

**Solutions Architect Associate exam**

<https://www.demystify.cloud/>

Sourav Banerjee

**AWS Certified Solutions Architect  
Associate**

# Index

<b>Compute</b> Amazon EC2 ..... 3 AWS Batch ..... 5 AWS Elastic Beanstalk ..... 7 AWS Lambda ..... 10 AWS SAR ..... 12 AWS Fargate ..... 13	AWS CloudTrail ..... 65 Amazon CloudWatch ..... 67 AWS Config ..... 69 AWS License Manager ..... 71 AWS Organizations ..... 72 AWS Systems Manager ..... 74
<b>Containers</b> Amazon EKS ..... 14 Amazon ECS ..... 16 Amazon ECR ..... 18	<b>Developer Tools</b> AWS CodeBuild ..... 76 AWS CodeCommit ..... 77 AWS CodeDeploy ..... 78 AWS X-Ray ..... 79
<b>Storage</b> Amazon S3 ..... 19 AWS Backup ..... 21 Amazon EBS ..... 22 Amazon EFS ..... 24 Amazon FSx for Windows File Server ..... 26 Amazon FSx for Lustre ..... 27 Amazon S3 Glacier ..... 28 AWS Snowball ..... 29 AWS Storage Gateway ..... 30	<b>Migration &amp; Transfer</b> AWS DMS ..... 80
<b>Database</b> Amazon Aurora ..... 33 Amazon DocumentDB ..... 35 Amazon DynamoDB ..... 36 Amazon ElastiCache ..... 39 Amazon Keyspaces ..... 41 Amazon Neptune ..... 42 Amazon RDS ..... 43 Amazon Redshift ..... 46	<b>Networking &amp; Content Delivery</b> Amazon API Gateway ..... 81 AWS Cloud Map ..... 83 Amazon CloudFront ..... 85 AWS PrivateLink ..... 87 AWS Transit Gateway ..... 88 AWS Direct Connect ..... 90 AWS ELB ..... 92 Amazon Route 53 ..... 94 Amazon VPC ..... 96
<b>Security, Identity, &amp; Compliance</b> AWS IAM ..... 48 Amazon Cognito ..... 50 AWS Directory Service ..... 52 Amazon RAM ..... 54 AWS Secrets Managers ..... 55 AWS Security Hub ..... 57	<b>Front End Web &amp; Mobile</b> AWS AppSync ..... 100
<b>Cryptography &amp; PKI</b> AWS KMS ..... 58 AWS Certificate Manager ... 60	<b>Application Integration</b> Amazon EventBridge ..... 102 Amazon SNS ..... 104 Amazon SQS ..... 106 AWS Step Functions ..... 107 Amazon SWF ..... 109
<b>Management &amp; Governance</b> AWS Auto Scaling ..... 62 AWS CloudFormation ..... 63	<b>Billing &amp; Cost Management</b> AWS Cost Explorer ..... 111 AWS Budgets ..... 113 AWS Cost & Usage Reports ..... 115 Reserved Instance Reporting ..... 116
	<b>Monitoring</b> AWS Personal Health Dashboard ..... 117
	<b>AWS Management Console</b> AWS Management Console ..... 118

# AWS EC2

## What is AWS EC2?

- EC2 stands for Elastic Compute Cloud.
- Amazon EC2 is the virtual machine in the Cloud Environment.
- Amazon EC2 provides scalable capacity. Instances can scale up and down automatically based on the traffic.
- You do not have to invest in the hardware.
- You can launch as many servers as you want and you will have complete control over the servers and can manage security, networking, and storage.

## Instance Type:

- Instance type is providing a range of instance types for various use cases.
- The instance is the processor and memory of your EC2 instance.

## EBS Volume:

- EBS Stands for Elastic Block Storage.
- It is the block-level storage that is assigned to your single EC2 Instance.
- It persists independently from running EC2.
  - Types of EBS Storage
    - General Purpose (SSD)
    - Provisioned IOPS (SSD)
    - Throughput Optimized Hard Disk Drive
    - Cold Hard Disk Drive
    - Magnetic

**Instance Store:** Instance store is the ephemeral block-level storage for the EC2 instance.

- Instance stores can be used for faster processing and temporary storage of the application.

## AMI: AMI Stands for Amazon Machine Image.

- AMI decides the OS, installs dependencies, libraries, data of your EC2 instances.
- Multiple instances with the same configuration can be launched using a single AMI.

**Security Group:** A Security group acts as a virtual firewall for your EC2 Instances.

- It decides the type of port and kind of traffic to allow.
- Security groups are active at the instance level whereas Network ACLs are active at the subnet level.
- Security Groups can only allow but can't deny the rules.
- The Security group is considered stateful.
- By default, in the outbound rule all traffic is allowed and needs to define the inbound rules.

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

- Amazon EC2 instances use two keys, one is the public key which is attached to your EC2 instance.
- Another is the private key which is with you. You can get access to the EC2 instance only if these keys get matched.
- Keep the private key in a secure place.

**Tags:** Tag is a key-value name you assign to your AWS Resources.

- Tags are the identifier of the resource.
- Resources can be organized well using the tags.

**Pricing:**

- You will get different pricing options such as On-Demand, Savings Plan, Reserved Instances, and Spot Instances.

# AWS Batch

## What is AWS Batch?

AWS Batch allows developers, scientists, and engineers to run thousands of computing jobs in the AWS platform. It is a managed service that dynamically maintains the optimal compute resources like CPU, Memory based on the volume of submitted jobs. The User just has to focus on the applications (like shell scripts, Linux codes or java programs).

It executes workloads on **EC2 (including Spot instances)** and **AWS Fargate**.

## Components:

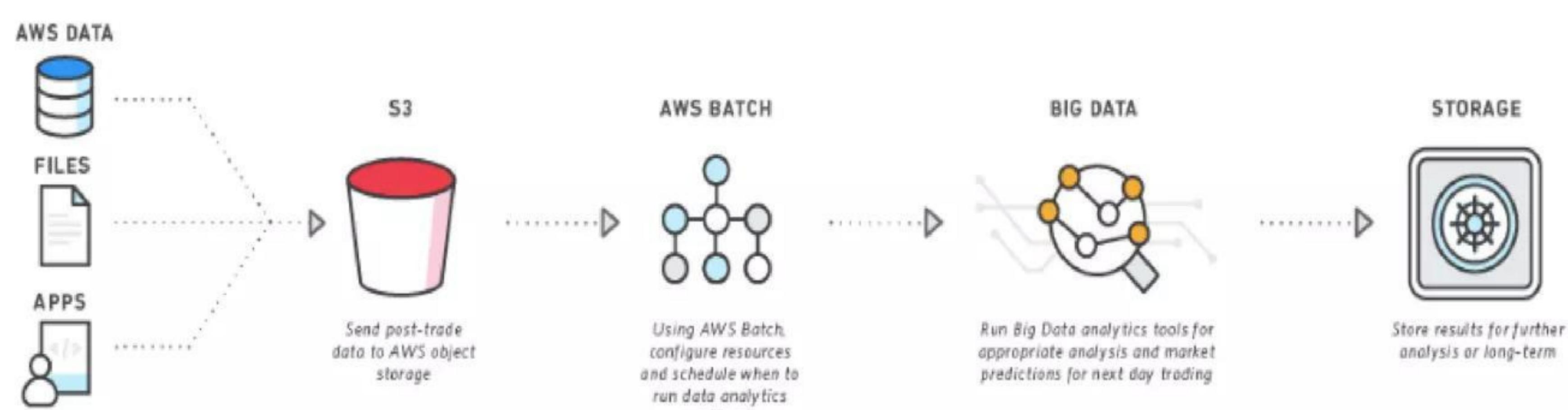
- **Jobs** - are the fundamental applications running on Amazon EC2 machines in containerised form.
- **Job Definitions** – define how the job is meant to be run. Like the associated IAM role, vCPU requirement, and container properties.
- **Job Queues** – Jobs reside in the Job queue where they wait till they are scheduled.
- **Compute Environments** – Each job queue is linked to a computing environment which in itself contains the EC2 instance to run containerized applications.
- There are two types of environments: **Managed** where the user gives min and max vCPU, EC2 instance type and AWS runs it on your behalf and **Unmanaged** where you have your own ECS agent.
- **Scheduler** – maintains the execution of jobs submitted to the queue as time and dependencies.

