AWS Solution Architect

Compute and Related Features

Learning Objectives

By the end of this lesson, you will be able to:

- Create an AMI image for fast deployment to streamline the deployment process and enhance reliability
- Launch and connect to a Windows instance for effective disaster recovery strategy
- Deploy various load balancers to improve scalability to achieve a scalable, resilient, and high-performing infrastructure
- Create a routing request in ALB to enable path-based routing, optimizing performance, security, and scalability

TECHNOLOGY

Introduction to Amazon EC2

simplilearn

Elastic Compute Cloud (EC2)

Amazon Elastic Compute Cloud (Amazon EC2) is a web service that provides scalable computing capacity in the Amazon Web Services (AWS) cloud.

Benefits of EC2

- Elastic Web-scale Computing
- Flexible Cloud Hosting Services
- AWS Integration
- Reliability and Security
- Low Cost
- Completely Controlled

- EC2 increases or decreases the storage capacity in minutes.
- It launches thousands of server instances simultaneously.

Benefits of EC2

- Elastic Web-scale Computing
- Flexible Cloud Hosting Services
- AWS Integration
- Reliability and Security
- Low Cost
- Completely Controlled

- EC2 launches numerous operating systems, instance types, and software in minutes.
- It allows the users to choose the memory, CPU, instance storage, and boot partition size that is best for their applications and OS.

Benefits of EC2

- Elastic Web-scale Computing
- Flexible Cloud Hosting Services
- AWS Integration
- Reliability and Security
- Low Cost
- Completely Controlled

EC2 is integrated with other AWS products, such as Amazon S3, Amazon RDS, and Amazon SQS, to provide a complete IT architecture solution.

Benefits of EC2

- Elastic Web-scale Computing
- Flexible Cloud Hosting Services
- AWS Integration
- Reliability and Security
- Low Cost
- Completely Controlled

- Amazon EC2 provides a highly reliable environment where replacement instances can be quickly and consistently deployed.
- Users can easily create secure and robust networks to run their Amazon EC2 instances using the Virtual Private Cloud (VPC).

Benefits of EC2

- Elastic Web-scale Computing
- Flexible Cloud Hosting Services
- AWS Integration
- Reliability and Security
- Low Cost
- Completely Controlled

- AWS charges the users in seconds. They only pay for what they use.
- The rates are lower than the existing on-premise infrastructure.

Benefits of EC2

- Elastic Web-scale Computing
- Flexible Cloud Hosting Services
- AWS Integration
- Reliability and Security
- Low Cost
- Completely Controlled

- The users have complete control over their instances. They have root access to all the instances.
- The users can use web service APIs to stop their instance while keeping the data on their boot partition and resume it.

Amazon EC2 Storage

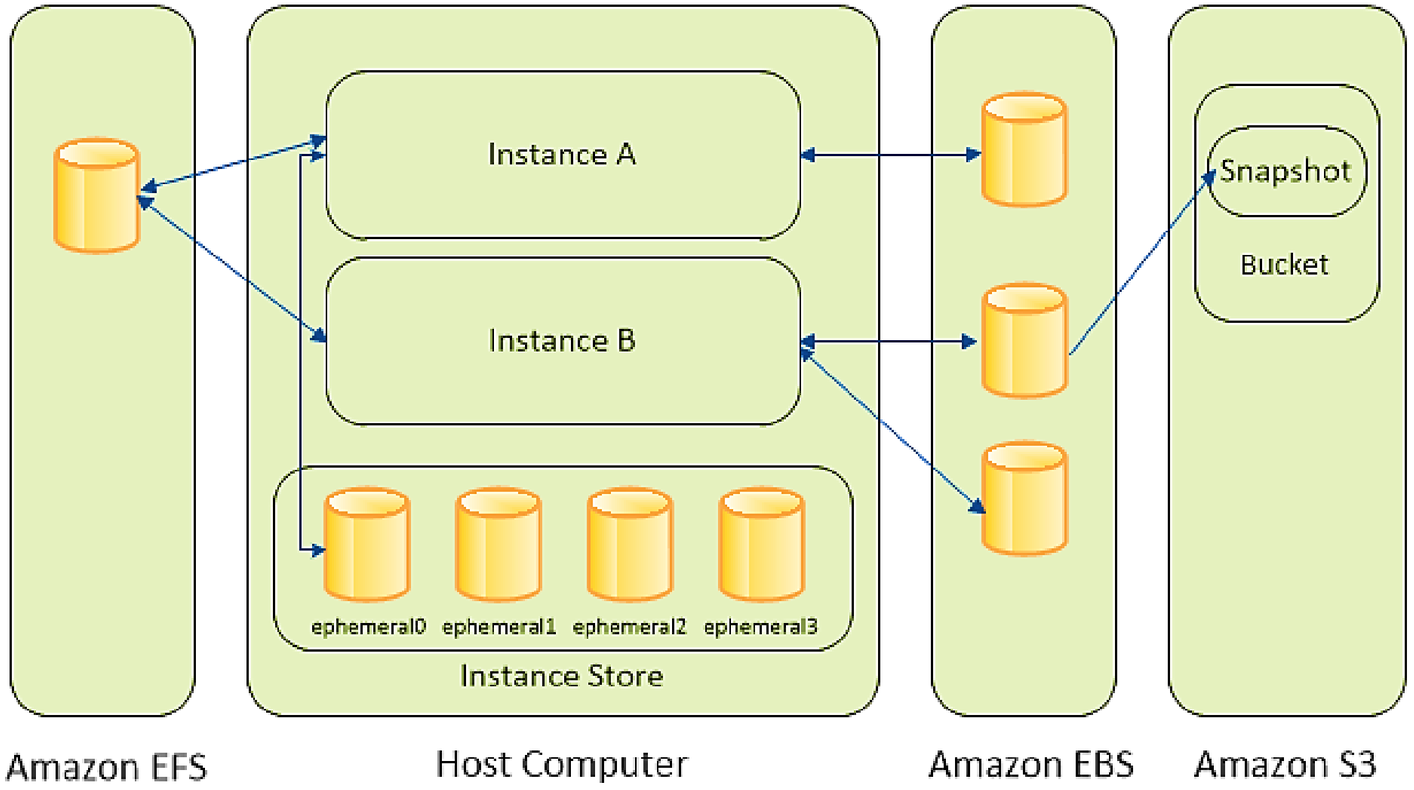
Amazon EC2 offers flexible, cost-effective, and simple data storage options for instances.

The storage options include the following:

- Amazon EBS with EC2
- Amazon EC2 instance store
- Amazon EFS with EC2
- Amazon S3 with EC2

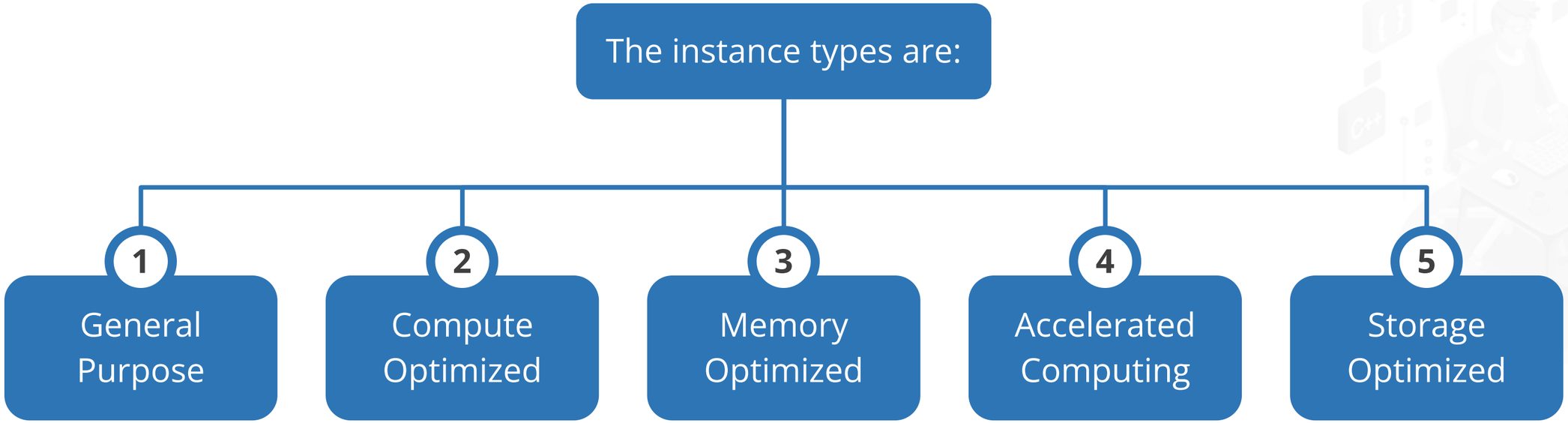
Amazon EC2 Storage: Linux Instances

The following image depicts the relationship between the storage options and Linux instances:



