SRI VASAVI ENGINEERING COLLEGE

REPORT FOR HACKTHON ON DEVOPS WITH AWS

AWS HACKATHON DOCUMENTATION

TEAM MEMBERS:

G.ANU DEEPIKA

B.UMA DEVI

K.SRIRAM

M.TARUN

R.KISHORE

M.JOGENDRA

SEPTEMBER 29, 2023

BRAINOVISION

AWS PROJECT REPORT – DOCUMENTATION

What Is AWS?

AWS (Amazon Web Services) is a comprehensive, evolving cloud computing platform provided by Amazon that includes a mixture of infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS) and packaged-software-as-a-service (SaaS) offerings. AWS services can offer an organization tools such as compute power, database storage and content delivery services.

Amazon.com Web Services launched its first web services in 2002 from the internal infrastructure that Amazon.com built to handle its online retail operations. In 2006, it began offering its defining IaaS services

AWS TOOLS

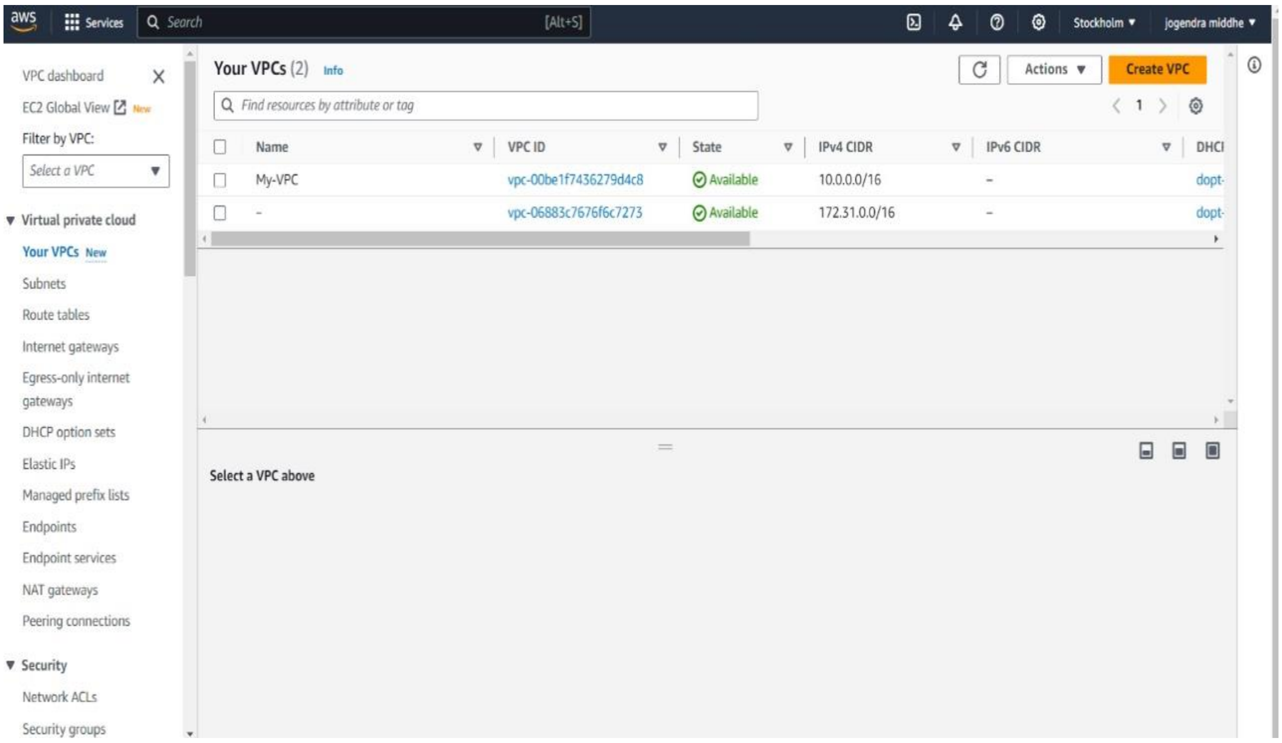
AWS provides a wide range of tools and services However, there are a few tools that are widely used and considered fundamental in many AWS deployments. Here are some of the most commonly used tools of AWS:

- Amazon EC2 (Elastic Compute Cloud)
  It provides a virtual server in the cloud, allowing you to run applications and services on a variety of operating systems.it also used to create various instances.
- Amazon RDS (Relational Database Service)
  It Provides managed database services for various relational database engines, including Amazon Aurora, MySQL, PostgreSQL, Oracle, and SQL Server.
- Amazon VPC (Virtual Private Cloud)
  It Offers a logically isolated virtual network within the AWS cloud. It allows you to launch AWS resources in a defined virtual network, providing control over IP addressing, subnets, and network gateways.
- *Amazon SQS (Simple Queue Service)*
  It A fully managed message queuing service that enables you to decouple and scale microservices, distributed systems, and serverless applications.
- AWS CloudTrail
  It Captures and logs API activity and events within your AWS infrastructure, providing audit trails for compliance, security analysis, and troubleshooting.

STEP-1:

Creating a Virtual Private Cloud(VPC):

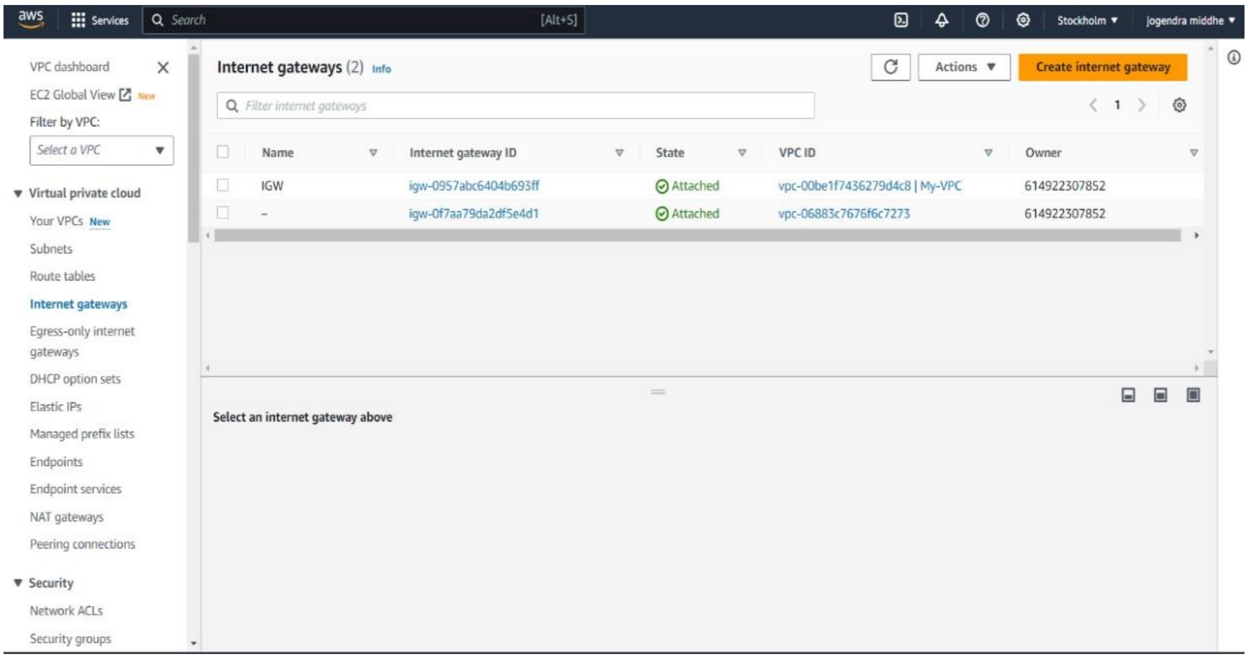
- Go to VPC dashboard click on the “create VPC” to start the VPC creation.
- Configure the VPC settings:
  - Give a name for your VPC.
  - Specify the IPv4 CIDR block for your VPC's IP address range (10.0.0.0/16).
  - you can also assign an IPv6 CIDR block to your VPC.
- Configure the VPC's subnets:
  - Specify the IPv4 CIDR block for your first subnet (e.g., 10.0.0.0/24).
  - Choose the availability zone where you want to create the subnet.
  - Repeat this step to create additional subnets if needed.
- Configure the VPC's route table:
  - Create a new route table or select an existing one.
  - Associate the subnets created in the previous step with the route table
- Configure the VPC's internet gateway:
  - Create a new internet gateway or select an existing one.
  - Attach the internet gateway to your VPC.
- Configure the VPC's security groups:
  - Create new security groups or select existing ones.
  - Define the inbound and outbound rules for each security group to control network traffic.
- Review all the configuration details and settings for your VPC. If everything looks correct, click on the "Create VPC" button to create your VPC.



Your VPCs (2) Info

Name	VPC ID	State	IPv4 CIDR	IPv6 CIDR	DHCI
My-VPC	vpc-00be1f7436279d4c8	Available	10.0.0.0/16	-	dopt-
-	vpc-06883c7676f6c7273	Available	172.31.0.0/16	-	dopt-

INTERNET GATEWAYS (IGW)



Internet gateways (2) Info

Name	Internet gateway ID	State	VPC ID	Owner
IGW	igw-0957abc6404b693ff	Attached	vpc-00be1f7436279d4c8 | My-VPC	614922307852
-	igw-0f7aa79da2df5e4d1	Attached	vpc-06883c7676f6c7273	614922307852

ROUTE TABLES (RT):

➢ Go to the "Route Tables" section: Within the selected VPC, click on the "Route Tables" option in the left navigation menu. This will display the list of existing route tables in the selected VPC**.**

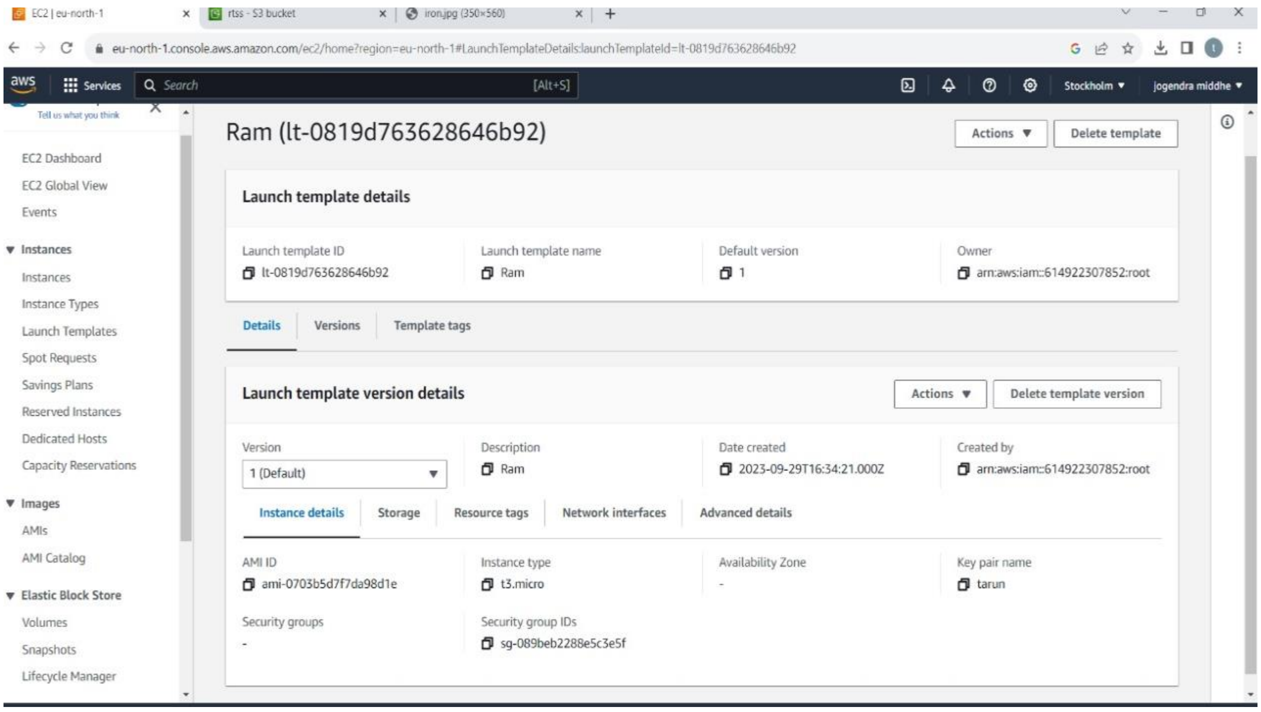
- Create a new route table: Click on the "Create Route Table" button to create a new route table within the selected VPC.
- Configure the route table settings:
  - Provide a name for the route table to identify it.
  - Select the VPC in which you want to create the route table.
  - Choose the desired subnet associations for the route table. Subnets can be associated with multiple route tables, and each subnet must be associated with at least one route table.
- Configure the routes:
  - Click on the "Edit routes" button to add or edit routes in the route table.
  - Add the desired routes by specifying the destination IP range and the target (e.g., an internet gateway, a virtual private gateway, or a NAT gateway)
- Save the route table: Click on the "Save" button to save the configured route table.
- Associate subnets with the route table:
  - In the "Associations" tab of the route table, click on the "Edit subnet associations" button.
  - Select the subnets you want to associate with the route table and click on the "Save" button.
- Review the route table: Verify the route table settings, associations, and routes in the AWS Management Console.

