



अखिल भारतीय तकनीकी शिक्षा परिषद्
All India Council for Technical Education



SRI VASAVI
ENGINEERING COLLEGE
(AUTONOMOUS)

(Approved by AICTE, New Delhi and Permanently Affiliated to JNTUK, Kakinada)

FOR HACKTHON ON DEVOPS WITH AWS

DATE: 29TH JULY, 2023

VENUE: Sri Vasavi Engineering College

Team Members:

1. N. Samyuktha (22A81A05N3)
2. N. Yugendra Babu (22A81A05N6)
3. S. Ravi Kumar (22A81A05O9)
4. D. Ajay Roshan (22A81A05L6)
5. E. Bala Krishna (23A85A0523)

Team number : 36

aws Documentation

Find user guides, developer guides, API references, tutorials, and more.



INTRODUCTION TO AWS:

AWS (Amazon Web Services) is a comprehensive cloud computing platform offered by Amazon.com. It provides a wide range of cloud services that enable businesses and individuals to build and deploy applications and infrastructure in a flexible, scalable, and cost-effective manner. AWS offers a robust set of tools, technologies, and services that cater to various computing needs, including storage, networking, databases, machine learning, analytics, and more.

OVERVIEW OF AWS:

Cloud computing is now an essential part of all businesses in every industry.

Amazon Web Services is a prevalent form that increases efficiency while

assisting several business practices. Dating back to the 2000s, organizations

depended absolutely on servers that are purchased servers. In contrast, such

servers had functions that are limited with steep prices, including a server that is

functioning requiring numerous validations.

The more business keeps experiencing growth, the more optimization practices

and servers are needed. Getting such items showed unproductively and at times

excessively costly.

The benefits of Amazon Web Services have been the answer to many problems.

Organizations that use AWS have instantly available servers; also, AWS offers

various improved storage options, workloads, and enhanced security measures.

HISTORY OF AWS:

Amazon Web Services was launched in 2002. The company intended to sell the

infrastructure that is not in use as a service or offering it to customers, wherein

the purpose was met enthusiastically. Amazon had its first AWS product launched in the year 2006. After four years, in the year 2012, Amazon had a huge occasion to gather customer input concerning AWS. To date, the organization continues to hold similar events, like Reinvent, that lets customers share feedback concerning AWS.

In 2015, Amazon publicized that the revenue of AWS has amounted to \$7.8 billion. From then and 2016, measures had been launched by AWS aiding customers to migrate their services to AWS. Such actions, including the growing and appreciating features of AWS, made further economic growth. In the year 2016, Amazon's revenue augmented to \$12.2 billion in 2016. Presently, AWS provides customers with 160 products and services.

Services that are commonly used and provided by Amazon Web Services

1. Compute Services:

- Amazon Elastic Compute Cloud (EC2): Provides virtual servers in the cloud, allowing users to quickly scale computing resources.
- AWS Lambda: Enables serverless computing, allowing developers to run code without managing servers.

2. Storage Services:

- Amazon Simple Storage Service (S3): Offers scalable object storage for storing and retrieving data.
- Amazon Elastic Block Store (EBS): Provides persistent block-level storage volumes for EC2 instances.

3. Database Services:

- Amazon Relational Database Service (RDS): Managed relational database service supporting multiple database engines.
- Amazon DynamoDB: Fully managed NoSQL database for high-performance applications.

4. Networking Services:

- Amazon Virtual Private Cloud (VPC): Offers isolated virtual networks in the AWS cloud.
- AWS Direct Connect: Provides dedicated network connections between on-premises environments and AWS.

5. Machine Learning and AI Services:

- Amazon SageMaker: Fully managed machine learning service for building, training, and deploying models.
- Amazon Rekognition: Deep learning-based image and video analysis service.
- Amazon Comprehend: Natural language processing service for analyzing text.

6. Analytics Services:

- Amazon Redshift: Data warehousing service for analyzing large datasets.
- Amazon Athena: Serverless interactive query service for analyzing data stored in S3.

7. Security and Identity Services:

- AWS Identity and Access Management (IAM): Manages user access and permissions to AWS resources.
- Amazon Cognito: Provides user authentication and authorization for mobile and web applications.

8. Deployment and Management Services:

- AWS CloudFormation: Infrastructure as code service for provisioning and managing AWS resources.
- AWS Elastic Beanstalk: Platform as a service (PaaS) for deploying and managing applications.

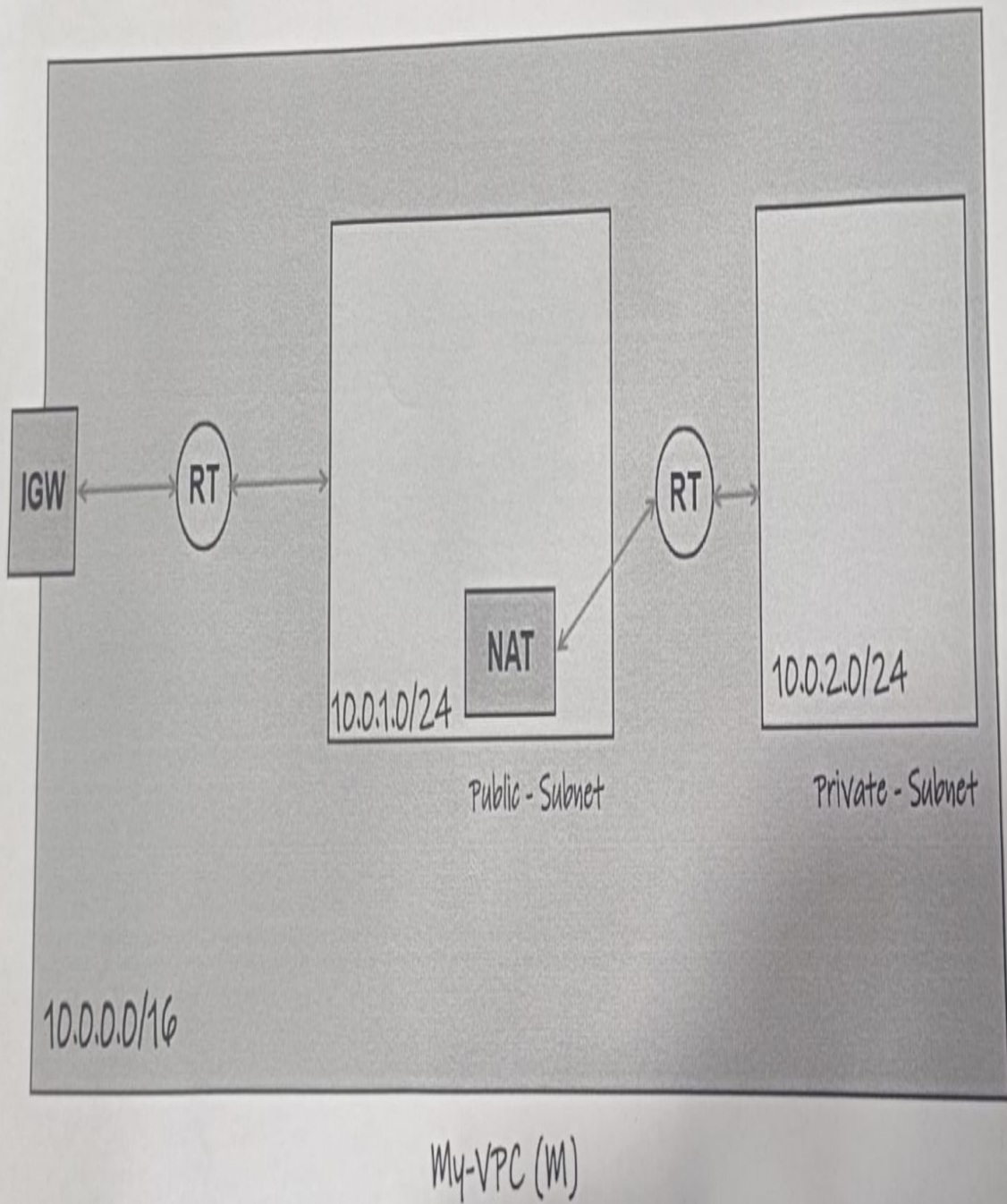
HACKATHON:

