LATE BHAUSAHEB HIRAY S.S. TRUST'S INSTITUTE OF COMPUTER APPLICATION

ISO 9001-2015 CERTIFIED

A PROJECT REPORT ON

VIRTUAL PRIVATE CLOUD
INFRASTRUCTURE PROVISIONING

FOR

BRAINOVISION SOLUTIONS INDIA PVT. LTD.

SUBMITTED BY

VARUN KUMAR

UNDER THE GUIDANCE OF

Dr. RASHMITA PRADHAN

MASTER OF COMPUTER APPLICATION
SEMESTER IV

UNIVERSITY OF MUMBAI
2023-2024

BRAIN O VISION

शिक्षा मंत्रालय
MINISTRY OF
EDUCATION

CERTIFICATE

OF INTERNSHIP COMPLETION

TO

Date: 01-06-2024

Varun Kumar
Late Bhausaheb Hiray Smarnika Samiti Trust's Institute of Computer Application

This is to certify that **Varun Kumar** has successfully completed his program with **BrainOvision Solutions Pvt. Ltd.** He was working on **AMAZON WEB SERVICES** and was actively & diligently involved in the projects and tasks assigned. During the span, we found him punctual and hardworking person.
His feedback and evolution proved that he is a quick learner. Congratulations and Best Wishes.

ROLE	: **AWS INTERN**
INTERN ID	: **B24SD0140**
START DATE	: **17-02-2024**
END DATE	: **30-05-2024**

Yours Faithfully

BRAIN O VISION SOLUTIONS INDIA PVT. LTD. HYDERABAD

Ganesh Nag Doddi
Founder & CEO
Brainovision Solutions India Pvt Ltd

Dr. Buddha Chandrashekar
Chief Coordinating Officer – AICTE
All India Council for Technical Education

BRAIN O VISION

Date: 26-05-2024

CERTIFICATE OF COMPLETION

This is to certify that **Varun Kumar** from **Late Bhausaheb Hiray S.S. Trust's Institute of Computer Application,Mumbai** successfully completed **Major** project Work at **BrainOvision Solutions India Pvt. Ltd.** from **20-03-2024 to 24-05-2024** during the project he worked on the project entitled **"Virtual Private Cloud Infrastructure Provisioning".**

we found him punctual and hardworking and interested to learn the technologies During the project work.His demonstrated Good skills with self -motivate attitude towards learning .

He is association with the team was fruitful .We wish all the best for future!

Yours faithfully
Name: Narayana Swamy N
Designation: Manager
BRAIN O VISION SOLUTIONS INDIA PVT LTD

+91 95029 35039
Info.brainovision@gmail.com
www.brainovision.in

Brain O Vision Solutions Pvt. Ltd...
Mohan's Elite, 1st Floor, H.No:2-56/5/50, Madhapur, Khanamet, Hyd - 500 081.
www.fb.com/brainovisionsolutions

PERFORMANCE APPRAISAL FROM PROJECT EMPLOYEE

INSTRUCTIONS: The immediate supervisor is asked to evaluate the student objectively comparing him with other students of comparable academic records with other personnel assigned to the same on similarly classified or with corporate standards.

Evaluation Criteria	Exceptional	Very Good	Average	Marginal	Unacceptable	Non-Applicable	Comments (if needed, write on back side of the page or on fresh new sheet then attach the same)
Relations with others		√					
Judgments		√					
Ability to learn		√					
Communication Skills			√				
Technical Skills		√					
Teamwork Skills		√					
Dependability		√					
Quality of work	√						
Educational Preparation for the assignment		√					
Potential for Greater Responsibility		√					
Comparison with students is at the same level from other Institutions		√					
Overall Performance		√					
Attendance: Regular							**Punctuality:** Regular

Name of the Student : **Varun Kumar**
Course Title : Master of Computer Applications
Name of the College : Late Bhausaheb Hiray S.S Trust's Institute of Computer Application, Bandra (East), Mumbai-400 051.
Name of Project : **Virtual Private Cloud Infrastructure Provisioning**
Name of Project Guide : **Mr.Narayana Swamy N**
Company Name : **BRAINOVISION SOLUTIONS INDIA PVT. LTD.**

LATE BHAUSAHEB HIRAY S.S. TRUST'S INSTITUTE OF COMPUTER APPLICATION

ISO 9001-2015 CERTIFIED

S.N. 341, Next to New English School, Govt. Colony, Bandra (East),Mumbai – 400051, Tel: 91-22-26570986/892, Telefax: 91-22-2657 3181
Website: www.hiray.org.in, E-mail: director@hiray.org.in

Date :

CERTIFICATE OF APPROVAL

This is to certify that the project report titled

VIRTUAL PRIVATE CLOUD INFRASTRUCTURE PROVISIONING

Is a bona-fide record of the work done by

VARUN KUMAR

UNDER THE GUIDANCE OF

Dr. RASHMITA PRADHAN

----------	----------	----------
HOD	External Guide	Internal Guide

ACKNOWLEDGEMENT

I started this project as part of the curriculum of Late Bhausaheb Hiray Smarnika Samiti Trust's Institute of Computer Application – MCA (Sem IV). The 6-month duration of this internship was an amazing experience in world of professionals. During this work I have gained both practical as well as theoretical knowledge of great significance.

Now after the completion of the project I feel very pleased to present this project report. Also, would like to take this opportunity to thank all those who helped me to successfully complete theproject.

I am extremely grateful to my Internal Project Guide **Dr. Rashmita Pradhan** for her constant and valuable encouragement.

At **BRAINOVISION SOLUTIONS INDIA PVT. LTD.**, I am thankful to everyone, for their kind co-operation and motivation in my project. I would also like to extend my thanks to my External Project Guide & his team for their co- operation and motivation. Without his guidance and help this project would not have been possible.

Finally, my heartfelt appreciation to my teammates for their valuable advice and timely support.

Table of Contents

Contents

1. Introduction

1.1 Synopsis of the Project

1.1.1 Company Profile

Brainovision Solutions India Private Limited.

is a an organization which deals with the wing of software development and technical education.

This is the place for students and faculty and other companies to find solution for all your requirements Such as internships, academic projects (Mini & Major project) , online courses, Workshops , faculty development programs and to hire a perfect skilled candidates.

All the certificate will be issued from our corporate company Brainovision Solutions India Private Limited.

If you are the one who dreams to be a technical Pro and wants to get placed in MNCs then this is the place to have to stop and start learning practically in corporate environment.

1.1.2 Objective

- The objective of this project is to establish a secure, scalable, and efficient infrastructure using Amazon Web Services Virtual Private Cloud (VPC) services. By implementing a VPC, the project aims to enhance the deployment and management of web applications, ensuring high availability through Auto Scaling and load balancing. Additionally, robust security measures, including Network Access Control Lists and IAM roles, will protect resources and data. This setup allows for flexible resource management, reducing costs and operational overhead while providing a reliable and secure environment for application hosting and data storage.

1.1.3 Scope

- Using Amazon Web Services Virtual Private Cloud services provides numerous advantages over traditional on-premises infrastructure. From hosting applications to ensuring secure data management, VPCs enhance operational efficiency. They offer scalable, flexible solutions for managing resources, ensuring high availability through features like Auto Scaling and load balancing. Additionally, VPCs provide robust security measures, including Network Access Control Lists and IAM roles, simplifying administration and monitoring. Overall, VPCs streamline the deployment and management of cloud-based applications, providing significant benefits to businesses in terms of cost, scalability, and security.

1.1.4 Problem Definition

- The problem definition for this project involves setting up a secure and scalable cloud infrastructure using Amazon Web Services Virtual Private Cloud to host web applications and manage data. Key challenges include configuring public and private subnets for optimal security and performance, ensuring high availability with Auto Scaling and load balancing, and implementing robust security measures such as Network Access Control Lists (NACLs) and IAM roles. Additionally, the project requires seamless integration with S3 for data storage and management. Tools like AWS Management Console, EC2, and RDS will be used to configure and manage the infrastructure, ensuring reliable and efficient operation.

1.1.5 Existing System

- **The existing system includes:**

1. VPC Creation
2. IGW Creation
3. Subnet Creation and Configuration
4. EC2 instance Creation
5. Security Configuration
6. Route Tables Setup
7. NAT Gateway Deployment
8. Jump/Bastion Host Setup
9. Database Configuration
10. Monitoring and Logging

1.1.6 Proposed System

- **The proposed system includes similar process to existing system with some different functionalities:**

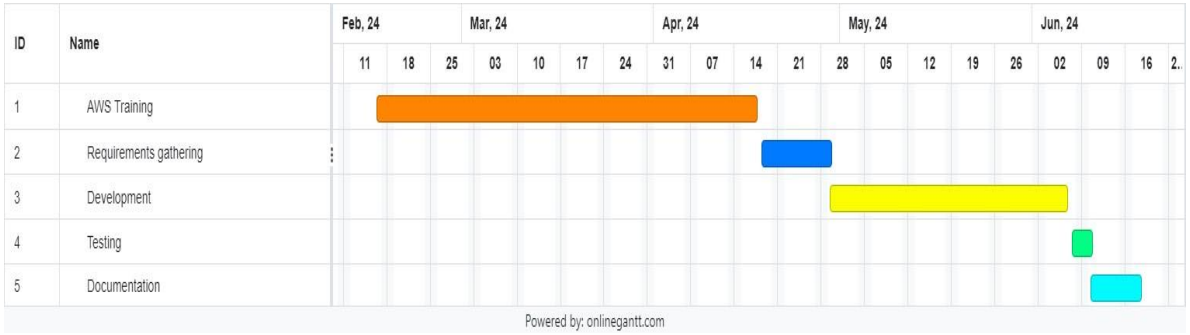
1. VPC Creation
2. IGW Creation
3. Subnet Creation and Configuration
4. EC2 instance Creation
5. Security Configuration
6. Route Tables Setup
7. NAT Gateway Deployment
8. Auto Scaling and Load Balancing
9. Jump/Bastion Host Setup
10. Database Configuration
11. S3 Bucket Integration
12. Monitoring and Logging
13. IAM Role Configuration
14. Application Deployment

2. System Analysis

2.1 System Planning and Schedule

2.1.1 GANTT Chart

- A Gantt chart is a useful graphical tool which shows activities or tasks performed against time. It is also known as visual presentation of a project where the activities are broken down and displayed on a chart which makes it is easy to understand and interpret.
- A Gantt chart is a popular tool in project management. It basically drills down activities which need to be done by a fixed time period. It is commonly used for tracking project schedules.
- Understanding the interlinkage between activities is very important to monitor and Gantt charts help the project manager to do just that. It conveys the information about the completion of other activities in the project. This information is important because of the interlinkages between various activities and if one activity gets delayed it will have an impact on others.
- Gantt chart is a useful tool in planning and scheduling the projects. It keeps the management updated as to when the project will get completed. It also keeps the management informed.
- They are commonly used in scheduling production processes, employee roster or scheduling, events scheduling, production processes, etc. Microsoft Excel can also be used to create Gantt charts apart from other independent software available in the market.