- Create a new route table: Click on the "Create Route Table" button to create a new route table within the selected VPC.

<u>STEP 2:</u>

Create an Auto Scaling Group:

- In the EC2 Auto Scaling console, click on "Auto Scaling Groups" in the sidebar.
- Click on "Create an Auto Scaling group" .



eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#LaunchTemplateDetails:launchTemplateId=lt-0819d763628646b92

Ram (lt-0819d763628646b92)

Launch template details

Launch template ID: lt-0819d763628646b92
Launch template name: Ram
Default version: 1
Owner: arn:aws:iam::614922307852:root

Launch template version details

Version: 1 (Default)
Description: Ram
Date created: 2023-09-29T16:34:21.000Z
Created by: arn:aws:iam::614922307852:root

AMI ID: ami-0703b5d7f7da98d1e
Instance type: t3.micro
Availability Zone: -
Key pair name: tarun
Security groups: -
Security group IDs: sg-089beb2288e5c3e5f

- Select the launch configuration you created in the previous step.
- Configure the desired minimum, maximum, and desired capacity of instances.
- Configure any additional settings such as scaling policies and tags.
- Save your Auto Scaling group.

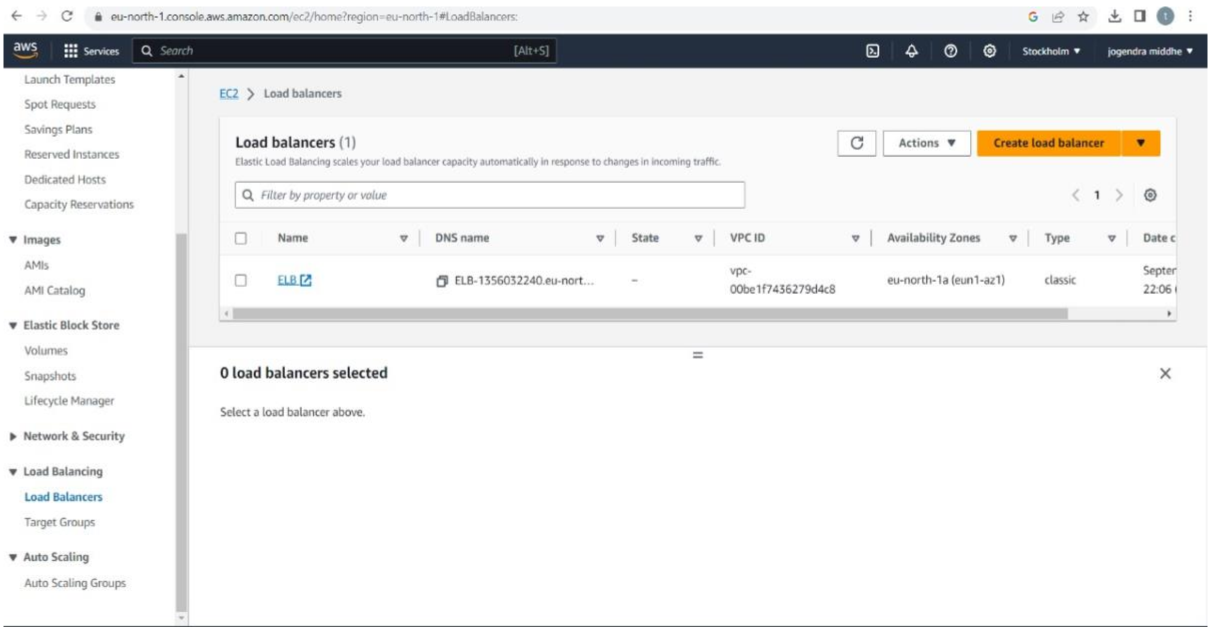
Create an Elastic Load Balancer:

- Step 1: Sign in to the AWS Management Console
- Log in to your AWS account using your credentials.
-
- Step 2: Navigate to the Load Balancers Console
-
- Go to the AWS Management Console.
- In the "Services" menu, select "EC2."
- Under the "Load Balancing" section in the EC2 Dashboard, click "Load Balancers."

- Step 3: Create a Load Balancer
-
- Click the "Create Load Balancer" button.
- Step 4: Choose Load Balancer Type
-
- Select "Application Load Balancer" as the load balancer type.
- Step 5: Configure Load Balancer
-
- Provide a name for your load balancer.
- Select the VPC (Virtual Private Cloud) where you want to create the load balancer.
- Configure listeners (the protocol and port) for your load balancer. For example, you might set up HTTP on port 80 and HTTPS on port 443.
- Choose a security policy (if using HTTPS).
- Configure Availability Zones where the load balancer will distribute traffic.
- Optionally, configure IP address type and idle timeout.
- Step 6: Configure Routing
-
- Set up target groups to route traffic to instances. You can create new target groups or use existing ones.
- Define target group properties, including the protocol and port for communication with instances.
- Optionally, configure health checks to ensure instances are healthy.
- Step 7: Configure Advanced Settings (Optional)

-
- Optionally, configure advanced settings such as connection draining, sticky sessions, and cross-zone load balancing.
- Step 8: Add Tags (Optional)
-
- Optionally, add tags to your load balancer for better organization and management.
- Step 9: Review and Create
-
- Review all your settings to ensure they're correct.
- Click the "Create" button to create the Application Load Balancer.
- Step 10: Wait for Creation
- AWS will create your load balancer, which may take a few minutes. Once it's created, you can see its status in the Load Balancers console.
-
- Your Application Load Balancer is now set up and ready to distribute traffic to the instances in your target group. Make sure to update your DNS records or point your applications to the DNS name provided by the load balancer to start routing traffic through it.
-
- Remember that this guide is for creating an Application Load Balancer. If you need to create a Network Load Balancer or a Classic Load Balancer, the steps will be slightly different, but the general process

is similar. Choose the appropriate load balancer type based on your specific requirements.



Load balancers (1)

Elastic Load Balancing scales your load balancer capacity automatically in response to changes in incoming traffic.

Name	DNS name	State	VPC ID	Availability Zones	Type	Date c
ELB	ELB-1356032240.eu-nort...	-	vpc-00be1f7436279d4c8	eu-north-1a (eun1-az1)	classic	Septer 22:06

0 load balancers selected

Select a load balancer above.

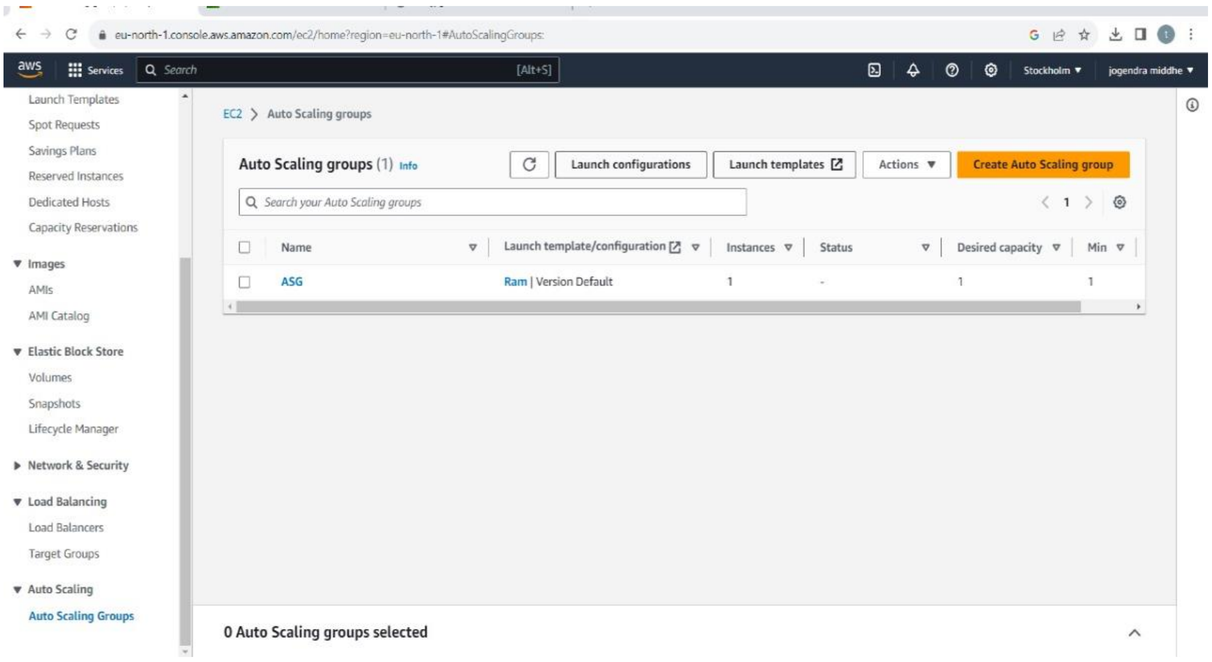
Configure the Auto Scaling Group with the Load Balancer:

- Step 1: Sign in to the AWS Management Console
- Log in to your AWS account using your credentials.
-
- Step 2: Navigate to the EC2 Auto Scaling Console
-
- In the AWS Management Console, go to the "Services" menu.
- Under "Compute," select "EC2 Auto Scaling."
- Step 3: Create an Auto Scaling Group
-
- Click on the "Create Auto Scaling group" button to start the wizard.
-

- Step 4: Choose a Launch Template or Launch Configuration
-
- You need to specify the Amazon Machine Image (AMI), instance type, and other details for your instances. You can choose to use an existing launch template or launch configuration, or you can create a new one.
-
- Using an Existing Launch Template or Launch Configuration:
-
- Select the launch template or launch configuration from the list.
- Creating a New Launch Template:
-
- Click on "Create a launch template."
- Fill in the required details, such as the AMI, instance type, key pair, security groups, etc.
- Save the launch template.
- Step 5: Configure Auto Scaling Group Details
-
- Specify a name for your Auto Scaling Group.
- Set the desired capacity, minimum size, and maximum size. These define how many instances the group should maintain.
- Configure network settings, including the VPC, subnets, and security groups.

- Optionally, configure advanced details like IAM roles and instance termination policies.
- Step 6: Configure Scaling Policies
-
- Define scaling policies for your Auto Scaling Group. You can create both target tracking and step scaling policies to automatically adjust the group's size based on metrics like CPU utilization or custom CloudWatch metrics.
- Step 7: Configure Notifications (Optional)
-
- Set up Amazon SNS notifications to be alerted when scaling events occur.
-
- Step 8: Add Tags (Optional)
-
- Add tags to your Auto Scaling Group for better organization and management.
-
- Step 9: Review and Create
-
- Review all your settings to ensure they're correct. Make sure to double-check your scaling policies and notifications.
-
- Click "Create Auto Scaling group" to create the group.
-
- Step 10: Monitor the Auto Scaling Group
-

- Once your Auto Scaling Group is created, it will start launching instances based on your configuration. You can monitor the scaling activities and instances in the ASG console.
-
- Your Auto Scaling Group is now set up, and it will automatically adjust the number of instances based on your scaling policies to meet the desired capacity and respond to scaling events.
-
- Remember to configure your launch templates or configurations, security groups, IAM roles, and other settings appropriately to ensure your instances behave as expected within your application environment.

