized into several sections:

- Compute**: EC2, Lightsail, Lambda, Batch, Elastic Beanstalk, Serverless Application Repository, AWS Outposts, EC2 Image Builder.
- Storage**: S3, EFS, FSx, S3 Glacier, Storage Gateway, AWS Backup.
- Database**: RDS, DynamoDB, ElastiCache, Neptune, Amazon Redshift, Amazon QLDB.
- Blockchain**: Amazon Managed Blockchain.
- Satellite**: Ground Station.
- Quantum Technologies**: Amazon Braket.
- Management & Governance**: AWS Organizations, CloudWatch, AWS Auto Scaling, CloudFormation, CloudTrail, Config, OpsWorks, Service Catalog, Systems Manager, AWS AppConfig, Trusted Advisor, Control Tower, AWS License Manager, AWS Well-Architected Tool.
- Analytics**: Athena, EMR, CloudSearch, Elasticsearch Service, Kinesis, QuickSight, Data Pipeline, AWS Data Exchange, AWS Glue, AWS Lake Formation, MSK.
- Business Applications**: Alexa for Business, Amazon Chime, WorkMail, Amazon Honeycode.
- End User Computing**: WorkSpaces, AppStream 2.0, WorkDocs, WorkLink.
- Security, Identity, & Compliance**: IAM, Resource Access Manager, Cognito, Secrets Manager, GuardDuty, Inspector, Amazon Macie, AWS Single Sign-On, Certificate Manager, Key Management Service, CloudHSM, Directory Service.
- Internet Of Things**: IoT Core, FreeRTOS, IoT 1-Click, IoT Analytics, IoT Device Defender, IoT Device Management, IoT Events, IoT Greengrass, IoT SiteWise, IoT Things Graph.

A web-based console accessed through a standard web browser



# Launching Cloud Services: Command Line

---

## Command Line

This command launches a virtual server (instance) on AWS



```
aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro
```



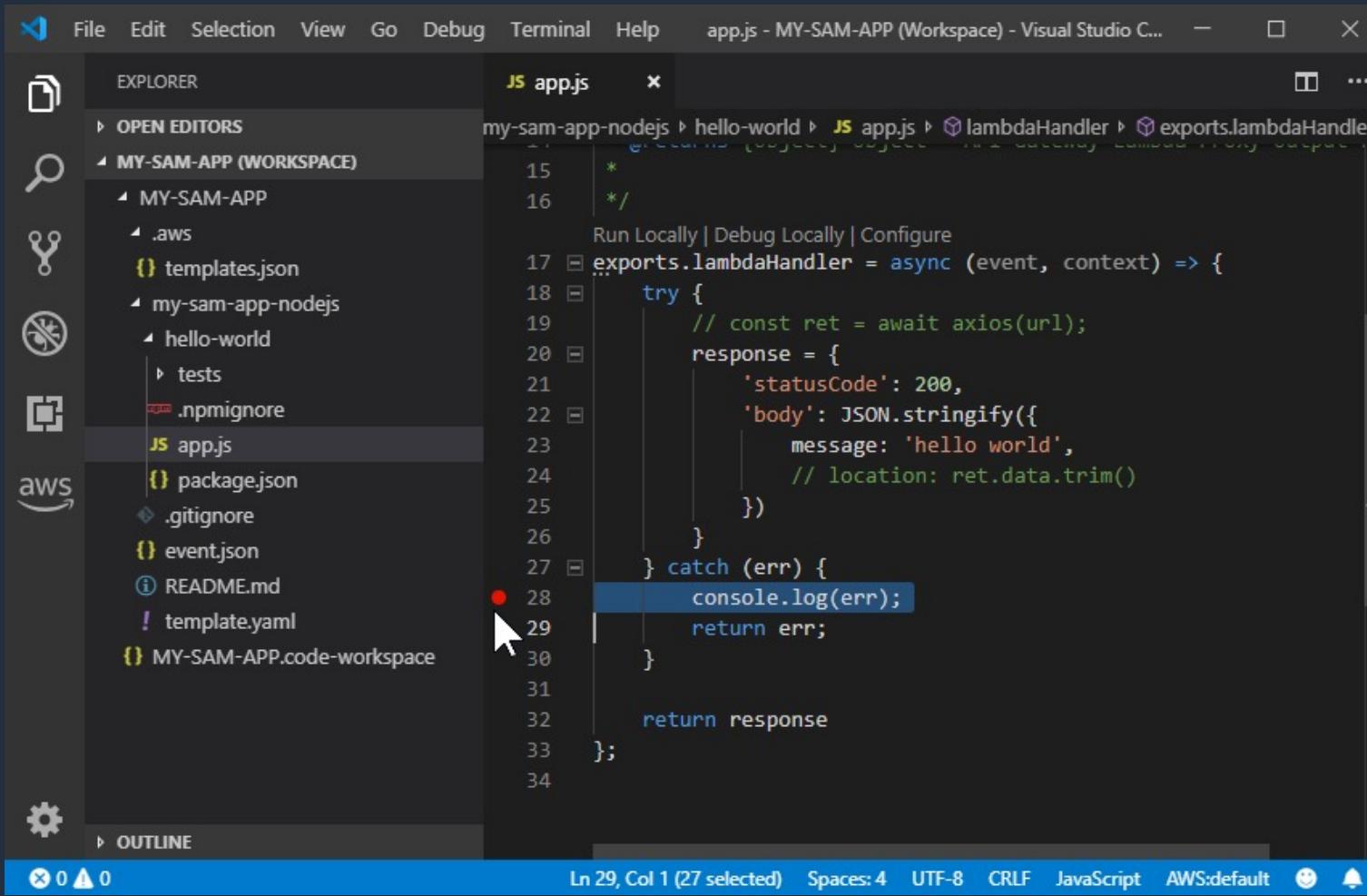
```
aws s3 ls s3://mys3databucket
```

This command lists the contents of a storage container (bucket) on Amazon S3



# Launching Cloud Services: Software Development Kit

A developer writes code in an **integrated development environment** (IDE)



The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows the project structure under "MY-SAM-APP (WORKSPACE)".
- Editor:** The file "app.js" is open, showing JavaScript code for a Lambda function. The code defines a "lambdaHandler" function that makes an axios request to "url" and returns a JSON response with "hello world". It also handles errors by logging them to the console.
- Status Bar:** Shows the current line (Ln 29, Col 1), character count (27 selected), and file encoding (UTF-8).

```
15 *  
16 */  
17 exports.lambdaHandler = async (event, context) => {  
18   try {  
19     const ret = await axios(url);  
20     response = {  
21       statusCode: 200,  
22       body: JSON.stringify({  
23         message: 'hello world',  
24         // location: ret.data.trim()  
25       })  
26     }  
27   } catch (err) {  
28     console.log(err);  
29     return err;  
30   }  
31  
32   return response  
33 };  
34 
```

The code leverages the **SDK** to work with cloud services

# Create your AWS Free Tier Account



# Configure Account and Create a Billing Alarm



# AWS CLI and CloudShell

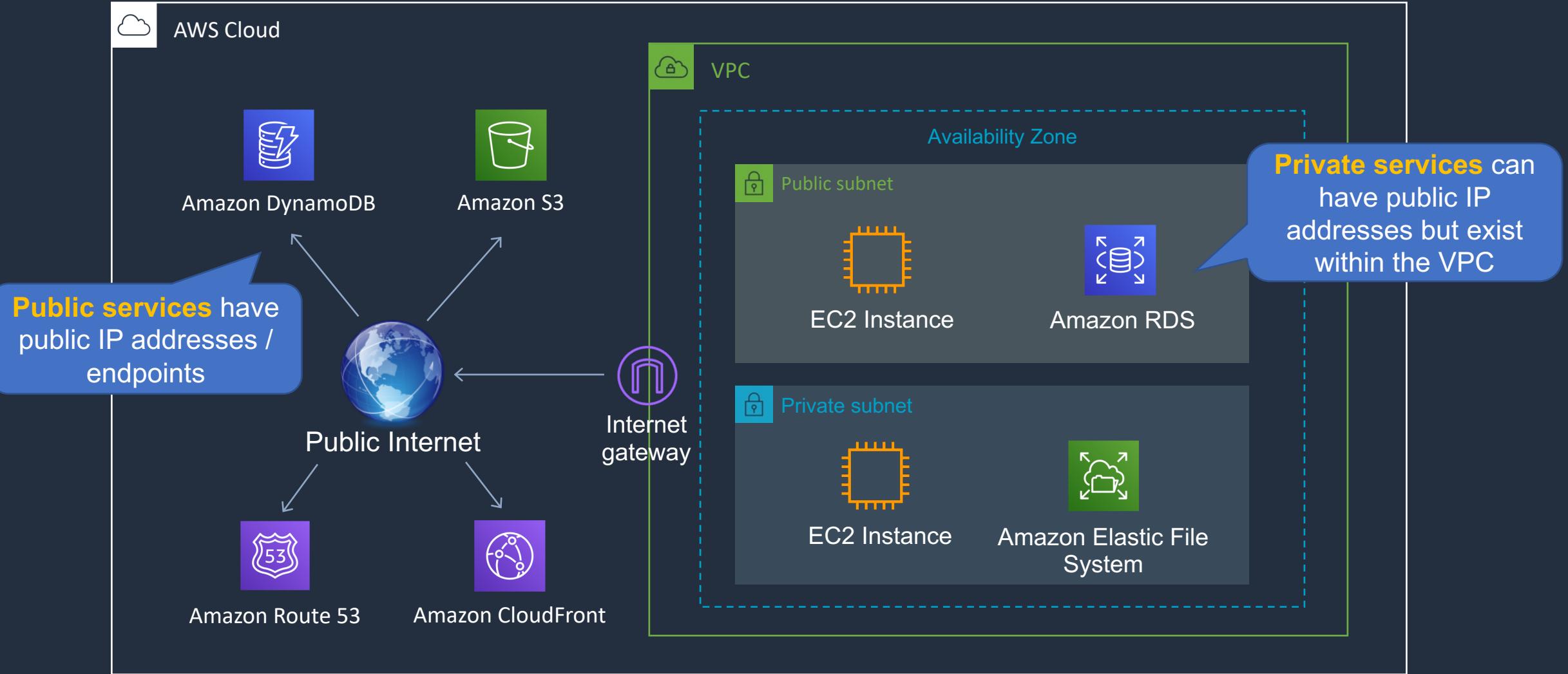


# AWS Public and Private Services





# AWS Public and Private Services



# The 6 Advantages of Cloud Computing





# The 6 Advantages of Cloud Computing

## 1. Trade capital expense for variable expense

CAPEX



Purchase servers



Tax deductible over depreciation lifetime

OPEX



Pay as you go



Tax deductible in same year



# The 6 Advantages of Cloud Computing

## 2. Benefit from massive economies of scale





# The 6 Advantages of Cloud Computing

---

---

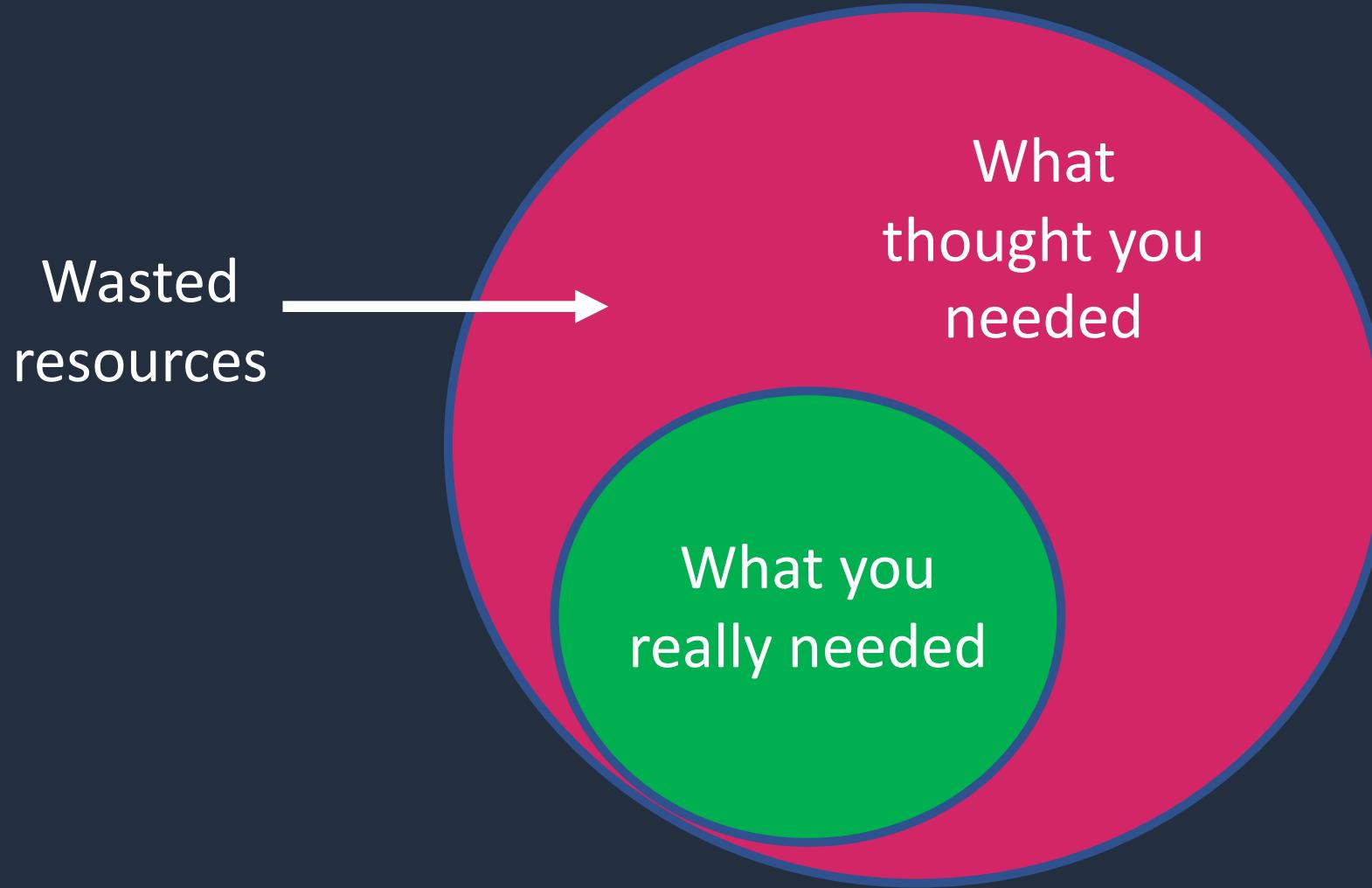
## 2. Benefit from massive economies of scale

- Aggregated usage across hundreds of thousands of customers = lower variable costs for customers



# The 6 Advantages of Cloud Computing

## 3. Stop guessing capacity





# The 6 Advantages of Cloud Computing

---

---

## 4. Increase speed and agility



Speed = deploy resources easily and quickly

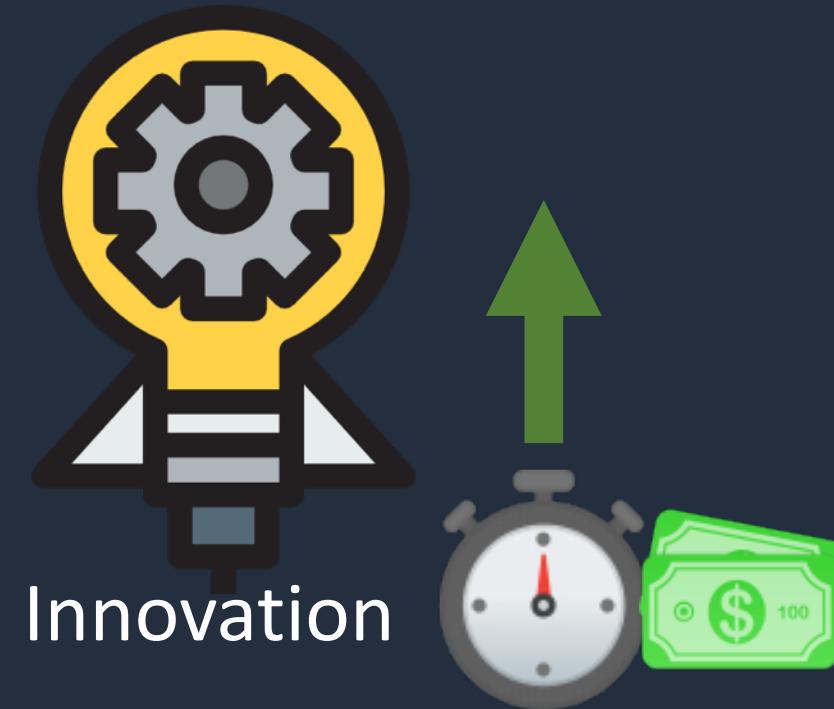


Agility = react to change ; speed to market



# The 6 Advantages of Cloud Computing

5. Stop spending money running and maintaining data centers





# The 6 Advantages of Cloud Computing

## 6. Go global in minutes



# SECTION 4

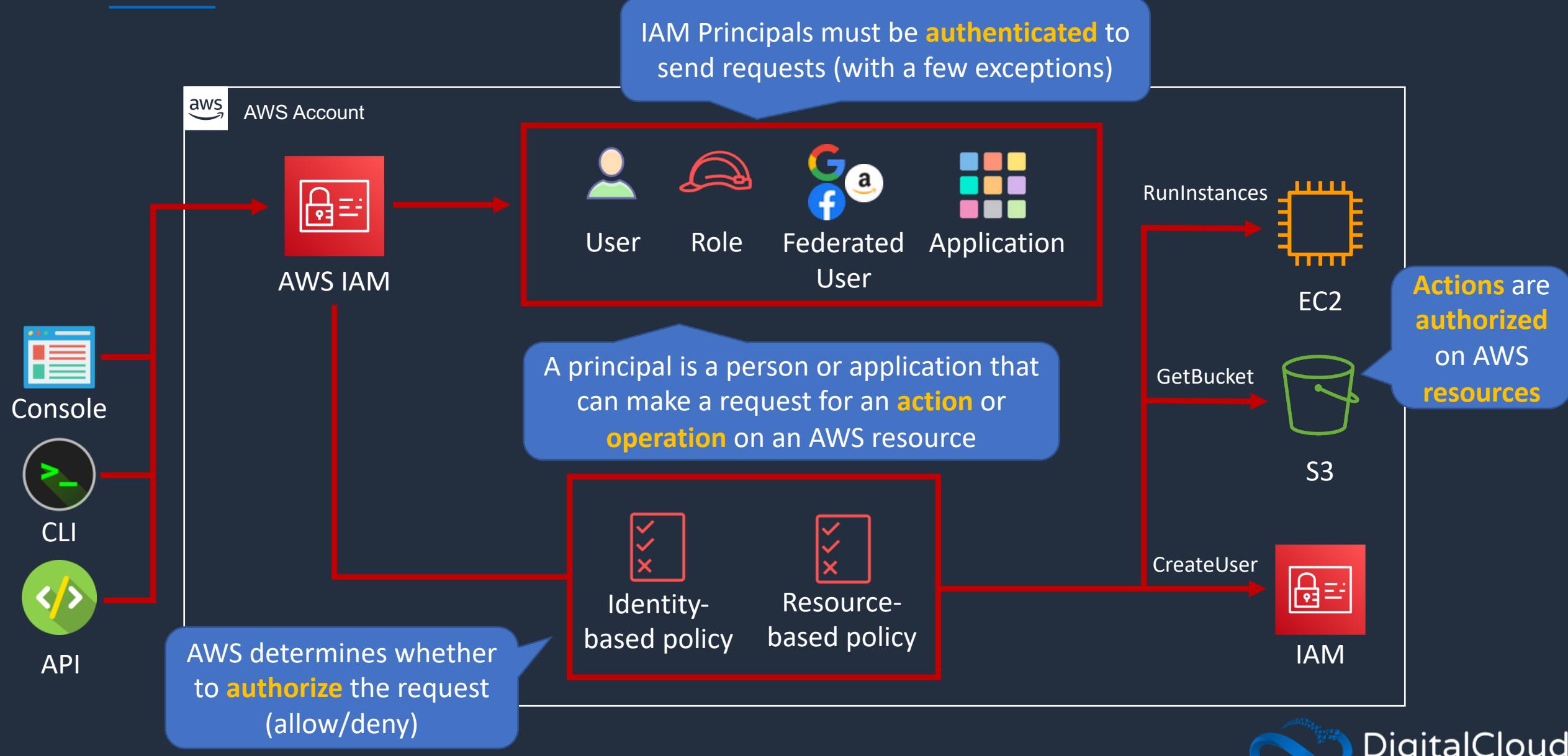
## Identity and Access Management (AWS IAM)

# AWS IAM Overview





# AWS Identity and Access Management (IAM)

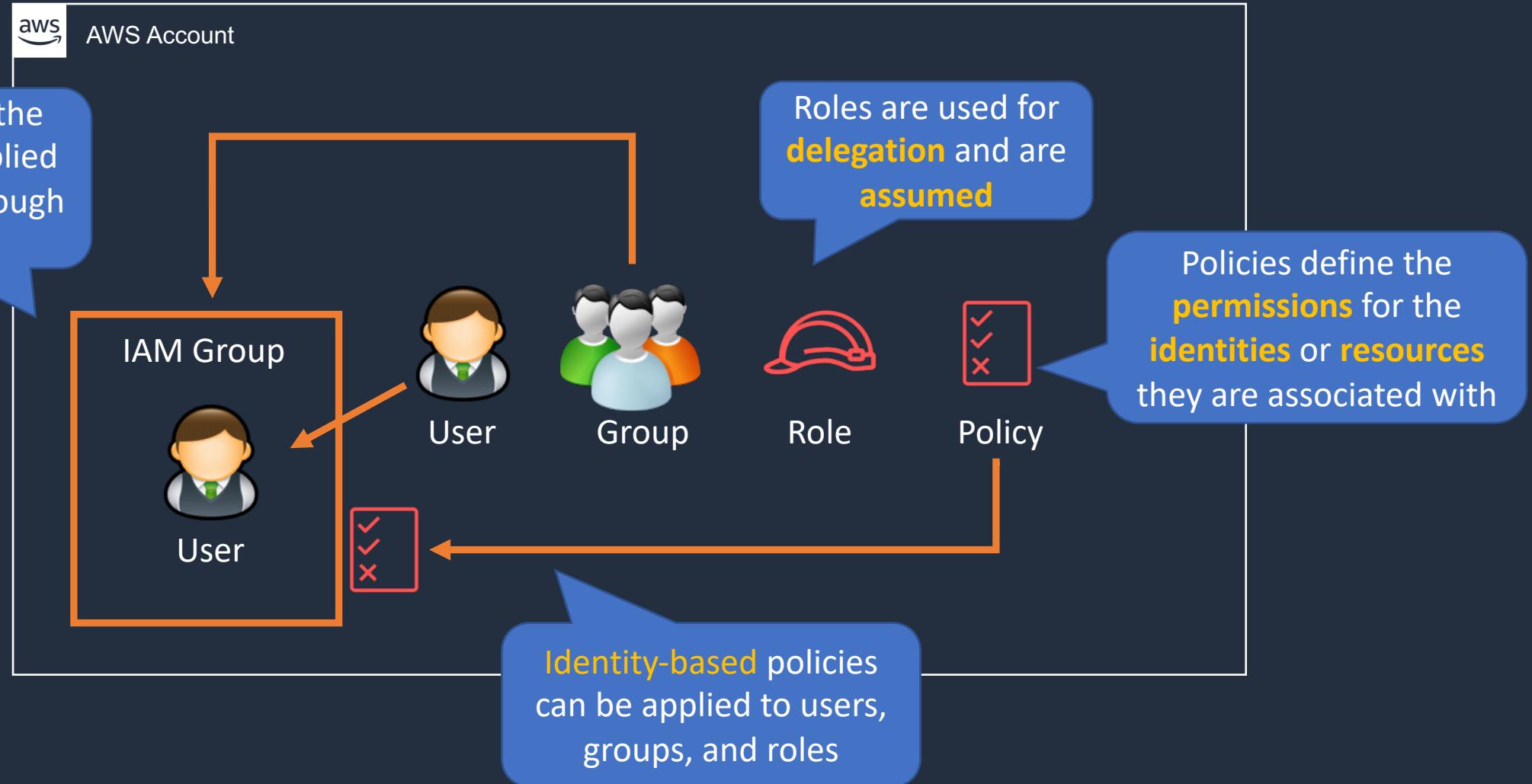


# IAM Users, Groups, Roles, and Policies



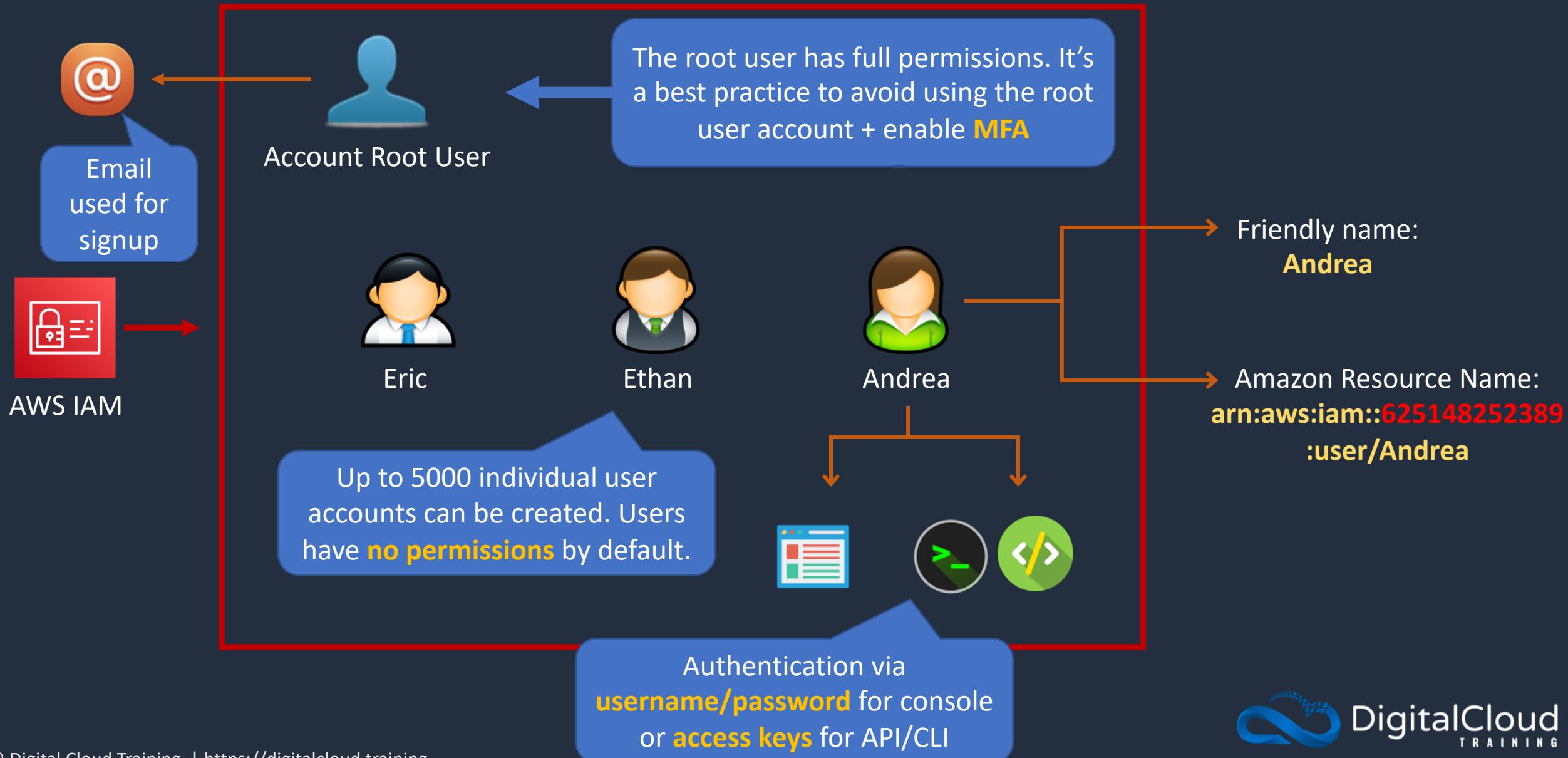


# Users, Groups, Roles and Policies



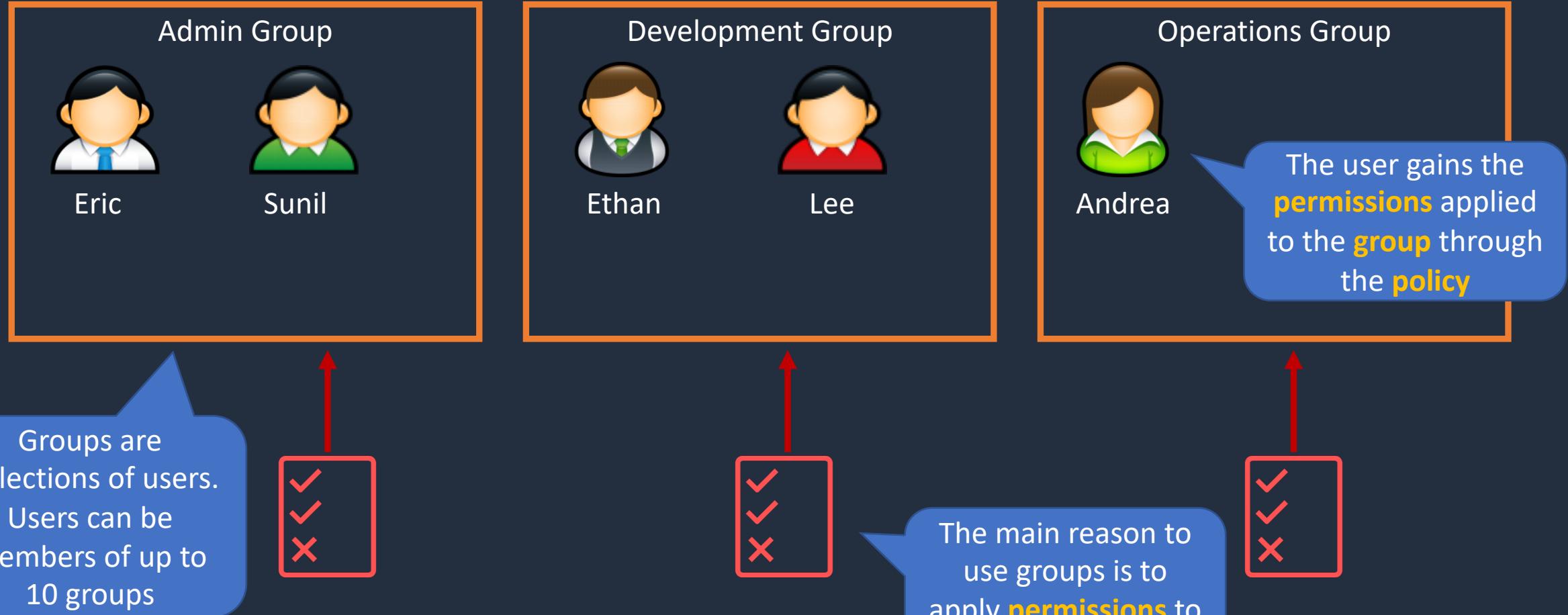


# IAM Users





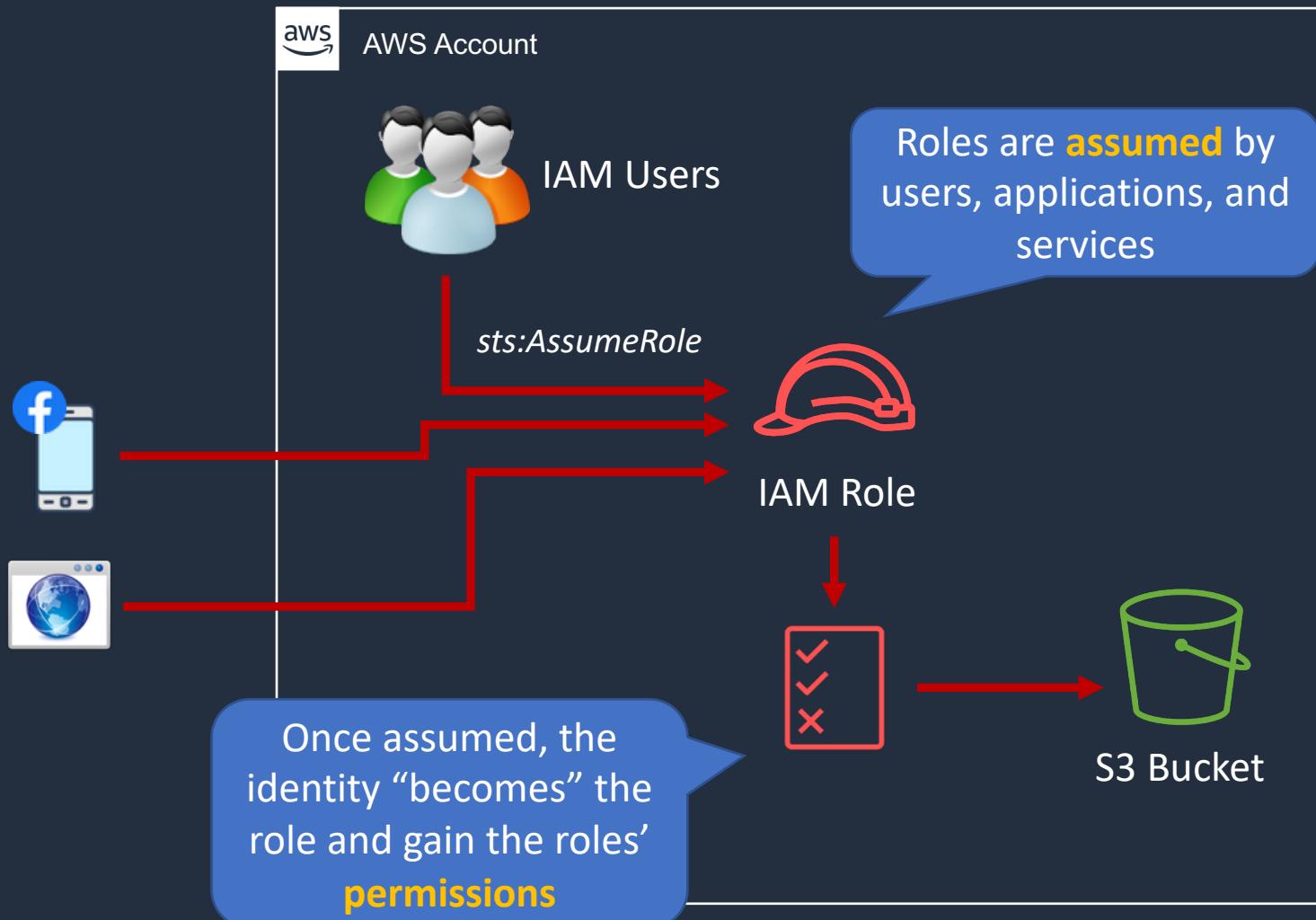
# IAM Groups





# IAM Roles

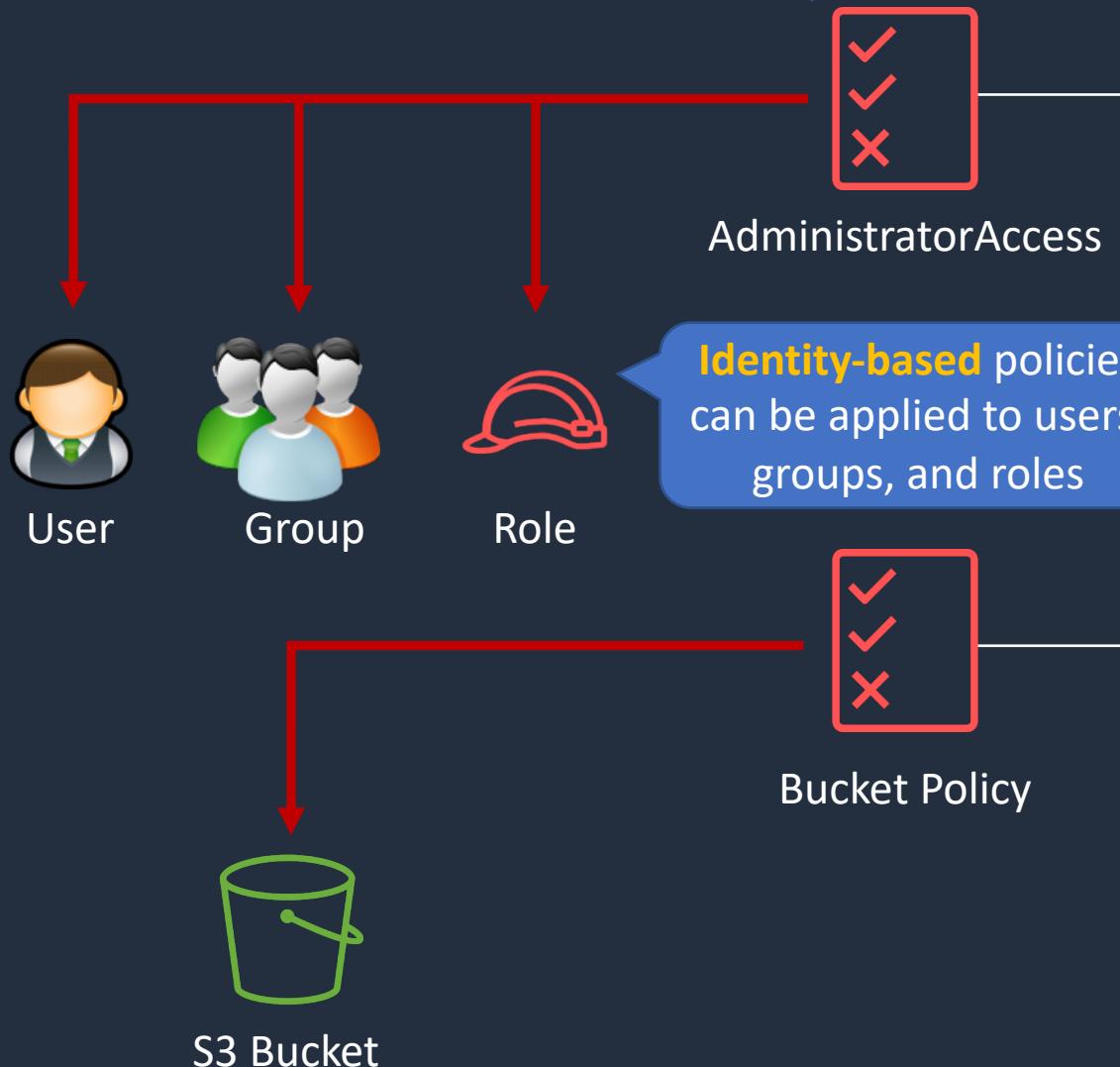
An **IAM role** is an IAM **identity** that has specific **permissions**





# IAM Policies

Policies are **documents** that define **permissions** and are written in **JSON**



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

All permissions are **implicitly denied** by default

```
{  
  "Version": "2012-10-17",  
  "Id": "Policy1561964929358",  
  "Statement": [  
    {  
      "Sid": "Stmt1561964454052",  
      "Effect": "Allow",  
      "Principal": {  
        "AWS": "arn:aws:iam::515148227241:user/Paul"  
      },  
      "Action": "s3:*",  
      "Resource": "arn:aws:s3:::dctcompany",  
      "Condition": {  
        "StringLike": {  
          "s3:prefix": "Confidential/*"  
        }  
      }  
    }  
  ]  
}
```

Resource-based policies apply to **resources** such as S3 buckets or DynamoDB tables

# Setup Individual User Account

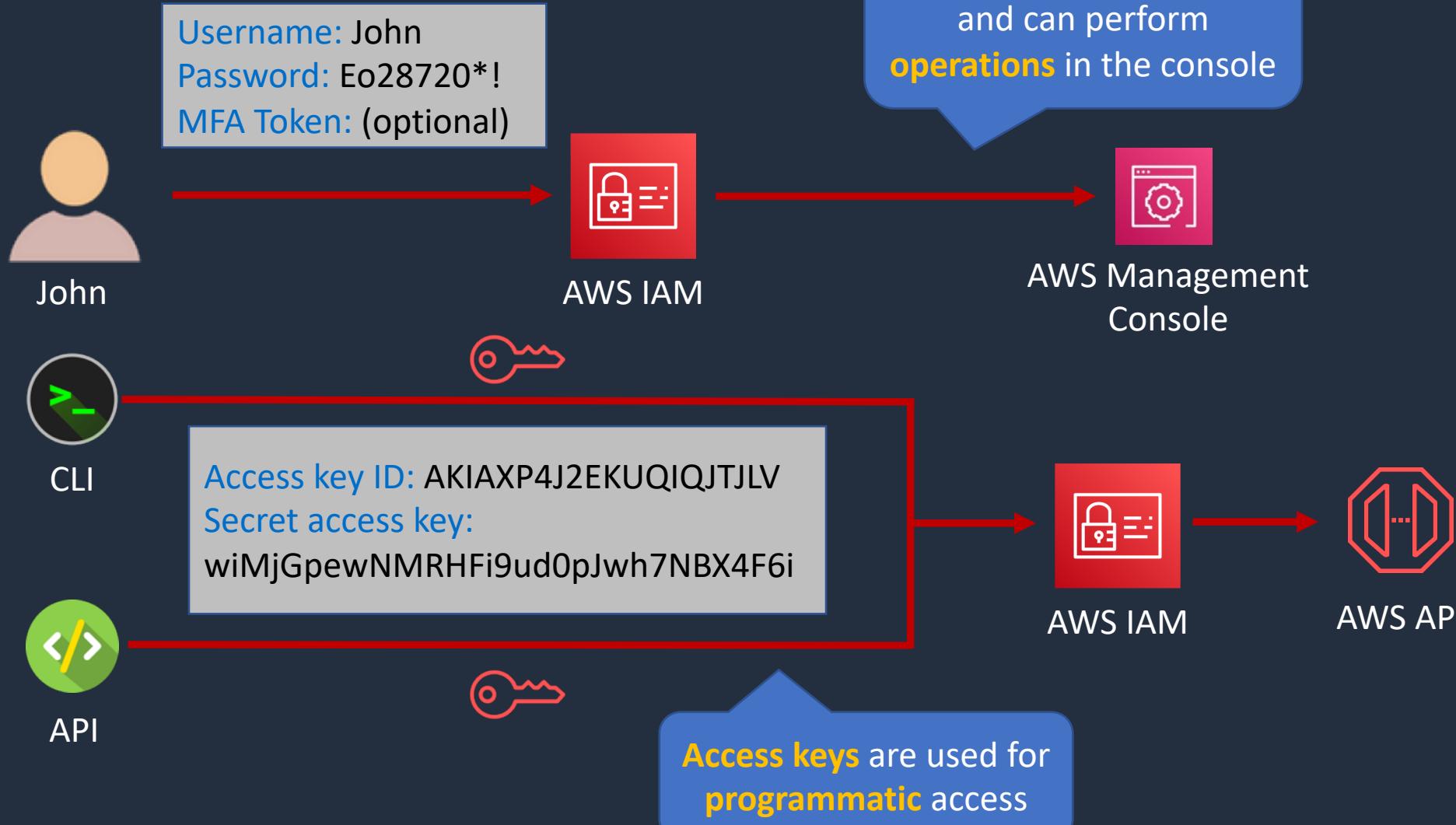


# IAM Authentication and MFA





# IAM Authentication Methods





# Multi-Factor Authentication

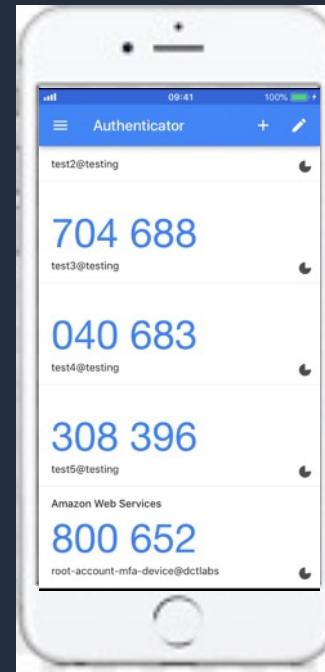
---

Something you **know**:

EJPx!\*21p9%

Password

Something you **have**:



Something you **are**:





# Multi-Factor Authentication

Something you **know**:



IAM User

EJPx!\*21p9%

Password

Something you **have**:



Virtual MFA

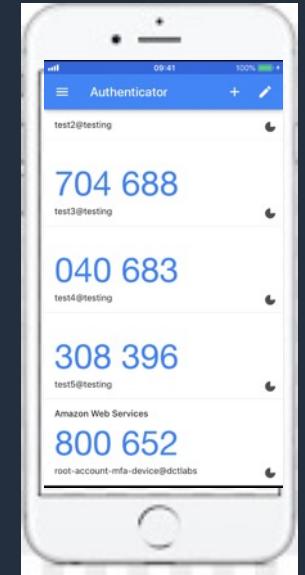


Physical MFA

e.g. Google Authenticator on  
your smart phone



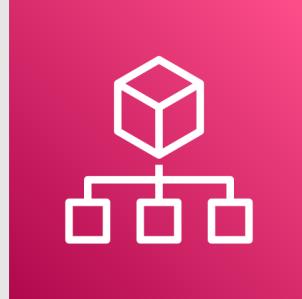
Physical tokens can  
be purchased from  
**third parties**



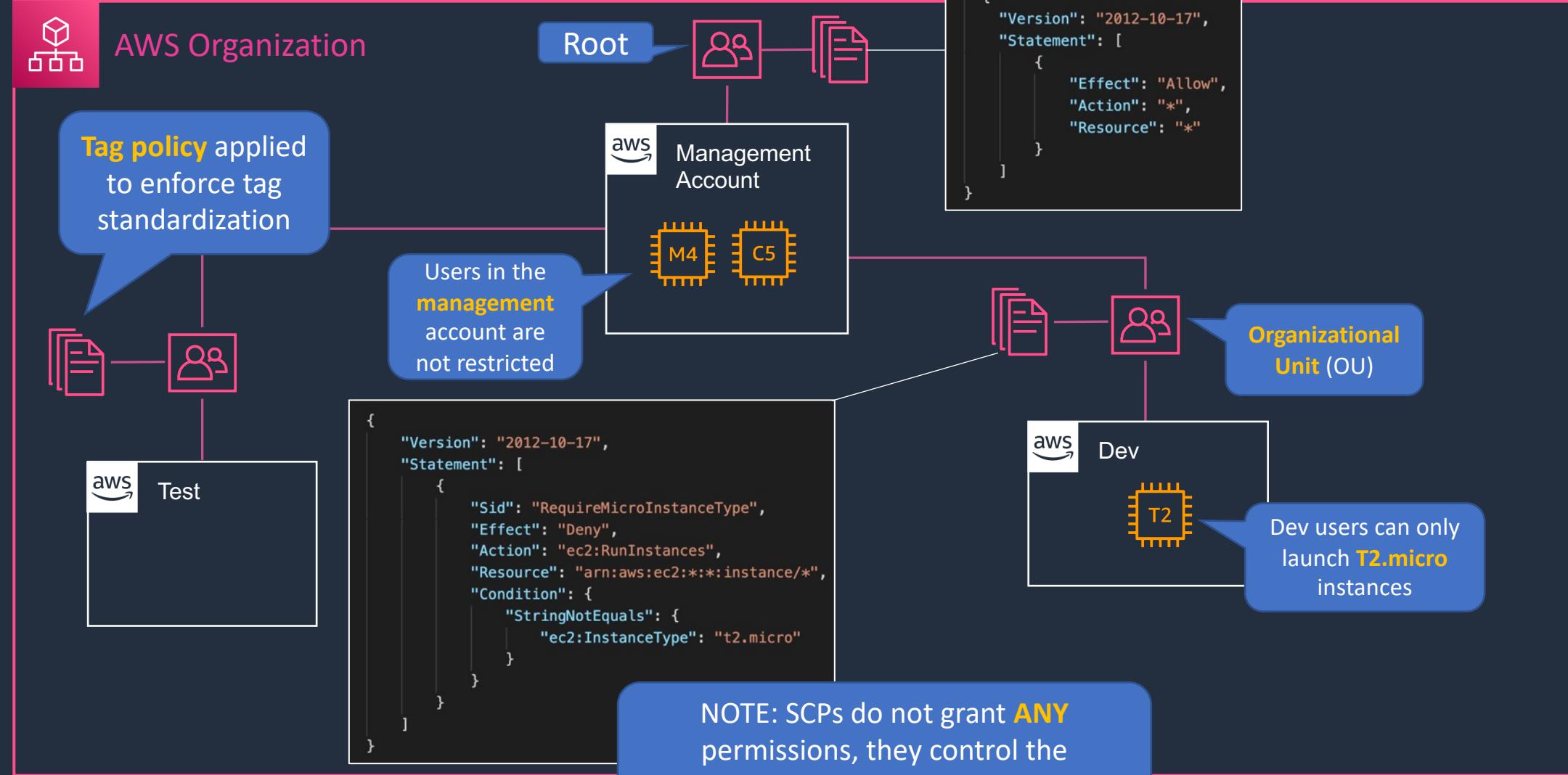
# Setup Multi-Factor Authentication (MFA)



# Service Control Policies (SCPs)



# Service Control Policies



SCPs control the maximum available permissions

# IAM Password Policy



# IAM Best Practices





# AWS IAM Best Practices

- Lock away your AWS account root user access keys
- Create individual IAM users
- Use groups to assign permissions to IAM users
- Grant least privilege
- Get started using permissions with AWS managed policies
- Use customer managed policies instead of inline policies
- Use access levels to review IAM permissions
- Configure a strong password policy for your users
- Enable MFA



# AWS IAM Best Practices

---

---

- Use roles for applications that run on Amazon EC2 instances
- Use roles to delegate permissions
- Do not share access keys
- Rotate credentials regularly
- Remove unnecessary credentials
- Use policy conditions for extra security
- Monitor activity in your AWS account

# SECTION 5

## AWS Compute Services

# Computing Basics





# Computing Basics

---

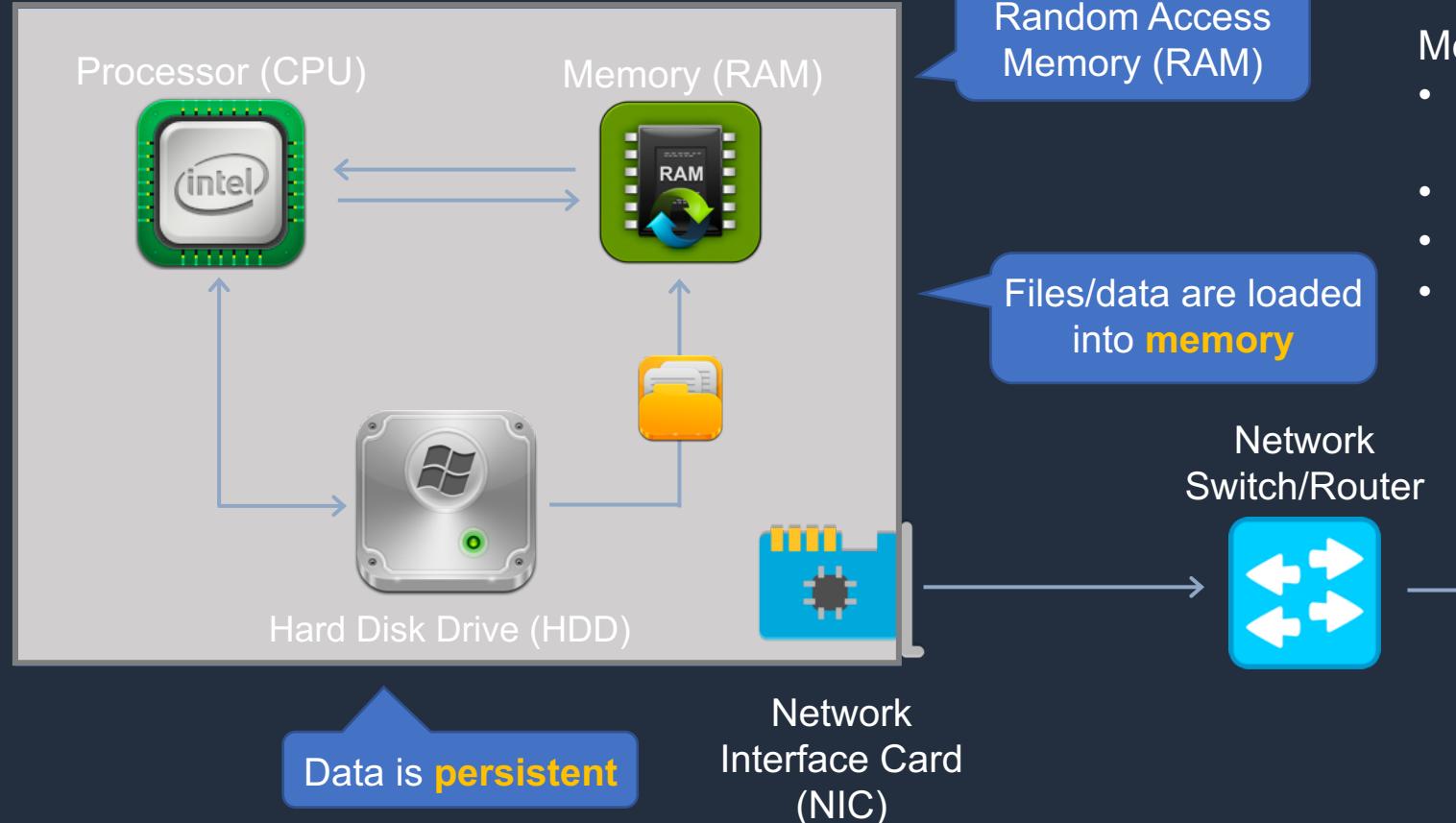




# Computing Basics

Central Processing Unit (CPU)

RAM is non-persistent storage



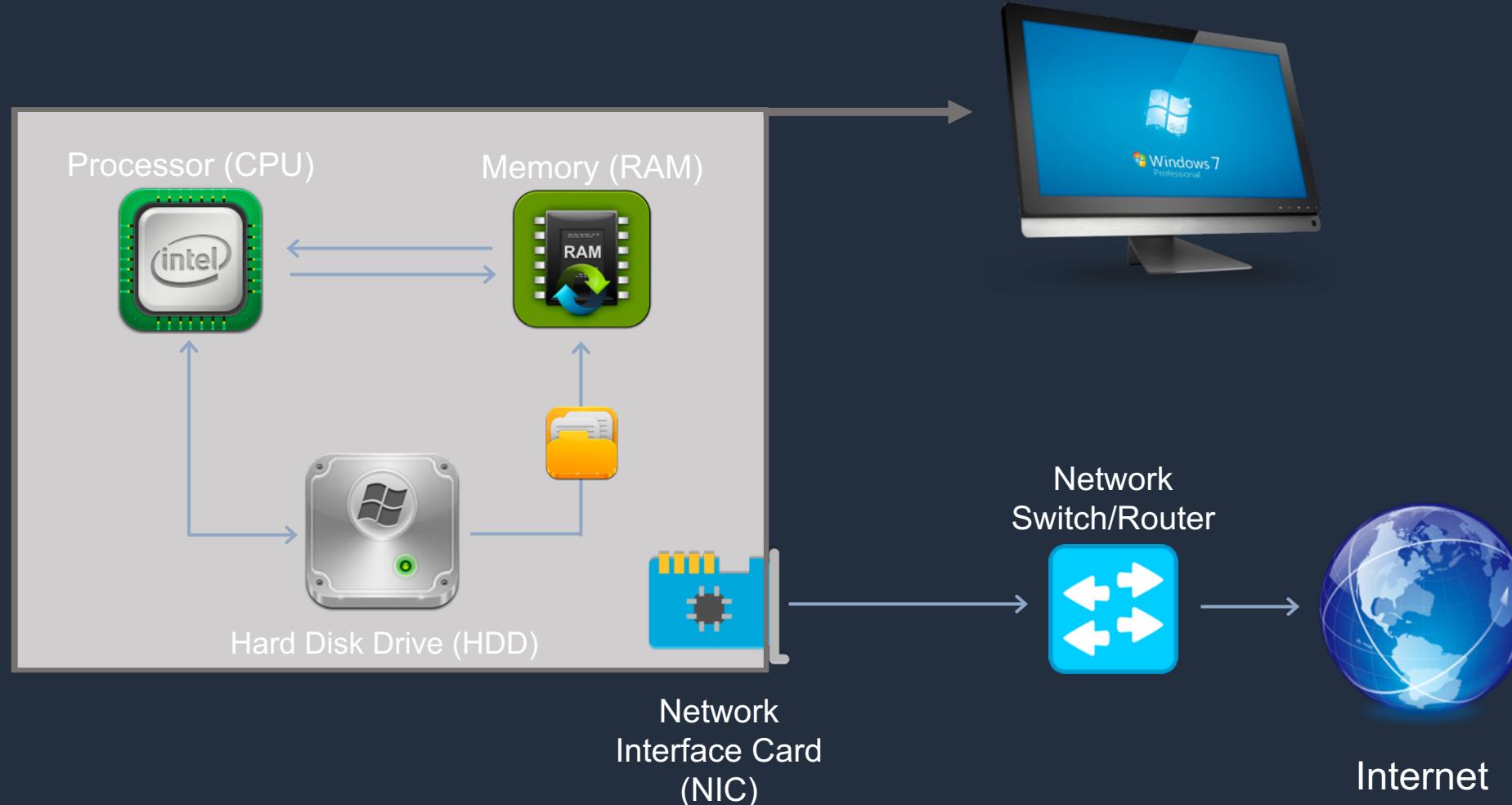
Measurements:

- CPU is measured in Gigahertz (Ghz)
- RAM is measured in Gigabyte (GB)
- HDD is measured in Gigabyte (GB)
- NIC is measured in Megabits per second (Mbps) or Gigabits per second (Gbps)



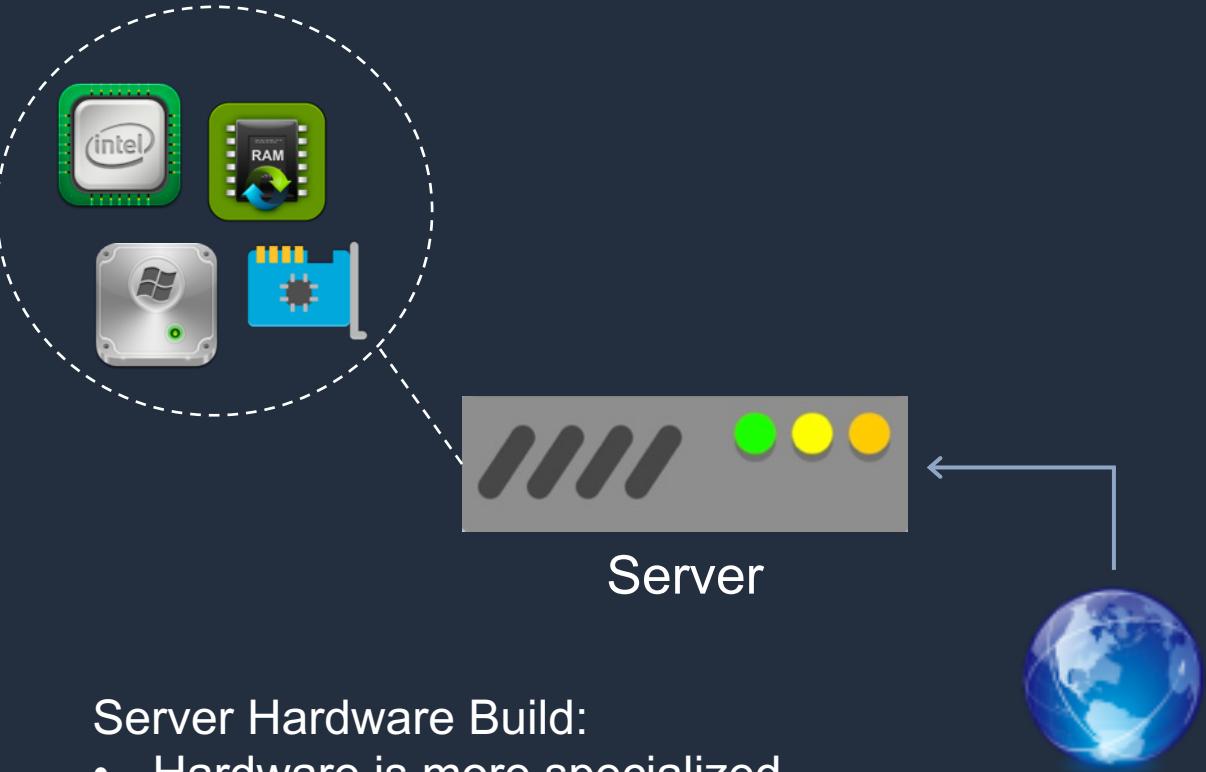
# Computing Basics

---





# Servers vs Desktops/Laptops



## Server Hardware Build:

- Hardware is more specialized
- Much higher prices compared to desktops / laptops
- Includes redundancy