2.1.2 PERT TABLE

Training	
Start Date	15-Feb
End Date	14-Apr

Requirements gathering	
Start Date	15-Apr
End Date	26-May

Development	
Start Date	26-May
End Date	05-Jun

Testing	
Start Date	05-Jun
End Date	08-Jun

Documentation	
Start Date	08-Jun
End Date	12-Jun

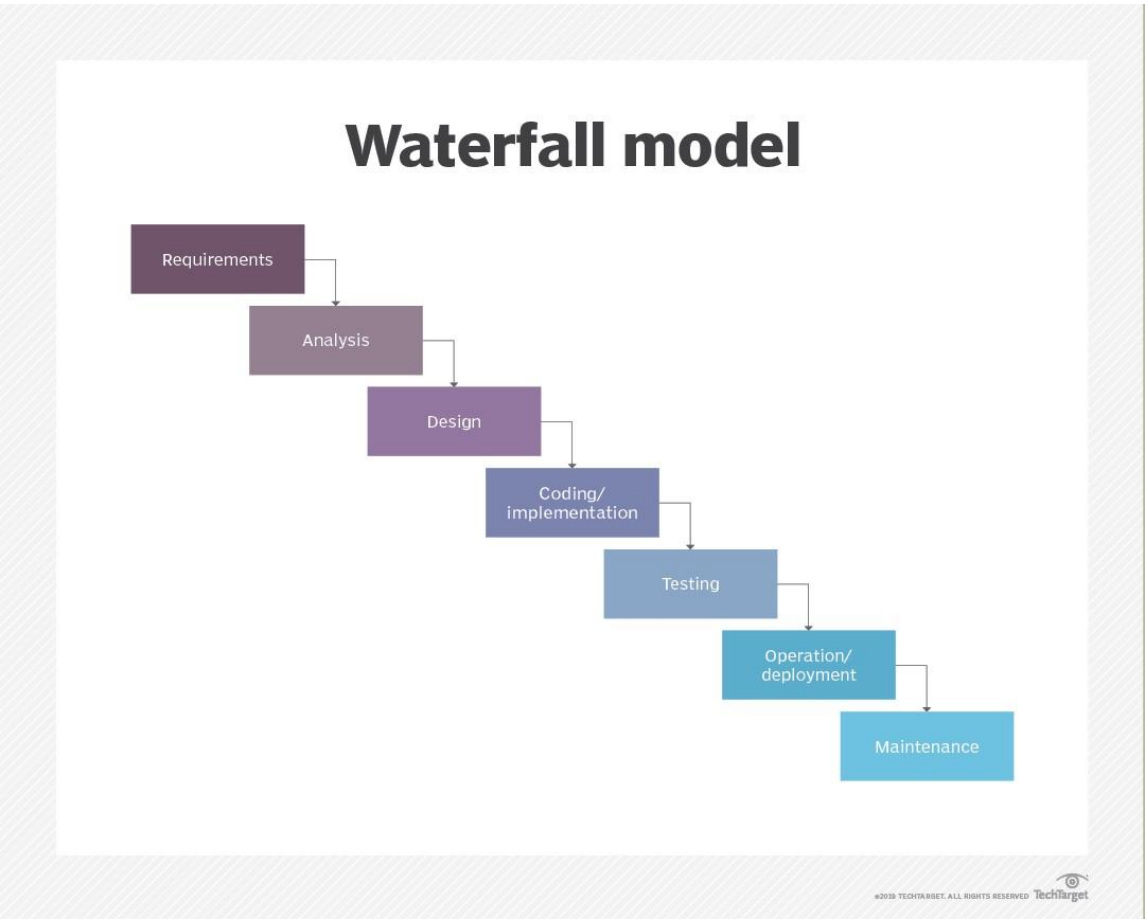
2.1.3 Waterfall Model

- The waterfall model doesn't include a project's end user or client as much as other development methodologies. Users are consulted during the initial stages of gathering and defining requirements, incorporating client feedback after that. By leaving the client out of the main part of the waterfall process, the development team moves quickly through the phases of a project.
- This methodology is good for teams and projects that want to develop a project according to fixed or unchanging requirements set forth at the beginning of the project. Waterfall projects have a high degree of process definition with little or no output variability. Waterfall is also a good choice if the project is constrained by cost or time.
- Projects based on the waterfall model are well defined, predictable and have specific documentation. They also have the following characteristics:
  a) fixed requirements;
  b) ample resources;
  c) an established timeline;
  d) well-understood technology; and
  e) unlikely to require significant changes.
- In software development, if an application needs to work on the first try at the risk of losing customers, waterfall is a suitable method because it sets out to achieve that goal. Contrast that with Agile project management and development methodology. Agile methods use ongoing reiteration, which is an iterative approach that involves designing, developing and testing software in repeated cycles that build upon each other.

- **Phases of the waterfall model**
- When used for a software development process, the waterfall methodology has seven stages:
  a) **Requirements**. Potential requirements, deadlines and guidelines for the project are analyzed and placed into a formal requirements document, also called a functional specification. This stage of development defines and plans the project without mentioning specific processes.
  b) **Analysis**. The system specifications are analyzed to generate product models and business logic to guide production. This is also when financial and technical resources are audited for feasibility.
  c) **Design**. A design specification document is created to outline technical design requirements, such as the programming language, hardware, data sources, architecture and services.
  d) **Coding and implementation**. The source code is developed using the models, logic and requirement specifications designated in the prior phases. Typically, the system is coded in smaller components, or units, before being put together.
  e) **Testing**. This is when quality assurance, unit, system and beta tests identify issues that must be resolved. This may cause a forced repeat of the coding stage for debugging. If the system passes integration and testing, the waterfall continues forward.

f) **Operation and deployment.** The product or application is deemed fully functional and is deployed to a live environment.

g) **Maintenance**. Corrective, adaptive and perfective maintenance is carried out indefinitely to improve, update and enhance the product and its functionality. This could include releasing patch updates and new versions.



3. System Design

3.1. Technology We Use

3.1.1 Software Tools/Resources to be used

1. **AWS Management Console:**
   a. **Purpose:** A web-based interface for managing AWS services.
   b. **Features:** Provides access to various AWS services, including EC2, RDS, VPC, IAM, and S3, for configuring and managing cloud resources.
2. **Amazon EC2:**
   a. **Purpose:** Provides scalable computing capacity in the AWS cloud.
   b. **Features:** Allows deployment of instances (virtual servers), configuring auto-scaling, and load balancing.
3. **Amazon RDS:**
   a. **Purpose:** Managed relational database service.
   b. **Features:** Simplifies database setup, operation, and scaling. Supports various database engines, including MySQL, PostgreSQL, and SQL Server.
4. **Amazon S3:**
   a. **Purpose:** Scalable object storage service.
   b. **Features:** Securely stores and retrieves any amount of data. Provides IAM roles for fine-grained access control.
5. **AWS IAM:**
   a. **Purpose:** Identity and Access Management.
   b. **Features:** Controls access to AWS services and resources securely. Allows creation of users, groups, and roles with specific permissions.
6. **AWS VPC:**
   a. **Purpose:** Virtual Private Cloud service.
   b. **Features:** Provides isolated cloud resources within the AWS cloud. Supports custom IP address ranges, subnets, route tables, and gateways.
7. **SSH (Secure Shell):**
   a. **Purpose:** Secure protocol for operating network services over an unsecured network.
   b. Features: Enables secure access to the Jump/Bastion host and other instances within the VPC.

3.1.2 Selection of Technology/Specific Requirements

3.1.2.1 Hardware to Be Used

- **Processor:** Intel Pentium Processor or Higher (1.6 GHz or faster processor)
- **Memory:** Minimum 8GB RAM or higher
- **Storage:** Minimum 500GB Hard disk

3.1.2.2 Software to Be Used

- **Operating System:**

  - Windows 7 or newer
  - Linux (any modern distribution)
  - macOS 10.8 or later

3.2 Detailed Life Cycle of the Project

1. **VPC Creation:**

a) Define and create a VPC with a CIDR block of 10.0.0.0/16.
b) Set up subnets (public and private).
c) Configure route tables and gateways (Internet Gateway for public subnet, NAT Gateway for private subnet).

2. **Security Configuration:**

a) Configure Network Access Control Lists (NACLs) for both subnets.
b) Set up Security Groups for instance-level security.
c) Define IAM roles and policies for secure access management.

3. **Resource Deployment:**

a) Launch EC2 instances for web servers in the public subnet.
b) Deploy an RDS instance for the database in the private subnet.
c) Set up a Jump/Bastion host in the public subnet for secure access.

4. **Auto Scaling and Load Balancing:**

a) Create an Auto Scaling Group (ASG) for EC2 instances.
b) Configure an Elastic Load Balancer (ELB) to distribute traffic.

5. **Data Storage and Management:**

a) Create an S3 bucket for storing application data.
b) Implement IAM roles to control access to the S3 bucket.
c) Set up lifecycle policies for data management.

6. **Monitoring and Logging:**

a) Configure CloudWatch for monitoring EC2 instances, RDS, and other resources.
b) Set up CloudTrail for logging API calls and actions.
c) Implement alarms and notifications for critical events.

7. **Application Deployment:**

a) Deploy web applications on EC2 instances.
b) Configure environment variables and necessary dependencies.
c) Set up CI/CD pipelines for continuous deployment.

8. **Backup and Recovery:**

a) Implement RDS automated backups and snapshots.
b) Set up S3 backup policies for critical data.
c) Configure recovery procedures and test regularly.