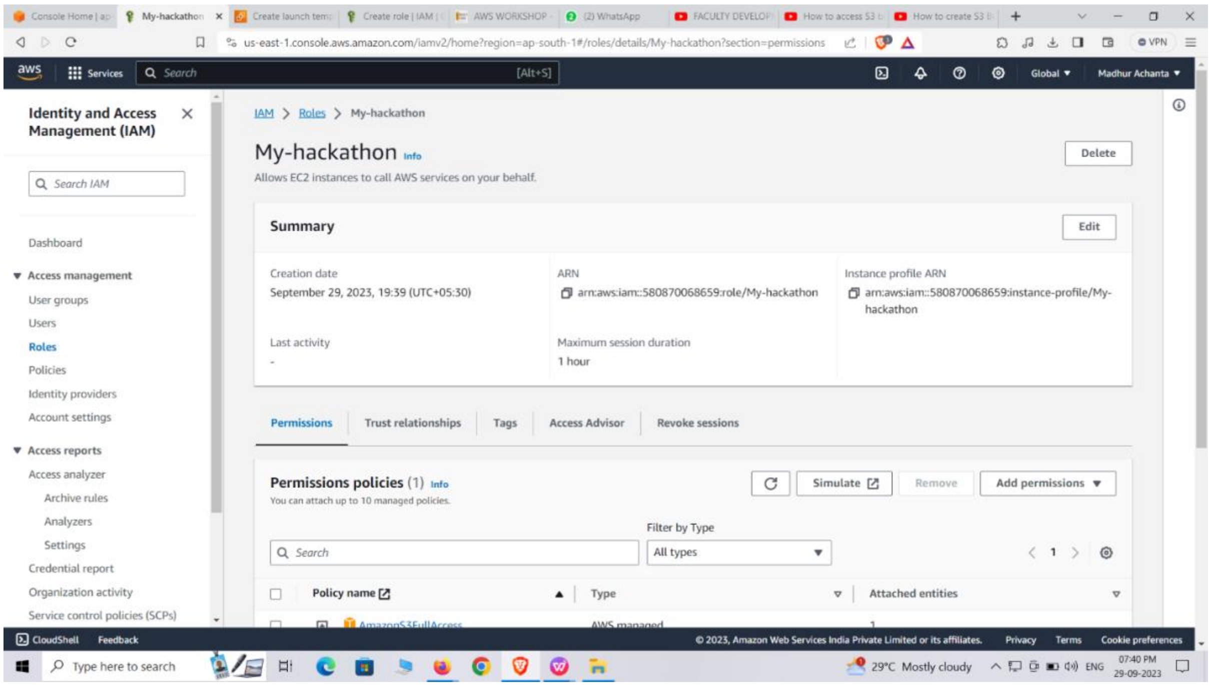


Auto Scaling groups (1) Info

Name	Launch template/configuration	Instances	Status	Desired capacity	Min
ASG	Ram | Version Default	1	-	1	1

0 Auto Scaling groups selected

<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.



us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/roles/details/My-hackathon?section=permissions

My-hackathon

Allows EC2 instances to call AWS services on your behalf.

Summary

Creation date: September 29, 2023, 19:39 (UTC+05:30)

ARN: arn:aws:iam::580870068659:role/My-hackathon

Instance profile ARN: arn:aws:iam::580870068659:instance-profile/My-hackathon

Last activity: -

Maximum session duration: 1 hour

Permissions policies (1)

You can attach up to 10 managed policies.

Filter by Type: All types

Policy name	Type	Attached entities
AmazonS3FullAccess	AWS managed	1

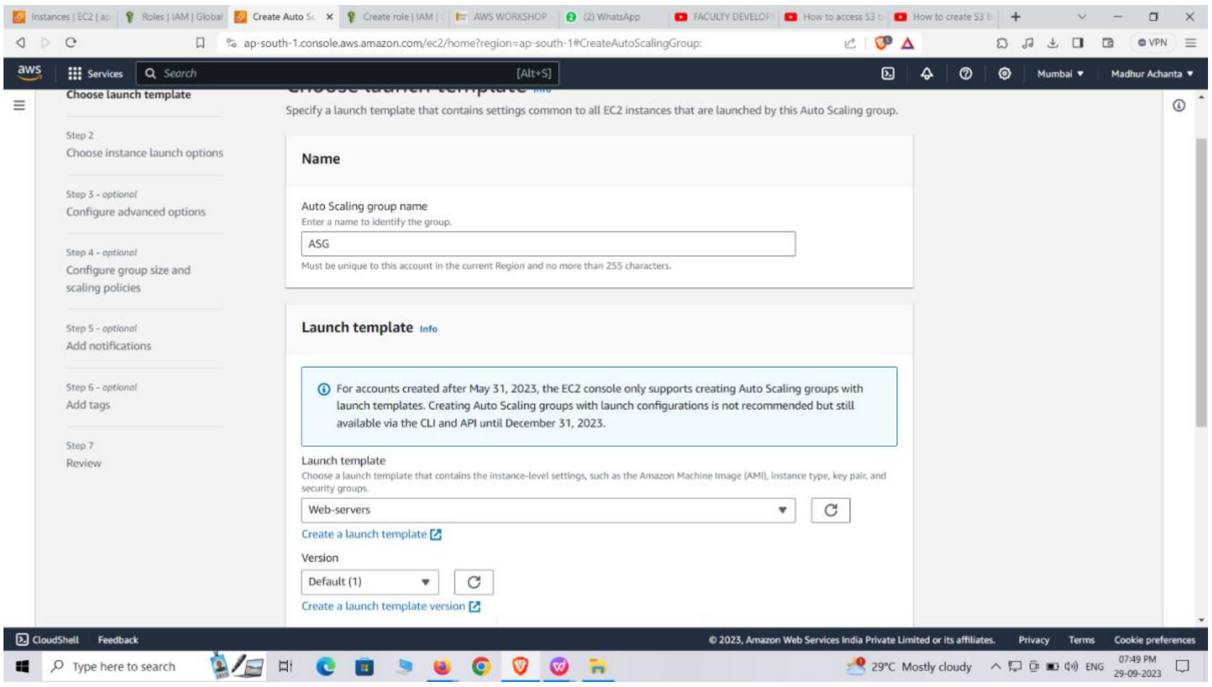
© 2023, Amazon Web Services India Private Limited or its affiliates.

<u>To set up a web server in AWS:</u>

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>CREATING IAM:</u>


Choose launch template

Specify a launch template that contains settings common to all EC2 instances that are launched by this Auto Scaling group.

Name

Auto Scaling group name
Enter a name to identify the group.
ASG
Must be unique to this account in the current Region and no more than 255 characters.

Launch template Info

For accounts created after May 31, 2023, the EC2 console only supports creating Auto Scaling groups with launch templates. Creating Auto Scaling groups with launch configurations is not recommended but still available via the CLI and API until December 31, 2023.

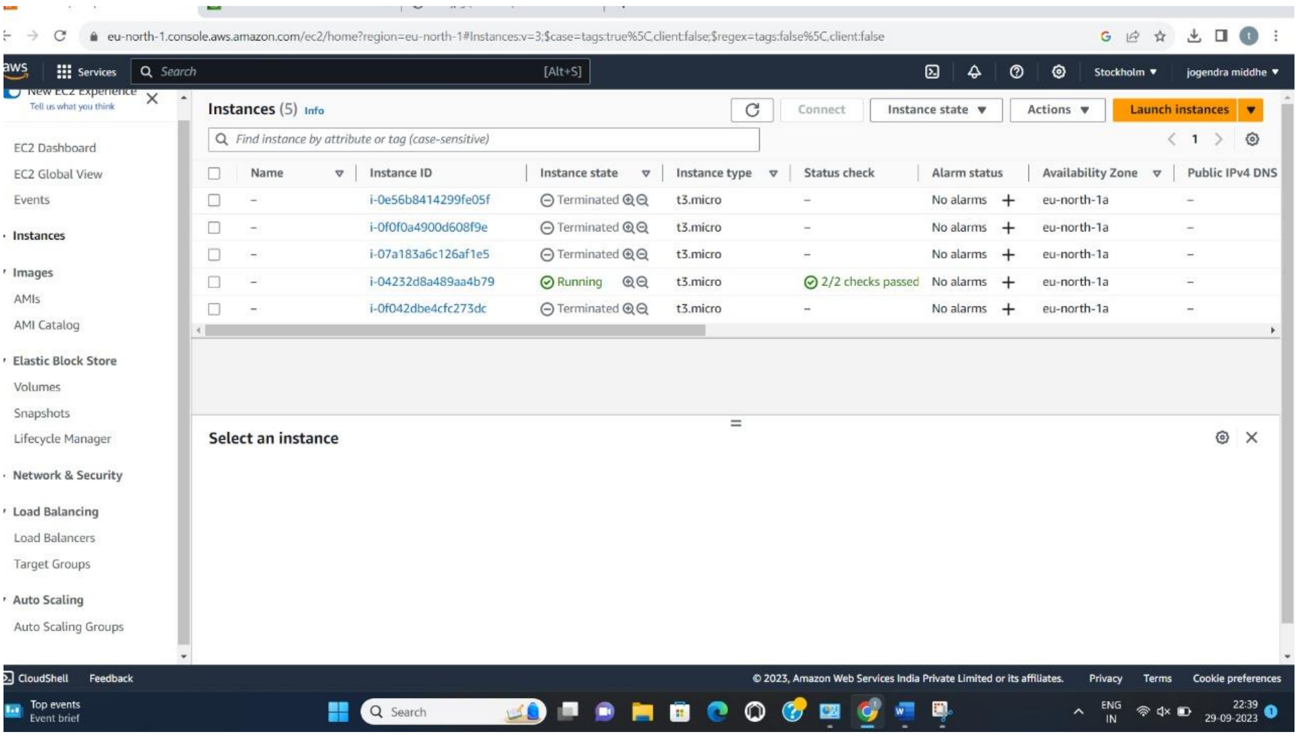
Launch template
Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
Web-servers
Create a launch template

Version
Default (1)
Create a launch template version

After following the above steps, we can connect to the webserver

USER DATA:

```
sudo su
yum update -y
yum install httpd -y
cd/var/www/html
echo “how”>index.html
service httpd start
chkconfig httpd on
```



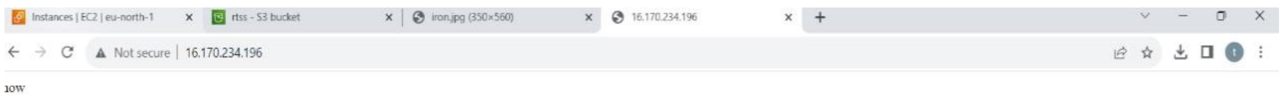
eu-north-1.console.aws.amazon.com/ec2/home?region=eu-north-1#Instances:v=3;$case=tags:true%5C,client:false;$regex=tags:false%5C,client:false

Instances (5) Info

Find instance by attribute or tag (case-sensitive)