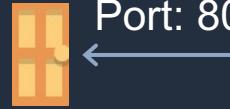




# Client / Server Computing



Web Server



Port: 80

Protocol: HTTP



The client application finds the server by **IP address**

A **port** is like a door into the server



File Server



Port: 445

Protocol: SMB



Email Server



Port: 25

Protocol: SMTP

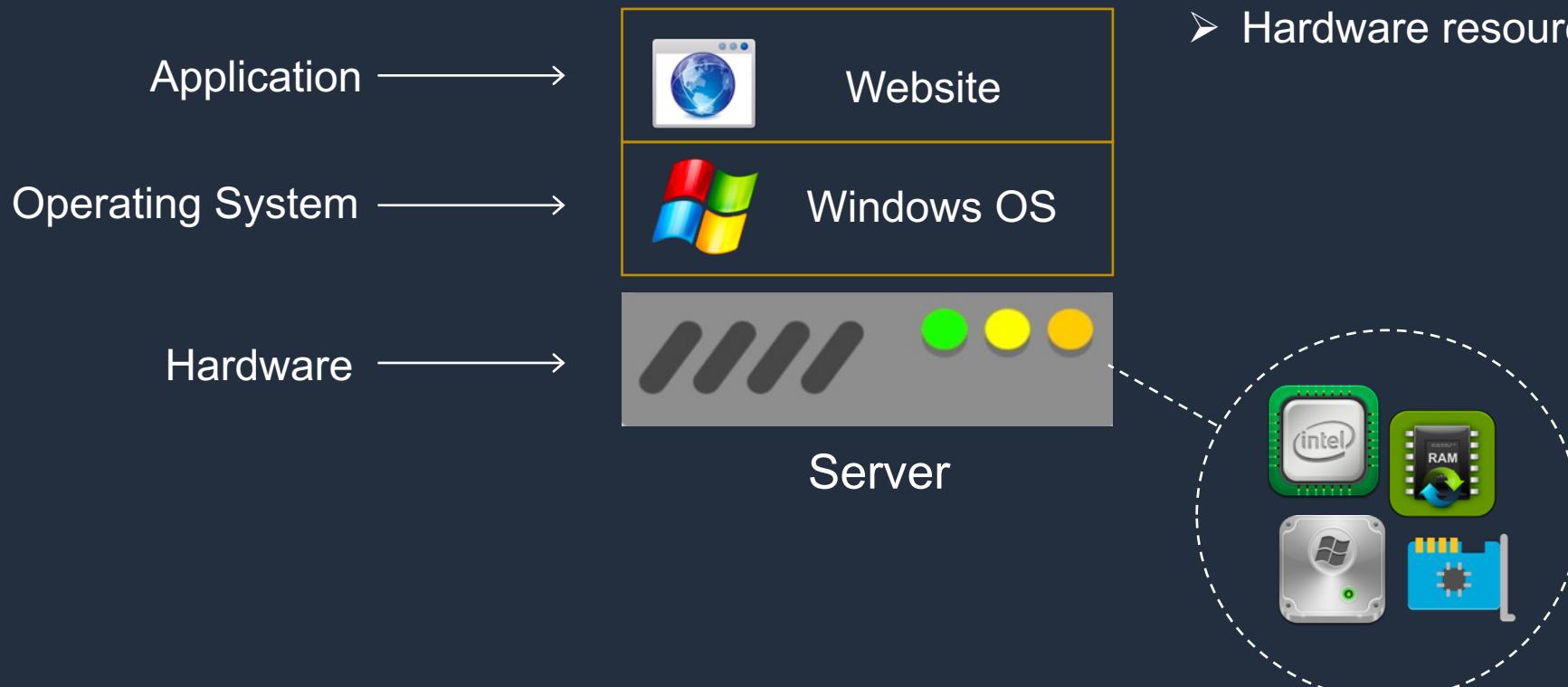


# Server Virtualization





# Without Server Virtualization

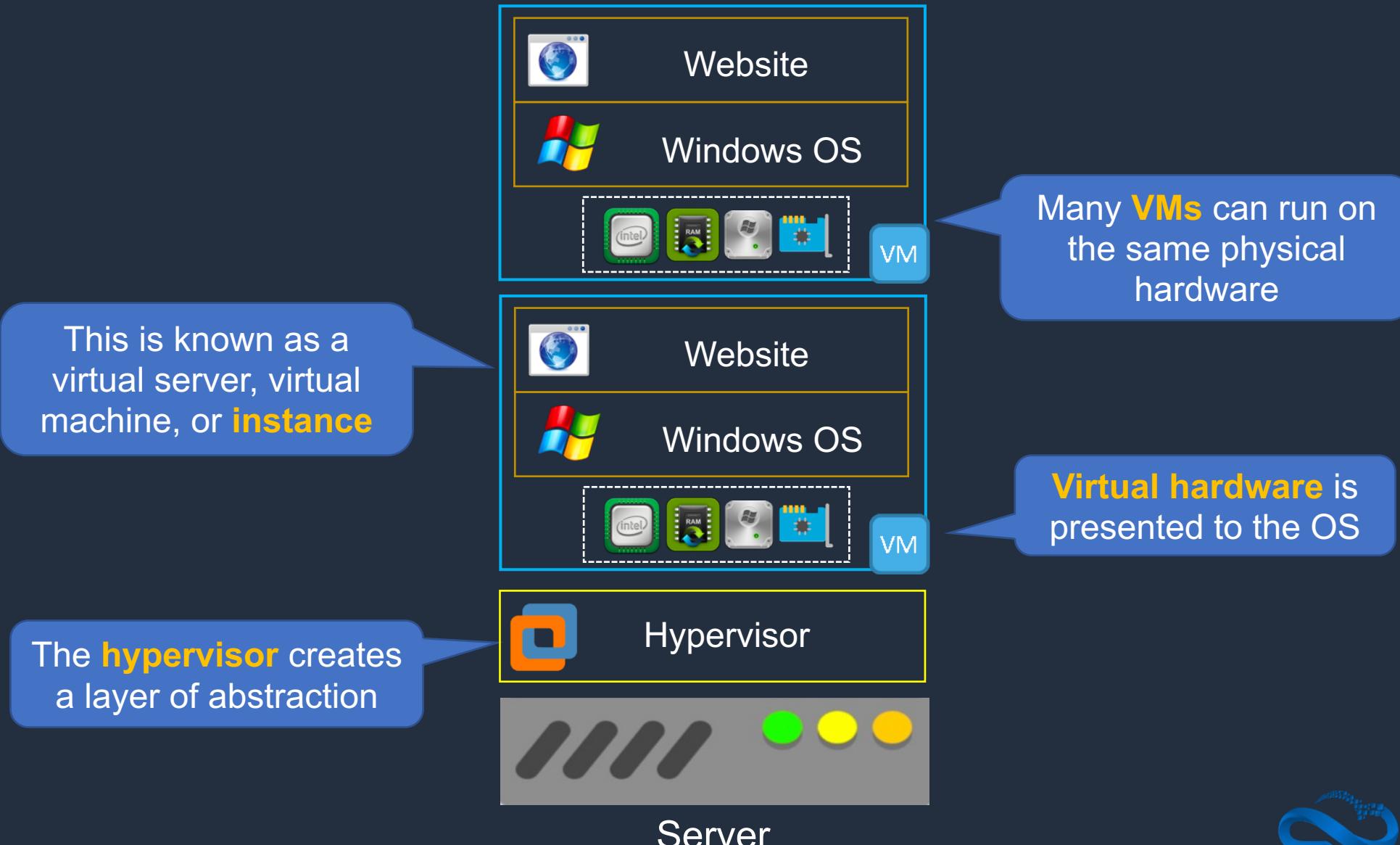


## Limitations:

- OS is tied to hardware (no portability)
- Hardware resources may be underutilized

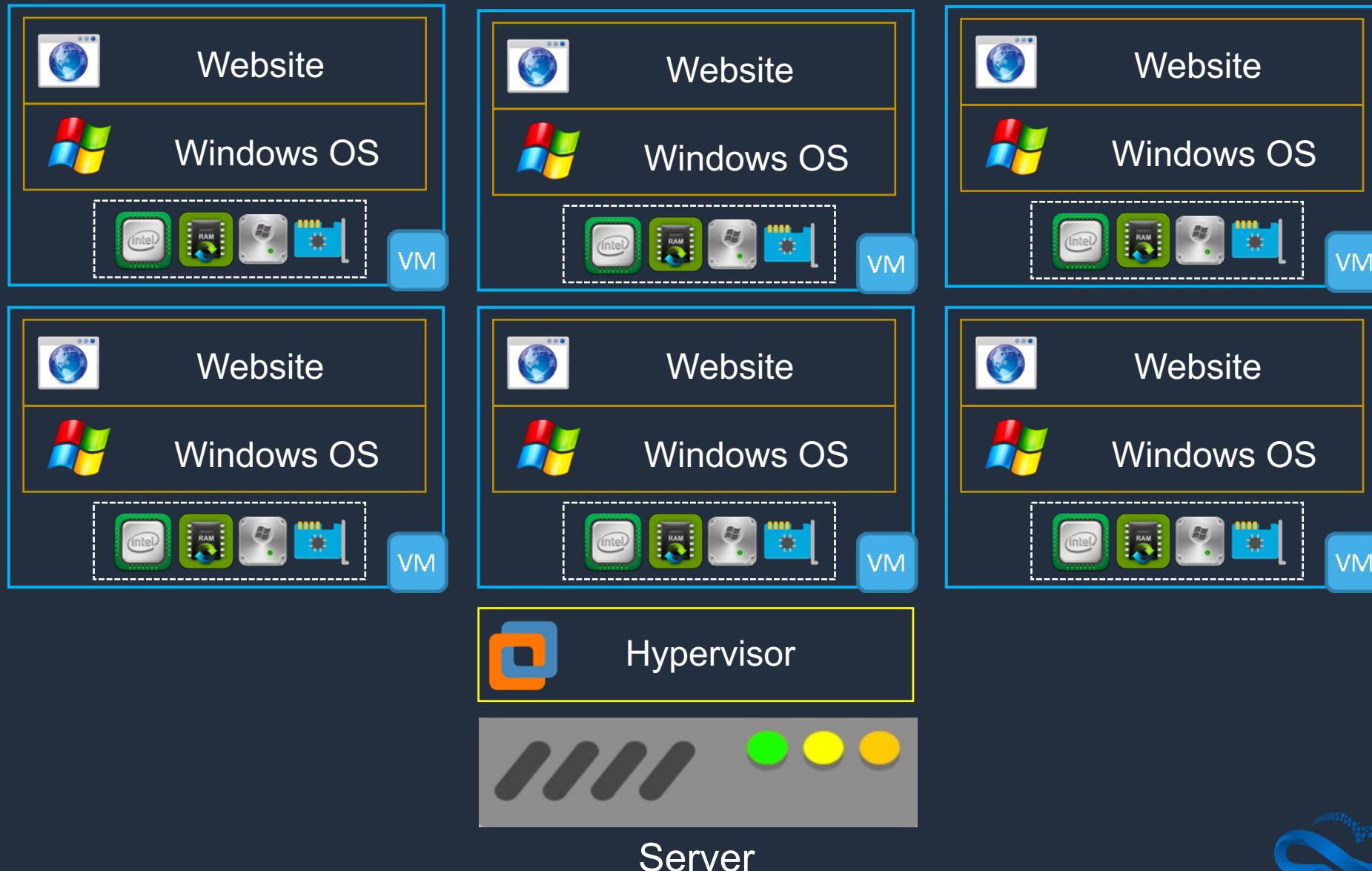


# Server Virtualization





# Server Virtualization

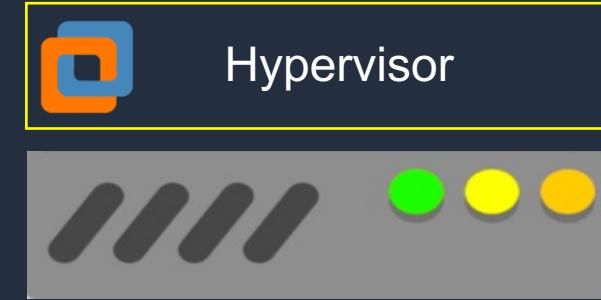




# Server Virtualization



Server



Server



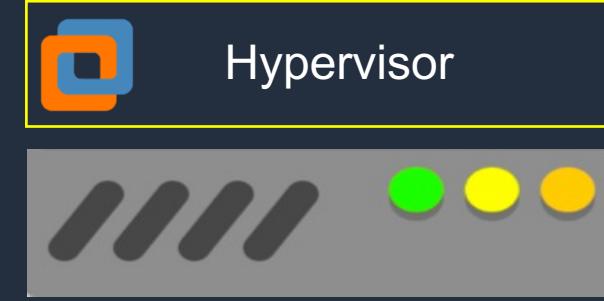
# Server Virtualization



Server

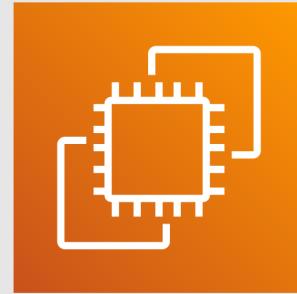


Hypervisor



Server

# Amazon Elastic Compute Cloud (EC2)



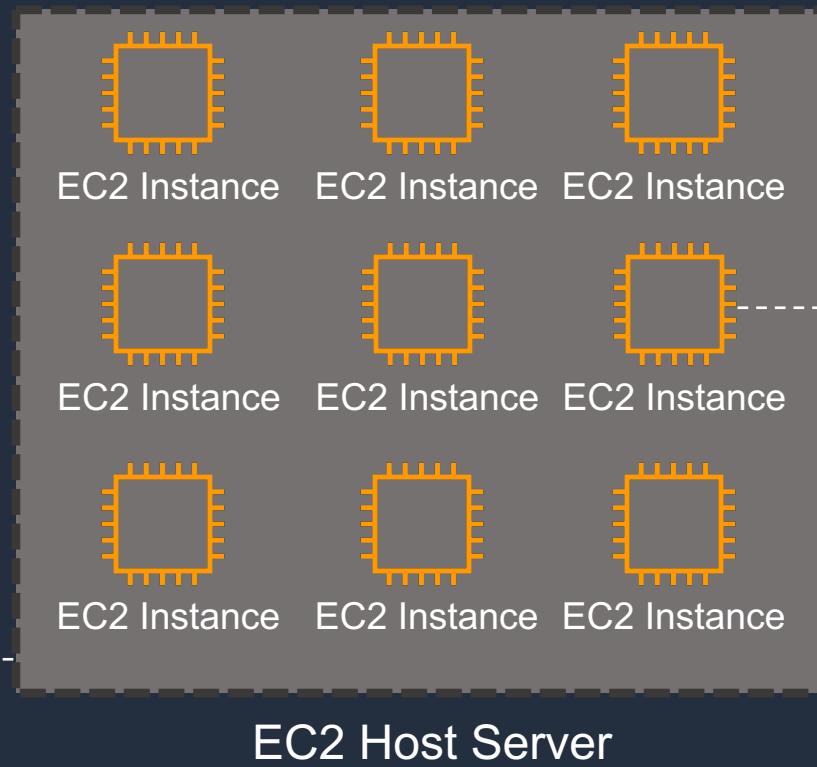


# Amazon EC2

EC2 instances run  
Windows, Linux, or  
MacOS

An **EC2 instance**  
is a virtual server

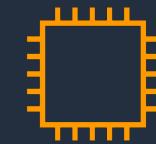
EC2 hosts are  
**managed by AWS**



A selection of **instance types**  
come with varying combinations  
of CPU, memory, storage and  
networking



# Launching an EC2 Instance



Select an **instance type**

An **AMI** defines the configuration of the instance

EC2 Instance

A **snapshot** is a point-in-time backup of an instance



Amazon Machine Image (AMI)



EBS Snapshot



Linux



Microsoft Windows

You can customize your instance and create a **custom AMI**



Customized AMI

Family	Type	vCPUs	Memory (GiB)
General purpose	t2.micro	1	1
Compute optimized	c5n.large	2	5.25
Memory optimized	r5ad.large	2	16
Storage optimized	d2.xlarge	4	30.5
GPU instances	g2.2xlarge	8	15

The **instance type** defines the hardware profile (and cost)



# Benefits of Amazon EC2

---

---

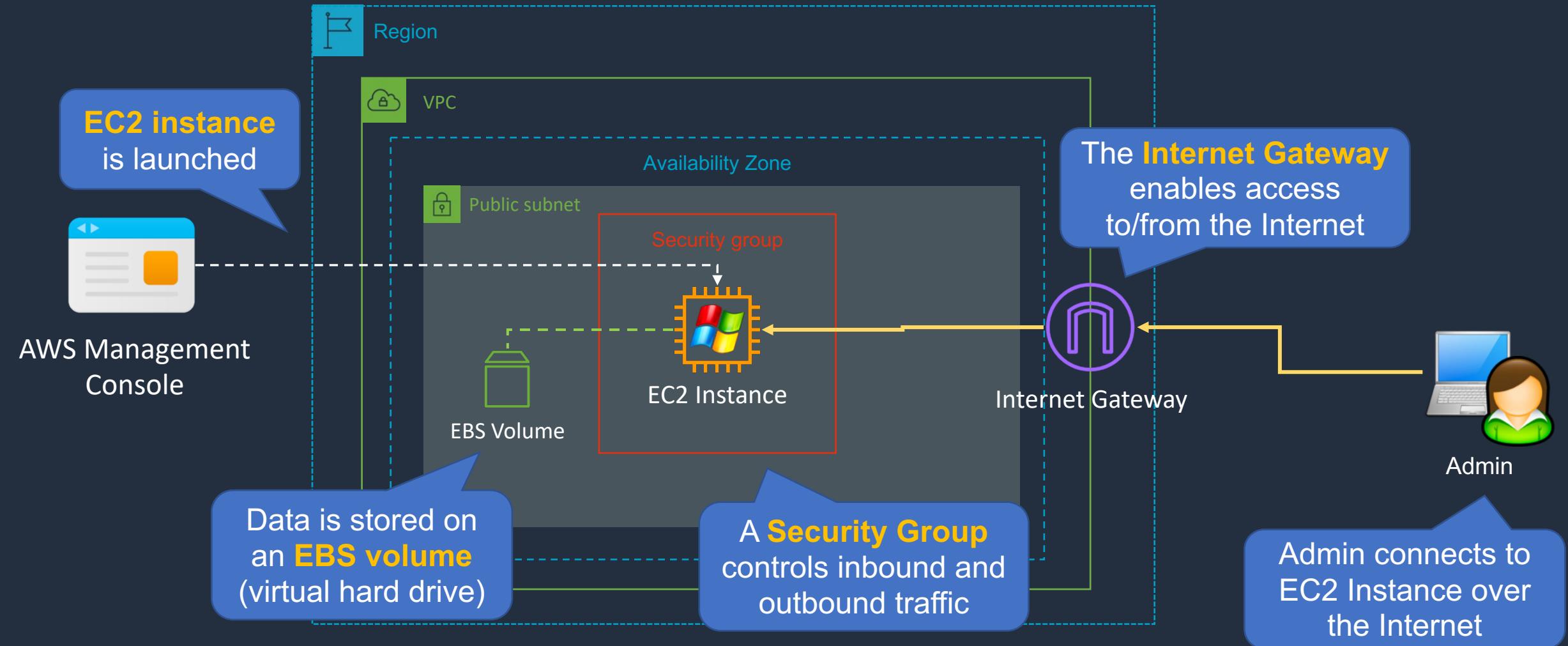
- **Elastic computing** – easily launch hundreds to thousands of EC2 instances within minutes
- **Complete control** – you control the EC2 instances with full root/administrative access
- **Flexible** – Choice of instance types, operating systems, and software packages
- **Reliable** – EC2 offers very high levels of availability and instances can be rapidly commissioned and replaced
- **Secure** – Fully integrated with Amazon VPC and security features
- **Inexpensive** – Low cost, pay for what you use

# Launch EC2 Instances (Windows + Linux)





# Amazon EC2 Instance in a Public Subnet



# EC2 Instance Connect and SSH



# RDP to Windows Instance



# Amazon EC2 User Data and Metadata





# Amazon EC2 User Data

The code is run when the instance starts for the **first time**



AWS Management Console

**Batch** and **PowerShell** scripts can be run on Windows

User data i  As text  As file  Input is already base64 encoded

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd
systemctl enable httpd
```

Limited to  
**16 KB**



EC2 Instance

EC2 Instance with a **web service** is launched



# Amazon EC2 Metadata

- Instance metadata is data about your EC2 instance
- Instance metadata is available at `http://169.254.169.254/latest/meta-data`
- Examples:



```
[ec2-user@ip-172-31-42-248 ~]$ curl http://169.254.169.254/latest/meta-data
ami-id
ami-launch-index
ami-manifest-path
block-device-mapping/
events/
hibernation/
hostname
identity-credentials/
instance-action
instance-id
instance-life-cycle
instance-type
local-hostname
local-ipv4
```



# Amazon EC2 Metadata

- Examples ctd.:

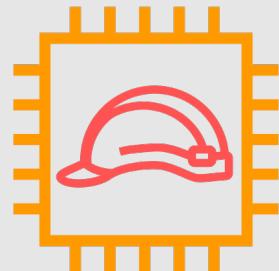
```
[ec2-user@ip-172-31-42-248 ~]$ curl http://169.254.169.254/latest/meta-data/local-ipv4  
172.31.42.248[ec2-user@ip-172-31-42-248 ~]$
```

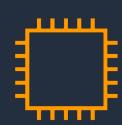
```
[ec2-user@ip-172-31-42-248 ~]$ curl http://169.254.169.254/latest/meta-data/public-ipv4  
3.26.54.18[ec2-user@ip-172-31-42-248 ~]$
```

# [HOL] Launch Instance with User Data and Metadata

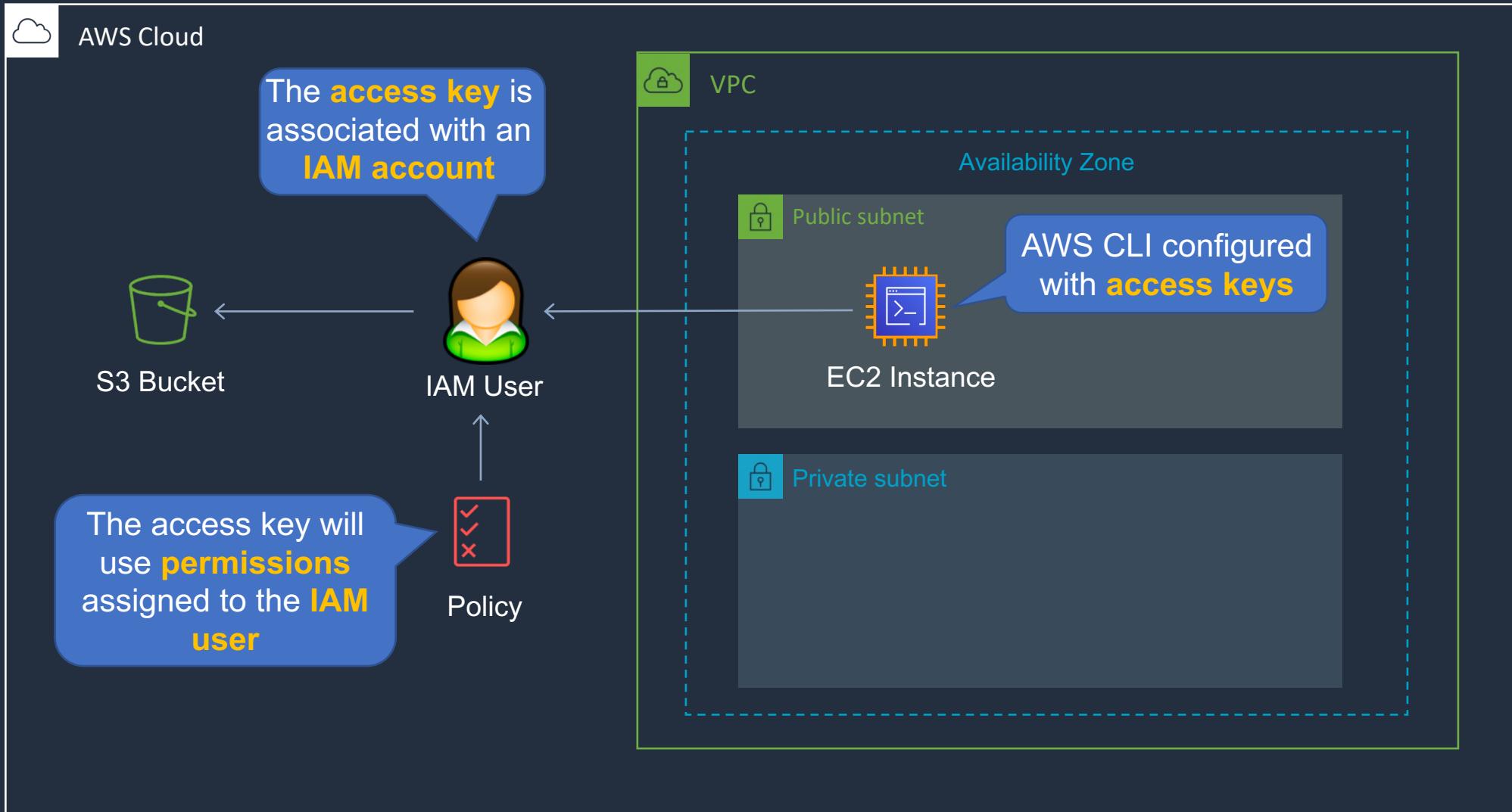


# Accessing Services – Access Keys and IAM Roles



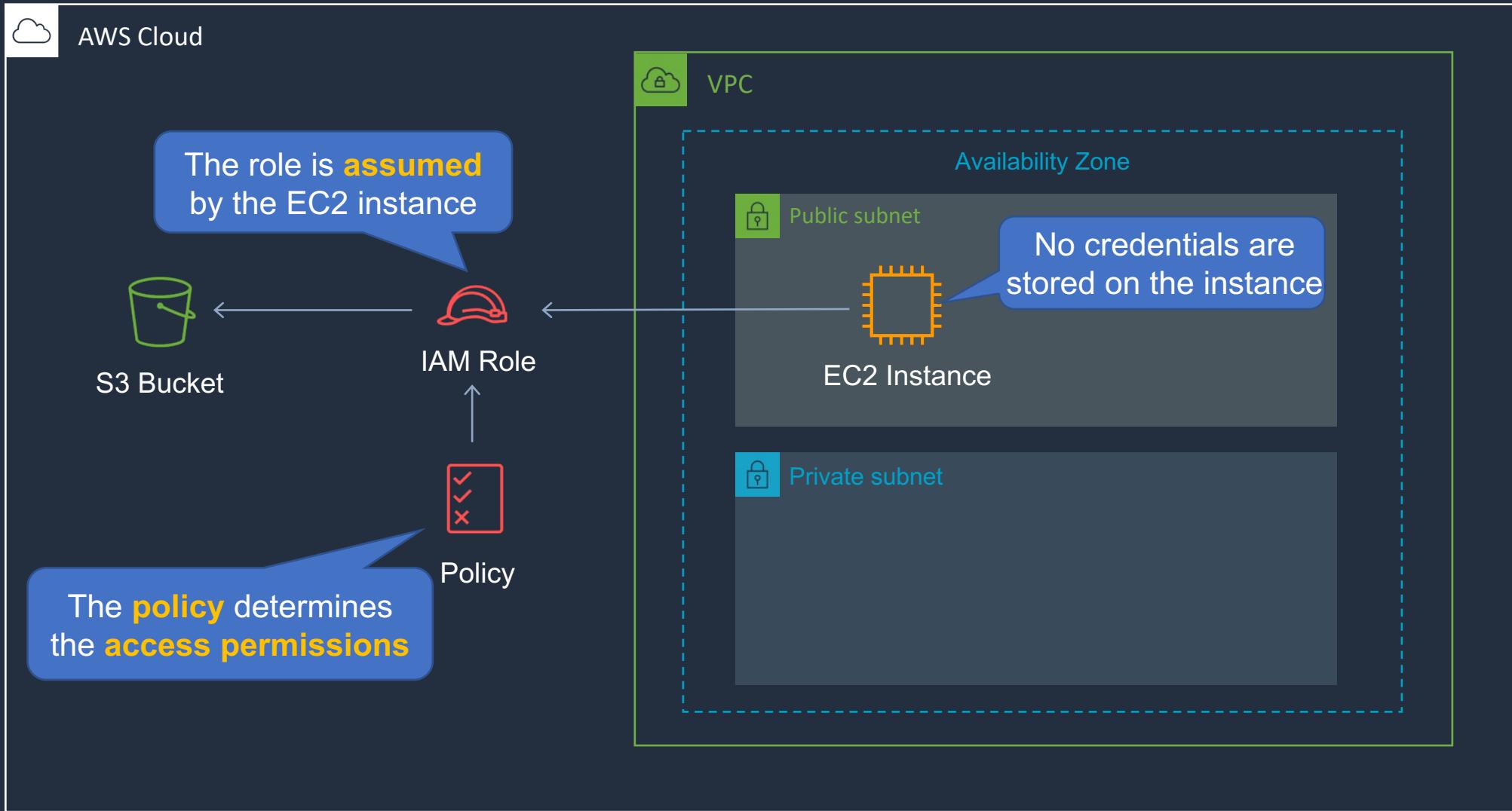


# Access Keys





# Amazon EC2 Instance Profiles (IAM Roles for EC2)



# Access Keys and IAM Roles



# AWS Batch





# AWS Batch



Launch a **Batch Job**



Job **Definition**

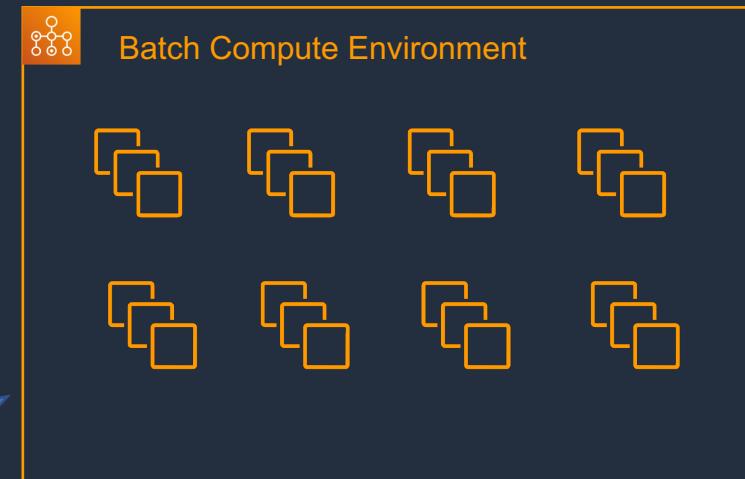


Job **Queue**

A job is submitted to a **queue** until **scheduled** onto a compute environment

A job is a unit of work such as a **shell script**, **executable** or **Docker container image**

Batch **launches**, **manages**, and **terminates** resources as required (EC2 and ECS/Fargate)



Managed or **unmanaged** resources used to run the job

# Amazon LightSail





# Amazon LightSail

- Low cost and ideal for users with less technical expertise
- Compute, storage, and network
- Preconfigured virtual servers

## Amazon's Simple Cloud Server

Amazon Lightsail - Powerful virtual servers built for reliability & performance

**FREE TIER OFFER**

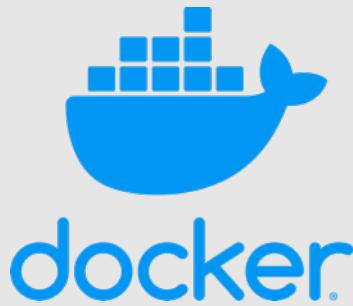
**1 MONTH FREE\***

1 Month Free Trial      Starting at \$3.50/month      Starting at \$8/month

- Virtual servers, databases and load balancers
- SSH and RDP access
- Can access Amazon VPC

**Exam tip:** typically comes up in use cases where an easy method of deploying a virtual server is required by a user with little or no AWS expertise

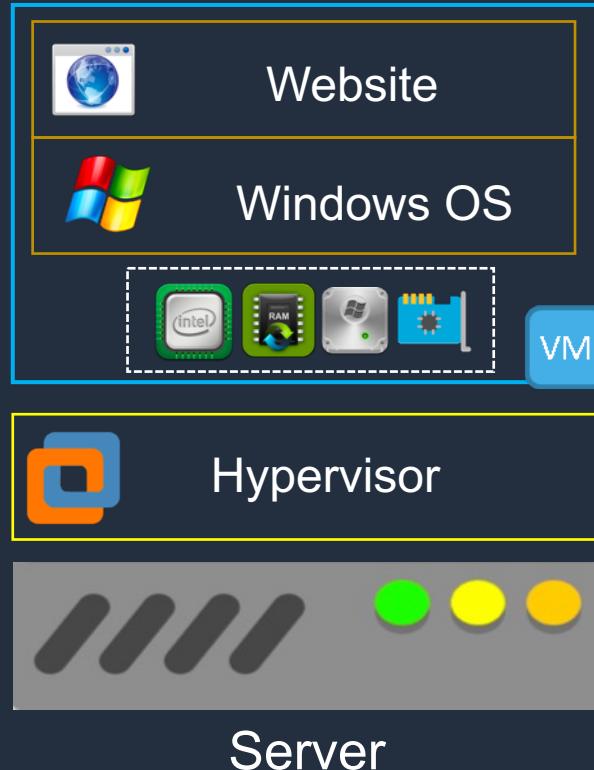
# Docker Containers and Microservices





# Server Virtualization vs Containers

Every VM-instance needs an **operating system** which uses significant resources





# Docker Containers

Containers **start up** very **quickly**

A **container** includes all the code, settings, and dependencies for running the application



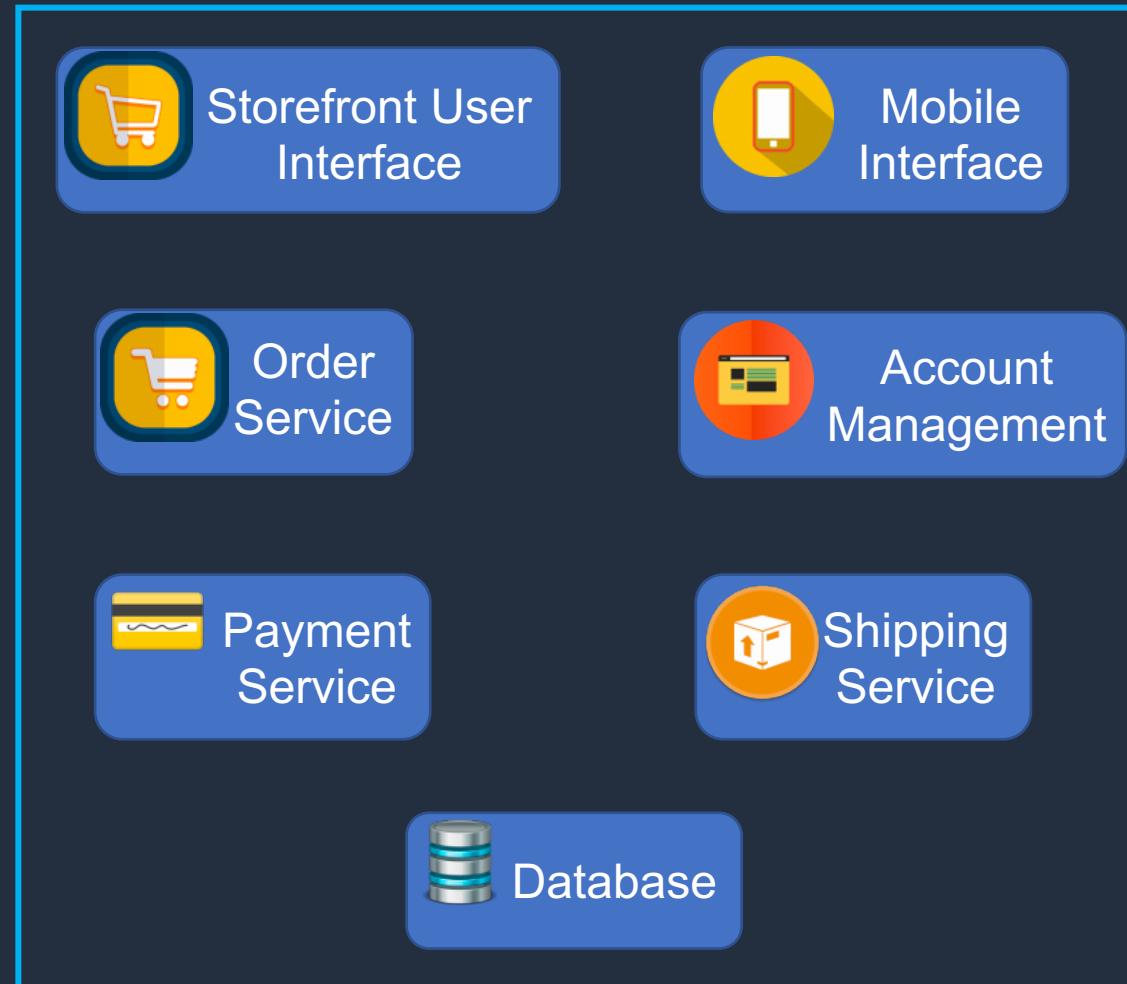
Containers are very resource **efficient**

Each container is **isolated** from other containers

# Monolithic Application

---

---



# Monolithic Application

Updates to, or failures of, any single component can take down the whole application



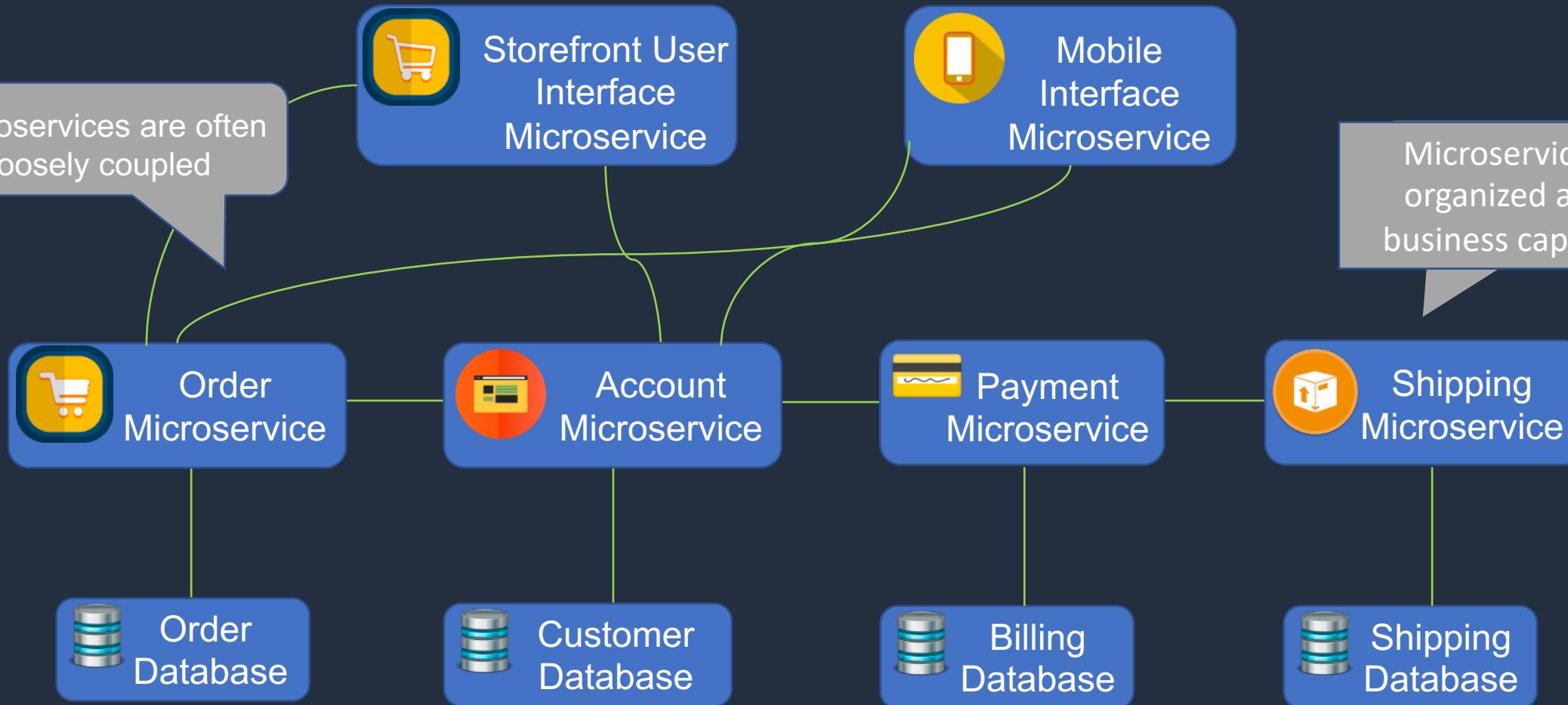
The user interface, business logic, and data access layer are combined on a single platform



# Microservices Application

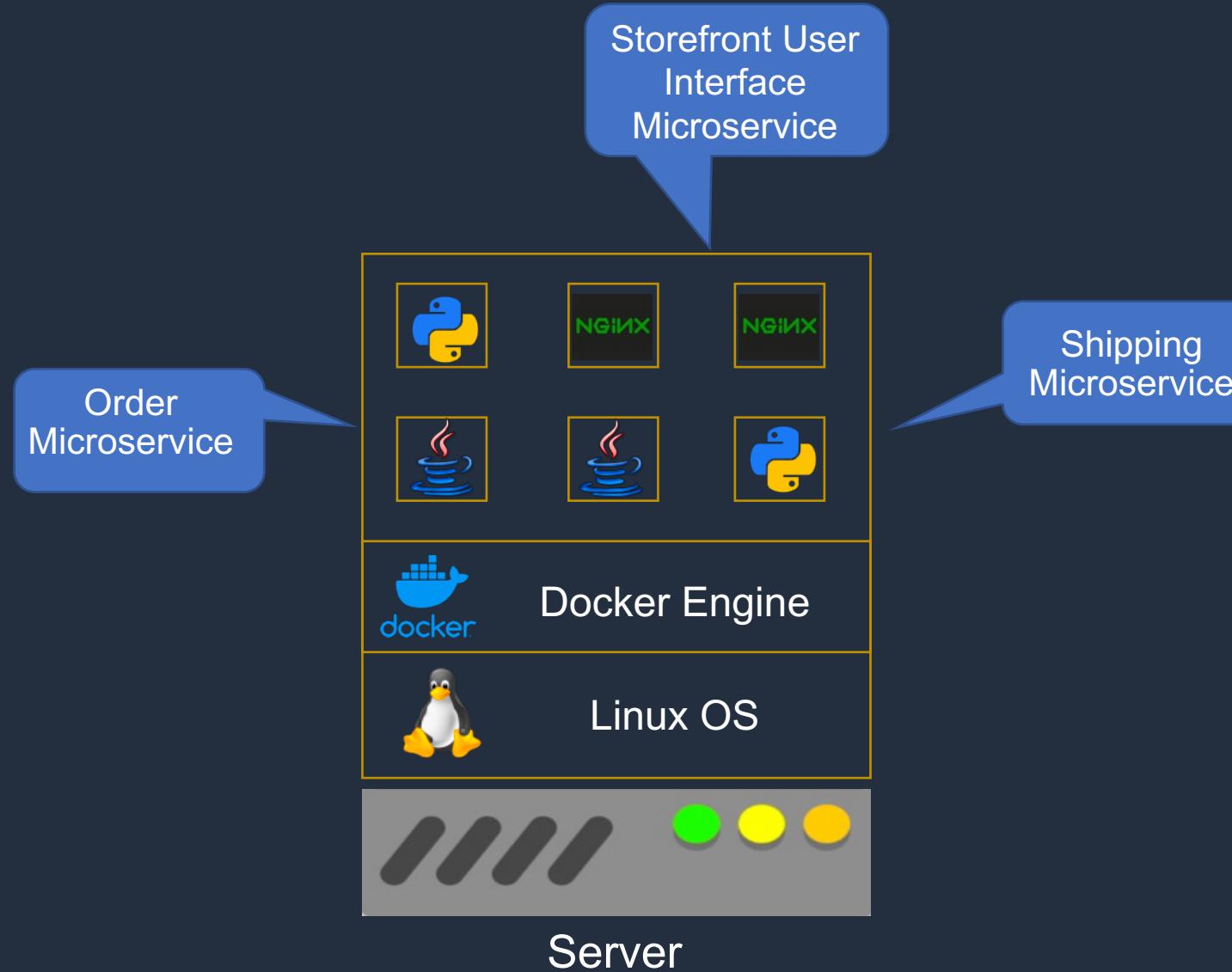
A microservice is an independently deployable unit of code

Microservices are often loosely coupled





# Microservices Application





# Microservices Application

Microservices can also be  
**spread** across hosts

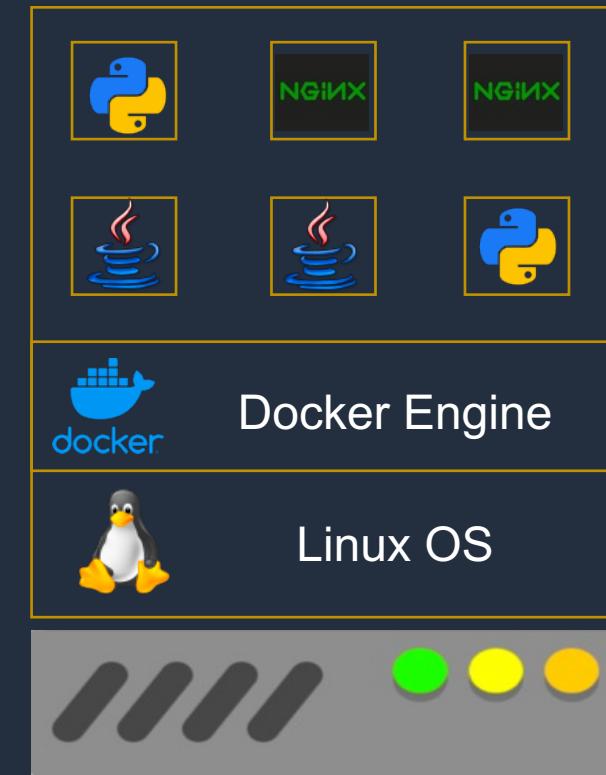
**Many instances** of each microservice  
can run on each host



Server



Server



Server

# Amazon Elastic Container Service (ECS)





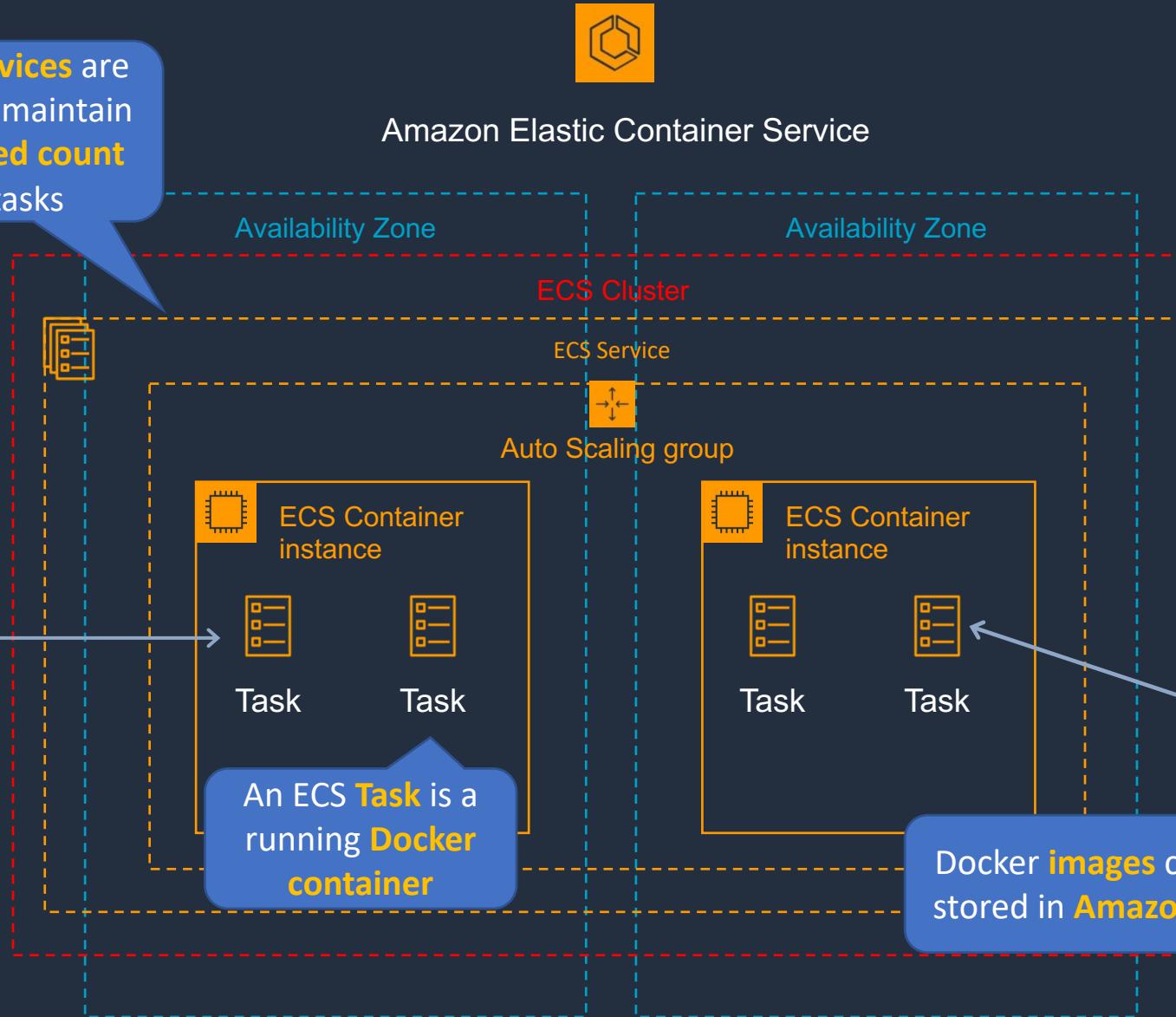
# Amazon ECS

An ECS **Task** is created from a **Task Definition**

Task Definition

