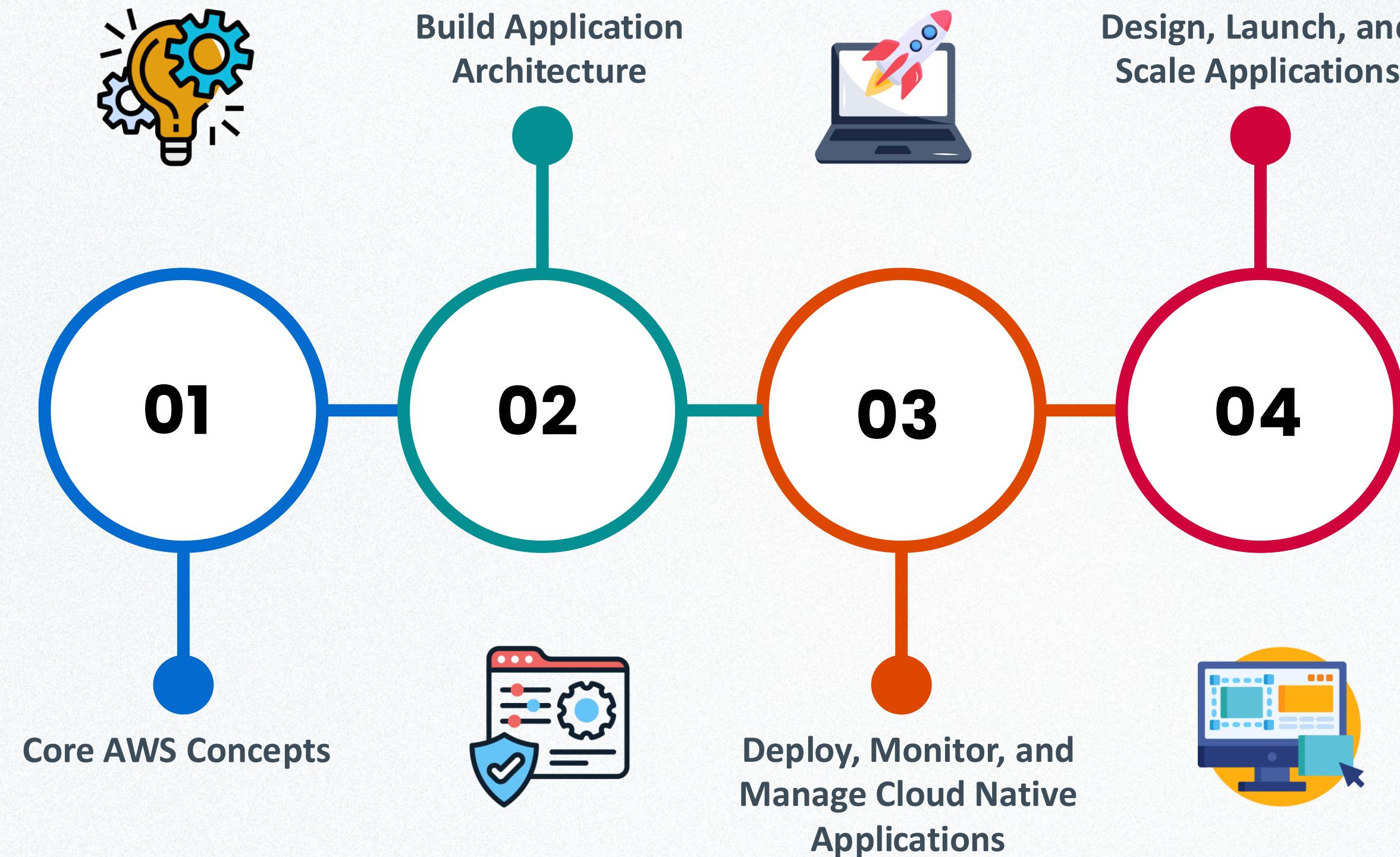




Build and Scale with AWS

A Hands-on Beginner's Guide





The Ultimate Linux Bootcamp for
DevOps SRE & Cloud Engineers



Practical Kubernetes –
Beyond CKA and CKAD



Automation with Ansible -
Hands-on DevOps



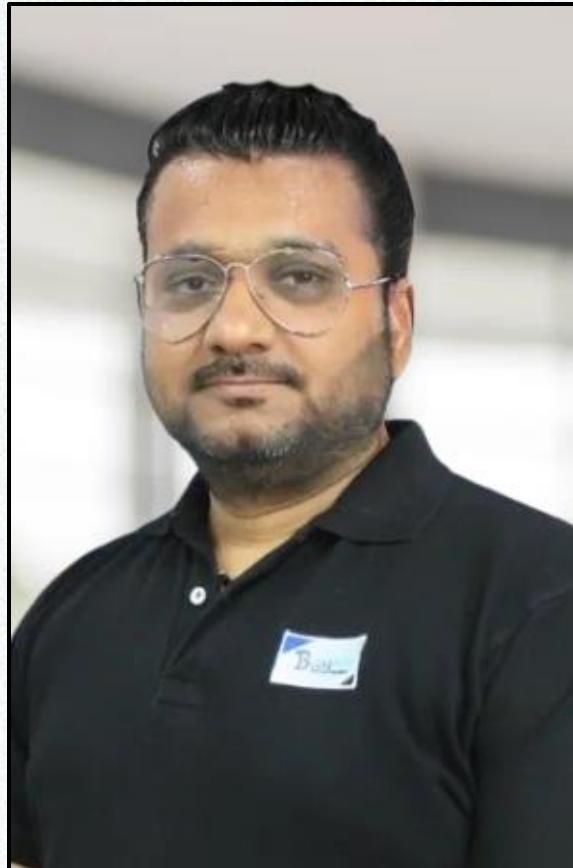
Kubernetes and
Cloud Native Associate



Argo CD for the Absolute
Beginners - Hands-On



Mastering Docker Essentials
- Hands-on



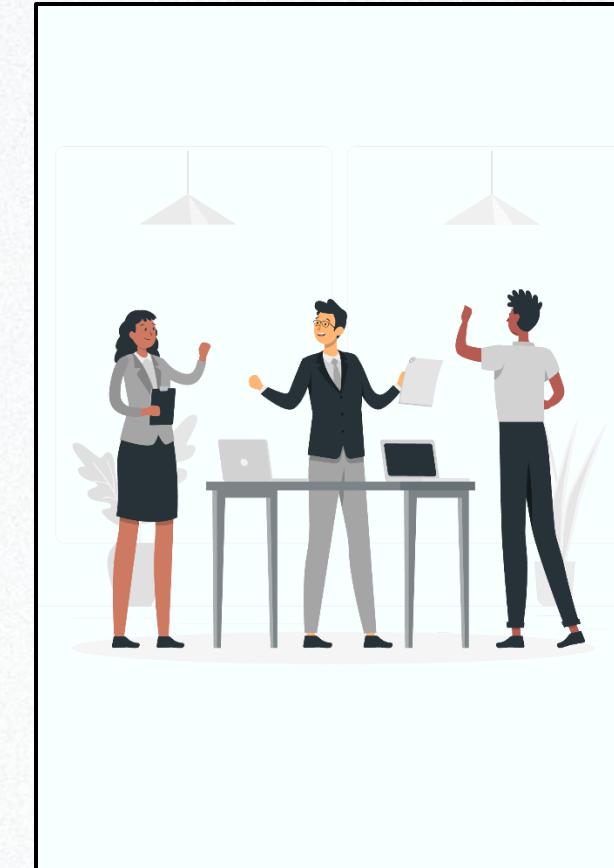
Kulbhushan Mayer



Yogesh Raheja



Deepthi Narayan



Thinknyx Team

How Will This Course Work?



How Will This Course Work?



Hands-On Demonstrations



Practice Exercises

Cloud Fundamentals with AWS

Section 1



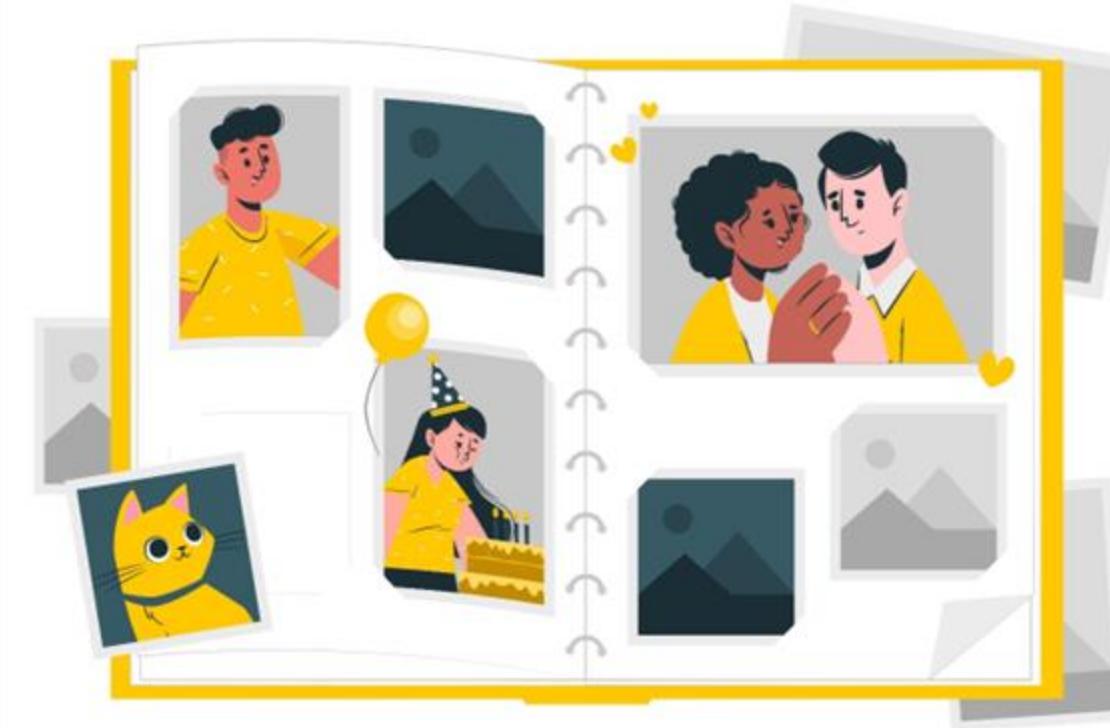
Cloud Fundamentals

- ❖ Advantages of cloud over on-premises infrastructure
- ❖ Fundamental concepts
- ❖ Essential characteristics
- ❖ Service models
- ❖ Deployment models

Introduction to Cloud

Introduction to Cloud

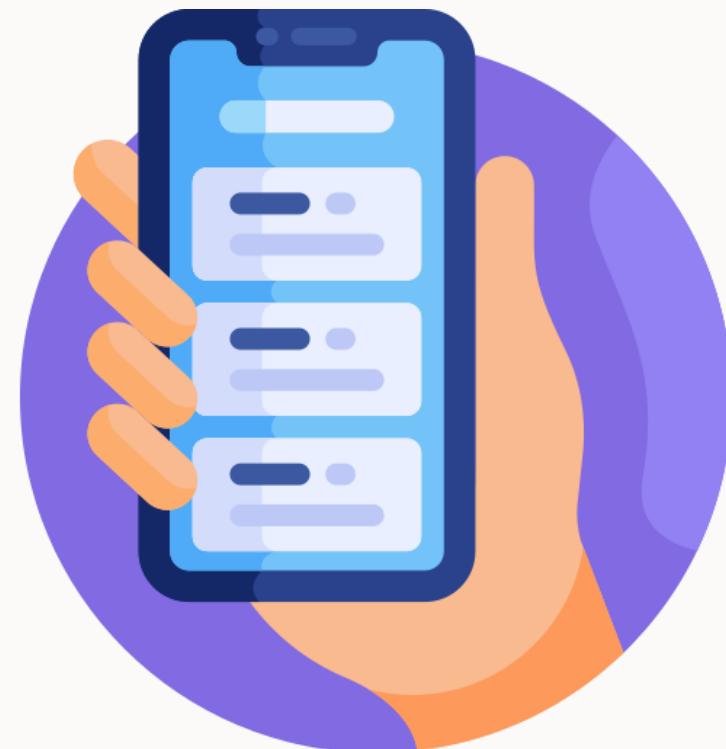
➤ What is Cloud Computing?



25 to 30 Pictures

Introduction to Cloud

➤ What is Cloud Computing?



Introduction to Cloud

➤ What is Cloud Computing?



GooglePhotos



Google Drive



OneDrive



iCloud

- ✓ Even if you lose your device, your files stay safe in the cloud

Introduction to Cloud

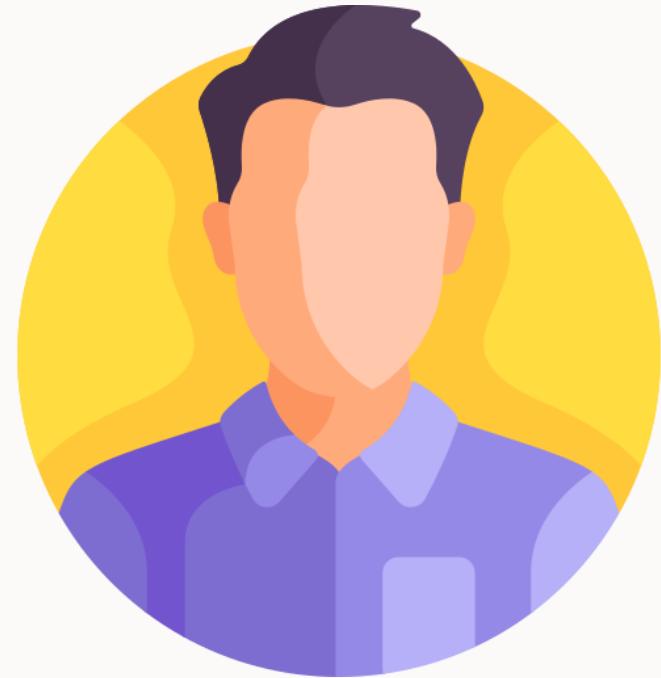
➤ What is Cloud Computing?

- ✓ Access your data anytime, anywhere
- ✓ Works across devices with an internet connection
- ✓ Seamless switching between devices
- ✓ Ideal for remote work and streaming



Introduction to Cloud

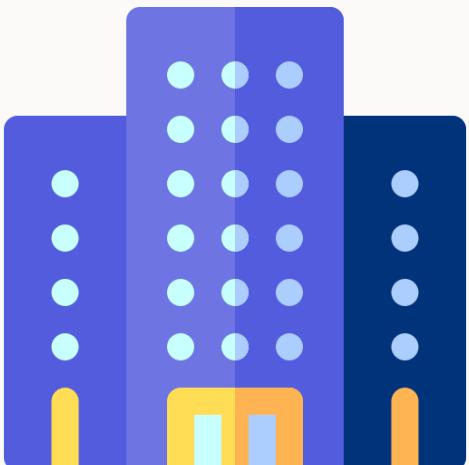
➤ What is On-Premises Computing?



Introduction to Cloud

➤ What is On-Premises Computing?

- ✓ **Traditional IT Model** – Organization owns, manages, and maintains hardware and software
- ✓ **Legacy Systems** – Supports mainframes, Solaris, HP-UX, and other enterprise technologies
- ✓ **Physical Infrastructure** – Servers, storage, and networking housed in dedicated data centers or server rooms



- ✓ **Humble Beginnings** – Many businesses start small, just like Google began in a garage
- ✓ **Traditional Approach** – Initial reliance on minimal infrastructure before growing
- ✓ **Gradual Expansion** – Organizations scale up, investing in their own data centers

Introduction to Cloud

➤ Limitations of On-Premises Computing

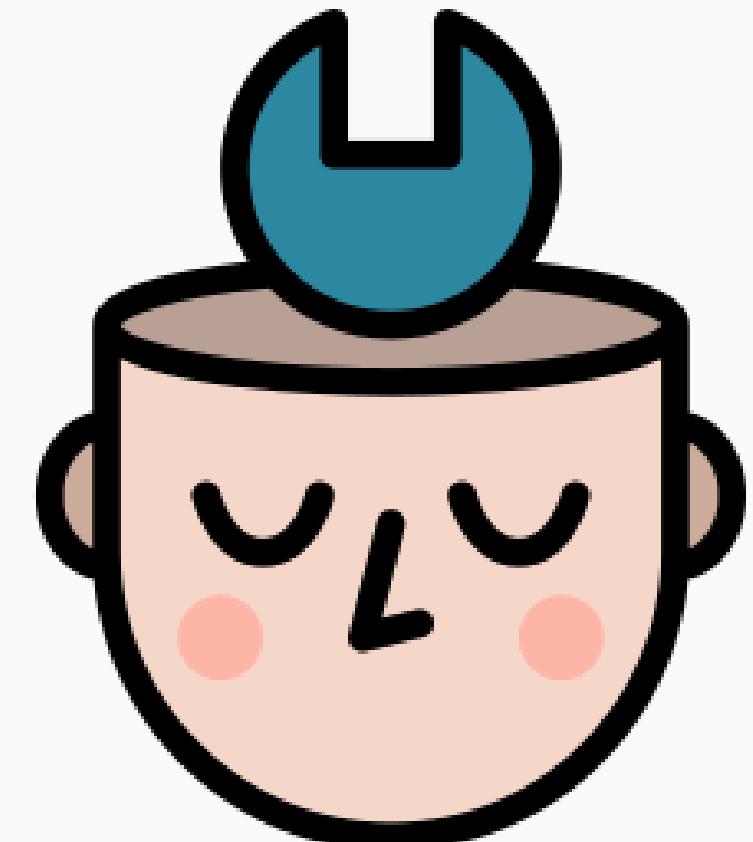
- **High Costs & Upfront Investments** – Requires significant capital for hardware, storage, and infrastructure, with risky resource planning



Introduction to Cloud

➤ Limitations of On-Premises Computing

- **Complex Maintenance & IT Overhead** – Needs a dedicated IT team for upkeep, increasing costs and operational complexity



Introduction to Cloud

➤ Limitations of On-Premises Computing

- **Scalability Challenges**

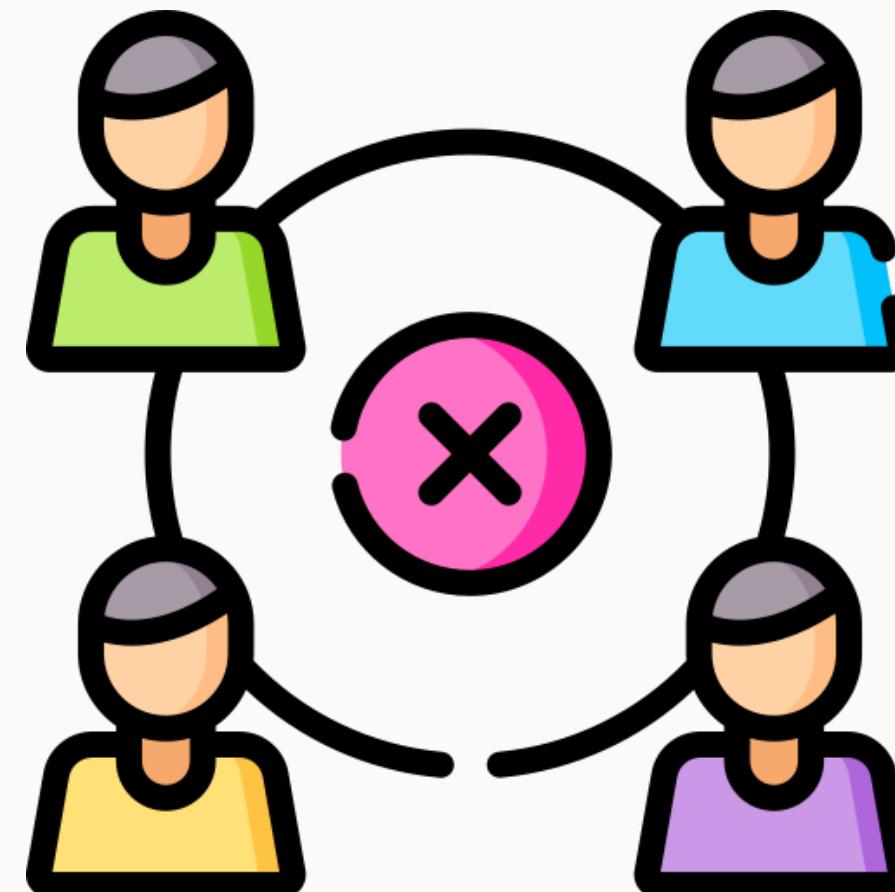
- **Overprovisioning** – Excess capacity leads to wasted resources
- **Underprovisioning** – Insufficient capacity causes performance issues and downtime



Introduction to Cloud

➤ Limitations of On-Premises Computing

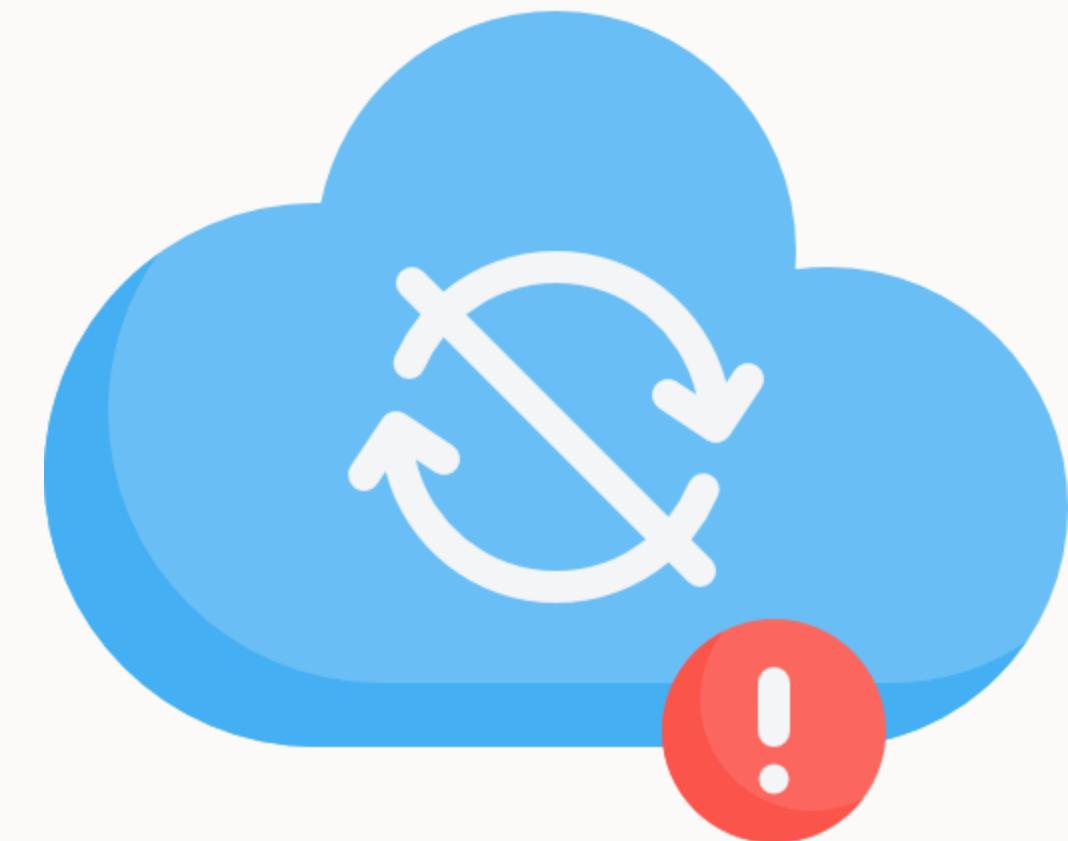
- **Limited Accessibility & Collaboration** – Restricted remote access hinders global teamwork



Introduction to Cloud

➤ Limitations of On-Premises Computing

- **Backup & Disaster Recovery Challenges** – Managing backups and recovery solutions is costly and complex, increasing the risk of data loss



Introduction to Cloud

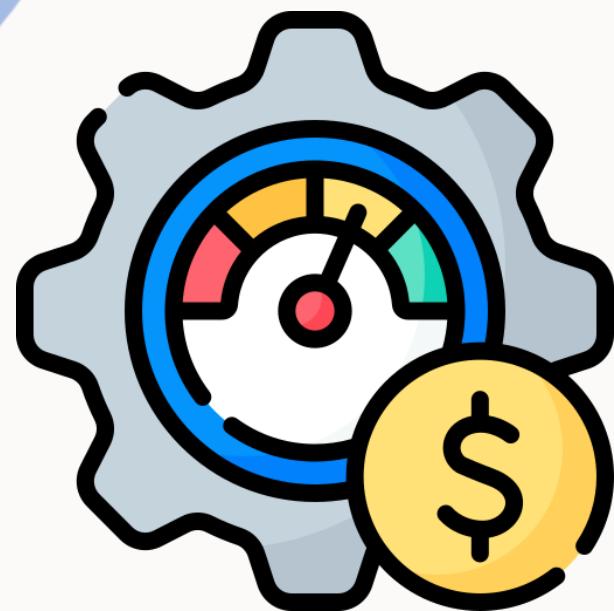
➤ Why Cloud Computing?

- More flexible
- Scalable
- Cost-effective

Introduction to Cloud

➤ Advantages of moving to the cloud

✓ Cost Efficiency: Pay for What You Use



No Upfront Investment – Avoid large capital expenses on hardware

Pay-as-You-Go – Only pay for the resources consumed

On-Demand Scaling – Increase or decrease capacity based on need

Avoid Overprovisioning – Prevent unnecessary spending on idle resources

Example: Black Friday – Temporarily scale up servers and scale down post-event

Optimized Costs – Reduce waste and improve financial efficiency

Introduction to Cloud

➤ Advantages of moving to the cloud

✓ Simplified Maintenance & Reduced IT Overhead

Managed Infrastructure – Cloud providers handle hardware, updates, and security

Less IT Burden – No need for large in-house IT teams for maintenance

Automatic Updates – Software and security patches are handled by the provider

Focus on Innovation – IT teams can prioritize business growth instead of server management

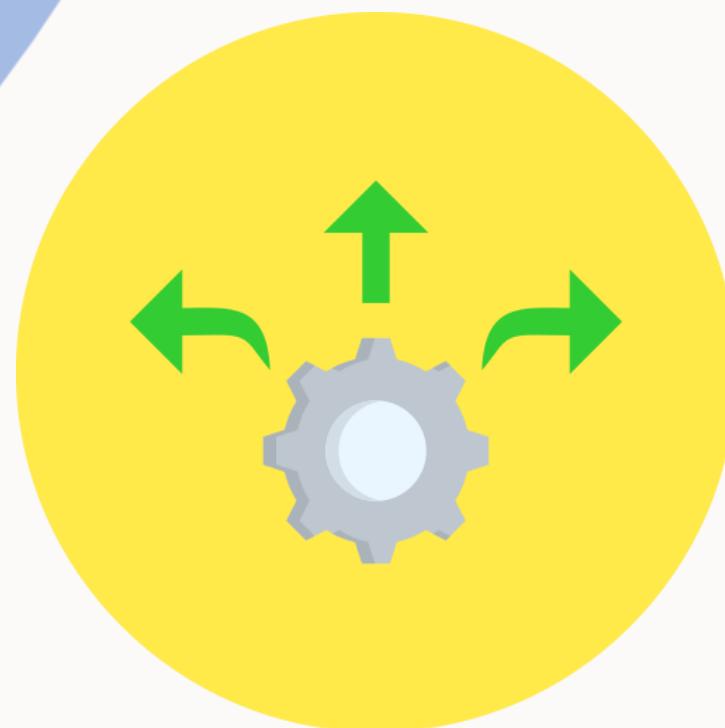
Cost Savings – Reduce expenses on IT staff and infrastructure upkeep



Introduction to Cloud

➤ Advantages of moving to the cloud

✓ Scalability & Flexibility



Instant Scaling – Expand or reduce resources in real time

No Hardware Delays – Avoid weeks/months of setup for on-premises infrastructure

Optimized Performance – Scale to meet demand without service disruptions

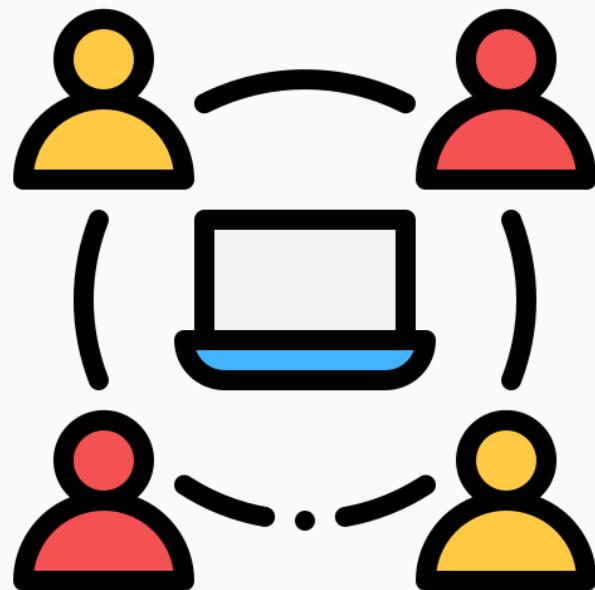
Ideal for Startups – Start small and grow as needed without upfront costs

Cost-Effective Expansion – Pay only for the resources used, adjusting as business needs change

Introduction to Cloud

➤ Advantages of moving to the cloud

✓ Global Accessibility & Remote Collaboration



Access from Anywhere – Work from any location with an internet connection

No Geographic Limits – Collaborate across different time zones seamlessly

Supports Remote & Hybrid Work – Enables flexible work environments

Real-Time Collaboration – Teams can work on shared files and applications instantly

Increased Productivity – Employees stay connected and efficient, regardless of location

Introduction to Cloud

➤ Advantages of moving to the cloud

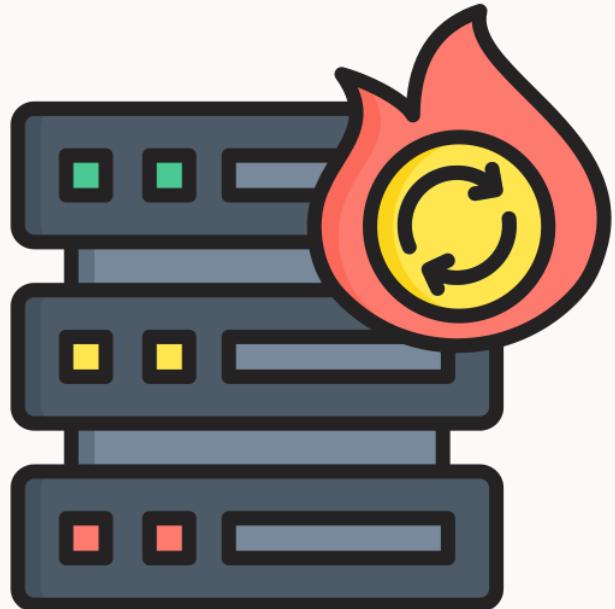
- ✓ **Built-in Backup, Security & Disaster Recovery**

Automated Backups – No manual effort required for data protection

Disaster Recovery – Minimize downtime with built-in failover solutions

Data Redundancy – Replication across multiple geographic locations

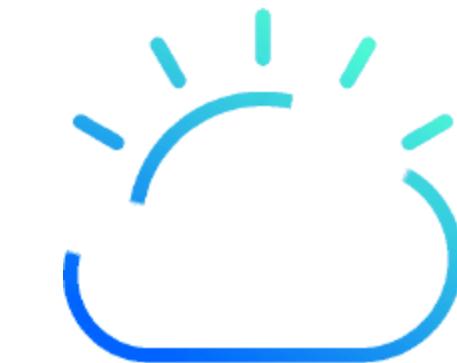
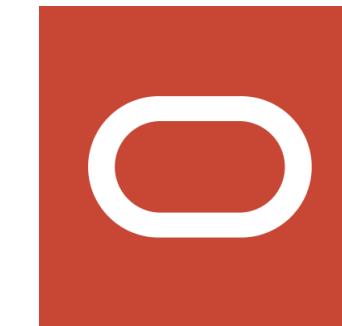
Enhanced Security – Encryption, access controls, and compliance measures



Introduction to Cloud

➤ Cloud Computing: From Product to Service

Cloud computing delivers computing as a service rather than a product



Essentials Characteristics

Essentials Characteristics

➤ NIST

- ✓ National Institute of Standards and Technology
- ✓ U.S. agency setting technology standards
- ✓ Enhances cybersecurity and innovation
- ✓ Defines cloud computing framework
- ✓ Essential characteristics, service models, and deployment models

Essentials Characteristics

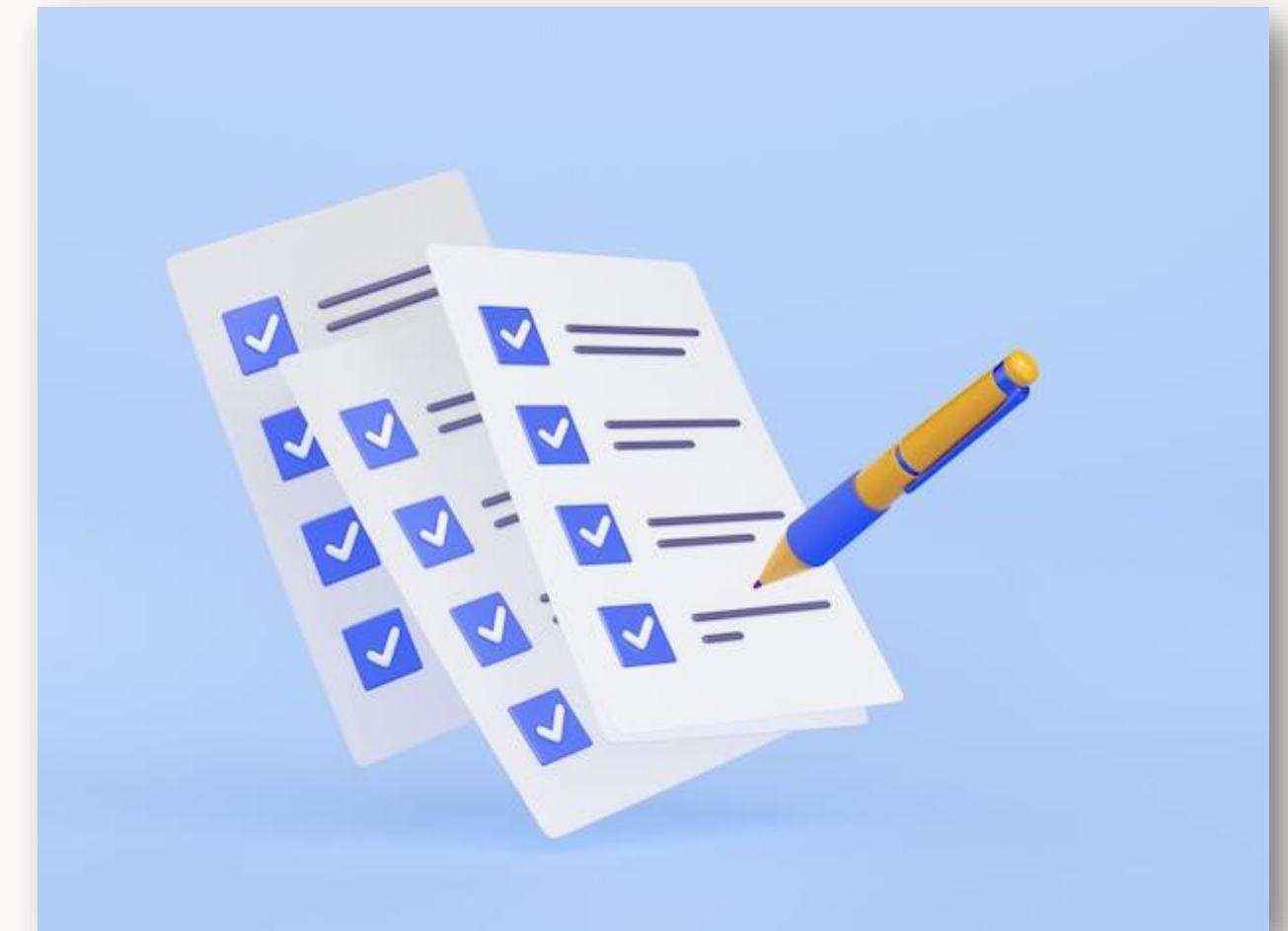
➤ Five essential characteristics

- ✓ On-Demand Self-Service
- ✓ Broad Network Access
- ✓ Resource Pooling
- ✓ Rapid Elasticity
- ✓ Measured Service

Essentials Characteristics

➤ On-Demand Self-Service

- ✓ Provision resources without human intervention
- ✓ Includes servers, storage, and networking
- ✓ Scales up or down as needed
- ✓ Quick setup—ready in seconds



Essentials Characteristics

➤ Broad Network Access

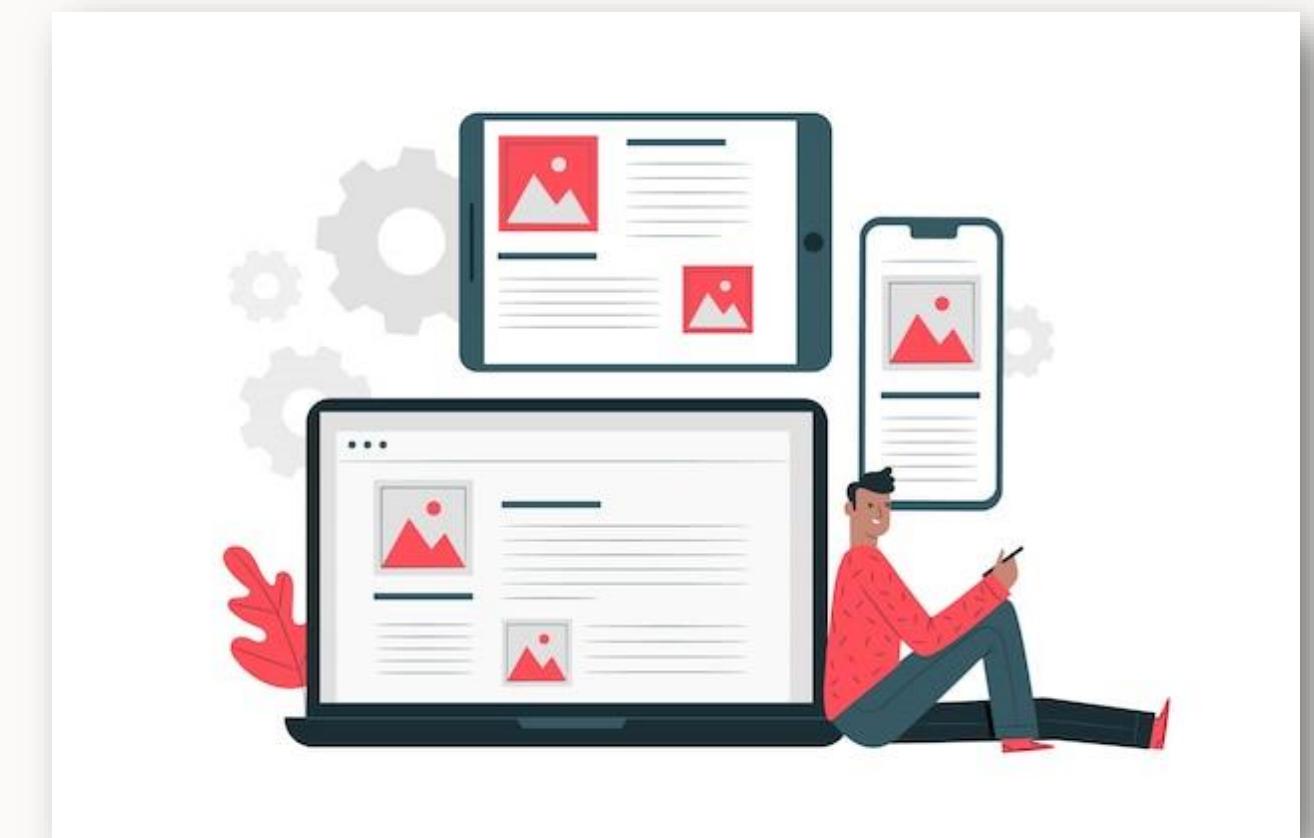
- ✓ Access cloud services from any device via the internet
- ✓ Enables remote access and collaboration
- ✓ Removes location restrictions
- ✓ Like Google Drive or iCloud—upload on one device, access on another



Google Drive



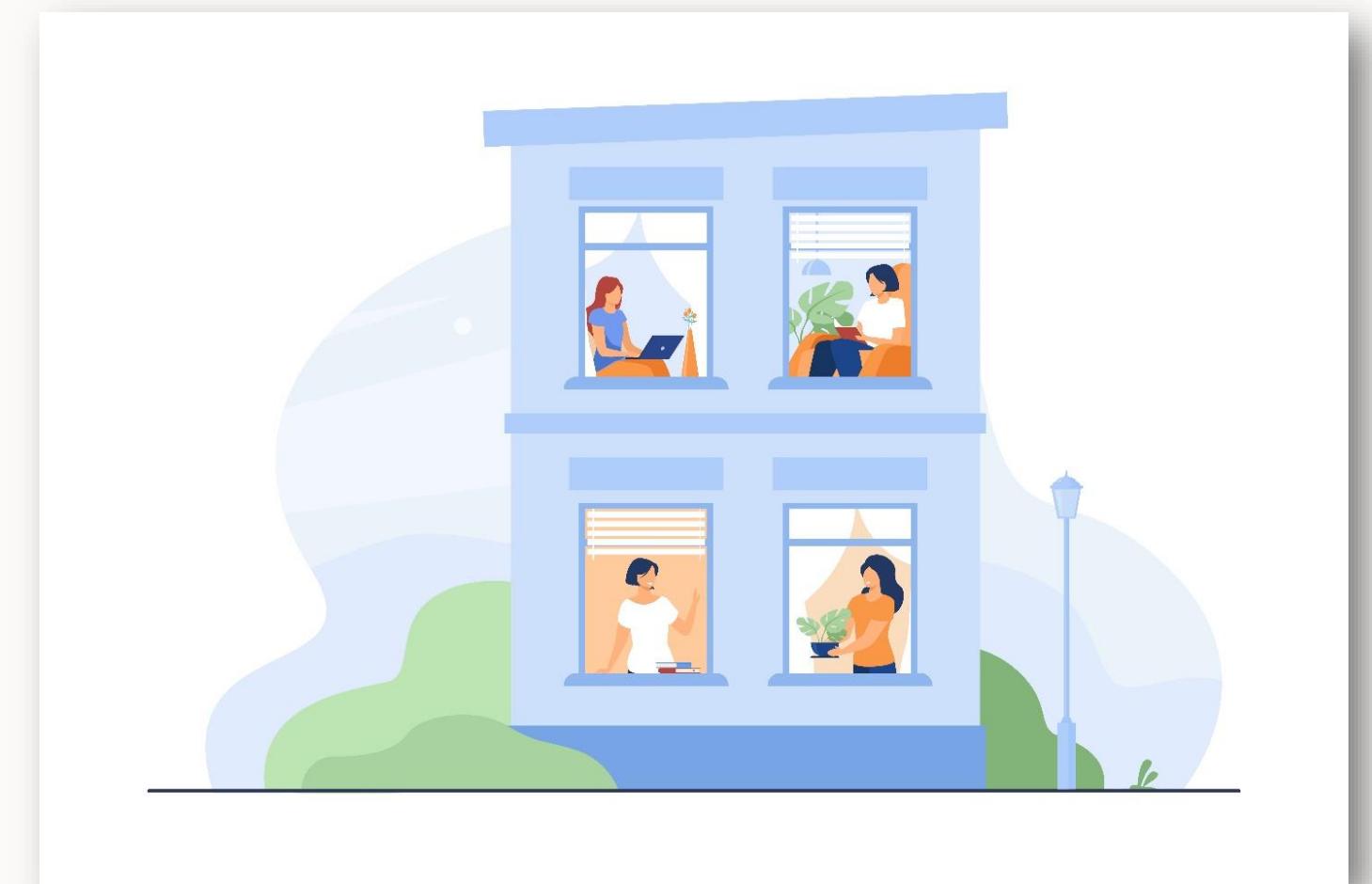
iCloud



Essentials Characteristics

➤ Resource Pooling

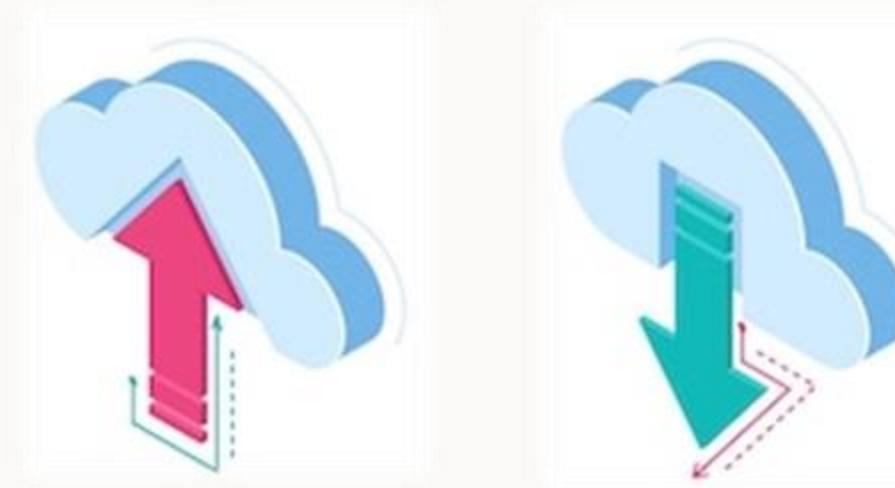
- ✓ Multi-tenant model for efficient resource sharing
- ✓ Dynamically allocates storage, processing, and bandwidth
- ✓ Users don't control exact resource locations
- ✓ Can specify regions for compliance and performance



Essentials Characteristics

➤ Rapid Elasticity

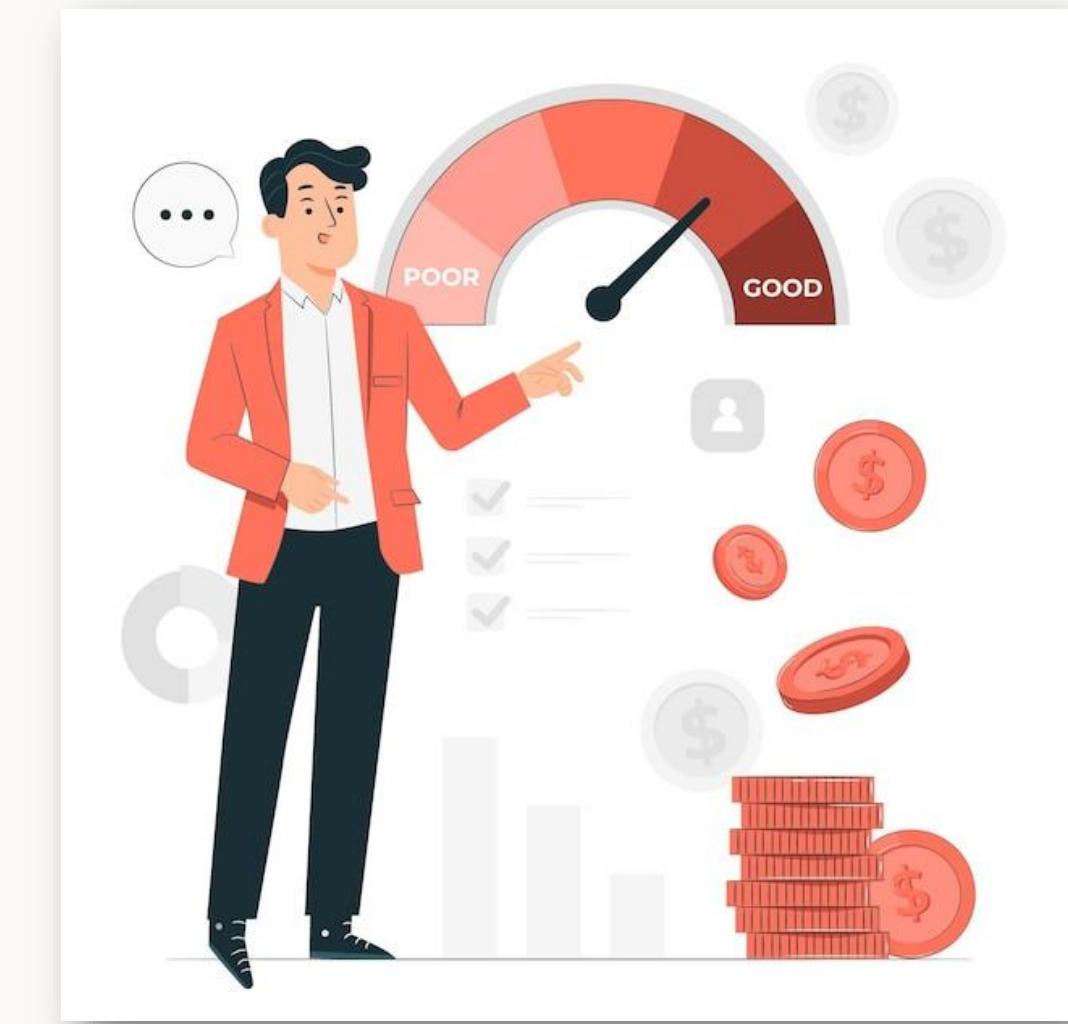
- ✓ Automatically scales resources based on demand
- ✓ Handles normal loads and traffic spikes (e.g., Black Friday)
- ✓ Ensures performance without overprovisioning



Essentials Characteristics

➤ Measured Service

- ✓ Pay-as-you-go billing based on usage
- ✓ Monitored, controlled, and optimized resource consumption
- ✓ Ensures cost transparency and efficient IT spending



Service Models

Service Models

➤ What are your options?



