SRI VASAVI ENGINEERING COLLEGE (AUTONOMOUS)
(Approved by AICTE,New Delhi and Permanently Affliated to JNTUK,Kakinada)

REPORT FOR HACKATHON WITH AWS

DATE: 29thseptember,2023

VENUE: Sri Vasavi Engineering College(Autonomous)

BATCH NO : 41 ,CSE-D

TEAMMEMBERS:

D.JYOTHSNA(22A81A05L3)

SK.SAHEDHA(22A81A05P1)

G.PRAVEENA(22A81A05M0)

G.SATISH(22A81A05M2)

P.CHAITANYA(22A81A05O4)

P.HEMANTH SAI(22A81A05O6)

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.

BENEFITS OF AWS:

Easy to use

AWS is designed to allow application providers, ISVs, and vendors to quickly and securely host your applications – whether an existing application or a new SaaS-based application. You can use the AWS Management Console or well-documented web services APIs to access AWS’s application hosting platform.

Flexible

AWS enables you to select the operating system, programming language, web application platform, database, and other services you need. With AWS, you receive a virtual environment that lets you load the software and services your application requires. This eases the migration process for existing applications while preserving options for building new solutions.

Cost-Effective

You pay only for the compute power, storage, and other resources you use, with no long-term contracts or up-front commitments. For more information on comparing the costs of other hosting alternatives with AWS, see the AWS Economics Center.

Reliable

With AWS, you take advantage of a scalable, reliable, and secure global computing infrastructure, the virtual backbone of Amazon.com's multi-billion dollar online business that has been honed for over a decade.

Scalable and high-performance

Using AWS tools, Auto Scaling, and Elastic Load Balancing, your application can scale up or down based on demand. Backed by Amazon's massive infrastructure, you have access to compute and storage resources when you need them.

Secure

AWS utilizes an end-to-end approach to secure and harden our infrastructure, including physical, operational, and software measures. For more information, see the AWS Security Center.

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.

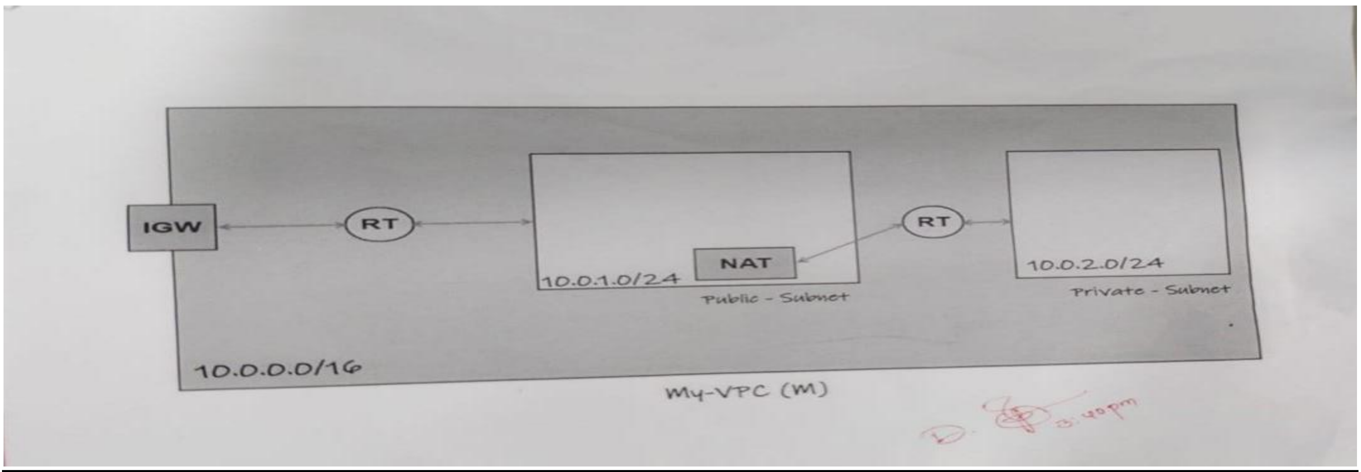
HACKATHON:

The AWS Hackathon, held on 30 sept 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.

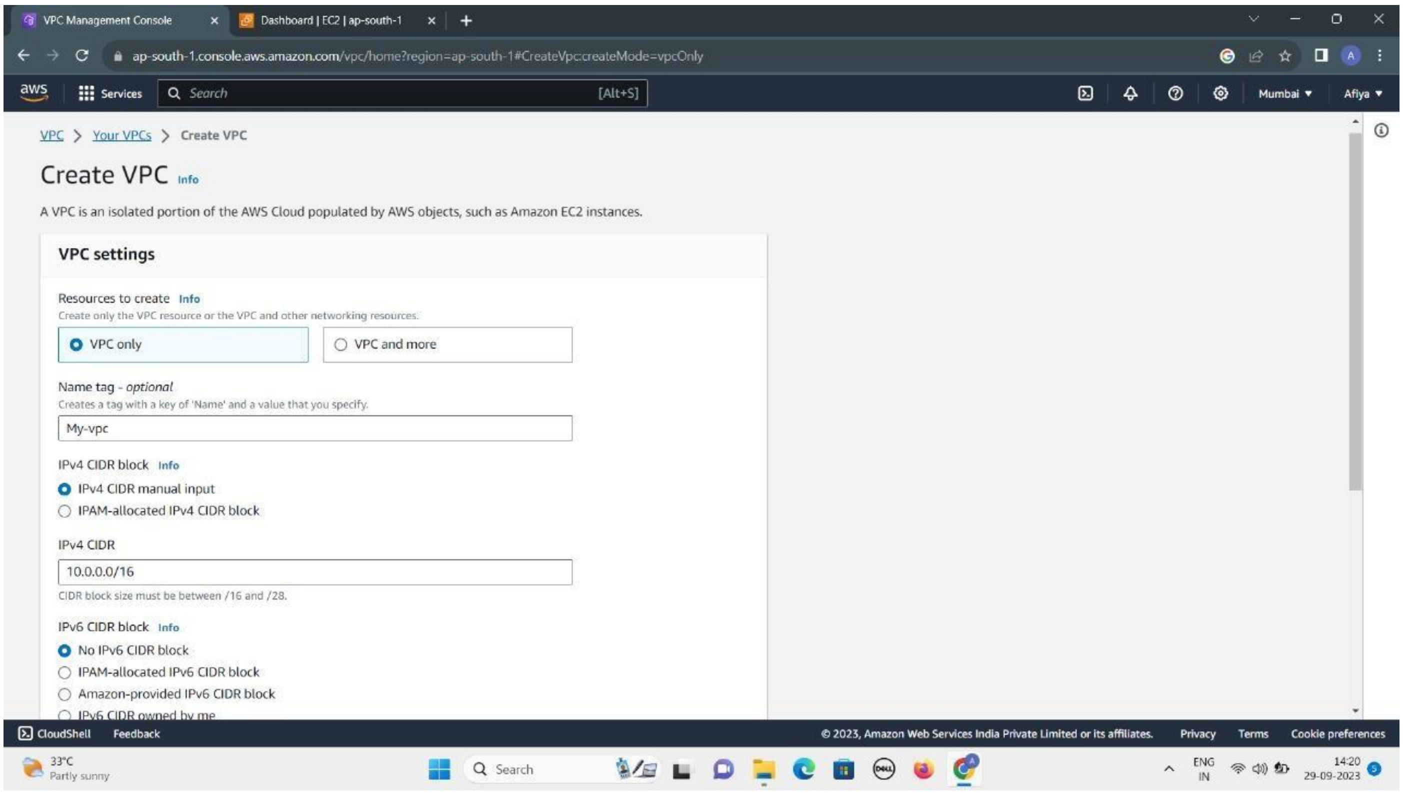
Problem Statement:

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 fol lowing 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:


VPC Management Console
Dashboard | EC2 | ap-south-1
ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#CreateVpc:createMode=vpcOnly
Services
Search
[Alt+S]
Mumbai
Afiya
VPC > Your VPCs > Create VPC

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
[✓ VPC only] VPC and more

Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
My-vpc

IPv4 CIDR block Info
[✓ IPv4 CIDR manual input]
IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
[✓ No IPv6 CIDR block]
IPAM-allocated IPv6 CIDR block
Amazon-provided IPv6 CIDR block
IPv6 CIDR owned by me

Create a VPC:

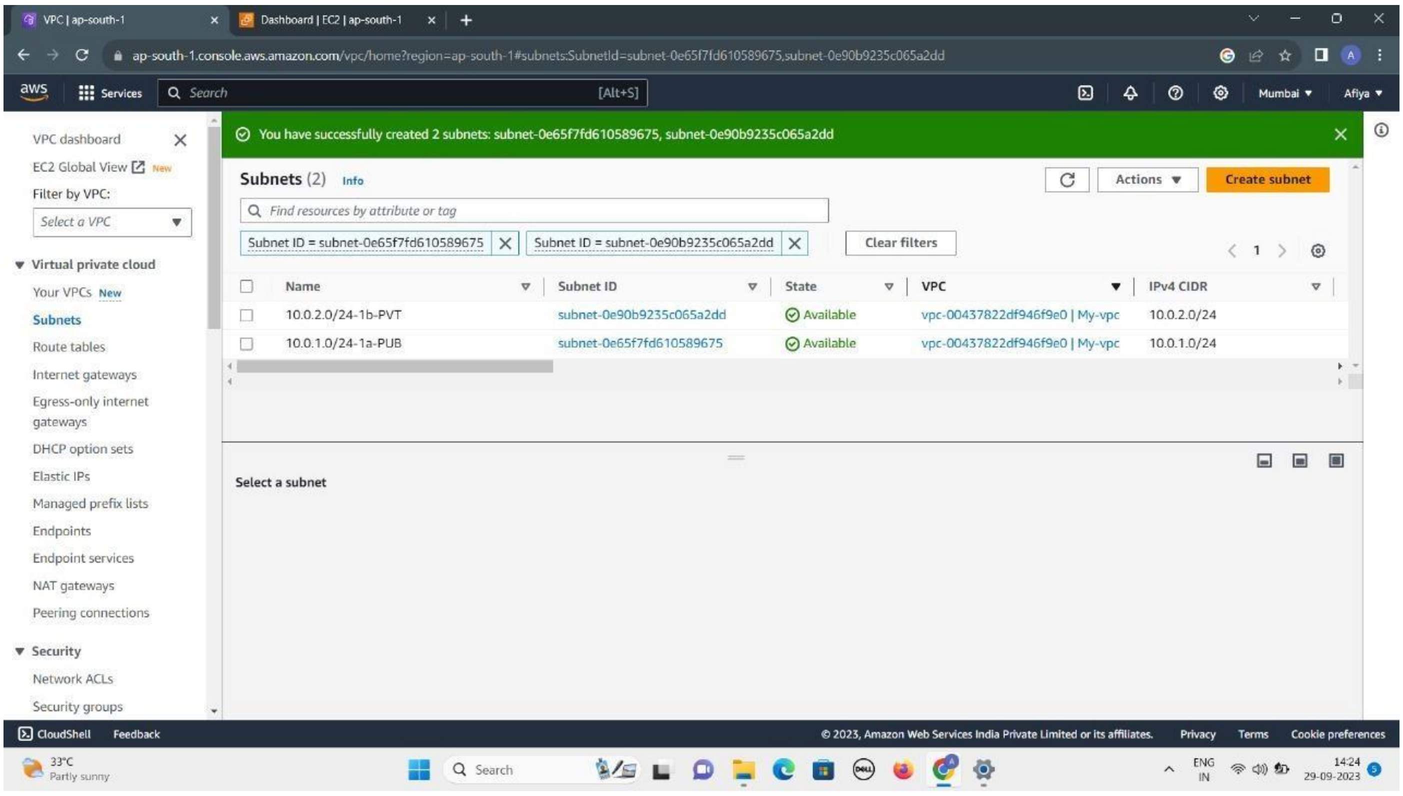
Open the Amazon VPC service in the AWS Management Console.