```
{  
  "containerDefinitions": [  
    {  
      "name": "wordpress",  
      "links": [  
        "mysql"  
      ],  
      "image": "wordpress",  
      "essential": true,  
      "portMappings": [  
        {  
          "containerPort": 80,  
          "hostPort": 80  
        }  
      ],  
      "memory": 500,  
      "cpu": 10  
    }  
  ]  
}
```

ECS **Services** are used to maintain a **desired count** of tasks

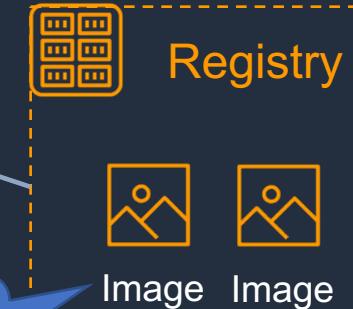


## Amazon Elastic Container Service

An Amazon **ECS Cluster** is a logical grouping of **tasks** or **services**

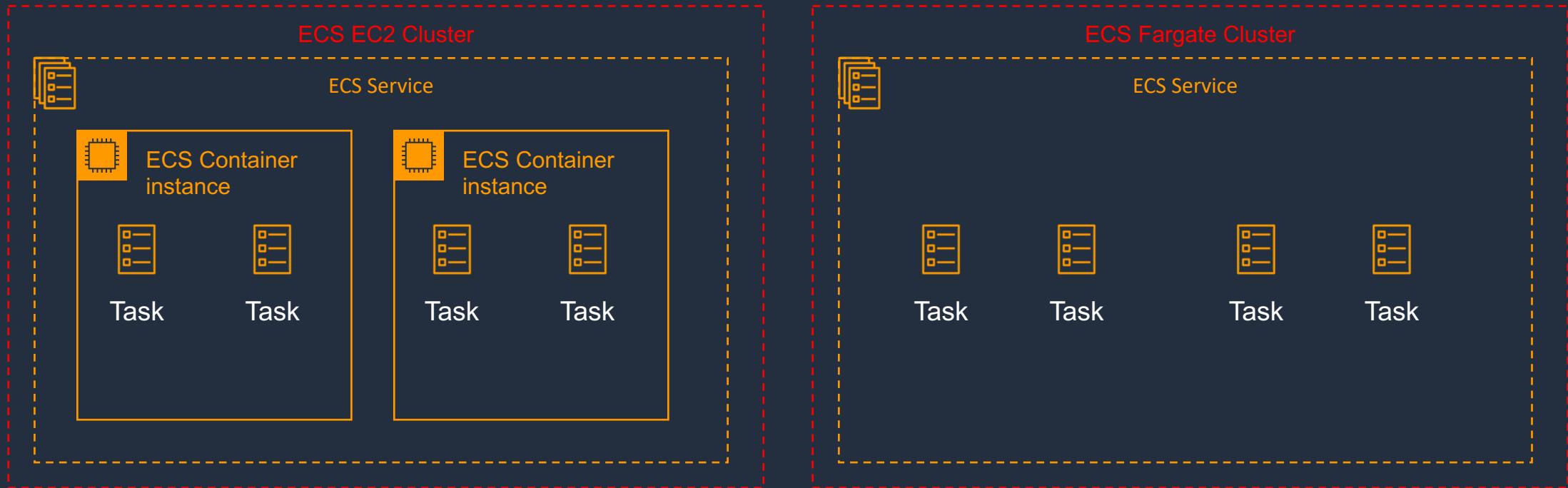


Amazon Elastic Container Registry





# Amazon ECS



## EC2 Launch Type

- You explicitly provision EC2 instances
- You're responsible for managing EC2 instances
- Charged per running EC2 instance
- EFS and EBS integration
- You handle cluster optimization
- More granular control over infrastructure

## Fargate Launch Type

- Fargate automatically provisions resources
- Fargate provisions and manages compute
- Charged for running tasks
- No EFS and EBS integration
- Fargate handles cluster optimization
- Limited control, infrastructure is automated

# Launch Docker Container on ECS



# SECTION 6

## AWS Storage Services

# Block vs File vs Object Storage





# Hard Drives

Hard drives are **block-based** storage systems

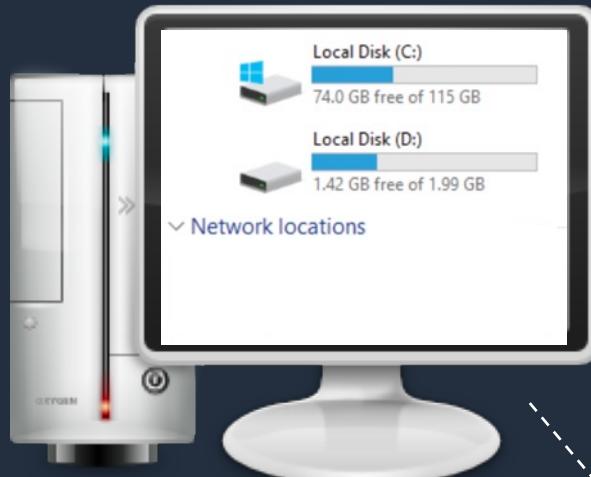


The Operating System (OS) can be used to create **volumes**. A volume can be partitioned and formatted

Hard drives are block-based storage systems



# Network Attached Storage



The Operating System (OS) sees a **filesystem** that is mapped to a local drive letter

The NAS “shares” **filesystems** over the network



Network Switch



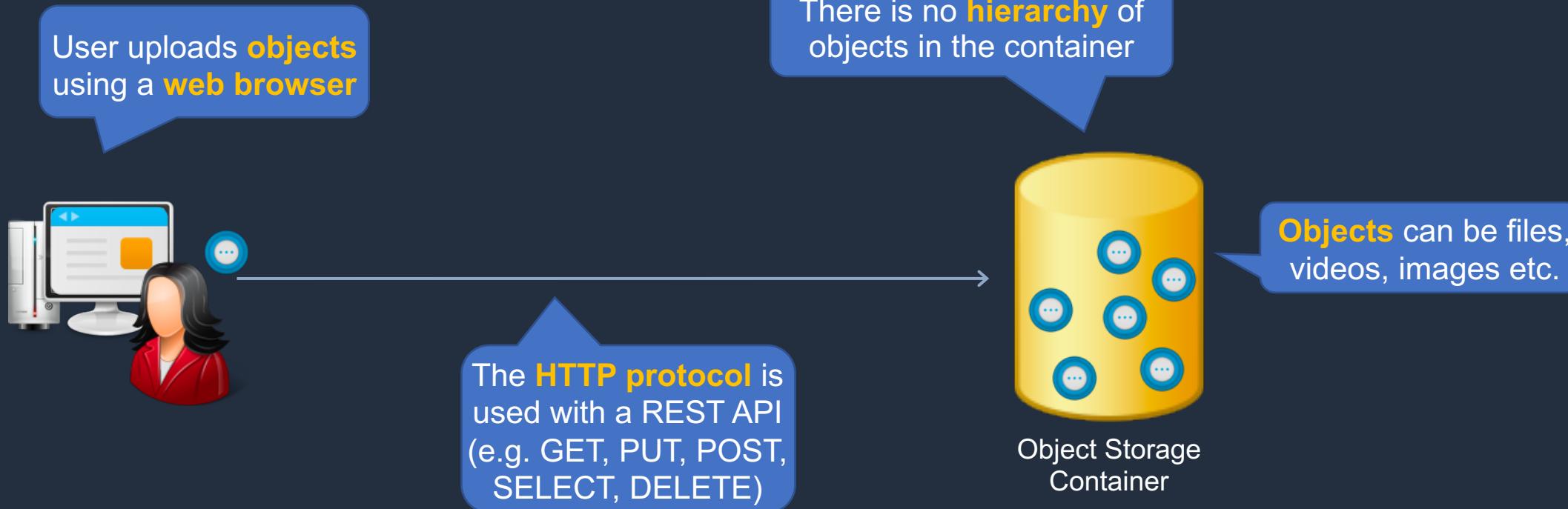
Network Attached Storage Server (NAS)



NAS devices are file-based storage systems



# Object Storage Systems





# Block, File, and Object Storage

The OS sees **volumes** that can be partitioned and formatted

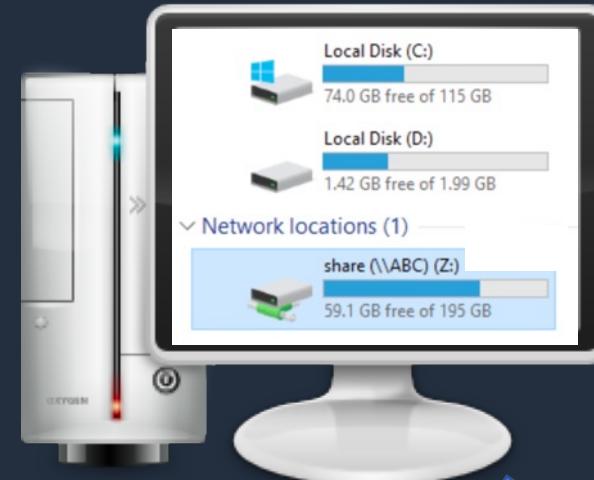
Block Storage



The OS reads/writes at the **block level**. Disks can be internal, or network attached

A **filesystem** can be shared by many users/computers

File Storage



A **filesystem** is “mounted” to the OS using a **network share**

Massively scalable, low cost

Object Storage



Object Storage Container

There is **no hierarchy** of objects in the container

Uses a **REST API**



# AWS Storage Services

---

## Block Storage



Amazon Elastic Block  
Store

## File Storage



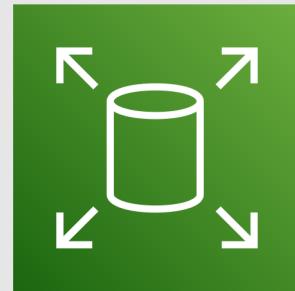
Amazon Elastic  
File System

## Object Storage



Amazon Simple  
Storage Service (S3)

# Amazon Elastic Block Store (EBS)

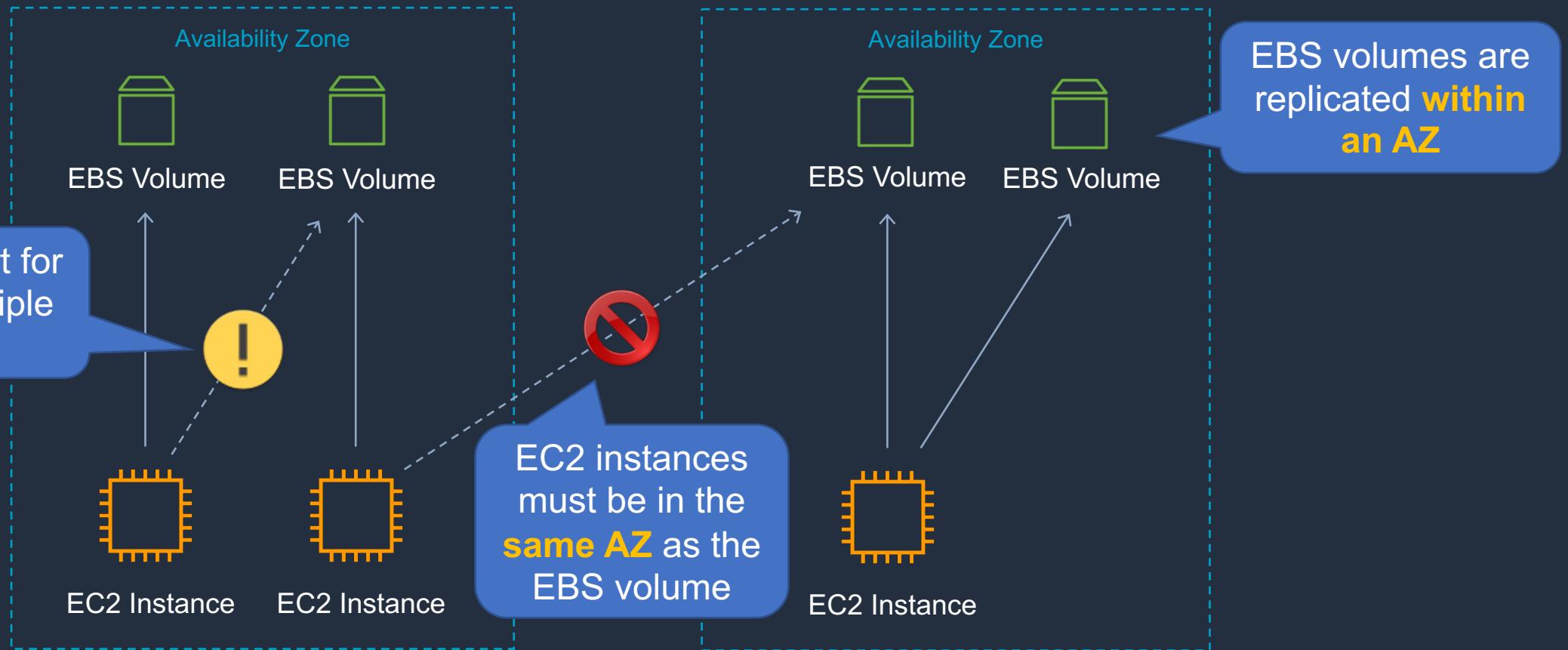




# Amazon EBS



## Amazon Elastic Block Store (EBS)





# Amazon EBS SSD-Backed Volumes

New and **not** on  
the exam yet

New and **not** on  
the exam yet

	General Purpose SSD		Provisioned IOPS SSD		
Volume type	gp3	gp2	io2 Block Express ‡	io2	io1
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)		99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
Use cases	<ul style="list-style-type: none"><li>Low-latency interactive apps</li><li>Development and test environments</li></ul>		Workloads that require sub-millisecond latency, and sustained IOPS performance or more than 64,000 IOPS or 1,000 MiB/s of throughput		<ul style="list-style-type: none"><li>Workloads that require sustained IOPS performance or more than 16,000 IOPS</li><li>I/O-intensive database workloads</li></ul>
Volume size	1 GiB - 16 TiB		4 GiB - 64 TiB	4 GiB - 16 TiB	
Max IOPS per volume (16 KiB I/O)	16,000		256,000	64,000 †	
Max throughput per volume	1,000 MiB/s	250 MiB/s *	4,000 MiB/s	1,000 MiB/s †	
Amazon EBS Multi-attach	Not supported		Not supported	Supported	
Boot volume	Supported				



# Amazon EBS HDD-Backed Volumes

	Throughput Optimized HDD	Cold HDD
<b>Volume type</b>	st1	sc1
<b>Durability</b>	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)
<b>Use cases</b>	<ul style="list-style-type: none"><li>• Big data</li><li>• Data warehouses</li><li>• Log processing</li></ul>	<ul style="list-style-type: none"><li>• Throughput-oriented storage for data that is infrequently accessed</li><li>• Scenarios where the lowest storage cost is important</li></ul>
<b>Volume size</b>	125 GiB - 16 TiB	125 GiB - 16 TiB
<b>Max IOPS per volume (1 MiB I/O)</b>	500	250
<b>Max throughput per volume</b>	500 MiB/s	250 MiB/s
<b>Amazon EBS Multi-attach</b>	Not supported	Not supported
<b>Boot volume</b>	Not supported	Not supported

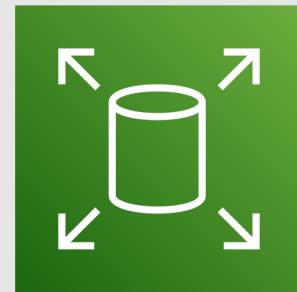


# Amazon EBS

---

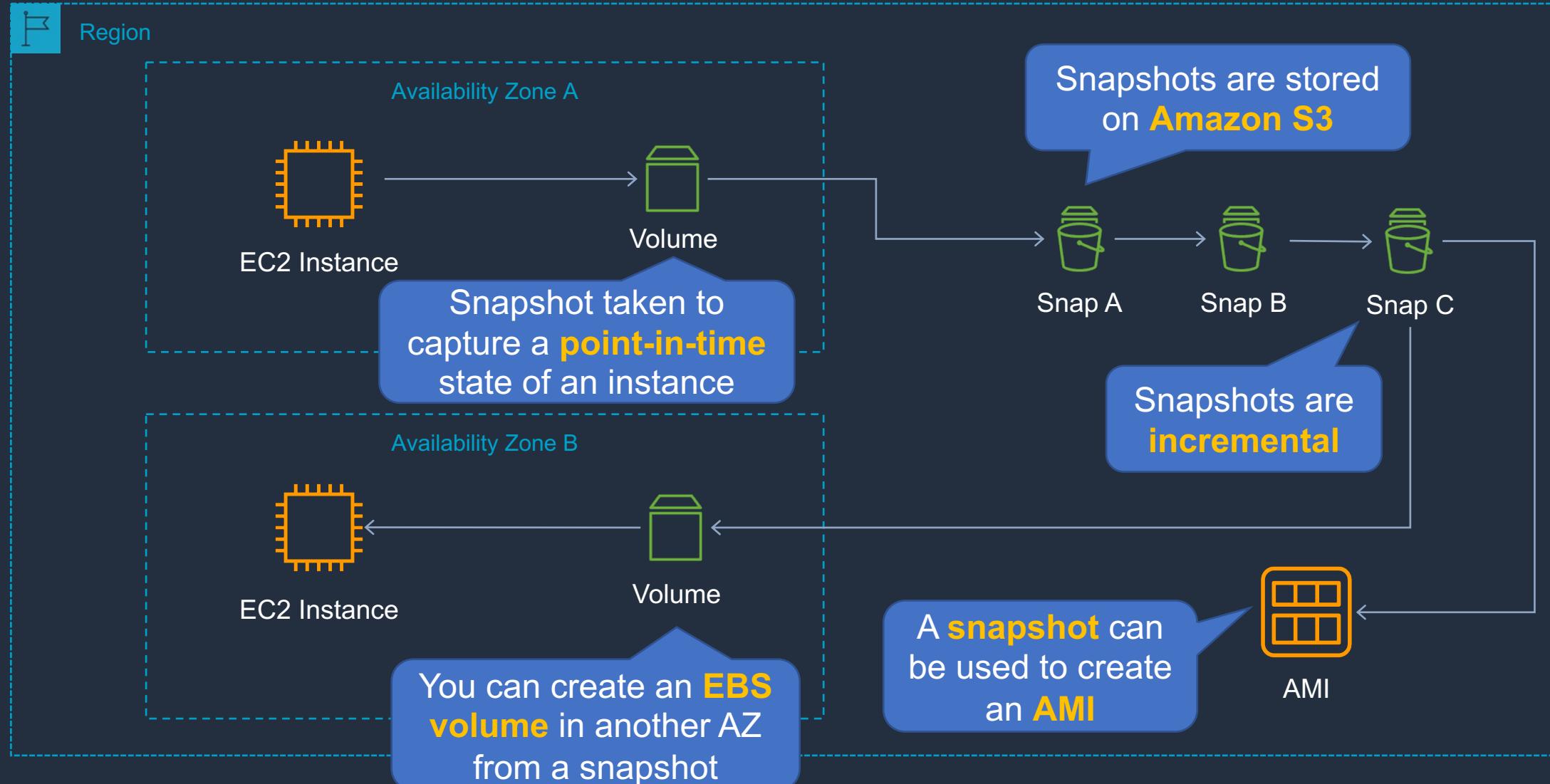
- EBS volume data persists **independently** of the life of the instance
- EBS volumes do not need to be attached to an instance
- You can attach multiple EBS volumes to an instance
- You can use multi-attach to attach a volume to multiple instances but with some constraints
- EBS volumes must be in the **same AZ** as the instances they are attached to
- Root EBS volumes **are deleted** on termination by default
- Extra non-boot volumes **are not deleted** on termination by default

# Amazon EBS Snapshots and DLM





# Amazon EBS Snapshots



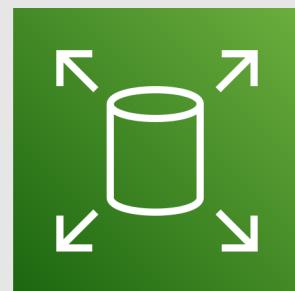


# Amazon Data Lifecycle Manager (DLM)

---

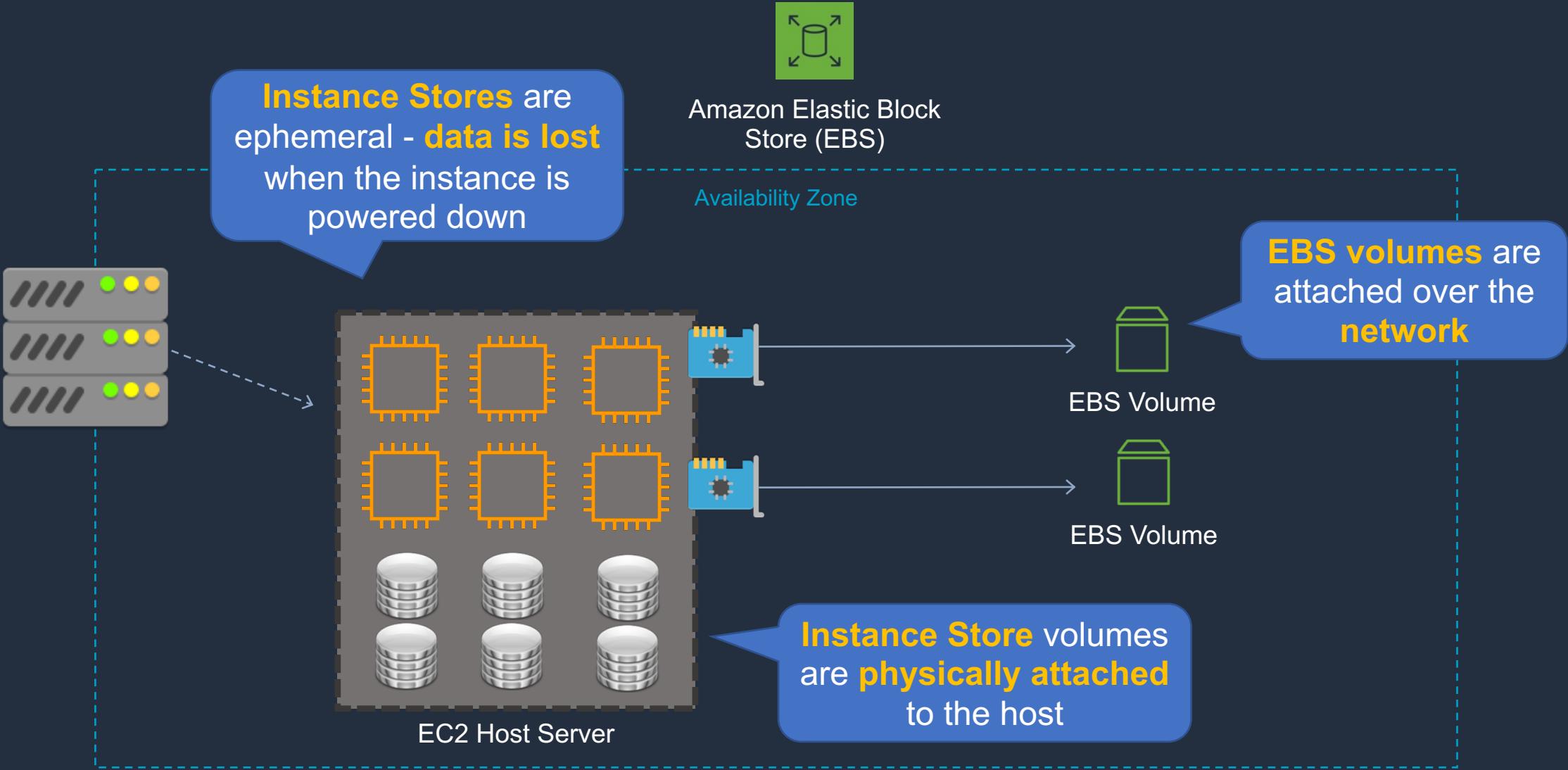
- DLM automates the creation, retention, and deletion of EBS snapshots and EBS-backed AMIs
- DLM helps with the following:
  - Protects valuable data by enforcing a regular backup schedule
  - Create standardized AMIs that can be refreshed at regular intervals
  - Retain backups as required by auditors or internal compliance
  - Reduce storage costs by deleting outdated backups
  - Create disaster recovery backup policies that back up data to isolated accounts

# EC2 Instance Store Volumes





# EBS vs instance store



# EBS Volumes and Snapshots



# Amazon Machine Images (AMI)





# Amazon Machine Images (AMIs)

---

---

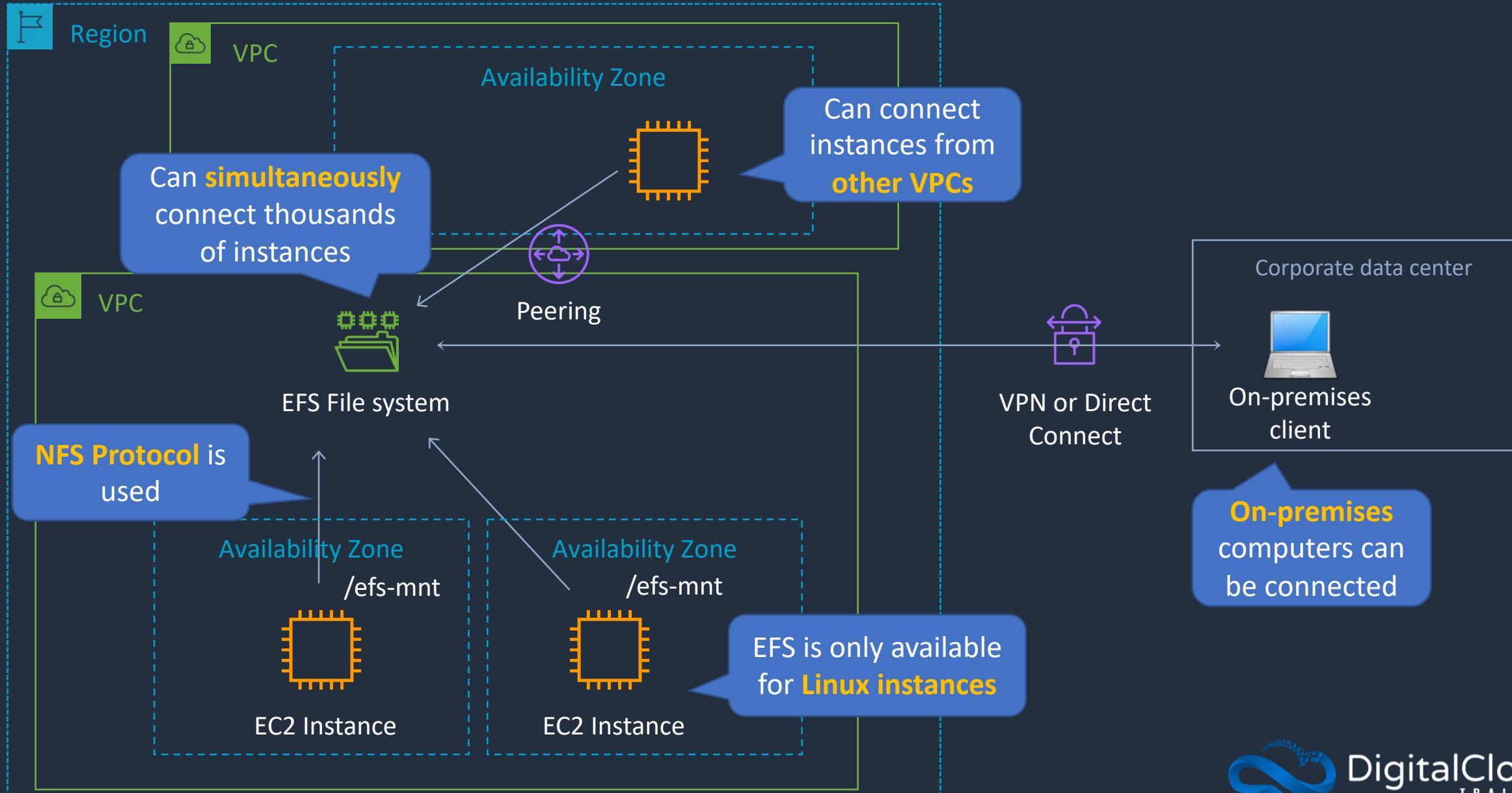
- An **Amazon Machine Image** (AMI) provides the information required to launch an instance
- An AMI includes the following:
  - One or more EBS snapshots, or, for instance-store-backed AMIs, a template for the root volume of the instance (for example, an operating system, an application server, and applications)
  - Launch permissions that control which AWS accounts can use the AMI to launch instances
  - A block device mapping that specifies the volumes to attach to the instance when it's launched
- AMIs come in three main categories:
  - **Community AMIs** - free to use, generally you just select the operating system you want
  - **AWS Marketplace AMIs** - pay to use, generally come packaged with additional, licensed software
  - **My AMIs** - AMIs that you create yourself

# Amazon Elastic File System (EFS)





# Amazon EFS



# Amazon Simple Storage Service (S3)





# Amazon S3



Bucket

A **bucket** is a container for objects

[http://\*bucket\*.s3.\*aws-region\*.amazonaws.com](http://bucket.s3.amazonaws.com)

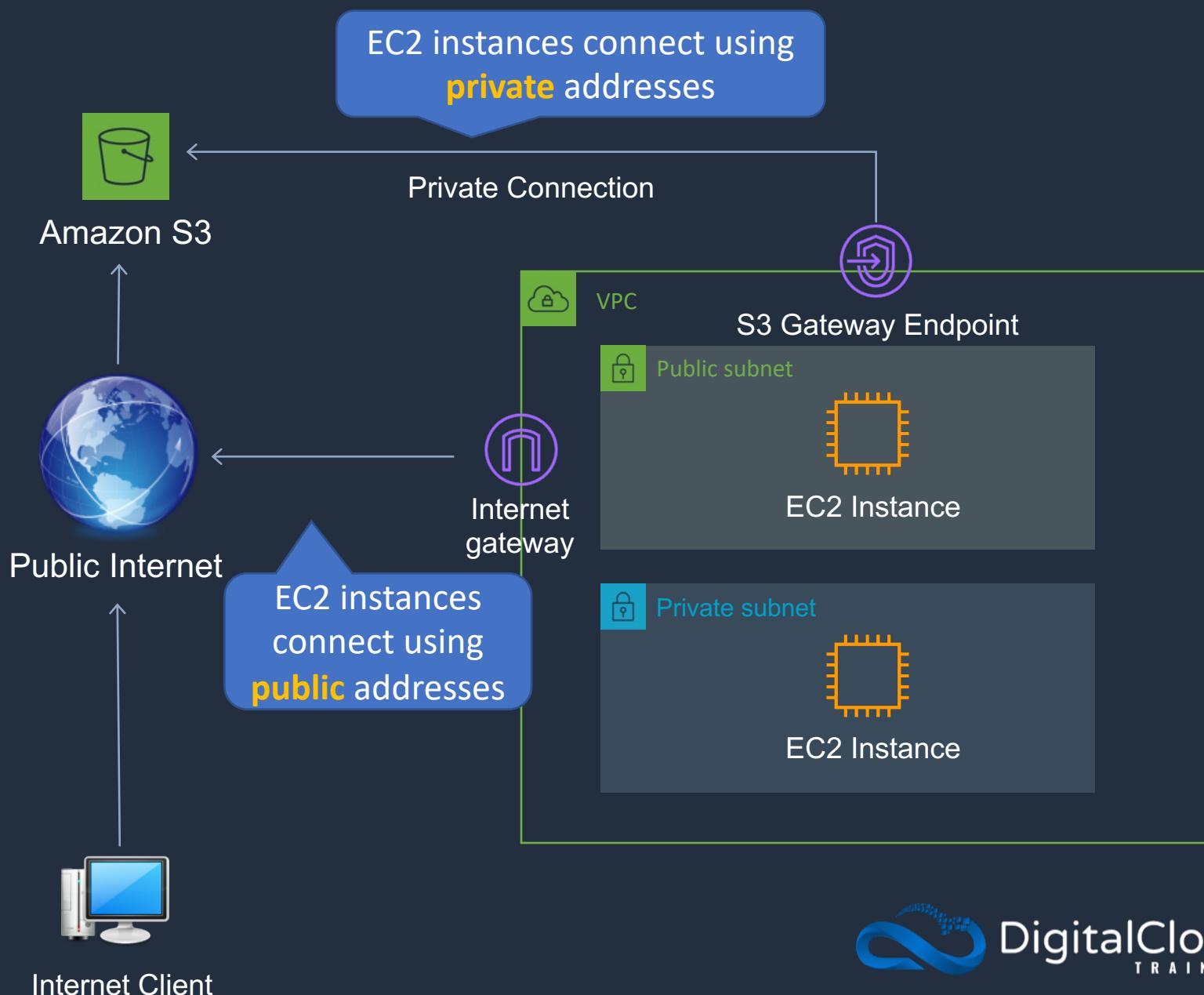
[http://s3.\*aws-region\*.amazonaws.com/\*bucket\*](http://s3.<i>aws-region</i>.amazonaws.com/<i>bucket</i>)



Object

An object consists of:

- Key (name of objects)
- Version ID
- Value (actual data)
- Metadata
- Subresources
- Access control information





# Amazon S3

---

- You can store any type of file in S3
- Files can be anywhere from 0 bytes to 5 TB
- There is unlimited storage available
- S3 is a universal namespace so bucket names must be unique globally
- However, you create your buckets within a REGION
- It is a best practice to create buckets in regions that are physically closest to your users to reduce latency



# Amazon S3 – Additional Features

---

---

S3 Capability	What it Does
Transfer Acceleration	Speed up data uploads using CloudFront in reverse
Requester Pays	The requester rather than the bucket owner pays for requests and data transfer
Events	Trigger notifications to SNS, SQS, or Lambda when certain events happen in your bucket
Static Web Hosting	Simple and massively scalable static website hosting
Versioning and Replication	Retain versions of objects and replicate objects within and across AWS Regions

# Amazon S3 Storage Classes





# Amazon S3 Availability and Durability

---

---

## Availability

- Measures how readily available the service is
- Measured as a percentage
- S3 availability SLA varies between storage classes

## Durability

- Measures the likelihood of data loss
- All storage classes offer 99.99999999% durability
- This means that if you store 100 billion objects in S3, you will lose one object at most



# Amazon S3 Storage Classes

	S3 Standard	S3 Intelligent Tiering	S3 Standard-IA	S3 One Zone-IA	S3 Glacier	S3 Glacier Deep Archive
<b>Designed for durability</b>	99.99999999%	99.99999999%	99.99999999%	99.99999999%	99.99999999%	99.99999999%
<b>Designed for availability</b>	99.99%	99.9%	99.9%	99.5%	99.99%	99.99%
<b>Availability SLA</b>	99.9%	99%	99%	99%	99.9%	99.9%
<b>Availability Zones</b>	≥3	≥3	≥3	1	≥3	≥3
<b>Minimum capacity charge per object</b>	N/A	N/A	128KB	128KB	40KB	40KB
<b>Minimum storage duration charge</b>	N/A	30 days	30 days	30 days	90 days	180 days
<b>Retrieval fee</b>	N/A	N/A	Per GB retrieved	Per GB retrieved	Per GB retrieved	Per GB retrieved
<b>First byte latency</b>	milliseconds	milliseconds	milliseconds	milliseconds	select minutes or hours	select hours
<b>Storage type</b>	Object	Object	Object	Object	Object	Object
<b>Lifecycle transitions</b>	Yes	Yes	Yes	Yes	Yes	Yes

# Create Amazon S3 Bucket



# S3 Versioning, Replication and Lifecycle Rules





# Amazon S3 Versioning

---

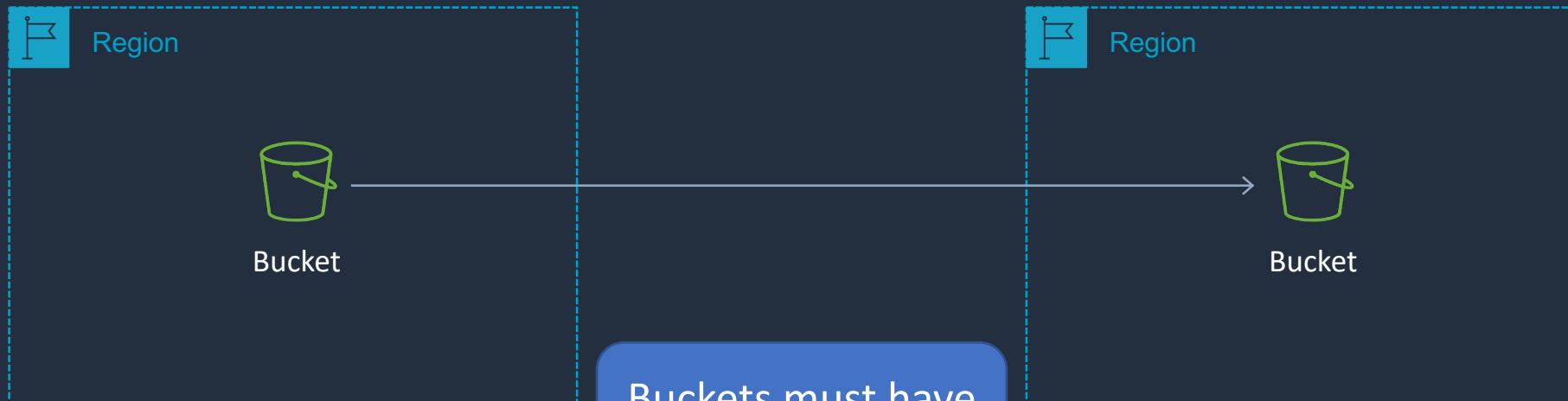
- Versioning is a means of keeping **multiple variants** of an **object** in the same bucket
- Use versioning to preserve, retrieve, and restore every version of every object stored in your Amazon S3 bucket
- Versioning-enabled buckets enable you to recover objects from accidental deletion or overwrite



# Amazon S3 Replication

---

## Cross-Region Replication (CRR)



Buckets must have  
**versioning** enabled



## Same-Region Replication (SRR)

# Configure Replication and Lifecycle



# Configure S3 Static Website



# S3 Permissions and Bucket Policies



# Archiving with S3 Glacier





# Amazon S3 Glacier

- Extremely low cost and you pay only for what you need with no commitments of upfront fees
- Two classes **Glacier** and **Glacier Deep Archive**
- Three options for access to archives, listed in the table below:

	Expedited	Standard	Bulk
Data access time (Glacier)	1-5 minutes	3-5 hours	5-12 hours
Data access time (Deep Archive)	N/A	12 hours	48 hours



# Object Lock and Glacier Vault Lock

## S3 Object Lock

- Store objects using a write-once-read-many (WORM) model
- Prevent objects from being deleted or overwritten for a fixed time or indefinitely

## S3 Glacier Vault Lock

- Also used to enforce a WORM model
- Can apply a policy and lock the policy from future edits
- Use for compliance objectives and data retention

# AWS Storage Gateway





# AWS Storage Gateway

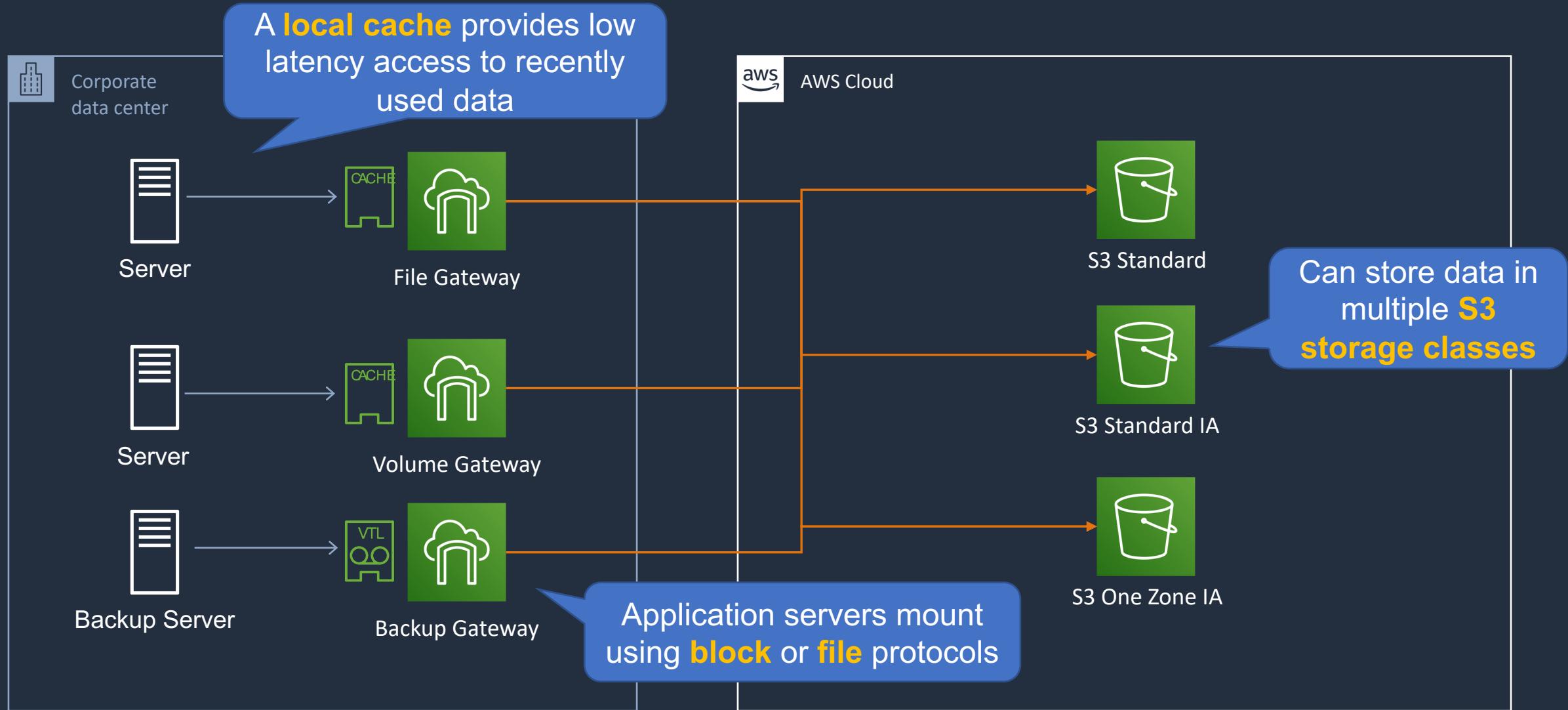
---

---

- Hybrid cloud storage service
- Access cloud storage from on-premises applications
- Enables access to proprietary object storage (S3) using standard protocols
- Use cases:
  - Moving backups to the cloud
  - Using on-premises file shares backed by cloud storage
  - Low latency access to data in AWS for on-premises applications
  - Disaster recovery



# AWS Storage Gateway



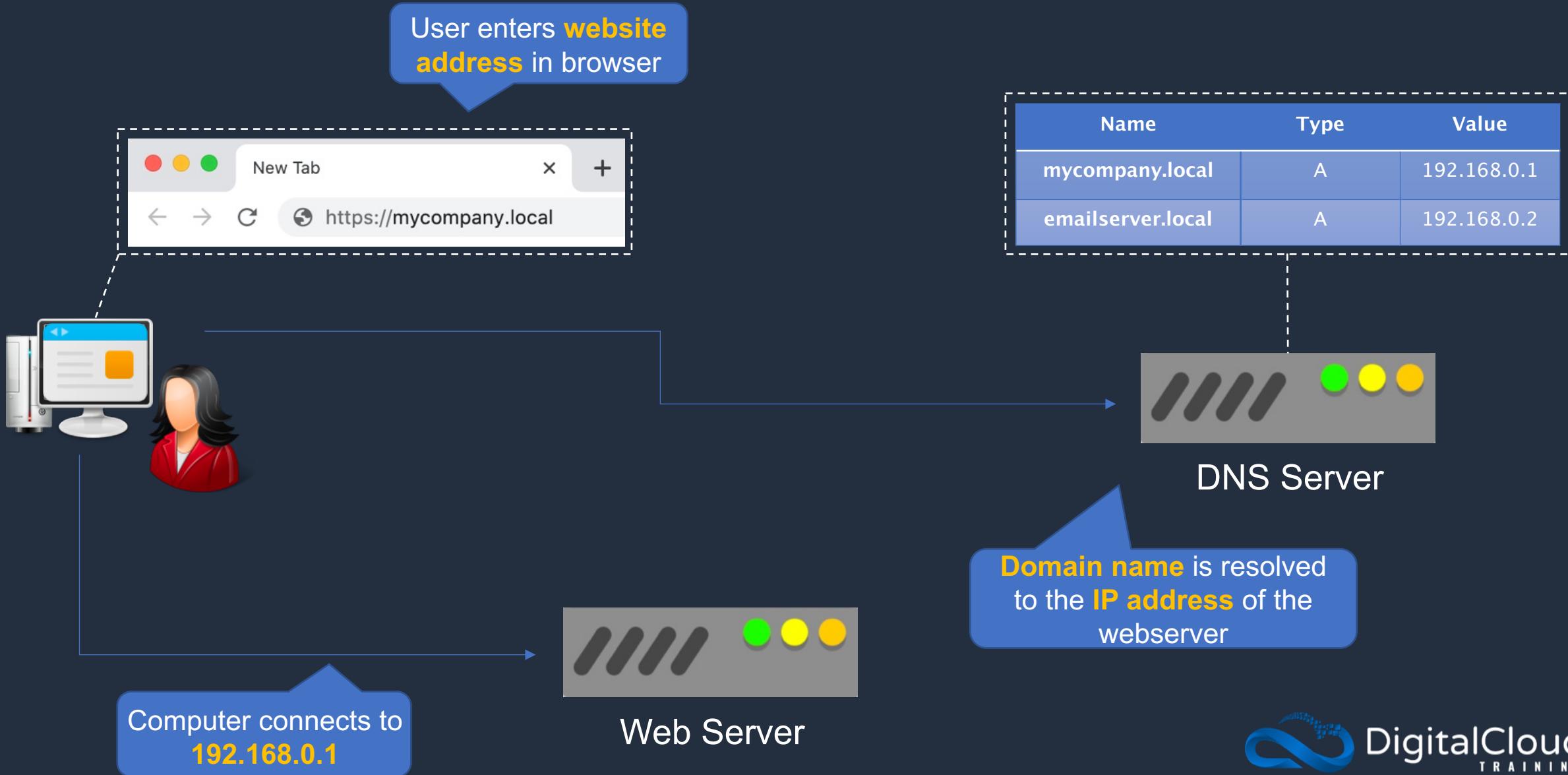
# SECTION 7

## DNS, Elastic Load Balancing, and Auto Scaling

# DNS and Amazon Route 53

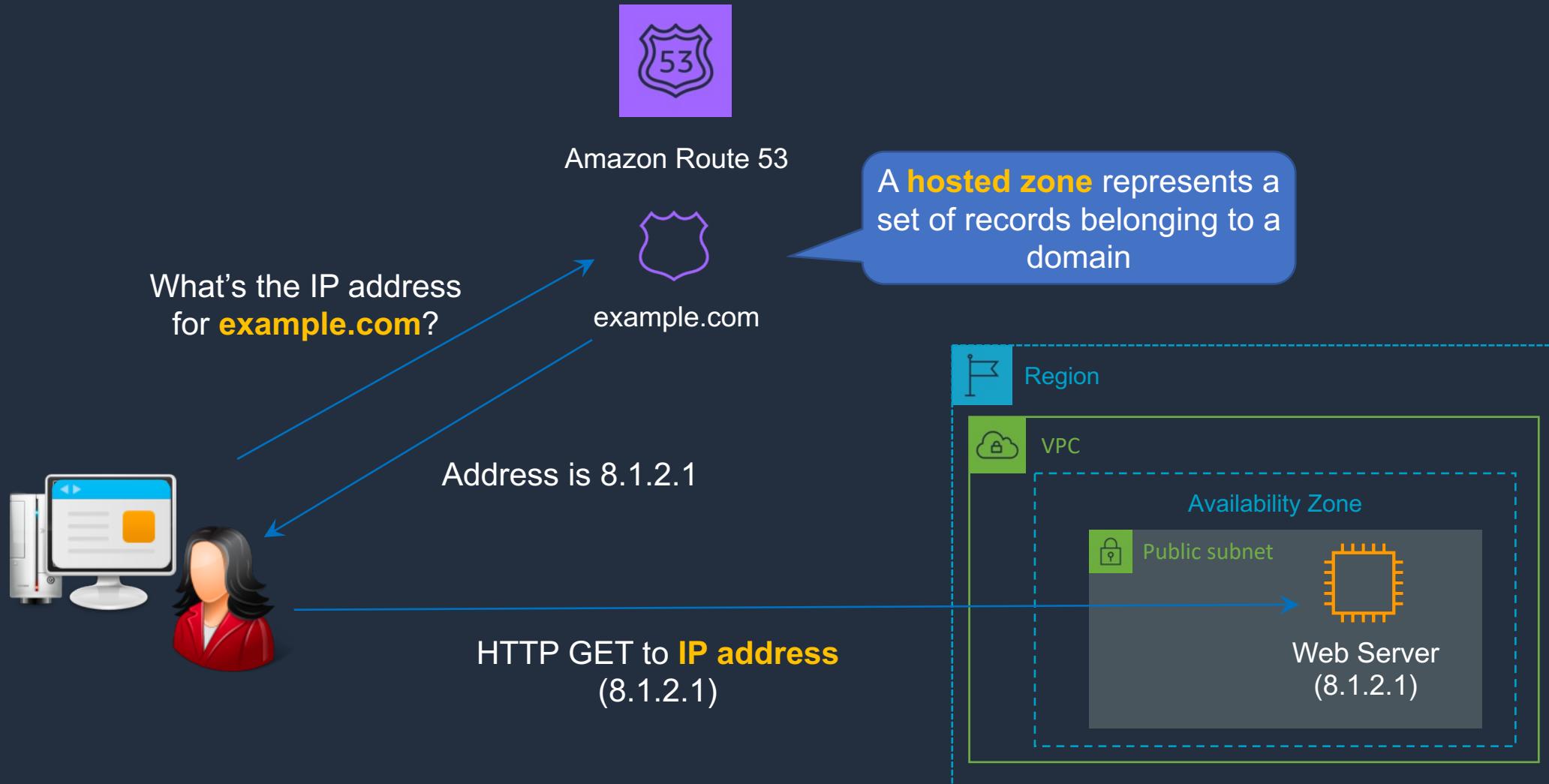


# The Domain Name System (DNS)





# Amazon Route 53



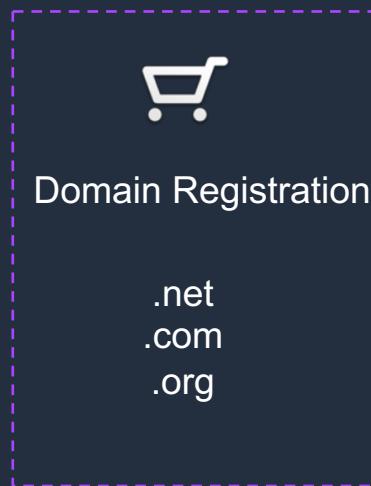
# Amazon Route 53 Routing Policies

Routing Policy	What it does
<b>Simple</b>	Simple DNS response providing the IP address associated with a name
<b>Failover</b>	If primary is down (based on health checks), routes to secondary destination
<b>Geolocation</b>	Uses geographic location you're in (e.g. Europe) to route you to the closest region
<b>Geoproximity</b>	Routes you to the closest region within a geographic area
<b>Latency</b>	Directs you based on the lowest latency route to resources
<b>Multivalue answer</b>	Returns several IP addresses and functions as a basic load balancer
<b>Weighted</b>	Uses the relative weights assigned to resources to determine which to route to

# Amazon Route Features



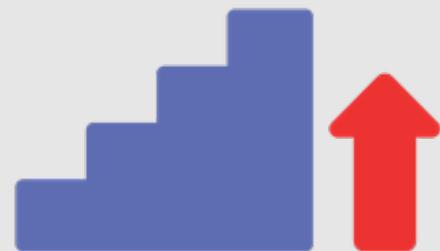
Amazon Route 53



# Register Domain with Route 53 (Optional)



# Elasticity: Scaling Up vs Out





# Scaling Up (vertical scaling)

---

---





# Scaling Up (vertical scaling)

Scaling up means  
**adding** resources  
to the instance



Limitation is that you have a **single point of failure** (SPOF)



# Scaling Out (horizontal scaling)

Scaling out provides greater **resiliency**

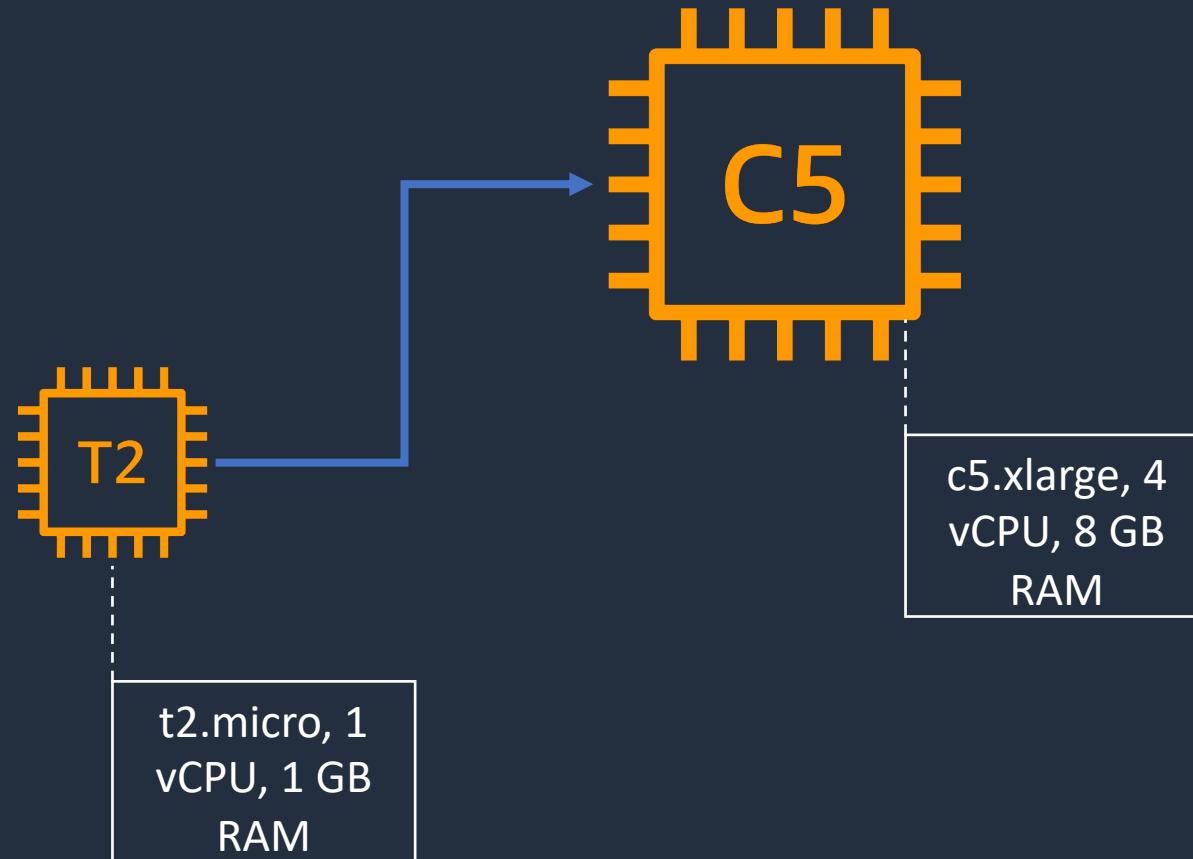


Scaling out can be used to add almost unlimited capacity



# Scaling Up (vertical scaling)

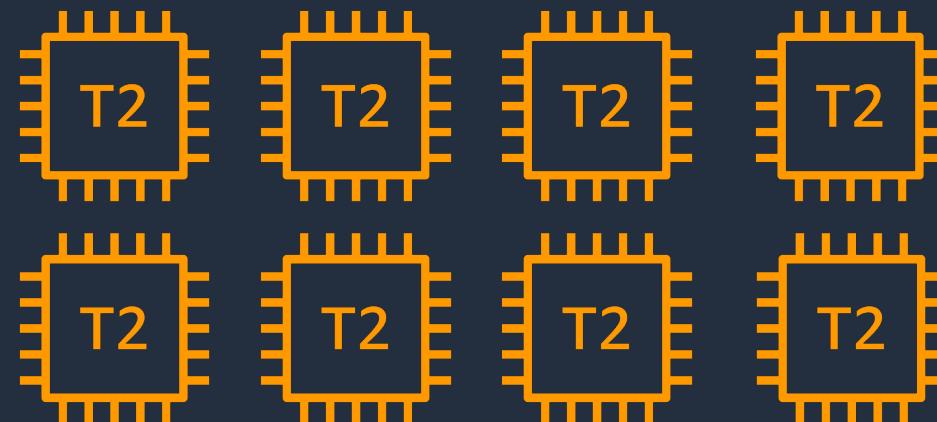
---



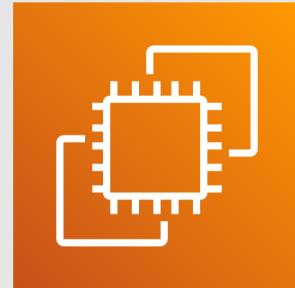


# Scaling Out (horizontal scaling)

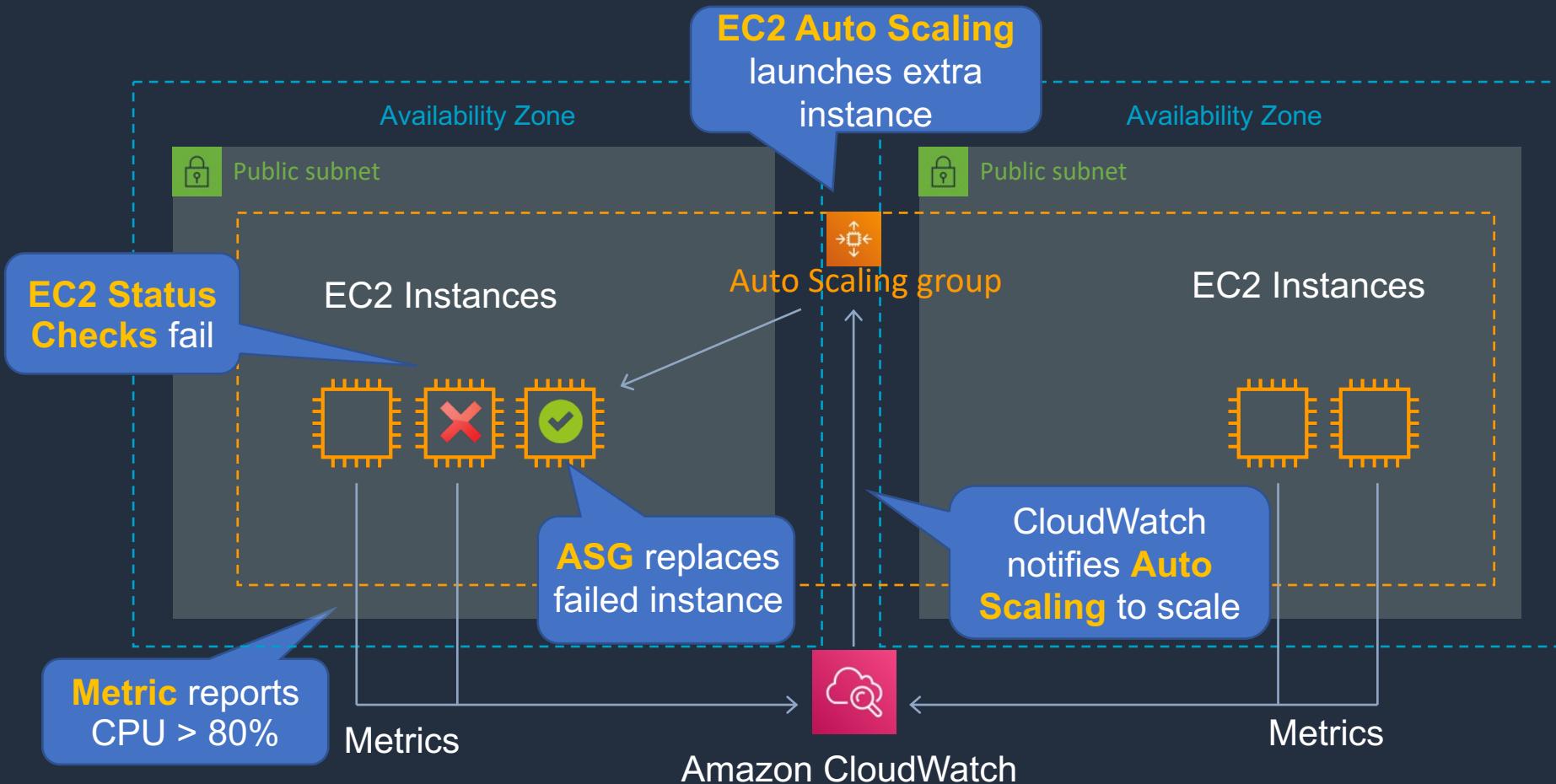
---



# Amazon EC2 Auto Scaling



# Amazon EC2 Auto Scaling





# Amazon EC2 Auto Scaling

---

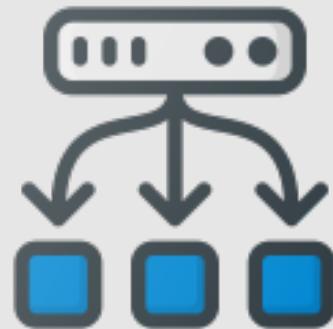
---

- EC2 Auto Scaling **launches** and **terminates** instances dynamically
- Scaling is horizontal (scales out)
- Provides **elasticity** and **scalability**
- Responds to EC2 status checks and CloudWatch metrics
- Can scale based on demand (performance) or on a schedule
- Scaling policies define how to respond to changes in demand

# Create an Auto Scaling Group

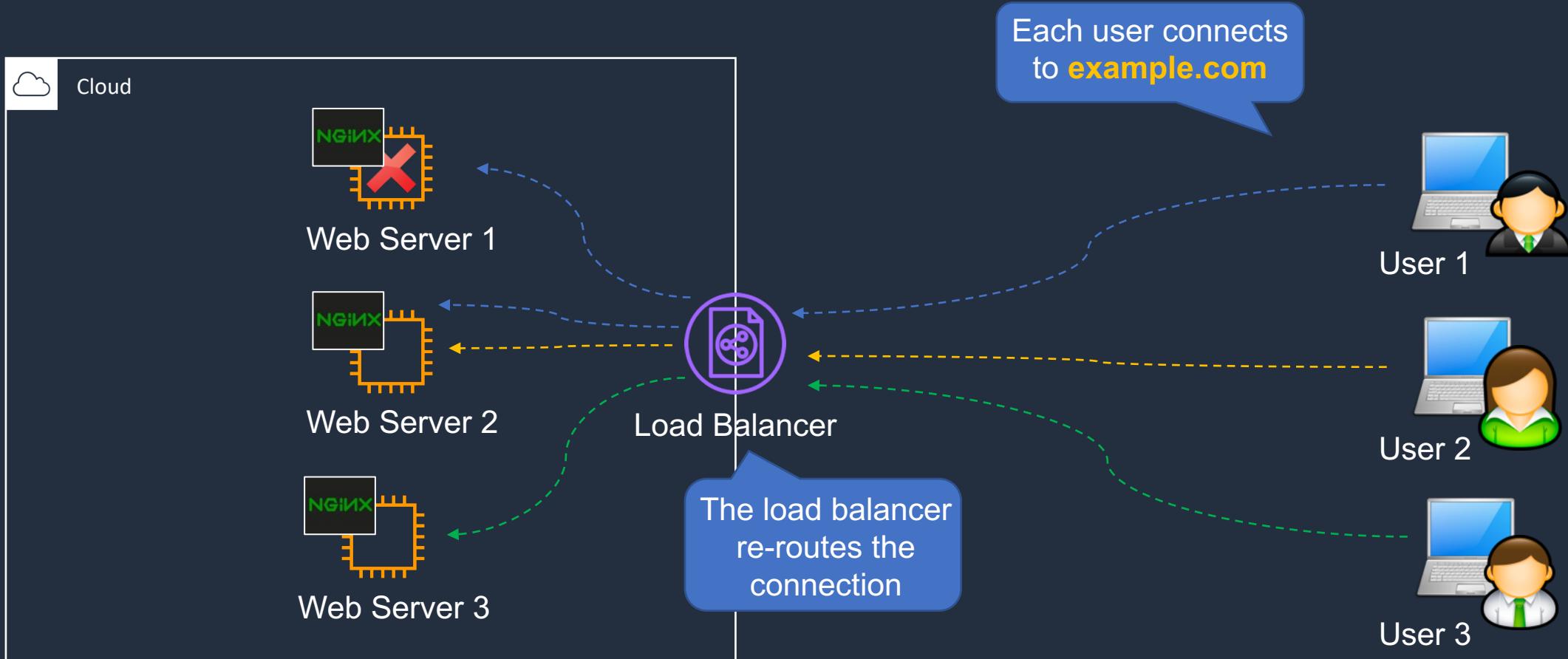


# Load Balancing and High Availability





# Load Balancing and High Availability





# Fault Tolerance

---

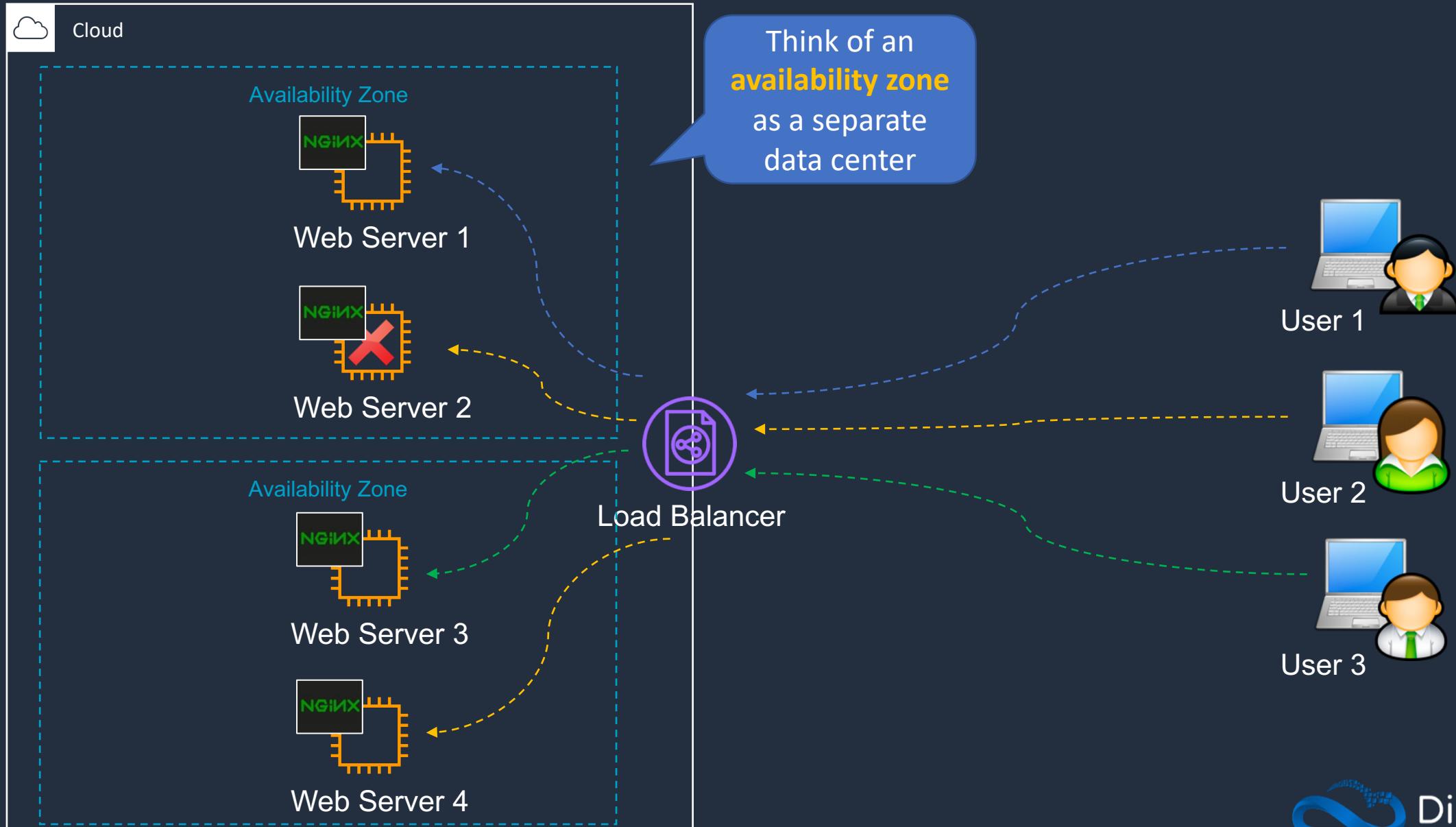
**Redundant** components allow the system to continue to operate



The system may fail if there is no built-in **redundancy**

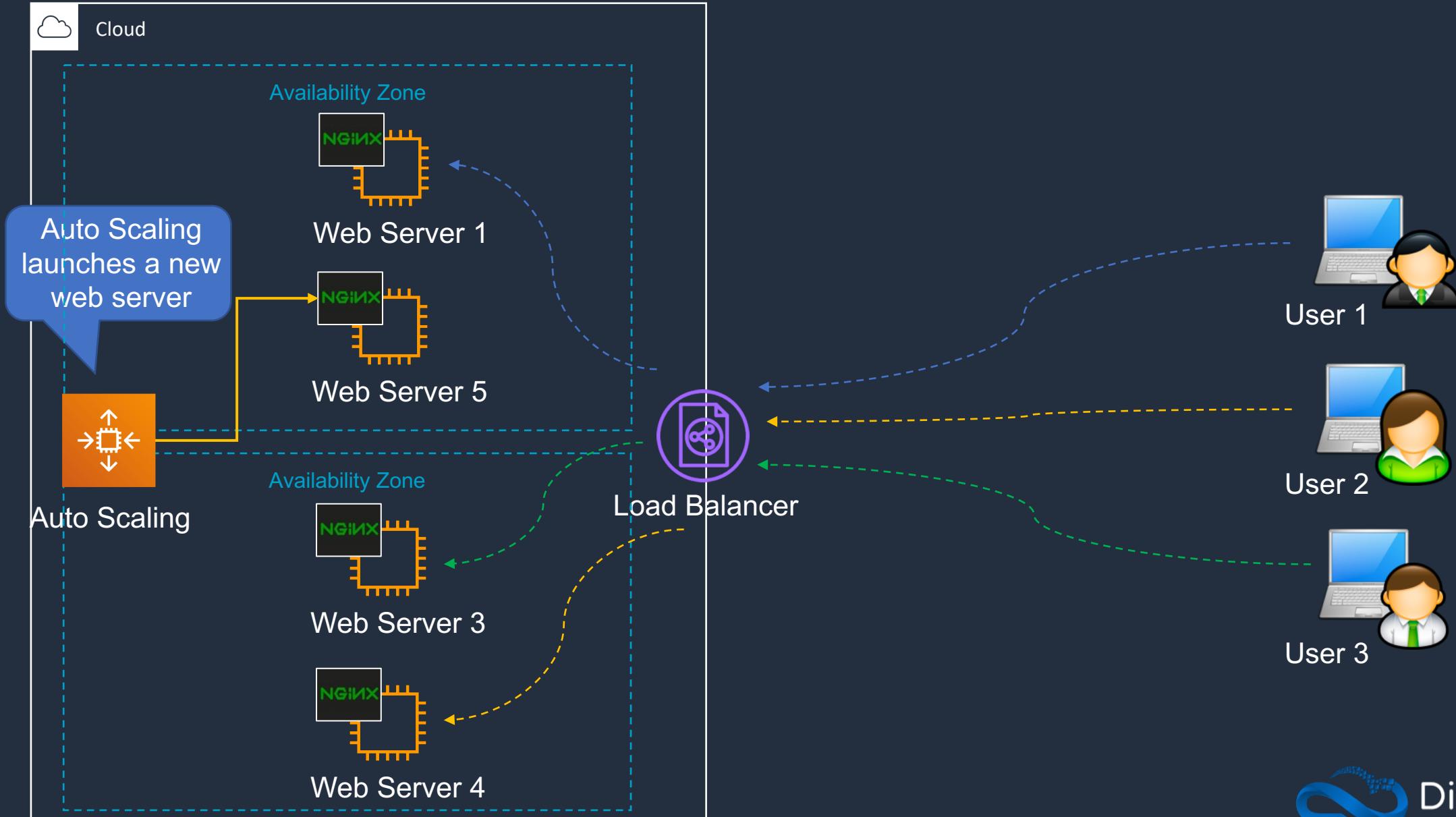


# High Availability and Fault Tolerance



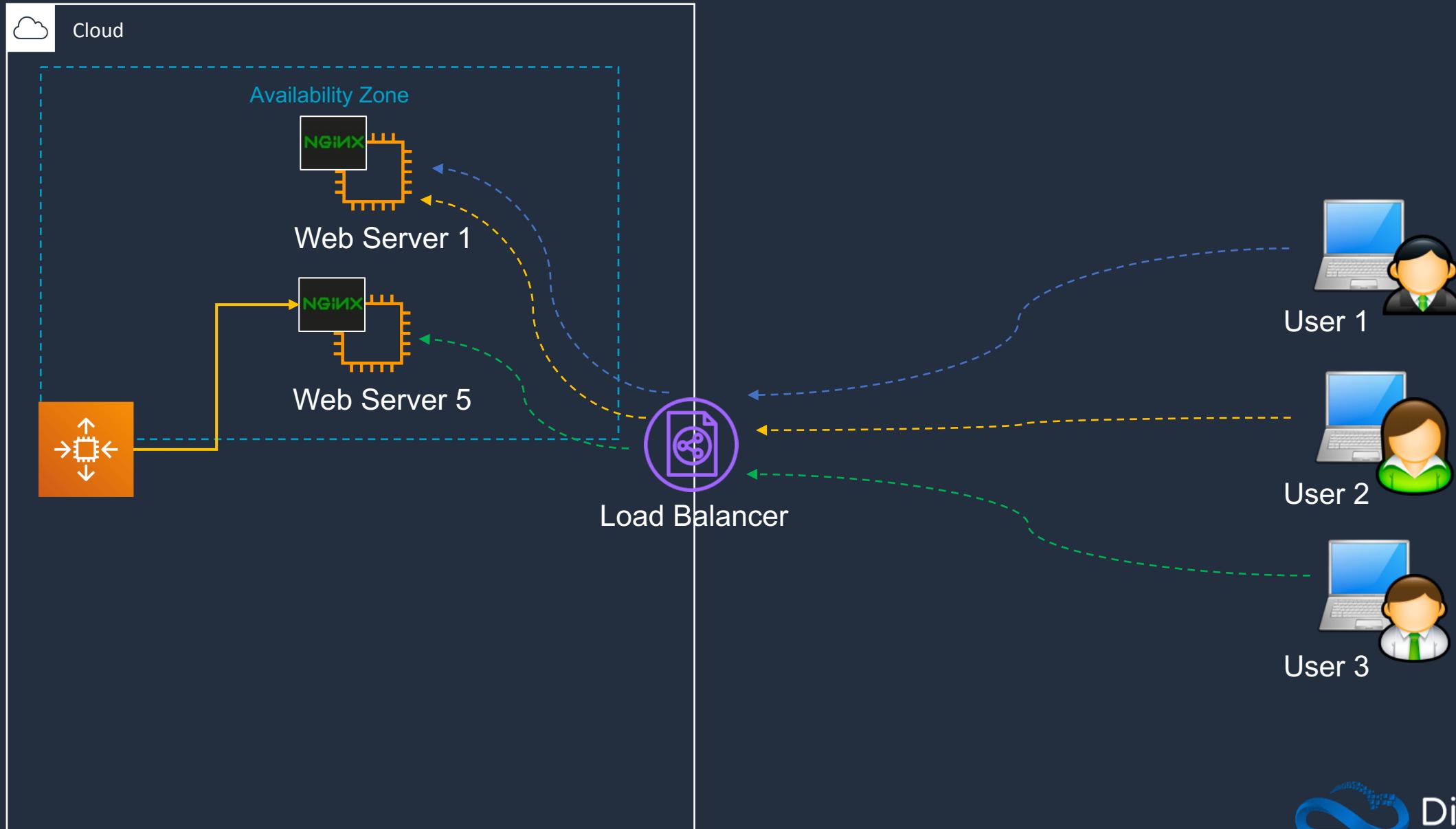


# High Availability and Fault Tolerance

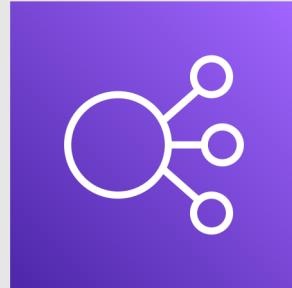




# High Availability and Fault Tolerance

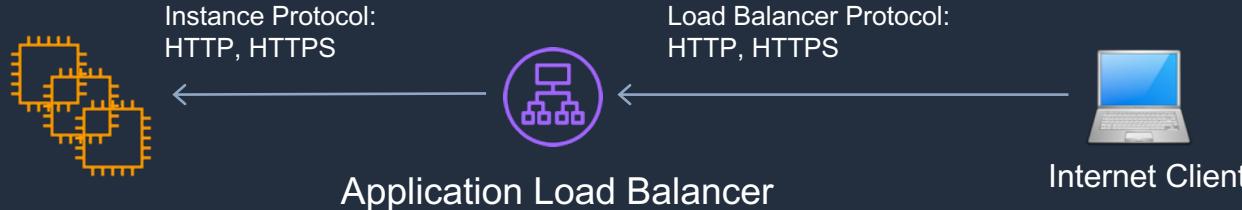


# Amazon Elastic Load Balancer (ELB)



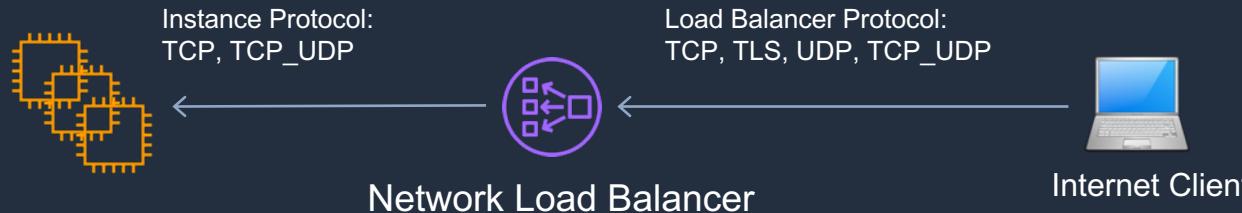


# Types of Elastic Load Balancer (ELB)



## Application Load Balancer

- Operates at the request level
- Routes based on the content of the request (layer 7)
- Supports advanced routing



## Network Load Balancer

- Operates at the connection level
- Routes connections based on IP protocol data (layer 4)
- Offers ultra high performance, low latency and TLS offloading at scale



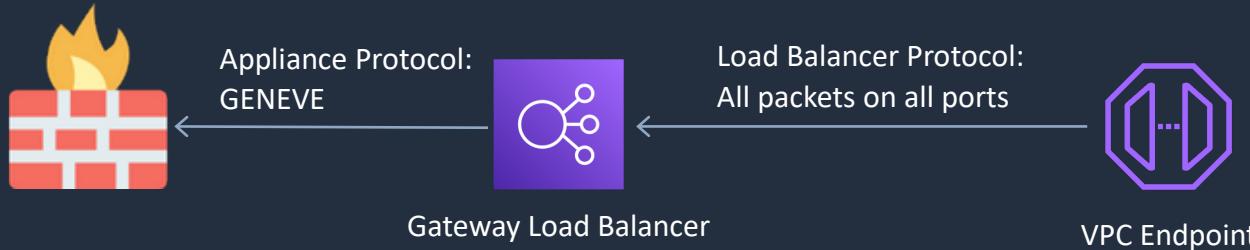
# Types of Elastic Load Balancer (ELB)

Old and **shouldn't** be  
the exam anymore



## Classic Load Balancer

- Old generation; not recommended for new applications
- Performs routing at Layer 4 and Layer 7
- Use for existing applications running in EC2-Classic



## Gateway Load Balancer

- Used in front of virtual appliances such as firewalls, IDS/IPS, and deep packet inspection systems

New and **not** yet  
on the exam

# Attach an Application Load Balancer

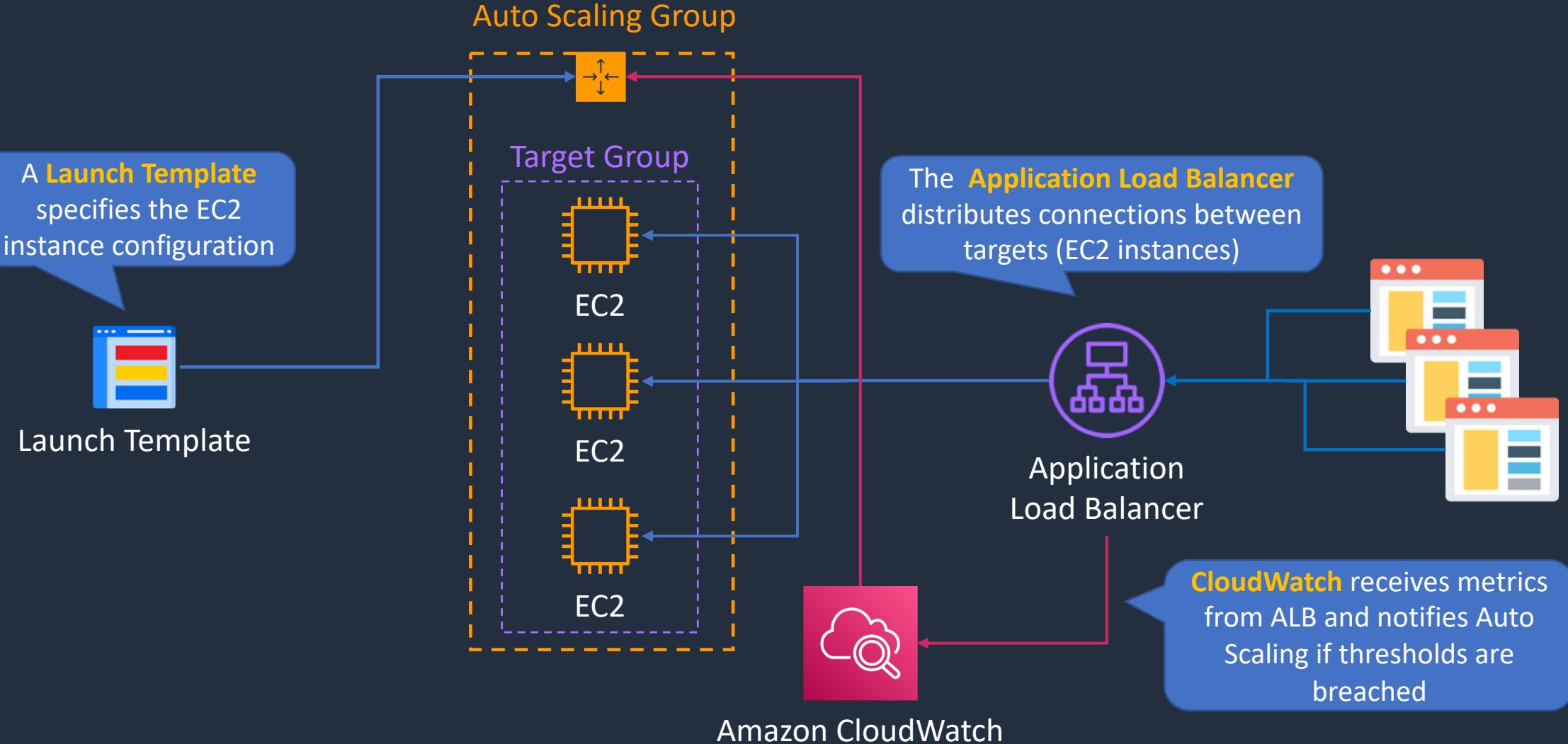


# Elastically Scale the Application

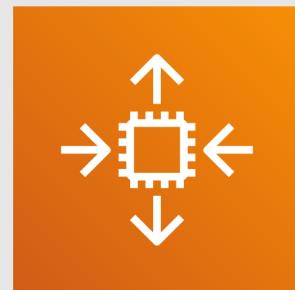




# Elastically Scale the Application



# Scaling Policies





# Scaling Policies

---

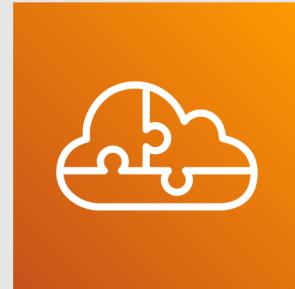
---

- **Target Tracking** – Attempts to keep the group at or close to the metric
- **Simple Scaling** – Adjust group size based on a metric
- **Step Scaling** – Adjust group size based on a metric – adjustments vary based on the size of the alarm breach
- **Scheduled Scaling** – Adjust the group size at a specific time

# SECTION 8

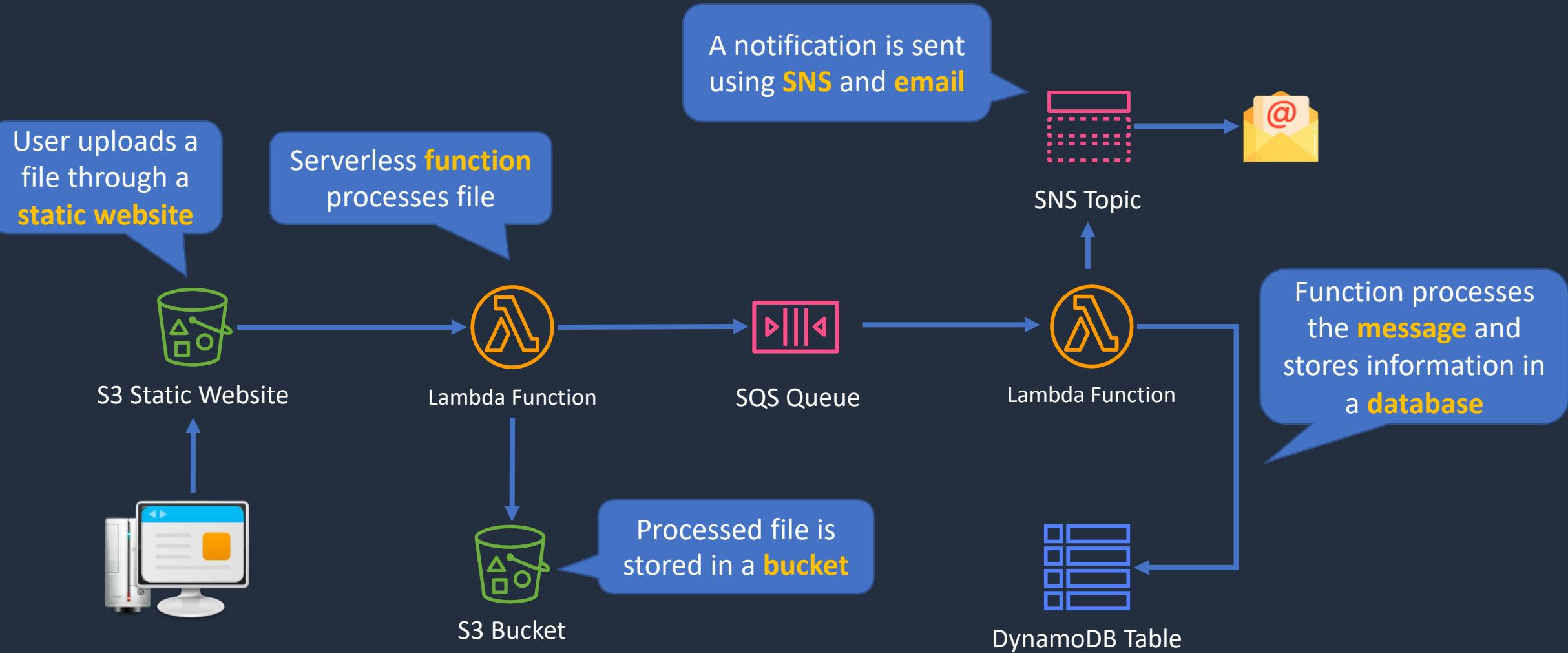
## Application Services

# Serverless Services





# Serverless Services





# Serverless Services

---

- With serverless there are **no instances** to manage
- You don't need to provision hardware
- There is no management of operating systems or software
- Capacity provisioning and patching is handled automatically
- Provides automatic scaling and high availability
- Can be very cheap!



# Serverless Services

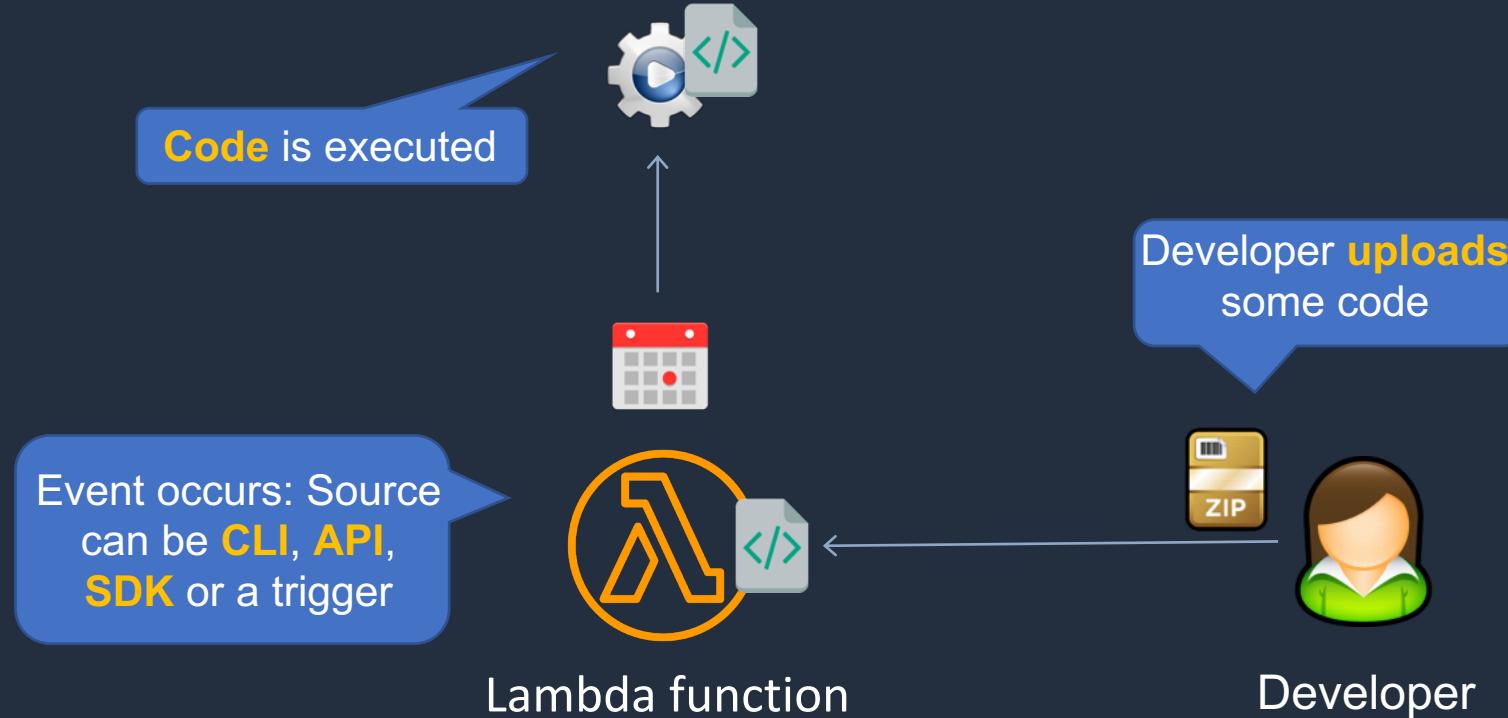
---

- Serverless services include:
  - AWS Lambda
  - AWS Fargate
  - Amazon EventBridge
  - AWS Step Functions
  - Amazon SQS
  - Amazon SNS
  - Amazon API Gateway
  - Amazon S3
  - Amazon DynamoDB

# AWS Lambda Functions



# AWS Lambda Functions





# AWS Lambda Functions

---

- AWS Lambda executes code only when needed and scales automatically
- You pay only for the compute time you consume (you pay nothing when your code is not running)
- Benefits of AWS Lambda:
  - No servers to manage
  - Continuous scaling
  - Millisecond billing
  - Integrates with almost all other AWS services



# AWS Lambda Functions

---

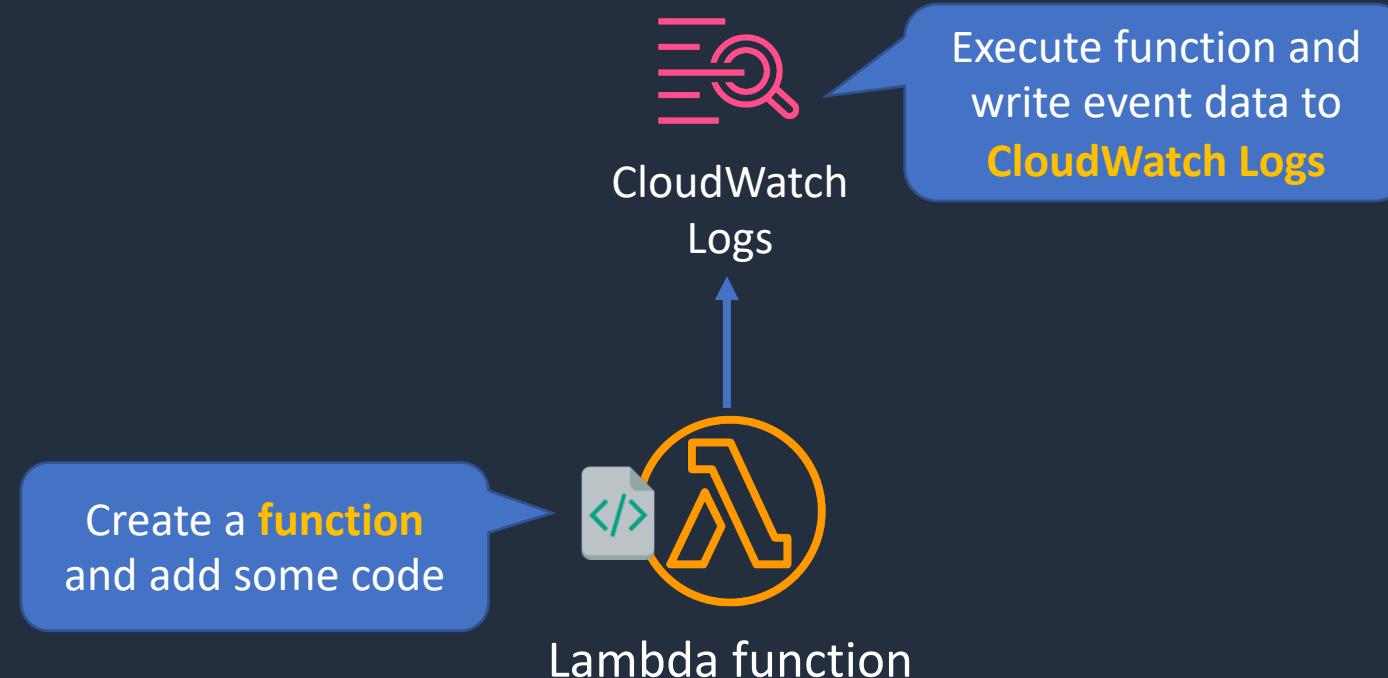
- Primary use cases for AWS Lambda:
  - Data processing
  - Real-time file processing
  - Real-time stream processing
  - Build serverless backends for web, mobile, IOT, and 3rd party API requests

# Create a Simple Lambda Function





# Create a Simple Lambda Function

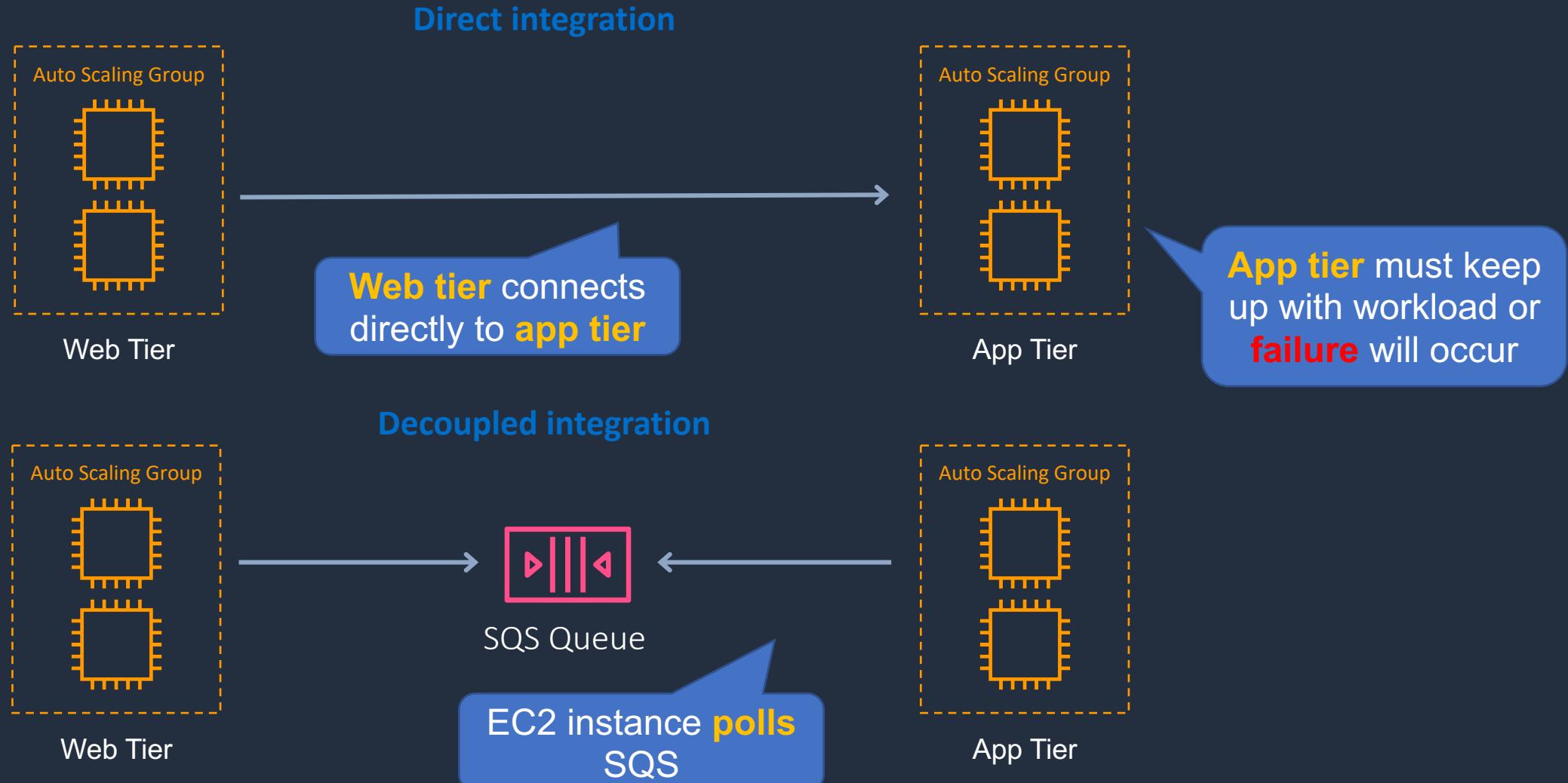


# Application Integration Services





# Amazon Simple Queue Service (SQS)





# Amazon SQS

---

---

- SQS offers a reliable, highly-scalable, hosted queue for storing messages in transit between computers
- SQS is used for distributed/decoupled applications
- SQS uses a message-oriented API
- SQS uses pull based (polling) not push based



# Amazon MQ

---

---

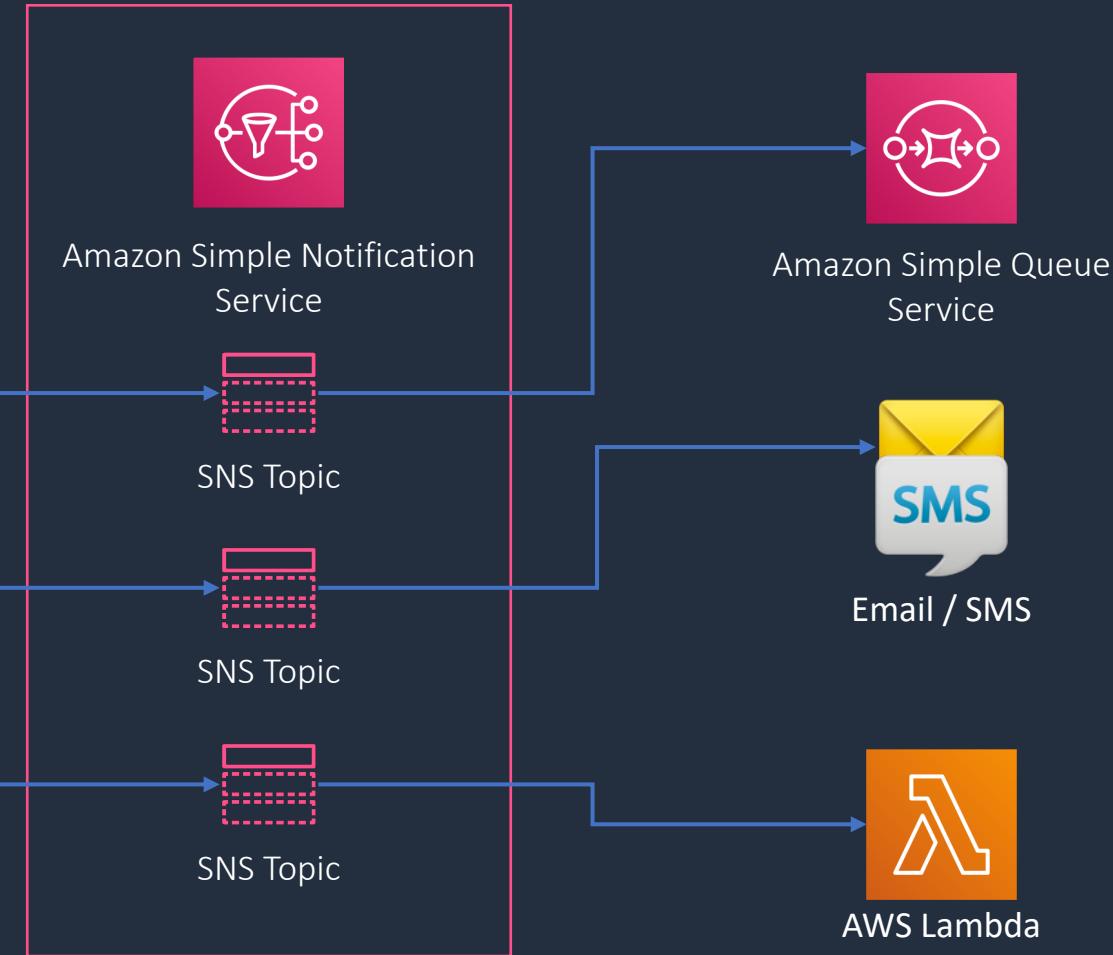
- Message broker service
- Similar to Amazon SQS
- Based on Apache Active MQ and RabbitMQ
- Used when customers require industry standard APIs and protocols
- Useful when migrating existing queue-based applications into the cloud



# Amazon Simple Notification Service (SNS)

## PUBLISHERS

- Amazon EC2
- Amazon CloudWatch
- Amazon Simple Storage Service



## SUBSCRIBERS



# Amazon SNS

---

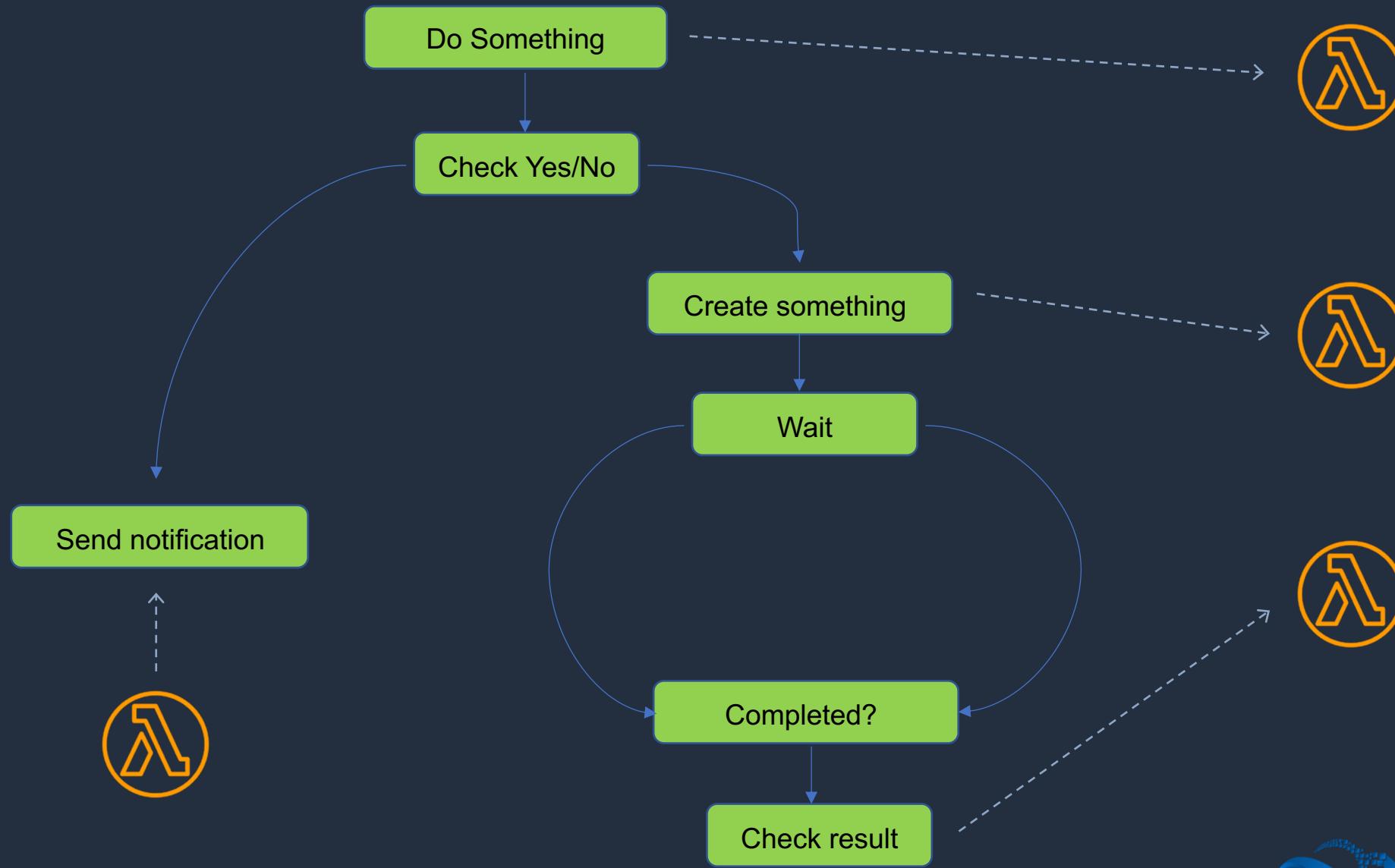
---

- Amazon SNS is used for building and integrating loosely-coupled, distributed applications
- Provides instantaneous, push-based delivery (no polling)
- Uses simple APIs and easy integration with applications
- Offered under an inexpensive, pay-as-you-go model with no up-front costs



# AWS Step Functions

---





# AWS Step Functions

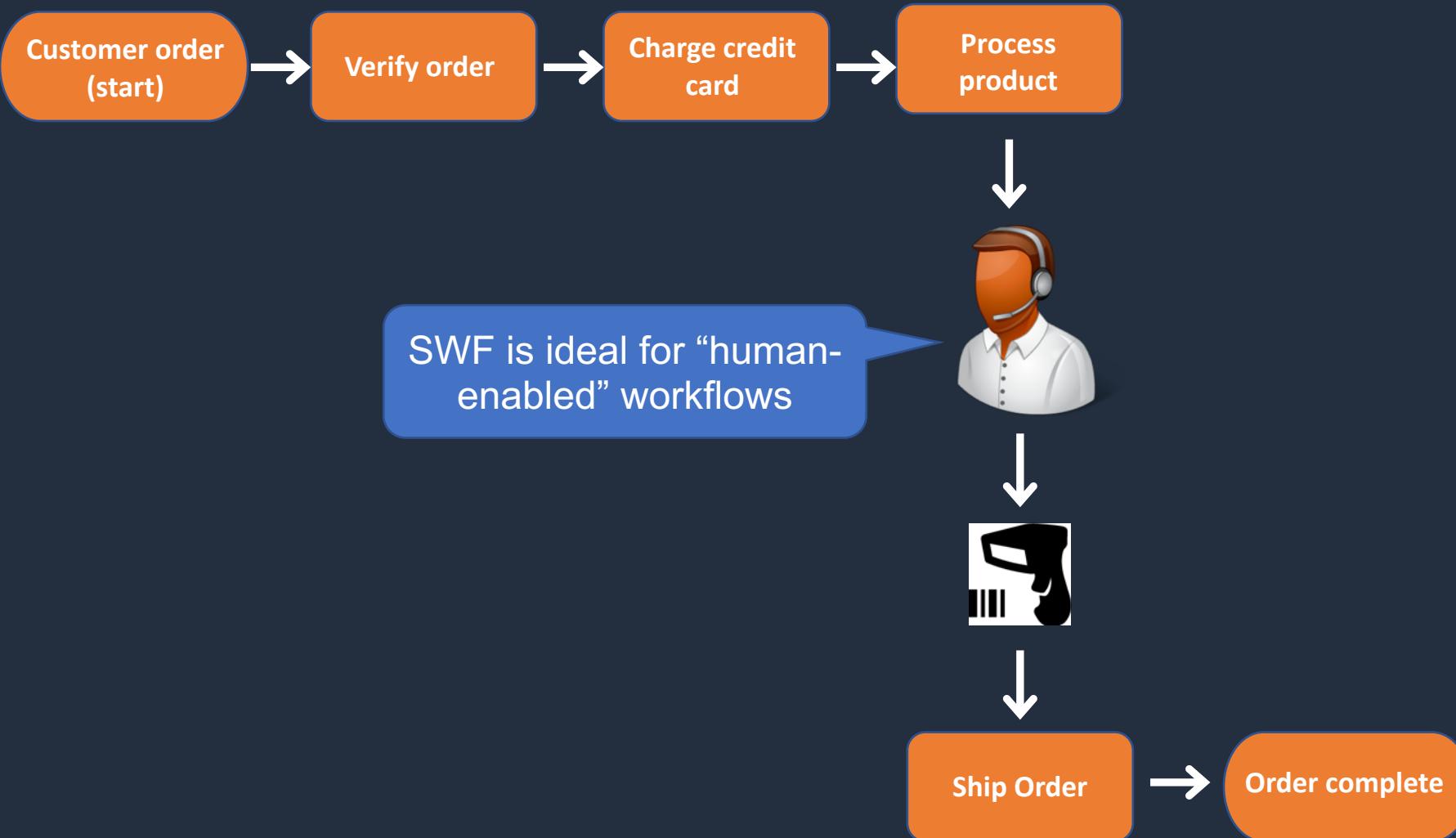
---

---

- AWS Step Functions makes it easy to coordinate the components of distributed applications as a series of steps in a visual workflow
- You can quickly build and run state machines to execute the steps of your application in a reliable and scalable fashion



# AWS Simple Workflow Service (SWF)





# Amazon SWF

---

- Amazon Simple Workflow Service (SWF) is a web service that makes it easy to coordinate work across distributed application components
- Create distributed asynchronous systems as workflows
- Best suited for human-enabled workflows like an order fulfilment system or for procedural requests
- AWS recommends that for new applications customers consider Step Functions instead of SWF



# Application Integration Services Comparison

Service	What it does	Example use cases
Simple Queue Service	Messaging queue; store and forward patterns	Building distributed / decoupled applications
Simple Notification Service	Set up, operate, and send notifications from the cloud	Send email notification when CloudWatch alarm is triggered
Step Functions	Out-of-the-box coordination of AWS service components with visual workflow	Order processing workflow
Simple Workflow Service	Need to support external processes or specialized execution logic	Human-enabled workflows like an order fulfilment system or for procedural requests  Note: AWS recommends that for new applications customers consider Step Functions instead of SWF
Amazon MQ	Message broker service for Apache Active MQ and RabbitMQ	Need a message queue that supports industry standard APIs and protocols; migrate queues to AWS

# Amazon EventBridge / CloudWatch Events



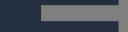
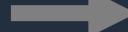


# Amazon EventBridge

EventBridge used to be known as **CloudWatch Events**

## Event Source

EC2 instance  
**terminated** event

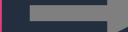


EventBridge  
event bus

## Event

## Rule

Send **SNS** notification



## Target



# Amazon EventBridge

**Event matching pattern**  
You can use pre-defined pattern provided by a service or create a custom pattern

Pre-defined pattern by service  
 Custom pattern

**Service provider**  
AWS services or custom/partner services

AWS

**Service name**  
The name of partner service selected as the event source

EC2

**Event type**  
The type of events as the source of the matching pattern

EC2 Instance State-change Notification

Any state  
 Specific state(s)

terminated X

Any instance  
 Specific instance Id(s)

i-1234567890abcdef0

**Event pattern**

Copy Edit