9. **Cost Management:**

a) Monitor usage and spending through AWS Cost Explorer.
b) Implement Reserved Instances and Savings Plans where applicable.
c) Review and optimize resource allocation regularly.

10. **User Access and Permissions:**

a) Configure IAM users, groups, and roles.
b) Define and apply permissions policies.
c) Set up Multi-Factor Authentication (MFA) for enhanced security.

11. **Documentation and Training:**

a) Document the VPC setup, configurations, and procedures.
b) Train relevant personnel on managing and maintaining the VPC.
c) Update documentation regularly to reflect changes and

4.AWS

4.1 About AWS

AWS is the world's most comprehensive and broadly adopted cloud platform, offering over 200 fully featured services from data centers globally. Some key points about AWS:

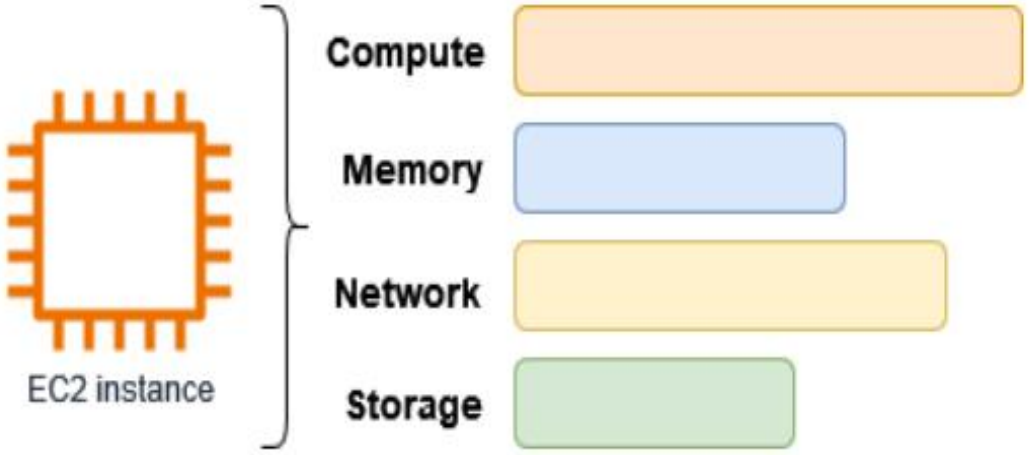
1. AWS provides a wide range of cloud computing services, including compute, storage, databases, networking, security, analytics, and more. These services help organizations move faster, lower IT costs, and scale.
2. AWS is trusted by millions of customers, including the fastest-growing startups, largest enterprises, and leading government agencies, to power a wide variety of workloads.
3. With AWS, organizations no longer need to plan for and procure servers and other IT infrastructure weeks or months in advance. Instead, they can instantly spin up hundreds or thousands of servers in minutes and deliver results faster.
4. AWS's rapid pace of innovation allows customers to focus on what's most important to them and their end users, without having to worry about the underlying infrastructure.

4.2 EC2

Amazon EC2 (Elastic Compute Cloud) is a core service within the AWS cloud platform that provides scalable computing capacity. Here are some key points about EC2:

1. EC2 allows you to launch and manage virtual server instances, called EC2 instances, in the AWS cloud. These instances can be easily scaled up or down based on your computing needs.
2. EC2 instances come in a variety of configurations, called instance types, that offer different combinations of CPU, memory, storage, and networking capacity. You can choose the instance type that best fits your application requirements.
3. EC2 instances can be launched using Amazon Machine Images (AMIs), which are pre-configured templates that include the operating system, application server, and other software.
4. To connect to your EC2 instances, you can use Secure Shell (SSH) for Linux/Unix instances or Remote Desktop Protocol (RDP) for Windows instances. You'll need to use the key pair (private and public keys) created during the instance launch process.

5. EC2 instances are deployed within a Virtual Private Cloud (VPC), which is a logically isolated section of the AWS cloud. This allows you to control the network settings, security, and access to your instances.



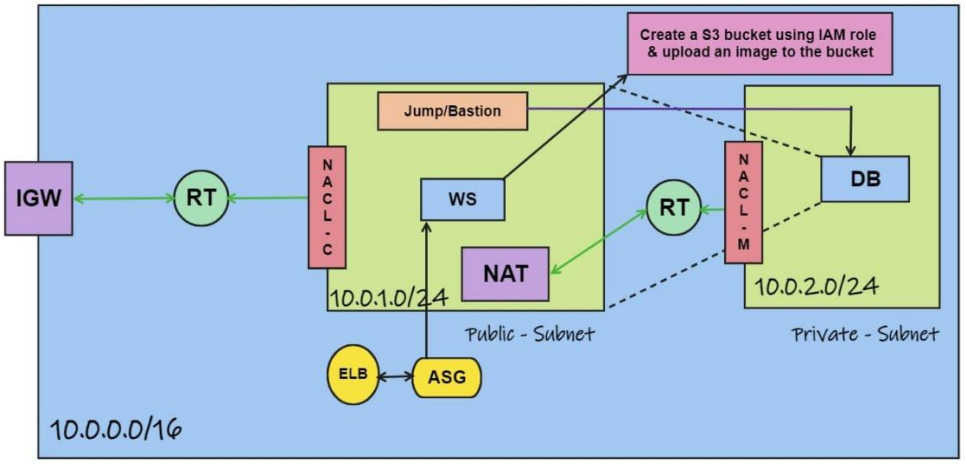
4.3 VPC

A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It allows you to launch AWS resources, such as Amazon EC2 instances, in a logically isolated virtual network that you define. Here are some key points about VPCs:

1. **Subnets**: Within the VPC, you can create multiple subnets. These subnets can be either public or private, depending on their routing configuration.

   a. **Public Subnets**: Subnets with a route table that has a route to an Internet Gateway, allowing instances in that subnet to communicate with the internet.
   b. **Private Subnets**: Subnets without a direct route to the internet, typically used for resources that don't need to be publicly accessible.

2. **Internet Gateway**: An internet gateway is a VPC component that allows communication between instances in the VPC and the internet.
3. **NAT Gateway**: A NAT (Network Address Translation) gateway is used to provide internet access for instances in private subnets. It allows outbound internet access while preventing inbound access.
4. **Route Tables**: Route tables control the routing of network traffic within the VPC. Each subnet is associated with a route table, which determines where the traffic is directed.
5. **Security Groups**: Security groups act as virtual firewalls, controlling inbound and outbound traffic to your EC2 instances.
6. **Network ACLs (NACLs)**: NACLs are an additional layer of security, controlling inbound and outbound traffic at the subnet level.
7. **EC2 Instances**: EC2 (Elastic Compute Cloud) instances can be launched within the VPC, either in public or private subnets, depending on their requirements.
8. **RDS Instances**: RDS (Relational Database Service) instances can be deployed within the VPC, typically in private subnets, to provide secure and scalable database services.

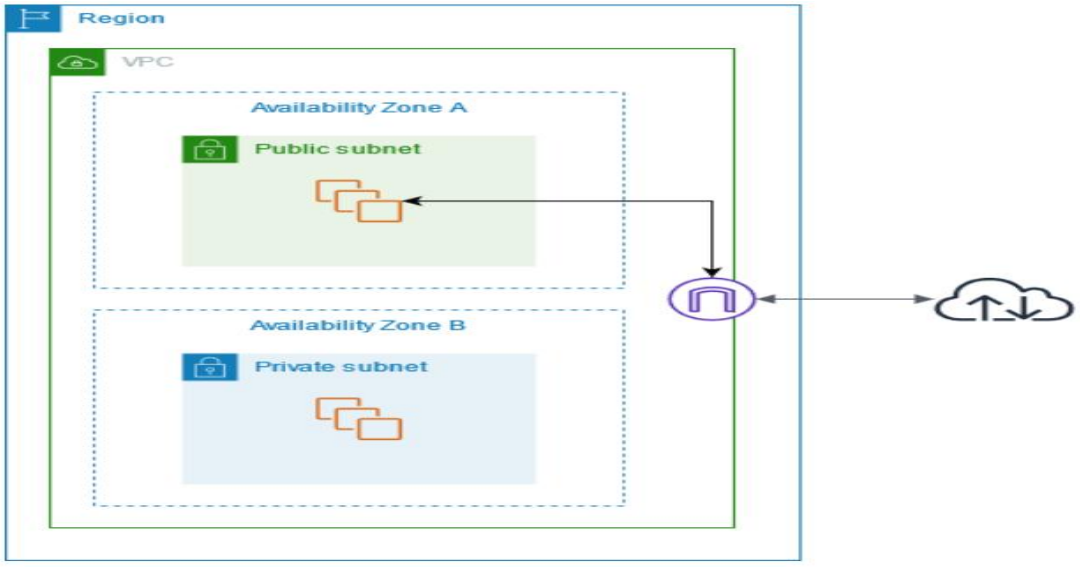


My-VPC (M)

4.4 IGW

An internet gateway (IGW) is a key component in a VPC that enables communication between resources in your VPC and the internet. Here are the key points about IGWs:

1. An IGW is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.
2. When you create a VPC, it does not automatically come with an IGW. You need to create and attach an IGW to your VPC to enable internet access.
3. After creating an IGW, you need to update the route table of your public subnets to route internet-bound traffic (0.0.0.0/0) to the IGW.
4. Instances launched in public subnets with a route to the IGW can access the internet directly. Instances in private subnets can access the internet by using a Network Address Translation (NAT) gateway.

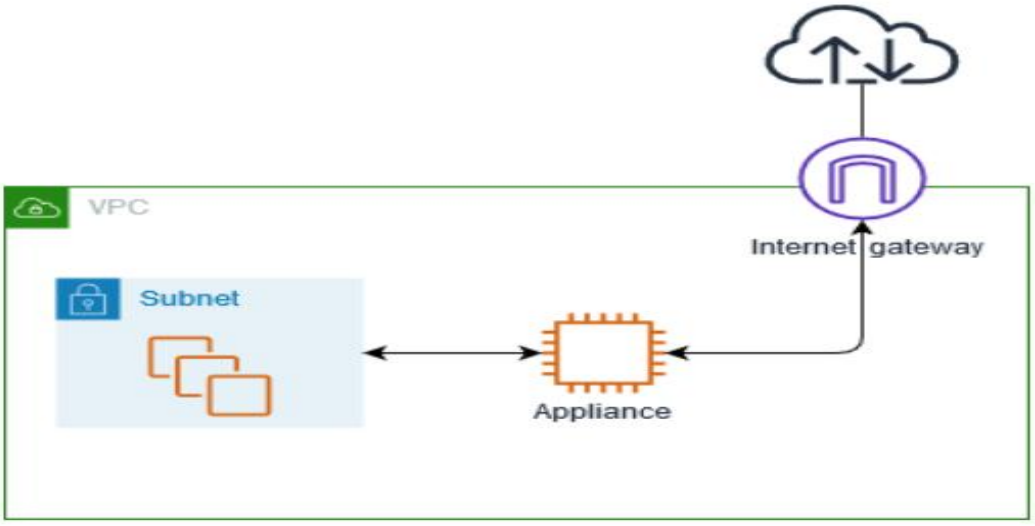


4.5 Route Tables

.