Service Models

➤ What are your options?

1. Make everything from scratch
2. Use pre-made ingredients
3. Get pizza delivered
4. Dine out at a restaurant



Service Models

Make at home

- Dining Table
- Soda
- Electric/Gas
- Oven
- Fire
- Pizza Dough
- Tomato Sauce
- Toppings
- Cheese

On-Premise

- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking



Set up and maintain their own servers, networks, and security

You Manage

Service Provider Manages

Service Models

Take and bake

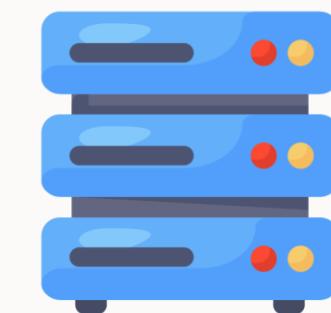
- Dining Table
- Soda
- Electric/Gas
- Oven
- Fire
- Pizza Dough
- Tomato Sauce
- Toppings
- Cheese

IaaS

- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

 You Manage

 Service Provider Manages



- Rent virtual machines on**
- AWS EC2
 - Google Compute Engine
 - Microsoft Azure Virtual Machines



Service Models

Pizza delivery

- Dining Table
- Soda
- Electric/Gas
- Oven
- Fire
- Pizza Dough
- Tomato Sauce
- Toppings
- Cheese

PaaS

- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

You Manage

Service Provider Manages



Deploy your code on

- AWS Elastic Beanstalk
- Google App Engine
- Microsoft Azure App Services

Service Models

Dine out

- Dining Table
- Soda
- Electric/Gas
- Oven
- Fire
- Pizza Dough
- Tomato Sauce
- Toppings
- Cheese

SaaS

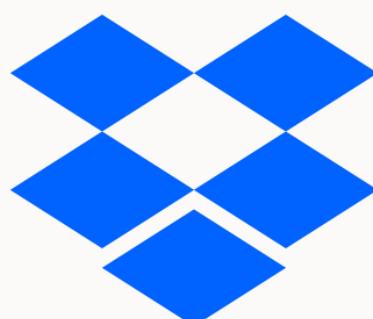
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking



Google Drive



Office 365



You Manage

Service Provider Manages

Service Models

➤ Comparing Flexibility & Pricing of Cloud Service Models

Model	Flexibility	Cost
On-Premises	Full control over everything	Most expensive (hardware & maintenance)
IaaS	High (manage software, provider handles hardware)	Pay-as-you-go, cost-effective for scaling
PaaS	Moderate (focus on apps, provider handles infra)	Higher than IaaS, but reduces operational work
SaaS	Low (use the software, no management required)	Subscription-based, cost-effective but least customizable

Deployment Models

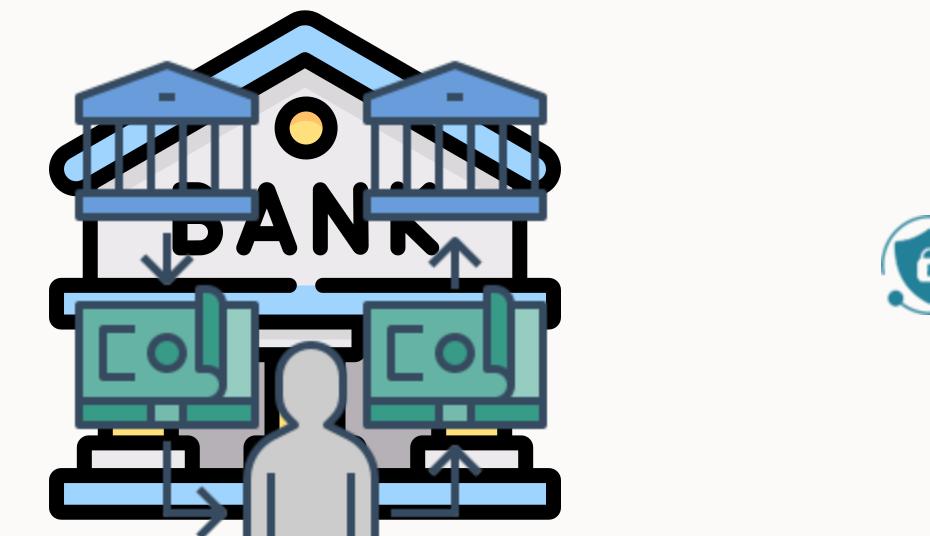
Deployment Models



Deployment Models

➤ Private Cloud

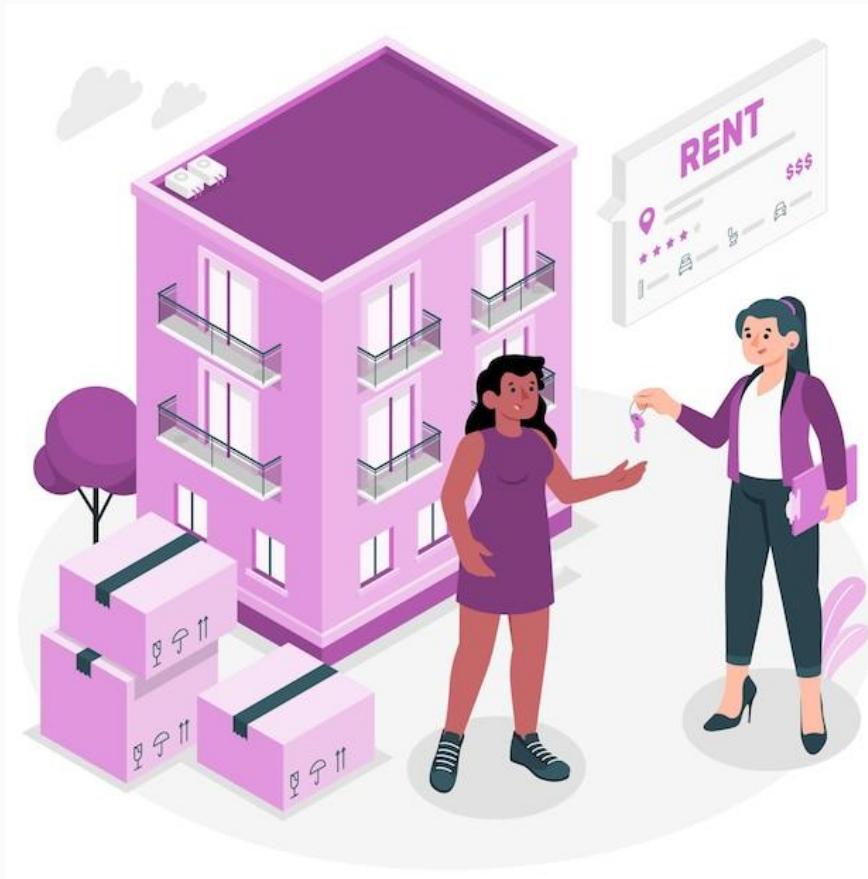
- ✓ Like owning a house—full control over security and maintenance
- ✓ Dedicated cloud infrastructure for a single organization
- ✓ Can be hosted on-premises or by a third-party provider



Deployment Models

➤ Public Cloud

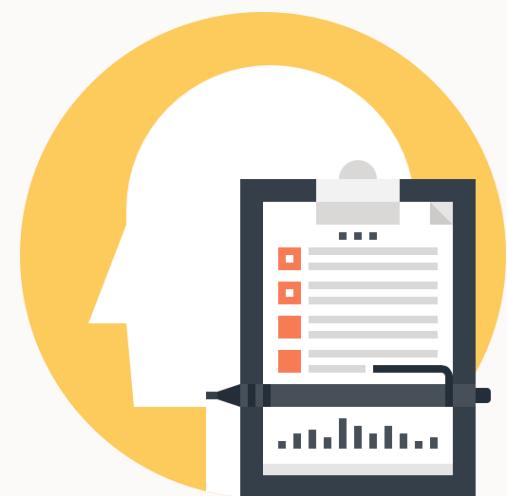
- ✓ Like renting an apartment—use resources as needed
- ✓ Cloud providers (AWS, Azure, Google Cloud) offer resources to multiple customers
- ✓ Hosted over the internet



Deployment Models

➤ Hybrid Cloud

- ✓ Like owning a house and renting a vacation home
- ✓ Businesses store sensitive data in a private cloud
- ✓ Use public cloud for less critical tasks



Deployment Models

➤ Community Cloud

- ✓ Like a gated community—shared resources with private spaces
- ✓ Designed for organizations with similar needs (e.g., government agencies, universities)
- ✓ Multiple organizations share the cloud infrastructure

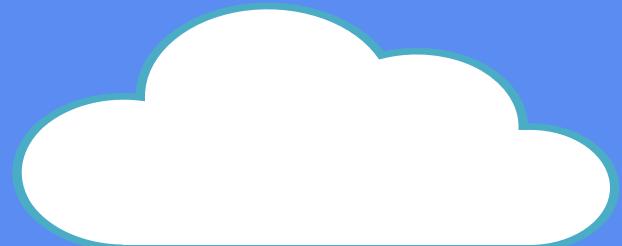


Summary

- Importance and advantages of cloud computing
- **NIST definition:** Five essential characteristics, three service models, and four deployment models

Cloud Fundamentals with AWS

Section 2



AWS

- ❖ AWS global infrastructure
- ❖ Types of AWS services
- ❖ Setting up your AWS account
- ❖ Guided tour of the AWS dashboard

Introduction to AWS

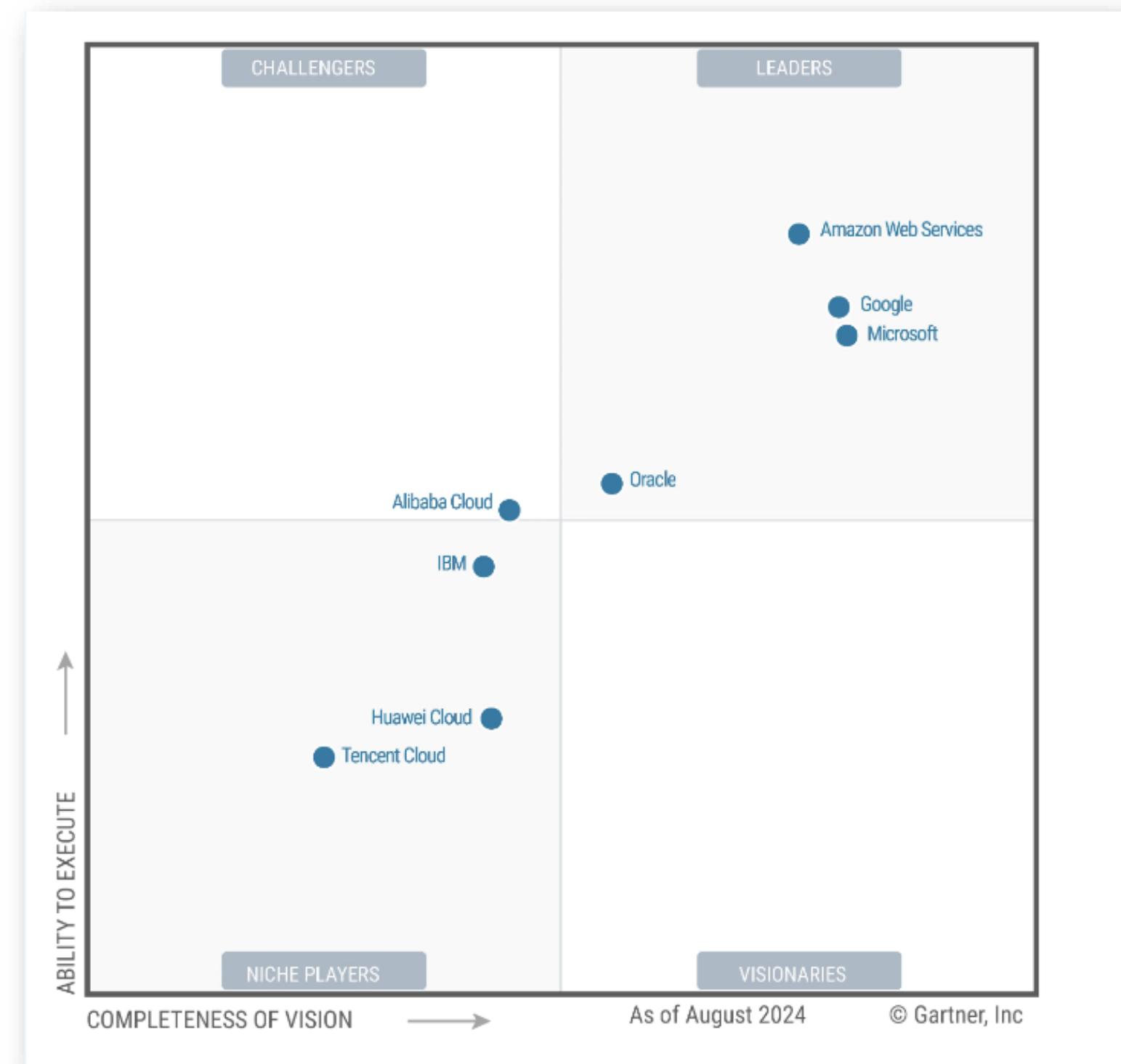
Introduction to AWS

➤ What is AWS?

- ✓ Amazon Web Services
- ✓ Launched in 2006
- ✓ Provides on-demand technology services
- ✓ Pay-as-you-go pricing model
- ✓ No upfront hardware investments required
- ✓ Manages infrastructure for businesses
- ✓ Enables focus on growth and innovation
- ✓ Allows quick deployment and scaling of applications
- ✓ Helps in cost optimization

Introduction to AWS

➤ AWS: A Market Leader



Courtesy Gartner

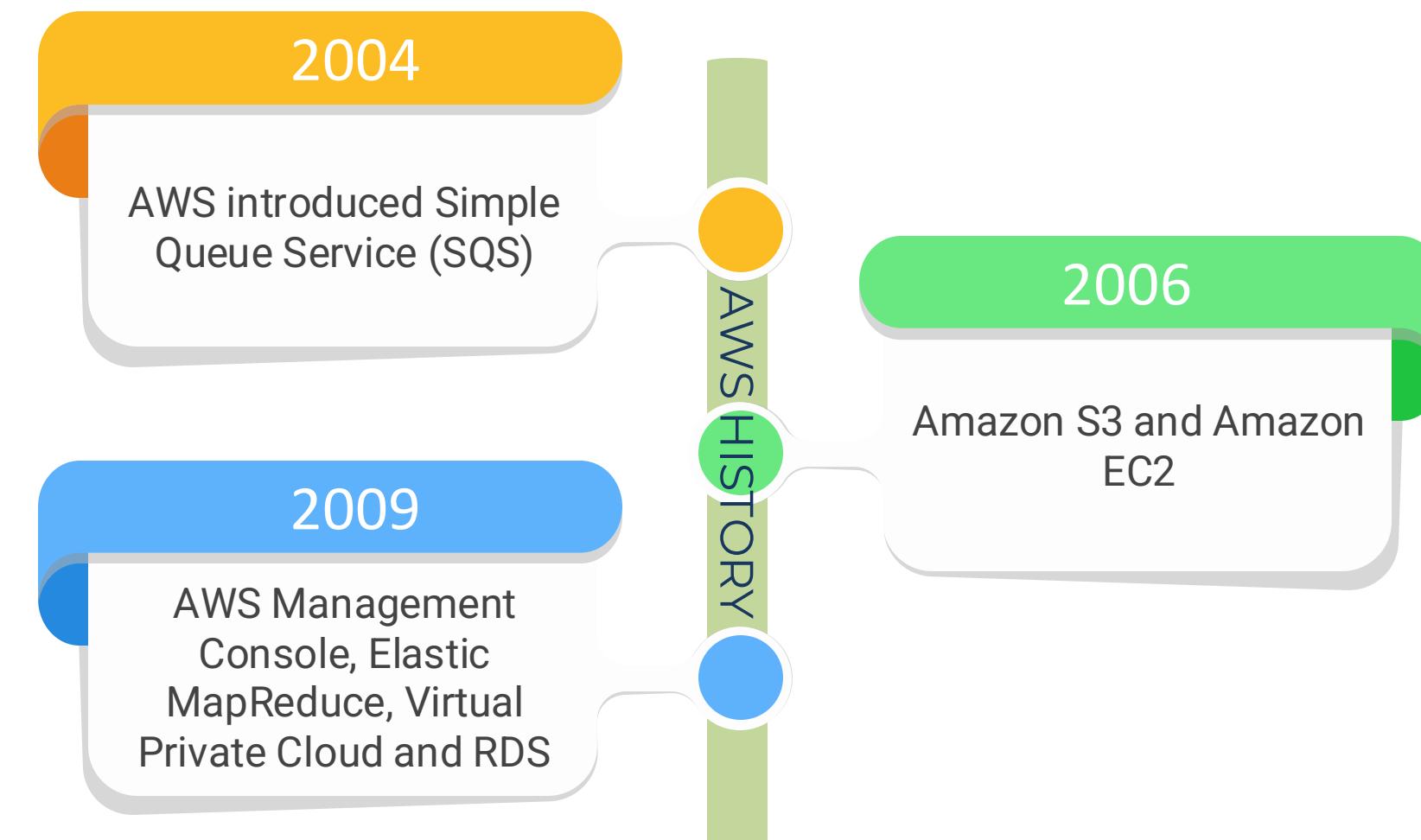
Introduction to AWS

➤ AWS History



Introduction to AWS

➤ AWS History



Introduction to AWS

➤ AWS History



Introduction to AWS

➤ AWS History



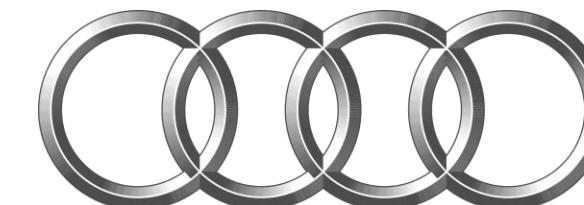
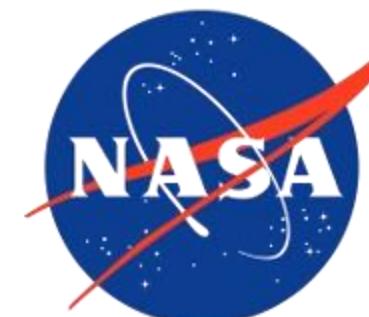
Introduction to AWS

➤ AWS History

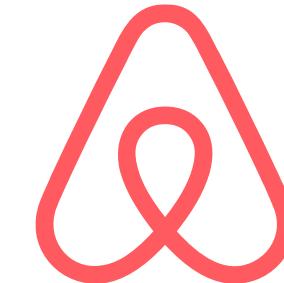


Introduction to AWS

➤ AWS History



Lenovo™



Coca-Cola



Introduction to AWS

➤ Key Benefits of AWS

✓ **Security**

- ✓ Strong data security with custom-built infrastructure
- ✓ 24/7 monitoring
- ✓ Encrypted data across the global network
- ✓ Full customer control over encryption, movement, and retention



Introduction to AWS

➤ Key Benefits of AWS

✓ Availability

- ✓ High availability with multiple Availability Zones
- ✓ Minimum of three independent Availability Zones per Region
- ✓ Isolates issues to prevent widespread impact
- ✓ Ensures continued operation even if one Region faces problems



Introduction to AWS

➤ Key Benefits of AWS

✓ Performance



- ✓ Low-latency, high-performance networking
- ✓ 400 GbE fiber backbone
- ✓ Local Zones for improved proximity to users
- ✓ Wavelength for ultra-low latency needs

Introduction to AWS

➤ Key Benefits of AWS

✓ Scalability

- ✓ Quickly scale resources as needed
- ✓ Reduces over-provisioning
- ✓ Minimizes costs
- ✓ Meets growing demands efficiently



Introduction to AWS

➤ Key Benefits of AWS

✓ **Flexibility**

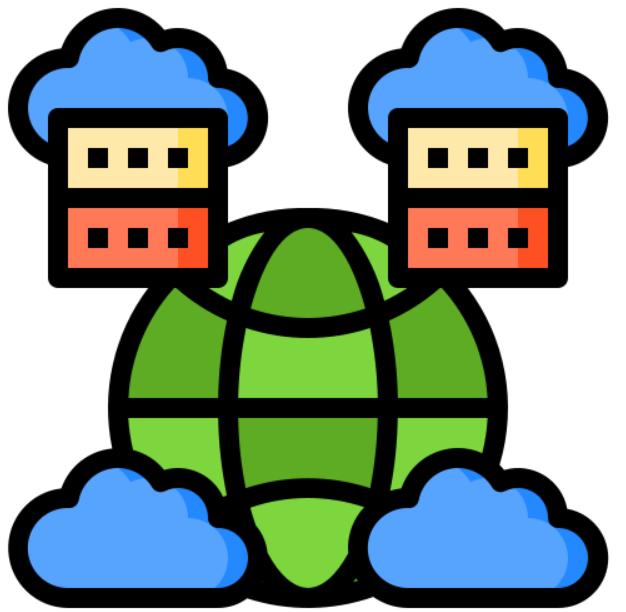
- ✓ Deploy workloads across Regions, Availability Zones, Local Zones, Wavelength, and Outposts
- ✓ Unified network and AWS services across all options



Introduction to AWS

➤ Key Benefits of AWS

✓ Global Reach



- ✓ Largest global infrastructure
- ✓ Deploy workloads closer to users
- ✓ Supports satellite services with AWS Ground Station

Regions, Availability Zones & Edge Locations

Regions, Availability Zones & Edge Locations

114 Availability Zones

36 launched Regions
each with multiple Availability Zones

700+ CloudFront POPs
and 13 Regional edge caches

Regions, Availability Zones & Edge Locations

➤ AWS Region

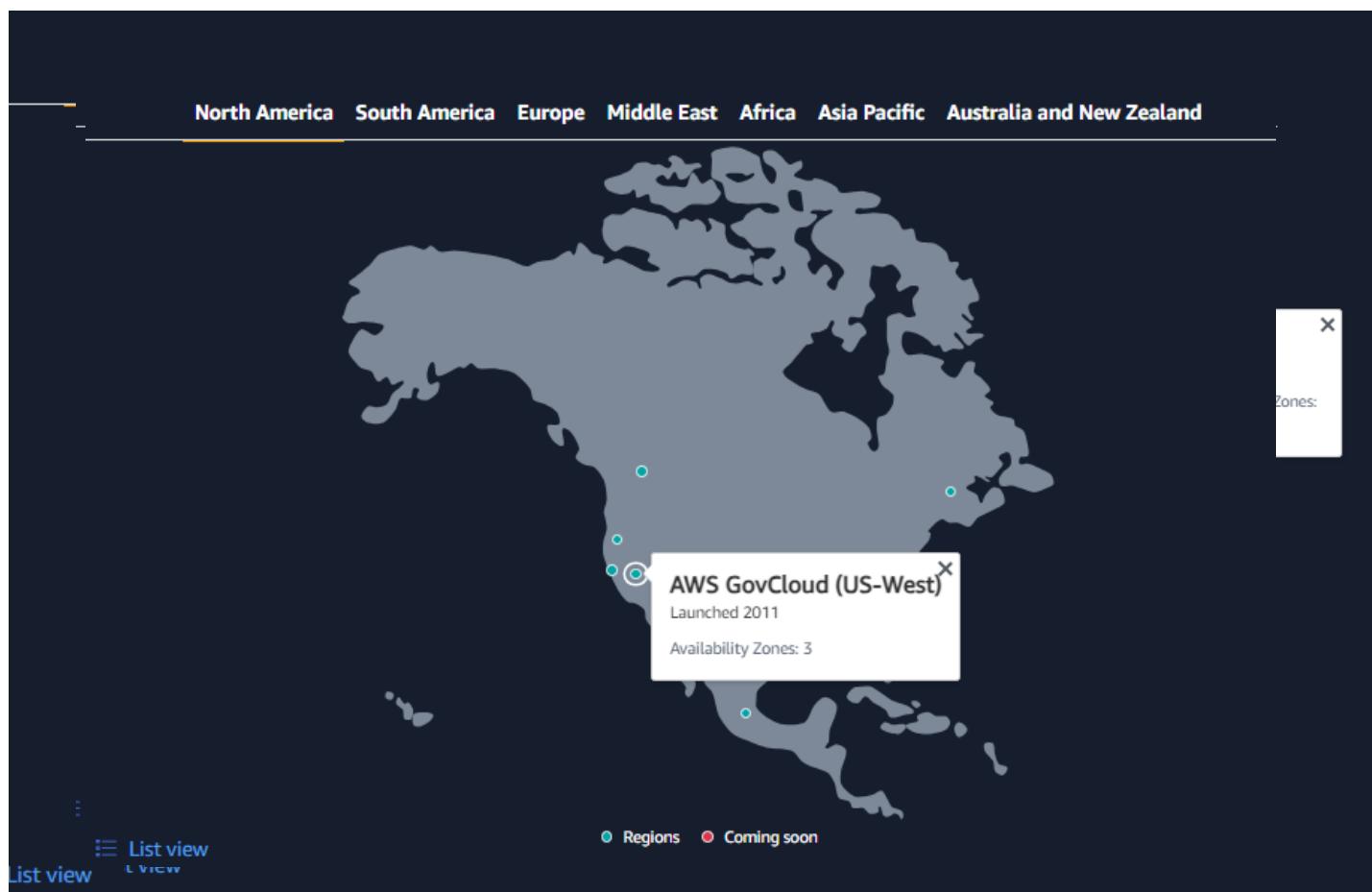
- ✓ Physical location with multiple data centers
- ✓ Each Region has at least three Availability Zones
- ✓ Availability Zones are isolated but connected

[North America](#) [South America](#) [Europe](#) [Asia Pacific](#) [Africa](#) [Middle East](#)

Regions, Availability Zones & Edge Locations

➤ AWS Region

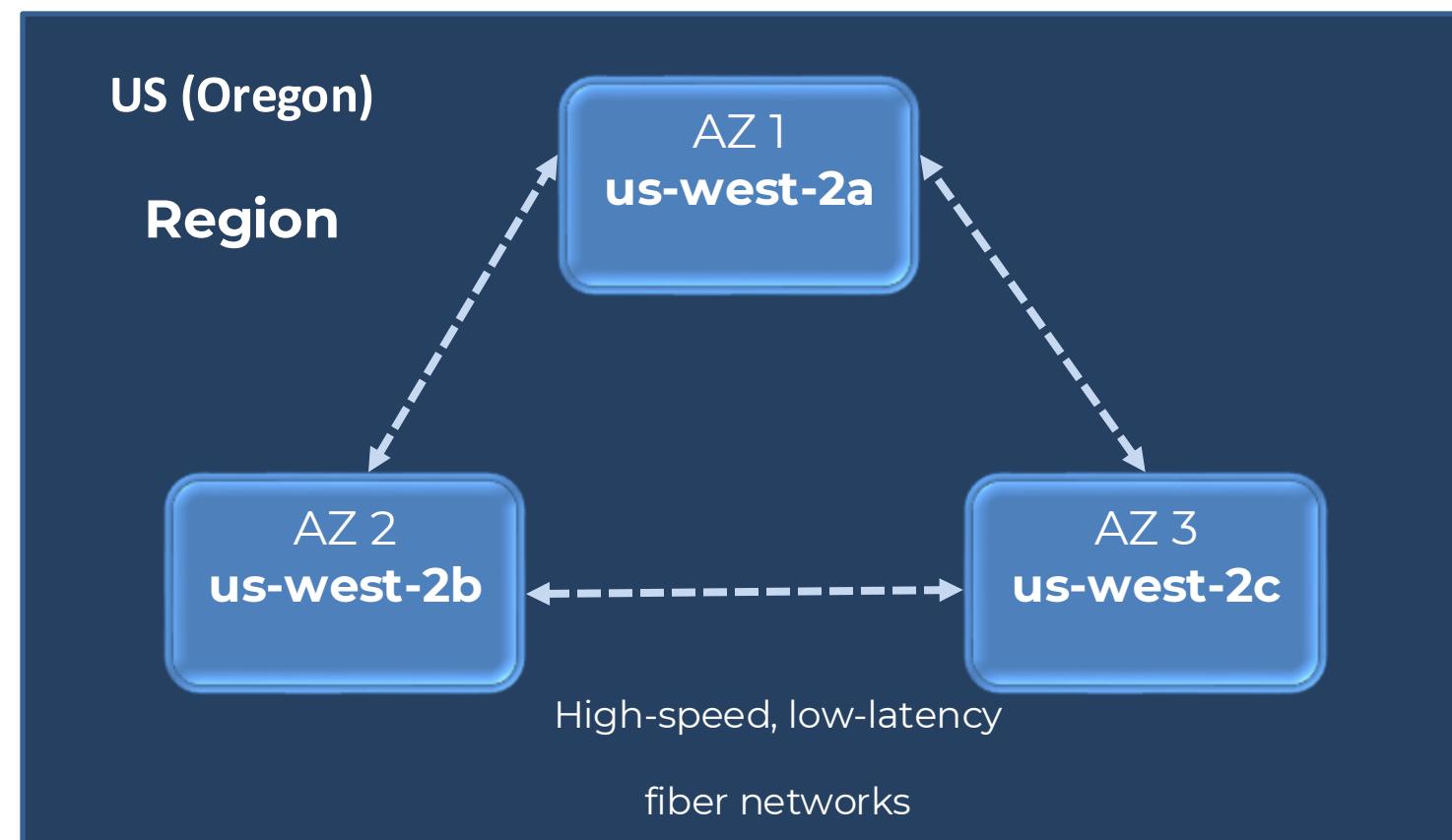
- ✓ Physical location with multiple data centers
- ✓ Each Region has at least three Availability Zones
- ✓ Availability Zones are isolated but connected



Regions, Availability Zones & Edge Locations

➤ Availability Zone (AZ)

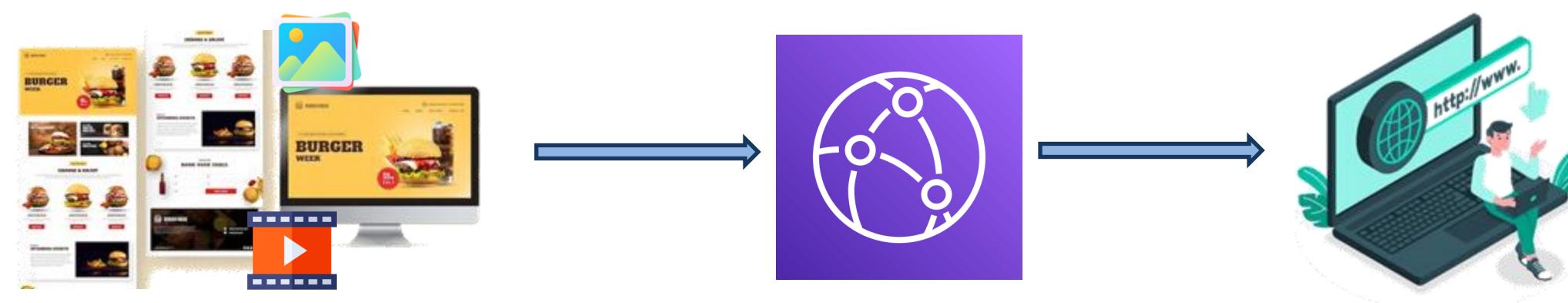
- ✓ One or more data centers within an AWS Region
- ✓ Each AZ has independent power, networking, and connectivity
- ✓ Ensures high availability and fault tolerance
- ✓ Connected with high-speed, low-latency fiber networks
- ✓ Provides scalability and resilience against local disruptions



Regions, Availability Zones & Edge Locations

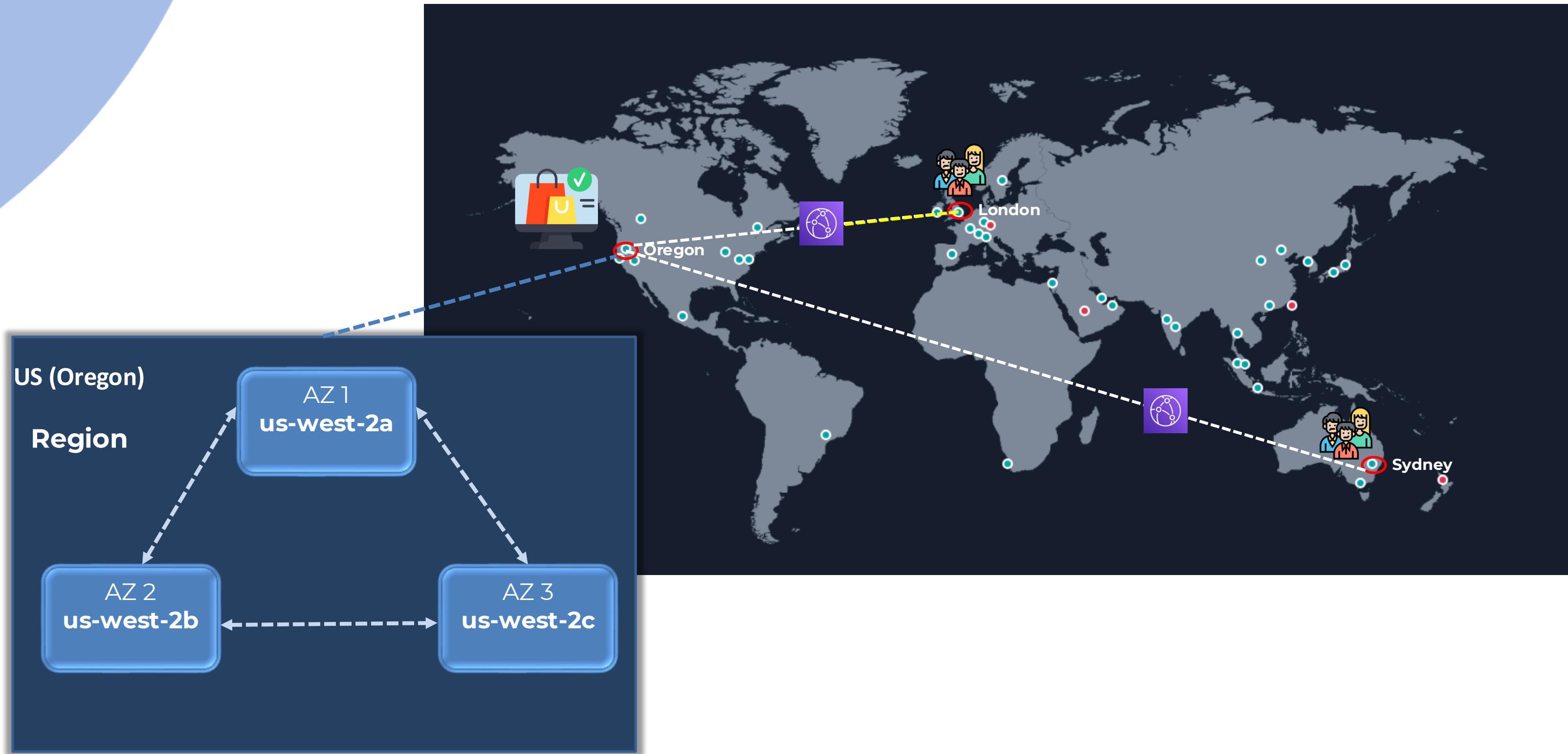
➤ Edge Location

- ✓ Part of AWS CloudFront network
- ✓ Delivers content quickly to users worldwide
- ✓ Caches images, videos, and web pages
- ✓ Brings content closer to users for faster access



Regions, Availability Zones & Edge Locations

➤ Real-World Example



Regions, Availability Zones & Edge Locations

➤ Evaluating Regions

- ✓ **Compliance** – Choose a Region that meets local regulations and data residency requirements
- ✓ **Latency** – Select a Region close to users to minimize network latency and improve performance
- ✓ **Cost** – AWS pricing varies by Region; opting for a lower-cost Region can reduce expenses
- ✓ **Services and Features** – New AWS services are deployed first in larger Regions, impacting availability

Regions, Availability Zones & Edge Locations

➤ In summary

- ✓ AWS Regions host data across multiple Availability Zones for reliability
- ✓ Edge Locations speed up content delivery to users
- ✓ Choosing a Region depends on compliance, latency, cost, and available services

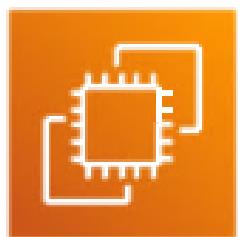
Types of Services in AWS

Types of Services in AWS

Analytics	Game Development
Application Integration	Internet of Things
Blockchain	Machine Learning
Business Applications	Management & Governance
Cloud Financial Management	Media Services
Compute	Migration & Transfer
Containers	Networking & Content Delivery
Customer Enablement	Quantum Technologies
Database	Robotics
Developer Tools	Satellite
End User Computing	Security, Identity, & Compliance
Front-end Web & Mobile	Storage

Types of Services in AWS

➤ Compute services



Amazon EC2

Scalable virtual servers



AWS Lambda

Running code without managing servers



Amazon Elastic Beanstalk

Easy application deployment and management

Which allow you to run applications and workloads on the cloud

Types of Services in AWS

➤ Storage services



Scalable object storage

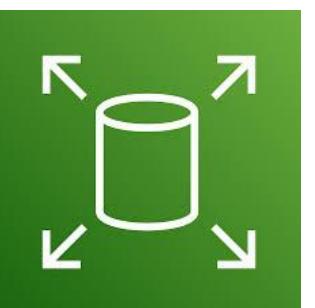
Amazon S3



Scalable file storage for EC2 instances

Designed to store and manage your data

Amazon EFS



High-performance block storage for EC2 instances

Amazon EBS

Types of Services in AWS

➤ Database services



Managed relational databases

Amazon RDS



NoSQL databases

Amazon DynamoDB



In-memory caching

Amazon ElastiCache

Help you store and manage
your data

Types of Services in AWS

➤ Networking services



Amazon VPC

Creating isolated networks



Amazon Route 53

DNS and domain name management



Amazon API Gateway

Creating and managing APIs

Types of Services in AWS

➤ Security & Identity services



AWS IAM

Managing user permissions



AWS KMS

Encrypting data

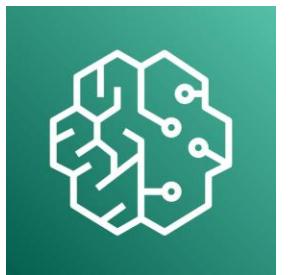


AWS Shield

Protecting against DDoS attacks

Types of Services in AWS

➤ Machine Learning services



Building, training, and deploying models

Amazon SageMaker



Image and video analysis

AWS Rekognition



Natural language processing

AWS Comprehend

Types of Services in AWS

➤ Analytics services



Amazon EMR

Big data processing



Amazon Athena

Querying data in S3 using SQL



AWS Glue

Data integration and ETL (extract, transform, load) tasks

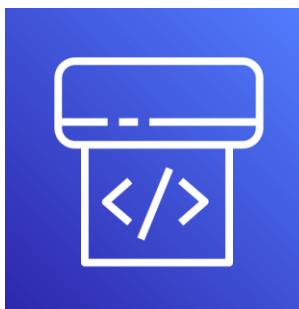
Types of Services in AWS

➤ Developer Tools



AWS CodeDeploy

Automating application deployments



AWS CodePipeline

Automating release processes



AWS Cloud9

Cloud-based development environments

Types of Services in AWS

➤ Containerized services



Running Docker containers at scale

Amazon ECS



Kubernetes-managed containers

Amazon EKS



Storing container images

Amazon ECR

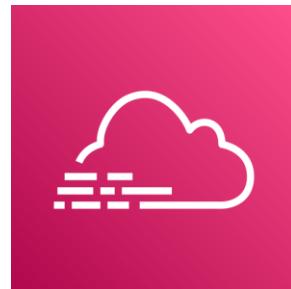
Types of Services in AWS

➤ Management & Governance tools



Amazon CloudWatch

Monitoring resources and applications



AWS CloudTrail

Logging API activity

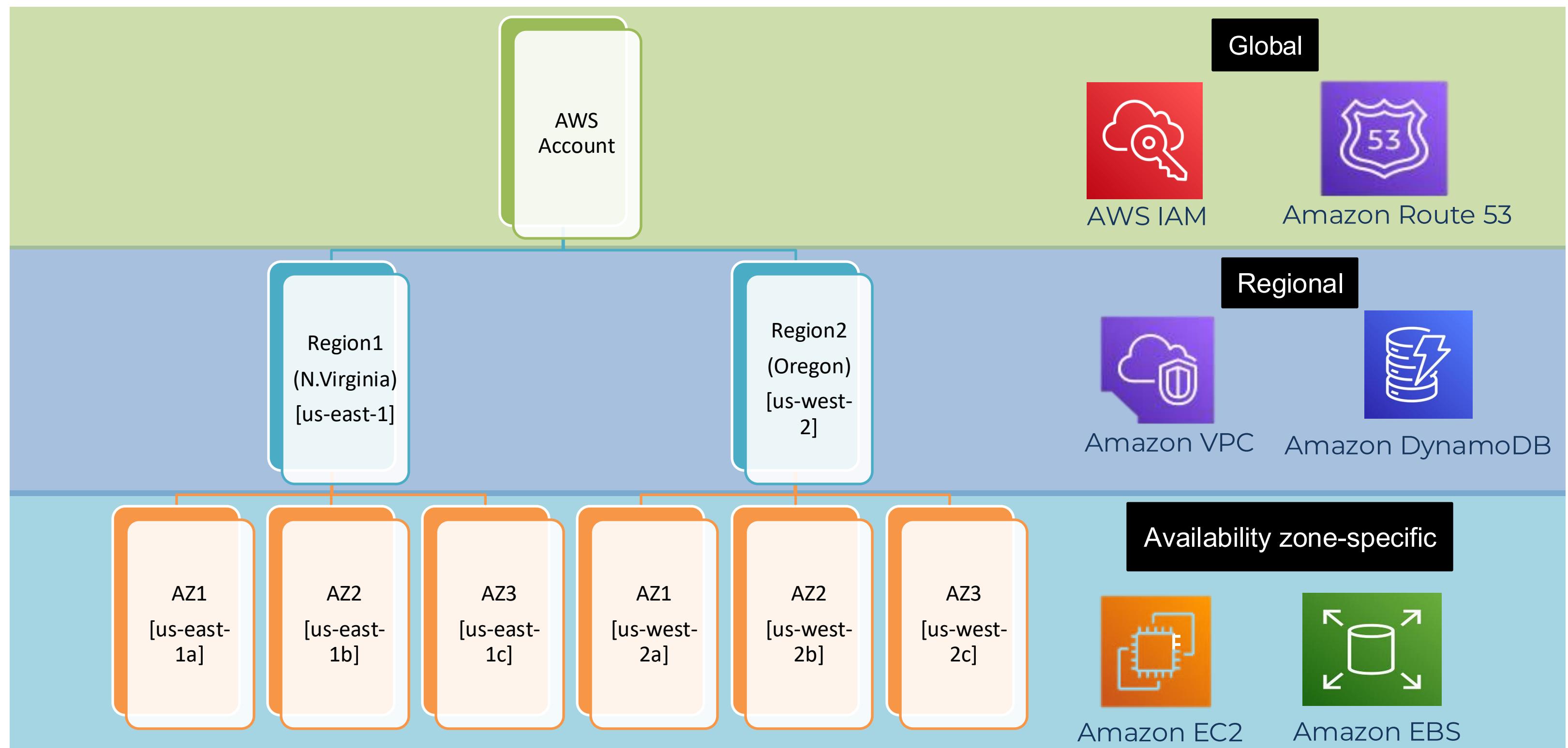


AWS Systems Manager

Operational insights and automation

Types of Services in AWS

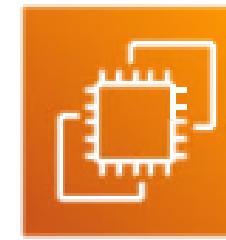
➤ Scope and Accessibility of AWS Services