The AWS Hackathon, held on 29th September 2023, at Sri Vasavi Engineering college, was a highly successful event that brought together talented developers and innovators to showcase their skills and create innovative solutions using Amazon Web Services (AWS) technologies. The hackathon provided an opportunity for participants to collaborate, learn, and explore the vast capabilities of AWS.

Problem Statement:

We were assigned a task to do within the allotted time. There were four levels, and after completing each one, our performance was certified so that we may move on to the following level. The task of building a Virtual Private Cloud (VPC) was assigned to us. Numerous more ideas are used into the development of VPC. These ideas incorporate several AWS Services. The following problem statements were provided with a graphic representation:

LEVEL-1:



VPC (Virtual Private Cloud) is an AWS service that allows users to create and manage their own isolated virtual network within the AWS cloud. It provides a secure and private environment for running applications and services. The main components of VPC are:

The screenshot displays the AWS Management Console interface for VPC management. The main content area, titled 'Your VPCs (2)', contains a table with the following data:

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCP
-	vpc-08841d58a44c2dd03	Available	172.31.0.0/16	-	dopt-
hp-vpc-01	vpc-0a251b752994a8f61	Available	10.0.0.0/16	-	dopt-

The left sidebar provides navigation links for various AWS services, including 'Virtual private cloud' (with sub-links like 'Your VPCs', 'Subnets', 'Route tables', etc.) and 'Security'. The bottom of the image shows a Windows taskbar with a search bar, several application icons, and a system clock indicating 14:15 on 29-09-2023.

Create a VPC:

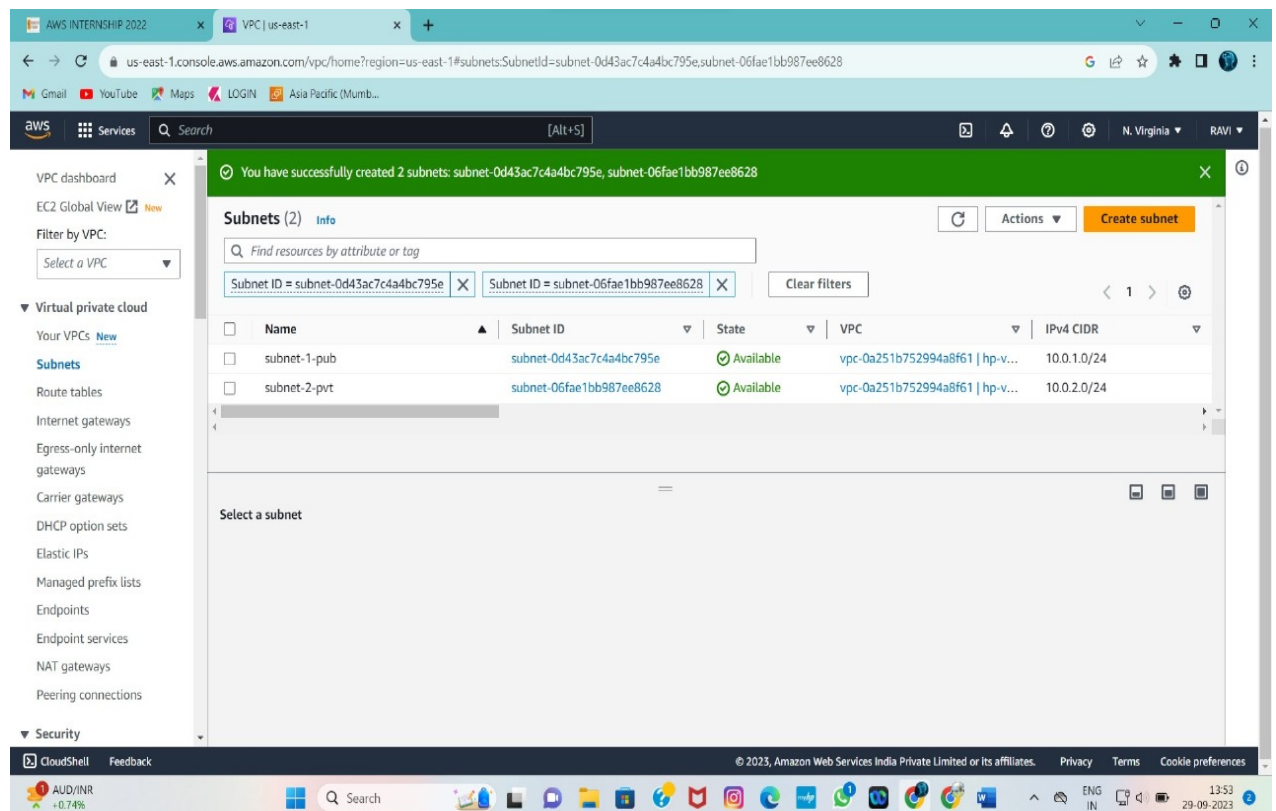
Open the Amazon VPC service in the AWS Management Console.

Select “your VPC’S” from the left side console

Click on "Create VPC" and provide the necessary details, such as the IP address range.

Select "Create VPC" to create the VPC.

Subnets: Subnets divide the IP address range of a VPC into smaller segments and are associated with specific availability zones (AZs) in a region.



Create Subnets:

Click on "Subnets" in the left-side of console.

Click on "Create subnet" and provide the subnet details.

For the public subnet:

Associate it with the VPC created in step 1.

Select an availability zone (AZ) and specify the IP address range.

Choose a route table that allows traffic to an internet gateway.

For the private subnet:

Associate it with the same VPC.