Click on "Your VPCs" in the left-hand navigation pane.

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

Click "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.



You have successfully created 2 subnets: subnet-0e65f7fd610589675, subnet-0e90b9235c065a2dd

Subnets (2) Info

Subnet ID = subnet-0e65f7fd610589675   Subnet ID = subnet-0e90b9235c065a2dd

Name	Subnet ID	State	VPC	IPv4 CIDR
10.0.2.0/24-1b-PVT	subnet-0e90b9235c065a2dd	Available	vpc-00437822df946f9e0 | My-vpc	10.0.2.0/24
10.0.1.0/24-1a-PUB	subnet-0e65f7fd610589675	Available	vpc-00437822df946f9e0 | My-vpc	10.0.1.0/24

Select a subnet

Create Subnets:

Click on "Subnets" in the left-hand navigation pane.

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:

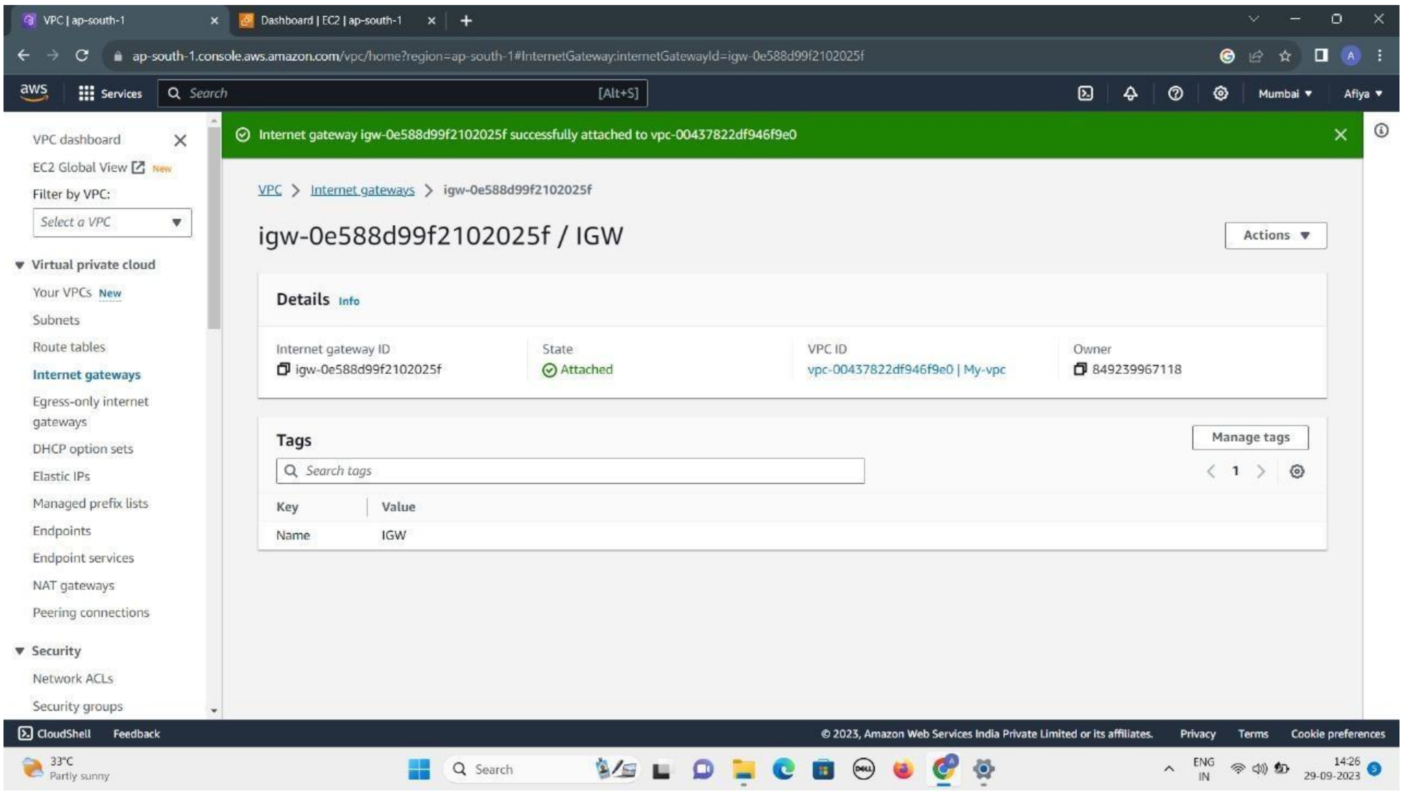
Associate it with the same VPC.

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.



Internet gateway igw-0e588d99f2102025f successfully attached to vpc-00437822df946f9e0

VPC > Internet gateways > igw-0e588d99f2102025f

igw-0e588d99f2102025f / IGW

Actions

Details Info

Internet gateway ID	State	VPC ID	Owner
igw-0e588d99f2102025f	Attached	vpc-00437822df946f9e0 | My-vpc	849239967118

Tags

Manage tags

Search tags

Key	Value
Name	IGW

Create an Internet Gateway:

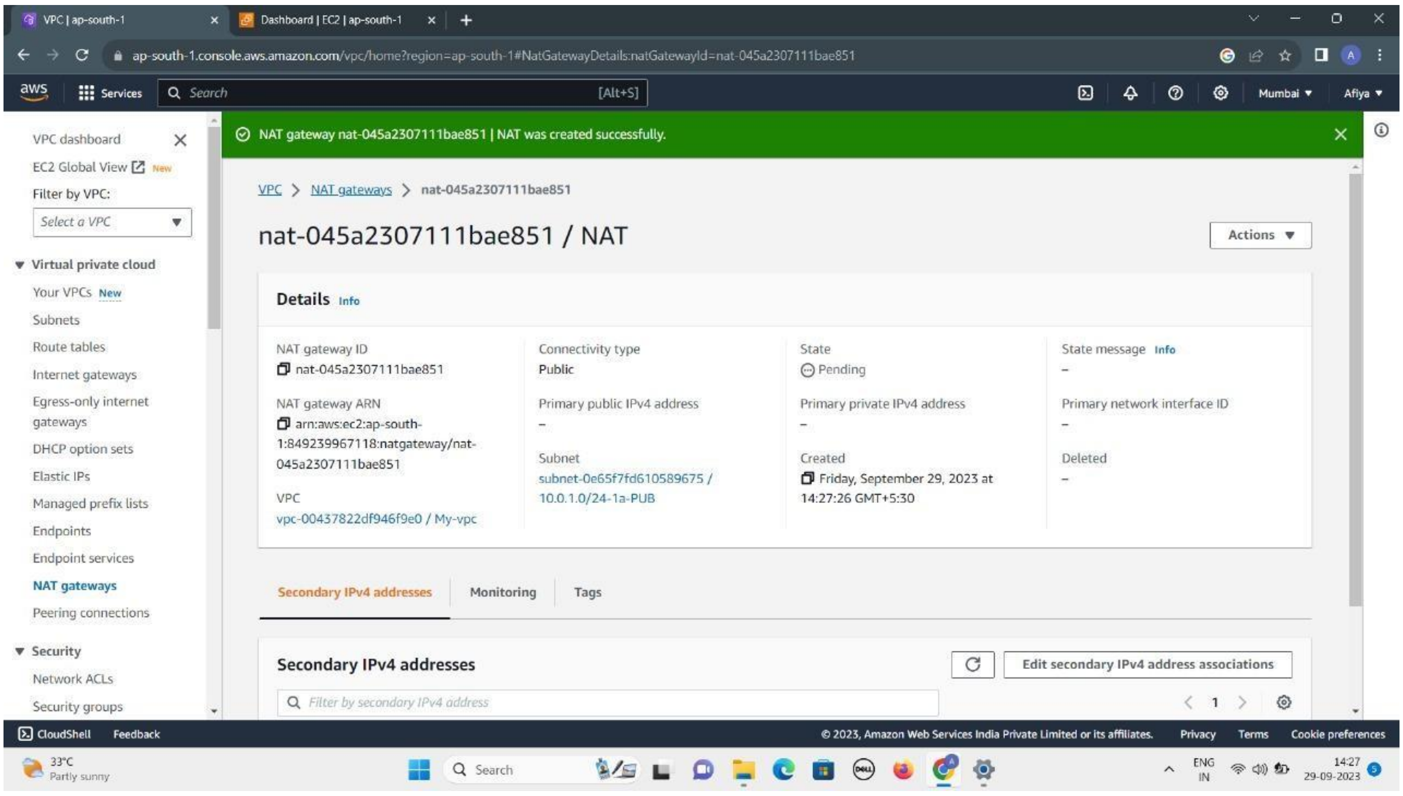
Click on "Internet Gateways" in the left-hand navigation pane.

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.


NAT gateway nat-045a2307111bae851 | NAT was created successfully.