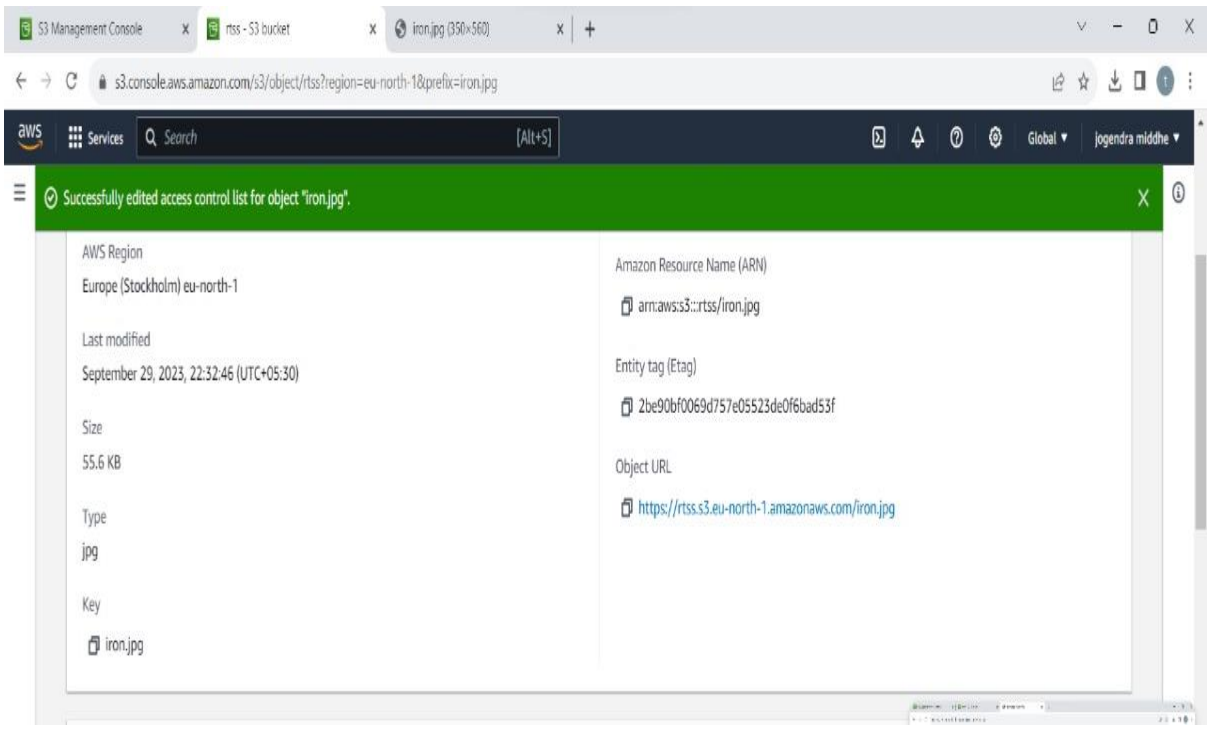
Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
–	i-0e56b8414299fe05f	Terminated	t3.micro	–	No alarms	eu-north-1a	–
–	i-0f0f0a4900d608f9e	Terminated	t3.micro	–	No alarms	eu-north-1a	–
–	i-07a183a6c126af1e5	Terminated	t3.micro	–	No alarms	eu-north-1a	–
–	i-04232d8a489aa4b79	Running	t3.micro	2/2 checks passed	No alarms	eu-north-1a	–
–	i-0f042dbe4cfc273dc	Terminated	t3.micro	–	No alarms	eu-north-1a	–

Select an instance



16.170.234.196

low



Successfully edited access control list for object "iron.jpg".

AWS Region
Europe (Stockholm) eu-north-1

Last modified
September 29, 2023, 22:32:46 (UTC+05:30)

Size
55.6 KB

Type
jpg

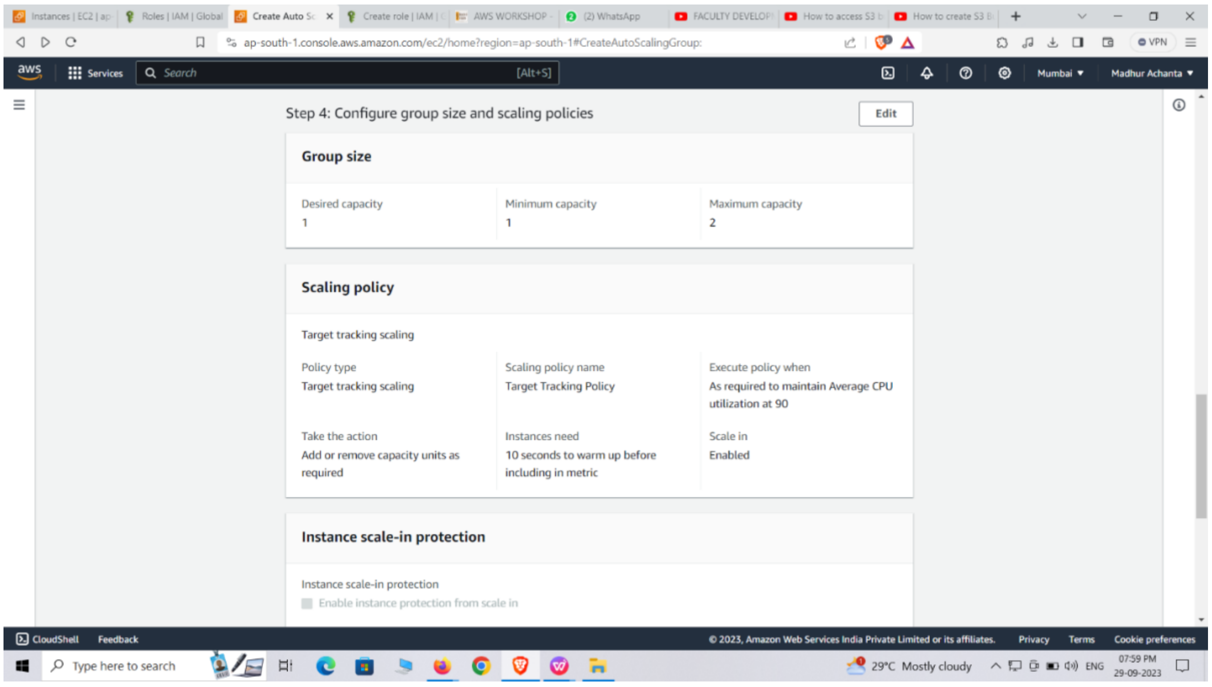
Key
iron.jpg

Amazon Resource Name (ARN)
arn:aws:s3:::rtss/iron.jpg

Entity tag (Etag)
2be90bf0069d757e05523de0f6bad53f

Object URL
https://rtss.s3.eu-north-1.amazonaws.com/iron.jpg

CREATING BUCKET (S3):



Step 4: Configure group size and scaling policies

Group size

Desired capacity	Minimum capacity	Maximum capacity
1	1	2

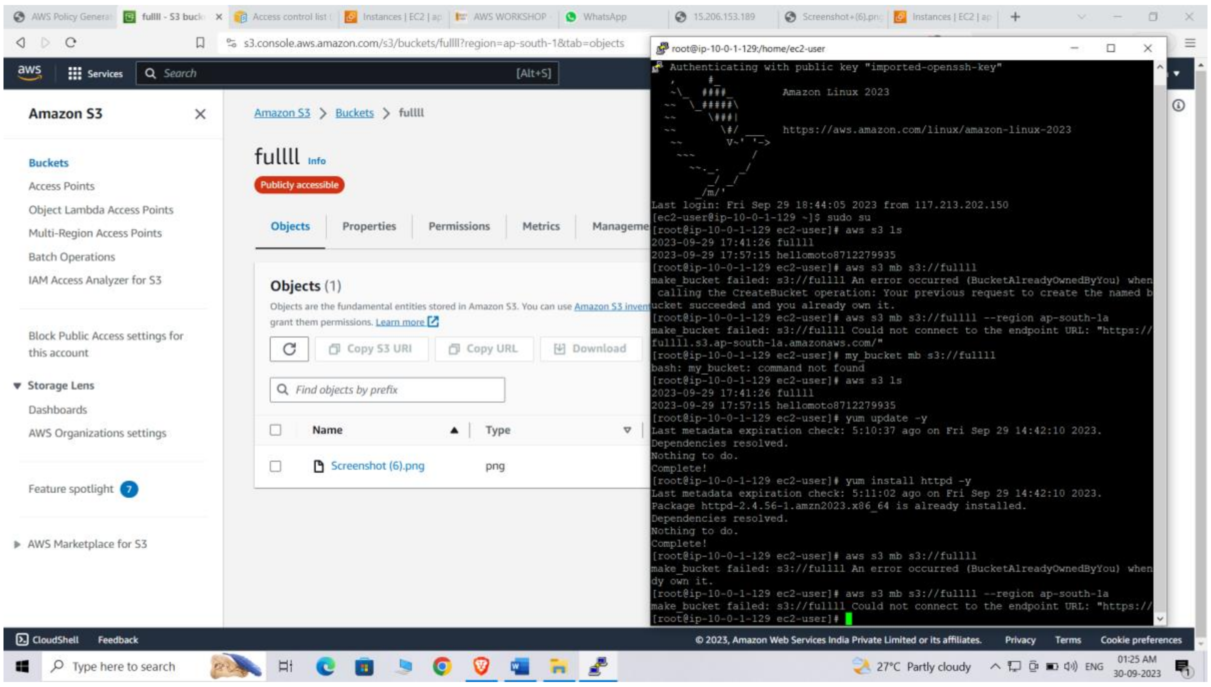
Scaling policy

Target tracking scaling

Policy type Target tracking scaling	Scaling policy name Target Tracking Policy	Execute policy when As required to maintain Average CPU utilization at 90
Take the action Add or remove capacity units as required	Instances need 10 seconds to warm up before including in metric	Scale in Enabled

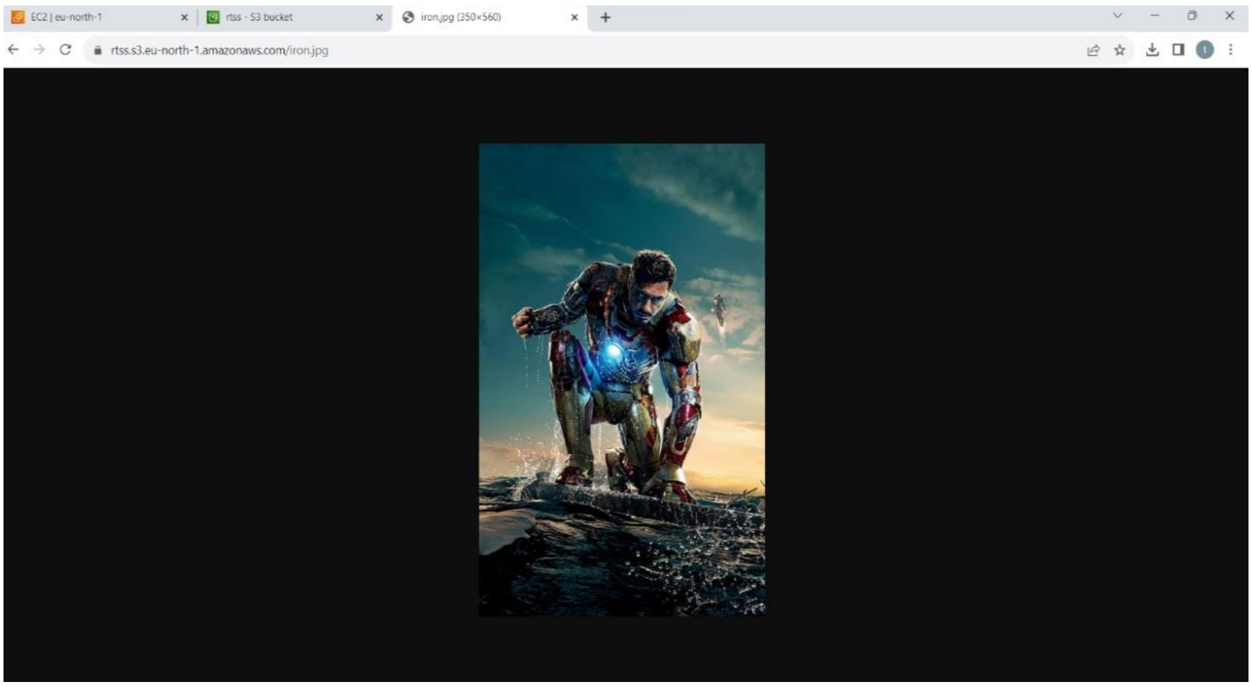
Instance scale-in protection

Instance scale-in protection
Enable instance protection from scale in


s3.console.aws.amazon.com/s3/buckets/fullll?region=ap-south-1&tab=objects
Amazon S3
Buckets
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
AWS Organizations settings
Feature spotlight
AWS Marketplace for S3
Amazon S3 > Buckets > fullll
fullll Info
Publicly accessible
Objects
Properties
Permissions
Metrics
Manageme
Objects (1)
Copy S3 URI
Copy URL
Download
Find objects by prefix
Name
Type
Screenshot (6).png
png
root@ip-10-0-1-129:/home/ec2-user

```
Authenticating with public key "imported-openssh-key"
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
Last login: Fri Sep 29 18:44:05 2023 from 117.213.202.150
[ec2-user@ip-10-0-1-129 ~]$ sudo su
[root@ip-10-0-1-129 ec2-user]# aws s3 ls
2023-09-29 17:41:26 fullll
2023-09-29 17:57:15 hellomoto8712279935
[root@ip-10-0-1-129 ec2-user]# aws s3 mb s3://fullll
make_bucket failed: s3://fullll An error occurred (BucketAlreadyOwnedByYou) when
 calling the CreateBucket operation: Your previous request to create the named b
ucket succeeded and you already own it.
[root@ip-10-0-1-129 ec2-user]# aws s3 mb s3://fullll --region ap-south-1a
make_bucket failed: s3://fullll Could not connect to the endpoint URL: "https://
fullll.s3.ap-south-1a.amazonaws.com/"
[root@ip-10-0-1-129 ec2-user]# my_bucket mb s3://fullll
bash: my_bucket: command not found
[root@ip-10-0-1-129 ec2-user]# aws s3 ls
2023-09-29 17:41:26 fullll
2023-09-29 17:57:15 hellomoto8712279935
[root@ip-10-0-1-129 ec2-user]# yum update -y
Last metadata expiration check: 5:10:37 ago on Fri Sep 29 14:42:10 2023.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-1-129 ec2-user]# yum install httpd -y
Last metadata expiration check: 5:11:02 ago on Fri Sep 29 14:42:10 2023.
Package httpd-2.4.56-1.amzn2023.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-10-0-1-129 ec2-user]# aws s3 mb s3://fullll
make_bucket failed: s3://fullll An error occurred (BucketAlreadyOwnedByYou) when
dy own it.
[root@ip-10-0-1-129 ec2-user]# aws s3 mb s3://fullll --region ap-south-1a
make_bucket failed: s3://fullll Could not connect to the endpoint URL: "https://
[root@ip-10-0-1-129 ec2-user]#
```