Select a different availability zone (AZ) and specify the IP address range.

Choose a route table that does not have a direct route to an internet gateway.

Click "Create subnet" to create each subnet.

Internet Gateway: An Internet Gateway enables communication between a VPC and the internet.

33°C Partly sunny Search 13:55 29-09-2023

Create an Internet Gateway:

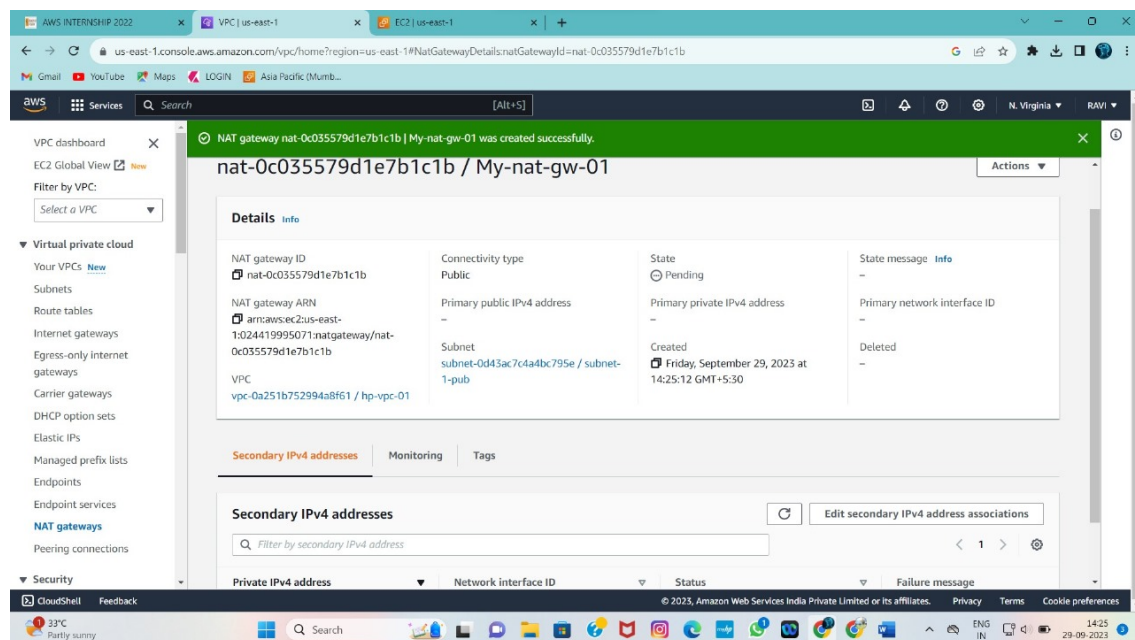
Click on "Internet Gateways" in the left-side console.

Click on "Create internet gateway" and provide a name.

Select the created internet gateway and click on "Actions."

Choose "Attach VPC" and select the VPC created in step 1.

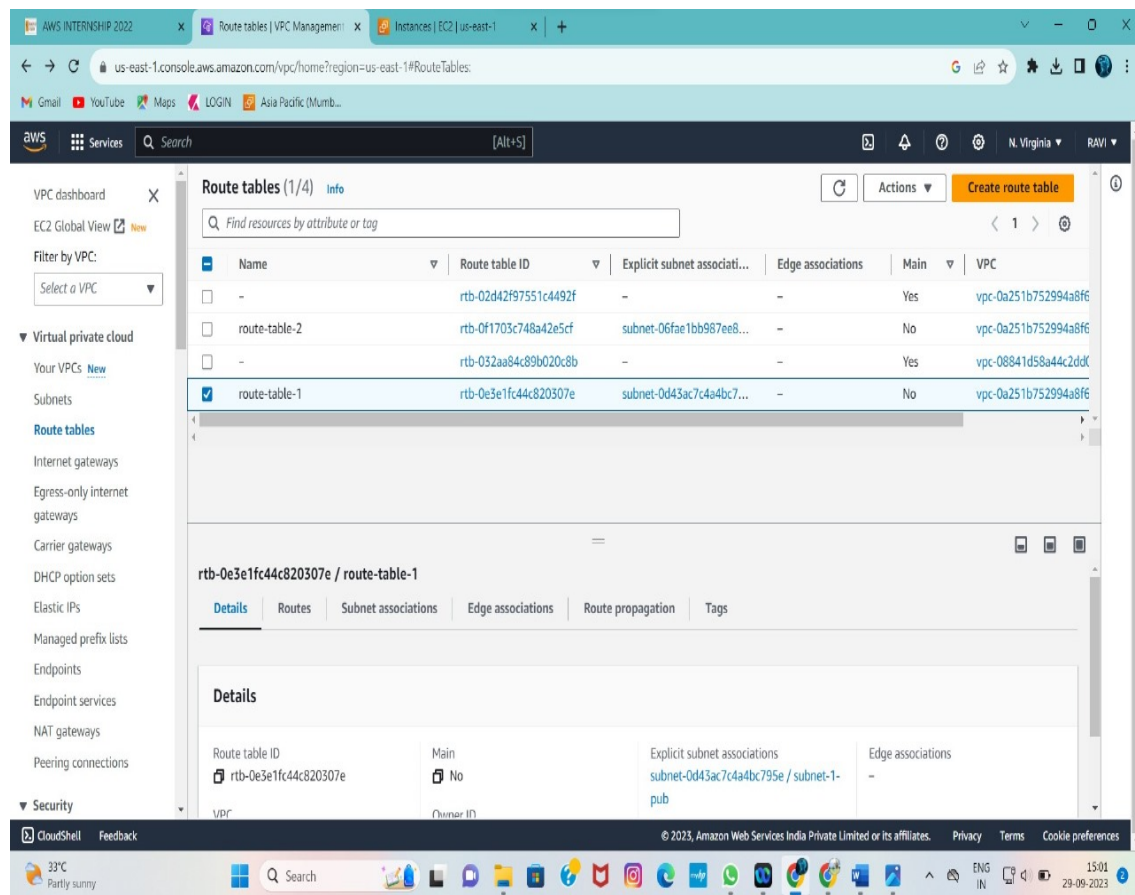
Network Address Translation (NAT) is a technology that allows instances within a private subnet in AWS to access the internet by translating their private IP addresses to public IP addresses.



To configure NAT in AWS:

1. Create a NAT Gateway in a public subnet, associate an Elastic IP address, and route the private subnet's outbound traffic to the NAT Gateway.
2. Modify the private subnet's route table to direct the internet-bound traffic to the NAT Gateway, enabling instances in the private subnet to communicate with the internet while remaining protected.

Route table is a networking component in AWS that controls the traffic flow within a VPC by specifying the destination and target for network packets, determining where they should be sent. It acts as a routing guide for incoming and outgoing network traffic in a virtual network.



Creating Route Tables:

Click on "Route Tables" in the left-hand navigation pane.