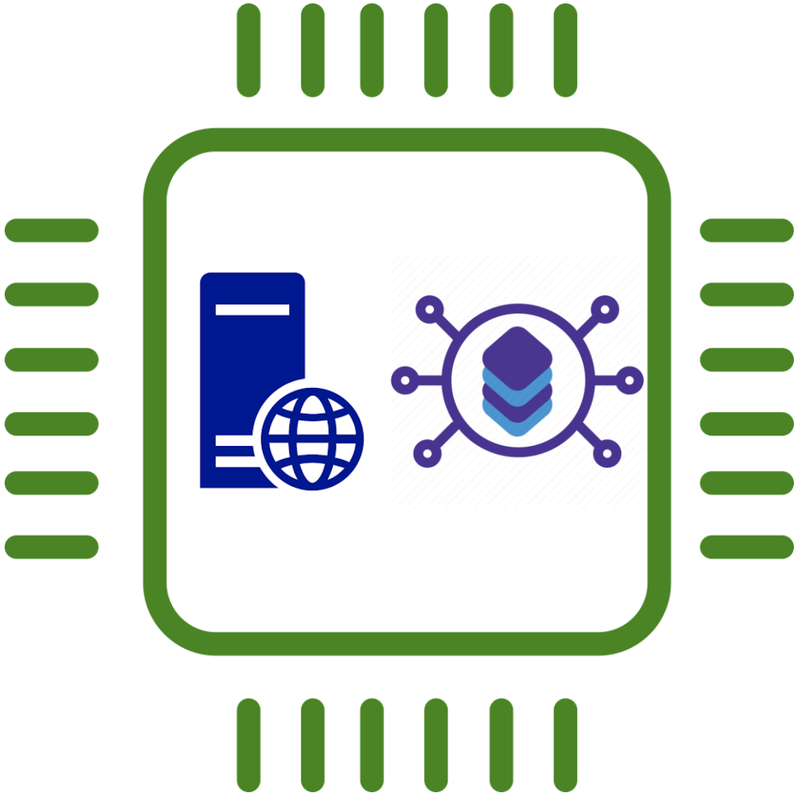
Instances

An instance is a virtual server in the AWS cloud. Amazon EC2 offers various instance types, allowing users to select the CPU, memory, storage, and networking resources required to run their applications.



General Purpose Instances

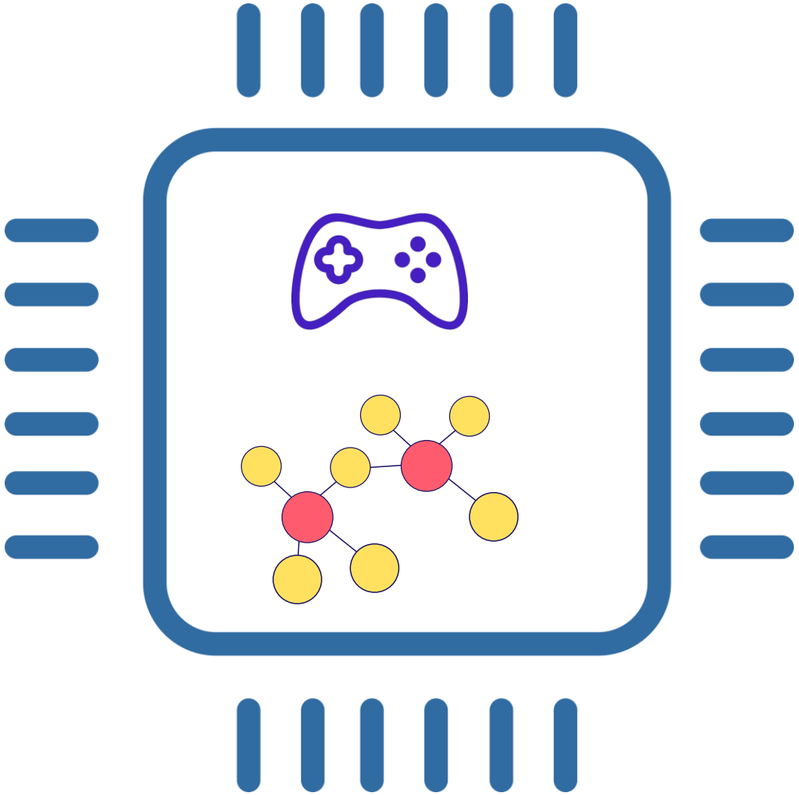
These instances are virtual machines offering balanced compute, memory, and networking for diverse workloads.



- These instances are useful for applications like web servers and code repositories that require such resources in equal parts.
- The examples are Mac, T4g, T3, T3a, T2, M6g, M6i, M6a, M5, M5a, M5n, M5zn, M4, and A1 instances.

Compute Optimized Instances

These instances are suitable for compute-intensive applications that benefit from high-performance processors.



- These instances are useful for batch processing workloads, dedicated gaming servers and ad server engines, scientific modeling, and so on.
- The examples are C7g, C6g, C6gn, C6i, C6a, Hpc6a, C5, C5a, C5n, and C4 instances.

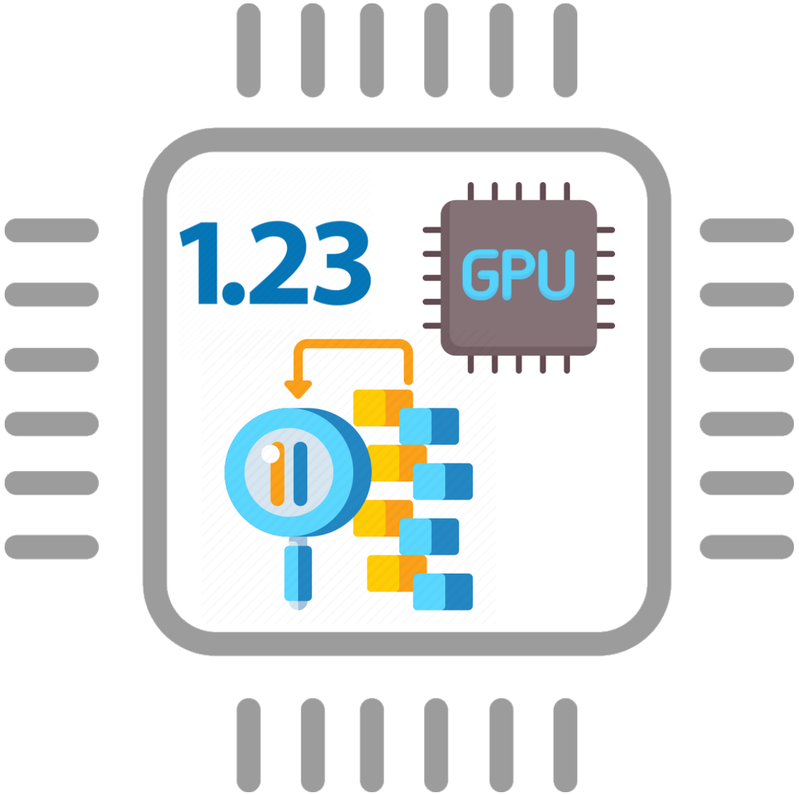
Memory Optimized Instances

These instances are intended to provide rapid performance for workloads that process large data sets in memory.

- These instances are useful for databases, electronic design automation (EDA) workloads, real-time analytics, and real-time caching servers.
- The examples are R6g, R6i, R5, R5a, R5b, R5n, R4, X2gd, X2idn, X2iezn, X1e, X1, high memory, and z1d instances.

Accelerated Computing Instances

These instances use hardware accelerators or co-processors to execute tasks more effectively than software running on CPUs.



- These instances are useful for performing tasks such as floating-point number calculations, graphic processing, data pattern matching, and so on.
- The examples are P4, P3, P2, DL1, Trn1, Inf1, G5, G5g, G4dn, G4ad, G3, F1, and VT1 instances.

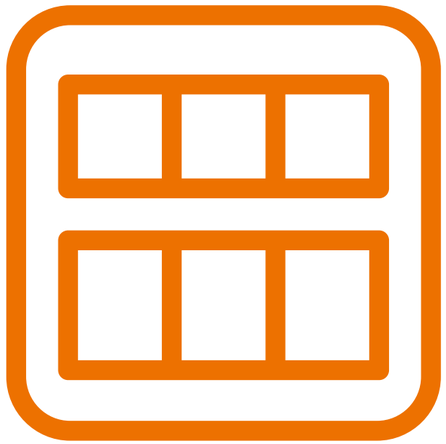
Storage Optimized Instances

These instances are designed to provide applications with tens of thousands of low-latency, random I/O operations per second (IOPS).