## Best Practices:

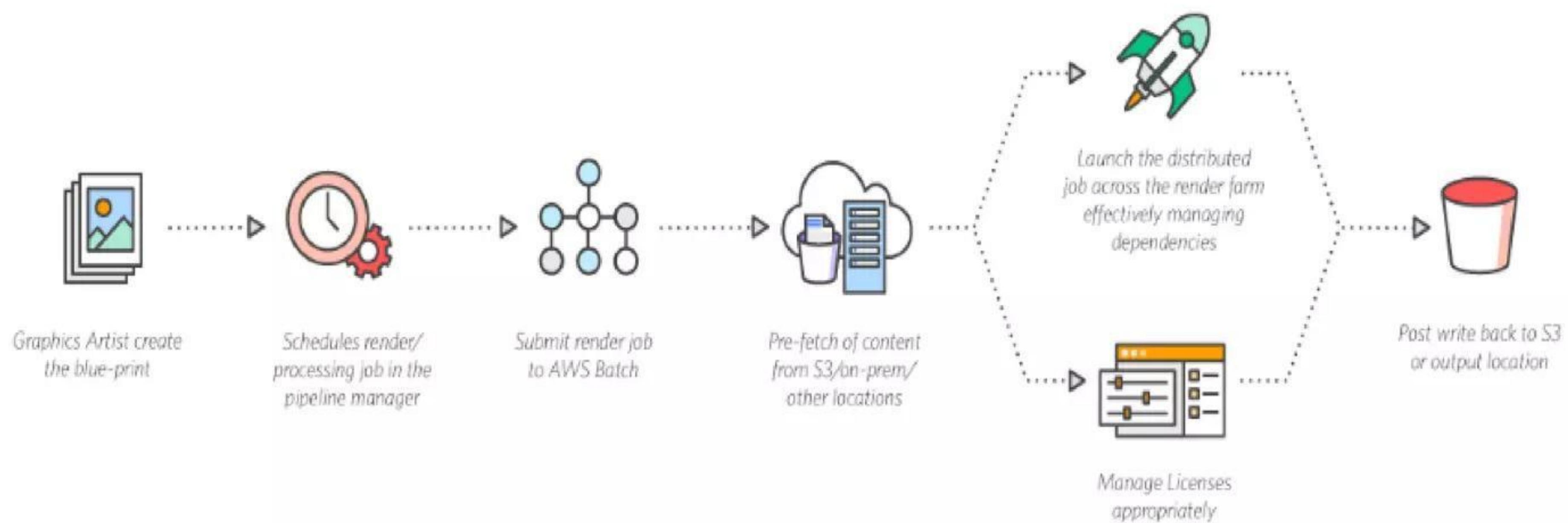
- Use Fargate if you want to run the application without getting into EC2 infrastructure details. Let the AWS batch manage it.
- Use EC2 if your work scale is very large and you want to get into machine specifications like memory, CPU, GPU.
- Jobs running on Fargate are faster on startup as there is no time lag in scale-out operation, unlike EC2 where launching new instances may take time.

## Use Cases:

- **Stock markets and Trading** – The trading business involves daily processing of large scale data and loading them into a Data warehouse for analytics. So that your predictions and decisions are quick enough to make a business grow on a regular basis.



- Media houses and the Entertainment industry – Here a large amount of data in the forms of audio, video and photos are being processed daily to cater to their customers. These application workloads can be moved to containers on AWS Batch.



### Pricing:

- There is no charge for AWS Batch rather you pay for the resources like EC2 and Fargate you use.

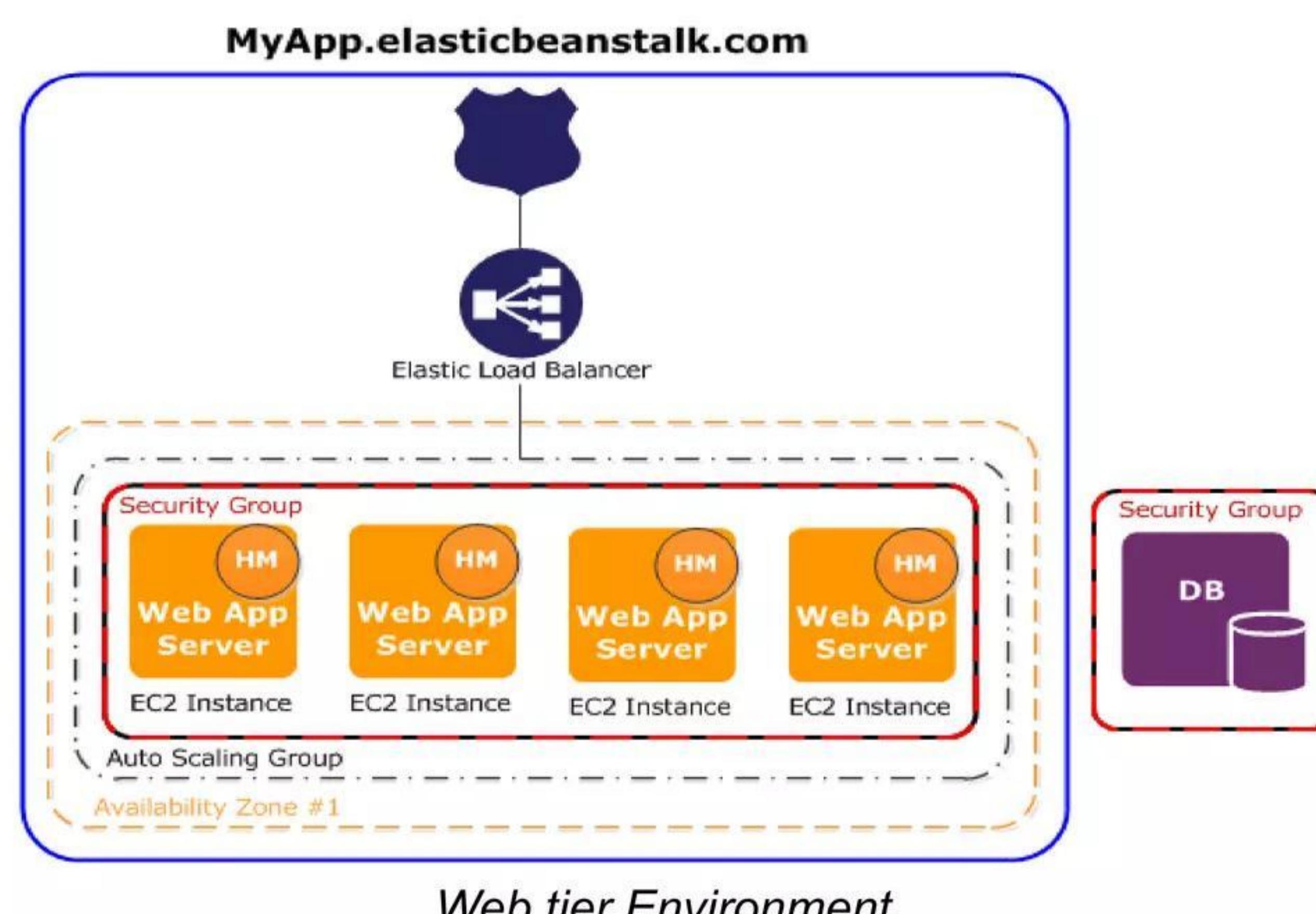
# AWS Beanstalk

## What is Amazon Elastic Beanstalk?

- Beanstalk is a compute service for deploying and scaling applications developed in many popular languages.
- Developers can focus on writing code and don't need to worry about the underlying infrastructure required to run the application.
- AWS Elastic Beanstalk is the best way to deploy your application in the fastest and simplest way.
- It provides the user interface/dashboard to monitor your application.
- It gives you the flexibility to choose AWS resources such as Amazon EC2 Instance along with the pricing options which suit your application needs.

