

UNIT-II

Introduction to AWS:

What is AWS?, three ways to interact with AWS, AWS Global Infrastructure Overview, AWS services and service categories.

Compute: Compute services overview, Introduction to Amazon EC2, AMI's, Volumes, and Snapshots.

Storage: AWS S3: S3 durability and redundancy S3 Buckets, S3 Uploading Downloading, S3 Permissions.

What is AWS?

- The full form of AWS is **Amazon Web Services**.
- It is a platform that offers **flexible, reliable, scalable, easy-to-use and, cost-effective cloud computing solutions**.
- AWS is a comprehensive, easy to use computing platform offered Amazon. The platform is developed with a combination of **infrastructure as a service (IaaS), platform as a service (PaaS) and packaged software as a service (SaaS) offerings**.
- Amazon Web Services (AWS) is the world's most comprehensive and broadly adopted cloud, offering over **200 fully featured services from data centers globally**.
- Millions of **customers—including the fastest-growing startups, largest enterprises, and leading government agencies**—are using AWS to lower costs, become more agile, and innovate faster.

AWS provides services to customers when required without any prior commitment or upfront investment.

Pay-As-You-Go enables the customers to procure services from AWS.

- ❖ Computing
- ❖ Programming models
- ❖ Database storage
- ❖ Networking



History of AWS

2002- AWS services launched

2003: In 2003, Chris Pinkham and Benjamin Black presented a paper on how Amazon's own internal infrastructure should look like. They suggested to sell it as a service and prepared a business case on it. They prepared a six-page document and had a look over it to proceed with it or not. They decided to proceed with the documentation.

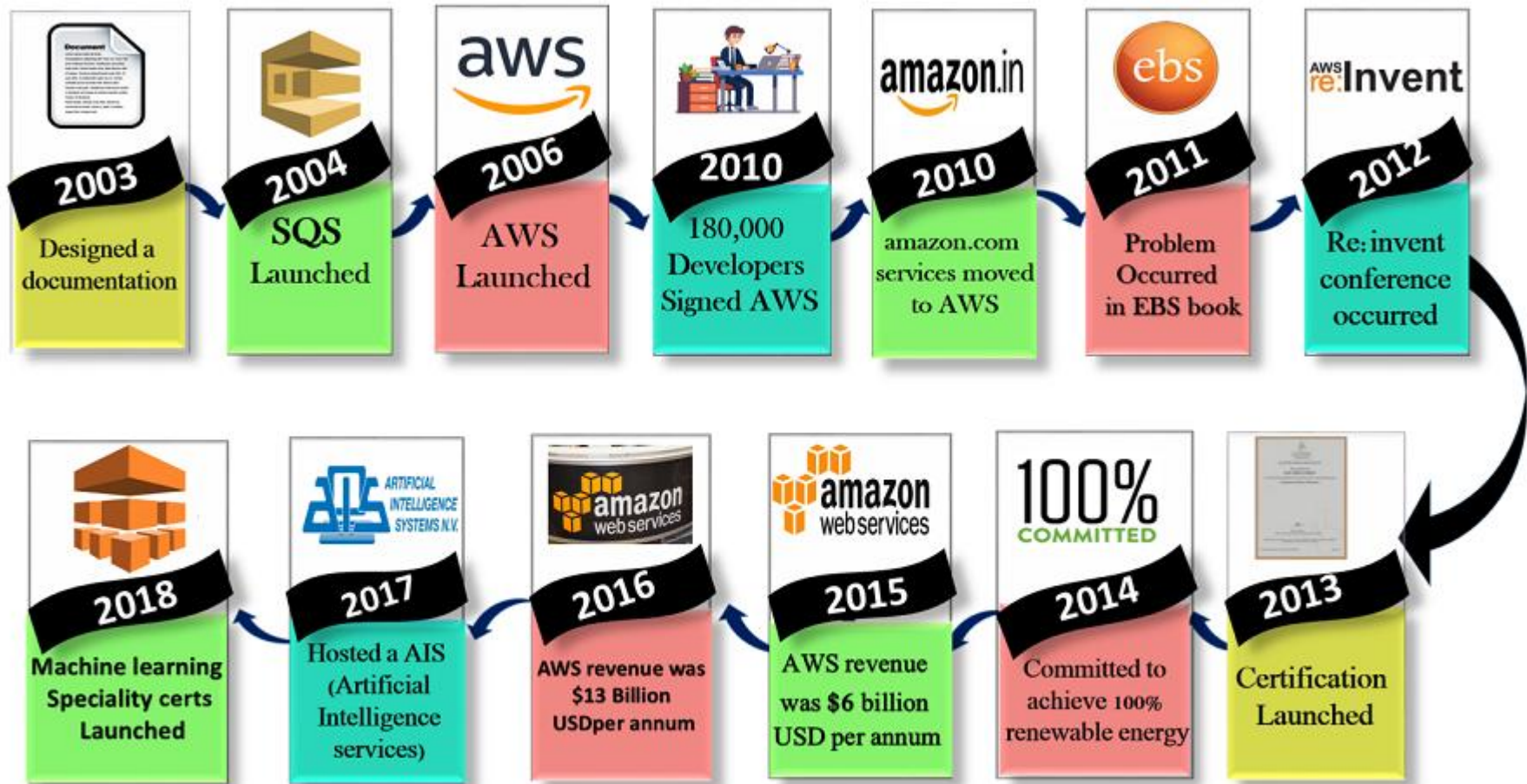
2004: SQS stands for "Simple Queue Service" was officially launched in 2004. A team launched this service in Cape Town, South Africa.

2006: AWS (Amazon Web Services) was officially launched.

2007: In 2007, over 180,000 developers had signed up for the AWS.

2010: In 2010, amazon.com retail web services were moved to the AWS, i.e., amazon.com is now running on AWS.

2011: AWS suffered from some major problems. Some parts of volume of EBS (Elastic Block Store) was stuck and were unable to read and write requests. It took two days for the problem to get resolved.



2012: AWS hosted a first customer event known as re:Invent conference. First re:invent conference occurred in which new products were launched. In AWS, another major problem occurred that affects many popular sites such as Pinterest, Reddit, and Foursquare.

2013: In 2013, certifications were launched. AWS started a certifications program for software engineers who had expertise in cloud computing.

2014: AWS committed to achieve 100% renewable energy usage for its global footprint.

2015: AWS breaks its revenue and reaches to \$6 Billion USD per annum. The revenue was growing 90% every year.

2016: By 2016, revenue doubled and reached \$13Billion USD per annum.

2017: In 2017, AWS re: invent releases a host of Artificial Intelligence Services due to which revenue of AWS doubled and reached \$27 Billion USD per annum.

2018: In 2018, AWS launched a **Machine Learning Speciality Certs**. It heavily focussed on automating Artificial Intelligence and Machine learning.

AWS is made up of many different [cloud computing](#) products and services.

The highly profitable division of Amazon provides servers, storage, networking, remote computing, email, mobile development, and security.

AWS can be broken into three main products: **EC2**, Amazon's virtual machine service, **Glacier**, a low-cost cloud storage service, and **S3**, Amazon's storage system.

AWS is so large and present in the computing world that it's far outpaced its competitors.

As of the first quarter of 2021, one independent analyst reports AWS has over a third of the market at 32.4%, with Azure following behind at 20%, and Google Cloud at 9%.

AWS has 81 availability zones in which its servers are located.

These serviced regions are divided in order to allow users to set geographical limits on their services (if they so choose), but also to provide security by diversifying the physical locations in which data is held.

AWS spans 245 countries and territories.

Categories of AWS services:



Analytics



Application
Integration



AR and VR



Blockchain



Business
Applications



Compute



Cost
Management



Customer
Engagement



Database



Developer Tools



End User
Computing



Game Tech



Internet
of Things



Machine
Learning



Management and
Governance



Media Services



Migration and
Transfer



Mobile



Networking and
Content Delivery



Robotics



Satellite



Security, Identity, and
Compliance



Storage

What Is aws



Amazon S3



Amazon ECS



Amazon
CloudTrail



AWS
CloudFormation



AWS RDS



AWS
CLI



AWS
Lambda



amazon
REDSHIFT



AWS
CLI



Amazon EBS



Amazon EFS



Amazon ELB



Elastic
Beanstalk



AWS
VPC



AWS SQS



AWS
Route 53



AWS IAM



amazon
SNS



Amazon DynamoDB

Three ways to access the AWS:

[1.AWS Management Console](#)

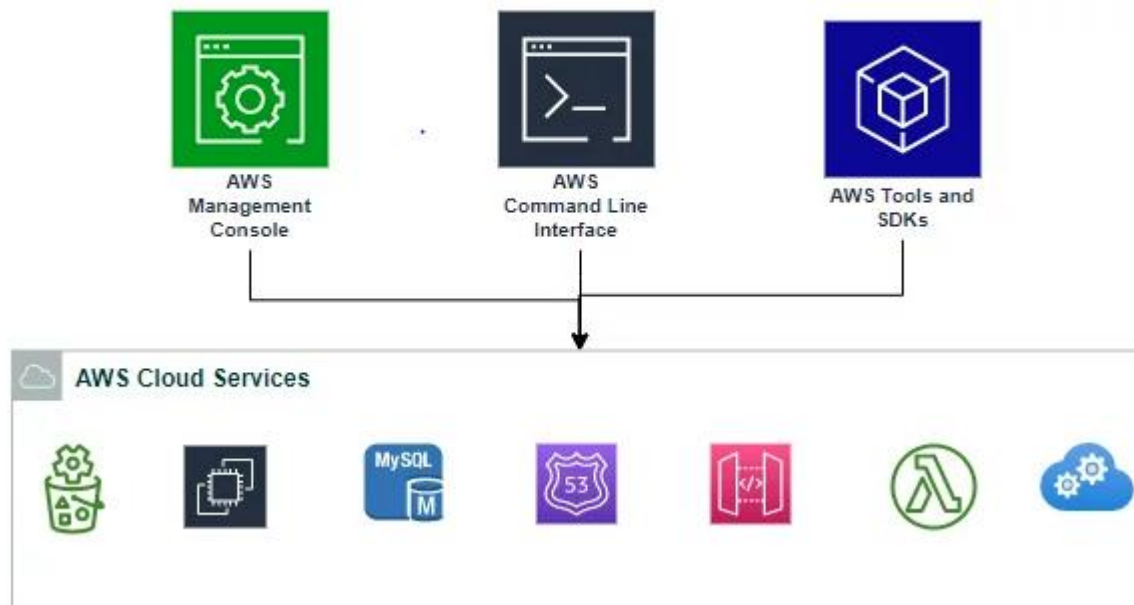
[2 AWS CLI \(Command Line Interface\)](#)

[3 AWS SDK \(Software Development Kits\)](#)

AWS consists of many cloud services that you can use in combinations as per organizational requirements.

Users need to create an AWS Account to access and manage AWS resources.

Interacting with AWS services

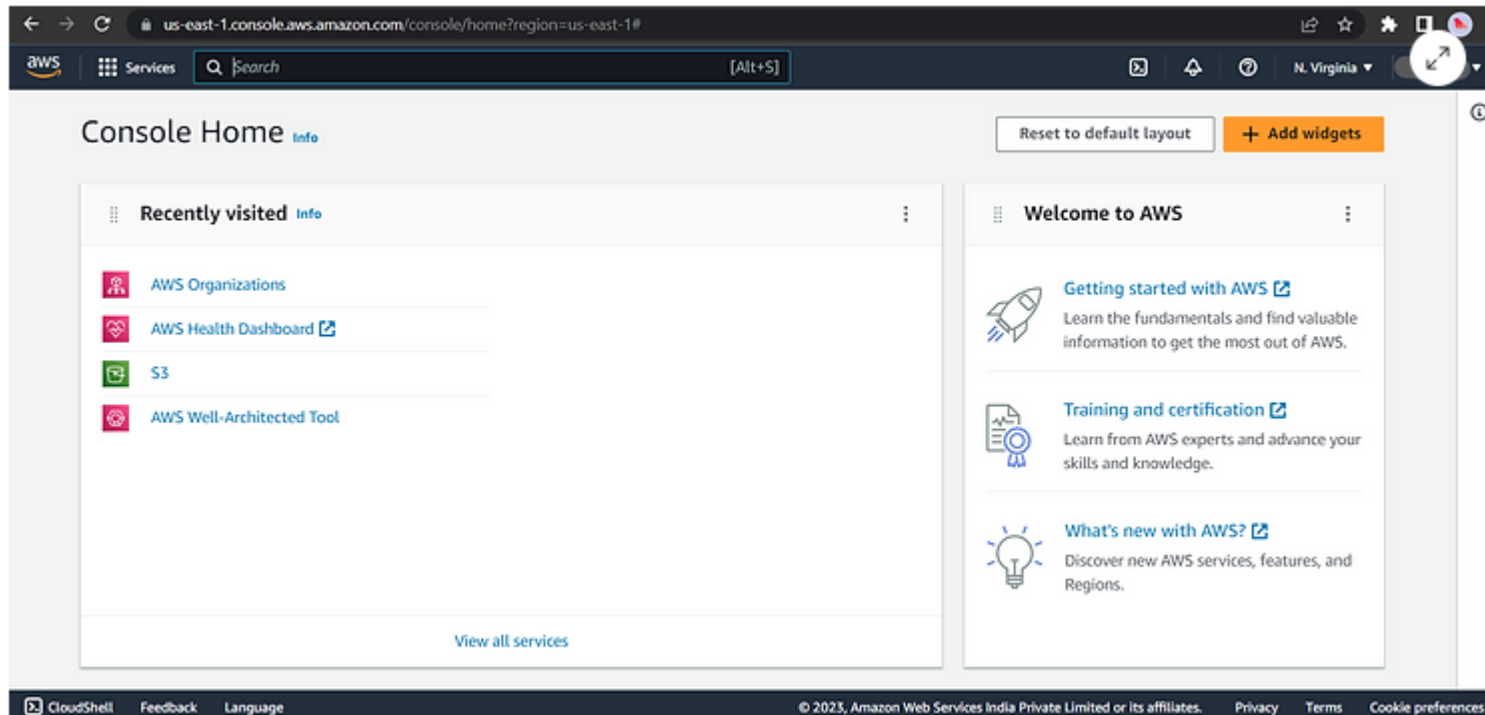


AWS Management Console

AWS Management Console provides a web-based user interface that allows users to interact with AWS services through a graphical user interface (GUI).

It's designed for users who are new to AWS and provides an easy-to-use interface for managing AWS resources.

The console provides access to all AWS services and resources, making it a great tool for users who are looking for a quick and easy way to get started with AWS.



AWS Command Line Interface (CLI)

The **AWS Command Line Interface (CLI)** is a tool that allows users to interact with AWS services through a command-line interface (CLI).

It provides users with the ability to automate common tasks and perform complex operations through a set of command-line tools.

The CLI supports Windows, macOS, and Linux, making it a versatile tool for developers and administrators.

The screenshot shows the AWS Command Line Interface (CLI) landing page. The top navigation bar includes the AWS logo, links for Products, Solutions, Pricing, Documentation, Learn, Partner Network, AWS Marketplace, Customer Enablement, Events, and Explore More. A 'Sign In to the Console' button is also present. The main content area is titled 'AWS Command Line Interface' and features a description of the tool, a 'Get Started with AWS for Free' section with a 'Create Free Account' button, and a list of related links including Documentation, Tools, and Release Notes. The page also includes a sidebar with 'RESOURCES' and 'RELATED LINKS'.

RESOURCES

- [AWS Command Line Interface](#)

RELATED LINKS

- [Documentation](#)
- [Tools](#)
- [Release Notes](#)

Get Started with AWS for Free

[Create Free Account](#)

AWS Command Line Interface

The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

The AWS CLI v2 offers several [new features](#) including improved installers, new configuration options such as AWS IAM Identity Center (successor to AWS SSO), and various interactive features.

Windows
Download and run the [64-bit Windows installer](#).

MacOS
Download and run the [MacOS PKG installer](#).

Linux
Download, unzip, and then run the [Linux installer](#).

Amazon Linux
The AWS CLI comes pre-installed on [Amazon Linux AMI](#).

Release Notes
Check out the [Release Notes](#) for more information on the latest version.

Getting Started » **AWS CLI Reference »** **GitHub Project »** **Community Forum »**

AWS provides multiple options to choose from.

Two prominent methods:

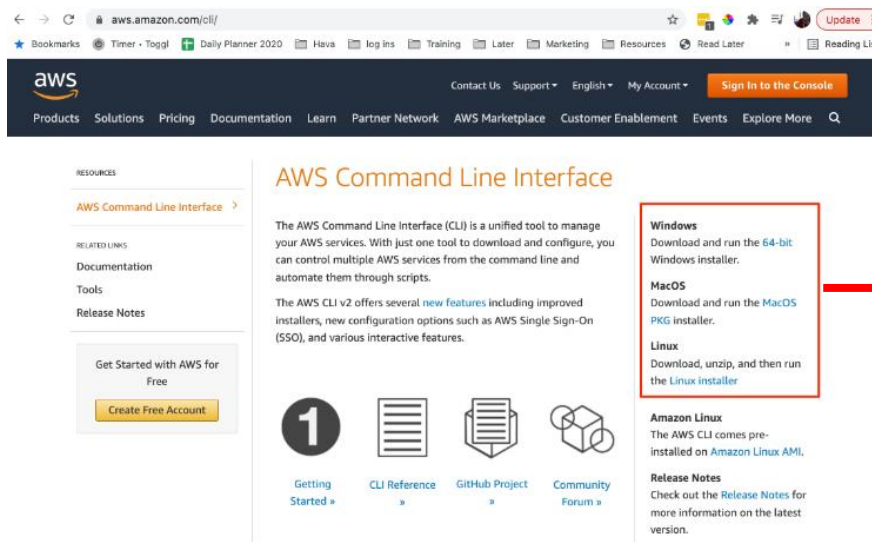
- **Local Installation of the AWS CLI**
- **CloudShell Integration**

1. Local Installation

With local installation, bring the power of the AWS CLI directly to your local development environment.

By installing the AWS CLI on your machine, you enable quick and convenient access to a excess amount of AWS services right from your terminal. This method provides **seamless integration between your workflow and the AWS ecosystem**, allowing you to script, automate, and manage resources efficiently.

To install the AWS CLI you can go to <https://aws.amazon.com/cli/>



On the right hand side of the screen you will find the download instructions for Windows, MacOS or Linux. If you are using Amazon Linux, the AWS CLI comes pre-installed.

```
Command Prompt
Microsoft Windows [Version 10.0.19045.3324]
(c) Microsoft Corporation. All rights reserved.

C:\Users\sudip>aws --version
aws-cli/2.13.11 Python/3.11.4 Windows/10 exe/AMD64 prompt/off

C:\Users\sudip>
```

Configuring the AWS Profile

But first, you need to configure your AWS profile to interact with your AWS resources. Your AWS profile is a **set of credentials** you can use to authenticate yourself using the AWS CLI.

```
C:\Users\sudip>aws configure
AWS Access Key ID [None]: 
AWS Secret Access Key [None]: 
Default region name [None]: 
Default output format [None]:
```

CloudShell Integration

AWS CloudShell is a **browser-based shell** that you can access directly from your AWS management console. CloudShell pre-authenticates the access on the shell which runs on an **Amazon Linux computing** environment. CloudShell comes **pre-configured with the AWS CLI, SDKs, and other useful tools**. This means you can access the power of the AWS CLI without any installation, setup, or configuration on your local machine. With **CloudShell, you can focus on managing your resources and executing commands without worrying about environment setup**.