VPC > NAT gateways > nat-045a2307111bae851

nat-045a2307111bae851 / NAT

Details

NAT gateway ID: nat-045a2307111bae851	Connectivity type: Public	State: Pending	State message: –
NAT gateway ARN: arn:aws:ec2:ap-south-1:849239967118:natgateway/nat-045a2307111bae851	Primary public IPv4 address: –	Primary private IPv4 address: –	Primary network interface ID: –
VPC: vpc-00437822df946f9e0 / My-vpc	Subnet: subnet-0e65f7fd610589675 / 10.0.1.0/24-1a-PUB	Created: Friday, September 29, 2023 at 14:27:26 GMT+5:30	Deleted: –

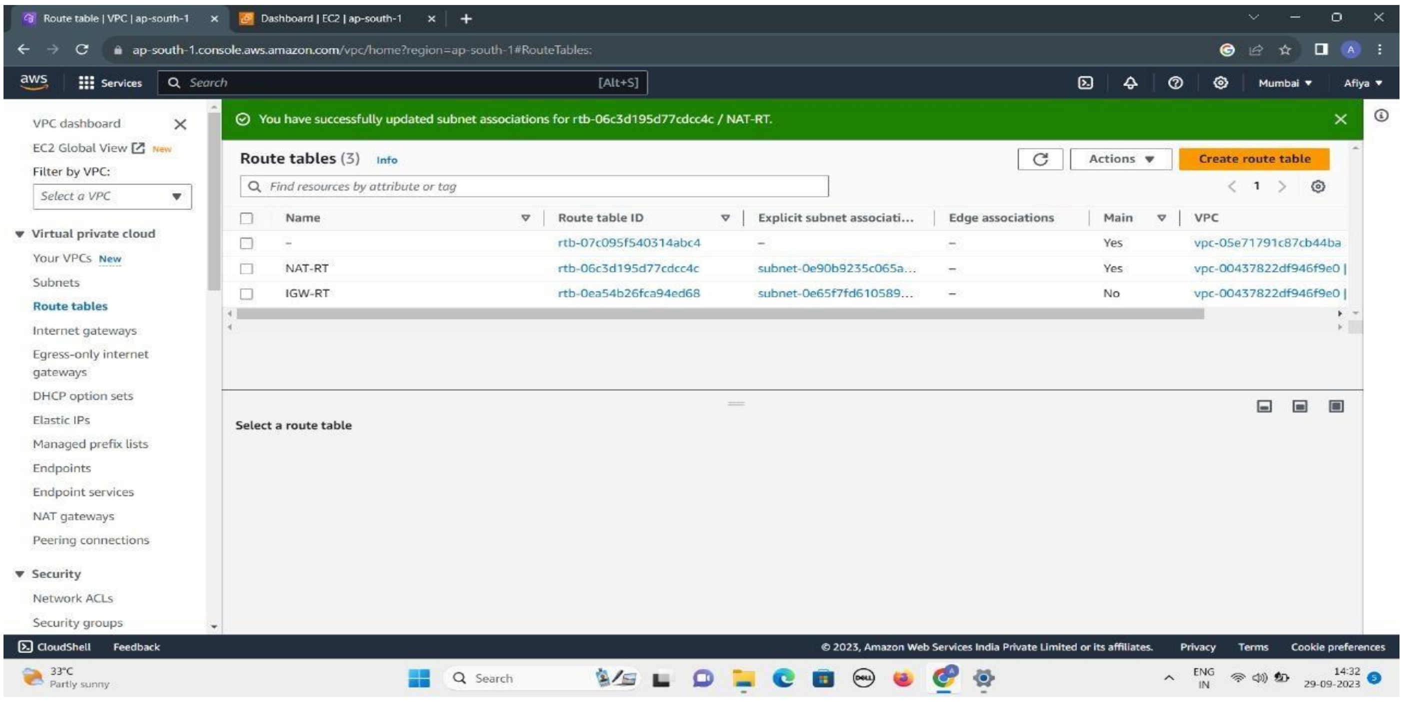
Secondary IPv4 addresses

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.



You have successfully updated subnet associations for rtb-06c3d195d77cdcc4c / NAT-RT.

Route tables (3)

Name	Route table ID	Explicit subnet associati...	Edge associations	Main	VPC
-	rtb-07c095f540314abc4	–	–	Yes	vpc-05e71791c87cb44ba
NAT-RT	rtb-06c3d195d77cdcc4c	subnet-0e90b9235c065a...	–	Yes	vpc-00437822df946f9e0
IGW-RT	rtb-0ea54b26fca94ed68	subnet-0e65f7fd610589...	–	No	vpc-00437822df946f9e0

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.

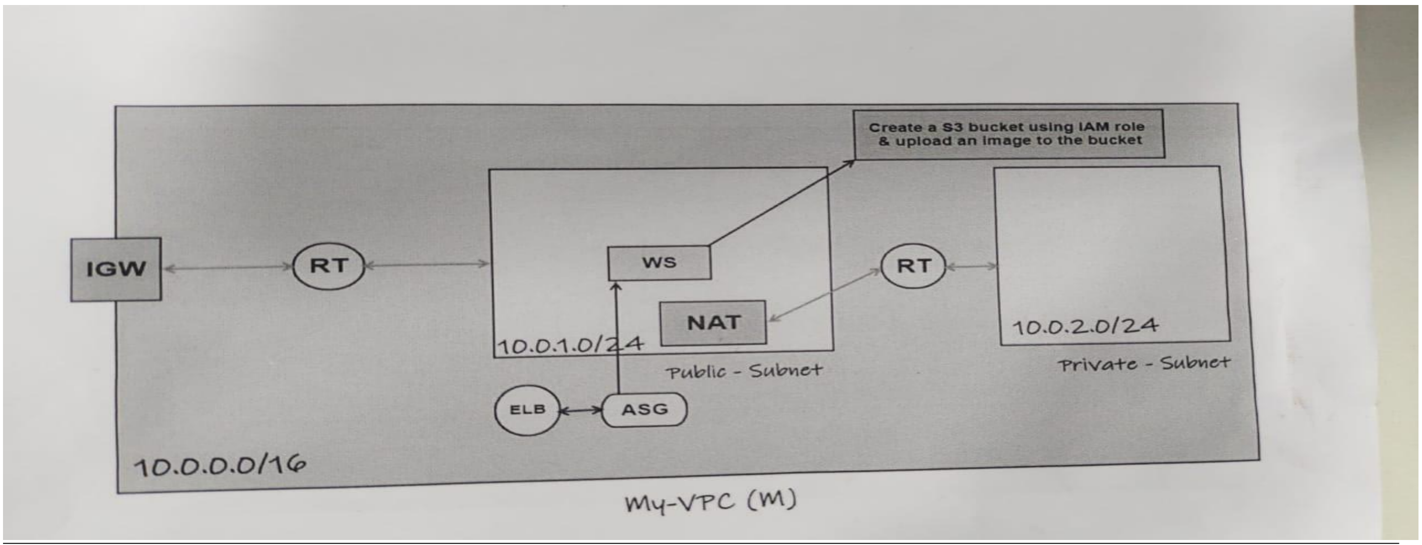
Configure Security Groups:

Click on "Security Groups" in the left-hand navigation pane.

Create security groups for the public and private subnets, specifying inbound and outbound rules as needed.

LEVEL-2

We need to launch a webserver in the public subnet using Auto Scaling Group and should be connected with s3 bucket using IAM role&upload a image to the bucket.



- 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.

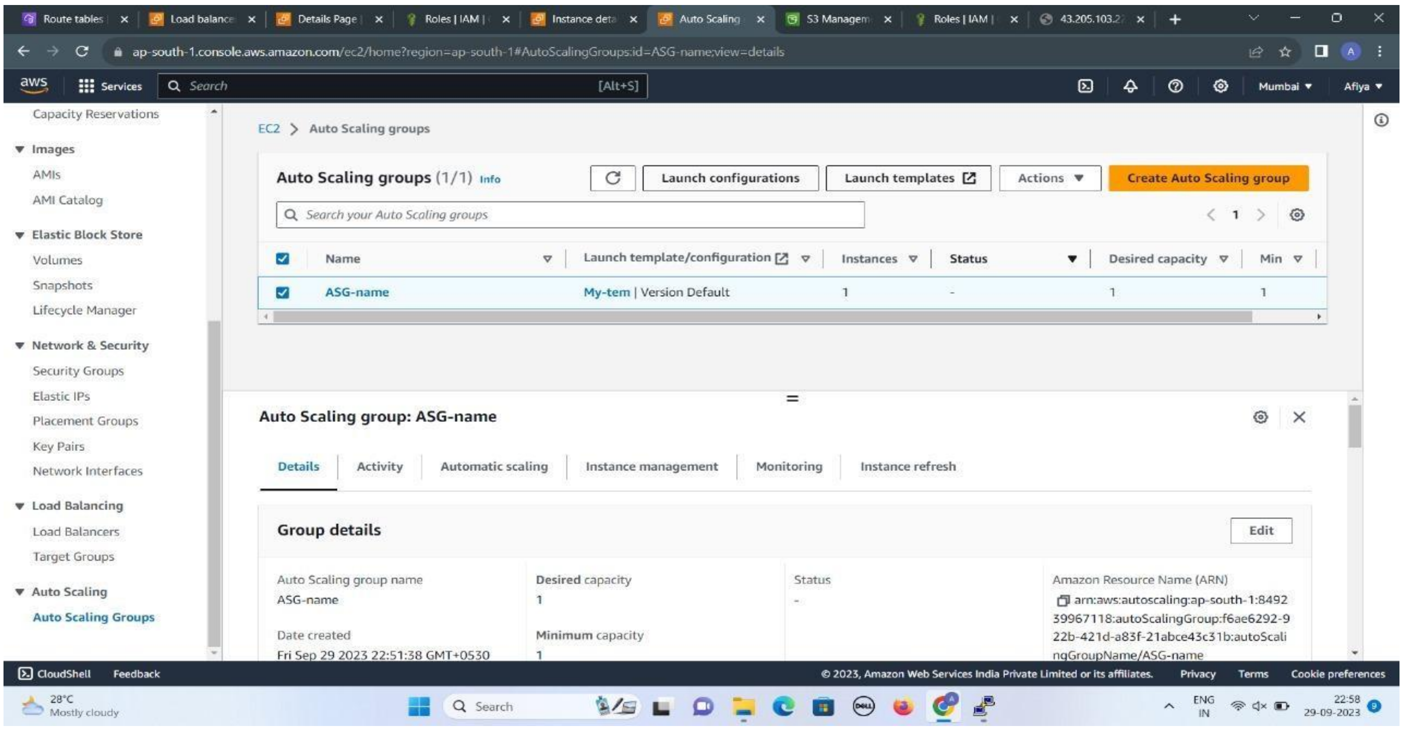
Choose "Classic Load Balancer" as the load balancer type.

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.