Create two route tables, one for the public subnet and one for the private subnet.

For the public route table:

Add a route that allows traffic to an internet gateway (0.0.0.0/0).

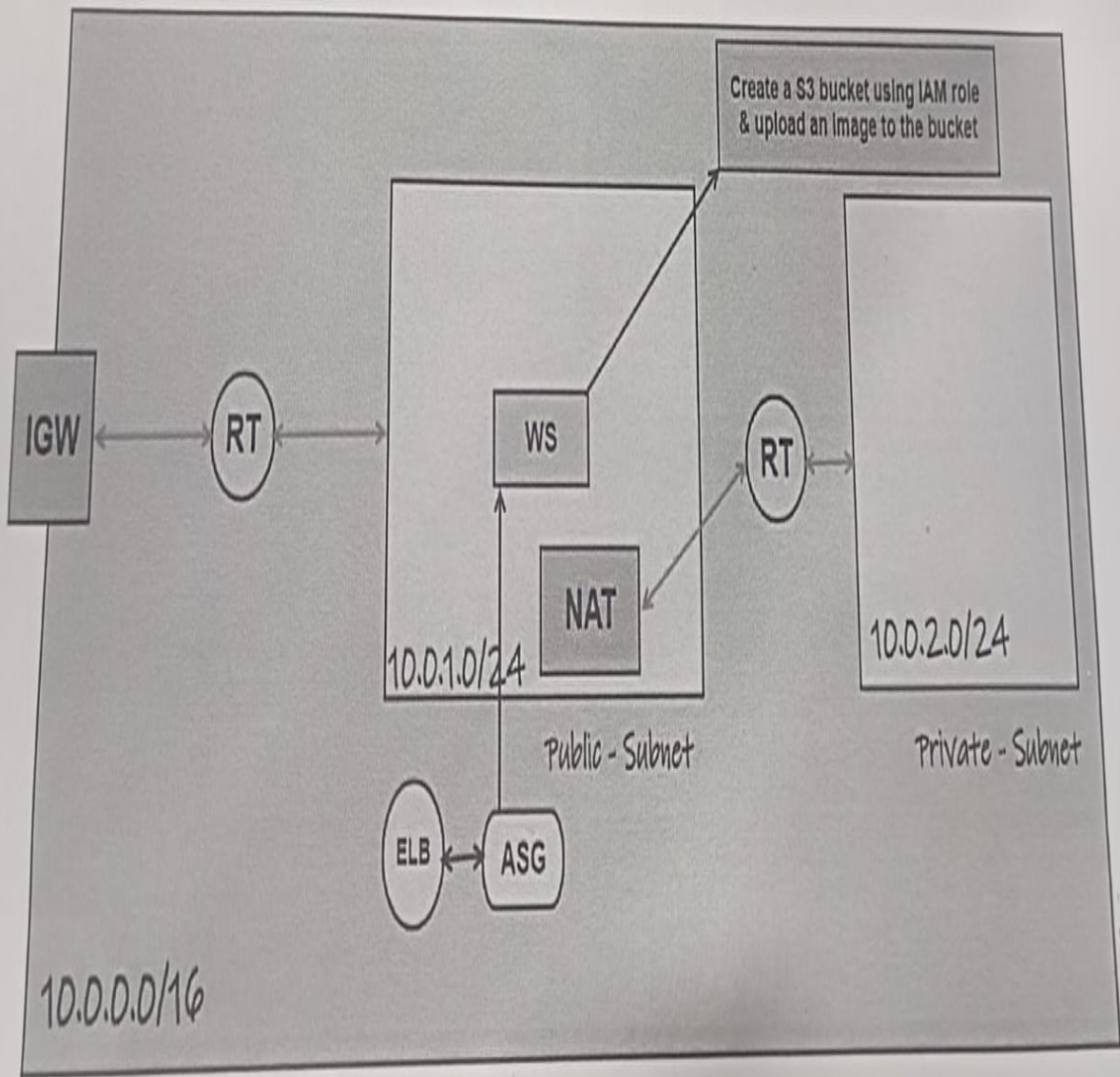
Associate it with the public subnet.

For the private route table:

Associate it with the private subnet.

LEVEL-2:

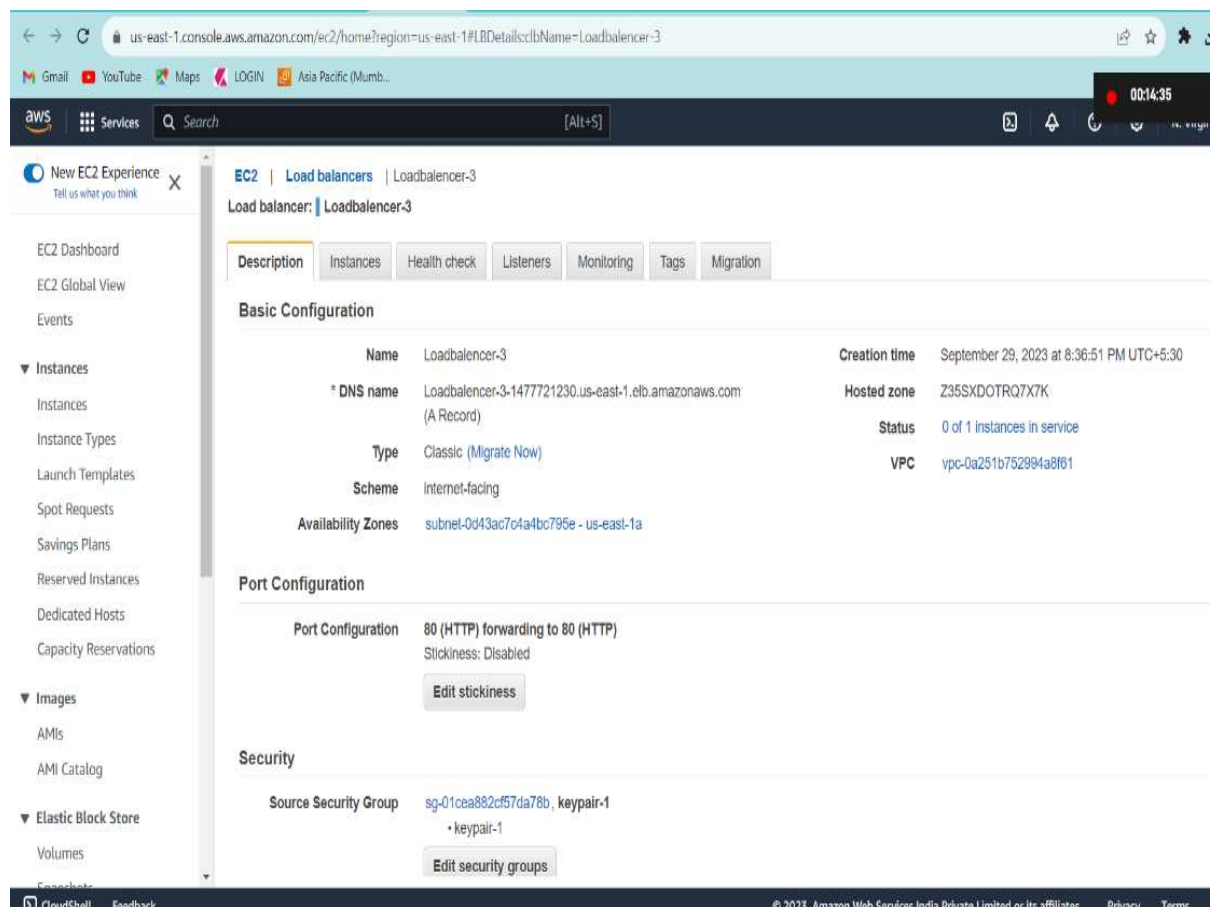
We need to launch a webserver in the public subnet using Auto Scaling group and hostS3bucket and deploy a project with the help of IAM role &upload an image to the bucket.



