LEARNING AWS

Hands –On!

Course Syllabus

- **Module 1:** Introduction to Cloud Computing and AWS
- **Module 2:** Amazon Elastic Compute Cloud and Amazon Elastic Block Store
- **Module 3:** Amazon Simple Storage Service and Amazon Glacier Storage
- **Module 4:** Amazon Virtual Private Cloud
- **Module 5:** Databases
- **Module 6:** Authentication and Authorisation

Course Syllabus

- **Module 7:** CloudTrail, CloudWatch, and AWS Config
- **Module 8:** Domain Name System and Network Routing
- **Module 9:** Reliability Pillar
- **Module 10:** Performance Efficiency Pillar
- **Module 11:** Security Pillar
- **Module 12:** Cost Optimisation Pillar
- **Module 13:** Operational Excellence Pillar

Module 1: Introduction to Cloud Computing and AWS

Cloud Computing and Virtualisation

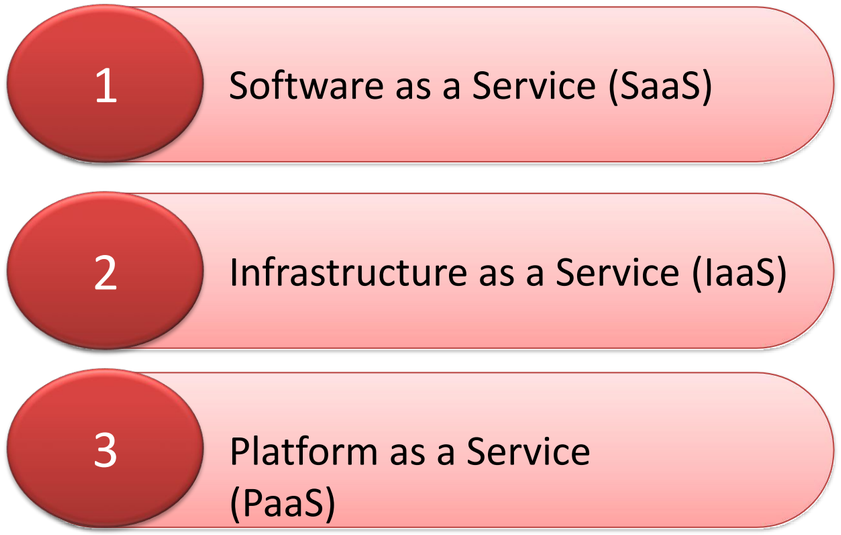
Cloud Computing

- Cloud computing is a computer model that allows workloads to run in a virtualised environment
- In cloud computing, virtualisation is used to replace physical files, servers, networks, files, programmes, devices, and infrastructure with computer-generated versions hosted and managed by a service provider
- Providers utilise management software to automate repeated procedures and manage the data, security features, storage capacity, and processing power needed to send data between user devices and the cloud

Cloud Computing and Virtualisation

(Continued)

- Cloud computing services are usually classified into one of three categories:



Cloud Computing and Virtualisation

1. Software as a Service (SaaS)

- The most common sort of cloud-based service is SaaS, which allows users to access software via a browser or app without the need for any hardware installation or maintenance

2. Infrastructure as a Service (IaaS)

- IaaS manages the customer's software, hardware, servers, storage, and any other critical requirements. IaaS consumers, unlike SaaS users, only pay for what they use on a weekly or monthly basis. Some suppliers will even let you pay by the hour. IaaS allows for regular, quick growth in both directions, even if it is not always the most cost-effective alternative

3. Platform as a Service (PaaS)

- PaaS is a cloud environment that facilitates the creation and deployment of applications. From building and testing to deployments and upgrades, vendors supply everything a company needs to manage the whole development lifecycle from a single central location

Cloud Computing and Virtualisation

Virtualisation

- Virtualisation is a method of generating computer-generated versions of servers, apps, data centres, and other forms of hardware that act like their physical counterparts
- Virtualisation software uses a "hypervisor," that allows a single computer to host many virtual machines (VMs)
- Virtual machines (VMs) are software containers that run their own operating systems and act like standalone computers despite only using a small portion of the underlying hardware
- For more efficient hardware utilisation, the hypervisor also assigns computing power to each VM as needed

Cloud Computing and Virtualisation

Characteristics of virtualisation

1. **Resource Sharing**

- Virtualisation allows users to build many computing environments from a single host machine (or a network of connected servers). This allows users to manage the number of active servers, save power usage, and limit the number of active servers

2. **Isolation**

- Self-contained virtual machines (VMs) in virtualisation software provide an isolated online environment for guest users (which includes not only people but also applications, OSs, and gadgets). This separation keeps sensitive data safe while allowing guests to stay connected

Cloud Computing and Virtualisation

3. Availability

- Virtualisation software has various advantages over physical servers, including increased uptime, availability, fault tolerance, and help users avoid downtime, which reduces productivity and poses security and safety risks

4. Aggregation

- Virtualisation allows various devices to share resources from a single machine, but it may also be used to integrate multiple devices into a single powerful host. Cluster management software is required for aggregation, which links a homogeneous set of computers or servers to create a unified resource centre

5. Reliability

- Automated load balancing, which runs redundant servers on different host machines to prevent disruptions, ensures continual uptime for virtualisation platforms

VIRTUAL SERVER

- It is a Virtualized Instance of a computer system that runs an operating system and applications.
- Virtual server is like having many computers inside one, all working together to do different tasks for different people, even though they're really just using parts of the same big computer. It's like having a bunch of secret hidden computers that you can use without needing to buy lots of real computers.
- Virtualization technology allows multiple virtual servers to coexist on a single physical server, each operating as an isolated and independent entity.
- The concept of virtual servers within virtualization provides a way to make more efficient use of hardware resources, enhance flexibility, and streamline management of IT environments. This technology has been a key driver in data center optimization and the cloud computing revolution.

AWS Cloud

- Cloud computing is defined as an internet-based computing platform that connects large groups of remote servers for centralised data storage and online access to computer services and resources

- Organisations can use shared computing and storage resources instead of constructing, operating, and expanding infrastructure on their own when they use cloud computing

- The following features are enabled by cloud computing:

  - On-demand resources can be provisioned and released

  - Depending on the load, resources can be automatically scaled up or down

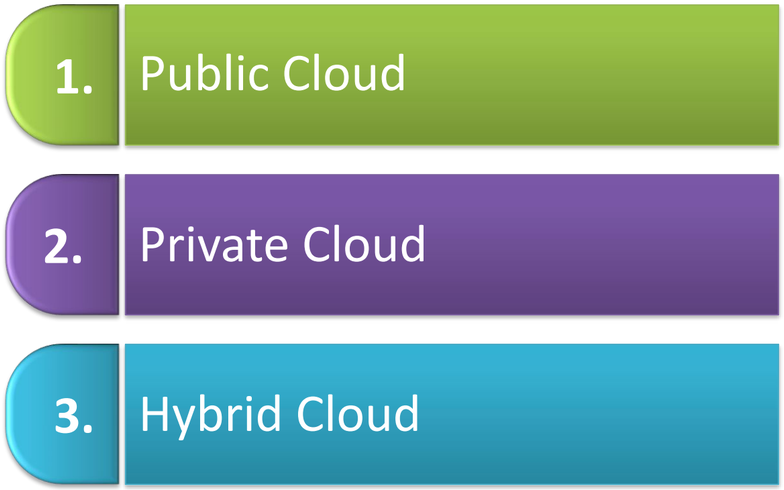
  - Resources can be accessed securely over the internet

  - Pay-as-you-go models are available from cloud service providers, in which clients are charged based on the type of resources and utilisation

AWS Cloud

Types of Clouds

- There are three types of clouds:



AWS Cloud

1. **Public Cloud**

- Third-party service providers make resources and services available to their clients via the Internet in the public cloud. The data and security of the customer are stored on the service provider's infrastructure

2. **Private Cloud**

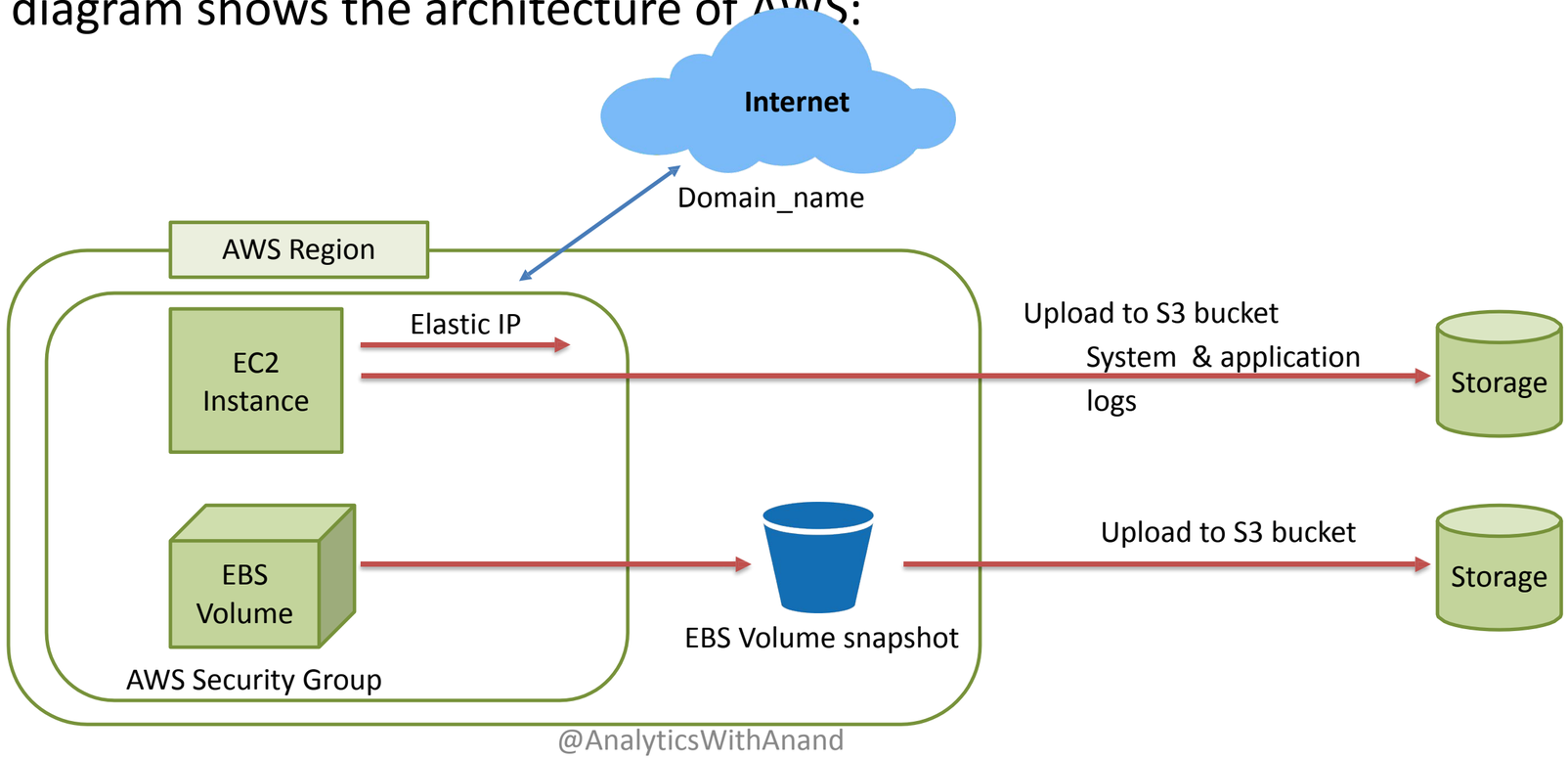
- A private cloud offers essentially identical functionality to a public cloud, but the data and services are managed only for the customer's company by the business or a third party. As this type of cloud has complete control over the infrastructure, security issues are minimised

3. **Hybrid Cloud**

- A hybrid cloud combines private and public cloud resources. The decision to run on a private or public cloud is usually influenced by a number of factors, including the sensitivity of data and applications, industry certifications and needed standards, regulations, and so on

AWS Platform Architecture

- EC2 allows customers to create virtual computers with various settings based on their needs
- It enables several configuration options, various pricing options, and mapping of individual server, etc.
- The following diagram shows the architecture of AWS:



AWS Platform Architecture

Load balancing

- Load balancing refers to the distribution of hardware or software loads across web servers, which improves the server's and application's efficiency
- In traditional web application architectures, a hardware load balancer is a standard network appliance
- Elastic Load Balancing is an AWS service that distributes traffic to Amazon EC2 instances among various accessible sources and allows for dynamic addition and removal of Amazon EC2 hosts from the load-balancing cycle
- Elastic Load Balancing may dynamically increase and decrease load-balancing capacity in response to traffic demands, as well as provide sticky sessions for more complicated routing requirements

AWS Platform Architecture

Amazon Cloud-front

- Amazon Cloud-front is responsible for content delivery which is used to deliver the website. Using a global network of edge locations, it may include static, dynamic, and streaming content
- Content requests from users are automatically directed to the nearest edge site, which enhances speed
- Amazon Cloud-front is suitable to work with other Amazon Web Services such as Amazon EC2 and Amazon S3
- In addition to it, it also works with any non-AWS origin server and saves the original files in the same way

AWS Platform Architecture

Security Management

- Security groups, similar to an inbound network firewall, are a feature of Amazon's Elastic Compute Cloud (EC2) that allows us to designate which protocols, ports, and source IP ranges are allowed to access our EC2 instances

- One or more security groups can be set to each EC2 instance, each of which routes the necessary traffic to each instance. Security groups restrict access to EC2 instances by using specific subnets or IP addresses

Elastic Caches

- Amazon Elastic Cache is a cloud-based memory cache management service. Cache plays a significant role in memory management because it reduces stress on services and increases performance and scalability on the database tier by caching frequently used data

AWS Platform Architecture

Amazon RDS

- Amazon RDS (Relational Database Service) is a database engine that works similarly to MySQL, Oracle, and Microsoft SQL Server. Amazon RDS supports the same queries, applications, and tools

- It automatically updates database software and handles backups according to the user's preferences. Point-in-time recovery is also supported. There are no upfront costs, and we only pay for the resources that we utilise

Storage & Backups

- Web application data and assets can be stored, accessed, and backed up using the WS cloud

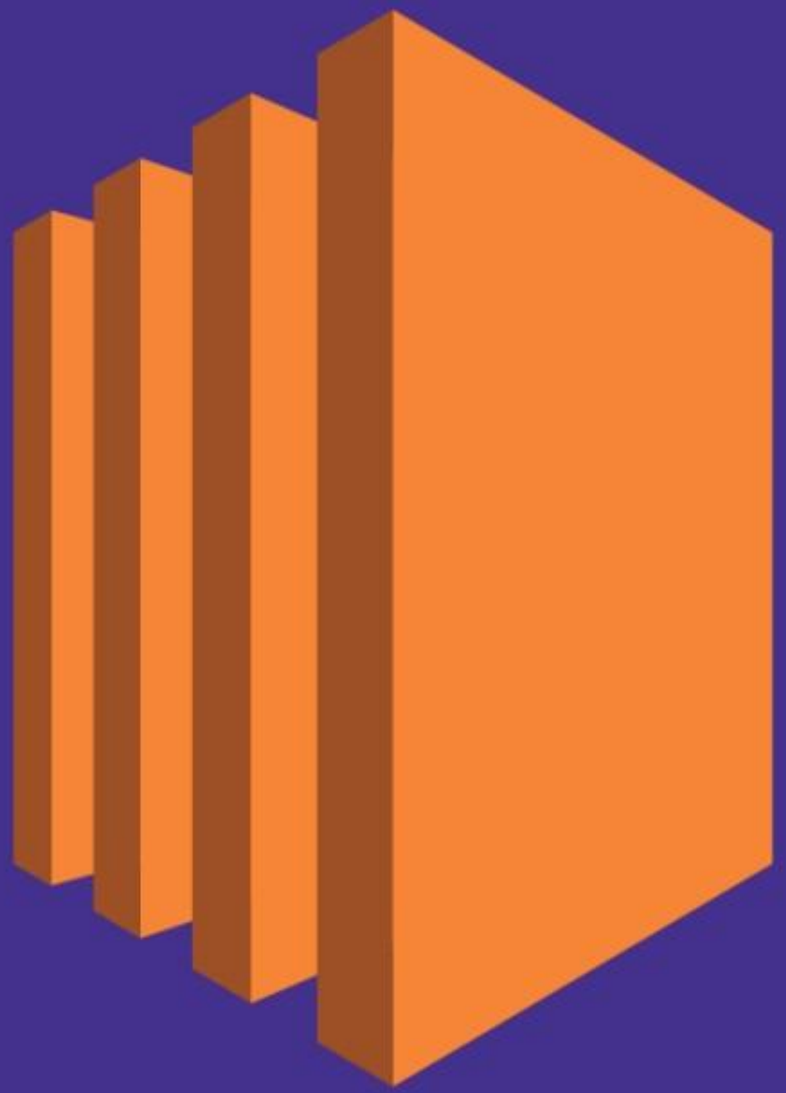
- The Amazon S3 (Simple Storage Service) provides a simple web-services interface for storing and retrieving any amount of data from anywhere on the internet at any time

AWS Platform Architecture

(Continued)

- Amazon S3 stores data as an objects inside the resources known as buckets
- The user can store as many objects as they need in the bucket and read, write, and remove them

Module 2: Amazon EC2 and Amazon Elastic Block Store

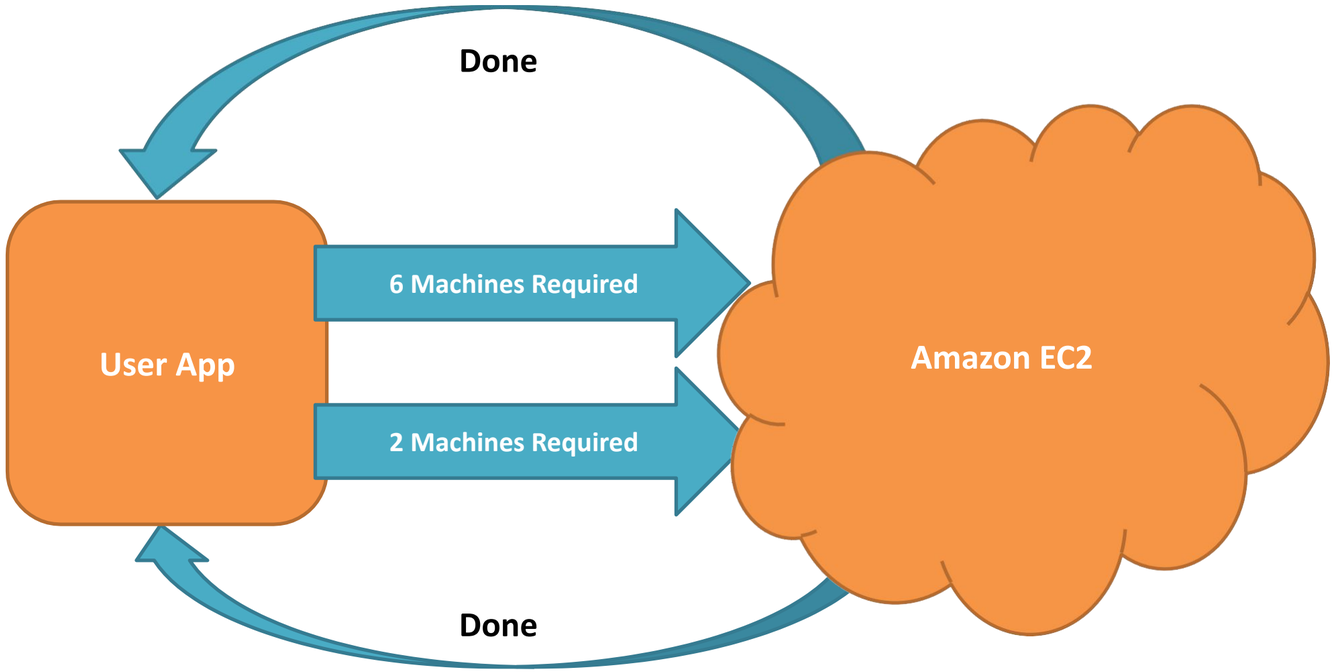


EC2

- Amazon EC2 (Amazon Elastic Compute Cloud) gives scalable computing capacity in the AWS (Amazon Web Services cloud)
- By using Amazon EC2 eliminates your need for investing in hardware upfront so that you may develop and deploy apps faster
- You may use Amazon EC2 for launching as many or as few virtual servers as you require, configuring security and networking, and also managing storage
- It enables you for scaling up or down for handling changes in requirements or spikes in popularity, decreasing your need for forecasting traffic

EC2

Working of Amazon EC2



EC2

(Continued)

- The following are the features that are provided by Amazon EC2:
  - Virtual computing environments, which are known as instances
  - Preconfigured templates for your instances, which is known as AMIs (Amazon Machine Images), which package the bits you require for your server (including the operating system (OS) and additional software)
  - Several configurations of CPU, storage, memory, and networking capacity for your instances, which is known as instance types
  - Storage volumes for temporary data that are deleted when you terminate or stop your instance, which is known as instance store volumes
  - The persistent storage volumes for your data using Amazon EBS (Amazon Elastic Block Store), which is known as Amazon EBS volumes

EC2

(Continued)

- Various physical locations for your resources, like Amazon EBS volumes and instances, which is known as Regions and Availability Zones
- A firewall which enables you for specifying the ports, protocols, and source IP ranges which can reach your instances by using security groups
- The static IPv4 addresses for dynamic cloud computing, which is known as Elastic IP addresses
- Metadata, also known as tags, which you can create and assign to your resources of Amazon EC2
- The virtual networks you may create which are logically isolated from the rest of the AWS cloud and which you may optionally connect to your network, which is known as VPCs (virtual private clouds)

INSTANCE

- In the cloud, an instance is a virtual server. The configuration of an instance at launch is a copy of the AMI which you defined when you launched an instance

- You may launch various instance's types from a single AMI. The type of an instance essentially determines the host computer's hardware used for your instance

- Every instance type gives heterogeneous compute and memory capabilities. Select an instance type on the basis of the amount of memory and computing power which you need for the software or application which you plan for running on the instance

KEY PAIR

- A key pair refers to a pair of cryptographic keys used for secure communication and authentication within the AWS ecosystem. The key pair consists of two components: a public key and a private key.

- **Public Key**: This key is meant to be shared openly. It is used by AWS services to encrypt data that only the corresponding private key can decrypt. Public keys are used for encrypting data or verifying the authenticity of a message or entity.

- **Private Key**: This key is kept secret and is known only to the owner. It is used for decrypting data that has been encrypted with the corresponding public key. Private keys are also used for digitally signing messages or data to prove their authenticity

- Key pairs provide a secure and convenient way to access and manage your resources within AWS while maintaining a high level of security.

- When you launch a new Amazon Elastic Compute Cloud (EC2) instance, you can associate a key pair with it. This allows you to securely log into the instance using SSH (for Linux instances) or Remote Desktop (for Windows instances) without needing to use a password. The private key is used to authenticate yourself to the instance.

SECURITY GROUP

- A security group in Amazon Web Services (AWS) is a fundamental component of network security that acts as a virtual firewall for your Amazon EC2 instances.
- A security group controls inbound and outbound traffic to and from these instances, allowing you to specify which traffic is allowed or denied based on rules you define.
- Each instance can be associated with one or more security groups.
- To manage security groups, you can use the AWS Management Console, AWS Command Line Interface (CLI), or AWS SDKs. Always ensure you follow security best practices to safeguard your resources and data in the cloud.

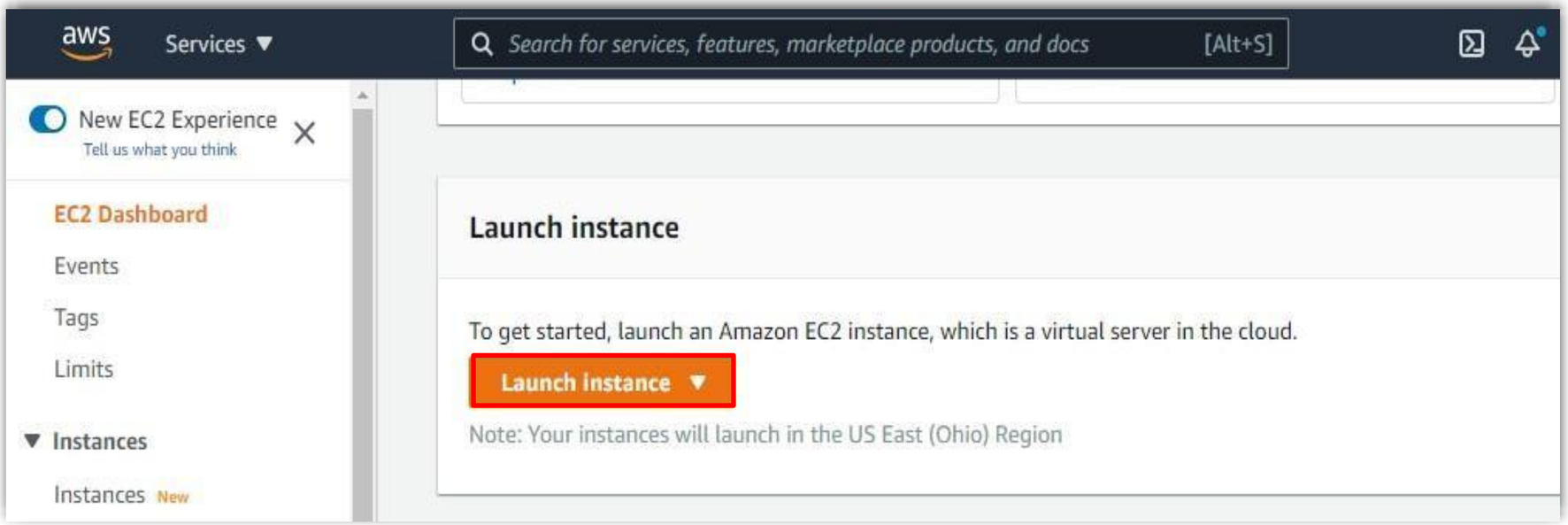
STACK

- Stack is a collection of AWS resources that are managed and provisioned together as a unit.
- AWS provides a service called AWS CloudFormation that allows you to create and manage stacks in a declarative way, using templates written in JSON or YAML.
- A stack is created when you use the CloudFormation service to deploy a template. The stack represents the set of resources defined in the template. Stacks are uniquely named within an AWS region and your AWS account.