Types of Services in AWS

➤ Understanding AWS Services Across IaaS, PaaS, and SaaS Models

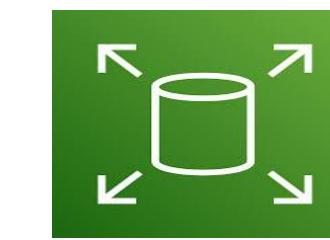
□ Infrastructure as a Service (IaaS)



Amazon EC2



Amazon VPC



Amazon EBS



Amazon S3

- ✓ Flexibility to build and manage your own infrastructure

Types of Services in AWS

➤ Understanding AWS Services Across IaaS, PaaS, and SaaS Models

□ Platform as a Service (PaaS)



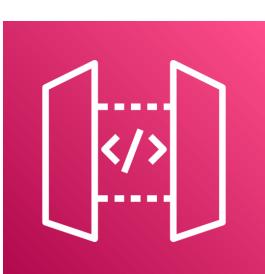
Amazon RDS



Amazon Elastic Beanstalk



Amazon DynamoDB



Amazon API Gateway

- ✓ Deploying and scaling web applications, without having to manage the underlying infrastructure

Types of Services in AWS

➤ Understanding AWS Services Across IaaS, PaaS, and SaaS Models

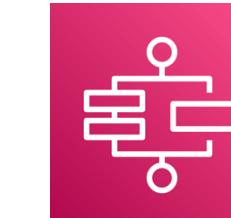
□ Function as a Service (FaaS)



AWS Lambda



AWS Glue



Step Functions



EventBridge

- ✓ Let you run code in response to events—without needing to manage servers

Types of Services in AWS

➤ Understanding AWS Services Across IaaS, PaaS, and SaaS Models

□ Container as a Service (CaaS)



Amazon ECS



Amazon EKS



Amazon App Runner



AWS Lambda

- ✓ Provide managed container orchestration so you can run and scale containerized applications easily

Types of Services in AWS

➤ Understanding AWS Services Across IaaS, PaaS, and SaaS Models

□ Software as a Service (SaaS)



Amazon SageMaker



AWS Rekognition



AWS Textract



AWS WorkSpaces

- ✓ Focus on using the service without worrying about the backend

Types of Services in AWS

➤ Understanding AWS Services Across IaaS, PaaS, and SaaS Models

□ Infrastructure as a Service (IaaS)

- ✓ EC2, VPC, EBS, S3, IAM, Direct Connect, Route 53, Elastic Load Balancing (ELB)

□ Platform as a Service (PaaS)

- ✓ RDS, Elastic Beanstalk, Fargate, DynamoDB, App Runner, API Gateway, Aurora Serverless

□ Function as a Service (FaaS)

- ✓ Lambda, Glue, Step Functions, EventBridge

□ Container as a Service (CaaS)

- ✓ ECS, EKS, App Runner, Lambda (for containers)

□ Software as a Service (SaaS)

- ✓ SageMaker, Rekognition, Textract, WorkSpaces, Chime, QuickSight, Comprehend, Bedrock



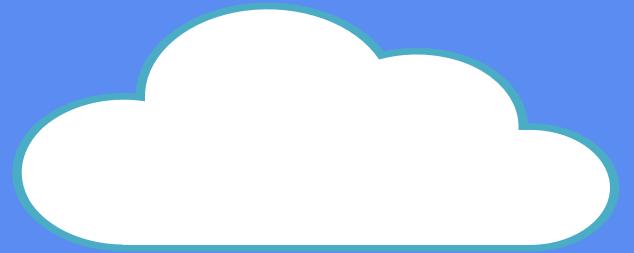
Demonstration | AWS Account Setup & Dashboard Walkthrough

Summary

- Introduction to AWS, its history, and benefits
- **Key concepts:** Regions, Availability Zones, and Edge Locations
- Overview of AWS service categories and types
- Understanding AWS services across IaaS, PaaS, and SaaS models
- Hands-on AWS account setup and dashboard walkthrough

Cloud Fundamentals with AWS

Section 4



IAM (Identity and Access Management)

- ❖ IAM
- ❖ Policies, Groups, Users, and Roles
- ❖ Demo - Creating an IAM policy
- ❖ Demo - Setting up an IAM user

Introduction to IAM

Introduction to IAM

➤ What is IAM?

- ✓ Global AWS service for securely managing AWS resources
- ✓ Helps to securely manage access to your AWS resources
- ✓ Enables user, group, role, and policy management for fine-grained access control
- ✓ Ensures security, flexibility, and least privilege by granting only necessary permissions

Introduction to IAM

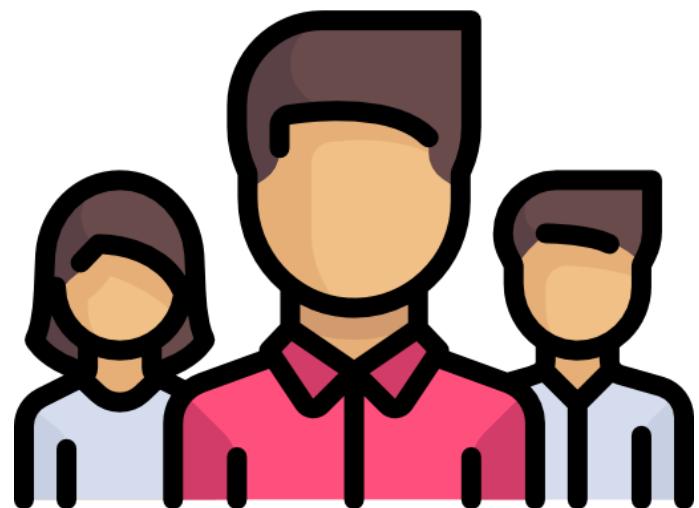
➤ Why Use IAM?

- ✓ Controls authentication and authorization for AWS resource access
- ✓ Grants permissions securely without sharing passwords or access keys
- ✓ Enhances security with Multi-Factor Authentication (MFA) for added protection
- ✓ Manages access at scale with users, groups, roles, and policies for efficient administration

Introduction to IAM

➤ Key IAM Components

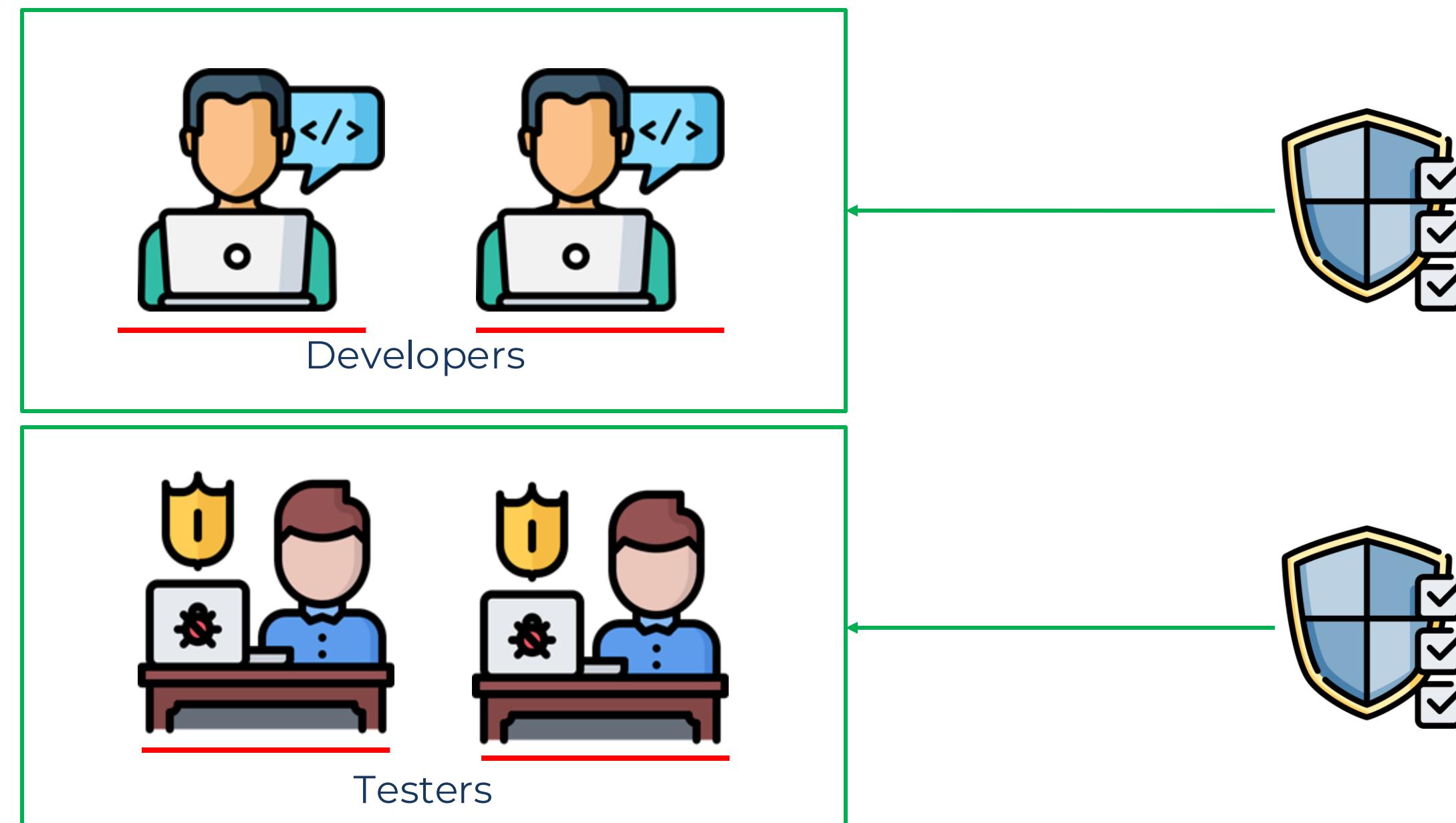
✓ IAM Users



Introduction to IAM

➤ Key IAM Components

✓ IAM Groups



Note: Keep in mind that groups cannot sign in directly, they only serve as a way to manage permissions collectively

Introduction to IAM

➤ Key IAM Components

- ✓ **IAM Policies**
- ✓ Policies define who can do what in AWS
- ✓ JSON-based documents defining allowed or denied actions on AWS resources
- ✓ Policies attach to users, groups, or roles for fine-grained access control

Policy Type	Description	Modifiable	Scope	Best For
AWS Managed Policies	Pre-defined by AWS, updated periodically for new services and best practices	✗ No	Multiple identities	Quick setup with AWS-recommended permissions
Customer Managed Policies	Created by users for customized access control	✓ Yes	Multiple identities	Custom permissions with full control
Inline Policies	Directly attached to a single IAM identity, deleted with the identity	✓ Yes	Single identity (User, Group, or Role)	One-off, specific use cases

Introduction to IAM

➤ Key IAM Components

- ✓ **IAM Roles**
- ✓ Temporary credentials which are not associated with a specific user
- ✓ Allows to assume permissions without exposing long-term credentials

Introduction to IAM

➤ Understanding IAM Policy Structure

- ✓ IAM policies are JSON documents defining permissions
- ✓ Controls who can perform what actions on which resources, and under what conditions
- ✓ AWS supports seven types of policies
 - Identity-based Policies
 - Resource-based Policies
 - Permissions Boundaries
 - Service Control Policies (SCPs)
 - Resource Control Policies (RCPs)
 - Access Control Lists (ACLs)
 - Session Policies

Introduction to IAM

➤ Understanding IAM Policy Structure

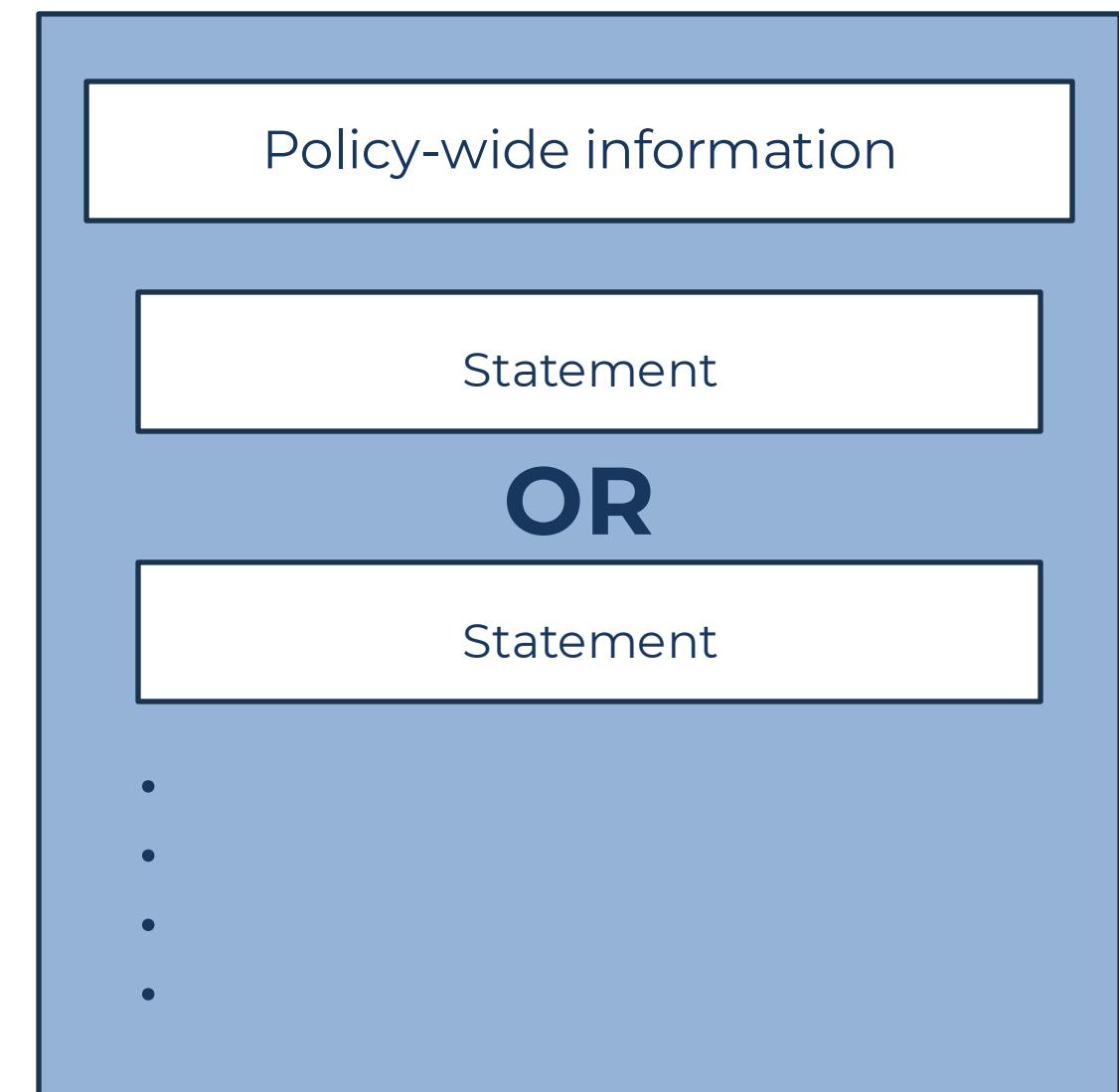
Identity-Based Policies	Resource-Based Policies
Attached to IAM identities (users, roles, or groups)	Attached directly to a resource
Define allowed actions on resources	Specify which principals (users, roles, or accounts) can access the resource

Introduction to IAM

➤ Structure of an IAM Policy

- ✓ Each IAM policy consists of:

1. Policy-wide information (like the version number)
2. One or more statements, each defining a specific permission



Introduction to IAM

➤ Example - Policy Breakdown

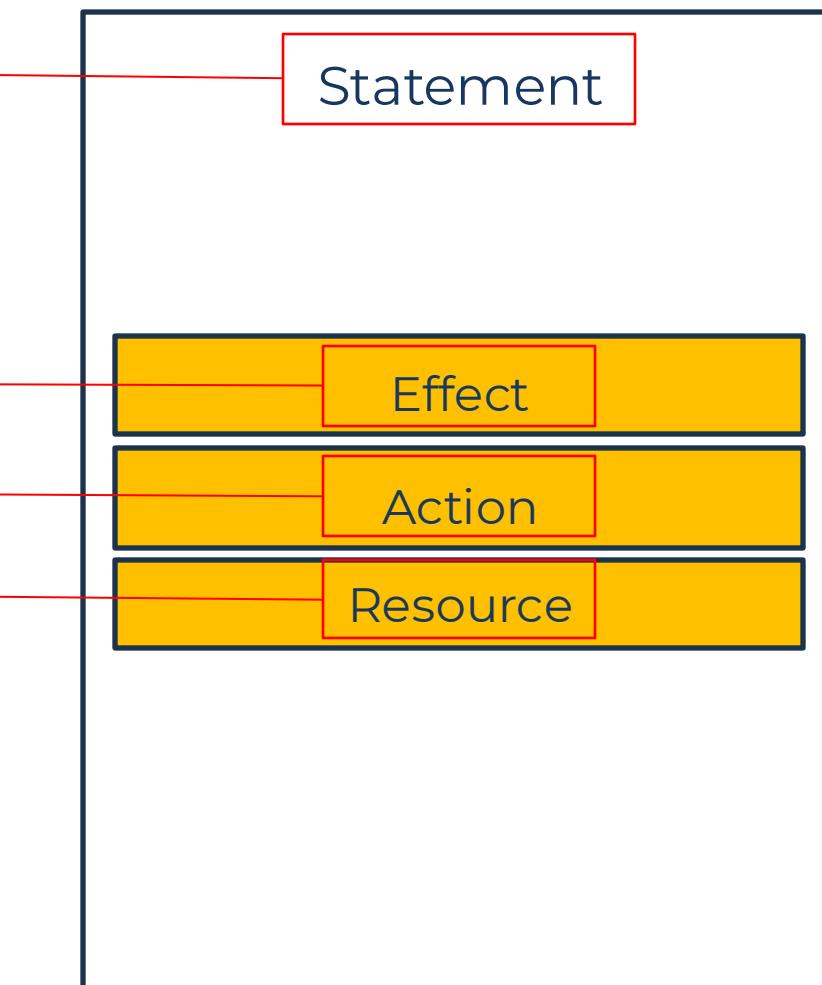
Administrator Access Policy

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

"Version": "2012-10-17", → IAM policy language version
 "Statement": → Container where permissions are defined

"Effect": "Allow", → Specifies whether the policy grants or denies permissions
 "Action": "*", → Defines the actions allowed
 "Resource": "*" → Specifies which AWS services or denied on AWS services resources the policy applies to