Route tables are a key component of a VPC that control the routing of network traffic within the VPC and to external networks. Here are the key points about route tables in a VPC:

1. Each VPC has a main route table that is automatically created and associated with all subnets by default. This main route table contains a local route that enables communication between resources within the VPC.
2. You can create additional custom route tables and associate them with specific subnets within the VPC. This allows you to have different routing configurations for different parts of your VPC.
3. Route tables contain routes that specify the destination CIDR block and the target (e.g., internet gateway, virtual private gateway, network interface) for the traffic. The most specific route matching the destination is used for routing the traffic.
4. To add a route to a route table, you can use the AWS CLI command aws ec2 create-route or the AWS Tools for Windows PowerShell cmdlet Register-EC2RouteTable.
5. When using AWS CloudFormation to create a VPC, you cannot directly reference the main route table. Instead, you need to create custom route tables and associate them with your subnets.



4.6 Subnet

A VPC subnet is a range of IP addresses within a VPC. Here are the key points about VPC subnets:

1. Subnets are used to segment the VPC's IP address range into smaller, more manageable networks.
2. Each subnet must reside entirely within a single Availability Zone and cannot span multiple zones.
3. Subnets can be classified as either public or private based on their ability to communicate directly with the internet:

   a. Public subnets have a route to an internet gateway, allowing instances in those subnets to communicate directly with the internet.

b. Private subnets do not have a direct route to the internet and require a NAT gateway or other mechanism to access the internet.

4. When creating a subnet, you specify the Availability Zone and the IPv4 CIDR block for the subnet. You can also optionally specify an IPv6 CIDR block.
5. You can modify certain subnet settings after creation, such as the auto-assign public IP address feature and the resource-based hostname settings.

Public Subnet:

1. A public subnet is a subnet that has a route table with a route to an internet gateway, allowing instances in that subnet to communicate with the internet.
2. Instances in a public subnet are assigned a public IP address and can be directly accessed from the internet.
3. Public subnets are typically used for resources that need to be accessible from the internet, such as web servers.

Private Subnet:

1. A private subnet is a subnet that does not have a direct route to the internet.
2. Instances in a private subnet are assigned a private IP address and cannot be directly accessed from the internet.
3. Private subnets are typically used for resources that do not need to be directly accessible from the internet, such as application servers or databases.

Accessing Instances in Private Subnets:

1. To access instances in a private subnet from the internet, you can use a bastion host (also known as a jump box) in a public subnet.
2. The bastion host acts as a secure entry point to connect to instances in the private subnet using SSH or RDP.
3. Alternatively, you can use an Elastic Load Balancer (ELB) in a public subnet to manage and distribute traffic to instances in a private subnet.

Subnet CIDR Blocks:

1. The CIDR block (Classless Inter-Domain Routing) of a subnet determines whether it is public or private.
2. Typically, a public subnet has a CIDR block that falls within the range of 0.0.0.0/0, while a private subnet has a CIDR block that falls within a more restricted range, such as 10.0.0.0/16 or 172.16.0.0/16.

4.7 Servers

an overview of how web servers and database servers work in the context of AWS:

1. **Web Servers**:

a) Web servers are responsible for handling and responding to HTTP/HTTPS requests from clients (e.g., web browsers).
b) In AWS, you can host web servers on Amazon EC2 instances, which provide the compute capacity to run your web application code.
c) To improve scalability and availability, you can use Elastic Load Balancing (ELB) to distribute traffic across multiple EC2 instances.

d) For static web content, you can also use Amazon S3 and Amazon CloudFront to provide fast, secure, and scalable content delivery.

1. **Database Servers**:

1. Database servers are responsible for storing and managing the data used by your web application.
2. In AWS, you have several options for hosting your database:

a) Amazon RDS (Relational Database Service) - Managed service for popular relational databases like MySQL, PostgreSQL, Oracle, and SQL Server.
b) Amazon DynamoDB - Fully managed NoSQL database service for high-performance, low-latency applications.
c) Amazon Aurora - Highly scalable, cloud-native relational database that is compatible with MySQL and PostgreSQL.

2. **Connecting Web Servers and Database Servers**:

1. Web servers typically connect to the database servers to retrieve or store data required by the web application.
2. In AWS, you can configure the network settings (e.g., security groups, network ACLs) to allow secure communication between the web servers and database servers.
3. For better security and isolation, you can place the web servers in a public subnet and the database servers in a private subnet, and use a NAT Gateway to enable outbound internet access for the private subnet.

4.8 S3

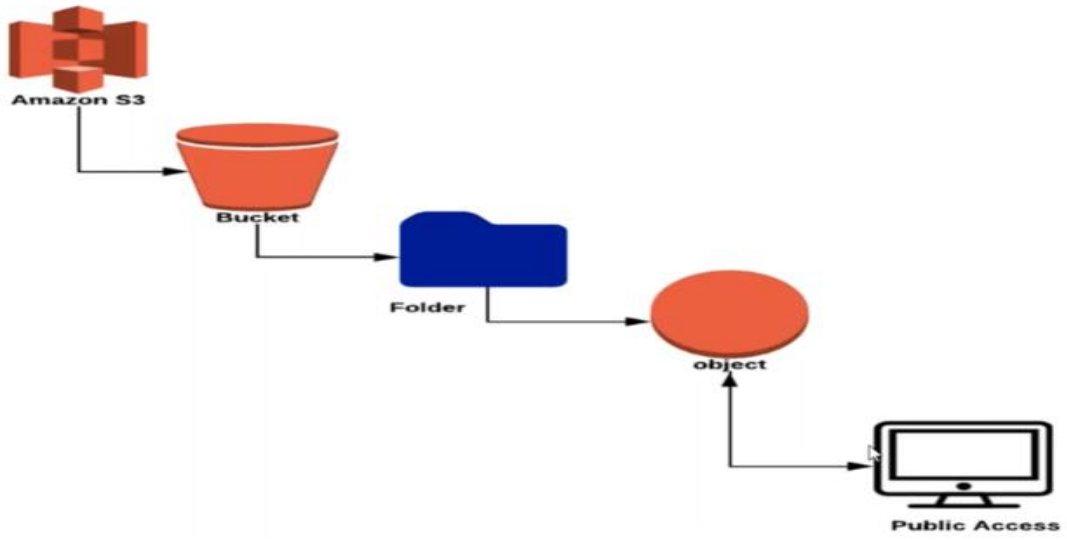
About Amazon S3 (Simple Storage Service) services:

1. Security:

a) Amazon S3 is secure by default. Only the bucket owner has access to the buckets they create.
b) You can use access control mechanisms like bucket policies to selectively grant permissions to users and groups.
c) Amazon S3 supports secure data transfer via SSL/HTTPS endpoints.
d) Amazon S3 automatically encrypts all object uploads by default.

2. S3 Capabilities:

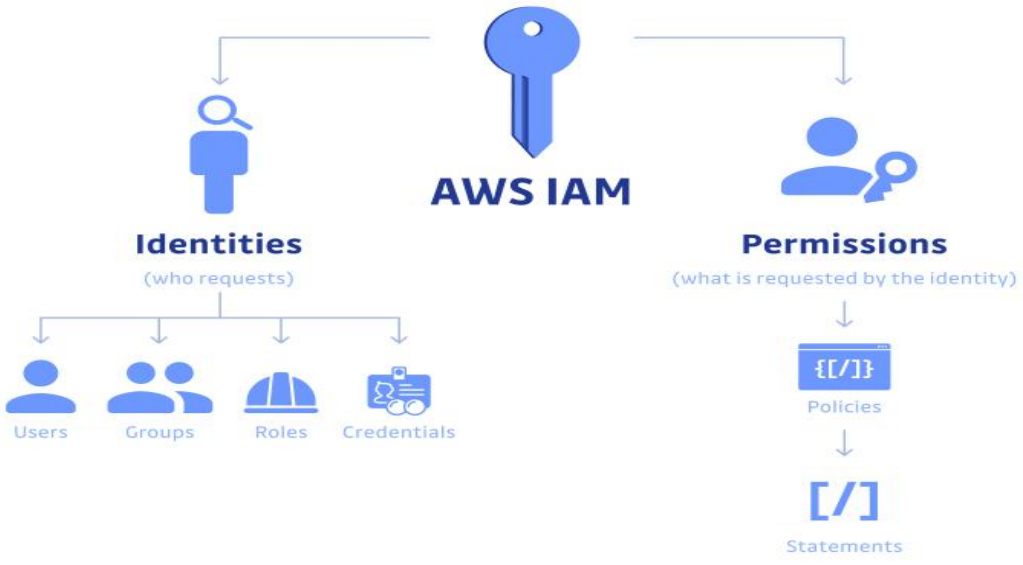
a) Amazon S3 provides a simple web service interface to store and retrieve any amount of data from anywhere.
b) It is highly scalable, and you only pay for what you use, allowing you to start small and grow as needed.
c) S3 is designed to be highly flexible, allowing you to store any type and amount of data.
d) You can use S3 to build a wide range of applications, from simple FTP to sophisticated web applications.



4.9 IAM

AWS Identity and Access Management (IAM):

1. IAM is an AWS service that helps administrators securely control access to AWS resources. It allows you to manage user identities and their permissions.
2. IAM provides fine-grained access control across all of AWS. You can use IAM policies to manage permissions for your workforce and systems, ensuring the principle of least privilege.
3. IAM supports attribute-based access control (ABAC), which allows you to create fine-grained permissions based on user attributes like department, job role, or team name. This can help reduce the number of distinct permissions required.
4. IAM is integrated with several other AWS services. You can refer to the AWS documentation to see the list of services that work with IAM.
5. When using AWS resources, you often need to provide an IAM role. This role determines the level of access you want to grant, including the operations and AWS resources (e.g., Amazon S3 buckets) that the user or system can access.