LAUNCHING EC2

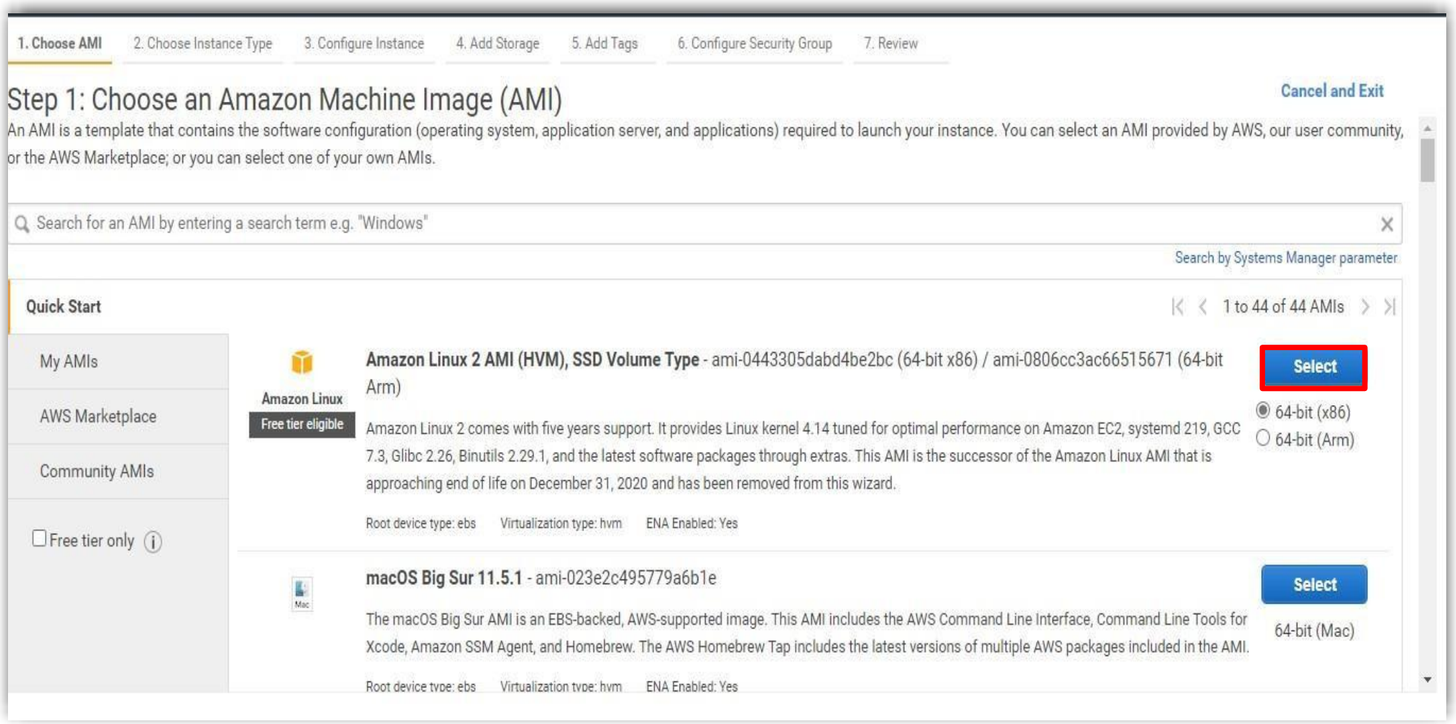
- Perform the following steps to make EC2 instances:

Step 1: Click on the **Launch instance**



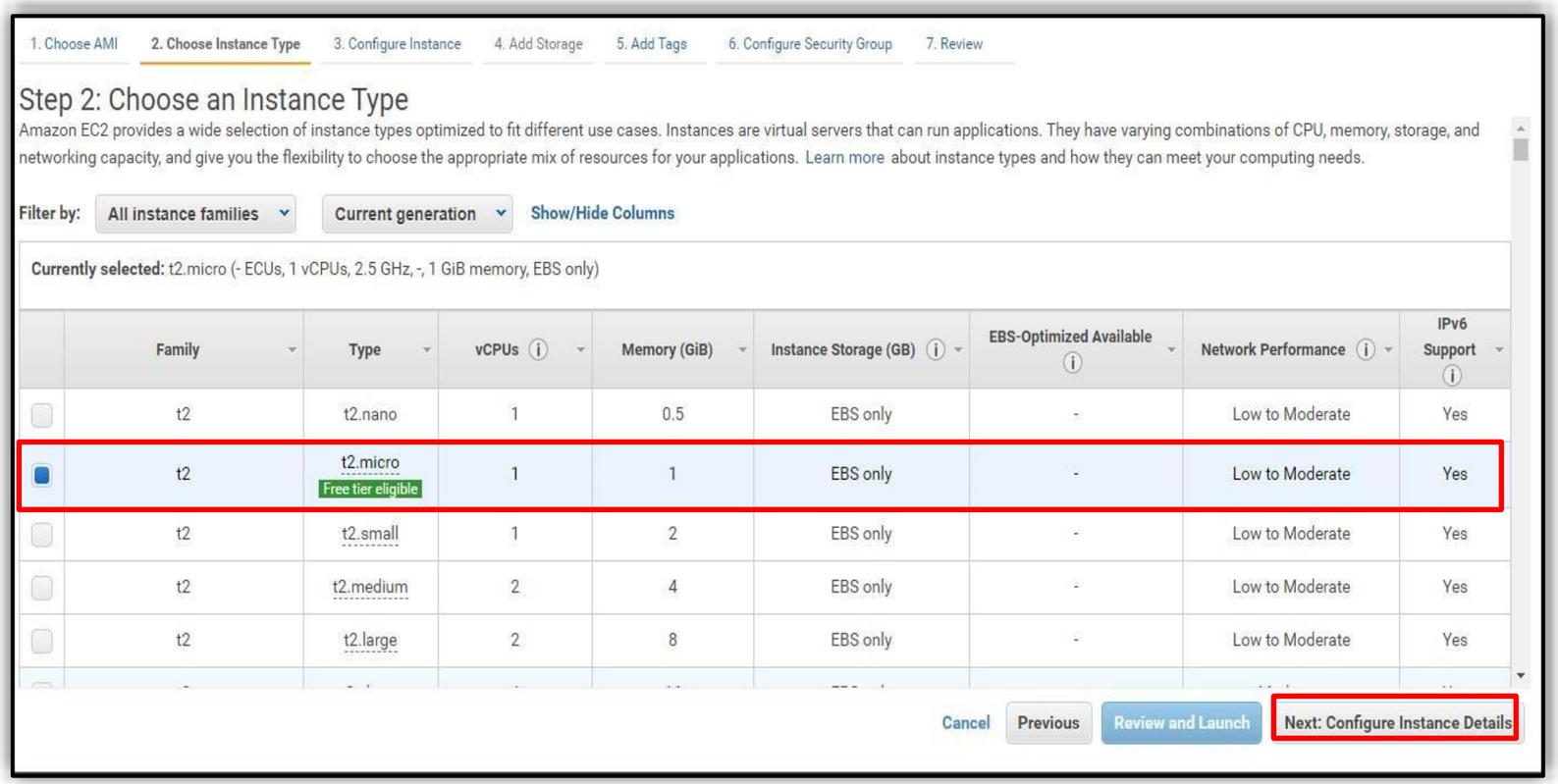
EC2

Step 2: Choose **Amazon Machine Image** then, click on **Select**



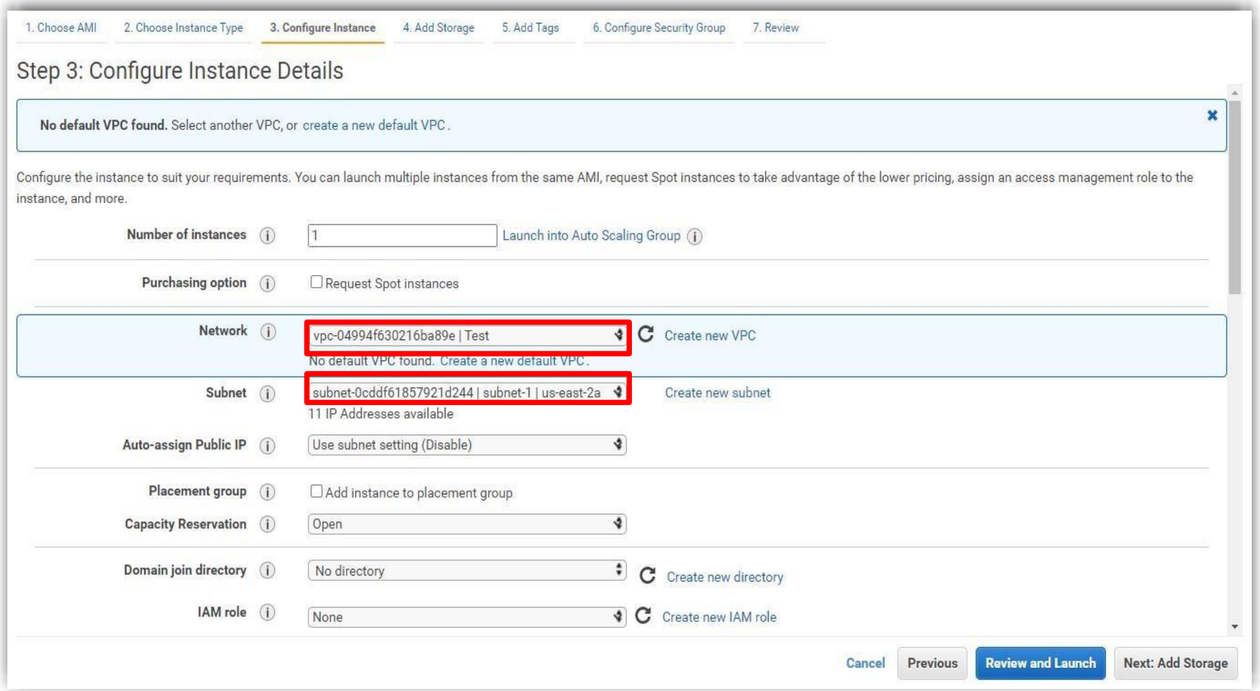
EC2

Step 3: Choose **t2 micro** as an instance type then, click on **Configure Instance Details**



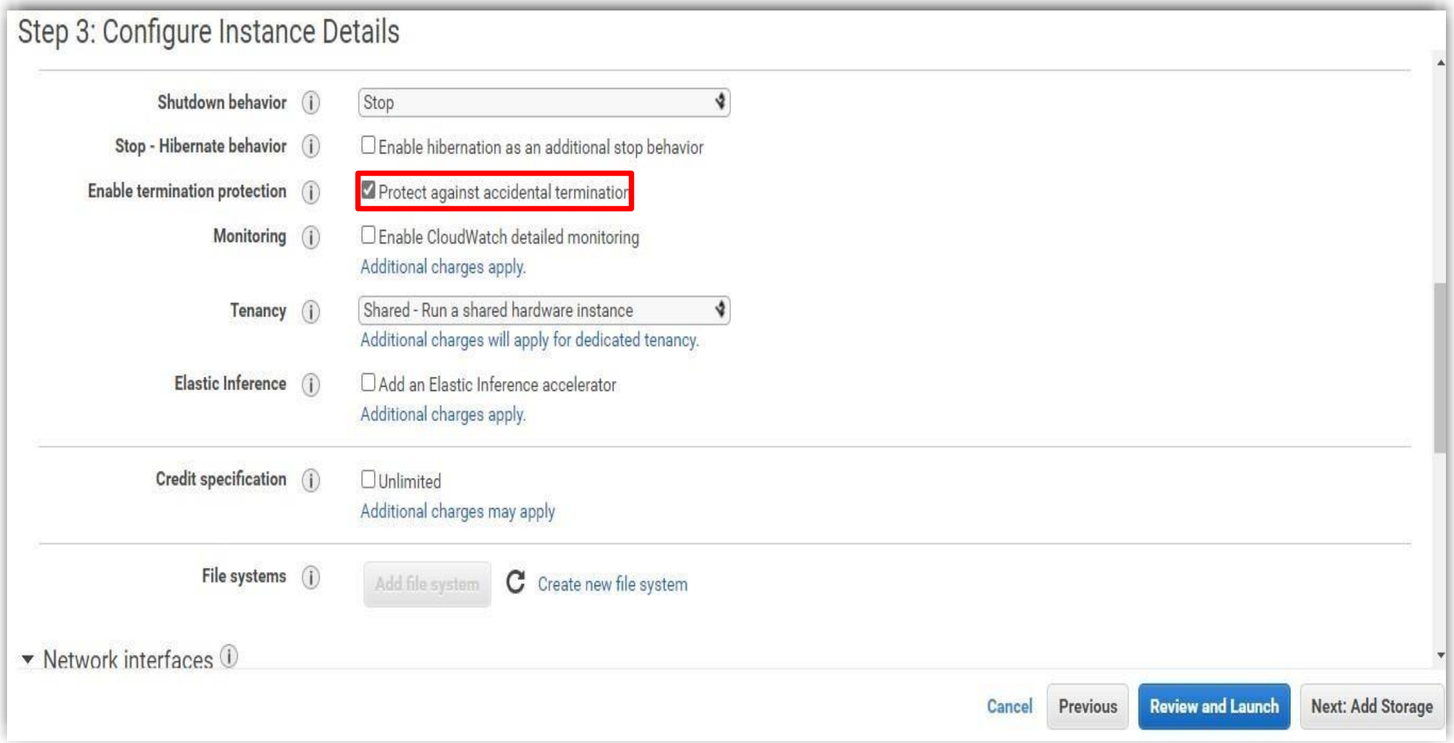
EC2

Step 4: Enter values in **Network** and **Subnet** field



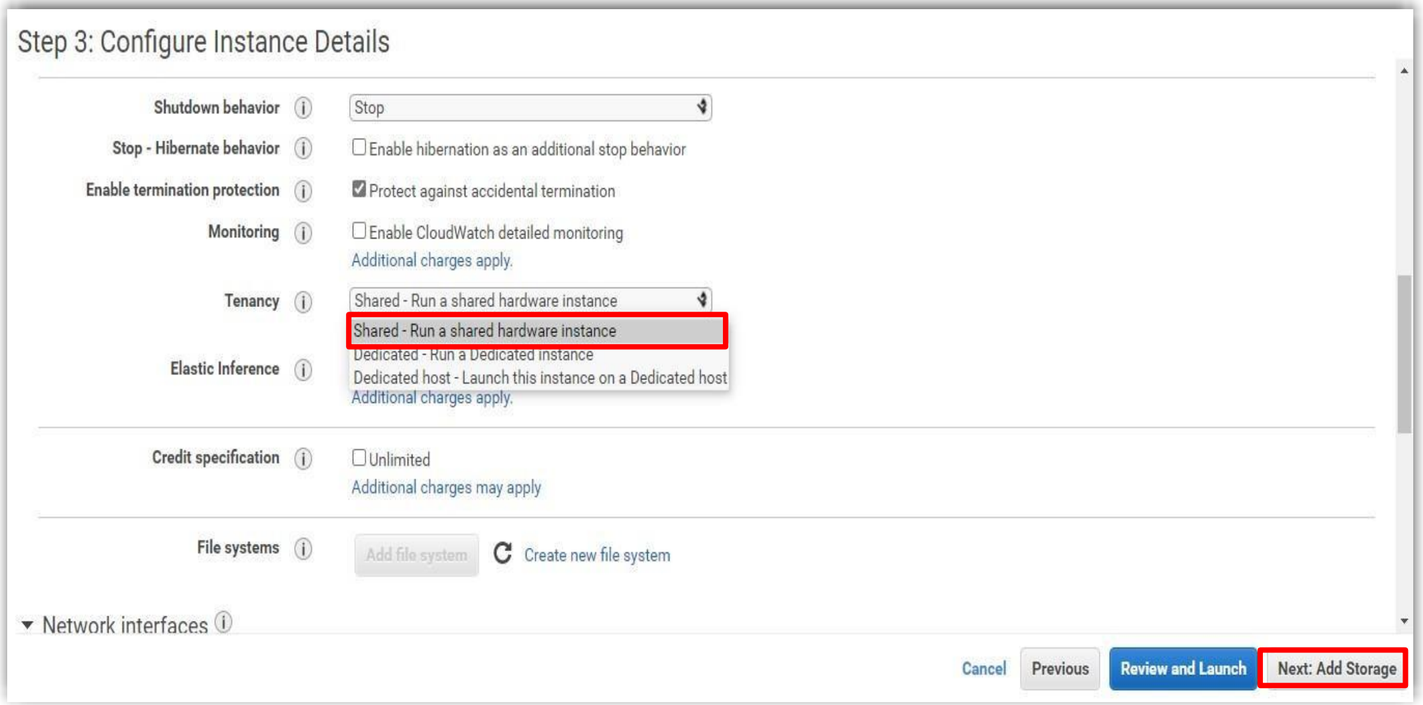
EC2

Step 5: Tick the checkbox of **Protect against accidental termination**



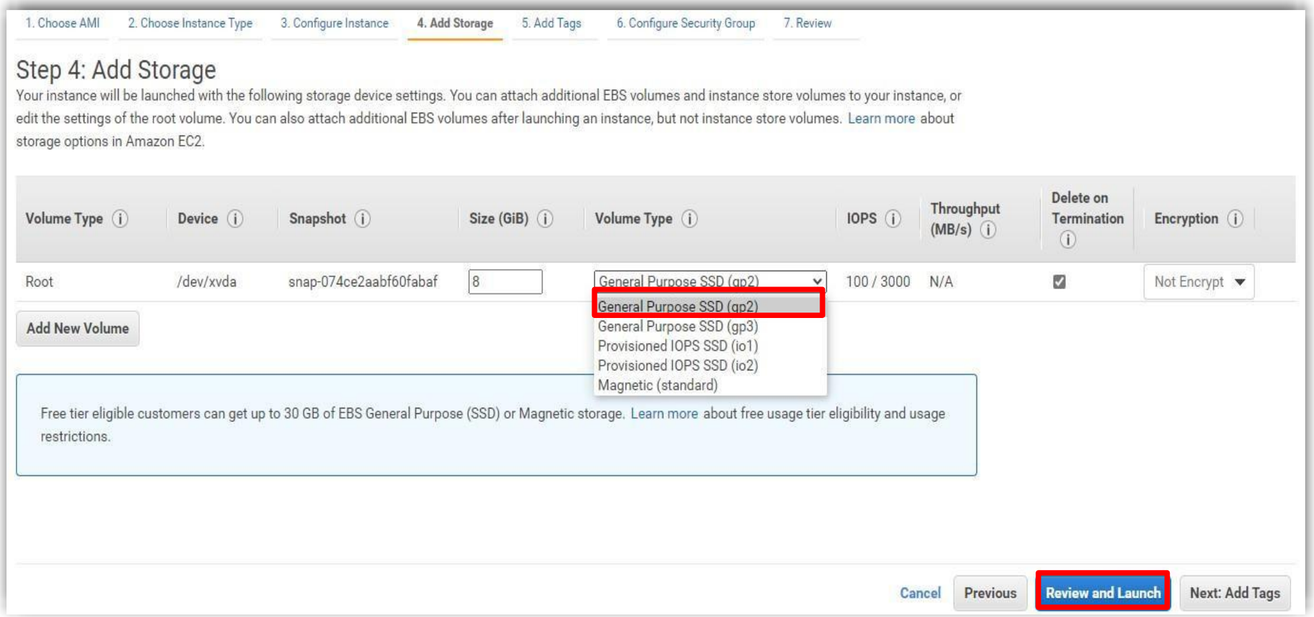
EC2

Step 6: Choose **Share-Run a shared hardware instance** Tenancy. Click on **Next: Add Storage**



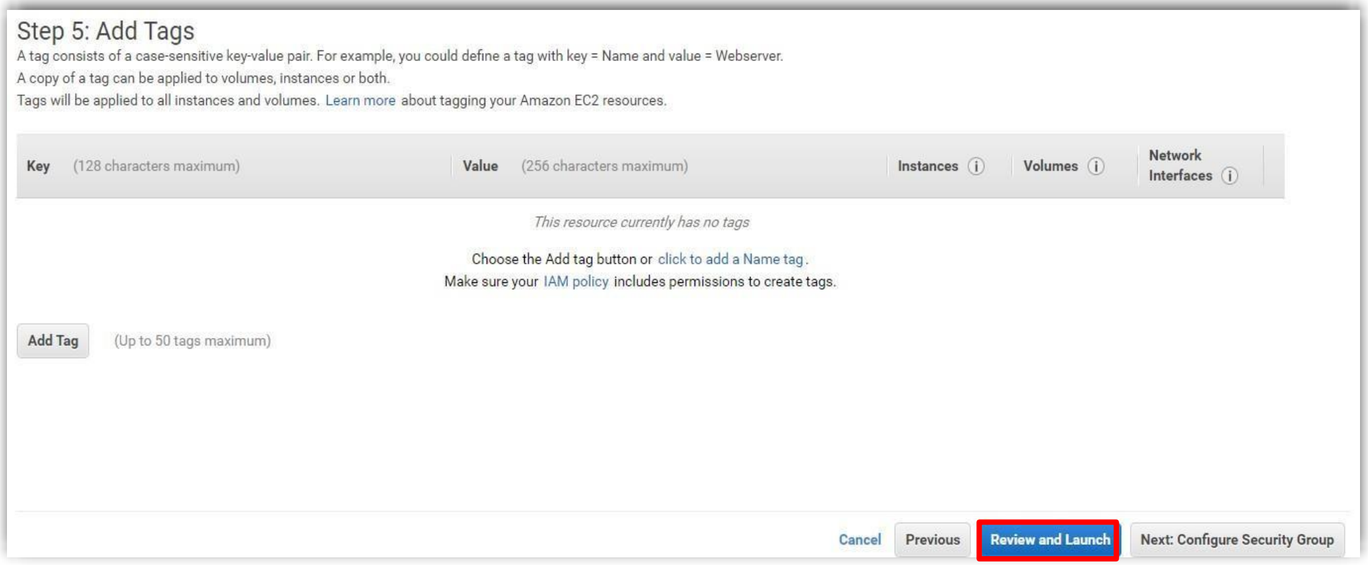
EC2

Step 7: Choose **Volume Type** as **General Purpose SSD (gp2)** and click on **Review and Launch**



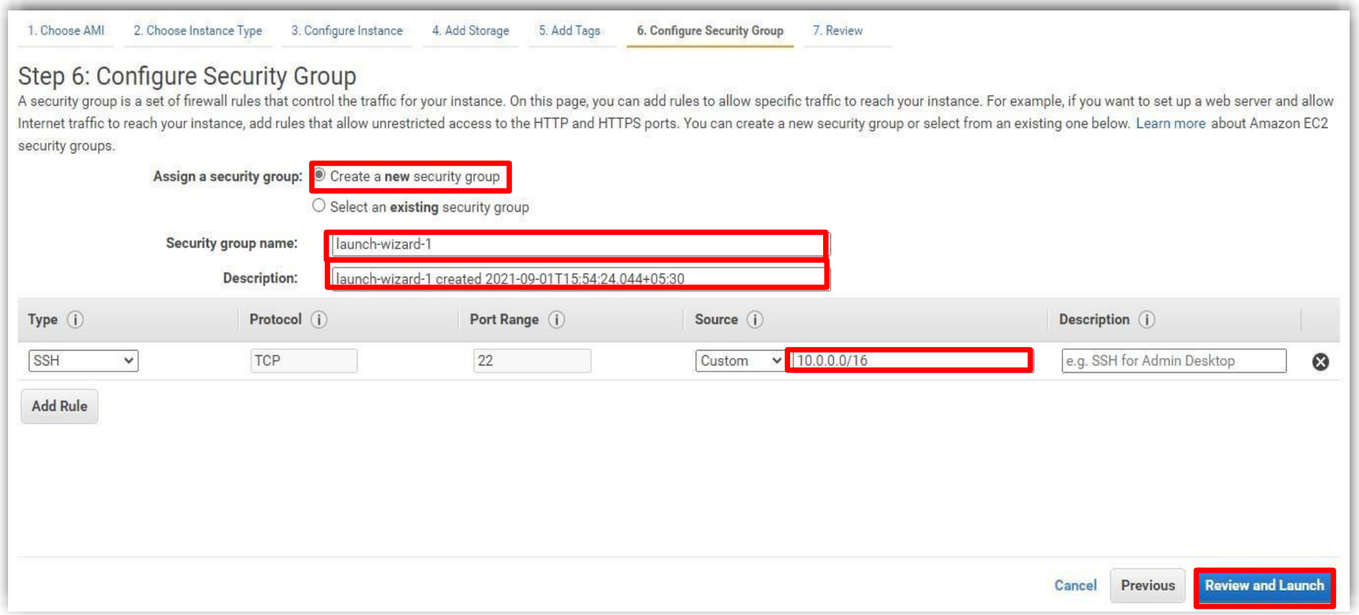
EC2

Step 8: Click on **Review and Launch**



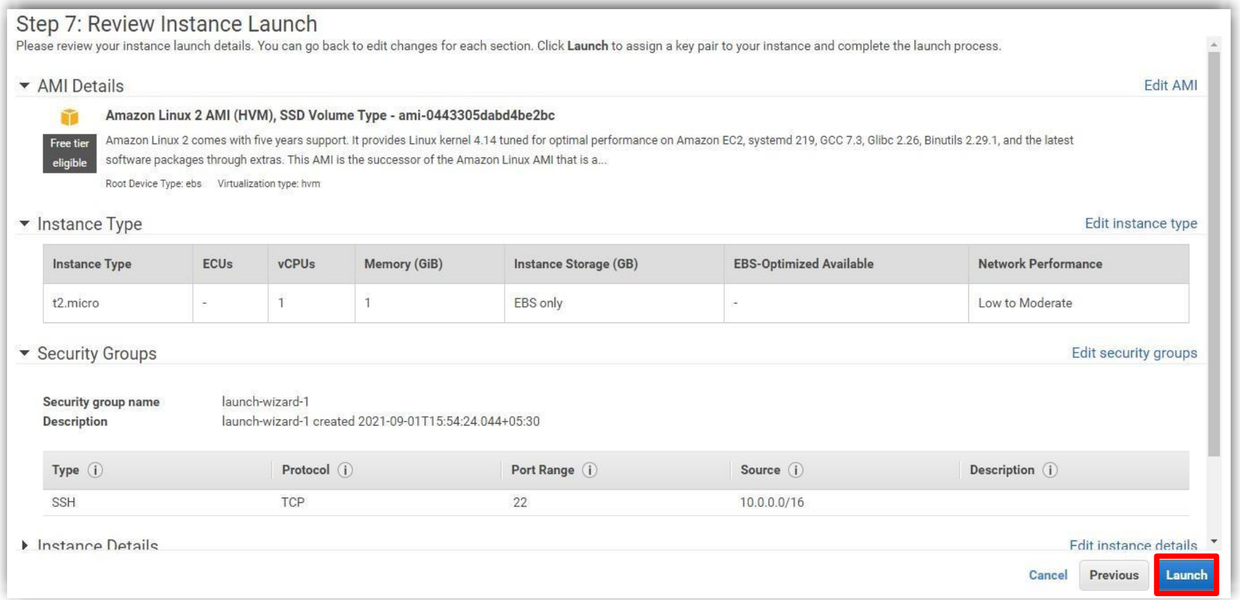
EC2

Step 9: Choose **Create a new security group** then, set the **Security group name** and **Description**. Choose Source as Custom and add custom IP. Click on **Review and Launch**



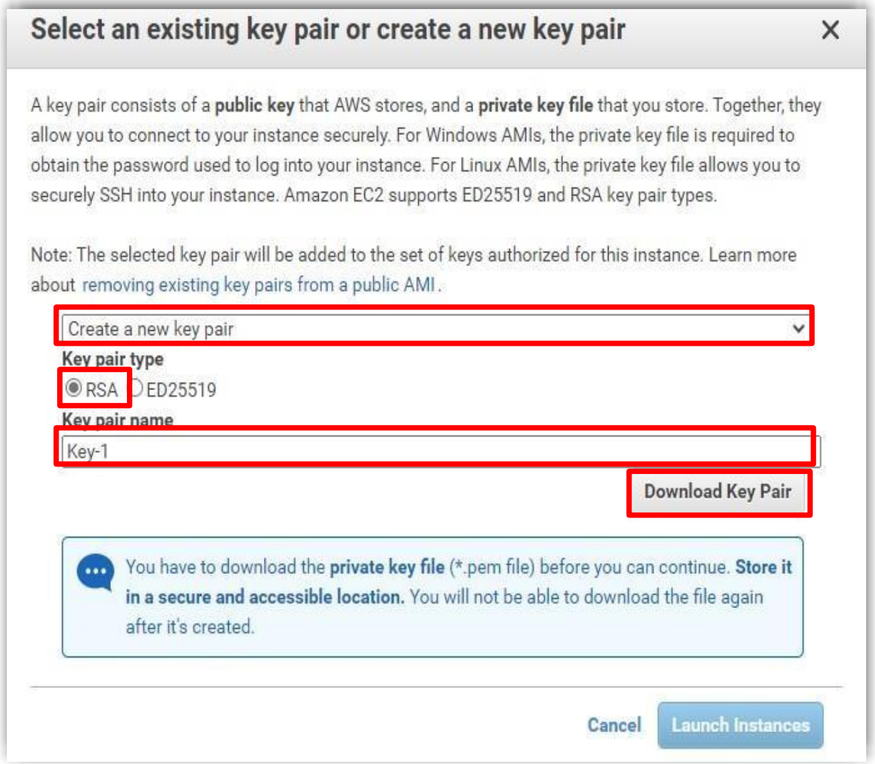
EC2

Step 10: Click **Launch**



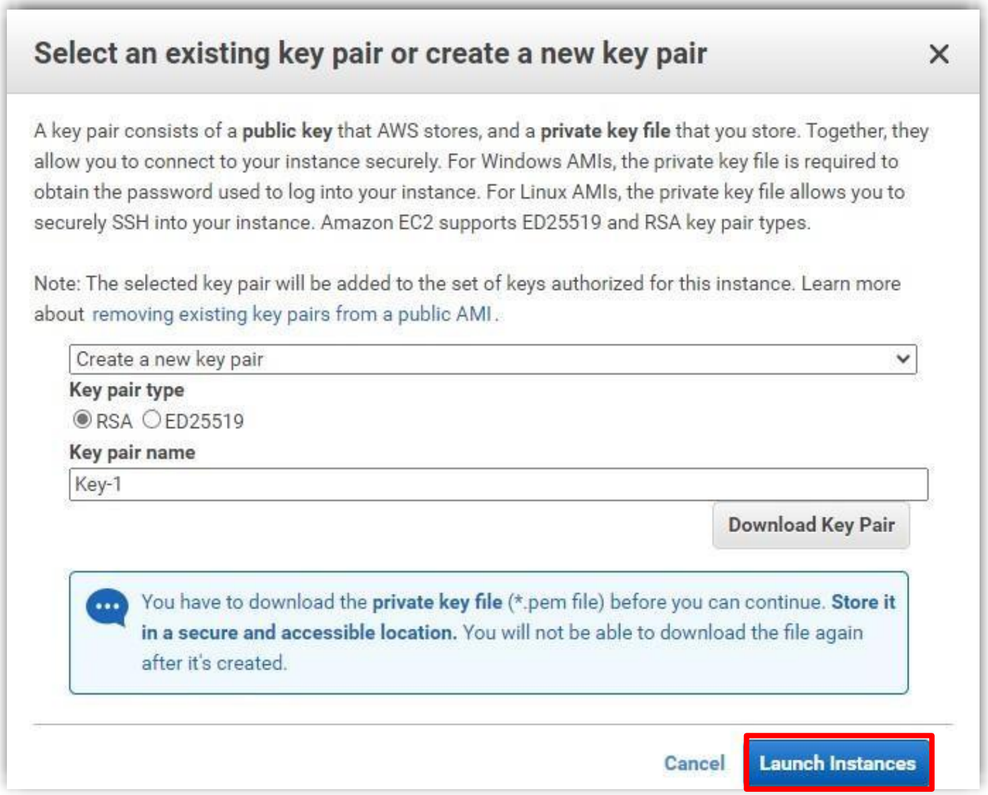
EC2

Step 11: Choose **Create a new key pair** then, choose Key pair Type as **RSA**. Enter the Key pair name and click on the **Download Key Pair**