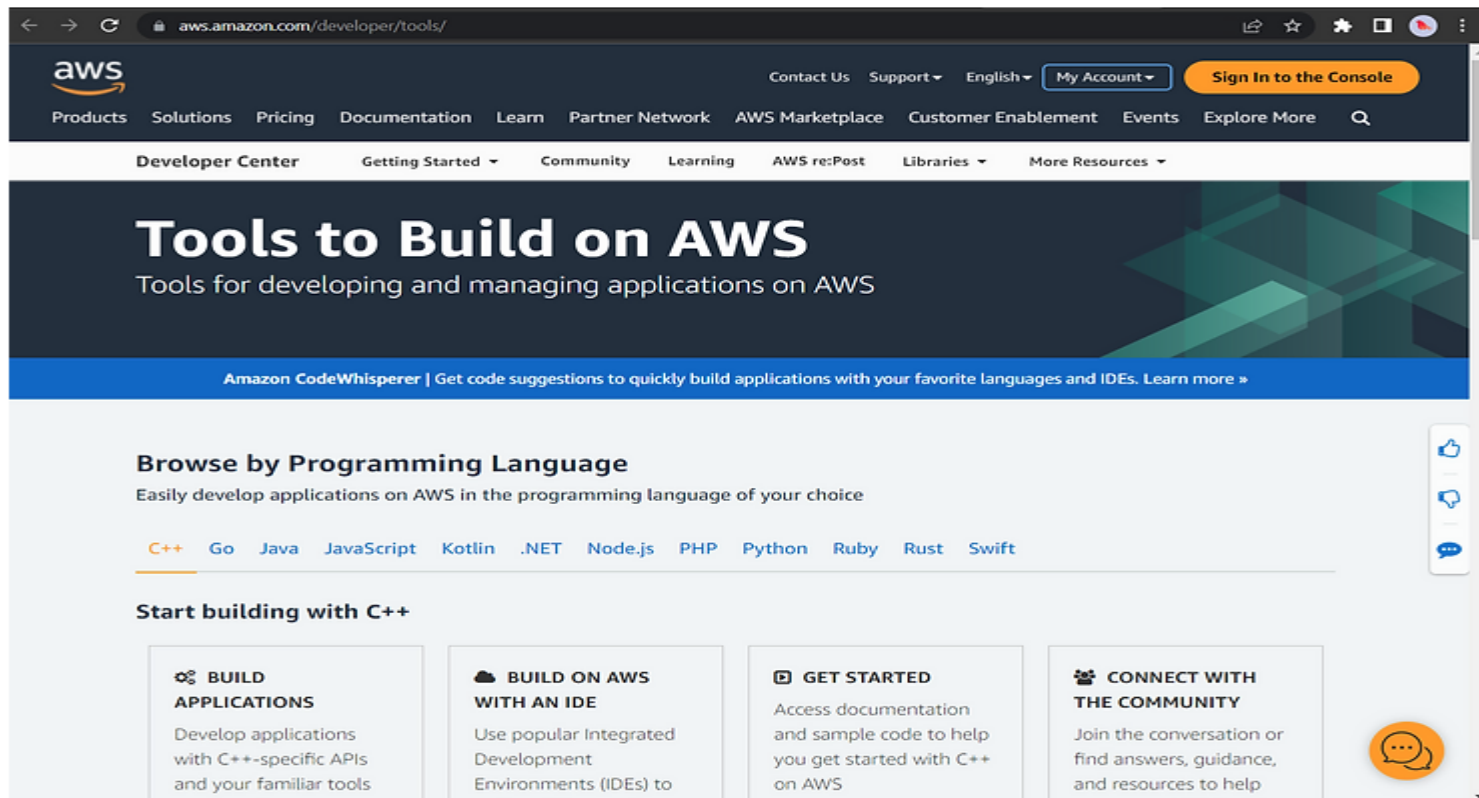
AWS CloudShell

AWS Software Development Kits (SDKs)

The **AWS Software Development Kits (SDKs)** are libraries that enable developers to interact with AWS services from their applications.

The SDKs are available in several programming languages, including Java, Python, .NET, and JavaScript.

They provide a simple and consistent programming interface that makes it easy to develop and deploy applications on AWS.



AWS Global Infrastructure:

Amazon Web Services is a global public cloud provider, and as such, it has to have a global network of infrastructure to run and Manage its many growing cloud services that support customers around the world.

The AWS Global Infrastructure Contains the below components

- ❖ Availability Zones (AZs)
- ❖ Regions
- ❖ Edge Locations
- ❖ Regional Edge Caches
- ❖ Local Zones
- ❖ Wavelength Zones
- ❖ Outposts.

Availability Zones:

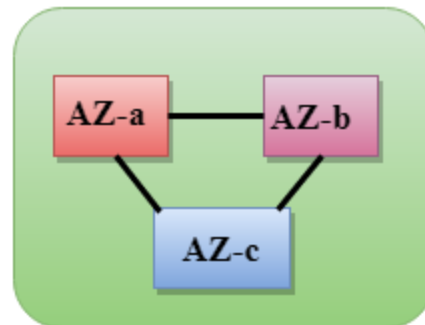
AZs are essentially the physical data centers of AWS.

This is where the actual compute, storage, network, and database resources are hosted that we as consumers provision within our Virtual Private Clouds (VPCs).

The things which interact with the cloud sits inside the data centers.

An availability zone can be a several data centers, but if they are close together, they are counted as 1 availability zone.

Each AZ will always have at least one other AZ that is geographically located within the same area, usually a city, linked by highly resilient and very low latency private fiber-optic connections.



Data Center :

a data center holds servers and/or data storage. This covers the actual hardware, the room in which it is kept, the power and backup systems, the environmental controls, and anything else required to keep those servers operational.

A data center might consist of a single server or a large complex with many servers mounted on racks.

Like a personal computer but with a higher power, each server has a processor, storage, and memory.

- Systems for handling, processing, exchanging, and storing data throughout the enterprise.
- Physical infrastructure facilitating data communications and processing.
- Utilities include network security access, cooling, energy, and uninterruptible power supply (UPSes)

Companies that provide [public cloud](#) computing services, like Amazon or Microsoft, have data centers that they make available to other businesses.



Regions:

Region is a collection of availability zones that are geographically located close to one other.

This is generally indicated by AZs within the same city.

AWS has deployed them across the globe to allow its worldwide customer base to take advantage of low latency connections.

Every Region will act independently of the others, and each will contain at least two Availability Zones.

Having global regions also allows for compliance with regulations, laws, and governance relating to data storage (at rest and in transit).

For example, you may be required to keep all data within a specific location, such as Europe.

Having multiple regions within this location allows an organization to meet this requirement.

Region and Availability Zone Naming Conventions:

AWS has a specific naming convention for both Regions and Availability Zones.

Depending on where you are viewing and using the Region name, it can be represented as two different names for the same Region.

Regions have both a 'friendly' name, indicating a location that can be viewed within the Management Console and a Code Name that is used when referencing regions programmatically.

<i>Name</i>	<i>Code Name</i>
<i>US East (N. Virginia)</i>	us-east-1
<i>US East (Ohio)</i>	us-east-2
<i>US West (N. California)</i>	us-west-1
<i>US West (Oregon)</i>	us-west-2
<i>Africa (Cape Town)</i>	af-south-1
<i>Asia Pacific (Hong Kong)</i>	ap-east-1
<i>Asia Pacific (Hyderabad)</i>	ap-south-2
<i>Asia Pacific (Jakarta)</i>	ap-southeast-3
<i>Asia Pacific (Melbourne)</i>	ap-southeast-4
<i>Asia Pacific (Mumbai)</i>	ap-south-1



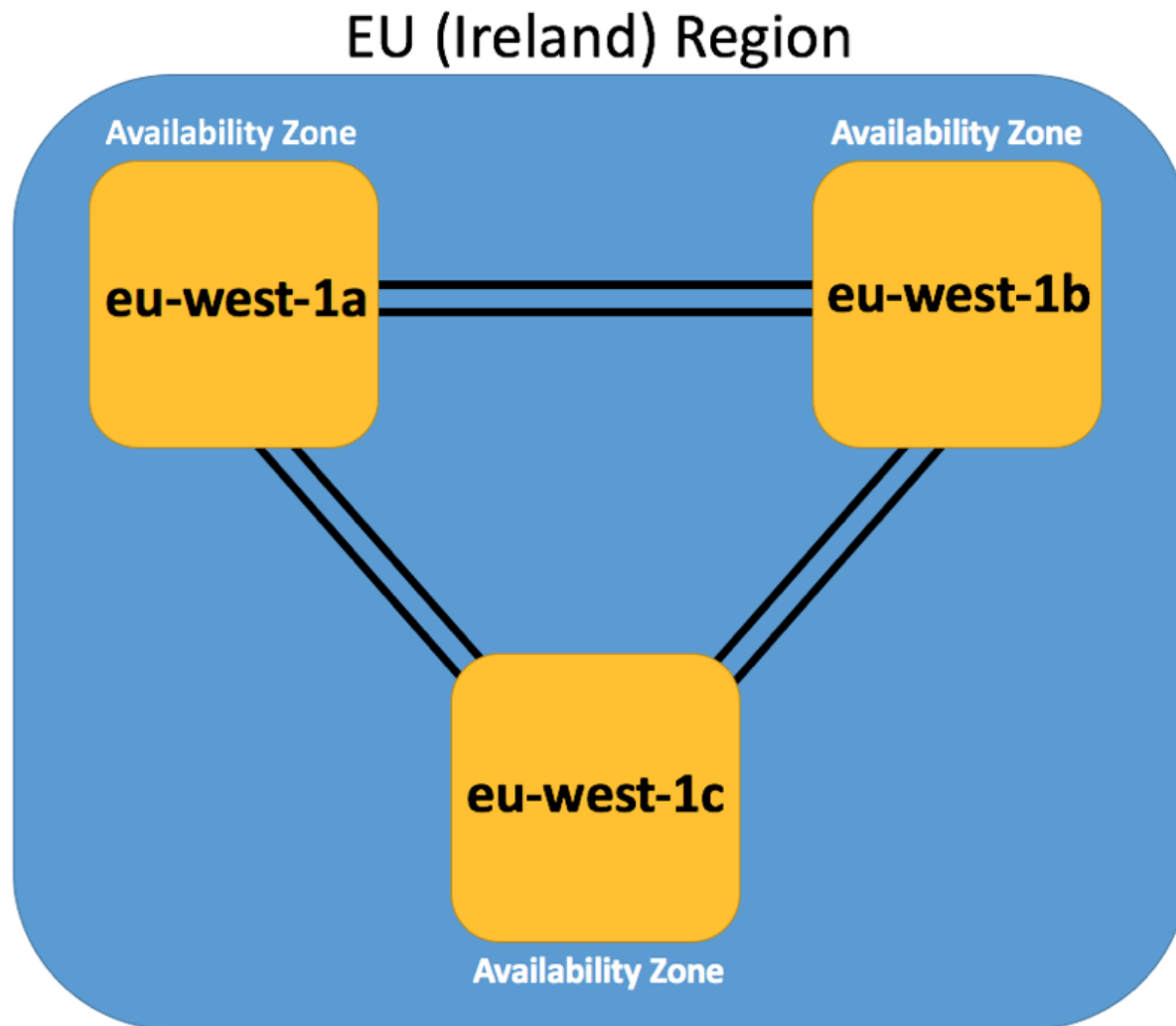
AWS REGIONS

Availability Zones are always referenced by their Code Name, which is defined by the AZs Region Code Name that the AZ belongs to, followed by a letter. For example, the AZs within the eu-west-1 region (EU Ireland), are:

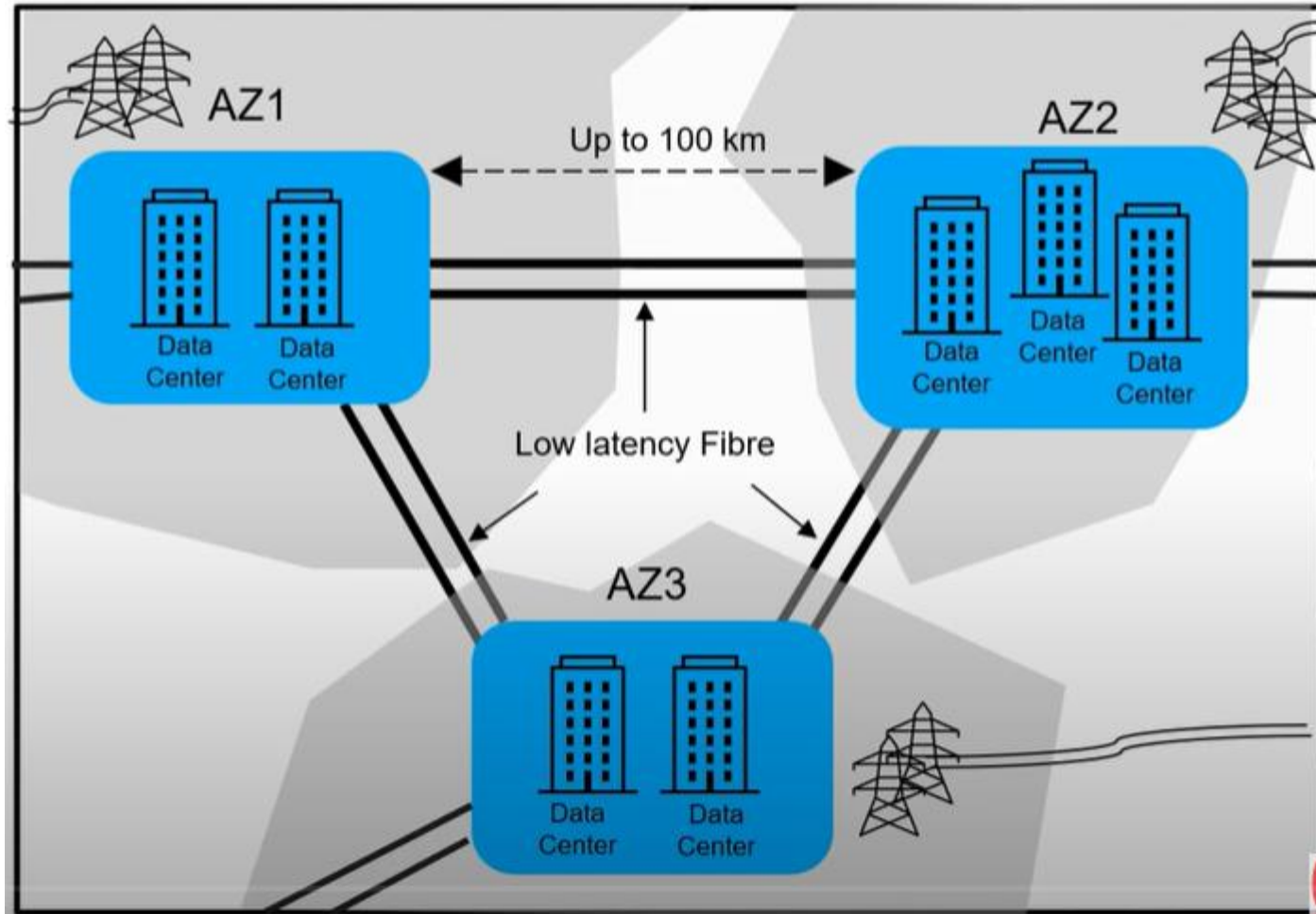
eu-west-1a

eu-west-1b

eu-west-1c



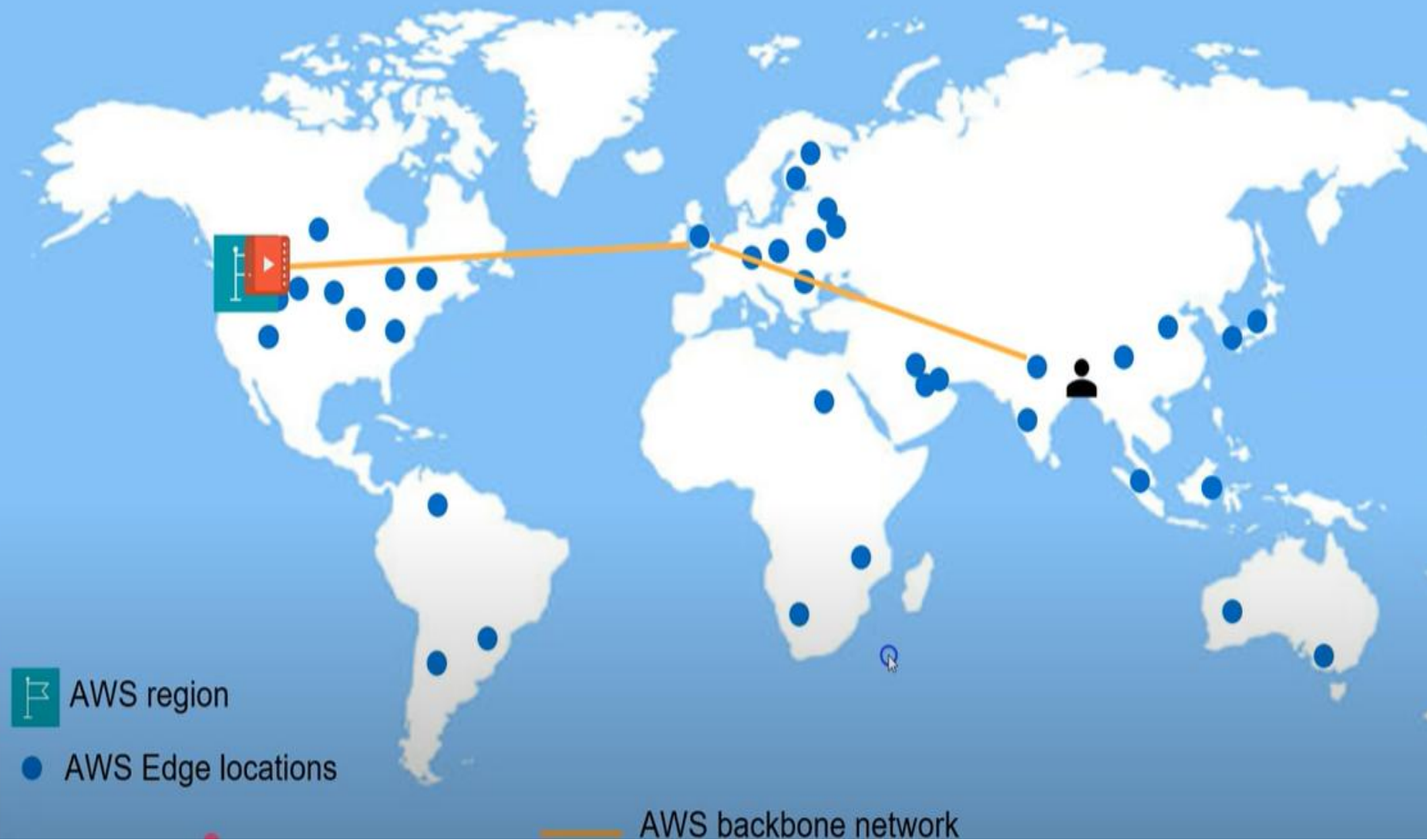
AWS Region



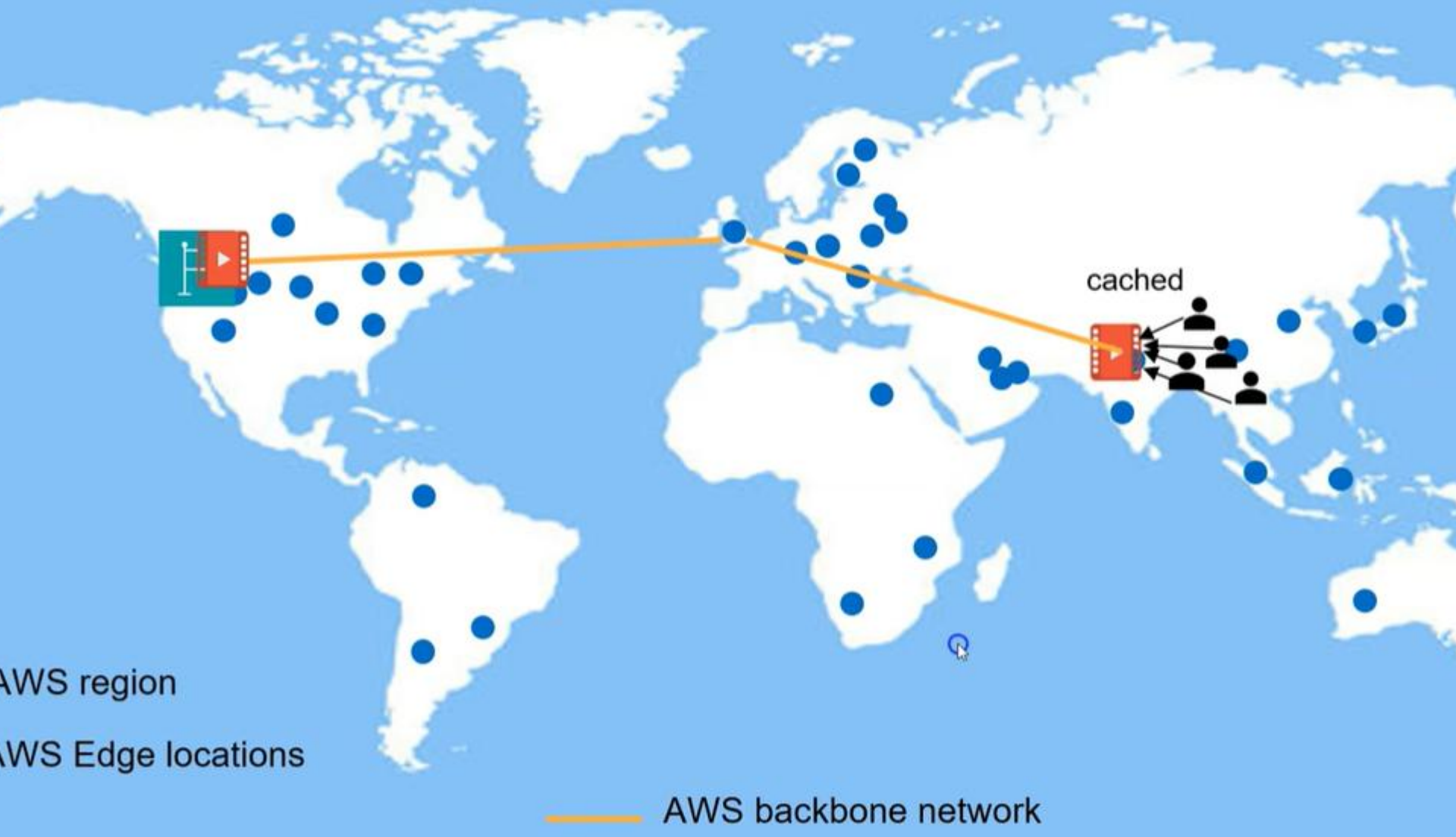
Without AWS edge network



Amazon CloudFront



Amazon CloudFront



Amazon CloudFront

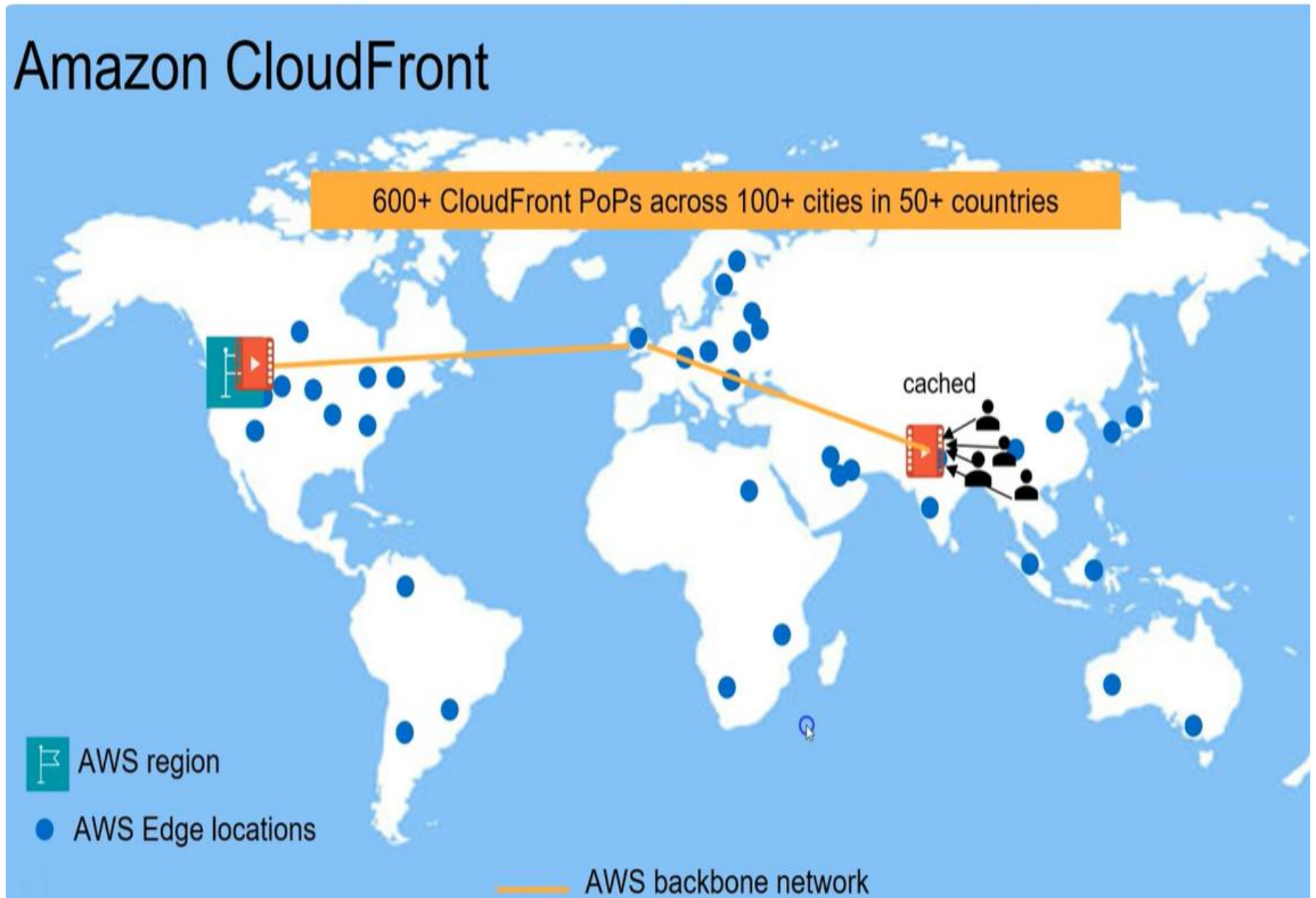
600+ CloudFront PoPs across 100+ cities in 50+ countries

 AWS region

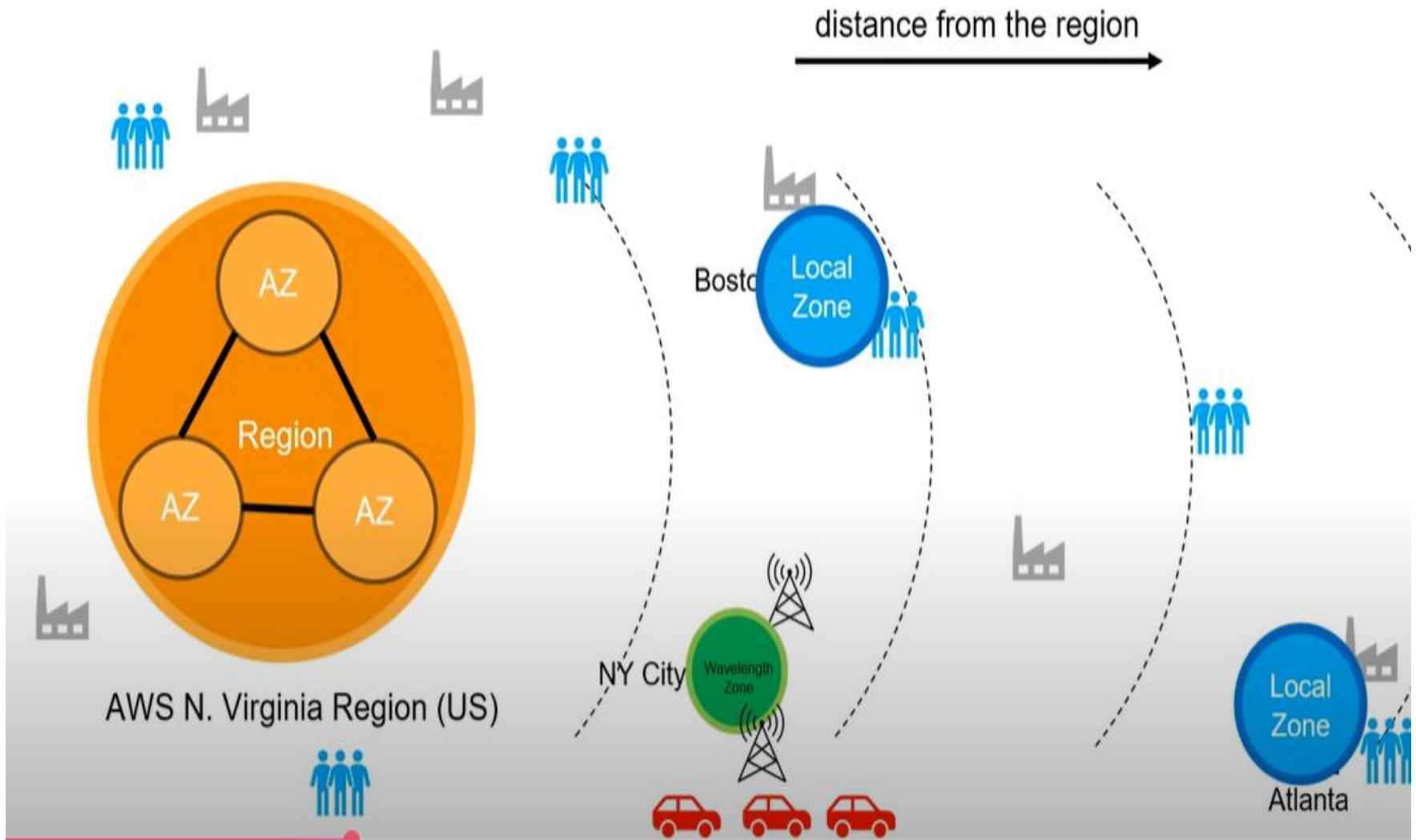
 AWS Edge locations

 AWS backbone network

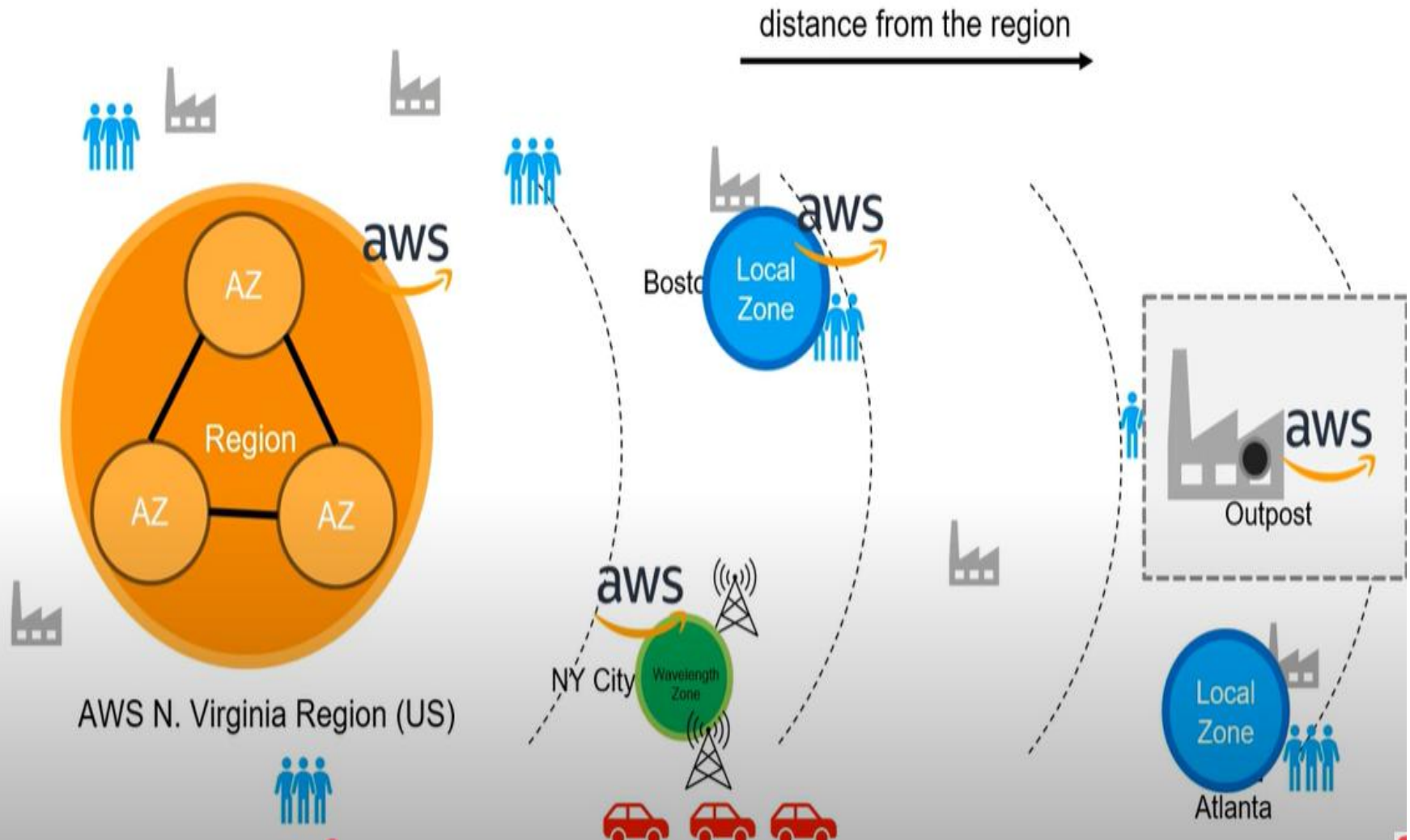
cached



Beyond the AWS Region..



Beyond the AWS Region..



Edge Locations:

Edge Locations are AWS sites deployed in major cities and highly populated areas across the globe.

While Edge Locations are not used to deploy your main infrastructures such as [EC2 instances](#), [EBS storage](#), VPCs, or RDS resources like AZs, they are used by AWS services such as **AWS CloudFront** and **AWS Lambda@Edge** (currently in Preview) to cache data and reduce latency for end-user access by using the Edge Locations as a global Content Delivery Network (CDN).

Regional Edge Cache:

In November 2016, AWS announced a new type of Edge Location, called a Regional Edge Cache.

These sit between your CloudFront Origin servers and the Edge Locations.

A Regional Edge Cache has a larger cache-width than each of the individual Edge Locations, and because data expires from the cache at the Edge Locations, the data is retained at the Regional Edge Caches.

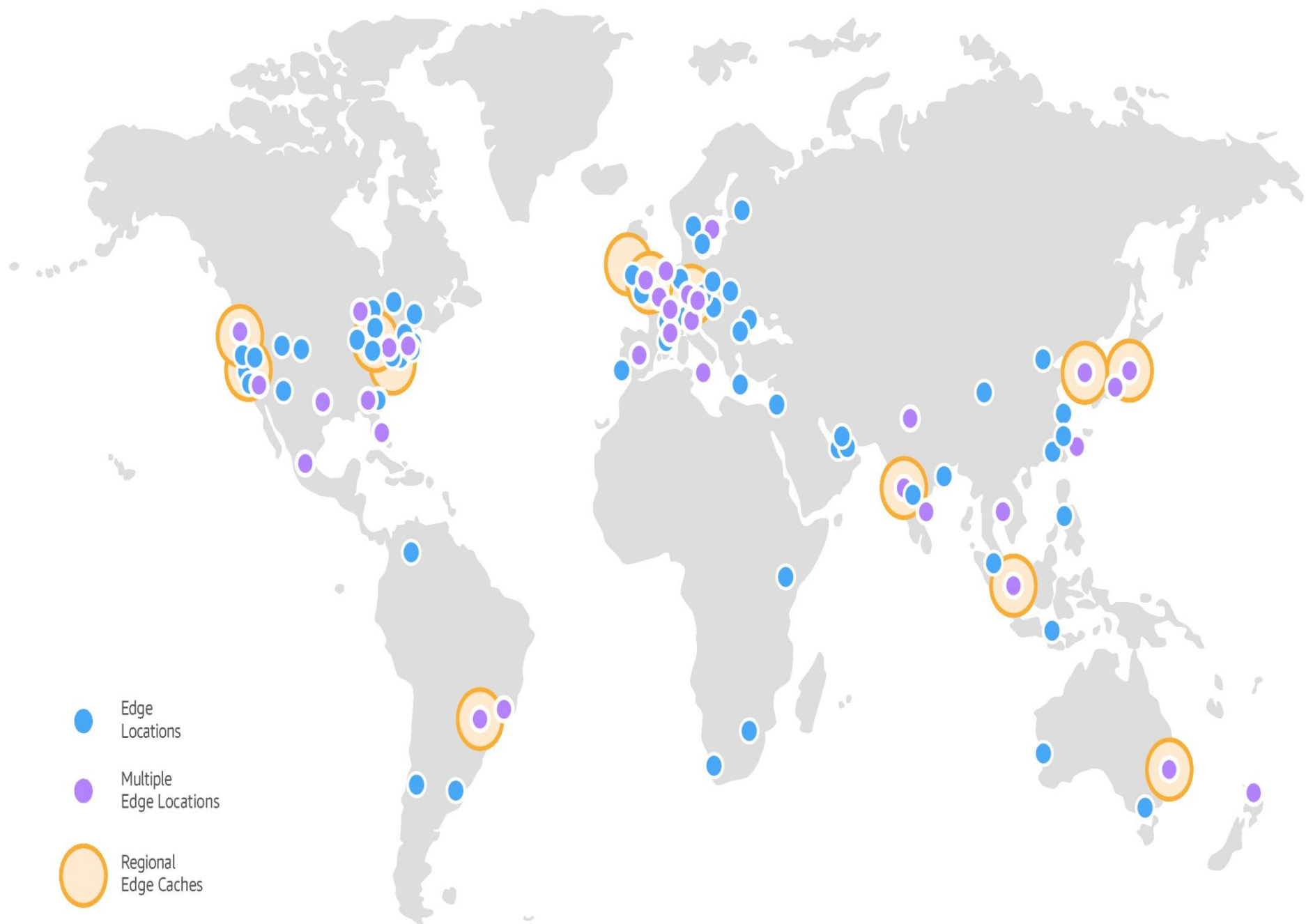
Therefore, when data is requested at the Edge Location that is no longer available, the Edge Location can retrieve the cached data from the Regional Edge Cache instead of the Origin servers, which would have a higher latency.