```
1 {
2   "source": ["aws.ec2"],
3   "detail-type": ["EC2 Instance State-change N
4   "detail": {
5     "state": ["terminated"],
6     "instance-id": ["i-1234567890abcdef0"]
7   }
8 }
```

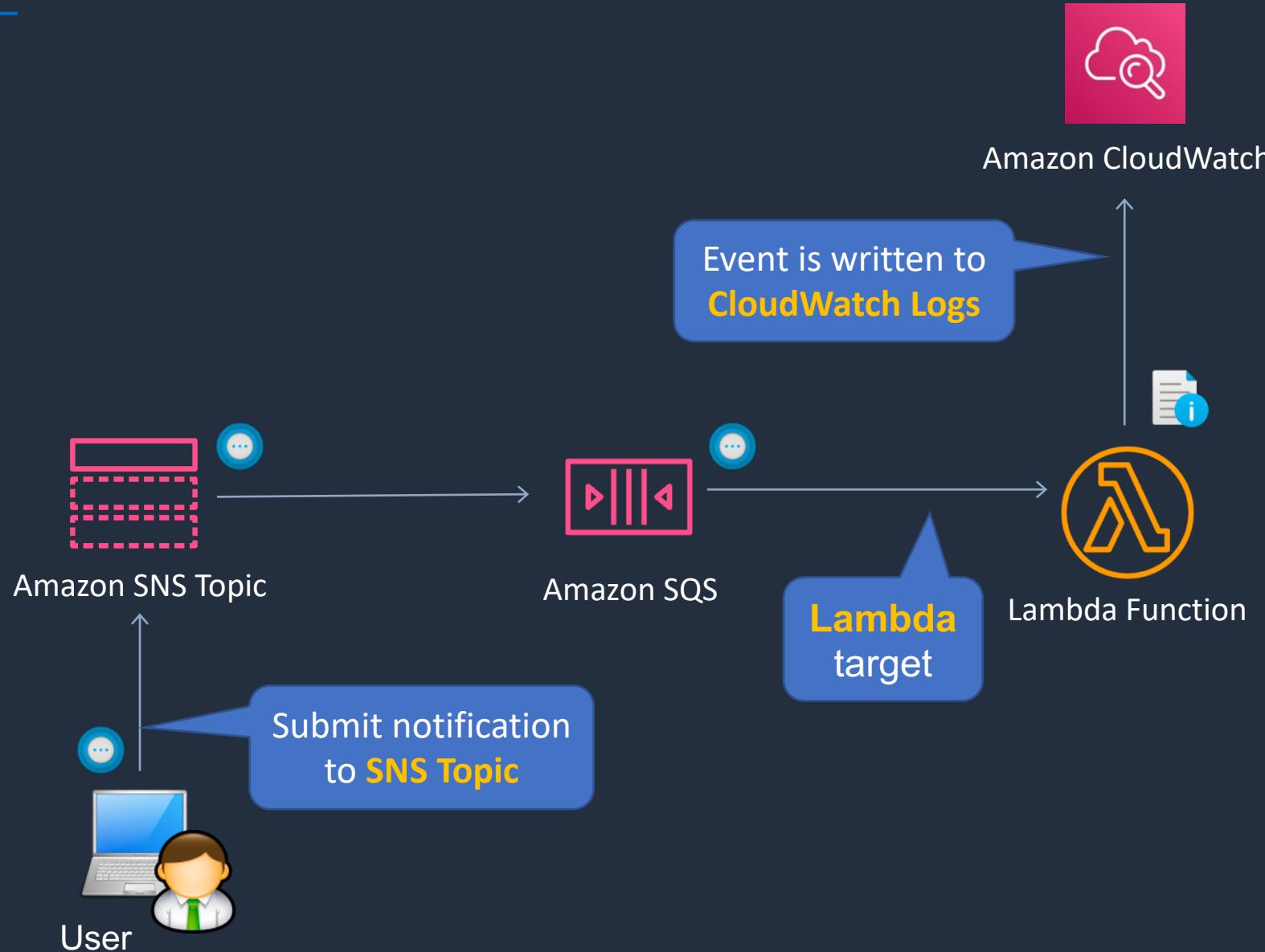
```
{
  "version": "0",
  "id": "6a7e8feb-b491-4cf7-a9f1-bf3703467718",
  "detail-type": "EC2 Instance State-change Notification",
  "source": "aws.ec2",
  "account": "111122223333",
  "time": "2017-12-22T18:43:48Z",
  "region": "us-west-1",
  "resources": [
    "arn:aws:ec2:us-west-1:123456789012:instance/i-1234567890abcdef0"
  ],
  "detail": {
    "instance-id": "i-1234567890abcdef0",
    "state": "terminated"
  }
}
```

# Create an Event-Driven Application

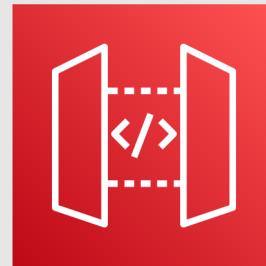




# Simple Event-Driven Application

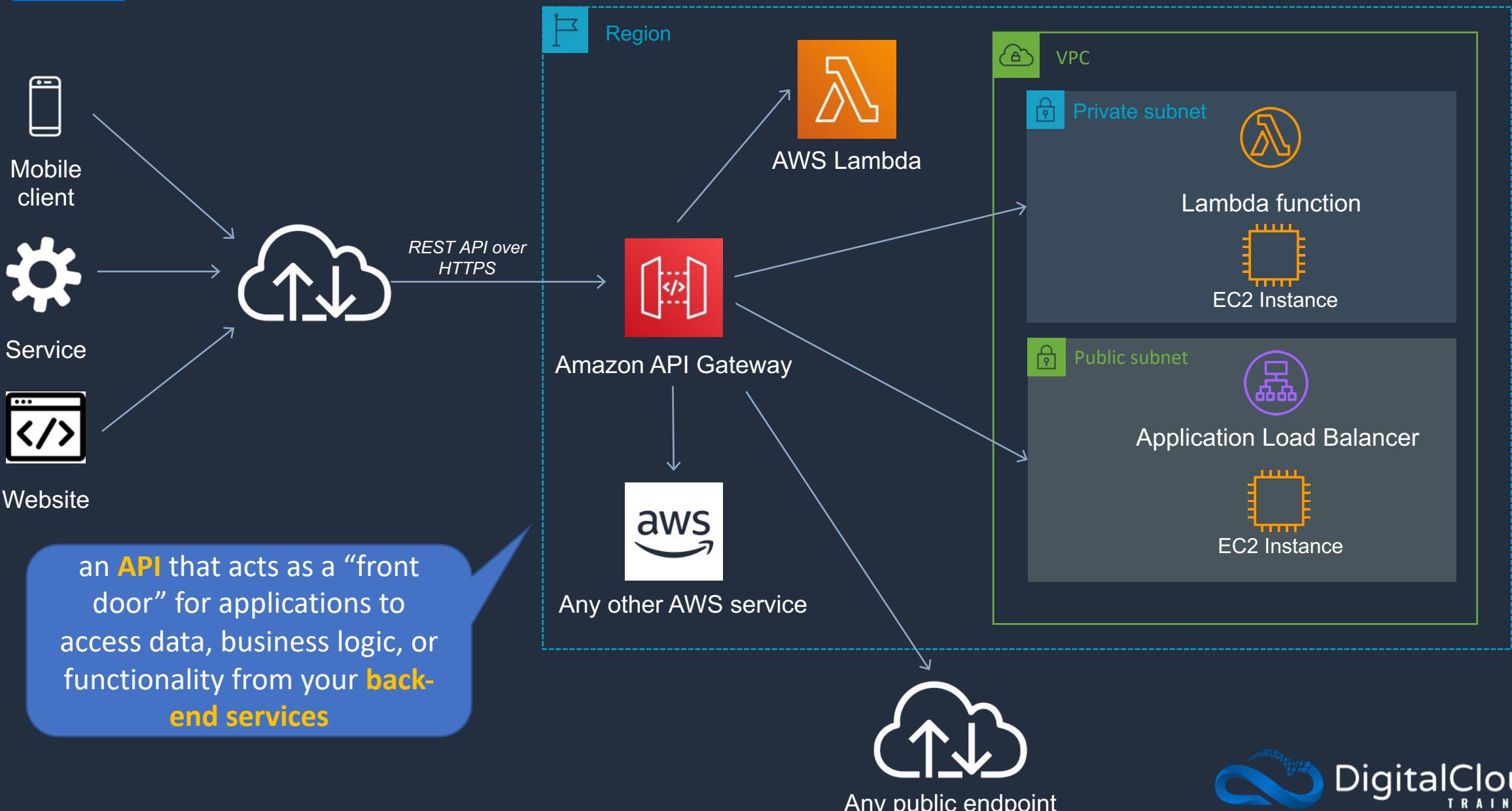


# Amazon API Gateway





# Amazon API Gateway



# SECTION 9

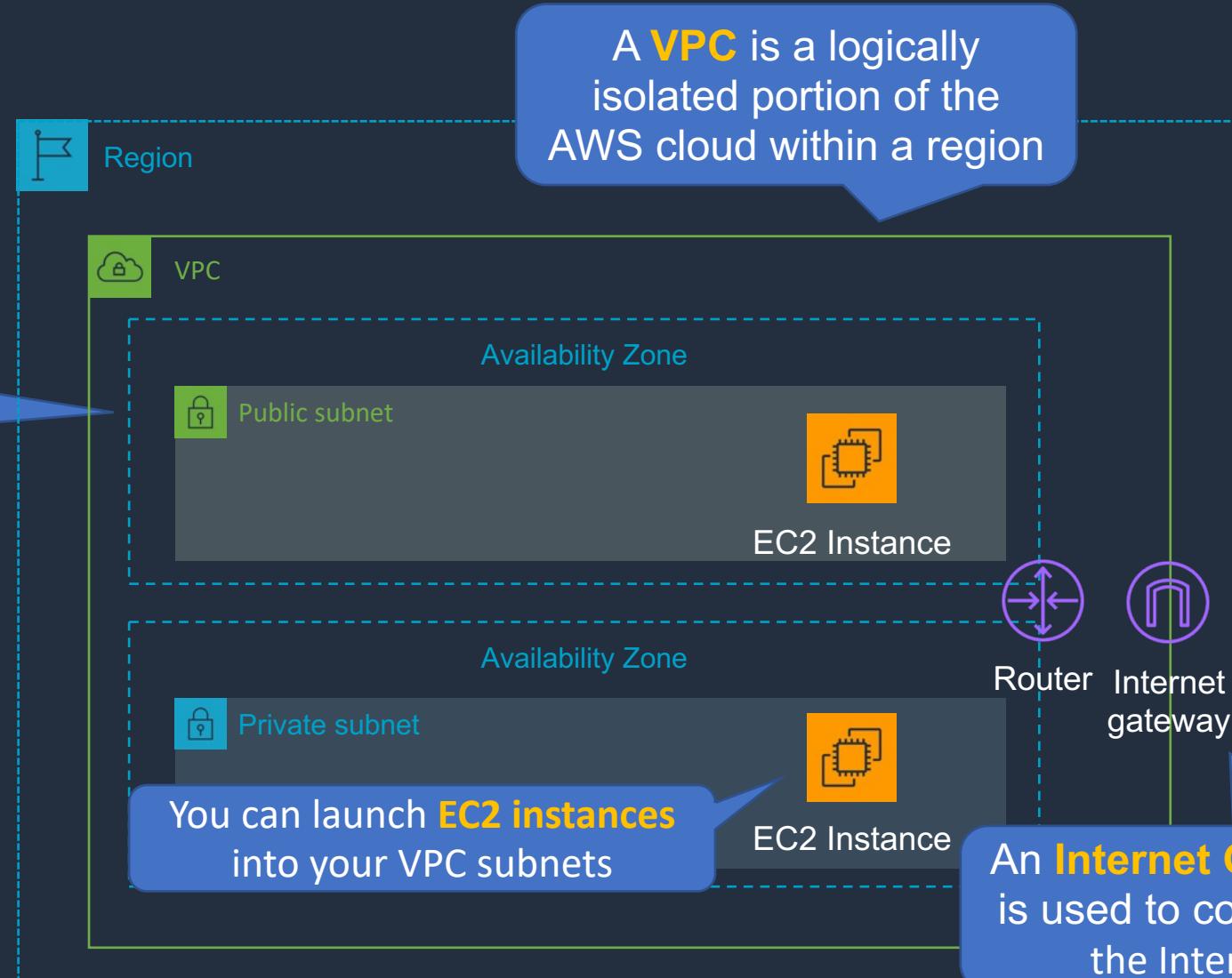
## Amazon VPC, Networking, and Hybrid

# Amazon Virtual Private Cloud (VPC)





# Amazon VPC





# Amazon VPC



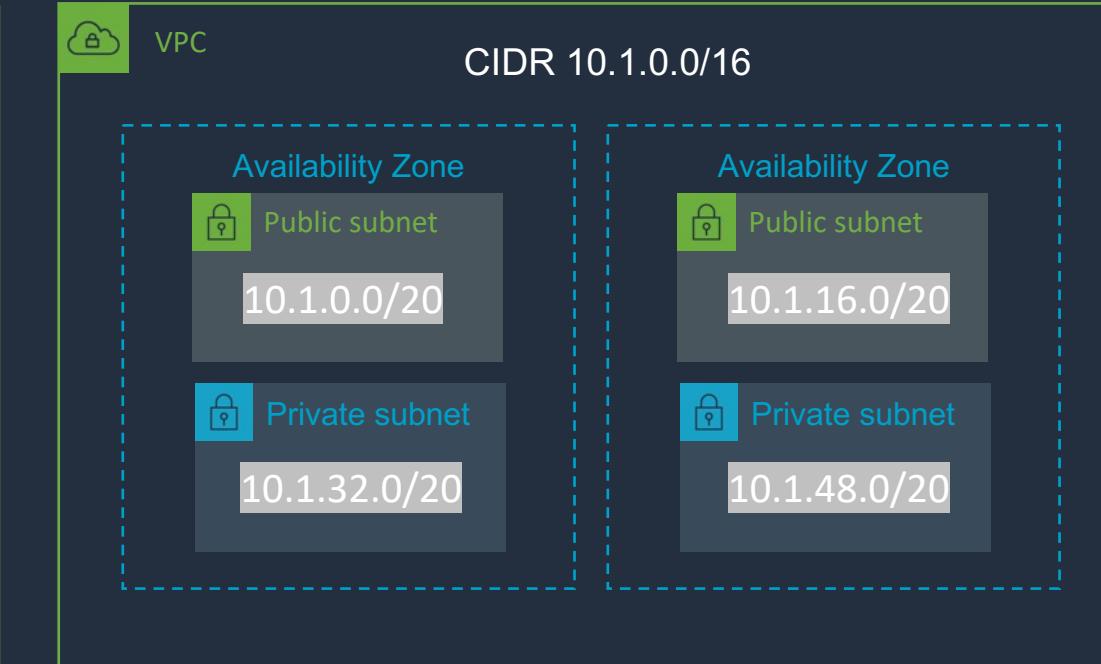
Region

Each **VPC** has a different block of IP addresses

**CIDR** stands for Classless Interdomain Routing



Each subnet has a block of **IP addresses** from the CIDR block



You can create **multiple VPCs** within each region



# Amazon VPC

VPC Component	What it is
<b>Virtual Private Cloud (VPC)</b>	A logically isolated virtual network in the AWS cloud
<b>Subnet</b>	A segment of a VPC's IP address range where you can place groups of isolated resources
<b>Internet Gateway/Egress-only Internet Gateway</b>	The Amazon VPC side of a connection to the public Internet for IPv4/IPv6
<b>Router</b>	Routers interconnect subnets and direct traffic between Internet gateways, virtual private gateways, NAT gateways, and subnets
<b>Peering Connection</b>	Direct connection between two VPCs
<b>VPC Endpoints</b>	Private connection to public AWS services
<b>NAT Instance</b>	Enables Internet access for EC2 instances in private subnets managed by you
<b>NAT Gateway</b>	Enables Internet access for EC2 instances in private subnets (managed by AWS)
<b>Virtual Private Gateway</b>	The Amazon VPC side of a Virtual Private Network (VPN) connection
<b>Customer Gateway</b>	Customer side of a VPN connection
<b>AWS Direct Connect</b>	High speed, high bandwidth, private network connection from customer to aws
<b>Security Group</b>	Instance-level firewall
<b>Network ACL</b>	Subnet-level firewall



# Amazon VPC

---

---

- A virtual private cloud (VPC) is a virtual network dedicated to your AWS account
- Analogous to having your own DC inside AWS
- It is logically isolated from other virtual networks in the AWS Cloud
- Provides complete control over the virtual networking environment including selection of IP ranges, creation of subnets, and configuration of route tables and gateways
- You can launch your AWS resources, such as Amazon EC2 instances, into your VPC



# Amazon VPC

---

- When you create a VPC, you must specify a range of IPv4 addresses for the VPC in the form of a Classless Inter-Domain Routing (CIDR) block; for example, 10.0.0.0/16
- A VPC spans all the Availability Zones in the region
- You have full control over who has access to the AWS resources inside your VPC
- By default you can create up to 5 VPCs per region
- A default VPC is created in each region with a subnet in each AZ

# Create a Custom VPC

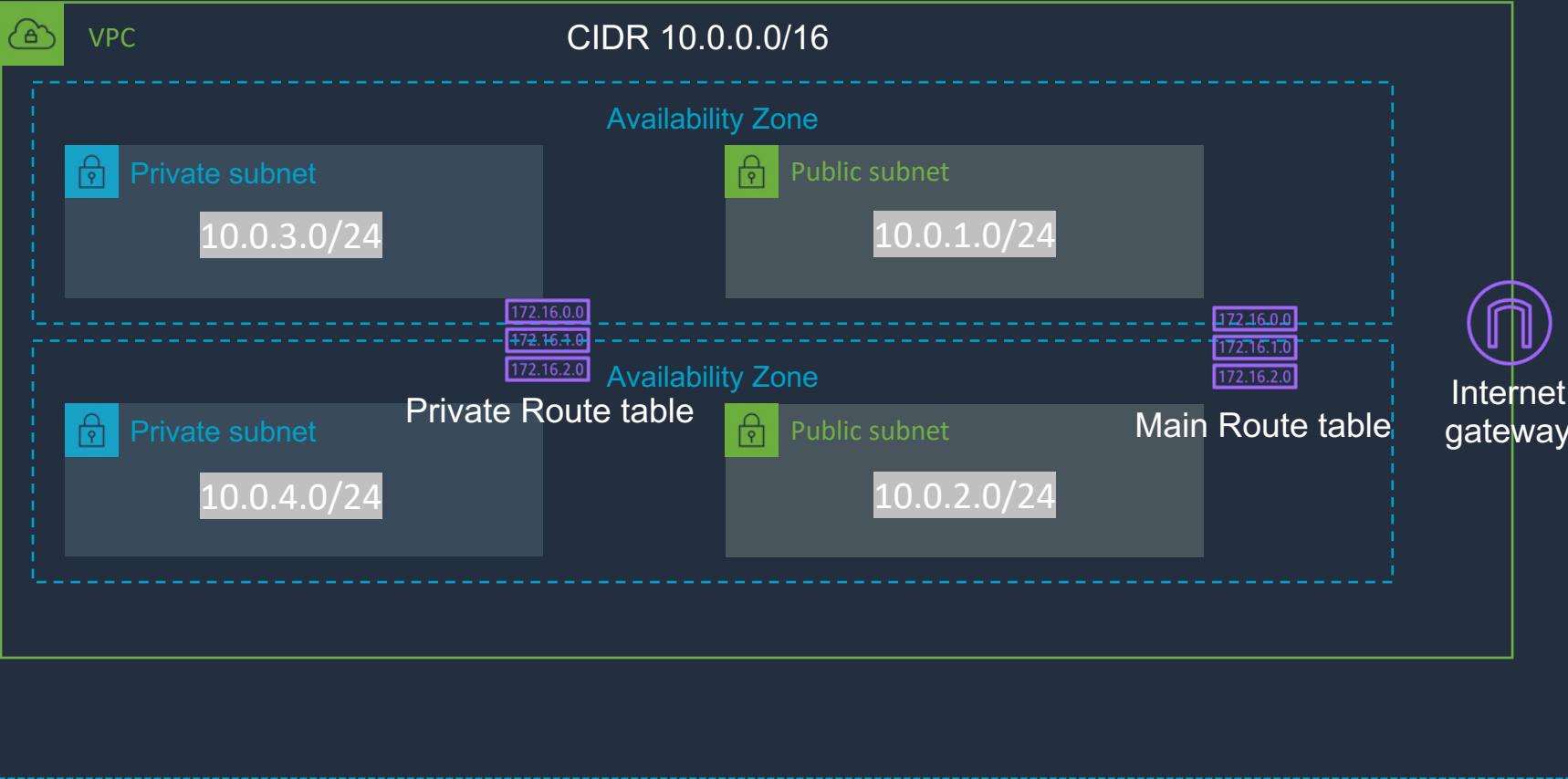




# Create a Custom VPC



Region



Main Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

Private Route Table

Destination	Target
10.0.0.0/16	Local

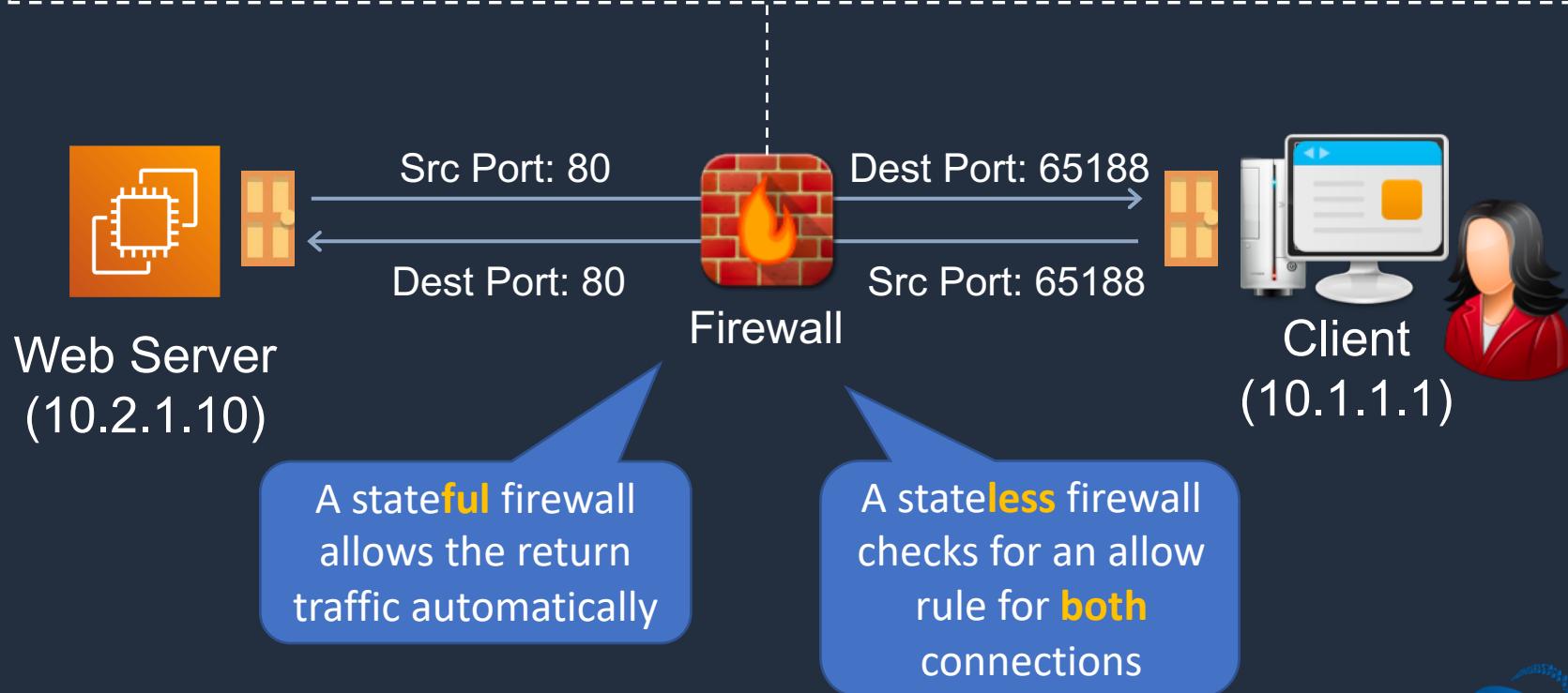
# Security Groups and Network ACLs





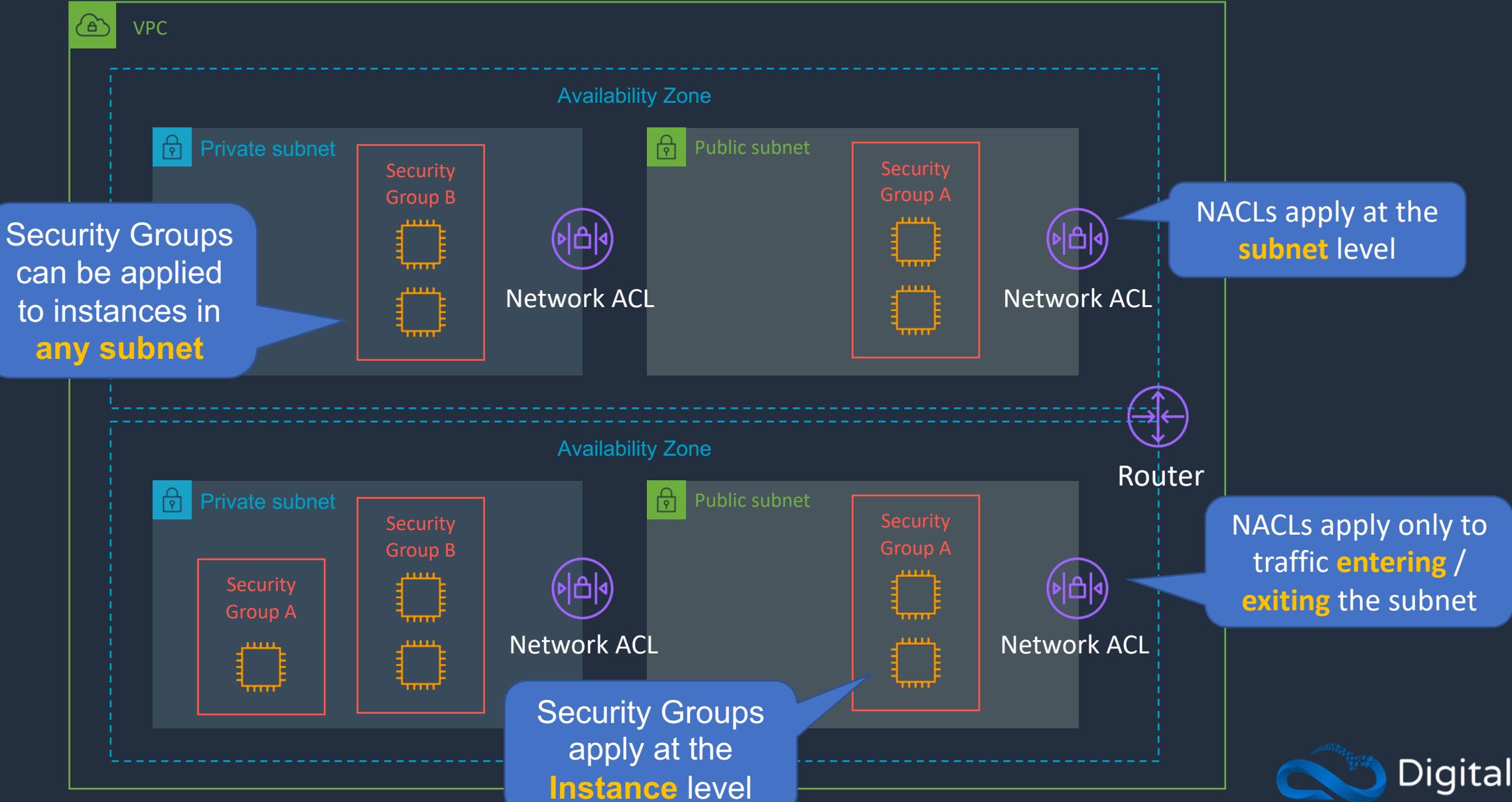
# Stateful vs Stateless Firewalls

PROTOCOL	SOURCE IP	DESTINATION IP	SOURCE PORT	DESTINATION PORT
HTTP	10.1.1.1	10.2.1.10	65188	80
HTTP	10.2.1.10	10.1.1.1	80	65188





# Security Groups and Network ACLs





# Security Group Rules

Security groups support  
**allow** rules only

## Inbound rules

Type	Protocol	Port range	Source
SSH	TCP	22	0.0.0.0/0
RDP	TCP	3389	0.0.0.0/0
RDP	TCP	3389	::/0
HTTPS	TCP	443	0.0.0.0/0
HTTPS	TCP	443	::/0
All ICMP - IPv4	ICMP	All	0.0.0.0/0

Separate rules  
are defined for  
outbound traffic

A source can be an **IP  
address or security  
group ID**



# Network ACLs

## Inbound Rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL Traffic	ALL	ALL	::/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

## Outbound Rules

Rule #	Type	Protocol	Port Range	Destination	
100	ALL Traffic	ALL	ALL	0.0.0.0/0	ALLOW
101	ALL Traffic	ALL	ALL	::/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY
*	ALL Traffic	ALL	ALL	::/0	DENY

NACLs have an explicit deny

Rules are processed in order

# Configure Security Groups and NACLs



# Public, Private and Elastic IP Addresses



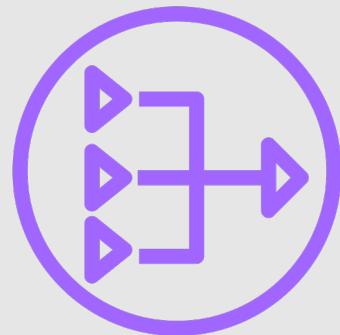
# → Public, Private and Elastic IP addresses

Name	Description
Public IP address	<p>Lost when the instance is stopped</p> <p>Used in Public Subnets</p> <p>No charge</p> <p>Associated with a private IP address on the instance</p> <p>Cannot be moved between instances</p>
Private IP address	<p>Retained when the instance is stopped</p> <p>Used in Public and Private Subnets</p>
Elastic IP address	<p>Static Public IP address</p> <p>You are charged if not used</p> <p>Associated with a private IP address on the instance</p> <p>Can be moved between instances and Elastic Network Adapters</p>

# Working with IP Addresses

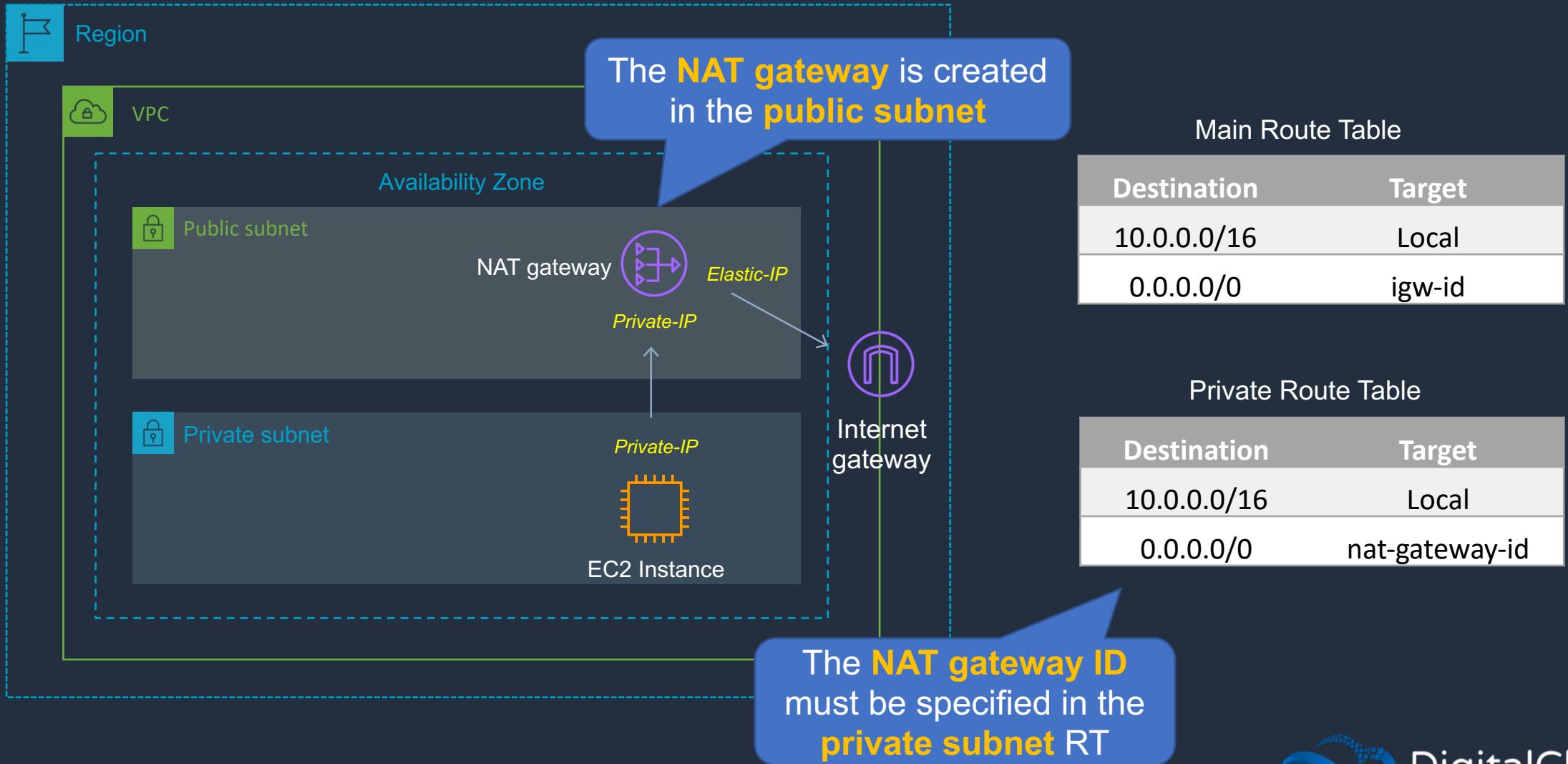


# NAT Gateways and NAT Instances



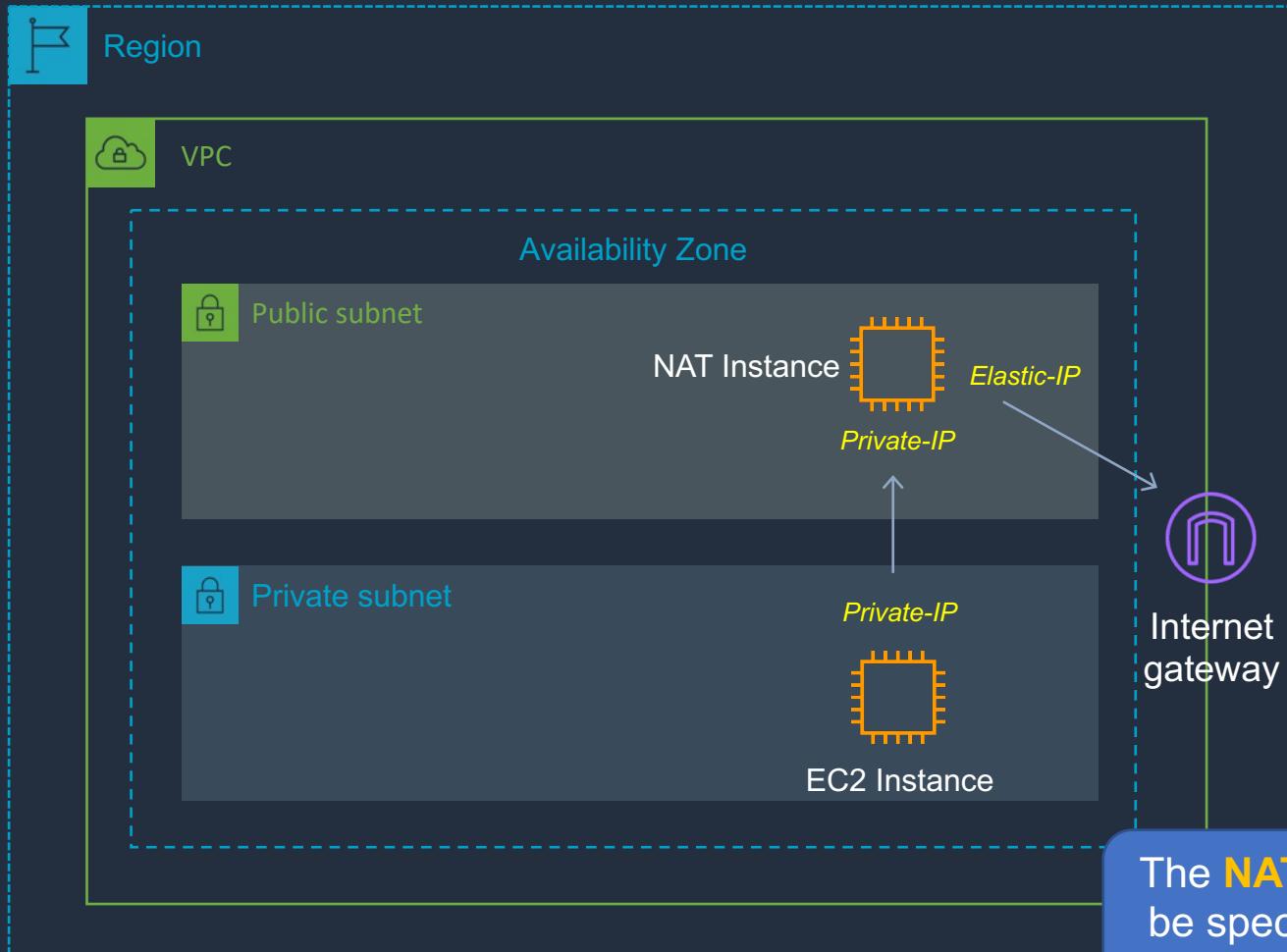


# NAT Gateways





# NAT Instances



Main Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	igw-id

Private Route Table

Destination	Target
10.0.0.0/16	Local
0.0.0.0/0	nat-instance-id

The **NAT instance ID** must  
be specified in the **private  
subnet RT**



# NAT Instance vs NAT Gateway

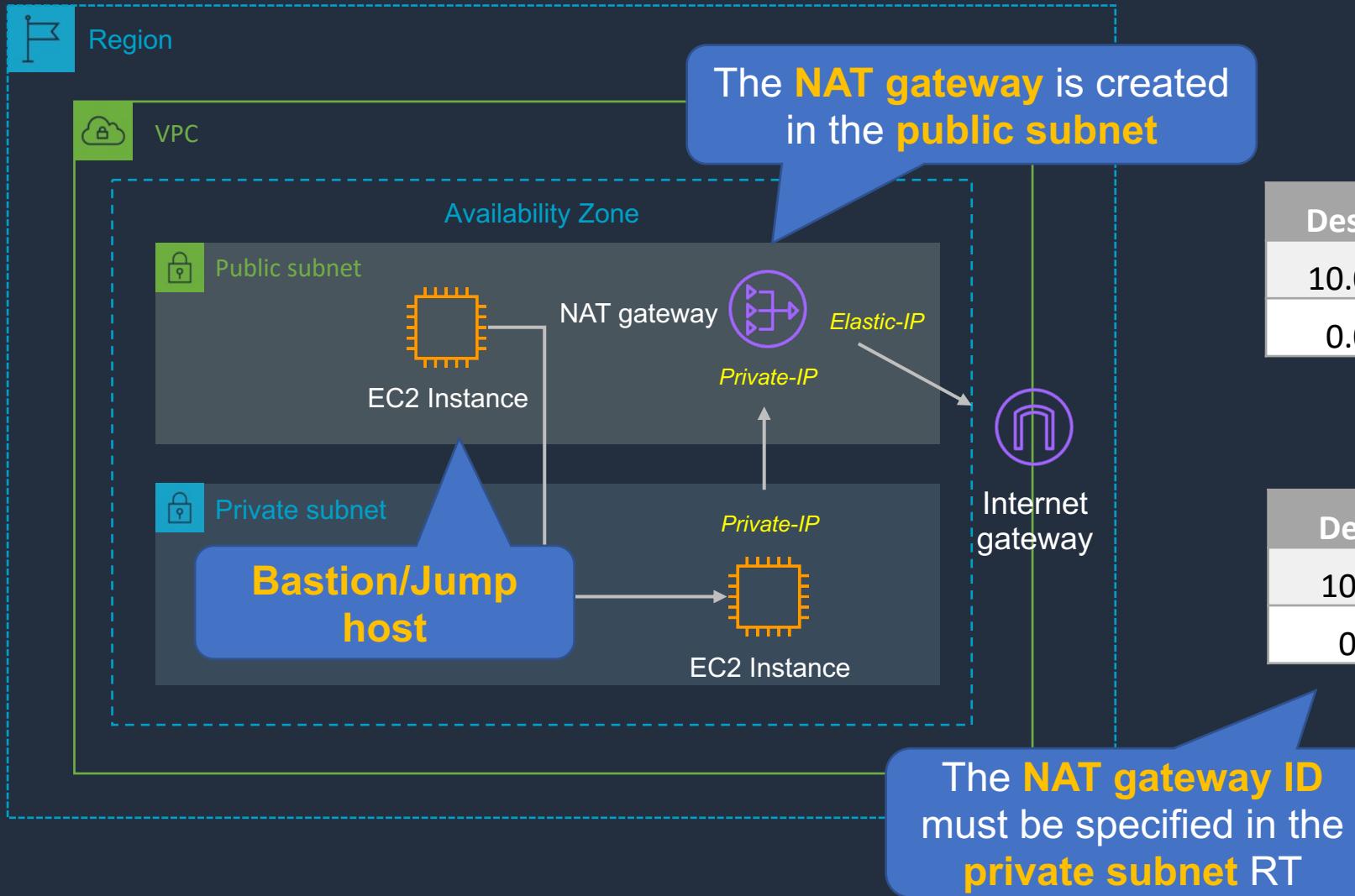
NAT Instance	NAT Gateway
Managed by you (e.g. software updates)	Managed by AWS
Scale up (instance type) manually and use enhanced networking	Elastic scalability up to 45 Gbps
No high availability – scripted/auto-scaled HA possible using multiple NATs in multiple subnets	Provides automatic high availability within an AZ and can be placed in multiple AZs

# Deploy a NAT Gateway





# NAT Gateways

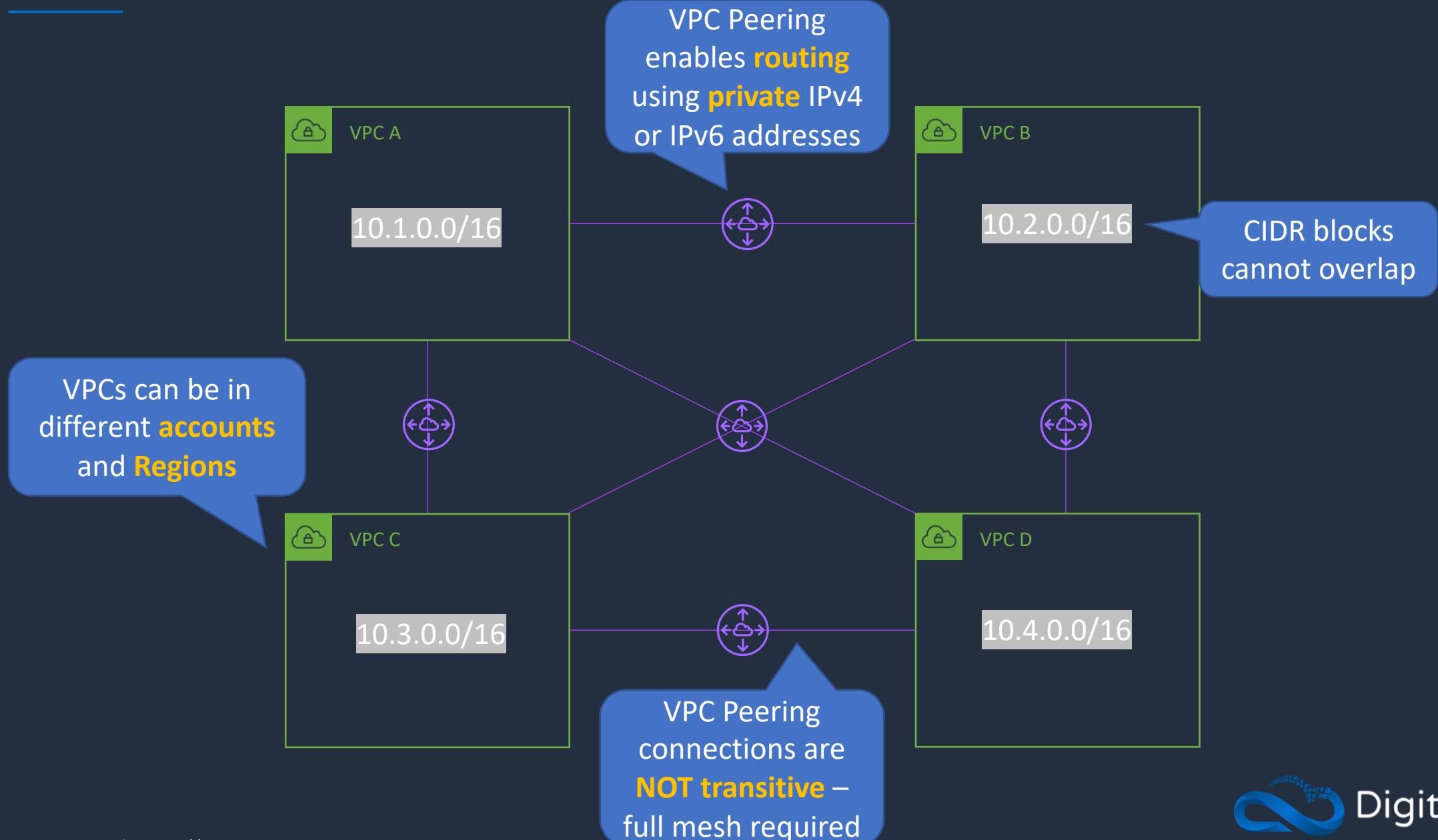


# Amazon VPC Peering





# VPC Peering

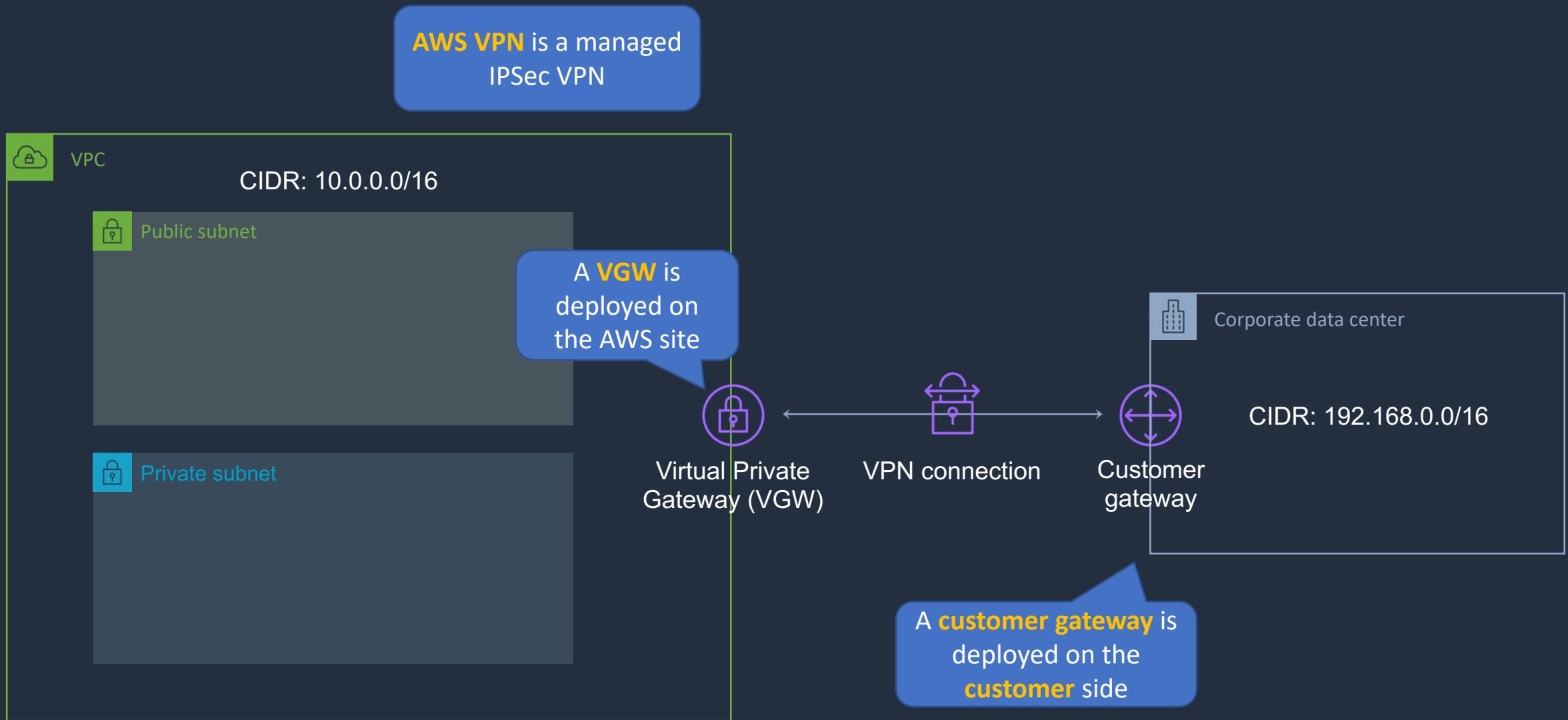


# Amazon VPN and AWS Direct Connect



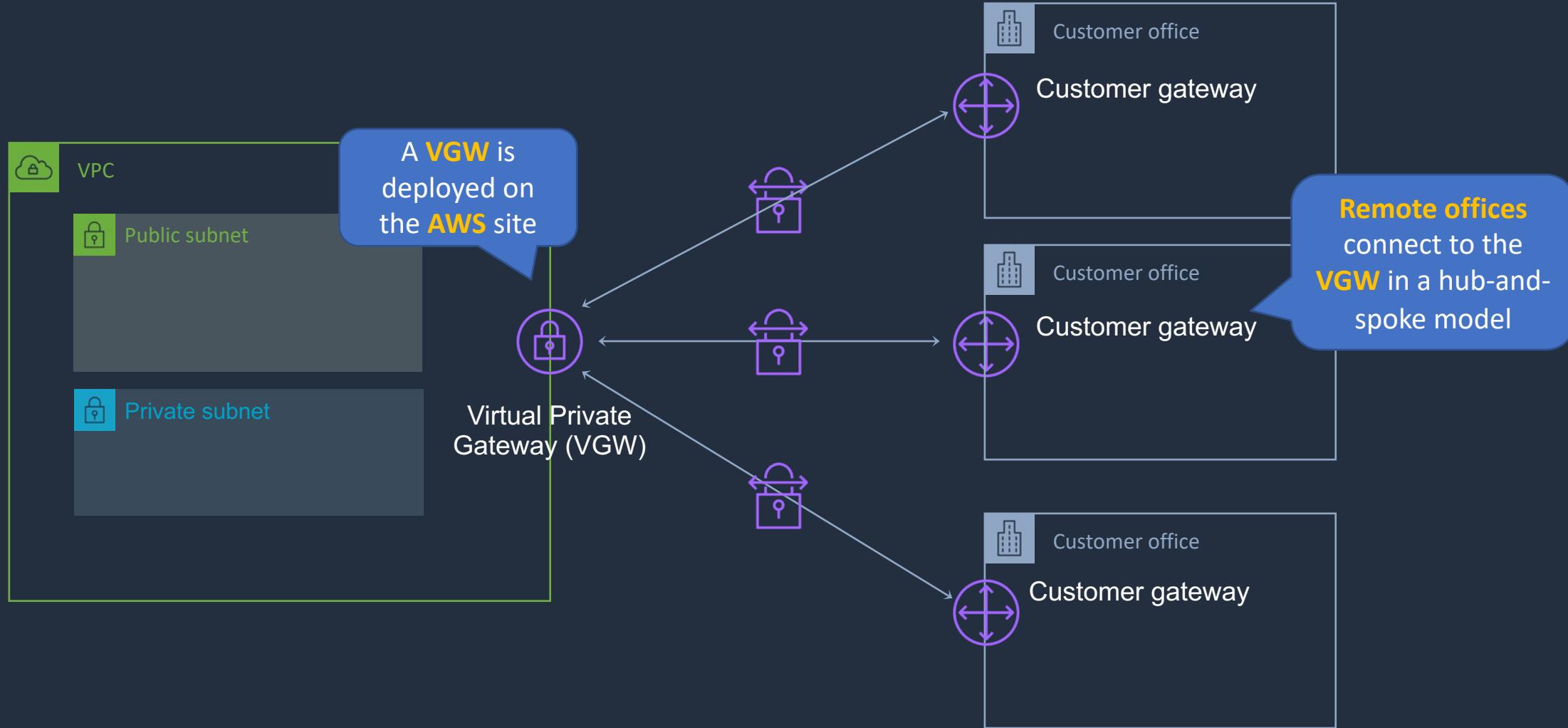


# AWS Site-to-Site VPN





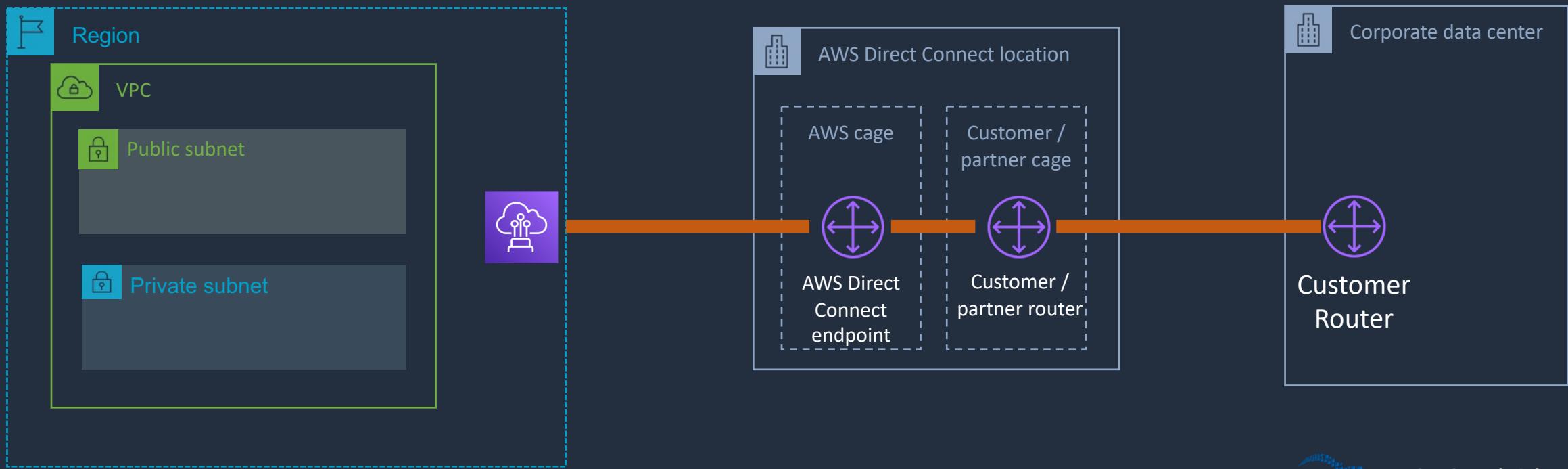
# AWS VPN CloudHub



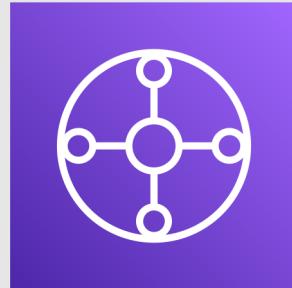


# AWS Direct Connect

- **Private** connectivity between AWS and your data center / office
- Consistent network experience – increased **speed/latency** & **bandwidth/throughput**
- Lower costs for organizations that transfer **large** volumes of data



# AWS Transit Gateway

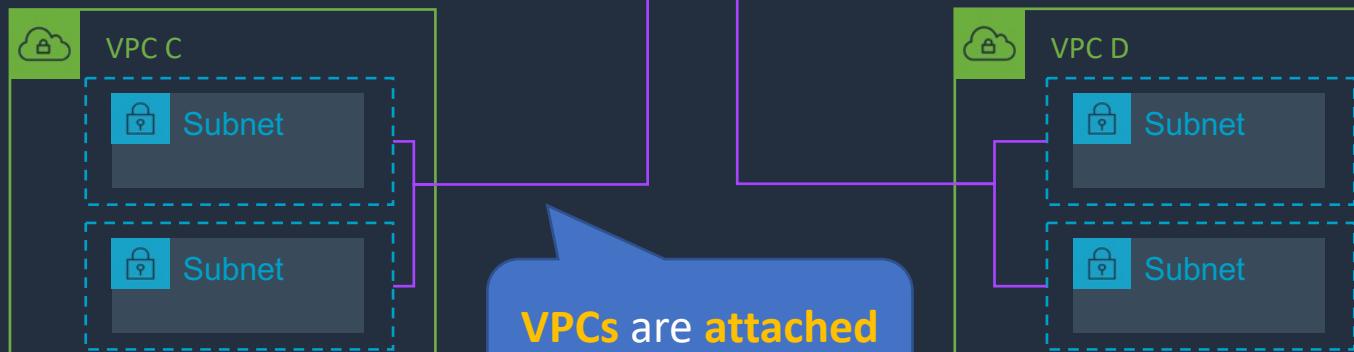




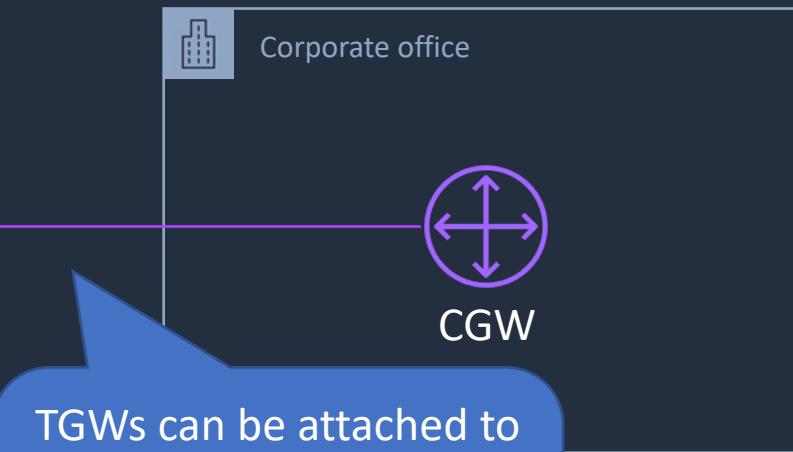
# AWS Transit Gateway



**Transit Gateway** is a network transit hub that interconnects **VPCs** and **on-premises** networks

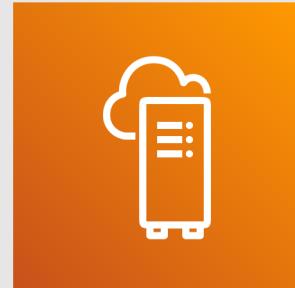


**VPCs are attached** to Transit Gateway



TGWs can be attached to **VPNs, Direct Connect Gateways, 3<sup>rd</sup> party appliances and TGWs** in other Regions/accounts

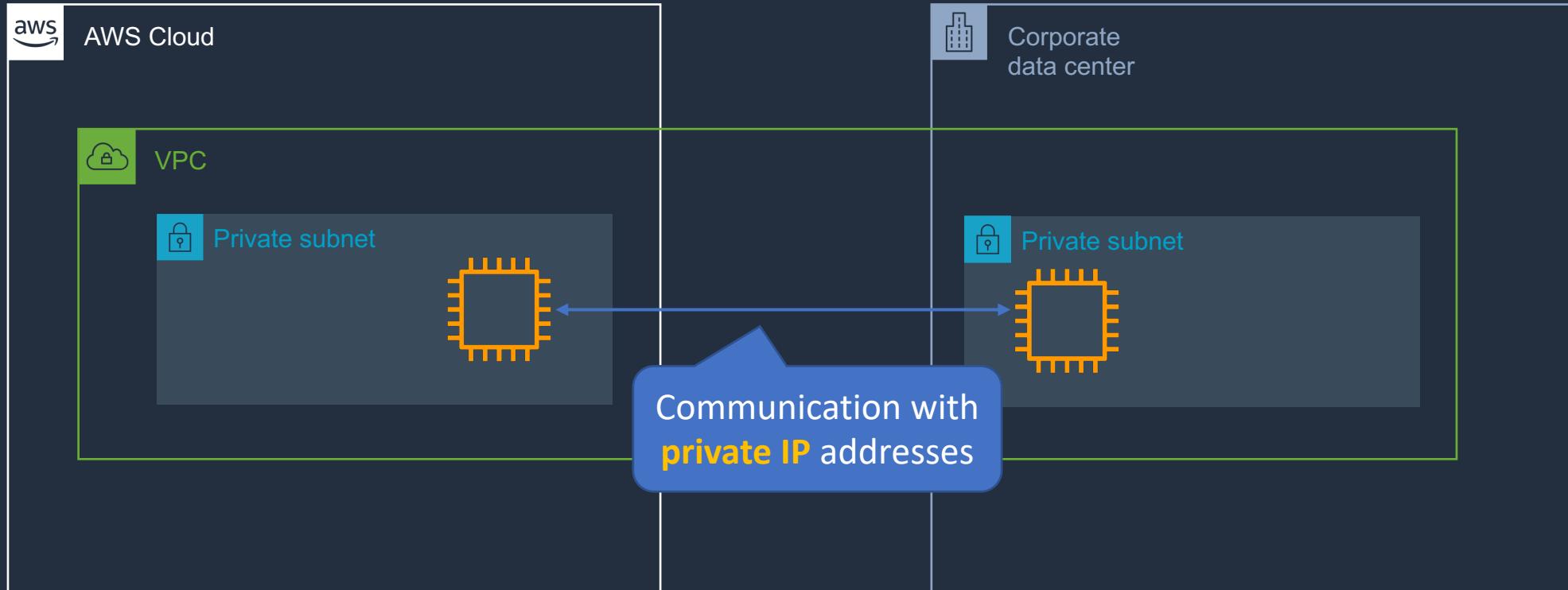
# AWS Outposts





# AWS Outposts

---





# AWS Outposts

Services you can run on AWS Outposts include:

- Amazon EC2
- Amazon EBS
- Amazon S3
- Amazon VPC
- Amazon ECS/EKS
- Amazon RDS
- Amazon EMR

# SECTION 10

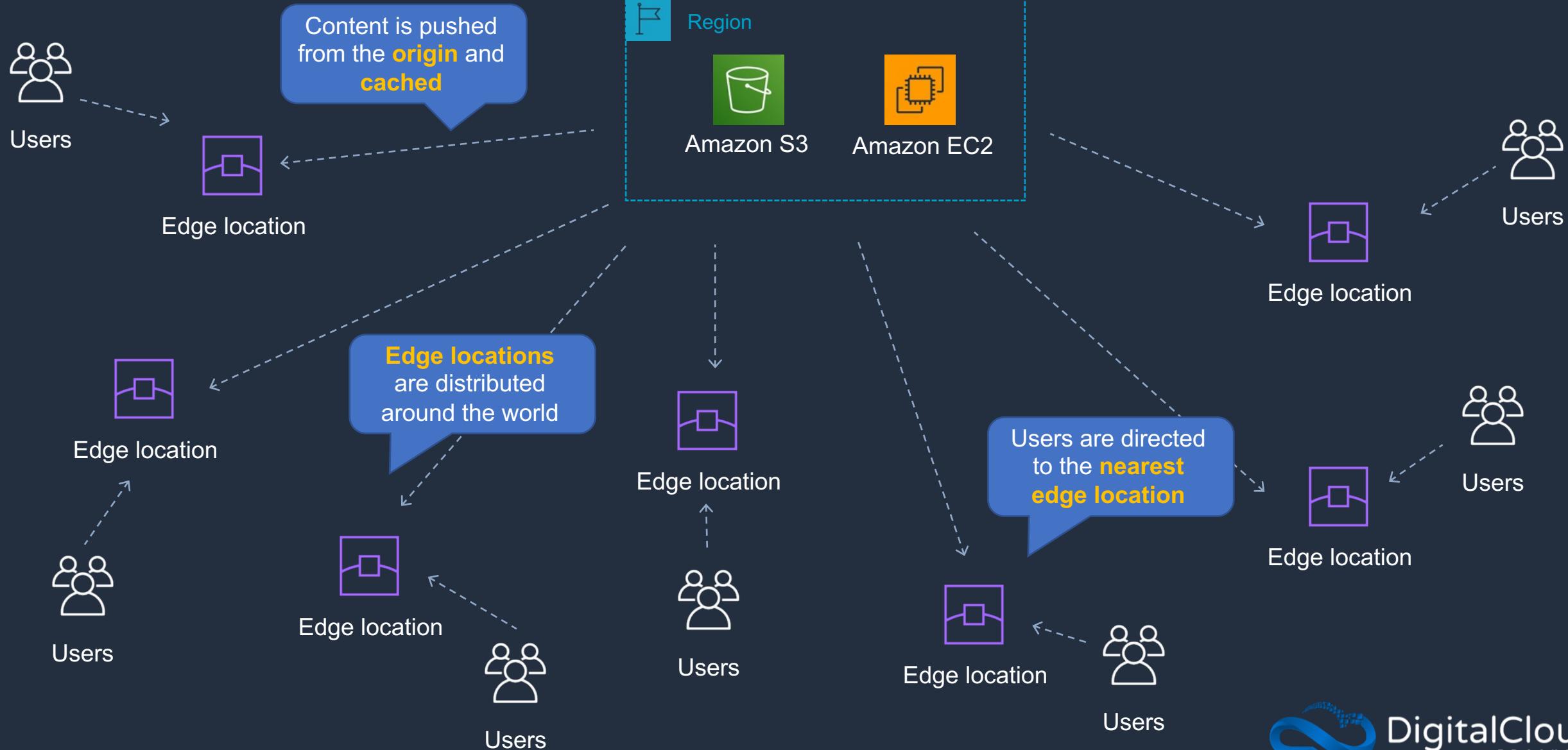
## Deployment and Automation

# Caching and Amazon CloudFront





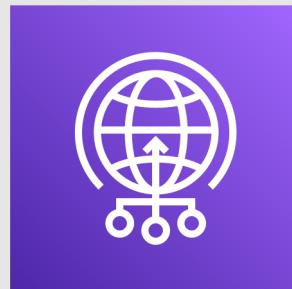
# Amazon CloudFront



# S3 Static Website with CloudFront



# AWS Global Accelerator



# AWS Global Accelerator



Users in US

Connect via Edge Location

Requests are routed to the **optimal endpoint**



us-east-1



Edge location

User traffic ingresses using the closest **Edge Location**



Global Accelerator  
Addresses:  
51.45.2.12  
53.58.31.89

Static **anycast** IP addresses

AWS Global Network

Traffic traverses the **AWS global network**



ap-southeast-2

Users are **redirected** to another **endpoint**

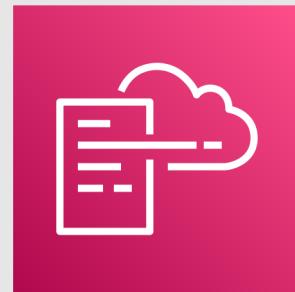


# AWS Global Accelerator vs CloudFront

---

- Both use the AWS global network and edge locations
- CloudFront improves performance for cacheable content and dynamic content
- GA improves performance for a wide range of applications over TCP and UDP
- GA proxies connections to applications in one or more AWS Regions
- GA provides failover between AWS Regions

# AWS CloudFormation





# AWS CloudFormation

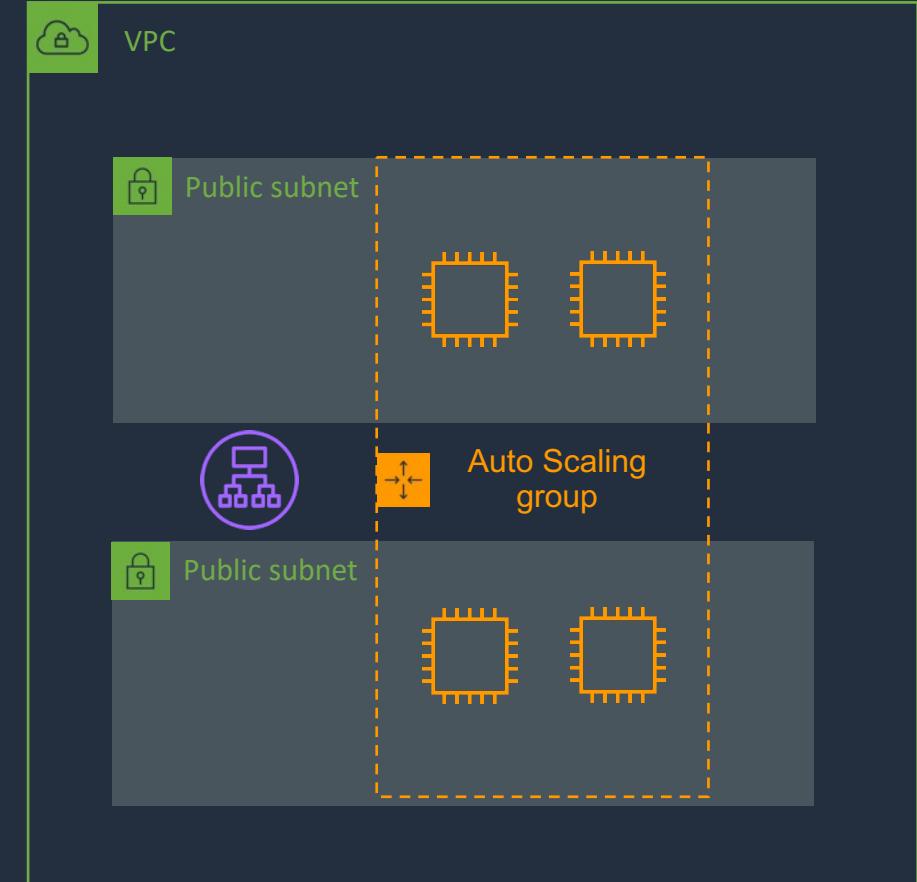
Infrastructure patterns are defined in a **template** file using **code**



CloudFormation **builds** your infrastructure according to the **template**

AWS CloudFormation