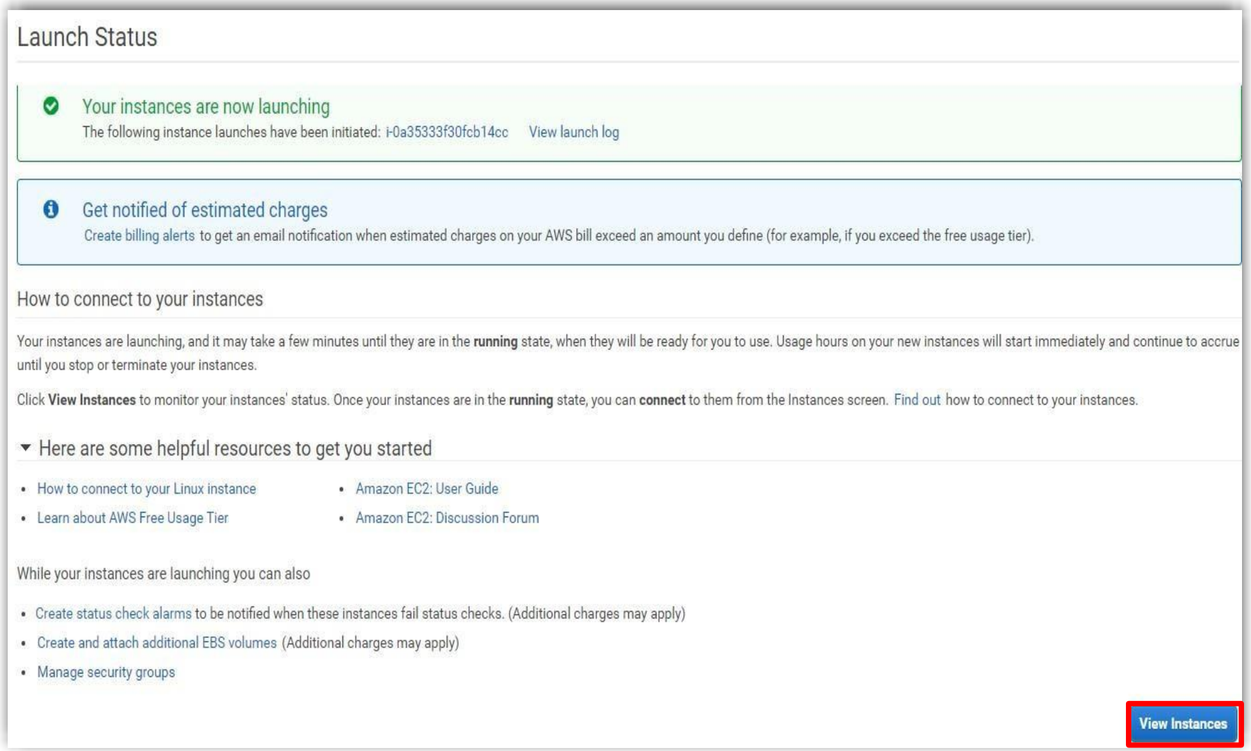
EC2

Step 12: Click **Launch Instances**



EC2

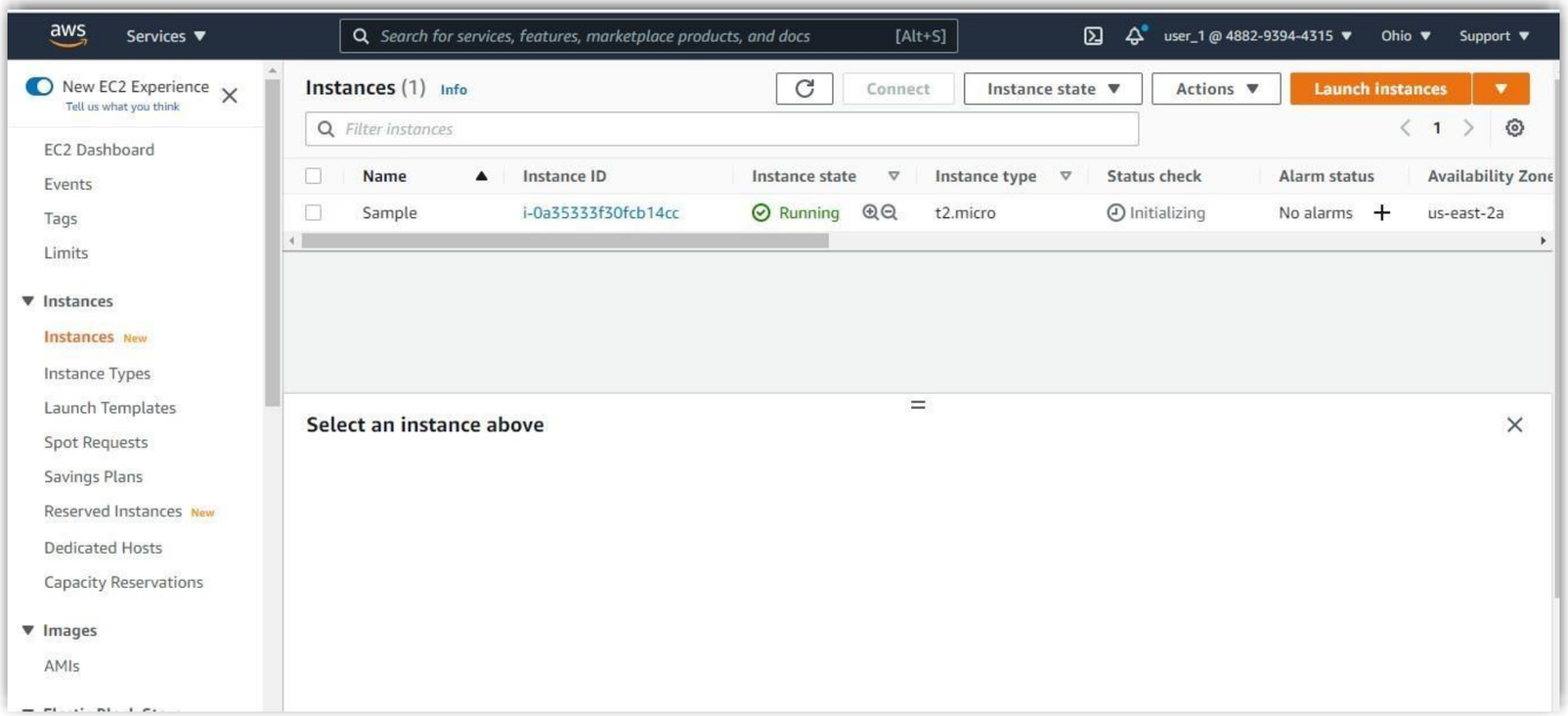
Step 13: Instance launched successfully. Click on the **View Instance**



EC2

(Continued)

- Instance made successfully



EC2

Storage Volumes

- You can create an Amazon EBS volume to attach to any EC2 instance in the same Availability Zone. You can only attach an encrypted EBS volume to supported instance types when you create it

- If you're creating a volume for high-performance storage, use a Provisioned IOPS SSD volume (io1 or io2) and attach it to an instance with enough bandwidth for supporting your application, like an EBS- optimised instance. For Throughput Optimised HDD (st1) and Cold HDD (sc1) volumes, the same advice applies

- Empty EBS volumes receive the maximum performance as soon as they're available and don't need to be initialised. However, before you can access storage blocks on volumes created from snapshots, they must be initialised

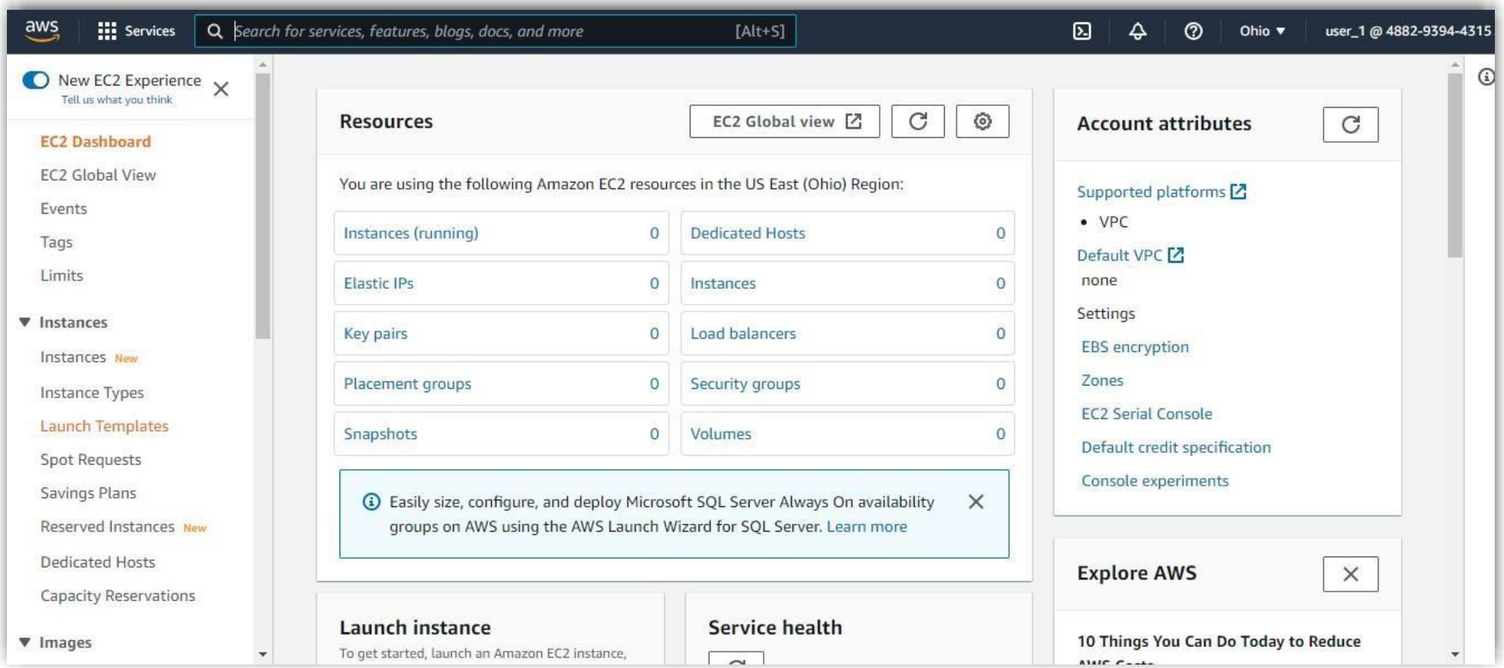
- This preliminary action takes time and can significantly increase the latency of an input or Output operation when each block is accessed for the first time. After all blocks have been downloaded and written to the volume, volume performance is achieved

EC2

(Continued)

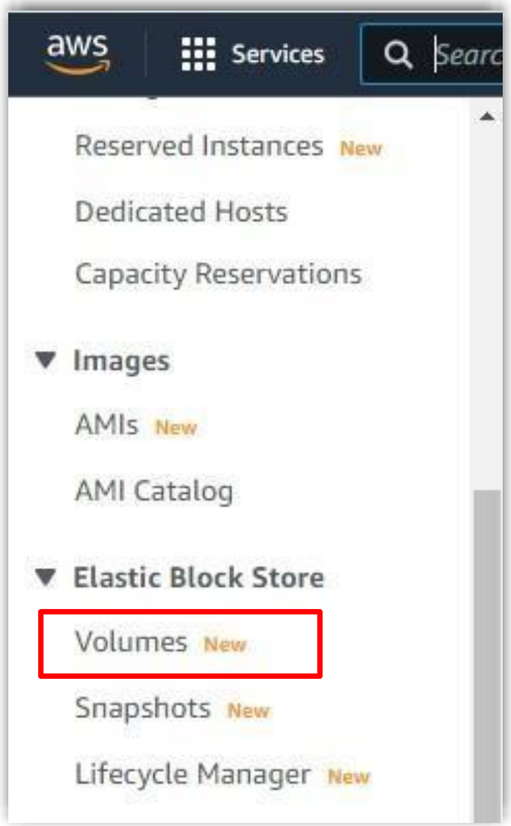
- Empty volumes receive their maximum performance as soon as that they're available and don't need to be initialised. The following are the steps to create an empty EBS volume using the console:

Step 1: Click on this link https://console.aws.amazon.com/ec2/ to open **Amazon EC2** console



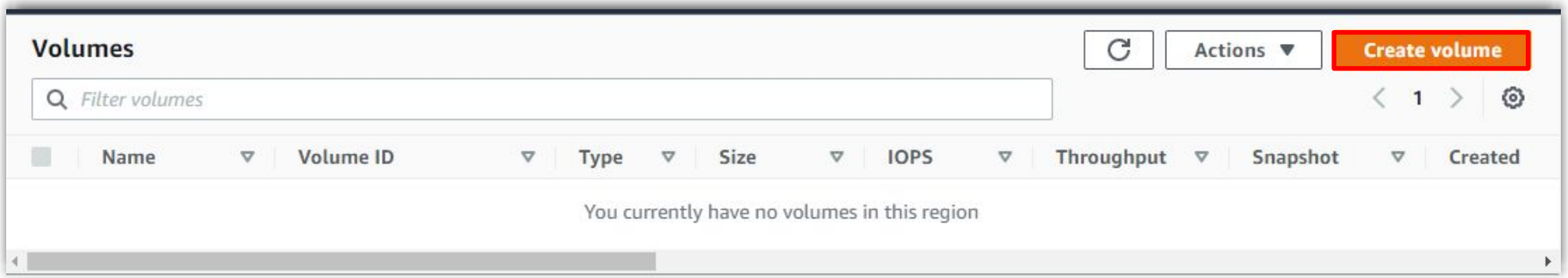
EC2

Step 2: Choose **Volumes** from the navigation pane



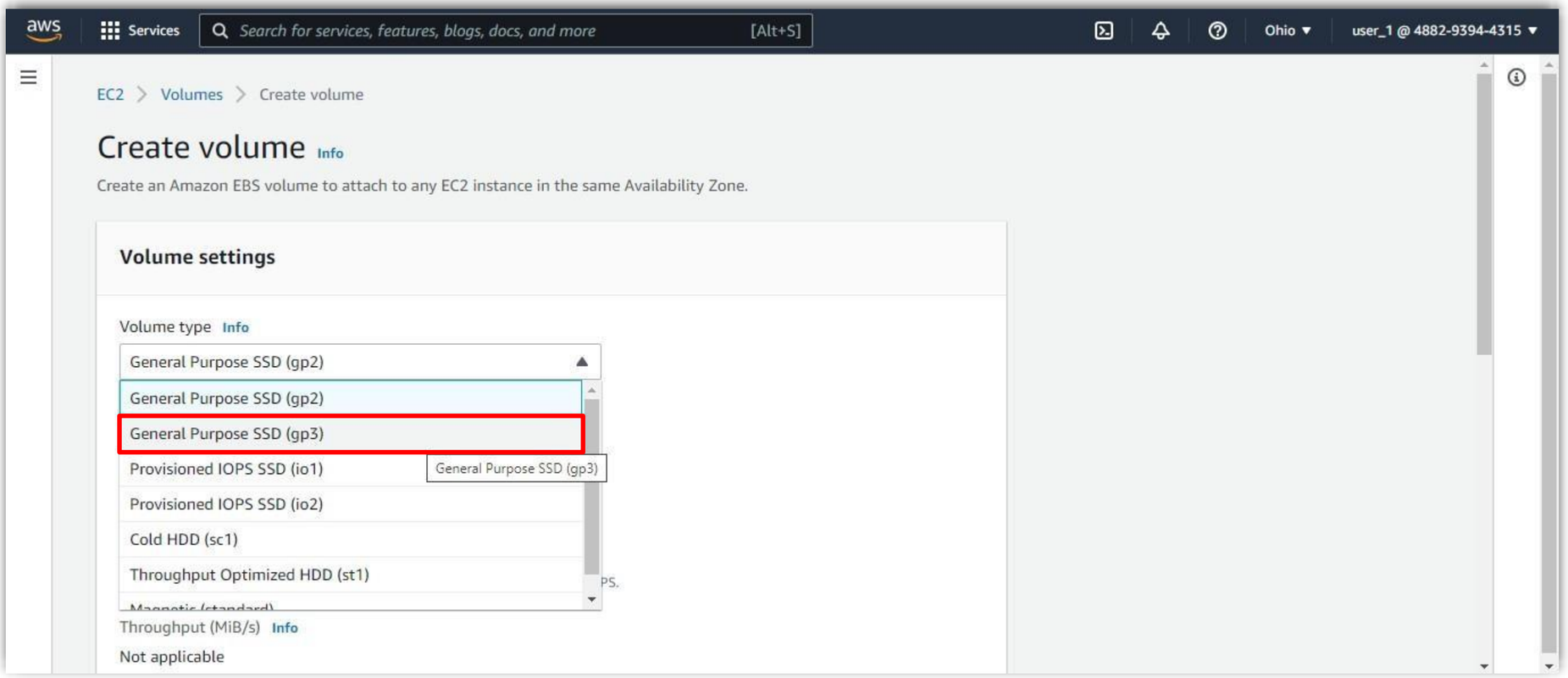
EC2

Step 3: Click on **Create volume**



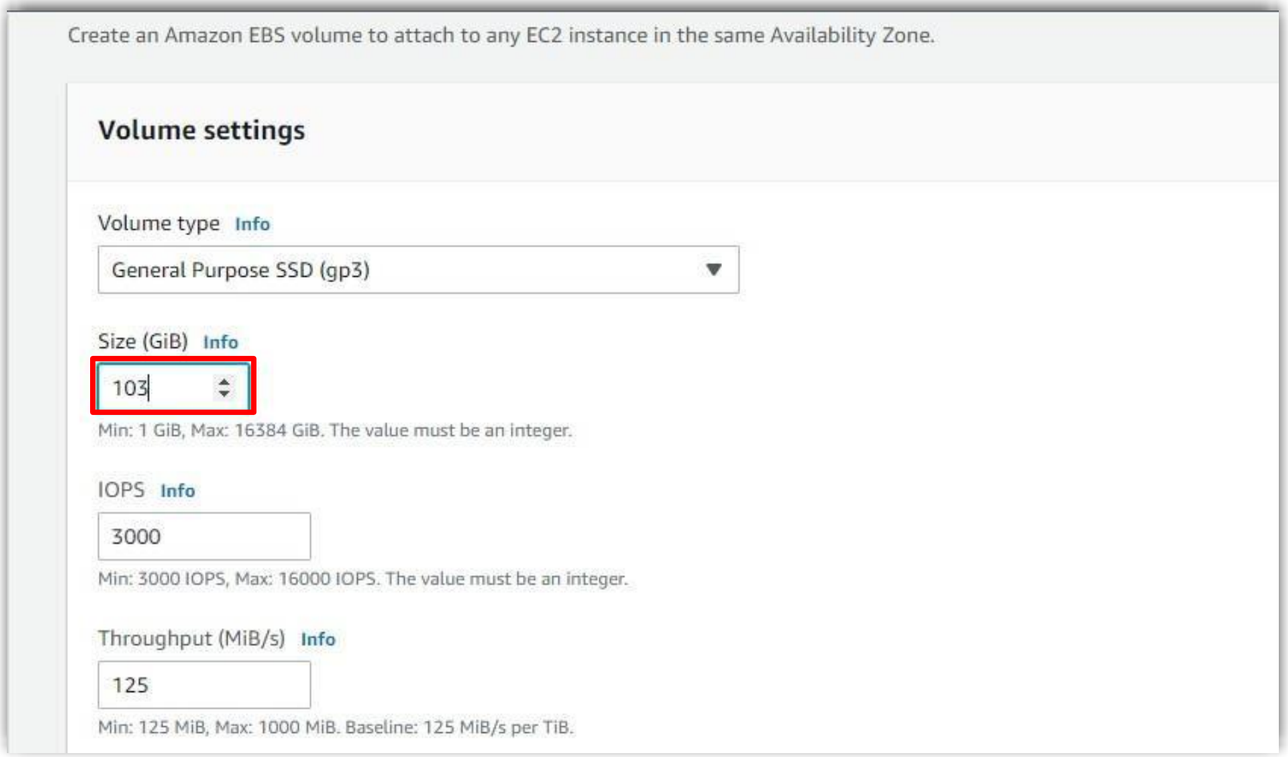
EC2

Step 4: Choose the volume type to create



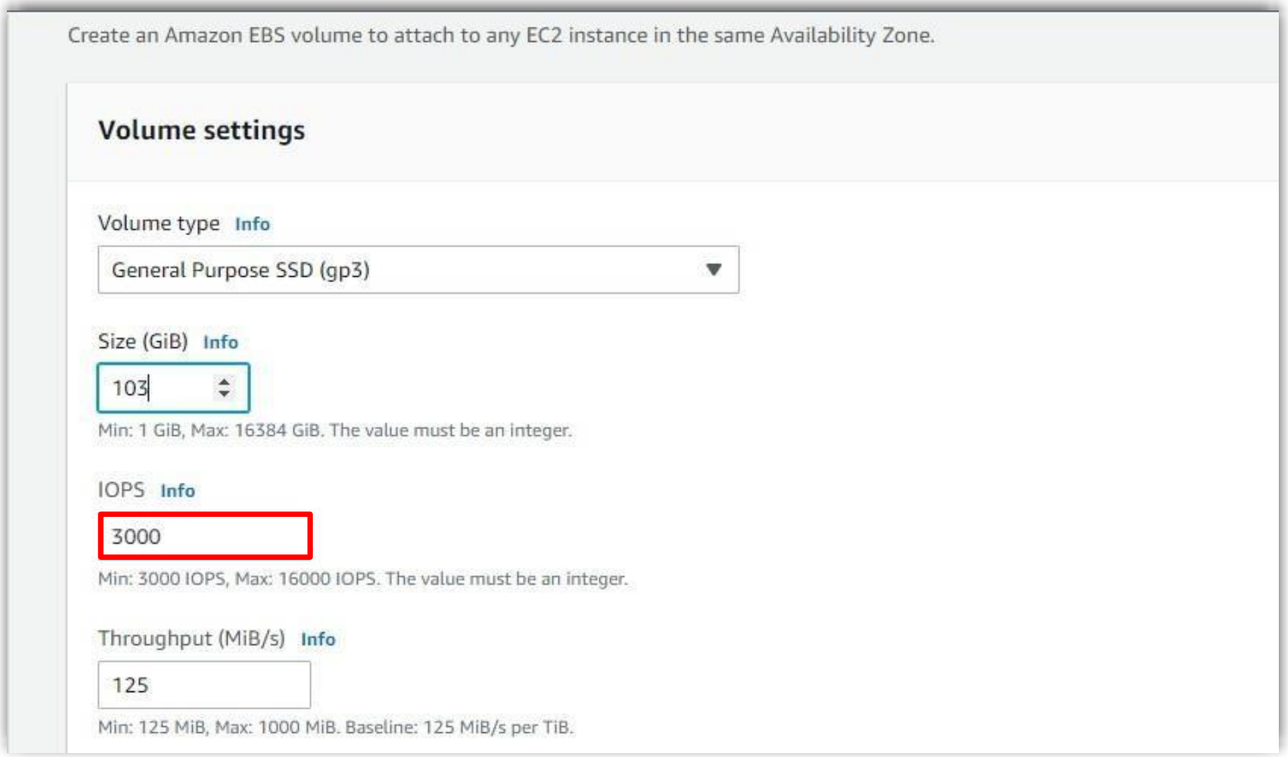
EC2

Step 5: Enter the volume's size in GiBs for Size



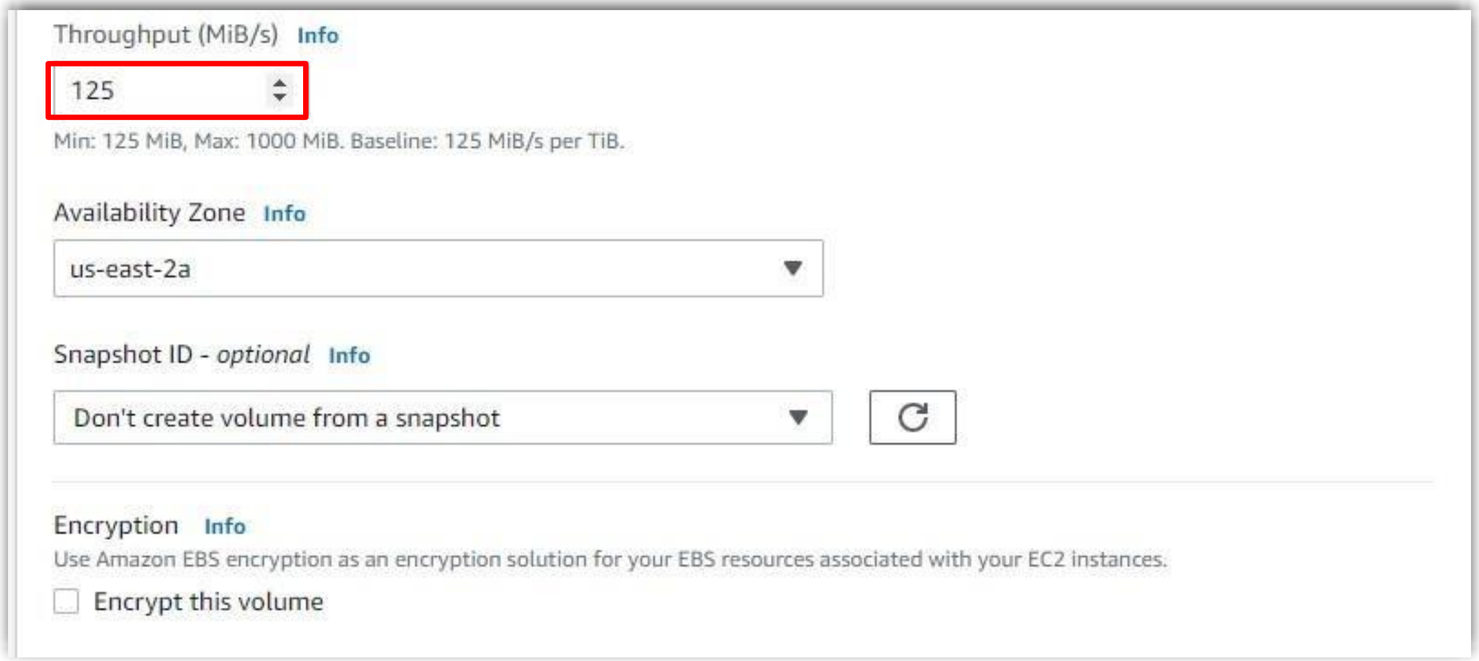
EC2

Step 6: Enter the volume's maximum number of input/output operations per second (IOPS)



EC2

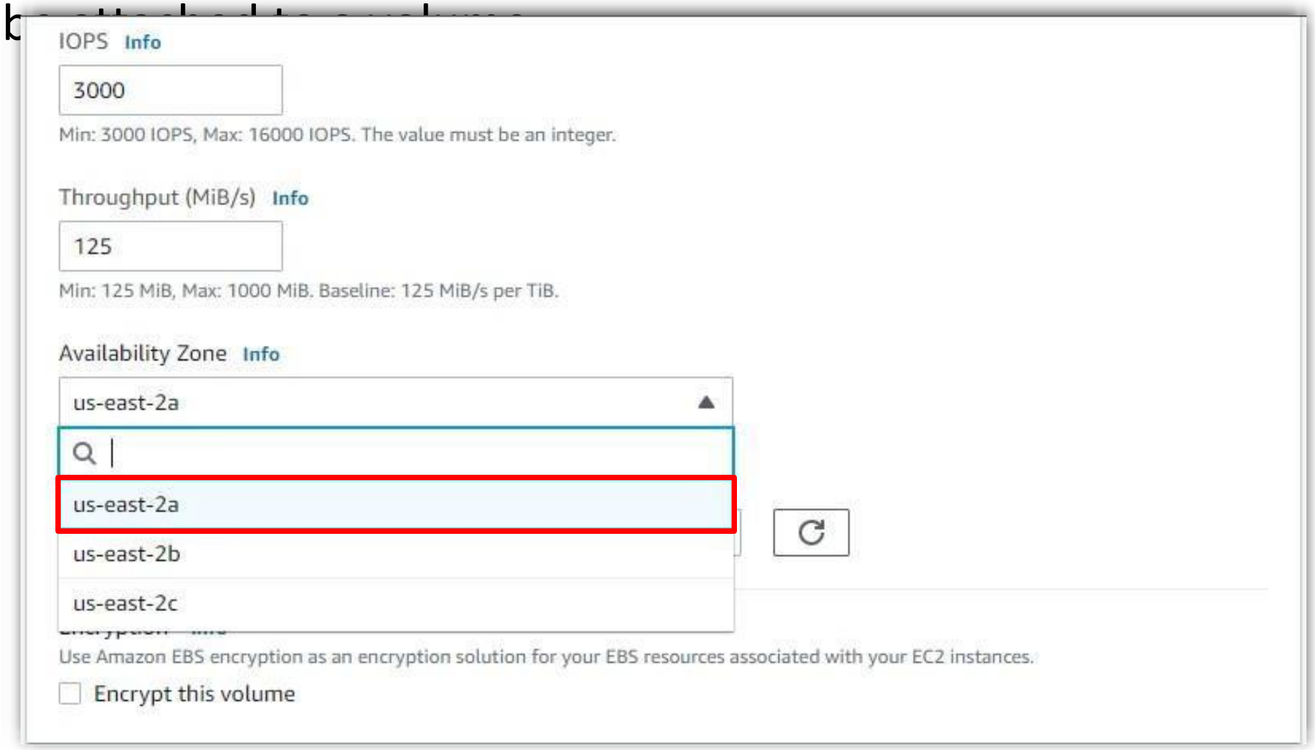
Step 7: In MiB/s, enter the throughput that the volume should give