IAM Policy Syntax



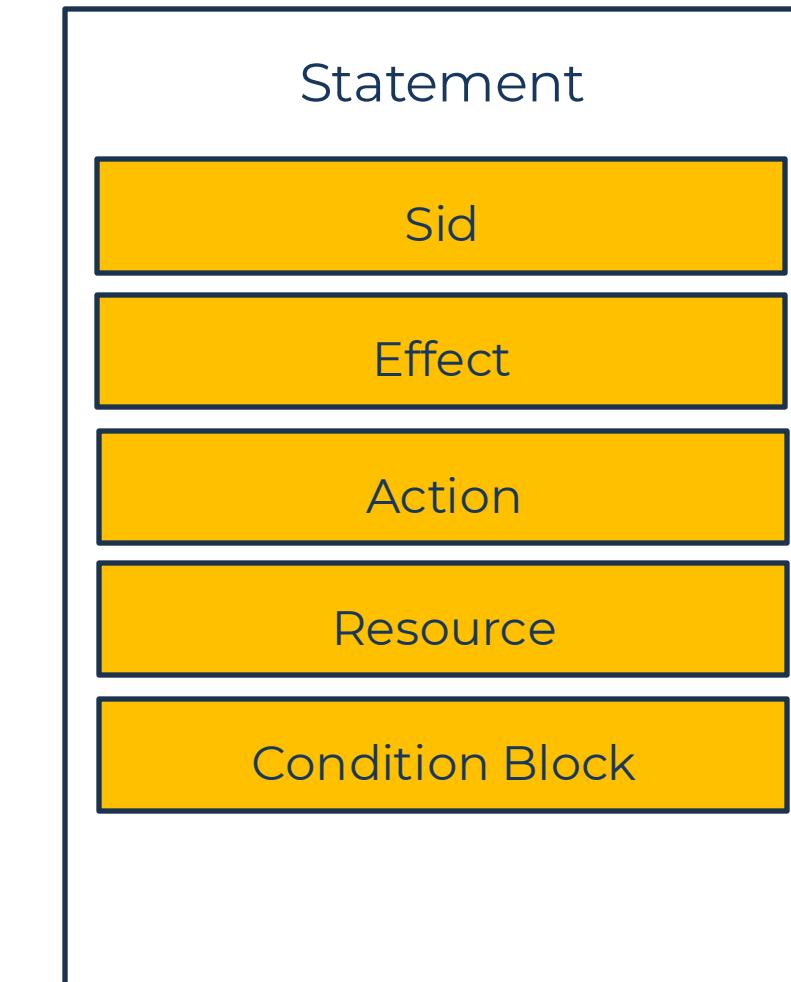
Introduction to IAM

Version

Identity-based Policy

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": "ec2:*",
            "Resource": "*",
            "Condition": {
                "ForAnyValue:StringEqualsIfExists": {
                    "aws:RequestedRegion": "us-east-2"
                }
            }
        },
        {
            "Sid": "VisualEditor1",
            "Effect": "Allow",
            "Action": "rds:*",
            "Resource": "*"
        }
    ]
}
```

IAM Policy Syntax



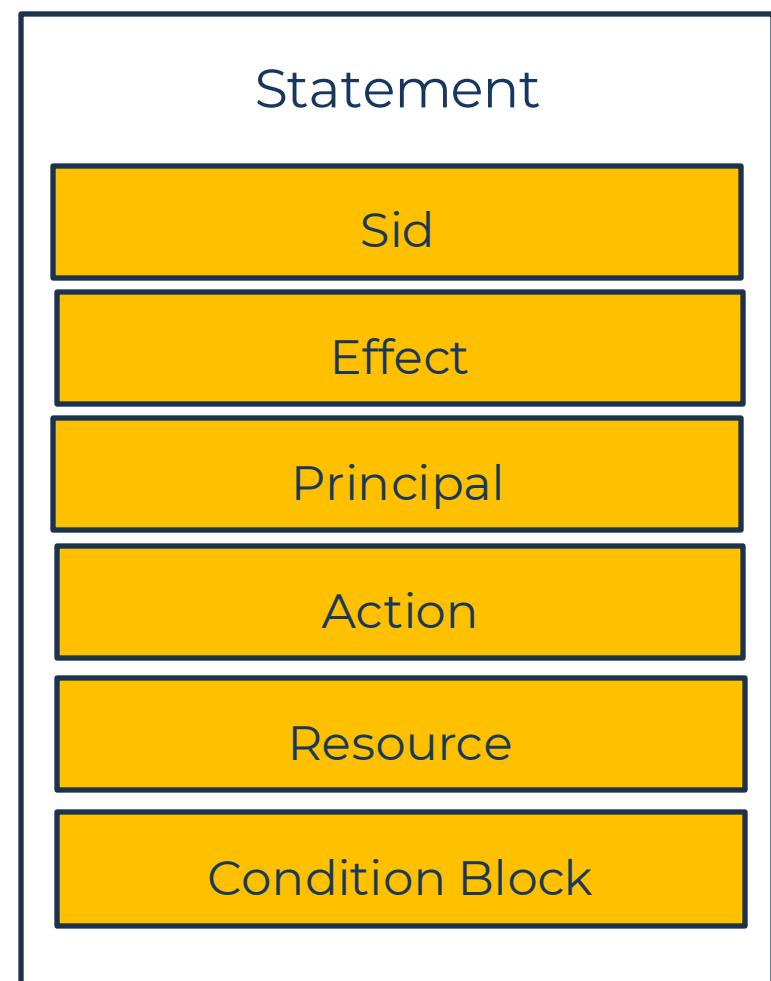
Conditions

- ✓ IP Address Restrictions – Imposes restrictions based on IP addresses using aws:SourceIp
- ✓ Region-Based Access – Restricts access to specific AWS regions using aws:RequestedRegion

Introduction to IAM

Resource-based Policy

IAM Policy Syntax



Introduction to IAM

➤ Visual Editor

Instead of manually writing JSON, AWS provides a Visual Editor to simplify policy creation:

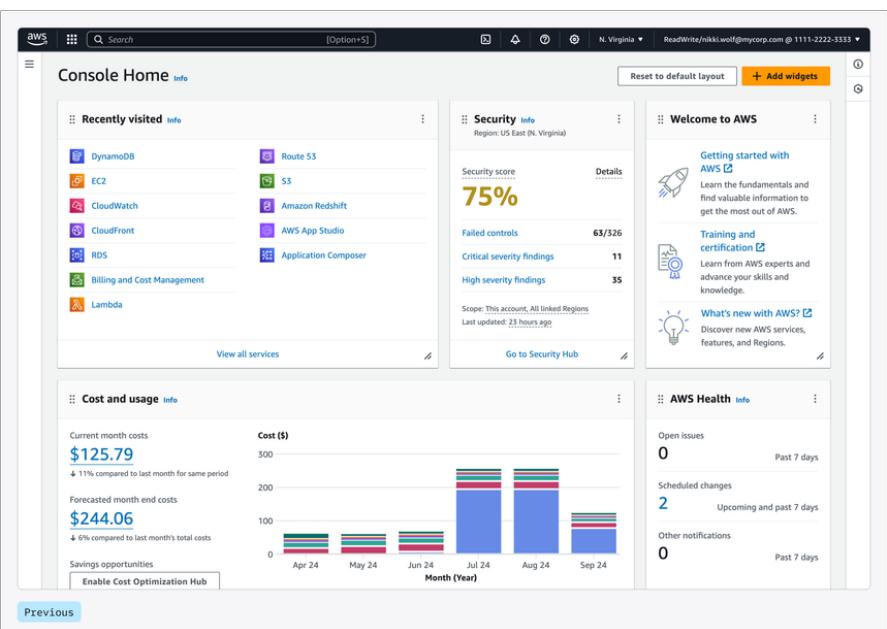
- ✓ Choose a service: **S3** or **EC2**
- ✓ Select the actions: **Read**, **Write**, or **Delete**
- ✓ Specify resources using **ARNs**
- ✓ Add the conditions: **IP** or **Region-based Access**



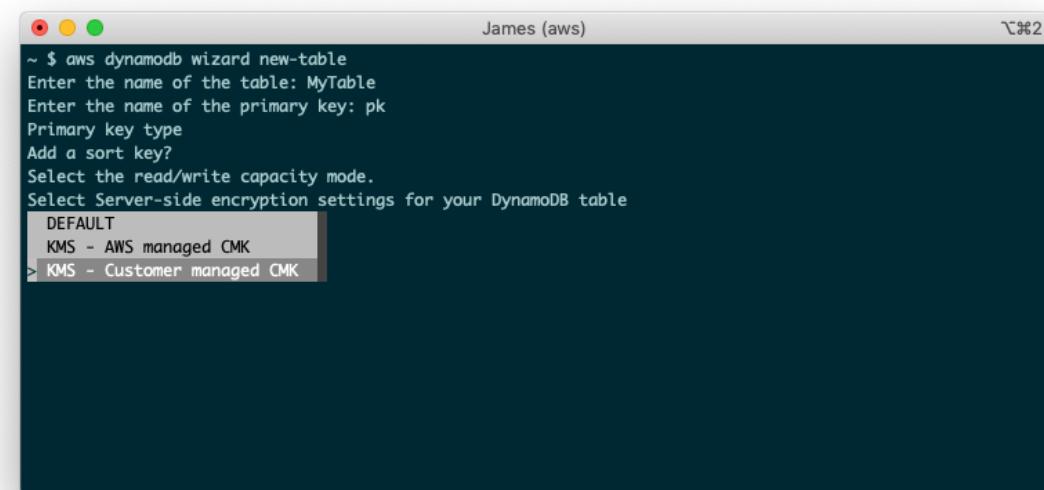
Introduction to IAM

➤ Managing IAM Efficiently

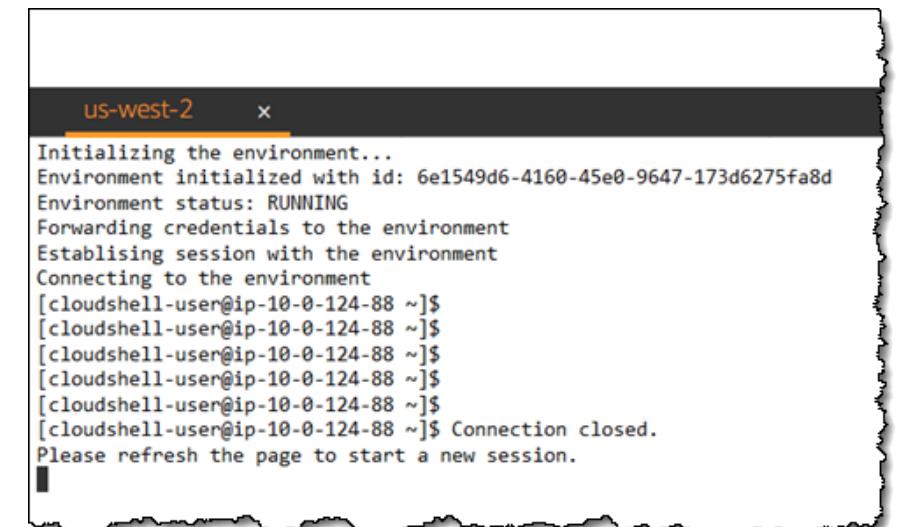
✓ AWS Management Console



✓ AWS CLI & AWS Tools for Windows PowerShell



✓ AWS CloudShell



✓ AWS SDKs



✓ IAM Query API



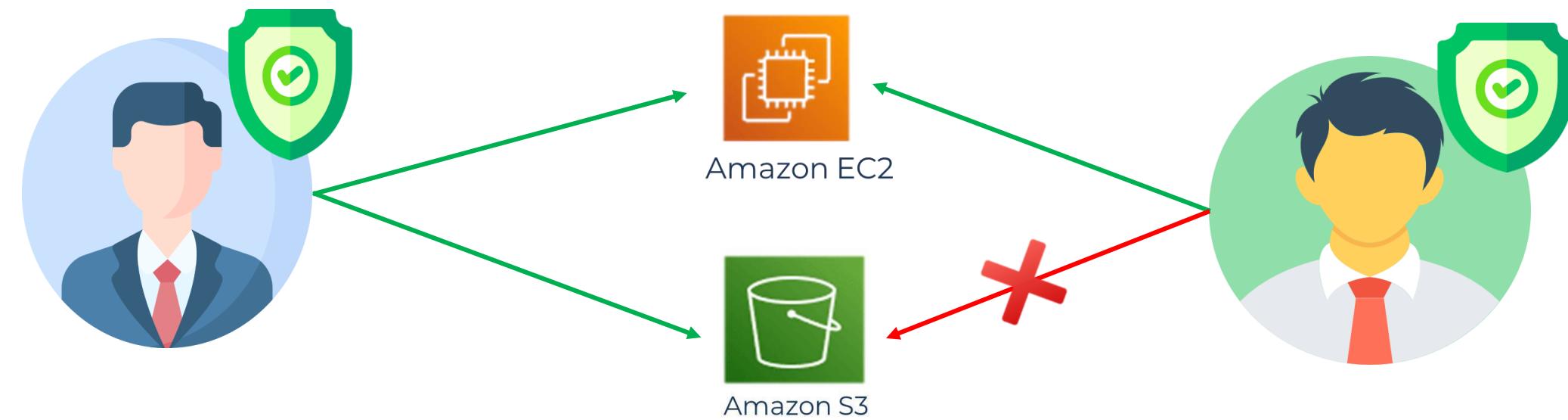
Introduction to IAM

➤ Understanding IAM Through an Analogy



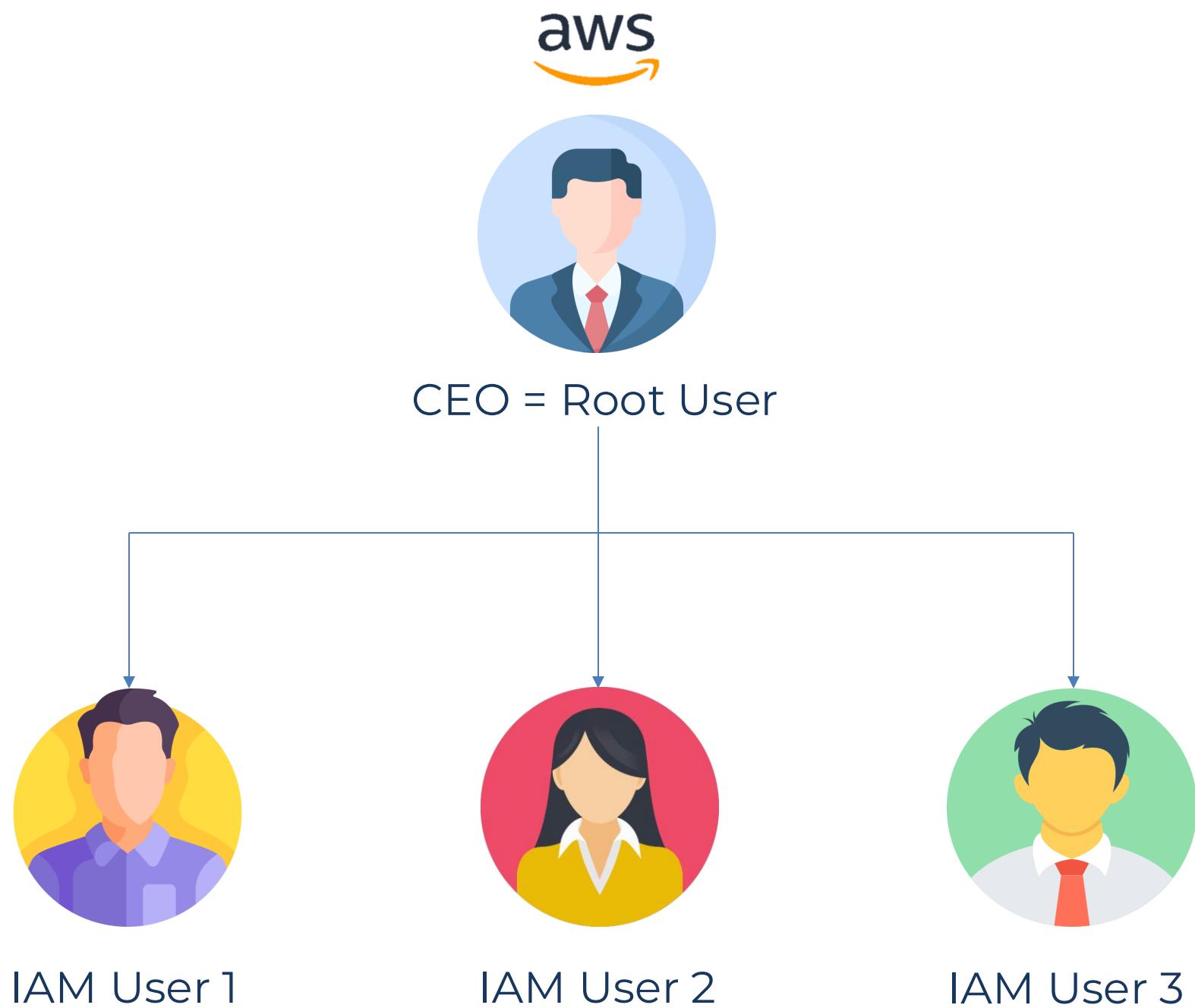
Introduction to IAM

➤ Understanding IAM Through an Analogy



Introduction to IAM

➤ Understanding IAM Through an Analogy



Introduction to IAM

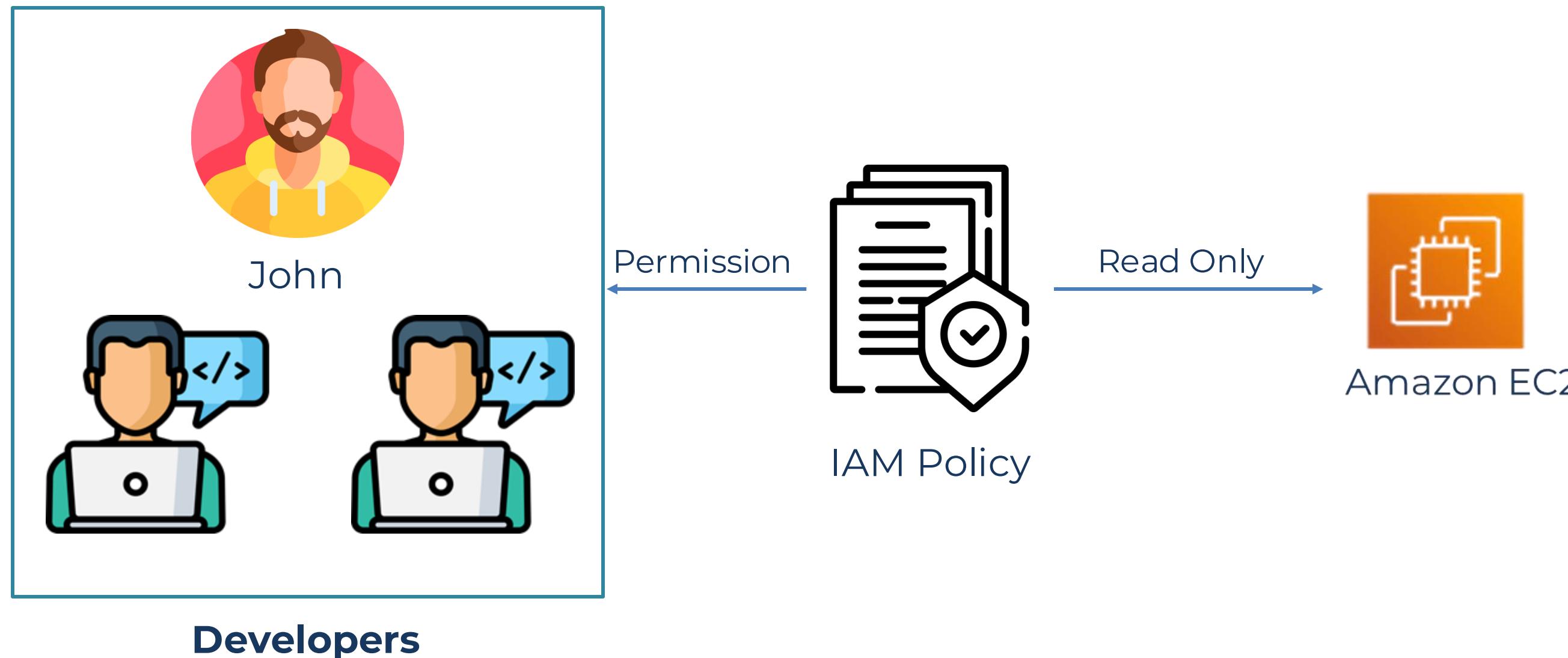
➤ IAM Users and Policies

- ✓ IAM users need explicit permissions to act
- ✓ Permissions are defined using IAM Policies - JSON-based documents



Introduction to IAM

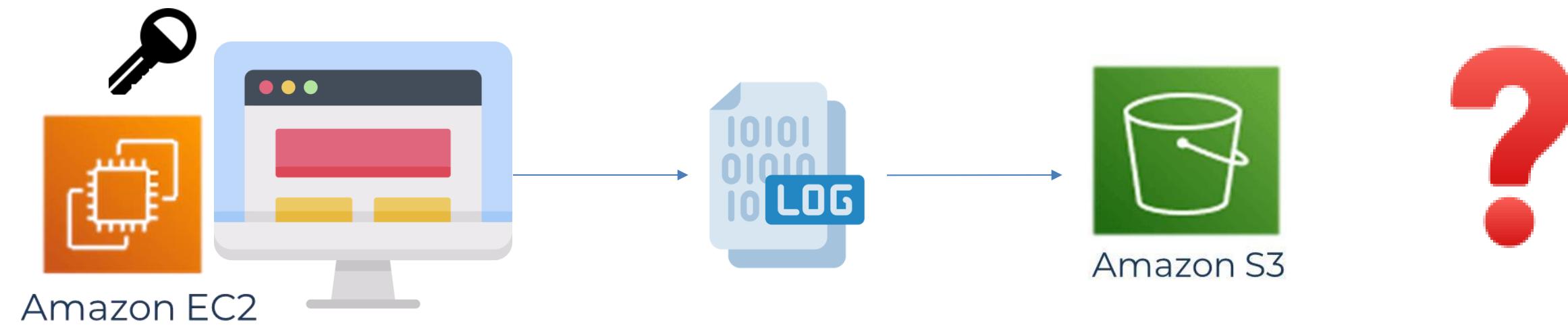
➤ IAM Groups – Efficient Access Management



- ✓ Simplified management
- ✓ Scalability

Introduction to IAM

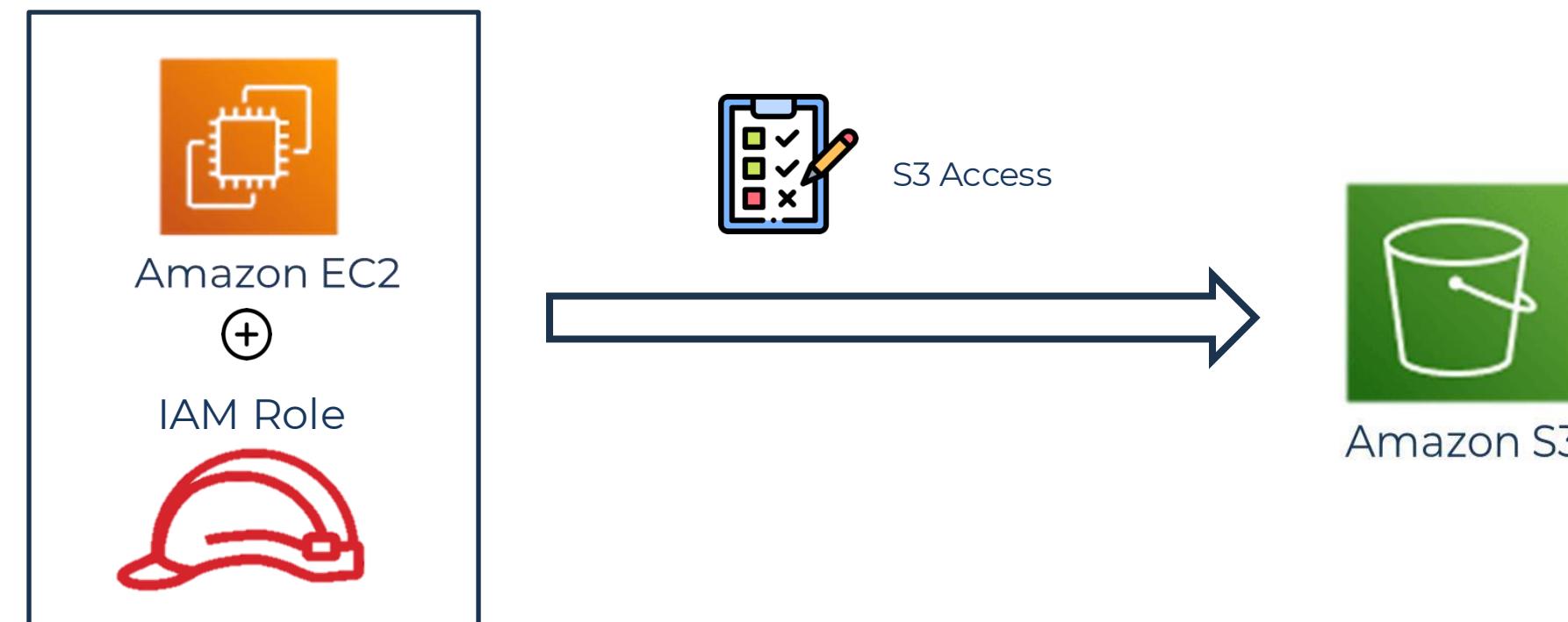
➤ IAM Roles – Secure Access for AWS Services



Security X
AWS best practices X
Storing Credentials X

Introduction to IAM

➤ IAM Roles – Secure Access for AWS Services



- ✓ IAM Roles are widely used in AWS for:
 - Granting EC2 access to S3, RDS, or DynamoDB
 - Allowing Lambda functions to interact with databases or APIs
 - Enabling Kubernetes (EKS) workloads to access AWS services securely
- ✓ IAM Roles do not require passwords or access keys



Demonstration | Creating Policy based on Sample Application



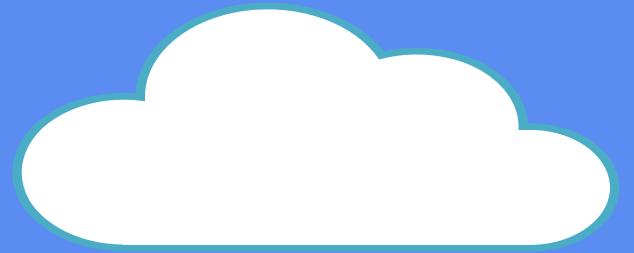
Demonstration | Create New IAM User and Login

Summary

- IAM
 - Policies
 - Groups
 - Users
 - Roles
 - How they work together to manage access in AWS
- **Demo 1:** Created a custom IAM policy to grant access to EC2 resources
- **Demo 2:** Attached the policy to an IAM user

Cloud Fundamentals with AWS

Section 5



EC2 (Elastic Compute Cloud)

- ❖ EC2
- ❖ Why EC2 is important & its Key Features
- ❖ EC2 pricing models
- ❖ Launch templates
- ❖ **Demo 1** - Create and access Linux and Windows EC2 instances
- ❖ **Demo 2** - Deploy a sample application on a Linux instance
- ❖ **Demo 3** - Create an AMI and launch instances from it
- ❖ **Demo 4** - Set up a launch template

Introduction to EC2

Introduction to EC2

➤ Advantages of On-Premises Infrastructure

- ✓ Full control over hardware
- ✓ Predictable performance

Introduction to EC2

➤ Challenges of On-Premises Infrastructure

1. Physical Infrastructure

- ✓ Requires racks, cooling systems, and power management

2. High Upfront Investment & Deployment Delays

- ✓ Significant initial costs
- ✓ Deploying new servers can take weeks or even months

3. Security and Maintenance

- ✓ Demand dedicated IT teams

4. Scaling Limitations

- ✓ Requires purchasing and installing additional hardware
- ✓ Scaling isn't always efficient

Introduction to EC2

➤ Challenges of On-Premises Infrastructure

5. High Availability and Disaster Recovery

- ✓ Entirely falls on the organization
- ✓ Adds complexity and cost

6. Limited Agility

- ✓ Difficult to adapt quickly to changing business needs
- ✓ Launching new applications, expanding to new regions, or responding to traffic spikes requires lengthy procurement cycles and manual configurations
- ✓ Limits innovation and slows time-to-market

Introduction to EC2

➤ Amazon EC2: A Smarter Alternative

- ✓ Elastic Compute Cloud

□ Fast Deployment

- ✓ Launch virtual servers in minutes
- ✓ Eliminates costly hardware investments
- ✓ Removes long provisioning times

Introduction to EC2

➤ Amazon EC2: A Smarter Alternative

□ Flexible Instance Types

- ✓ Wide range of instance types
- ✓ Organizations can quickly adapt to different workloads
 - Compute-intensive tasks
 - Memory-heavy applications
 - General-purpose computing

Introduction to EC2

➤ Amazon EC2: A Smarter Alternative

□ Scalability with Auto Scaling

- ✓ Automatically adjusts the number of EC2 instances based on demand
- ✓ Keeps applications responsive during peak usage
- ✓ Reduces costs when traffic decreases
- ✓ Scale instantly and dynamically—no waiting weeks for new hardware

Introduction to EC2

➤ Amazon EC2: A Smarter Alternative

□ High Availability with ELB and Availability Zones

- ✓ Elastic Load Balancing (ELB) distributes traffic across multiple instances
- ✓ Prevents any single point of failure
- ✓ AWS operates across multiple availability zones
- ✓ Ensures seamless failover in case of infrastructure issues

Introduction to EC2

➤ Amazon EC2: A Smarter Alternative

□ Flexible and Resilient Storage Options

- ✓ Amazon EBS for persistent storage
- ✓ Amazon S3 for scalable object storage
- ✓ Amazon RDS for managed databases

Introduction to EC2

➤ Amazon EC2: A Smarter Alternative

□ Enhanced Security

- ✓ IAM roles, security groups, and network ACLs
- ✓ Precise access control and network protection
- ✓ Minimizes risks associated with traditional data centers

Introduction to EC2

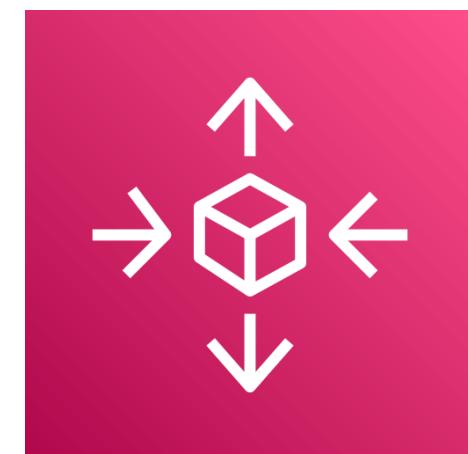
➤ Amazon EC2: A Smarter Alternative

□ Agility and Innovation

- ✓ Launch new products
- ✓ Respond to market shifts
- ✓ Expand globally
- ✓ Move faster, innovate sooner, and scale effortlessly



Load Balancing



Auto Scaling

Introduction to EC2

➤ What is EC2?

- ✓ **Amazon Elastic Compute Cloud** provides on-demand, scalable computing capacity



EC2

- ✓ Giving you the flexibility to choose and configure the operating system, software, and applications based on your needs

Introduction to EC2

➤ Features of EC2

- ✓ Amazon Machine Images
- ✓ Instance types
- ✓ Amazon EBS volumes
- ✓ Key pairs
- ✓ Security groups
- **Configure**
- **Secure**
- **Optimize**

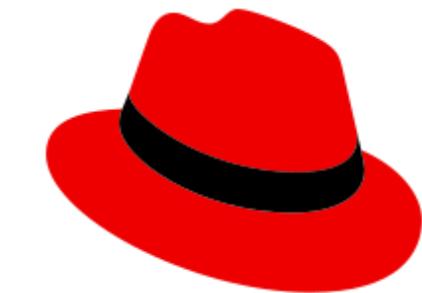
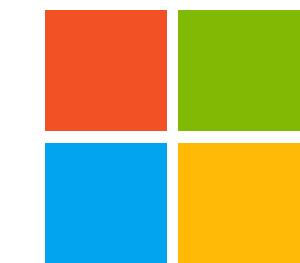
Introduction to EC2

➤ Amazon Machine Image

- ✓ Just like a physical server needs an OS, an EC2 instance requires a preconfigured template to run
- ✓ AMIs provide the OS, software, and settings to launch instances quickly and consistently

Introduction to EC2

- Amazon Machine Image
 - AWS offers a diverse range of AMIs



Introduction to EC2

➤ Amazon Machine Image

□ AWS offers a diverse range of AMIs

- ✓ **Amazon Linux** – Developed and optimized by **AWS** for performance and security
- ✓ **Ubuntu** – Maintained by **Canonical**, offering frequent updates and long-term support (LTS) versions
- ✓ **Windows Server** – Provided directly by **Microsoft**, with various editions tailored for enterprise applications
- ✓ **MacOS** – Available for Apple-based workloads, hosted on **dedicated Mac** instances
- ✓ **Red Hat Enterprise Linux (RHEL)** – Managed by **Red Hat**, designed for enterprise environments with full support and security updates
- ✓ **SUSE Linux Enterprise Server (SLES)** – Developed by **SUSE**, known for stability and optimized for cloud and enterprise workloads

Introduction to EC2

➤ Advantages of AMI

□ Region-Specific but Transferable

- ✓ **AMIs are Region-Specific:** Each AMI is tied to a specific AWS Region
- ✓ **Not Restricted:** You can copy an AMI to another AWS Region when needed
- ✓ **Seamless Replication:** Easily replicate environments across multiple locations
- ✓ **Efficiency:** No need to start from scratch when deploying in a new region

Introduction to EC2

➤ Advantages of AMI

□ Preconfigured for Quick Deployment

- ✓ **AWS Default AMIs:** Prebuilt images with popular OS and essential software
- ✓ **Instant Instance Launch:** No need for manual installation

Introduction to EC2

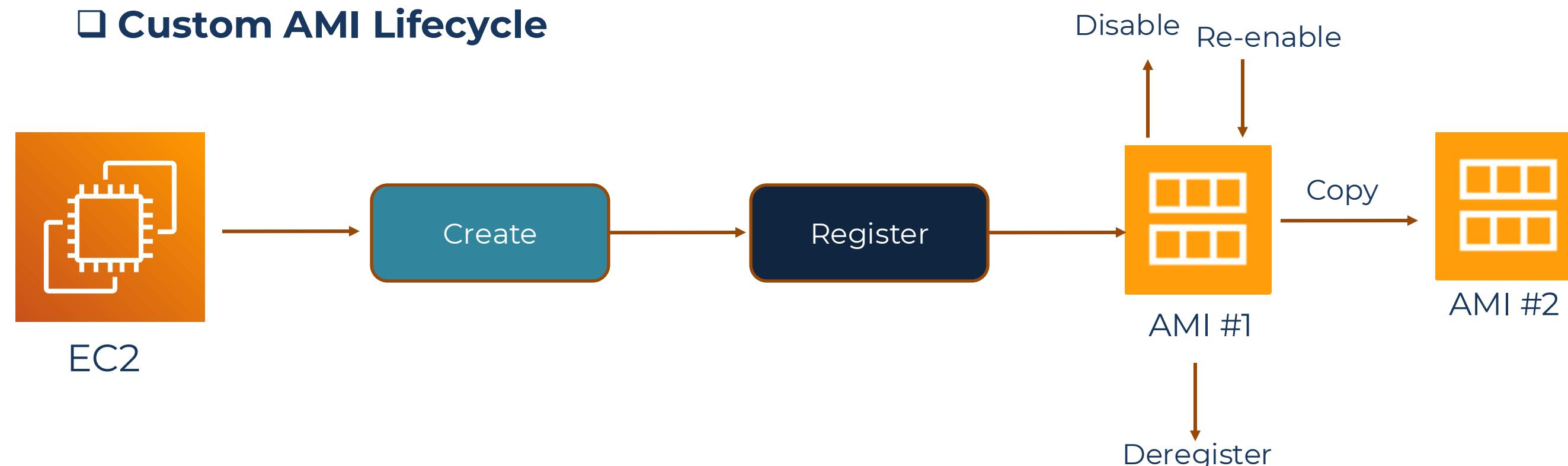
➤ Advantages of AMI

□ Custom AMIs for Efficiency

Why Use Custom AMIs?

- ✓ Pre-install applications, settings, and security configurations.
- ✓ Ensure quick and consistent deployment of multiple instances.
- ✓ Saves time and effort while maintaining uniform configurations.

□ Custom AMI Lifecycle



Introduction to EC2

➤ Advantages of AMI

□ Easily Scale with Multiple Instances

- ✓ **Consistent Deployments:** Use a single AMI to launch multiple identical instances
- ✓ **No Manual Configuration:** Pre-configured AMI ensures uniform setup across all servers
- ✓ **Fast Deployment:** Just launch and go—no repetitive setup required

Introduction to EC2

➤ Advantages of AMI

□ Share or Sell Your AMI

- ✓ **Sharing Options:** Grant access to specific AWS accounts
- ✓ **AWS Marketplace:** List your AMI for sale to a broader audience
- ✓ **Benefits to Others:** Help users save time with your pre-configured setup

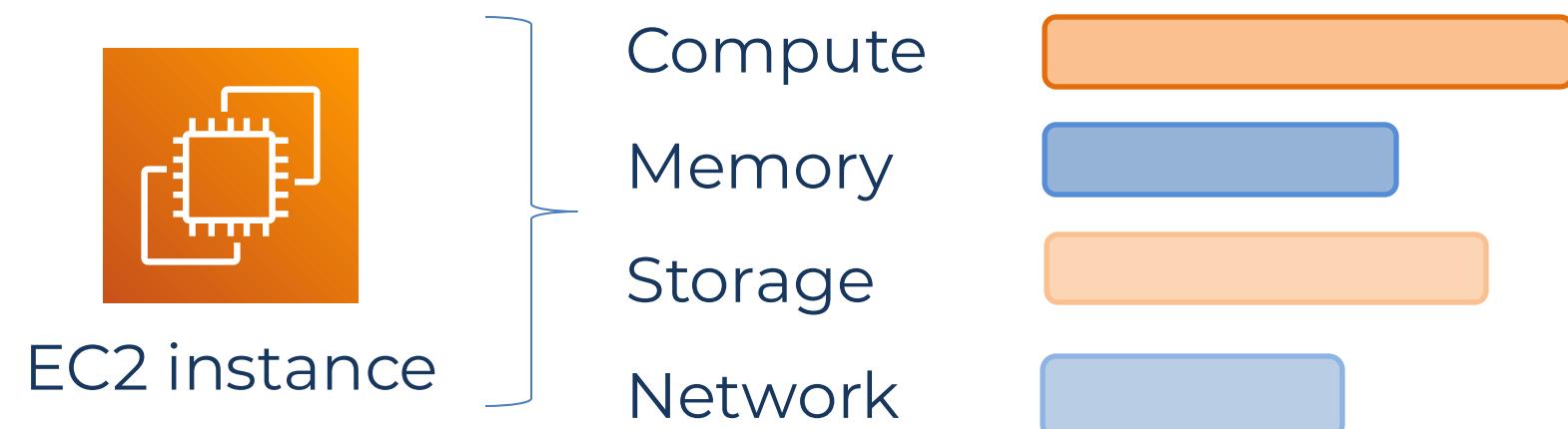
Introduction to EC2

➤ Amazon Machine Image

- Quick Start AMIs
- My AMIs
- AWS Marketplace AMIs
- Community AMIs

Introduction to EC2

➤ Instance Types



Introduction to EC2

➤ Instance Types

□ EC2 instances are divided into six primary categories

- ✓ **General Purpose** – Balanced compute, memory, and networking, ideal for web servers and code repositories
- ✓ **Compute Optimized** – High-performance processors for tasks like batch processing, media transcoding, and machine learning inference
- ✓ **Memory Optimized** – Designed for large in-memory workloads like real-time big data analytics
- ✓ **Storage Optimized** – Ideal for applications requiring high-speed access to large datasets, such as databases and big data processing
- ✓ **Accelerated Computing** – Uses GPUs or specialized hardware for AI, graphics processing, and complex calculations
- ✓ **High-Performance Computing (HPC)** – Optimized for running large-scale simulations and deep learning models

Introduction to EC2

➤ Instance Types

PAGE CONTENT

General Purpose

General Purpose

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

Compute Optimized

M8g	M7g	M7i	M7i-flex	M7a	Mac	M6g	M6i	M6in	M6a	M5	M5n	M5zn
M5a	M4	T4g	T3	T3a	T2							

Memory Optimized

[Amazon EC2 M8g instances](#) are powered by AWS Graviton4 processors. They deliver the best price performance in Amazon EC2 for general purpose workloads.

Accelerated Computing

Features:

- Powered by custom-built AWS Graviton4 processors
- Larger instance sizes with up to 3x more vCPUs and memory than M7g instances
- Features the latest DDR5-5600 memory
- Optimized for Amazon EBS by default
- Supports [Elastic Fabric Adapter \(EFA\)](#) on m8g.24xlarge, m8g.48xlarge, m8g.metal-24xl, and m8g.metal-48xl
- Powered by the [AWS Nitro System](#), a combination of dedicated hardware and lightweight hypervisor

Storage Optimized

HPC Optimized

Instance Features

Measuring Instance Performance

Instance size	vCPU	Memory (GiB)	Instance storage (GB)	Network bandwidth (Gbps)	Amazon EBS bandwidth (Gbps)
m8g.medium	1	4	EBS-only	Up to 12.5	Up to 10

You can visit: aws.amazon.com/ec2/instance-types

Introduction to EC2

➤ Understanding Amazon EC2 Instance Naming Convention



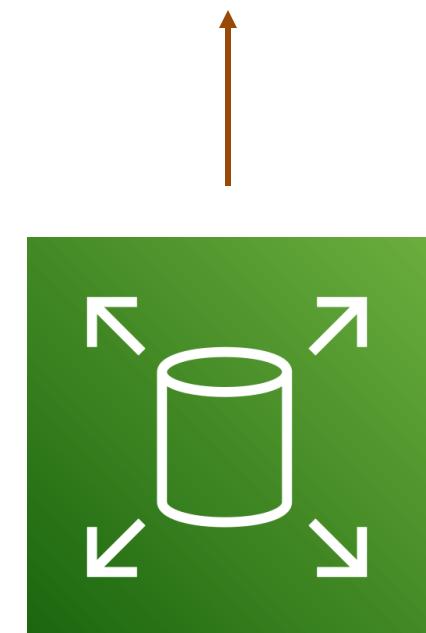
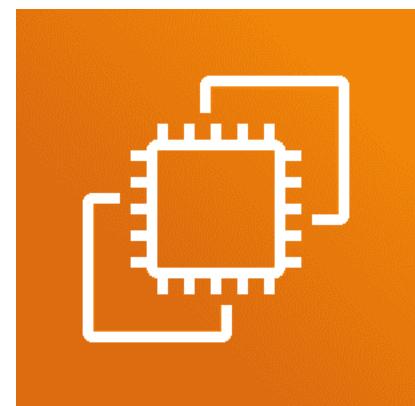
Introduction to EC2

➤ Amazon EBS volumes

- ✓ AWS automatically creates a **root volume** that contains the **boot image**
- ✓ One root volume
- ✓ Add more storage anytime
- ✓ Even after launch
- ✓ Without stopping the instance

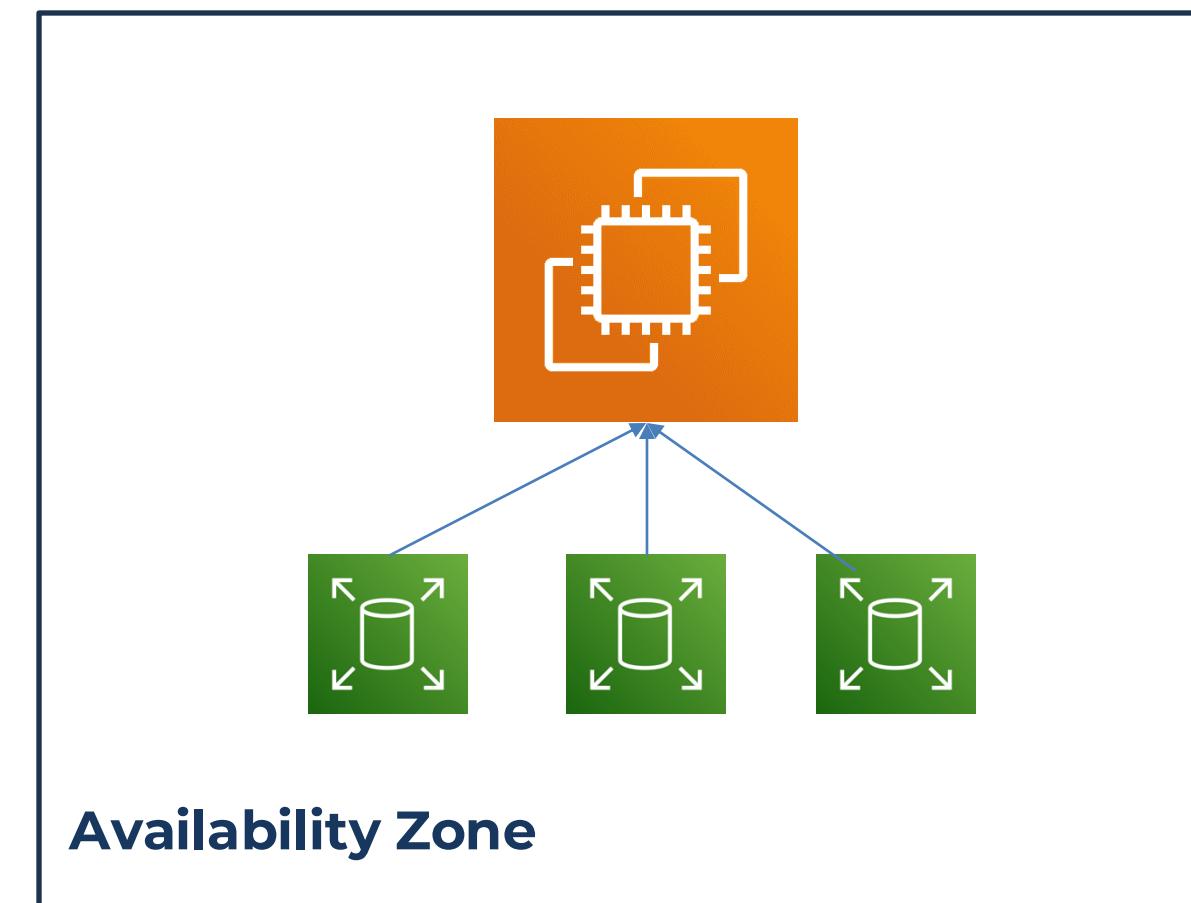
Introduction to EC2

➤ Amazon EBS volumes



Introduction to EC2

➤ Amazon EBS volumes



Some volume and instance types even support Multi-Attach

Introduction to EC2

➤ Amazon EBS volumes

- **AWS recommends using EBS-backed AMIs**
 - ✓ Faster launch times and persistent storage
 - ✓ EBS volumes are flexible
 - You can resize or increase performance
 - You can create EBS snapshots to back up data or move it

Accounts

Regions

Availability
Zones

Introduction to EC2

➤ Amazon EBS volumes

- ✓ By default, EBS root volumes are deleted when the instance is terminated

DeleteOnTermination = False

Introduction to EC2

➤ EBS Volume Types

- **SSD-backed volumes**
- **HDD-backed volumes**

Introduction to EC2

➤ EBS Volume Types

□ SSD-backed volumes

- General Purpose SSD (**gp3, gp2**)
- Provisioned IOPS SSD (**io1, io2**)

□ HDD-backed volumes

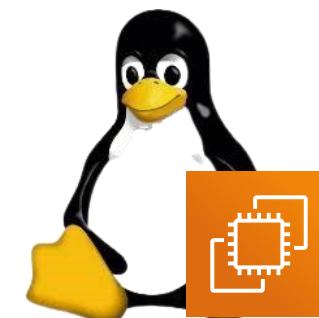
- Throughput Optimized HDD (**st1**)
- Cold HDD (**sc1**)

Only gp2/gp3 and io1/io2 can be used as boot volumes

Introduction to EC2

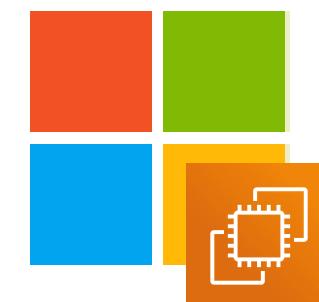
➤ Key Pairs

- ✓ **Public Key** – Stored on your instance by AWS
- ✓ **Private Key** – Stored securely on your computer and used for authentication



- ✓ Private key allows you to securely SSH

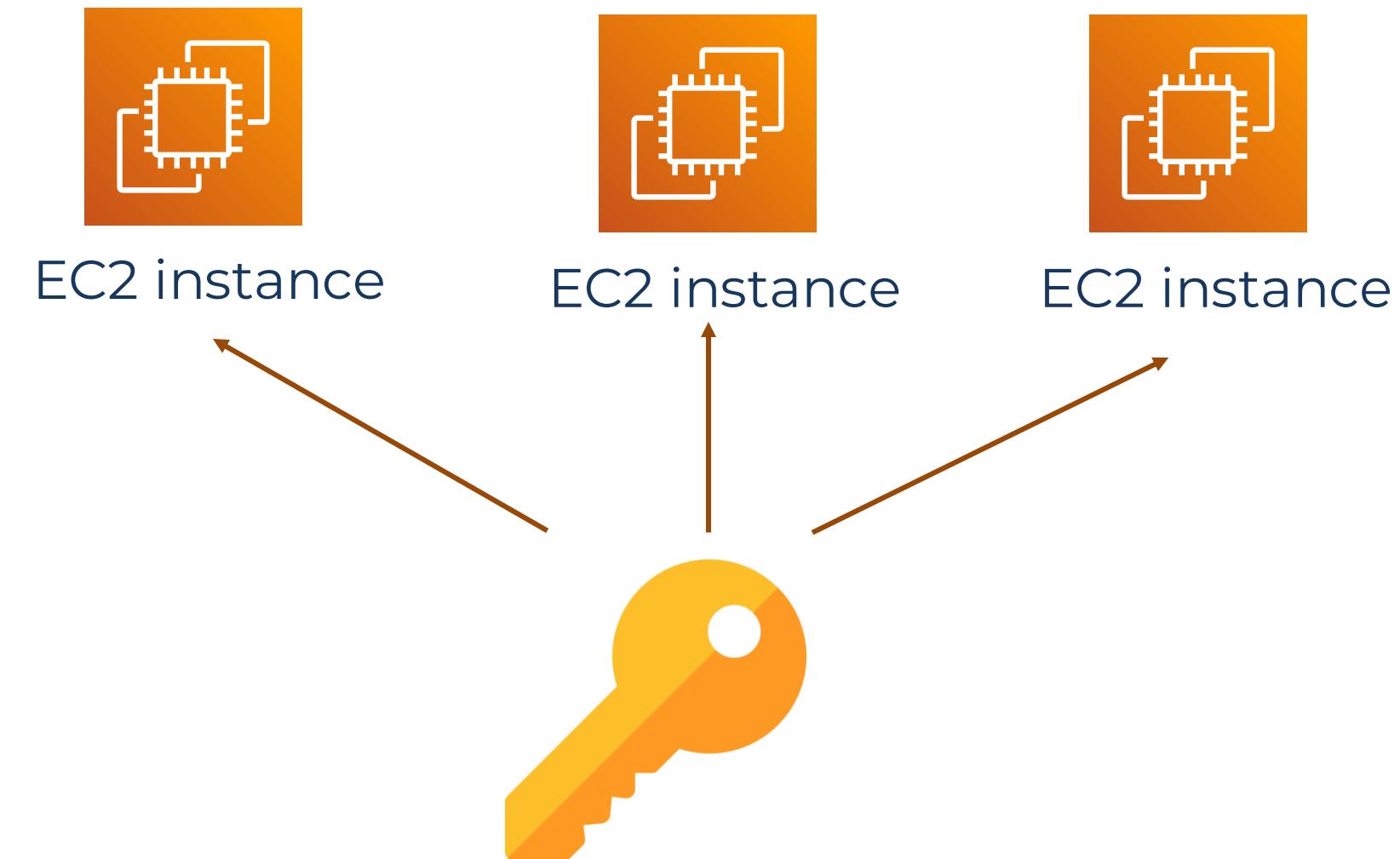
`~/.ssh/authorized_keys`



- ✓ Private key is required to decrypt the administrator password

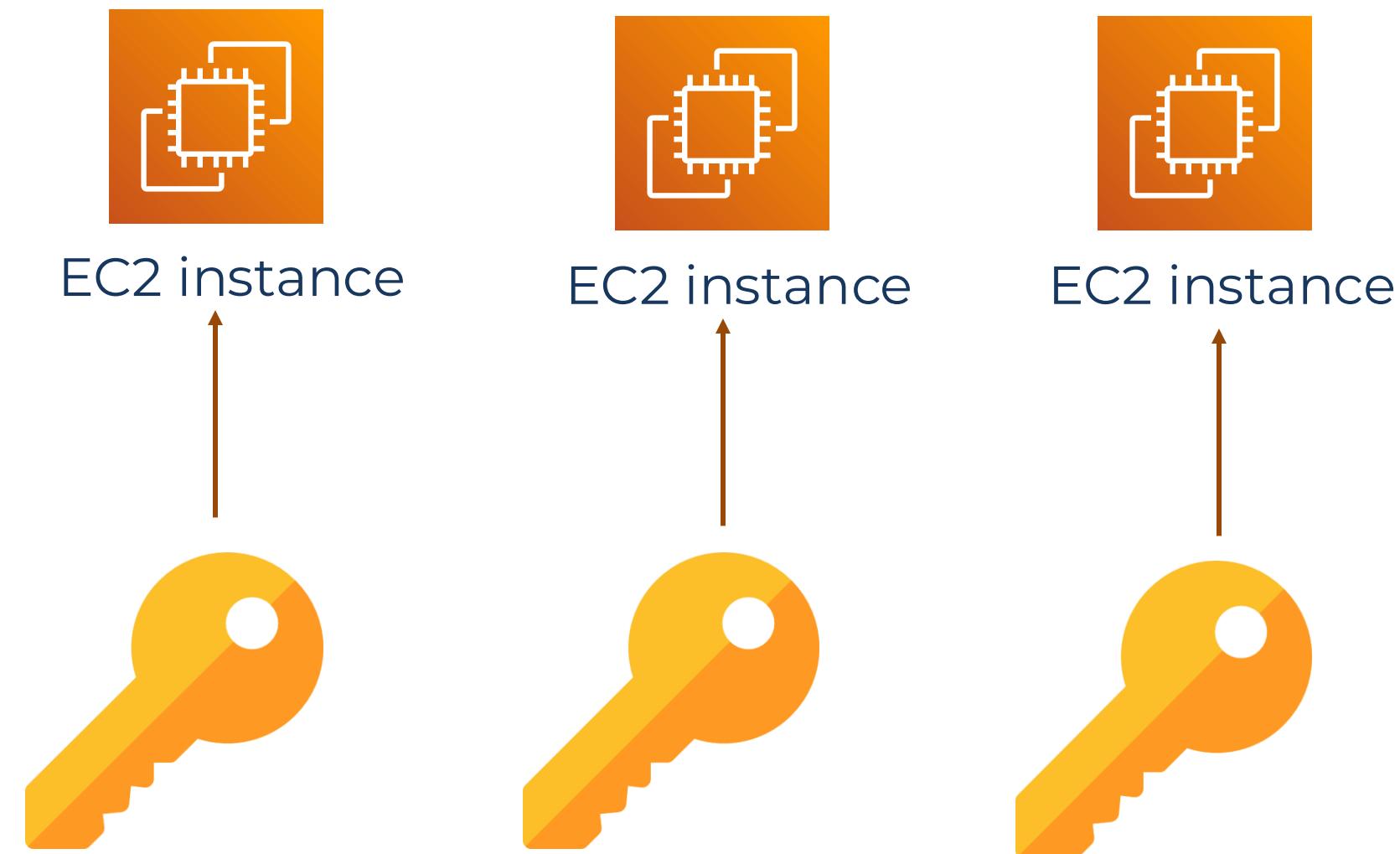
Introduction to EC2

➤ Key Pairs



Introduction to EC2

➤ Key Pairs



Introduction to EC2

➤ Key Pairs

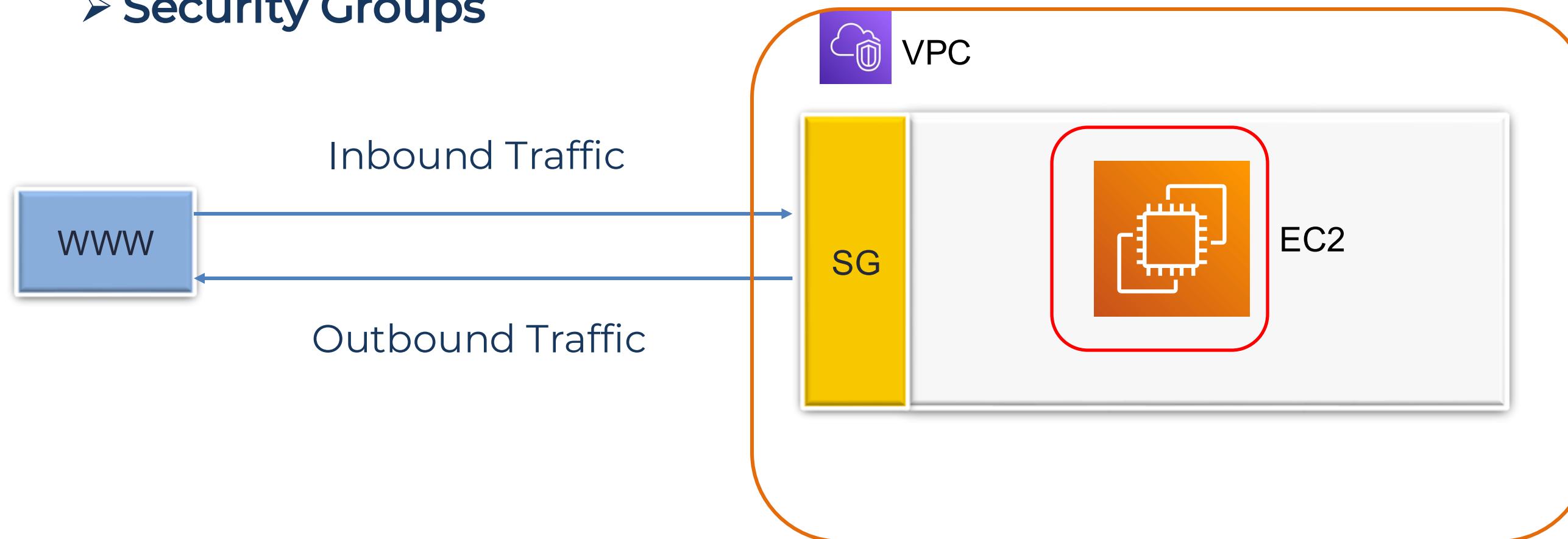


Private Key

- ✓ Amazon EC2 does not keep a copy
- ✓ Store it securely

Introduction to EC2

➤ Security Groups



- ✓ Security group acts as a virtual firewall for EC2 instances
- ✓ Controls the inbound and outbound traffic of EC2 instances
- ✓ Security groups operate only within the VPC
- ✓ You can modify security group rules anytime, and changes apply immediately

Introduction to EC2

➤ How Security Groups Work

□ Stateful Nature

- ✓ **Automatic Response Handling** – Outgoing request responses are allowed automatically
- ✓ **Inbound & Outbound Sync** – Inbound traffic permits its response to leave without explicit outbound rules

□ Evaluating Rules

- ✓ **Rule Evaluation**- EC2 evaluates all rules across all security groups attached to an instance
- ✓ **Multiple Rules Handling**- If multiple rules exist for a port, EC2 considers all of them

Introduction to EC2

➤ How Security Groups Work

□ Two rules for port 22 (SSH access)

1. Rule 1: Allows access from 203.0.113.1 (a specific IP)
2. Rule 2: Allows access from anywhere (0.0.0.0/0)

Since **Rule 2 is more permissive**, anyone can connect via SSH



✗ “Allow from Anywhere” for SSH or RDP in production

✓ Restrict access to **specific IPs**

Introduction to EC2

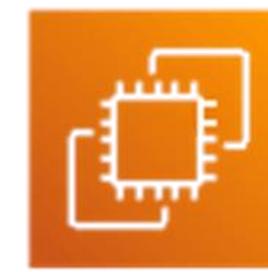
➤ Security Group Rule Example

- Four key components

Type	Protocol	Port Range	Source
SSH	TCP	22	192.168.1.100/32

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload



Amazon EC2

Flexibility

Cost savings

Resource availability

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ On-Demand Instances: Pay-as-You-Go Flexibility

- ✓ Billed per second (minimum 60 seconds)
- ✓ No upfront costs or long-term commitments
- ✓ Flexible and scalable resource allocation

Best for:

- ✓ Short-term, unpredictable workloads
- ✓ Development and testing environments
- ✓ Applications that require immediate scalability

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ Savings Plans: Lower Costs for Predictable Workloads

- ✓ Offer up to 72% cost savings compared to On-Demand pricing
- ✓ In exchange for a 1- or 3-year commitment

Types:

- Compute Savings Plans (applies to EC2, AWS Lambda, and AWS Fargate)
- SageMaker Savings Plans (for machine learning workloads)

Best for:

- ✓ Steady, long-running applications
- ✓ Businesses committed to a set level of compute usage

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ Reserved Instances: Predictable Cost Savings

- ✓ Offer discounted EC2 pricing
- ✓ In exchange for a commitment to a specific instance configuration
- ✓ Unlike Savings Plans, Reserved Instances lack flexibility
- ✓ Still provide up to 72% savings

Best for:

- ✓ Long-running applications with predictable workloads
- ✓ Businesses looking to reduce EC2 costs with fixed configurations
- ✓ Workloads that don't require flexibility in instance types or regions

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ Spot Instances: Maximize Cost Efficiency with Unused Capacity

- ✓ Allow you to use spare AWS capacity
- ✓ At up to 90% discount compared to On-Demand prices
- ✓ AWS may reclaim Spot Instances when demand increases

Best for:

- ✓ Batch processing, big data, and fault-tolerant applications
- ✓ Machine learning training and containerized workloads
- ✓ Applications with flexible start and stop times

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ Capacity Reservations: Guaranteed Compute Resources

- ✓ Secure instance availability in a specific Availability Zone
- ✓ Suitable for any duration
- ✓ Prevent capacity shortages during peak demand

Types of Capacity Reservations

On-Demand Capacity Reservations

- ✓ Reserve capacity for any instance type, duration, or workload

Capacity Blocks for ML

- ✓ Pre-book EC2 P5 or P4d instances
- ✓ Designed for machine learning workloads
- ✓ No long-term commitments

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ Capacity Reservations: Guaranteed Compute Resources

- ✓ Secure instance availability in a specific Availability Zone
- ✓ Suitable for any duration
- ✓ Prevent capacity shortages during peak demand

Best for:

- ✓ Business-critical applications needing reserved capacity
- ✓ Workloads requiring high availability or regulatory compliance

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ Dedicated Instances and Dedicated Hosts

- ✓ Both run workloads on dedicated hardware
- ✓ Dedicated Hosts offer more control over instance placement

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ Dedicated Instances: Isolated Hardware for Your Account

- ✓ Ensure physical isolation at the host hardware level
- ✓ May share hardware with other instances within the same AWS account

Best for:

- ✓ Workloads requiring isolation from other AWS accounts
- ✓ Meeting compliance or regulatory requirements
- ✓ Running applications on dedicated hardware without needing control over placement

Introduction to EC2

➤ Choosing the Right EC2 Pricing Model for Your Workload

□ Dedicated Hosts: Full Server Control & Licensing Benefits

- ✓ Provide a fully dedicated EC2 server
- ✓ Allow full control over instance placement (host affinity)
- ✓ Optionally share capacity with other AWS accounts
- ✓ Support using existing software licenses

Best for:

- ✓ Workloads requiring dedicated physical servers
- ✓ Organizations using Windows Server or SQL Server licenses
- ✓ Applications needing consistent host placement

Introduction to EC2

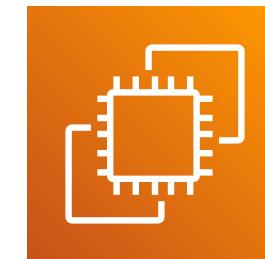
➤ Choosing the Right EC2 Pricing Model for Your Workload

- ✓ If you need flexibility, go for **On-Demand**
- ✓ For cost savings, **Savings Plans** or **Spot Instances** are great choices
- ✓ For critical applications, **Capacity Reservations** or **Dedicated Hosts** provide reliability

Introduction to EC2

➤ Launch Templates

AMI



EC2 instance

Launch Templates

- ✓ More powerful and flexible way to automate instance launches
- ✓ Pre-configured setup

Introduction to EC2

➤ Launch Templates

Aspect	Amazon Machine Image (AMI)	Launch Template
Purpose	Blueprint for EC2 instances	Deployment plan for EC2 instances
What it Defines	OS, software, and storage setup	Networking, security, instance type, and scaling options
Function	Specifies what will be installed	Specifies how the instance will be configured

Introduction to EC2

➤ Launch Templates

Versioning

Modifiability

**Predefined Instance
Settings**

**Auto Scaling
Compatibility**

**Flexible
Deployment**

Introduction to EC2

➤ User Data

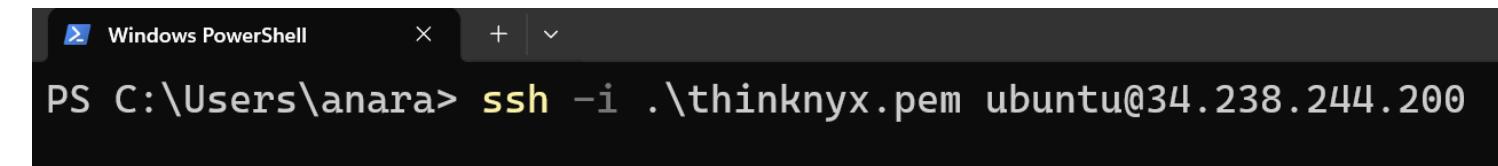
- ✓ User Data allows us to run automation scripts when the instance starts
 - Install software, configure the environment, or launch services
 - Run as root, we don't need to use sudo

Introduction to EC2

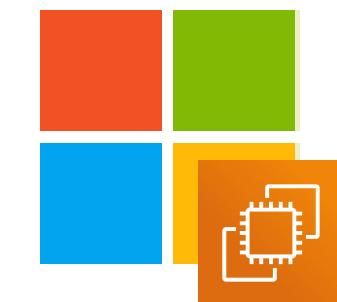
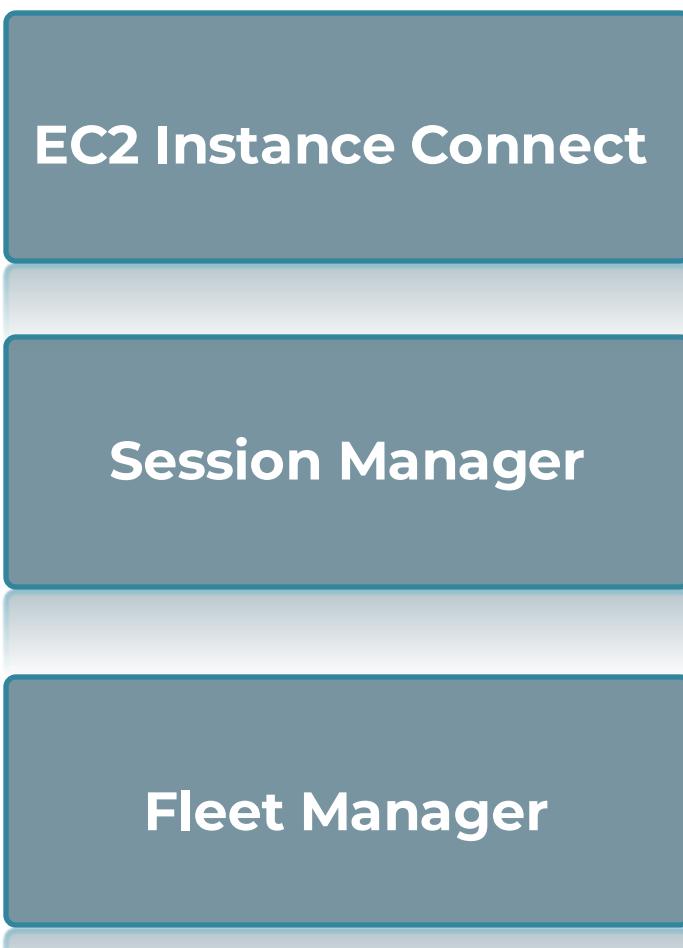
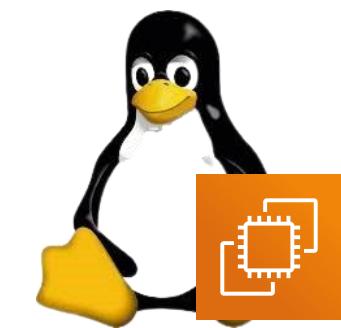
➤ Connect to EC2 instance



Amazon EC2

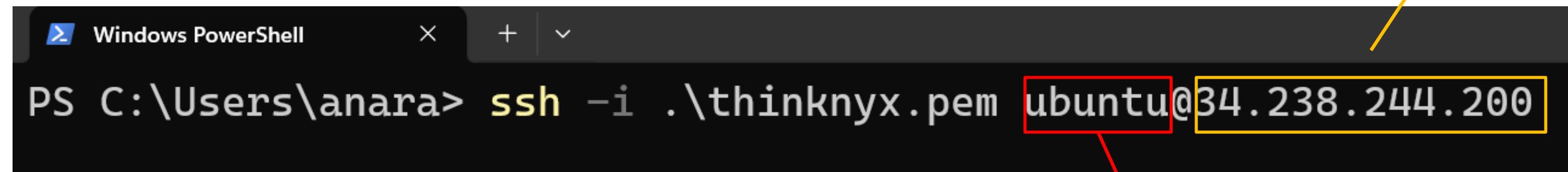