<u>Auto Scaling Group (ASG):</u>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.



Auto Scaling groups (1/1) Info

	Name	Launch template/configuration	Instances	Status	Desired capacity	Min
✓	ASG-name	My-tem | Version Default	1	-	1	1

Auto Scaling group: ASG-name

Group details

Auto Scaling group name
ASG-name

Date created
Fri Sep 29 2023 22:51:38 GMT+0530

Desired capacity
1

Minimum capacity
1

Status
-

Amazon Resource Name (ARN)
arn:aws:autoscaling:ap-south-1:849239967118:autoScalingGroup:f6ae6292-922b-421d-a83f-21abce43c31b:autoScalingGroupName/ASG-name

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)

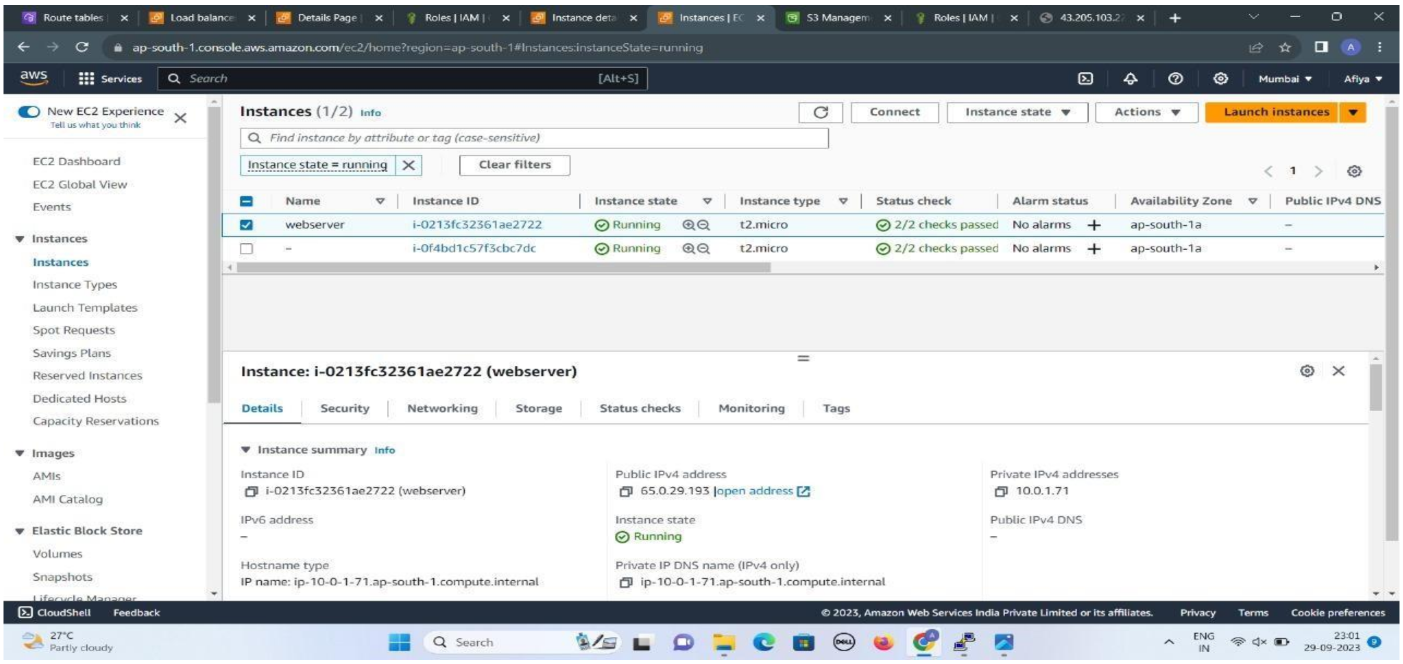
   - Set up email notifications for scaling events or policy breaches if desired.

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.

<u>Webserver:</u>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 (1/2) Info

Instance state = running

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
✓	webserver	i-0213fc32361ae2722	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
	–	i-0f4bd1c57f3cbc7dc	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–

Instance: i-0213fc32361ae2722 (webserver)

Instance summary Info

Instance ID: i-0213fc32361ae2722 (webserver)
Public IPv4 address: 65.0.29.193 | open address
Private IPv4 addresses: 10.0.1.71
IPv6 address: –
Instance state: Running
Public IPv4 DNS: –
Hostname type: IP name: ip-10-0-1-71.ap-south-1.compute.internal
Private IP DNS name (IPv4 only): ip-10-0-1-71.ap-south-1.compute.internal

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

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.

<u>S3 Bucket:</u>

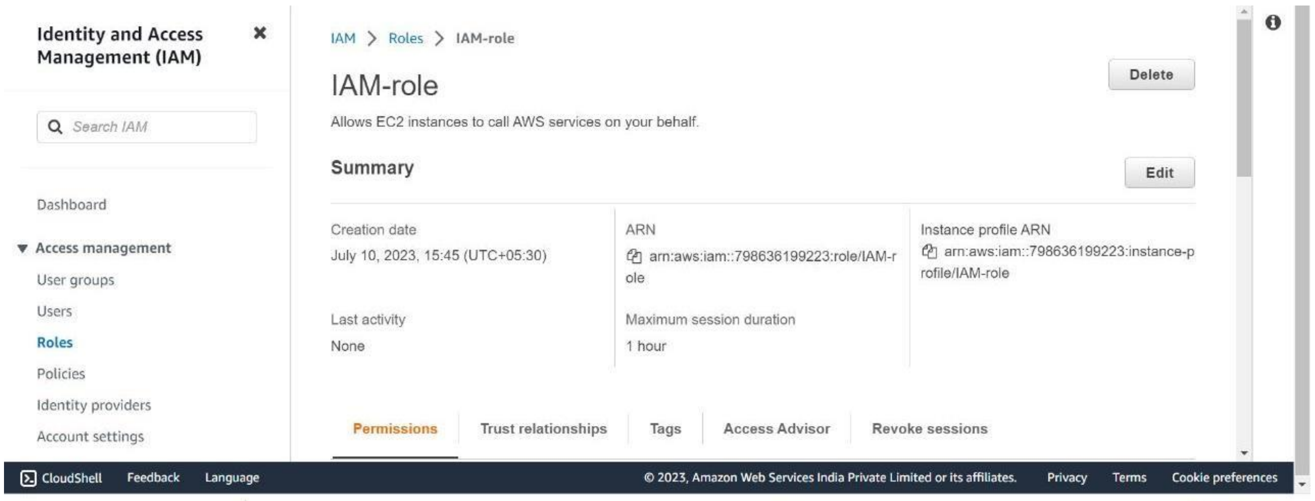
An Amazon S3 bucket is a <u>public cloud </u>storage resource available in Amazon Web Services (<u>AWS</u>) Simple Storage Service (<u>S3</u>) platform. It provides <u>object-based storage</u>, where data is stored inside S3 buckets in distinct units called objects instead of files.

To create an S3 bucket using an IAM role and upload an image to the bucket, you can follow these steps:

1. Navigate to the Amazon S3 service.
2. Click on "Create bucket" to start the bucket creation process.
3. Provide a unique name for your bucket.
4. Choose the region where you want your bucket to be located.
5. Configure any additional settings, such as versioning, logging, or encryption, if desired.
6. Under "Set permissions", select the option to grant access to an IAM role.
7. Choose the IAM role that has the necessary permissions to access the bucket.
8. Click on "Create" to create your bucket.

To create an IAM (Identity and Access Management) role in AWS:

1. Open the IAM service in the AWS Management Console.
2. Click on "Roles" in the left-hand navigation pane.
3. Click on "Create role" button.
4. Select the desired type of trusted entity (AWS service, another AWS account, or a web identity).
5. Choose the appropriate permissions for the role
6. Add tags (optional) to label and categorize the role.
7. Review the configuration and click on "Create role" to create the IAM role.



IAM-role

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date: July 10, 2023, 15:45 (UTC+05:30)

ARN: arn:aws:iam::798636199223:role/IAM-role

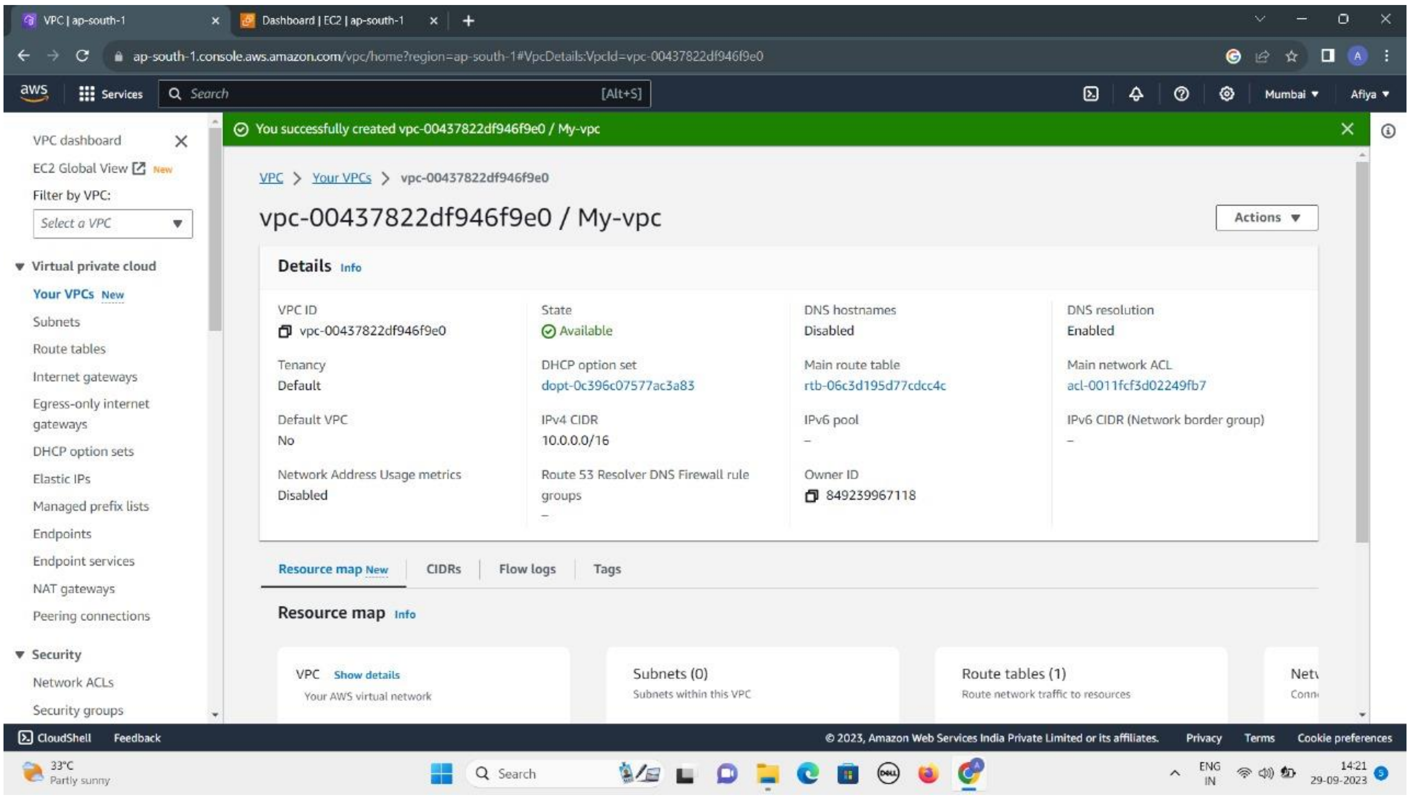
Instance profile ARN: arn:aws:iam::798636199223:instance-profile/IAM-role

Last activity: None

Maximum session duration: 1 hour

Once the bucket is created, you can upload an image to it by following these steps:

1. Select the bucket you just created from the list of buckets.
2. Click on the "Upload" button.
3. Choose the image file you want to upload from your local machine.
4. Configure any additional settings, such as access permissions or metadata, if desired.
5. Click on "Upload" to upload the image to the bucket.