EC2

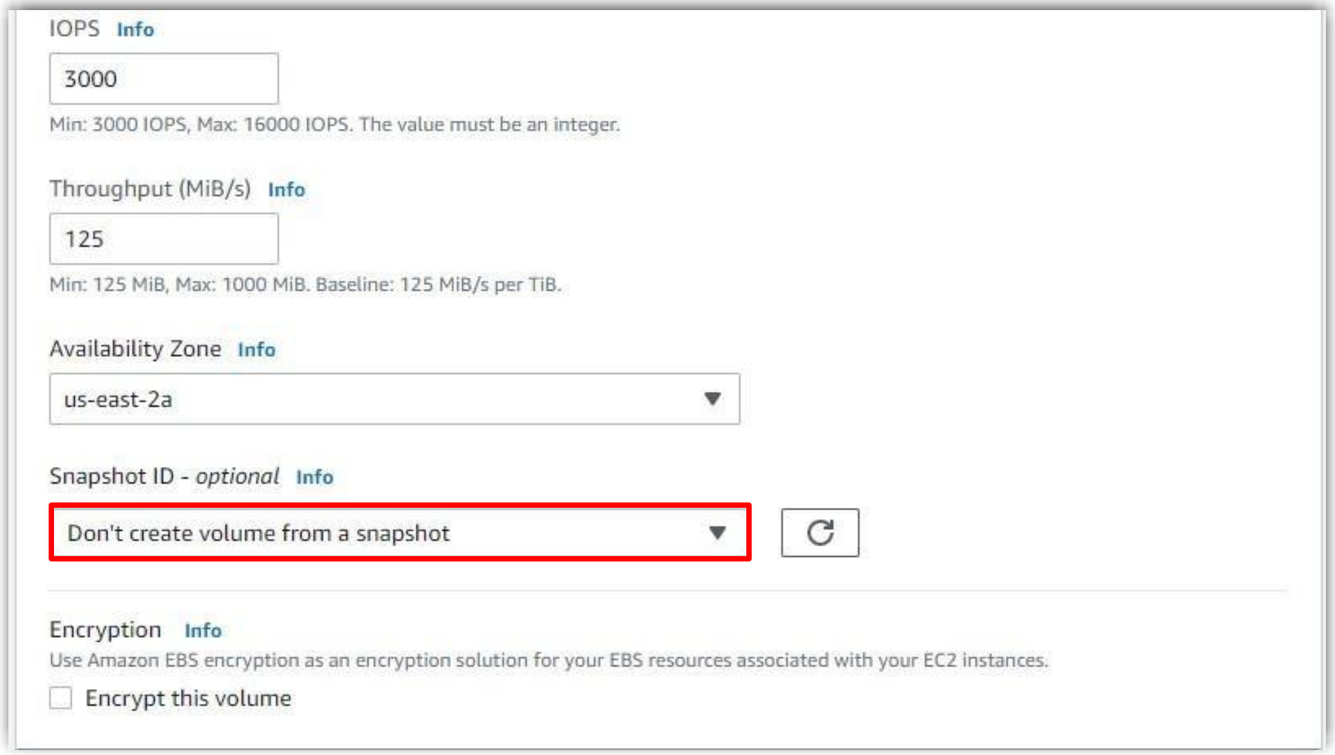
Step 8: Choose the **Availability Zone** where the volume will be created. Only instances in the same

Availability Zone can b



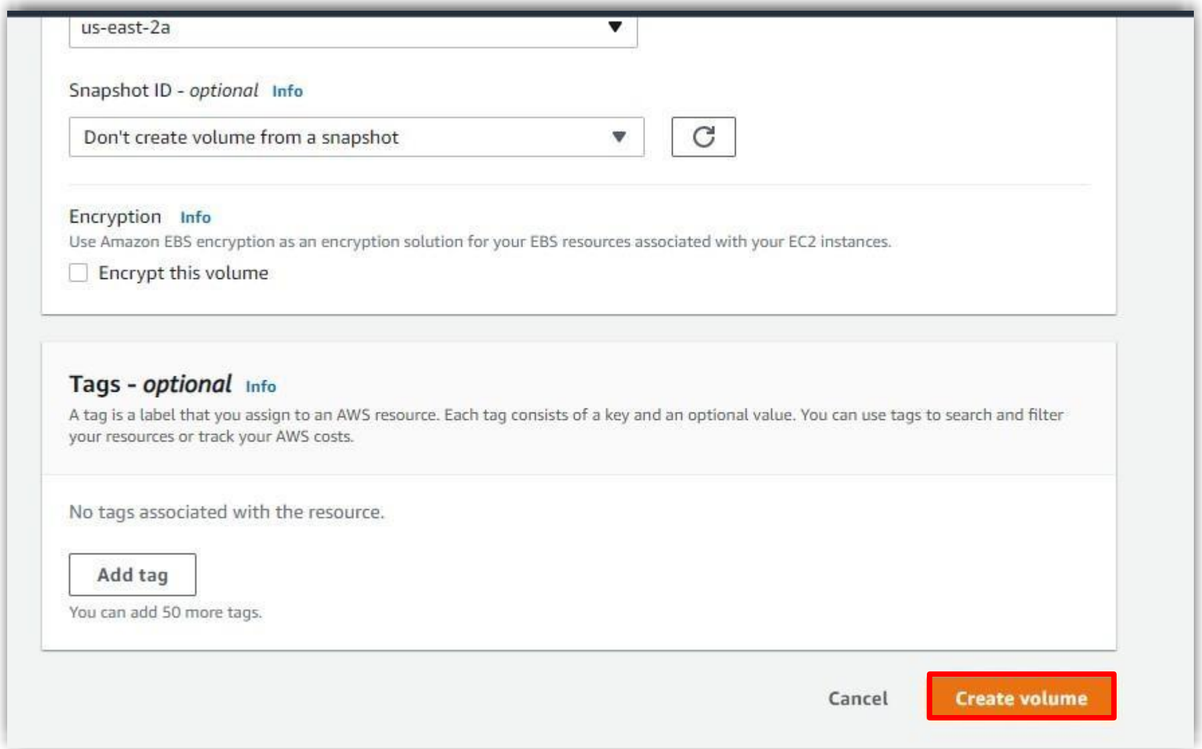
EC2

Step 9: For **Snapshot ID**, keep the default value



EC2

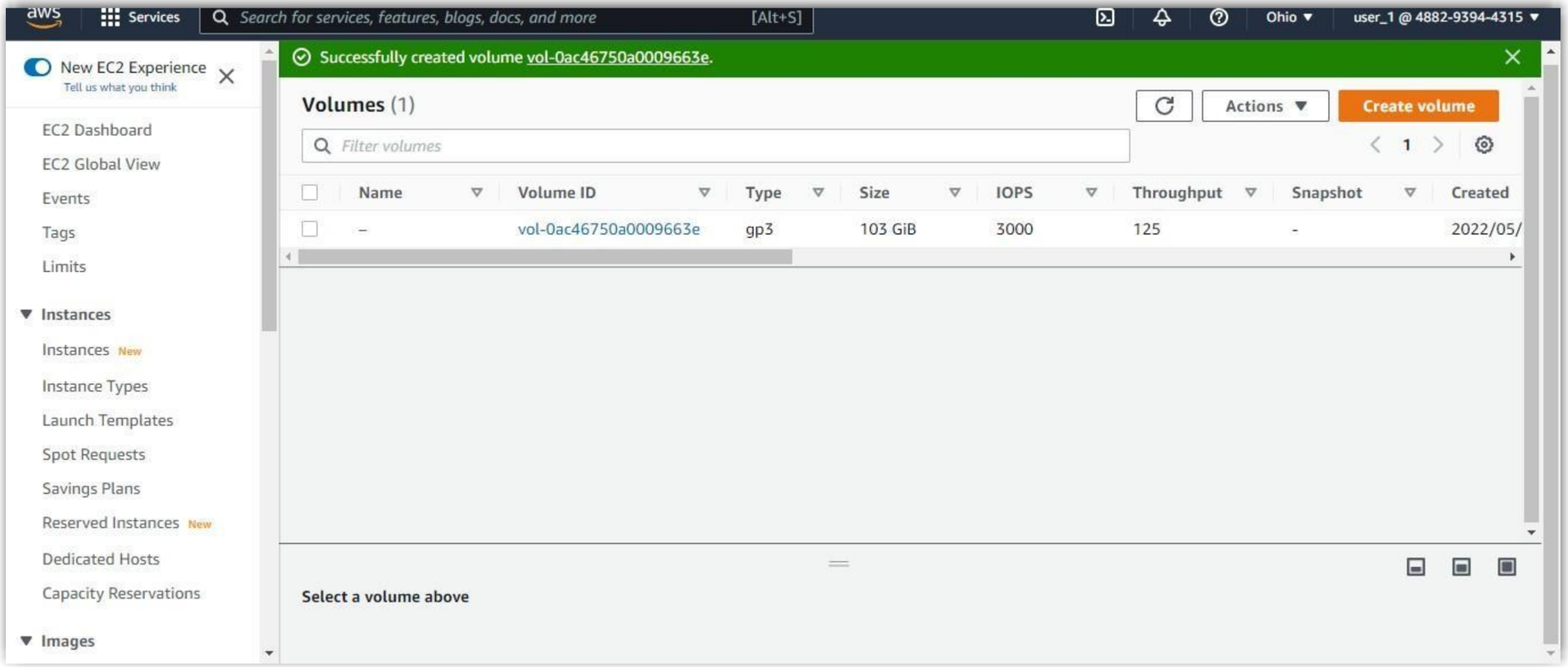
Step 10: Click on **Create volume**



EC2

(Continued)

- A volume has been successfully created:



EC2 Instance

Securing

Overview of Cloud Security

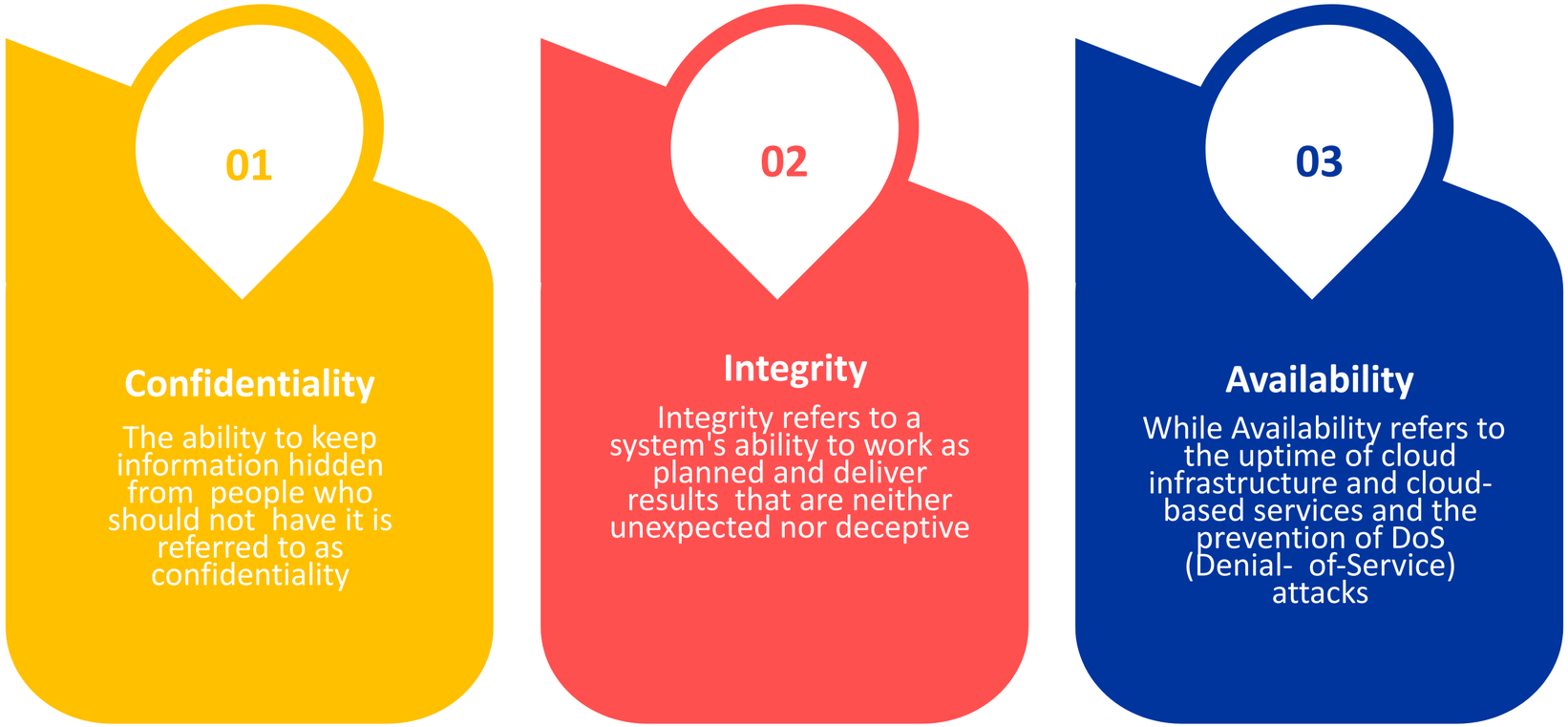
- The security of cloud resources is becoming increasingly important as more and more organisations migrate their on-premises resources to the cloud, either through the lift-n-shift strategy or a one-time migration process

- To protect your cloud data, many of the same policies, security controls, technologies, and processes that protect physical data centres, networks, and computing environments are delivered as a service

EC2 Instance

(Continued)

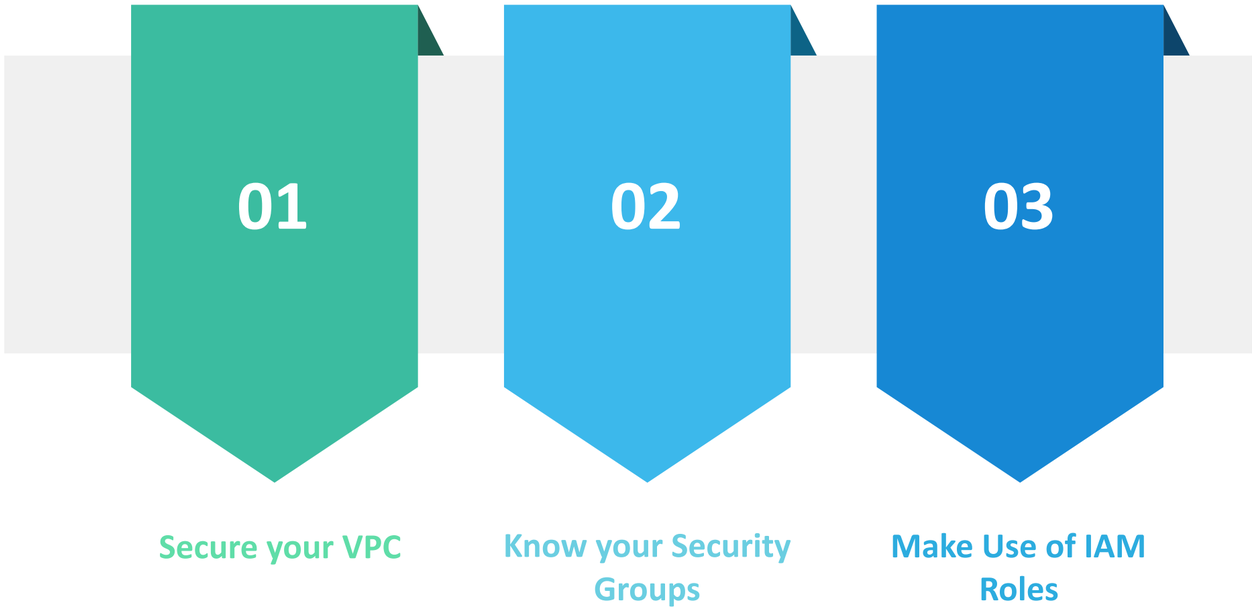
- Secure cloud services must have three qualities:



EC2 Instance

(Continued)

- The following are the four important techniques that help you secure your EC2 instances:



EC2 Instance

1. **Secure your VPC**

- Amazon VPC (Amazon Virtual Private Cloud) is an Amazon Web Services feature that enables users to create a logically isolated virtual network in which to provision your AWS resources

- A VPC is made up of IP addresses, subnets, network interfaces, route tables, gateways, endpoints, and other components

- VPC is the physical host for EC2 Instances, and if the physical host that runs your EC2 instance is compromised in any way, your instance may be compromised as well

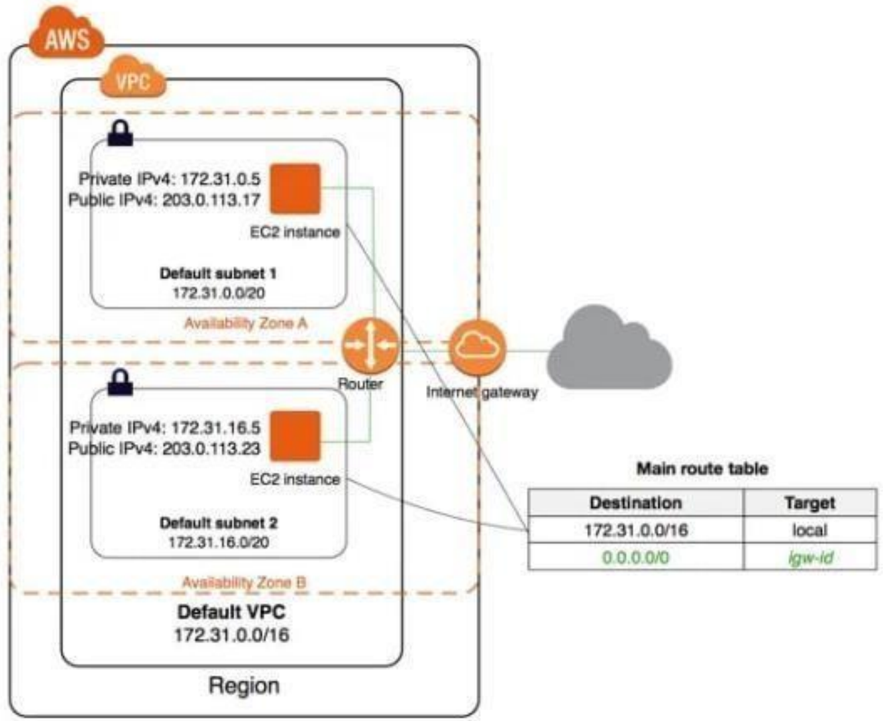
EC2 Instance

(Continued)

- Creating a new Virtual Private Cloud from scratch is one technique to ensure that your VPC is secure

- Although AWS offers a default Virtual Private Cloud in each region of your account, it is advisable to establish your own new VPCs rather to using the default VPC

- Because subnets in the de facto VPC are associated with your primary routing table, which does not set any limits on inbound or outbound traffic, the default VPC's security configuration is essentially open



EC2 Instance

(Continued)

- As a result, it's usually advisable to create new VPCs and then define custom route tables for each VPC when it comes to securing your AWS resources . This is because all other resources in your AWS account rely on it to communicate and exchange information

2. Know your Security Groups

- To control the inbound and outbound flow of network traffic in your environment, AWS provides two types of virtual firewalls. The two types of firewalls are NACLs (Network Access Control Lists) and security groups

- Security groups are necessary for securing communication with EC2 instances, while NACLs are important to secure communication with VPCs since they control access to subnets in your cloud environment

- You must first ensure that you have a clear understanding of how security groups work before you can secure your EC2 instances

EC2 Instance

(Continued)

- Amazon has a good description of how to use security groups for securing inbound traffic for Linux instances, and you can obtain a good understanding of how to use security groups to protect other types of instances by working through the scenario they present

- The fundamental principles are straightforward: assign one or more security groups to your instance, and add rules to all security groups to enable specific types of traffic to your instance. Always remember the cardinal rules of access control: **least privilege** and **least access**

3. Make Use of IAM Roles

- When you first set up your AWS environment, you'll be given security credentials that provide you access to all of your AWS resources, including EC2

- At your own risk, use these default AWS credentials to provide users, apps, or services access to your instances

EC2 Instance

(Continued)

- Instead, use AWS Identity and Access Management (IAM) to restrict access to your instances and other resources like storage for users, apps, and services
- IAM is an Amazon Web Services feature that enables you to create users and groups and assign them their own security credentials
- IAM can also be used to create JSON-format policies for using the EC2 APIs to perform various tasks on instances
- IAM enables you to create roles, which is especially useful for controlling instance security. IAM roles allow managing AWS credentials for applications that run on EC2 instances
- This is important because API calls made by applications must be signed with valid AWS credentials. Create an IAM role and assign it to your EC2 instance in most cases

EC2 Instance

(Continued)

- The required permissions for the IAM role are determined by an IAM policy you created to provide secure access to another AWS resource, such as an S3 bucket
- After that, the instance is launched in EC2 and the IAM role creates temporary credentials for the instance to access the bucket
- The advantage of using IAM roles in this manner is that temporary credentials are used instead of root credentials to access the bucket, which is more secure because root credentials are not revealed

Module 3: Amazon S3 and Amazon Glacier Storage

S3

- Amazon S3, or Simple Storage Service, is a storage service provided for the internet
- Using Amazon S3, users can store and retrieve any amount of data irrespective of time from any where on the Internet
- The developers are provisioned with the highly scalable, reliable, fast, and inexpensive data storage infrastructure
- S3 stores data as objects in resources called buckets. Downloading of uploading of data can take place from S3 using the AWS service APSs
- A user can perform various operations on buckets. These operations include storing objects, deleting objects, and performing read and write operations. All these objects can be accessed through a web browser

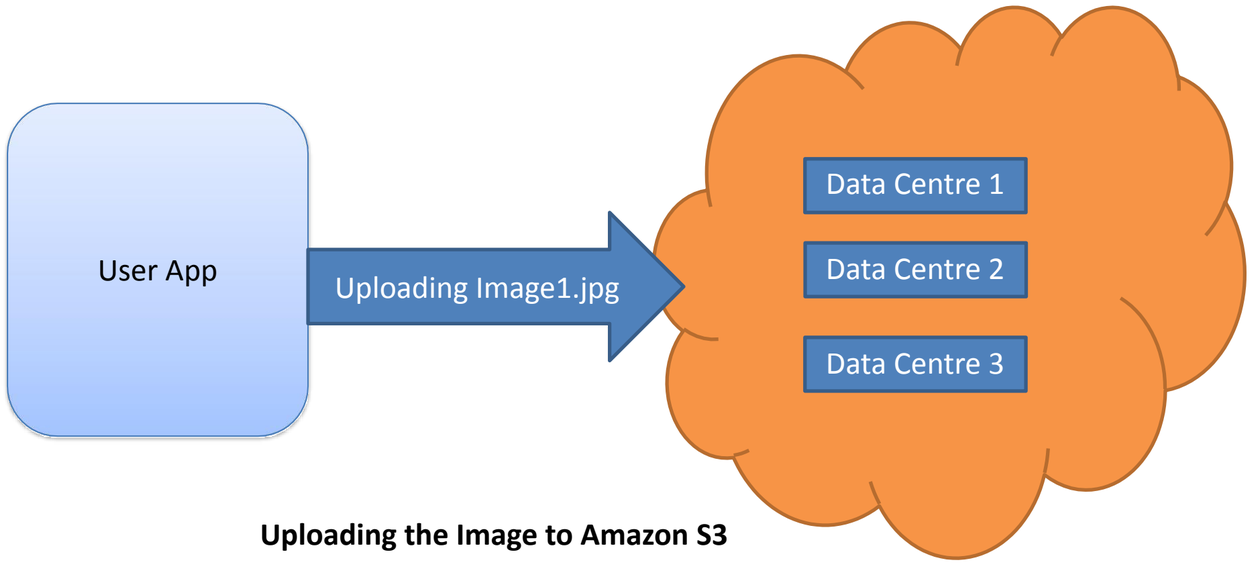
S3

(Continued)

- S3 is a virtually infinite storage resource for web applications. Amazon S3 can be used without the user having to sign any contracts

- As per the pricing policy, the user will have to pay for only the services used. S3 has the capability to store objects as large as 5 Terabytes

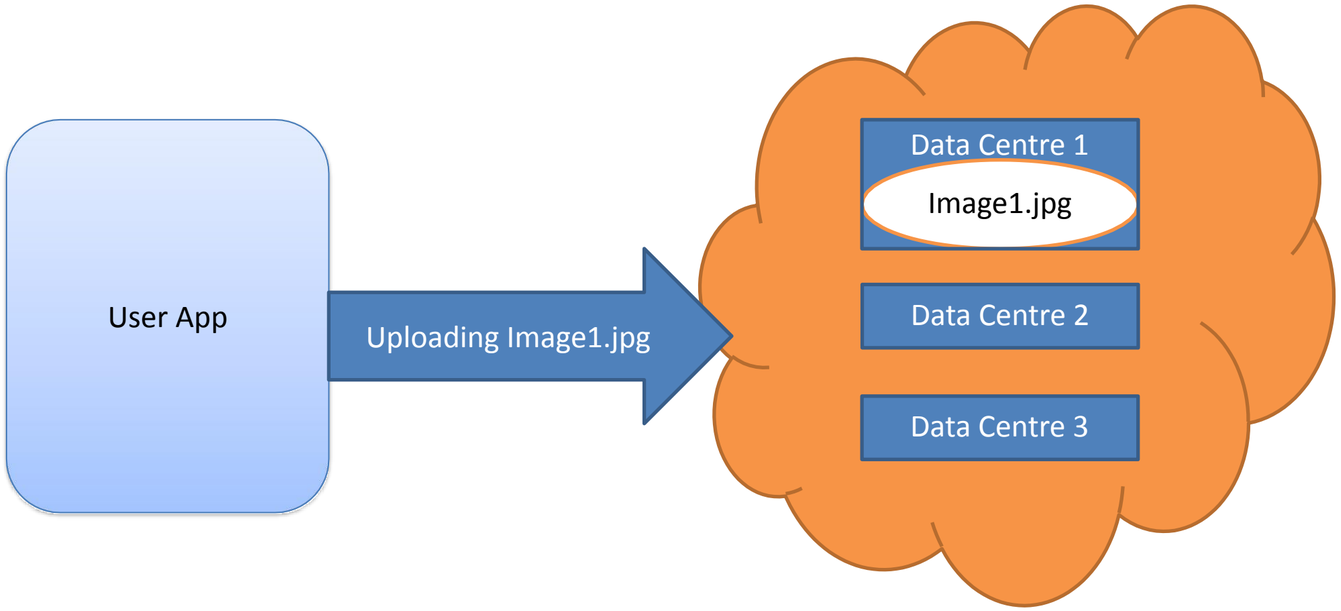
Amazon S3 Operations



Uploading the Image to Amazon S3

S3

(Continued)



Data Centre 1 Processing the Image

S3

(Continued)