My-VPC (M)

→ This was launched in the existing VPC only

Classic Load Balancer: A Classic Load Balancer operates at the transport layer (Layer-4) and routes traffic based on network information such as IP addresses and ports. It performs health checks on registered instances and automatically distributes incoming traffic across those instances to ensure efficient utilization and fault tolerance.



Steps to create a Classic Load Balancer:

Open the Amazon EC2 service in the AWS Management Console.

Click on "Load Balancers" in the left-hand navigation pane.

Click on the "Create Load Balancer" button.

Choose "Classic Load Balancer" as the load balancer type.

Configure the basic settings:

Provide a name for your load balancer.

Select the appropriate VPC or the EC2-Classic network.

Define the health check parameters to monitor the health of the instances.

Review the configuration and click on the "Create" button.

Auto Scaling Group (ASG): An Auto Scaling Group manages a group of EC2 instances and automatically adjusts the number of instances based on scaling policies and health checks. ASG maintains a desired capacity of instances, scales up or down based on defined conditions, and replaces unhealthy instances to ensure the desired state of the application.

The screenshot shows the AWS Management Console interface for Launch Templates. The left sidebar contains navigation links for various AWS services. The main content area displays the 'Launch Templates (1/1)' page. A table lists the launch templates, with one entry: 'lt-04c07d17377729521' named 'Tem'. Below the table, the details for the selected template are shown, including its ID, name, default version (1), and owner (arn:aws:iam::024419995071:root). The page also shows the 'Launch template details' section with tabs for Details, Versions, and Template tags.

Launch Template ID	Launch Template Name	Default Version	Latest Version	Create Time	Created By
lt-04c07d17377729521	Tem	1	1	2023-09-29T20:12:09.000Z	arn:aws:iam::024419995071:root

Launch template details

Launch template ID	Launch template name	Default version	Owner
lt-04c07d17377729521	Tem	1	arn:aws:iam::024419995071:root

Launch template version details

Steps to create an Auto Scaling Group:

1. Open the Amazon EC2 service in the AWS Management Console.
2. Click on "Auto Scaling Groups" in the left-hand navigation pane.
3. Click on the "Create Auto Scaling Group" button.
4. Configure the Auto Scaling Group:
 - Provide a name for your Auto Scaling Group.
 - Select the appropriate launch template or launch configuration.
 - Specify the desired capacity, minimum and maximum instances, and the subnet(s) for your instances.
5. Configure scaling policies(optional)
6. Configure notifications (optional)
 - Set up email notifications for scaling events or policy breaches if desired.
8. Configure tags (optional):
 - Add any tags to label and categorize your Auto Scaling Group.
9. Review the configuration and click on the "Create Auto Scaling Group" button.

Webserver: A web server in AWS is a virtual machine or container running web server software to host websites and web applications in the cloud.

Instances | EC2 | us-east-1

deepthi-9 - S3 bucket

deepthi-9.s3.amazonaws.com/iri

mpng (1017x572)

New tab

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#instances:v=3;\$case=tag:true%5C,client:false;\$region=tag:false%5C,client:false

GmailYouTubeMapsLOGINAsia Pacific (Mumb...

awsServicesSearch[Alt+S]

N. VirginiaRAV

Events

Instances

Instances

Instance Types

Launch Templates

Spot Requests

Savings Plans

Reserved Instances

Dedicated Hosts

Capacity Reservations

Images

AMIs

AMI Catalog

Elastic Block Store

Volumes

Snapshots

Lifecycle Manager

Network & Security

Security Groups

Elastic IPs

Instances (1/1) Info

Connect

Instance state

Actions

Launch instances

Find instance by attribute or tag (case-sensitive)

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
webserver	i-0a9bb05bed5f9bcc4	Running	t2.micro	2/2 checks passed	No alarms	us-east-1a	-

Instance: i-0a9bb05bed5f9bcc4

Details

Security

Networking

Storage

Status checks

Monitoring

Tags

Instance summary Info

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0a9bb05bed5f9bcc4	-	10.0.1.157
IPv6 address	Instance state	Public IPv4 DNS
-	Running	-

CloudShellFeedback

© 2023, Amazon Web Services India Private Limited or its affiliates. PrivacyTermsCookie preferences