© 2023, Amazon Web Services India Private Limited or its affiliates.
27°C Partly cloudy
01:25 AM
30-09-2023

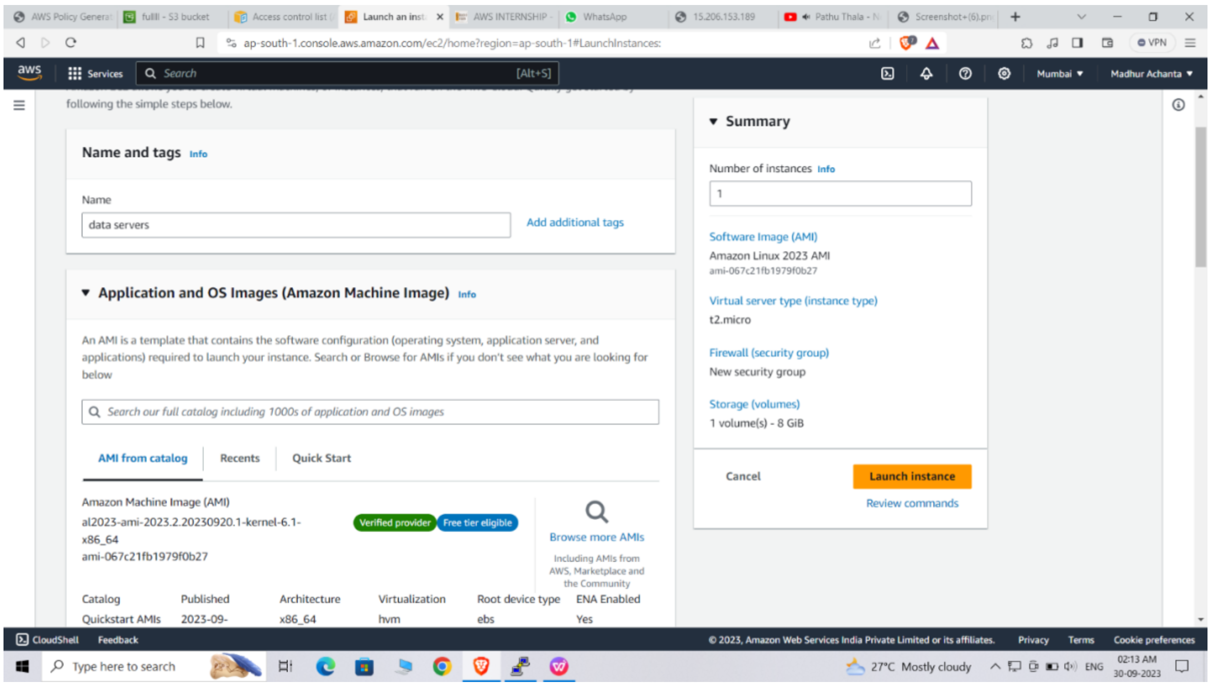

EC2 | eu-north-1
rtss - S3 bucket
iron.jpg (350×560)
rtss.s3.eu-north-1.amazonaws.com/iron.jpg

STEP-3:

Launching Database :-

Click on the launch instances

Give name as data servers


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

following the simple steps below.

Name and tags Info

Name: data servers

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

AMI from catalog | Recents | Quick Start

Amazon Machine Image (AMI)
al2023-ami-2023.2.20230920.1-kernel-6.1-x86_64
ami-067c21fb1979f0b27

Verified provider | Free tier eligible

Browse more AMIs
Including AMIs from AWS, Marketplace and the Community

Catalog	Published	Architecture	Virtualization	Root device type	ENA Enabled
Quickstart AMIs	2023-09-	x86_64	hvm	ebs	Yes

Summary

Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI
ami-067c21fb1979f0b27

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

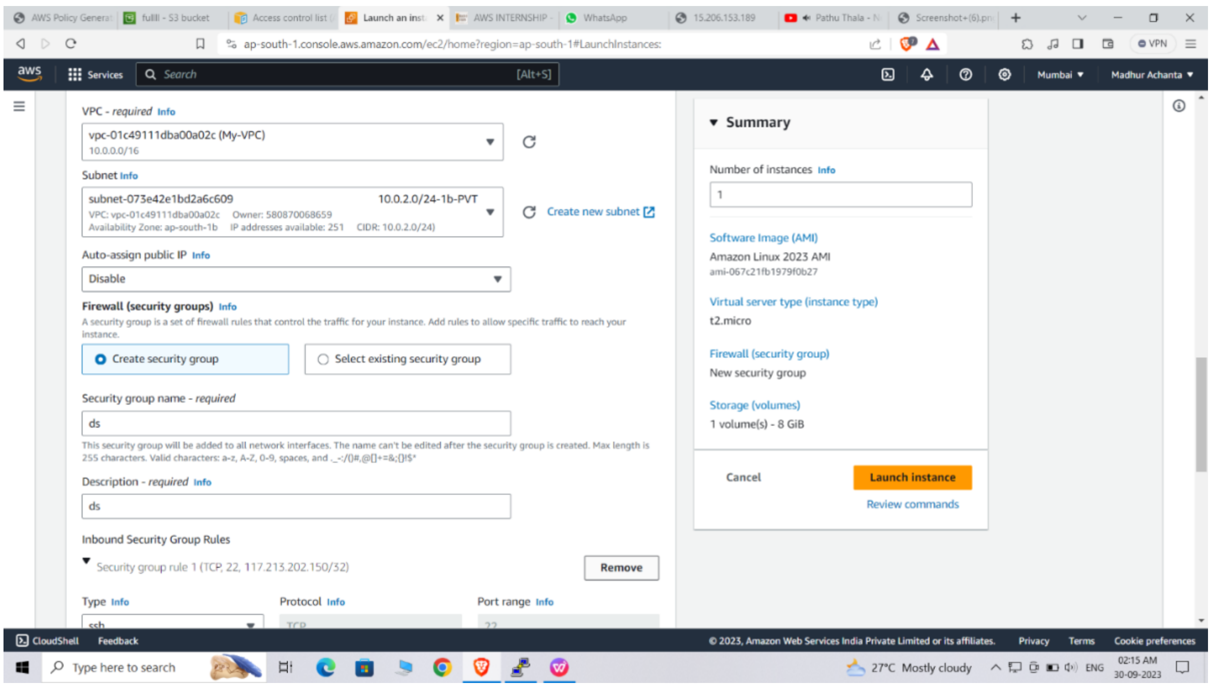
Cancel | Launch instance | Review commands

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

Select AMI(Amazon Machine Image) as linux

Select t2.micro which is free instances

Now select the existing key pair which is named as hack


VPC - *required* Info

vpc-01c49111dba00a02c (My-VPC)
10.0.0.0/16

Subnet Info

subnet-073e42e1bd2a6c609 10.0.2.0/24-1b-PVT
VPC: vpc-01c49111dba00a02c Owner: 580870068659
Availability Zone: ap-south-1b IP addresses available: 251 CIDR: 10.0.2.0/24)

Auto-assign public IP Info

Disable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

[✓ Create security group] Select existing security group

Security group name - *required*

ds

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info

ds

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 117.213.202.150/32)

Type: ssh | Protocol: TCP | Port range: 22

Summary

Number of instances: 1

Software Image (AMI)
Amazon Linux 2023 AMI
ami-067c21fb1979f0b27

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

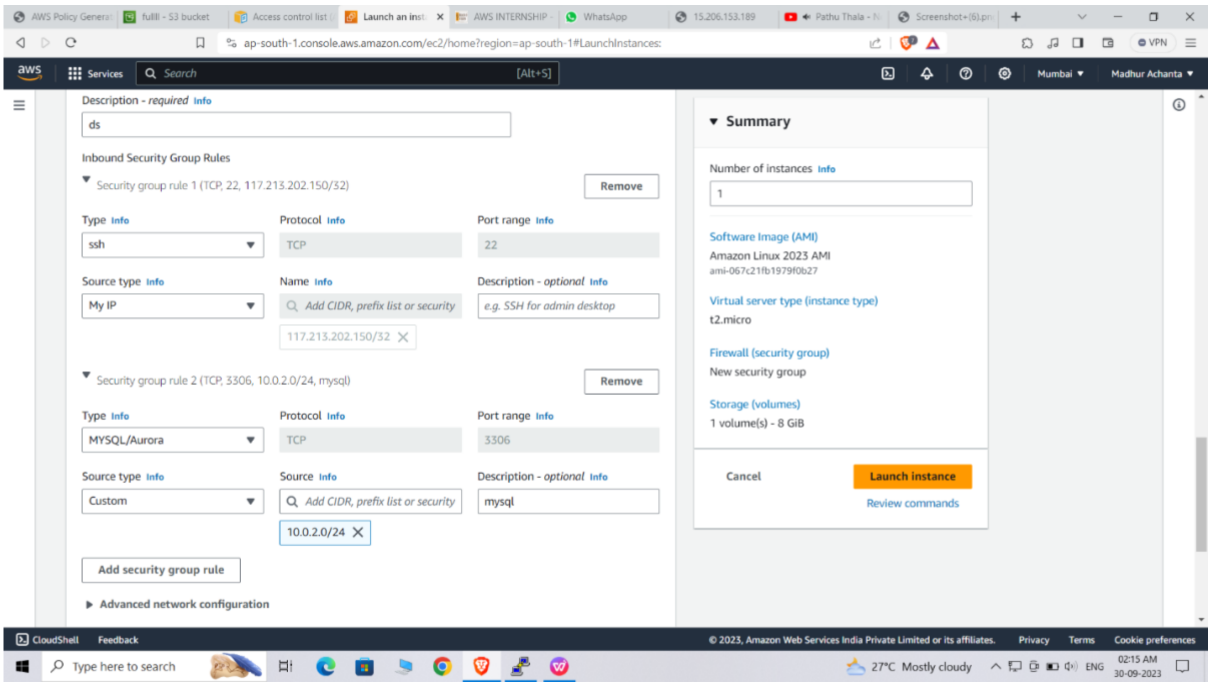
Storage (volumes)
1 volume(s) - 8 GiB

Now go to the network settings and select VPC & PRIVATE subnet

Now go to the the inbound security group in that select source type as my IP

Now select the type as MYSQL/Aurora and give the source as 10.0.2.0/24

And give the description as mysql give linux scripts and click on the launch instances



Description - *required* Info

ds

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 117.213.202.150/32) Remove

Type	Protocol	Port range
ssh	TCP	22

Source type	Name	Description - *optional*
My IP	Add CIDR, prefix list or security / 117.213.202.150/32	e.g. SSH for admin desktop

Security group rule 2 (TCP, 3306, 10.0.2.0/24, mysql) Remove

Type	Protocol	Port range
MYSQL/Aurora	TCP	3306

Source type	Source	Description - *optional*
Custom	Add CIDR, prefix list or security / 10.0.2.0/24	mysql