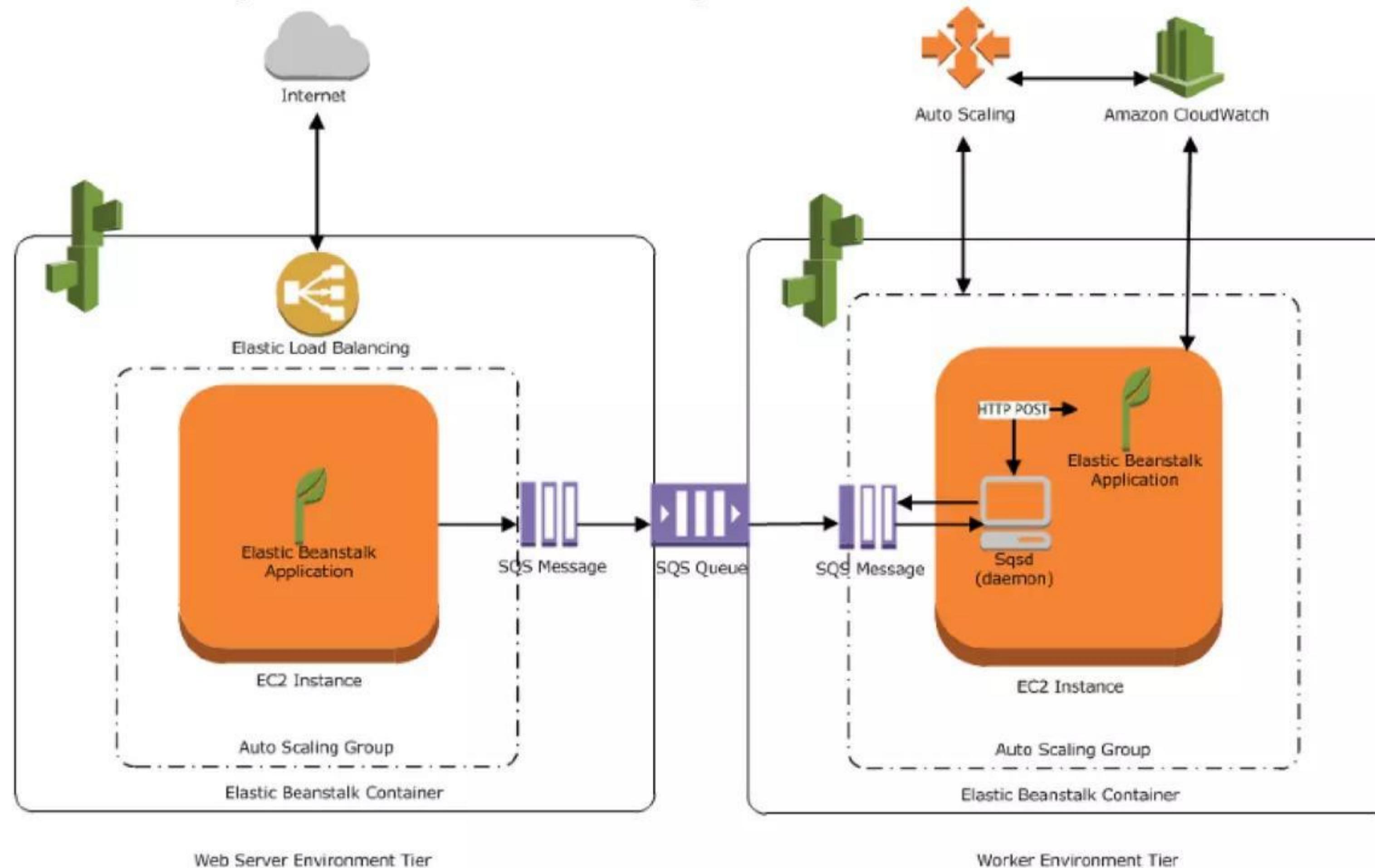
AWS Elastic Beanstalk supports two types of Environment:

- **Web Tier Environment**
  - This application hosted on the Web Server Environment handles the HTTP and HTTPS requests from the users.
  - Beanstalk Environment: When an environment is launched, Beanstalk automatically assigns various resources to run the application successfully.
  - Elastic Load Balancer: Request is received from the user via Route53 which forwards the request to ELB. Then ELB distributes the request among various EC2 Instances of the Autoscaling group.
  - Auto Scaling Group: Auto Scaling will automatically add or remove EC2 Instance based on the load in the application.
  - Host Manager: Software components inside every EC2 Instance which is responsible for the following:
    - Log files generation
    - Monitoring
    - Events in Instance



- **Worker Environment**

- A worker is a background process that helps applications for handling heavy resource and time-intensive operations.
- It is responsible for database clean up, report generation that helps to remain up and running.
- In the Worker Environment, Beanstalk installs a Daemon on each EC2 Instance in the Auto Scaling Group.
- Daemon pulls requests from the SQS queue and executes the task based on the message received.
- After execution, SQS will delete the message, and in case of failure, it will retry to send the message.



## Platform Supported

- .Net (on Linux or Windows)
- Docker
- GlassFish
- Go
- Java
- Node.js
- Python
- Ruby
- Tomcat

## Deployment Models:

**All at Once:** Deployment will start taking place in all the instances at the same time. It means all your EC2 Instances will be out of service for a short time. Your application will be completely down for the same duration.

**Rolling** – Deploy the new version in batches; unlike all at once, one group of instances will run the old version of the application. That means there will not be complete downtime just like all at once.

**Rolling with additional batch** - Deploy the new version in batches. But before that, provision an additional group of instances to compensate for the updating one.

**Immutable** – Deploy the new version to a separate group of instances, and the update will be immutable.

**Traffic splitting** – Deploy the new version to a separate group of instances and split the incoming traffic between the older and the new ones.

**Pricing:**

- Amazon will not charge you for AWS Elastic Beanstalk.
- Instead, you will be paying for the resources such as EC2 Instance, ELB and Auto Scaling group where your application is hosted.

# AWS Lambda

## What is AWS Lambda?

- AWS Lambda is a **serverless** compute service through which you can run your code without provisioning any Servers.
- It only runs your code when needed and also scales automatically when the request count increases.
- AWS Lambda follows the Pay per use principle – it means there is no charge when your code is not running.
- Lambda allows you to run your code for any application or backend service with zero administration.
- Lambda can run code in response to the events. Example – update in DynamoDB Table or change in S3 bucket.
- You can even run your code in response to HTTP requests using Amazon API Gateway.

## What is Serverless computing?

- Serverless computing is a method of providing backend services on a pay per use basis.
- Serverless/Cloud vendor allows you to write and deploy code without worrying about the underlying infrastructure.
- Servers are still there, but you are not managing them, and the vendor will charge you based on usage.

## When do you use Lambda?

- When using AWS Lambda, you are only responsible for your code.
- AWS Lambda manages the memory, CPU, Network, and other resources.
- It means you cannot log in to the compute instances or customize the operating system.
- If you want to manage your own compute resources, you can use other compute services such as EC2, Elastic Beanstalk.
- There will be a level of abstraction which means you cannot log in to the server or customize the runtime.

## How does Lambda work?



## **Lambda Functions**

- A function is a block of code in Lambda.
- You upload your application/code in the form of single or multiple functions.
- You can upload a zip file, or you can upload a file from the S3 bucket as well.
- After deploying the Lambda function, Lambda automatically monitors functions on your behalf, reporting metrics through Amazon CloudWatch.

## **Lambda Layers**

- A Lambda layer is a container/archive which contains additional code such as libraries, dependencies, or custom runtimes.
- AWS Lambda allows five layers in a function.
- Layers are immutable.
- A new version will be added if you publish a new layer.
- Layers are by default private but can be shared and made public explicitly.

## **Lambda Event**

- Lambda Event is an entity that invokes the lambda function.
- Lambda supports synchronous invocation of Lambda Functions.
- Lambda supports the following sources as an event:
  - AWS DynamoDB
  - AWS SQS
  - AWS SNS
  - CloudWatch Event
  - API Gateway
  - AWS IoT
  - Kinesis
  - CloudWatch Logs

## **Language Supported in AWS Lambda**

- NodeJS
- Go
- Java
- Python
- Ruby

## **Lambda@Edge**

- It is the feature of Amazon CloudFront which allows you to run your code closer to the location of Users of your application.
- It improves performance and reduces latency.
- Just like lambda, you don't have to manage and provision the infrastructure around the world.
- Lambda@Edge runs your code in response to the event created by the CDN.

## **Pricing:**

- Charges will be calculated based on the number of requests for the function executed in a particular duration.
- Duration will be counted on a per 100-millisecond basis.
- Lambda Free tier usage includes 1 million free requests per month.
- It also comes with 400,000 GB-Seconds of compute time per month.

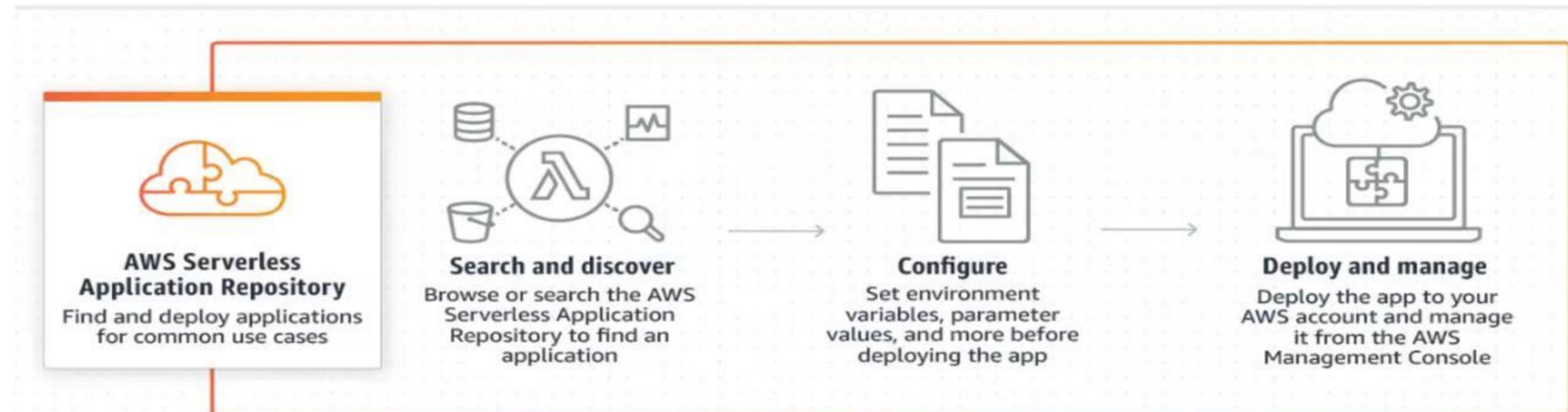
# AWS Serverless Application Repository

## What is AWS Serverless Application Repository?

It is a managed repository for serverless applications. It is used by organizations and independent developers to store and share reusable applications.