```
PS C:\Users\anara> ssh -i .\thinknyx.pem ubuntu@34.238.244.200
```

A screenshot of a Windows PowerShell window. The command entered is "ssh -i .\thinknyx.pem ubuntu@34.238.244.200".

Introduction to EC2

➤ Connect to EC2 instance



```
Windows PowerShell
PS C:\Users\anara> ssh -i .\thinknyx.pem ubuntu@34.238.244.200
```

A screenshot of a Windows PowerShell window. The title bar says "Windows PowerShell". The command entered is "ssh -i .\thinknyx.pem ubuntu@34.238.244.200". The IP address "34.238.244.200" is highlighted with a yellow box, and the word "ubuntu" is highlighted with a red box. A yellow arrow points from the text "Public IP" to the yellow box around the IP address. A red arrow points from the text "Username" to the red box around the word "ubuntu".

- For an Amazon Linux AMI, the username is `ec2-user`.
- For a CentOS AMI, the username is `centos` or `ec2-user`.
- For a Debian AMI, the username is `admin`.
- For a Fedora AMI, the username is `fedora` or `ec2-user`.
- For a RHEL AMI, the username is `ec2-user` or `root`.
- For a SUSE AMI, the username is `ec2-user` or `root`.
- For an Ubuntu AMI, the username is `ubuntu`.
- For an Oracle AMI, the username is `ec2-user`.
- For a Bitnami AMI, the username is `bitnami`.

Introduction to EC2

➤ Understanding Public vs. Private vs. Elastic IPs

Private IP

- ✓ Internal VPC communication
- ✓ Static (doesn't change when instance stops/starts)

Public IP

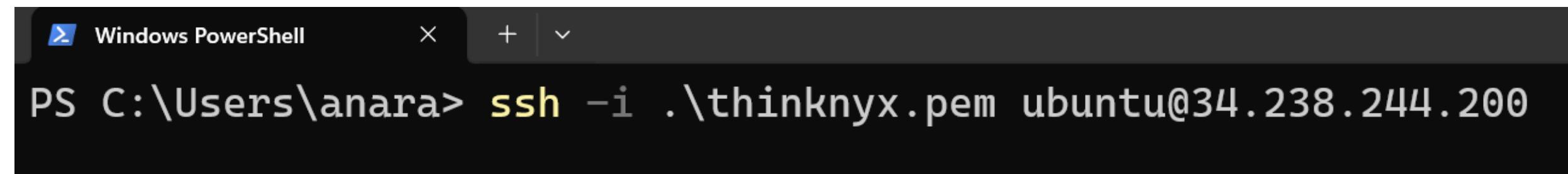
- ✓ Internet access
- ✓ Dynamic (changes when instance stops/starts)

Elastic IP (EIP)

- ✓ Static public IP
- ✓ Persistent (remains the same until released)

Introduction to EC2

Upcoming Demonstration