Add security group rule

Advanced network configuration

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI ami-067c21fb1979f0b27

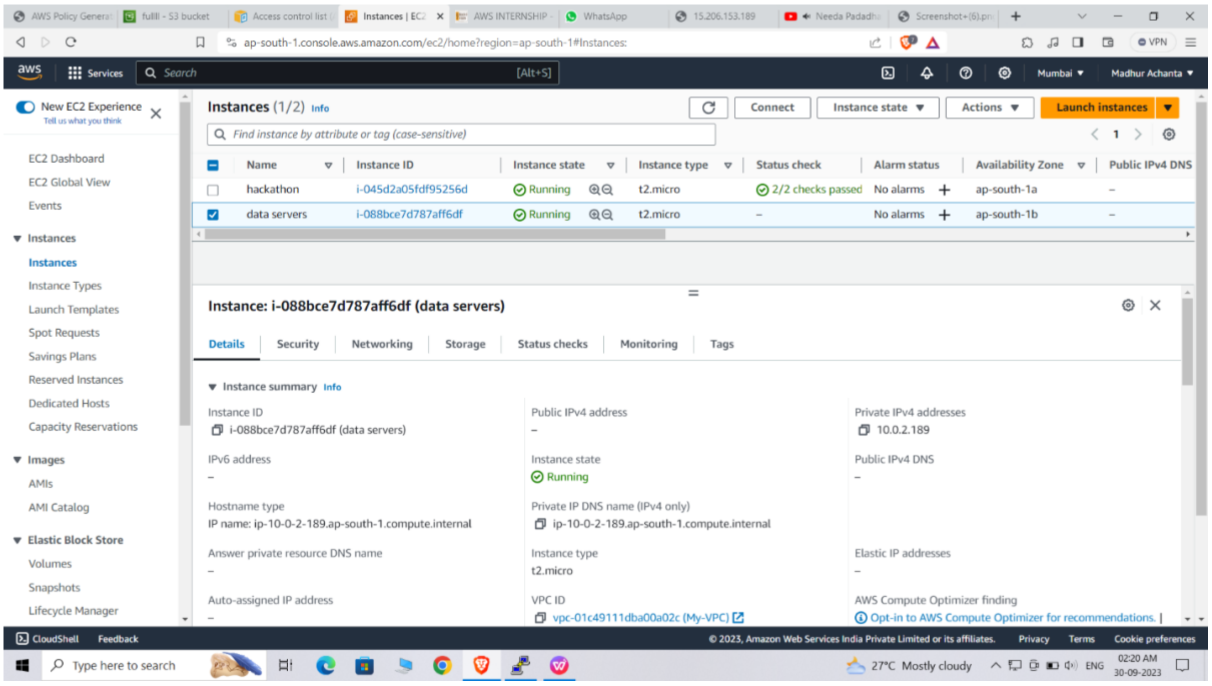
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Cancel | Launch instance | Review commands

Then the instance has been launched



Instances (1/2) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
hackathon	i-045d2a05fdf95256d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
data servers	i-088bce7d787aff6df	Running	t2.micro	–	No alarms	ap-south-1b	–

Instance: i-088bce7d787aff6df (data servers)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

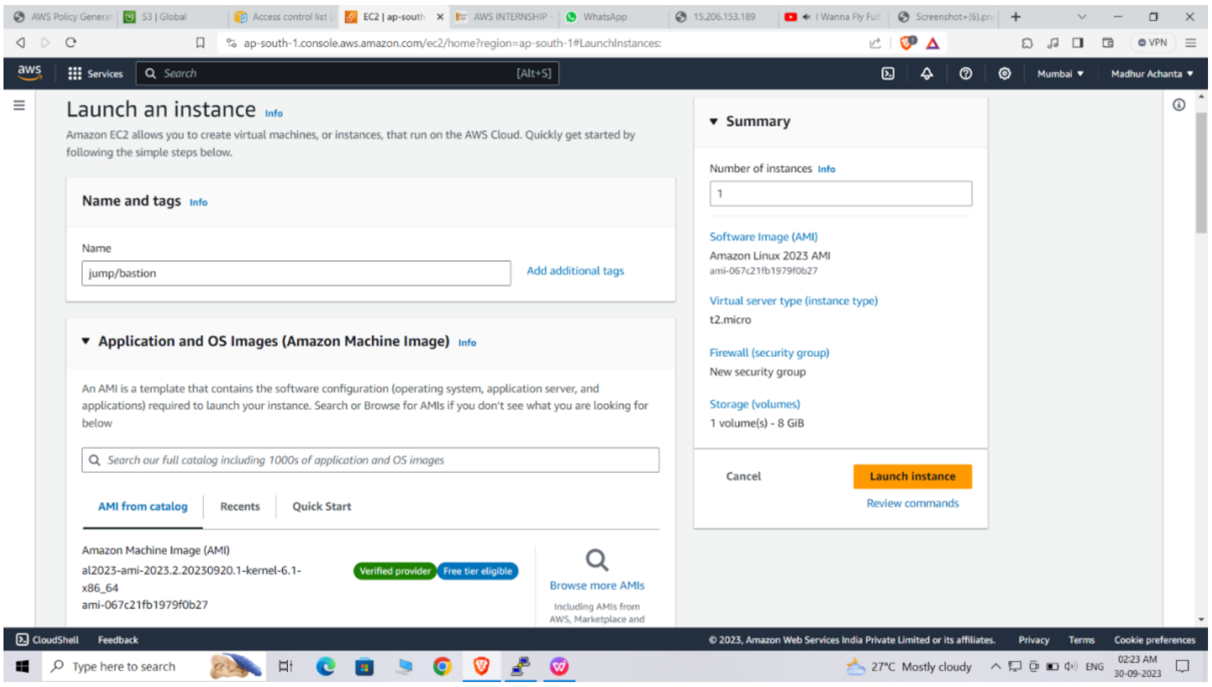
Instance ID: i-088bce7d787aff6df (data servers)	Public IPv4 address: –	Private IPv4 addresses: 10.0.2.189
IPv6 address: –	Instance state: Running	Public IPv4 DNS: –
Hostname type: IP name: ip-10-0-2-189.ap-south-1.compute.internal	Private IP DNS name (IPv4 only): ip-10-0-2-189.ap-south-1.compute.internal	
Answer private resource DNS name: –	Instance type: t2.micro	Elastic IP addresses: –
Auto-assigned IP address: –	VPC ID: vpc-01c49111dba00a02c (My-VPC)	AWS Compute Optimizer finding: Opt-in to AWS Compute Optimizer for recommendations.

<u>Creating instances</u>:-

For creating launch instances

Click on launch instances

Give name as jump/baston


Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: jump/bastion

Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

AMI from catalog | Recents | Quick Start

Amazon Machine Image (AMI)
al2023-ami-2023.2.20230920.1-kernel-6.1-x86_64
ami-067c21fb1979f0b27
Verified provider | Free tier eligible

Browse more AMIs
Including AMIs from AWS, Marketplace and

Summary

Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI
ami-067c21fb1979f0b27

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

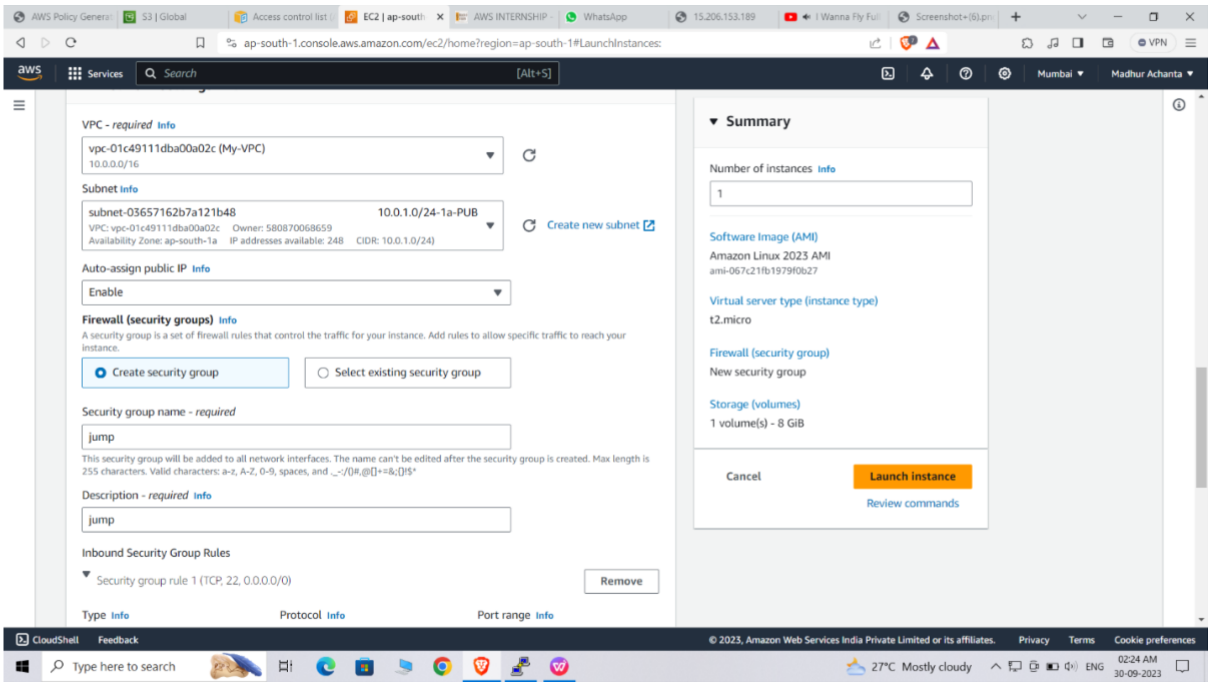
Storage (volumes)
1 volume(s) - 8 GiB

Cancel | Launch instance | Review commands

Now select the AMI(Amazon Machine Image) as linux

Now select instances type as t2.micro which is free tier

And give the existing key pair


VPC - *required* Info
vpc-01c49111dba00a02c (My-VPC)
10.0.0.0/16

Subnet Info
subnet-03657162b7a121b48 10.0.1.0/24-1a-PUB
VPC: vpc-01c49111dba00a02c Owner: 580870068659
Availability Zone: ap-south-1a IP addresses available: 248 CIDR: 10.0.1.0/24)

Create new subnet

Auto-assign public IP Info: Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

● Create security group ○ Select existing security group

Security group name - *required*: jump

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and ._-:/()#,@[]+=&;{}!$*

Description - *required* Info: jump

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0) Remove

Type Info | Protocol Info | Port range Info

Summary

Number of instances Info: 1

Software Image (AMI)
Amazon Linux 2023 AMI
ami-067c21fb1979f0b27

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

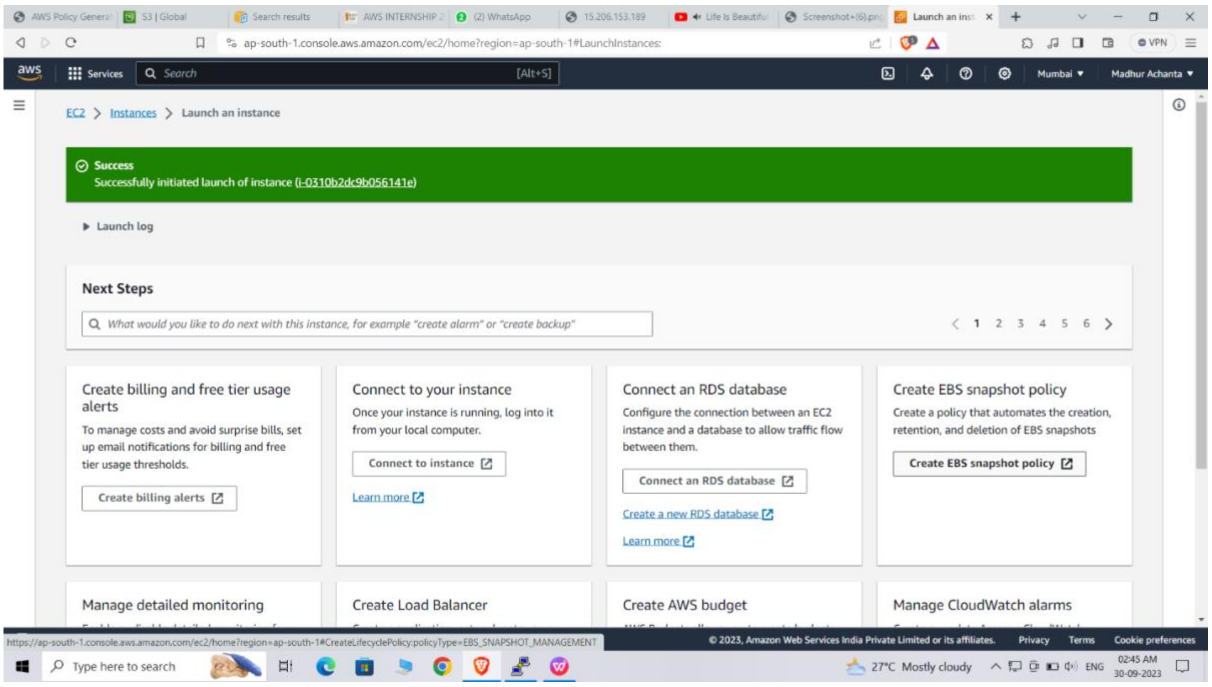
Cancel | Launch instance | Review commands