- These instances are useful for workloads that demand high sequential read and write access to large data sets on local storage.
- The examples are Im4gn, Is4gen, I4i, I3, I3en, D2, D3, D3en, and H1 instances.

Amazon Machine Images (AMI)

An Amazon Machine Image (AMI) is an AWS-supported and maintained image that contains the information required to launch an instance.



- Users must specify an AMI when they launch an instance.
- Users can launch multiple instances from a single AMI when they require multiple instances with the same configuration.
- Users can use different AMIs to launch instances when they require instances with different configurations.

Amazon Machine Images (AMI)

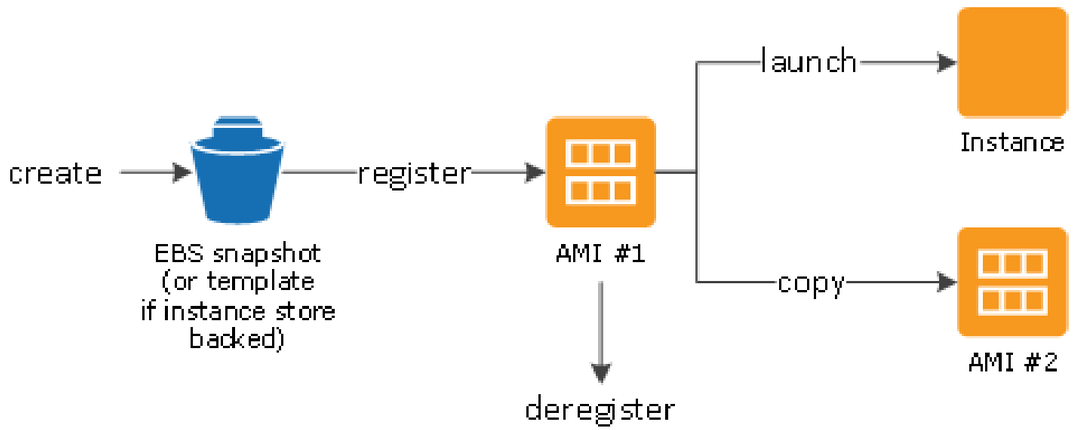
AMI

AMI is a virtual instance that includes:

- A template for the root volume of the instance
- Launch permissions to control AMI launch instances
- A block device mapping that specifies volumes to attach to the instance

AMI Lifecycle

The following image represents the AMI lifecycle:



- After creating and registering an AMI, users can use it to launch new instances.
- Users can copy an AMI within the same AWS region or to different AWS Regions.
- When an AMI is no longer required, users can deregister it.

Amazon Linux AMI

It is a Linux image offered by AWS that is both officially supported and continuously maintained.

The following are the features of the Amazon Linux AMI:

Updates regularly to include the latest components

Offers components at no additional charge to Amazon EC2 users

Allows repository access to MySQL, PostgreSQL, Python, Ruby, and Tomcat

Provides a secure, stable, and high-performance execution environment

Includes packages that enable easy integration with AWS services

AMI Distribution

An AMI can be kept private, shared with a specific list of AWS accounts, or made public.

AMI Types

An AMI can be categorized as either an EBS-backed AMI or an instance store-backed AMI.

Characteristic	Amazon EBS-backed AMI	Amazon instance store-backed AMI
Boot time for an instance	Usually less than 1 minute	Usually less than 5 minutes
Size limit for a root device	64 TiB	10 GiB
Root device volume	EBS volume	Instance store volume
Data persistence	By default, the root volume is deleted when the instance terminates. Data on any other EBS volumes persists after instance termination by default.	Information stored on instance store volumes remains intact solely throughout the duration of the instance's existence.

Creating an AMI Image

Duration:10 min

Problem Statement:

You have been assigned a task to create an AMI image from an EC2 instance for creating a backup or launching additional instances with the same configuration.

Outcome:

You can create an AMI from an EC2 instance, providing a reliable backup solution and a straightforward method for scaling and launching new instances with the same setup.

Note: Refer to the demo document for detailed steps: 01_Creating_an_AMI_image

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create an AMI image

Automating App Installation on EC2 Instance

Duration:10 min

Problem Statement:

You have been assigned a task to automate application installation during EC2 instance launch in AWS using User Data.

Outcome:

By automating application installation using user data scripts, the process of configuring and deploying EC2 instances becomes more efficient, reliable, and scalable, leading to improved operational effectiveness and reduced time to market for new applications and services.

Note: Refer to the demo document for detailed steps:
02_Automating_App_Installation_on_EC2 Instance

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Install apps when launching an instance

Quick Check

You need to launch a web server on AWS with specific software pre-installed. Which Amazon EC2 feature allows you to select a pre-configured virtual machine with the necessary software stack?