```
PS C:\Users\anara> ssh -i .\thinknyx.pem ubuntu@34.238.244.200
```





Demonstration | Creating Instances in AWS (Linux)



Demonstration | Creating Instances in AWS (Windows)



Demonstration | Accessing Linux Server



Demonstration | Creating AMI using EC2



Demonstration | Creating Launch Templates



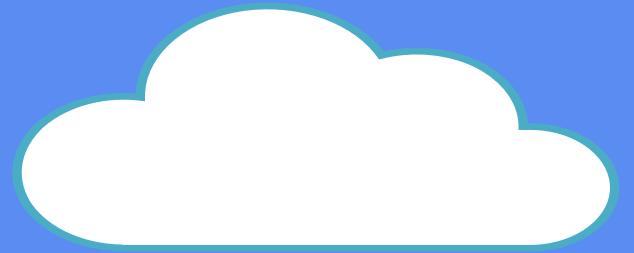
Demonstration | Create EC2 using Launch Templates

Summary

- EC2
- Core concepts and Key features - AMIs, Instance Types, EBS Volumes, Key Pairs, and Security Groups
- Pricing models - On-Demand, Savings Plans, Reserved Instances, Spot Instances, Dedicated Hosts, Dedicated Instances, and Capacity Reservations
- **Demo 1:** Created and access Linux and Windows EC2 instances
- **Demo 2:** Deployed a sample application on a Linux instance
- **Demo 3:** Created an AMI and launched instances from it
- **Demo 4:** Used launch templates for efficient instance provisioning

Cloud Fundamentals with AWS

Section 6



Introduction to Load Balancers

- ❖ AWS Load Balancers—NLB, ALB, and Gateway
- ❖ Target Groups and Listeners
- ❖ **Demonstration**
 - Set up a Target Group
 - Create a Load Balancer
 - Access the application using its DNS



Amazon EC2



Understanding Load Balancer

Understanding Load Balancer

➤ What is Load Balancing?

- ✓ Load balancing is the process of distributing incoming network traffic
- ✓ Across multiple servers that host an application
- ✓ Ensures no single server becomes overwhelmed

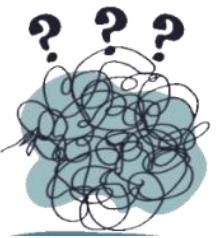
Performance

Availability

Reliability

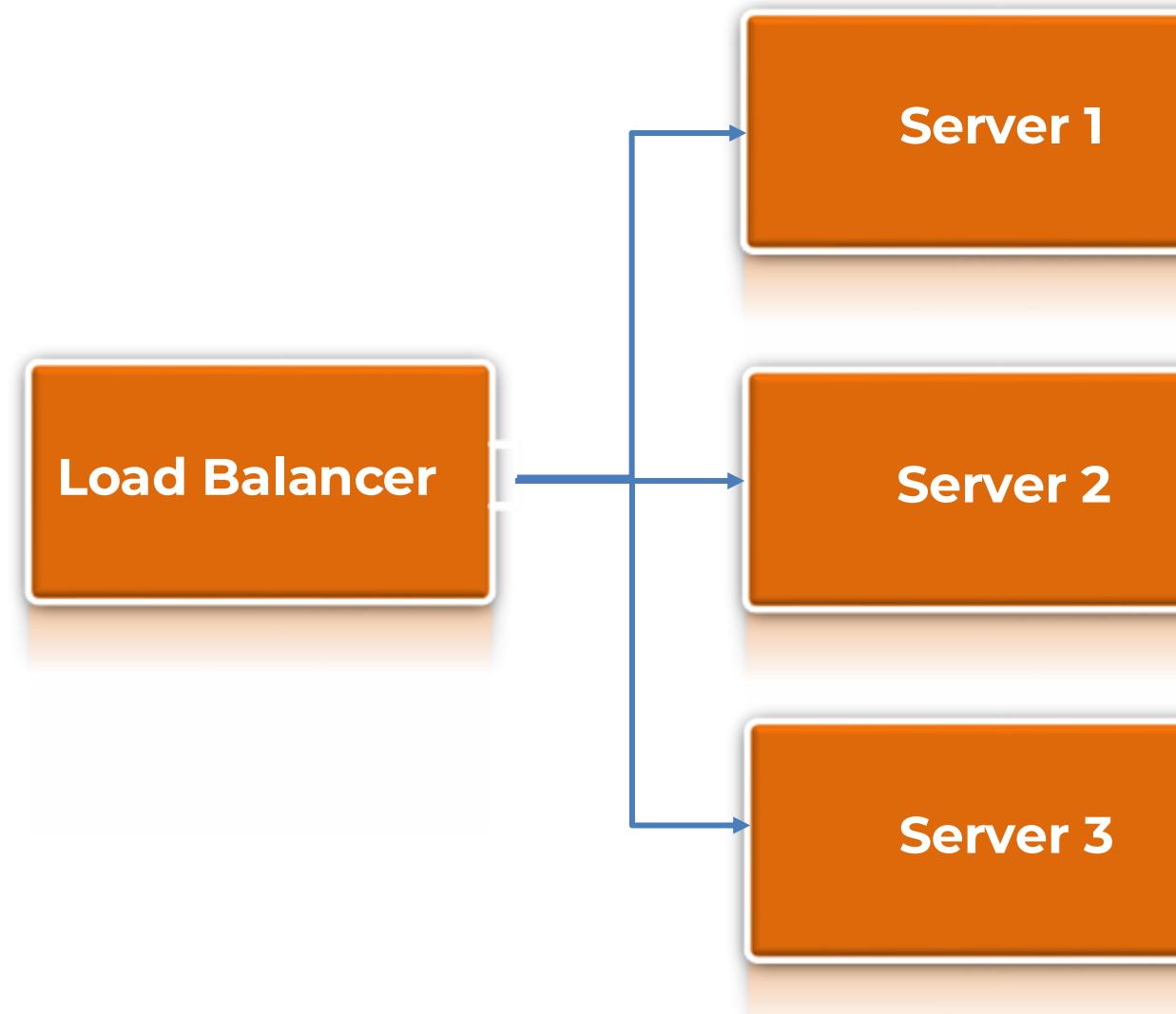
Understanding Load Balancer

➤ What is Load Balancing?



Understanding Load Balancer

➤ What is Load Balancing?



✓ Keeps everything balanced and efficient

Understanding Load Balancer

➤ Why Use a Load Balancer?

Availability

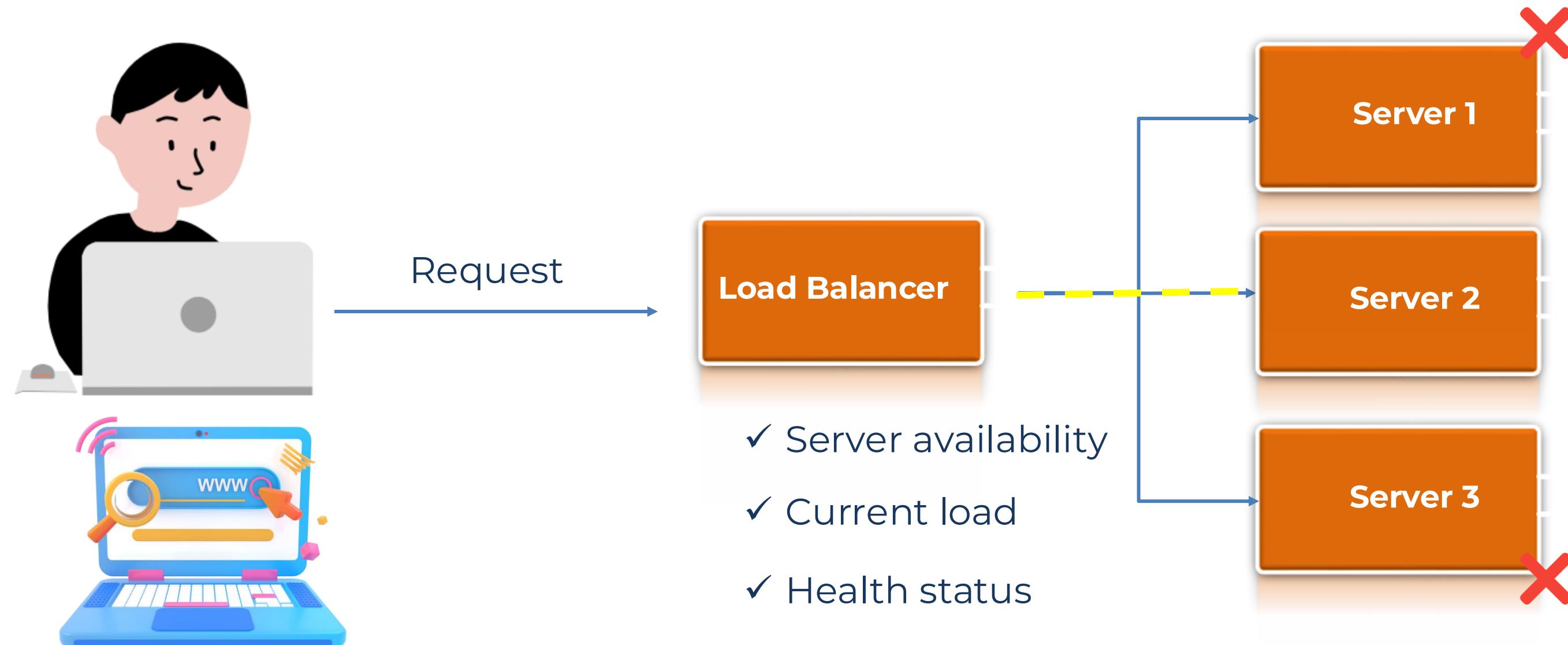
Scalability

Security

Performance

Understanding Load Balancer

➤ How Does It Work?



- ✓ Process takes milliseconds
- ✓ Ensures that traffic is always directed efficiently
- ✓ Helps maintain speed, reliability, and availability for users

Understanding Load Balancer

➤ Load Balancing Algorithms

- **Static algorithms**
- **Dynamic algorithms**

Understanding Load Balancer

➤ Load Balancing Algorithms

□ Static algorithms

- ✓ Don't change based on real-time server conditions
- **Round Robin**, where each server takes turns handling requests in a simple rotation
- **Weighted Round Robin**, where more powerful servers are assigned a greater share of the traffic
- **IP Hash**, which consistently routes a user to the same server based on their IP address

Understanding Load Balancer

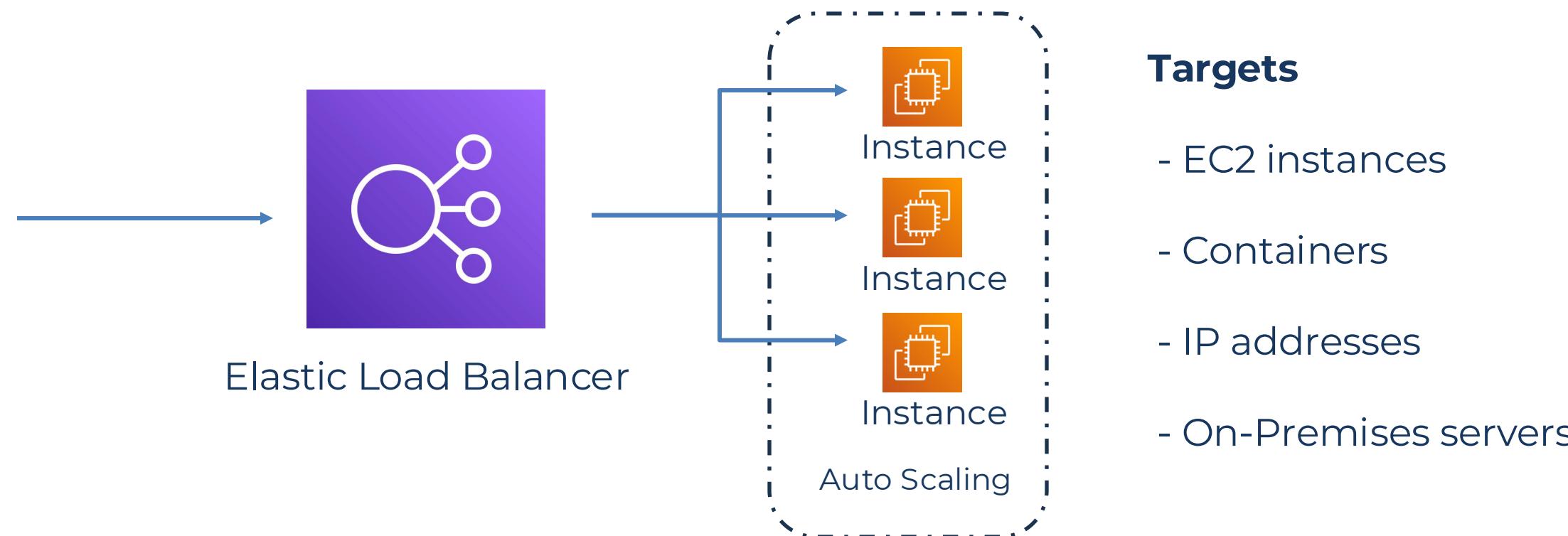
➤ Load Balancing Algorithms

□ Dynamic algorithms

- ✓ They take into account the current state of each server
 - **Least Connections**, which sends new requests to the server with the fewest active connections
 - **Least Response Time**, which prioritizes the server responding the fastest
 - **Resource-Based**, which looks at things like CPU and memory usage before deciding
-
- ✓ **Load balancer ensures that no single server is overwhelmed**
 - ✓ **Users always receive a fast, reliable response**

Understanding Load Balancer

➤ Load Balancing in AWS using ELB



Targets

- EC2 instances
- Containers
- IP addresses
- On-Premises servers

- ✓ Removes the complexity of manually configuring
- ✓ Focus on your applications
- ✓ Highly available, scalable, and integrates seamlessly with other AWS services
- ✓ AWS Certificate Manager (ACM), and AWS Web Application Firewall (WAF) for added security

Understanding Load Balancer

➤ Four Types of AWS Load Balancers

□ Application Load Balancer (ALB)

- ✓ ALB operates at Layer 7 (Application Layer)
- ✓ Routes traffic based on content-level information (URL, host headers, HTTP methods)
- ✓ Ideal for modern web applications, microservices, and APIs
- ✓ Enables sophisticated routing (e.g., /login to one service, /checkout to another)
- ✓ Improves load balancing efficiency at the application level

Understanding Load Balancer

➤ Four Types of AWS Load Balancers

□ Network Load Balancer (NLB)

- ✓ Layer 4 Load Balancer (Transport Layer)
- ✓ Ultra-Low Latency & High Throughput
- ✓ Handles Millions of Requests per Second
- ✓ Supports TCP & UDP Protocols
- ✓ Optimized for Static IP Addresses
- ✓ Ideal for Volatile Traffic (Gaming, Finance, etc.)

Understanding Load Balancer

➤ Four Types of AWS Load Balancers

□ Gateway Load Balancer (GWLB)

- ✓ Operates at Layer 3 (Network Layer)
- ✓ Simplifies deployment, scaling, and management of third-party virtual appliances (e.g., firewalls, IDS)
- ✓ Acts as a single entry and exit point for traffic flow
- ✓ Combines Layer 3 packet forwarding with Layer 4 load balancing
- ✓ Ensures session stickiness and high availability for security services

Understanding Load Balancer

➤ Four Types of AWS Load Balancers

□ Classic Load Balancer (CLB)

- ✓ Classic Load Balancer (CLB): AWS's legacy load balancer
- ✓ Layer Support: Works at both Layer 4 (Transport) and Layer 7 (Application)
- ✓ Feature Limitations: Lacks advanced features of ALB (Application Load Balancer) and NLB (Network Load Balancer)
- ✓ Primary Use Case: Supports older applications in the EC2-Classic network

Understanding Load Balancer

➤ ELB Components

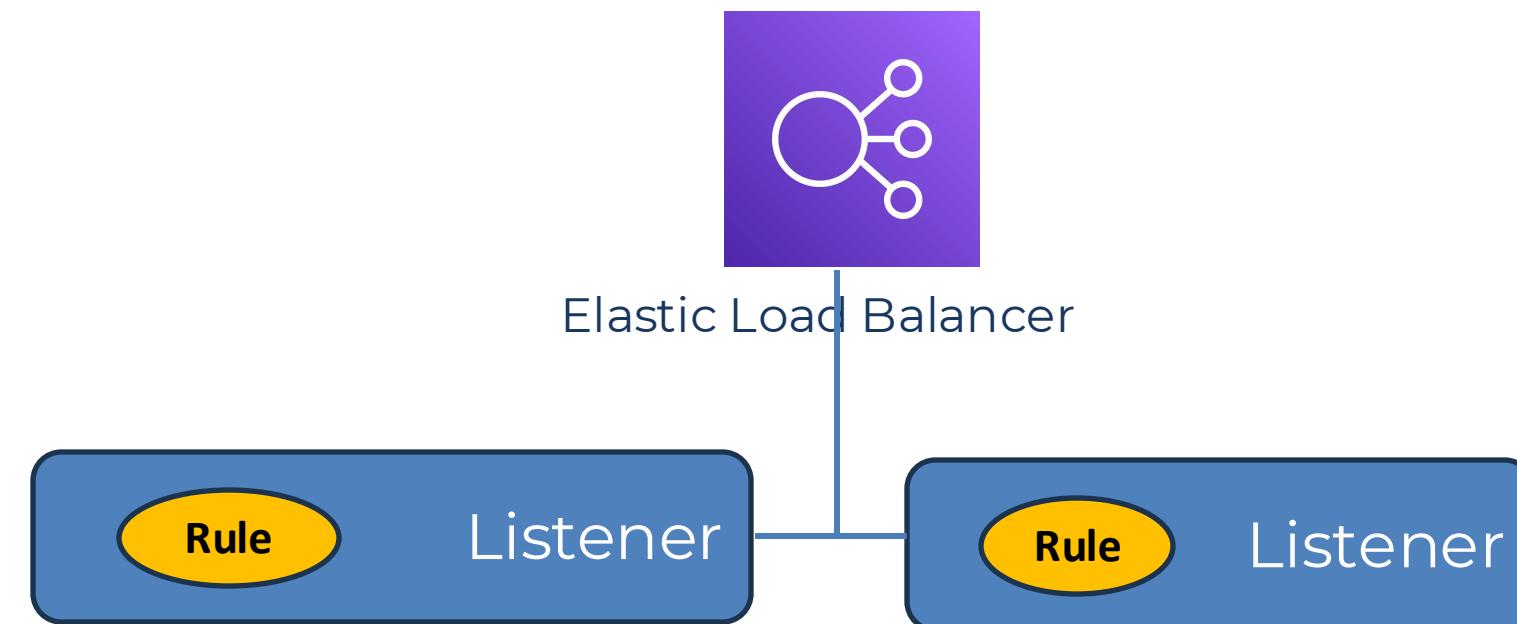
Listeners

Target Groups

Health Checks

Understanding Load Balancer

➤ ELB Components



Protocol

- HTTP
- HTTPS
- TCP



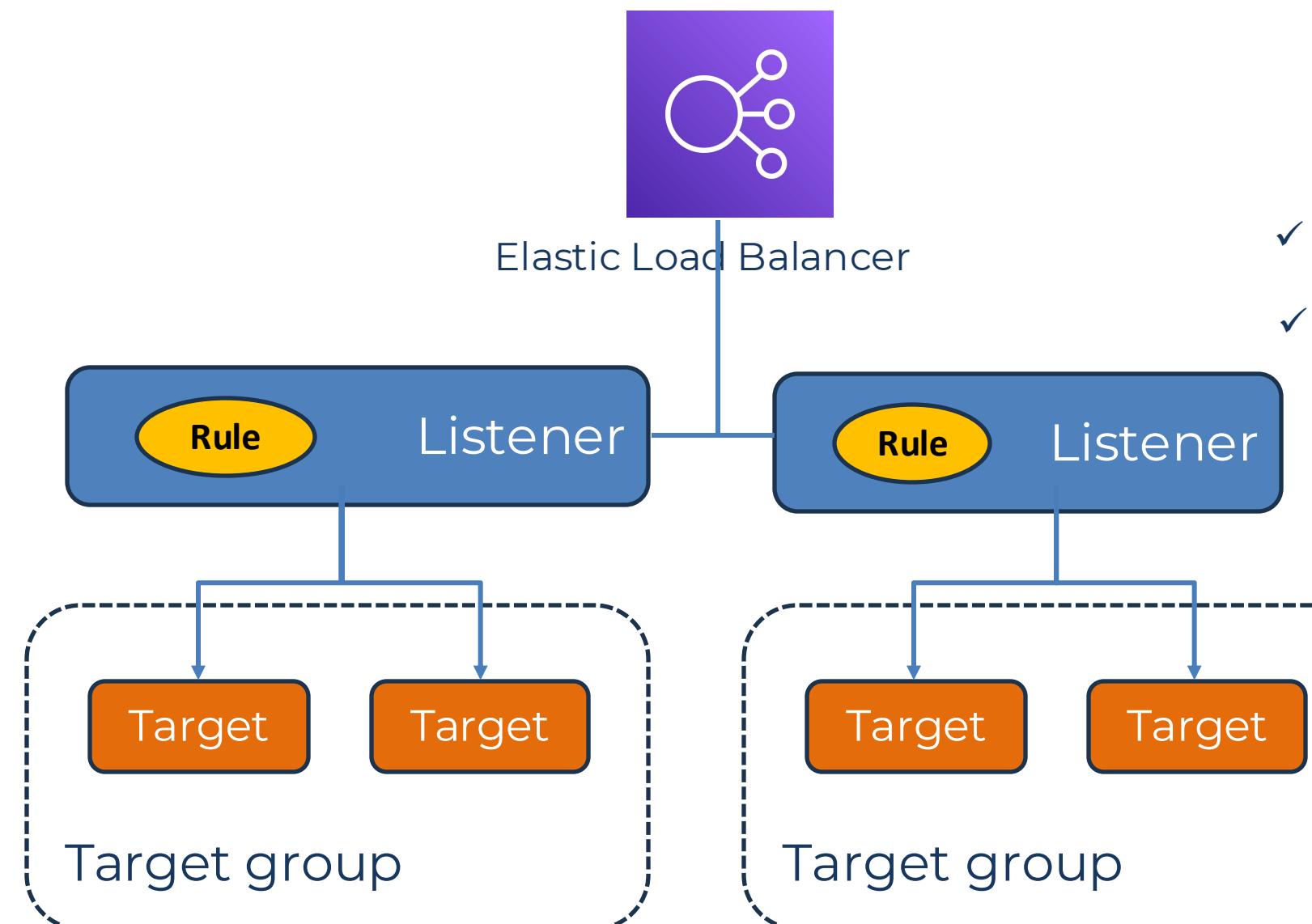
Port

- 80 or 443

- ✓ Route to URL path, the hostname, or query strings

Understanding Load Balancer

➤ ELB Components

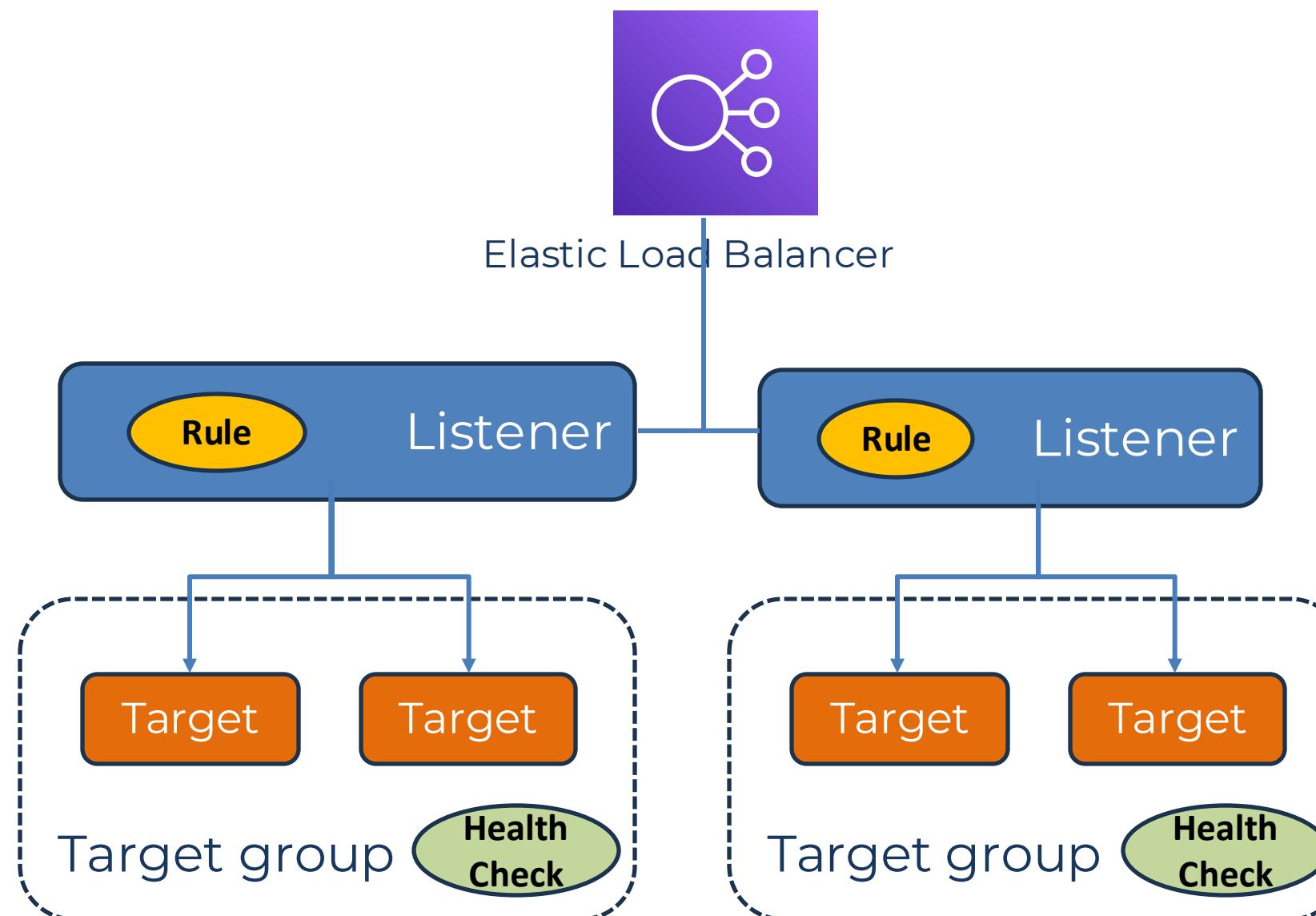


- ✓ Web servers might be listening on HTTP port 80
- ✓ Application servers listen on HTTPS port 443

- ✓ A target group is a collection of resources that the load balancer can send traffic to
- ✓ EC2 instances, containers, IP addresses, or even Lambda functions in the case of an ALB
- ✓ Its own set of configuration settings

Understanding Load Balancer

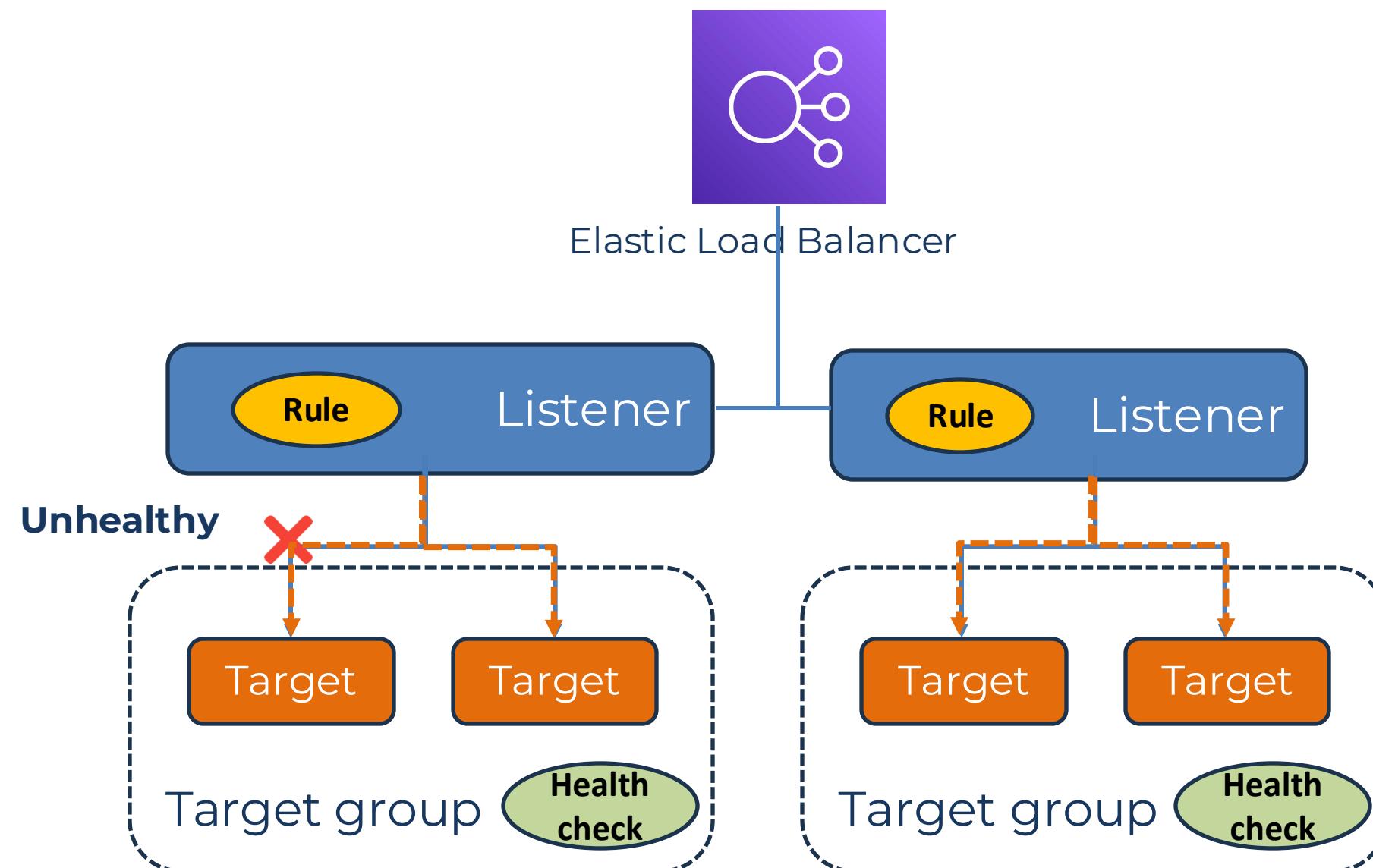
➤ ELB Components



- ✓ Are automated tests that the load balancer performs
- ✓ Make sure the targets in a group are healthy and ready to handle traffic
- ✓ Protocol and path to check
- ✓ How often the checks run
- ✓ How many times a target must fail before it's considered unhealthy

Understanding Load Balancer

➤ ELB Components





Demonstration | Creating Target Group and Register EC2



Demonstration | Creating Load Balancer with Listener



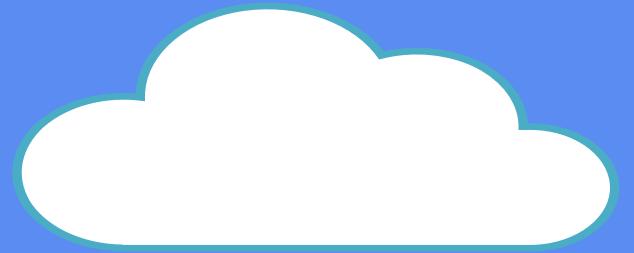
Demonstration | Create and Replace EC2 from Target Group

Summary

- Types of AWS Load Balancers: NLB, ALB, and Gateway
- Key components: Target Groups and Listeners
- **Demonstration Recap**
 - ✓ Created a Target Group and registered EC2 instances
 - ✓ Set up a Load Balancer with a Listener
 - ✓ Accessed the app via Load Balancer DNS
 - ✓ Replaced an EC2 instance in the Target Group
 - ✓ Verified the new version was accessible via the same URL

Cloud Fundamentals with AWS

Section 7



Autoscaling Groups

- ❖ Autoscaling Groups
- ❖ Hands-on Demos
 - Create Auto Scaling Group with launch templates, set up health checks and target groups
 - Instance refreshes
 - Simulate scaling actions based on CPU usage and health check failures

Introduction to Autoscaling Groups

Autoscaling Groups

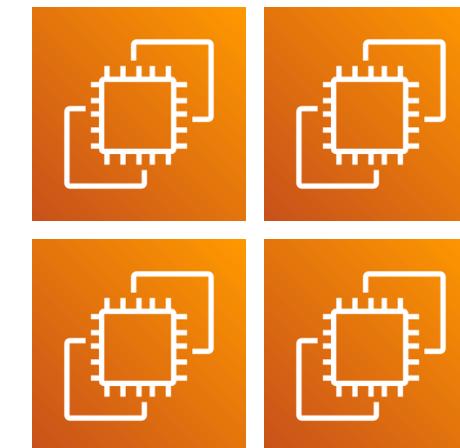


Autoscaling Groups

➤ Amazon EC2 Auto Scaling

- ✓ A service that automatically adjusts the number of Amazon EC2 instances in your environment

Decrease in Demand



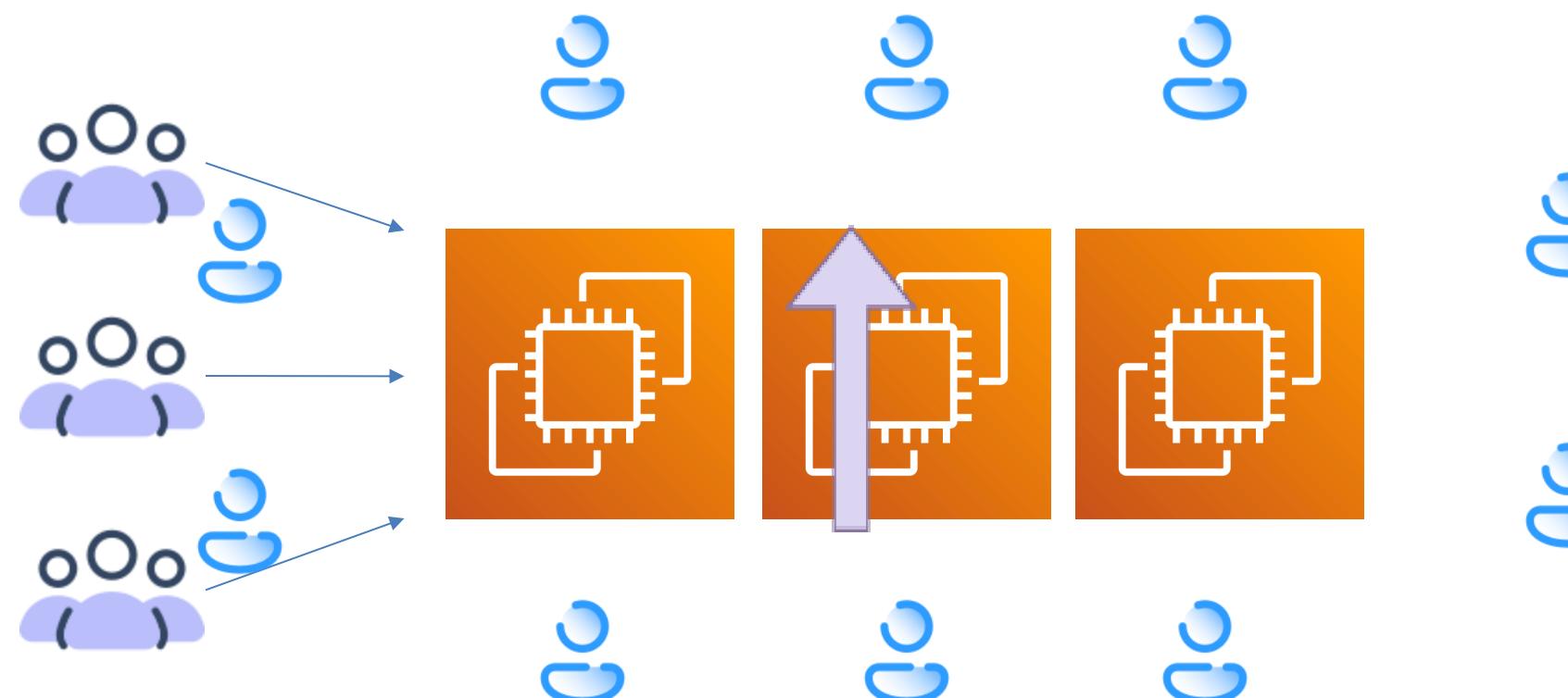
Flexible
Automated



Autoscaling Groups

➤ Working of Autoscaling

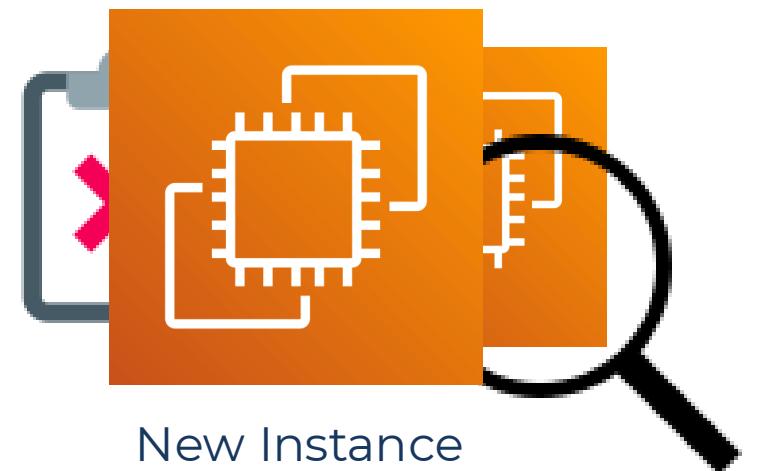
- ✓ Auto Scaling Group or ASG
- ✓ Manages a group of EC2 instances
 - The **minimum number** of instances that should always be running
 - The **maximum number** you're willing to scale to
 - The **desired capacity**, or target number of instances under normal conditions



Autoscaling Groups

➤ Key Features

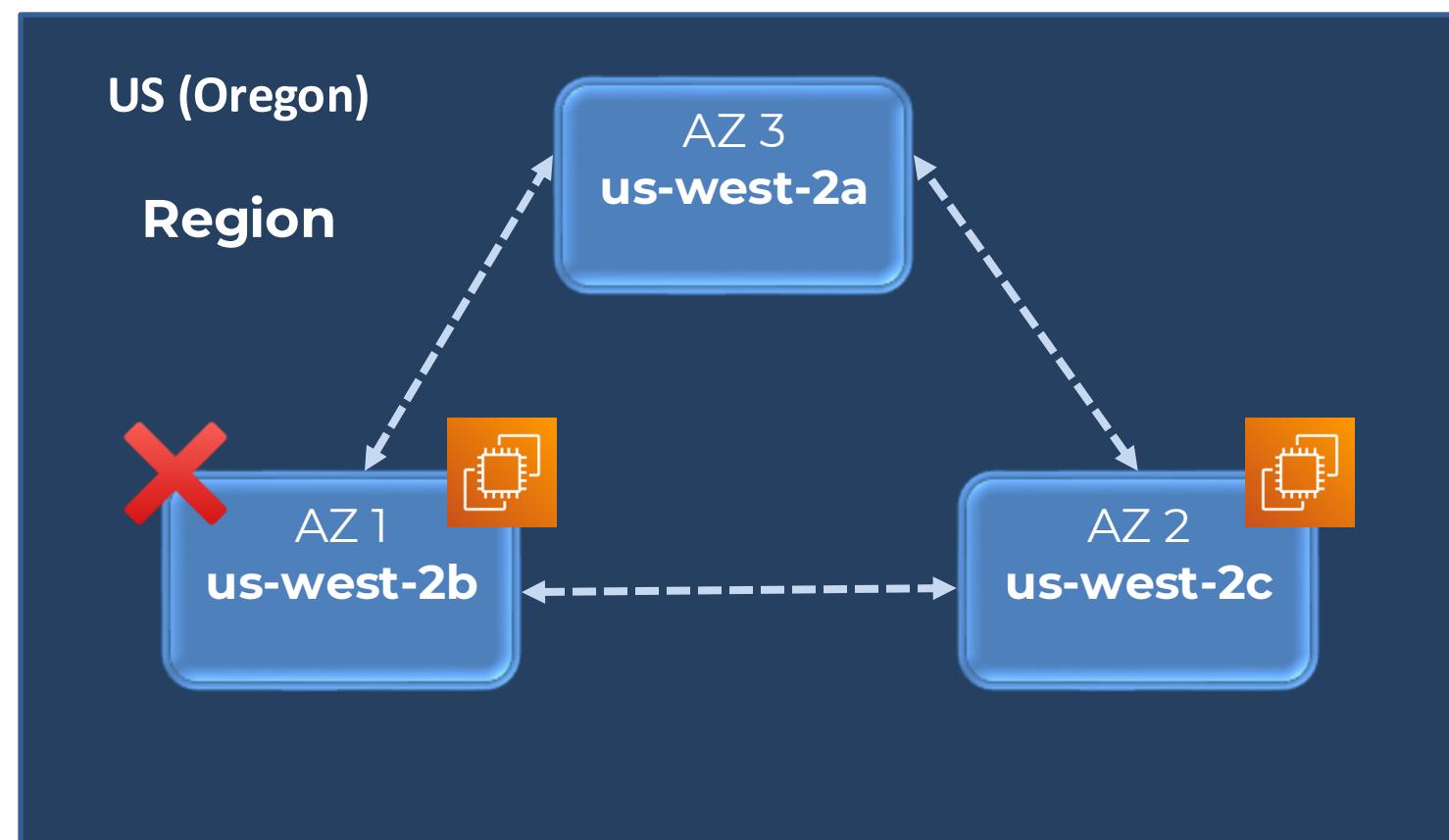
- ✓ **Automated Health Monitoring and Recovery**



Autoscaling Groups

➤ Key Features

- ✓ Multi-AZ Deployments for High Availability



Autoscaling Groups

➤ Key Features

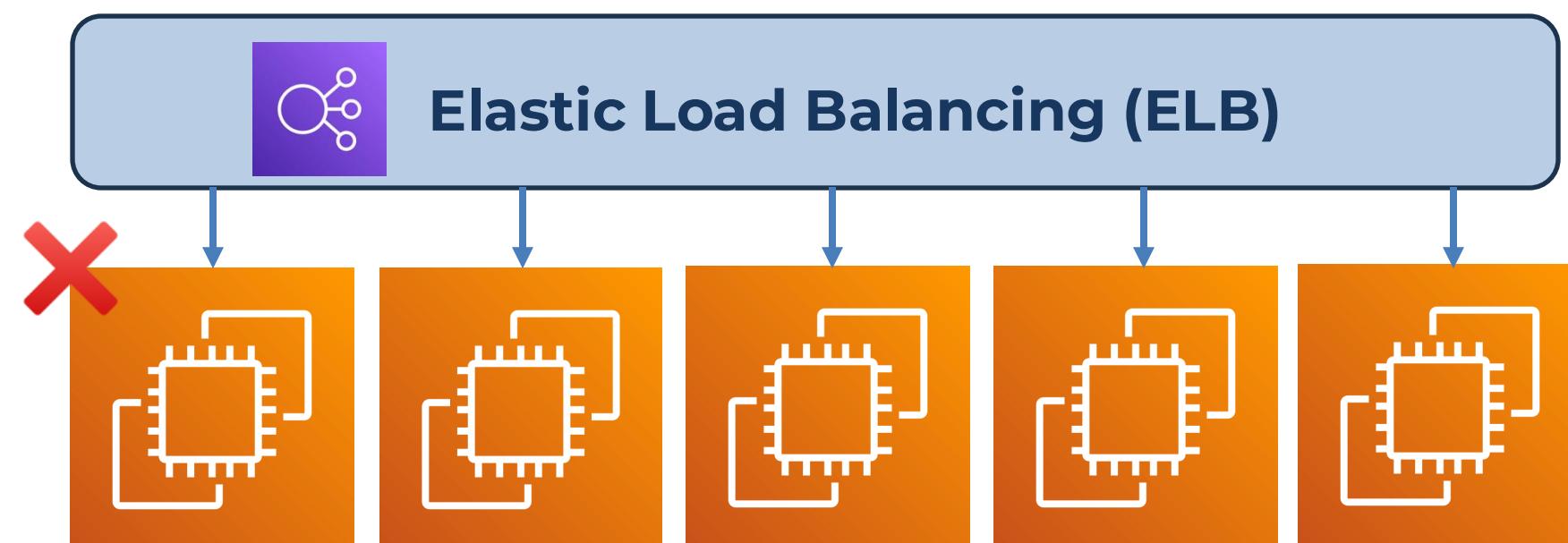
✓ Flexible Resource Strategies

- Mix different instance types and purchase options
- Combining On-Demand Instances with Spot Instances
- Optimizes cost and performance

Autoscaling Groups

➤ Key Features

- ✓ Integration with Elastic Load Balancing (ELB)



Autoscaling Groups

➤ Key Features

✓ Scaling Options

- **Dynamic Scaling** adjusts resources in real-time based on CPU usage or request rates
- **Predictive Scaling** uses ML to forecast traffic and scale resources proactively
- **Scheduled Scaling** adjusts capacity at set times for predictable demand changes

Autoscaling Groups

➤ Key Features

✓ **Instance Refresh for Zero-Downtime Updates**

- **Instance Refresh** updates applications by gradually replacing instances
- Ensures continuous availability during updates
- Helps roll out configuration changes smoothly

Autoscaling Groups

➤ Key Features

✓ Lifecycle Hooks for Custom Automation

- **Lifecycle Hooks:** Control the instance launch and termination process
- **Custom Actions:** Install software or shut down services before state transition

✓ Scale-In Protection for Critical Instances

- **Scale-in Protection** keeps critical instances running until manually removed

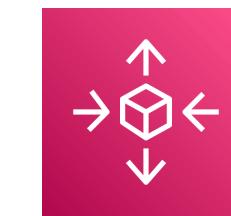
Autoscaling Groups

➤ Business Benefits

- ✓ **High Availability:** Ensures uptime during failures or traffic spikes
- ✓ **Cost Efficiency:** Aligns resources to demand, reducing costs
- ✓ **Operational Simplicity:** Automates scaling and healing, reducing manual effort

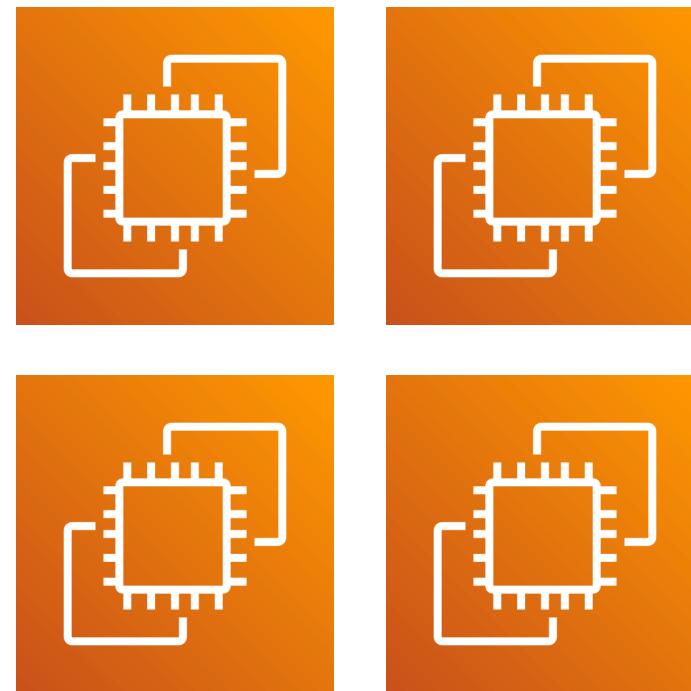
Autoscaling Groups

➤ Example



Autoscaling

During Evening



Autoscaling Groups

➤ Summary

✓ **Amazon EC2 Auto Scaling helps organizations:**

- Maintain application performance and availability
- Automatically adjust capacity to meet changing demands
- Optimize costs by scaling in and out as needed



Demonstration | Autoscaling with Launch Templates in Public Subnets

Autoscaling Groups

➤ Autoscaling with Launch Templates in Public Subnets

- ✓ Previously, we understood how we can leverage load balancers to provide a single static URL
 - No impact on the URL despite backend server or IP address change
 - Manual addition to the target group
- ✓ Some common situations
 - Upgrading application to the newer version
 - Rollback or degrade to the previous version
 - Adding more servers to handle traffic spike
 - Reducing servers during low traffic
- ✓ Manually, it becomes inefficient and error prone

Autoscaling Groups

➤ Autoscaling with Launch Templates in Public Subnets

- ✓ **Auto Scaling Groups**
- ✓ Automates EC2 launch, termination, and management based on scaling policies and health checks
- ✓ Ensures optimum servers according to the demand



Demonstration | Instance Refresh

Autoscaling Groups

➤ Understanding Instance Refresh

- ✓ Instance refresh helps to recreate/replace an existing instance with a new one
- ✓ Useful for upgrading or updating the instance
- ✓ Allows to automatically bring up new servers



Demonstration | Upgrade Application Version



Demonstration | Simulating EC2 Autoscaling on Health Check Failure



Demonstration | Simulating ELB Autoscaling on Health Check Failure



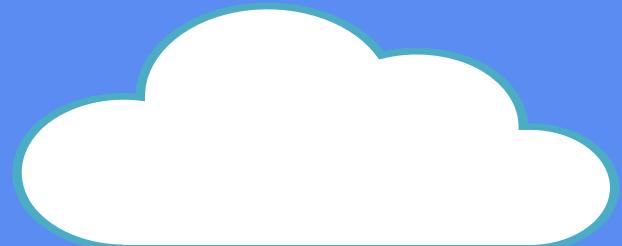
Demonstration | Auto Scaling: Scale-Out on High CPU

Summary

- The core concepts of Auto Scaling Groups
- Created Autoscaling group with Launch Templates in Public Subnets
- Explored the instance refresh feature to upgrade the application version
- Simulated Auto Scaling responses to health check failures at EC2 and ELB levels
- Tested scaling actions by increasing CPU utilization

Cloud Fundamentals with AWS

Section 9



S3

- ❖ Amazon S3, its key features, and practical use cases
- ❖ Hands-on Demos
 - How to create an S3 bucket
 - Set up IAM policies and roles for secure access
 - Integrate S3 into our application

Introduction to S3

S3

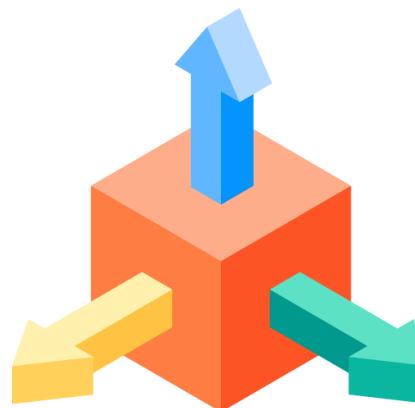


S3

➤ Amazon S3: AWS Storage service



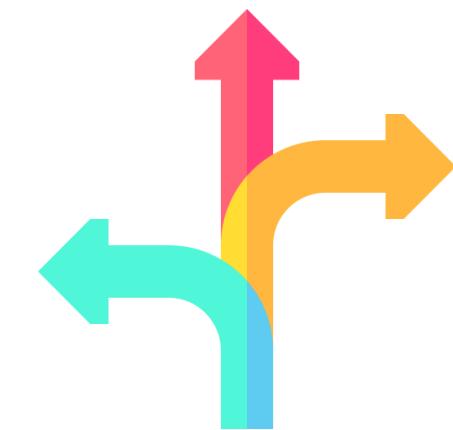
Secure



Scalable



Durable



Flexible

S3

➤ Understanding S3 Storage Classes



S3

➤ Understanding S3 Storage Classes

- ✓ Designed for eleven nines of durability—that's 99.999999999 %
- ✓ Availability varies based on the class
- ✓ Most storage classes are available across at least three availability zones
- ✓ One-Zone storage classes are stored in a single availability zone

S3

➤ Performance-Sensitive Storage Classes

- ✓ Workloads that require millisecond access and frequent data retrieval, Amazon S3 offers:

S3 Standard

- High durability, availability, and fast access
- Best for frequently accessed data

S3 Express One Zone

- Ultra-low latency storage for fast, consistent access in a single availability zone

S3

➤ Infrequent Access Storage Classes

- ✓ For infrequent access with millisecond retrieval, Amazon S3 offers:

S3 Standard Infrequent Access (Standard-IA)

- Ideal for backups and older data requiring quick access

S3 One Zone Infrequent Access

- Like Standard-IA but in one zone, lower cost, slightly higher risk

S3

➤ S3 Glacier – Long-Term Archival Storage

- ✓ For data that is rarely accessed and stored for long periods, AWS offers:

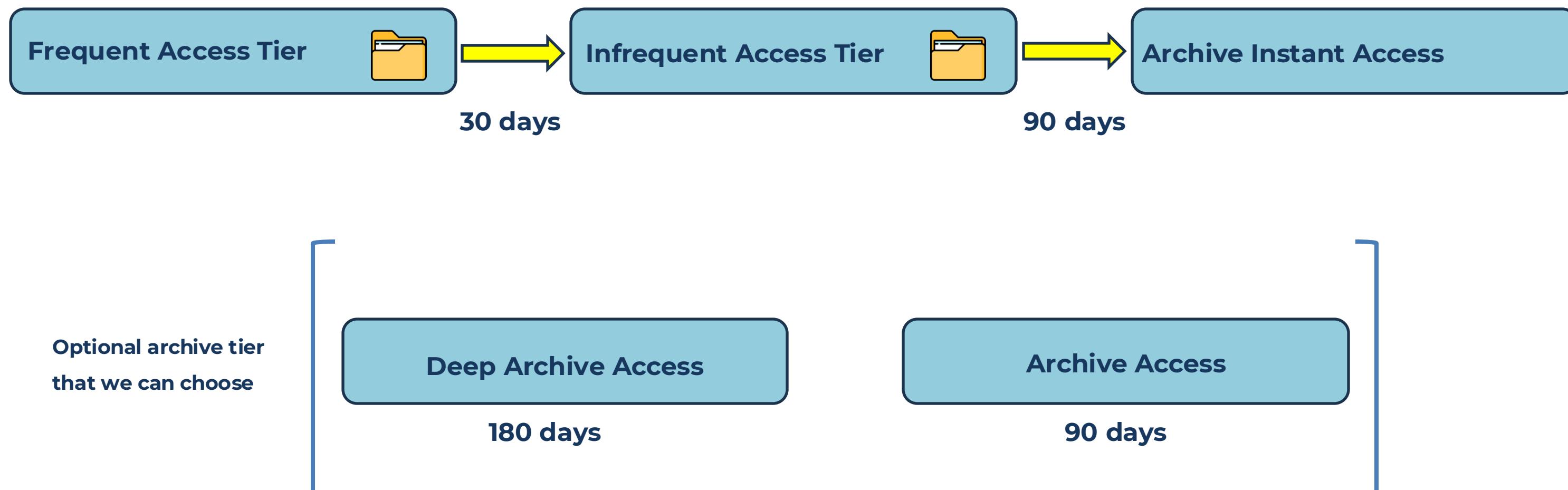
Storage Class	Best for	Retrieval Time
S3 Glacier Instant Retrieval	Rarely accessed but needs real-time retrieval	Milliseconds
S3 Glacier Flexible Retrieval	Archival data with occasional retrieval	Minutes to hours
S3 Glacier Deep Archive	Data accessed once or twice a year	Hours (12–48)

- ✓ Cost-effective storage with retrieval time
- ✓ Ideal for compliance archives and long-term backups

S3

➤ S3 Intelligent-Tiering – Automated Cost Optimization

- ✓ S3 Intelligent-Tiering optimizes costs by automatically moving data across access tiers.



Note: Objects smaller than 128 KB always remain in the Frequent Access tier.

S3

➤ Choosing the Right Storage Class

For frequent access with low latency → S3 Standard or S3 Express One Zone

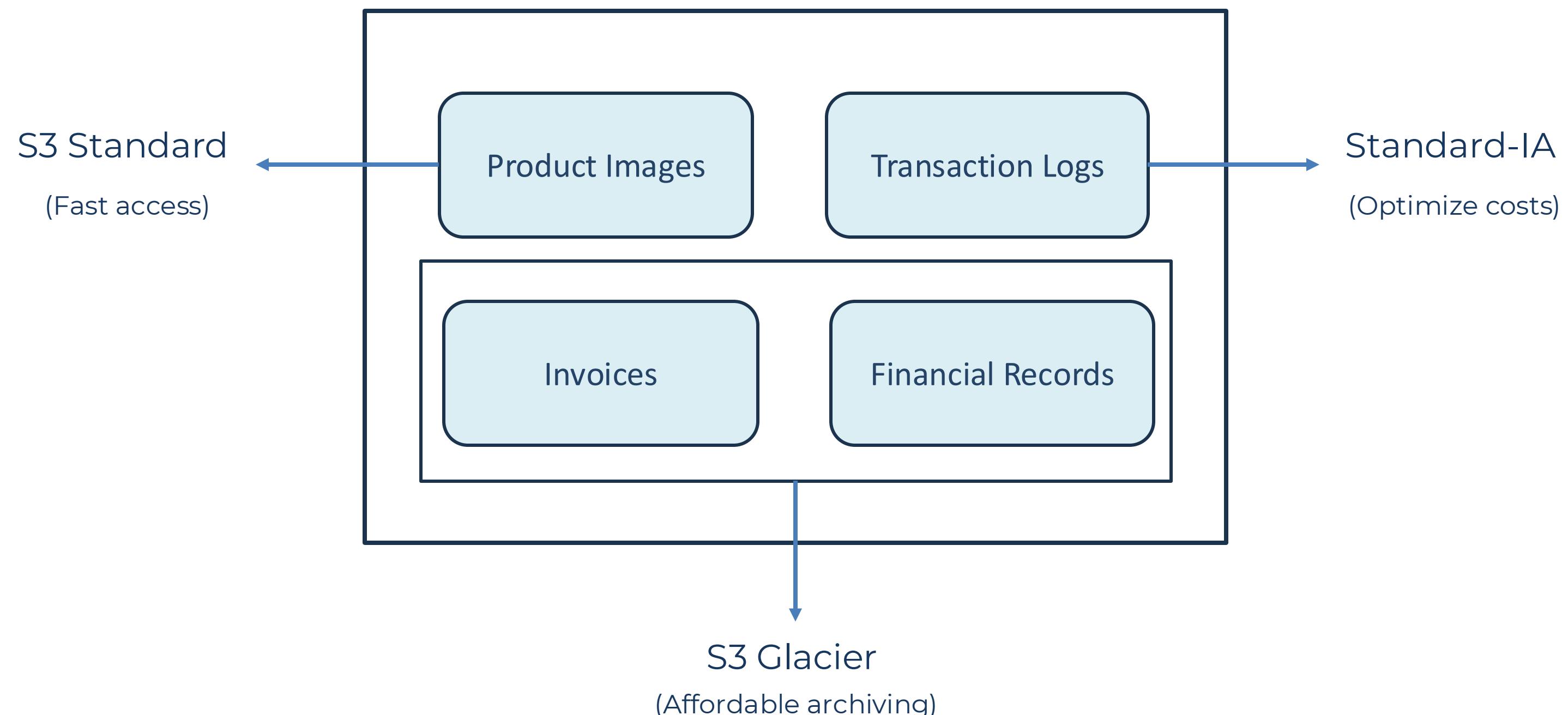
For occasional but still needs millisecond retrieval → Standard-IA or One Zone-IA

For long-term archival with minimal access → S3 Glacier

For changing or unpredictable access patterns → S3 Intelligent-Tiering

S3

➤ Choosing the Right Storage Class

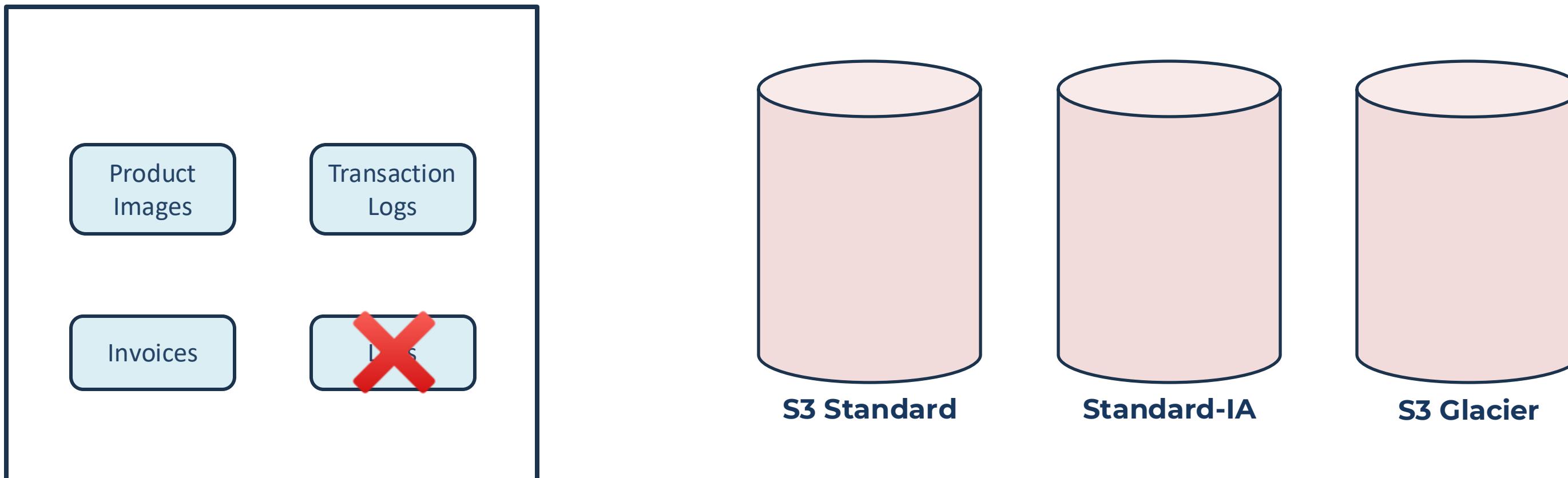


S3

➤ Optimizing Storage Management in Amazon S3

□ S3 Lifecycle – Automating Cost Optimization

- ✓ Automatically transition objects between storage classes or delete them when no longer needed

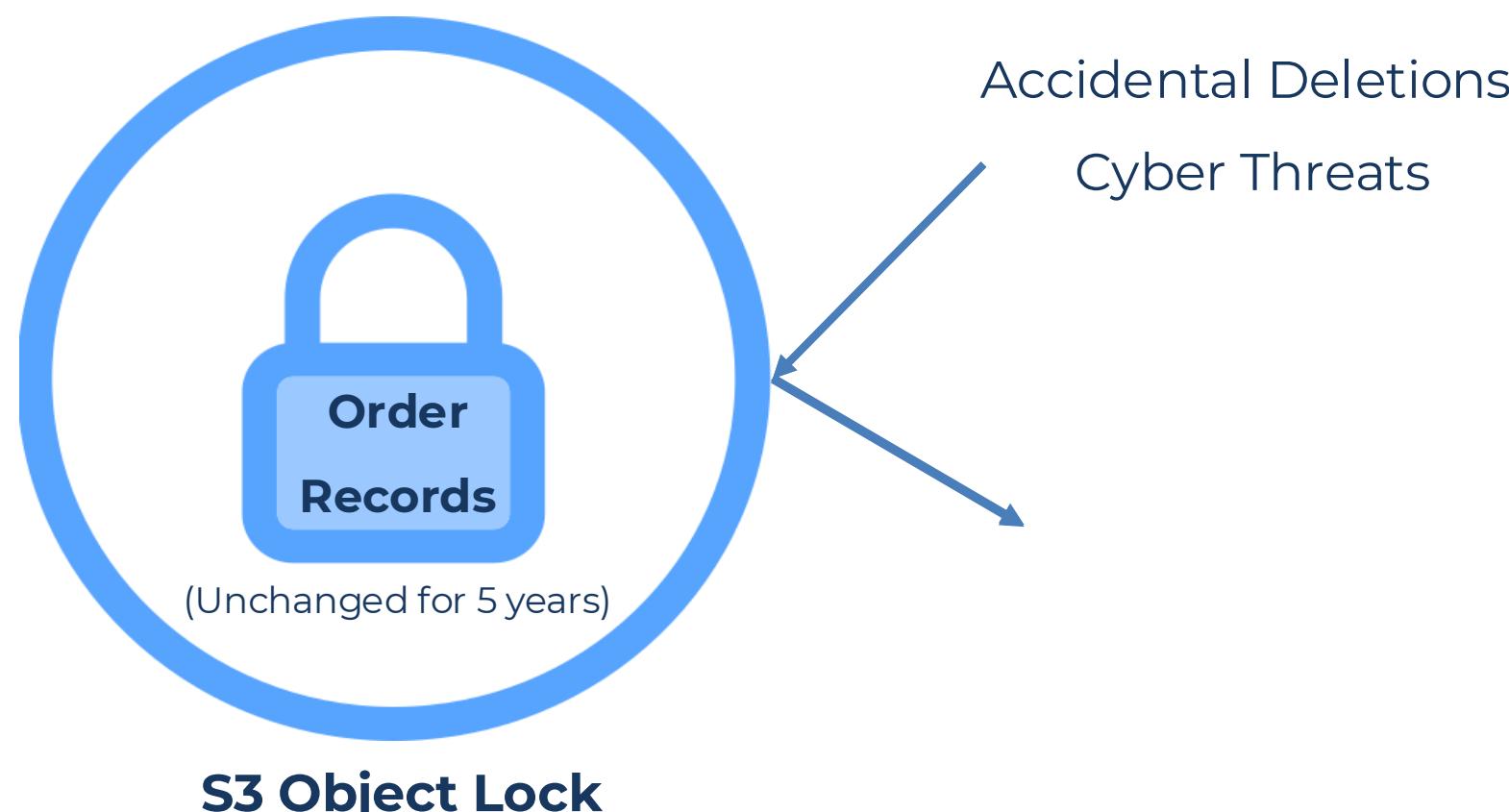


S3

➤ Optimizing Storage Management in Amazon S3

□ S3 Object Lock – Prevent Accidental or Malicious Deletion

- ✓ S3 Object Lock ensures regulatory compliance & data integrity
- ✓ Stores objects in **Write-Once-Read-Many (WORM)** format
- ✓ Prevents deletion or modification for a set duration or indefinitely



S3

➤ Optimizing Storage Management in Amazon S3

□ S3 Replication – Ensuring Data Redundancy and Low Latency

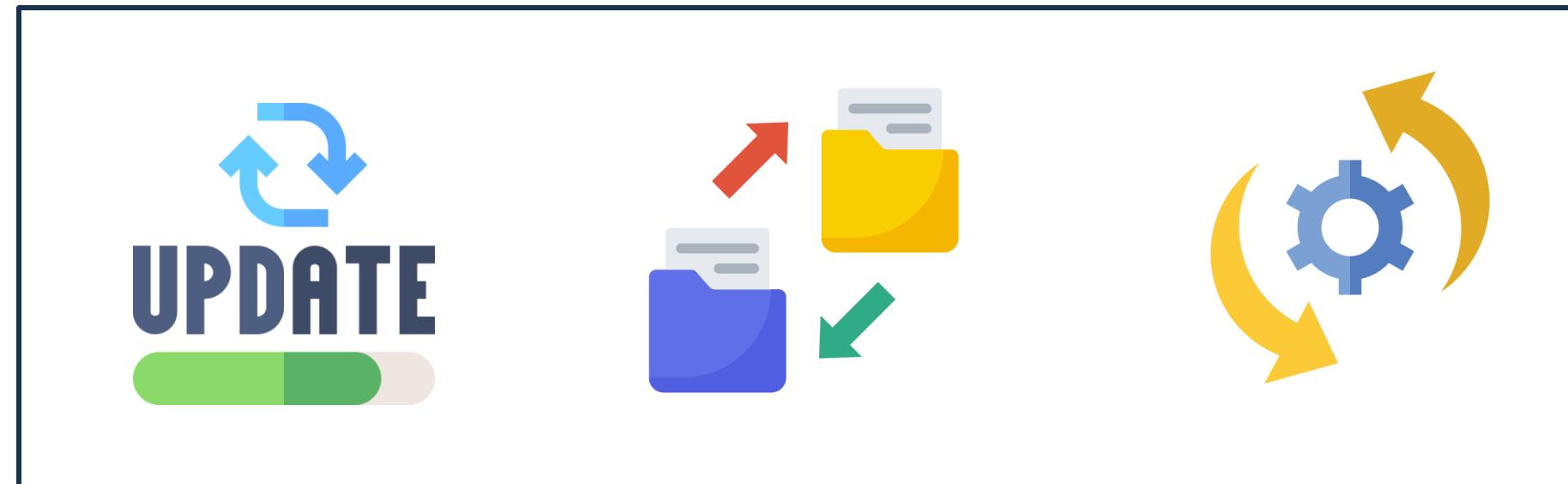
- ✓ Helps to copy objects from one S3 bucket to another
- ✓ Within the same AWS region or across different regions
- ✓ Useful for -
 - **Disaster Recovery** – Replicate critical files in another region
 - **Low-Latency Access** – Store data closer to users for faster access
 - **Compliance** – Ensure data redundancy for regulatory needs



S3

➤ Optimizing Storage Management in Amazon S3

□ S3 Batch Operations – Managing Large-Scale Data Efficiently



S3 Batch Operations – Perform bulk actions with a single request

Use Cases

Copy for Audits – Move archived documents to Standard storage

Invoke Lambda – Trigger functions to process datasets

Restore Files – Retrieve millions from Glacier for audits

S3

□ Securing and Managing Access in Amazon S3



Amazon S3



Security



Access Management of Data



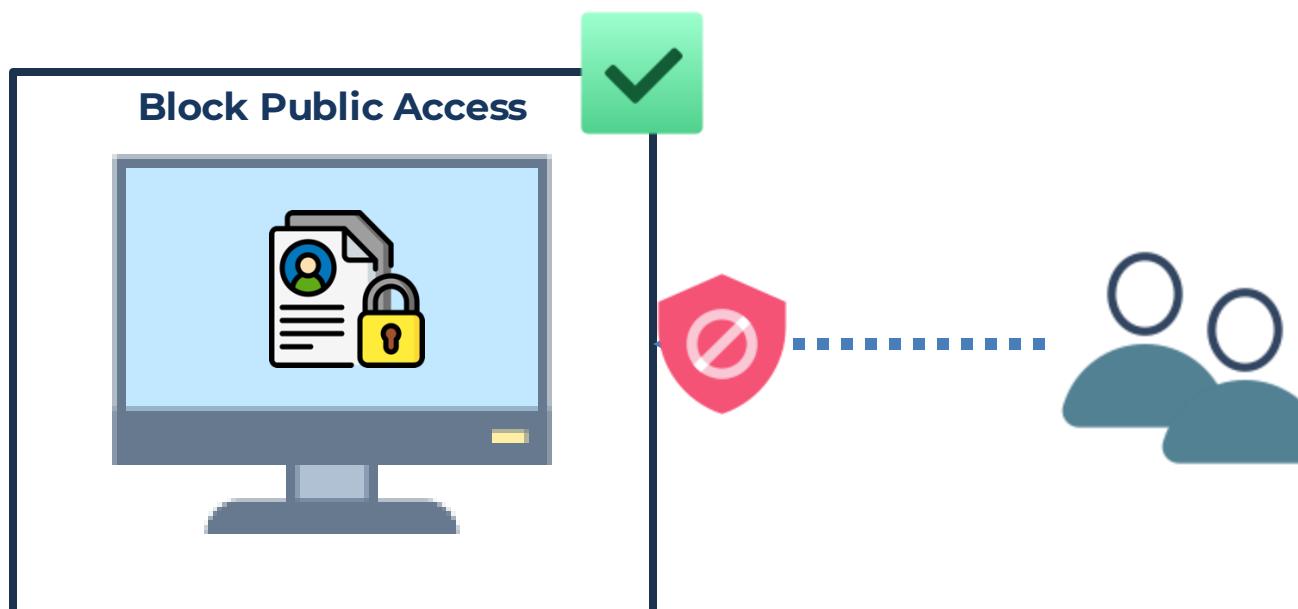
- ✓ By default, S3 buckets and objects are private, accessible only to the creator
- ✓ S3 offers tools to audit, restrict, and manage permissions securely

S3

➤ Securing and Managing Access in Amazon S3

□ Blocking Public Access – Keeping Data Secure

- ✓ Accidental public exposure can cause security breaches and compliance issues
- ✓ S3 Block Public Access feature keeps buckets and objects private by default

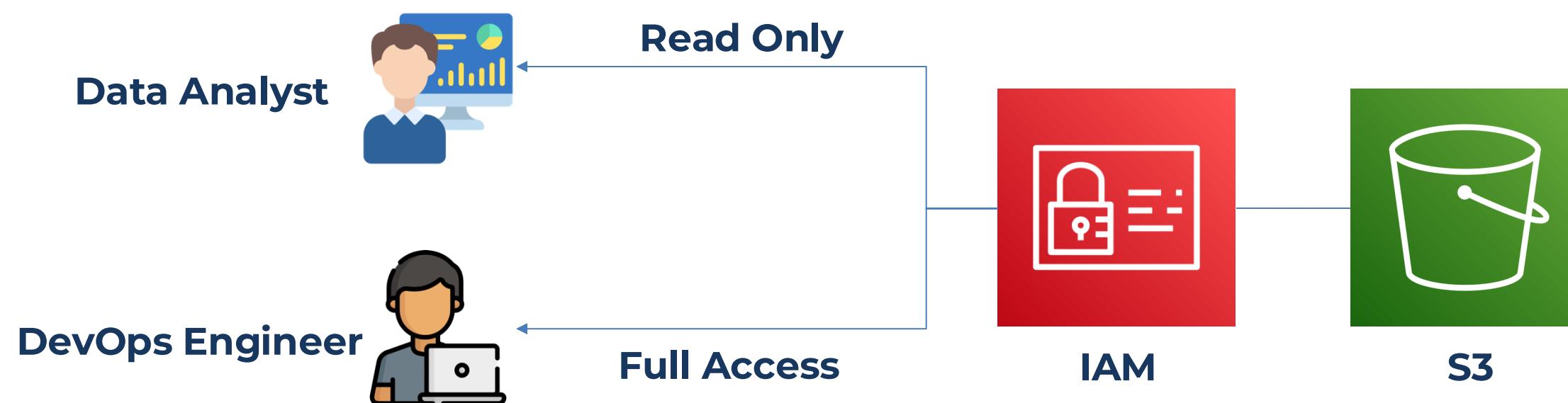


S3

➤ Securing and Managing Access in Amazon S3

□ IAM – Controlling Access with User Roles and Policies

- ✓ IAM defines who can access S3 resources and what actions they can perform
- ✓ Controls access based on job roles by assigning user roles and permissions



S3

➤ Securing and Managing Access in Amazon S3

□ Bucket Policies – Fine-Tuning Access at the Resource Level

- ✓ Control permissions directly at the S3 bucket level
- ✓ Useful for setting up organization-wide rules for accessing S3 resources

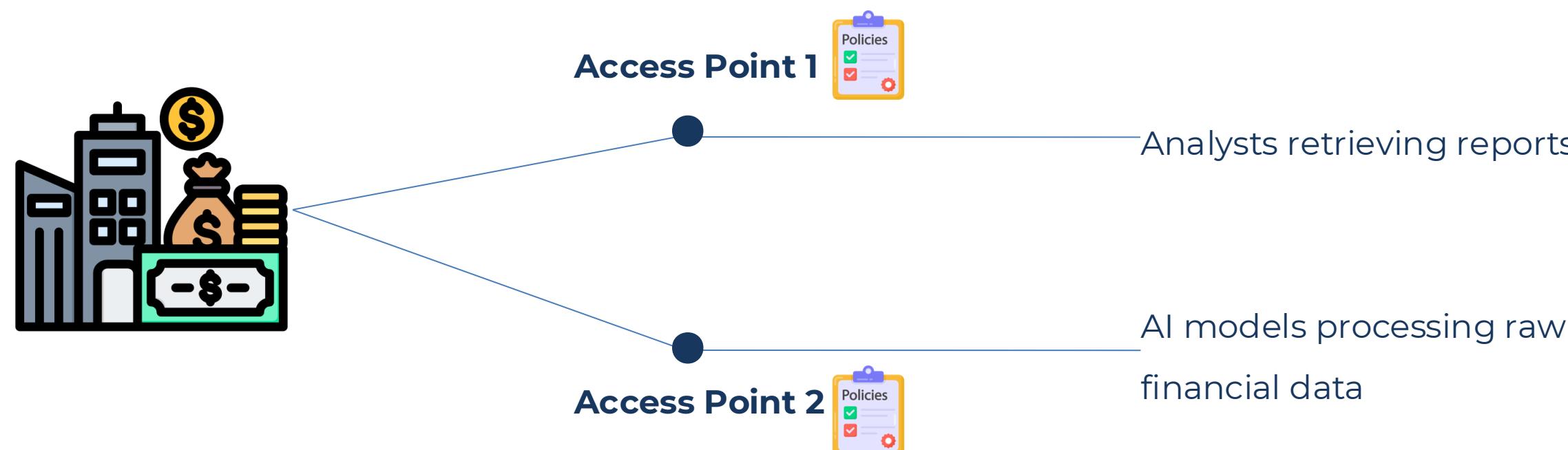


S3

➤ Securing and Managing Access in Amazon S3

□ S3 Access Points – Simplifying Access for Large-Scale Data

- ✓ S3 Access points control team access via named endpoints



S3

➤ Securing and Managing Access in Amazon S3

□ ACLs – Legacy Access Control for Specific Use Cases

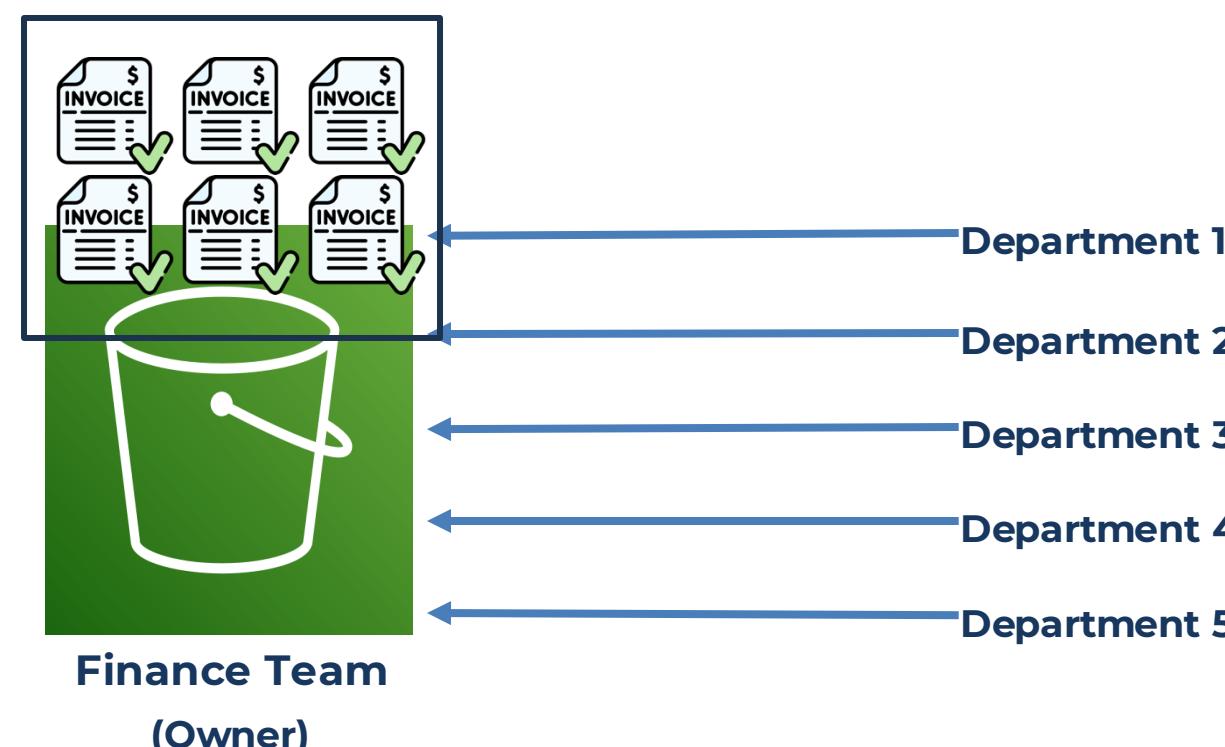
- ✓ Allow granting read and write permissions at a granular level
- ✓ Not recommended for modern access management
- ✓ IAM policies and bucket policies offer a simpler and more flexible approach
- ✓ Use Cases -
 - **Shared Ownership:** Multiple AWS accounts upload objects
 - **Third-Party Access:** Specific file permissions needed

S3

➤ Securing and Managing Access in Amazon S3

□ S3 Object Ownership – Ensuring Centralized Control

- ✓ Ensures the bucket owner has full control over all objects
- ✓ Useful when,
 - Multiple users or accounts upload data to a shared bucket
 - Disable ACLs and manage access via IAM or bucket policies



S3

➤ Securing and Managing Access in Amazon S3

□ IAM Access Analyzer – Monitoring and Auditing Access Policies

- ✓ IAM Access Analyzer for S3 helps organizations: the bucket owner has full control over all objects
 - Audit bucket policies to identify security risks
 - Detect public access or overly permissive sharing
 - Ensure compliance with internal and regulatory security requirements



S3

➤ Strong Consistency

❑ Operators:

- ✓ PUT
- ✓ DELETE
- ✓ GET (Read)

❑ Applies to:

- ✓ S3 Select
- ✓ ACLs
- ✓ Object Tags
- ✓ Metadata

❑ Benefits:

- ✓ Guarantees accurate and immediate access to your data across all AWS Regions

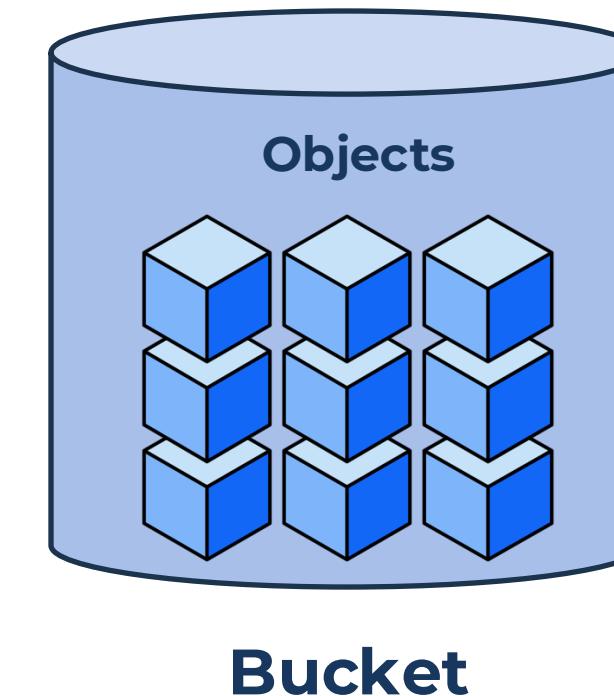
S3

➤ S3 Seamless Integration with AWS Services and Beyond

Features	Description
Lambda	Automate workflows without extra compute resources
Athena or Redshift Spectrum	Enables direct analysis of data in S3
S3 Object Lambda	Customize GET, HEAD, LIST requests (e.g., filter, redact, resize)
S3 Resource Change	Trigger workflows using SNS, SQS, or Lambda on data changes
Connects with vast partner ecosystem	Built-in support for backup, disaster recovery, and data lakes

S3

➤ Understanding Amazon S3 Structure and Organization



- ✓ AWS account has a default quota of 10,000 general-purpose buckets
- ✓ Each object includes data and metadata like last modified date, content type, and custom attributes

S3

➤ Understanding Amazon S3 Structure and Organization

- ✓ **Unique key (or key name)**
- ✓ If versioning is enabled, each object gets a unique **Version ID**
- ✓ Objects can be uniquely identified using **Bucket Name + Object Key + Version ID**



S3

➤ Understanding Amazon S3 Structure and Organization

- ✓ Amazon S3 supports objects up to **5 TiB**
- ✓ For files over **100 MB**, multipart upload boosts speed and reliability by uploading parts in parallel
- ✓ S3 uses a **region-based** storage model
- ✓ Bucket names must **be globally unique**
- ✓ Data is stored and accessed within **specific regions**
- ✓ Designed for scalable, organized, and efficient cloud storage



Demonstration | Use Case Explanation and S3 Bucket Creation

S3

➤ Autoscaling with Launch Templates in Public Subnets

- ✓ Previously, we understood how we can leverage load balancers to provide a single static URL
 - No impact on the URL despite backend server or IP address change
 - Manual addition to the target group
- ✓ Some common situations
 - Upgrading application to the newer version
 - Rollback or degrade to the previous version
 - Adding more servers to handle traffic spike
 - Reducing servers during low traffic
- ✓ Manually, it becomes inefficient and error prone



Demonstration | Creating Policy & Role



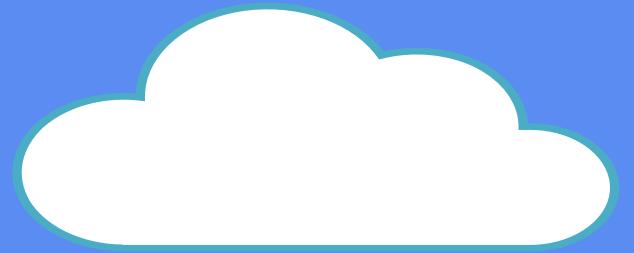
Demonstration | Update LT with IAM Profile & Upgrade ASG

Summary

- Amazon S3 and its key features
- Created an S3 bucket
- Configured IAM policies and roles
- Integrated our application with S3 using launch templates

Cloud Fundamentals with AWS

Section 10

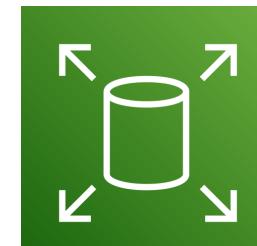


RDS

- ❖ Amazon RDS, its benefits, and key concepts
- ❖ Hands-on Demos
 - Create a PostgreSQL instance
 - Configure access
 - Manage backups
 - Perform instance upgrades
 - Restore from Snapshots
 - Delete databases
 - Update our launch template to connect our application to the database

Introduction to RDS

Introduction to RDS



EBS

- ✓ **EBS** provides block storage for a single EC2 instance

✗ Built-in querying

✗ Indexing

✗ Transactional capabilities



EFS

- ✓ **EFS** enables shared file storage



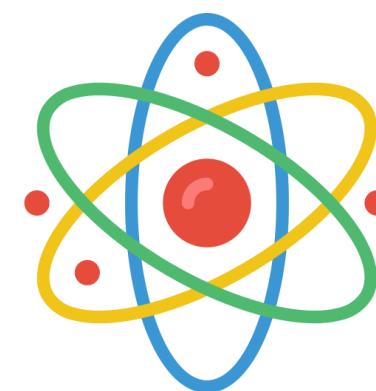
S3

- ✓ **S3** offers scalable object storage



Introduction to RDS

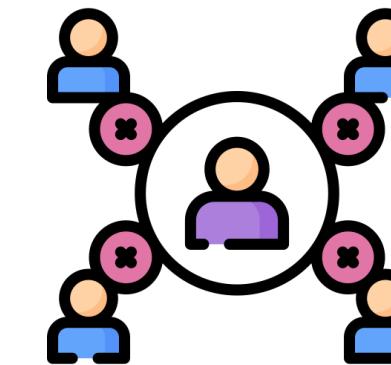
- ✓ Databases help manage structured data efficiently
- ✓ Enforce relationships
- ✓ Ensure consistency



Atomicity



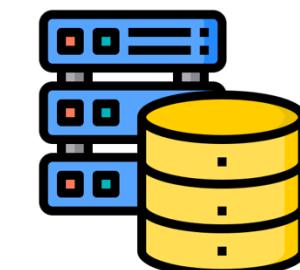
Consistency



Isolation



Durability



- ✓ Fast lookups
- ✓ Complex queries
- ✓ Real-time data processing

Introduction to RDS

- ✓ **Distributed Applications:** Modern apps require different database types for scalability & performance

➤ AWS Databases and Their Purpose

- ✓ **AWS Purpose-Built Databases:** Relational, key-value, document, in-memory, graph, time-series, ledger
- ✓ **Optimized for Use Cases:** Each database type enhances speed, efficiency, and innovation

□ Challenges with Traditional Databases

- ✓ Struggle with scalability, high availability, and cost

AWS Database Benefits:

- | | |
|---|--|
| ✓ Serverless, auto-scaling, self-healing | ✓ Eliminates restrictive licensing |
| ✓ Handles millions of requests per second | ✓ Fully managed solutions for developers |
| ✓ 3-5x better performance | |

Introduction to RDS

➤ AWS Database Services

RELATIONAL

Amazon RDS



Amazon Aurora



- Strong consistency
- Schema enforcement

Database Services

NON-RELATIONAL



Key-value
Amazon DynamoDB



Document
Amazon DocumentDB



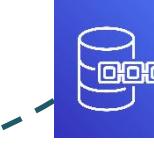
Memory
Amazon ElastiCache



Graph
Amazon Neptune



Time-series
Amazon Timestream



Ledger
Amazon QLDB

- ✓ Handle large-scale workloads
- ✓ Store flexible JSON data structures
- ✓ Lightning-fast responses
- ✓ Highly connected datasets
- ✓ Managing time-based data
- ✓ Ensure immutable transaction records with cryptographic verification

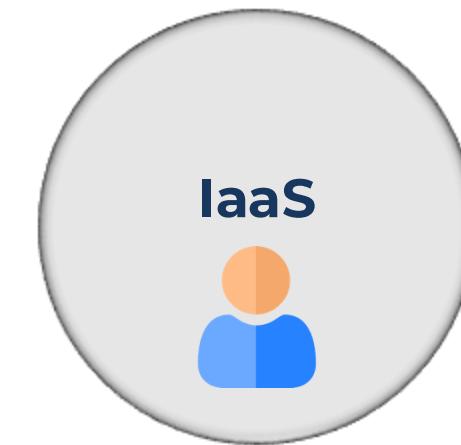
Introduction to RDS

➤ Understanding Amazon RDS as a PaaS Solution

- ✓ RDS is a fully managed cloud-based solution designed to simplify the deployment, management, and scaling of relational databases.

Cost-effective

Flexible infrastructure



- ✓ It abstracts the complexities of infrastructure management
- ✓ Provides a ready-to-use environment for deploying and managing databases
- ✓ RDS automates installation, configuration, and maintenance
- ✓ Including backups, patching, scaling, and failover

Users must **install, configure, and maintain** their own database software on virtual machines

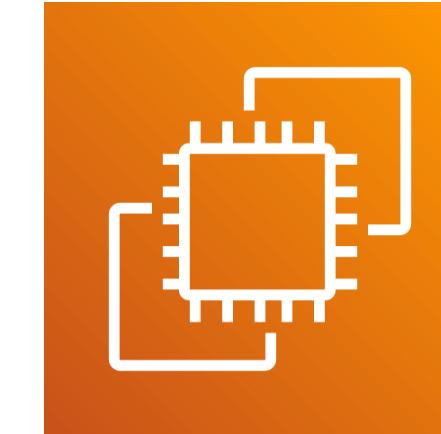
Introduction to RDS

➤ Understanding Amazon RDS as a PaaS Solution

- Three common database management approaches



On-premises



Amazon EC2



Amazon RDS

Introduction to RDS

➤ Understanding Amazon RDS as a PaaS Solution

□ On-premises

- **Full Responsibility:** Manage hardware, OS, database software, backups, security, and scaling
- **High Maintenance:** Requires regular updates, patches, and monitoring
- **Manual Scaling:** Expensive and time-consuming capacity planning

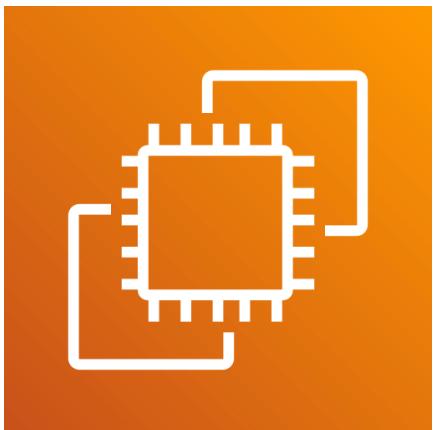


Introduction to RDS

➤ Understanding Amazon RDS as a PaaS Solution

□ Amazon EC2

- **No Physical Hardware Management:** Eliminates on-premise infrastructure needs
- **Manual Maintenance Required:** Installation, backups, patching, and scaling must be handled manually
- **Higher Risk of Misconfigurations:** Increased chances of setup errors
- **Performance Challenges:** Requires tuning and monitoring for efficiency
- **Downtime Risks:** Failures may lead to extended outages without automated recovery



Introduction to RDS

➤ Understanding Amazon RDS as a PaaS Solution

□ Amazon RDS

- **Automated Management:** Handles backups, patching, scaling, and failover automatically
- **High Availability:** Ensures minimal downtime with multi-AZ deployments
- **Security Integration:** Supports AWS IAM and VPC for streamlined access control
- **Focus on Applications:** Reduces administrative overhead, allowing businesses to optimize applications



Introduction to RDS

➤ Key Benefits of Amazon RDS



- Manual database maintenance
- Ensuring reliability
- Scalability
- Security

Introduction to RDS

➤ Key Benefits of Amazon RDS



Amazon RDS

- ✓ Supports multiple familiar database engines

Engine options

Engine type [Info](#)

<input type="radio"/> Aurora (MySQL Compatible) 	<input checked="" type="radio"/> Aurora (PostgreSQL Compatible) 
<input type="radio"/> MySQL 	<input type="radio"/> PostgreSQL 
<input type="radio"/> MariaDB 	<input type="radio"/> Oracle 
<input type="radio"/> Microsoft SQL Server 	<input type="radio"/> IBM Db2 

Introduction to RDS

➤ Key Benefits of Amazon RDS



Amazon RDS

- ✓ **Automation:** Handles software updates, backups, and failure recovery

Backup & Recovery:

- ✓ Automatic backups for point-in-time recovery
- ✓ Manual snapshots for long-term retention

High Availability:

- ✓ Multi-AZ deployments ensure automatic failover
- ✓ Read replicas enhance performance by distributing workloads

Introduction to RDS

➤ Key Benefits of Amazon RDS



Amazon RDS

Security:

- ✓ Integration with AWS IAM for access control
- ✓ Encryption for data at rest and in transit
- ✓ VPC support for controlled network access

Business Benefits:

- ✓ Reduces manual administration, optimizes costs, and enhances database performance and reliability

Introduction to RDS

➤ Amazon RDS Parameter Groups

RDS > Parameter groups > Create parameter group

Create parameter group

Parameter group details
To create a parameter group, choose a parameter group family, then name and describe your parameter group

Parameter group family
DB family that this DB parameter group will apply to

Group name
Identifier for the DB parameter group

Description
Description for the DB parameter group

Cancel **Create**

Memory allocation

Query optimization

Connection limits



- ✓ Default parameter group
- ✓ Cannot be modified

Introduction to RDS

➤ Amazon RDS Parameter Groups

Custom Configurations:

- ✓ Create a custom parameter group for specific DB settings

Immediate vs. Reboot Changes:

- ✓ Dynamic parameters apply immediately
- ✓ Static parameters require a reboot

Optimization & Consistency:

Helps fine-tune performance and ensure uniform settings across instances

Testing Best Practices:

Always test changes before applying them to production

Introduction to RDS

➤ Amazon RDS DB Instance Overview

- ✓ Amazon RDS DB instance is the core building block of RDS



Introduction to RDS

➤ Amazon RDS DB Instance Overview

□ DB Engines

- ✓ Amazon RDS supports multiple database engines, each its own unique features

IBM Db2

IBM Db2

MySQL



MariaDB



Oracle

ORACLE®

Microsoft SQL Server



PostgreSQL



- Windows-based enterprise applications

- Extensibility and advanced features

Introduction to RDS

➤ Amazon RDS DB Instance Overview

□ Instance Types

CPU

Memory

Network

- **General-purpose**
 - ✓ Balanced combination of compute, memory, and network
- **Memory-optimized**
 - ✓ High-speed memory access
- **Compute-optimized**
 - ✓ CPU-intensive applications
- **Burstable performance**
 - ✓ Occasional high CPU usage

Introduction to RDS

➤ Amazon RDS DB Instance Overview

□ Instance Storage Types

- **General-purpose SSD**

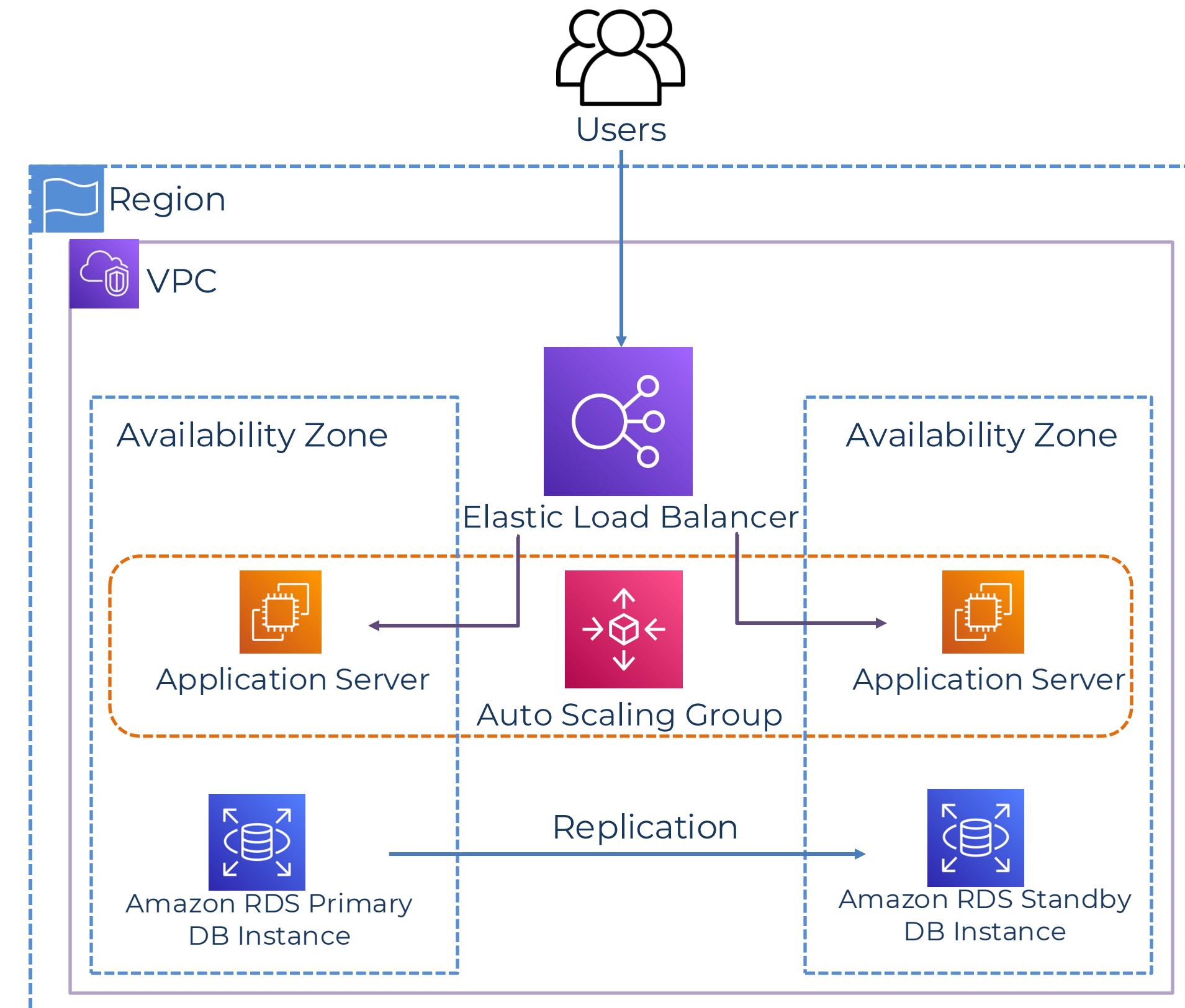
- ✓ Cost-effective and suitable for most workloads

- **Provisioned IOPS**

- ✓ High-performance applications with heavy read and write requirements

Introduction to RDS

➤ Amazon RDS in a Scalable Architecture





Demonstration | Creating Database Instance - Postgres



Demonstration | Accessing the Postgres Database using Postgres User



Demonstration | Creating Snapshot



Demonstration | Enabling Automated Backups



Demonstration | Upgrading Database Instance

RDS

➤ Upgrading Database Instance

- ✓ Previously, we enabled automated backups in Amazon RDS
- ✓ Backups are retained for 7 days by default, but can be increased
- ✓ Automated backups protect data and also allow easy restoration
- ✓ Amazon RDS is a PaaS offering
- ✓ RDS enables easy database upgrades with minimal downtime



Demonstration | Creating Database from Snapshot

RDS

Snapshot Creation



Automated Backups



Database Upgrade





Demonstration | Deleting Database Instance



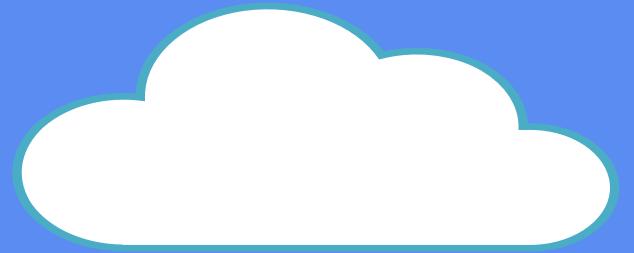
Demonstration | Use Case: Update LT with IAM Profile & Upgrade ASG

Summary

- Amazon RDS, including its benefits, and key concepts
- Explored setting up a PostgreSQL instance, managing backups, restoring from snapshots, upgrading databases, and scaling with Auto Scaling Groups
- Integrating PostgreSQL

Cloud Fundamentals with AWS

Section 11



Simple Notification Service

- ❖ Amazon SNS
- ❖ Create a topic and integrate SNS
- ❖ Update the IAM role
- ❖ Refresh the Launch Template
- ❖ Update our Auto Scaling Group

Introduction to SNS

Introduction to SNS



E-Commerce



E-mail



SMS



Mobile
notifications

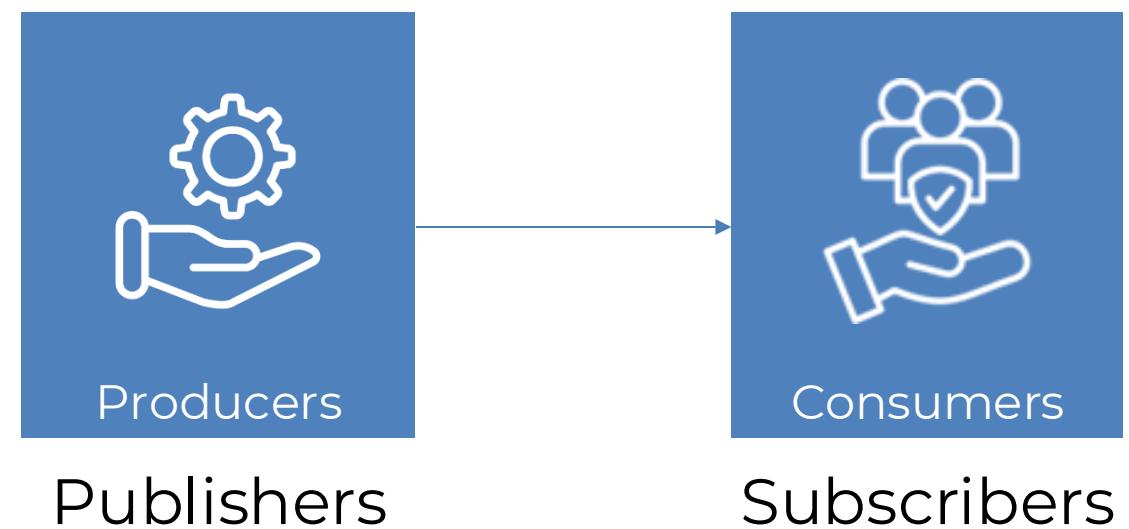
- ✓ Your order has been received
and is being processed

SNS

Introduction to SNS

➤ What is Amazon SNS?

- ✓ Fully managed messaging service from AWS
- ✓ Makes easy to coordinate communication between different systems and users
- ✓ Allows you to send messages from



- ✓ Publish-Subscribe Model

Introduction to SNS

➤ Amazon SNS Messaging Patterns

Application-to-Application (A2A)

- ✓ Messaging allows SNS to integrate different applications or services

SQS
Queues

For order
processing

AWS
Lambda

To update
inventory

Analytics
Pipelines

Decouple and
automate
microservices
communication

Introduction to SNS

➤ Amazon SNS Messaging Patterns

Application-to-Person (A2P)

- ✓ Messaging enables SNS to send notifications directly to people

SMS Alerts

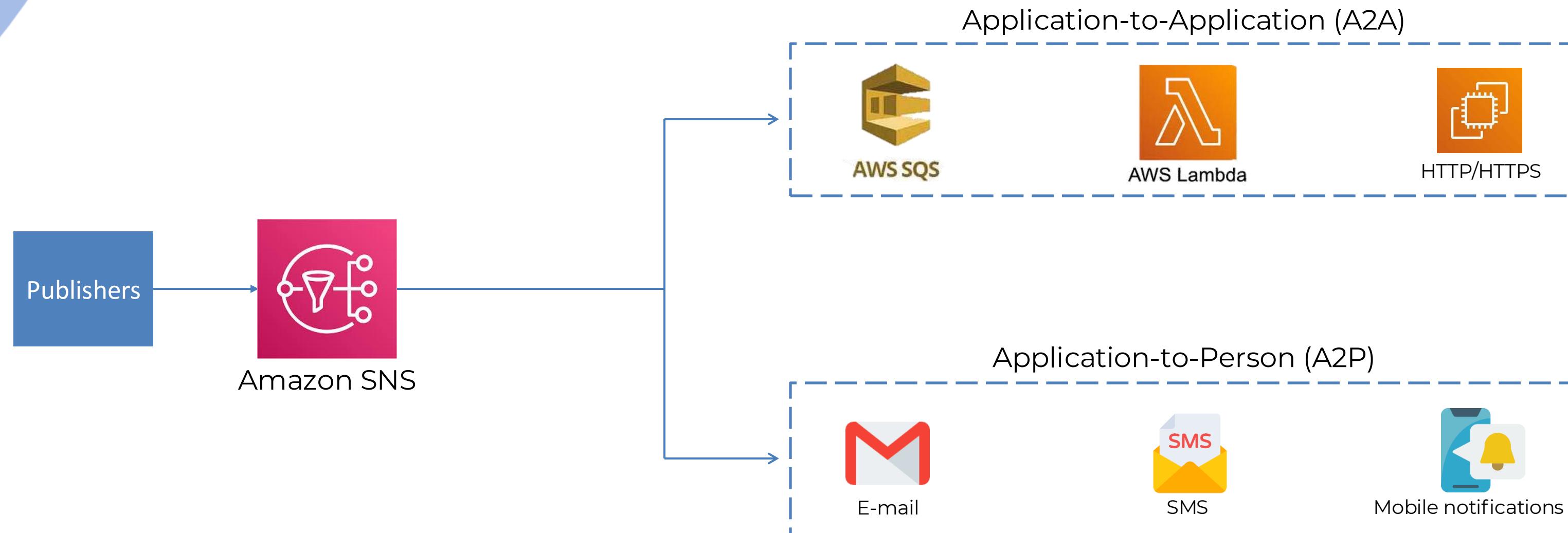
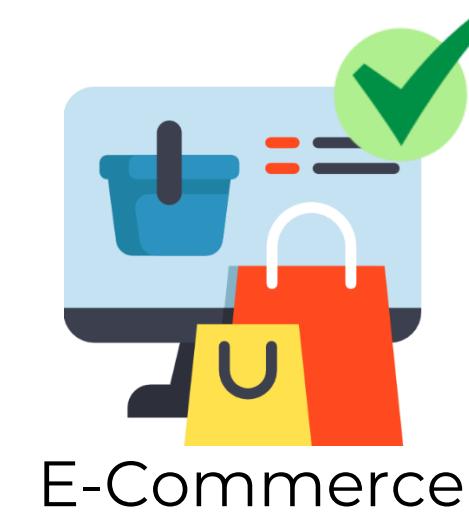
Email Confirmations

Mobile Push Notifications

Customer engagement, Alerts, and Real-time updates

Introduction to SNS

➤ How SNS Works



Introduction to SNS

➤ Key Features of Amazon SNS

A2A & A2P

Fan-Out
Architecture

Message
Filtering

FIFO Topics

Durable
Message
Delivery

Secure

- Dead-letter queues
- AWS Key Management Service encryption
- PrivateLink
- Resource policies

Introduction to SNS

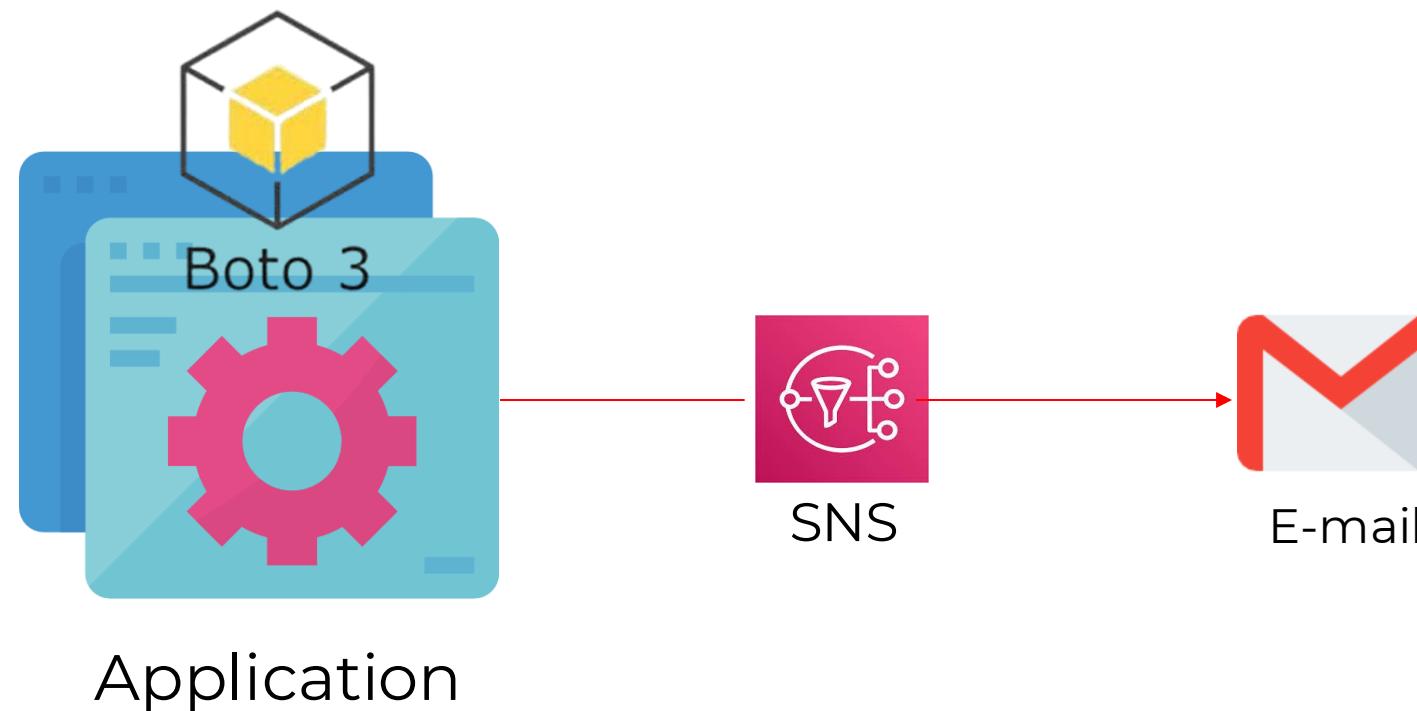
➤ Boto3

- ✓ AWS SDK for Python
- ✓ Allows us to create, configure, and manage AWS services through Python code
- ✓ Provides an easy way to interact with services like EC2, S3, and, in our case, Amazon SNS



Introduction to SNS

➤ Boto3