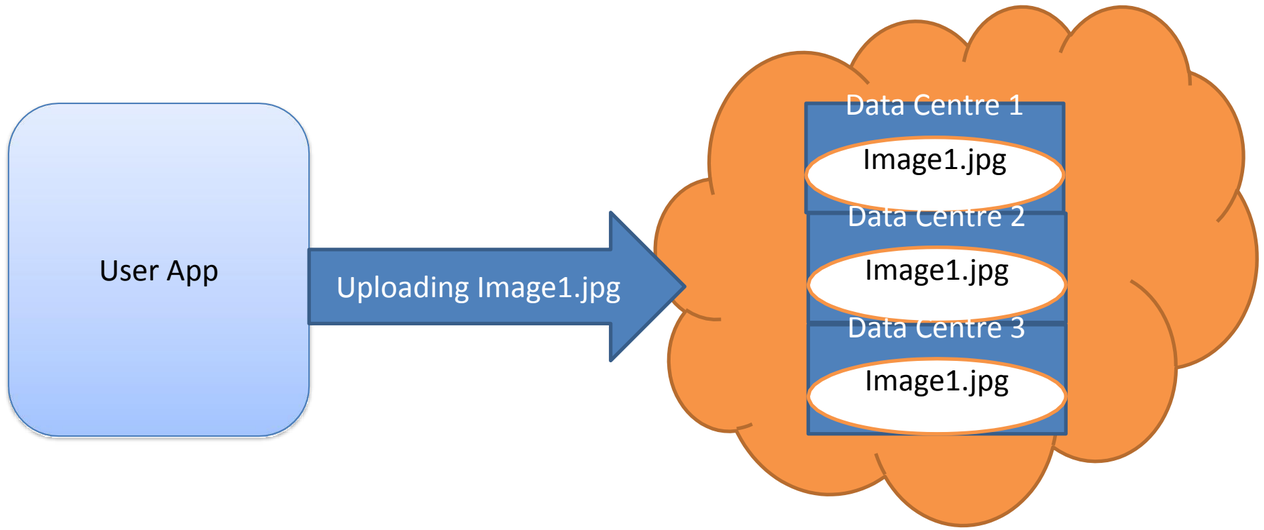
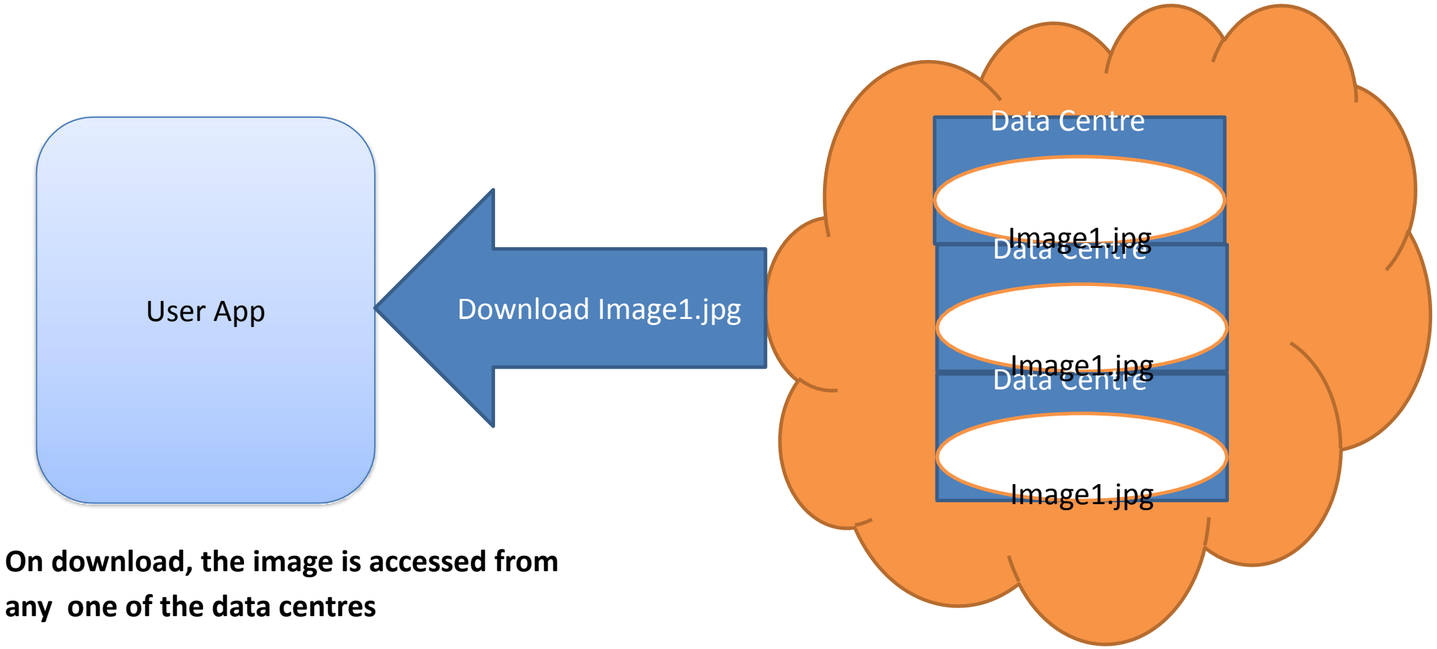


Image is replicated to various data centres

S3

(Continued)

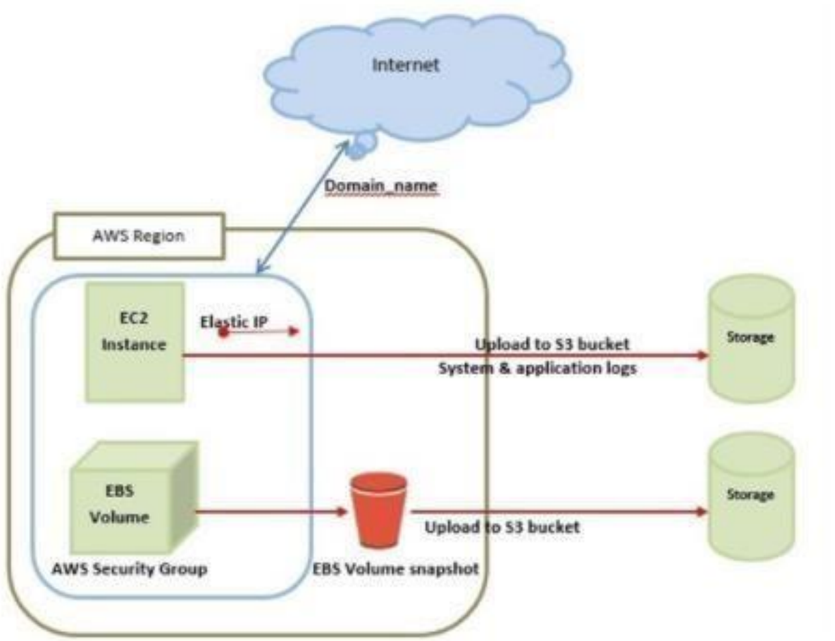


On download, the image is accessed from any one of the data centres

S3

Service Architecture

- The following diagram is the basic structure of AWS Architecture which provides cloud computing services accordingly



S3

(Continued)

- It is considered the fundamental structure of AWS architecture or AWS EC2.EC2 stands for Elastic Compute Cloud, and it allows clients or users to use different configurations in their project or method depending on their needs

- Moreover, there are different amazing options like pricing options, configuration servers, individual server mapping, etc. S3 which is present in the AWS architecture is named as Simple Storage Services

- By using this S3, users can easily recover or else store data via different data types using Application Programming Interface calls. No computing element for the services as well

S3

Durability and Availability

- **Availability** measures how readily available service is, e.g. you may have a hairdresser or barber that you like to visit. When the provider is available then only you can use these services
- Adding more barbers (or hairdressers) that you can use will improve the availability level
- It's improbable that an issue will affect all of our barbers at the same time if we have numerous barbers ready to go in different places
- Generally, availability is measured as a percentage, e.g. the service level agreement for S3 is that it will be available 99.99% of the time

S3

(Continued)

- **Durability** is used to measure the likelihood of data loss, e.g. suppose you have an important document in your safe at home
- If you make a copy of it as well as store it in a safe deposit box at the bank, you have just enhanced the durability of that document. It is much less likely that all copies will be destroyed at the same time. AWS measures durability as a percentage
- For example, the S3 Standard Tier is designed for 99.999999999% durability. It indicates that if you store 100 billion objects in S3, you will lose one object
- By spreading data and service across regions as well as availability zones, you can improve reliability and durability

S3

Object Lifecycle

- Define S3 Lifecycle configuration rules for objects that have a well-defined lifecycle. For example:
  - If you upload periodic logs to a bucket, your application might require them for a week or a month. After that, you might want to delete them
  - Some documents are often accessed for a limited period. After that, they are infrequently accessed. At some point, you might not require real-time access to them, but your organisation or regulations might need you to archive them for a particular period. After that, you can delete them
  - You might upload some types of data to Amazon S3 primarily for archival purposes, e.g. you might archive digital media, financial as well as healthcare records, raw genomic sequence data, long- term database backups, and data that need to be retained for regulatory compliance
- You can tell Amazon S3 to transition objects to less-expensive storage classes, or archive or delete them with S3 Lifecycle configuration rules

Amazon Glacier

- For data preservation and long-term backup, Amazon S3 Glacier is a safe, durable, and exceptionally low-cost Amazon S3 storage class
- Using S3 Glacier, consumer can store their data cost effectively for long time such as for months, years, or even decades
- S3 Glacier allows the consumers for offloading the administrative burdens to operate and scale the storage to AWS
- They do not need to worry for the capacity planning, data replication, hardware provisioning, hardware failure recovery and detection, or time-consuming hardware migrations
- S3 Glacier is one of the many other storage classes for the Amazon S3. For Amazon S3 core concepts including access points , buckets, storage classes and objects

Amazon Glacier

S3 Glacier User

1. **What is Amazon S3 Glacier**: The Amazon SDKs that you can use for interacting with the service

2. **Getting Started:** The Getting Started with Amazon S3 Glacier leads you via the process to create a vault, upload archives, create jobs for downloading archives, to retrieve the job output, and delete the archives

- Amazon S3 is supporting the lifecycle configuration on the S3 bucket, which allows you for transitioning the objects to the S3 Glacier storage class for the archival

- When you are transitioning the Amazon S3 objects to the S3 Glacier storage class, Internally, the Amazon S3 utilises the S3 Glacier for durable storage at a lower cost

- However, the objects are stored in S3 Glacier, they remain Amazon S3 objects that you are managing in the Amazon S3, and you can not able to access them directly via S3 Glacier

Other Storage-Related Services

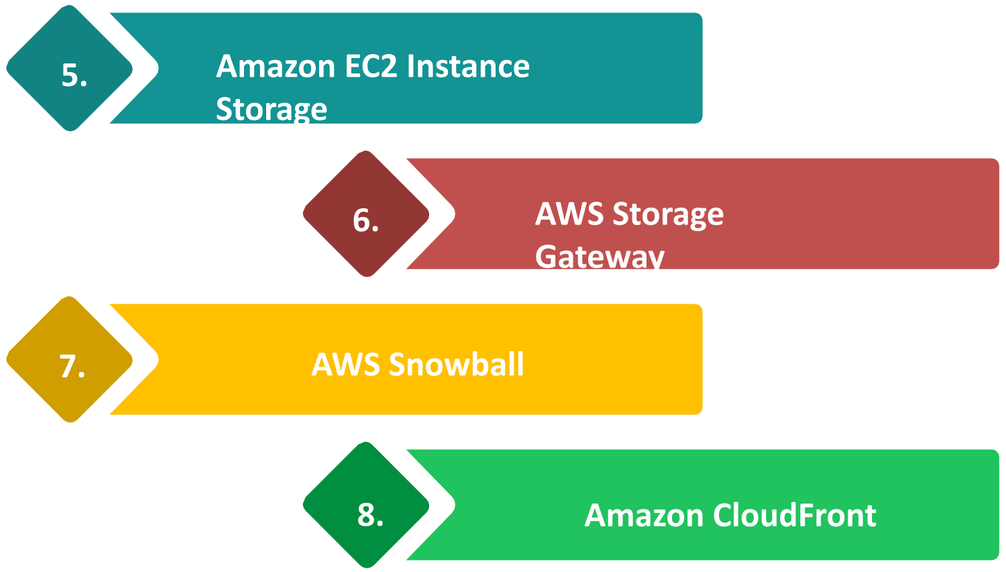
AWS Storage Services

- Amazon Web Services offers low-cost data storage with a high availability and durability. It provides storage choices to back up the information, disaster recovery, and archiving

- The following are the list of the main storage services available on the AWS Cloud:

Other Storage-Related Services

(Continued)



Module 4: Amazon Virtual Private Cloud

Introduction

- The Amazon Virtual Private Cloud (Amazon VPC) allows you to deploy AWS resources into a defined virtual network

- This virtual network closely matches a typical network you'd run in your own data centre, with the benefit of AWS' scalable infrastructure

Access Amazon VPC

- You can use any of the following Interfaces to create, access, and manage your VPCs:

1. **Amazon Web Services Management Console:** Provides a web interface through which you can access your virtual private clouds

2. **Amazon Command Line Interface (Amazon CLI):** It supports Windows, Mac, and Linux and provides commands for a wide range of Amazon services, including Amazon VPC

Introduction

3. **Amazon SDKs:** It supports Windows, Mac, and Linux and provides commands for a wide range of Amazon services, including Amazon VPC

4. **Query API:** HTTPS requests are used to call low-level API activities. The Query API is the most direct approach to accaess Amazon VPC, but it demands that your application handle low-level features like hash generation and error handling

Subnets

- AWS VPC is a logical network isolation for your EC2 instances. Placing them in a subnet can isolate instances with similar properties. All the EC2 instances and RDS instances can be logically isolated using VPC

Subnet:

- You can divide your VPC network into logical sub-networks. These are called subnets. A subnet is a generic networking concept. In AWS, you can create a public or a private subnet within a VPC

- A public subnet allows you to connect to the internet whereas a private subnet does not allow you to connect to the internet. However, you can configure subnets to allow inbound or outbound traffic for the instances

- In VPC, you can assign an IP address to an EC2 instance that will identify it in a subnet uniquely

VPC CIDR Blocks

- CIDR refers to as classless Inter-Domain Routing, CIDR Notation is a technique that allocates IP addresses and routes Internet Protocol packets. CIDR notation is the syntax for specifying IP addresses and their associated routing prefix

- It appends the IP address with a slash and a decimal number. The IP address may denote a distinct subnet address, or it may denote the starting address of a network. The decimal number denotes the maximum size of the network

- For example: For IP range 10.10.0.0/24, 10. 10.0 denotes the network part, which cannot be changed. Therefore, you can create an IP address in the range of 10.10.0.0 to 10.10.0255

Elastic Network Interface

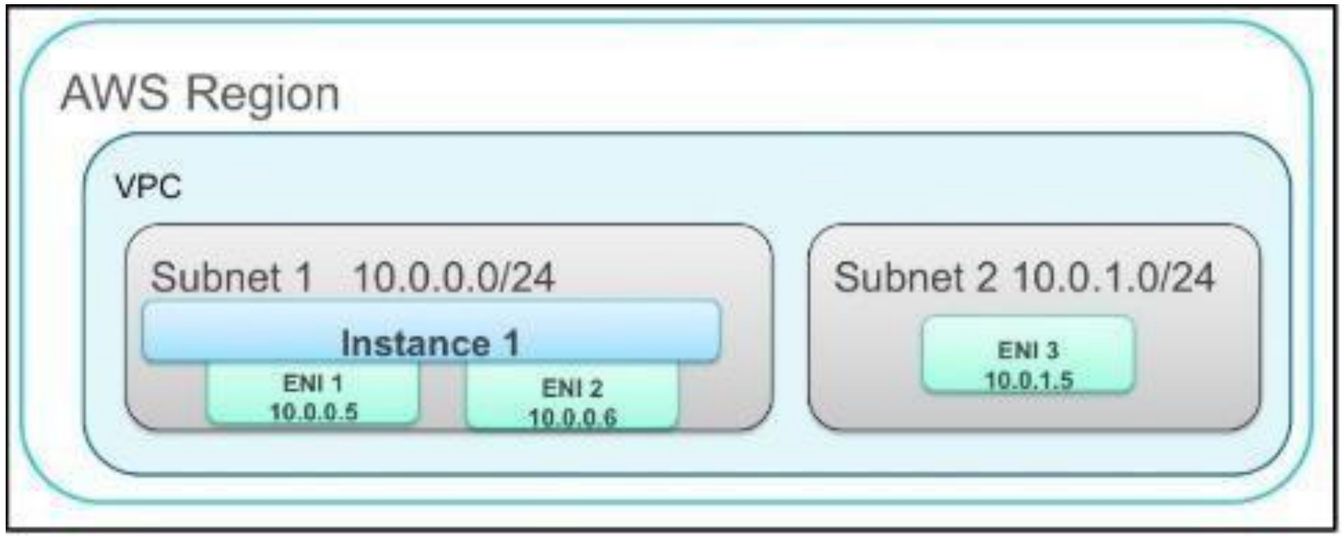
- Each instance in the VPC has a network interface, much like a NIC card in a physical machine
- The default network interface is assigned a private IP address from the IP address range assigned to your VPC
- More than one network interface can be added to the instance. This is called an Elastic Network Interface or ENI
- The number of ENI that you can attach to an instance depends on the instance type

Elastic Network Interface

- An ENI can have:
  - A primary private IP address
  - Or more secondary IP addresses
  - 1 Elastic IP address
  - A MAC address
  - 1 or more security groups
  - A source and destination check flag
  - A description

Elastic Network Interface

- Example of an EC2 instance with two Elastic Network Interface (ENI):



Internet Gateway

- The component of VPC that lets the instance in the VPC to talk to the internet. It is horizontally scaled, redundant, and highly scalable

- There are two purposes served by an Internet Gateway:

  - Providing a target in your VPC route tables for internet - routable traffic

  - Performing Network Address Translation (NAT) for instances that have been assigned public IPv4 addresses

Configuring Routes

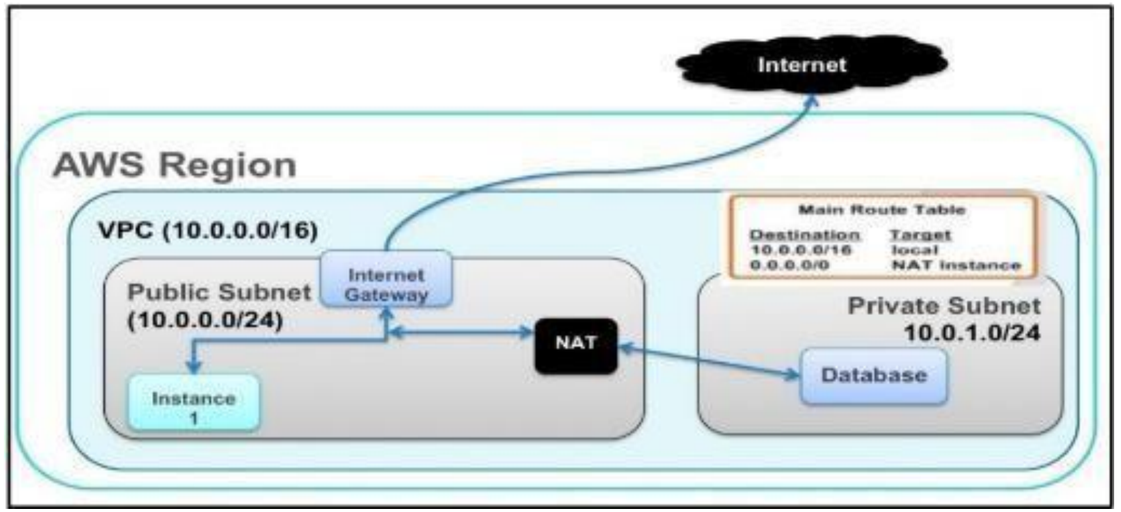
- The route table contains rules that specify where the traffic should be routed to. These rules are called routes. A VPC has a routing table associated with it

Internet Gateway

- The following are the basic points to remember about a routing table:
  - A VPC has an implicit router
  - Every subnet is associated with a routing table. If it is not associated with any routing table, then it is associated with a default route table of the VPC
  - The main route table can be replaced with the custom route table
  - Each route table specifies a destination CIDR and a target

Internet Gateway

- Sample route table for the Private Subnet:



Route Tables

- A route table is a set of rules that determines where network traffic from your subnet or gateway is directed

Route table concepts

- The basic concepts for route tables are as follows:

Main route table: The route table that comes with your VPC by default. It is responsible for all subnets that are not expressly linked to another route table

Custom route table: A route table for your VPC that you design

Destination: The IP address range to which traffic should be routed (destination CIDR). For instance, a corporate network with the CIDR 172.16.0.0/12

Target: An internet gateway, for example, is a gateway, network interface, or connection that sends destination traffic

Route Tables

Route table association: A route table and a subnet, internet gateway, or virtual private gateway are linked

Subnet route table: A route table that is linked to a subnet

Local route: A default communication route within the VPC

Propagation: A virtual private gateway can propagate routes to the route tables automatically using route propagation

Gateway route table: A route table linked to an internet gateway or a virtual private gateway

Transit gateway route table: A route table linked to a transport gateway

Local gateway route table: A route table connected to an Outposts local gateway

Security Groups

Security in VPC

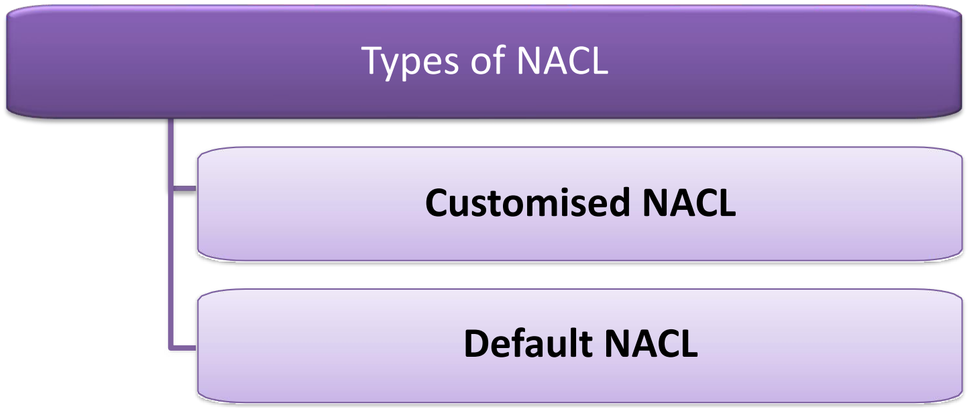
- The Virtual Private Cloud provides its users with various levels of security
- The VPC is protected by Security Groups, NACLs(Network Access Control Lists) and flow logs
- The Security Groups acting as firewalls control both inbound and outbound traffic to/from the internet
- NACL’s act on each subnet to check whether access to a subnet is authorised or not
- Flow logs are used for storing information about IP traffic that is going out or coming into the VPC through the network interfaces

Network Access Control Lists

- NACL stands for Network Access Control List, which provides a security layer to the Amazon Web Services architecture
- NACL acts as a firewall, ensuring the security of VPCs and subnets
- It contributes to the creation of a security layer that regulates and efficiently handles traffic within the subnets
- It is an optional VPC layer that adds another layer of protection to the Amazon service
- Virtual private Cloud (VPC) is a container that holds subnets
- Subnets act as data storage containers

Network Access Control Lists

Types of NACl



Customised NACL: It is also known as a user-defined NACL, and its primary function is to block all incoming and outgoing traffic until a rule to manage the traffic is added

Default NACL: Customised NACL, on the other hand, permits all traffic to flow in and out of the network. It also comes with a specific rule that is assigned a number and cannot be changed or removed. When a request does not match its associated rule, it refuses access

Elastic IP Addresses

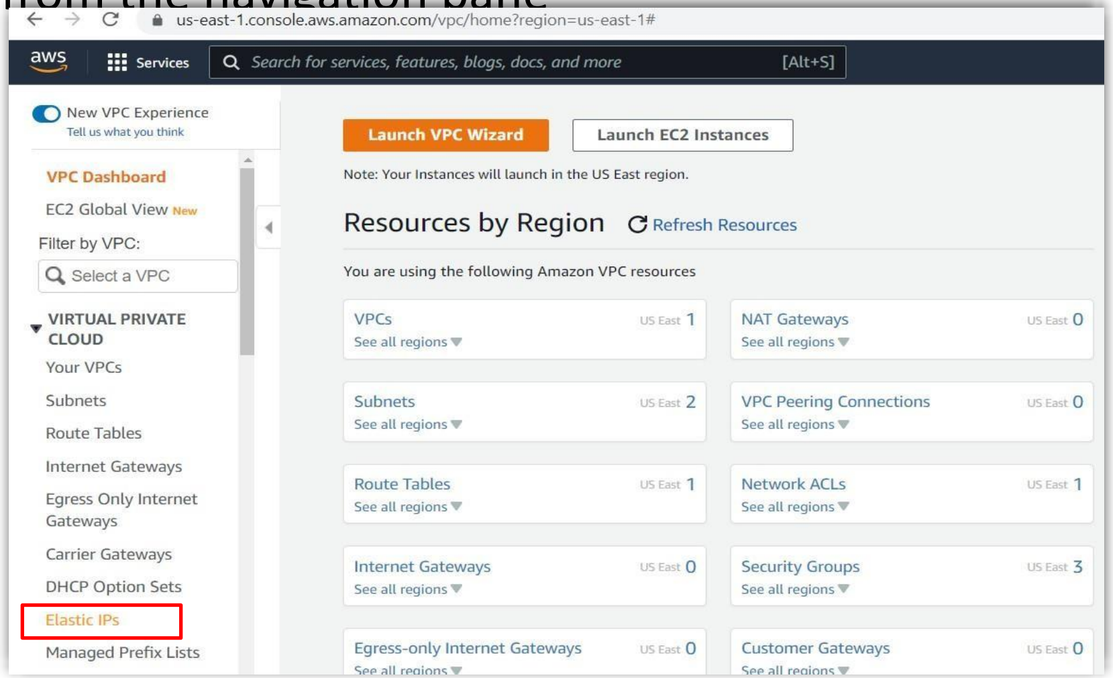
- An Elastic IP address is a public IPv4 address that is static and designed for cloud computing
- Any instance or network interface in any VPC in your account can be assigned an Elastic IP address
- You can use an Elastic IP address to mask an instance's failure by immediately remapping the address to another instance in your VPC
- An Elastic IP address is a reserved public IP address that you can assign to any EC2 instance in a specific area until you release it
- The default public IP address is replaced when you associate an Elastic IP address with an EC2 instance
- It will replace this hostname if an external hostname was assigned to the instance via your launch settings; otherwise, it will create one for the instance

Elastic IP Addresses

Allocate an Elastic IP Address

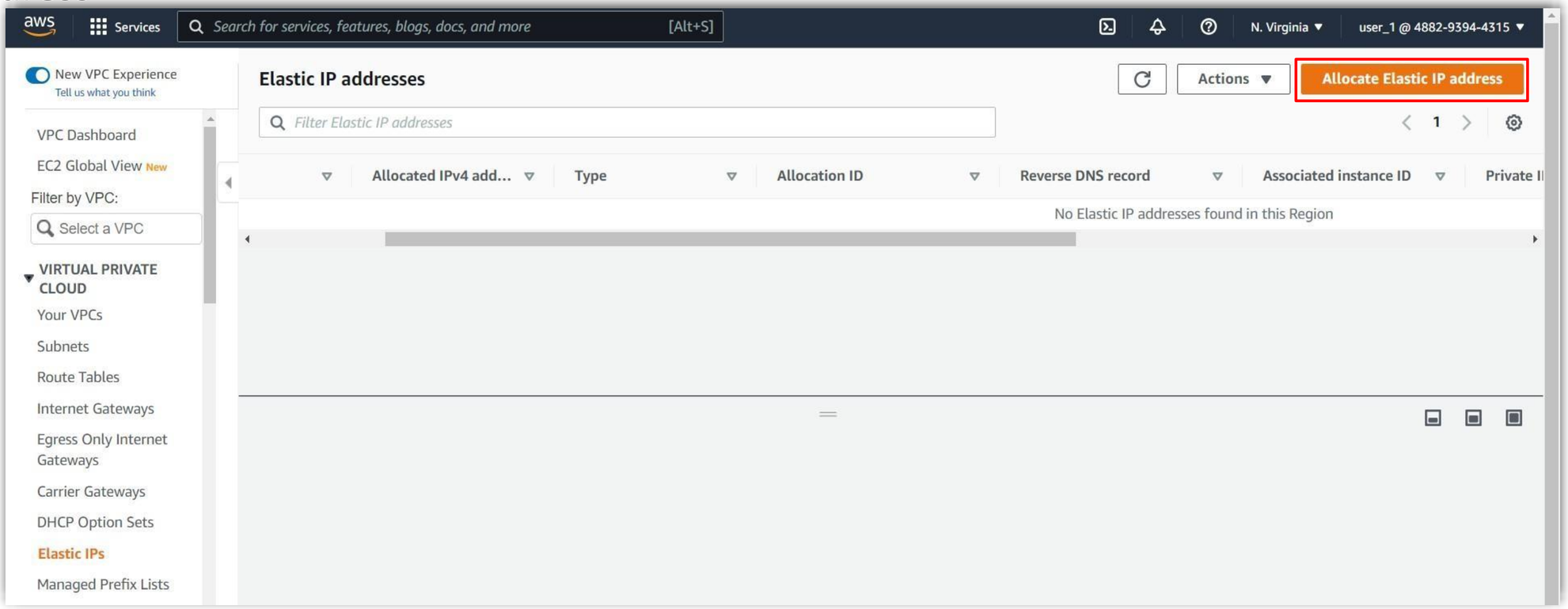
Step 1: Go to **Amazon VPC console** at https://console.aws.amazon.com/vpc/