```
1 "AWSTemplateFormatVersion": "2010-09-09",
2
3 "Description": "AWS CloudFormation Sample Template WordPress_Multi_AZ: WordPress is web
4
5 "Parameters": {
6   "VpcId": {
7     "Type": "AWS::EC2::VPC::Id",
8     "Description": "VpcId of your existing Virtual Private Cloud (VPC)",
9     "ConstraintDescription": "must be the VPC Id of an existing Virtual Private Cloud."
10 },
11
12 "Subnets": {
13   "Type": "List<AWS::EC2::Subnet::Id>",
14   "Description": "The list of SubnetIds in your Virtual Private Cloud (VPC)",
15   "ConstraintDescription": "must be a list of at least two existing subnets associated
16 },
```





# AWS CloudFormation

---

---

- Infrastructure is provisioned consistently, with fewer mistakes (human error)
- Less time and effort than configuring resources manually
- Free to use (you're only charged for the resources provisioned)
- A template is a YAML or JSON template used to describe the end-state of the infrastructure you are either provisioning or changing
- CloudFormation creates a Stack based on the template
- Can easily rollback and delete the entire stack as well

# Create CloudFormation Stack



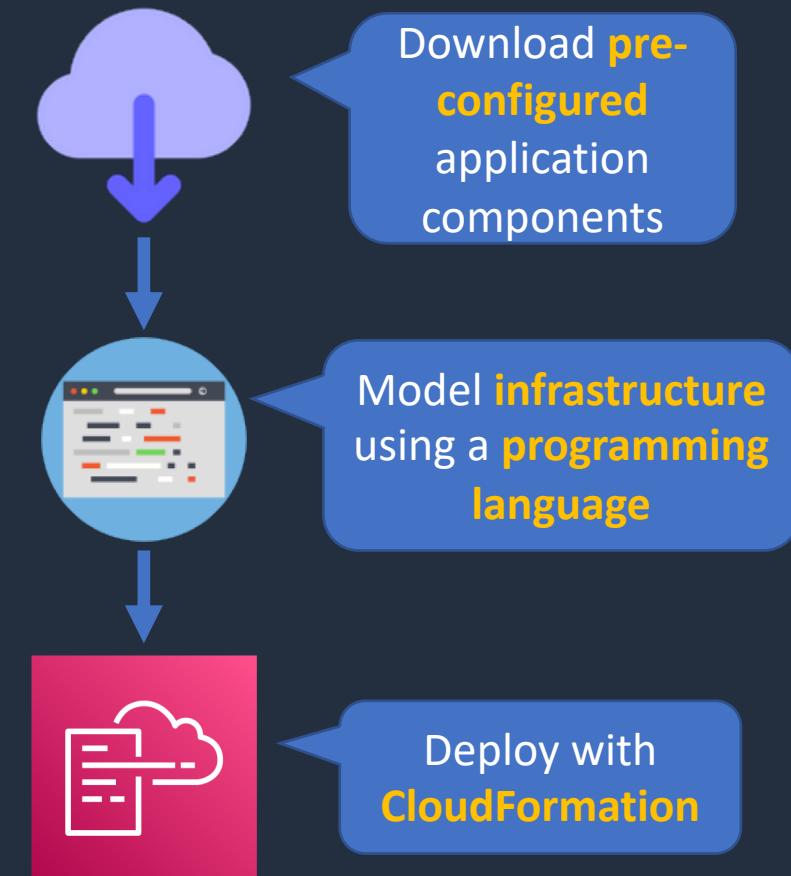
# AWS Cloud Development Kit (CDK)





# AWS Cloud Development Kit

- Open-source software development framework to define your cloud application resources using **familiar programming languages**
- Preconfigures cloud resources with proven defaults using **constructs**
- Provisions your resources using **AWS CloudFormation**
- Enables you to model application infrastructure using TypeScript, Python, Java, and .NET
- Use existing IDE, testing tools, and workflow patterns

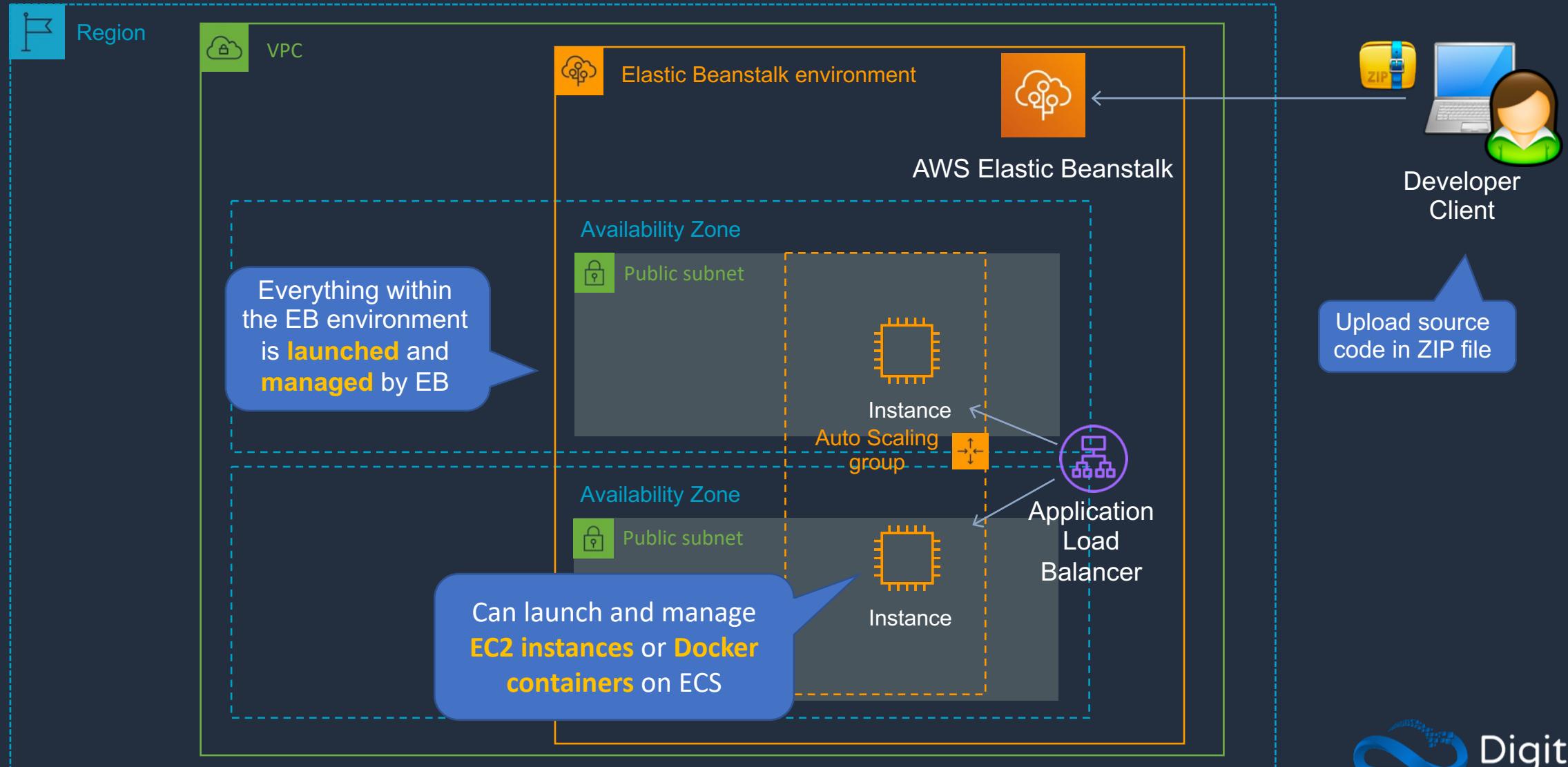


# AWS Elastic Beanstalk





# AWS Elastic Beanstalk





# AWS Elastic Beanstalk

---

---

- Supports Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker web applications
- Integrates with VPC
- Integrates with IAM
- Can provision most database instances
- Allows full control of the underlying resources
- Code is deployed using a WAR file or Git repository



# AWS Elastic Beanstalk

CloudFormation	Elastic Beanstalk
“Template-driven provisioning”	“Web apps made easy”
Deploys infrastructure using code	Deploys applications on EC2 (PaaS)
Can be used to deploy almost any AWS service	Deploys web applications based on Java, .NET, PHP, Node.js, Python, Ruby, Go, and Docker
Uses JSON or YAML template files	Uses ZIP or WAR files (or Git)
Similar to Terraform	Similar to Google App Engine

# Deploy a Web Application



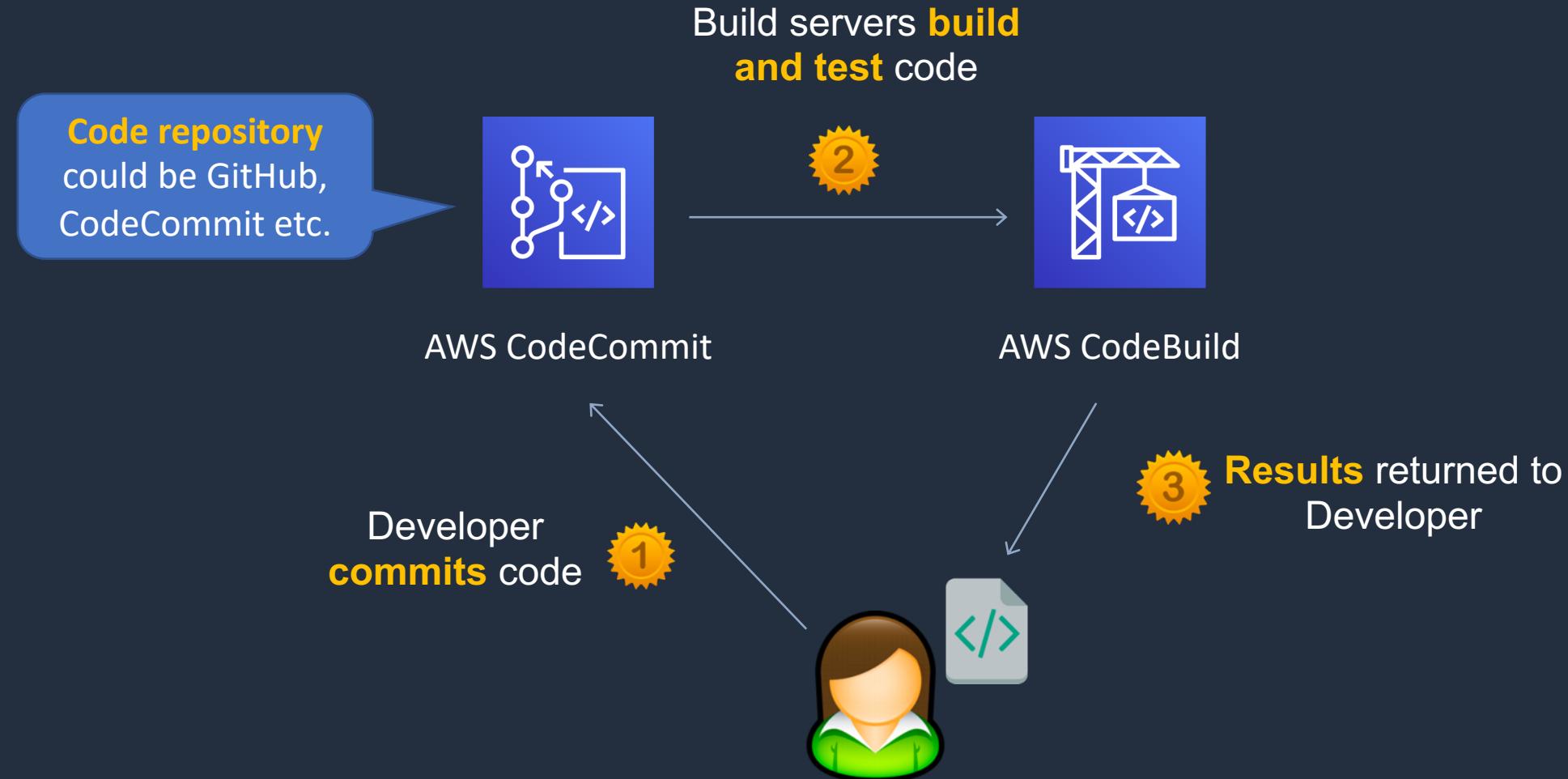
# AWS Developer Tools (Code\*)





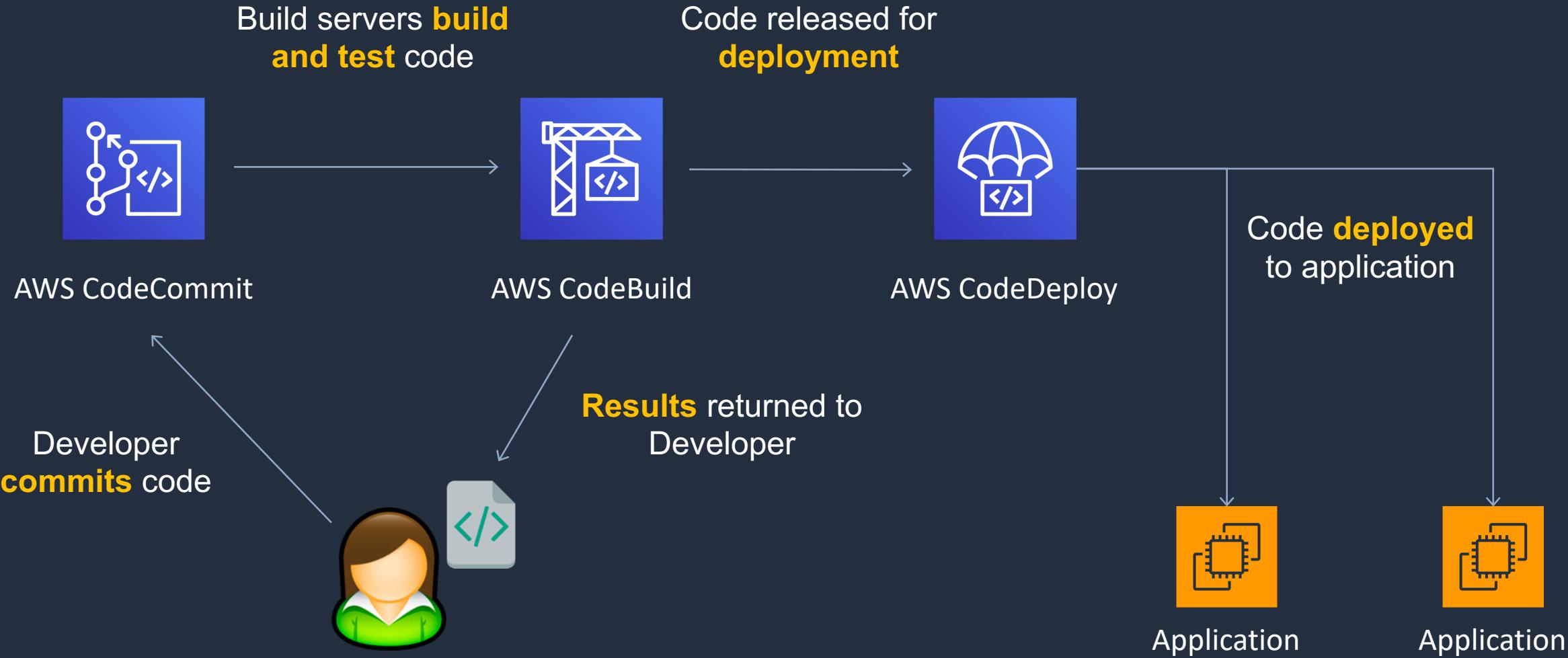
# Continuous Integration

---





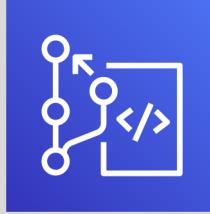
# Continuous Integration and Continuous Delivery





# Continuous Integration and Continuous Delivery

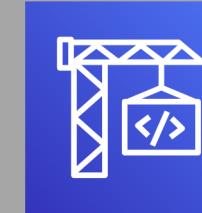
CODE



AWS CodeCommit

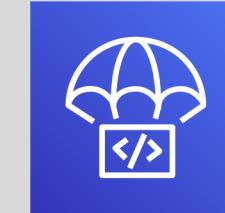
BUILD & TEST

AWS CodePipeline



AWS CodeBuild

DEPLOY

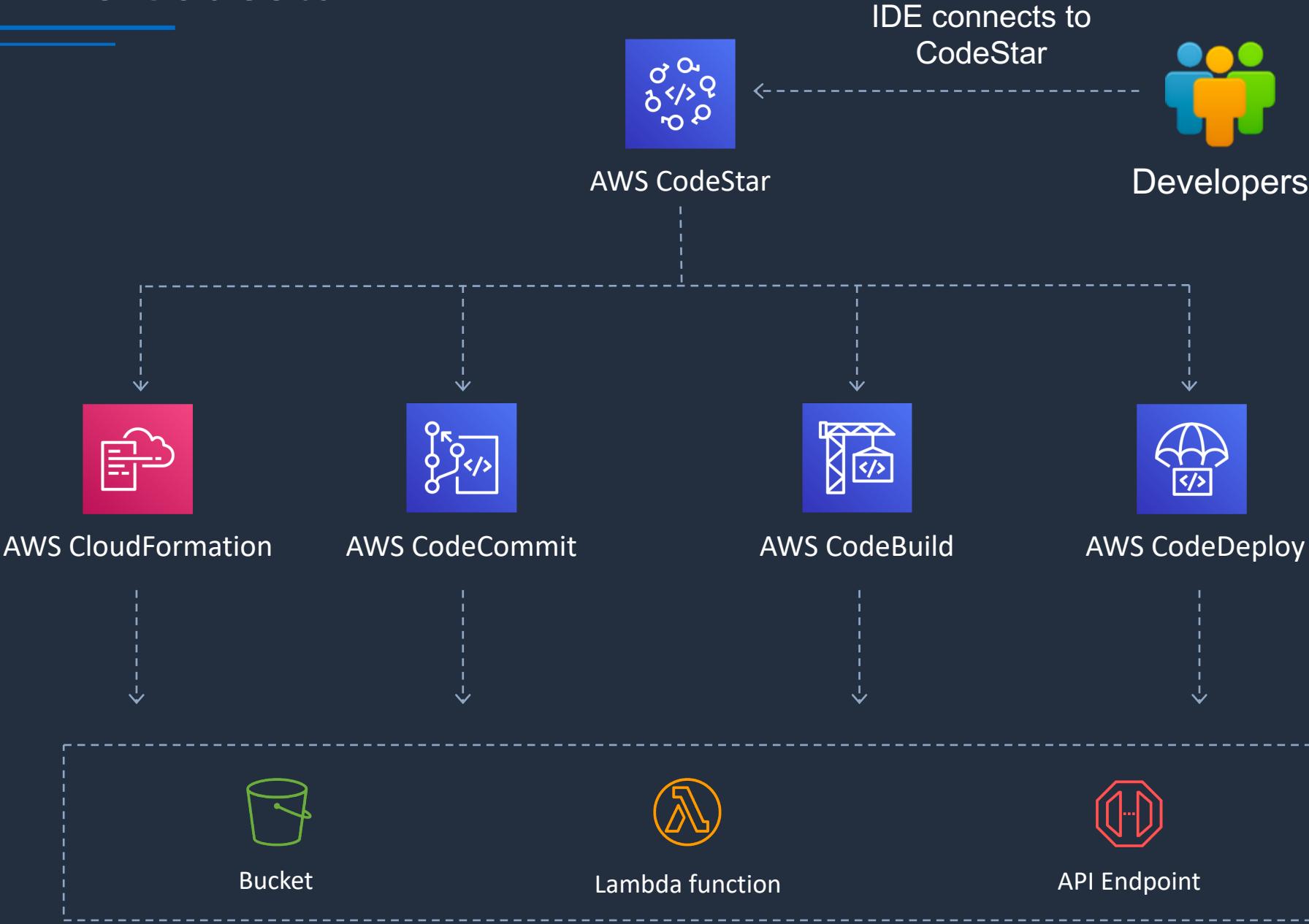


AWS CodeDeploy





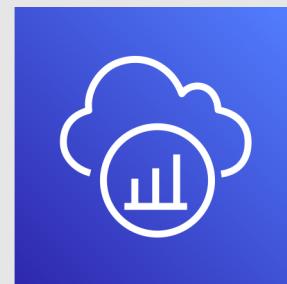
# AWS CodeStar



# AWS CodeStar



# AWS X-Ray





# AWS X-Ray

Records **latency** writing to a DynamoDB table



Client



Records **latency** from client to application



# AWS X-Ray

---

---

- AWS X-Ray helps developers analyze and debug production, distributed applications, such as those built using a microservices architecture
- AWS X-Ray supports applications running on:
  - Amazon EC2
  - Amazon ECS
  - AWS Lambda
  - AWS Elastic Beanstalk
- Need to integrate the X-Ray SDK with your application and install the X-Ray agent

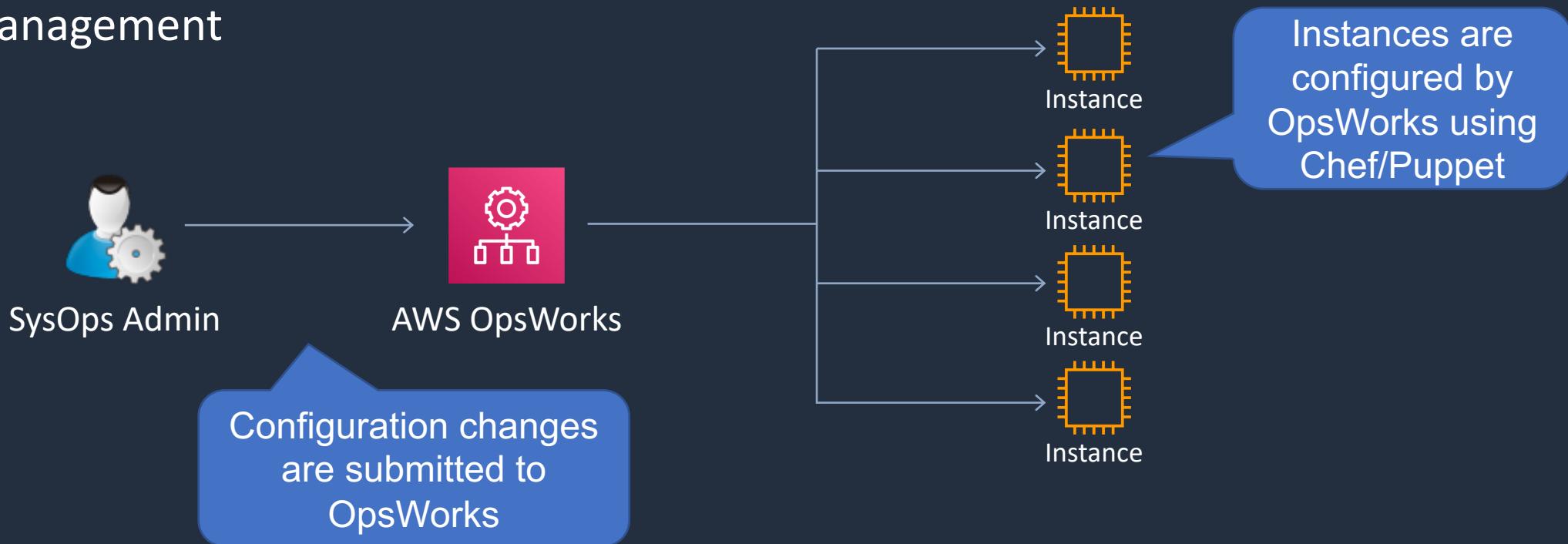
# AWS OpsWorks





# AWS OpsWorks

- AWS OpsWorks is a configuration management service that provides managed instances of **Chef** and **Puppet**
- Updates include patching, updating, backup, configuration and compliance management



# SECTION 11

## Databases and Analytics

# Types of Database





# Relational vs Non-Relational

---

---

Key differences are how data are **managed** and how data are **stored**

Relational	Non-Relational
Organized by tables, rows and columns	Varied data storage models
Rigid schema (SQL)	Flexible schema (NoSQL) – data stored in key-value pairs, columns, documents or graphs
Rules enforced within database	Rules can be defined in application code (outside database)
Typically scaled vertically	Scales horizontally
Supports complex queries and joins	Unstructured, simple language that supports any kind of schema
Amazon RDS, Oracle, MySQL, IBM DB2, PostgreSQL	Amazon DynamoDB, MongoDB, Redis, Neo4j



# Relational Database

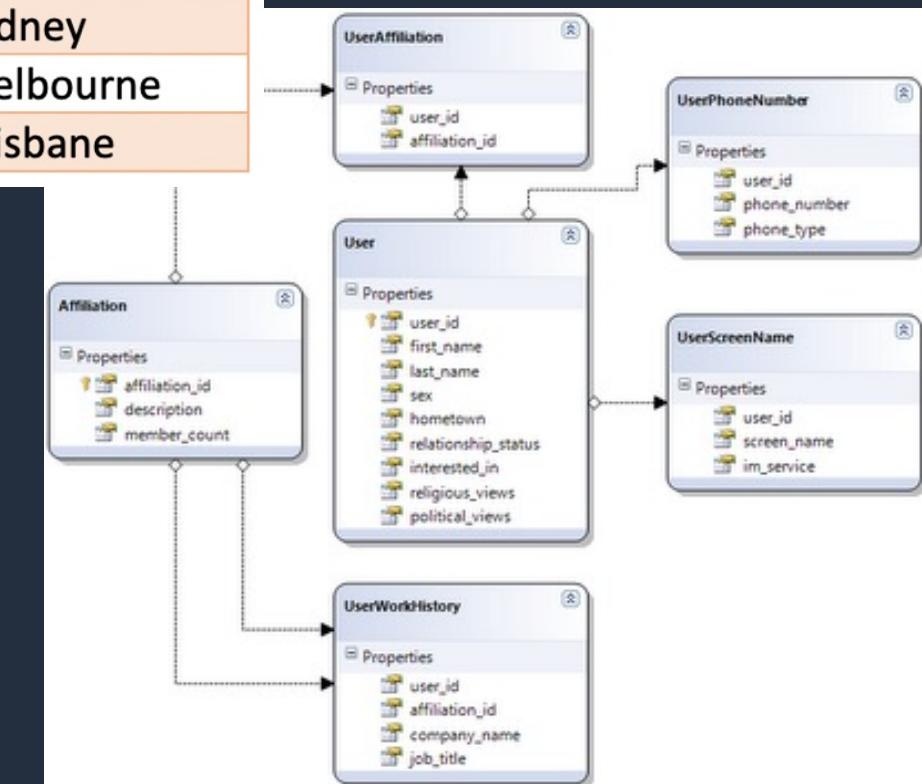
EmployeeID	FirstName	LastName	JobRole	Location
00001	Paul	Peterson	Senior Developer	Sydney
00002	Kaleigh	Annette	Assistant Manager	Brisbane
00003	Carl	Wood	Sales Support	Sydney
00004	Vinni	Jones	Customer Services	Melbourne
00005	Stefanie	Howard	IT Architect	Brisbane

Relational Database

Structured Query Language (SQL) query:

```
SELECT FirstName  
FROM employees  
WHERE Location = Sydney
```

Relational Database – Multiple Tables



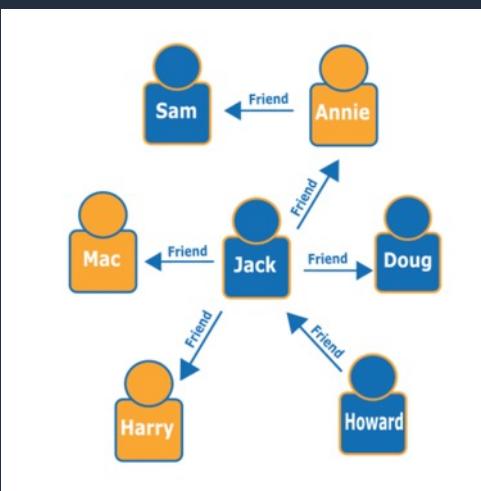


# Types of Non-Relational DB (NoSQL)

Key-value – e.g. Amazon DynamoDB



Graph – e.g. Amazon Neptune



Document – e.g. MongoDB

```
JSON
1 [           ...
2 {           ...
3   "year" : 2013,
4   "title" : "Turn It Down, Or Else!",
5   "info" : {
6     "directors" : [ "Alice Smith", "Bob Jones" ],
7     "release_date" : "2013-01-18T00:00:00Z",
8     "rating" : 6.2,
9     "genres" : [ "Comedy", "Drama" ],
10    "image_url" : "http://ia.media-imdb.com/images/N/09ERWAU7FS797A37LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@._V1_SX400_.jpg",
11    "plot" : "A rock band plays their music at high volumes, annoying the neighbors.",
12    "actors" : [ "David Matthewman", "Jonathan G. Neff" ]
13  }
14 },
15 {
16   "year": 2015,
17   "title": "The Big New Movie",
18   "info": {
19     "plot": "Nothing happens at all.",
20     "rating": 0
21   }
22 }
23 ]
```



# Operational vs Analytical

---

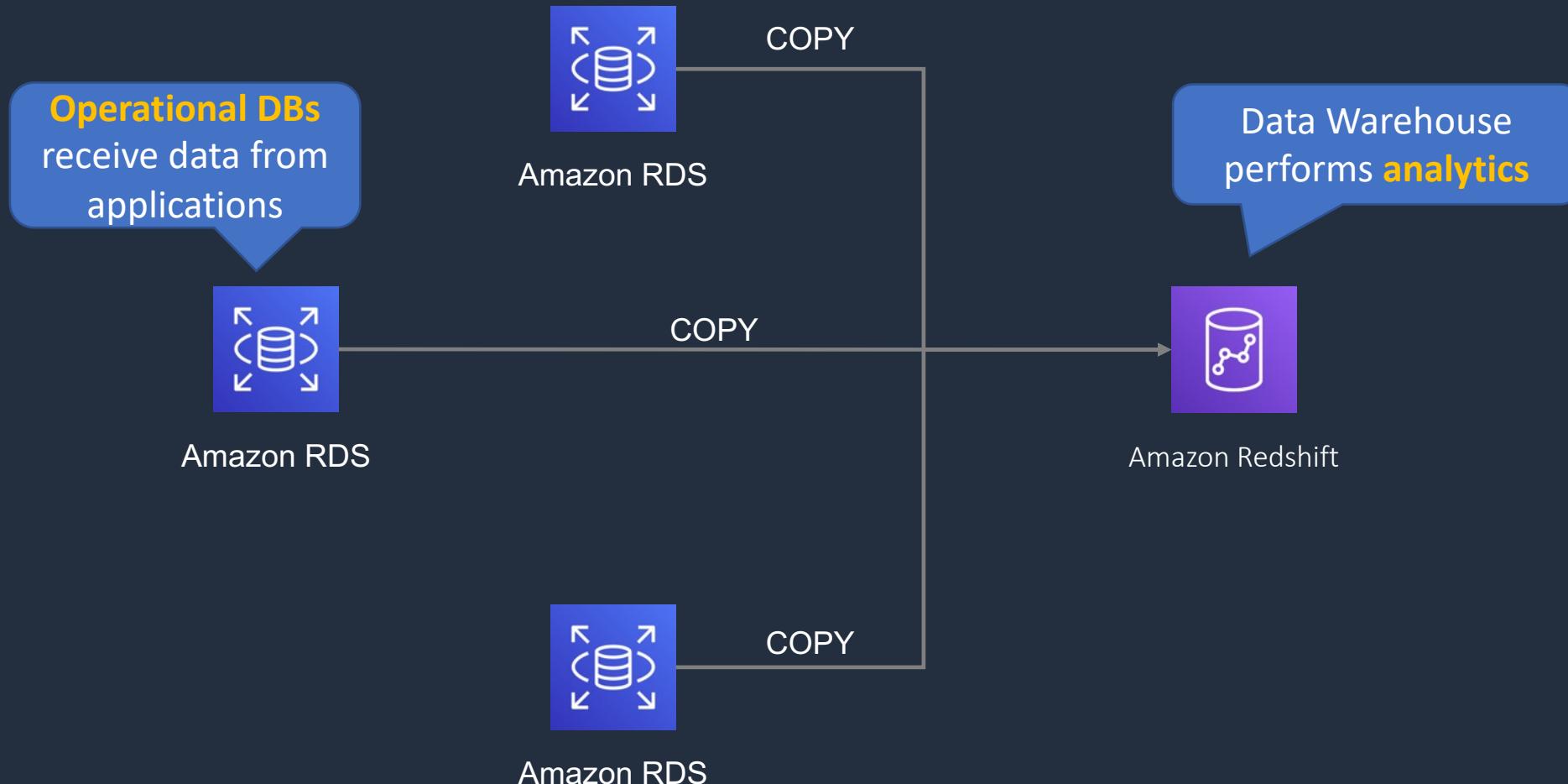
Key differences are **use cases** and how the database is **optimized**

Operational / transactional	Analytical
Online Transaction Processing (OLTP)	Online Analytics Processing (OLAP) – the source data comes from OLTP DBs
Production DBs that process transactions. E.g. adding customer records, checking stock availability (INSERT, UPDATE, DELETE)	Data warehouse. Typically, separated from the customer facing DBs. Data is extracted for decision making
Short transactions and simple queries	Long transactions and complex queries
Relational examples: Amazon RDS, Oracle, IBM DB2, MySQL	Relational examples: Amazon RedShift, Teradata, HP Vertica
Non-relational examples: MongoDB, Cassandra, Neo4j, HBase	Non-relational examples: Amazon EMR, MapReduce



# Operational vs Analytical

---





# AWS Databases

Data Store	Use Case
Database on EC2	<ul style="list-style-type: none"><li>• Need full control over instance and database</li><li>• Third-party database engine (not available in RDS)</li></ul>
Amazon RDS	<ul style="list-style-type: none"><li>• Need traditional relational database</li><li>• e.g. Oracle, PostgreSQL, Microsoft SQL, MariaDB, MySQL</li><li>• Data is well-formed and structured</li></ul>
Amazon DynamoDB	<ul style="list-style-type: none"><li>• NoSQL database</li><li>• In-memory performance</li><li>• High I/O needs</li><li>• Dynamic scaling</li></ul>
Amazon RedShift	<ul style="list-style-type: none"><li>• Data warehouse for large volumes of aggregated data</li></ul>
Amazon ElastiCache	<ul style="list-style-type: none"><li>• Fast temporary storage for small amounts of data</li><li>• In-memory database</li></ul>
Amazon EMR	<ul style="list-style-type: none"><li>• Analytics workloads using the Hadoop framework</li></ul>

# Amazon Relational Database Service (RDS)





# Amazon RDS

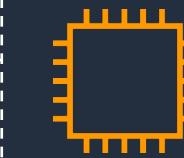
---

RDS is a **managed**, relational database



Amazon RDS

RDS runs on **EC2 instances**, so you must choose an **instance type**



EC2

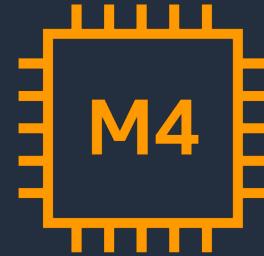
RDS supports the following database engines:

- Amazon Aurora
- MySQL
- MariaDB
- Oracle
- Microsoft SQL Server
- PostgreSQL



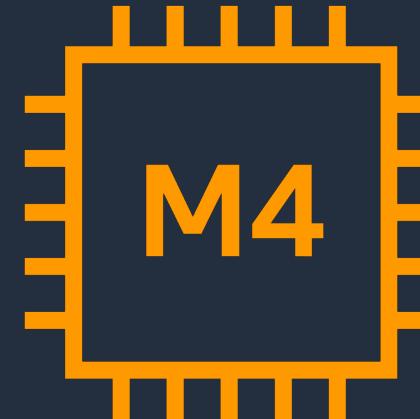
# Amazon RDS Scaling Up (vertically)

---



M4 instance

**db.m4.large** 2  
vCPUs, 8 GiB  
RAM

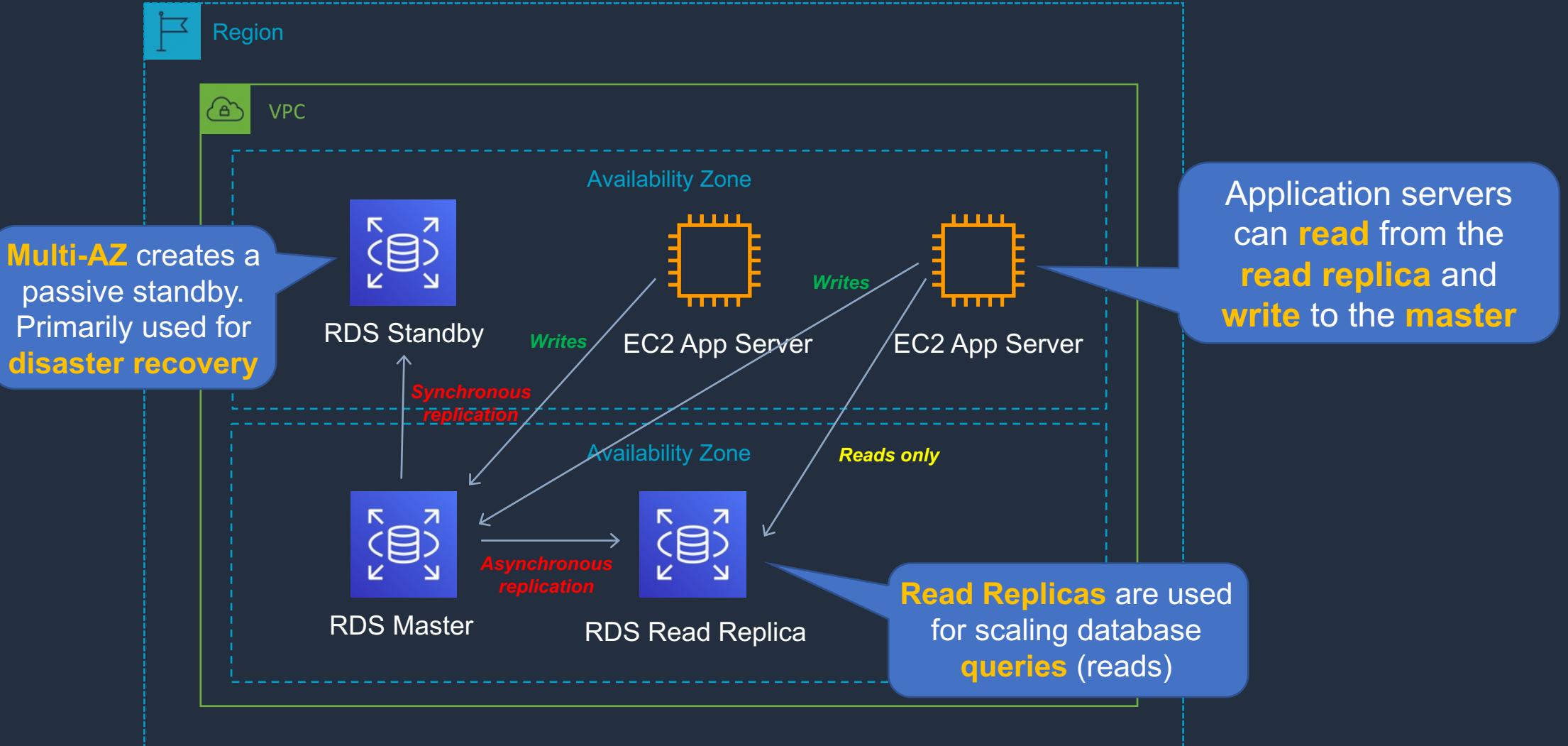


M4 Instance

**db.m4.2xlarge**  
4 vCPUs, 32  
GiB RAM



# Disaster Recovery (DR) and Scaling Out (Horizontally)





# Amazon RDS

- RDS uses EC2 instances, so you must choose an instance family/type
- Relational databases are known as Structured Query Language (SQL) databases
- RDS is an Online Transaction Processing (OLTP) type of database
- Easy to setup, highly available, fault tolerant, and scalable
- Common use cases include online stores and banking systems
- You can encrypt your Amazon RDS instances and snapshots at rest by enabling the encryption option for your Amazon RDS DB instance (during creation)
- Encryption uses AWS Key Management Service (KMS)



# Amazon RDS

---

- Amazon RDS supports the following database engines:
  - SQL Server
  - Oracle
  - MySQL Server
  - PostgreSQL
  - Aurora
  - MariaDB
- Scales up by increasing instance size (compute and storage)
- Read replicas option for read heavy workloads (scales out for reads/queries only)
- Disaster recovery with Multi-AZ option

# Create Amazon RDS Database



# Amazon Aurora





# Amazon Aurora

---

- Amazon Aurora is an AWS database offering in the RDS family
- Amazon Aurora is a MySQL and PostgreSQL-compatible relational database built for the cloud
- Amazon Aurora is up to five times faster than standard MySQL databases and three times faster than standard PostgreSQL databases
- Amazon Aurora features a distributed, fault-tolerant, self-healing storage system that auto-scales up to 64TB per database instance

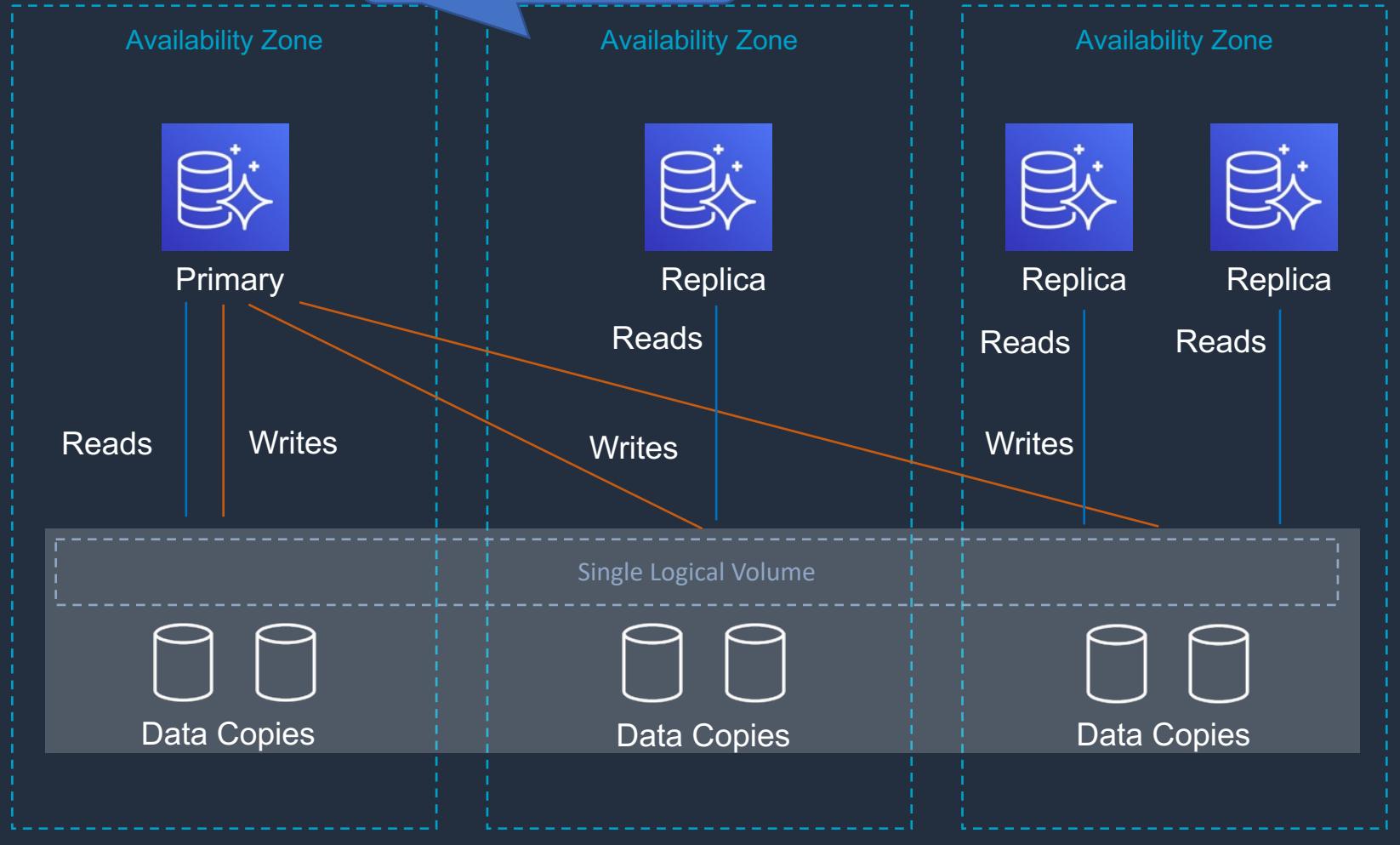


# Amazon Aurora



Region

Aurora Replicas are  
within a region



## Aurora Fault Tolerance

- Fault tolerance across 3 AZs
- Single logical volume
- Aurora Replicas scale-out read requests
- Can **promote** Aurora Replica to be a new primary or create new primary
- Can use **Auto Scaling** to add replicas



# Amazon Aurora Key Features

Aurora Feature	Benefit
<b>High performance and scalability</b>	Offers high performance, self-healing storage that scales up to 64TB, point-in-time recovery and continuous backup to S3
<b>DB compatibility</b>	Compatible with existing MySQL and PostgreSQL open source databases
<b>Aurora Replicas</b>	In-region read scaling and failover target – up to 15 (can use Auto Scaling)
<b>MySQL Read Replicas</b>	Cross-region cluster with read scaling and failover target – up to 5 (each can have up to 15 Aurora Replicas)
<b>Global Database</b>	Cross-region cluster with read scaling (fast replication / low latency reads). Can remove secondary and promote
<b>Multi-Master</b>	Scales out writes within a region. In preview currently and will not appear on the exam
<b>Serverless</b>	On-demand, autoscaling configuration for Amazon Aurora - does not support read replicas or public IPs (can only access through VPC or Direct Connect - not VPN)

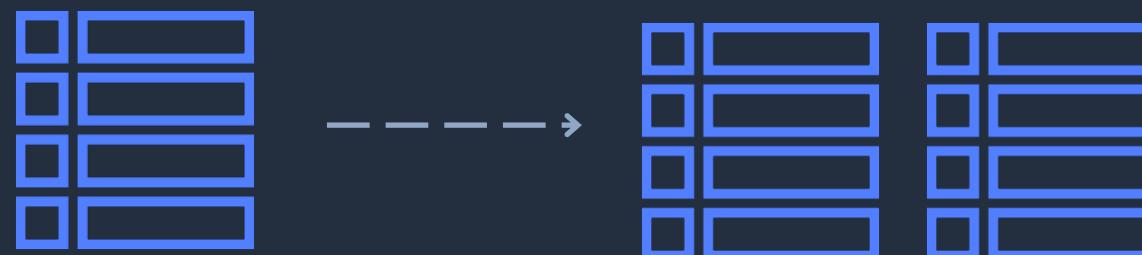
# Amazon DynamoDB





# Amazon DynamoDB

- Fully managed NoSQL database service
- Key/value store and document store
- It is a non-relational, key-value type of database
- Fully serverless service
- Push button scaling



DynamoDB Table



# Amazon DynamoDB

---

- DynamoDB is made up of:

- Tables
- Items
- Attributes

userid	orderid	book	price	date
user001	1000092	ISBN100..	9.99	2020.04..
user002	1000102	ISBN100..	24.99	2020.03..
user003	1000168	ISBN2X0..	12.50	2020.04..



# Amazon DynamoDB Key Features

---

---

DynamoDB Feature	Benefit
Serverless	Fully managed, fault tolerant, service
Highly available	99.99% availability SLA – 99.999% for Global Tables!
NoSQL type of database with Name / Value structure	Flexible schema, good for when data is not well structured or unpredictable
Horizontal scaling	Seamless scalability to any scale with push button scaling or Auto Scaling
DynamoDB Accelerator (DAX)	Fully managed in-memory cache for DynamoDB that increases performance (microsecond latency)
Backup	Point-in-time recovery down to the second in last 35 days; On-demand backup and restore
Global Tables	Fully managed multi-region, multi-master solution

# Create Amazon DynamoDB Table

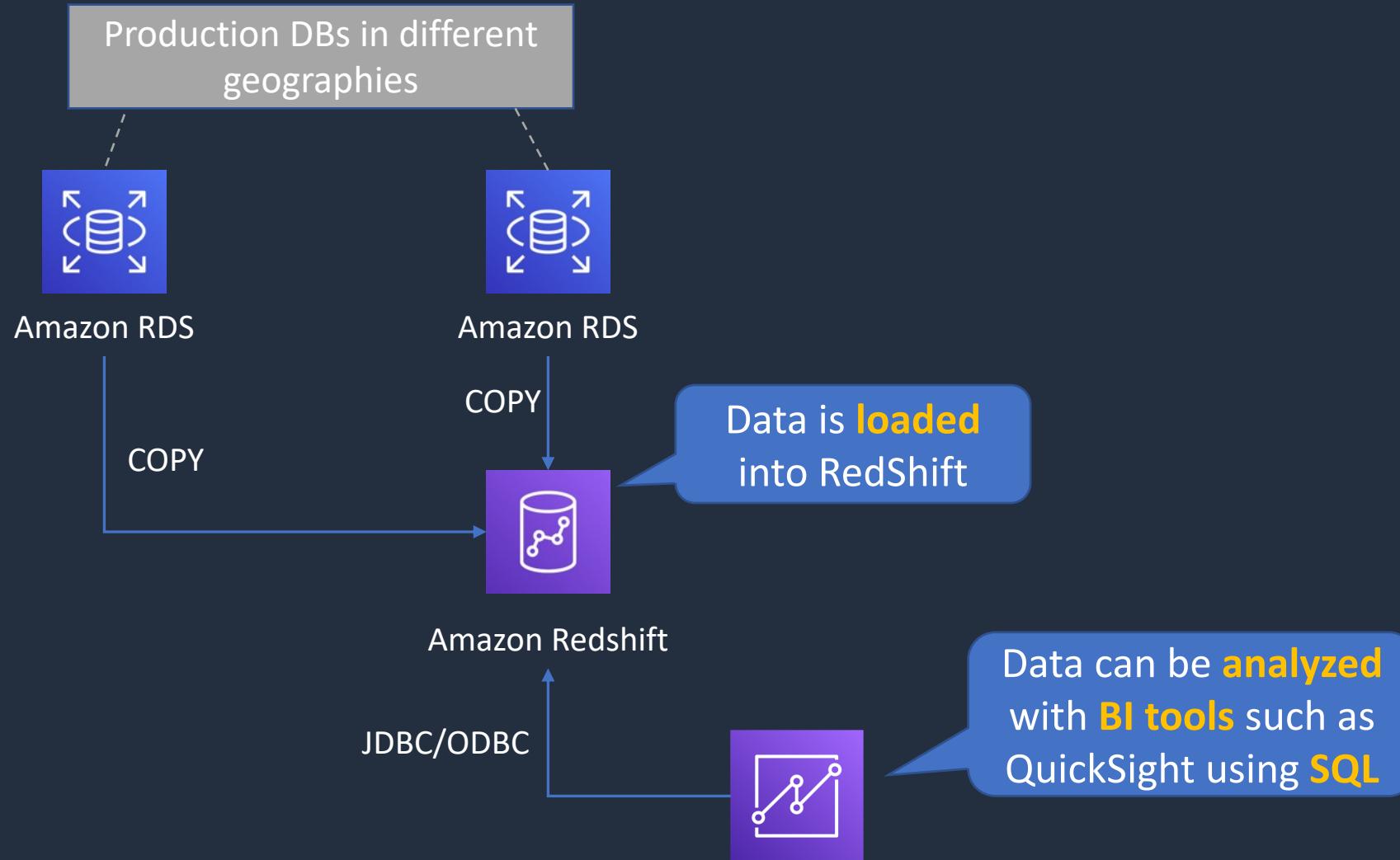


# Amazon RedShift





# Amazon Redshift





# Amazon RedShift

---

---

- Amazon Redshift is a fast, fully managed data warehouse that makes it simple and cost-effective to analyze all your data using standard SQL and existing Business Intelligence (BI) tools
- RedShift is a SQL based data warehouse used for analytics applications
- RedShift is a relational database that is used for Online Analytics Processing (OLAP) use cases
- RedShift uses Amazon EC2 instances, so you must choose an instance family/type
- RedShift always keeps three copies of your data
- RedShift provides continuous/incremental backups

# Amazon Elastic Map Reduce (EMR)





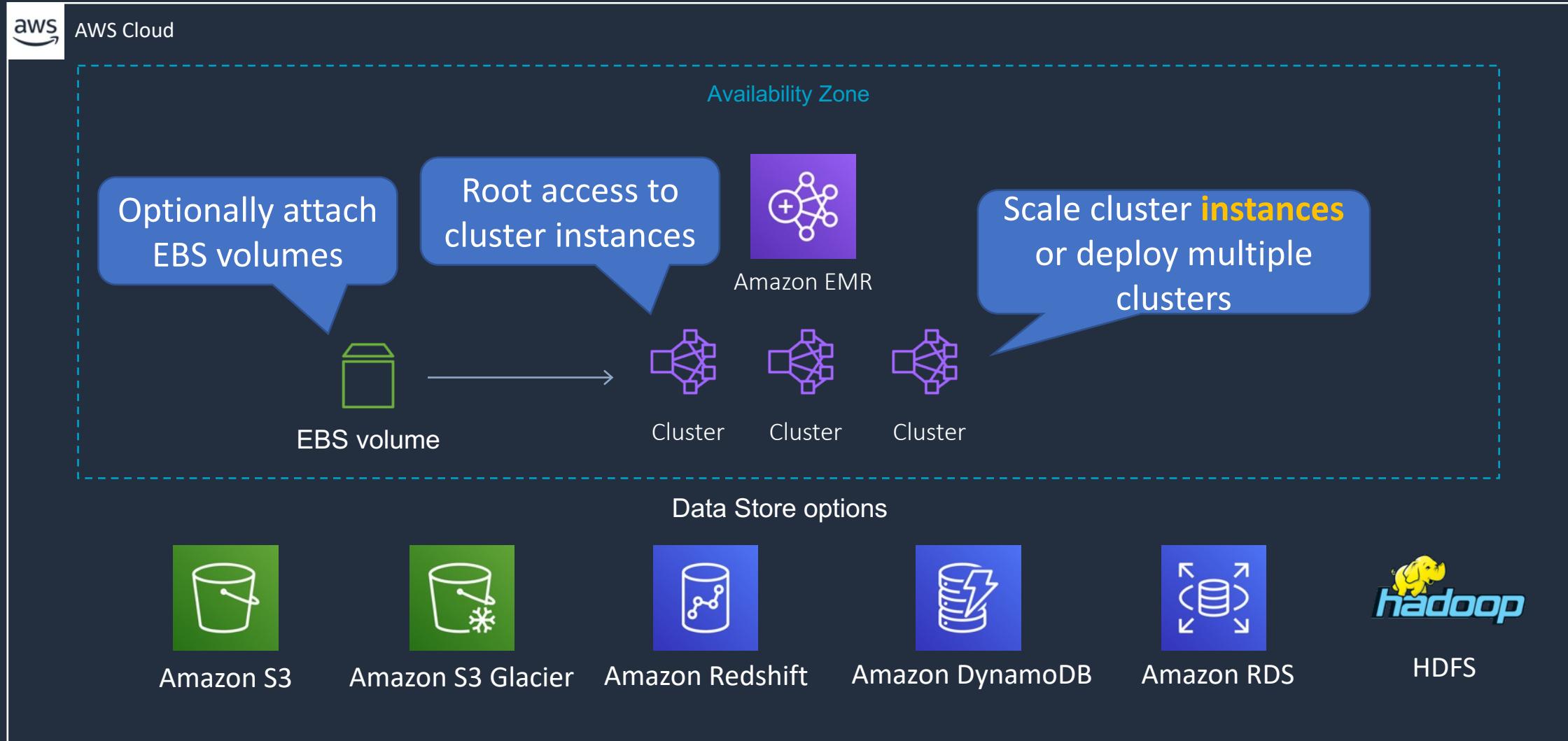
# Amazon EMR

---

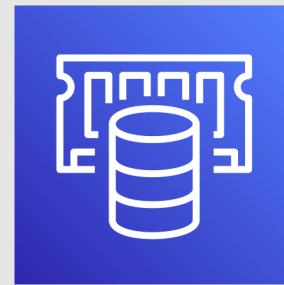
- Managed cluster platform that simplifies running big data frameworks including **Apache Hadoop** and **Apache Spark**
- Used for processing data for analytics and business intelligence
- Can also be used for transforming and moving large amounts of data
- Performs extract, transform, and load (ETL) functions



# Amazon EMR



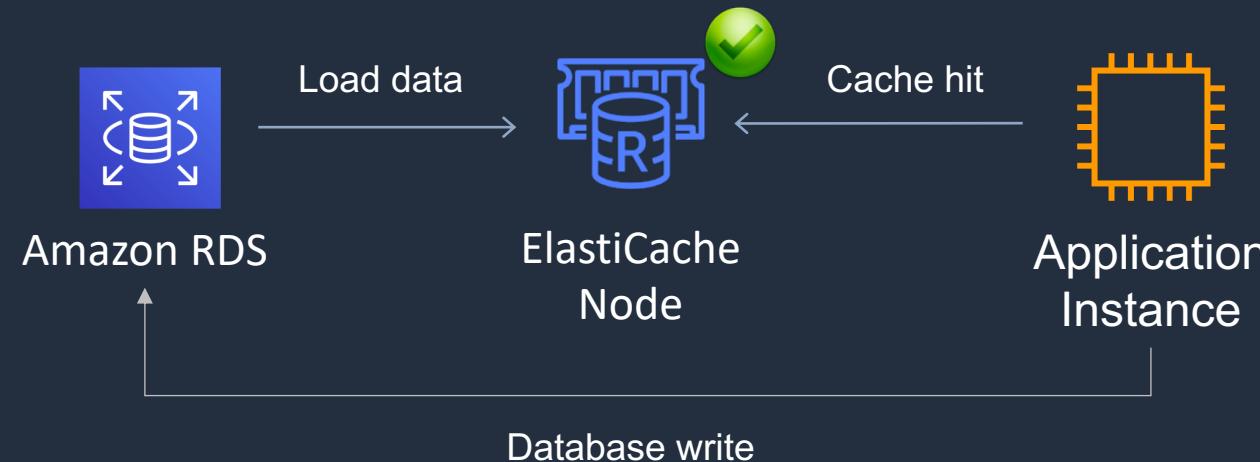
# Amazon ElastiCache





# Amazon ElastiCache

- Fully managed implementations **Redis** and **Memcached**
- ElastiCache is a **key/value** store
- In-memory database offering high performance and low latency
- Can be put in front of databases such as RDS and DynamoDB



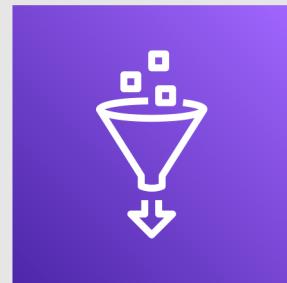
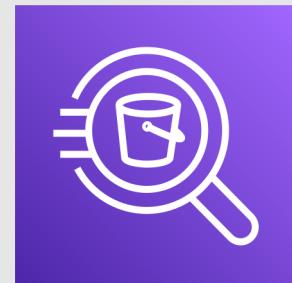


# Amazon ElastiCache

- ElastiCache nodes run on Amazon EC2 instances, so you must choose an instance family/type

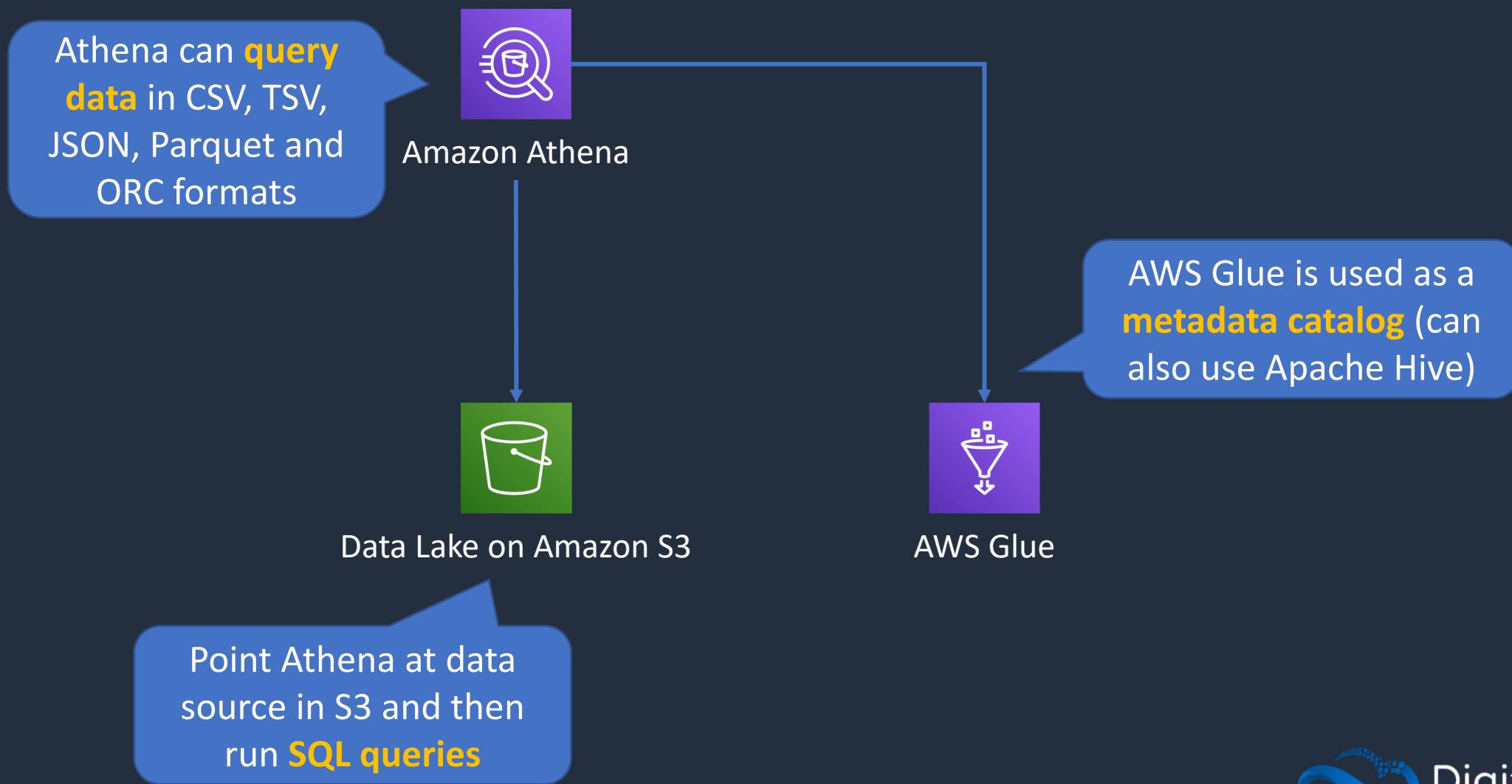
Use Case	Benefit
Web session store	In cases with load-balanced web servers, store web session information in Redis so if a server is lost, the session info is not lost, and another web server can pick it up
Database caching	Use Memcached in front of AWS RDS to cache popular queries to offload work from RDS and return results faster to users
Leaderboards	Use Redis to provide a live leaderboard for millions of users of your mobile app
Streaming data dashboards	Provide a landing spot for streaming sensor data on the factory floor, providing live real-time dashboard displays

# Amazon Athena and AWS Glue





# Amazon Athena and AWS Glue





# Amazon Athena

---

- Athena queries data in S3 using SQL
- Can be connected to other data sources with Lambda
- Data can be in CSV, TSV, JSON, Parquet and ORC formats
- Uses a managed Data Catalog (AWS Glue) to store information and schemas about the databases and tables



# AWS Glue

---

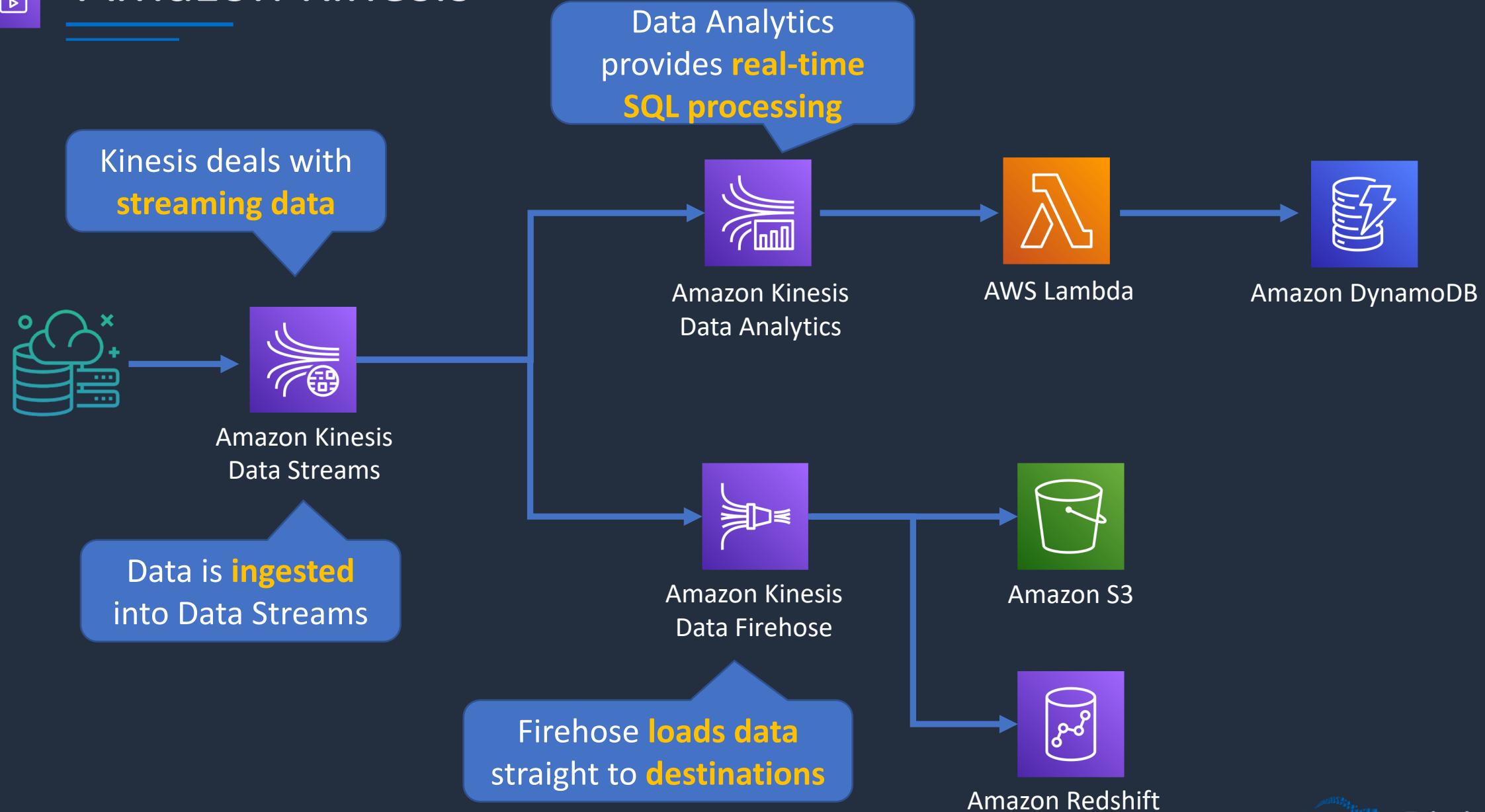
- Fully managed extract, transform and load (ETL) service
- Used for preparing data for analytics
- AWS Glue runs the ETL jobs on a fully managed, scale-out Apache Spark environment
- Works with data lakes (e.g. data on S3), data warehouses (including RedShift), and data stores (including RDS or EC2 databases)

# Amazon Kinesis





# Amazon Kinesis





# Amazon Kinesis

---

Examples of streaming data use cases include:

- Purchases from online stores
- Stock prices
- Game data (statistics and results as the gamer plays)
- Social network data
- Geospatial data (think uber.com)
- IoT sensor data



# Amazon Kinesis

---

## Kinesis Data Streams

- Producers send data which is stored in shards for up to 7 days
- Consumers process the data and save to another service

## Amazon Kinesis Data Firehose

- No shards, completely automated and elastically scalable
- Saves data directly to another service such as S3, Splunk, RedShift, or Elasticsearch

## Amazon Kinesis Data Analytics

- Provides real-time SQL processing for streaming data

# Other Databases and Analytics Services





# Other Databases and Analytics Services

---

## AWS Data Pipeline

- Processes and moves data between different AWS compute and storage services
- Save results to services including S3, RDS, DynamoDB, and EMR

## Amazon QuickSight

- Business intelligence (BI) service
- Create and publish interactive BI dashboards for Machine Learning-powered insights

## Amazon Neptune

- Fully managed graph database service



# Other Databases and Analytics Services

---

---

## Amazon DocumentDB

- Fully managed document database service (non-relational)
- Supports MongoDB workloads
- Queries and indexes JSON data

## Amazon QLDB

- Fully managed ledger database for immutable change history
- Provides cryptographically verifiable transaction logging

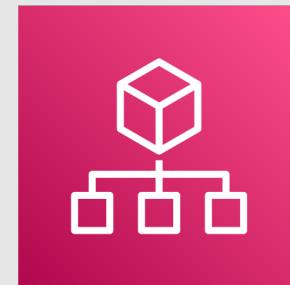
## Amazon Managed Blockchain

- Fully managed service for joining public and private networks using Hyperledger Fabric and Ethereum

# SECTION 12

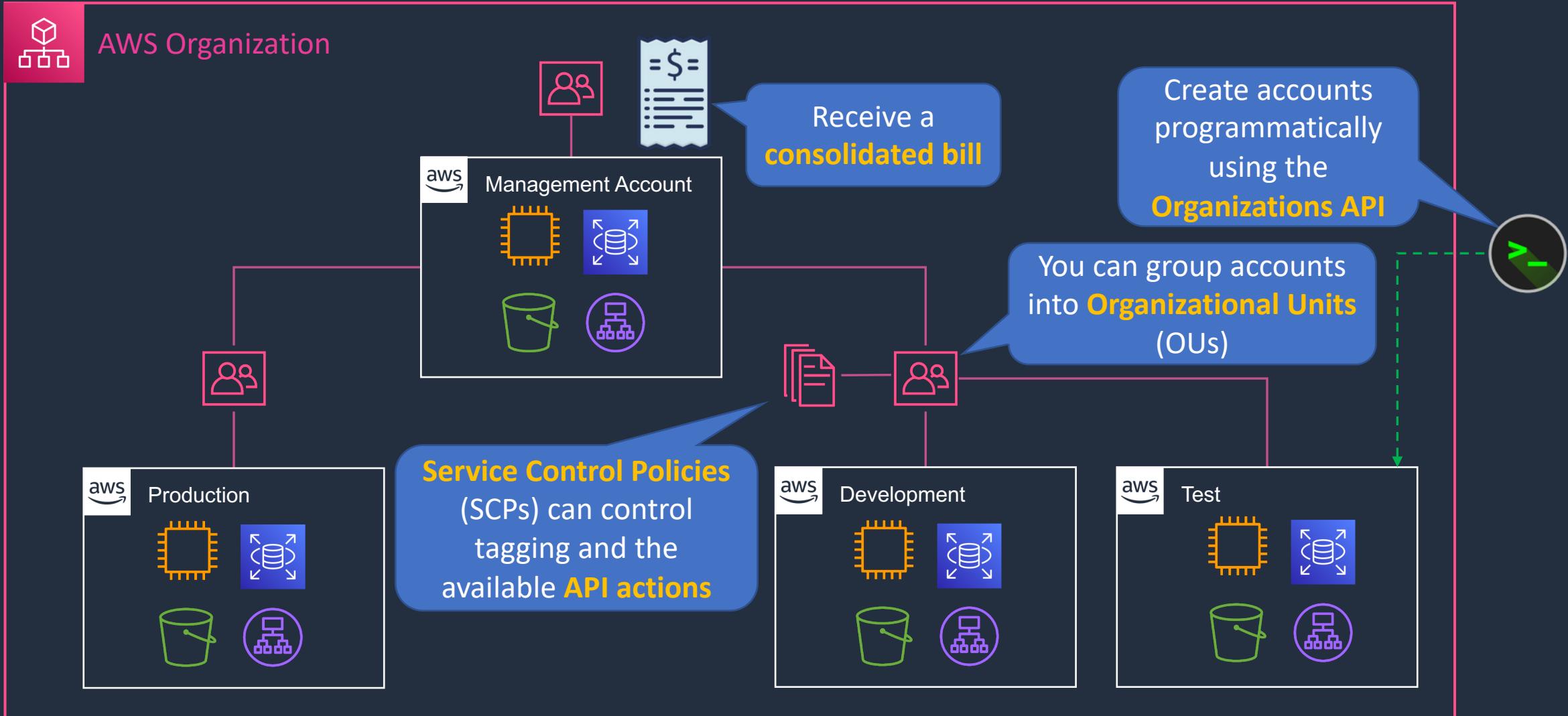
## Management and Governance

# AWS Organizations





# AWS Organizations





# AWS Organizations

---

---

- AWS organizations allows you to consolidate multiple AWS accounts into an organization that you create and centrally manage
- Available in two feature sets:
  - **Consolidated Billing**
  - **All features**
- Includes root accounts and organizational units
- Policies are applied to root accounts or OUs
- Consolidated billing includes:
  - **Paying Account** – independent and cannot access resources of other accounts
  - **Linked Accounts** – all linked accounts are independent

# AWS Organizations



# AWS Control Tower





# AWS Control Tower

- Simplifies the process of creating multi-account environments
- Sets up governance, compliance, and security guardrails for you
- Integrates with other services and features to setup the environment for you including:
  - AWS Organizations, SCPs, OUs, AWS Config, AWS CloudTrail, Amazon S3, Amazon SNS, AWS CloudFormation, AWS Service Catalog, AWS Single Sign-On (SSO)



# AWS Control Tower

Examples of guardrails AWS Control Tower can configure for you include:

- Disallowing public write access to Amazon Simple Storage Service (Amazon S3) buckets
- Disallowing access as a root user without multi-factor authentication
- Enabling encryption for Amazon EBS volumes attached to Amazon EC2 instances

# AWS Systems Manager





# AWS Systems Manager

---

---

- Manages many AWS resources including Amazon EC2, Amazon S3, Amazon RDS etc.
- Systems Manager Components:
  - **Automation**
  - **Run Command**
  - **Inventory**
  - **Patch Manager**
  - **Session Manager**
  - **Parameter Store**



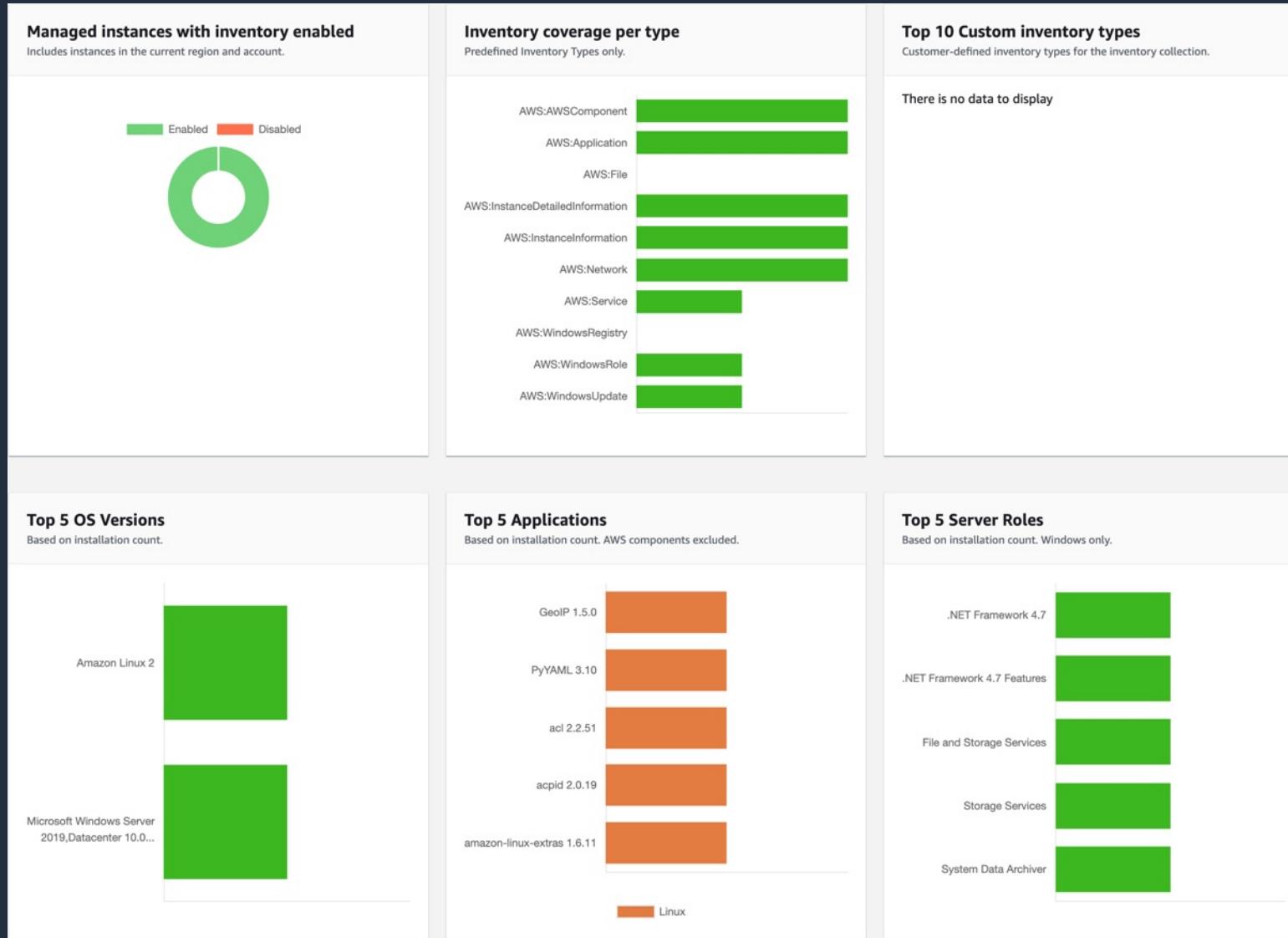
# AWS Systems Manager



This command checks  
for missing **updates**



# AWS Systems Manager



Inventory



# AWS Systems Manager

---

---

## Patch Manager

- Deploy operating system and software patches automatically across large groups of Amazon EC2 or on-premises instances

## Compliance

- Scan managed instances for patch compliance and configuration inconsistencies



# AWS Systems Manager

---

## Session Manager

- Secure remote management of your instances at scale without logging into your servers
- Replaces the need for bastion hosts, SSH, or remote PowerShell

## Parameter Store

- Parameter Store provides secure, hierarchical storage for configuration data management and secrets management

# Manage EC2 Instances



# AWS Service Catalog





# AWS Service Catalog

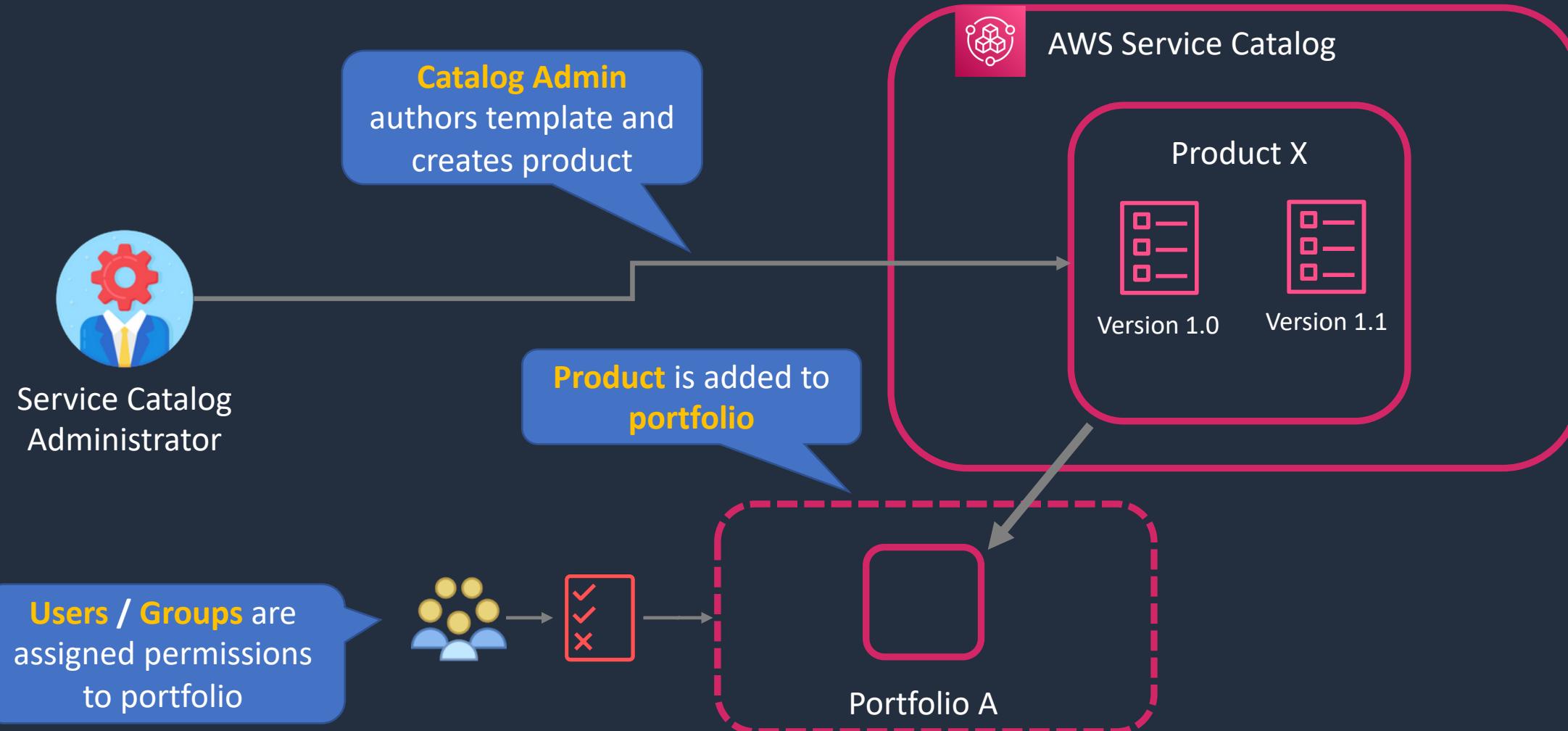
---

---

- AWS Service Catalog allows organizations to create and manage **catalogs of IT services** that are approved for use on AWS
- AWS Service Catalog allows you to **centrally manage** commonly deployed IT services
- IT services can include virtual machine images, servers, software, and databases and multi-tier application architectures
- Enables users to quickly deploy only the approved IT services they need



# AWS Service Catalog



# AWS Config

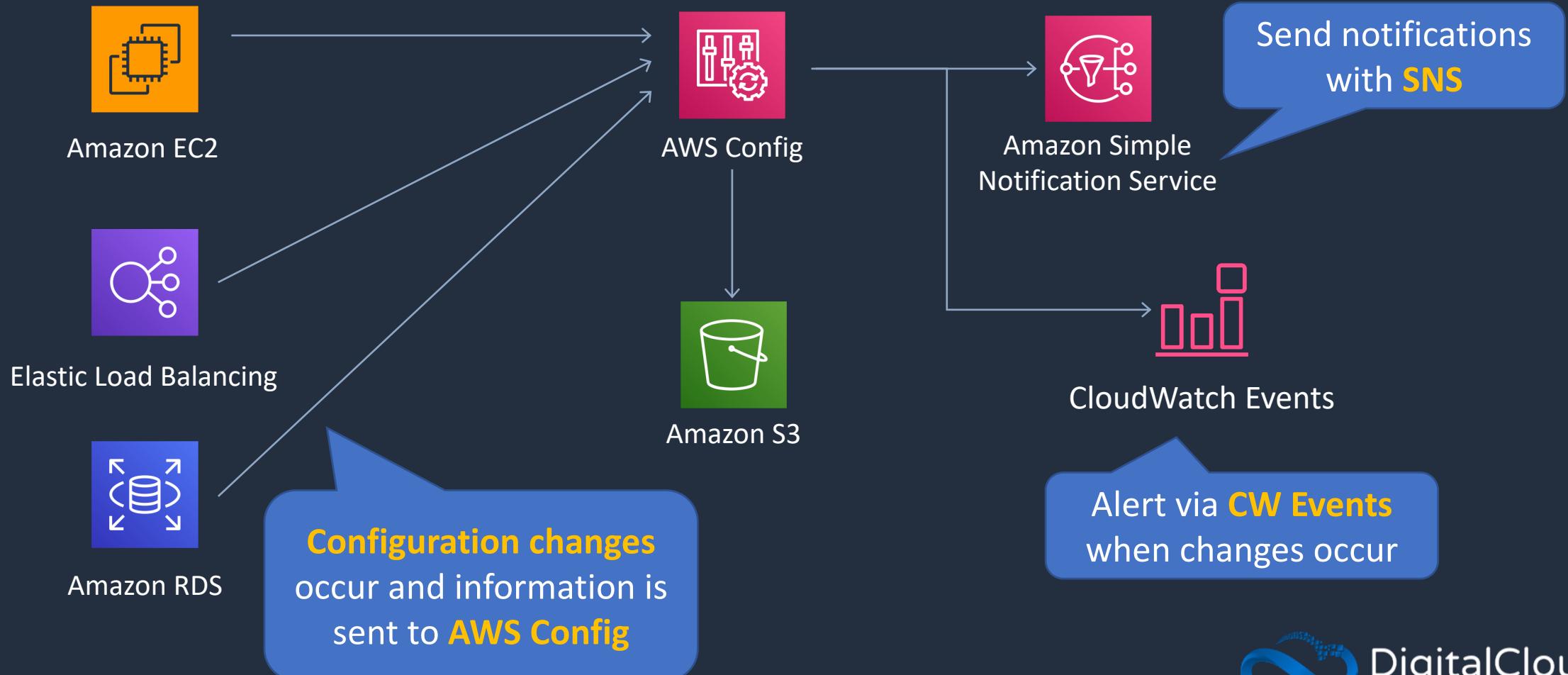




# AWS Config

AWS Config evaluates the **configuration** against desired configurations

Example Services:





# AWS Config

Example Rule	Description
s3-bucket-server-side-encryption-enabled	Checks that your Amazon S3 bucket either has S3 default encryption enabled or that the S3 bucket policy explicitly denies put-object requests without server side encryption
restricted-ssh	Checks whether security groups that are in use disallow unrestricted incoming SSH traffic
rds-instance-public-access-check	Checks whether the Amazon Relational Database Service (RDS) instances are not publicly accessible
cloudtrail-enabled	Checks whether AWS CloudTrail is enabled in your AWS account

# Configuration Compliance



# AWS Trusted Advisor





# AWS Trusted Advisor

---

---

- Trusted Advisor is an online resource that helps to reduce cost, increase performance and improve security by optimizing your AWS environment
- Trusted Advisor provides real time guidance to help you provision your resources following best practices
- Advisor will advise you on **Cost Optimization**, **Performance**, **Security**, and **Fault Tolerance**

# AWS Health API and Dashboards





# AWS Personal Health Dashboard

---

- AWS Personal Health Dashboard provides alerts and remediation guidance when AWS is experiencing events that may **impact you**
- Personal Health Dashboard gives you a **personalized** view into the performance and availability of the AWS services underlying your AWS resources
- Also provides proactive notification to help you plan for scheduled activities



# AWS Service Health Dashboard

Not personalized information so may not be relevant to you

No proactive notification of scheduled activities

## Current Status - Jun 7, 2020 PDT

Amazon Web Services publishes our most up-to-the-minute information on service availability in the table below. Check back here any time to get current status information, or subscribe to an RSS feed to be notified of interruptions to each individual service. If you are experiencing a real-time, operational issue with one of our services that is not described below, please inform us by clicking on the "Contact Us" link to submit a service issue report. All dates and times are Pacific Time (PST/PDT).

North America	South America	Europe	Africa	Asia Pacific	Middle East	Contact Us
Recent Events	Details				RSS	
No recent events.						
Remaining Services	Details				RSS	
Alexa for Business (N. Virginia)	Service is operating normally					
Amazon API Gateway (Montreal)	Service is operating normally					
Amazon API Gateway (N. California)	Service is operating normally					
Amazon API Gateway (N. Virginia)	Service is operating normally					
Amazon API Gateway (Ohio)	Service is operating normally					
Amazon API Gateway (Oregon)	Service is operating normally					
Amazon AppStream 2.0 (N. Virginia)	Service is operating normally					
Amazon AppStream 2.0 (Oregon)	Service is operating normally					
Amazon Athena (Montreal)	Service is operating normally					
Amazon Athena (N. Virginia)	Service is operating normally					
Amazon Athena (Ohio)	Service is operating normally					
Amazon Athena (Oregon)	Service is operating normally					

Shows current status information on service availability

# SECTION 13

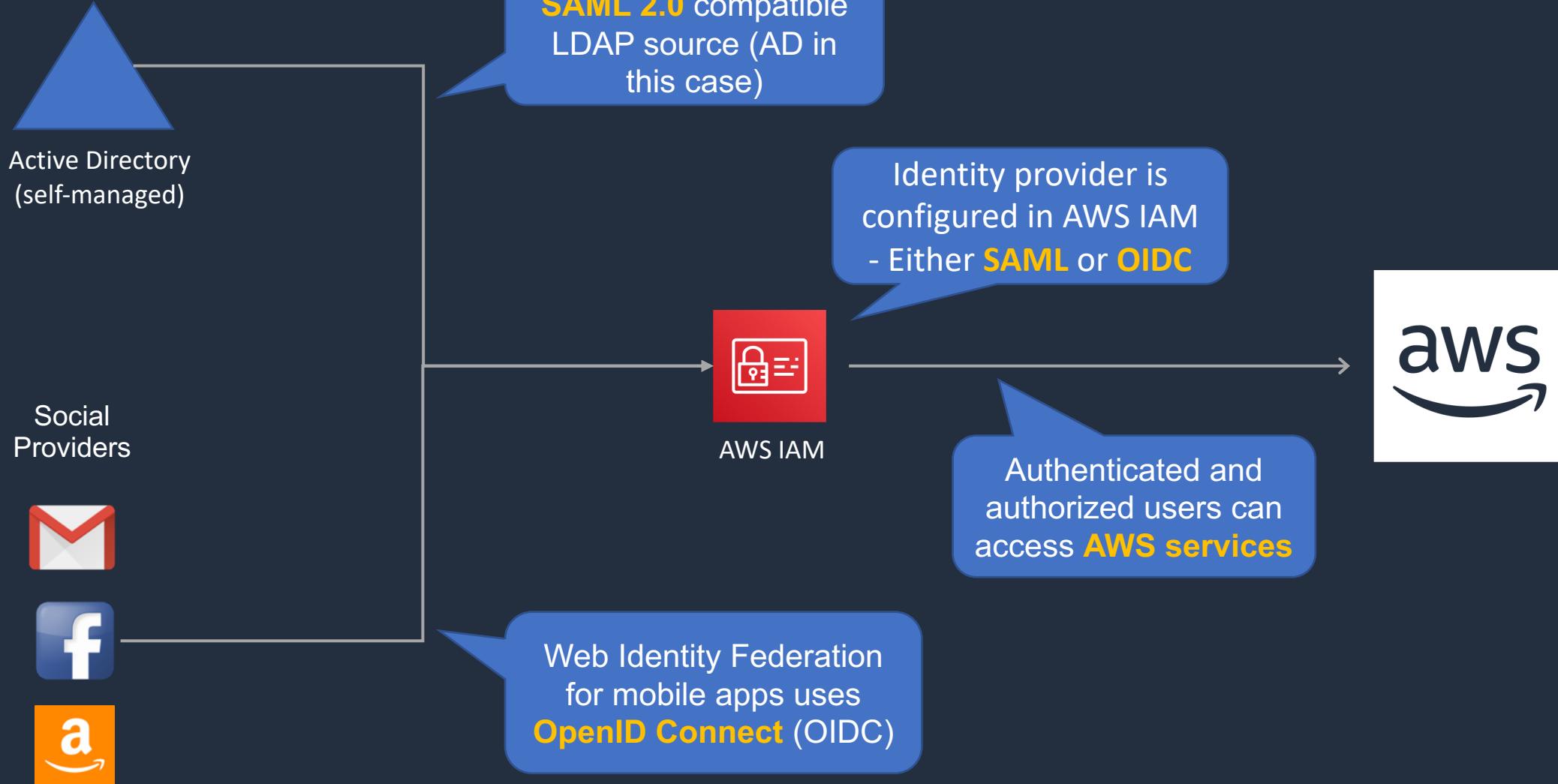
## AWS Cloud Security and Identity

# Identity Providers and Federation



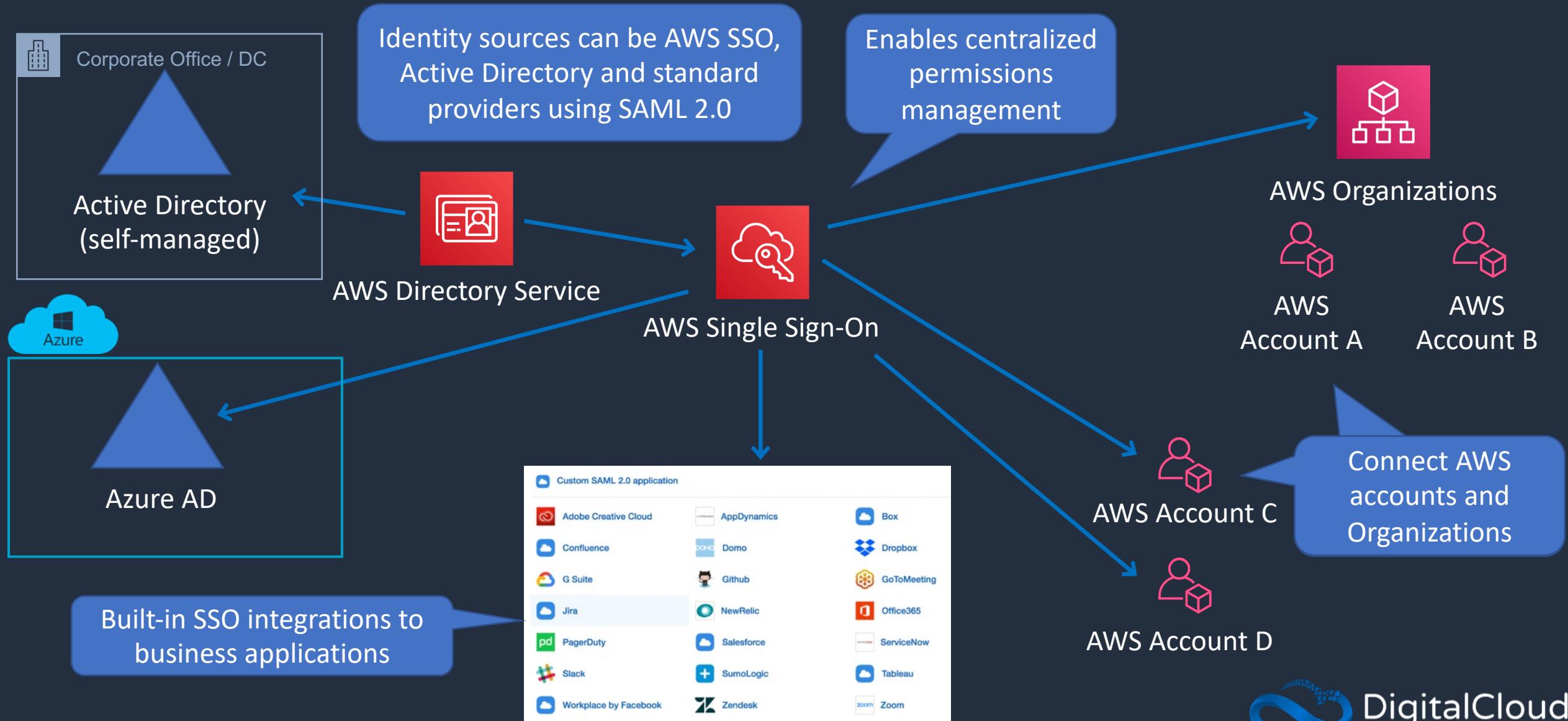