```
# Initialize the SNS client
sns_client = boto3.client(
    'sns',
    region_name=os.getenv('AWS_REGION') # Optional, depending on your region
)
```

Create an SNS client **with `boto3.client('sns')`**

```
# Publish the message to SNS
sns_client.publish(
    TopicArn=SNS_TOPIC_ARN,
    Message=sns_message_body,
    Subject=sns_subject
)
```

Use the **publish method** to send messages to an SNS topic

Introduction to SNS

- Three important parameters with the publish method

Request Syntax

```
response = client.publish(  
    TopicArn='string',  
    TargetArn='string',  
    PhoneNumber='string',  
    Message='string',  
    Subject='string',  
    MessageStructure='string',  
    MessageAttributes={  
        'string': {  
            'DataType': 'string',  
            'StringValue': 'string',  
            'BinaryValue': b'bytes'  
        }  
    },  
    MessageDeduplicationId='string',  
    MessageGroupId='string'  
)
```

Unique identifier of the SNS topic

Actual content of the email body

Becomes the subject line of the
email sent to the recipient



Demonstration | Creating SNS Topic & Add Subscription



Demonstration | Use Case Explanation & Code Merge

Simple Notification Service

```
def send sns notification(user_name, user_position, resume_url, user_experience, user_ctc, user_expected_ctc, user_phone_number):
    """
    Sends a notification to an SNS Topic with the provided user details.
    """

    # Construct the SNS message body
    sns_message_body = f"A new profile has been uploaded. You can view the profile at: {resume_url}\n\nDetails:\n" \
        f"Name: {user_name}\nPosition Applied: {user_position}\nExperience: {user_experience}\nCTC: {user_ctc}\n" \
        f"Expected CTC: {user_expected_ctc}\nPhone Number: {user_phone_number}"

    # Construct the SNS message subject
    sns_subject = f"{user_name} has applied for {user_position}"

    try:
        # Publish the message to SNS
        sns_client.publish(
            TopicArn=SNS_TOPIC_ARN,
            Message=sns_message_body,
            Subject=sns_subject
        )
        print(f"Notification sent to SNS for {user_name} applying for {user_position}")
    