A. EC2 Instance Types

B. Elastic Block Store (EBS)

C. Auto Scaling

D. Amazon Machine Image (AMI)

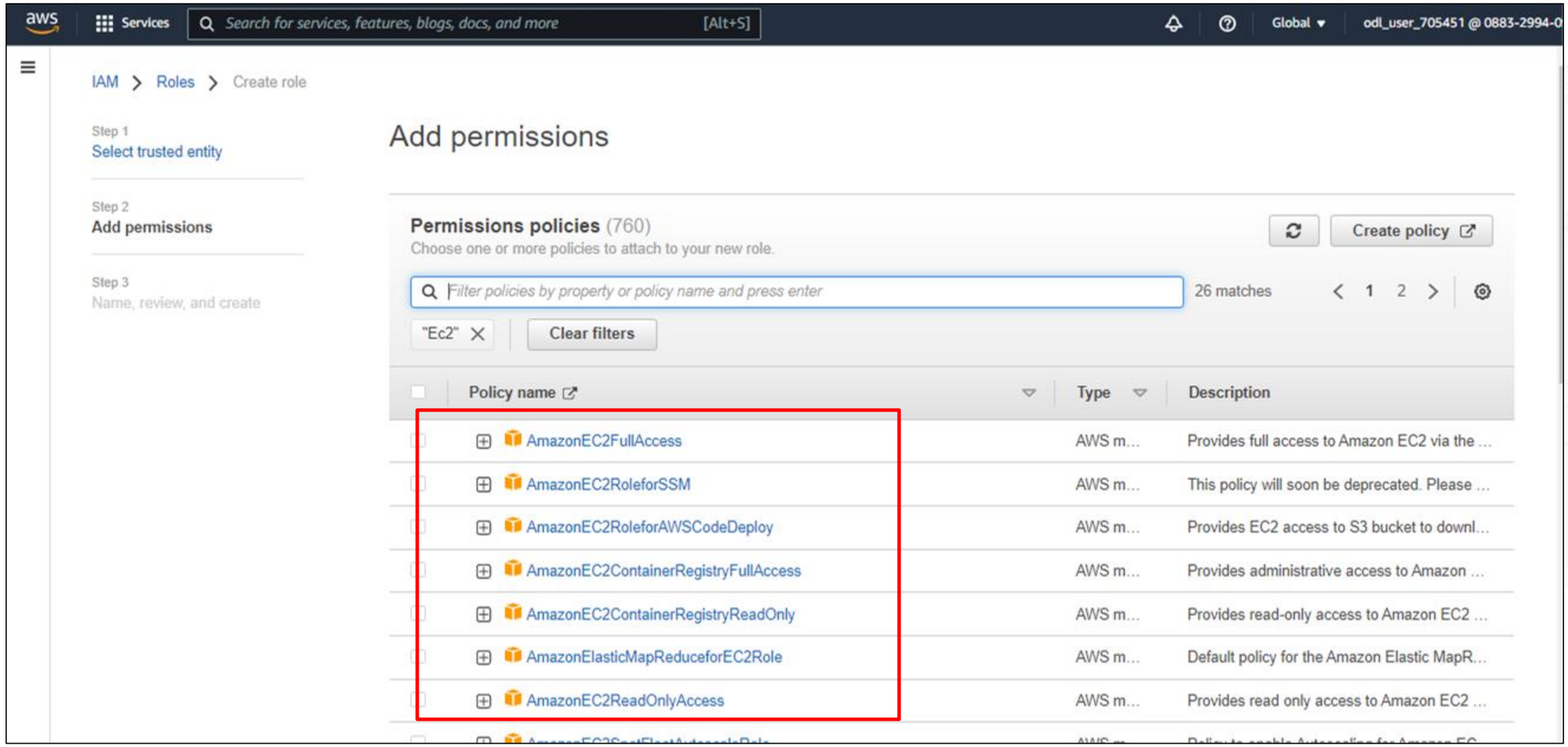
EC2 Instance Metadata

IAM Roles

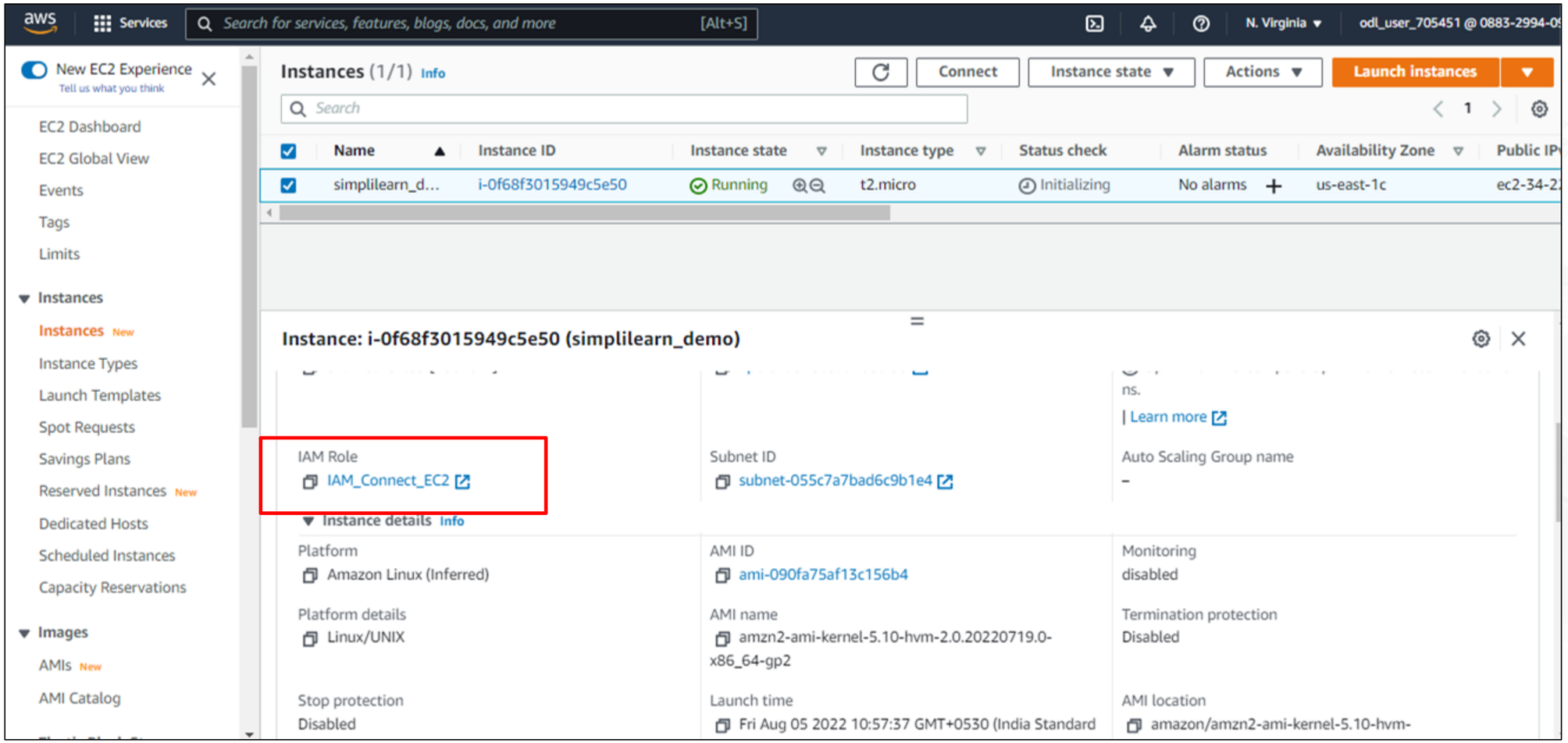
An IAM role is included with an instance profile on Amazon EC2. The IAM console automatically creates an instance profile when users create an IAM role, giving it the same name as the role it belongs to.

Connection Between IAM and EC2

The connection between IAM Roles and EC2 instances by applying necessary permissions:



Connection Between IAM and EC2



Using IAM Roles to Access S3 Bucket

Duration:10 min

Problem Statement:

You have been assigned a task to securely access Amazon S3 (Simple Storage Service) buckets from an EC2 instance using IAM (Identity and Access Management) roles.

Outcome:

By implementing IAM roles for accessing S3 buckets from EC2 instances, you can achieve a secure, manageable, and scalable solution for accessing AWS resources.

Note: Refer to the demo document for detailed steps:
03_Using_IAM_Roles_to_Access_S3_Bucket

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create an IAM role
2. Connect IAM Profile to EC2
3. Validate access to the S3 bucket

Starting and Accessing a Windows Instance on EC2

Duration:10 min

Problem Statement:

You have been assigned a task launch a Windows-based EC2 instance on Amazon Web Services (AWS) and securely access it using the Remote Desktop Protocol (RDP).

Outcome:

This process ensures a secure and efficient setup of a Windows-based EC2 instance on AWS, ready for any required tasks or applications.

Note: Refer to the demo document for detailed steps:
04_Starting_and_Accessing_a_Windows_Instance_on_EC2

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Launch an EC2 instance
2. Access the EC2 instance via RDP

Quick Check

You need to securely allow your EC2 instance to access an S3 bucket without embedding credentials in your application code. Which approach should you use?

A. Store the S3 credentials in the EC2 instance metadata

B. Attach an IAM role to the EC2 instance

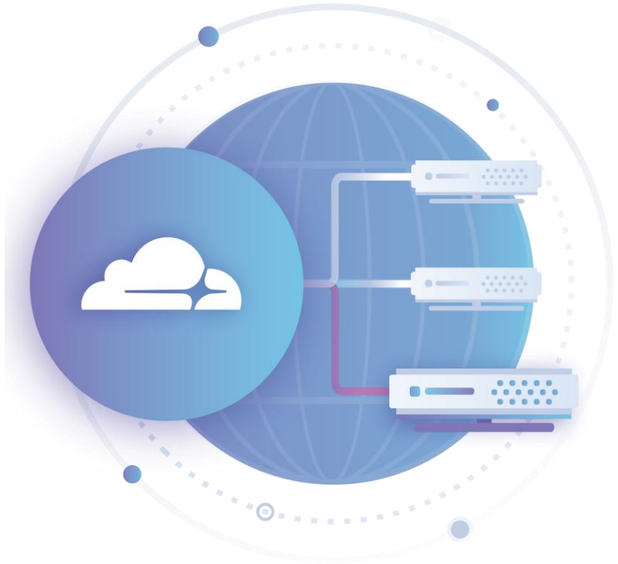
C. Configure access permissions in the S3 bucket policy manually

D. Use AWS Secrets Manager to store and retrieve S3 credentials

Elastic Load Balancer

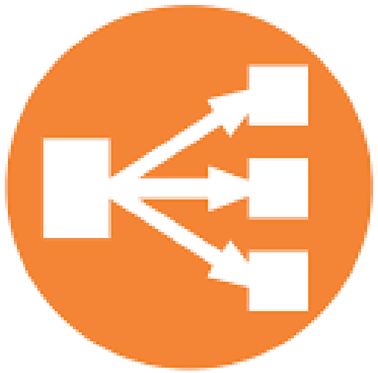
What Is Load Balancing?

Load balancing refers to the distribution of network traffic across multiple servers or instances of virtual machines that host the application.



Amazon Elastic Load Balancing (ELB)

ELB is a load balancing service offered by AWS.



It distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, and Lambda functions.