### Features:

- AWS Serverless Application Repository has applications for **Alexa Skills, chatbots, IoT, real-time media processing** from many publishers.
- All the applications provided by AWS come under MIT open source license while publicly available applications by other users come under Open Source Initiative (OSI).
- All applications published on Serverless Application Repository are carefully examined by AWS for the correct set of permissions so that the customer knows which application can be accessed.
- AWS **CodePipeline** can be used to link GitHub with Serverless Application Repository.
- Before publishing, describe the application using AWS SAM, package it using CLI, and publish via CLI or SDK or console.
- Applications can be shared within all accounts of AWS Organizations. Users cannot share applications across other Organizations.
- AWS Serverless Application Repository is **integrated with AWS Lambda**. The application can be downloaded and with API Gateway it can trigger the Lambda function. See below diagram -



### Use Case:

- Used for various AWS Alexa skills and integration with IoT devices.
- Used for chatbots that remove inappropriate messages, images from channels.
- Used in Twitter leadership boards.

### Pricing:

- There is no charge for this service itself but you pay for the resources used in the application

# AWS Fargate

## What is AWS Fargate?

AWS Fargate is a serverless compute service that is used for containers by Amazon Elastic Container Service (ECS) and Amazon Elastic Kubernetes Service (EKS).

- It eliminates the tasks required to provision, configure, or scale groups of virtual machines like Amazon EC2 to run containers.
- It packages the application in containers, by just specifying the CPU and memory requirements with IAM policies. Fargate task does not share its underlying kernel, memory resources, CPU resources, or elastic network interface (ENI) with another task.
- It does not support all the task definition parameters that are available in Amazon ECS tasks. Only a few are valid for Fargate tasks with some limitations.
- Kubernetes can be integrated with AWS Fargate by using controllers. These controllers are responsible for scheduling native Kubernetes pods onto Fargate.
- Security groups for pods in EKS can not be used when pods running on Fargate.
- The following storage types are supported for Fargate tasks:
  - Amazon EFS volumes for persistent storage
  - Ephemeral storage for nonpersistent storage

## Benefits:

- Fargate allows users to focus on building and operating the applications rather than focusing on securing, scaling, patching, and managing servers.
- Fargate automatically scales the compute environment that matches the resource requirements for the container.
- Fargate provides built-in integrations with other AWS services like Amazon CloudWatch Container Insights.

## Price details:

- Charges are applied for the amount of vCPU and memory consumed by the containerized applications.
- Fargate's Savings Plans provide savings of up to 50% in exchange for one or three-year long term commitment.
- Additional charges will be applied if containers are used with other AWS services.

# **Amazon Elastic Kubernetes Service(EKS)**

## **What is Amazon Elastic Kubernetes Service(EKS)?**

Amazon Elastic Kubernetes Service (Amazon EKS) is a service that enables users to manage Kubernetes applications in the AWS cloud or on-premises.

Any standard Kubernetes application can be migrated to EKS without altering the code.

The EKS cluster consists of two components:

- Amazon EKS control plane
- Amazon EKS nodes

The Amazon EKS control plane consists of nodes that run the Kubernetes software, such as etcd and the Kubernetes API server. Amazon EKS runs its own Kubernetes control plane without sharing control plane infrastructure across other clusters or AWS accounts.

To ensure high availability, Amazon EKS runs Kubernetes control plane instances across multiple Availability Zones. It automatically replaces unhealthy control plane instances and provides automated upgrades and patches for the new control planes.

The two methods for creating a new Kubernetes cluster with nodes in Amazon EKS:

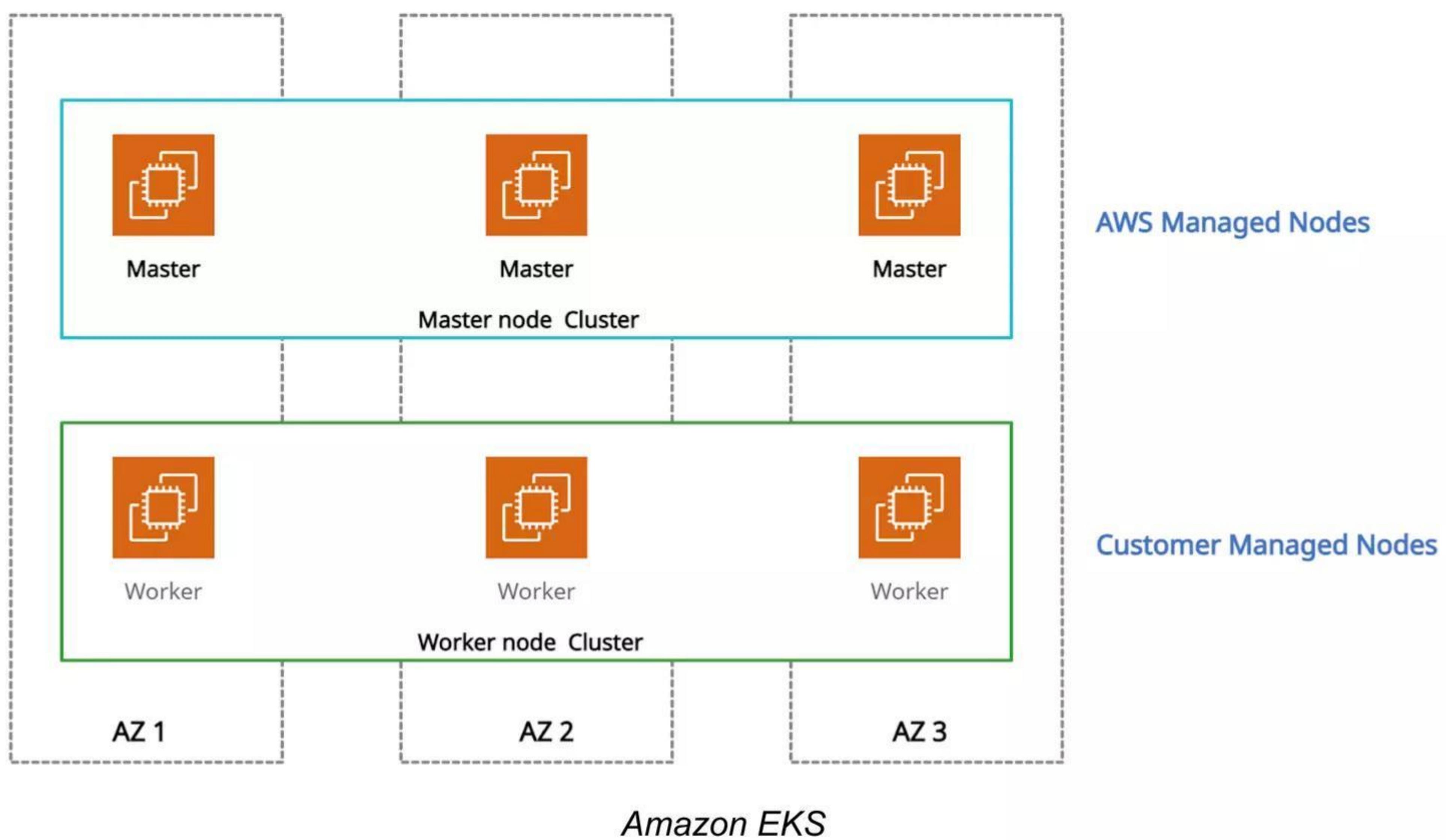
- eksctl – A command-line utility that consists of kubectl for creating and managing Kubernetes clusters on Amazon EKS.
- AWS Management Console and AWS CLI

There are methods that Amazon EKS cluster uses to schedule pods using single or combined node groups:

- Self-managed nodes - consist of one or more Amazon EC2 instances that are deployed in an Amazon EC2 Auto Scaling group
- Amazon EKS Managed node groups - helps to automate the provisioning and lifecycle management of nodes.
- AWS Fargate - run Kubernetes pods on AWS Fargate

Amazon Elastic Kubernetes Service is integrated with many AWS services for unique capabilities:

- Images - Amazon ECR for container images
- Load distribution - AWS ELB (Elastic Load Balancing)
- Authentication - AWS IAM
- Isolation - Amazon VPC



### Use Cases:

- Using Amazon EKS, Kubernetes clusters and applications can be managed across hybrid environments.
- EKS with Kubeflow can model machine learning workflows using the latest EC2 GPU-powered instances.
- Users can execute batch workloads on the EKS cluster using the Kubernetes Jobs API across AWS compute services such as Amazon EC2, Fargate, and Spot Instances.