You successfully created vpc-00437822df946f9e0 / My-vpc

VPC > Your VPCs > vpc-00437822df946f9e0

vpc-00437822df946f9e0 / My-vpc

Details Info

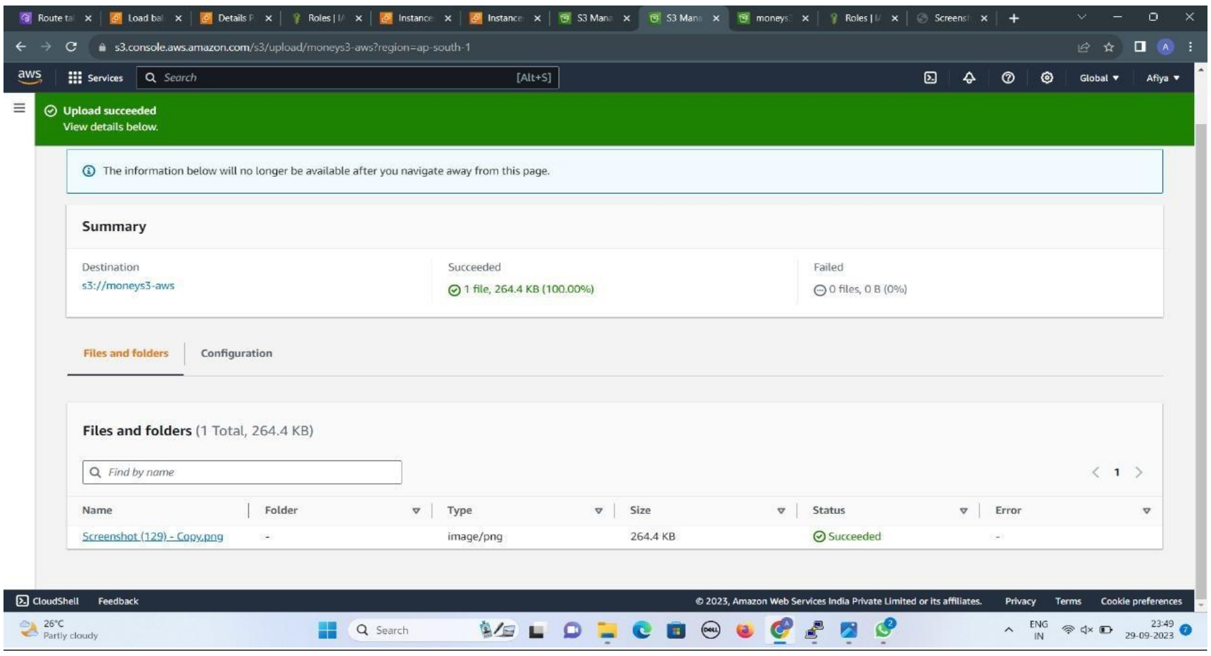
VPC ID	State	DNS hostnames	DNS resolution
vpc-00437822df946f9e0	Available	Disabled	Enabled
Tenancy	DHCP option set	Main route table	Main network ACL
Default	dopt-0c396c07577ac3a83	rtb-06c3d195d77cdcc4c	acl-0011fcf3d02249fb7
Default VPC	IPv4 CIDR	IPv6 pool	IPv6 CIDR (Network border group)
No	10.0.0.0/16	–	–
Network Address Usage metrics	Route 53 Resolver DNS Firewall rule groups	Owner ID	
Disabled	–	849239967118	

Resource map Info

VPC Show details — Your AWS virtual network

Subnets (0) — Subnets within this VPC

Route tables (1) — Route network traffic to resources



s3.console.aws.amazon.com/s3/upload/moneys3-aws?region=ap-south-1

Upload succeeded
View details below.

The information below will no longer be available after you navigate away from this page.

Summary

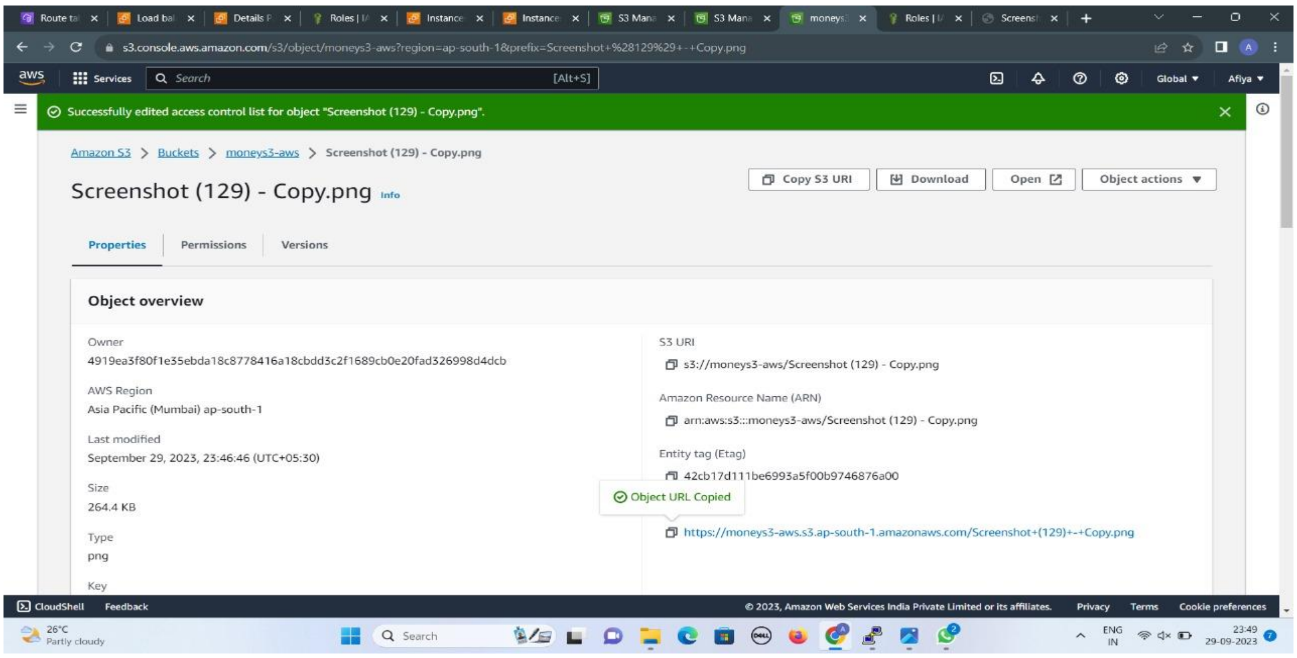
Destination	Succeeded	Failed
s3://moneys3-aws	1 file, 264.4 KB (100.00%)	0 files, 0 B (0%)

Files and folders | Configuration

Files and folders (1 Total, 264.4 KB)

Name	Folder	Type	Size	Status	Error
Screenshot (129) - Copy.png	-	image/png	264.4 KB	Succeeded	-

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



s3.console.aws.amazon.com/s3/object/moneys3-aws?region=ap-south-1&prefix=Screenshot+%28129%29+-+Copy.png

Successfully edited access control list for object "Screenshot (129) - Copy.png".

Amazon S3 > Buckets > moneys3-aws > Screenshot (129) - Copy.png

Screenshot (129) - Copy.png

Properties | Permissions | Versions

Object overview

Owner
4919ea3f80f1e35ebda18c8778416a18cbdd3c2f1689cb0e20fad326998d4dcb

AWS Region
Asia Pacific (Mumbai) ap-south-1

Last modified
September 29, 2023, 23:46:46 (UTC+05:30)