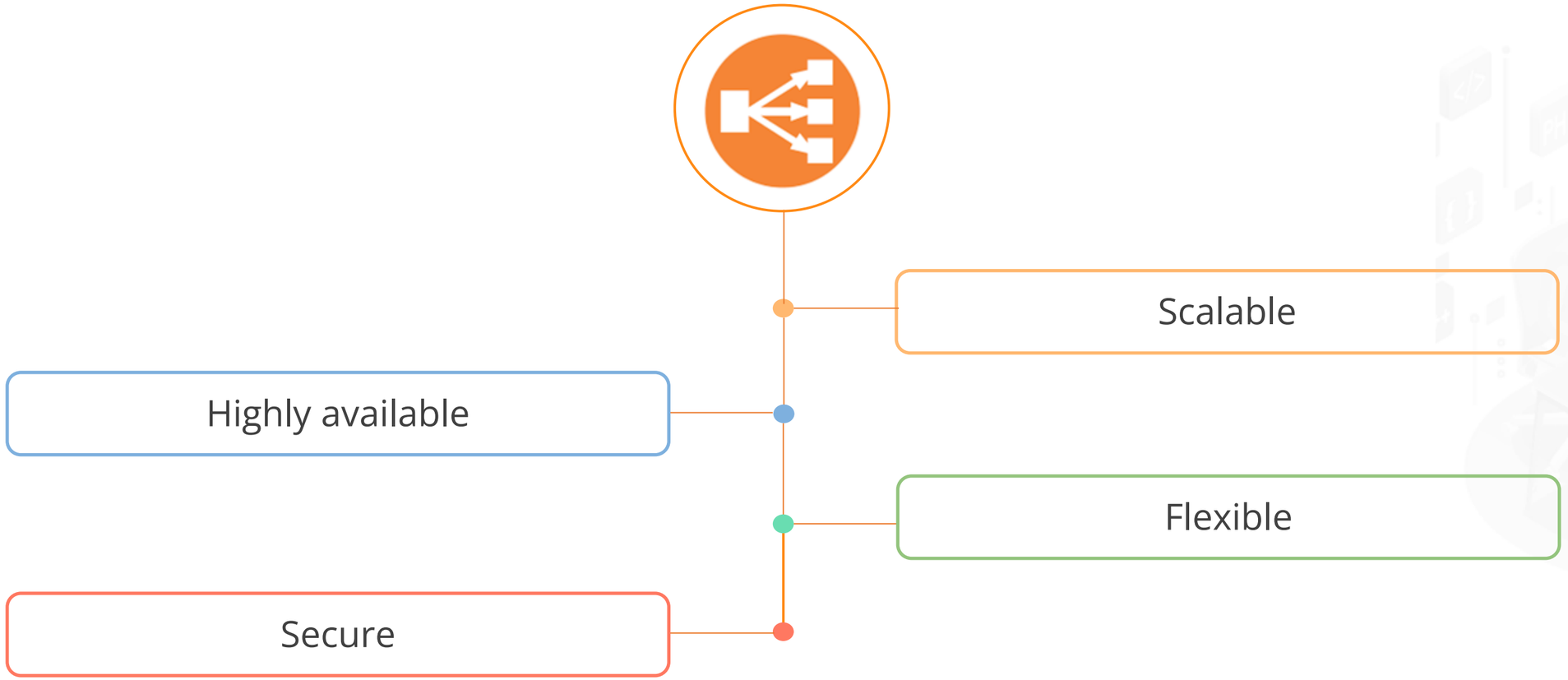
Importance of Load Balancing

Load balancing is needed to:

01	Prevent traffic overload on any server
02	Improve application responsiveness
03	Increase availability of applications
04	Avoid single point of failure in servers

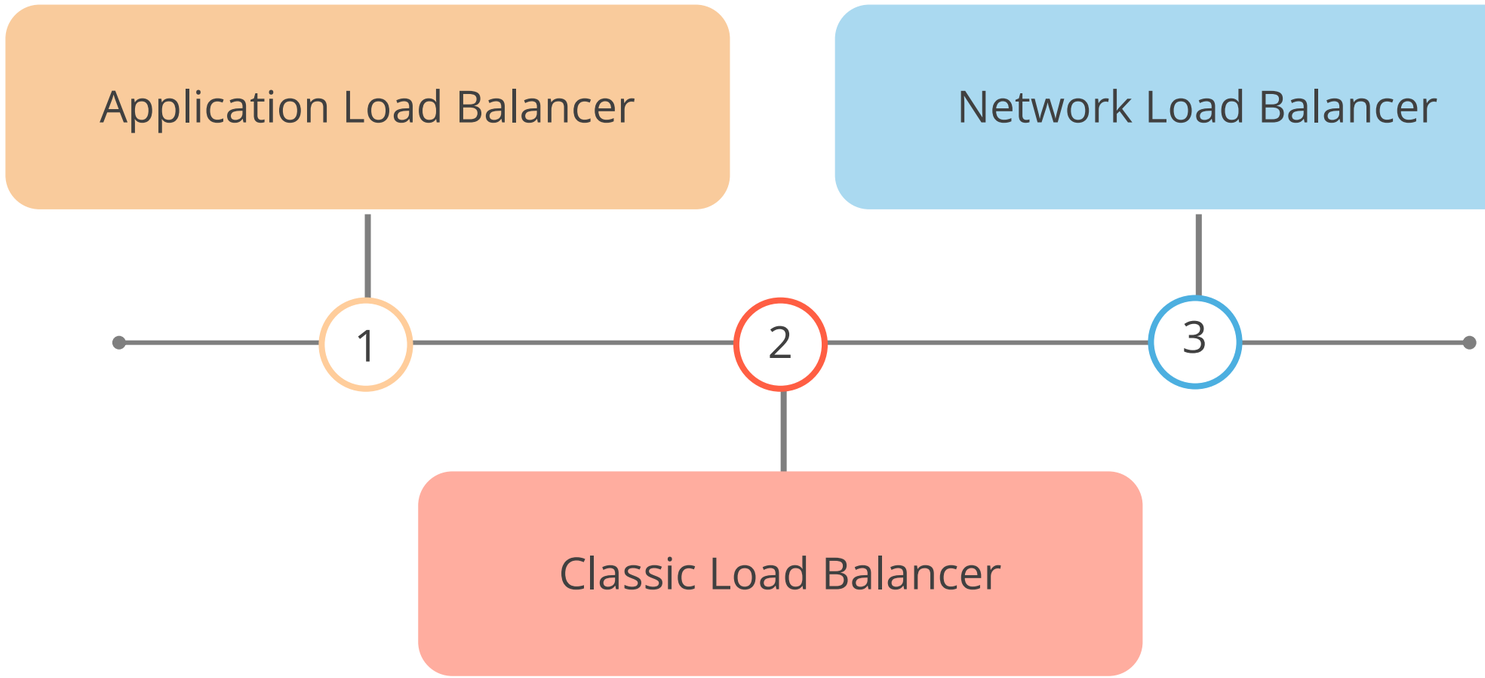
Benefits of Amazon ELB

Amazon ELB offers the following benefits:



Types of Amazon Load Balancers

Amazon ELB offers the following types of load balancers:



Application Load Balancer

01	It is used for load balancing of HTTP and HTTPS traffic.
02	It routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) based on the content of the request.
03	It operates on layer 7.

Network Load Balancer

01	It is used for load balancing of TCP, UDP, and TLS traffic.
02	It routes traffic to targets within Amazon Virtual Private Cloud (Amazon VPC) regardless of the content of the request.
03	It operates on layer 4.

Classic Load Balancer

01	It provides basic load balancing across multiple Amazon EC2 instances.
02	It is best suited for applications built on the EC2-Classic network.
03	It operates on both layer 4 and layer 7.

HTTP 503: Service Unavailable Error

HTTP 503: Service Unavailable Error

The **HTTP 503: Service unavailable error** indicates that either the load balancer or the registered instances are causing the error.

HTTP 503: Service Unavailable Error: Causes

The following are a few causes and solutions for the HTTP 503: Service unavailable error:

Cause 1:
Insufficient capacity in the load balancer to handle the request.

Solution 1:
This should be a transient issue and should not last more than a few minutes. If it persists, go to the AWS Support Center for assistance.

HTTP 503: Service Unavailable Error: Causes

The following are a few causes and solutions for the HTTP 503: Service unavailable error:

Cause 2:
There are no registered instances.

Solution 2:
A user needs to register at least one instance in every Availability Zone, where the load balancer is configured to respond. Verify this by looking at the **HealthyHostCount** metrics in CloudWatch. If a user is unable to ensure that an instance is registered in each Availability Zone, enabling cross-zone load balancing is recommended.

HTTP 503: Service Unavailable Error: Causes

The following are a few causes and solutions for the HTTP 503: Service unavailable error:

Cause 3:
There are no healthy instances.

Solution 3:
A user needs to ensure that they have healthy instances in every Availability Zone, where their load balancer is configured to respond. This can be verified by looking at the **HealthyHostCount** metric.

HTTP 503: Service Unavailable Error: Causes

The following are a few causes and solutions for the HTTP 503: Service unavailable error:

Cause 4:
The surge queue is full.

Solution 4:
A user needs to ensure that their instances have sufficient capacity to handle the request rate. This can be verified by looking at the **SpillOverCount** metric.

Setting Up a Web Server on an EC2 Instance

Duration:10 min

Problem Statement:

You have been assigned a task to set up a web server on an EC2 instance, making it accessible via its public IPv4 address.

Outcome:

You can deploy a basic web server and ensure it is accessible over the internet, demonstrating a foundational knowledge of AWS infrastructure and web server setup.