Local Zones:

a new type of infrastructure deployment designed to place core AWS Compute, Storage, Networking, and Database services near highly populated areas such as major cities that do not already have an AWS Region nearby.

AWS Local Zones allow customers in these areas to deploy resources and applications that require single-digit millisecond latency that would otherwise not be attainable given the geographic distance to the nearest Regions.

All AWS Local Zones are connected to a parent Region, allowing you to seamlessly connect to all other AWS services via a secure, dedicated high-speed connection.



- Edge Locations
- Multiple Edge Locations
- Regional Edge Caches

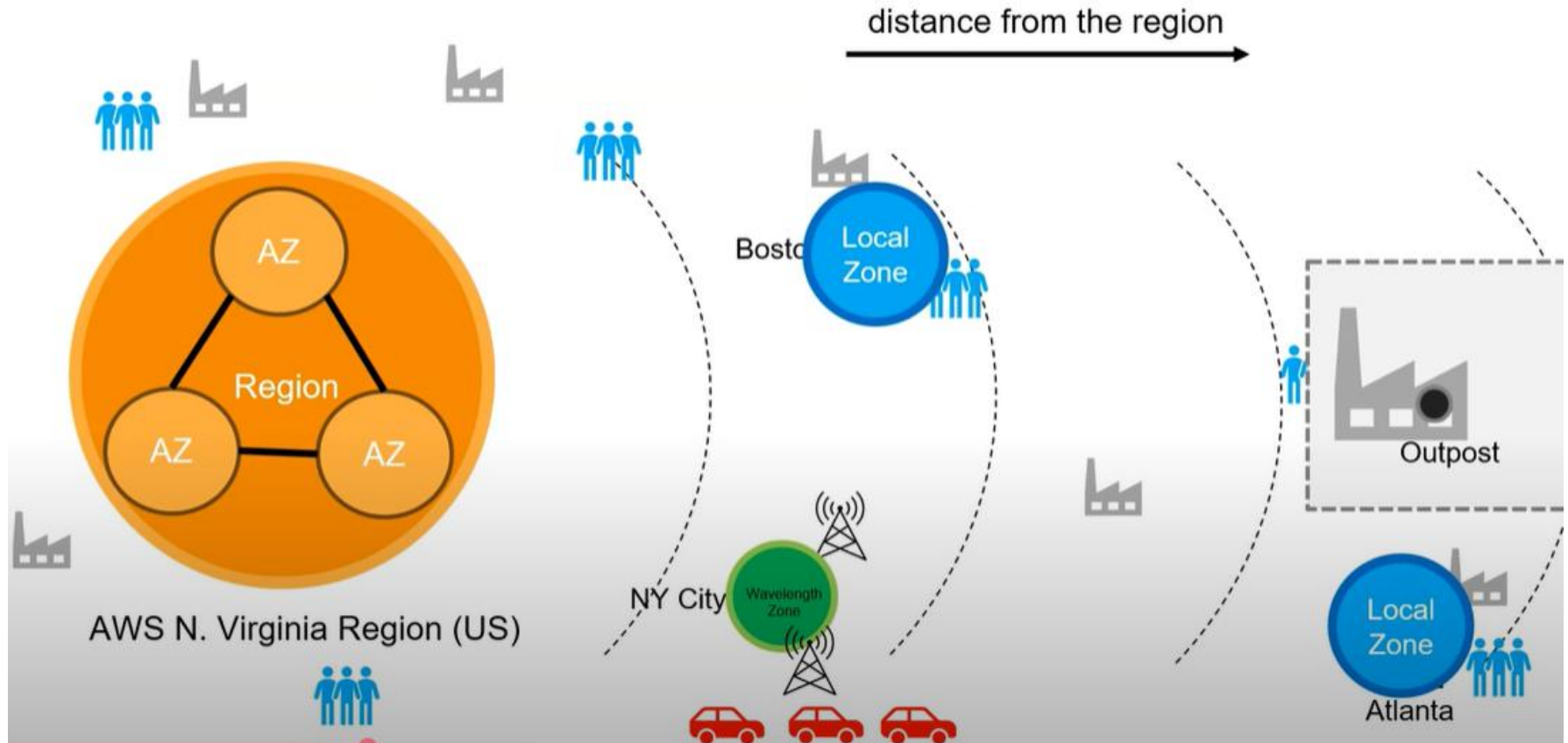
all Local Zones will be listed alongside the Availability Zones within that Region and can be selected when deploying everything from VPC subnets, to EC2 instances and EBS volumes, to ECS and EKS clusters.

In August 2023, AWS announced Dedicated Local Zones, which offer dedicated, fully managed infrastructure that is built for the exclusive use of a specific customer or community.

Dedicated Local Zones can be deployed in an existing on-premises data center or other locations that may be dictated by a customer or community's requirements to comply with security or other data sovereignty regulations for mission-critical and other sensitive workloads.

These are especially useful in the public sector and other industries where strict governance controls are necessary to comply with local laws and regulations.

Beyond the AWS Region..



Wavelength Zones

AWS Wavelength Zones also place core AWS services closer to large end user bases and are connected to a parent Region via a secure, dedicated high-speed connection.

AWS Wavelength Zones are embedded within 5G mobile broadband networks and are deployed within the data centers of large telecommunications providers.

Deploying AWS resources such as VPC subnets, EC2 instances, and EBS volumes to an AWS Wavelength Zone allows end users to connect to these resources without ever leaving the mobile provider's network.

By reducing the number of network hops and eliminating the need for any traffic to traverse the public internet, developers can offer ultra-low latency and increased reliability for 5G applications such as live video streaming and interactive gaming.

AWS Wavelength Zones are currently available through Verizon in the United States, KDDI in Japan, SK Telecom in South Korea, Vodafone in the UK and Germany, and Bell in Canada.

Outposts:

AWS Outposts brings the capabilities of the AWS cloud to your on-premises data center. This includes the same hardware used by AWS within their data centers, allowing you to use native AWS services, including the same tools and APIs you would use when running your infrastructure within AWS.

Outposts may be connected to AWS using either a Direct Connect or VPN connection.

Outposts allow you to run AWS services such as EC2, ECS, EKS, S3, RDS, and EMR on-premises.

AWS Outposts are fully managed, you do not need to maintain a level of patch management across your infrastructure or worry about installing or updating any software.

AWS will ensure your Outposts are patched and updated as needed.

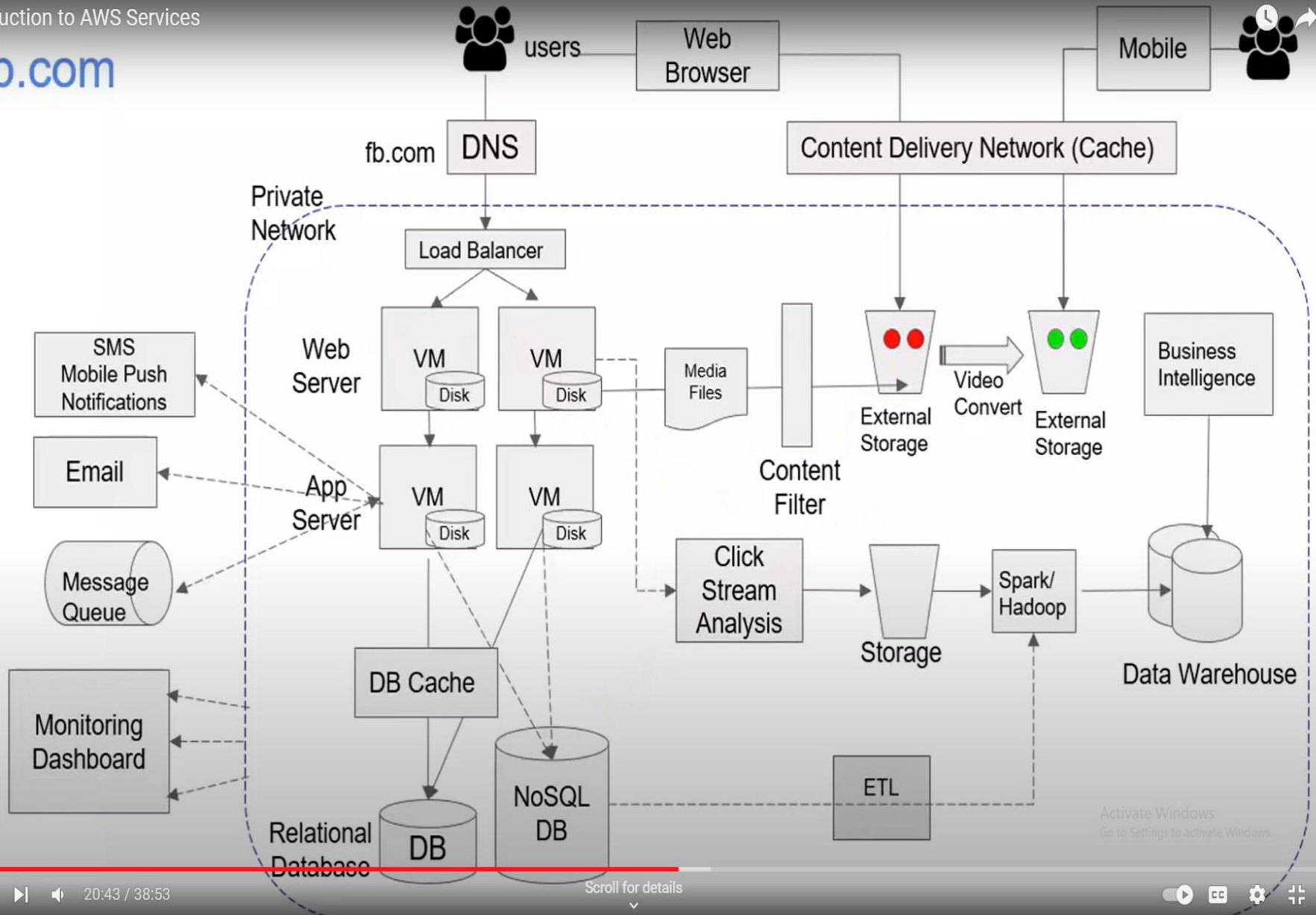
AWS services and service categories

Service Category	Description	Service Examples
Compute	Compute services are the processing power needed to run your application. The AWS compute portfolio provides tools to develop, deploy, run, and scale your applications in the AWS Cloud.	<ul style="list-style-type: none">• Amazon Elastic Compute Cloud (Amazon EC2)• AWS Elastic Beanstalk• AWS Fargate• AWS Lambda
Storage	Storage is one of the core building blocks of AWS. These services hold the information used by your applications. They provide reliable, scalable, and secure options to store, transmit, and back up your data.	<ul style="list-style-type: none">• Amazon Simple Storage Service (Amazon S3)• Amazon Elastic Block Store (Amazon EBS)• Amazon S3 Glacier
Networking & Content Delivery	Your network is the foundation of your infrastructure. AWS Networking & Content Delivery services are used to support your content delivery network, such as virtual private cloud, direct connections, and load balancing needs.	<ul style="list-style-type: none">• Amazon Virtual Private Cloud (Amazon VPC)• Amazon Route 53• Amazon CloudFront

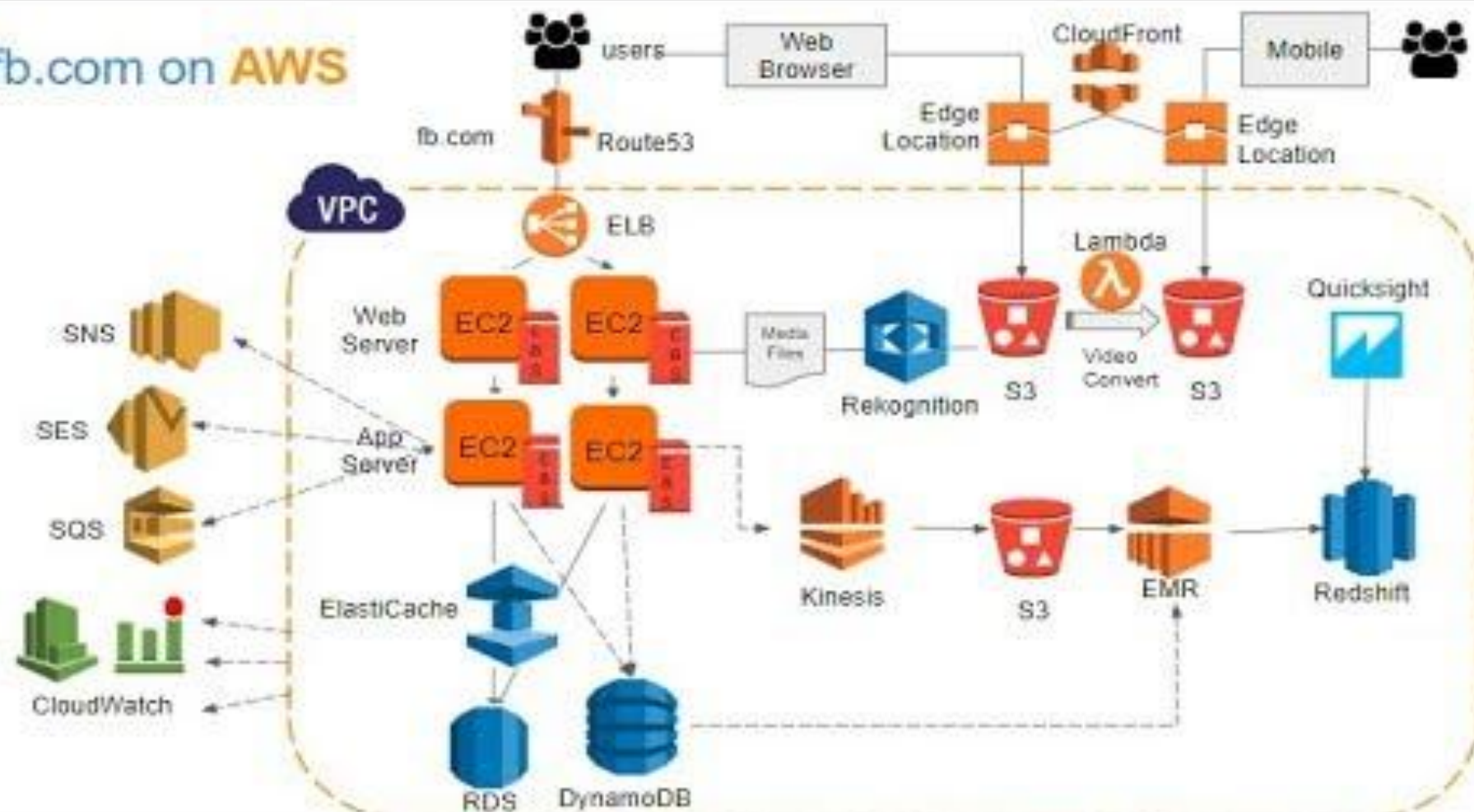
Database	Database services range from relational databases that are easy to set up and operate, to NoSQL databases built for higher demand and low latency—AWS offers a solution for all database needs.	<ul style="list-style-type: none"> • Amazon Aurora • Amazon Relational Database Service (Amazon RDS) • Amazon DynamoDB
Analytics	Analytics services enable you to transform raw data into meaningful information. They help you collect, visualize, and analyze your data to get the answers you need to run your business.	<ul style="list-style-type: none"> • Amazon Athena • Amazon Redshift • Amazon Kinesis
AWS Cost Management	Use these services to access information about your costs and usage, organize your costs across business lines, understand cost drivers and usage trends, set budgets, and optimize your AWS usage.	<ul style="list-style-type: none"> • AWS Cost Explorer • AWS Budgets • AWS Cost and Usage Report