27°C Mostly cloudy

Search

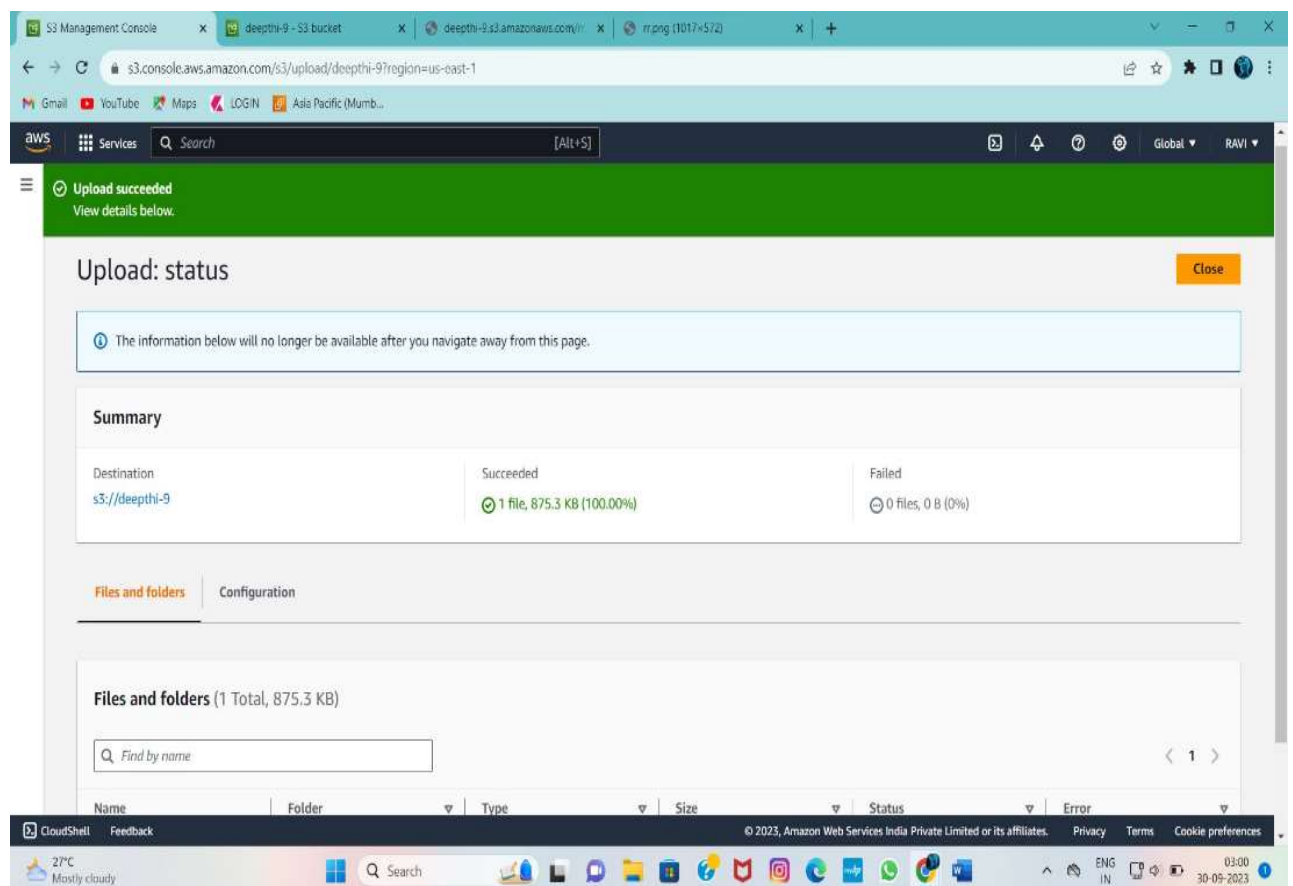
ENG IN

03:02 30-09-2023

To set up a web server in AWS:

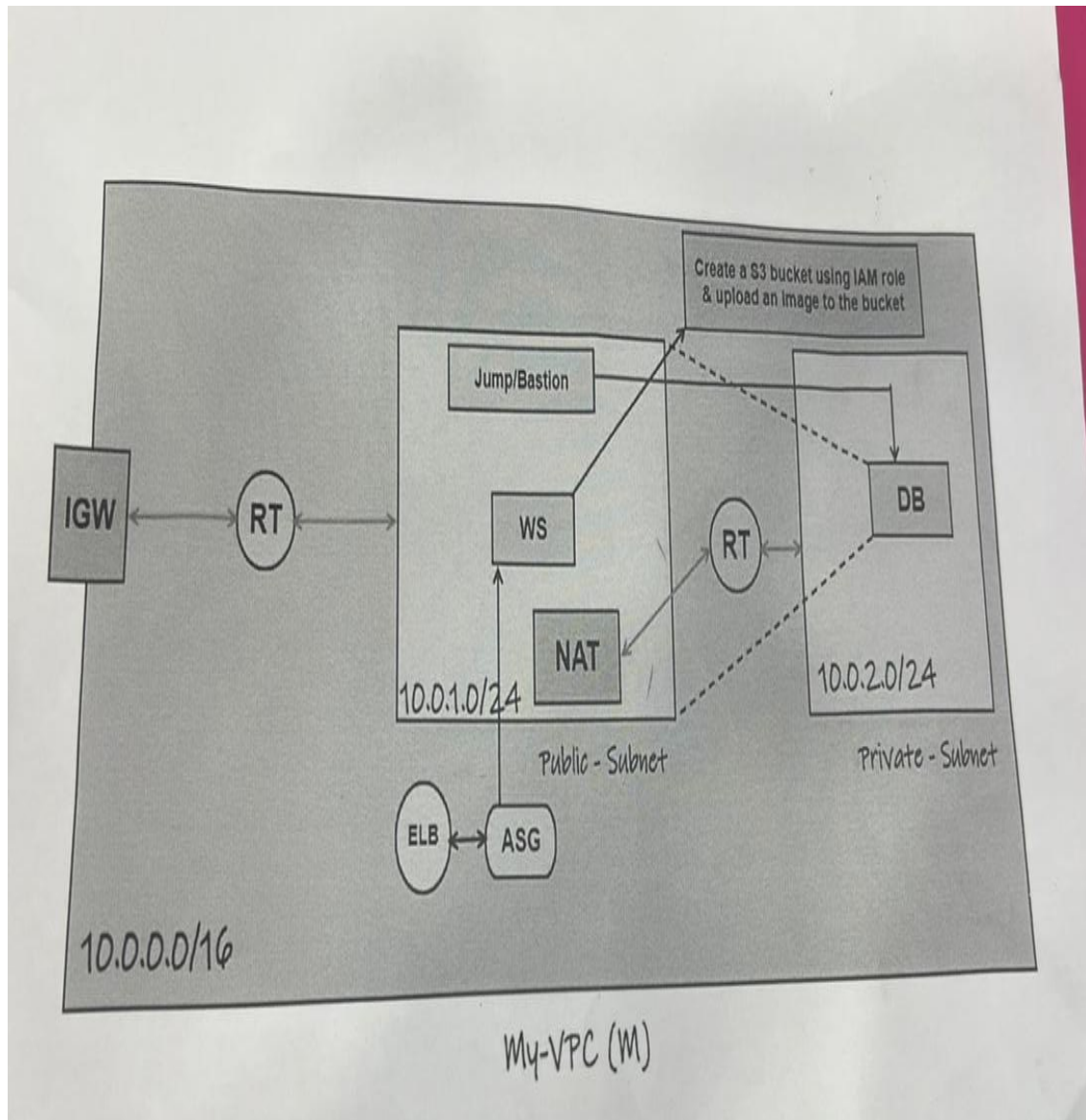
1. Launch an EC2 instance, choose an appropriate Amazon Machine Image (AMI) with pre-installed web server software, configure security groups, and assign an Elastic IP if needed.
2. Connect to the instance, install any additional dependencies or modules, configure the web server software (e.g., Apache, Nginx), upload website files, and configure DNS or load balancer settings to direct traffic to the web server.