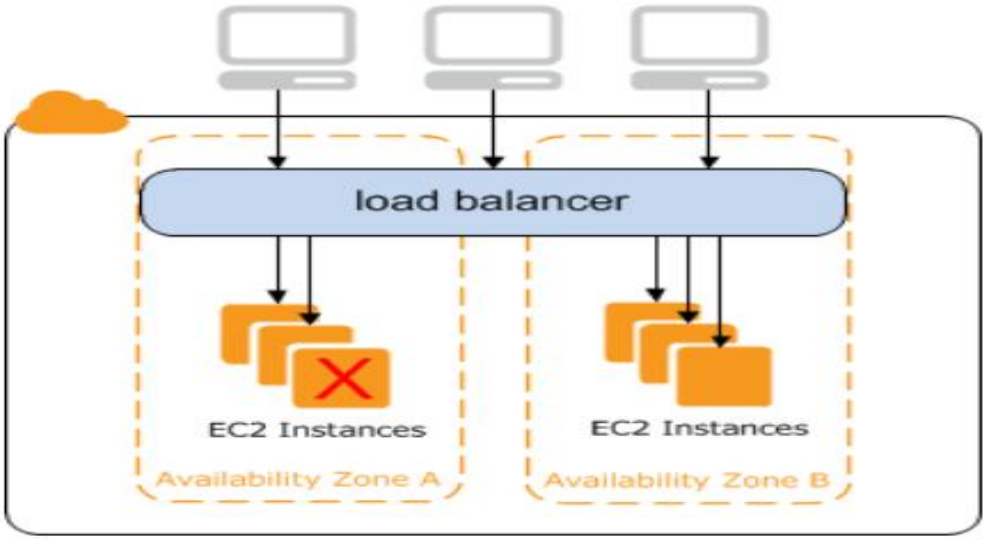
4.10 Load balancers

AWS Load Balancers:

1. **Load Balancer Types**: AWS provides multiple load balancer options to suit different application requirements:
1. Application Load Balancer (ALB) - Best suited for HTTP/HTTPS traffic and advanced request routing for microservices and containers.
2. Network Load Balancer (NLB) - Best suited for TCP traffic where extreme performance and low latency are required.
2. **Load Balancer Features**:

   a. **Preserve Source IP**: NLB preserves the client's source IP address, allowing backend applications to see the original client IP.
   b. **Static IP Support**: NLB automatically provides a static IP per Availability Zone that can be used as the frontend IP.
   c. **Elastic IP Support**: NLB allows you to assign an Elastic IP per Availability Zone as the frontend IP.
   d. **DNS Failover**: If there are no healthy targets or the load balancer nodes are unhealthy, Route 53 will direct traffic to load balancer nodes in other Availability Zones.
   e. **Sticky Sessions**: ALB supports both duration-based and application-based sticky sessions to route requests from the same client to the same target.

3. **Load Balancer Selection**: When choosing a load balancer, consider factors like protocol, target type, application requirements, and placement (Region, Local Zone, Outpost).

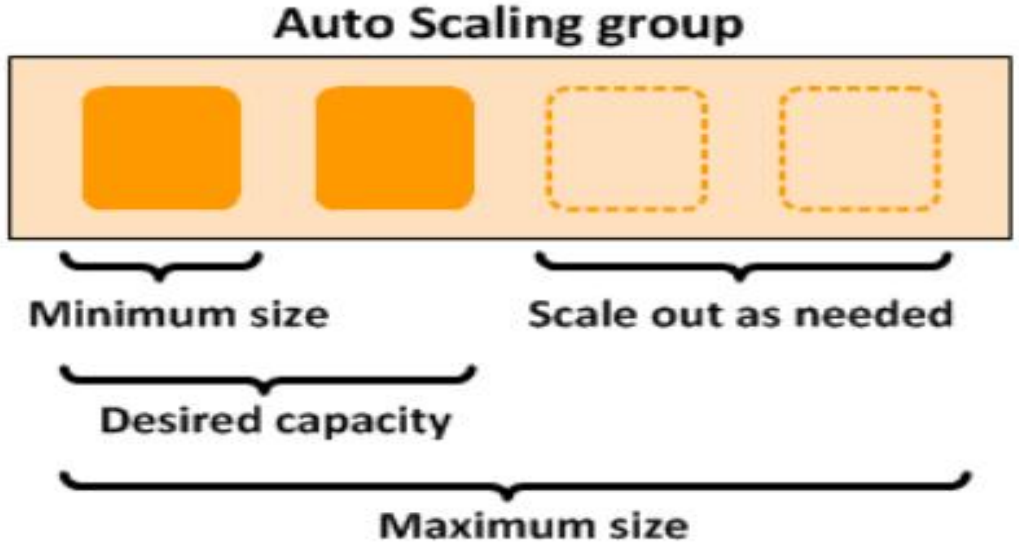


4.11 ASG

AWS Auto Scaling Groups (ASGs):

1. An ASG is a collection of EC2 instances that share similar characteristics and are treated as a logical grouping for the purposes of fleet management and dynamic scaling.
2. ASGs allow you to automatically adjust the number of instances in the group based on criteria you define, such as maintaining a fixed number of instances or scaling up/down based on demand.
3. ASGs can help ensure application availability by automatically detecting and replacing unhealthy instances.

4. When instances are launched, ASGs will distribute the desired capacity across the Availability Zones you specify to maintain balance.
5. ASGs support the use of Spot Instances, which can provide steep discounts compared to On-Demand prices. When a Spot Instance is terminated, the ASG will attempt to launch a replacement instance.
6. You can use AWS Auto Scaling to manage the scaling of your ASGs, which simplifies the scaling experience by allowing you to scale collections of related resources that support your application.

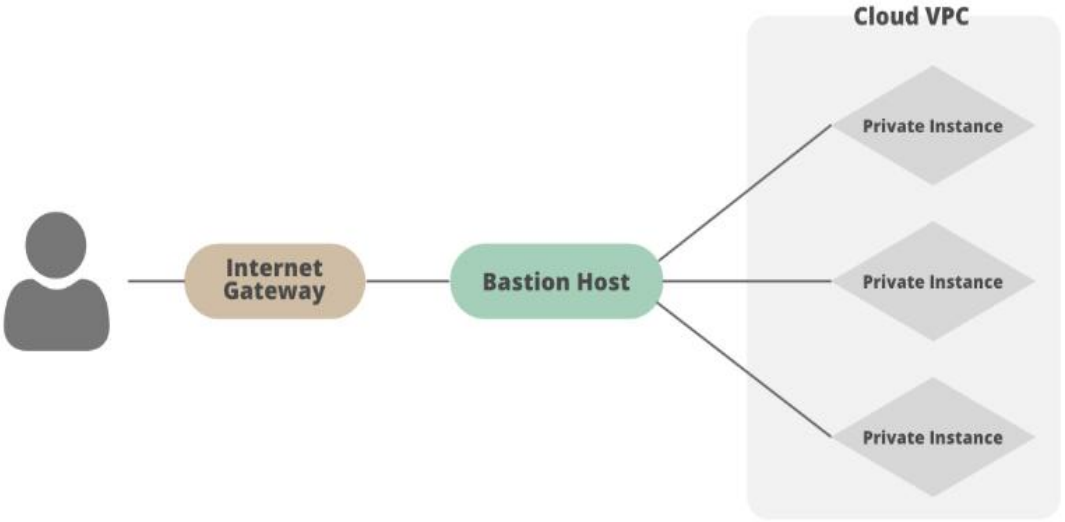


4.12 Bastion host

A bastion host, also known as a jump box, is a special-purpose computer on a network specifically designed and configured to withstand attacks. The primary purpose of a bastion host is to provide a controlled and secure access point for users to connect to private resources within a virtual private cloud (VPC) or private network.

Here are the key points about using a bastion host in AWS:

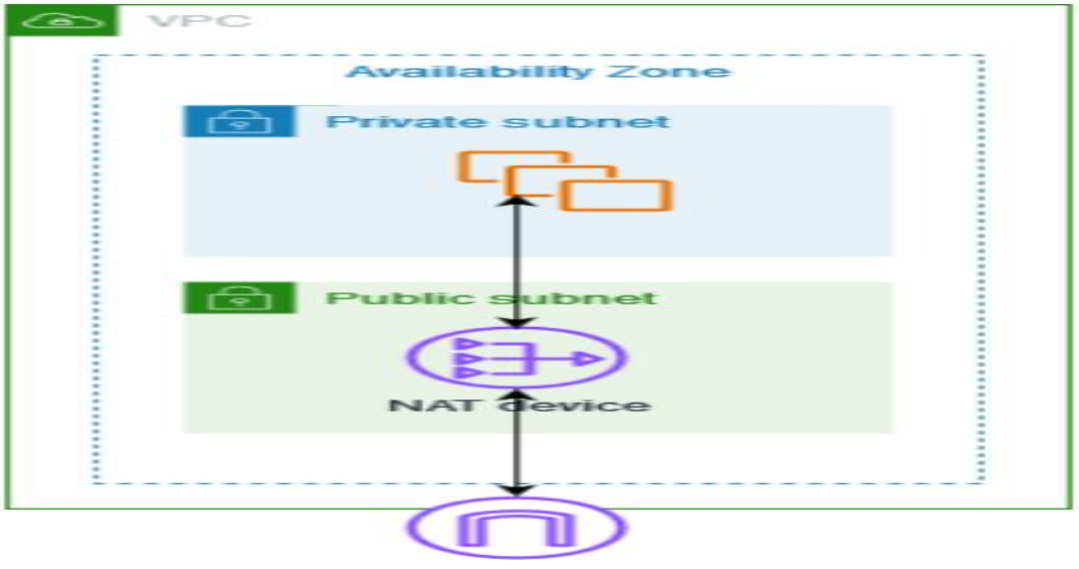
1. Bastion hosts are typically launched in a public subnet, allowing them to be accessed from the internet. This provides a secure entry point to connect to resources in private subnets.
2. To connect to a private EC2 instance from outside the VPC, you would first connect to the bastion host using SSH or RDP, and then from the bastion host, you can connect to the private instance.
3. To allow the bastion host to connect to the private EC2 instance, you need to update the security group of the private instance to allow SSH access from the bastion host's private IP address.
4. If you have a custom Network ACL (NACL) attached to the private subnet, you must also whitelist the private IP range of the bastion host in the NACL rules.
5. When connecting to the private EC2 instance from the bastion host, you can use the same SSH key pair that was used to launch the private instance.
6. For Windows instances, you can use a tool like PuTTY and Pageant to manage your SSH keys and connect to the private instance through the bastion host.