Migration & Transfer	AWS offers services for migrating any workload—application, website, database, storage, physical or virtual server (or even an entire data center) to AWS.	<ul style="list-style-type: none">• AWS Database Migration Service• AWS Snowball• AWS DataSync
Security, Identity & Compliance	These services provide tools to achieve compliance and protect your infrastructure and data from both internal and external threats and exposures.	<ul style="list-style-type: none">• AWS Identity and Access Management (IAM)• Amazon Inspector• AWS Shield• AWS Security Hub
Management & Governance	This category includes end-to-end IT lifecycle management—control and secure your environment, reduce costs, simplify compliance, and enhance operational efficiency.	<ul style="list-style-type: none">• Amazon CloudWatch• AWS CloudFormation• AWS CloudTrail• AWS Trusted Advisor

fb.com

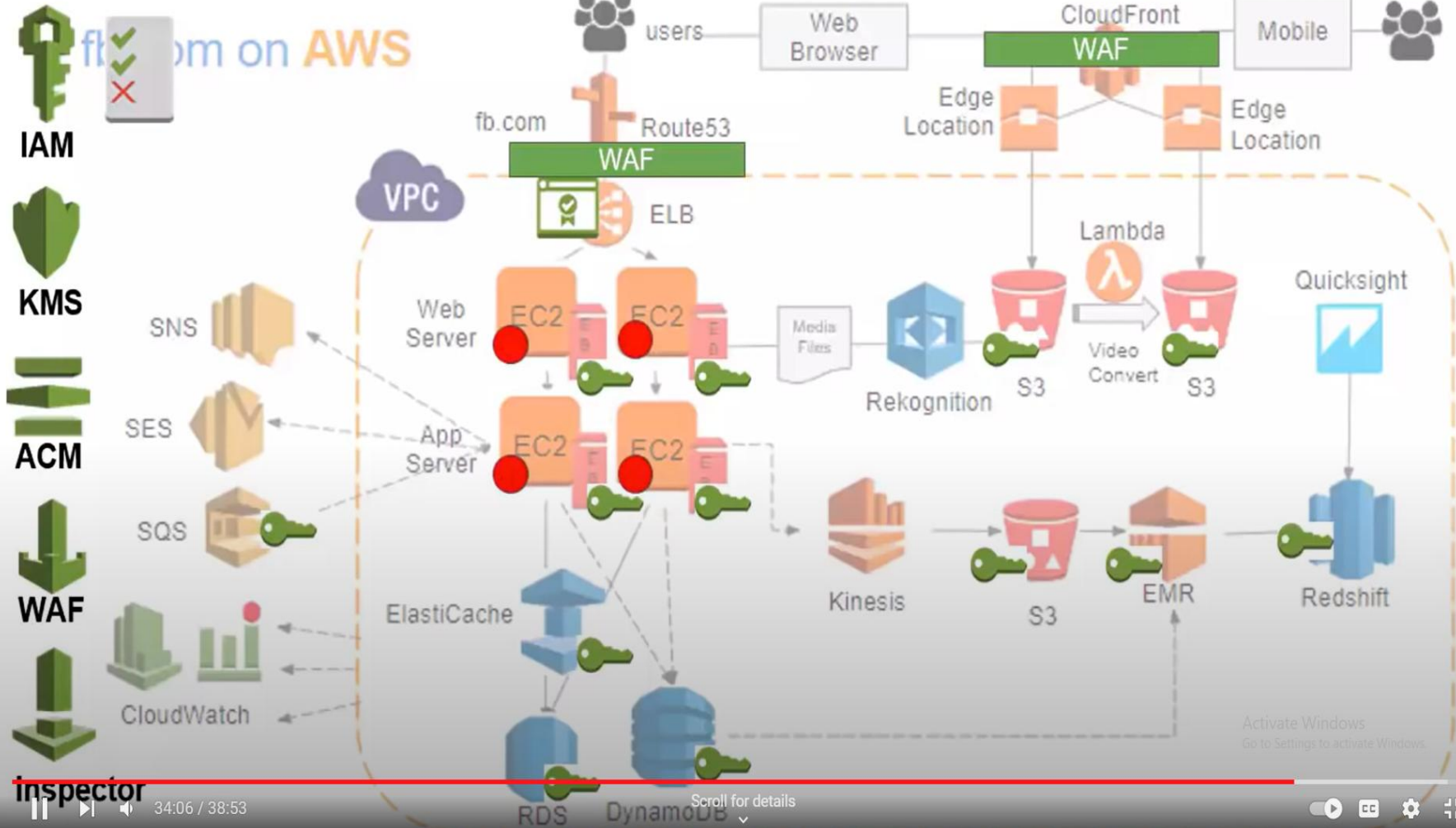


fb.com on AWS



AWS Security Services

Press **Esc** to exit full screen



Inspector

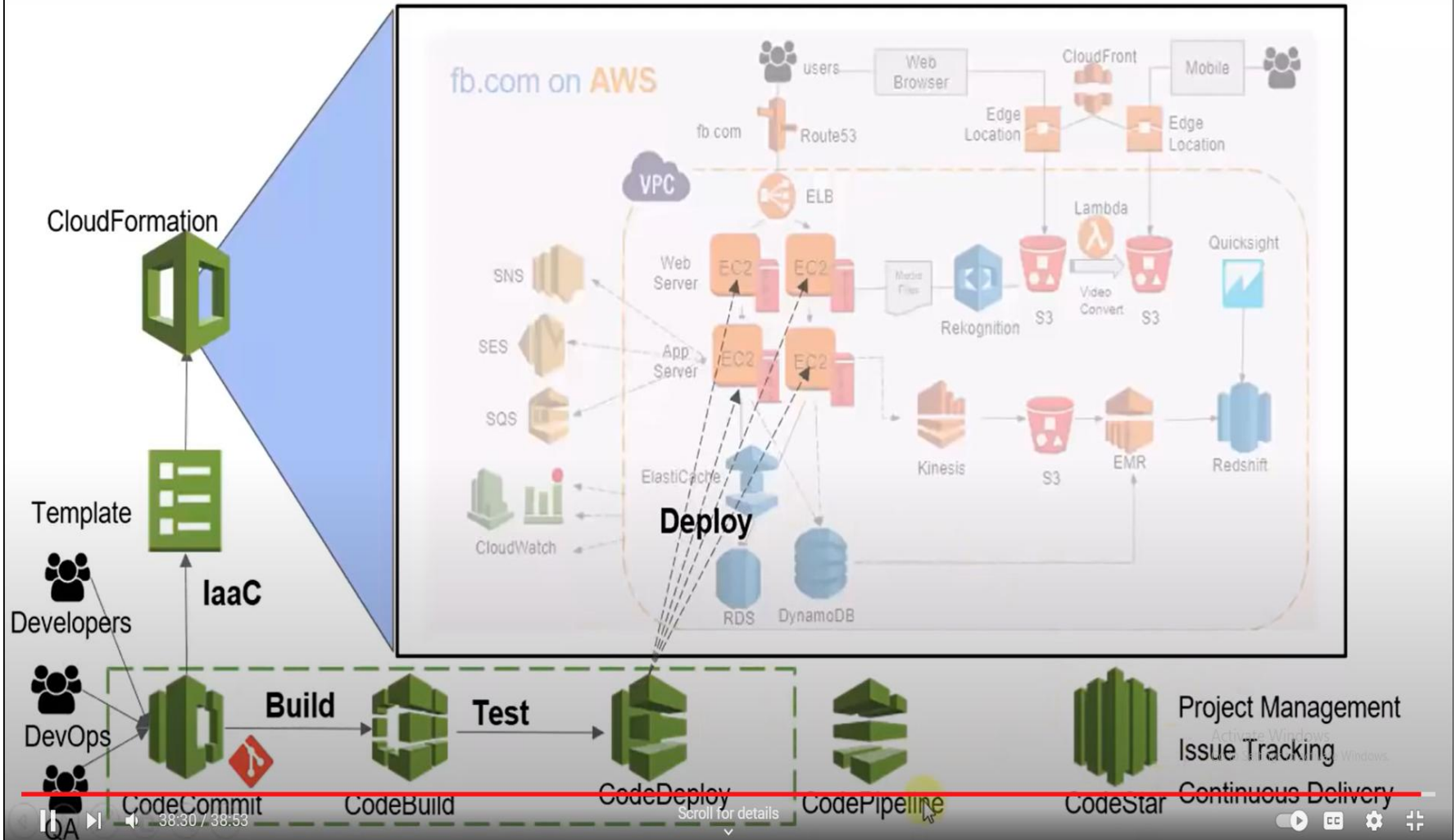
34:06 / 38:53

Scroll for details

Activate Windows
Go to Settings to activate Windows.

AWS Development and DevOps Services

AWS Region



AWS Analytics Services

Amazon Athena

Amazon Cloudsearch

Amazon Elasticsearch Service

Amazon EMR

Amazon FinSpace

Amazon Kinesis

Amazon Managed Streaming for Apache Kafka

Amazon Redshift

Amazon QuickSight

AWS Data Exchange

AWS Data Pipeline

AWS Glue

AWS Lake Formation

AWS Application Integration Services

AWS Step Functions

Amazon AppFlow

Amazon EventBridge

Amazon Managed Workflows for Apache Airflow

Amazon MQ

Amazon Simple Notification Service

Amazon Simple Queue Service

Amazon AppSync

AWS Blockchain Services

Amazon Managed Blockchain

Amazon Quantum Ledger Database (QLDB)

AWS Business Applications Services

Alexa for Business

Amazon Chime

Amazon Honeycode

Amazon WorkDocs

Amazon WorkMail

AWS Cloud Financial Management Services

AWS Cost Explorer

AWS Budgets

AWS Cost and Usage Report

Reserved Instance Reporting

Savings Plans

AWS Compute Services

Amazon EC2 Auto Scaling

AWS App Runner

AWS Batch

AWS Elastic Beanstalk

AWS Outposts

AWS Serverless Application Repository

AWS Snow Family

AWS Wavelength

VMWare Cloud on AWS

AWS Container Services

Amazon Elastic Container Registry

Amazon Elastic Container Service (ECS)

Amazon ECS Anywhere

Amazon Elastic Kubernetes Service

Amazon EKS Distro

AWS App2Container

AWS Copilot

AWS Fargate

Red Hat OpenShift Service on AWS

AWS Customer Engagement Services

Amazon Connect

Amazon Pinpoint

Amazon Simple Email Service

AWS Database Services

Amazon Aurora

Amazon Quantum Ledger Database

Amazon DynamoDB

Amazon RDS

Amazon DocumentDB

Amazon RDS on VMware

Amazon ElastiCache

Amazon Redshift

Amazon Keyspaces

Amazon Timestream

Amazon Neptune

Amazon Database Migration Service

AWS Glue

AWS Developer Tools

Amazon CodeGuru

Amazon Correto

AWS Cloud Development Kit

AWS Cloud9

AWS CloudShell

AWS CodeArtifact

AWS CodeBuild

AWS CodeCommit

AWS CodeDeploy

AWS CodePipeline

AWS CodeStar

AWS Command Line Interface

AWS Device Farm

AWS Fault Injection Simulator

AWS Tools and SDKs

AWS X-Ray

AWS End User Computing Services

Amazon AppStream 2.0

Amazon WorkDocs

Amazon WorkLink

Amazon WorkSpaces

AWS Front-End Web and Mobile Services

AWS Amplify

Amazon API Gateway

Amazon Location Service

Amazon Pinpoint

AWS Device Farm

AWS Game Tech Services

Amazon GameLift

Amazon Lumberyard

AWS Internet of Things Services

AWS IoT Core

AWS IoT Events

AWS Greengrass

AWS IoT SiteWise

AWS IoT 1-Click

AWS IoT Things Graph

AWS IoT Analytics

AWS Partner Device Catalog

AWS IoT Button

FreeRTOS

AWS IoT Device Defender

AWS IoT Management

AWS Machine Learning Services

Amazon SageMaker

Amazon Augmented AI

Amazon CodeGuru

Amazon Comprehend

Amazon DevOps Guru

Amazon Elastic Inference

Amazon Forecast

Amazon Fraud Detector

Amazon Healthlake

Amazon Kendra

Amazon Lex

Amazon Lookout for Equipment

Amazon Lookout for Metrics

Amazon Lookout for Vision

Amazon Minitron

Amazon Personalize

Amazon Polly

Amazon Rekognition

Amazon SageMaker Data Wrangler

Amazon SageMaker Ground Truth

Amazon Textract

Amazon Translate

Amazon Transcribe

Amazon Deep Learning AMIs

AWS Deep Learning Containers

AWS DeepComposer

AWS DeepLens

AWS DeepRacer

AWS Inferentia

AWS Panorama

PyTorch on AWS

Apache MXNet on AWS

TensorFlow on AWS

AWS Management and Governance Services

Amazon CloudWatch	AWS Distro for OpenTelemetry
AWS Auto Scaling	AWS Launch Wizard
AWS Chatbot	AWS License Manager
AWS CloudFormation	AWS Management Console
AWS CloudTrail	AWS Managed Services
AWS Compute Optimizer	Amazon Managed Service for Grafana
AWS Config	AWS Management Service for Prometheus
AWS Control Tower	AWS OpsWorks
AWS Console Mobile Application	AWS Organizations
	AWS Personal Health Dashboard

AWS Proton

AWS Service Catalog

AWS Systems Manager

AWS Trusted Advisor

AWS Well-Architected Tool

AWS Migration and Transfer Services

AWS Migration Hub

AWS Application Discovery Service

AWS Application Migration Service
(CloudEndure Migration)

AWS Database Migration Service

AWS Transfer Family

Migration Evaluator (formerly TSO Logic)

AWS Media Services

Amazon Elastic Transcoder

Amazon Interactive Video Service

Amazon Kinesis Video Streams

AWS Elemental MediaConnect

AWS Elemental MediaConvert

AWS Elemental MediaLive

AWS Elemental MediaPackage

AWS Elemental MediaStore

AWS Elemental MediaTailor

AWS Elemental Appliances and Software

AWS Nimble Studio

AWS Networking and Content Delivery Services

Amazon VPC

Amazon CloudFront

Amazon Route 53

Amazon PrivateLink

AWS App Mesh

AWS Cloud Map

AWS Direct Connect

AWS Global Accelerator

AWS Transit Gateway

Elastic Load Balancing

AWS Quantum Technologies Services

Amazon Braket

AWS Robotics Services

AWS RoboMaker

AWS Satellite Services

AWS Ground Station

AWS VR and AR Services

Amazon Sumerian

AWS Storage Services

Amazon Simple Storage System (S3)

Amazon Elastic Block Store (EBS)

Amazon Elastic File System (EFS)

Amazon FSx for Lustre

Amazon FSx for Windows File Server

Amazon S3 Glacier

AWS Backup

AWS Snow Family

AWS Storage Gateway

CloudEndure Disaster Recovery

AWS Serverless Services

AWS Lambda

Amazon API Gateway

Amazon DynamoDB

Amazon EventBridge

Amazon Simple Notification System (SNS)

Amazon Simple Queue Service (SQS)

AWS Fargate

AWS Step Functions

AWS Security, Identity, and Compliance Services

AWS Identity & Access Management

AWS Directory Service

Amazon Cognito

AWS Firewall Manager

Amazon Detective

AWS Resource Access Manager

Amazon GuardDuty

AWS Secrets Manager

Amazon Inspector

AWS Security Hub

Amazon Macie

AWS Shield

AWS Artifact

AWS Single Sign-On

AWS Audit Manager

AWS WAF

AWS Certificate Manager

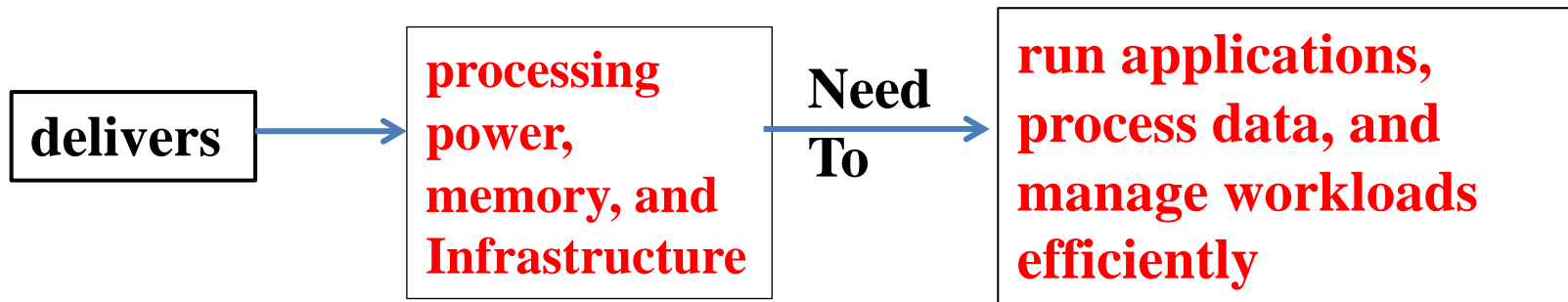
AWS CloudHSM

AWS Compute services overview

AWS compute services are a **line of infrastructure as a service (IaaS) products** from Amazon Web Services (AWS) that allow customers to **purchase computing power** as needed instead of maintaining their own physical servers.

AWS Compute was one of the **first services to be provided by AWS back in 2006** providing only a **single EC-2 instance** type.

AWS Compute Services refers to a **set of cloud-based solutions** provided by Amazon Web Services (AWS) that

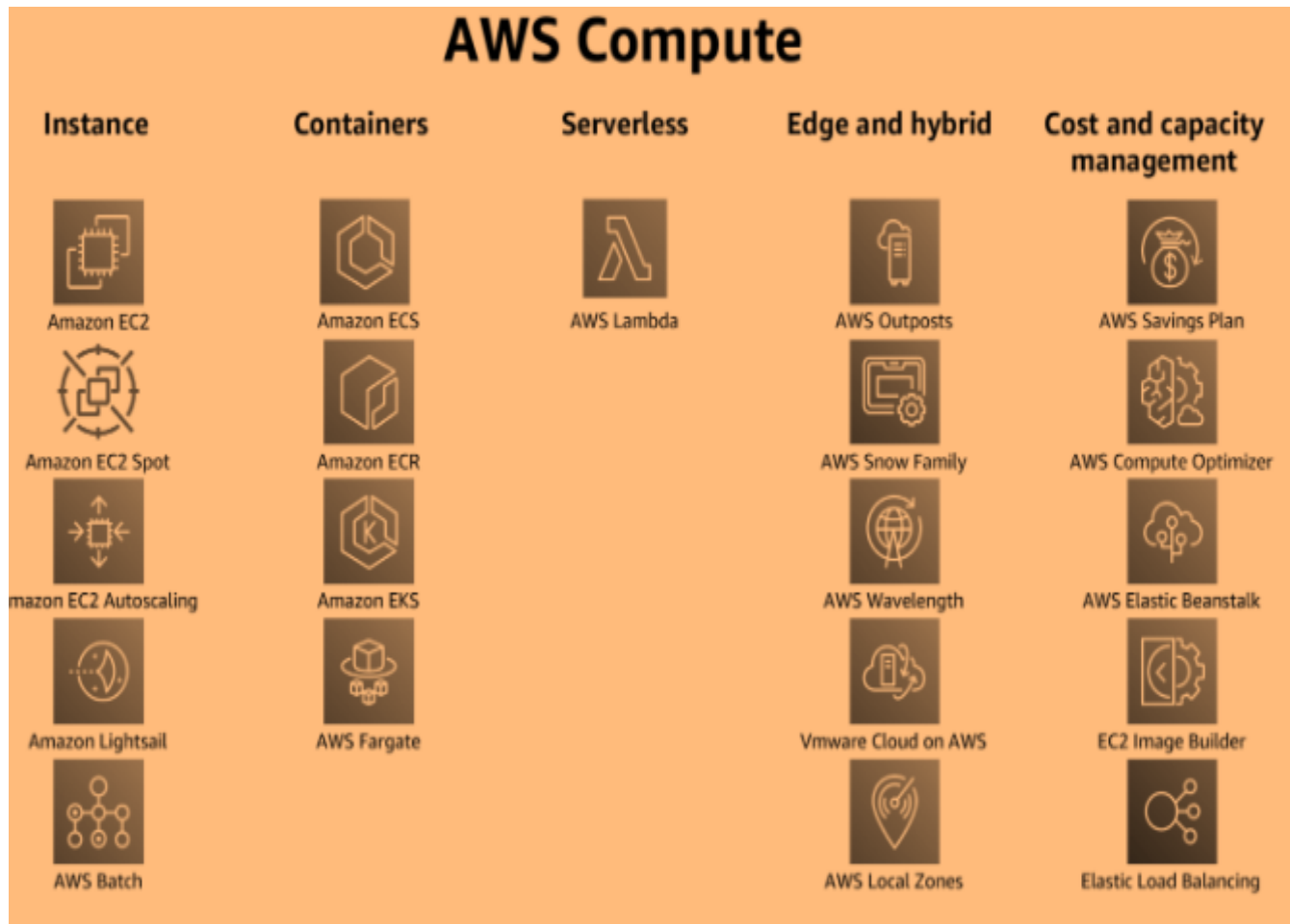


Instead of managing physical hardware or data centers,

AWS Compute Services allows users to run and scale their applications in the cloud, providing flexibility, scalability, and cost-efficiency.