### Price details:

- \$0.10 per hour is charged for each Amazon EKS cluster created.
- Using EKS with EC2 - Charged for AWS resources (e.g. EC2 instances or EBS volumes).
- Using EKS with AWS Fargate - Charged for CPU and memory resources starting from the time to download the container image until the Amazon EKS pod terminates.

# Amazon Elastic Container Service

## What is Amazon ECS?

Amazon Elastic Container Service (Amazon ECS) is a regional container orchestration service like Docker that allows to execute, stop, and manage containers on a cluster.

A container is a standard unit of software development that combines code, its dependencies, and system libraries so that the application runs smoothly from one environment to another.

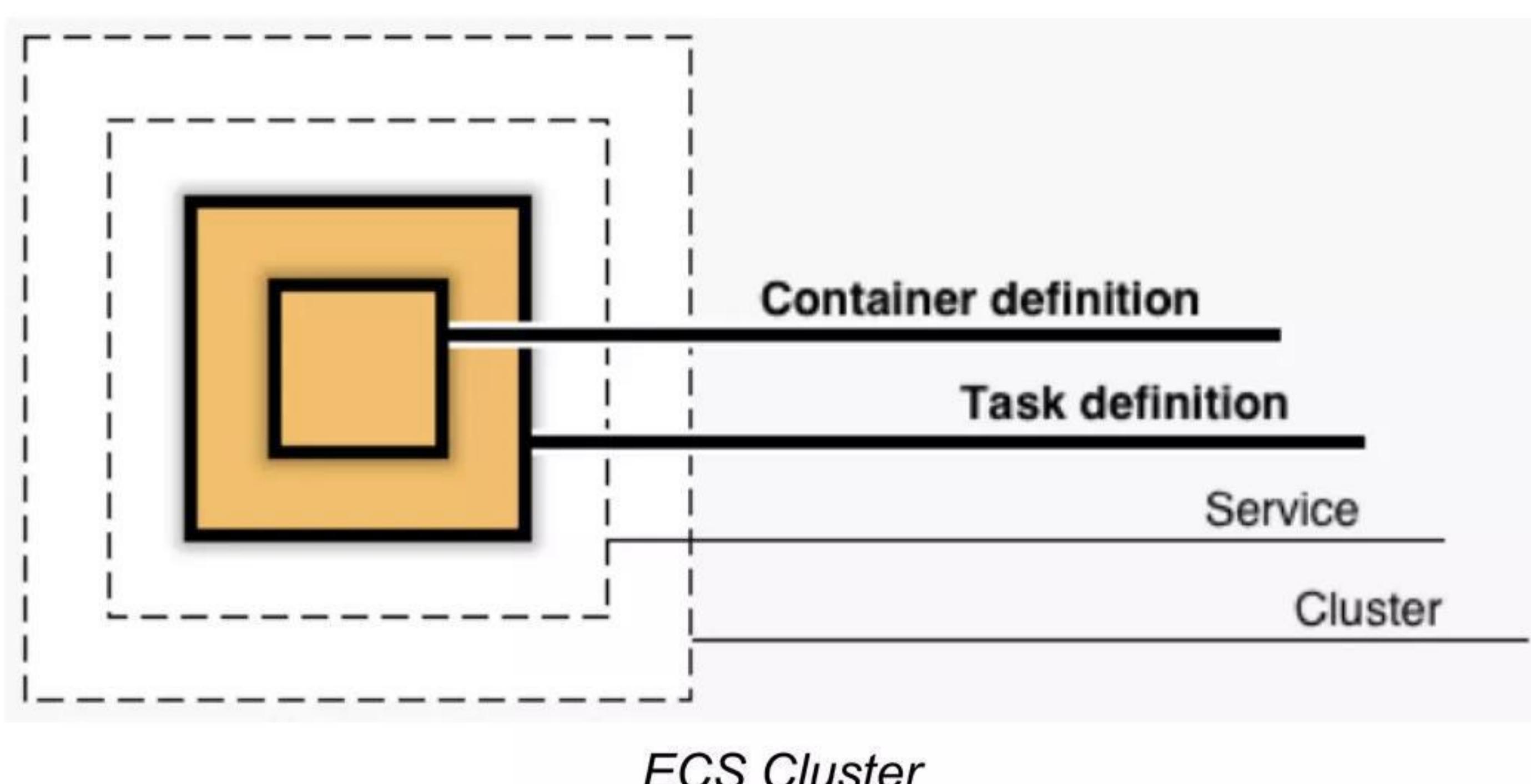
Images are created from a Dockerfile (text format), which specifies all of the components that are included in the container. These images are then stored in a registry from where they can then be downloaded and executed on the cluster.

All the containers are defined in a task definition that runs a single task or tasks within a service. The task definitions (JSON format) defines which container images should run across the clusters. A service is a configuration that helps to run and maintain several tasks simultaneously in a cluster.

ECS cluster is a combination of tasks or services that can be executed on EC2 Instances or AWS Fargate, a serverless compute for containers. When using Amazon ECS for the first time, a default cluster is created.

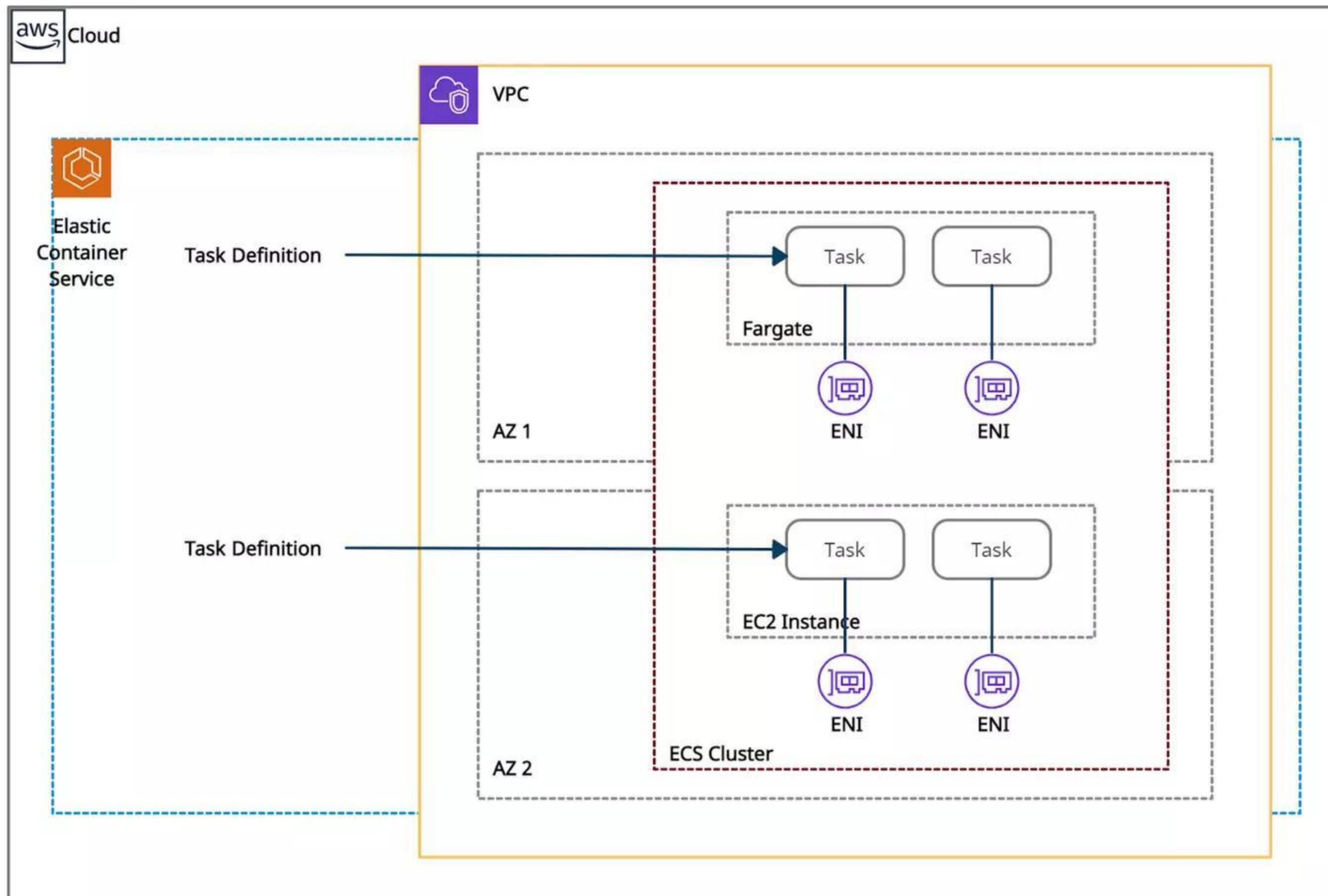
The container agent runs on each instance within an Amazon ECS cluster. It sends data on the resource's current running tasks and resource utilization to Amazon ECS. It starts and stops the tasks whenever it receives a request from Amazon ECS.