4.13 NAT Gateway

An overview of what a NAT Gateway is in AWS:

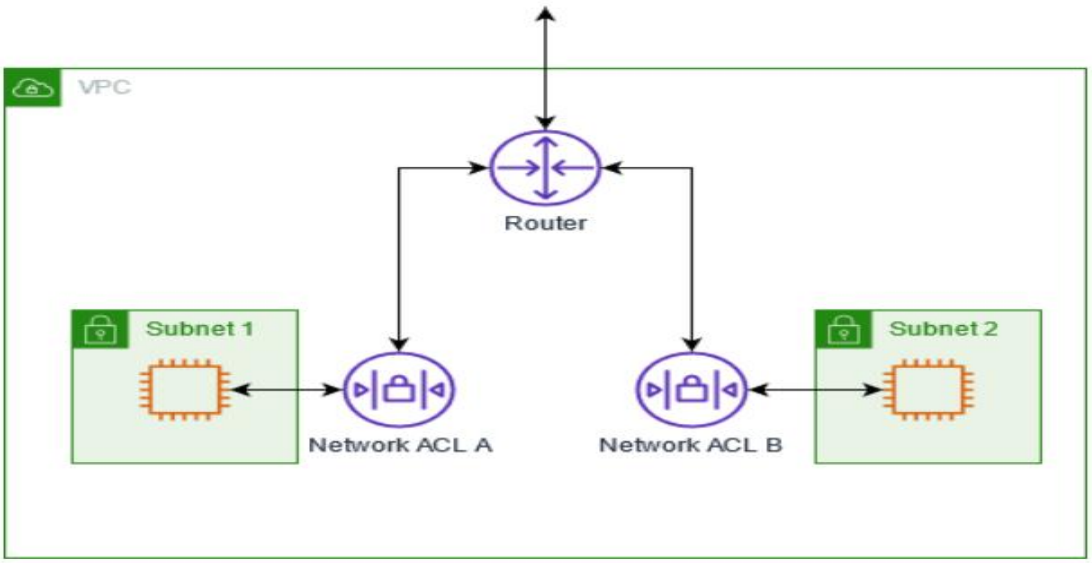
1. A NAT (Network Address Translation) Gateway is a managed service provided by AWS that enables instances in a private subnet to connect to the internet or other AWS services, while preventing the internet from initiating a connection with those instances.
2. The NAT Gateway performs both address translation and port address translation (PAT) to enable this connectivity. It is a highly available and scalable NAT service that removes the need to manage your own NAT instances.
3. You can use a NAT Gateway to allow instances in a private subnet to access the internet or other AWS services, while keeping those instances private and inaccessible from the internet.
4. There are two types of traffic that can go through a NAT Gateway - internet-bound traffic and AWS-bound traffic. The NAT Gateway processing charges are based on the amount of traffic (in GB) that traverses the gateway, regardless of the traffic type.
5. To identify whether the NAT Gateway processing charges are predominantly due to internet-bound or AWS-bound traffic, you can use AWS Cost Explorer. This allows you to analyze the data transfer metrics and narrow down the root cause.



4.14 NACL

A Network ACL (NACL) is a stateless firewall that controls inbound and outbound traffic at the subnet level in a VPC. Here are the key points about NACLs in AWS:

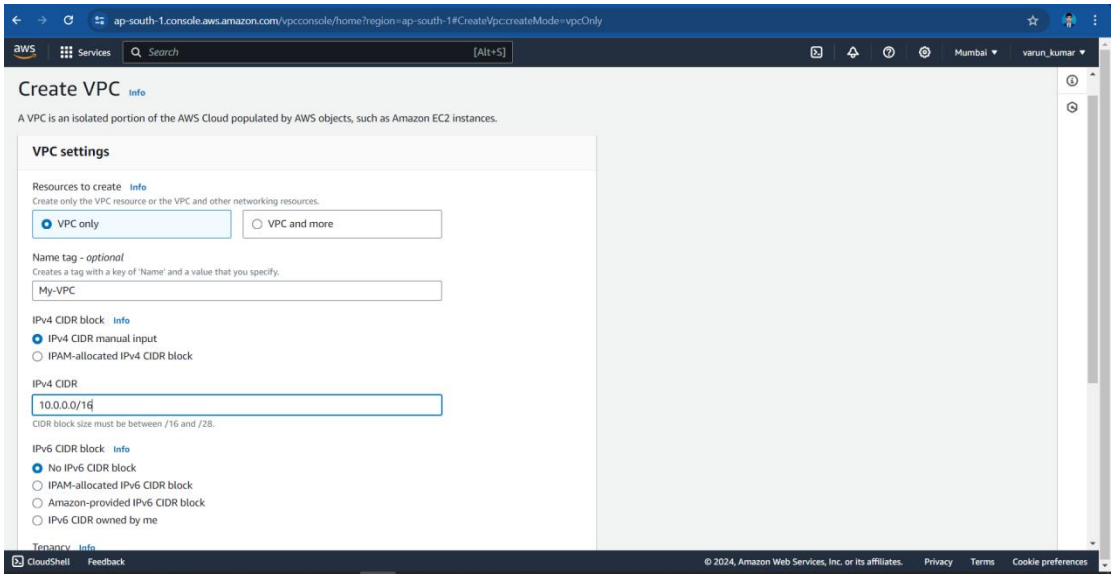
1. NACLs are an optional additional layer of security for your VPC, complementing the security group rules applied at the instance level.
2. NACLs have separate inbound and outbound rules, and each rule can either allow or deny traffic based on IP addresses, ports, and protocols.
3. NACL rules are evaluated in numerical order, starting from the lowest rule number. The first rule that matches the traffic is applied, so the order of rules is important.
4. NACLs are stateless, meaning they do not track connection state like security groups. Each packet is evaluated independently based on the NACL rules.
5. AWS Firewall Manager now allows customers to centrally create, deploy, and manage NACL rules across multiple accounts in an AWS Organization.
6. In AMS Managed Multi-Account Landing Zones, the use of NACLs is limited to application accounts, and they can only be used as a deny list to allow AMS monitoring and management operations.

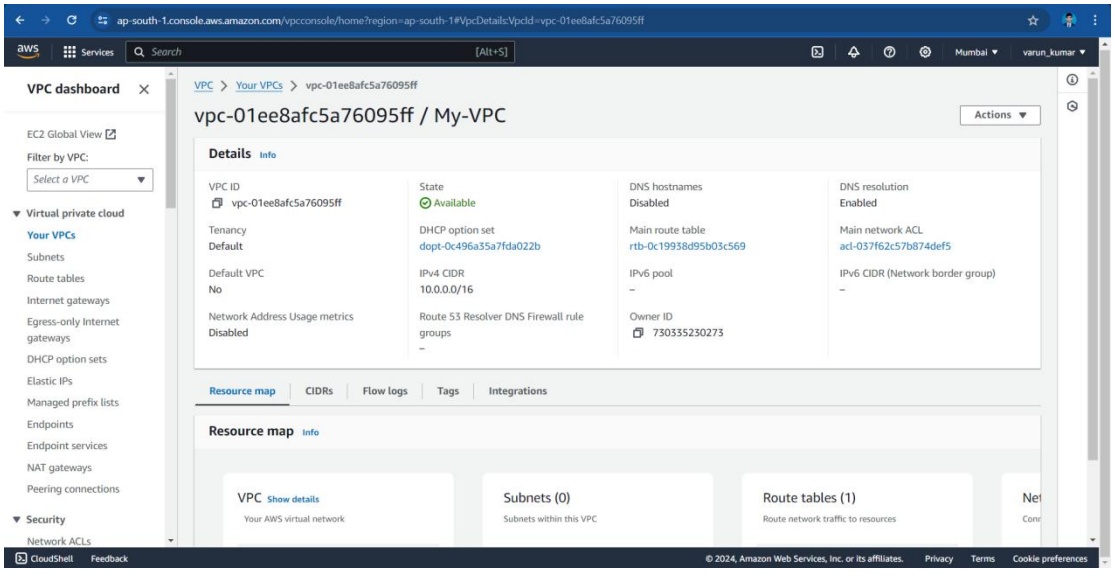


5.Project Implementation and Screenshots

5.1 Creating a VPC (Virtual Private Cloud):

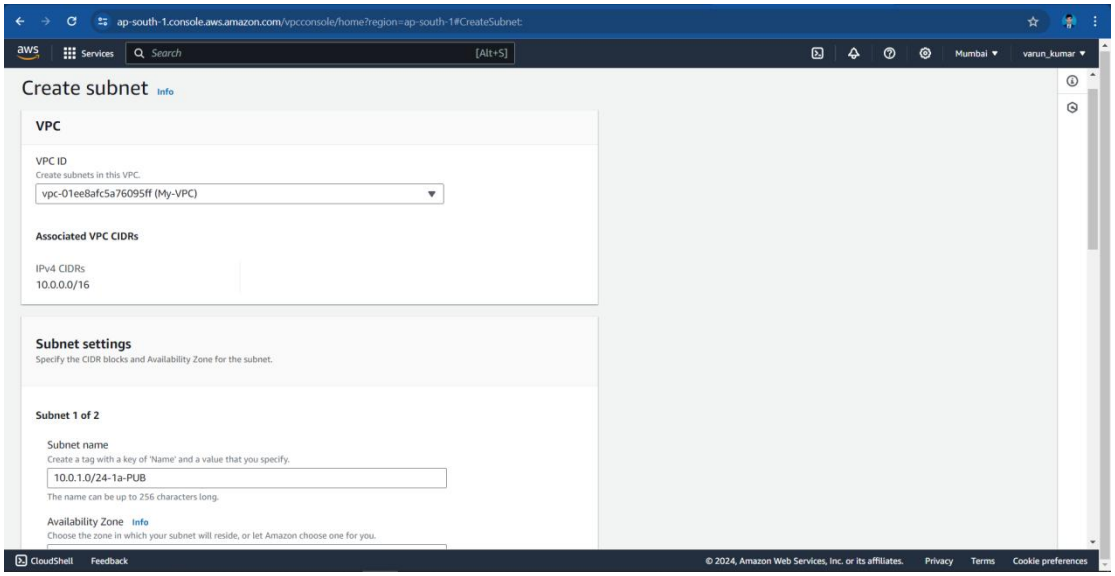
- In the VPC dashboard click on the “create VPC” Button to start the VPC creation wizard.
- Configure the VPC settings:
  - Provide a name for your VPC.
  - Specify the IPv4 CIDR block for your VPC's IP address range (10.0.0.0/16).
  - Optionally, you can assign an IPv6 CIDR block to your VPC.
- Configure 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.
- Review all the configuration details and settings for your VPC. If everything looks correct, click on the "Create VPC" button to create your VPC.