Note: Refer to the demo document for detailed steps:
05_Setting_Up_a_Web_Server_on_an_EC2_Instance

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create a VPC, a subnet, and an internet gateway
2. Create Route tables
3. Create an EC2 web server instance

Using a Classic Load Balancer to Distribute Traffic

Duration:10 min

Problem Statement:

You have been assigned a task to demonstrate the process of creating a Classic Load Balancer in AWS EC2 and deploying it to multiple instances in different availability zones.

Outcome:

The classic Load Balancer will efficiently distribute incoming traffic across multiple EC2 instances, enhancing the application's availability and reliability.

Note: Refer to the demo document for detailed steps:
06_Using_a_Classic_Load_Balancer_to_Distribute_Traffic

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create a security group
2. Launch instances with different availability zones
3. Create the Classic Load Balancer
4. Deploy the Classic Load Balancer to an EC2 instance

Configuring an Application Load Balancer

Duration:10 min

Problem Statement:

You have been assigned a task to configure an Application Load Balancer in AWS to distribute traffic across multiple EC2 instances for load balancing and redundancy.

Outcome:

The Application Load Balancer will efficiently manage and distribute incoming traffic to multiple EC2 instances, enhancing the application's availability, scalability, and performance.

Note: Refer to the demo document for detailed steps:
07_Configuring_Application_Load_Balancer

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create a target group
2. Launch EC2 instances
3. Configure the target group
4. Create a Load Balancer
5. Test the Load Balancer

Creating Route Requests in ALB

Duration:10 min

Problem Statement:

You have been assigned a task to set up routing requests in Amazon Web Services (AWS) using an Application Load Balancer (ALB).

Outcome:

The Application Load Balancer will efficiently route incoming requests based on defined rules, ensuring optimal traffic distribution and high availability for the application.

Note: Refer to the demo document for detailed steps: 08_Creating_Route_Requests_in_ALB

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Set up the prerequisites for EC2
2. Create a routing request in ALB

Creating a Network Load Balancer

Duration:10 min

Problem Statement:

You have been assigned a task to To create a Network Load Balancer (NLB) in the Amazon Web Services (AWS) environment for distributing traffic across the specified availability zones using multiple listeners and target groups.

Outcome:

The Network Load Balancer will effectively manage and distribute traffic, ensuring optimal performance, availability, and reliability of the application.

Note: Refer to the demo document for detailed steps:
09_Creating_a_Network_Load_Balancer

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Set up the Network Load Balancer

Quick Check

You notice that your application is intermittently returning an HTTP 503: Service Unavailable error. What could be a likely cause of this issue?

A. The server is returning a 200 OK response.

B. The load balancer is directing traffic to an unhealthy instance.

C. The DNS configuration is incorrect.

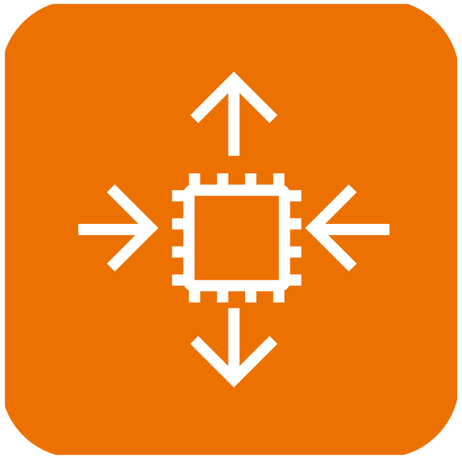
D. The database query is returning too many results.

ELB Logs

Auto Scaling

Auto Scaling

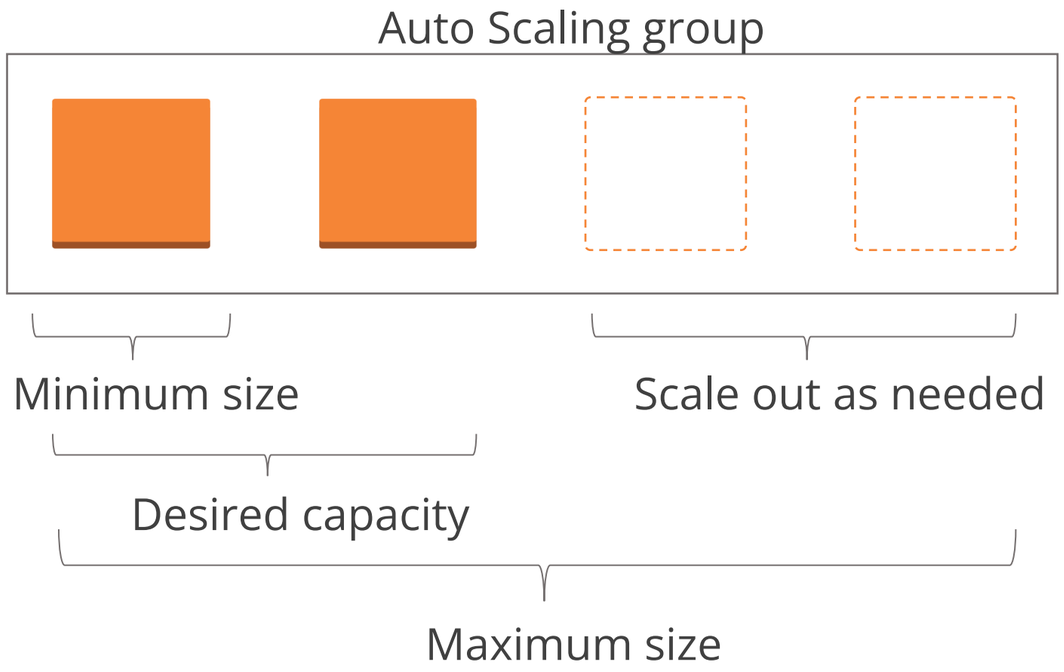
Amazon EC2 Auto Scaling helps the users to maintain application availability.



According to the conditions defined by the users, EC2 auto scaling allows them to automatically add or remove EC2 instances.

Auto Scaling Groups

A collection of EC2 instances is called an Auto Scaling group.



The users can specify the minimum number of instances in each group, and auto scaling ensures that the group never goes below the minimum size.

Auto Scaling

The difference between the load balancer and auto scaling are:

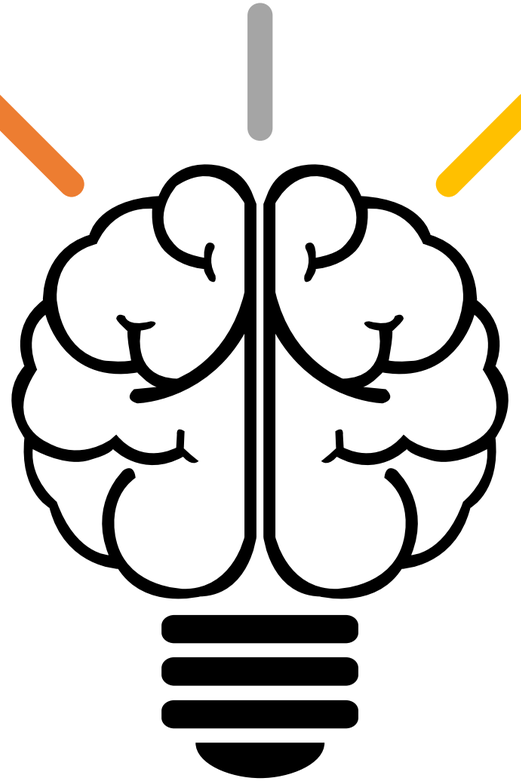
Load Balancer	Auto Scaling
Elastic load balancer only monitors each instance's health, distributes traffic, and connects each request to the right target groups.	The user's ability to specify policies based on predefined criteria is enabled by auto scaling, which enables multiple instances to work parallelly.