A task is the representation of a task definition. The number of tasks to run on your cluster is specified after the task definition is created within Amazon ECS. The task scheduler is responsible for attaching tasks within your cluster based on the task definitions.



Application Load Balancers offer some attractive features:

- It enables containers to use dynamic host port mapping. For that, multiple tasks from the same service are allowed per container instance.
- It supports path-based routing and priority rules due to which multiple services can use the same listener port on a single Application Load Balancer.



*Amazon Elastic Container Service*

Amazon ECS can be integrated with:

- AWS Identity and Access Management
- Amazon EC2 Auto Scaling
- Elastic Load Balancing
- Amazon Elastic Container Registry
- AWS CloudFormation

It decreases time consumption by eliminating user tasks to install, operate, and scale cluster management infrastructure. With API calls, Docker-enabled applications can be launched and stopped.

It powers other services such as Amazon SageMaker, AWS Batch, Amazon Lex. It also integrates with AWS App Mesh, to provide rich observability, controls traffic and security features to the applications.

#### **Use Cases:**

The two main use cases in Amazon ECS are:

- Microservices - They are built by the architectural method that decomposes or decouples complex applications into smaller and independent services.
- Batch Jobs - Docker containers are best suited for batch job workloads. Batch jobs are short-lived packages processed under Docker image. So they can be deployed anywhere, such as in an Amazon ECS task.

#### **Pricing details:**

- Amazon ECS provides two charge models:
  - Fargate Launch Type Model - pay for the amount of vCPU and memory resources.
  - EC2 Launch Type Model - pay for the AWS resources created to store and run the application.

# Amazon Elastic Container Registry

## What is Amazon Elastic Container Registry?

Amazon Elastic Container Registry (ECR) is a managed service that allows users to store, manage, share, and deploy container images and artifacts. It is mainly integrated with Amazon Elastic Container Service (ECS), for simplifying the production workflow.

## Features:

- It stores both the containers which are created, and any container software bought through AWS Marketplace.
- It is integrated with Amazon Elastic Container Service (ECS), Amazon Elastic Kubernetes Service (EKS), and AWS Lambda, and AWS Fargate for easy deployments.
- AWS Identity and Access Management (IAM) enables resource-level control of each repository within ECR.
- It supports public and private container image repositories. It allows sharing container applications privately within the organization or publicly for anyone to download.
- A separate portal called Amazon ECR Public Gallery, helps to access all public repositories hosted on Amazon ECR Public.
- It stores the container images in Amazon S3 because S3 provides 99.99999999% (11 9's) of data durability.
- It allows cross-region and cross-account replication of the data for high availability applications.
- Encryption can be done via HTTPS while transferring container images. Images are also encrypted at rest using Amazon S3 server-side encryption or by using customer keys managed by AWS KMS.
- It is integrated with continuous integration and continuous delivery and also with third-party developer tools.
- Lifecycle policies are used to manage the lifecycle of the images.

## Pricing details:

- Using AWS Free Tier, new customers get 500 MB-month of storage for one year for private repositories and 50 GB-month of free storage for public repositories.
- Without Sign-up, 500 GB of data can be transferred to the internet for free from a public repository each month.
- By signing-up to an AWS account, or authenticating to ECR with an existing AWS Account, 5 TB of data can be transferred to the internet for free from a public repository each month.

# Amazon S3

## What is Amazon S3?

S3 stands for Simple Storage Service.

Amazon S3 is object storage that allows us to store any kind of data in the bucket. It provides availability in multiple AZs, durability, security, and performance at a very low cost.

Any type of customer can use it to store and protect any amount of data for use cases, like static and dynamic websites, data analytics, and backup.

## Basics of S3?

- It is object-based storage.
- Files are stored in Buckets.
- The bucket is a kind of folder.
- Folders can be from 0 to 5 TB.
- S3 bucket names must be unique globally.
- When you upload a file in S3, you will receive an HTTP 200 code if the upload was successful.
- S3 offers Strong consistency for PUTs of new objects, overwrites or delete of current object and List operations.
- By Default, all the Objects in the bucket are private.

## Properties of Amazon S3.

- **Versioning:** This allows you to keep multiple versions of Objects in the same bucket.
- **Static Website Hosting:** S3 can be used to host a Static Website, which does not require any server-side Technology.
- **Encryption:** Encrypt Object at rest with Amazon S3 Managed keys (SSE-S3), or Amazon KMS Managed Keys (SS3-KMS).
- **Objects Lock:** Block Version deletion of the object for a defined period. Object lock can be enabled during the bucket creation only.
- **Transfer Acceleration:** Transfer Acceleration takes advantage of Amazon CloudFront's globally distributed edge locations and enables the fast, easy, and secure transfer of files.

## Permissions & Management.

- **Access Control List: ACLs** used to grant read/write permission to another AWS Account.
- **Bucket Policy:** It uses JSON based access policy advance permission to your S3 Resources.
- **CORS:** CORS stands for Cross-Origin Resource Sharing. It allows cross-origin access to your S3 Resources.

## Charges:

You will be charged based on multiple factors:

- Storage
- Requests
- Storage Management Pricing (Life Cycle Policies)
- Transfer Acceleration
- Data Transfer Pricing

## Miscellaneous Topic

- **Access Point:** By creating Access Point, you can make S3 accessible over the internet.
- **Life Cycle:** By Configuring Lifecycle, you can make a transition of objects to different storage classes.
- **Replication:** This feature will allow you to replicate data between buckets within the same or different region.

## Storage Class/Pricing model of S3

- S3 Standard
- S3 Standard-IA (Infrequent Access)
- S3 Intelligent Tiering (No need to mentioned Life Cycle Policy)
- S3 One Zone-IA (Kept in a Single Zone)
- S3 Glacier (For Archiving Purpose)
- S3 Glacier Deep Archive (For Archiving Purpose)

Storage class	Suitable for	Durability	Availability	Availability Zones	Min. storage days
S3 Standard	accessed data frequently	100%	99.99%	>= 3	None
S3 Standard-IA	accessed data infrequently	100%	99.90%	>= 3	30 days
S3 Intelligent-Tiering	Storage for unknown access patterns	100%	99.90%	>= 3	30 days
S3 One Zone-IA	Non-critical data	100%	99.50%	1	30 days
S3 Glacier	For long term Data Archival. e.g., 3 years – 5 years	100%	99.99%	>= 3	90 days
S3 Glacier Deep Archive	For long term Data Archival. e.g., 3 years – 5 years	100%	99.99%	>= 3	180 days
RRS (Reduced Redundancy Storage)	Frequently accessed for non-critical data but not recommended	99%	99.99%	>= 3	NA

# AWS Backup

## What is AWS Backup?

AWS Backup is a secure service that automates and governs data backup (protection) in the AWS cloud and on-premises.

## Features:

- It offers a backup console, backup APIs, and the AWS Command Line Interface (AWS CLI) to manage backups across the AWS resources like instances and databases.
- It offers backup functionalities based on policies, tags, and resources.
- It provides scheduled backup plans (policies) to automate backup of AWS resources across AWS accounts and regions.
- It offers incremental backup to minimize storage costs. The first backup backs up a full copy of the data and then only the successive incremental backup changes.
- It provides backup retention plans to retain and expire backups automatically. Automated backup retention also helps to minimize storage costs for backup.
- It provides a dashboard in the AWS Backup console to monitor backup and restore activities.
- It offers an enhanced solution by providing separate encryption keys for encrypting multiple AWS resources.
- It provides lifecycle policies configured to transition backups from Amazon EFS to cold storage automatically.
- It is tightly integrated with Amazon EC2 to schedule backup jobs and the storage (EBS) layer. It also simplifies recovery by restoring whole EC2 instances from a single point.
- It supports cross-account backup and restores either manually or automatically within the AWS organizations.
- It allows backups and restores to different regions, especially during any disaster, to reduce downtime and maintain business continuity.
- It integrates with Amazon CloudWatch, AWS CloudTrail, and Amazon SNS to monitor, audit API activities and notifications.

## Use cases:

- It can use AWS Storage Gateway volumes for hybrid storage backup. AWS Storage Gateway volumes are secure and compatible with Amazon EBS, which helps restore volumes to on-premises or the AWS environment.

## Price details:

- AWS charges monthly based on the amount of backup storage used and the amount of backup data restored.

## AWS EBS - Elastic Block Store

### What is AWS EBS?

Amazon Elastic Block Store (AWS EBS) is a persistent block-level storage (volume) service designed to be used with Amazon EC2 instances. **EBS is AZ specific** & automatically replicated within its AZ to protect from component failure, offering high availability and durability.