# Federation with IAM



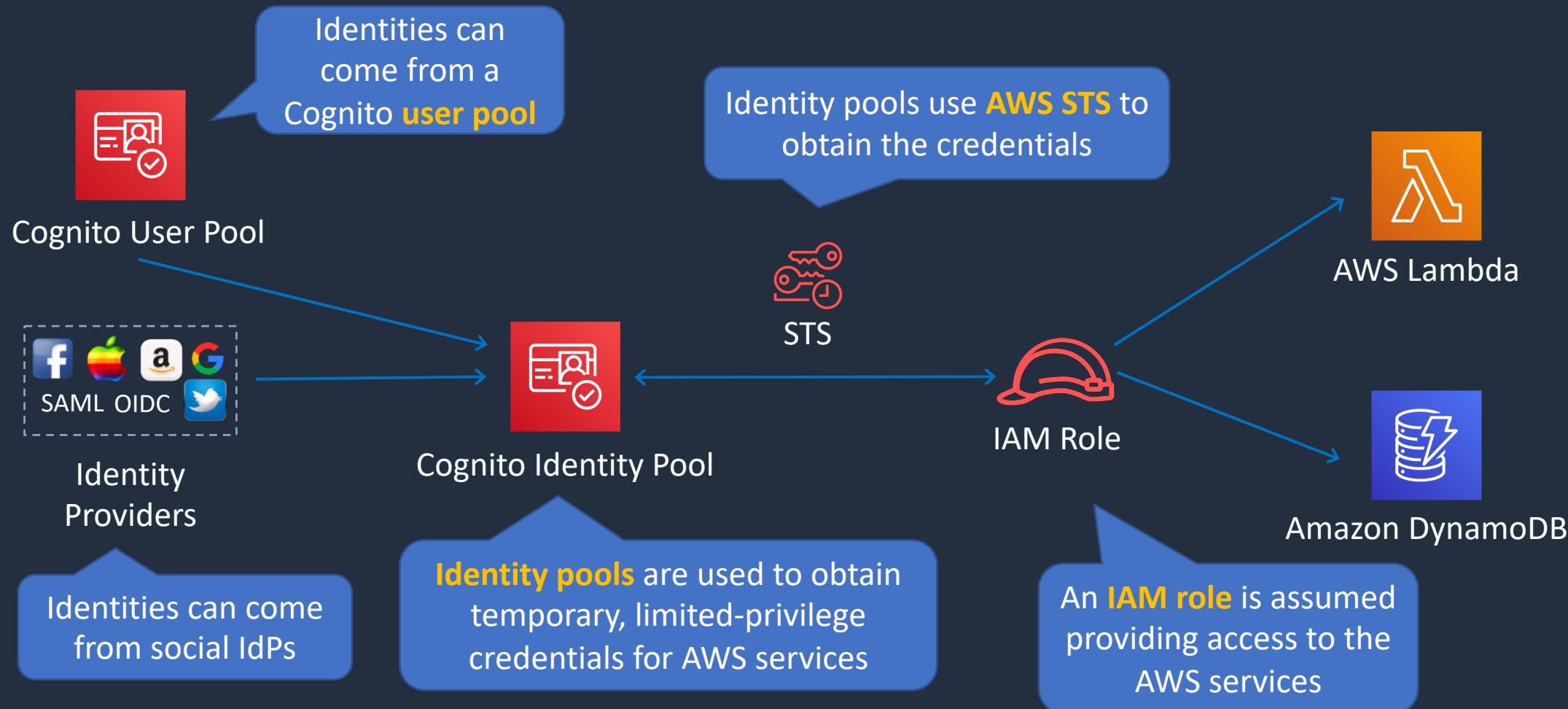


# AWS Single Sign-On (SSO)

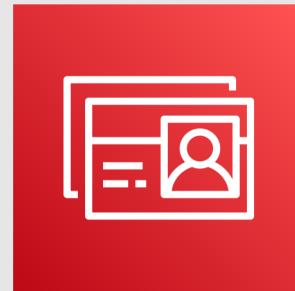




# AWS Cognito (Mobile and Web Apps)



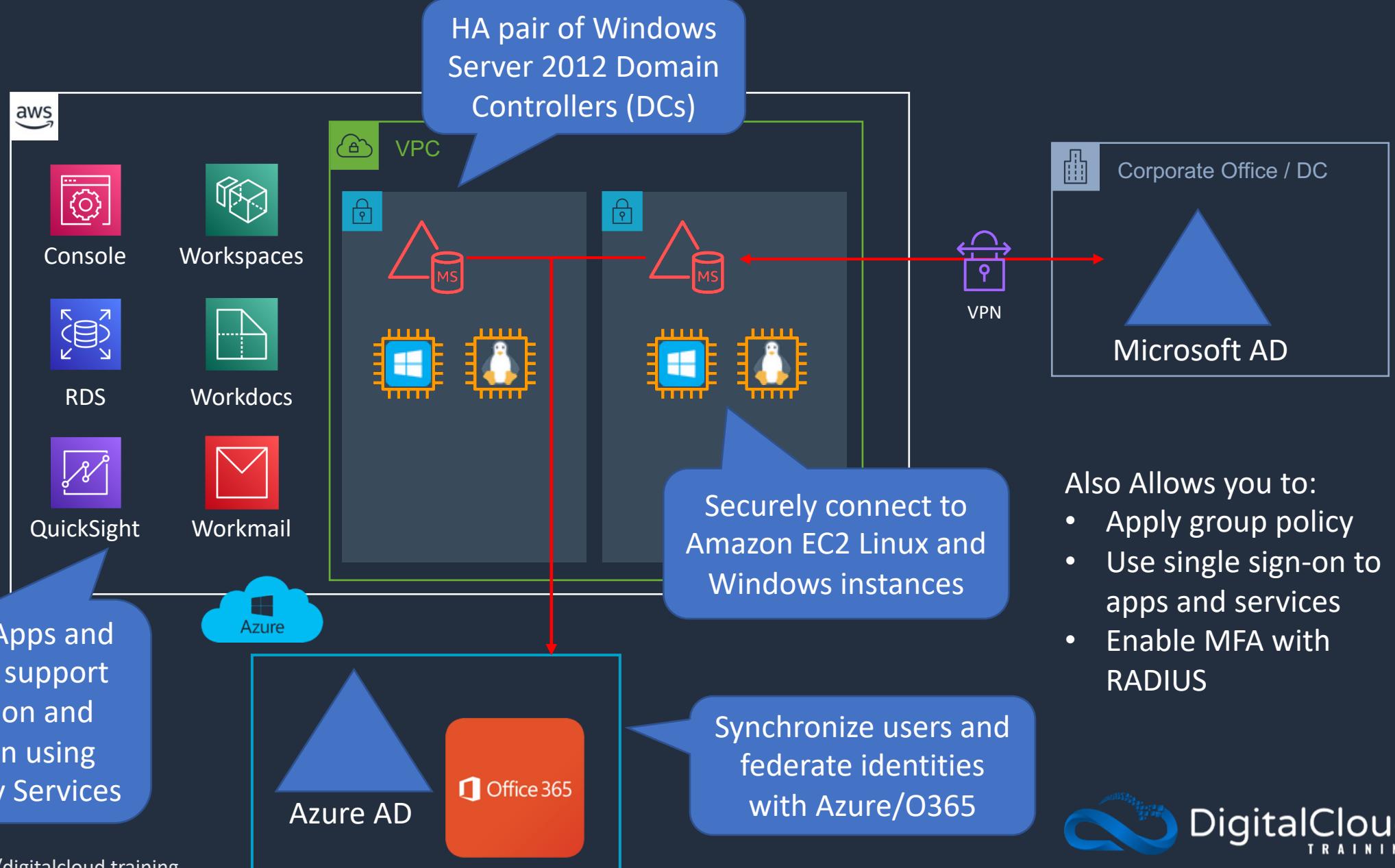
# AWS Directory Service





# AWS Managed Microsoft Active Directory

Managed implementation of Microsoft Active Directory running on Windows Server 2012 R2



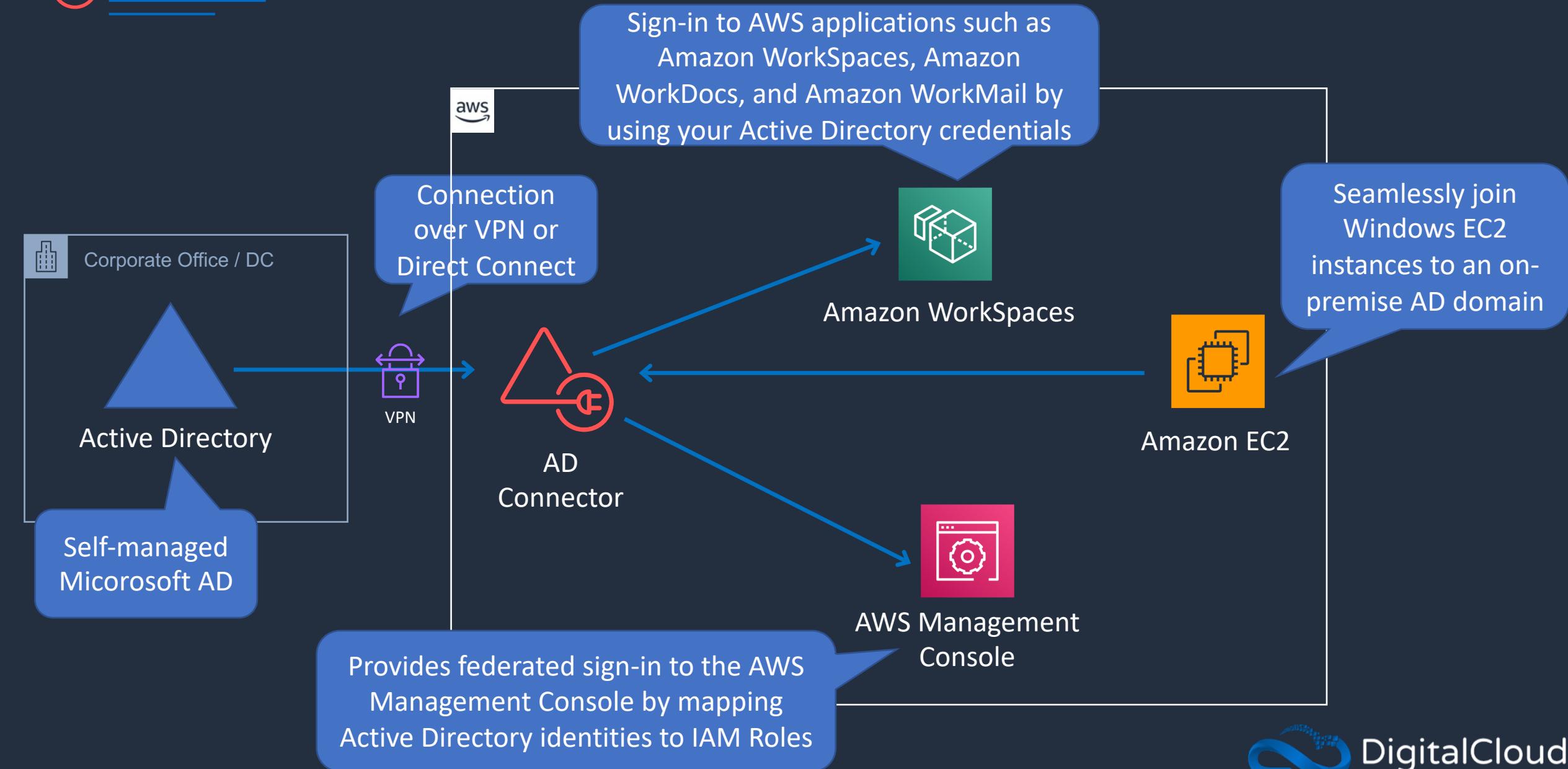


# AWS Managed Microsoft Active Directory

- Fully managed AWS services on AWS infrastructure
- Best choice if you have more than 5000 users and/or need a trust relationship set up
- You can setup trust relationships to extend authentication from on-premises Active Directories into the AWS cloud
- On-premise users and groups can access resources in either domain using SSO
- Can be used as a standalone AD in the AWS cloud



# AD Connector





# AD Connector

---

- AD Connector is a directory gateway for redirecting directory requests to your on-premise Active Directory.
- AD Connector eliminates the need for directory synchronization and the cost and complexity of hosting a federation infrastructure
- Connects your existing on-premise AD to AWS
- Best choice when you want to use an existing Active Directory with AWS services.



# AWS Directory Services Options

---

Directory Service	Service Description	Use Case
AWS Directory Service for Microsoft Active Directory	AWS-managed full Microsoft AD running on Windows Server 2012 R2	Enterprises that want hosted Microsoft Active Directory
AD Connector	Allows on-premises users to log into AWS services with their existing AD credentials	Single sign-on for on-premises employees
Simple AD	Low scale, low cost, AD implementation based on Samba	Simple user directory, or you need LDAP compatibility

# Protecting Secrets





# Systems Manager Parameter Store

- Provides secure, hierarchical storage for configuration data management and secrets management
- It is highly scalable, available, and durable
- You can store data such as passwords, database strings, and license codes as parameter values
- You can store values as plaintext (unencrypted data) or ciphertext (encrypted data)
- You can then reference values by using the unique name that you specified when you created the parameter



Amazon EC2



Parameter  
Store



Retrieve database connection string



Amazon RDS





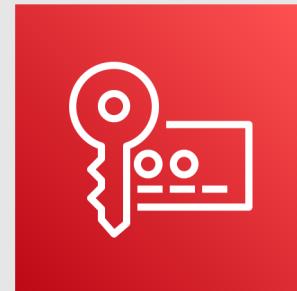
# AWS Secrets Manager

---

---

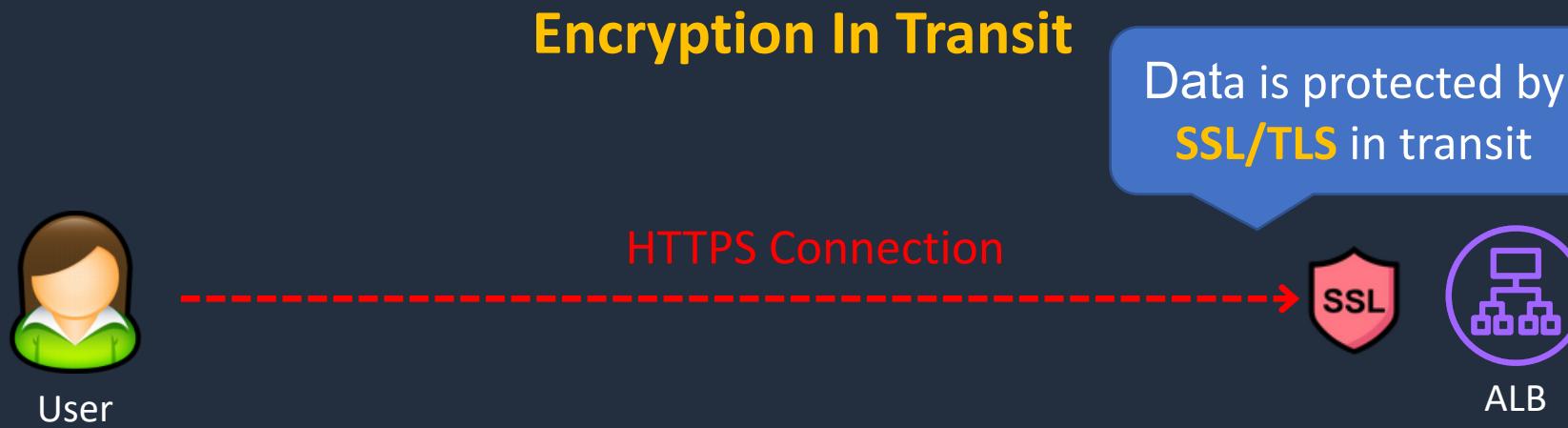
- Similar to Parameter Store
- Allows native and automatic rotation of keys
- Fine-grained permissions
- Central auditing for secret rotation

# Encryption



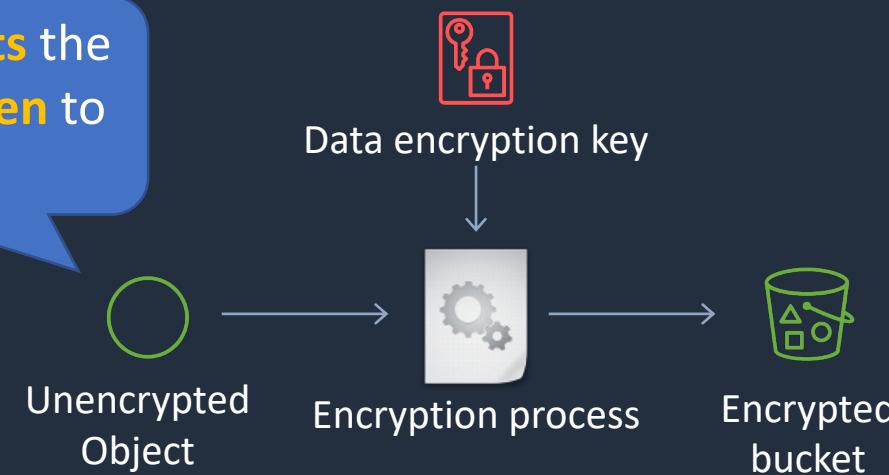


# Encryption In Transit vs At Rest



## Encryption At Rest

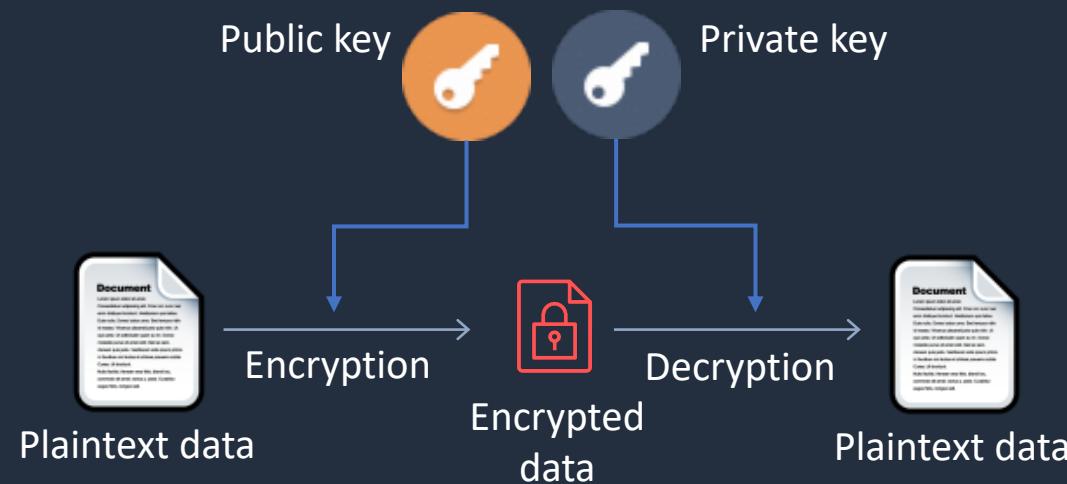
Amazon S3 encrypts the object as it is written to the bucket it





# Asymmetric Encryption

- Asymmetric encryption is also known as public key cryptography
- Messages encrypted with the public key can only be decrypted with the private key
- Messages encrypted with the private key can be decrypted with the public key
- Examples include SSL/TLS and SSH





# AWS Certificate Manager (ACM)

---

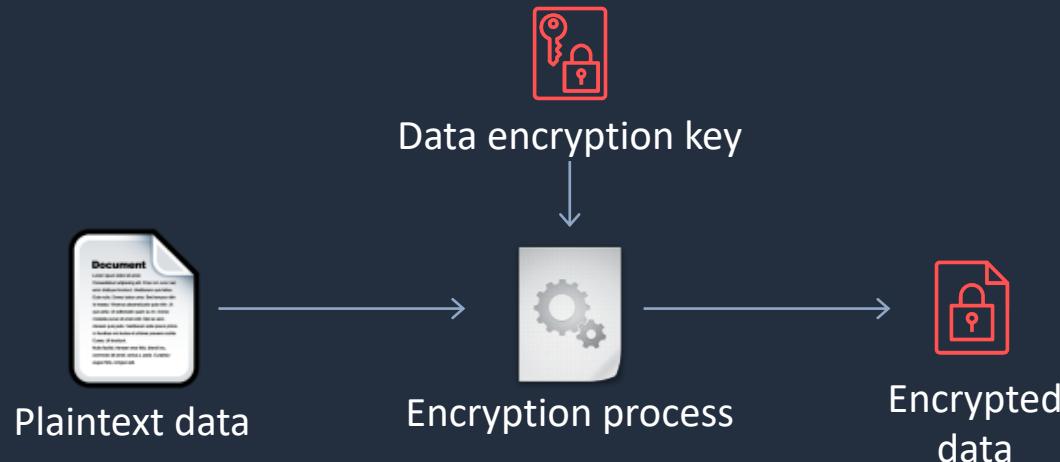
---

- Create, store and renew SSL/TLS X.509 certificates
- Single domains, multiple domain names and wildcards
- Integrates with several AWS services including:
  - **Elastic Load Balancing**
  - **Amazon CloudFront**
  - **AWS Elastic Beanstalk**
  - **AWS Nitro Enclaves**
  - **AWS CloudFormation**

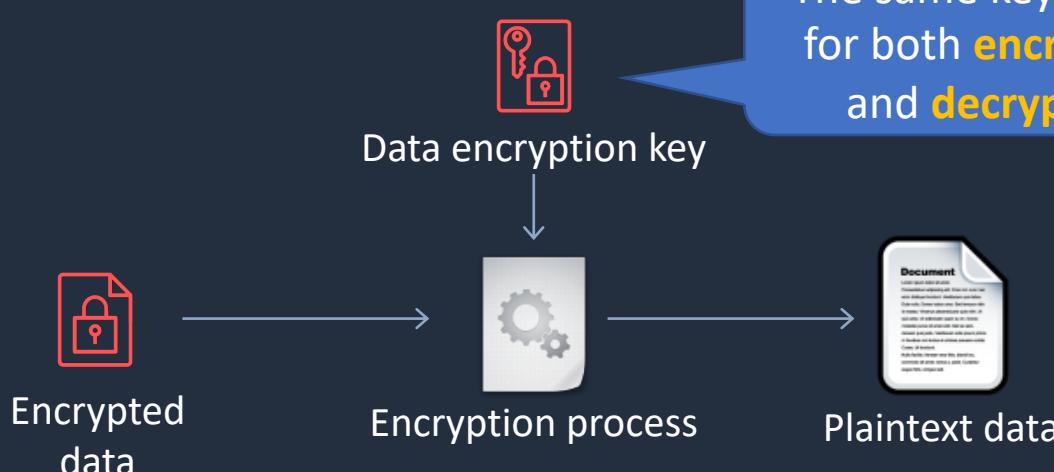


# Symmetric Encryption

## Encryption



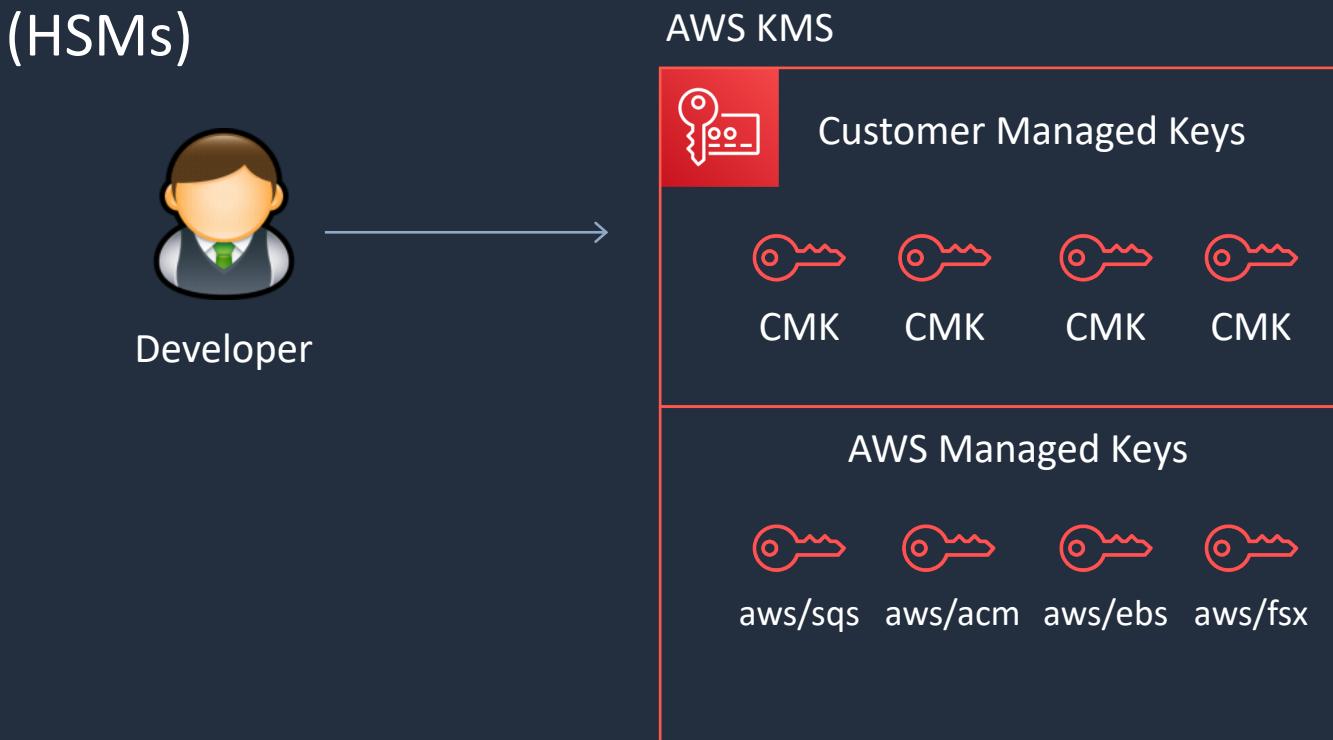
## Decryption





# AWS Key Management Service (KMS)

- Create and manage **symmetric** and **asymmetric** encryption keys
- The **customer master keys** (CMKs) are protected by hardware security modules (HSMs)





# AWS CloudHSM

- AWS CloudHSM is a cloud-based hardware security module (HSM)
- Generate and use your own encryption keys on the AWS Cloud
- Manage your own encryption keys using FIPS 140-2 Level 3 validated HSMs
- CloudHSM runs in your VPC

	CloudHSM	AWS KMS
Tenancy	Single-tenant HSM	Multi-tenant AWS service
Availability	Customer-managed durability and available	Highly available and durable key storage and management
Root of Trust	Customer managed root of trust	AWS managed root of trust
FIPS 140-2	Level 3	Level 2 / Level 3 in some areas
3 <sup>rd</sup> Party Support	Broad 3 <sup>rd</sup> Party Support	Broad AWS service support

# Encryption on AWS



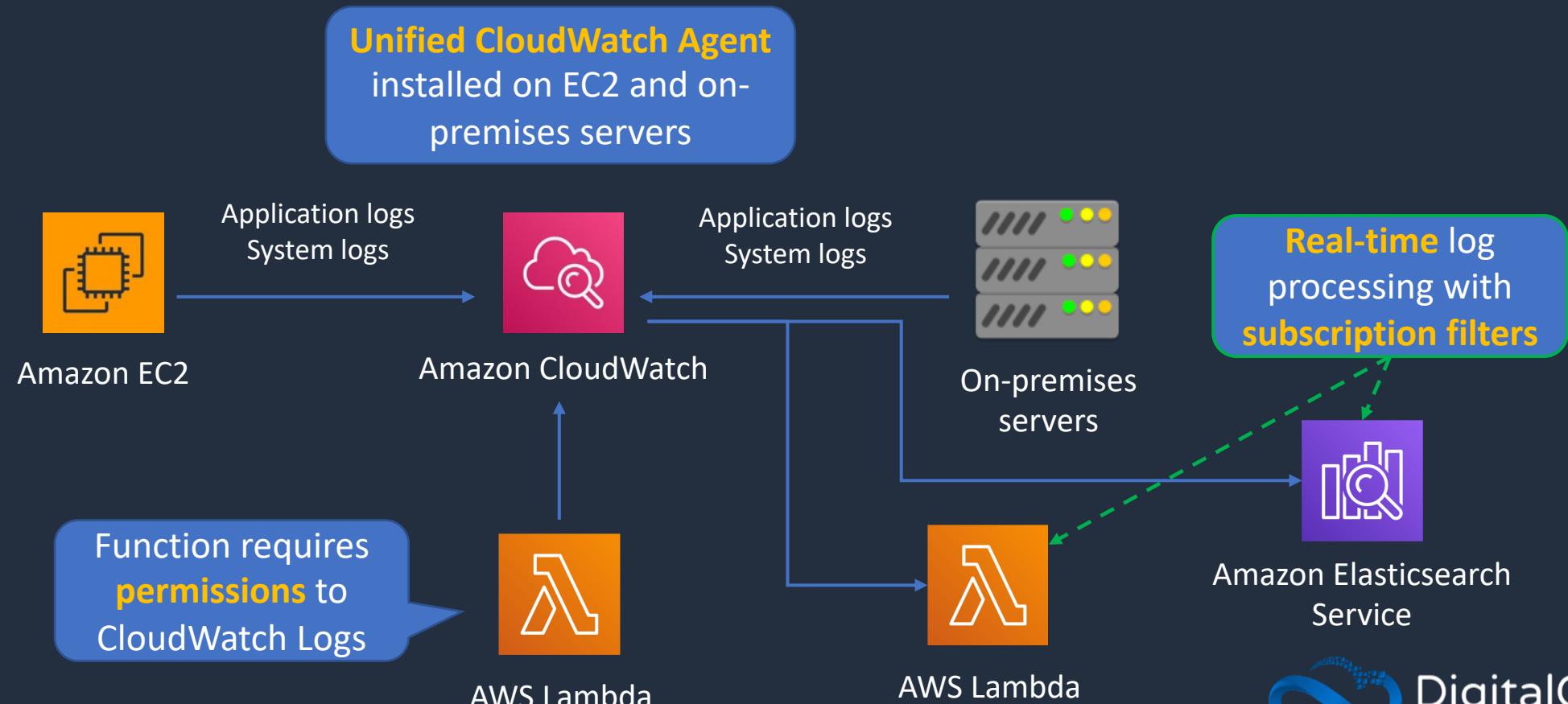
# Logging and Auditing





# Amazon CloudWatch Logs

- Gather application and system logs in CloudWatch
- Defined expiration policies and KMS encryption





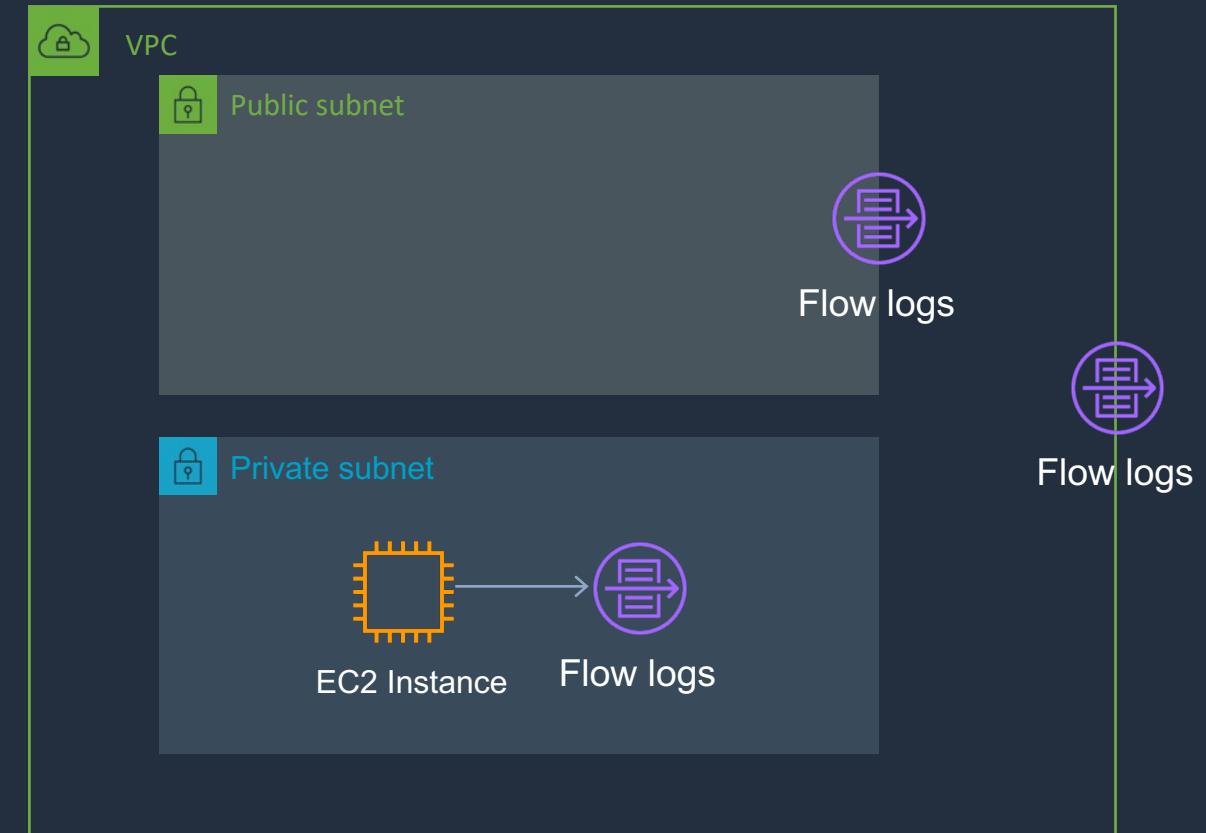
# AWS CloudTrail

- CloudTrail logs **API activity** for auditing
- By default, management events are logged and retained for 90 days
- A **CloudTrail Trail** logs any events to S3 for indefinite retention
- Trail can be within Region or all Regions
- CloudWatch Events can be triggered based on API calls in CloudTrail
- Events can be streamed to CloudWatch Logs



# VPC Flow Logs

- Flow Logs capture information about the IP traffic going to and from network interfaces in a VPC
- Flow log data is stored using Amazon CloudWatch Logs
- Flow logs can be created at the following levels:
  - VPC
  - Subnet
  - Network interface





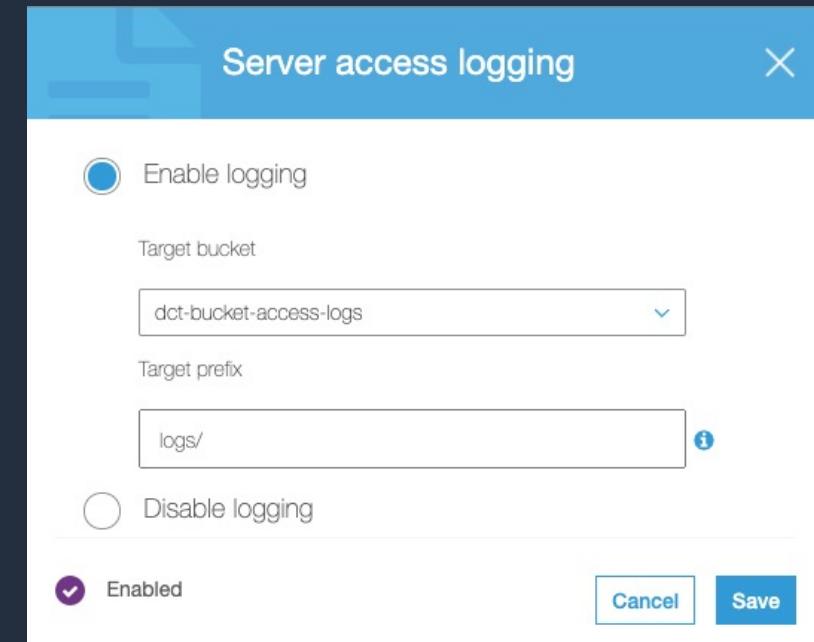
# Access Logs

## Elastic Load Balancing Access Logs

- capture detailed information about requests sent to the load balancer
- Use to analyze traffic patterns and troubleshoot issues
- Can identify requester, IP, request type etc.
- Can be optionally stored and retained in S3.

## S3 Access Logs

- Provides detailed records for the requests that are made to a bucket
- Details include the requester, bucket name, request time, request action, response status, and error code (if applicable)
- Disabled by default



# AWS CloudTrail



# Detect and Respond





# Amazon Detective

---

- Analyze, investigate, and quickly identify the root cause of potential security issues or suspicious activities
- Automatically collects data from AWS resources
- Uses machine learning, statistical analysis, and graph theory
- Creates a unified, interactive view of resources, users and interactions between them
- Data sources include VPC Flow Logs, CloudTrail, and GuardDuty



# AWS GuardDuty

- Intelligent threat detection service
- Detects account compromise, instance compromise, malicious reconnaissance, and bucket compromise
- Continuous monitoring for events across:
  - **AWS CloudTrail Management Events**
  - **AWS CloudTrail S3 Data Events**
  - **Amazon VPC Flow Logs**
  - **DNS Logs**



# Amazon Macie

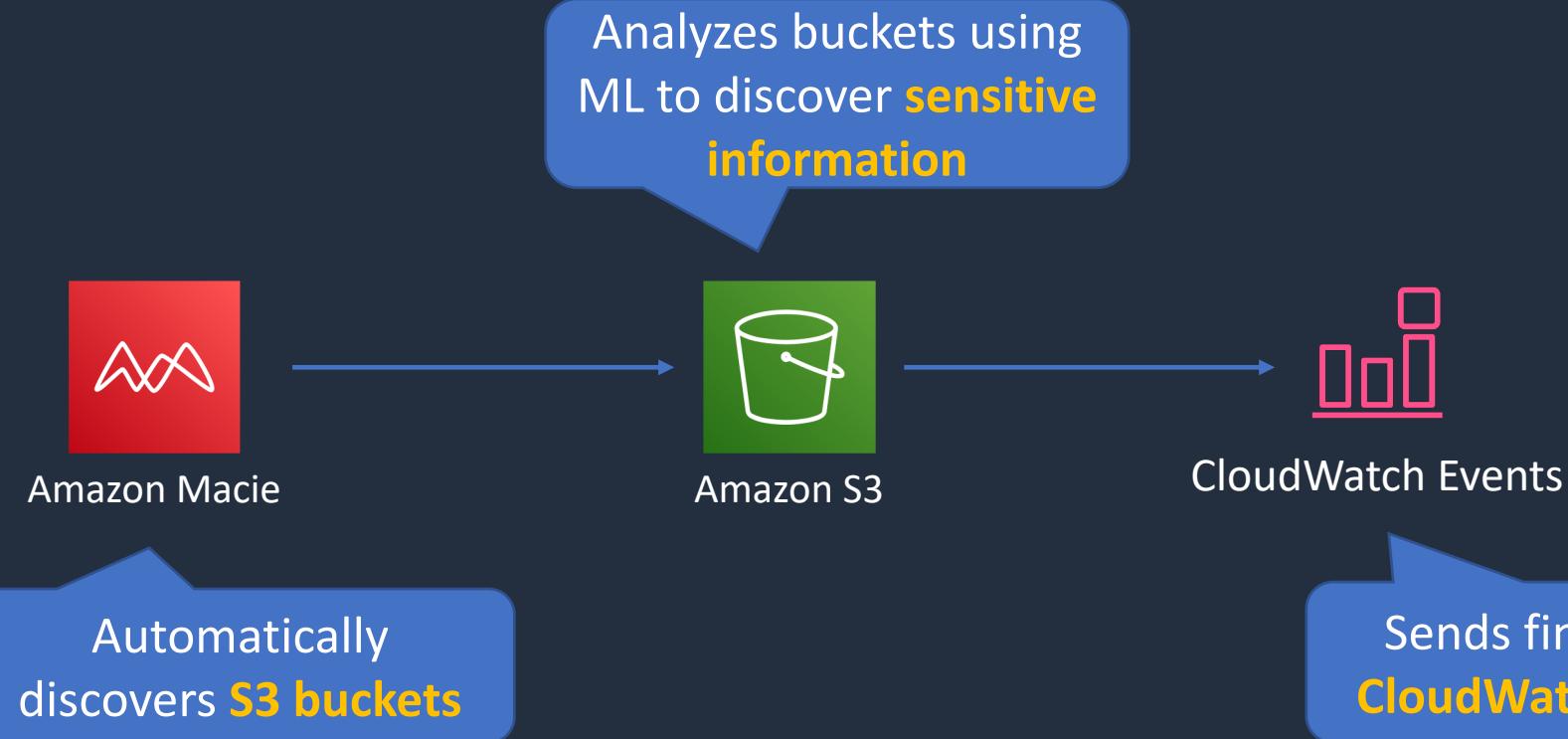
---

---

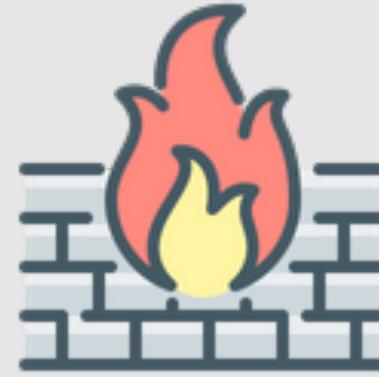
- Macie is a fully managed data security and data privacy service
- Uses machine learning and pattern matching to discover, monitor, and help you protect your sensitive data on Amazon S3
- Macie enables security compliance and preventive security
- Can Identify a variety of data types, including PII, Protected Health Information (PHI), regulatory documents, API keys, and secret keys



# Amazon Macie



# Firewalls and DDoS Protection





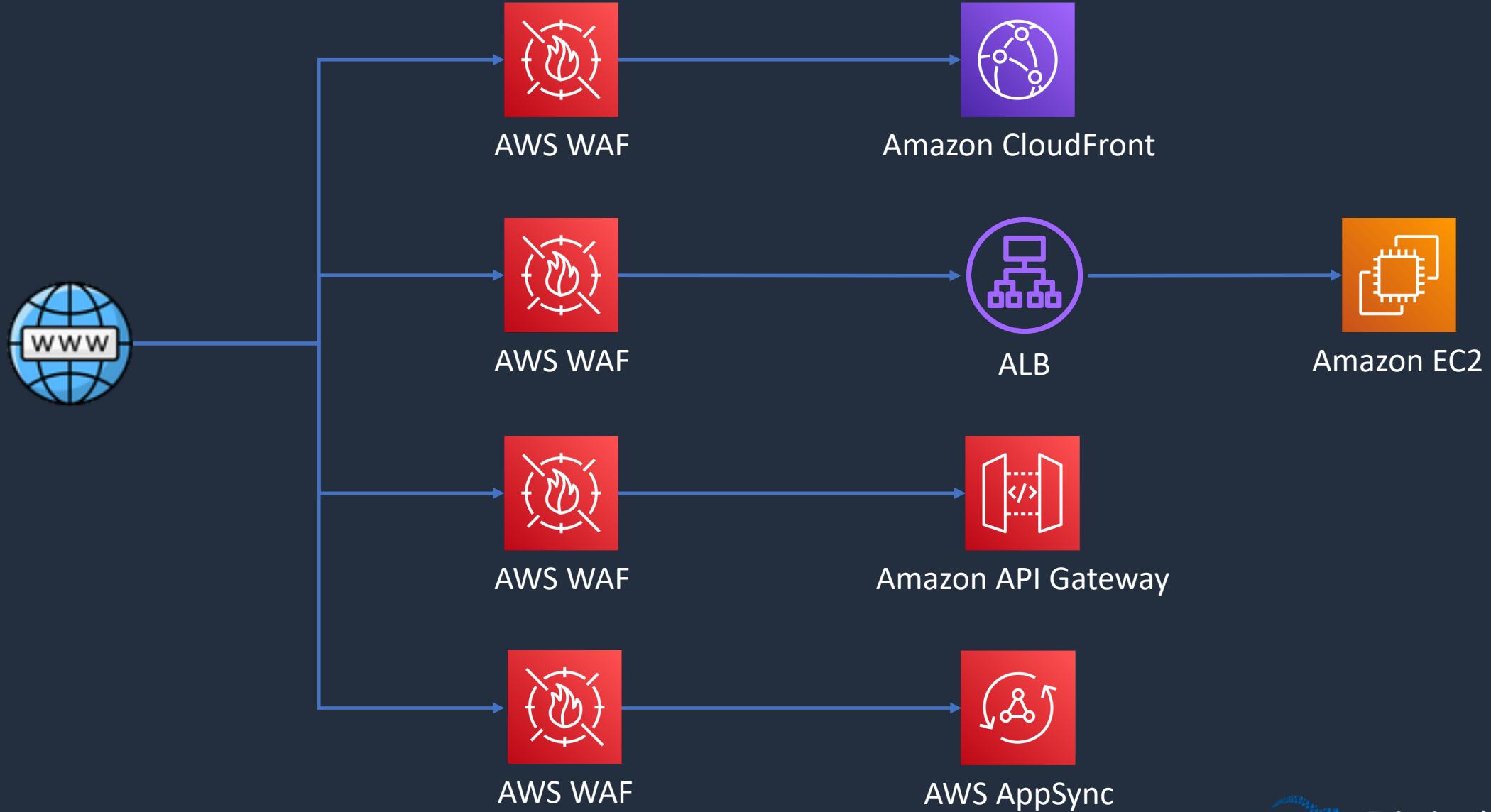
# AWS Web Application Firewall (WAF)

---

- AWS WAF is a **web application firewall**
- WAF lets you create rules to filter web traffic based on conditions that include IP addresses, HTTP headers and body, or custom URIs
- WAF makes it easy to create rules that block common web exploits like **SQL injection** and **cross site scripting**
- The rules are known as Web ACLs



# AWS Web Application Firewall (WAF)





# AWS Shield

- AWS Shield is a managed **Distributed Denial of Service** (DDoS) protection service
- Safeguards web application running on AWS with always-on detection and automatic inline mitigations
- Helps to minimize application downtime and latency
- Two tiers –
  - **Standard** – no cost
  - **Advanced** - \$3k USD per month and 1 year commitment
- Integrated with Amazon CloudFront (standard included by default)

# Compliance Services





# AWS Artifact

- AWS Artifact provides on-demand access to AWS' security and compliance reports and select online agreements
- Reports available in AWS Artifact include:
  - Service Organization Control (SOC) reports
  - Payment Card Industry (PCI) reports
- Provides certifications from accreditation bodies across geographies and compliance verticals that validate the implementation and operating effectiveness of **AWS security controls**
- Agreements available in AWS Artifact include the Business Associate Addendum (BAA) and the Nondisclosure Agreement (NDA)

# Security Management and Support





# AWS Security Hub

- Provides a comprehensive view of security alerts and security posture **across AWS accounts**
- Aggregates, organizes, and prioritizes security alerts, or findings, from multiple AWS services
- Continuously monitors your environment using automated security checks
- Configure security standards to validate against
  - AWS Foundational Security Best Practices v1.0.0
  - CIS AWS Foundations Benchmark v1.2.0
  - PCI DSS v3.2.1



# AWS Security Bulletins

- Security and privacy events affecting AWS services are published (also has an RSS feed)

▼ Content Type

Important  
 Informational

▼ Year

2021  
 2020  
 2019  
 2018  
 2017  
 2016  
 2015  
 2014

<a href="#">Sudo Security Issue (CVE-2021-3156)</a> AWS-2021-001, 01/27/2021
<a href="#">Xen Security Advisory (XSA-286)</a> AWS-2020-005, 10/23/2020
<a href="#">Container Networking Security Issue (CVE-2020-8558)</a> AWS-2020-002v2, 07/09/2020
<a href="#">Minimum Version of TLS 1.2 Required for FIPS Endpoints by March 31, 2021</a> AWS-2020-001, 03/31/2020
<a href="#">Kubernetes Security Issue (CVE-2019-11249)</a> AWS-2019-007, 08/15/2019
<a href="#">Kubernetes Security Issue (CVE-2019-11246)</a> AWS-2019-006, 07/02/2019
<a href="#">Linux Kernel TCP SACK Denial of Service Issues</a> AWS-2019-005, 06/17/2019



# AWS Trust & Safety Team

- Contact the **AWS Trust & Safety** team if AWS resources are being used for:
  - Spam
  - Port scanning
  - Denial-of-service attacks
  - Intrusion attempts
  - Hosting of objectionable or copyrighted content
  - Distributing malware
- Email address is: [abuse@amazonaws.com](mailto:abuse@amazonaws.com)

# Penetration testing





# Penetration Testing

---

---

- Penetration testing is the practice of testing one's own application's security for vulnerabilities by simulating an attack
- AWS allows penetration testing without prior approval for 8 AWS services



# Penetration Testing

---

---

## Permitted services

- Amazon EC2 instances, NAT Gateways, and Elastic Load Balancers
- Amazon RDS
- Amazon CloudFront
- Amazon Aurora
- Amazon API Gateways
- AWS Lambda and Lambda Edge functions
- Amazon Lightsail resources
- Amazon Elastic Beanstalk environments

## Prohibited Activities

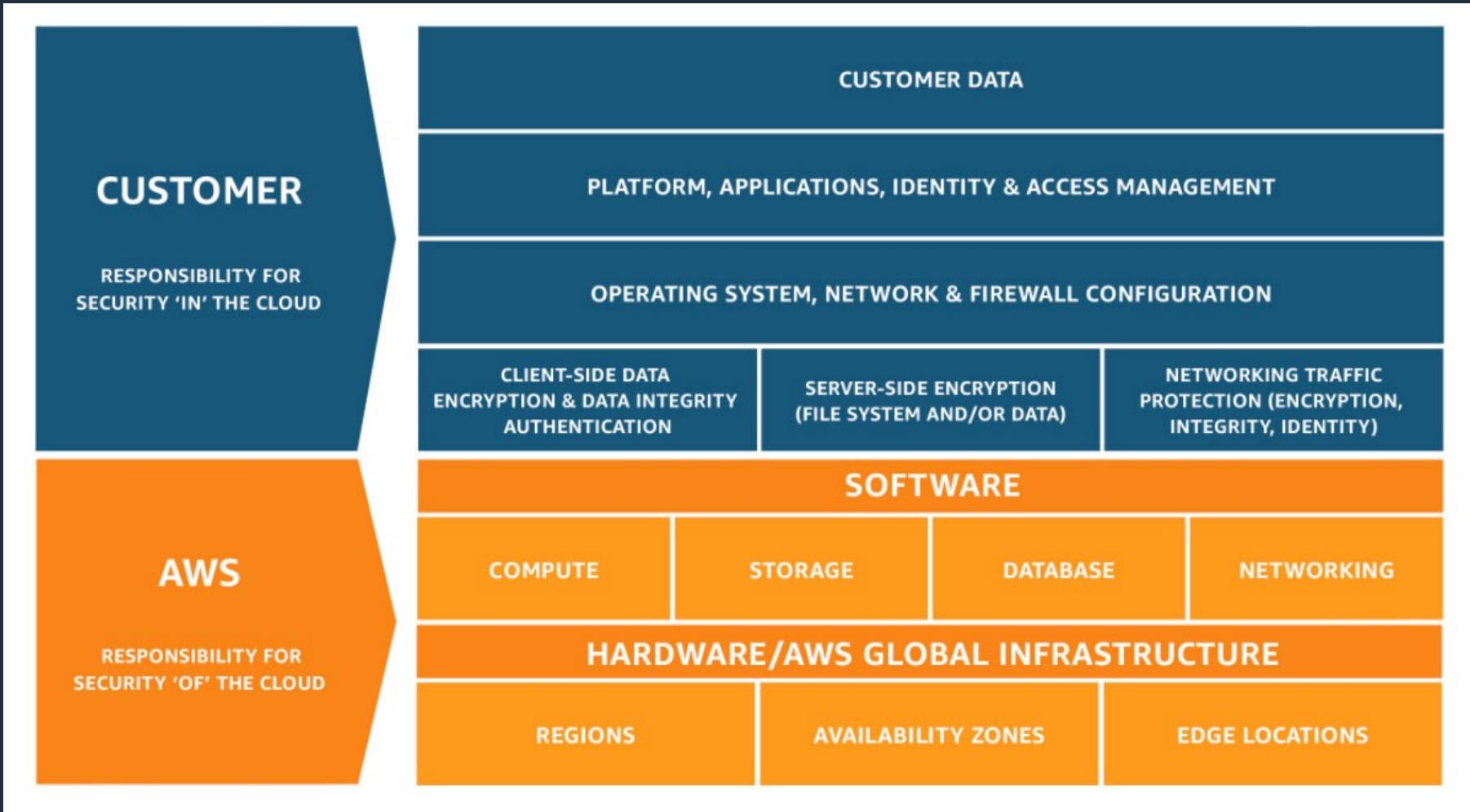
- DNS zone walking via Amazon Route 53 Hosted Zones
- Denial of Service (DoS), Distributed Denial of Service (DDoS), Simulated DoS, Simulated DDoS Port flooding
- Protocol flooding
- Request flooding (login request flooding, API request flooding)

# Shared Responsibility Model Review





# The AWS Shared Responsibility Model





# The AWS Shared Responsibility Model

## CUSTOMER RESPONSIBILITY



Bucket with objects



Role



Multi-Factor Authentication



Security Group



Patch management



Staff training



Data encryption



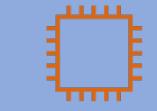
IAM User



Network ACL



SSL encryption



EC2 Instance



Auto Scaling



Elastic load balancer

## AWS RESPONSIBILITY



Data center security



Network router



Network switch



Server



Storage



Database Server



Disk drive

# SECTION 14

## Architecting for the Cloud

# AWS Well-Architected





# AWS Well-Architected

---

---

- AWS Well-Architected helps cloud architects build secure, high-performing, resilient, and efficient infrastructure for their applications and workloads
- Based on 5 pillars:
  - **Operational Excellence**
  - **Security**
  - **Reliability**
  - **Performance Efficiency**
  - **Cost Optimization**



# AWS Well-Architected

---

---

Consists of:

- AWS Well-Architected Guidance
- AWS Well-Architected Tool
- AWS Well-Architected Lenses
- AWS Architecture Center

# AWS Well-Architected Framework





# AWS Well-Architected Framework

---

---

- Helps you understand the pros and cons of decisions you make while building systems on AWS
- Based on 5 pillars:

## **Operational Excellence Pillar**

- Support development and run workloads effectively
- Gain insight into workload operations
- Continuously improve processes and procedures to deliver business value



# AWS Well-Architected

---

---

- Best practices for operational excellence:
  - Perform operations as code
  - Make frequent, small, reversible changes
  - Refine operations procedures frequently
  - Anticipate failure
  - Learn from all operational failures



## Security Pillar

- Protect data, systems, and assets to take advantage of cloud technologies to improve your security
- Best practices for security:
  - Implement a strong identity foundation
  - Enable traceability
  - Apply security at all layers
  - Automate security best practices
  - Protect data in transit and at rest
  - Keep people away from data
  - Prepare for security events



## Reliability Pillar

- Ensuring a workload can perform its intended function correctly and consistently when it's expected to
- This includes the ability to operate and test the workload through its total lifecycle
- Best practices for reliability:
  - Automatically recover from failure
  - Test recovery procedures
  - Scale horizontally to increase aggregate workload availability
  - Stop guessing capacity
  - Manage change in automation



## Performance Efficiency Pillar

- The ability to use computing resources efficiently to meet system requirements, and to maintain that efficiency as demand changes and technologies evolve
- Best practices for performance efficiency:
  - Democratize advanced technologies
  - Go global in minutes
  - Use serverless architectures
  - Experiment more often
  - Consider mechanical sympathy



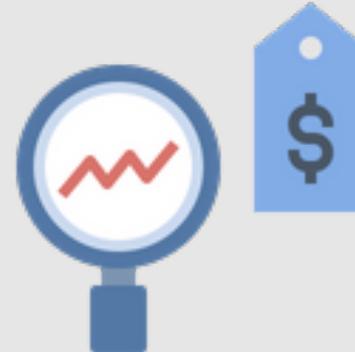
## Cost Optimization Pillar

- The ability to run systems to deliver business value at the lowest price point
- Best practices for cost optimization:
  - Implement Cloud Financial Management
  - Adopt a consumption model
  - Measure overall efficiency
  - Stop spending money on undifferentiated heavy lifting
  - Analyze and attribute expenditure

# SECTION 15

Accounts, Billing and Support

# AWS Pricing Fundamentals





# AWS Pricing Fundamentals

## Compute



**Amount** of resources such as CPU and RAM and **duration**

## Storage



Quantity of **data stored**

## Outbound Data Transfer



Quantity of data that is **transferred out** from all services



# AWS Pricing Fundamentals

---

---

## Pay-as-you-go

- Easily adapt to changing business needs
- Improved responsiveness to change
- Adapt based on needs, not forecasts
- Reduce risk over overpositioning of missing capacity



# AWS Pricing Fundamentals

---

## Save when you reserve

- Invest in reserved capacity (e.g. RDS and EC2)
- Save up to 75% compared to on-demand (pay-as-you-go)
- The more you pay upfront the greater the discount



# AWS Pricing Fundamentals

---

## Pay less by using more

- Pay less using volume-based discounts
- Tiered pricing means the more you use the lower the unit pricing

# Amazon EC2 Pricing Options





# Amazon EC2 Pricing Options

## On-Demand

Standard rate - no discount; no commitments; dev/test, short-term, or unpredictable workloads

## Spot Instances

Bid for unused capacity; up to 90% discount; can be terminated at any time; workloads with flexible start and end times

## Dedicated Hosts

Physical server dedicated for your use; Socket/core visibility, host affinity; pay per host; workloads with server-bound software licenses

## Reserved

1 or 3-year commitment; up to 75% discount; steady-state, predictable workloads and reserved capacity

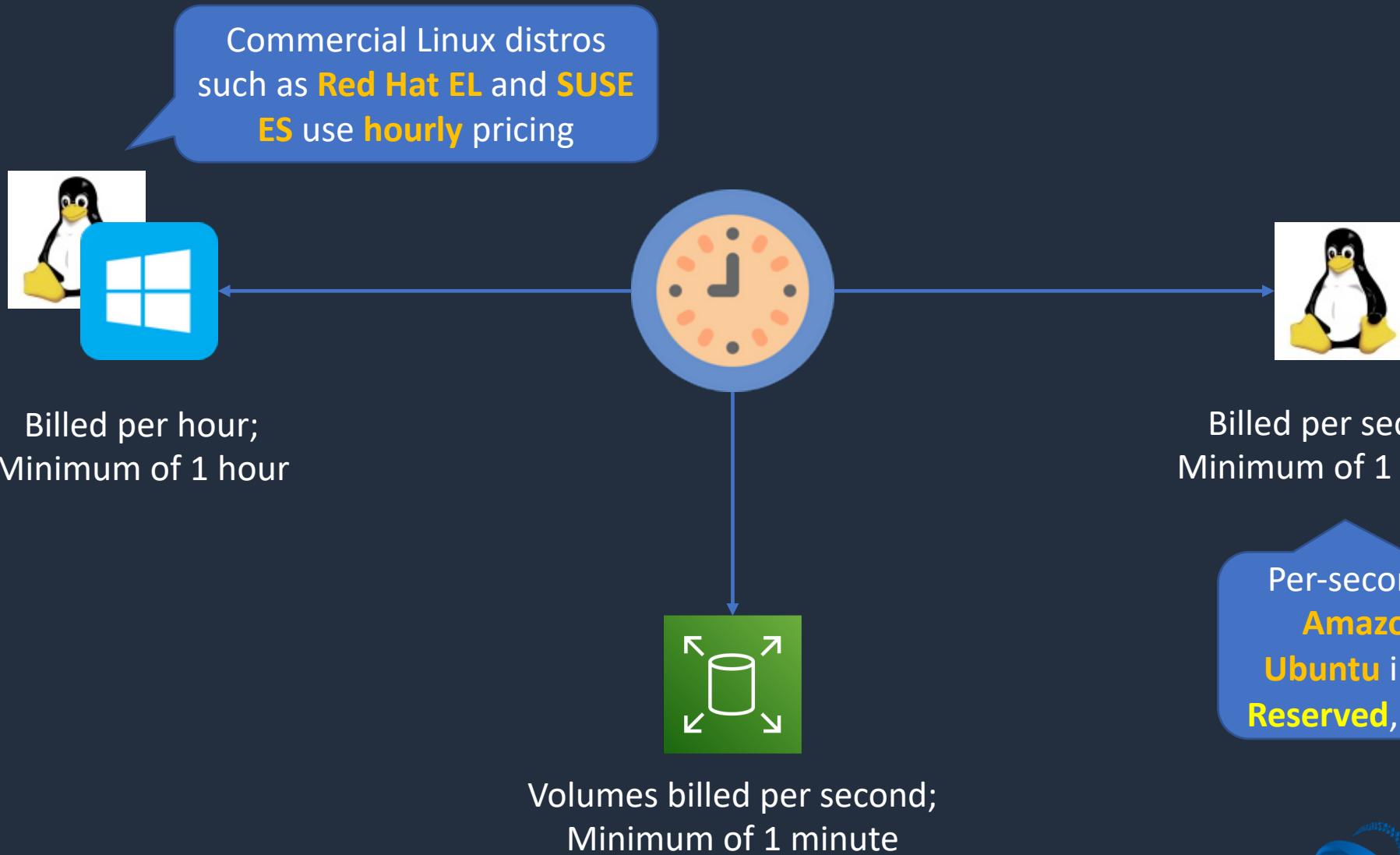
## Dedicated Instances

Physical isolation at the host hardware level from instances belonging to other customers; pay per instance

## Savings Plans

Commitment to a consistent amount of usage (EC2 + Fargate + Lambda); Pay by \$/hour; 1 or 3-year commitment

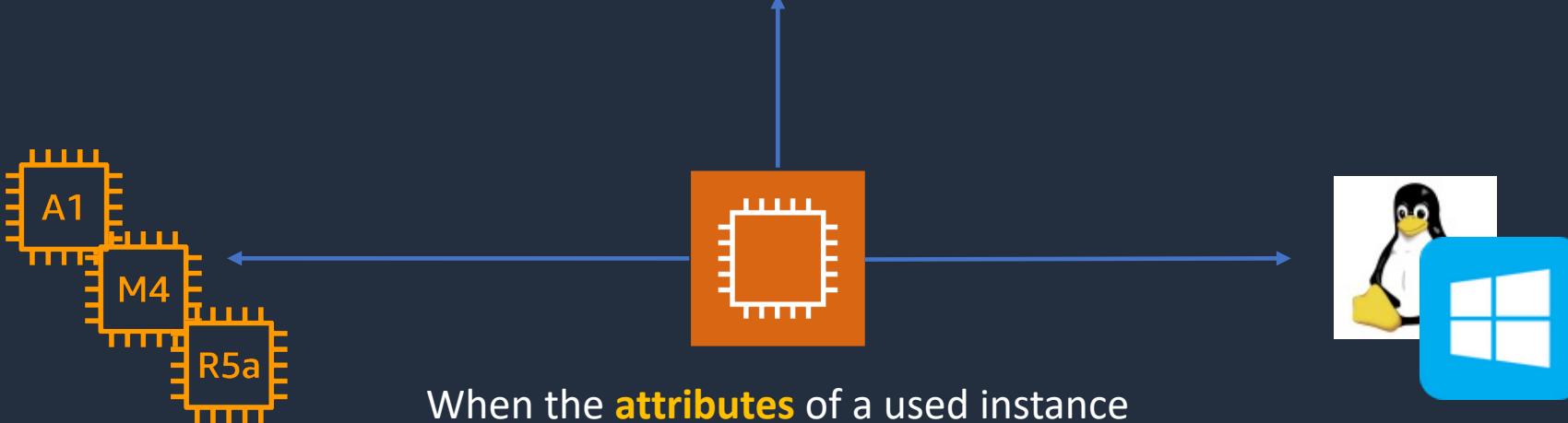
# \$ Amazon EC2 Billing





# Amazon EC2 Reserved Instances (RIs)

Tenancy: **Default** or **Dedicated**



When the **attributes** of a used instance match the **attributes** of an RI the discount is applied

Reserves capacity in specified AZ

Availability Zone

Region

Does **not** reserve capacity; discount applies to all AZs



# Amazon EC2 Reserved Instances (RIs)



Scheduled RI

- Match capacity reservation to **recurring schedule**
- Minimum **1200 hours** per year
- Example: Reporting app that runs 6 hours a day 4 days a week = 1248 hours per year

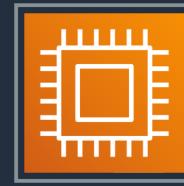


**Important**  
We do not have any capacity for purchasing Scheduled Reserved Instances or any plans to make it available in the future. To reserve capacity, use [On-Demand Capacity Reservations](#) instead. For discounted rates, use [Savings Plans](#).

This message started showing recently but exam may **not** reflect this yet

# \$ AWS Savings Plans

---



## Compute Savings Plan



1 or 3-year; hourly commitment to usage of **Fargate**, **Lambda**, and **EC2**; Any Region, family, size, tenancy, and OS



## EC2 Savings Plan



1 or 3-year; hourly commitment to usage of **EC2** within a **selected Region** and **Instance Family**; Any size, tenancy and OS



# Amazon EC2 Spot Instances



**Bid** for unused capacity at up to 90% discount



**2-minute warning** if AWS need to reclaim capacity – available via **instance metadata** and **CloudWatch Events**



**Spot Instance:** One or more EC2 instances



**Spot Fleet:** launches and maintains the number of Spot / On-Demand instances to meet specified target capacity



**EC2 Fleet:** launches and maintains specified number of Spot / On-Demand / Reserved instances in a **single API call**

Can define separate OD/Spot **capacity targets, bids, instance types, and AZs**



# Spot Block



Requirement:  
Uninterrupted for  
1-6 hours

Pricing is **30% - 45%** less  
than On-Demand



Solution: **Spot Block**