Size
264.4 KB

Type
png

Key

S3 URI
s3://moneys3-aws/Screenshot (129) - Copy.png

Amazon Resource Name (ARN)
arn:aws:s3:::moneys3-aws/Screenshot (129) - Copy.png

Entity tag (Etag)
42cb17d111be6993a5f00b9746876a00

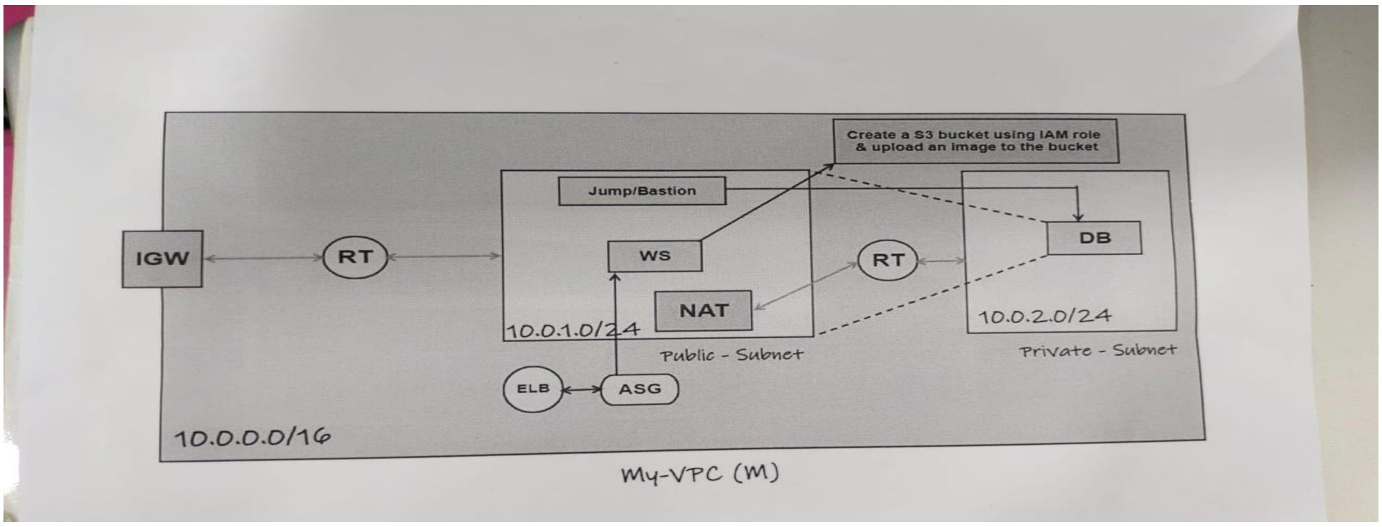
Object URL Copied

https://moneys3-aws.s3.ap-south-1.amazonaws.com/Screenshot+(129)+-+Copy.png

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

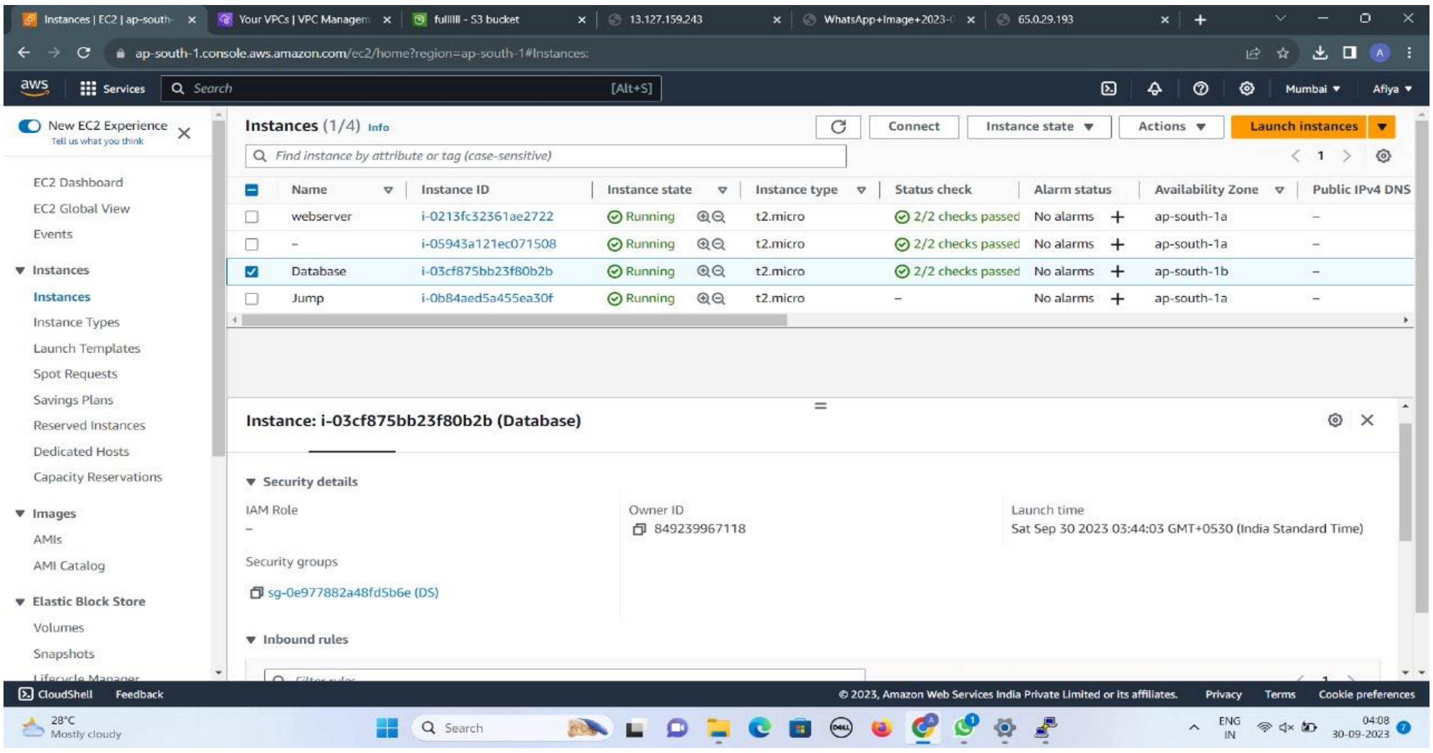
LEVEL-3

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.



Instances (1/4) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
webserver	i-0213fc32361ae2722	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
-	i-05943a121ec071508	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
Database	i-03cf875bb23f80b2b	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	–
Jump	i-0b84aed5a455ea30f	Running	t2.micro	–	No alarms	ap-south-1a	–

Instance: i-03cf875bb23f80b2b (Database)

Security details

IAM Role: –

Owner ID: 849239967118

Launch time: Sat Sep 30 2023 03:44:03 GMT+0530 (India Standard Time)

Security groups: sg-0e977882a48fd5b6e (DS)

Inbound rules

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

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.

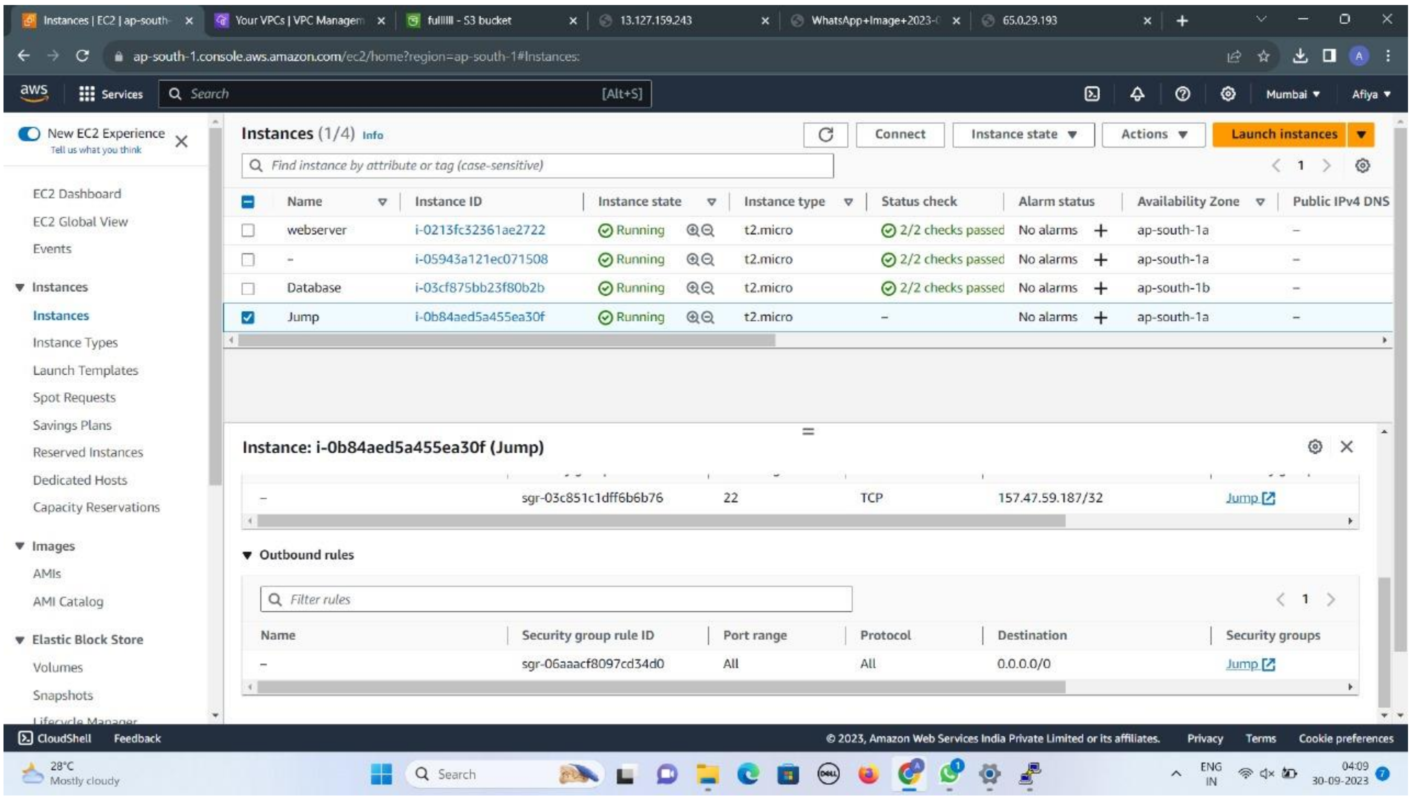
2. Configure security groups to control inbound and outbound traffic for the database server.

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.


ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#Instances:

Instances (1/4)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	webserver	i-0213fc32361ae2722	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
☐	-	i-05943a121ec071508	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
☐	Database	i-03cf875bb23f80b2b	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	–
☑	Jump	i-0b84aed5a455ea30f	Running	t2.micro	–	No alarms	ap-south-1a	–

Instance: i-0b84aed5a455ea30f (Jump)

–	sgr-03c851c1dff6b6b76	22	TCP	157.47.59.187/32	Jump

Outbound rules

Name	Security group rule ID	Port range	Protocol	Destination	Security groups
–	sgr-06aaacf8097cd34d0	All	All	0.0.0.0/0	Jump

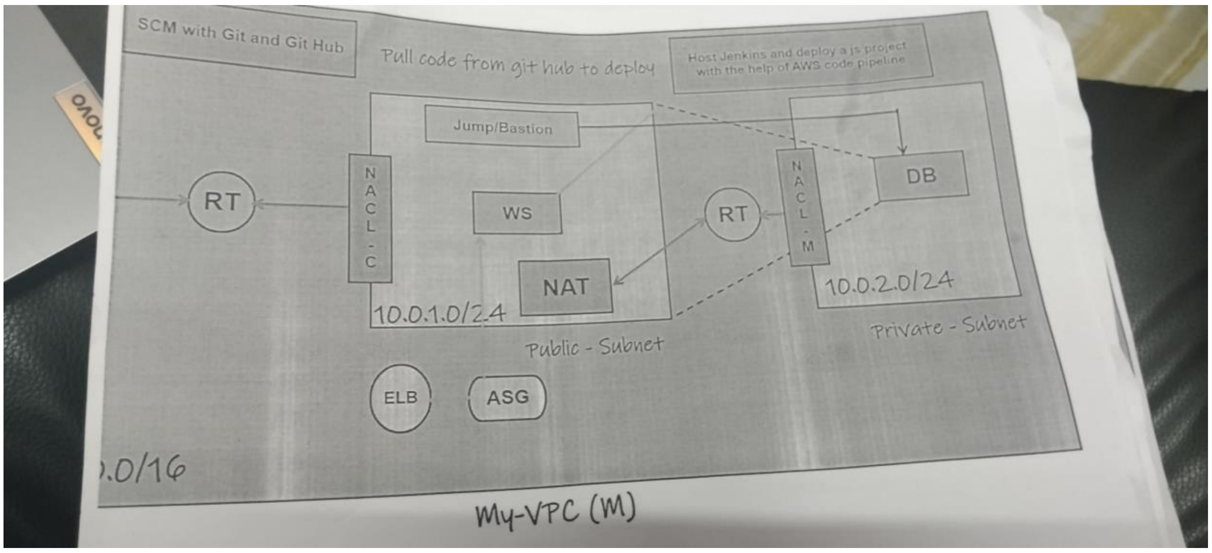
© 2023, Amazon Web Services India Private Limited or its affiliates.