### Types of EBS:

<b>SSD-backed volumes</b> (Solid State Drive)	<b>Optimized for transactional workloads</b> (small and frequent I/O) - IOPS	
<b>Types SSD</b>	<b>General Purpose SSD- gp2</b> (1 GiB — 16 TiB)  IOPS : 3000 to 20000 Max / Volume	Boot volumes Development /Test Low-latency Apps Virtual Desktops
	<b>Provisioned IOPS SSD (io1)</b> low-latency or high-throughput Consistent IOPS (16,000+ IOPS ) Transactional workloads	MongoDB / NoSQL MySQL / RDS Latency Critical Apps
<b>HDD-backed volumes:</b> (Magnetic Drive)	<b>Low-Cost throughput-intensive workloads</b> (Not Suitable for Low Latency(IOPS) -- i.e. booting)	
<b>Types HDD</b>	<b>Throughput Optimized HDD (st1)</b> Low Cost - Frequently accessed, throughput-intensive & Large-Sequential O/I -- 500 MB/s	Stream Processing Big Data Processing Data Warehouse
	<b>Cold HDD (sc1)</b> Lowest Cost - less frequently accessed data Throughput : 250 MiB/s	Colder Data requires fewer scans per day.

### Features:

- High Performance (Provides single-digit-millisecond latency for high-performance)
- Highly Scalable (Scale to petabytes)
- Offers high availability (guaranteed 99.999% by Amazon) & Durability
- Offers seamless encryption of data at rest through Amazon Key Management Service (KMS).
- Automate Backups through **data lifecycle policies** using EBS Snapshots to S3 Storage.
- EBS detached from an EC2 instance and attached to another one quickly.

## Key Points to Remember:

- **Backup/Migration:** To move a volume across AZs, you first need to take a snapshot.
- **Provisioned capacity:** capacity needs to be provisioned in advanced (GBs & IOPS)
- You can increase the capacity of the drive over time.
- It can be detached from an EC2 instance and attached to another one quickly.
- It's locked to **Single Availability Zone (AZ)**
- The default volume type is General Purpose SSD (gp2)
- EBS Volume can be mounted parallelly using RAID Settings:
  - RAID 0 (increase performance)
  - RAID 1 (increase fault tolerance)
- It's a network drive (i.e. not a physical drive).
- Unencrypted volume can be encrypted using an encrypted snapshot
- Snapshot of the encrypted volume is encrypted by default.
- When you share an encrypted snapshot, you must also share the customer-managed CMK used to encrypt the snapshot.

## Pricing:

- You will get billed for all the provisioned capacity & snapshots on S3 Storage + Sharing Cost between AZs/Regions

## EBS vs Instance Store

### Instance Store (ephemeral storage) :

- It is ideal for temporary block-level storage like buffers, caches, temporary content
- Data on an instance store volume persists only during the life of the associated instance. (As it is volatile storage - lose data if stop the instance/instance crash)
- **Physically attached to ec2 instance** - hence, the **lowest possible latency**.
- **Massive IOPS - High performance**
- Instance store backed Instances can be of maximum 10GiB volume size
- Instance store volume cannot be attached to an instance, once the Instance is up and running.
- Instance store volume can be used as root volume.
- You cannot create a snapshot of an instance store volume.

### EBS :

- Persistent Storage.
- Reliable & Durable Storage.
- EBS volume can be detached from one instance and attached to another instance.
- EBS boots faster than instance stores.

# AWS EFS - Elastic File Storage

## What is AWS EFS?

Amazon Elastic File System (Amazon EFS) provides a scalable, fully managed elastic distributed file system based on NFS. It is persistent file storage & can be easily scaled up to petabytes.

It is designed to share parallelly with thousands of EC2 instances to provide better throughput and IOPS. It is a regional service automatically replicated across multiple AZ's to provide High Availability and durability.

## Types of EFS Storage Classes:

Standard Storage	For frequently accessed files.
Infrequent Access Storage ( EFS-IA )	For files not accessed every day Cost-Optimized (costs only \$0.025/GB-month) Use EFS Lifecycle Management to move the file to EFS IA

## EFS Access Modes :

- 1) Performance Modes:
  - General Purpose: low latency at the cost of lower throughput.
  - Max I/O: high throughput at the cost of higher latency.
- 2) Throughput Modes :
  - Bursting (default): throughput grows as the file system grows
  - Provisioned: specify throughput in advance. (fixed capacity)

## Features:

- Fully Managed and Scalable, Durable, Distributed File System (NFSv4)
- Highly Available & Consistent low latencies. (EFS is based on SSD volumes)
- POSIX Compliant (NFS) Distributed File System.
- EC2 instances can access EFS across AZs, regions, VPCs & on-premises through AWS Direct Connect or AWS VPN.
- Provides EFS Lifecycle Management for the better price-performance ratio
- It can be integrated with AWS Datasync for moving data between on-premise to AWS EFS
- Supported Automatic/Schedule Backups of EFS (AWS Backups)
- It can be integrated with CloudWatch & CloudTrail for monitoring and tracking.
- EFS supports encryption at transit(TLS) and rest both. (AWS Key Management Service (KMS))
- Different Access Modes: Performance and Throughput for the better cost-performance tradeoff.
- EFS is more expensive than EBS.
- Once your file system is created, you cannot change the performance mode
- Not suitable for boot volume & highly transactional data (SQL/NoSQL databases)
- Read-after-write consistency for data access.
- Integrated with IAM for access rights & security.

### **Use Cases:** (Sharing Files Across instances/containers)

- Mission critical business applications
- Microservice based Applications
- Container storage
- Web serving and content management
- Media and entertainment file storage
- Database Backups
- Analytics and Machine Learning

### **Best Practices:**

- Monitor using cloudWatch and track API using CloudTrails
- Leverage IAM services for access rights and security
- Test before fully migrating mission critical workload for performance and throughput.
- Separate out your latency-sensitive workloads. Storing these workloads on separate volumes ensures dedicated I/O and burst capabilities.

### **Pricing:**

- Pay for what you have used based on Access Mode/Storage Type + Backup Storage.

## **Amazon FSx for Windows File Server**

### **What is Amazon FSx for Windows File Server?**

- Amazon FSx for Windows File Server is an FSx solution that offers a scalable and shared file storage system on the Microsoft Windows server.
- Using the Server Message Block (SMB) protocol with Amazon FSx Can access file storage systems from multiple windows servers.
- It offers to choose from HDD and SSD storage, offers high throughput, and IOPS with sub-millisecond latencies for Windows workloads.
- Using SMB protocol, Amazon FSx can connect file systems to Amazon EC2, Amazon ECS, Amazon WorkSpaces, Amazon AppStream 2.0 instances, and on-premises servers using AWS Direct Connect or AWS VPN.
- It provides high availability (Multi-AZ deployments) with an active and standby file server in separate AZs.
- It automatically and synchronously replicates data in the standby Availability Zone (AZ) to manage failover.
- Using AWS DataSync with Amazon FSx helps to migrate self-managed file systems to Windows storage systems.
- It offers identity-based authentication using Microsoft Active Directory (AD).
- It automatically encrypts data at rest with the help of AWS Key Management Service (AWS KMS). It uses SMB Kerberos session keys to encrypt data in transit.

### **Use cases:**

- Large organizations which require shared access to multiple data sets between multiple users can use Amazon FSx for Windows File Server.
- Using Windows file storage, users can easily migrate self-managed applications to AWS using AWS DataSync.
- It helps execute business-critical Microsoft SQL Server database workloads easily and automatically handles SQL Server Failover and data replication.
- Using Amazon FSx for Windows File Server, users can easily process media workloads with low latencies and high throughput.
- It enables users to execute high intensive analytics workloads, including business intelligence and data analytics applications.

### **Price details:**

- Charges are applied monthly based on the storage and throughput capacity used for the file system's file system and backups.
- The cost of storage and throughput depends on the deployment type, either single-AZ or multi-AZ.

## Amazon FSx for Lustre

### What is Amazon FSx for Lustre?

- Amazon FSx for Lustre is an FSx solution that offers scalable storage for the Lustre system (parallel and high-performance file storage system).
- It supports fast processing workloads like custom electronic design automation (EDA) and high-performance computing (HPC).
- It provides shared file storage with hundreds of gigabytes of throughput, sub-millisecond latencies, and millions of IOPS.
- It offers a choice between SSD and HDD for storage.
- It integrates with Amazon S3 to process data concurrently using parallel data-transfer techniques.
- It stores datasets in S3 as files instead of objects and automatically updates with the latest data to run the workload.
- It offers to select unreplicated file systems for shorter-term data processing.
- It can be used with existing Linux-based applications without any changes.
- It offers network access control using POSIX permissions or Amazon VPC Security Groups.
- It easily provides data-at-rest and in-transit encryption.
- AWS Backup can also be used to backup Lustre file systems.
- It integrates with SageMaker to process machine learning workloads.

### Use cases:

- The workloads which require shared file storage and multiple compute instances use Amazon FSx for Lustre for high throughput and low latency.
- It is also applicable in media and big data workloads to process a large amount of data.

### Price details:

- Charges are applied monthly in GB based on the storage capacity used for the file system.
- Backups are stored incrementally, which helps in storage cost savings.

# Amazon S3 Glacier

## What is Amazon S3 Glacier?

Amazon S3 Glacier is a web service with vaults that offer long-term data archiving and data backup.