Step 2: Click on **Elastic IPs** from the navigation pane



Elastic IP Addresses

Step 3: Click on **Allocate Elastic IP Address**



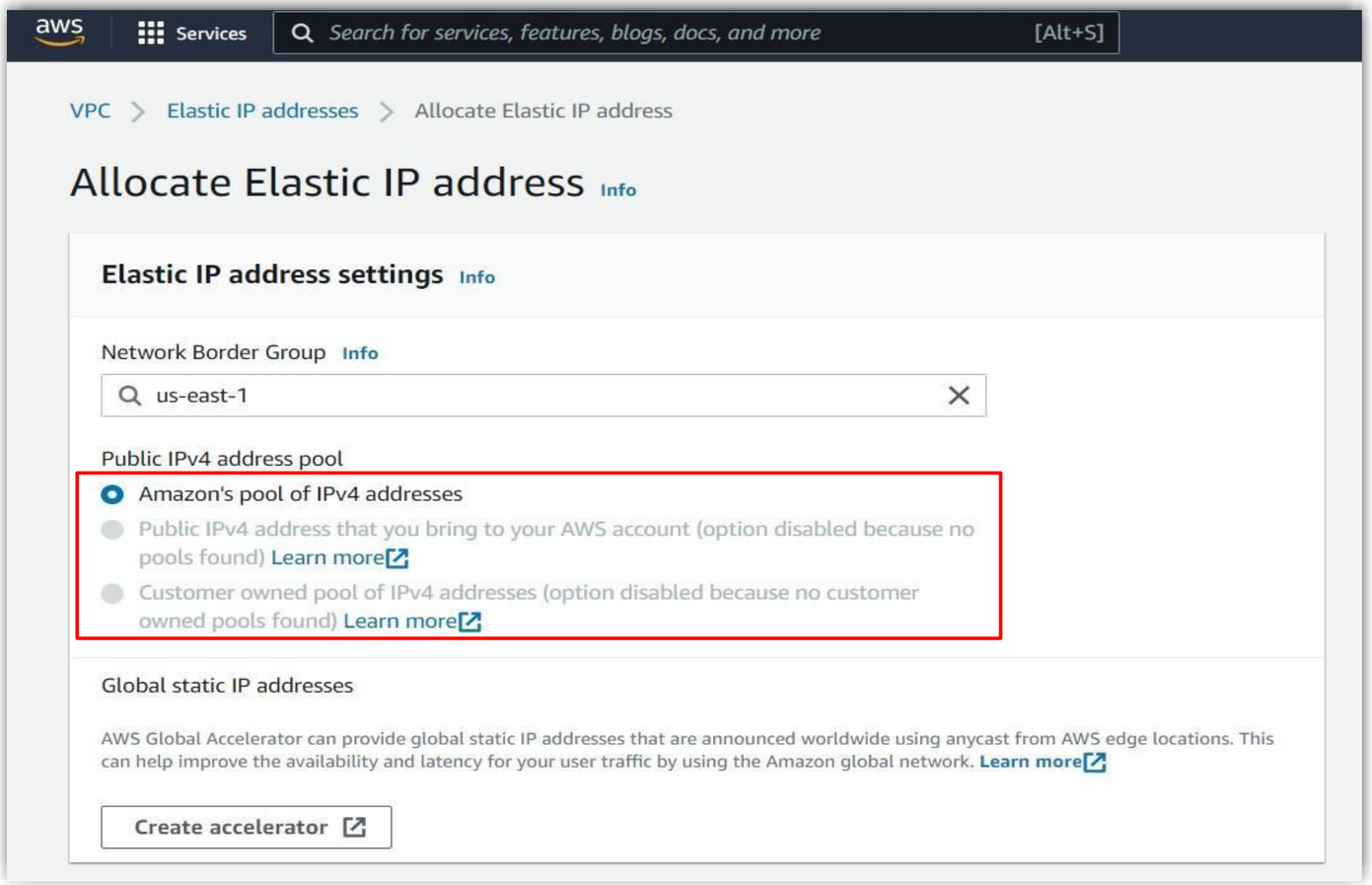
Elastic IP Addresses

Step 4: Choose one of the following for **Public IPv4 address pool:**

- **Amazon's pool of IP addresses:** If you want to be assigned an IPv4 address from Amazon's pool of IP addresses
- **My pool of public IPv4 addresses:** If you want to assign an IPv4 address from a pool of IP addresses that you have added to your AWS account. If you don't have any IP address pools, this option is disabled
- **Customer owned pool of IPv4 addresses:** If you want to assign an IPv4 address to an Outpost from a pool generated from your on-premises network. If you have an Outpost, you can use this option

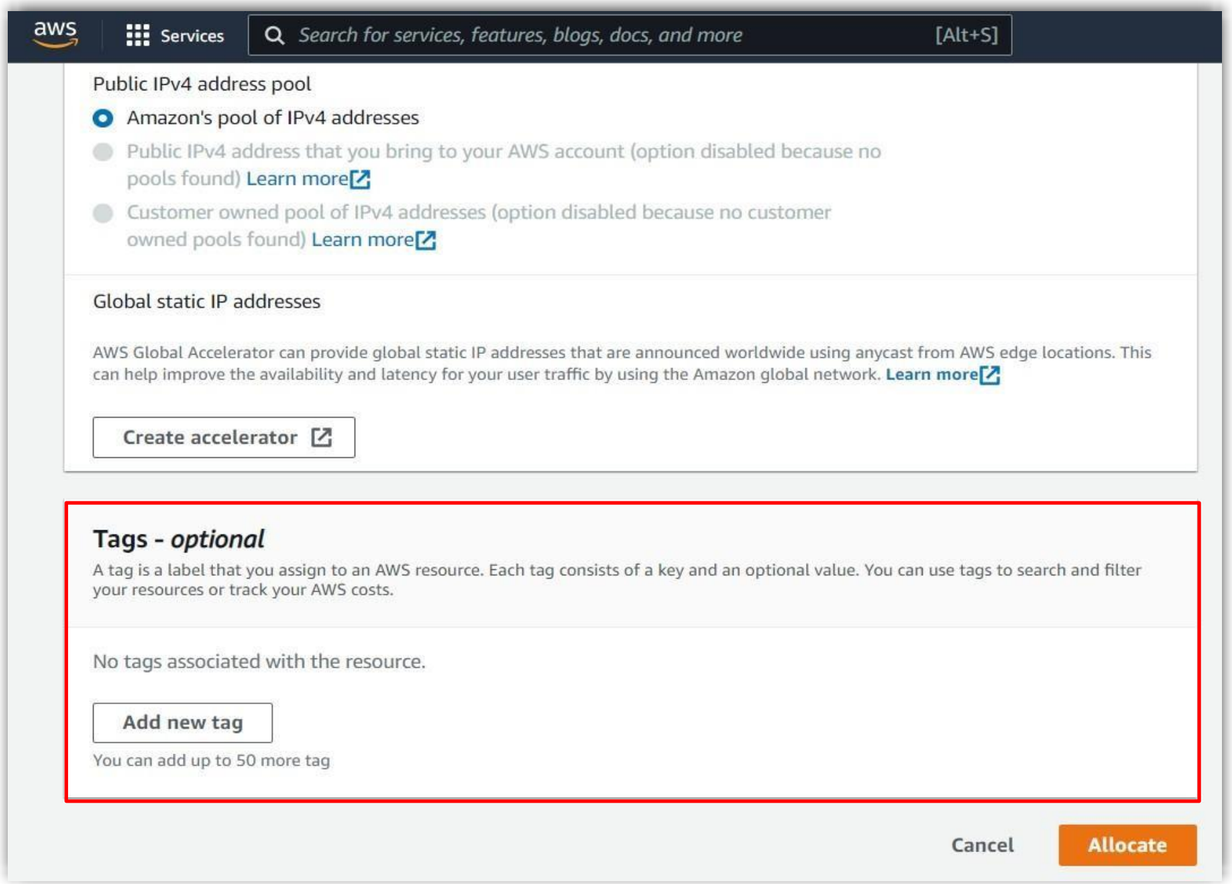
Elastic IP Addresses

(Continued)



Elastic IP Addresses

Step 5: Add or **Remove a tag** (optional)

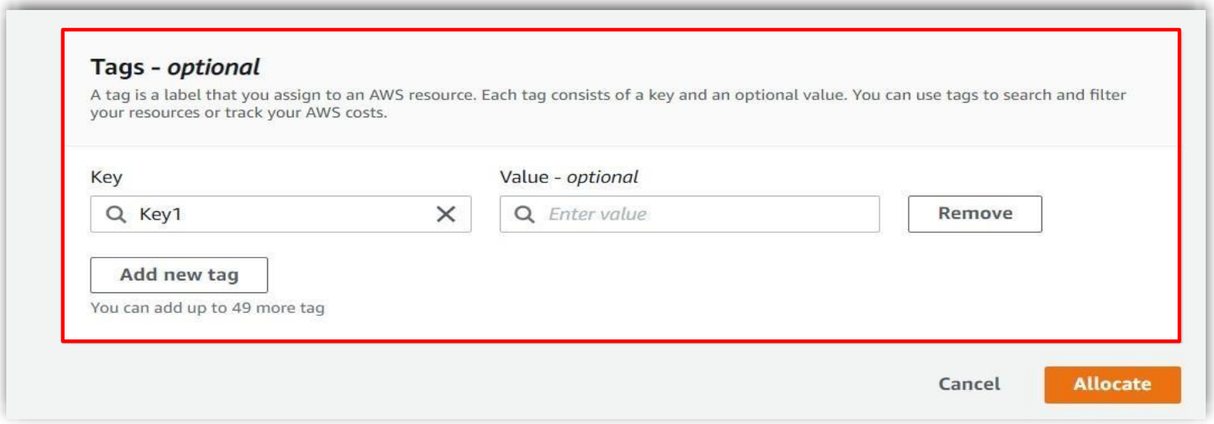


Elastic IP Addresses

(Continued)

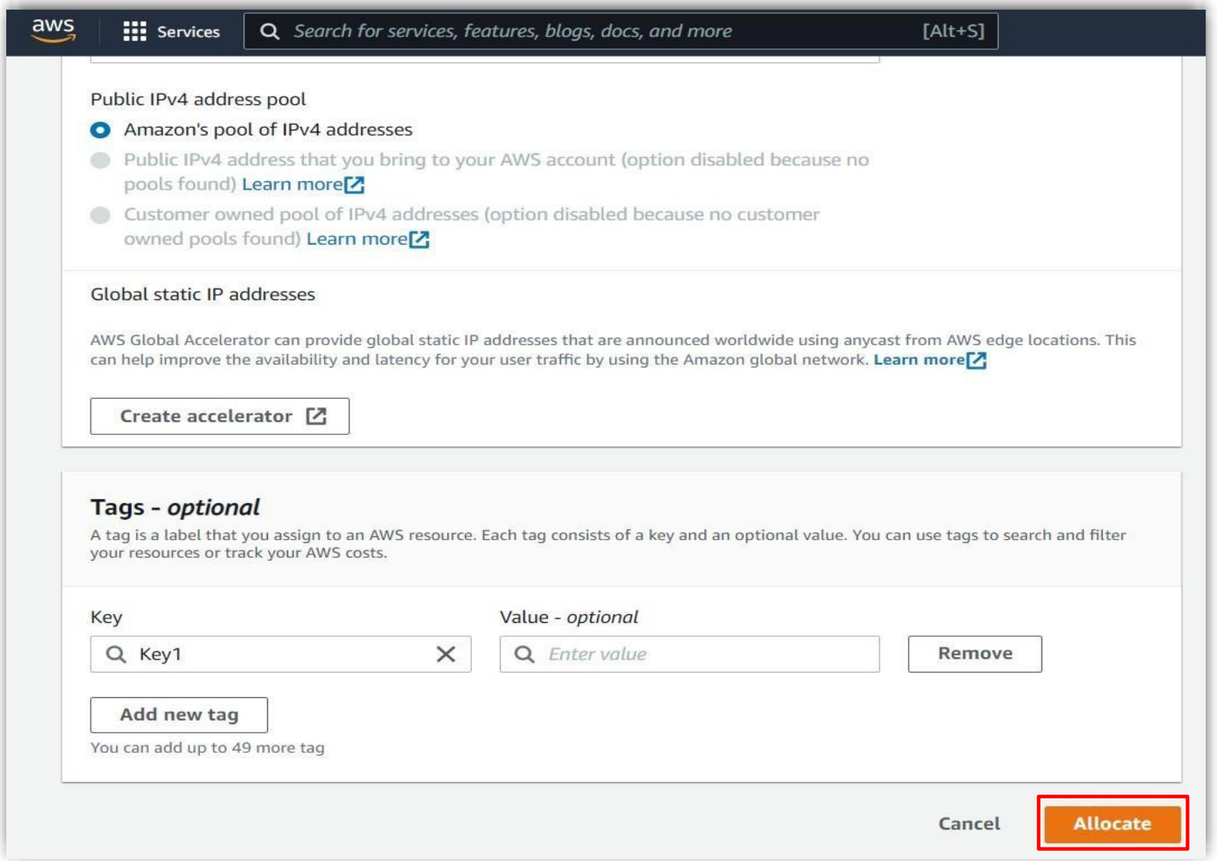
- Click on **Add a new tag** and do the following:
  - For **Key**, enter the key name
  - For **Value**, enter the key value

- Click on **Remove** to remove the tag's Key and Value



Elastic IP Addresses

Step 6: Click on **Allocate**



Network Address Translation

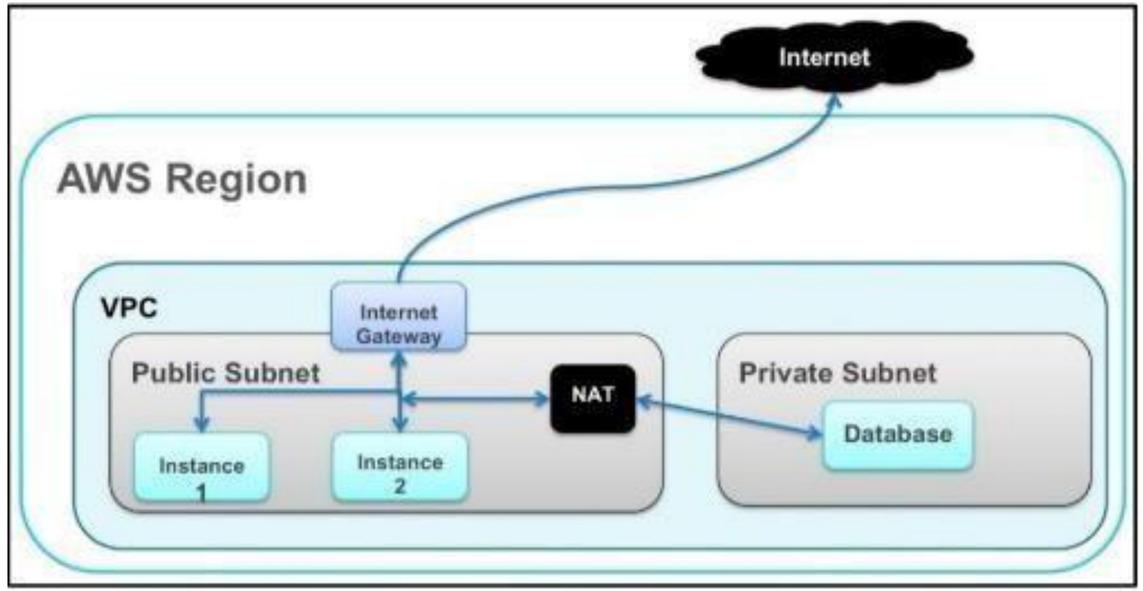
- A NAT instance enables the instances in the private subnet to connect to the internet through the public subnet. At the same time, it prevents inbound internet communications to the instances in the private subnet

- The main route table routes the traffic from the private subnet to the NAT instance in the public subnet. The NAT instance sends the traffic to the Internet Gateway

- The NAT instance specifies a high port number for the response. The NAT instance sends the information from the internet to the instance in the private subnet to the specified port number

Network Address Translation

- Example of a NAT instance:

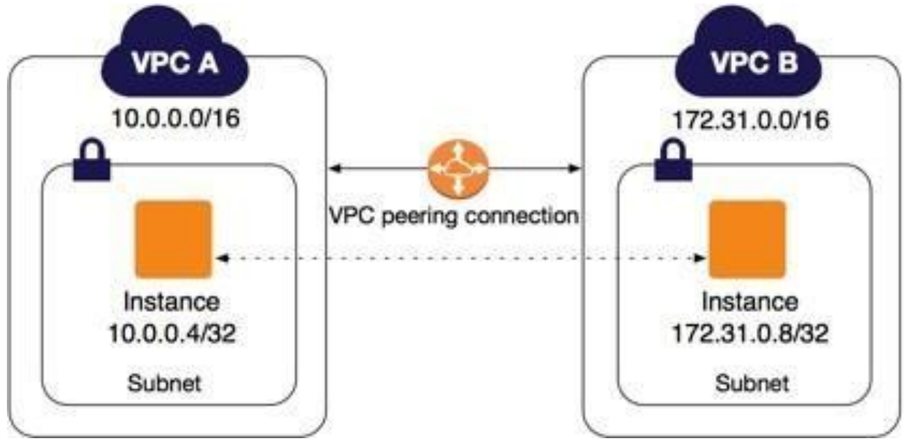


Network Address Translation Devices

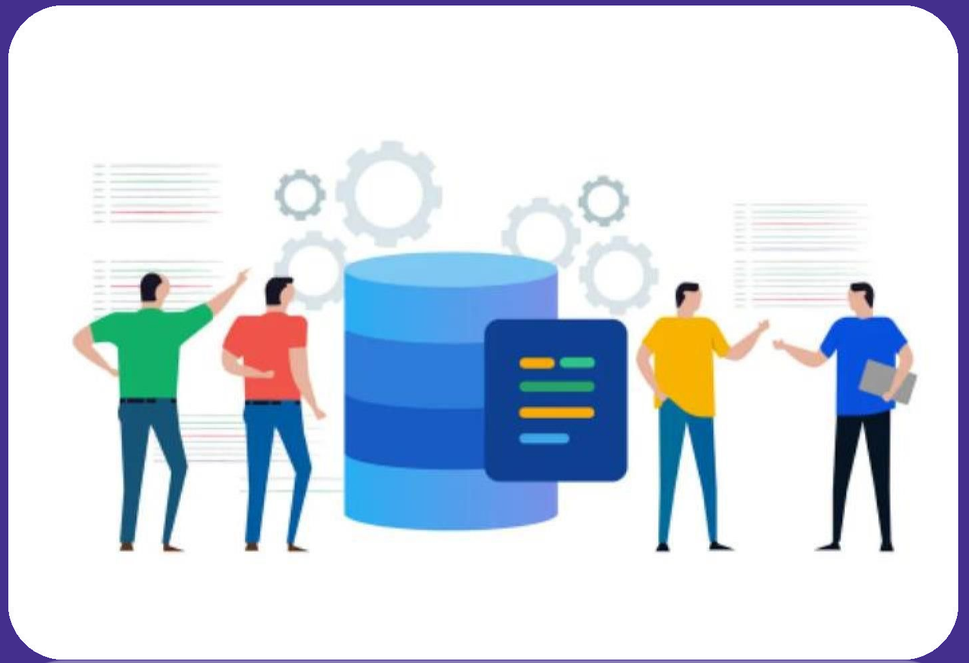
- A NAT device can be used to allow instances in private subnets to connect the internet, other VPCs, and on-premises networks
- These instances can communicate with services outside the VPC, but they can not accept connection requests
- The source IPv4 address of the instances is replaced by the NAT device's address by the NAT device
- The NAT device converts the addresses back to the original source IPv4 addresses while sending response traffic to the instances. You can either utilise an AWS-managed NAT device, a NAT gateway or establish your own NAT device, a NAT instance, on an EC2 instance

VPC Peering

- It allows you to start AWS resources into a virtual network that you have specified
- It is a networking connection within two VPCs that allows you to route traffic among them using private IPv4 addresses or IPv6 addresses
- If they are within the same network, then the instances can interact with each other. You also create a VPC peering connection between your own VPCs, or with a VPC in another AWS account
- The VPCs can be in various regions, also known as an inter-region VPC peering connection



Module 5: Databases



Relational Databases

- A relational database is a collection of data items having pre-defined relationships between them. These items are arranged in a tabular format, with columns and rows

- Tables hold data about the objects that will be represented in the database. A field keeps the actual value of an attribute, while each column in a table holds a specific type of data

- The table's rows represent a collection of related values for a single object or entity. A primary key can be assigned to each row in a table, and foreign keys can be used to relate rows from other tables. Without reorganising the database tables themselves, this data can be accessed in a variety of ways

Relational Databases

(Continued)

- The following services are included in the AWS database service:
  - **Amazon Relational Database Service:** It is compatible with six popular database engines
  - **Amazon Redshift:** It is a petabyte-scale data warehouse service
  - **AWS Database Migration Service:** It is a service that offers easy and inexpensive to migrate your databases to AWS cloud
  - **Amazon Elasticache:** It is an in-memory cache service with support for Redis and Memcached
  - **Amazon Aurora:** It is a MySQL-Compatible relational database with five times performance
  - **Amazon DynamoDB:** It is a flexible and fast NoSQL database service

Amazon Relational Database Service

- Amazon RDS (Amazon Relational Database Service) is defined as a web service that makes it easy to set up, administer, and scale a relational database in the Amazon Web Services Cloud

- It performs routine database management activities and provides cost-effective, resizable capacity for an industry-standard relational database

- Amazon RDS supports a variety of database instance types, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server, and offers six well-known database engines to choose from, including Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle Database, and SQL Server

- You can quickly migrate or replicate your existing databases to Amazon RDS using the AWS Database Migration Service

Amazon Relational Database Service

(Continued)

Benefits of Amazon (RDS):

Amazon Relational Database Service

1. **Easy to administer**

- Amazon RDS makes it simple to move from project design to deployment
- To gain access to the features of a production-ready relational database in minutes, use the Amazon RDS Management Console, simple API calls, or the AWS RDS Command-Line Interface
- There is no need for installing and maintaining database software and no need for infrastructure provisioning

2. **Highly scalable**

- With a few mouse clicks or an API request, you can scale your database's compute and storage capacity with little or no downtime
- You can use a variety of Amazon RDS engine types to launch one or more services
- To offload read traffic from your primary database instance, use Read Replicas

Amazon Relational Database Service

3. **Available and Durable**

- Amazon RDS is built on the same high-reliability architecture as used by other Amazon Web Services

- When you create a Multi-AZ DB Instance, Amazon RDS replicates the data to a backup instance in a separate Availability Zone synchronously (AZ)

- Automated backups, database snapshots, and automatic host replacement are just a few of the services that Amazon RDS has to offer for crucial production databases

4. **Fast and Secure**

- **Fast**

  - The most demanding database applications are supported by Amazon RDS

Amazon Relational Database Service

(Continued)

- You have two SSD-backed storage solutions to select from, one optimised for high-performance OLTP applications and the other for cost-effective general-purpose use

- Furthermore, Amazon Aurora offers performance comparable to professional databases at 1/10th of the cost

- **Secure**

  - Controlling network access to your database is simple with Amazon RDS

  - You can also operate your database instances in Amazon Virtual Private Cloud (Amazon VPC), which allows you to isolate your database instances while connecting to your existing IT infrastructure using an industry-standard secured IPsec VPN

  - Many Amazon RDS engine types support encryption in transport and at rest

Amazon Relational Database Service

5. **Inexpensive**

- You pay very low rates and only pay for what you use. You also have the choice of On-Demand pricing, which requires no upfront or long-term commitments, or Reserved Instance pricing, which provides even lower hourly prices

Amazon Redshift

- Amazon Redshift is a cloud-based, fully managed petabyte-scale data warehouse service. Starting with a few hundred gigabytes of data, you can scale up to a petabyte or more

- This allows you to acquire new insights for your business and customers by using your data. The first step in creating a data warehouse is to set up an Amazon Redshift cluster, which is a collection of machines

- You can upload your data set and then perform data analysis queries after you've provisioned your cluster

- Regardless of the size of the data set, Amazon Redshift provides quick query performance with the use of the same SQL-based tools and business intelligence apps you're already using

amazon REDSHIFT

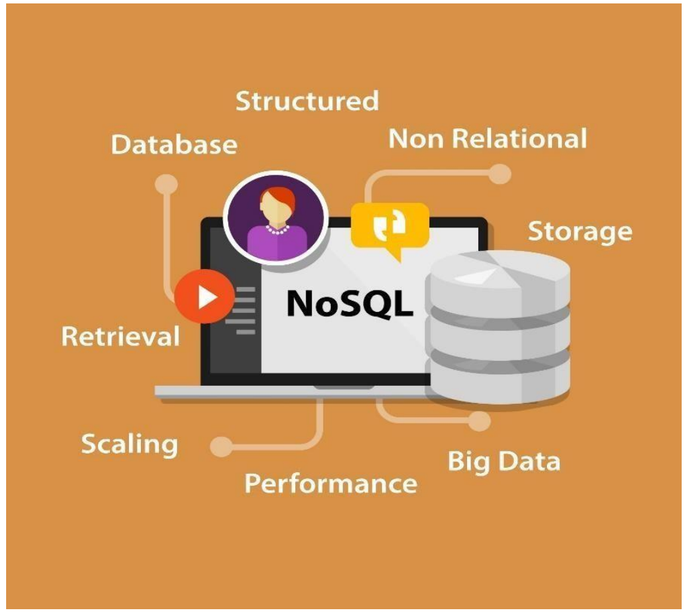
Amazon Redshift

- The following are the features of Redshift:

1	Faster Performance
2	Easy to setup, deploy and manage
3	Cost-effective
4	Scale quickly to meet your needs
5	Query your data lake
6	Secure

Non-Relational (No-SQL) Databases

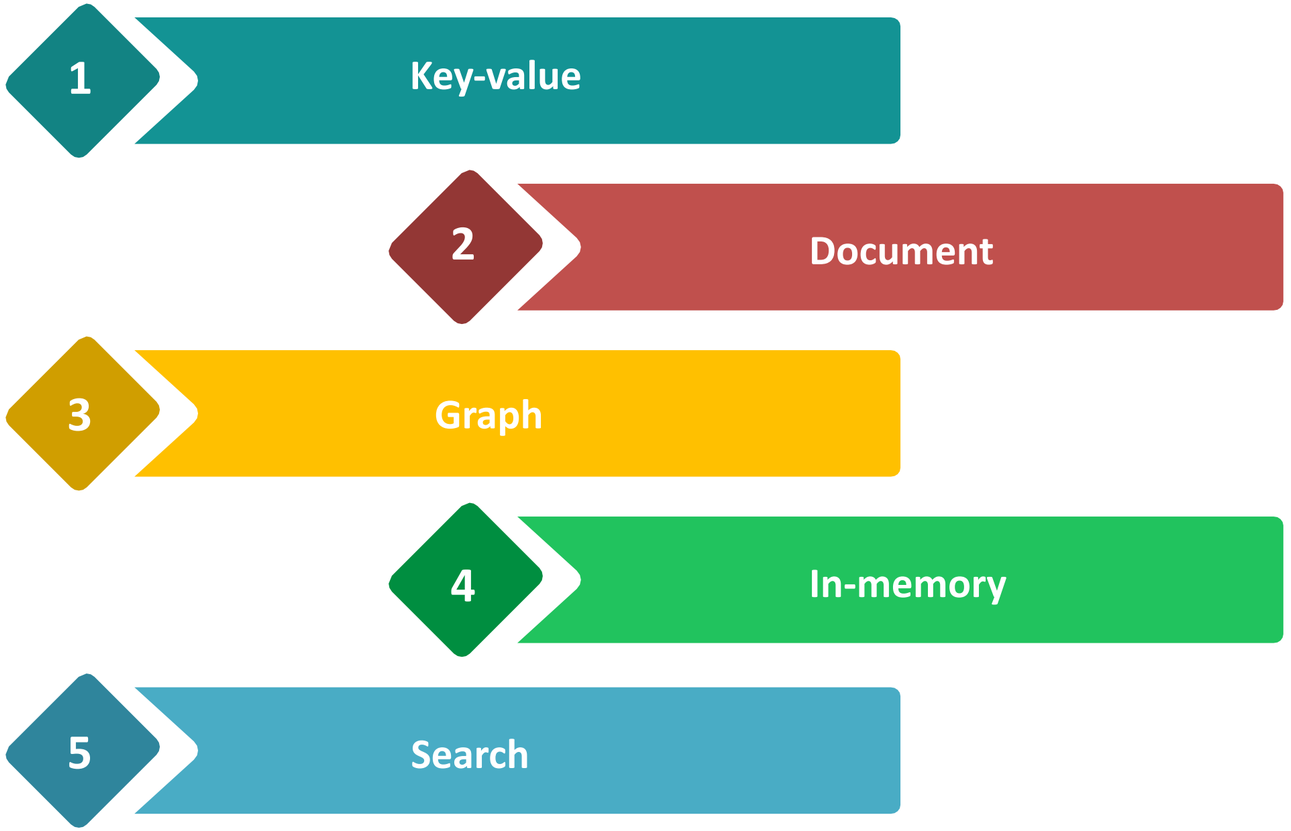
- NoSQL databases enable you to store data in a number of data models with a flexible schema
- These databases are generally simple to use for developers and provide the high performance and functionality that modern applications require
- AWS NoSQL databases can store enormous amounts of data while maintaining low latency



Non-Relational (No-SQL) Databases

(Continued)

- The following are the types of NoSQL Databases:



Non-Relational (No-SQL) Databases

1. **Key-value**

- Key-value databases are extremely partitionable and can scale horizontally to scales that other databases cannot achieve

- The key-value data architecture works especially effectively in applications like gaming, ad tech, and IoT

- For any size of workload, Amazon DynamoDB is designed to provide consistent single-digit millisecond latency

- The Snapchat Stories feature, which involves Snapchat's greatest storage write workload, transitioned to DynamoDB because of its consistent performance

Non-Relational (No-SQL) Databases

2. Document

- Data is frequently represented as an object or JSON-like document in application code because it is an efficient and intuitive data paradigm for developers
- Document databases make it easier to store and query data in a database by using the same document model format that developers use in their application code
- The semistructured, flexible, and hierarchical nature of documents and document databases enables them to develop with applications needs
- The document model works well with user profiles, catalogs, and content management systems where each document is different and develops over time
- Amazon MongoDB and DocumentDB (with MongoDB compatibility) are popular document databases that provide powerful and intuitive APIs for flexible and iterative development

Non-Relational (No-SQL) Databases

3. Graph

- The goal of a graph database is to make it simple to create and execute applications that interact with large, connected datasets

- Social networking, recommendation engines, fraud detection, and knowledge graphs are all common use cases for graph databases. Amazon Neptune is a graph database service that Amazon fully maintains

- Neptune supports the Property Graph model as well as the Resource Description Framework (RDF), with TinkerPop and RDF/SPARQL as graph API options. Neo4j and Giraph are two common graph databases

Non-Relational (No-SQL) Databases

4. **In-memory**

- Leaderboards, session storage, and real-time analytics are examples of gaming and ad-tech applications that demand microsecond response times and can see big increases in traffic at any time

- Low-latency, high-throughput applications, such as McDonald's, can't be served with disk-based data stores. Hence Amazon ElastiCache supports Memcached and Redis

- Another instance of purpose-built data storage is Amazon DynamoDB Accelerator (DAX). DynamoDB reads are significantly faster due to DAX

Non-Relational (No-SQL) Databases

5. Search

- Many apps generate log files that developers can use to troubleshoot problems. By indexing, collecting, and searching semistructured logs and metrics, Amazon Elasticsearch Service (Amazon ES) is designed to provide near-real-time visualisations and analytics of machine-generated data

- For a full-text search, Amazon ES is also a robust, high-performance search engine. Expedia uses over 150 Amazon ES domains, 30 TB of data, and 30 billion documents for a range of mission-critical use cases, including operational monitoring and troubleshooting, distributed application stack tracing, and price optimisation

DynamoDB

- Amazon DynamoDB is defined as a fully managed NoSQL database service that offers high performance and scalability
- DynamoDB offloads the administrative responsibilities of operating and expanding a distributed database, and you do not need to bother about setup and configuration, hardware provisioning, software patching, replication, or cluster scalability
- DynamoDB also supports encryption at rest, which reduces the time and effort required to safeguard sensitive data
- You can design database tables using DynamoDB to store and retrieve any amount of data and handle any request traffic

DynamoDB

- You can increase or decrease the throughput capacity of your tables without experiencing any downtime or performance reduction
- You can monitor resource use and performance indicators using the AWS Management Console
- DynamoDB supports the on-demand backup capability
- It enables you to create comprehensive backups of your tables for long-term storage and archive to meet regulatory compliance requirements

High Availability and Durability

- DynamoDB dynamically distributes your tables' data and traffic across a sufficient number of servers to meet your throughput and storage needs while ensuring consistent and fast performance

DynamoDB

(Continued)

- All of your data is kept on solid-state disks (SSDs) and is automatically replicated across various AWS Region Availability Zones, ensuring high availability and data durability

- To keep DynamoDB tables synchronised across AWS Regions, you can utilise global tables

Module 6: Authentication and Authorisation

Introduction

What is IAM?