Now go to the network settings and give VPC

And give subnet as PUBLIC subnet

Give SG as jump and description as jump

Now go to advanced settings and type linux scripts


EC2 > Instances > Launch an instance

Success
Successfully initiated launch of instance (i-0310b2dc9b056141e)

Launch log

Next Steps

What would you like to do next with this instance, for example "create alarm" or "create backup"

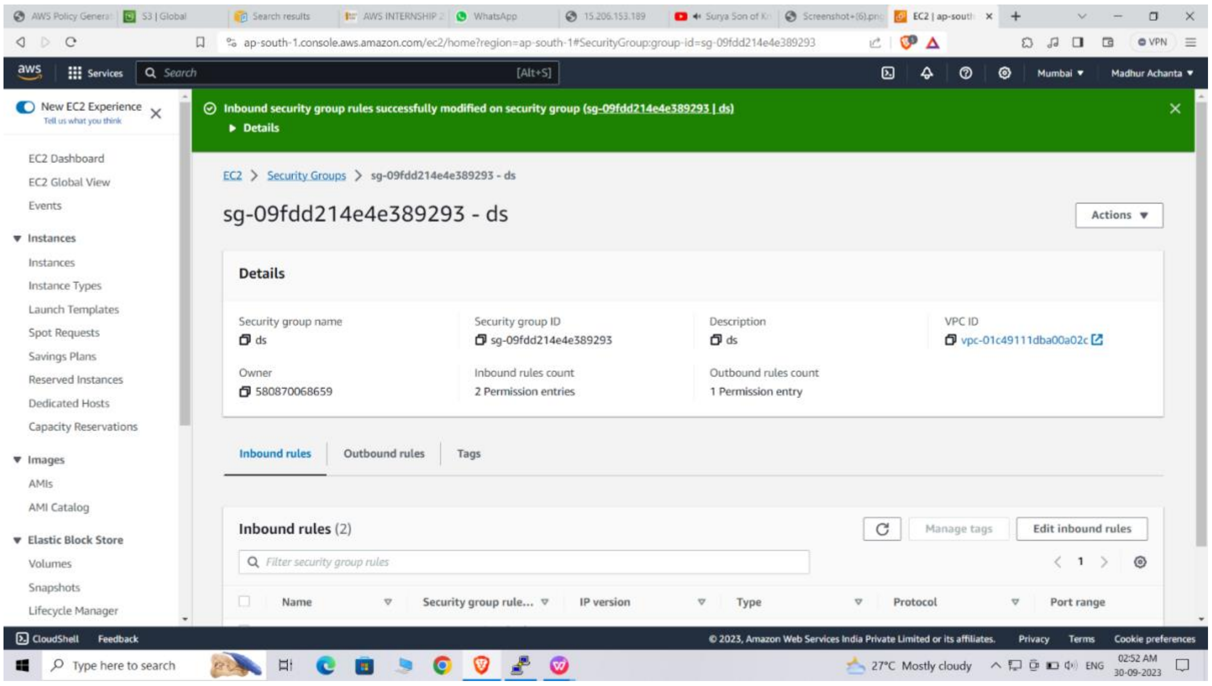
Create billing and free tier usage alerts
To manage costs and avoid surprise bills, set up email notifications for billing and free tier usage thresholds.
Create billing alerts

Connect to your instance
Once your instance is running, log into it from your local computer.
Connect to instance
Learn more

Connect an RDS database
Configure the connection between an EC2 instance and a database to allow traffic flow between them.
Connect an RDS database
Create a new RDS database
Learn more

Create EBS snapshot policy
Create a policy that automates the creation, retention, and deletion of EBS snapshots
Create EBS snapshot policy

Manage detailed monitoring
Create Load Balancer
Create AWS budget
Manage CloudWatch alarms


Inbound security group rules successfully modified on security group (sg-09fdd214e4e389293 | ds)
Details

EC2 > Security Groups > sg-09fdd214e4e389293 - ds

sg-09fdd214e4e389293 - ds

Details

Security group name	Security group ID	Description	VPC ID
ds	sg-09fdd214e4e389293	ds	vpc-01c49111dba00a02c
Owner	**Inbound rules count**	**Outbound rules count**	
580870068659	2 Permission entries	1 Permission entry	

Inbound rules | Outbound rules | Tags

Inbound rules (2)

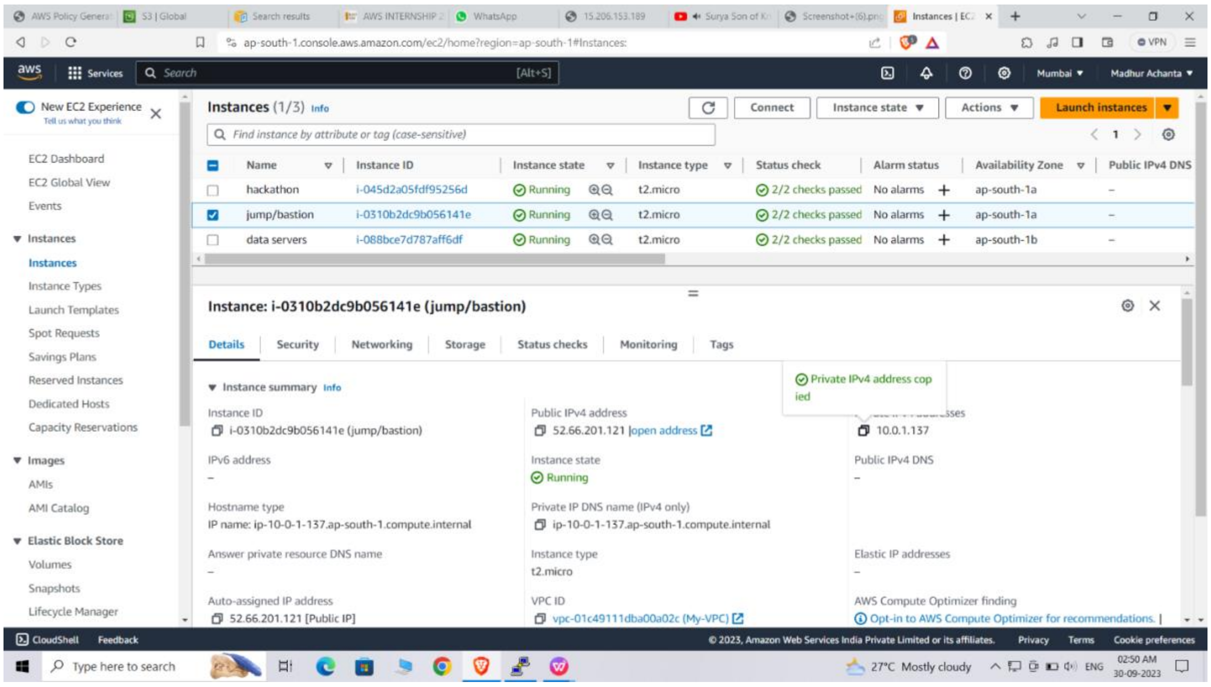
Name	Security group rule...	IP version	Type	Protocol	Port range

Connect the jump private IP address to database SSH

Now go to security group and edit the inbound rules

Remove IP given to SSH

Enter the copied address then save rules



Instances (1/3) Info

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	hackathon	i-045d2a05fdf95256d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
☑	jump/bastion	i-0310b2dc9b056141e	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
☐	data servers	i-088bce7d787aff6df	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	–

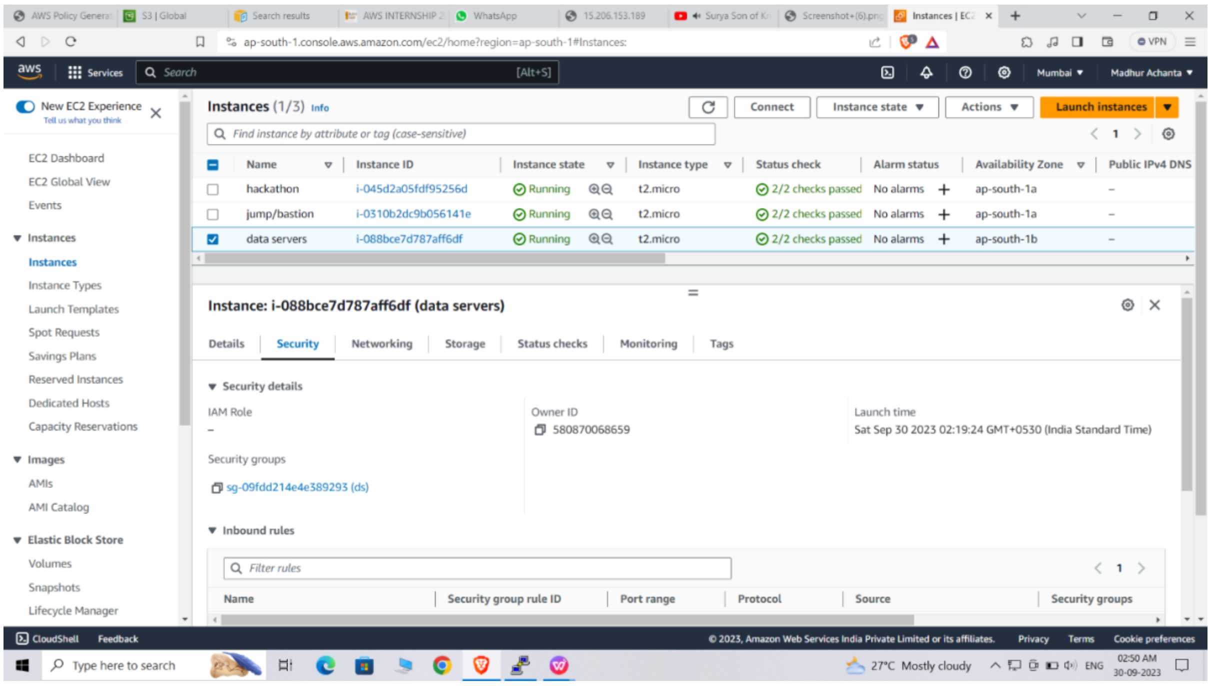
Instance: i-0310b2dc9b056141e (jump/bastion)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

Instance summary Info

Private IPv4 address copied

Instance ID i-0310b2dc9b056141e (jump/bastion)	Public IPv4 address 52.66.201.121 | open address	Private IPv4 addresses 10.0.1.137
IPv6 address –	Instance state Running	Public IPv4 DNS –
Hostname type IP name: ip-10-0-1-137.ap-south-1.compute.internal	Private IP DNS name (IPv4 only) ip-10-0-1-137.ap-south-1.compute.internal	
Answer private resource DNS name –	Instance type t2.micro	Elastic IP addresses –
Auto-assigned IP address 52.66.201.121 [Public IP]	VPC ID vpc-01c49111dba00a02c (My-VPC)	AWS Compute Optimizer finding Opt-in to AWS Compute Optimizer for recommendations.