It is the cheapest S3 storage class and offers 99.999999999% of data durability.

It helps to retain unlimited data like photos, videos, documents as TAR or ZIP file, data lakes, analytics, IoT, machine learning, and compliance data.

S3-Standard, S3 Standard-IA, and S3 Glacier storage classes, objects, or data are automatically stored across availability zones in a specific region.

S3 Glacier provides the following data retrieval options:

- Expedited retrievals -
  - It retrieves data in 1-5 minutes.
- Standard retrievals -
  - It retrieves data between 3-5 hours.
- Bulk retrievals -
  - It retrieves data between 5-12 hours.

## Features:

- It integrates with AWS IAM to allow vaults to grant permissions to the users.
- It integrates with AWS CloudTrail to log and monitor API call activities for auditing.
- A vault is a place for storing archives with certain functionalities like to create, delete, lock, list, retrieve, tag, and configure.
- Vaults can be set with access policies for additional security by the users.
- Amazon S3 Glacier jobs are the select queries that execute to retrieve archived data.
- It uses Amazon SNS to notify when the jobs complete.
- It uses 'S3 Glacier Select' to query specific archive objects or bytes for analytics instead of complete archives.
- S3 Glacier Select operates on uncompressed comma-separated values (CSV format) and output results to Amazon S3.
- Amazon S3 Glacier Select uses SQL queries using SELECT, FROM, and WHERE.
- It offers only SSE-KMS and SSE-S3 encryption.
- Amazon S3 Glacier does not provide real-time data retrieval of the archives.

## Use Cases:

- It helps to store and archive media data that can increase up to the petabyte level.
- Organizations that generate, analyze, and archive large data can make use of Amazon S3 Glacier and S3 Glacier Deep Archive storage classes.
- Amazon S3 Glacier replaces tape libraries for storage because it does not require high upfront cost and maintenance.

## Price details:

- Free Usage Tier - Users can retrieve with standard retrieval up to 10 GB of archive data per month for free.
- Data transfer out from S3 Glacier in the same region is free.

## AWS Snowball

### What is AWS Snowball?

- AWS Snowball is a storage device used to transfer a large amount of data ranging from 50TB - 80TB between Amazon Simple Storage Service and onsite data storage location at high speed.
- These devices are protected by the AWS Key Management Service to protect data in transit securely.
- AWS Snow Family Management Console helps to manage data transfer jobs using job management API.
- The Snowball client and the Amazon S3 Adapter for Snowball are used to perform data transfers on the Snowball device locally.
- If data transfers involve large files and multiple jobs, you might separate the data into several smaller data segments. Troubleshooting a large transfer can be more complicated than a small transfer. Parallelization helps to transfer data with Snowball at a faster rate.
- AWS Snowball is integrated with other AWS services such as AWS CloudTrail to capture all API calls as events and with Amazon Simple Notification Service (Amazon SNS) to notify about data transfer.
- AWS Snowball Edge is a type of Snowball device that can transport data at speeds faster than the internet and can do local processing and edge-computing workloads between the local environment and the AWS Cloud.
- Using Snowball Edge devices, one can execute EC2 AMIs and deploy AWS Lambda code on the devices to perform processing and analysis with the applications.

There are two other categories of the AWS Snow family:

- Snowball Edge Compute Optimized - provide block storage, object storage, and 40 vCPUs
- Snowball Edge Storage Optimized - provides block storage, object storage, and 52 vCPUs, and an optional GPU for high processing use cases.

### Use cases:

- AWS Snowball helps to transfer or receive large amounts of data with clients or partners regularly.
- AWS Snowball collects large data and performs analysis to overcome failure and improve safety, efficiency and productivity.

### Pricing details:

Charges are applied based on the following components:

- Service Fee per Job - region-specific.
- Per Day Fee to keep snowfall onsite - region-specific
- Data Transfer fee for Amazon S3:
  - Data transfer IN - free.
  - Data transfer OUT - region-specific.
- Shipping Costs - standard carrier rates.

# AWS Storage Gateway

## What is the AWS Storage Gateway?

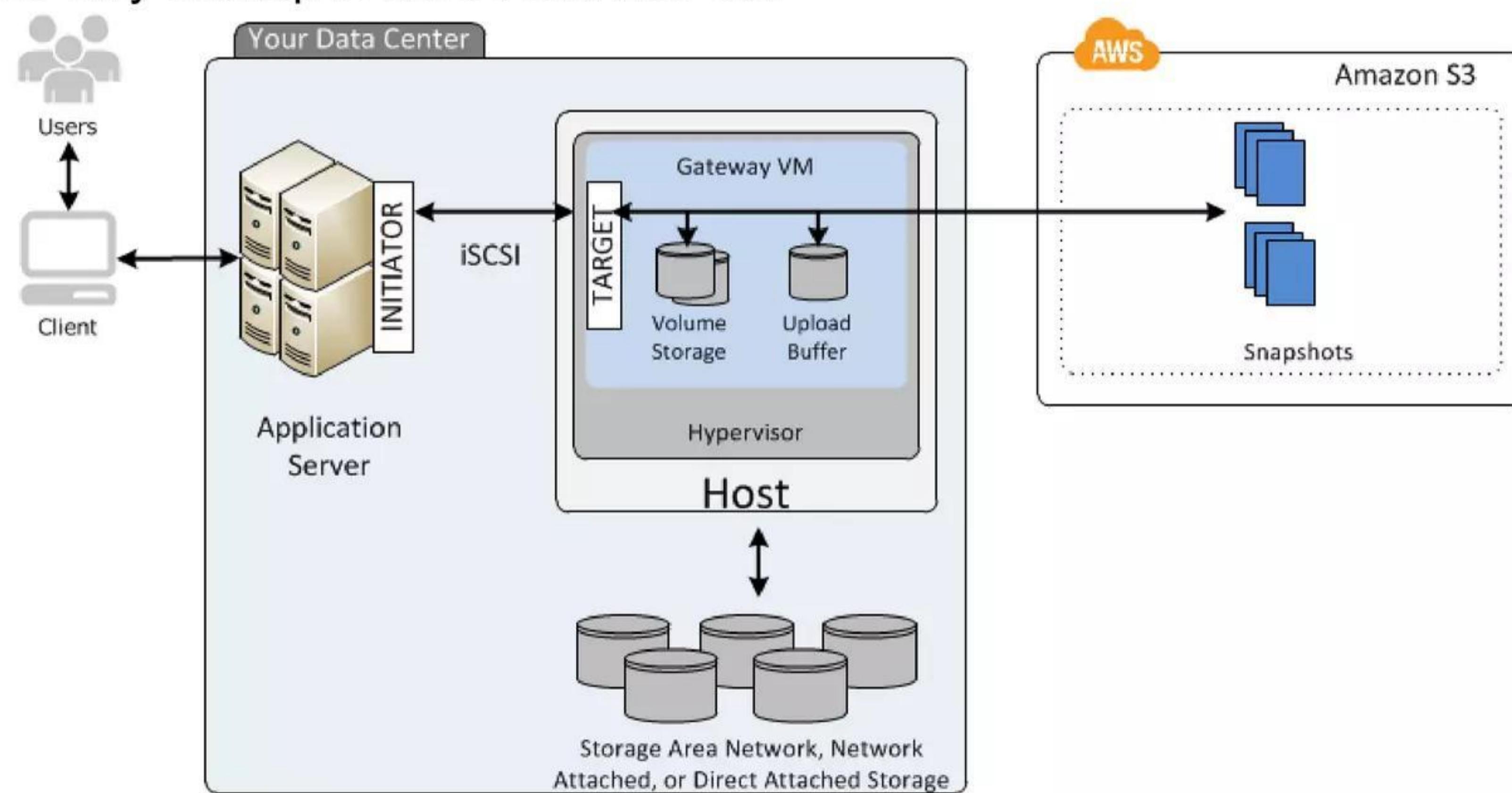
AWS Storage Gateway is a **hybrid cloud storage service** that allows your on-premise storage & IT infrastructure to seamlessly integrate with AWS Cloud Storage Services. It Can be AWS Provided Hardware or Compatible Virtual Machine.

## Purpose of Using AWS Storage Gateway(hybrid Cloud Storage) :

- To Fulfill Licencing Requirements.
- To Achieve Data-Compliance Requirements.
- To Reduce Storage & Management Cost.
- For Easy and Effective Application Storage-Lifecycle & Backup Automation.
- For Hybrid Cloud & Easy Cloud Migration.

## Volume Gateway (iSCSI)

- To Access Virtual Block-Level Storage Stored on-premise
- It can be asynchronously backed up and stored as a snapshot on AWS S3 for high reliability & durability.
  - **Storage Volume Gateway:** All Applications Data Stored on-premise and the only backup is stored on AWS S3.



- **Cache Volume Gateway:** Only Hot Data / Cached data is Stored on-premise and all other application data is stored on AWS S3.