- IAM (AWS Identity and Access Management) is the web service that provides securely, controlled access to the AWS resources
- A user uses IAM for controlling that who is signed in (authenticated) and has permissions (authorised) to use the resources
- When your AWS account is created, you start with a single sign-in identity that has full access to all services and resources of AWS in the account
- This identity is known as the AWS account root user and is accessed when you sign in with the email address and password that you used for creating an account

Introduction

Authentication

- A principal should be authenticated, that means signed in to AWS by using their credentials for sending a request to AWS

- Some of the services, for example, Amazon S3 and AWS STS, enable the few requests from anonymous users. Although, they are the exception to a rule

- For authenticating from the console as the root user, you should sign in with your email address and the password. As an IAM user, provide your account ID or alias, and after that user name and password

- For authenticating from the API or AWS CLI, you should provide the access key as well as a secret key. Moreover, you might need to provide further security information. For instance, AWS recommends that you utilise the MFA (multi-factor authentication) to increase your account security

Introduction

Authorisation

- You should also be authorised that, means allowed for completing your request. AWS checks for policies that apply to the request using values from the request context during authorization

- Use the policies for determining whether to enable or disenable the request. Most of the policies are stored in the AWS in the form of the JSON documents and it specifies, the permissions for the principal entities

- There are various types of policies that can impact whether the request is authorised. For providing the permissions for accessing the AWS resources in their account, you need only the policies based on the identity

Introduction

(Continued)

- Resource-based policies are popular to grant the cross-account access. The other types of policy are advanced featured and must be utilised carefully
- AWS checks every policy applying to the context of the request. If any single permissions policy contains a denied action, then the AWS denies the whole request and stopping evaluation. It is known as an explicit deny
- As the requests are denied by default, AWS authorises your request only when your every part of the request is permitted by the applicable permissions policies

Introduction

(Continued)

- The following are the rules followed by evaluation logic for the request within a single account:
  - All requests are denied by default. Generally, the requests use the credentials of the AWS account root user for the resources in a account that are always enabled
  - An explicit enable in any of the permissions policy (such as identity-based or resource-based) overrides this default
  - An existence of the Organisations SCP, IAM permissions boundary, or the session policy overrides the allow. If more than one policy type exists, they should all enable the request.
    Otherwise, it can be denied implicitly
  - An explicit deny in any of the policies overrides any allows

IAM Identities

- An AWS account root user or an IAM administrator for the account allows you to create the IAM identities. An IAM identity provides the access to an AWS account

- The collection of IAM users is called a user group that is managed as a unit. An IAM identity shows a user, and can be authenticated and authorised to perform the actions in AWS

- Every IAM identity can be associated with more than one policies. Policies define what actions a role, user or member of a user group can perform, on which resources of AWS and what will be conditions under which they will perform

IAM users

- An IAM user is an entity that is created by you in AWS. The IAM user shows the person or service who uses the IAM user for interacting with AWS

- A primary use for IAM users is to provide individuals with the capability to sign in to the AWS Management Console for interactive tasks and for making the programmatic requests to the AWS services with the use of the API or CLI

IAM Identities

(Continued)

- A user in the AWS includes a name, a password to sign in to an AWS Management Console, and up to two access keys that can be utilised with the API or CLI
- When you are creating an IAM user, you grant it permissions by making it a user group member that has appropriate permission policies attached or attaching policies to the user immediately
- Moreover, you can clone the permissions of an existing IAM user, which automatically makes a new user a member of the same user group and attaches policies to the user

IAM Identities

IAM User Groups

- An IAM user group is defined as a collection of IAM users. It can be used to specify the permission for a group of users, including permission that helps to manage those users
- For instance, you could have a user group called Admins and provide that user group the permissions types that administrators typically require. Any user in the user group automatically has permissions that are assigned to the user group
- If the new user is joining your organisation and must have administrator privileges, you can assign the appropriate permissions by adding a user to the user group

IAM Identities

IAM Roles

- An IAM role is same as a user, in that it is an identity with the permission policies that define what the identity can and cannot do in the AWS. Although, the role does not have any credentials (such as, password or access keys) associated with it

- Rather than being uniquely associated with a individual, a role is designed to be assumable by anyone who requires it. An IAM user can assume a role for temporarily taken on the multiple permissions for a particular task

- A role can be assigned to a federated user who signs in with the use of an external identity provider rather than IAM. AWS make use of the details that are passed by the identity provider for determining which role is mapped to a federated user

IAM Identities

Temporary Credentials in IAM

- Primarily, the temporary credentials are used with IAM roles, but there are also other uses. You can request the temporary credentials contain more restricted permissions set as compared to standard IAM user

- It prevents you from accidentally performing the tasks that are not permitted by the more restricted credentials

- A advantage of temporary credentials is that they expire automatically after a specific period of time. You have control across the duration that the credentials are valid

Module 7: CloudTrail, CloudWatch, and AWS Config

CloudTrail

- AWS CloudTrail is a service of AWS that allows you to enable governance, operational and compliance, and risk auditing for your AWS account. CloudTrail records actions made by a user, role, or AWS service as events
- Events contain actions taken in the AWS Management Console, AWS SDKs, APIs, and AWS Command Line Interface
- When you create an AWS account, CloudTrail is enabled. The activity is recorded in a CloudTrail event when activity happens in your AWS account
- By going to the Event History in the CloudTrail console, you can easily view the recent events. Create a trail in your AWS account for an ongoing record of your events and activity

CloudTrail

- Visibility into your AWS account activity is an important factor of operational and security best practices
- You can use CloudTrail to view, download, search, analyse, archive, and respond to account activity over your Amazon Web Services infrastructure
- You can recognise who or what performed which action when the event occurred, what resources were worked upon, and other details to help you analyse and respond to activity in your AWS account
- Optionally, you can allow AWS CloudTrail Insights on a trail to help you identify and respond to unusual activity
- You can merge CloudTrail into applications by using the API, verify the status of trails you create, automate trail creation for your organisation, and control how users view CloudTrail events

CloudTrail

How CloudTrail Works

- When you create an AWS account, CloudTrail is enabled. The activity is recorded in a CloudTrail event when activity happens in your AWS account. By going to the Event History in the CloudTrail console, you can easily view the recent events

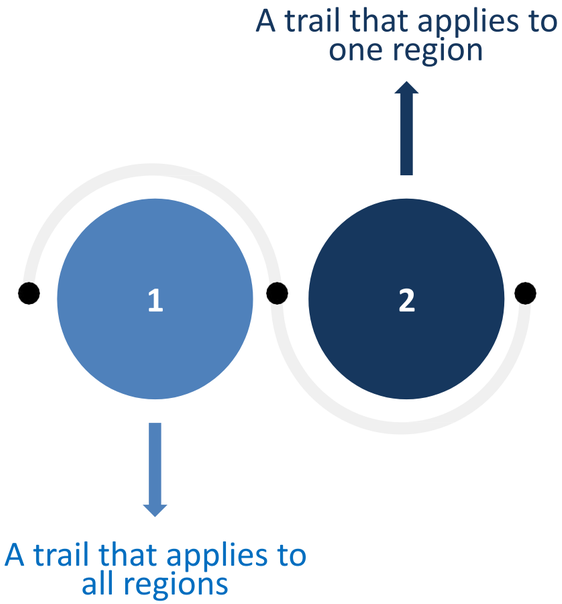
- You can view, search, and download the activity of the last 90 days in your AMS account using Event history. Also, you can create a CloudTrail trail to analyse, archive, and respond to modifications in your AWS resources

- With Amazon CloudWatch Logs and Amazon CloudWatch Events, you may also distribute and analyse the events in a trail. You can create a trail with the CloudTrail console, the CloudTrail API, or the AWS CLI

CloudTrail

(Continued)

- For an Amazon Web Services account, you can create two kinds of trails:



CloudTrail

1. **A trail that applies to all regions**

- CloudTrail records events in each region and sends the CloudTrail event log files to a specified S3 bucket while you create a trail that applies to all regions

- If you add a region after creating a trail that applies to all regions, the new region is automatically included, and events in that region are logged

- Because creating a trail in all regions is a suggested best practice for capturing activity across all regions in your account, it is the default option when creating a trail in the CloudTrail console

- By using the AWS CLI, you can only update a single-region trail to log all regions

CloudTrail

2. A trail that applies to one region

- When you create a trail that applies to a single region, CloudTrail records the events only in that region
- It then delivers the CloudTrail event log files to an Amazon S3 bucket that you select. Using the AWS CLI, you can only create a single-region trail
- If you create more single trails, you can have those trails distribute CloudTrail event log files to the same or separate Amazon S3 bucket
- This is the default choice when you create a trail using the CloudTrail API or the AWS CLI

CloudWatch

- Amazon CloudWatch monitors your AWS resources and the applications you execute on AWS in real- time
- CloudWatch can be used to collect and track metrics, which are variables that can be measured for your applications and resources
- The metrics of every AWS service you use are automatically shown on the CloudWatch home page. You may also create a custom dashboard to show metrics about your custom application as well as a custom group of metrics that you choose
- When a threshold is breached, you can create an alarm that watches metrics and sends notifications or automatically make changes to the resources you are monitoring
- For instance, you can monitor the disk reads, CPU usage, and writes of your Amazon EC2 instances and then use that data to evaluate whether you should deploy other instances to manage the increased load
- This data can be used to stop under-used instances to save money

CloudWatch

What is Amazon CloudWatch Logs

- Amazon CloudWatch can be used to monitor, store, and access your log files from AWS CloudTrail, Amazon Elastic Compute Cloud (Amazon EC2) instances, Route 53, and different sources

- CloudWatch Logs allows you to centralise the logs from all of your applications, systems, and AWS services that you can use into a single, highly scalable service

- You can view them then easily, search for specific error codes or patterns, filter them based on certain fields, or safely archive them for future analysis

- You can see all of your logs, regardless of sources, as a single and constant flow of events ordered by time, and you can query as well as sort them based on other sizes, combine them by certain fields, create custom calculations with powerful query language, and visualise log data in dashboard with CloudWatch Logs

CloudWatch

Using Amazon CloudWatch Alarms

- You can create both metric alarms and composite alarms in CloudWatch:

1. **Metric Alarm:**

   - A single CloudWatch metric or the result of the math expression based on CloudWatch is monitored by a metric alarm

   - The alarm performs one or more actions based on the expression's value or metric's value relative to a threshold across various time periods

   - Sending a notification to an Amazon SNS topic, an Amazon EC2 Auto Scaling action, performing an Amazon EC2 action or creating an OpsItem or incident in Systems Manager are all examples of actions

CloudWatch

2. Composite Alarm:

- A composite alarm has a rule expression that takes into account the alarm state of different alarms you have created

- Only if all the conditions of the rule are met does the composite alarm become an ALARM state. The alarms defined in a rule of composite alarm expression can have metrics alarms as well as other composite alarms.
- If the conditions of the rule are met, the composite alarm goes into ALARM state

AWS Config

- AWS Config gives you a detailed configuration of AWS resources in your AWS account
- This contains how the resources are connected to one another and how they were previously configured, so that you can see how the relationships and configurations change over time
- An AWS resource is an entity that can work within AWS, like an Amazon Elastic Compute Cloud (EC2) instance, a security group, an Amazon Virtual Private Cloud (VCP), or an Amazon Elastic Block Store (EBS) volume are examples of AWS resources
- You can do the following with AWS Config:
    - Examine your AWS resource configurations for required settings
    - View the connections among resources. You might want to locate all resources that use a specific security group, for example
    - Get a snapshot of the current configurations of the supported resources linked to your AWS account

AWS Config

(Continued)

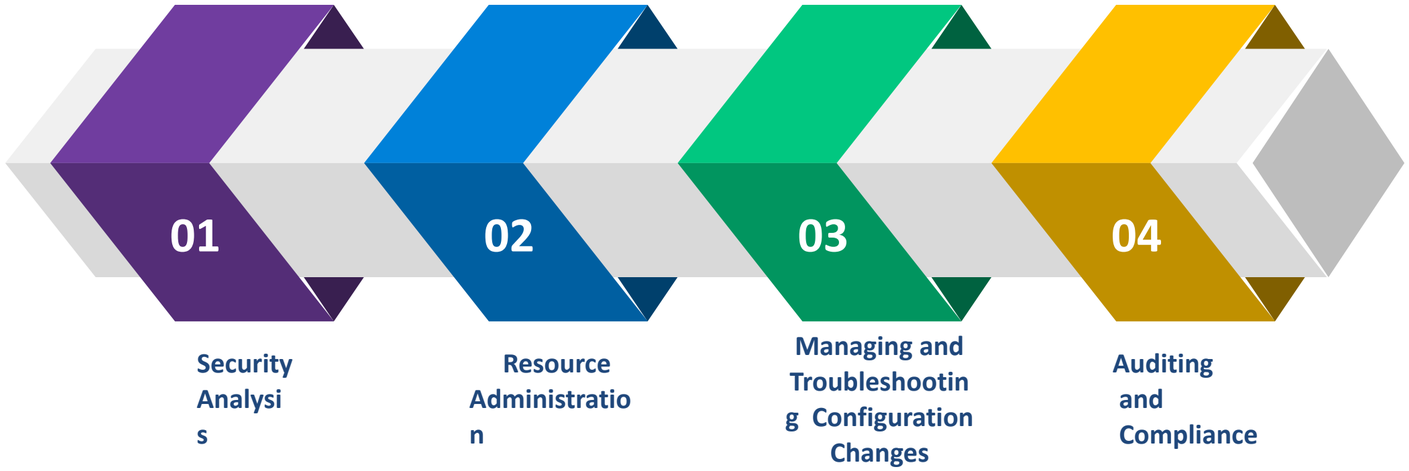
- Retrieve the settings for one or more resources in your account
- Obtain one or more resource's previous configurations
- When a resource is created, changed, or destroyed, you will get a notification

AWS Config

Ways to Use AWS Config

- You typically need AWS resources to execute your apps on AWS, which you must create and maintain collectively

- You need to keep track of your AWS as the demand for your application keeps growing. AWS Config is planned to help you manage your application resources in the following techniques:



Module 8: Domain Name System and Network Routing

Domain Name System

- All machines on the Internet use numbers to find and connect with one another, from your smartphone or laptop to the servers that deliver content for big retail websites

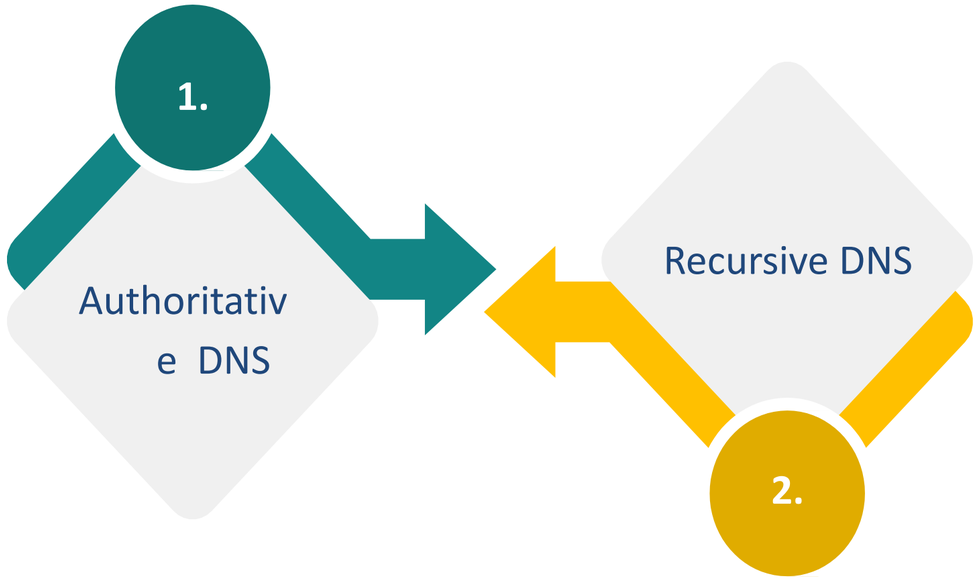
- These numbers are called as IP addresses. When you go to website through web browser, you do not need to remember and enter any long number. Rather than entering a domain name such as, example.com and still end up in a right place

- A DNS service for example, Amazon Route 53 is a globally distributed service translating the human readable names such as www.example.com into a numeric IP addresses such as 192.0.2.1 that computers are using for connecting to each other

- The DNS system of the internet is working as a phone book by managing the mapping between numbers and names. DNS servers are translating the requests for names into the IP addresses that control which server an end user will reach when they are typing a domain name into their web browser. These requests are known as queries

Domain Name System

Types of DNS Service



Domain Name System

1. Authoritative DNS

- An authoritative DNS service offers an update mechanism that the developers use for managing their public DNS names

- It then answers DNS queries, translate the domain names into the IP address so that the computers can interact with each other

- Authoritative DNS has the ultimate authority across a domain and which is responsible to provide the answers to the recursive DNS servers with the information of IP address. Amazon Route 53 is an authoritative DNS system

Domain Name System

2. Recursive DNS

- Generally, the clients do not make queries instantly to the authoritative DNS services. Rather than, they generally connect to another DNS service type known a resolver, or the recursive DNS service
- A recursive DNS service acts as the hotel concierge: while it does not own any DNS records, it acts like an intermediary who can get the DNS information on your behalf
- Recursive DNS answers the query of the DNS by providing the source or IP information if it has the DNS reference cached. If not, then it passes the query to more than one authoritative DNS servers for finding the information

Amazon Route 53

- It is a highly available and scalable cloud DNS web service. It is designed to provide the developers and businesses an extremely reliable as well as cost effective way for routing the end users to the internet applications by translating names such as, www.example.com into the numeric IP addresses for example, 192.0.2.1 that computers use for connecting to each other. It is also fully compliant with IPv6

- It effectively connects the user requests to the infrastructure running in the AWS – for example, Elastic Load Balancing load balancers, Amazon EC2 instances, or Amazon S3 buckets – and also used to the route users to infrastructure outside of AWS

- You can use Amazon Route 53 for configuring the DNS health checks, then it continually monitoring your applications' ability for recovering from the failures and controlling the application recovery with the Route 53 Application Recovery Controller

Amazon Route 53

(Continued)

- Amazon Route 53 Traffic Flow helps to manage the traffic globally via various types of routing that includes the Latency Based Routing, Geoproximity, Geo DNS, and Weighted Round Robin, all of which can be combined with the DNS Failover to allow various low-latency and fault-tolerant architectures

- You can easily manage how your end-users are routed to your application's endpoints with the use of Amazon Route 53 Traffic Flow's simple visual editor, whether in a single AWS region or distributed around the globe

- Moreover, it offers Domain Name Registration that you can purchase as well as manage the domain names like example.com and Amazon Route 53 that will automatically configure the DNS settings for your domains

Amazon Route 53

Benefits

1. **Highly available and reliable:** It is built using the highly available and reliable infrastructure of AWS. The DNS distributed nature helps to assure a consistent ability to route your end users to the application

2. **Flexible:** Amazon Route 53 Traffic Flow routes traffic on the basis of multiple criteria, for example, endpoint health, latency, and geographic location. You can also configure the multiple traffic policies and decide which policies are active at any specific time

3. **Designed for use with other Amazon Web Services:** It is designed for working well with another AWS offerings and features. You can use Amazon Route 53 for mapping the domain names to the Amazon EC2 instances, Amazon CloudFront distributions, Amazon S3 buckets, and other resources of AWS

Amazon Route 53

4. **Simple:** With the help of self-service sign-up, Amazon Route 53 can start answering your DNS queries within minutes. You can also configure the DNS settings with the AWS Management Console or easy-to-use API

5. **Fast:** Amazon Route 53 is designed to automatically route users to the optimal location, relying on the network conditions with the use of a global anycast network of DNS servers around the world

6. **Cost-Effective:** You need to pay just for the resources that you use, for example, the number of queries that the service answers for each domains hosted zones to manage the domains via the service, and optional characteristics, for example, traffic policies and health checks, all at a minimum cost and without the minimum usage commitments or any up-front fees

Amazon CloudFront

- It is a web service speeding up the distribution of static and dynamic web content, for example, .html, .css, .js, and image files, to your users
- CloudFront allow you to deliver the content via a worldwide network of data centres known as edge locations
- When the user is requesting the content that you serve with the CloudFront, the request is routed to the edge location providing the lowest latency (i.e. time delay); as a result, the content is delivered with a best possible performance
- If content is already in the edge location having the lowest latency, CloudFront immediately delivers it
- If the content is not in that edge location, then the CloudFront is retrieving it from an origin which is defined by you, for example, a MediaPackage channel, an Amazon S3 bucket, or an HTTP server (such as a web server) that you have identified as a source for the definitive content version

Amazon CloudFront

(Continued)

- It speeding up your content distribution by routing the each user request via the AWS backbone network to the location of the edge that can best serve the content
- Generally, it is a CloudFront edge server providing the fastest delivery to the viewer. With the use of AWS network dramatically decreases the number of networks that your users' requests should pass via, which is improving the performance
- Users get lower latency such as the time it takes for loading the file’s first byte—and the higher data transfer rates
- Moreover, you get increased availability and reliability as copies of the files (also called as objects) are now held (or cached) in various edge locations around the world

Module 9:

Reliability Pillar

Introduction

- The AWS Well-Architected Framework helps in understanding the benefits and drawbacks of decisions made while building workloads on AWS
- You will understand architectural best practices for designing and operating reliable, efficient, secure, and cost-effective workloads in the cloud by using the framework,
- It provides a technique to measure your architecture against best practices and specify areas for progress
- The following are the six pillars that support the AWS Well-Architected Framework:
  - Security
  - Operational Excellence
  - Cost Optimisation
  - Reliability
  - Sustainability
  - Performance Efficiency

Calculating Availability

- Availability is also called as service availability. It is commonly used metric for a target resiliency goal as well as quantitatively measuring resiliency
- The availability is defined as a percentage of time that a workload is accessible for use
- Available for use means that it performs its approved function successfully when needed. This percentage is calculated across a specific time, like a month, a year, or the last three years
- Availability is reduced whenever the application is not running normally, which includes both scheduled and unschedu est possible interpretation

$$\text{Availability} = \frac{\text{Available for Use Time}}{\text{Total Time}}$$

Calculating Availability

- Availability is defined as a percentage of uptime (like 99.9%) throughout a certain time period (commonly a month or year)
- The "number of nines" is commonly referred to in shorthand; for example, "five nines" means "99.999 percent available“
- Some clients choose to exclude scheduled service downtime (for instance, planned maintenance) from the formula's total time
- But, this is not recommended, as your users would usually want to use your services during these times

Calculating Availability

(Continued)

- The following table shows common application availability design objectives and the maximum length of time that interruptions can occur within a year while still meeting the target
- The table defines the examples of applications types that you see at each level of availability

Availability	Maximum Unavailability (per year)	Application Categories
99%	3 days 15 hours	Load jobs, data extraction, batch processing, and transfer
99.9%	8 hours 45 minutes	Internal tools such as knowledge management, project tracking
99.95%	4 hours 22 minutes	Point of sale, Online commerce
99.99%	52 minutes	Broadcast workload, video delivery
99.999%	5 minutes	ATM transactions, telecommunications workload

Calculating Availability

(Continued)

- Measuring availability based on request. It may be easier to count successful and failed requests rather than "time available for use“. The following formula can be used in such cases:

$$\text{Availability} = \frac{\text{Successful Responses}}{\text{Valid Requests}}$$

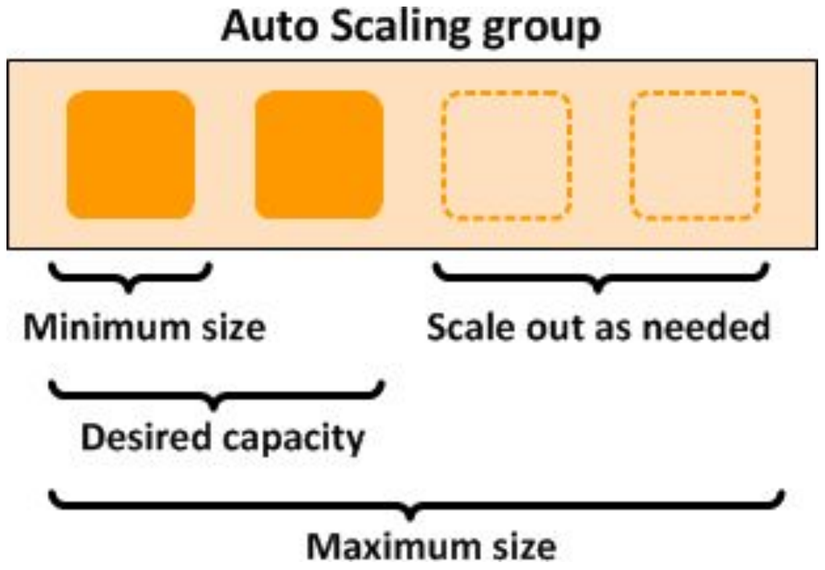
EC2 Auto Scaling

- Amazon EC2 Auto Scaling ensures that the exact number of Amazon EC2 instances available to manage the load of your application
- You can create a group of EC2 instances known as an "Auto Scaling group." The minimum number of instances in each Auto Scaling group can be specified, as well as Amazon EC2 Auto Scaling confirms that group never goes below the size specified
- You can define the maximum number of instances in each auto-scaling group, and Amazon EC2 Auto- Scaling will ensure that your group never exceeds this size
- Amazon EC2 Auto Scaling confirms that your group has this many instances if you specify the desired capacity, either when you create the group or at any point after that
- If you define scaling policies, Amazon EC2 Auto Scaling can launch or terminate instances when demand for your application increases or decreases

EC2 Auto Scaling

(Continued)

- For instance, the Auto Scaling group has a minimum size of one instance, maximum size of four instances and a desired capacity of two instances. The scaling policies that you describe adjust the number of instances based on the criteria that you specify within your minimum and maximum number of instances



Data Backup and Recovery

- Backup data, applications, and configuration in order to meet the needs for RPO (Recovery Point Objectives) and RTO (Recovery Time Objectives)

- **Recognise and back up all data that requires to be backed up, or reproduce the data from sources:**

  - Amazon S3 can be used as a backup destination for various data sources. AWS services such as Amazon RDS, Amazon EBS, and Amazon DynamoDB have built-in abilities to create backups. Or third-party backup software can be used. Alternatively, if the data can be replicated from other sources to achieve RPO, you may not need a backup

- **Secure and encrypt the backup:**

  - By using authorisation and authentication such as AWS Identity and Access Management (IAM) to detect access and using encryption to detect data integrity

Data Backup and Recovery

(Continued)

- **Perform data backup automatically:**
  - Configure a backup to be created automatically based on regular schedules, or by modifications in the dataset. EBS volumes, RDS instances, S3 objects, and DynamoDB tables can be configured for automatic backup. Third-party or AWS Market solutions can also be used
- **Perform periodic recovery of the data to verify backup integrity and processes:**
  - Validate that your backup procedure implementation meets your RTO (Recovery Time Objective) and RPO (Recovery Point Objective) by using a recovery test
  - You can stand up a testing environment on AWS and restore your backups there to consider RTO and RPO abilities and execute tests on data integrity and content
  - Further, Amazon DynamoDB and Amazon RDS enable point-in-time recovery (PITR). You can use a continuous backup to restore your dataset to the state it was in at a particular date and time

The Components of Reliability

- The reliability of the workload in the cloud is determined by various factors, the most important of which is resilience

- Resiliency is the workload ability to recover from infrastructure or service disruptions, dynamically obtain computing resources to meet demand, and reduce disruptions like transient or misconfiguration network problems