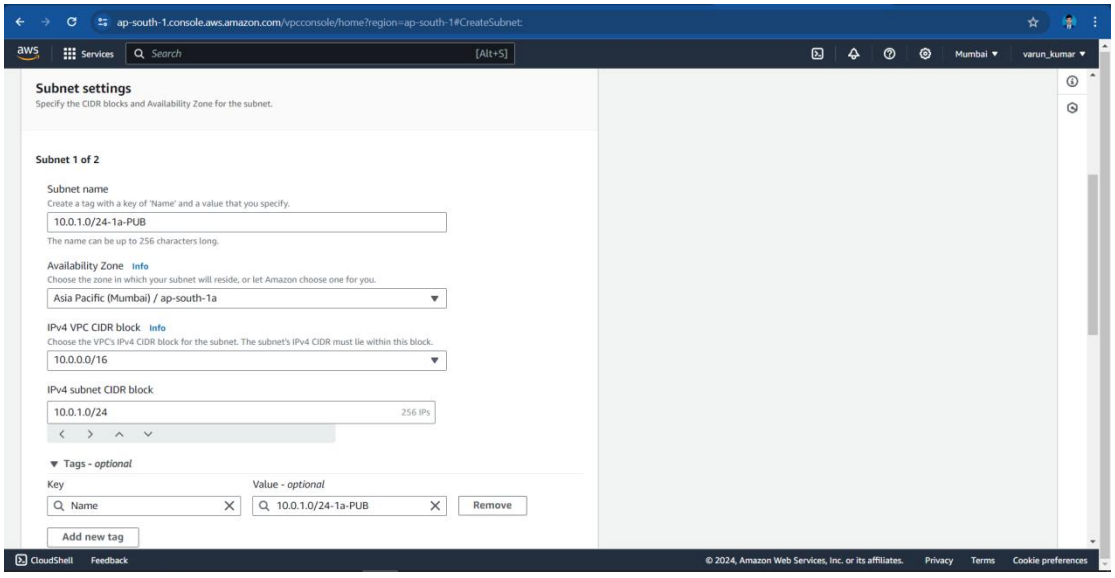
5.2 Creating Subnet

Click on the "Create Subnet" button to create a new subnet.



5.2.1 Creating Public Subnet

- Click on the "Create Subnet" button to create a new subnet.
- Configure the subnet settings:
  - Select the VPC in which you want to create the subnet.
  - Provide a name and a suitable CIDR block for the subnet. Ensure that the CIDR block falls within the IP address range of the VPC and doesn't overlap with other subnets
- Select the desired availability zone for the subnet. It's recommended to create subnets in multiple availability zones for high availability and fault tolerance.



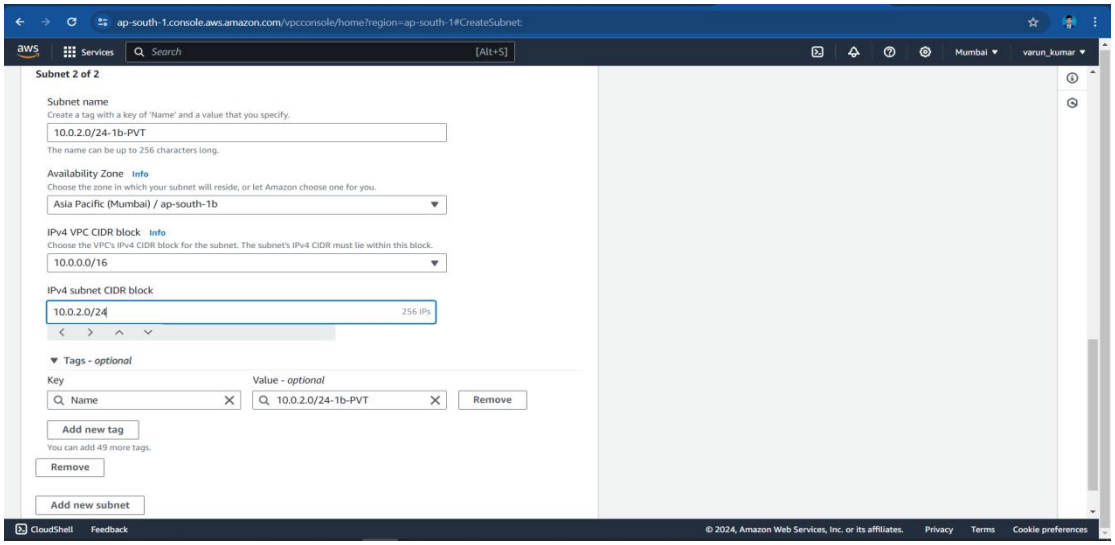
5.2.2 Creating Private Subnet

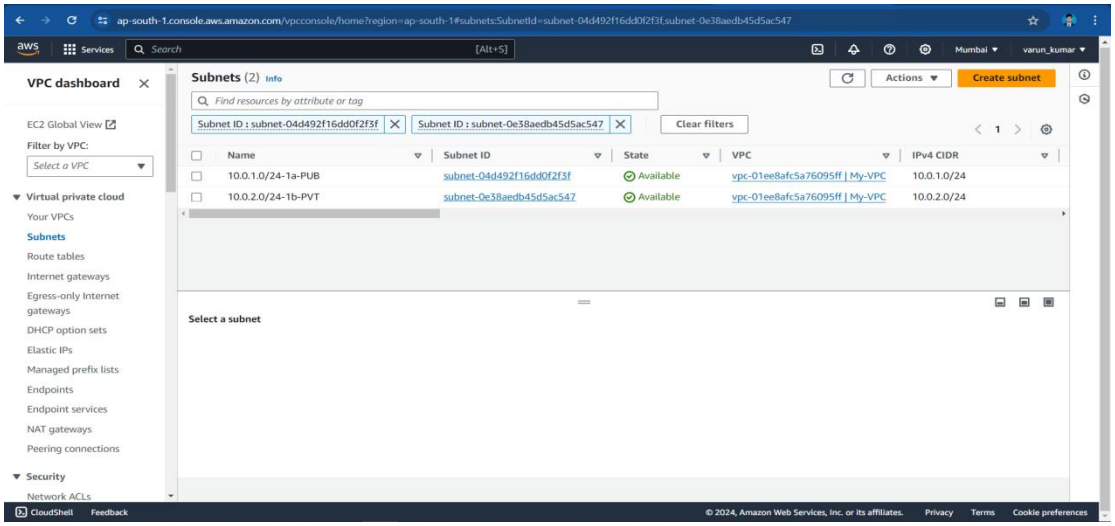
- Click on the "Create Subnet" button to create a new subnet.
- Configure the subnet settings:
  - Select the VPC in which you want to create the subnet.
  - Provide a name and a suitable CIDR block for the subnet. Ensure that the CIDR block falls within the IP address range of the VPC and doesn't overlap with other subnets
- Select the desired availability zone for the subnet. It's recommended to create subnets

in multiple availability zones for high availability and fault tolerance.

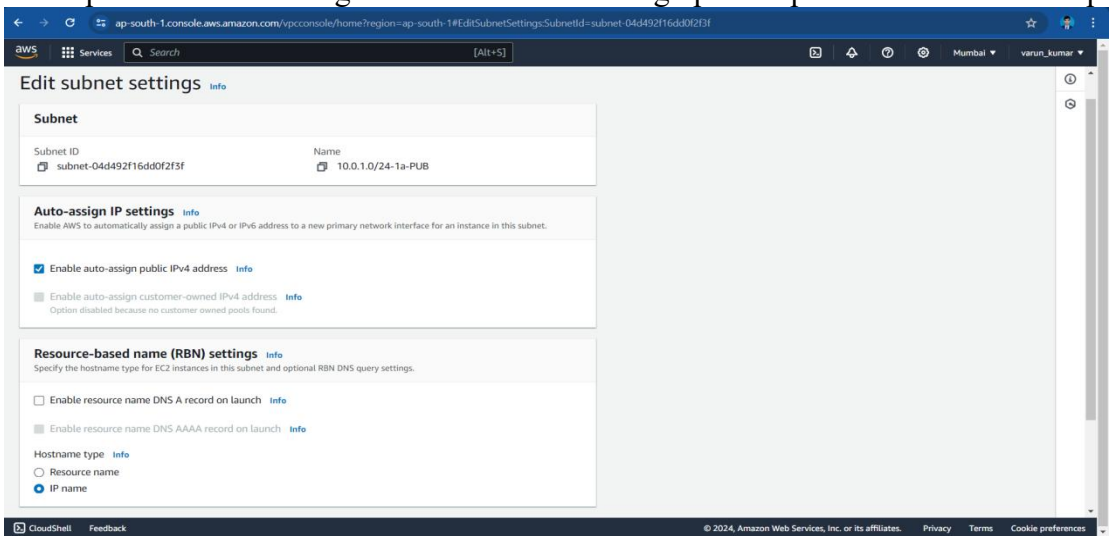
- Configure the subnet's route table:
- Choose an existing route table or create a new one for the subnet. To make the subnet public, associate it with a route table that has a route to an internet gateway.

Verify the details of the subnet, including the VPC, CIDR block, availability zone, route table, and NACL settings.