Auto Scaling: Features

Built-in scaling strategies

Automatic resource discovery

Predictive scaling

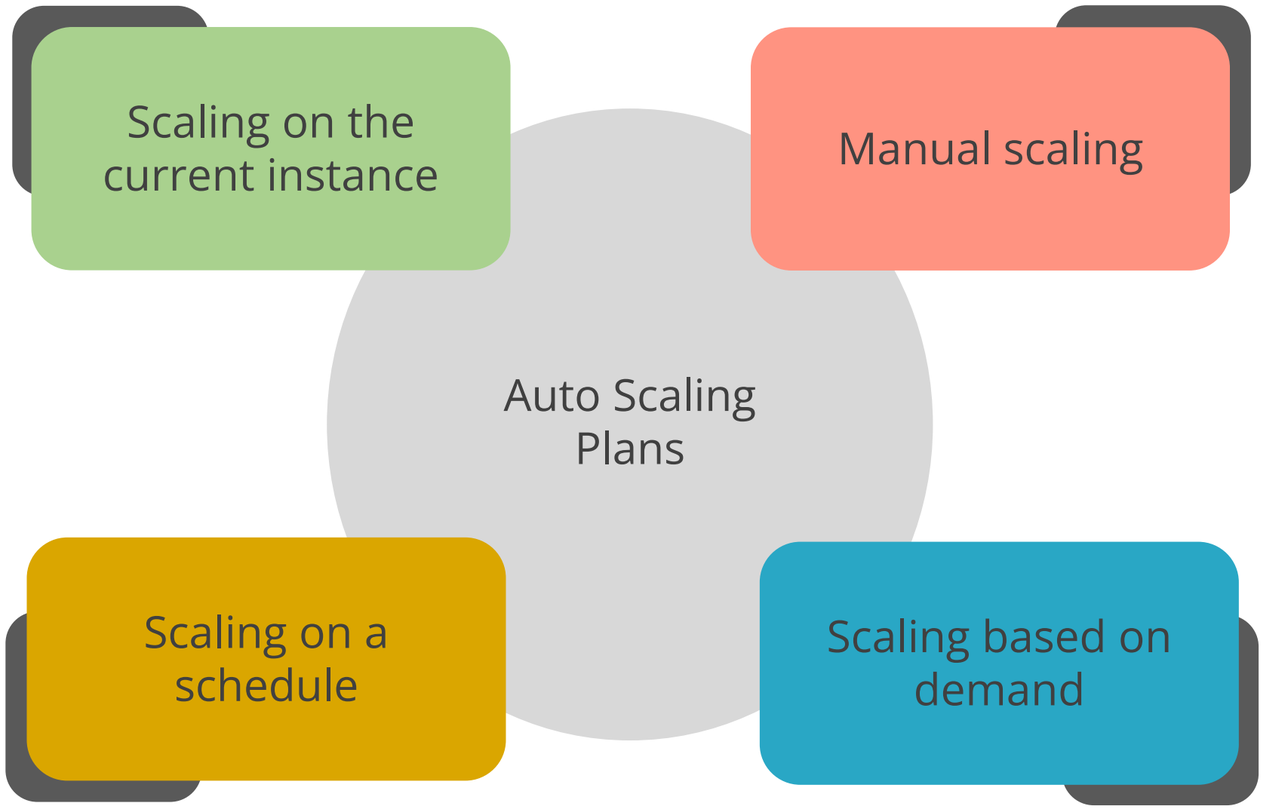


Unified scaling

Smart scaling policies

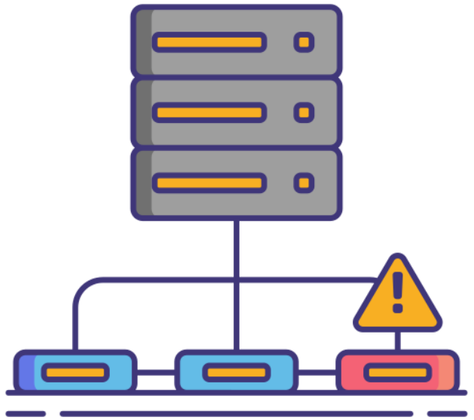
Auto Scaling Planning

Here are the four categories of auto scaling planning:



Auto Scaling Benefits

- Better fault tolerance
- Increased application availability
- Lower costs



Amazon EC2 auto scaling can determine the health of an instance. It can terminate the instance and replace it with a new one.

Auto Scaling Benefits

- Better fault tolerance
- Increased application availability
- Lower costs

Amazon EC2 auto scaling ensures that the application always has the right amount of computing and proactively provisions capacity with predictive scaling.

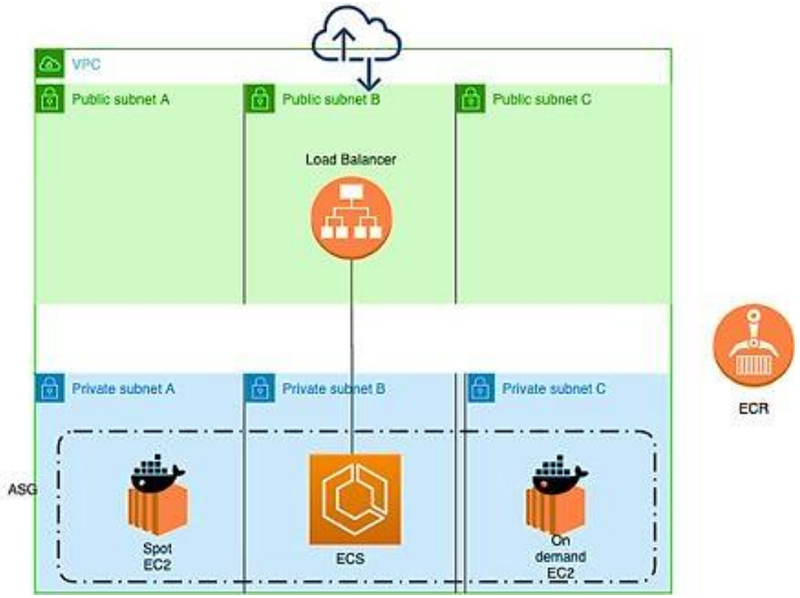
Auto Scaling Benefits

- Better fault tolerance
- Increased application availability
- Lower costs

Amazon EC2 auto scaling adds instances only when needed, and it can scale across purchase options to optimize performance and cost.

Fleet Management with Auto Scaling

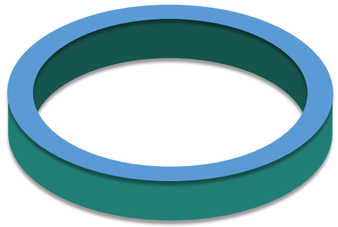
If the user's application runs on Amazon EC2 instances, the user has what is known as a fleet.



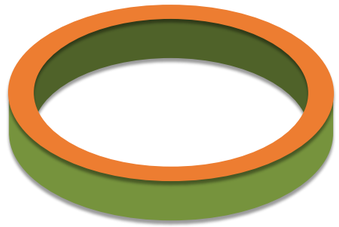
- The users can automate the management of a user’s fleet
- It is easy to set up.

Functions of Auto Scaling

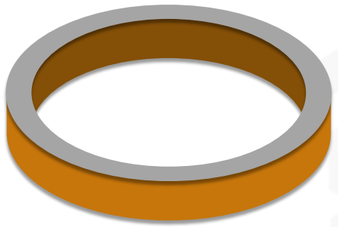
The three main functions that auto scaling performs to automate fleet management for EC2 instances are as follows:



Monitoring the health of running instances



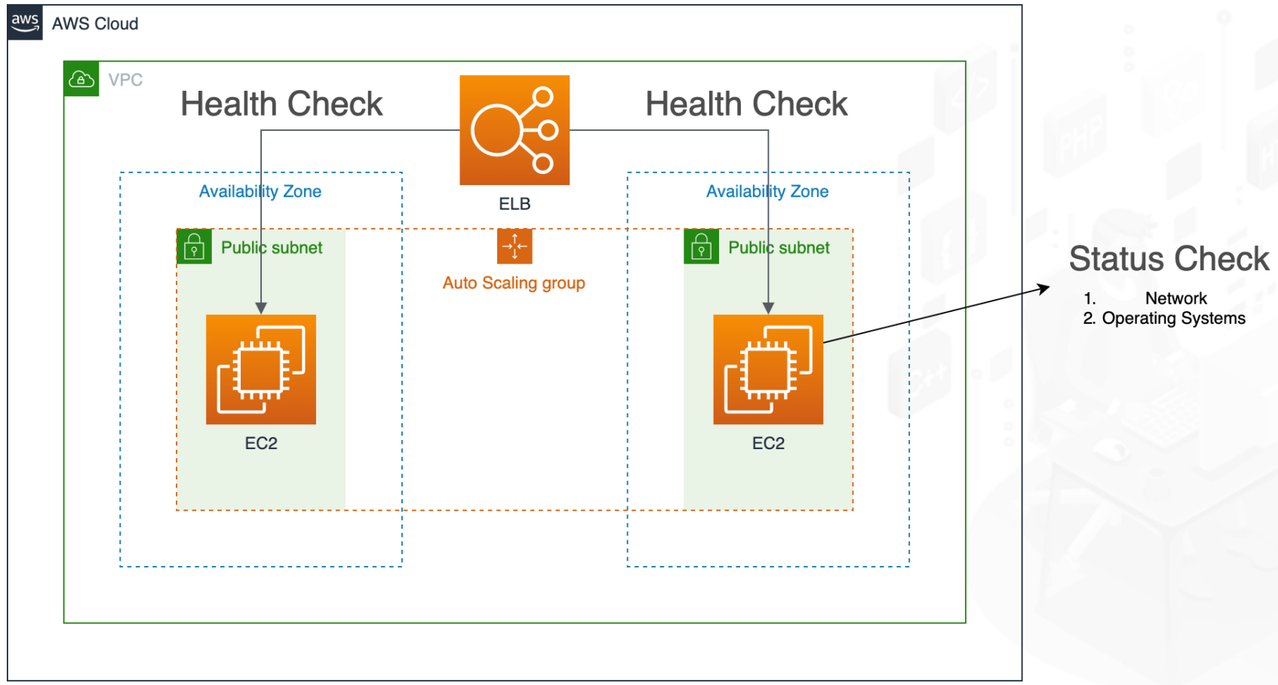
Replacing impaired instances automatically