- The following are the other factors affecting workload reliability:

  - Operational Excellence, which includes automation changes, use of playbooks for failure response, and ORRs (Operational Readiness Reviews) to verify that the application is ready for production operations

  - Performance efficiency that includes designing for maximum request rates as well as minimising latency for your workload

The Components of Reliability

(Continued)

- Security, which includes preventing unauthorised actors from causing harm to data or infrastructure, lowering availability. To maintain data security, encrypt backups, for example

- Cost optimisation involves trade-offs like whether to spend more on EC2 instances to meet static strength or to depend on automatic scaling when more capacity is required

Designing for Availability

- The availability that select AWS services were designed to fulfill. These values do not represent a Service Level Agreement or guarantee, but instead are part of each service's design goals
- In some particular issues, it differentiates portions of the service when the availability design goal differs meaningfully
- This list is not complete for all AWS services, and expect to update it with information about other services regularly
- The component availability objective is stated accordingly as Amazon Route 53, Amazon CloudFront, AWS Global Accelerator, and the AWS Identity as well as Access Management Control Plane deliver global services
- Other services provide services within an AWS Region and the availability goal is declared accordingly. Many services run within an availability zone, different from those in another availability zone

- It provides the availability design goal for a single availability zone, and then two or more than

Designing for Availability

(Continued)

Services	Components	Availability Design Goals
Amazon API Gateway	Control Plane	99.950%
	Data Plane	99.950%
Amazon Aurora	Control Plane	99.950%
	Single-AZ Data Plane	99.950%
	Multi-AZ Data Plane	99.990%
Amazon CloudSearch	Control Plane	99.950%
	Data Plane	99.950%
Amazon CloudFront	Control Plane	99.900%
	Data Plane (content delivery)	99.990%

Designing for Availability

- It is usually initially thought of an application's availability as a single goal for the application as a complete. Even though you can frequently find that certain elements of an application have distinct availability conditions

- Some systems, for example, may prioritise the ability to receive and store new data over the ability to retrieve existing data. Other systems emphasise real-time operations over actions that alter the configuration or environment of the system

- The availability of services may be very high during specific hours of the day but can tolerate extremely long periods of disruption beyond these hours

- These are some of the methods that you can use to break down a single application into constituent parts and determine the availability needs for each one

- It helps to concentrate the efforts (and expense) on the availability to meet certain requirements, rather than engineering the entire system to the toughest conditions

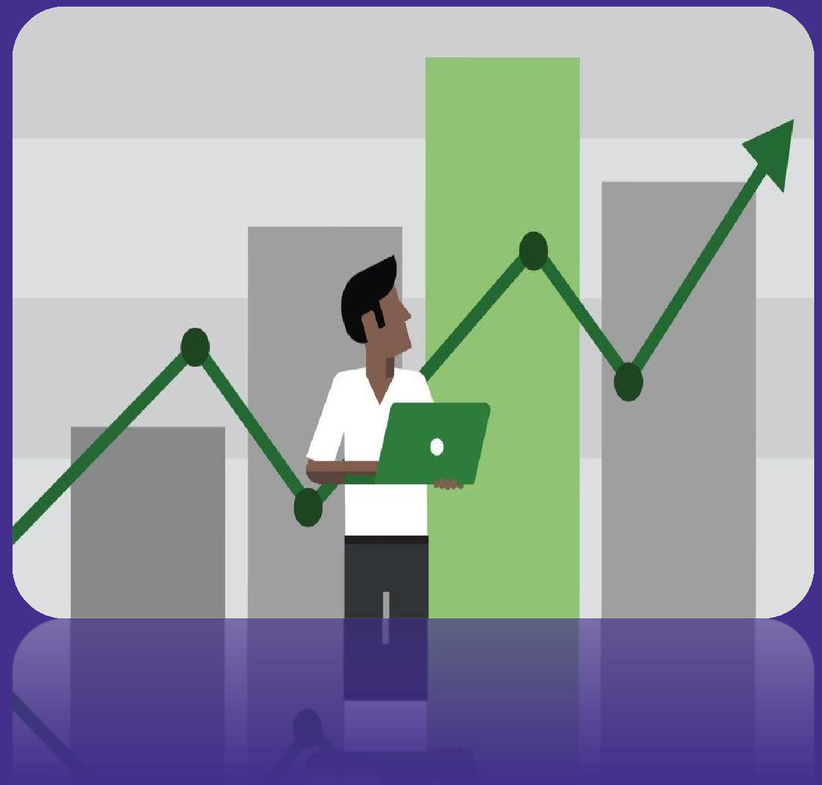
Designing for Availability

- Within AWS, you can generally divide the services into a "control plane" and a "data plane"
- Data planes are reliable for providing real-time service, while control planes are used for configuring the environment
- For instance, data plane operations such as Amazon RDS databases, Amazon EC2, and Amazon DynamoDB table read/write operations
- On the other hand, control plane operations are considered in the founding of new RDS databases or EC2 instances or adding or replacing table metadata in DynamoDB
- While high-level availability is necessary for all of these capabilities, the data planes generally have greater availability design goals than control planes
- As a result, workloads with high availability needs should avoid the run-time relying on control plane operations

Designing for Availability

- The availability requirements required for the workload should be aligned to the business criticality and requirements
- You can evaluate each workload by representing it in the business-critical framework with defined RPO, RTO, and availability
- Such an approach needs the people involved in implementing the workload to be aware of the framework and how their workload affects business requirements

Module 10: Performance Efficiency Pillar



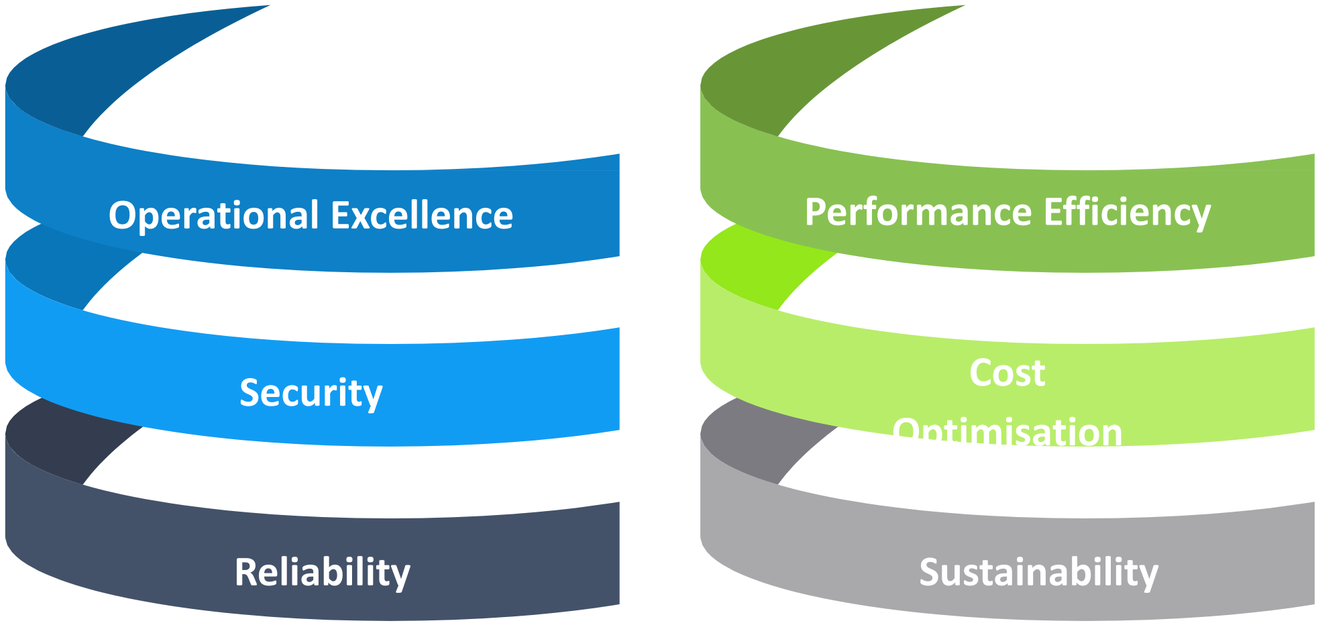
Introduction

Introduction

- The AWS Well-Architected Framework helps you to understand the benefits and drawbacks of decisions made by you while building the workloads on the AWS
- With Framework allows you to learn the architectural best practices to design and operate secure, reliable, cost-effective, and efficient workloads in the cloud
- It provides the way for consistently measuring the architectures against the best practices as well as identifying the areas for enhancement. Having a well-architected workload greatly helps in increasing the likelihood of business success

Introduction

- The framework is based on six pillars:



Introduction

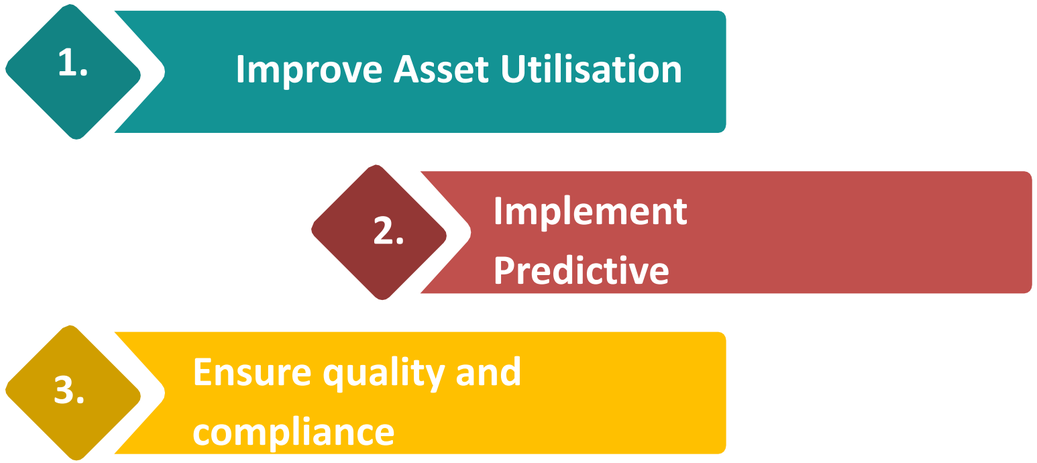
- It focuses on applying the performance efficiency pillar’s principles to the workloads. Traditionally, on-premises environments that achieve the high and lasting performance which is very challenging
- These principles will help you to build the architectures on the AWS that efficiently allows you to deliver the sustained performance across time
- It is intended for those in the technology roles, like (CTOs), chief technology officers, developers, architects, and operations team members

Optimising Performance for the Core AWS Services

- There is a requirement of optimising the assets, implementing the predictive maintenance, enhancing the utilisation, and ensuring the quality, health, as well as safety standards, and so on

- Modest enhancements in efficiency can bring transformative results. AWS offers the companies with everything necessary to immediately and easily manage data while assisting them to understand their operations and consumers

Benefits



Optimising Performance for the Core AWS Services

1. **Improve Asset Utilisation:** It increases the use of data analytics and AI for parsing the information of the consumer for behavioural patterns for better planning inventory, scheduling, and so on

2. **Implement Predictive Maintenance:** Leverage AWS to improve the predictive abilities such as safely adjusting routes, saving on energy and fuel costs, and maintains their equipment and fleet

3. **Ensure Quality and Compliance:** Consistently adhere to safety and health standards and help in building more effective processes

Infrastructure Automation

- Infrastructure-level automation benefits modern designs, whether monolithic or based on microservices
- With the help of virtual machines, IT teams can easily replicate the environments and create the operating system states templates that they required
- The host operating system became disposable and immutable. The idea bloomed and scale was added to the mix with a cloud technology
- There is no requirement for predicting the future when you can just provision on the demand for what you want and pay for what you use
- If an environment is not required anymore then you can shut down the resources. Spot compute can combine the on-demand provisioning, which allows you for requesting the unused compute capacity at a steep discount

Infrastructure Automation

- Microservices not just require the disposable infrastructure-as-code, moreover, they also need to be tested, built, and deployed automatically
- Continuous integration and delivery are essential for monoliths, but they are indispensable for the microservices
- Each service requires a pipeline, an individual can accommodate the several and diverse technology choices that a team makes
- An automated infrastructure supports repeatability for immediately setting up the environments. Development, integration, user acceptance testing (UAT) or performance testing, and production environments can all be dedicated to a single purpose
- Infrastructure defined as code and then instantiated can simply be moved back. It drastically decreases the risk of change and promoting the experiments and innovation

Module 11:

Security Pillar

Introduction

- You can understand the trade-offs with the support of the AWS Well-Architected Framework for the decisions you make during creating a workload on AWS
- Using the framework, you will learn existing architectural best practices for operating and designing a reliable, efficient, secure, and cost-effective workload in the cloud
- It provides a method to measure your workload aganist best practices and recognise the areas for improvement
- Workloads that are well-architected greatly increase the chances of business success
- The following are the six pillars that support the AWS Well-Architected Framework:
  - Security
  - Operational Excellence
  - Cost Optimisation
  - Reliability
  - Sustainability

Identity and Access Management

- For using the AWS services, you should grant your applications and users access to the resources in your AWS accounts

- Because you are running more workloads on the AWS, you need to robust the identity management and permissions in place for ensuring that the right individual have access to the correct resources under the correct conditions

- AWS provides a large selection of capabilities that helps you to manage your machine and human identities and also their permissions. The best practices for these abilities divided into two main areas:

Identity and Access Management

1. **Identity Management**

- There are two kinds of identities you must manage while approaching the operating secure AWS workloads:

  - **Human Identities**: The administrators, operators, developers, and consumers of your applications need an identity for accessing your AWS environments and applications. These can be your organisation's members or the external users with whom you can collaborate and who communicate with your AWS resources through the web browser, mobile app, client application, or interactive command-line tools

  - **Machine Identities:** Your workload applications, components, and operational tools need an identity for making the requests to the AWS services, such as for reading data. These identities contain machines that are running in your AWS environment, for example, AWS Lambda functions or Amazon EC2 instances

Identity and Access Management

2. Permissions Management

- Manage permissions for controlling the access to the machine and human identities requiring a access to AWS and your workloads. Permissions helps in controlling that who can access what, and under what conditions

- Set permissions to particular individual and machine identities for granting the access to a particular service actions on a particular resources

- In addition, specify conditions that should be true for access that is being granted. For instance, you can enable the developers for creating a new Lambda functions, but only in a particular Region

Detective Controls

- Detection has two parts: unexpected detection or unwanted configuration changes; and detecting an unexpected manner
- In an application delivery lifecycle, the first can take place at various places. You can check for unwanted configuration before a workload is launched by implementing checks in the CD/CI pipelines or source control by using infrastructure as code
- Then, you deploy a workload in production and non-productive environments, and check the configuration by using open-source, native AWS, or AWS Partner tools
- These checks may be a configuration that does not fulfil the safety principles or best practices or maybe changes made between a test and a deployed configuration
- For a running application, you can check if the configuration has changed unexpectedly, including external of a known deployment or automatic scaling event

Detective Controls

- For the other part of detection, unexpected behaviour, you can use tools to warn on an increase in a specific type of API call
- When unexpected and potentially unauthorised or malicious behaviour occurs within your AWS accounts, you can be alerted using Amazon GuardDuty
- You should also explicitly monitor for changing API calls that you do not expect to be used in your workload, as well as API calls that change the safety status
- Detection allows you to recognise a potential security misconfiguration, threat, or unusual manner
- It is an important part of the security lifecycle and can be used for a quality process, compliance or a legal obligation, as well as to efforts respond and threats identify

Detective Controls

- There are different types of detection methods. For instance, the logs can be analysed for exploits that are being used from your workload
- To ensure that you are meeting internal and external policies and needs, you should commonly analysis the detection methods related to your workload
- Automated warnings and notifications should be based on defined environments to allow your teams or tools to investigate
- These methods are necessary reactive factors that can help your organisation identify and understand the abnormal activity scope

Protecting Network

- Users can be located anywhere, between your workforce and your customers. You need to move away from the traditional models of trusting anyone as well as anything who has access to your network

- You can use the Zero Trust approach when you follow the principle of applying protection to all layers

- Zero-trust security is a model in which application components or microservices are measured separate from each other, and no microservices or component trusts another

- **Create network layers:**

  - Components like EC2 instances, Lambda functions, and RDS database clusters that share accessibility needs can be segmented into layers created by the subnet

  - For instance, RDS database cluster in a VPC that does not require Internet access should be placed in a subnet with no path to or from the Internet

Protecting Network

(Continued)

- **Control traffic at all layers:**
  - Examine the connectivity requirements of each component while architecting your network topology
  - For example, if a component needs Internet accessibility (inbound and outbound), edge services, VPCs, and connectivity to external data centers
- **Implement inspection and protection:**
  - Inspect and filter your traffic on each layer. You can examine your VPC configurations for possible involuntary access using the VPC Network Access Analyzer
  - You can define your network access requirements and recognize possible network paths that do not meet them

Protecting Network

(Continued)

- A web application firewall can help to secure components transacting via HTTP-based protocols from common attacks

- **Automate network protection:**

  - Automated security systems to provide a self-defence network based on threat intelligence and abnormality detection

  - For instance, prevention tools and intrusion detection can adapt to existing threats and reduce their impact

Data Encryption

- AWS provided the capability to add a security layer to your data at rest in the cloud, providing efficient and scalable encryption elements. These includes:
  - Most AWS services, like Amazon Redshift, Amazon S3, Amazon EBS, Amazon RDS, AWS Lambda, Amazon SageMaker, and Amazon ElastiCache, have data available at rest encryption capabilities
  - Server-side encryption (SSE) encrypts message queues for the transfer of sensitive data for Amazon SQS
  - Dedicated, hardware-based cryptographic key storage using AWS CloudHSM enables you to help fulfil your compliance needs
  - Flexible key management options, such as AWS Key Management Service, enable you to have the choice of having AWS manage your encryption keys or allowing you to have full control over your own keys

Module 12: Cost Optimisation Pillar

Introduction

- The AWS Well-Architected Framework helps you to understand the decisions you make when creating a workload on AWS
- The framework includes architectural best practices for operating as well as designing reliable, efficient, secure, and cost-effective workloads in the cloud
- It represents a way to measure your architectures continually against best practices and classify the areas for improvement
- The framework is based on six pillars:
  - Security
  - Operational Excellence
  - Cost Optimisation
  - Reliability
  - Sustainability
  - Performance Efficiency

Cost-Optimising Compute

- AWS allows you to take control of costs and consistently optimise your spending while building modern, scalable applications to fulfil your requirements

- The breadth of services and pricing options of AWS offers the flexibility to effectively manage your costs and still maintain the capacity and performance you need

- AWS is dedicated to helping the customers to succeed at the highest possible saving potential. During this time of crisis, it will work to create a plan that would satisfy your financial requirements

Choose the right pricing models

Match Capacity with Demand

Cost-Optimising Compute

Choose the Right Pricing Models

1. **Use Reserved Instances (RI) to reduce Amazon Redshift, Amazon RDS, Amazon ElastiCache, and Amazon OpenSearch Service costs**

- For several services, such as Amazon RDS and Amazon EC2, you can invest in reserved capacity. Reserved instances can save up to 72% over equivalent on-demand capability

- Reserved Instances come in three options: AURI (all-up-front), PURI (partial-up-front), and NURI (no-up-front)

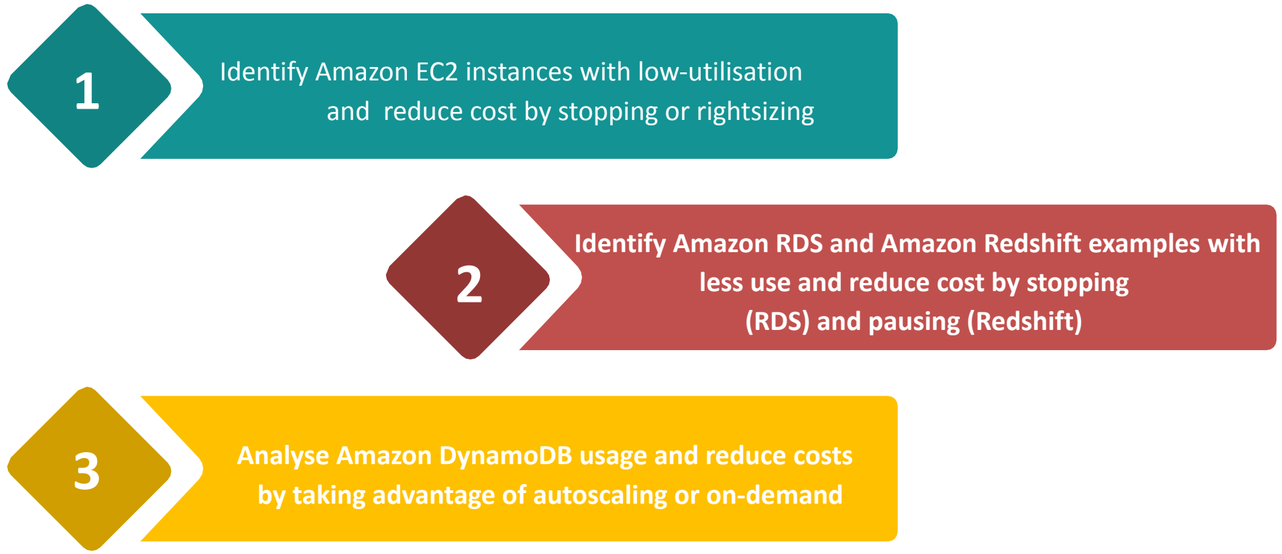
- Use the suggestions provided in the AWS Cost Explorer RI Purchase Recommendations, based on your use of Amazon Redshift, Amazon ElastiCache, Amazon RDS, and Amazon OpenSearch service

2. **Amazon EC2 Cost Savings**

- Use Amazon Spot Instances to save the cost of EC2 or Compute Savings Plans to reduce EC2, Lambda cost, and Fargate

Cost-Optimising Compute

Match Capacity with Demand



1. Identify Amazon EC2 instances with low-utilisation and reduce cost by stopping or rightsizing

- Use AWS Cost Explorer Resource Optimisation to get a report EC2 instances that are both underused or idle. Stopping or reducing these situations can help you save money. Use the AWS Operations Conductor to resize the EC2 instances automatically (based on the suggestions report from Cost Explorer)

Cost-Optimising Compute

2. **Identify Amazon RDS and Amazon Redshift examples with less use and reduce cost by stopping (RDS) and pausing (Redshift)**

- Use the Trusted Advisor Amazon RDS Idle DB instances check to recognise DB instances that have not had any connection for the past 7 days

- Use the Trusted Advisor Underutilized Redshift clusters check for Redshift to find clusters with no connections for the last 7 days and less than 5% cluster wide average CPU utilisation for 99 percent of the time

Cost-Optimising Compute

3. **Analyse Amazon DynamoDB usage and reduce costs by taking advantage of autoscaling or on- demand**

- Examine your DynamoDB usage in CloudWatch by monitoring two metrics, ConsumedWriteCapacityUnits as well as ConsumedReadCapacityUnits
- Use the AutoScaling feature to scale automatically (in as well as out) your DynamoDB table

Module 13: Operational Excellence Pillar

Introduction

- The operational excellence pillar defines how the organisation is supporting the objectives of the business, individuals' capability of running workloads effectively, gaining insight into their operations, and continually enhancing the supporting procedures and processes for delivering the business value

- Operational excellence in the cloud includes four areas:

Introduction

- The organisation’s leadership describes the objectives of the business. The organisation should understand needs and priorities and utilise these for organising and conducting work for supporting the business outcomes’ achievement

- Your workload should emit the information needed for supporting it. Implementing the services to allow the deployment, integration, and delivery of the individuals’ workload will allow an increased flow of beneficial changes into the production using automating of repetitive processes

- There may be risks inherent in the workload’s operation. You should understand those risks and make an informed decision for entering the production. Your teams should be able for supporting your workload

Introduction

- Operational and business metrics are derived from the desired business outcomes, will allow you to understand the workload health, your operations activities, as well as respond to the incidents

- Your priorities will change because your business requirements and business environment will be changed

- Use these as a feedback loop to continuously drive enhancement for the organisation and the operation of the workload

CloudFormation

- AWS CloudFormation is a service that helps you in modelling and setting up the AWS resources so that you can spend less time managing those resources and more time concentrating on your applications that are running in the AWS

- You can create a template describing all the resources of AWS that you want (such as Amazon RDS DB instances or Amazon EC2 instances), and CloudFormation takes care to provision and configure those resources. You do not require to individually create and configuring the AWS resources

Simplify Infrastructure Management

- For a scalable web application that provides a backend database, you might utilise an Auto Scaling group, and Amazon Relational Database Service database instance and an Elastic Load Balancing load balancer

- You might utilise each individual service for provisioning these resources, and then you can create the resources, you would have to configure them for working together. These tasks can add the time and complexity before an application even gets started

CloudFormation

Quickly Replicate Your Infrastructure

- If the application needs further availability, then you might replicate it in several regions so that if one region becomes unavailable, users can still utilise the application in another region

- The challenge with replicating your application is that it further needs you to replicate your resources

Easily Control and Track Changes to Your Infrastructure

- You might have underlying resources that you want for upgrading incrementally in some cases. For instance, you might change to the higher-performing type of instance in the Auto Scaling launch configuration so that you can decrease the maximum amount of instances in the Auto Scaling group

- If problems are occurring after completing the update, you might require to move back your infrastructure to the original settings. For doing it manually, you not only need to remember which resources were changed, moreover but you also need to know what were the original settings

CodeCommit

- It is a highly scalable, secure, managed source control service hosting the private Git repositories
- It helps to make it easier for teams to securely collaborate on the code with the contributions encrypted in transit as well as at rest. It also eliminates the requirement of managing your own source control system
- You can also use the CodeCommit for storing anything from code to the binaries. It supports the standard Git functionality, as a result, it works seamlessly with the existing Git-based tools

CodeCommit

Benefits

1. **Fully Managed:** AWS CodeCommit helps eliminate the need to maintain, host, back up, and scale your own source control servers. Automatically, the service scales for meeting the growing requirements of your project

2. **Secure:** It automatically encrypts your files in transit as well as at rest. CodeCommit is integrated with the IAM that allows you to customise the user-specific access to your repositories

3. **High Availability:** AWS CodeCommit has a highly redundant, scalable, and durable architecture. The service is designed to make the repositories highly accessible and available

4. **Collaborate on Code:** It assists you in collaborating on the code with team members through pull requests, merging, and merging. You can implement workflows, including the code feedback and reviews by default, and control who can make the changes to particular branches

CodeDeploy

- It is a completely managed deployment service automating the software deployments to the various computing services. For example, AWS Fargate, Amazon EC2, AWS Lambda, and the on- premises servers

- AWS CodeDeploy helps in making it easier for you to release the new features quickly, helps you to avoid the downtime throughout the application deployment, and handles the complexity of updating the applications

- It helps to automate the software deployments that eliminate the requirement for error-prone manual operations. The service scales for matching the deployment requirements

CodeDeploy

Benefits

1. **Automated Deployments:** AWS CodeDeploy completely automates the software deployments that allow you to deploy the reliably and quickly

2. **Centralised Control:** AWS CodeDeploy enables you to easily launch and tracking the status of the application deployments using AWS CLI or AWS Management Console

3. **Minimise Downtime:** AWS CodeDeploy helps in maximising the application availability throughout the software deployment process

4. **Easy to Adopt:** AWS CodeDeploy is platform and language agnostic, working with any application and providing the same experience whether you are deploying to AWS Fargate, Amazon EC2, or AWS Lambda

CodePipeline

- It is a completely managed continuous delivery service helping you in automating the release pipelines for quick and reliable applications and the infrastructure updates

- It also automates the test, builds, and deploy the phases of the release process every time there is a code change on the basis of the release model you define

- It allows you to quickly and reliably deliver the aspects and updates. You can easily integrate the AWS CodePipeline with third-party services, for example, GitHub, or with the custom plugin. With the AWS CodePipeline, you just pay for what you are using. There are no long-term commitments or upfront fees

CodePipeline

Benefits

1. **Rapid Delivery:** AWS CodePipeline helps in automating the software release process, enabling you to quickly release new aspects to the users. Using CodePipeline, you can rapidly iterate on the feedback and get new aspects to the users faster

2. **Get Started Fast:** Using AWS CodePipeline, you can immediately start to model the software release process. There are no servers for provisioning or setting up

3. **Configurable Workflow:** AWS CodePipeline enables you to model the various stages of the software release process with the use of the console interface, the AWS CLI, the AWS SDKs, or the AWS CloudFormation. You can easily specify the tests for running and customising the steps for deploying the application and its dependencies

4. **Easy to Integrate:** AWS CodePipeline can easily be extended for adapting to the particular requirements. You can use the pre-built plugins or the custom plugins in any step of the release process

AWS Systems Manager

- AWS Systems Manager (formerly called SSM) is an AWS service that the individual can use for viewing and controlling the infrastructure on the AWS
- With the Systems Manager console, an individual can view the operational data from several AWS services and automate the operational tasks over the AWS resources
- Systems Manager helps in maintaining the security and compliance by scanning the managed nodes and reporting on any of the policy violations which is detected
- Any machine configured for Systems Manager is known as a managed node. Systems Manager is supporting the edge devices, Amazon Elastic Compute, and Cloud instances, and on-premises servers and virtual machines that include the VMs in different cloud environments
- Systems Manager is supporting the Windows Server, Raspberry Pi OS, macOS, and several distributions of Linux for the operating systems

Keep Learning ☺️