To set up a jump bastion box in a VPC:

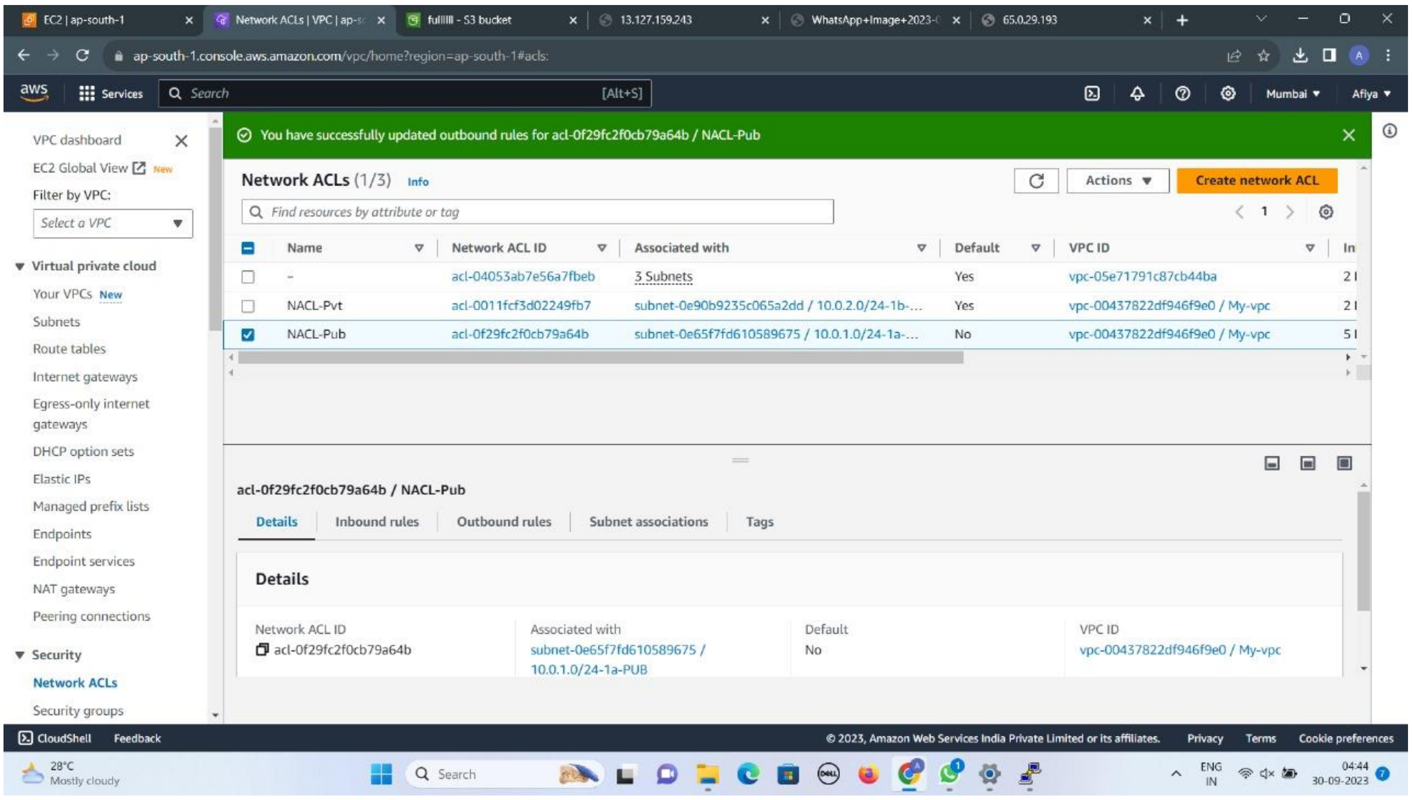
1. Launch an EC2 instance to serve as the jump bastion box, selecting an appropriate AMI and instance type.
2. Configure security groups for the jump bastion box to control inbound and outbound traffic.
3. Set up SSH key-based authentication to securely access the jump bastion box.
4. Configure the network 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

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 other will be he 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.



ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#acls:

VPC dashboard
EC2 Global View New
Filter by VPC: Select a VPC

Virtual private cloud
- Your VPCs New
- Subnets
- Route tables
- Internet gateways
- Egress-only internet gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- Endpoints
- Endpoint services
- NAT gateways
- Peering connections

Security
- Network ACLs
- Security groups

You have successfully updated outbound rules for acl-0f29fc2f0cb79a64b / NACL-Pub

Network ACLs (1/3) Info

	Name	Network ACL ID	Associated with	Default	VPC ID
☐	–	acl-04053ab7e56a7fbeb	3 Subnets	Yes	vpc-05e71791c87cb44ba
☐	NACL-Pvt	acl-0011fcf3d02249fb7	subnet-0e90b9235c065a2dd / 10.0.2.0/24-1b-...	Yes	vpc-00437822df946f9e0 / My-vpc
☑	NACL-Pub	acl-0f29fc2f0cb79a64b	subnet-0e65f7fd610589675 / 10.0.1.0/24-1a-...	No	vpc-00437822df946f9e0 / My-vpc

acl-0f29fc2f0cb79a64b / NACL-Pub

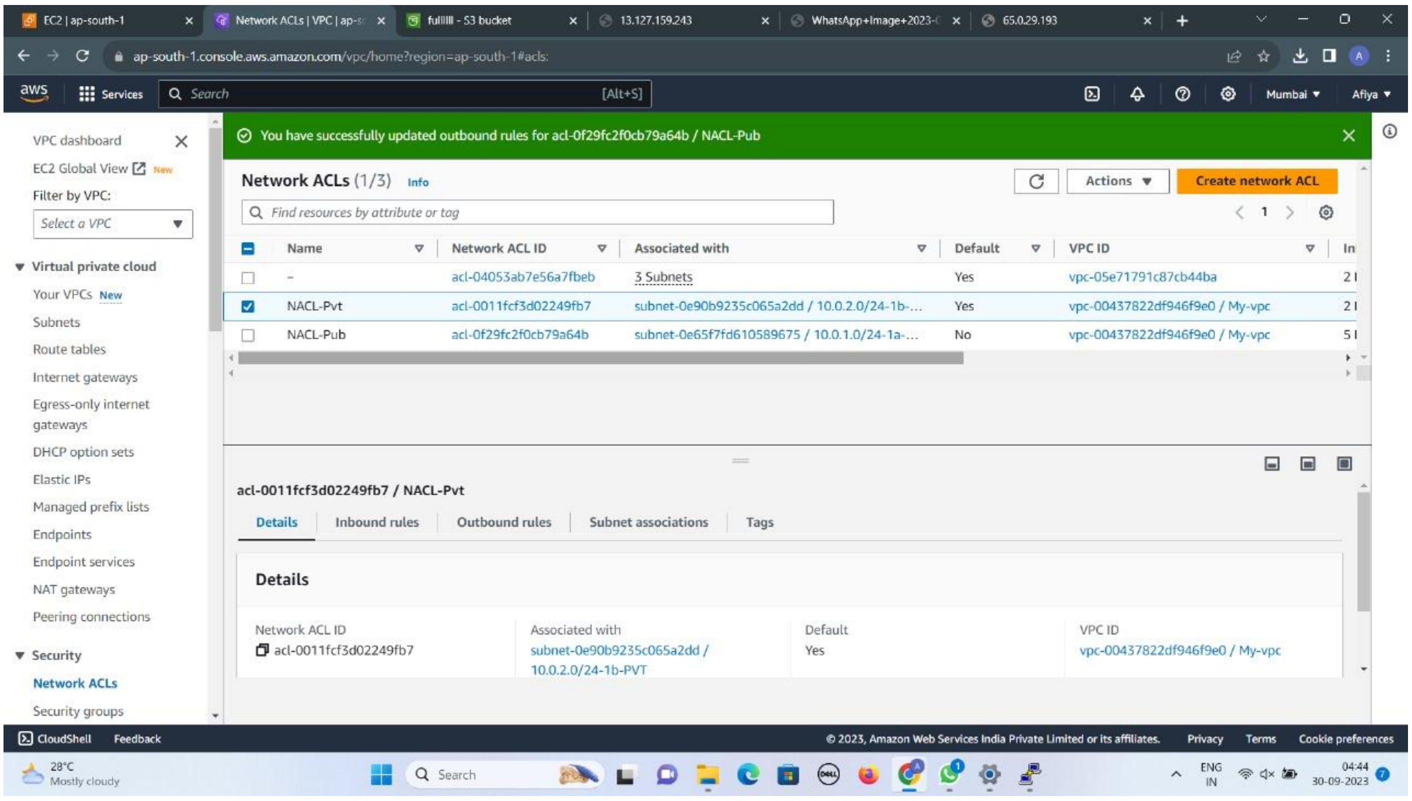
Details | Inbound rules | Outbound rules | Subnet associations | Tags

Details

Network ACL ID	Associated with	Default	VPC ID
acl-0f29fc2f0cb79a64b	subnet-0e65f7fd610589675 / 10.0.1.0/24-1a-PUB	No	vpc-00437822df946f9e0 / My-vpc

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

28°C Mostly cloudy 04:44 30-09-2023



ap-south-1.console.aws.amazon.com/vpc/home?region=ap-south-1#acls:

You have successfully updated outbound rules for acl-0f29fc2f0cb79a64b / NACL-Pub

Network ACLs (1/3) Info

	Name	Network ACL ID	Associated with	Default	VPC ID
☐	–	acl-04053ab7e56a7fbeb	3 Subnets	Yes	vpc-05e71791c87cb44ba
☑	NACL-Pvt	acl-0011fcf3d02249fb7	subnet-0e90b9235c065a2dd / 10.0.2.0/24-1b-...	Yes	vpc-00437822df946f9e0 / My-vpc
☐	NACL-Pub	acl-0f29fc2f0cb79a64b	subnet-0e65f7fd610589675 / 10.0.1.0/24-1a-...	No	vpc-00437822df946f9e0 / My-vpc

acl-0011fcf3d02249fb7 / NACL-Pvt

Details | Inbound rules | Outbound rules | Subnet associations | Tags

Details

Network ACL ID	Associated with	Default	VPC ID
acl-0011fcf3d02249fb7	subnet-0e90b9235c065a2dd / 10.0.2.0/24-1b-PVT	Yes	vpc-00437822df946f9e0 / My-vpc

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

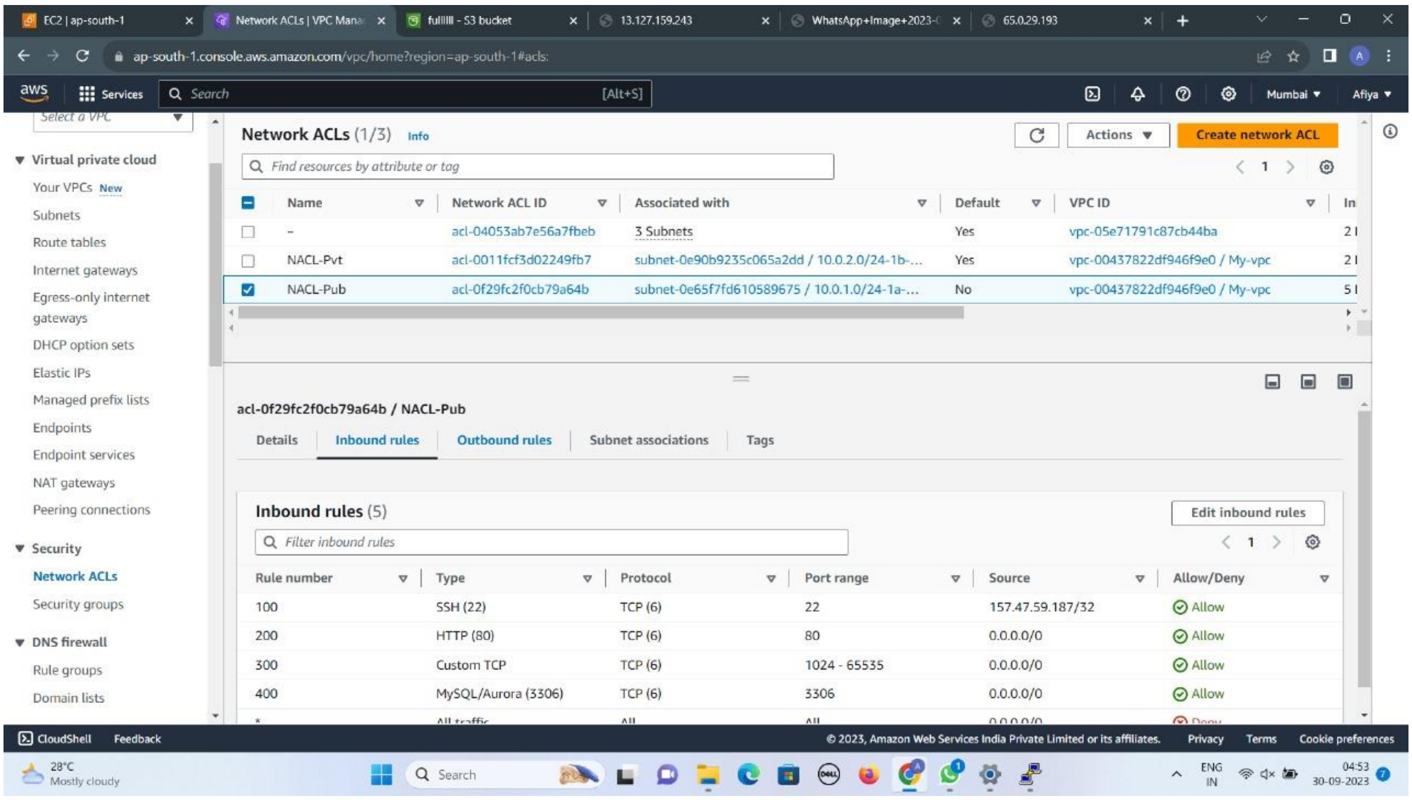
28°C Mostly cloudy 04:44 30-09-2023

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.



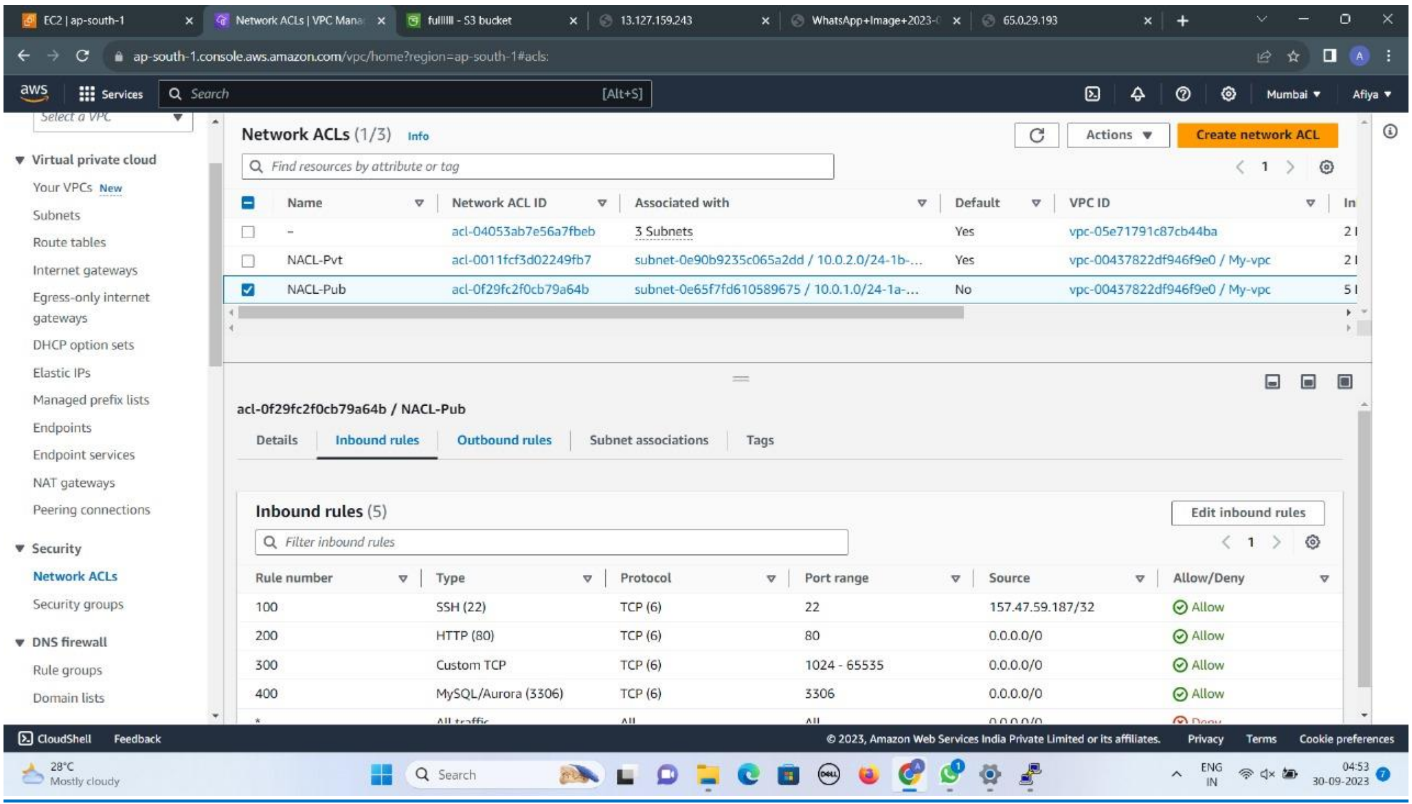
Network ACLs (1/3)

Name	Network ACL ID	Associated with	Default	VPC ID	In
-	acl-04053ab7e56a7fbeb	3 Subnets	Yes	vpc-05e71791c87cb44ba	2 I
NACL-Pvt	acl-0011fcf3d02249fb7	subnet-0e90b9235c065a2dd / 10.0.2.0/24-1b-...	Yes	vpc-00437822df946f9e0 / My-vpc	2 I
NACL-Pub	acl-0f29fc2f0cb79a64b	subnet-0e65f7fd610589675 / 10.0.1.0/24-1a-...	No	vpc-00437822df946f9e0 / My-vpc	5 I

acl-0f29fc2f0cb79a64b / NACL-Pub

Inbound rules (5)

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	SSH (22)	TCP (6)	22	157.47.59.187/32	Allow
200	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
300	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
400	MySQL/Aurora (3306)	TCP (6)	3306	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny



Network ACLs (1/3)

	Name	Network ACL ID	Associated with	Default	VPC ID	In
☐	-	acl-04053ab7e56a7fbeb	3 Subnets	Yes	vpc-05e71791c87cb44ba	2 I
☐	NACL-Pvt	acl-0011fcf3d02249fb7	subnet-0e90b9235c065a2dd / 10.0.2.0/24-1b-...	Yes	vpc-00437822df946f9e0 / My-vpc	2 I
☑	NACL-Pub	acl-0f29fc2f0cb79a64b	subnet-0e65f7fd610589675 / 10.0.1.0/24-1a-...	No	vpc-00437822df946f9e0 / My-vpc	5 I

acl-0f29fc2f0cb79a64b / NACL-Pub

Inbound rules (5)

Rule number	Type	Protocol	Port range	Source	Allow/Deny
100	SSH (22)	TCP (6)	22	157.47.59.187/32	Allow
200	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
300	Custom TCP	TCP (6)	1024 - 65535	0.0.0.0/0	Allow
400	MySQL/Aurora (3306)	TCP (6)	3306	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

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, and CodePipeline 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.

<<<Thank You>>>