The AWS compute platform includes **several different categories**, such as:

- Instances (virtual machines)
- Containers
- Serverless
- Edge and Hybrid locations
- Cost and capacity management



Category	Product name	Description
Instances	Amazon Elastic Compute Cloud (EC2)	Lets you rent and manage virtual servers in the cloud with elastic compute power. Uses an on-demand pricing model.
	Amazon Lightsail	Lets you quickly launch resources for small projects. Works best for preconfigured apps, like setting up a WordPress site.
	Amazon Batch	Lets you process large workloads in smaller chunks, or batches.
Containers	Amazon Elastic Container Service (ECS)	Container orchestration tool that lets you deploy and manage Docker containers.
	Amazon ECS Anywhere	Lets you run self-managed containers on your own infrastructure.
	Amazon Elastic Container Registry (ECR)	Lets you deploy container images to Amazon ECS, EKS, or on-premises.
	Amazon Elastic Kubernetes Service (EKS)	Lets you run Kubernetes applications and clusters in AWS cloud or on-premises.
	Amazon EKS Anywhere	Lets you run Kubernetes apps on your own infrastructure, similar to how ECS Anywhere lets you run self-managed containers.
	AWS Fargate	Lets you deploy and manage Docker containers using a serverless compute engine
	AWS App Runner	Lets you run containerized web apps without having to configure orchestrations, manage servers, or build pipelines.

Category	Product name	Description
Serverless	AWS Lambda	A serverless compute service that lets you run code without having to manage servers or clusters.
Edge and hybrid	AWS Outposts	Lets you take advantage of hybrid cloud by delivering AWS infrastructure and services to your on-premises or edge locations.
	AWS Snow Family	Lets you securely move data in and out of AWS through several different hardware options, as well as operate edge locations
	AWS Wavelength	Lets you take advantage of mobile edge computing infrastructure for fast applications on 5G devices
	VMware Cloud on AWS	Lets you run VMware workloads on AWS infrastructure through a managed service
	AWS Local Zones	Lets you run workloads in closer physical proximity to end users for latency-sensitive applications
Cost and capacity management	AWS Savings Plan	Lets you save up to 72 percent on compute services through a flexible pricing model
	AWS Compute Optimizer	Lets you improve improve price performance and optimize costs through automated recommendations from AWS
	AWS Elastic Beanstalk	Lets you easily deploy applications by automatically managing the AWS services you need to host and scale your application.
	EC2 Image Builder	Lets you easily and quickly build and maintain secure Linux or Windows Server images.
	Elastic Load Balancing (ELB)	Lets you improve availability by automatically distributing traffic across one or more Availability Zones (AZs).

Features of AWS Compute:

1. **Scalability:** These services let you **scale up or down** based on how much traffic you get. So, when your app gets popular, AWS can add more resources, and when things slow down, it scales back so you're not wasting money.
2. **Reliability:** AWS is pretty reliable because **they spread your stuff across different data centers**. If one goes down, your app keeps running somewhere else.
3. **Wide scope of categories:** AWS Compute encompasses a **wide range of services for different use cases**, from simple virtual machines (EC2), to serverless computing (AWS Lambda), to containerized applications (ECS, EKS), and even bare-metal compute resources (EC2 Bare Metal). This diversity means you can choose the right tool for any workload.
4. **Ethernet speed:** AWS offers **high-performance networking capabilities**. Instances in EC2 can be connected with high-bandwidth, low-latency networking features, which is essential for demanding applications like real-time analytics, gaming, or media streaming. This is made possible through technologies like **Elastic Network Adapter (ENA) and enhanced networking**.
5. **Multiple locations:** AWS operates across multiple regions and availability zones globally. This allows **you to run your workloads closer to your end users**, improving performance and reducing latency. It also provides resilience, as you can distribute resources across different geographical areas to maintain availability and disaster recovery capabilities.

1. Amazon Elastic Compute Cloud (EC2)

The EC2 service, which stands for Elastic Compute Cloud, enables anyone **to rent servers.**

Amazon EC2 is one of the **most prominent services of AWS** that has helped **businesses to run their applications over the public cloud seamlessly.**

The term **"Elastic"** in Amazon EC2 (Elastic Compute Cloud) refers to the **ability to quickly and dynamically scale compute resources up or down based on demand.**

With the use of the EC2 interface, the **developers can integrate several virtual machine instances and configure their scaling capacity.** The purpose of integrating Amazon EC2 into the business environment is to build apps for automating the scaling aspects based upon the needs and requirements at certain instances. Along with that, it becomes really easy to manage the storage and deploy virtual servers over the cloud with the use of Amazon EC2.

Amazon EC2 is **billed based on the size and hours of instance, operating system, and region.** Apart from that, the billing is also done based upon **resource consumption.** Setting up Amazon EC2 for your business operations is quite convenient and easy.

AMI(Amazon Machine Image) In AWS

An Amazon Machine Image (AMI) is a **template that contains all the information you need to launch a new EC2 instance**. AMI must be specified when an instance is launched.

An AMI can be used to launch **multiple new EC2 instances, all with the same configuration**.

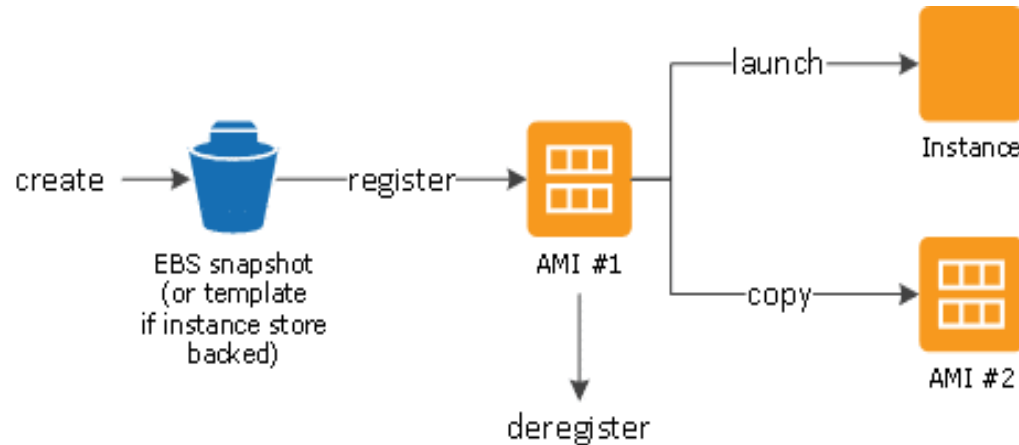
AMIs allow you to **pre-install any required packages** on an EC2 instance, **improve boot time, implement security controls for all instances** simultaneously, and launch **production-ready instances quickly when needed**.

AMIs provide a **blueprint for the root volume** which is needed to launch a particular instance. This blueprint typically includes the operating system, an application server, and applications.

Key Terminology associated with AWS AMI

- **EBS Snapshots** – To successfully meet the challenges related to storing data in the cloud, Amazon provides a storage service to be used with your EC2 instances, called Amazon Elastic Block Storage or Amazon EBS.
- **EC2 Instance** – Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud. Using Amazon EC2 eliminates your need to invest in hardware up-front so that you can develop and deploy applications faster.
- **Amazon S3** – Amazon Simple Storage Service (Amazon S3) is storage for the Internet. It is designed to make web-scale computing easier.

Lifecycle of Amazon Machine Image (AWS AMI)



Once an **AMI is created and registered, you can use it to launch new instances**. AMI can be copied to the same as well as different AWS Regions. Once the AMI is no longer required, you can deregister it.

Few AMIs are provided by Amazon and you can search for an AMI that meets the criteria for your instance. Certain AMIs are also provided by the community. Once the AMI is selected, you can launch an instance. It works just like using any other server.

After an AMI is created, it can either be **kept private or shared with a specific list of AWS accounts**. Once your work with the AMI is completed, you can deregister the AMI. After deregistering, the AMI can't be used to launch new instances but the existing instances that are already using AMI will not be affected.

AMI can be selected Based on the following characteristics

- **Regions** – Each region is designed to be isolated from the other regions to achieve the greatest possible fault tolerance and stability. Your account will determine the regions that are available to you.
- **Operating System** – Operating system types like Linux, macOS, Raspbian, and Windows Server are provided by AWS for the user to choose from.
- **Architecture** – There is a choice between 32-bit and 64-bit for the user to choose between.
- **Launch Permissions** – The launch permissions determine the availability to be either public, explicit and implicit.

Launch Permission	Description
public	The owner grants launch permissions to all AWS accounts.
explicit	The owner grants launch permissions to specific AWS accounts.
implicit	The owner has implicit launch permissions for an AMI.

- **backed by Amazon EBS** – When AMI is categorized as backed by Amazon EBS it means that the root device for an instance launched from the AMAZON AMI is an Amazon Elastic Block Store (Amazon EBS) volume created from an Amazon EBS snapshot
- **backed by instance store** – When backed by instance store, it means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

Types of AMI Images

Public Images

These are public-access machine images that are safe, secure, and customized.

Paid Images

These are images that a developer can sell to you. The images are available for purchase through the AWS Marketplace, an online store that offers software that runs on Amazon Web Services.

Private Images

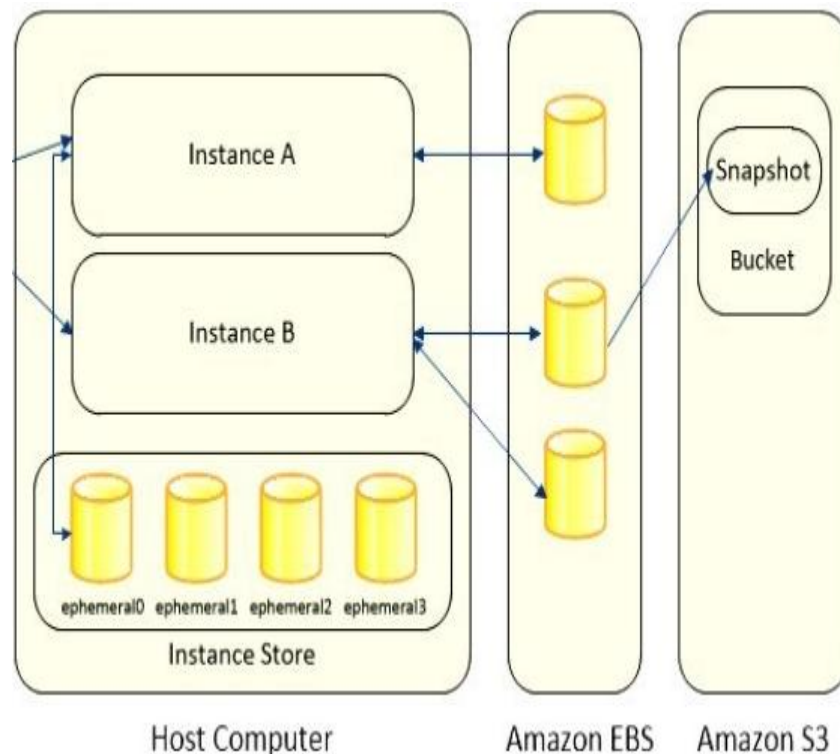
Only EC2 users who have been granted access to a private machine image by the developer can access it.

EC2 Volumes

EC2 Volumes (**called Elastic Block Storage by Amazon**) are essentially **disk images that can be mounted on any system running on EC2**, and continue to exist even if the system they were attached to is deleted.

They are a more permanent place to store data than the file system of an EC2 instance, and more easily accessible than S3.

The size of a volume is typically several GB, and is chosen at creation time. Amazon charges for volumes based on their size and the amount of time they exist.



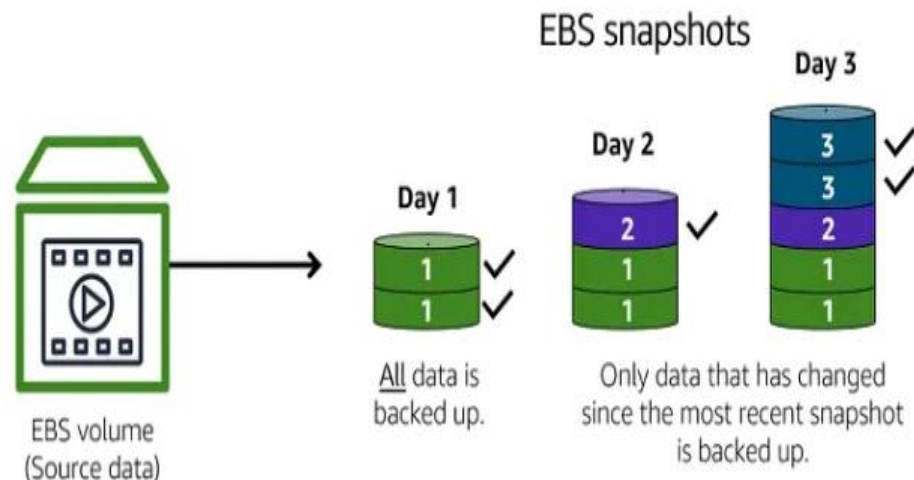
Snapshots

A snapshot is a base feature for **creating backups of your EBS volumes**.

A snapshot takes **a copy of the EBS volume and places it in Amazon S3**, where it is stored redundantly in multiple Availability Zones.

The initial snapshot is a full copy of the volume; ongoing snapshots store **incremental block-level changes only**.

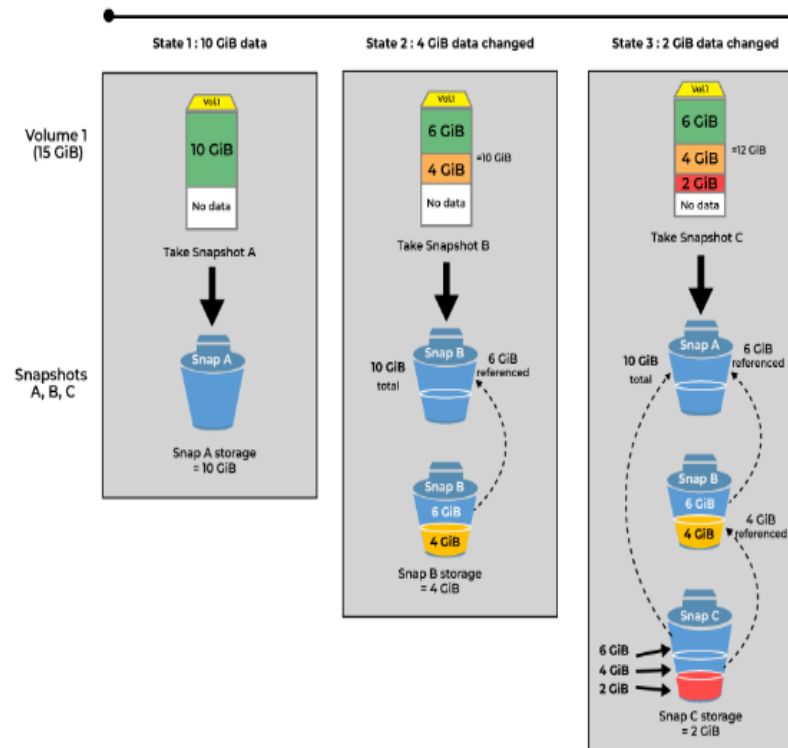
It is used to minimizing unnecessary storage, low cost, and speed backup. charges for your snapshot based on how amount of data stored in that snapshot. you cannot access the data in a snapshot. snapshot is used to restore the data in a specific state.



How Do Snapshots Work?

Snapshot is a backup of root storage that is attached to the EC2-Instance. The snapshot which you are going to take at first it consist of all the data which is present in the EBS eventually snapshots you are going to take will consist only the data which is newly added and this snapshots are also called incremental snapshots.

AWS will **charge you based on the amount of data is going to be backed up in single snapshot**. Snapshot source volume will not decide the cost will decide by the amount of data is being backed up. For example if size of EBS volume is 100 GiB and the data you have been backed up is 60 GiB using snapshot then amazon will only charges you for the 60 GiB only.



Amazon Simple Storage Services (Amazon S3)

Object storage

- Object storage is a technology that stores and manages data in an **unstructured format called objects**. Modern organizations create and analyze large volumes of unstructured data such as **photos, videos, email, web pages, sensor data, and audio files**.
- Cloud object storage systems distribute this data across multiple physical devices but allow users to access the content efficiently from a single, virtual storage repository.
- Object storage delivers **unlimited scalability at low per-gigabyte prices**.
- In object storage, objects are kept in a **single bucket**. Object storage combines the **pieces of data that make up a file, adds all the user-created metadata to that file, and attaches a custom identifier**. This lets you retrieve and analyze any object in the bucket, no matter the file type, based on its function and characteristics.

Amazon Simple Storage Services (Amazon S3)

- Amazon Simple Storage Service (Amazon S3) is **an object storage service** that offers industry-leading **scalability, data availability, security, and performance**.
- Customers of all sizes and industries can use Amazon S3 to store and protect any amount of data for a range of use cases, such as **data lakes, websites, mobile applications, backup and restore, archive, enterprise applications, IoT devices, and big data analytics**.
- Amazon S3 provides **management features** so that you can optimize, organize, and configure access to your data to meet your specific business, organizational, and compliance requirements.

Amazon S3 Core Components — Buckets and Objects

For Organizing, storing and retrieving data in Amazon S3 focuses on two key components:

Buckets and

Objects

that work together to create the storage system.

Amazon Simple Storage Services (Amazon S3)

Buckets

A bucket is a **container for objects**. Objects are saved in the buckets in Amazon S3. In order to store your data in Amazon S3, you first **create a bucket and specify a bucket name and AWS Region**. Then, you upload your data to that bucket as objects in Amazon S3.

It's also important to know that Amazon S3 buckets are globally unique. No other AWS account in the same region can have the same bucket names as yours unless you first delete your own buckets.