```



Demonstration | Update IAM Role to Access SNS Topic



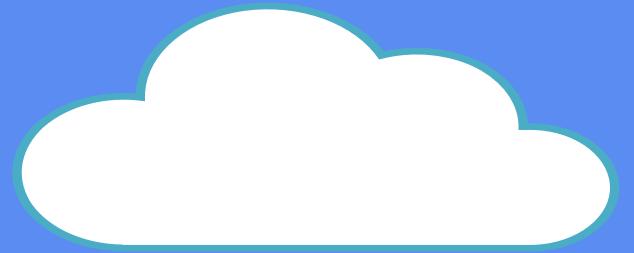
Demonstration | Update LT & Run Autoscaling Group Instance Refresh

Summary

- Created an SNS topic & connected it to our application
- Updated IAM role to allow app to publish to SNS
- Refreshed the LT & updated our Auto Scaling Group

Cloud Fundamentals with AWS

Section 12



VPC

- ❖ What is VPC
- ❖ Why it matters
- ❖ Key Components
- ❖ Hands-on Demos
 - Create a VPC
 - Deploy our application within the VPC

Introduction to VPC

Introduction to VPC

- ✓ VPC is a core networking service in AWS
- ✓ Allows you to create a logically isolated virtual network within the 
- ✓ Enables you to design and control your own networking environment



- ✓ Each AWS Region includes a default VPC when you first start using AWS
- ✓ Pre-configured setup for quick and easy resource deployment
- ✓ Simplifies launching services like EC2 instances and RDS databases
- ✓ Requires no complex networking configuration initially

Demerit

- ❑ Default VPC may not meet long-term architectural needs & security requirements

Introduction to VPC

➤ Why Create a Custom VPC?

1. Enhanced control over network configuration and security
2. Improved isolation for different applications or environments
3. Better compliance with organizational policies and industry standards
4. Optimized performance by structuring the network to fit specific workloads

Introduction to VPC

➤ Multiple VPCs for Greater Security and Flexibility



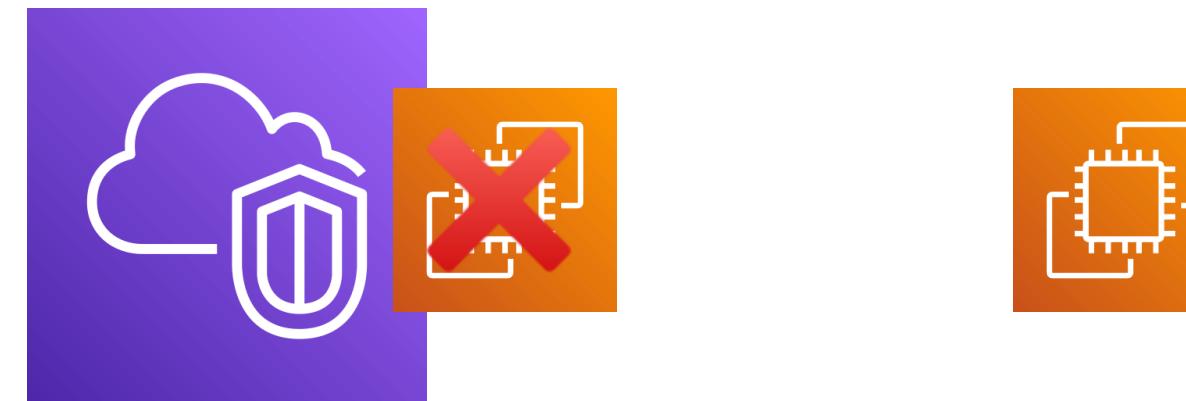
Security

Isolation

1. Apply distinct security policies to different environments
2. Implement customized access controls for individual teams or projects
3. Configure independent routing rules to manage traffic flow efficiently

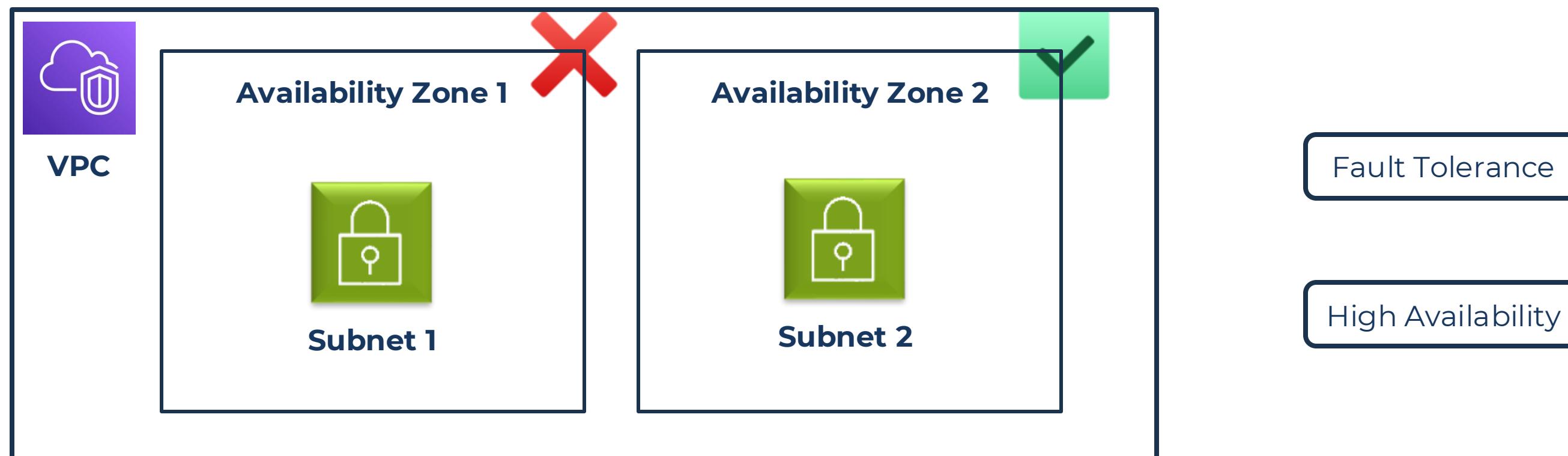
Introduction to VPC

➤ Understanding Subnets in a VPC



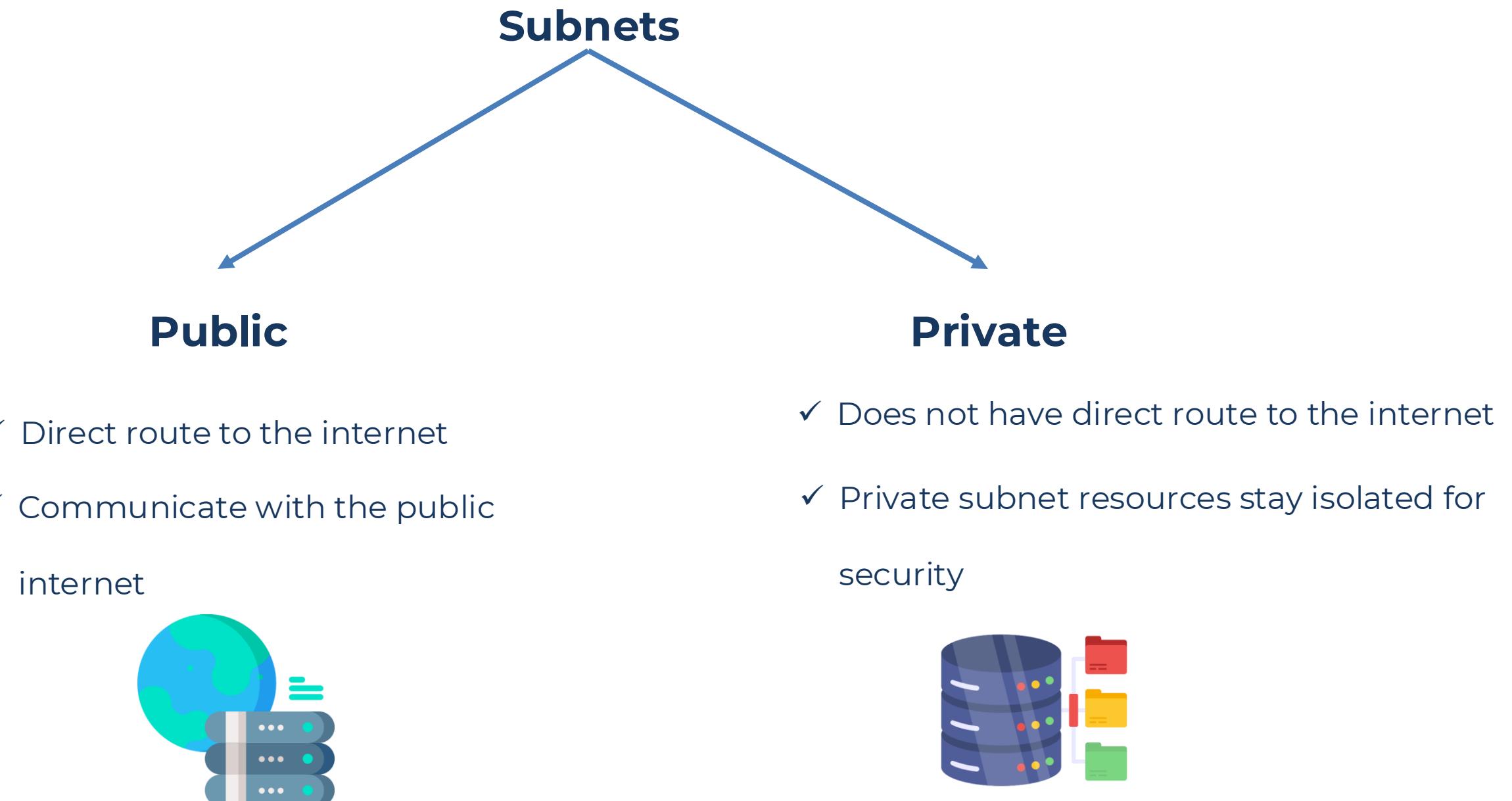
Subnet is a network within a network

Each subnet helps organize and manage network traffic more efficiently



Introduction to VPC

➤ Understanding Subnets in a VPC



Introduction to VPC

➤ VPC Creation

- ✓ When you create a new VPC, AWS sets up three things by default:
 - ❑ Route table
 - ❑ Network Access Control List, or NACL
 - ❑ Security group

Introduction to VPC

➤ Route Table and the Local Route



VPC

Route Table	
Example	
Destination	Target
10.10.0.0/16	local

- ✓ Local route that allows communication between all resources inside the VPC

1. Each subnet in a VPC must be associated with a route table
2. If not specified, AWS links it to the main route table by default
3. The route table determines how traffic flows in and out of the subnet

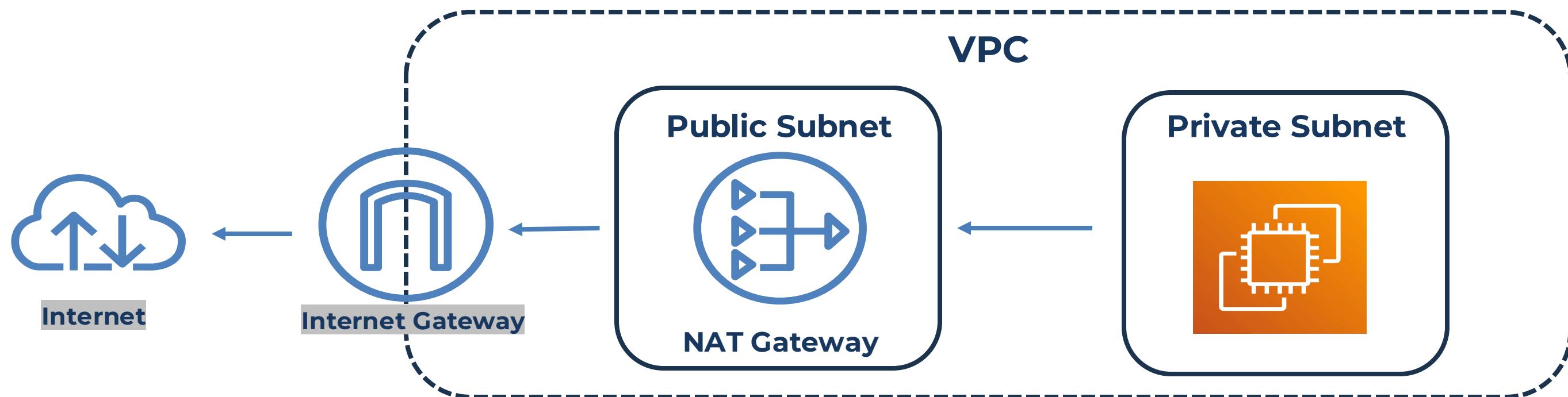
Introduction to VPC

➤ NACL versus Security Group

Aspect	Network ACL (NACL)	Security Group (SG)
Level	Subnet Level	Instance Level
Statefulness	Stateless – return traffic must be explicitly allowed	Stateful – return traffic is automatically allowed
Rules	Supports Allow & Deny rules	Supports Allow rules only
Evaluation	Rules evaluated in order (by rule number)	Rules evaluated as a whole
Use Case	Fine-grained subnet-level control (extra layers of security)	Simplified instance-level security

Introduction to VPC

➤ Internet Gateway and NAT Gateway



Public Subnet

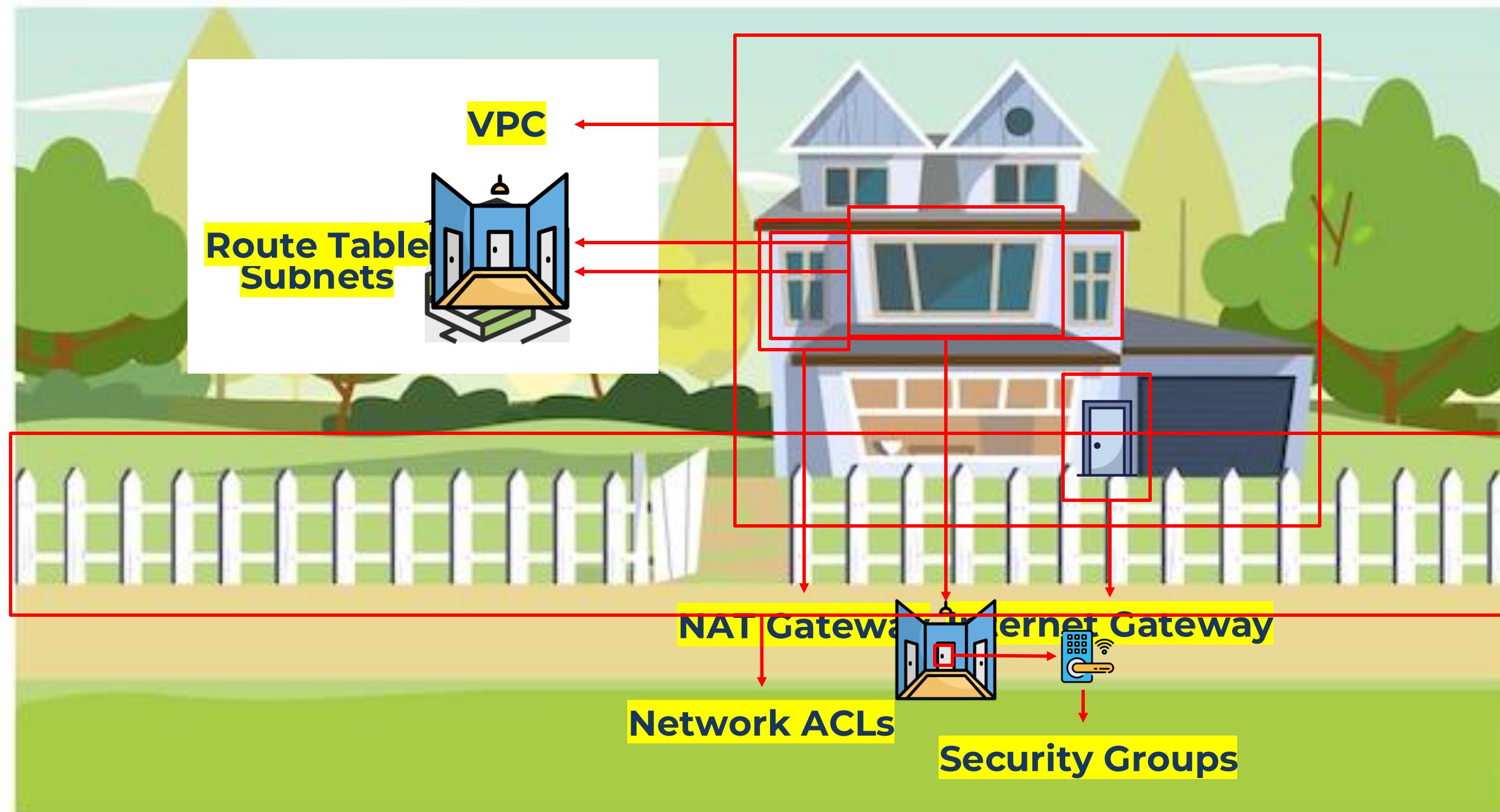
- ❑ Internet access is enabled by attaching an Internet Gateway (IGW) to the VPC
- ❑ The route table must send internet-bound traffic to the IGW
- ❑ Used for instances like web servers that need inbound and outbound internet access

Private Subnet

- ❑ No direct access to the internet
- ❑ For outbound traffic, use a NAT Gateway
- ❑ NAT Gateway sits in a public subnet, connected to the Internet Gateway
- ❑ Blocks unsolicited inbound connections

Introduction to VPC

➤ Understanding VPC through Analogy



Introduction to VPC

➤ Understanding CIDR Blocks

□ IP Address

- ✓ An **IPv4 address** consists of **32 bit**
- ✓ Grouped into **4 sections** of 8 bits each called **octets**
- ✓ Octets are separated by dots (e.g., 192.168.10.1)
- ✓ Each number (192, 168, 10, 1) represents 1 octet = **8 bits**
- ✓ In binary, 192.168.10.1 becomes: 11000000 . 10101000 . 00001010 . 00000001
- ✓ This 32-bit structure allows for a total of **2^{32} = over 4.29 billion** unique IP addresses globally

Introduction to VPC

➤ Understanding CIDR Blocks

□ CIDR

- ✓ CIDR stands for **Classless Inter-Domain Routing**
- ✓ Uses **slash** notation (e.g., /16, /24) to represent IP ranges
- ✓ The number after slash represents network bits
- ✓ For Example: 10.0.0.0/16 means the **first 16 bits** are fixed for the network

The remaining 16 bits can vary, allowing for **$2^{16} = 65,536$ IP**

addresses

- ✓ CIDR helps define the size of a network or subnet

Introduction to VPC

➤ Understanding CIDR Blocks

□ CIDR Block in VPC

- ✓ When creating a VPC in AWS, you must specify a CIDR block
- ✓ Example: **172 . 31 . 0 . 0 / 16**
- ✓ This provides a pool of **65,536 IP addresses**
- ✓ These can be divided into smaller subnet CIDRs

Introduction to VPC

➤ Understanding CIDR Blocks

□ Choosing Subnet CIDR Ranges

- ✓ Suppose each subnet should have 256 IP addresses
- ✓ $2^8 = 256$, so you need **8 bits** for host addresses
- ✓ If your VPC is /16, then your subnet CIDR would be /24 (16 + 8)
- ✓ A valid subnet could be: 172 . 31 . 1 . 0 / 24
- ✓ This gives you **256 IPs** ranging from 172.31.1.0 to 172.31.1.255

Introduction to VPC

➤ Understanding CIDR Blocks

□ Reserved IPs in Subnets

- ✓ Not all IPs in a subnet are usable

IP Address	Purpose
First IP	Network Address (e.g., 172.31.1.0)
Second IP	Reserved by AWS
Third IP	Reserved for AWS DNS
Fourth IP	Reserved for future use
Last IP	Broadcast address (e.g., 172.31.1.255)

- ✓ From 256 IPs in a /24 subnet, you get **251 usable IP addresses**

Introduction to VPC

➤ Understanding CIDR Blocks

□ Best Practices for Choosing Subnet CIDRs

- ✓ **Plan for scale** – Don't make subnets too small if traffic may grow
- ✓ **Avoid overlap** – No two subnets should have overlapped CIDRs
- ✓ **Separate environments** – Use different CIDR blocks for public, private, and database subnets
- ✓ **Stick to power-of-two sizes** – CIDR is binary-based, so keep IP counts like 16, 32, 64, 128, 256, etc

Introduction to VPC

➤ Understanding CIDR Blocks

❑ Example Subnet Planning from a /16 VPC

✓ **VPC CIDR: 10 . 0 . 0 . 0 / 16**

✓ You could divide it like:

❑ **10 . 0 . 1 . 0 / 24** – Public Subnet A (AZ1)

❑ **10 . 0 . 2 . 0 / 24** – Public Subnet B (AZ2)

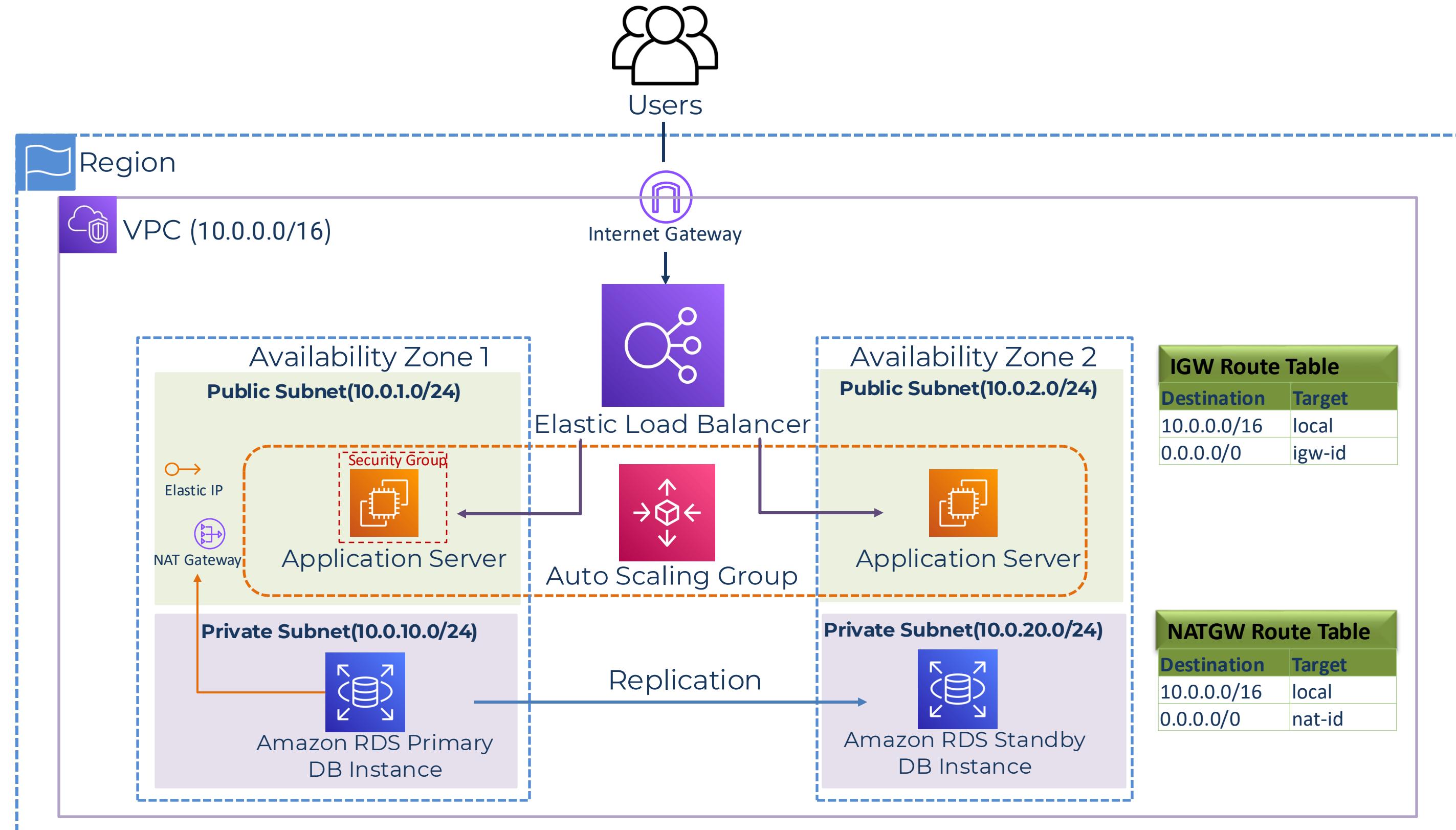
❑ **10 . 0 . 10 . 0 / 24** – Private Subnet for App Servers

❑ **10 . 0 . 20 . 0 / 24** – Private Subnet for Databases

✓ Each subnet has **251 usable IPs** and serves a clear purpose

Introduction to VPC

➤ Understanding the VPC and Its Core Components





Demonstration |Creating VPC



Demonstration | Creating Subnets - Public & Private



Demonstration | Creating Route Tables & Subnet Associations



Demonstration | Creating Internet Gateway & Attach with VPC



Demonstration | Adding Internet GW Route in Main Route Table



Demonstration | Create NAT Gateway & Add Route



Demonstration | Setting up Security Group for Sample Application



Demonstration | Create DB_Subnet_Group for RDS



Demonstration | Update Launch Template with Newly Created VPC

Summary

- VPC and its Key Components
- We created the VPC along with subnets, route tables, an internet gateway, and a NAT gateway
- We set up the security group, DB subnet group, and updated the launch templates to fit the new VPC setup

Cloud Fundamentals with AWS

Section 14



Route53 & Cert Manager

- ❖ Basics of Route 53 and ACM
- ❖ Set up a subdomain in Route 53
- ❖ Create an SSL certificate in ACM and link it to Route 53
- ❖ Attach the certificate to a Load Balancer
- ❖ Enable HTTPS access to the application

Introduction to Route 53

Introduction to Route 53

- ✓ A custom domain
- ✓ A reliable way to connect users to your website
- ✓ A system to monitor its availability



Route 53 is a highly available and scalable Domain Name System (DNS) web service from AWS

❑ Essential functions

Domain Registration

Traffic Routing

Health Monitoring



Introduction to Route 53

➤ Domain Registration

ecomartgreen.com

- **Route 53** console - no need to rely on third-party registrars



- ✓ Built-in health checks
- ✓ Monitors your resources continuously
- ✓ It automatically redirects users to a healthy backup - helping minimize downtime

Introduction to Route 53

➤ Route 53 Key Concepts

- **Domain:** ecomartgreen.com
- **Tool:** Amazon Route 53
- **Brand example:** EcoMartGreen

□ Domain name

- **Domain Name Definition:** Website address typed in browsers (e.g., ecomartgreen.com)
- **Website Identity:** Your online business identifier
- **Online Accessibility:** First step to making your business reachable online

Introduction to Route 53

➤ Route 53 Key Concepts

□ What is a domain registrar?

- ICANN-accredited company managing domain name reservation
- **Amazon Route 53** acts as a domain reseller
- Resells for accredited registrars (**e.g., Amazon Registrar, Gandi**)
- Allows direct domain registration (**e.g., ecomartgreen.com**) via Route 53 console
- Eliminates the need for a separate third-party registrar

Introduction to Route 53

➤ Route 53 Key Concepts

□ What is a domain registry?

- Owns & manages top-level domains (e.g., .com, .org)
- **Example:** VeriSign manages all **.com** domains
- Maintains master database & registration rules

Introduction to Route 53

➤ Route 53 Key Concepts

□ What are TLDs?

- **Generic TLDs** like **.com**, **.bike**, or **.shop**. These are unrestricted and available for anyone to use
- **Geographic TLDs** like **.us**, **.uk**, or **.tokyo**. These often have rules about who can register them

Introduction to Route 53

➤ Route 53 Key Concepts

□ What is DNS?

- The Internet's Phone Book
- Translates domain names into numerical IP addresses

ecomartgreen.com to **192.0.2.1**

- Enables computers to connect to websites

Introduction to Route 53

➤ Route 53 Key Concepts

□ Subdomains

- **Prefix to Domain:** Subdomains are prefixes added to your main domain name

blog.ecomartgreen.com
store.ecomartgreen.com



Subdomains

- **Deeper Levels:** You can create multiple levels of subdomains (**india.store.ecomartgreen.com**)
- Subdomains help organize your site into sections
- **DNS Record Required:** Routing traffic to a subdomain requires creating a DNS record (**A or CNAME**)
- **Destination Mapping:** The DNS record links the subdomain name to its specific server or destination

Introduction to Route 53

➤ Route 53 Key Concepts

□ What is a DNS record?

- A DNS record is a set of instructions that tells Route 53 how to respond when someone tries to access your domain or subdomains

□ Types of DNS Records

- **A record** – Maps your domain or subdomain to an IPv4 address

▪ Example

www.ecomartgreen.com to 192.0.2.12

- **CNAME record** – Maps one domain name to another

www.ecomartgreen.com to ecomartgreen.com

- **Alias record** – Unique to AWS, it works like a CNAME but at the root domain level



Introduction to Route 53

➤ Route 53 Key Concepts

□ Hosted Zone

- **Route 53 domain registration:** Auto-created public hosted zone
- **Hosted zone:** DNS control panel for your domain
 - Routing web traffic to your web server or load balancer
 - Managing email delivery
 - Creating and routing subdomains

store.ecomartgreen.com or blog.ecomartgreen.com

Introduction to Route 53

➤ Route 53 Key Concepts

□ NS and SOA Records

- **NS (Name Server)** records tell the internet which Route 53 name servers are authoritative for your domain
- **SOA (Start of Authority) record** holds information about your domain
 - Primary name server
 - Admin contact
 - DNS servers refresh cached data

□ Traffic Routing in Route 53

- DNS records direct traffic to resources (web, email, AWS, etc.)
- Routing policies offer granular control over traffic direction

Introduction to Route 53

➤ Route 53 Key Concepts

□ Routing Policies

Simple Routing Policy – Routes traffic to a single resource

Failover Routing Policy – Configures active-passive failover

Geolocation Routing Policy – Routes traffic based on user location

Geoproximity Routing Policy – Routes traffic based on resource location

Latency Routing Policy – Routes traffic to the resource with the best latency

IP-based Routing Policy – Routes traffic based on the client's IP

Multivalue Answer Routing Policy – Responds with up to eight healthy records randomly

Weighted Routing Policy – Distributes traffic based on specified weights

Introduction to Route 53

➤ Route 53 Key Concepts

□ Health Monitoring

- Route 53 checks if your website is up
- If a health check fails, it can reroute traffic to a healthy backup server
- Health checks can monitor HTTP, HTTPS, or TCP protocols

High availability is critical for user trust at businesses like EcoMart, where uptime is vital

Introduction to Cert Manager

Introduction to Cert Manager

ecomartgreen.com

- ✓ How encryption works
- ✓ Why HTTPS is important
- ✓ How ACM simplifies certificate management

Introduction to Cert Manager

➤ Encryption and Decryption

✓ Encryption

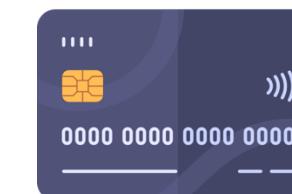
- ❑ Process of converting readable data, known as **plaintext**, into an unreadable format, called **ciphertext**

- ❑ Ensures **confidentiality** when data is transferred over the internet



✓ Decryption

- ❑ **Reverse** process which restores the original data from the encrypted version

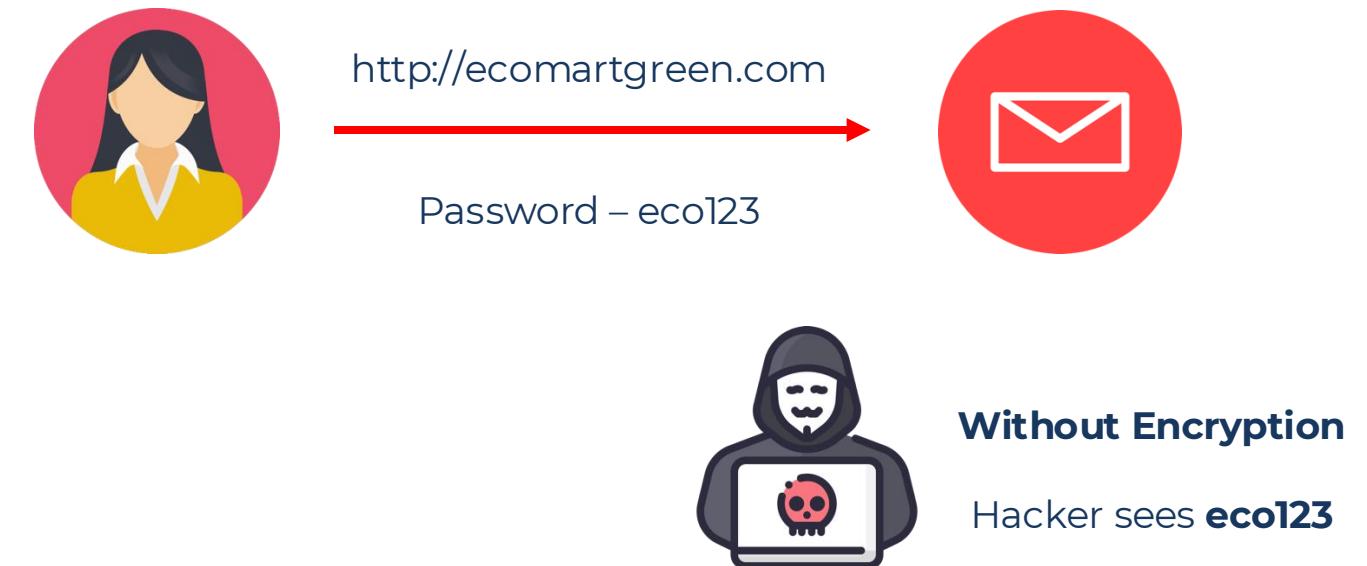


Introduction to Cert Manager

➤ Understanding HTTP and HTTPS

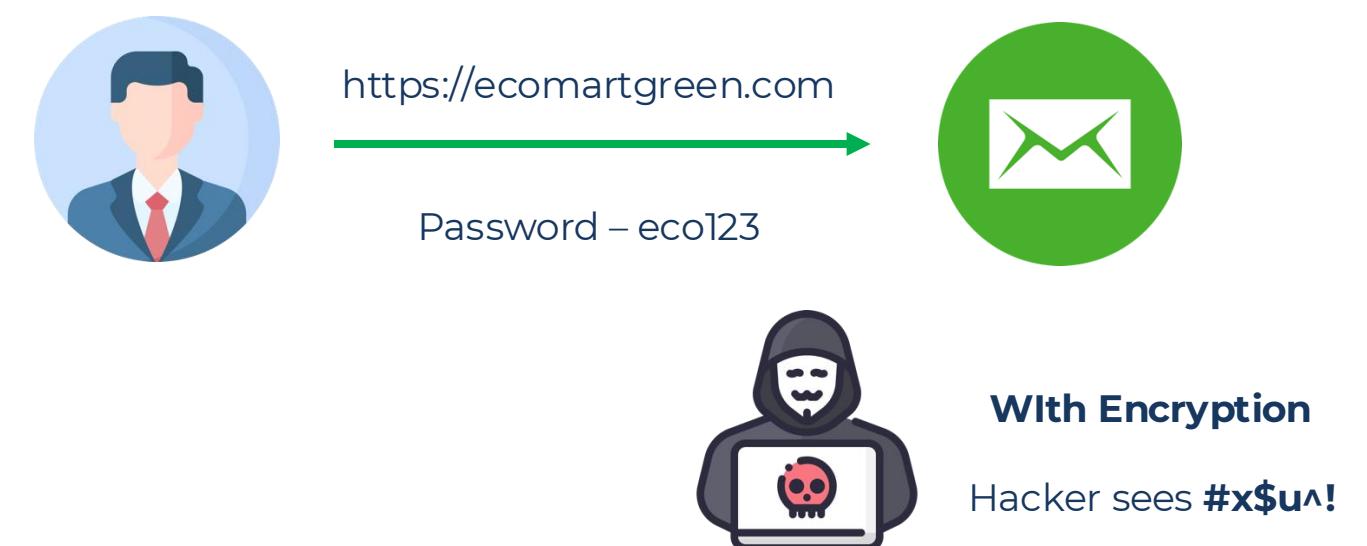
□ HTTP (Hypertext Transfer Protocol)

- ✓ The **standard** protocol for data exchange over the web
- ✓ **Not secured**, meaning anyone with access to the network can potentially intercept the information



□ HTTPS (HTTP Secure)

- ✓ Adds a layer of encryption to HTTP using **SSL or TLS protocols**
- ✓ Data transmitted between a user's browser and a web server is **encrypted** and **secure**

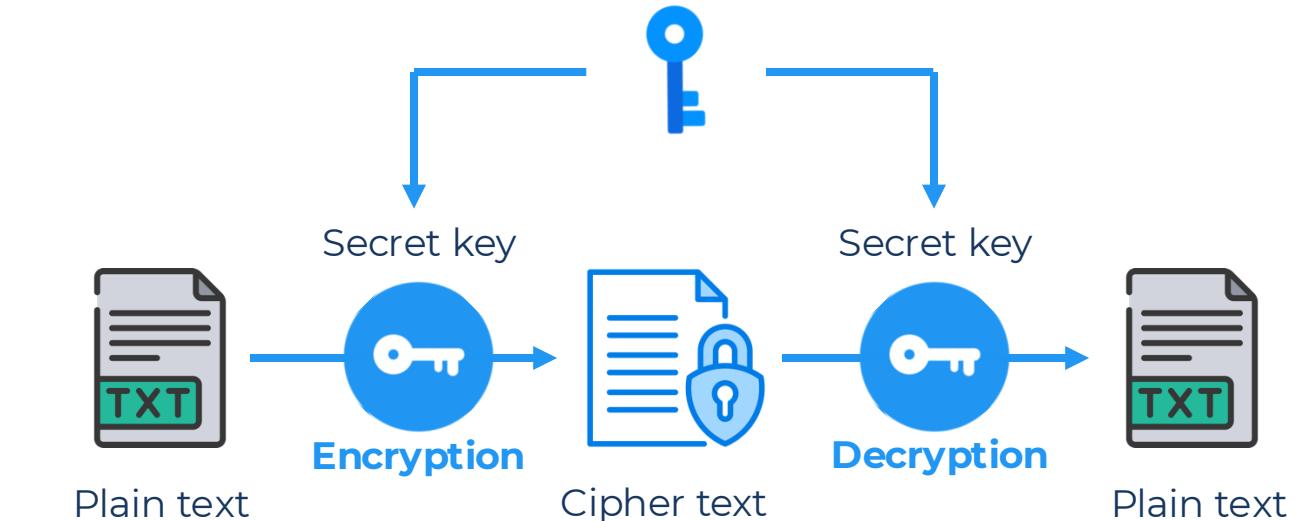


Introduction to Cert Manager

➤ Symmetric and Asymmetric Encryption

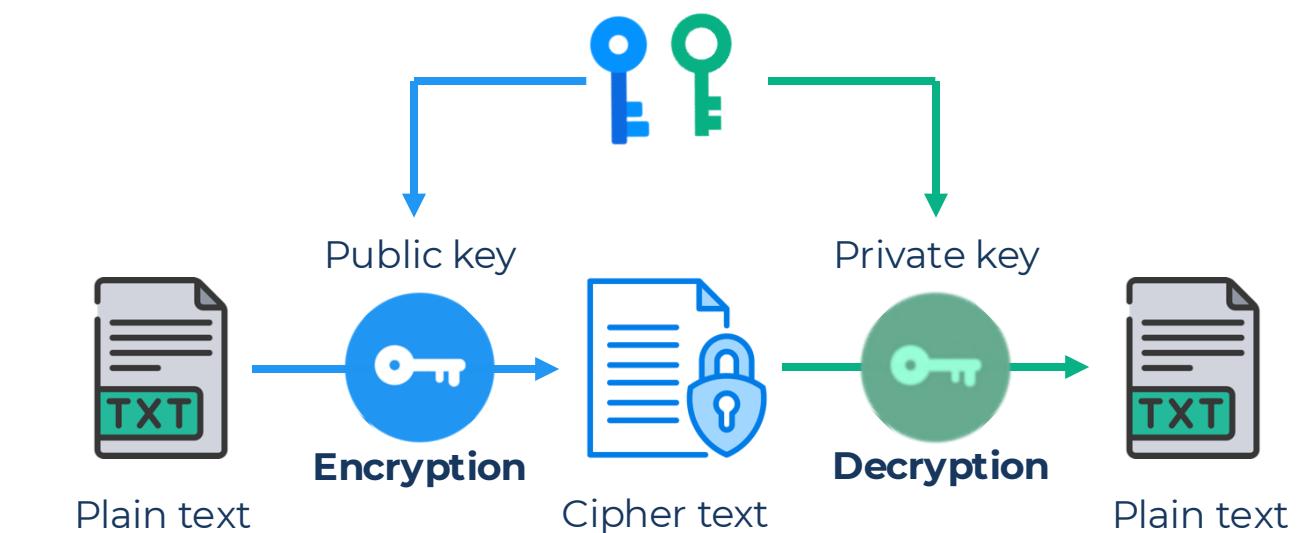
□ Symmetric Encryption

- ✓ Uses a **single key** for both encryption and decryption
- ✓ Fast and efficient, but needs a securely shared key



□ Asymmetric encryption

- ✓ Uses a **public key** for encryption and a **private key** for decryption
- ✓ More secure, especially when sharing keys between unknown parties
- ✓ Combines both methods to ensure a balance of **speed** and **security**



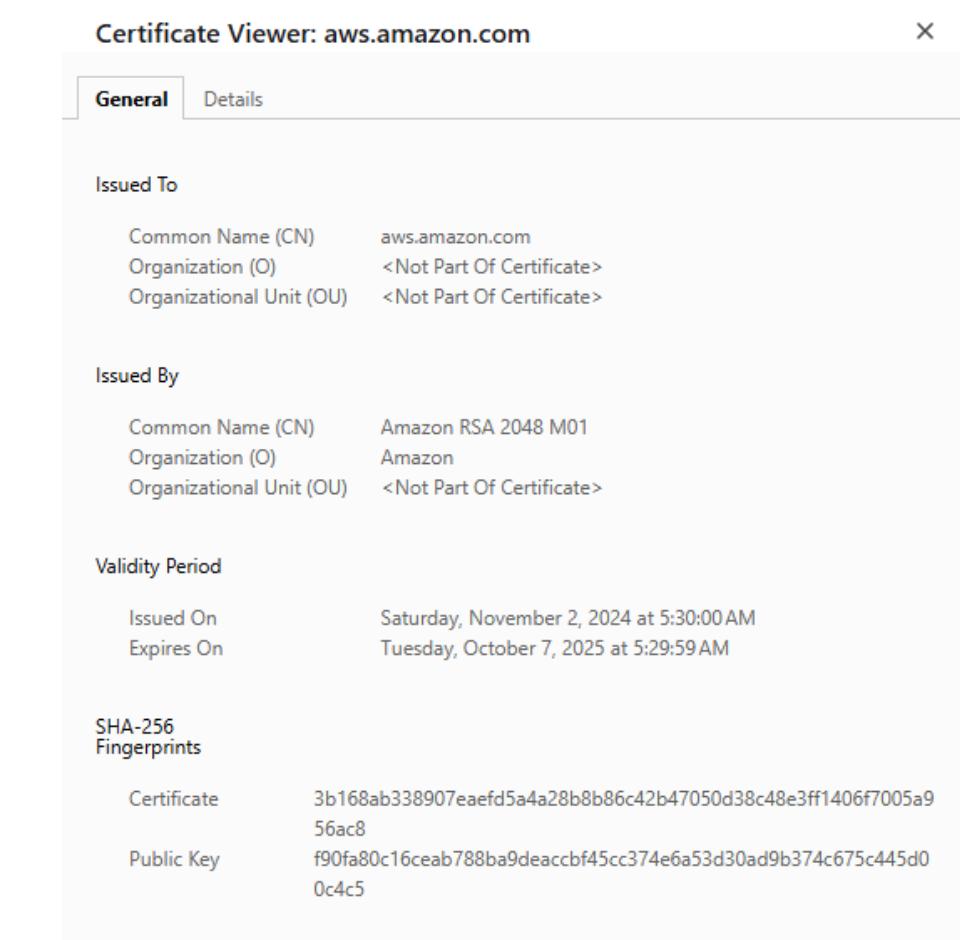
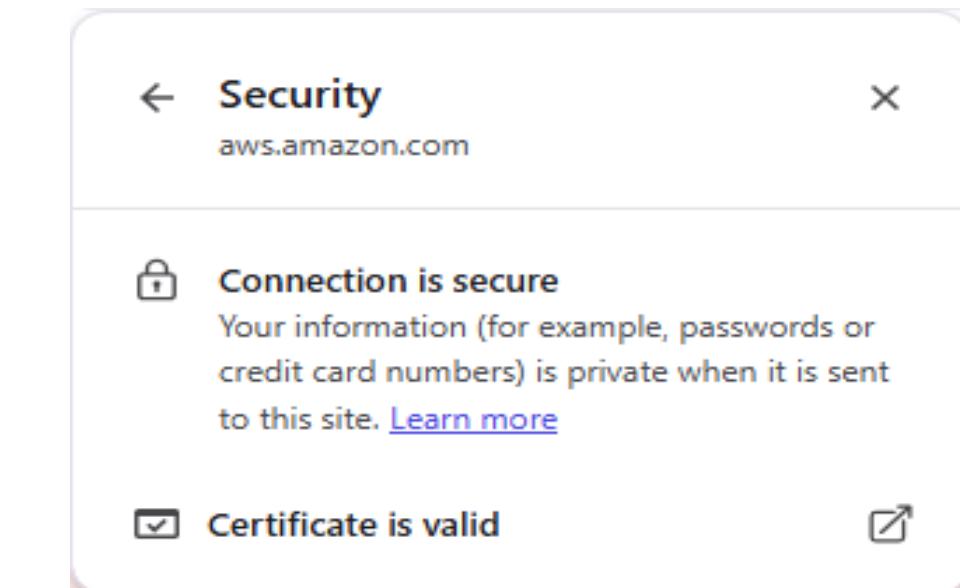
Introduction to Cert Manager

➤ SSL and TLS Certificates

- ✓ To enable HTTPS, a website must use an **SSL or TLS certificate**
- ✓ Authenticate website's identity and establishes a secure communication channel

□ SSL or TLS certificate is a **digital file**

- ✓ Website's public key
- ✓ Certificate's expiration date
- ✓ Domain name
- ✓ Signature of a trusted Certificate Authority (CA)



Introduction to Cert Manager

➤ Certificate Authorities and Chain of Trust

- ✓ Certificate Authorities issues certificates
- ✓ It is a trusted organizations that validate the legitimacy of websites



Chain of Trust

Ensures that the website you're connecting to is authentic

Introduction to Cert Manager

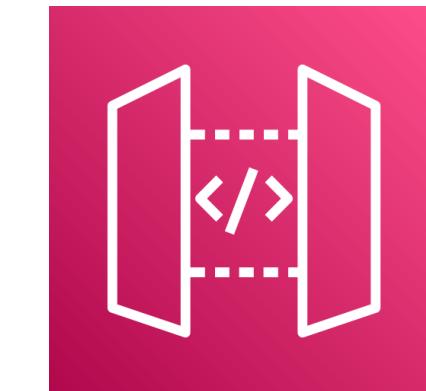
➤ Fully Qualified Domain Name (FQDN)

- ✓ It is the complete address of a website on the internet
- ✓ For example, www.ecomartgreen.com is the FQDN of the EcomartGreen website
- ✓ Certificates are issued to secure specific FQDNs
- ✓ Issued for wildcard domains, such as ***.ecomartgreen.com**, which secure all subdomains under the main domain

Introduction to Cert Manager

➤ Introducing AWS Certificate Manager (ACM)

- ❑ **AWS Certificate Manager, or ACM** handles,
 - ✓ Creation
 - ✓ Storage
 - ✓ Deployment
 - ✓ Automatic renewal of public and private SSL/TLS certificates
 - ✓ Easily request a certificate & associate it with AWS services



Introduction to Cert Manager

➤ Key Benefits of ACM

- ✓ Free public certificates for use with AWS services
- ✓ Support for single domains, multiple domains, and wildcard domains
- ✓ Automatic renewal of certificates to prevent service disruption
- ✓ Seamless integration with AWS services, reducing operational overhead
- ✓ Option to import third-party certificates if needed

Introduction to Cert Manager

➤ ACM and Private Certificates

- ✓ Supports private certificates through AWS Private Certificate Authority
- ✓ Ideal for internal applications and services that are not exposed to the internet
- ✓ Enables organizations to issue internal certificates via their own CA
- ✓ Private certificates cannot be used for public websites

Introduction to Cert Manager

➤ What's Inside an ACM Certificate

- ACM issues **X.509 v3** certificates valid for **13 months**
- ✓ **Basic Constraints**, identifying whether the certificate is a Certificate Authority
- ✓ **Authority Key Identifier**, linking the certificate to its issuer
- ✓ **Subject Key Identifier**, identifying the certificate's public key
- ✓ **Key Usage and Extended Key Usage**, defining what the certificate is used for
- ✓ **CRL Distribution Points**, providing a location to check certificate revocation status

ecomartgreen.com

- ✓ Provides a secure, reliable, managed solution for serving over HTTPS
- ✓ Ensures confidentiality, simplifies cert management, and builds trust

Introduction to Cert Manager

➤ Route 53 and Certificate Manager

- ACM offers two domain validation methods:

Method	Description	Renewal
Email Validation	ACM sends approval emails to domain contacts Certificate issued upon approval	Manual
DNS Validation	ACM provides CNAME records to be added to DNS Preferred for automatic renewal	Automatic

- ✓ When you request a certificate through ACM, you'll receive unique CNAME records
- ✓ These records are added to Route 53 to verify domain ownership
- ✓ ACM auto-renews certificates after CNAME setup
- ✓ Simplifies HTTPS management and boosts security



Demonstration | Setting Up Subdomain with A Record in Route 53



Demonstration | SSL Cert Setup with Cert Manager & Route 53



Demonstration | TLS Listener & Cert Setup

Summary

- Discussed Route 53 and ACM
- Set up a subdomain for an application in Route 53 using an A Record
- Created an SSL certificate in ACM and configured it with Route 53
- Added a TLS listener and attached the certificate to an Elastic Load Balancer
- Accessed the application securely via HTTPS

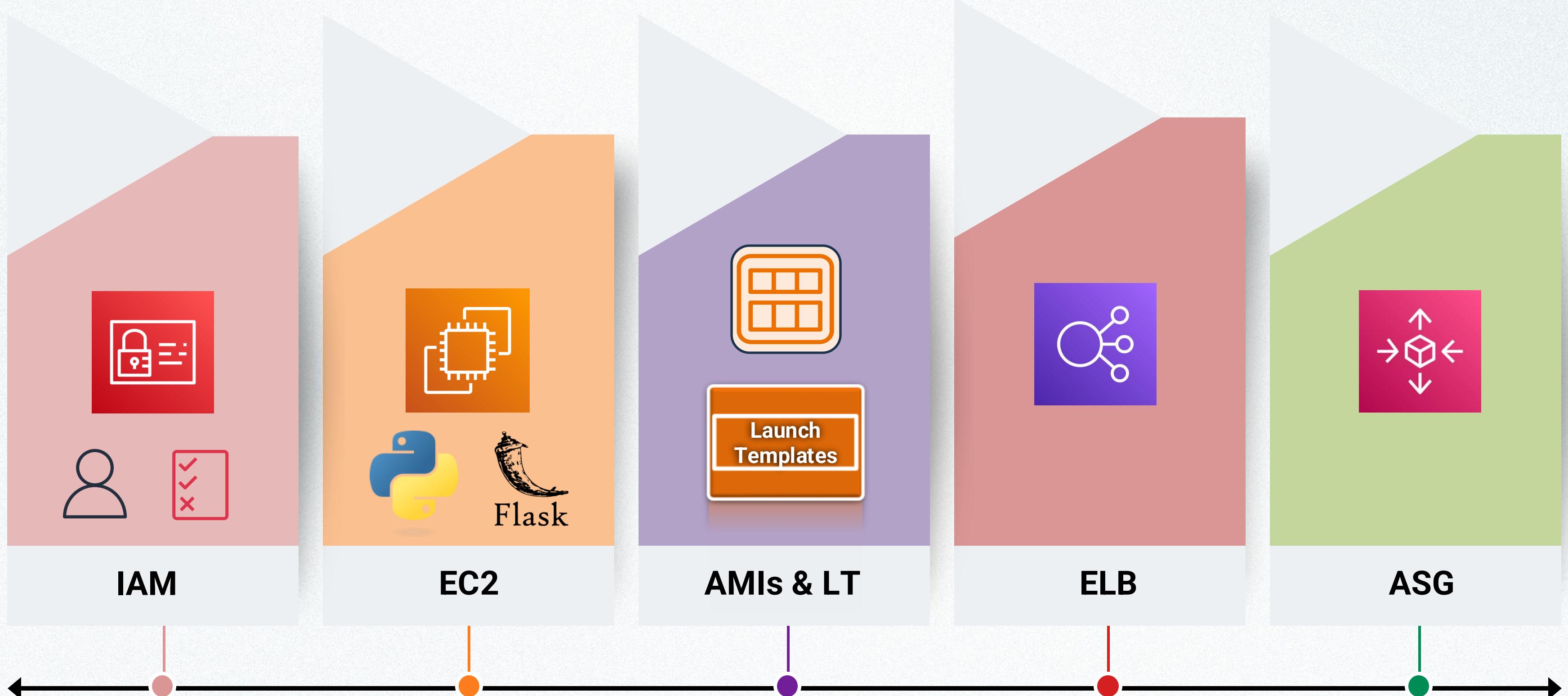


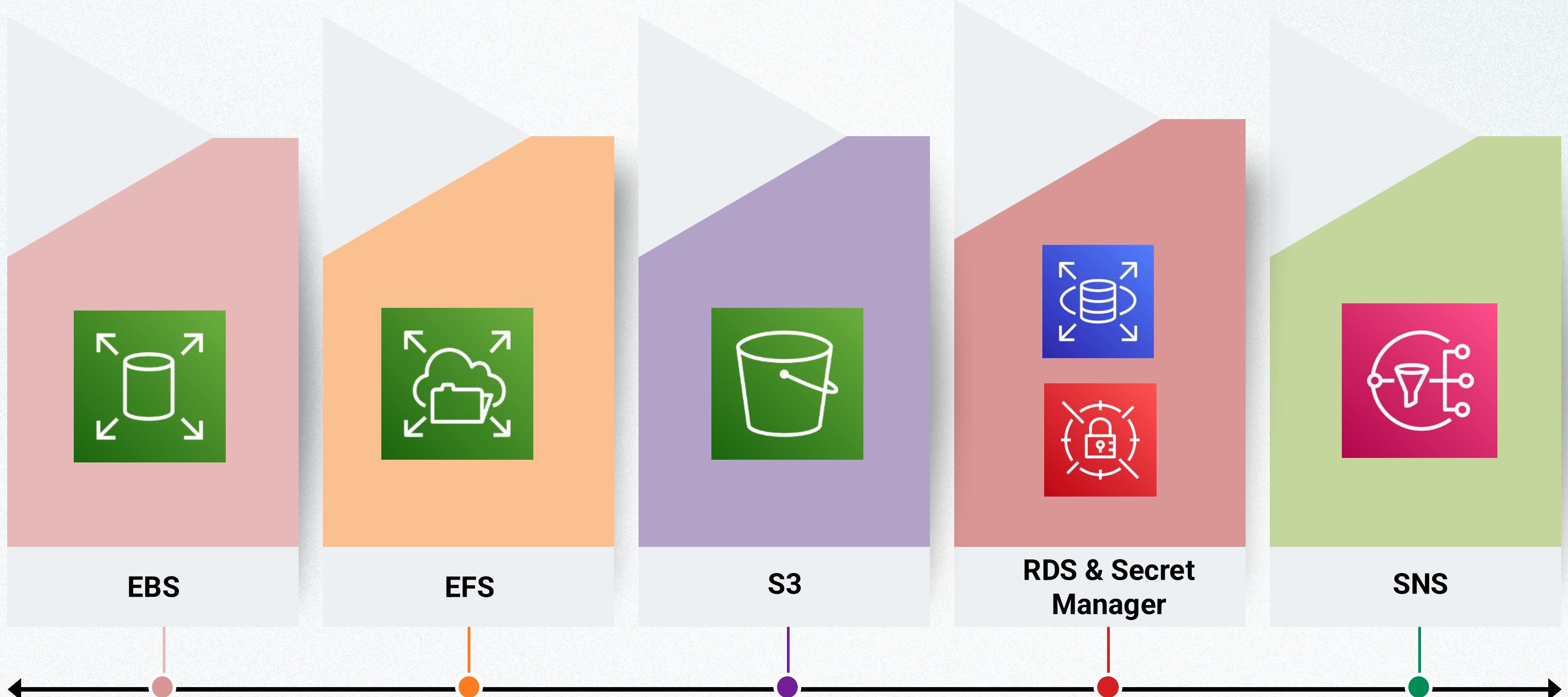
Build and Scale with AWS

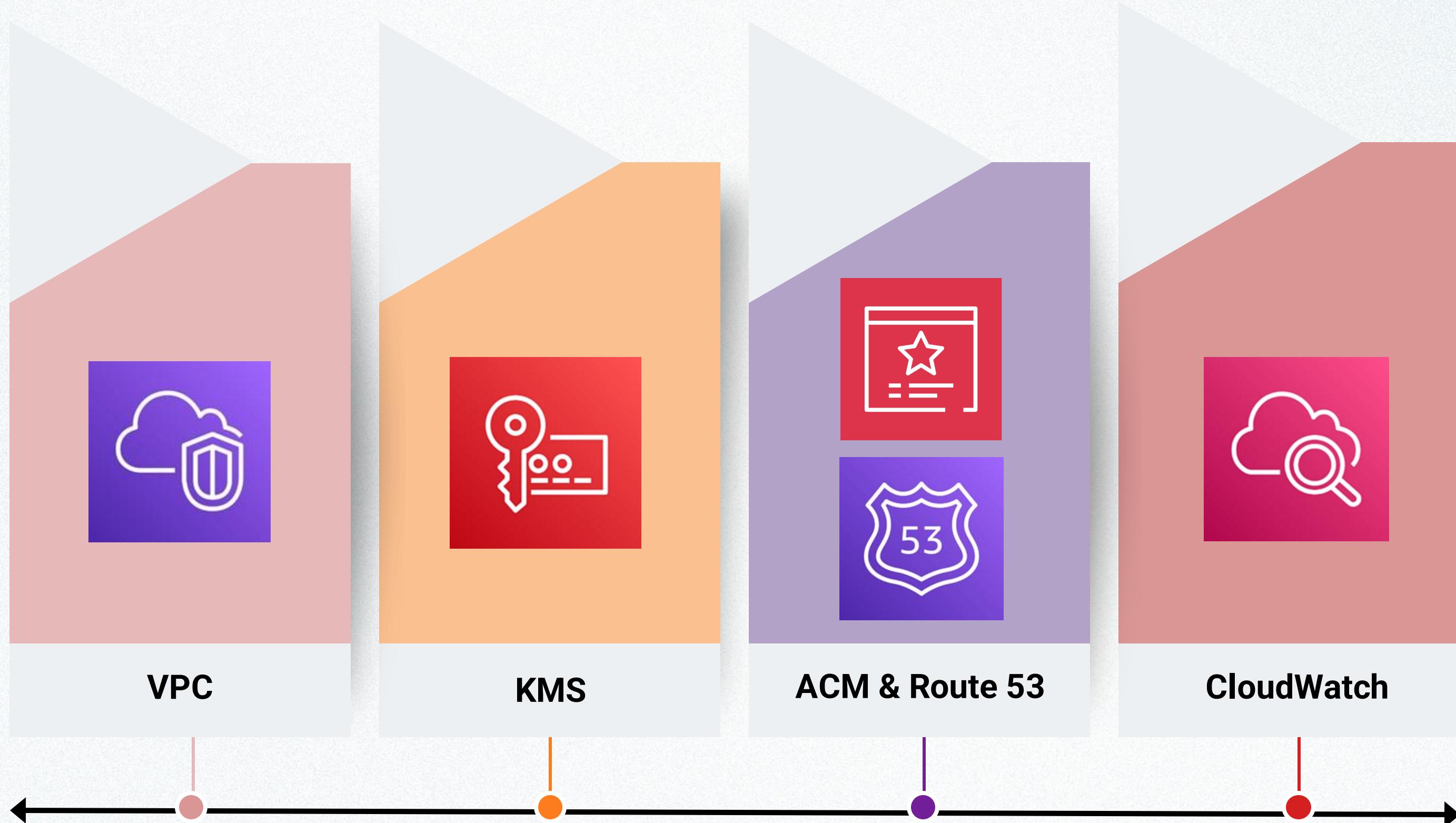
A Hands-on Beginner's

Guide

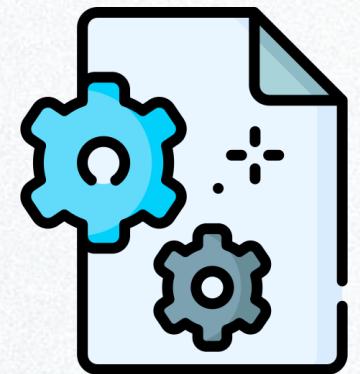








What's Next



Cache Management



AWS Elastic Beanstalk



AWS Lambda



NoSQL databases



Follow us on:

 @thinknyx

 @thinknyx

 @thinknyx-technologies

 @thinknyx

 @thinknyx-technologies