```
$ aws ec2 request-spot-instances \
--block-duration-minutes 360 \
--instance-count 5 \
--spot-price "0.25" ...
```



# Dedicated Instances and Dedicated Hosts

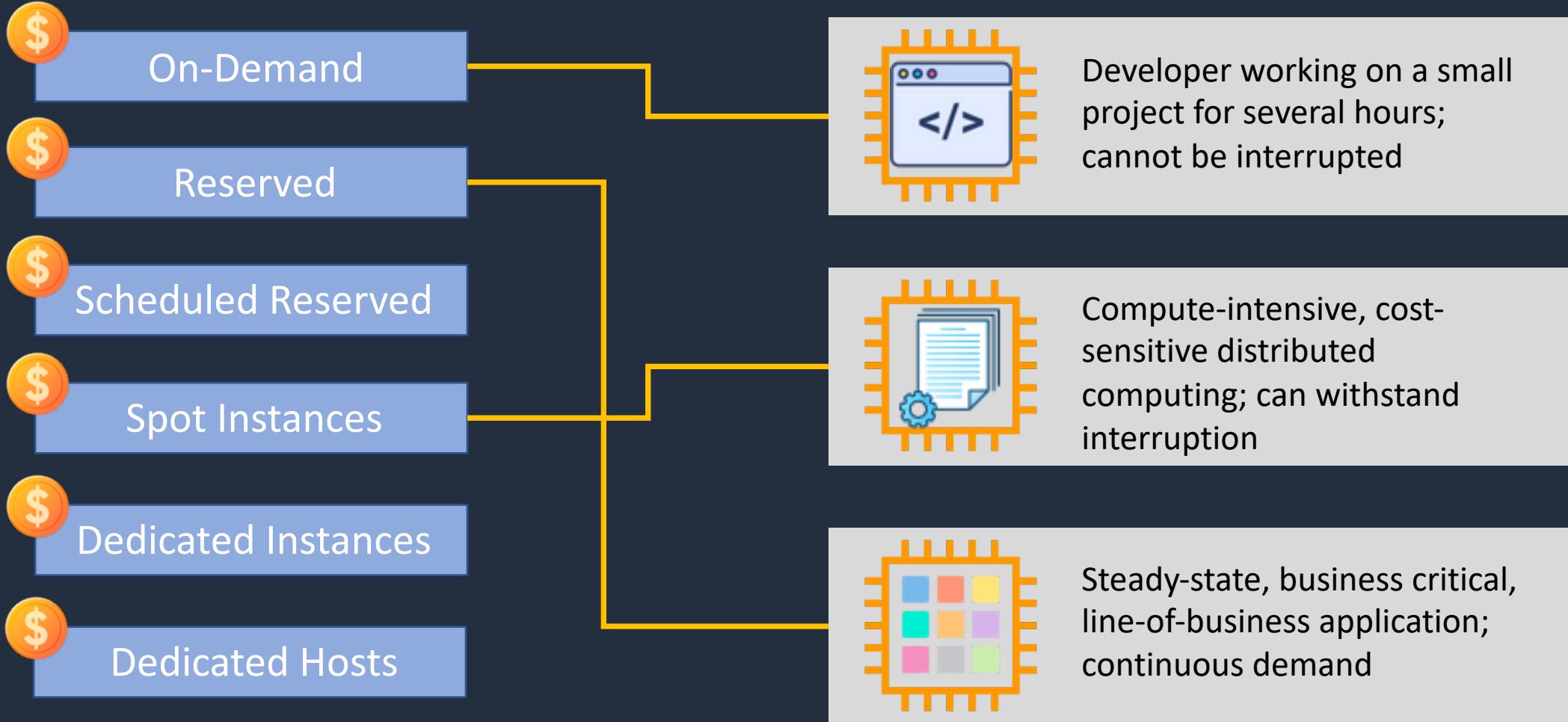
Characteristic	Dedicated Instances	Dedicated Hosts
Enables the use of dedicated physical servers	X	X
Per instance billing (subject to a \$2 per region fee)	X	
Per host billing		X
Visibility of sockets, cores, host ID		X
Affinity between a host and instance		X
Targeted instance placement		X
Automatic instance placement	X	X
Add capacity using an allocation request		X

# Amazon EC2 Pricing Use Cases



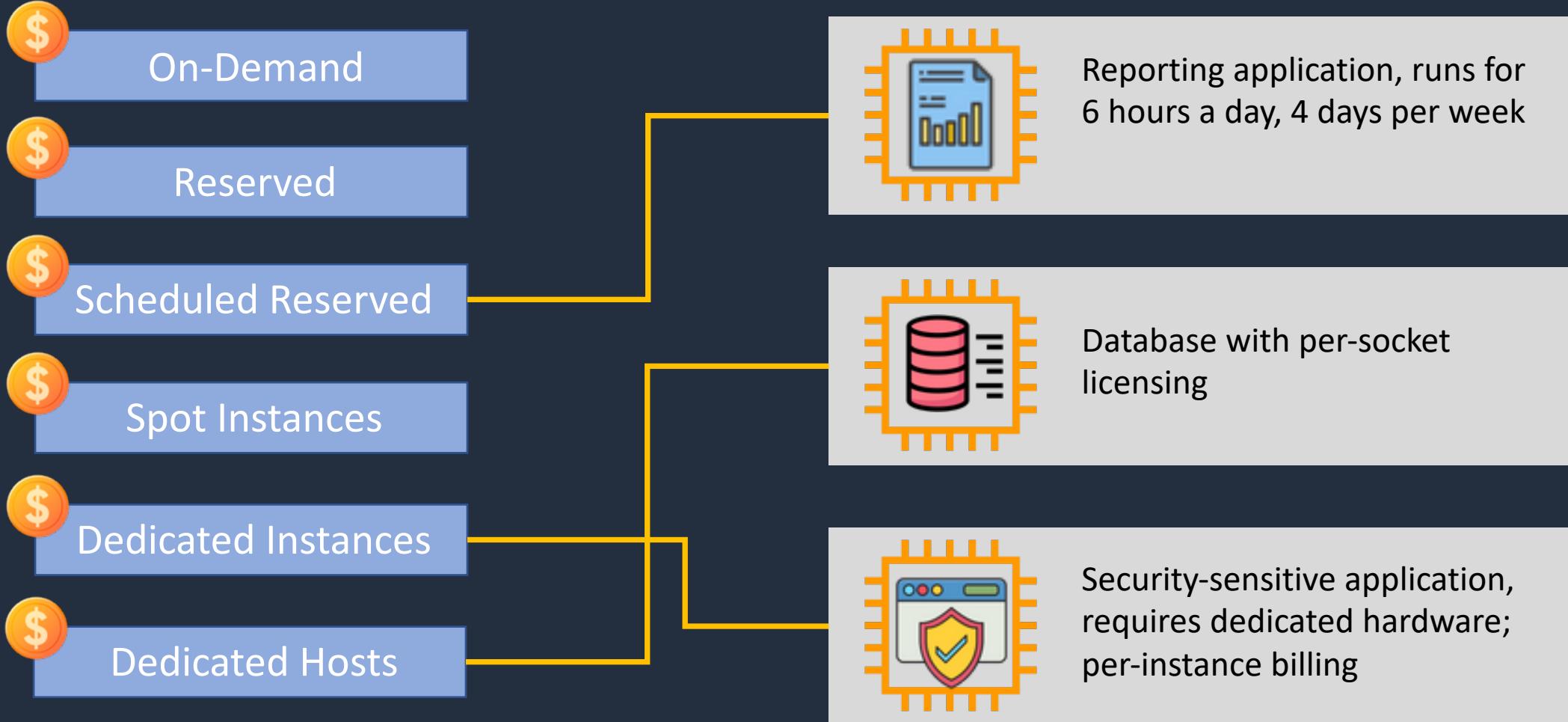


# Amazon EC2 Pricing Use Cases





# Amazon EC2 Pricing Use Cases



# Pricing for other AWS Services





# Amazon S3 Pricing

---

- **Storage class** – e.g. Standard or IA
- **Storage quantity** – data volume stored in your buckets on a per GB basis
- **Number of requests** – the number and type of requests, e.g. GET, PUT, POST, LIST, COPY
- **Lifecycle transitions requests** – moving data between storage classes
- **Data transfer** – data transferred out of an S3 region is charged



# Amazon S3 Glacier Pricing

- Three options for access to archives, listed in the table below:

	Expedited	Standard	Bulk
<b>Data access time</b>	1-5 minutes	3-5 hours	5-12 hours
<b>Data retrievals</b>	\$0.03 per GB	\$0.01 per GB	\$0.0025 per GB
<b>Retrieval requests</b>	On-Demand: \$0.01 per request  Provisioned: \$100 per Provisioned Capacity Unit	\$0.050 per 1,000 requests	\$0.025 per 1,000 requests



# Amazon EBS Pricing

---

- **Volumes** – volume storage for all EBS volumes type is charged by the amount of GB provisioned per month
- **Snapshots** – based on the amount of space consumed by snapshots in S3. Copying snapshots is charged on the amount of data copied across regions
- **Data transfer** – inbound data transfer is free, outbound data transfer charges are tiered



# Amazon RDS Pricing

---

- **Clock hours of server uptime** – amount of time the DB instance is running
- **Database characteristics** – e.g. database engine, size and memory class
- **Database purchase type** – e.g. On-Demand, Reserved.
- **Number of database instances**
- **Provisioned storage** – backup is included up to 100% of the size of the DB
- **Additional storage** – the amount of storage in addition to the provisioned storage is charged per GB per month



# Amazon RDS Pricing

---

- **Requests** – the number of input and output requests to the DB
- **Deployment type** – single AZ or multi-AZ
- **Reserved Instances** – RDS RIs can be purchased with No Upfront, Partial Upfront, or All Upfront terms. Available for Aurora, MySQL, MariaDB, Oracle and SQL Server



# Amazon DynamoDB Pricing

---

- Charged for reading, writing, and storing data
- **On-demand capacity mode**
  - Charged for reads and writes
  - No need to specify how much capacity is required
  - Good for unpredictable workloads
- **Provisioned capacity mode**
  - Specify number of reads and writes per second
  - Can use Auto Scaling
  - Good for predictable workloads
  - Consistent traffic or gradual changes



# Amazon CloudFront Pricing

---

- **Traffic distribution** – data transfer and request pricing, varies across regions, and is based on the edge location from which the content is served
- **Requests** – the number and type of requests (HTTP or HTTPS) and the geographic region in which they are made
- **Data transfer out** – quantity of data transferred out of CloudFront edge locations
- There are additional chargeable items such as invalidation requests, field-level encryption requests, and custom SSL certificates



# AWS Lambda Pricing

---

---

- **Number of requests**
- **Duration of request** – rounded up to the nearest millisecond
- Price is dependent on the amount of memory allocated to the function
- Free tier includes 1M free requests per month and 400,000 GB-seconds of compute time

# AWS Pricing Calculator



# AWS Support Plans





# AWS Support Plans

	Developer	Business	Enterprise
AWS Trusted Advisor Best Practice Checks	7 Core checks	Full set of checks	Full set of checks
Enhanced Technical Support	<ul style="list-style-type: none"><li>Business hours email access to Cloud Support Associates</li><li>Unlimited cases / 1 primary contact</li></ul>	<ul style="list-style-type: none"><li>24x7 phone, email, and chat access to Cloud Support Engineers</li><li>Unlimited cases / unlimited contacts (IAM supported)</li></ul>	<ul style="list-style-type: none"><li>24x7 phone, email, and chat access to Cloud Support Engineers</li><li>Unlimited cases / unlimited contacts (IAM supported)</li></ul>
Case Severity / Response Times*	<ul style="list-style-type: none"><li>General guidance: &lt; 24 hours**</li><li>System impaired: &lt; 12 hours**</li></ul>	<ul style="list-style-type: none"><li>General guidance: &lt; 24 hours</li><li>System impaired: &lt; 12 hours</li><li>Production system impaired: &lt; 4 hours</li><li>Production system down: &lt; 1 hour</li></ul>	<ul style="list-style-type: none"><li>General guidance: &lt; 24 hours</li><li>System impaired: &lt; 12 hours</li><li>Production system impaired: &lt; 4 hours</li><li>Production system down: &lt; 1 hour</li><li>Business-critical system down: &lt; 15 minutes</li></ul>



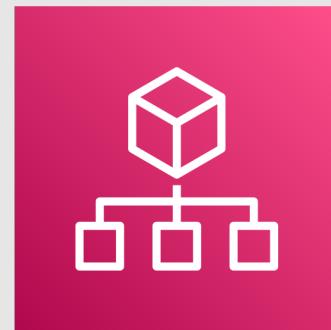
# AWS Support Plans

---

---

	Developer	Business	Enterprise
Architectural Guidance	General	Contextual to your use-cases	Consultative review and guidance based on your applications
Programmatic Case Management		AWS Support API	AWS Support API
Proactive Programs and Services		Access to Infrastructure Event Management for additional fee	<ul style="list-style-type: none"><li>• Infrastructure Event Management</li><li>• Well-Architected Reviews</li><li>• Access to proactive reviews, workshops, and deep dives</li></ul>
Technical Account Management			Designated Technical Account Manager (TAM) to proactively monitor your environment and assist with optimization and coordinate access to programs and AWS experts
Account Assistance			Concierge Support Team

# Consolidated Billing





# AWS Organizations

---

- Consolidated billing has the following benefits:
- One bill – You get one bill for multiple accounts
  - **Easy tracking** – You can track the charges across multiple accounts and download the combined cost and usage data
  - **Combined usage** – You can combine the usage across all accounts in the organization to share the volume pricing discounts and Reserved Instance discounts
  - **No extra fee** – Consolidated billing is offered at no additional cost



# AWS Organizations

Tier Description	Price Per GB	Price Per TB
First 1 TB/month	\$0.10	\$100.00
Next 49 TB/month	\$0.08	\$80.00
Next 450 TB/month	\$0.06	\$60.00

## Usage within Organization:

Account A (master) usage: 2 TB

Account B usage: 80 TB

Account C usage: 120 TB

Total: 202 TB

## Calculation:

First 1 TB = \$100.00

Next 49 TB = \$3,920.00

Next 157 TB = \$9,120.00

Total cost = \$13,140.00

# AWS Budgets





# AWS Budgets

All budgets (1)	Cost budgets (1)	Usage budgets (0)	Reservation budgets (0)	Savings Plans budgets (0)		
Budget name	Type	Current	Budgeted	Forecasted	Current vs. budgeted	Forecasted vs. budgeted
MyBudget	Cost	\$13.20	\$100.00	\$129.47	<div style="width: 13.2%; background-color: #0070C0;"></div> 13.2%	<div style="width: 129.47%; background-color: #E74C3C;"></div> 129.47%

**Set Custom Budgets** - set custom usage and reservation budgets

**Configure Alerts** – receive alerts when you exceed or are forecast to exceed your alert thresholds

**Integrated with other AWS services** – Includes Cost Explorer Chatbot, and Service Catalog

# AWS Cost Allocation Tags



# AWS Cost Management Tools





# AWS Cost Explorer

- The **AWS Cost Explorer** is a free tool that allows you to view charts of your costs
- You can view cost data for the past 13 months and forecast how much you are likely to spend over the next three months
- Cost Explorer can be used to discover patterns in how much you spend on AWS resources over time and to identify cost problem areas
- Cost Explorer can help you to identify service usage statistics such as:
  - Which services you use the most
  - View metrics for which AZ has the most traffic
  - Which linked account is used the most



# AWS Cost & Usage Report

---

---

- Publish AWS billing reports to an Amazon S3 bucket
- Reports break down costs by:
  - Hour, day, month, product, product resource, tags
- Can update the report up to three times a day
- Create, retrieve, and delete your reports using the AWS CUR API Reference



# AWS Price List API

---

---

- Query the prices of AWS services
- **Price List Service API** (AKA the Query API) – query with JSON
- **AWS Price List API** (AKA the Bulk API) – query with HTML
- Alerts via Amazon SNS when prices change

# AWS Cost Management Tools



# SECTION 16

Migration, Machine Learning and More

# AWS Migration and Transfer Services





# AWS Migration Tools



Region



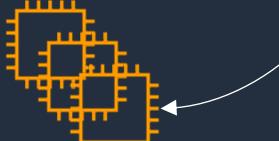
AWS Migration Hub



Amazon S3



Amazon RDS



EC2 Instances



EFS File system



AWS Server Migration Service



AWS Database Migration Service



AWS DataSync



VPN or Direct  
Connect connection

Corporate data center



Servers



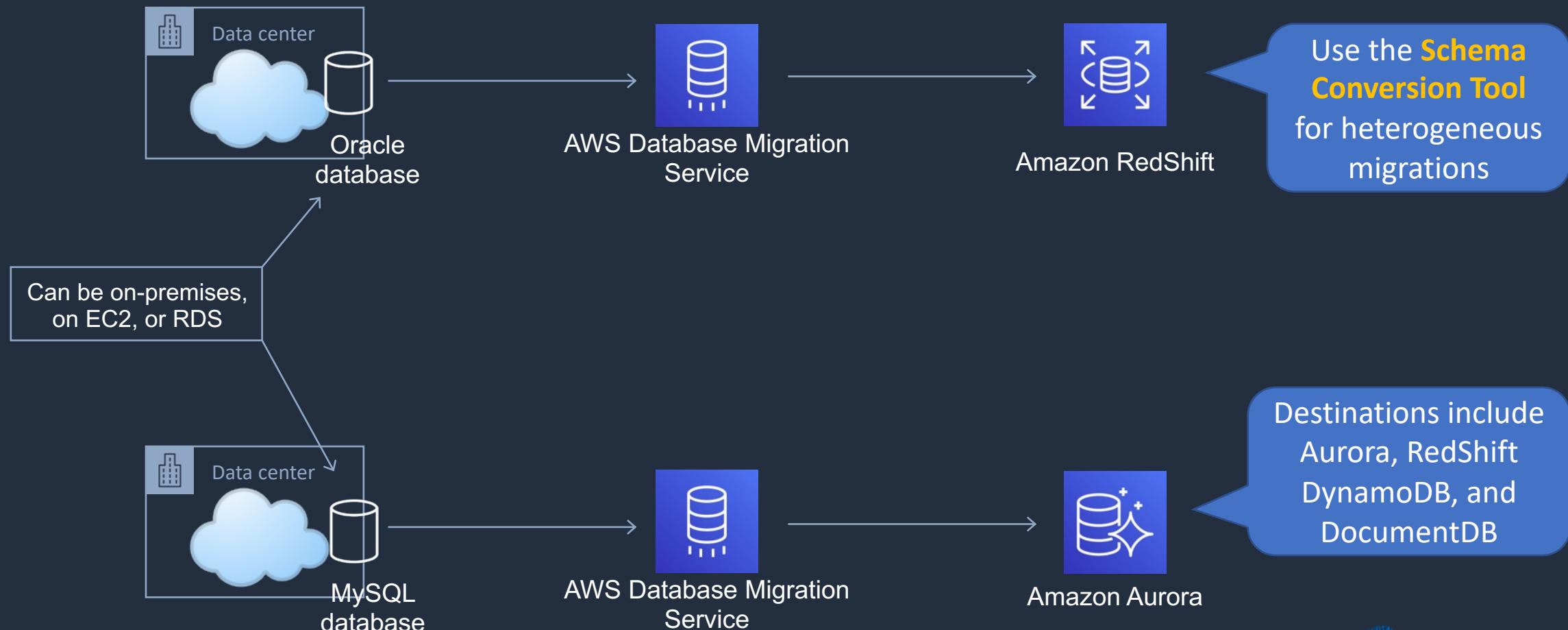
Database



NAS / File  
Server



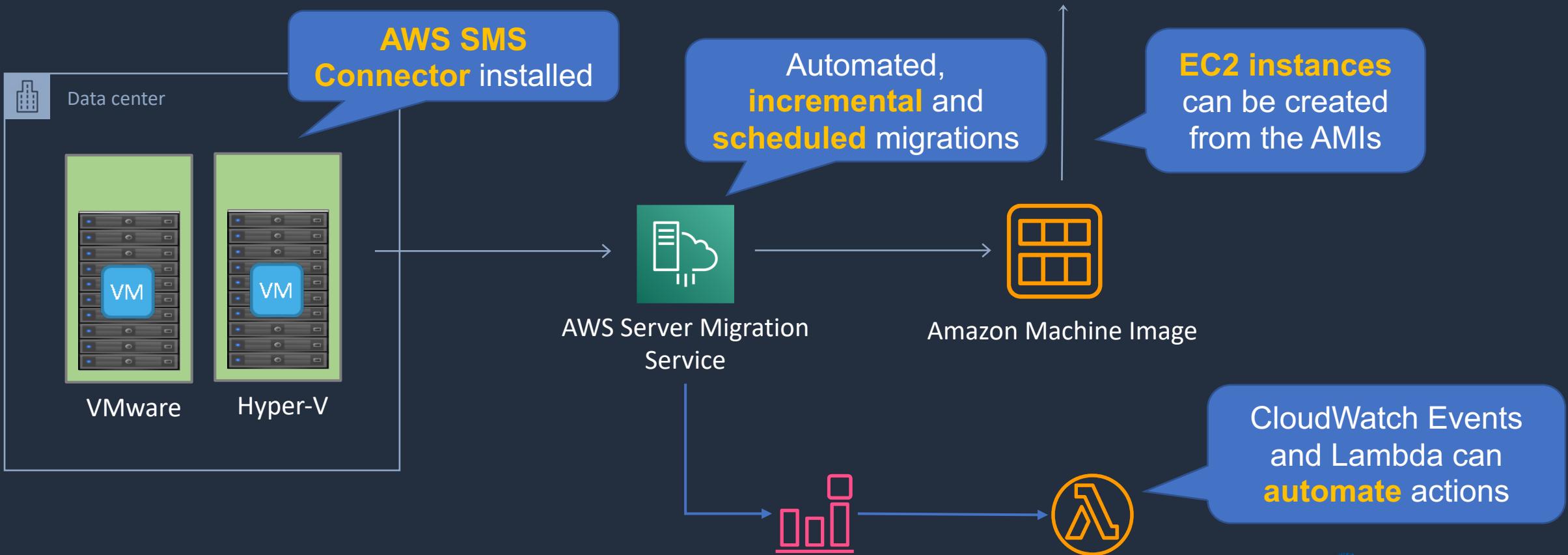
# AWS Database Migration Service (DMS)





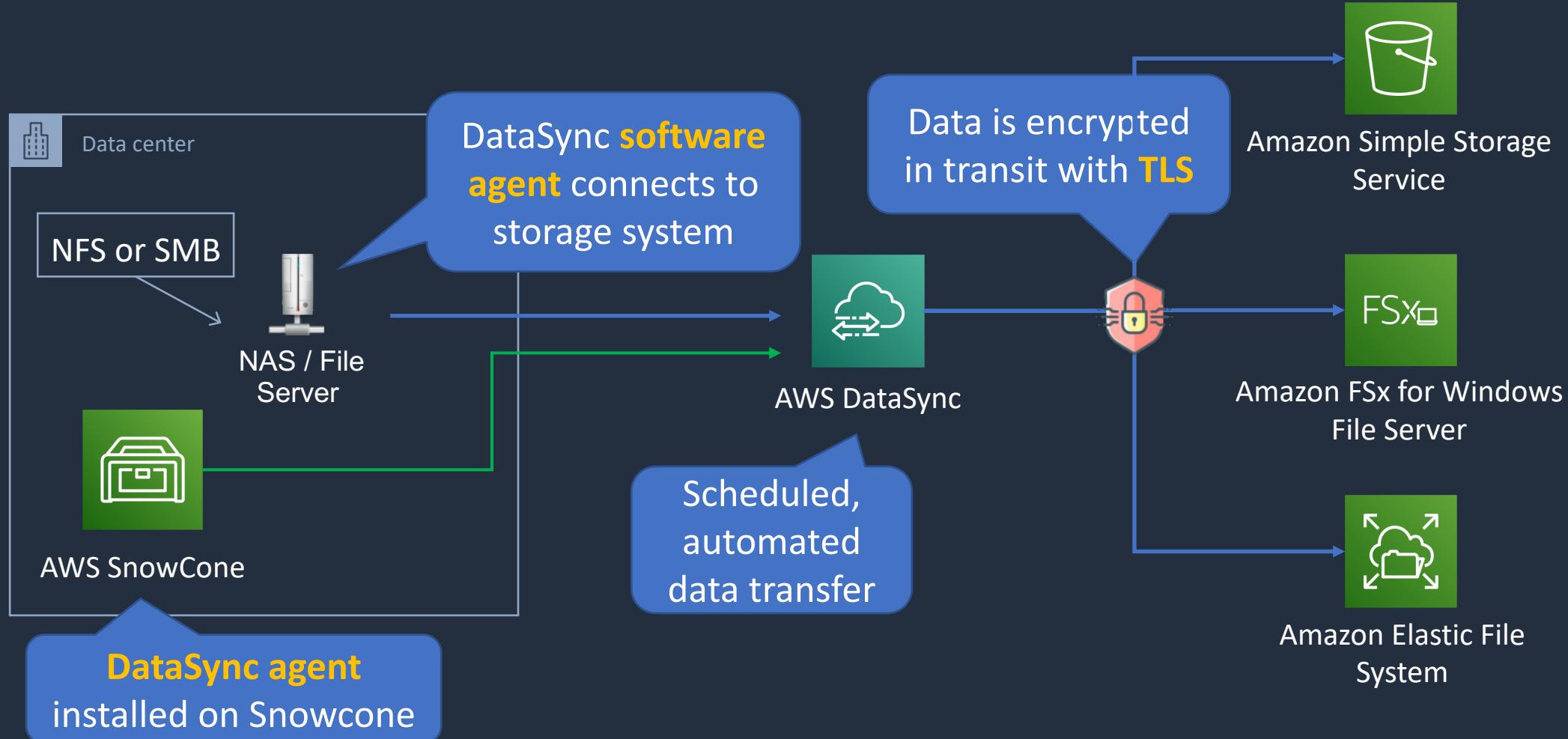
# AWS Server Migration Service (SMS)

- AWS SMS migrates VMware vSphere, Microsoft Hyper-V/SCVMM, and Azure virtual machines to Amazon EC2





# AWS DataSync





# AWS Snowball Family

- **AWS Snowball and Snowmobile** are used for migrating large volumes of data to AWS
- **Snowball Edge Compute Optimized**
  - Provides block and object storage and optional GPU
  - Use for data collection, machine learning and processing, and storage in environments with intermittent connectivity (edge use cases)
- **Snowball Edge Storage Optimized**
  - Provides block storage and Amazon S3-compatible object storage
  - Use for local storage and large-scale data transfer
- **Snowcone**
  - Small device used for edge computing, storage and data transfer
  - Can transfer data offline or online with AWS DataSync agent





# AWS Snowball Family

- Uses a secure storage device for physical transportation
- Snowball Client is software that is installed on a local computer and is used to identify, compress, encrypt, and transfer data
- Uses 256-bit encryption (managed with the AWS KMS) and tamper-resistant enclosures with TPM
- **Snowball** (80TB) (50TB ) “petabyte scale”
- **Snowball Edge** (100TB) “petabyte scale”
- **Snowmobile** – “exabyte scale” with up to 100PB per Snowmobile



# AWS Machine Learning Services





# AWS Rekognition

## Identify objects



## Perform facial analysis

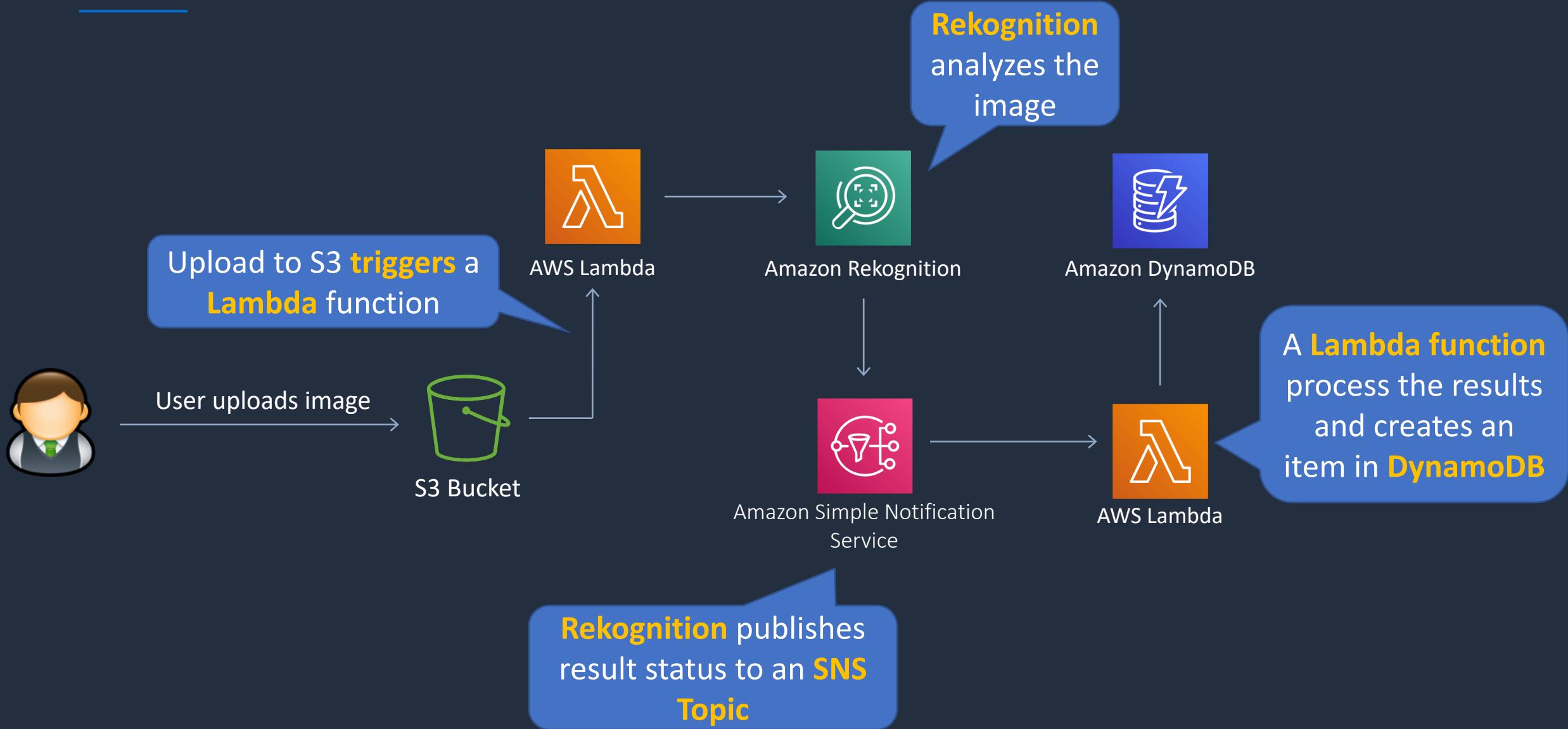


## Celebrity recognition





# AWS Rekognition in Event-Driven Architecture





# AWS Rekognition

---

- Add image and video analysis to your applications
- Identify objects, people, text, scenes, and activities in images and videos
- Processes videos stored in an Amazon S3 bucket
- Publish completion status to Amazon SNS Topic



# Amazon Transcribe

---

---

- Add speech to text capabilities to applications
- Recorded speech can be converted to text before it can be used in applications
- Uses a deep learning process called automatic speech recognition (ASR) to convert speech to text quickly and accurately



# Amazon Translate

---

- Neural machine translation service that delivers fast, high-quality, and affordable language translation
- Uses deep learning models to deliver more accurate and more natural sounding translation
- Localize content such as websites and applications for your diverse users



# Amazon SageMaker

---

- Helps data scientists and developers to prepare, build, train, and deploy high-quality machine learning (ML) models
- ML development activities including:
  - Data preparation
  - Feature engineering
  - Statistical bias detection
  - Auto-ML
  - Training and tuning
  - Hosting
  - Monitoring
  - Workflows



# Amazon Comprehend

---

- Natural-language processing (NLP) service
- Uses machine learning to uncover information in unstructured data
- Can identify critical elements in data, including references to language, people, and places, and the text files can be categorized by relevant topics
- In real time, you can automatically and accurately detect customer sentiment in your content



# Amazon Lex

---

---

- Conversational AI for Chatbots
- Build conversational interfaces into any application using voice and text
- Build bots to increase contact center productivity, automate simple tasks, and drive operational efficiencies across the enterprise

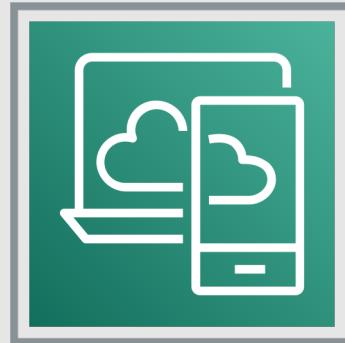


# Amazon Polly

---

- Turns text into lifelike speech
- Create applications that talk, and build entirely new categories of speech-enabled products
- Text-to-Speech (TTS) service uses advanced deep learning technologies to synthesize natural sounding human speech

# End User Computing





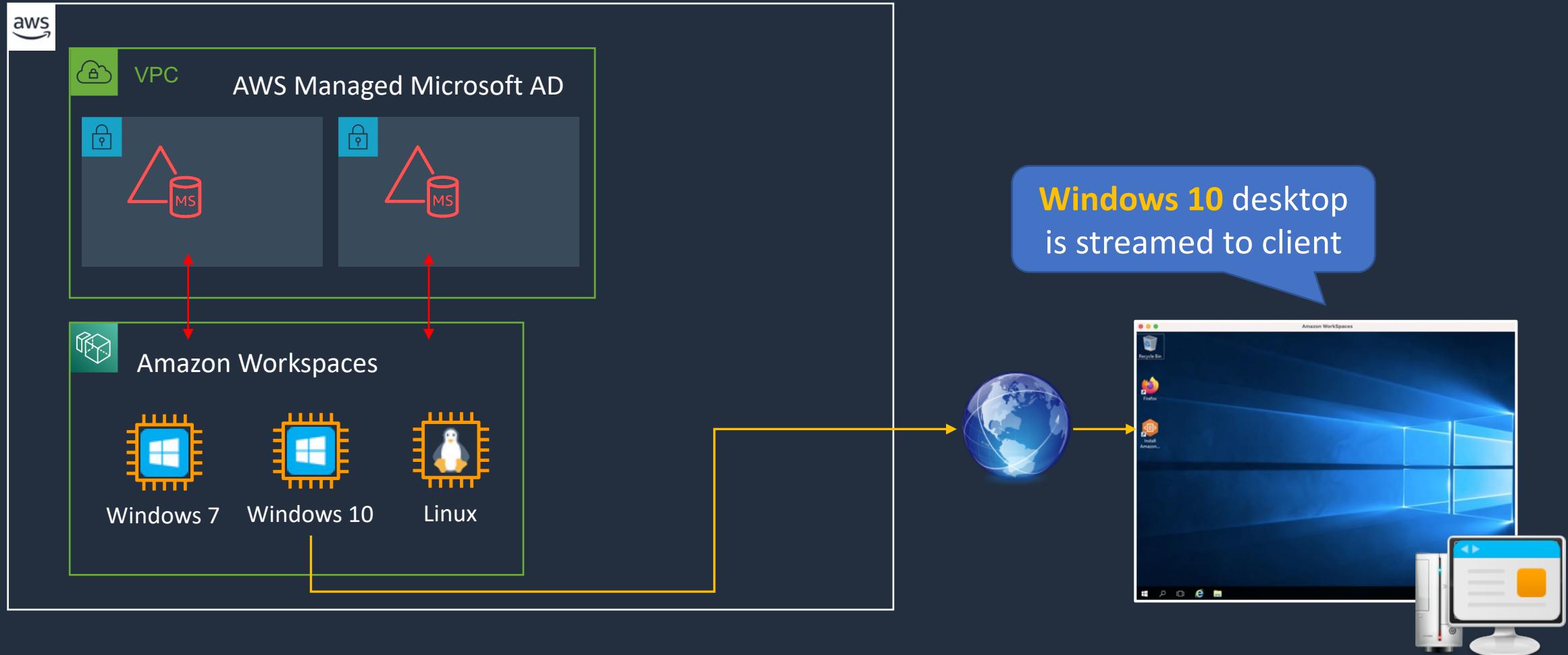
# Amazon Workspaces

---

- Managed **Desktop-as-a-Service** (DaaS) solution
- Provision either Windows or Linux desktops
- Simplifies delivery of desktops compared to traditional virtual desktop infrastructure (VDI) deployments



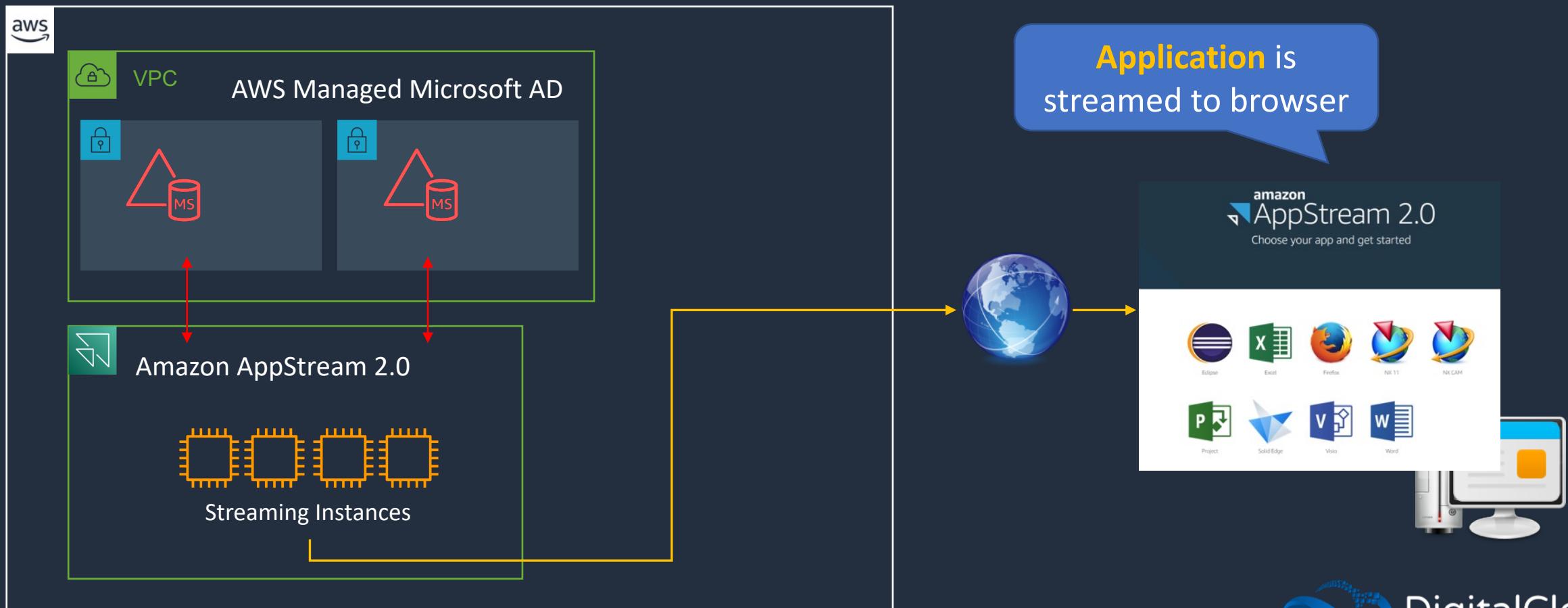
# Amazon Workspaces





# AWS AppStream 2.0

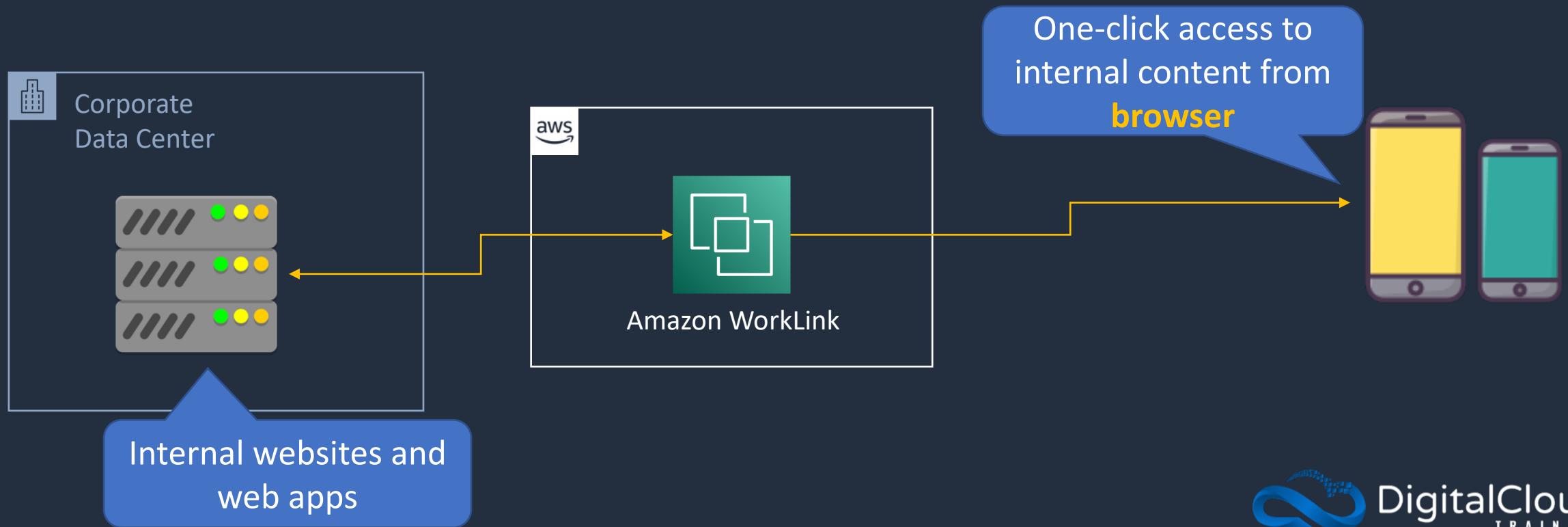
- Fully managed non-persistent application streaming service
- Alternative to popular products such as Citrix XenApp





# AWS WorkLink

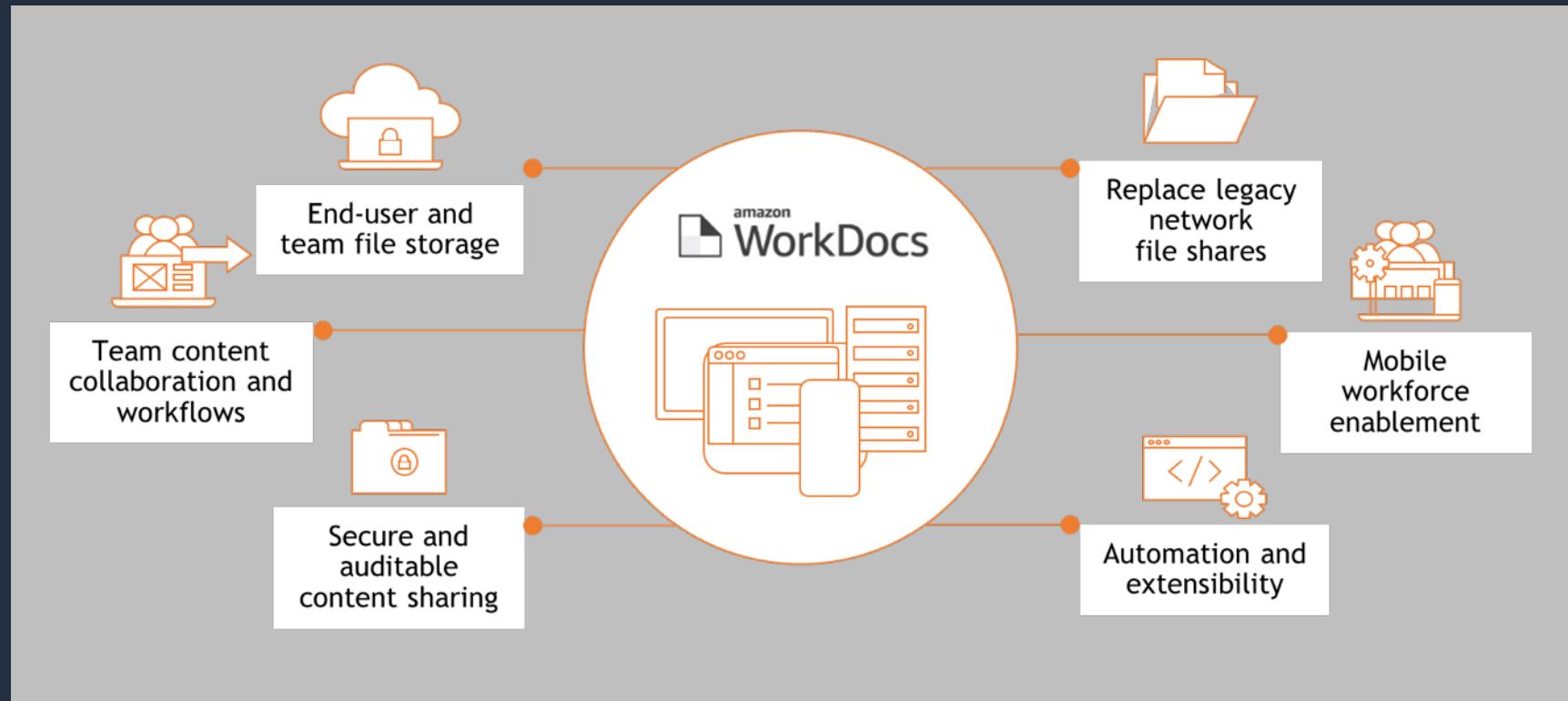
- Provides secure, one-click access to your internal websites and web apps using mobile phone browsers
- Does not require VPN client or App





# AWS WorkDocs

- Fully managed, secure content creation, storage, and collaboration service
- Create, edit, and share content that's centrally stored on AWS



# AWS IoT Core





# The Internet of Things (IoT)

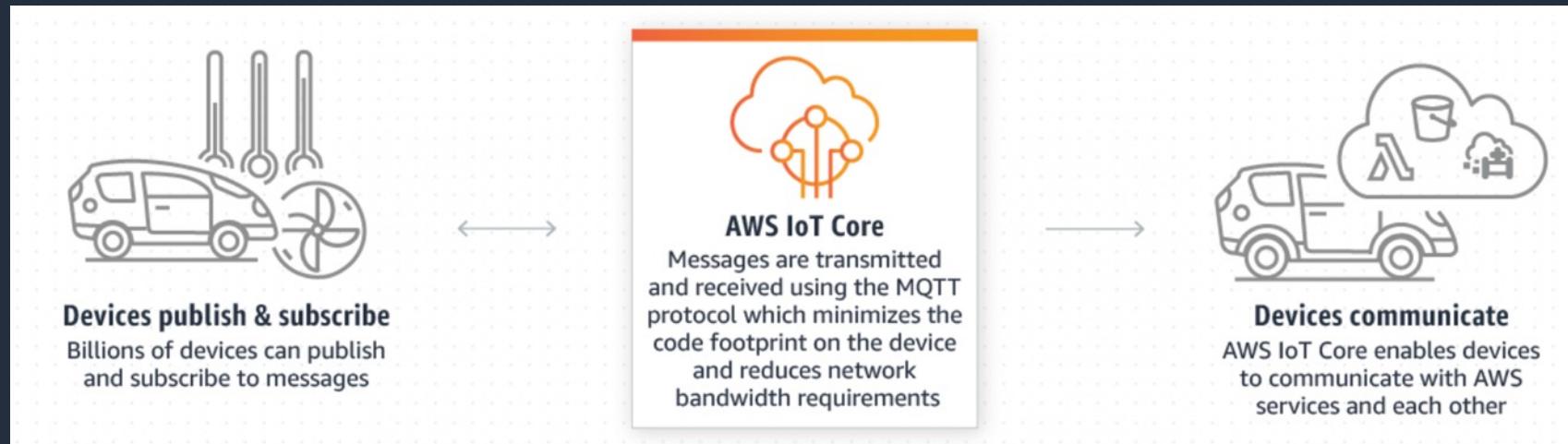
---

- Describes the network of physical objects that are embedded with sensors or software
- Each IoT device can communicate and exchange data with other devices and systems
- Use cases include:
  - Smart home automation
  - Smart healthcare
  - Manufacturing
  - Agriculture



# AWS IoT Core

- Lets you connect IoT devices to the AWS cloud without the need to provision or manage servers
- Can support billions of devices and trillions of messages



# SECTION 17

## Exam Preparation and Tips

# Booking your Exam



# Exam Preparation Tips





# Exam Preparation Tips

---

---

- Dedicate regular time to learning
- Use the free study plan
- Use practice tests early and regularly
- Review knowledge areas where you score poorly
- Don't book the exam until you're ready
- Non-native English speakers can request an extension (extra 30 minutes)
- If you've taken another AWS exam before, use your 50% discount voucher

# Exam Question Walkthrough