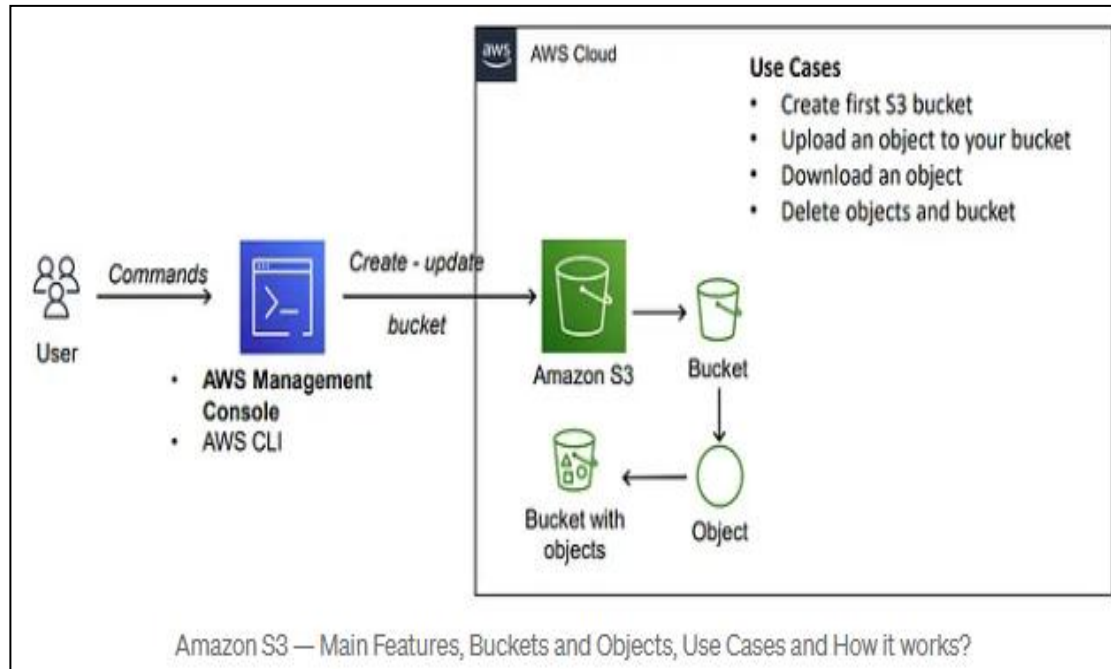
Objects

Objects are the fundamental entities stored in Amazon S3. Amazon S3 is an object storage service that stores data as objects within buckets. Objects are **data files, including documents, photos, videos and any metadata that describes the file**. Each **object has a key** (or key name), which is the unique identifier for the object within the bucket.

Objects consist of **object data and metadata**. The metadata is a set of name-value pairs that describe the object. These pairs include some default metadata, such as the date last modified, and standard HTTP metadata, such as Content-Type. We can also specify custom metadata at the time that the object is stored.

Amazon Simple Storage Services (Amazon S3)

Working of Amazon S3



When we create a bucket, we should give a bucket name and choose the AWS Region where the bucket will reside. After we create a bucket, we cannot change the name of the bucket or its Region. It's best practice to select a region that's geographically closest to you.

Objects that reside in a bucket within a specific region remain in that region unless you transfer the files somewhere else.

S3 Durability and Redundancy

In AWS S3, "durability" refers to the **high level of data reliability and resilience**, meaning that the service is designed to store your data with an **extremely low chance of loss**, typically **advertised as 99.99999999% (11 nines) data durability**, ensuring your data remains accessible even in case of hardware failures or regional outages due to its redundant storage across multiple Availability Zones (AZs).

How Durability is Achieved:

1) Multiple Copies Across Availability Zones: S3 automatically replicates your data across multiple **Availability Zones (AZs)** within a region. Each AZ is a distinct data center with its own power, networking, and cooling, designed to be independent of failures in other AZs.

- S3 stores **multiple copies** of your data across at least **three geographically separated Availability Zones** within the same region. If one AZ experiences failure, the data remains available from other AZs, ensuring continuity.

2) Automatic Data Integrity Checks: S3 continuously performs **integrity checks** on the data stored in its service, detecting any issues that might occur due to hardware failure, bit rot, or other potential data corruption. S3 automatically repairs any detected issues using the redundant copies stored in different AZs.

3) Self-Healing System: If an issue is detected in the data, S3 automatically repairs the data by retrieving the correct version from a healthy copy. This is part of its design to ensure the data remains consistent and intact.

S3 Durability and Redundancy

Redundancy in AWS S3 refers to the practice of **storing multiple copies of data across different physical locations** to ensure that your data is highly available and protected from hardware failures, data corruption, or other types of data loss. Redundancy is an important part of AWS S3's design to provide both **high durability** and **availability**.

S3 automatically replicates data across at least three Availability Zones (AZs) within a region. By storing multiple copies of your data across multiple AZs, S3 ensures that even if one AZ experiences a failure (e.g., network issues or hardware problems), your data remains intact and accessible from the other AZs.

Storage Classes:

1. **S3 Standard:** This storage class provides **high redundancy** by storing multiple copies of your data in **three or more Availability Zones** in a region. It's designed for frequently accessed data and ensures that your data is both highly available and durable.
2. **S3 Intelligent-Tiering:** This storage class automatically moves data between two access tiers (frequent and infrequent access) while ensuring redundancy across AZs, so your data remains highly available and protected.
3. **S3 Standard-IA (Infrequent Access):**
Designed for data that is accessed less frequently but requires immediate access when needed. Offers lower storage costs compared to the S3 Standard class, with a slightly higher retrieval fee. Suitable for backup and restore scenarios, long-term storage, and disaster recovery.

S3 Durability and Redundancy

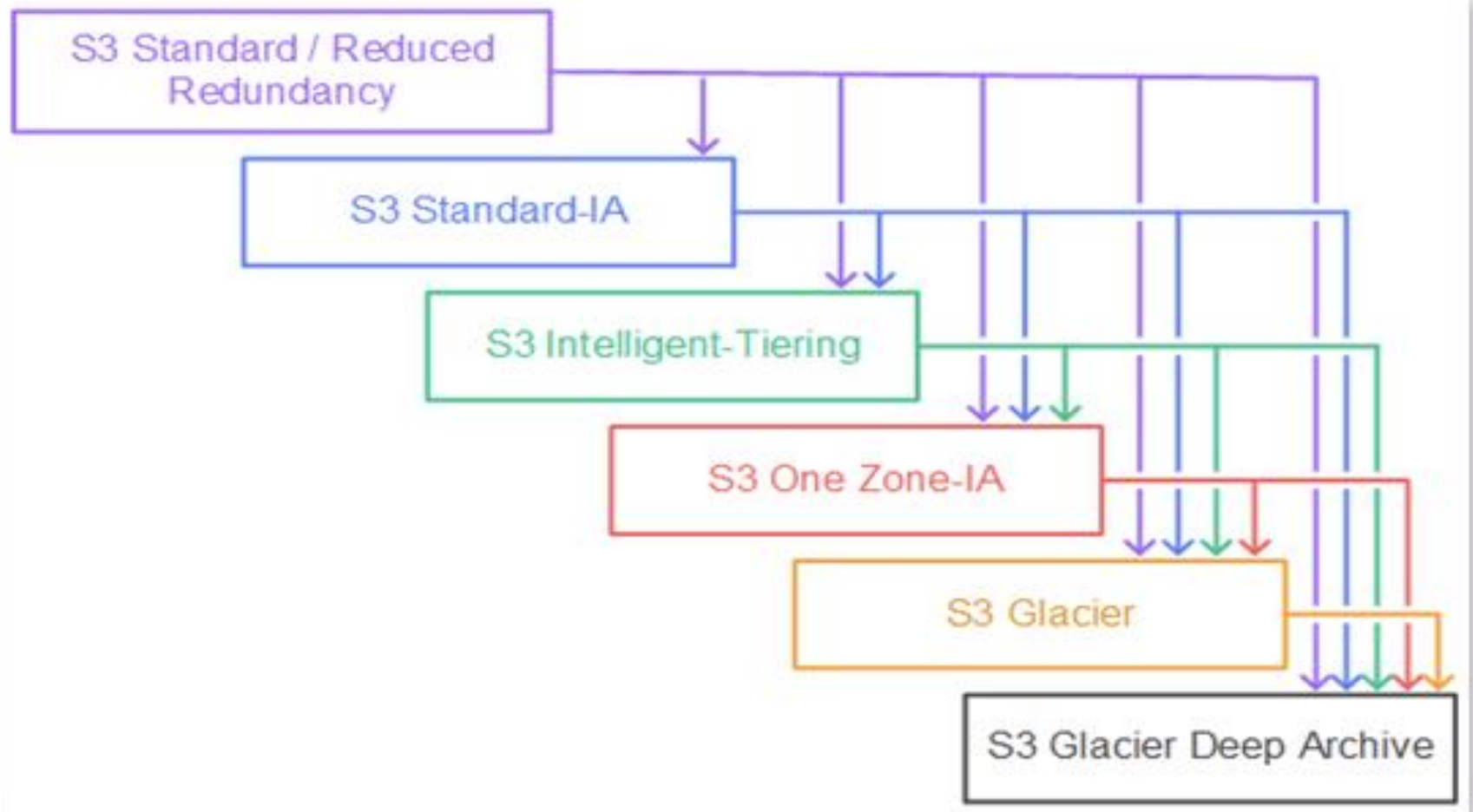
4. Amazon S3 One Zone - Infrequent Access: allows you to **replicate objects within the same region but into different buckets**, providing another layer of redundancy within a region. This is beneficial for keeping a backup copy of your data in a different bucket within the same region for higher availability

5. Amazon Glacier Flexible Retrieval: Amazon S3 Glacier Flexible Retrieval is a perfect solution for long storage and data archiving that don't require instant access. The service allows storing large or small volumes of data at a low cost. At the same time, the retrieval process may take several hours

6. Amazon Glacier Instant Retrieval: Amazon introduced the new archive storage class Amazon S3 Glacier Instant Retrieval. This class was designed to provide greater flexibility in storage management and efficiently reduce costs, and it became the most cost-effective and immediately accessible tool for long-term storage.

7. Amazon Glacier Deep : Even though these storage classes are designed for archival storage with lower access speeds, they still maintain redundancy and durability by replicating data across multiple AZs. **Glacier Deep Archive** provides even more cost-effective storage for long-term retention.

Storage Classes



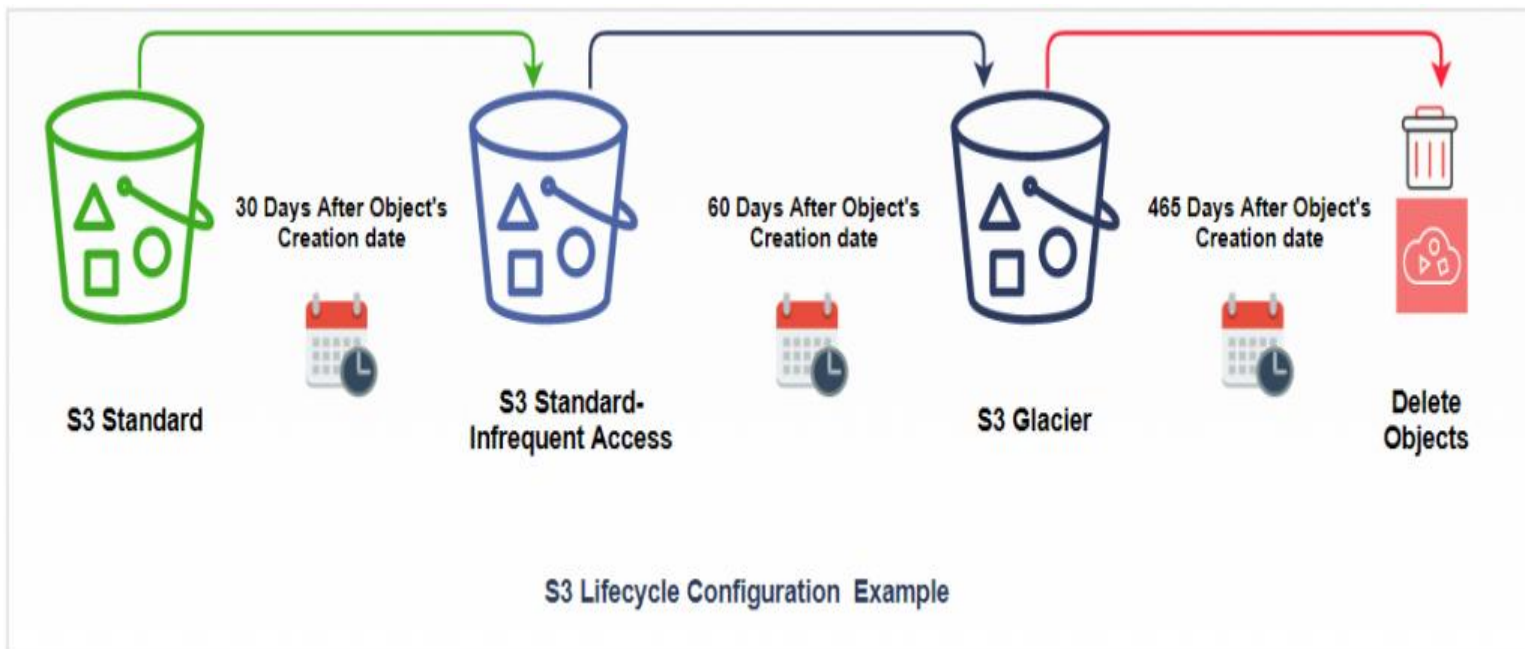
S3 Durability and Redundancy

8. Amazon S3 RRS: **Amazon S3 Reduced Redundancy Storage (RRS)** allows for reducing storage costs for uncritical data. This can be achieved by reducing redundancy and decreasing the number of replicas. The main difference between RRS and S3 Standard is reliability at 99,99%. This means that, on average, if you store 10,000 objects within a year, you are at risk of losing only one of them.

Versioning in S3 enables the ability to keep multiple versions of an object. This adds an extra layer of redundancy because even if an **object is accidentally deleted or overwritten, previous versions can still be recovered.**

S3 Lifecycle Management Configurations

S3 Lifecycle helps you store objects cost effectively throughout their lifecycle by transitioning them to lower-cost storage classes, or, deleting expired objects on your behalf. To manage the lifecycle of your objects, create an S3 Lifecycle configuration for your bucket. An S3 Lifecycle configuration is a set of rules that define actions that Amazon S3 applies to a group of objects.



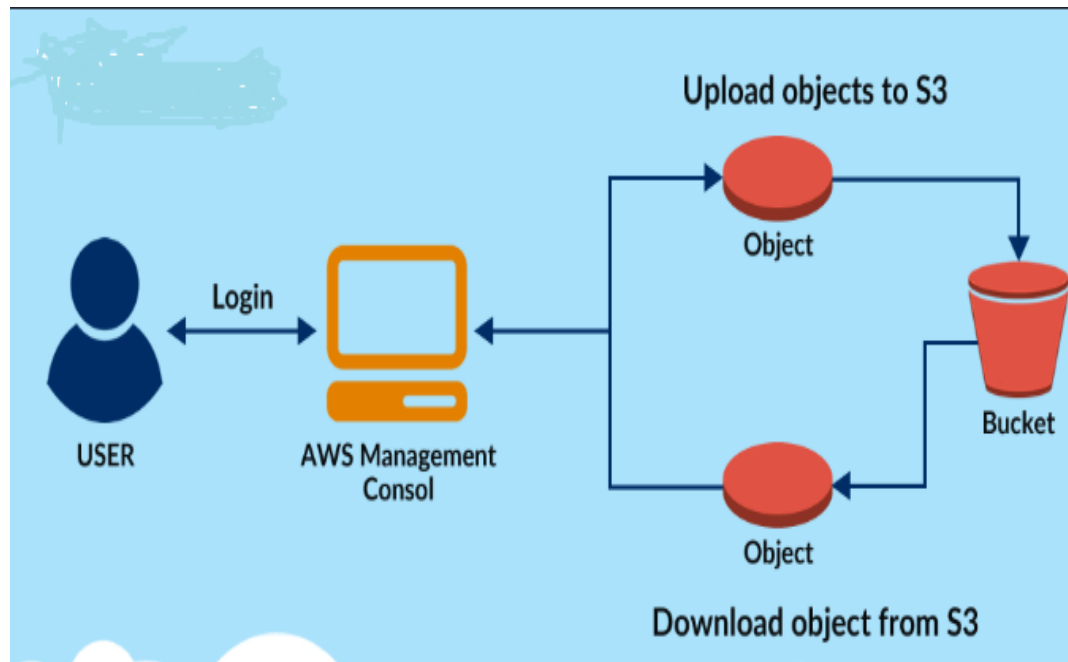
S3 Uploading Downloading

Uploading Objects:

Objects are uploaded to a specific bucket. You can specify the key, set metadata, choose the storage class, and configure encryption at upload time.

Retrieving Objects:

Objects can be downloaded or accessed directly if they are public or if you have the necessary permissions.



S3 Permissions

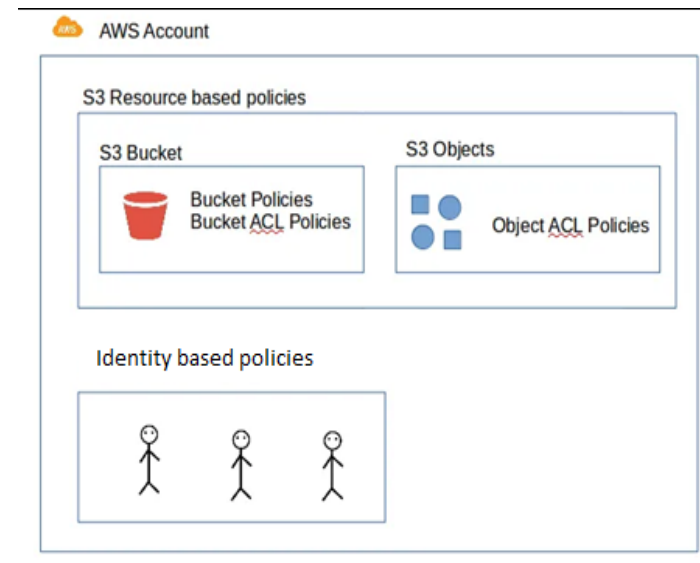
Amazon S3 buckets are **private by default**. This means that only the bucket owner can access the bucket, unless they grant access to others.

Amazon S3 permissions control access to the objects and buckets within S3. These **permissions are essential for managing who can perform actions like reading, writing, and deleting data in your S3 storage.**

We can **restrict access to S3 resources** by having **two types of Access Policies** .