Output is as shown in below:



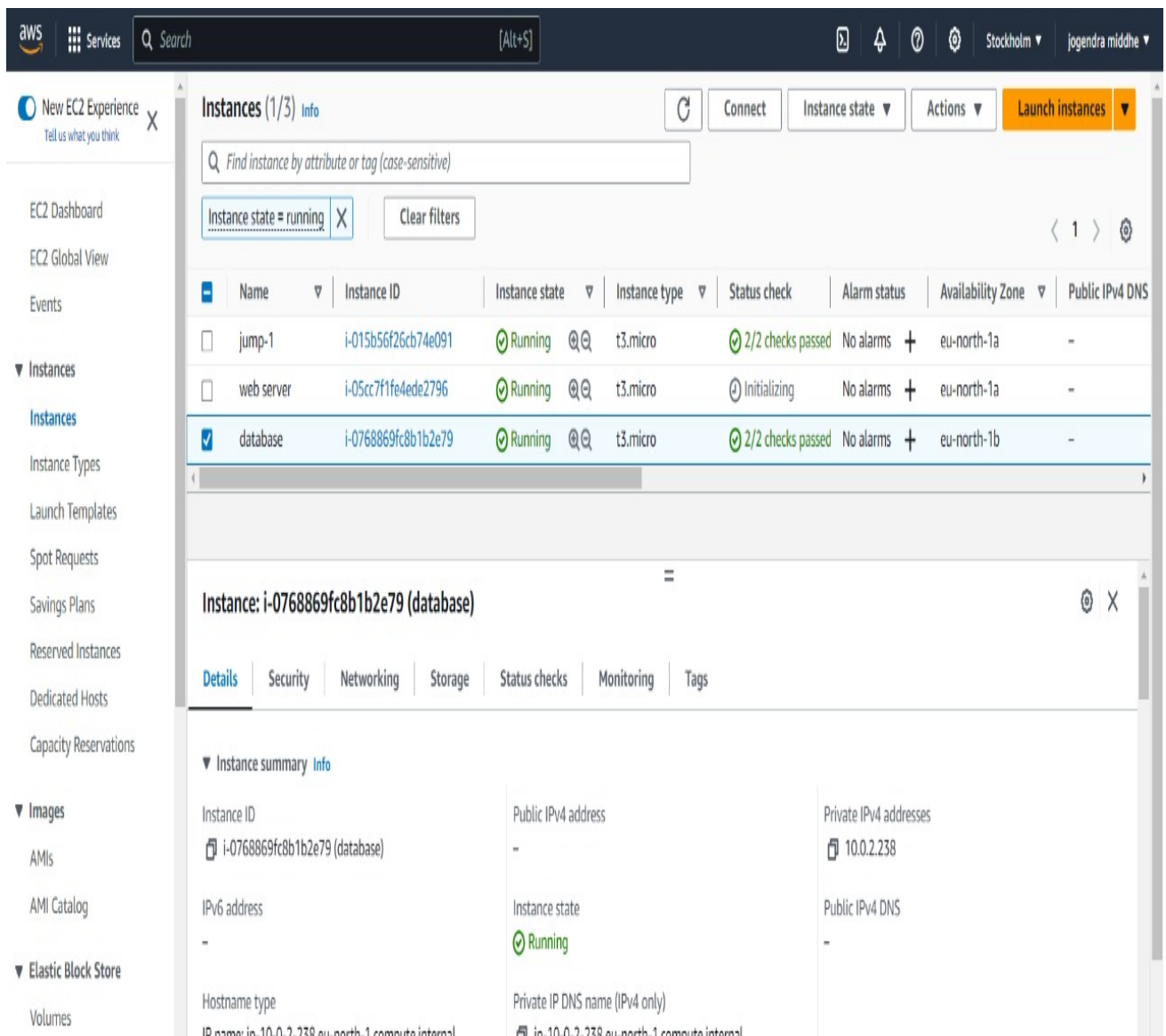
LEVEL-3:

At this level ,we need to launch a database instance in the private subnet, and connect it with the public subnet. We should also connect a Jump Bastion box to the database server from the public subnet.



The VPC which we have created earlier and used in the level 2 too will be used here.

A **database server** is a computer system designed to store, manage, and retrieve data efficiently. It provides a centralized repository for data storage and allows multiple clients or applications to access and manipulate the data securely.



The screenshot displays the AWS Management Console interface for EC2 instances. The left sidebar shows navigation options like EC2 Dashboard, EC2 Global View, Events, and a list of instance types. The main area shows a table of instances with columns for Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. The 'database' instance is selected, and its details are shown in the right pane.

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
jump-1	i-015b56f26cb74e091	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1a	-
web server	i-05cc7f1fe4ede2796	Running	t3.micro	Initializing	No alarms	eu-north-1a	-
database	i-0768869fc8b1b2e79	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1b	-

Instance: i-0768869fc8b1b2e79 (database)

Instance summary

Instance ID	Public IPv4 address	Private IPv4 addresses
i-0768869fc8b1b2e79 (database)	-	10.0.2.238

IPv6 address	Instance state	Public IPv4 DNS
-	Running	-

Hostname type	Private IP DNS name (IPv4 only)
IP name in 10.0.2.238 eu-north-1 compute internal	in-10.0.2.238 eu-north-1 compute internal

To set up a database server in a Virtual Private Cloud (VPC):

1. Launch an EC2 instance to serve as the database server, selecting an appropriate Amazon Machine Image (AMI) that includes the desired database software.
2. Configure security groups to control inbound and outbound traffic for the database server.
3. Install and configure the database software on the EC2 instance, setting up authentication, access control, and database-specific settings.
4. Configure the appropriate network settings within the VPC, such as DNS resolution, routing, and subnet associations.
5. Test the connectivity to the database server from other instances or applications within the VPC to ensure proper functionality.

A **jump bastion box**, also known as a jump host or jump server, is a dedicated server that acts as a secure access point to connect to other servers within a Virtual Private Cloud (VPC). It provides an additional layer of security by controlling and monitoring remote access to the network.

Instances (1/3) Info



Connect

Instance state ▼

Actions ▼

Launch instances ▼

Find instance by attribute or tag (case-sensitive)

Instance state = running X

Clear filters

< 1 > ⓘ

	Name ▼	Instance ID	Instance state ▼	Instance type ▼	Status check	Alarm status	Availability Zone ▼	Public IPv4 DNS
<input checked="" type="checkbox"/>	jump-1	i-015b56f26cb74e091	Running @ @	t3.micro	2/2 checks passed	No alarms +	eu-north-1a	-
<input type="checkbox"/>	web server	i-05cc7f1fe4ede2796	Running @ @	t3.micro	Initializing	No alarms +	eu-north-1a	-
<input type="checkbox"/>	database	i-0768869fc8b1b2e79	Running @ @	t3.micro	2/2 checks passed	No alarms +	eu-north-1b	-

Instance: i-015b56f26cb74e091 (jump-1)



Details

Security

Networking

Storage

Status checks

Monitoring

Tags

▼ Instance summary Info

Instance ID

i-015b56f26cb74e091 (jump-1)

IPv6 address

-

Hostname type

IP name: ip-10-0-1-63.eu-north-1.compute.internal

Public IPv4 address

16.171.47.177 [open address](#)