Balancing capacity across availability zones

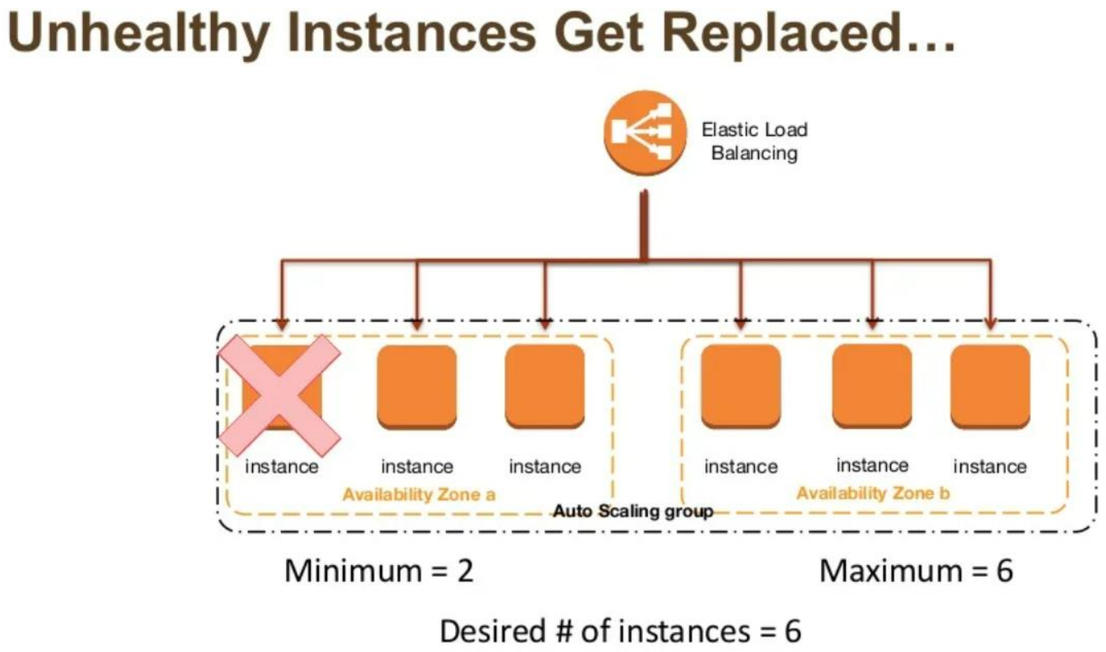
Monitoring the Health of Running Instances

- Auto Scaling monitors the health of all instances that are placed within an **Auto Scaling group** at regular intervals.
- If the instance is connected to an **Elastic Load Balancing** load balancer, it can also perform ELB health checks.



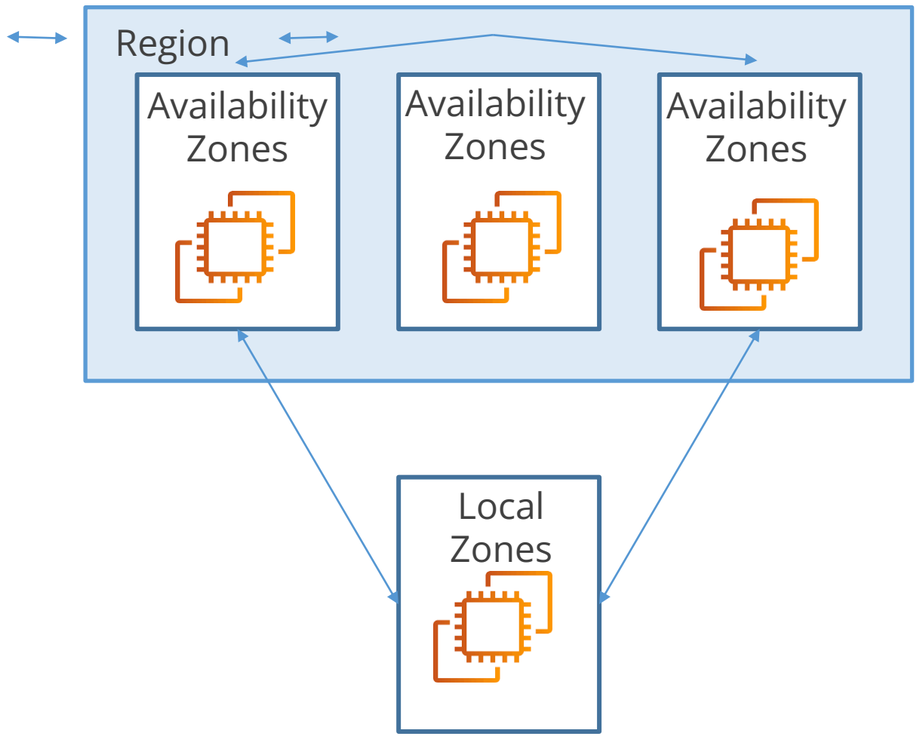
Replacing Impaired Instances Automatically

- When an impaired instance fails a health check, Auto Scaling terminates it and replaces it with a new one automatically.
- If the user is using an Elastic Load Balancing load balancer, Auto Scaling detaches the impaired instance from the load balancer before provisioning a new one and attaches it back to the load balancer.



Balancing Capacity Across Availability Zones

Balancing resources across **availability zones** is the best practice for well-architected applications, as this greatly increases aggregate system availability.



Auto Scaling automatically balances EC2 instances across zones when one configures multiple zones in the Auto Scaling group settings.

Configuring the Manual and Dynamic Scaling

Duration:15 min

Problem Statement:

You have been assigned a task to configure manual and dynamic scaling for an application using Amazon Web Services (AWS) tools and services for optimized resource management and performance.

Outcome:

You will gain a thorough understanding of configuring manual and dynamic scaling in AWS, leveraging Auto Scaling Groups, CloudWatch, and related AWS services to ensure your application can efficiently handle variable loads.

Note: Refer to the demo document for detailed steps:
10_Configuring_the_Manual_and_Dynamic_Scaling

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Set up predefined auto scaling group
2. Set up EC2 Auto Scaling with Load Balancer

Setting Up Auto Scaling Using Launch Templates

Duration:15 min

Problem Statement:

You have been assigned a task to set up an Auto Scaling group using a launch template in AWS for automated instance scaling and optimal resource management.

Outcome:

The Auto Scaling group will dynamically adjust the number of EC2 instances to meet the application's demand, ensuring optimal performance and cost-efficiency.

Note: Refer to the demo document for detailed steps:
11_Setting_Up_Auto_Scaling_Using_Launch_Templates

ASSISTED PRACTICE

Assisted Practice: Guidelines

Steps to be followed are:

1. Create a launch template in EC2
2. Create a launch configuration
3. Create an Auto Scaling group

Quick Check

Your application experiences varying levels of traffic throughout the day, and you want to automatically adjust the number of EC2 instances in response to this changing demand. Which feature should you use to achieve this?

A. Elastic Load Balancer

B. Auto Scaling Groups

C. Spot Instances

D. Reserved Instances

Key Takeaways

- Amazon EC2 provides a highly reliable environment where replacement instances can be quickly and consistently deployed.
- The users can launch multiple instances from a single AMI when they require multiple instances with the same configuration.
- AWS Lambda executes the code only when needed and scales automatically, from a few requests per day to thousands per second.
- Auto Scaling performs EC2 health checks at regular intervals, and if the instance is connected to an Elastic Load Balancer, it can also perform ELB health checks.

Scaling of EBS Volume for a Linux VM

Duration: 30 mins

Project agenda: Perform vertical scaling of EBS volume for a Linux VM

Description: Your company is experiencing business growth where solution deployment is happening with limited resources. In this case, the vertical scalability feature of AWS can be used to create a cost-optimised architecture.

Perform the following:

1. Create an EC2 Instance
2. Identify the EBS volume
3. Create a snapshot
4. Create a new volume
5. Detach the existing volume from the EC2 instance
6. Attach a new volume to the EC2 instance

LESSON-END PROJECT

TECHNOLOGY

Thank You

simplilearn