Instances (1/3) Info

Find instance by attribute or tag (case-sensitive)

	Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4 DNS
☐	hackathon	i-045d2a05fdf95256d	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
☐	jump/bastion	i-0310b2dc9b056141e	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1a	–
☑	data servers	i-088bce7d787aff6df	Running	t2.micro	2/2 checks passed	No alarms	ap-south-1b	–

Instance: i-088bce7d787aff6df (data servers)

Details | Security | Networking | Storage | Status checks | Monitoring | Tags

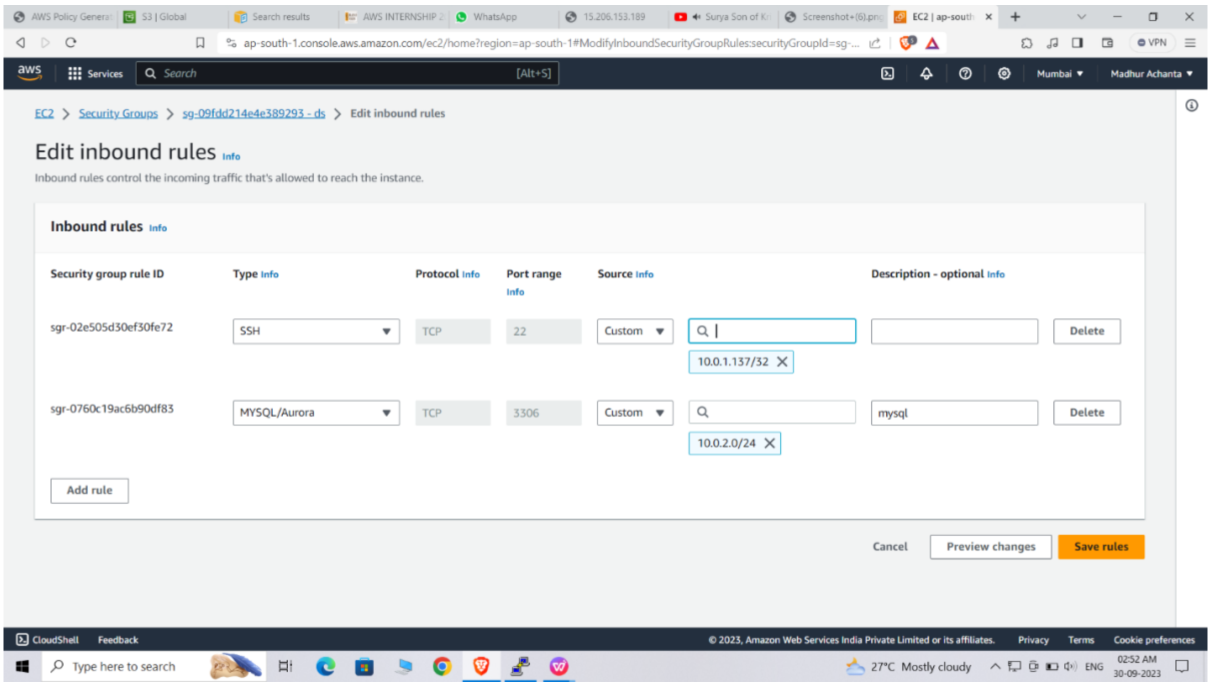
Security details

IAM Role –	Owner ID 580870068659	Launch time Sat Sep 30 2023 02:19:24 GMT+0530 (India Standard Time)
Security groups sg-09fdd214e4e389293 (ds)		

Inbound rules

Filter rules

Name	Security group rule ID	Port range	Protocol	Source	Security groups


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

EC2 > Security Groups > sg-09fdd214e4e389293 - ds > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-02e505d30ef30fe72	SSH	TCP	22	Custom	10.0.1.137/32	
sgr-0760c19ac6b90df83	MYSQL/Aurora	TCP	3306	Custom	10.0.2.0/24	mysql

STEP-4:

In this stage we created nacl for subnets.

In AWS, Network Access Control Lists (NACLs) are an optional layer of security that act as stateless firewalls for controlling inbound and outbound traffic at the subnet level. NACLs operate at the subnet level and evaluate rules in a sequential order to determine whether to allow or deny traffic. Each subnet in AWS can be associated with one NACL, and by default, a subnet is associated with the default NACL.

Here are the steps to create and attach a Network Access Control List (NACL) to subnets in AWS:

1. Open the Amazon VPC Console:

- Go to the Amazon VPC console in the AWS Management Console.

2. Create a Network Access Control List (NACL):

   - In the VPC dashboard, click on "Network ACLs" in the sidebar.

   - Click on "Create Network ACL" and specify a name and the VPC for the NACL.

   - Once created, the NACL will have default rules that allow all inbound and outbound traffic.

3. Configure Inbound and Outbound Rules:

   - Select the newly created NACL and click on the "Inbound Rules" or "Outbound Rules" tab.

   - Click on "Edit" and add or modify the rules according to your requirements.

   - Rules can be based on IP addresses, port ranges, protocols, and whether to allow or deny traffic.

   - Rules are evaluated in the order they appear, so ensure they are ordered correctly.

4. Associate the NACL with Subnets:

   - Click on the "Subnet Associations" tab in the NACL configuration.

   - Click on "Edit" and select the subnets you want to associate with the NACL.

   - You can choose to associate multiple subnets with the same NACL.

   - Make sure to confirm your changes to associate the NACL with the selected subnets.

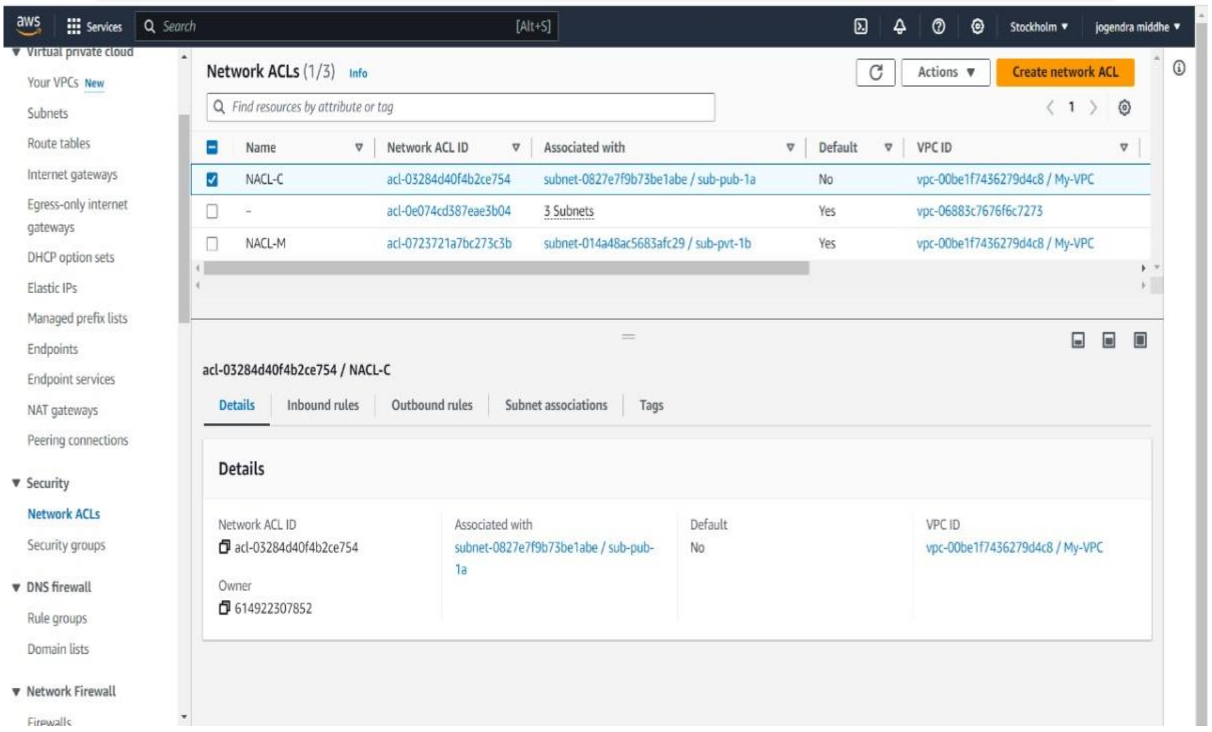
5. Verify and Test:

- Once the NACL is associated with the subnets, it will start governing the traffic.

- Verify that the NACL rules are correctly configured and are allowing or denying traffic as intended.

- Test the connectivity to ensure that the traffic is behaving as expected.

By following these steps, you can create and attach a Network Access Control List (NACL) to subnets in AWS. NACLs provide an additional layer of security by controlling inbound and outbound traffic at the subnet level, allowing you to define fine-grained rules for network traffic in your VPC.



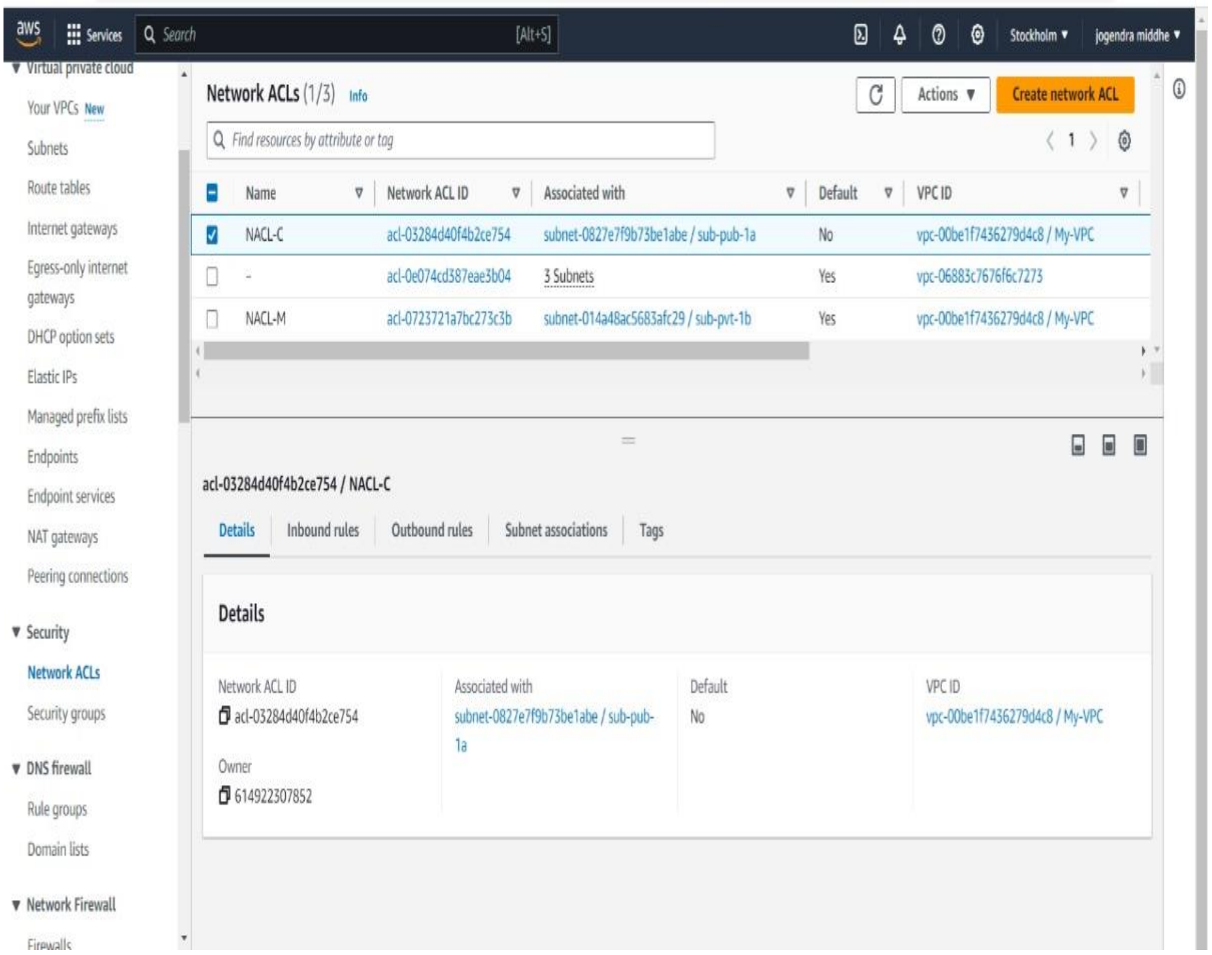
Network ACLs (1/3)

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-03284d40f4b2ce754 / NACL-C

Details

Network ACL ID: acl-03284d40f4b2ce754
Owner: 614922307852
Associated with: subnet-0827e7f9b73be1abe / sub-pub-1a
Default: No
VPC ID: vpc-00be1f7436279d4c8 / My-VPC



Network ACLs (1/3) Info

	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-03284d40f4b2ce754 / NACL-C

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

Details

Network ACL ID	Associated with	Default	VPC ID
acl-03284d40f4b2ce754	subnet-0827e7f9b73be1abe / sub-pub-1a	No	vpc-00be1f7436279d4c8 / My-VPC

Owner
614922307852

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.