1. Resource based Policies — [Bucket Policies / Bucket ACL Policies and Object ACL Policies]

2. Identity based Policies - IAM Policy



S3 Permissions

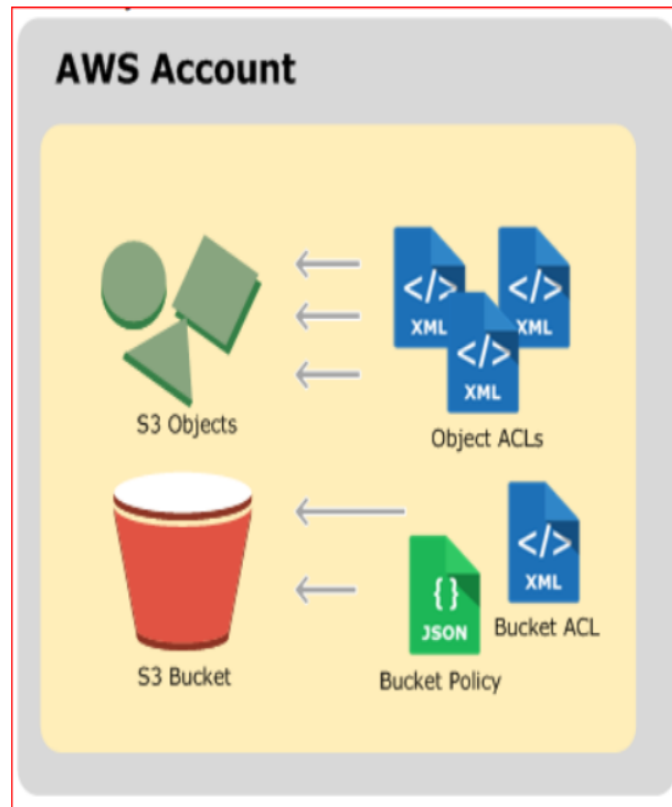
Resource-based policies:

Resource-based policies are access policies **we attach to our resources (buckets and objects)**. These policies define permissions that control who can access the resource and what actions they can perform on it.

Bucket and object permissions are independent of each other.

An object does not inherit the permissions from its bucket.

For example, if you create a bucket and grant write access to a user, you can't access that user's objects unless the user explicitly grants you access.



Bucket Policy

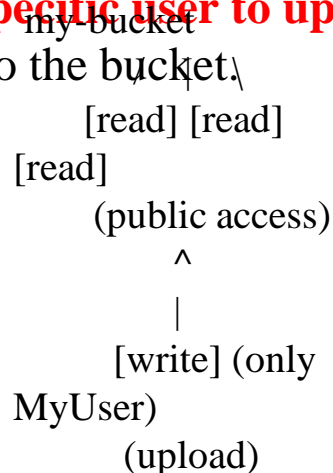
A **Bucket Policy** in AWS S3 is used to control access to an S3 bucket. It's a **JSON document** that defines **what actions are allowed or denied on the bucket and its objects**, and **who can perform those actions**.

You can use bucket policies to grant permissions based on specific criteria such as IP addresses, AWS accounts, or the type of actions.

Example :

Let's say you have a bucket called **my-bucket**, and you want to create a bucket policy that:

- 1) Allows **everyone to read** (download) the objects in the bucket.
- 2) Allows **only a specific user to upload (write)** objects to the bucket.



```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": "*",
      "Action": "s3:GetObject",
      "Resource": "arn:aws:s3:::my-
bucket/*"
    },
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:us
er/MyUser"
      },
      "Action": "s3:PutObject",
      "Resource": "arn:aws:s3:::my-
bucket/*"
    }
  ]
}
```

Bucket ACLs and Object ACLs are often managed using XML format instead of JSON

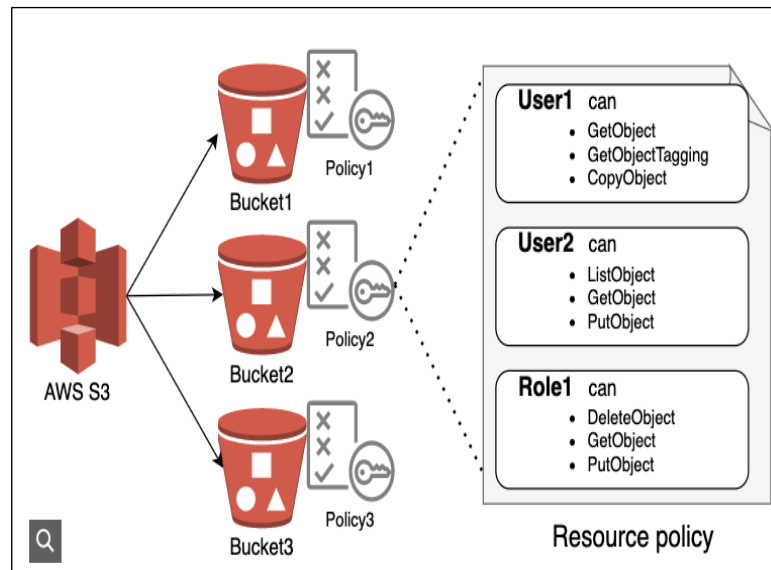
Bucket ACL Policies

A "Bucket ACL Policy" in Amazon S3 refers to an **Access Control List (ACL)** that is attached to an S3 bucket, allowing you to define **which users or groups can access the bucket** and the **level of access they have (read, write, full control)**, essentially managing permissions at the bucket level;

However, AWS generally **recommends using bucket policies instead of ACLs** as they offer more granular control and flexibility for managing access to S3 resources.

Object ACL Policies

Object ACLs control access to individual objects within the bucket. This is useful when you want to set different permissions for specific objects in the same bucket.



Identity based Policies - IAM (Identity and Access Management) Policy

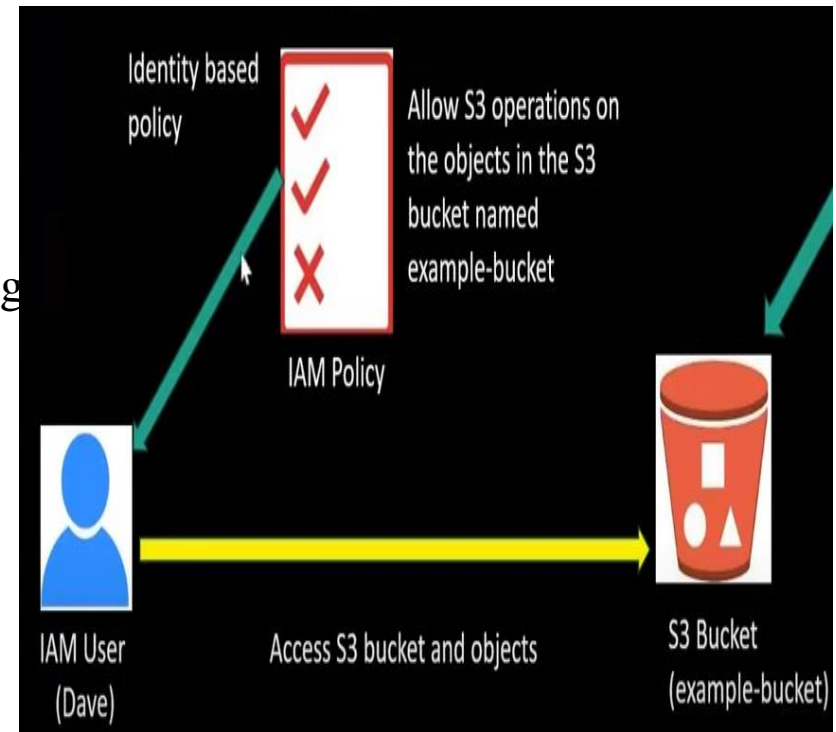
IAM is an AWS service for managing both **authentication** and **authorization** in determining who can access which resources in your AWS account. At the core of IAM's authorization system is an IAM policy.

Identity-based policies are policies that define **what actions an IAM identity (user, group, or role)** are allowed to take against specific resources. They are written in JSON format

Applying the Policy

To apply these IAM policies:

1. Go to the IAM console in AWS.
2. Create a new policy or edit an existing one.
3. Attach the policy to the appropriate IAM user, g



Difference between Backup vs Snapshot

Backup: It can be stored in any location, any server (or) any drive. It's a long-term process to complete the backup based on size. It is utilized to fully store the backup. Backups have different starting points and different endpoints.

Snapshot: It can be stored in a location, where the original data is present. It's a short-term process to complete the snapshot because it takes only the changed data. It is utilized for versioning. Snapshot has a recent picture of your server.

Benefits of AWS Snapshots

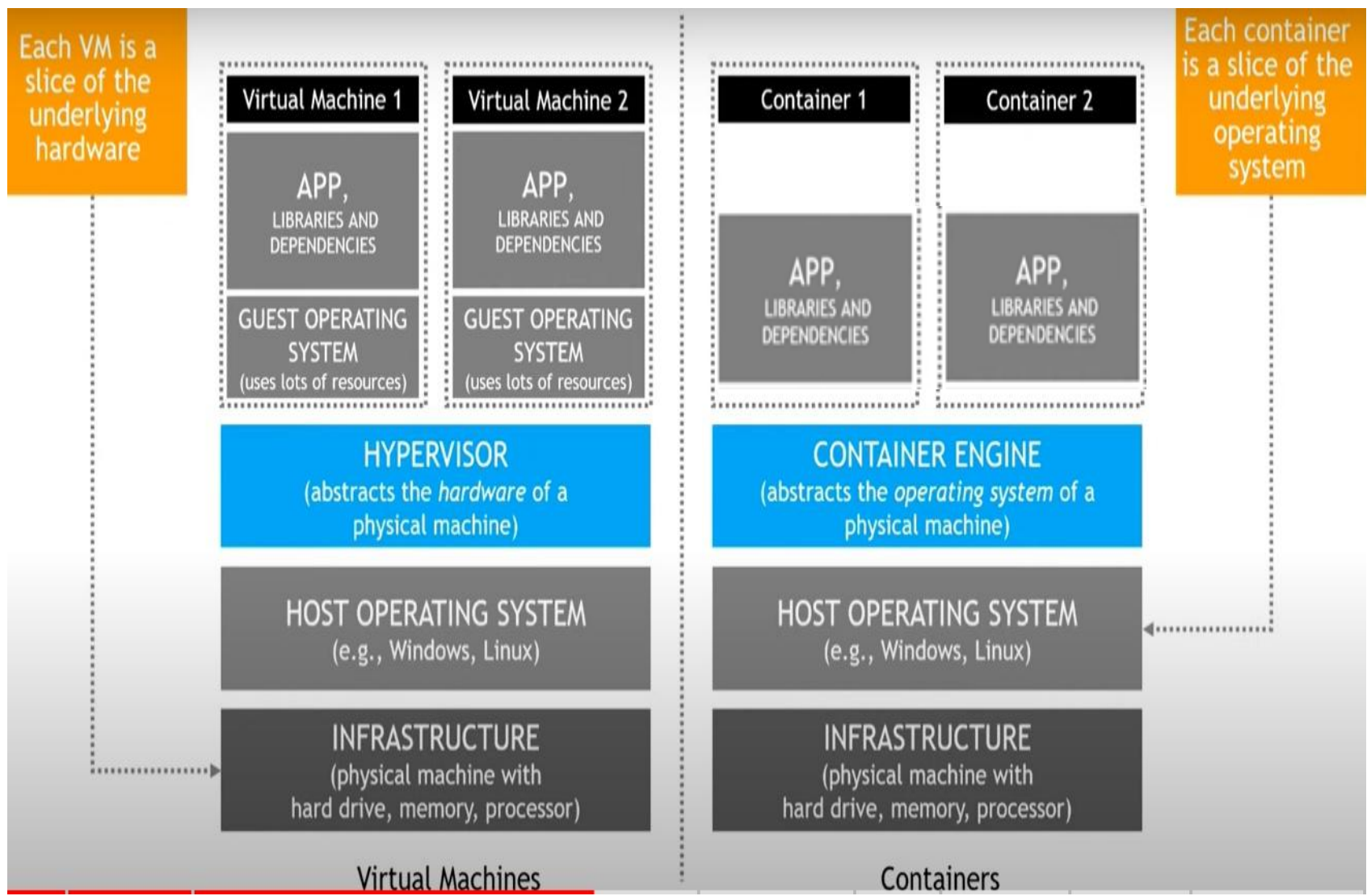
Some benefits of AWS Snapshots are listed below:

- 1. Automation:** The data stored in the AWS EBS will be dynamical means the data will continuously added to the volume so instead of performing manually you can automate the whole processes like you can set the timer in the snapshots configuration while creating AWS snapshots then snapshots will be automatically with in the certain intervals of time.
- 2. Cost-effective:** The design of AWS Snapshots will make them effective for the cost wise. Because AWS snapshots will follow the incremental type which the data added new to the existing storage will only been taken as an snapshots not the entire data. The pricing will only depends on the amount of data have been backed up.
- 3. Secure:** The data which is been backed up using AWS snapshots will have an encryption and also you can re-encrypt the snapshots depending on the requirement.
- 4. High Availability:** The snapshots which we have taken will directly stored to amazon S3 for long term purpose. S3 is designed for 99.99999999% (11 nines) durability, ensuring higher availability of your EBS Snapshots.

Container services

To understand what containers are, we need to understand what containers are replacing, It is replacing the **Virtual Machine**.

Virtual Machines (VM) vs Containers



Virtual Machines (VM)

Each **virtual machine** runs a unique guest **operating** system.

VMs with different operating systems can **run on the same physical server**—a UNIX VM can sit alongside a Linux VM, and so on.

Each VM has its **own binaries, libraries, and applications** that it services

The **hypervisor software**, which **simulates the physical infrastructure and divides the resources into numerous virtual machines**, runs on top of virtual machines. The host computer or virtual machine monitor are other names for the hypervisor.

VMs are usually **oversized, and many gigabytes in size** since each VM container has its guest operating system, kernel, binaries, libraries, and applications.

Containers

Containers sit on top of a physical server and its host OS—for example, Linux or Windows.

Each container **shares the host OS kernel** and, usually, **the binaries and libraries, too**.

Unlike virtual machines, container applications **virtualize the OS rather than the hardware**.

Only the application and its dependencies are packed in container technology because they share most of their requirements. Shared components **are read-only**.

They are **merely megabytes in size and far lighter than virtual machines**.

Containers also **reduce management overhead**. Because they share a common operating system, only a single operating system needs care and feeding for bug

AWS Container Services

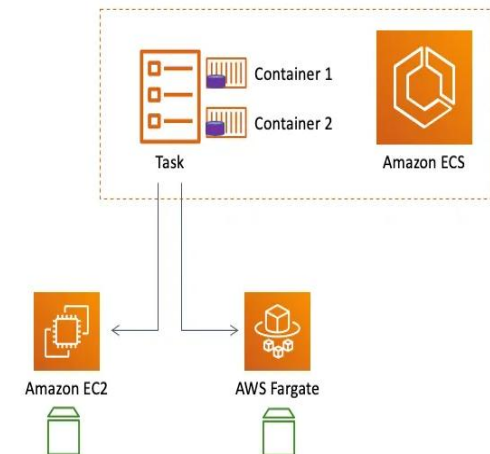
AWS offers several Container services like **Elastic Container Registry (ECR), Elastic Container Service (ECS), Kubernetes (EKS), and AWS Fargate.**

AWS ECS (Elastic Container Service)

One way of **using a Container engine** is via the AWS Elastic Container Service.. ECS has **inbuilt security features to protect your Containers**. AWS users can also automate the launch of several Containers using the Continuous Integration and Delivery CI/CD model.

ECS is an ideal choice for deploying **microservices-based applications, batch processing jobs, and machine learning models.**

ECS is also a **cheap way, yielding savings of up to 50% with the pay-as-you-go model.** AWS ECS also integrates seamlessly with other AWS services.



The EC2 launch type allows you to run your containers on **EC2 instances** that you manage, while the **Fargate launch type** allows you to run your containers without the need to manage any underlying infrastructure.

AWS EKS (Elastic Kubernetes Service)

Kubernetes is a portable, extensible, open source platform for **managing containerized workloads** and services, that facilitates both declarative configuration and automation.

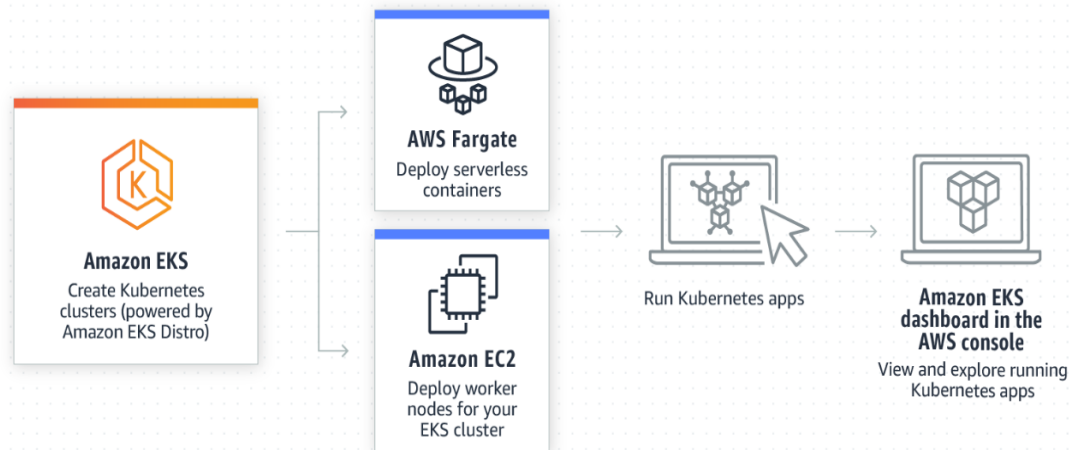
The AWS Kubernetes service is a **reliable way of running containerized apps** over Kubernetes.

Kubernetes is an open-source system used for running **Containerized workloads**.

EKS has an auto-scaling feature, which helps in right-sizing resources. In addition, EKS has **more security layers** comprised of patches and automatic security updates for the Kubernetes environment.

Like ECS, EKS supports both on premise and hybrid cloud environments.

The EKS console lets you view worker nodes and serverless Containers for your EKS workloads. The Elastic Kubernetes Service is ideal for Machine Learning workflows because of the Kubeflow feature



AWS ECR(Elastic Container Registry)

AWS users can manage the **storage and encryption of Container images using the ECR service.**

With the Elastic Container Registry Service, you can make **images fast by compressing them and making them available on-demand.**

ECR lets you import existing Containerized apps without installing any dependencies.

ECR lets you **securely share Container images** via the HTTPS connection. Automatic encryption adds an extra layer to stored resources. Users in highly regulated environments can meet compliance through ECR storage.



AWS Fargate

Use AWS Fargate to deploy **serverless Containers**.

AWS Fargate **integrates with ECS and EKS** to remove the need for server management. Instead of managing servers, you only manage the applications. Based on the architecture, AWS Fargate isolates the most secure applications.

AWS Fargate integrates with CloudWatch to pull critical application metrics. In the traditional model, AWS users had to spend time deploying a compute engine, paying for it, and rightsizing EC2. AWS Fargate removes all these inefficiencies and leads to a faster time to market.

Use AWS Fargate to build web applications, deploy APIs, and run microservices at optimized costs.