Instance state

Running

Private IP DNS name (IPv4 only)

ip-10-0-1-63.eu-north-1.compute.internal

Private IPv4 addresses

10.0.1.63

Public IPv4 DNS

-

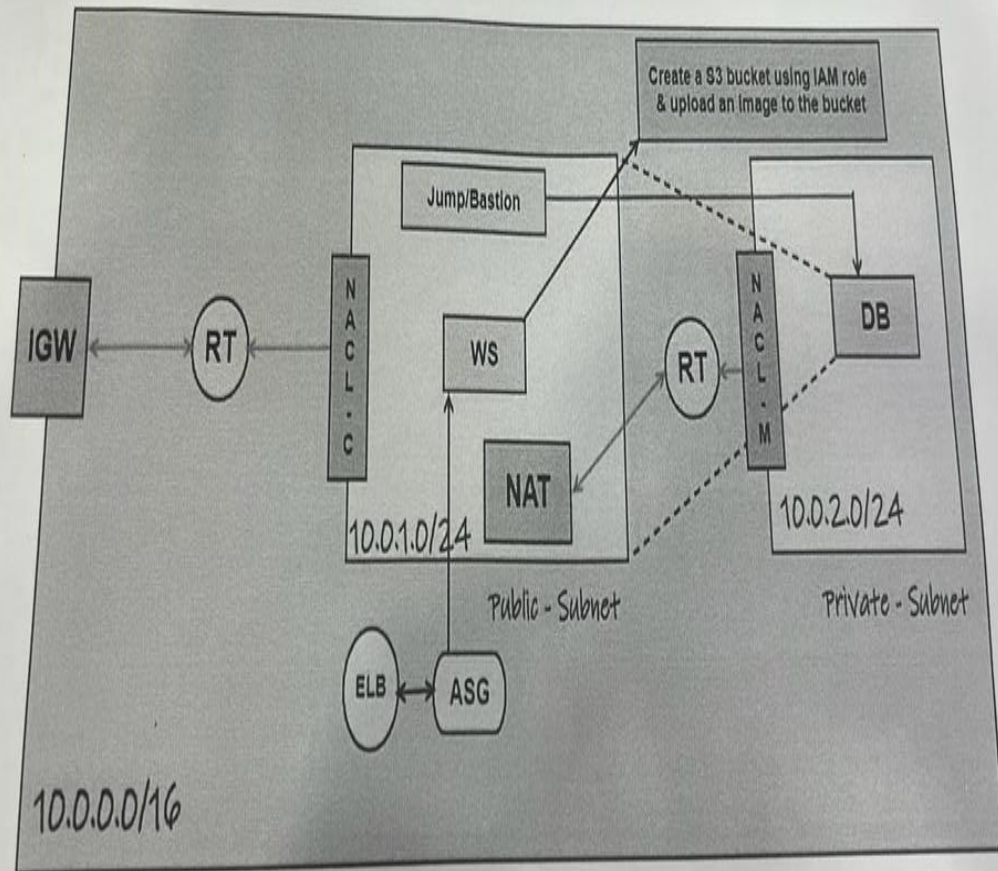
Private IP addresses

Private IP addresses

To set up a jump bastion box in a VPC:

1. Launch an EC2 instance to serve as the jump bastion box, selecting an appropriate AMI and instance type.
2. Configure security groups for the jump bastion box to control inbound and outbound traffic.
3. Set up SSH key-based authentication to securely access the jump bastion box.
4. Configure the network settings, such as routing and subnet associations, to allow the jump bastion box to communicate with other instances in the VPC.
5. Ensure the necessary firewall rules are in place to restrict access to the jump bastion box, allowing only authorized users or IP addresses.
6. Connect to the jump bastion box using SSH and then use it as a gateway to connect securely to the database server of the private subnet within the VPC.

LEVEL-4



My-VPC (M)

At level 4, using the previously created VPC, we need to associate the Network Access Control Lists to the public and private subnets. One of them will be the default and the other will be custom.

Network Access Control Lists (NACLs) are an optional layer of security for controlling inbound and outbound traffic at the subnet level in Amazon Web Services (AWS). NACLs act as stateless firewalls that allow or deny traffic based on rules.

The screenshot displays the AWS Management Console interface for Network ACLs. The left sidebar shows the navigation menu with categories like Virtual private cloud, Security, DNS firewall, and Network Firewall. The main content area is titled 'Network ACLs (1/3)' and includes a search bar and a table of existing ACLs. The table has columns for Name, Network ACL ID, Associated with, Default, and VPC ID. Three ACLs are listed: NACL-C, -, and NACL-M. NACL-M is selected, and its details are shown below the table. The details section includes tabs for Details, Inbound rules, Outbound rules, Subnet associations, and Tags. The 'Details' tab is active, showing the Network ACL ID, Associated with, Default, VPC ID, and Owner.

Name	Network ACL ID	Associated with	Default	VPC ID
NACL-C	acl-03284d40f4b2ce754	subnet-0827e7f9b73be1abe / sub-pub-1a	No	vpc-00be1f7436279d4c8 / My-VPC
-	acl-0e074cd387eae3b04	3 Subnets	Yes	vpc-06883c7676f6c7273
NACL-M	acl-0723721a7bc273c3b	subnet-014a48ac5683afc29 / sub-pvt-1b	Yes	vpc-00be1f7436279d4c8 / My-VPC

acl-0723721a7bc273c3b / NACL-M

Details

Network ACL ID	Associated with	Default	VPC ID
acl-0723721a7bc273c3b	subnet-014a48ac5683afc29 / sub-pvt-1b	Yes	vpc-00be1f7436279d4c8 / My-VPC

Owner

Owner
614922307852

To create a NACL in AWS:

1. Open the Amazon VPC service in the AWS Management Console.
2. Click on "Network ACLs" in the left-hand navigation pane.
3. Click on "Create network ACL" and specify a name and VPC association.
4. Define inbound and outbound rules for the NACL, specifying the allowed protocols, ports, and IP ranges.
5. Name the main NACL as the default one.
6. Associate the NACL with the public subnet created previously.

CONCLUSION:

Participating in the hackathon was a valuable learning experience. We faced several challenges, such as time constraints, technical hurdles, and coordination among team members. However, these challenges helped us develop problem-solving skills, collaboration, and adaptability.

Through the hackathon, we learned the importance of effective planning. Clear delegation of tasks, regular updates, and utilizing collaborative tools improved our productivity and efficiency.

We also recognized the significance of leveraging cloud services, specifically AWS, for rapid prototyping and deployment. Using AWS services like VPC, EC2, ELB, ASG, streamlined our development process and enabled us to create scalable and resilient solutions.

In future hackathons, we would aim to allocate more time for planning and ideation, allowing for better execution.

Additionally, conducting thorough research on AWS services and best practices beforehand would enhance our ability to leverage the platform effectively.

Overall, the hackathon experience provided us with practical insights into solving real-world challenges, sharpened our technical skills, and emphasized the importance of teamwork and innovation.