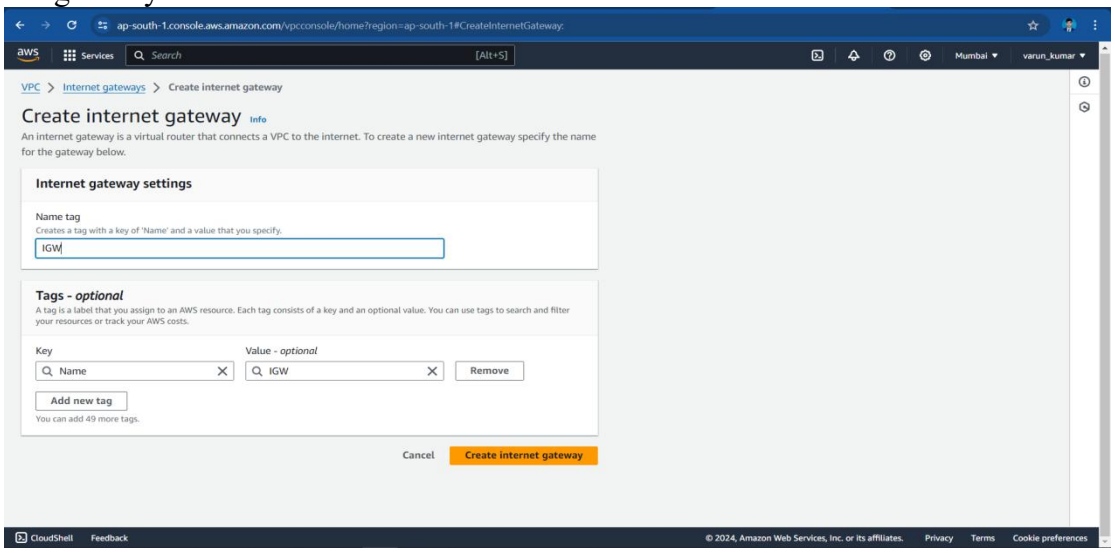


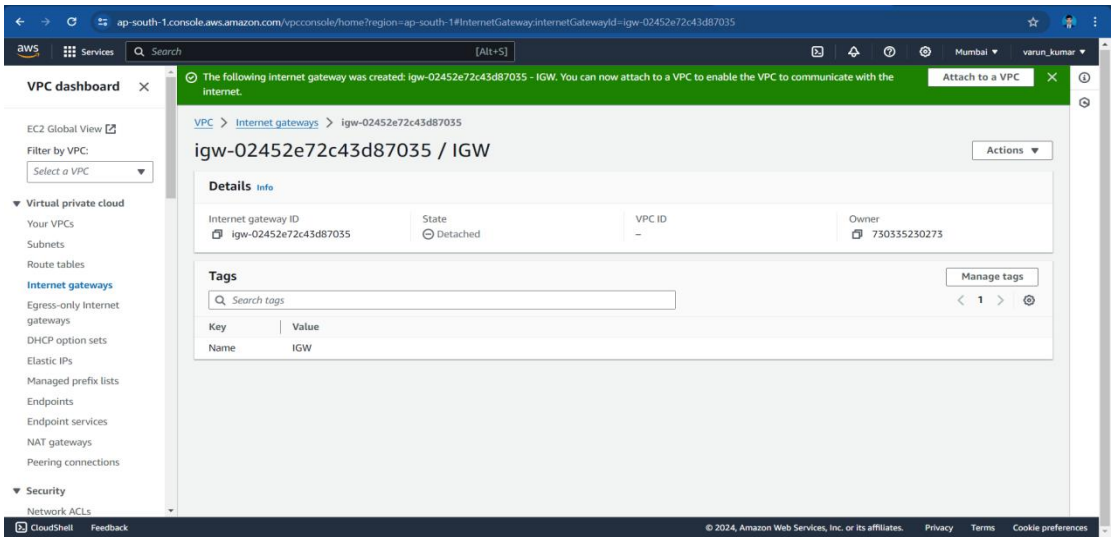
In the public subnet setting we enable the auto assign public ipv4 address to make it public.



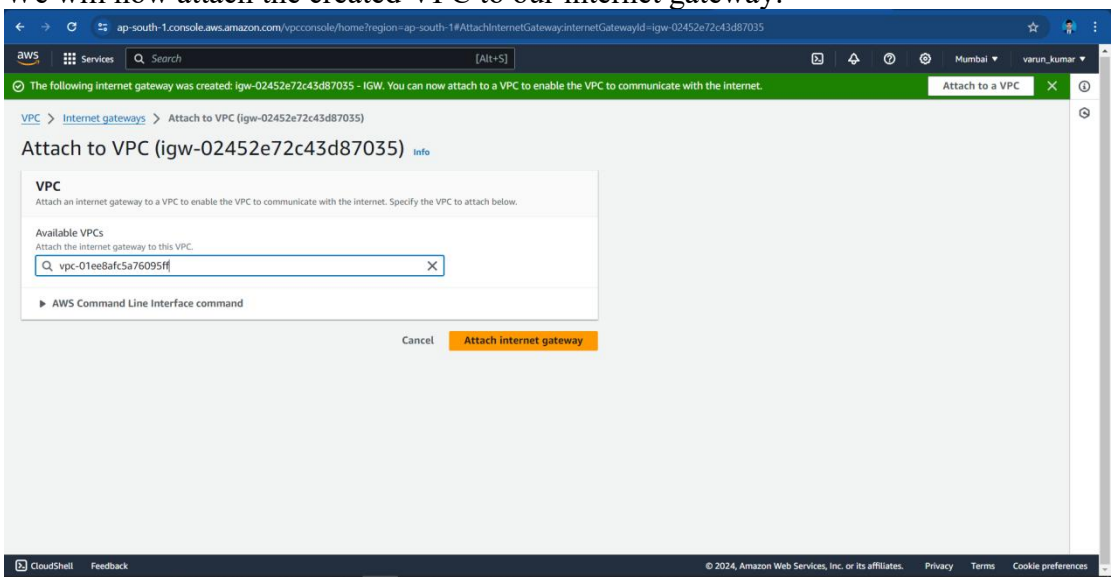
5.3 Creating Internet Gateway

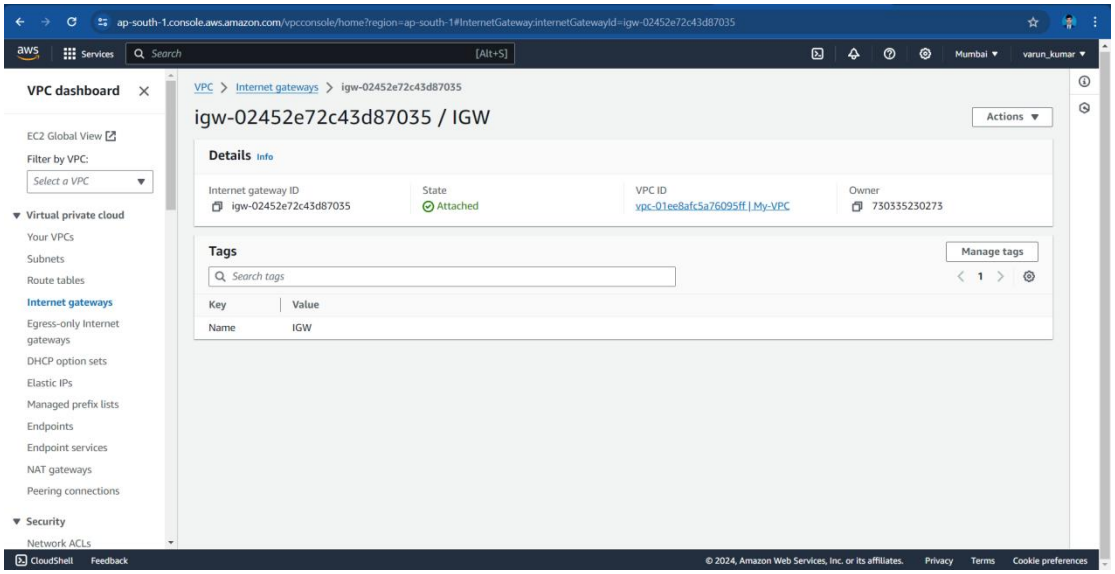
We will now create a internet gateway provide a name tag and click on create internet gateway.





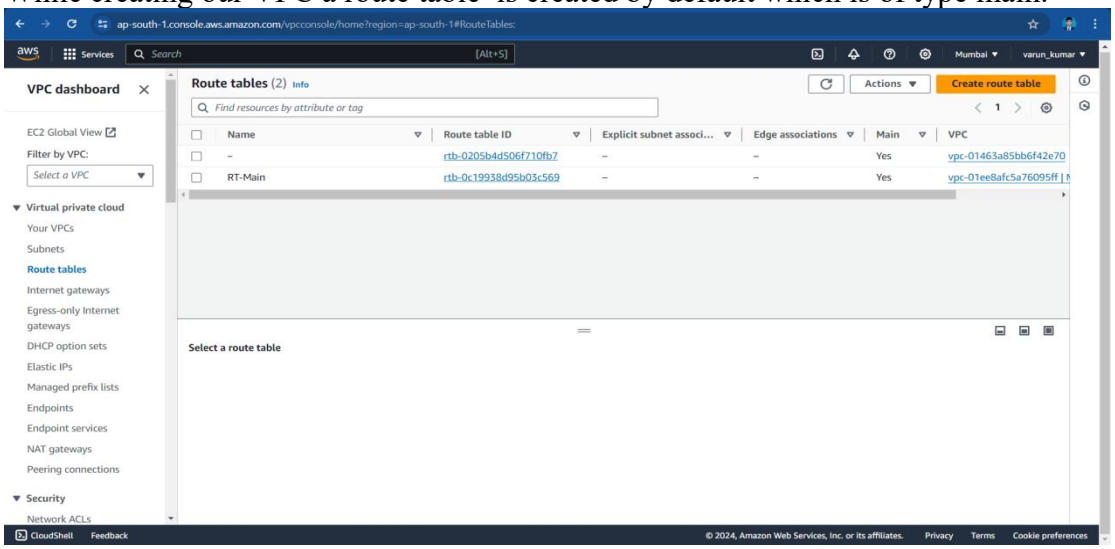
We will now attach the created VPC to our internet gateway.



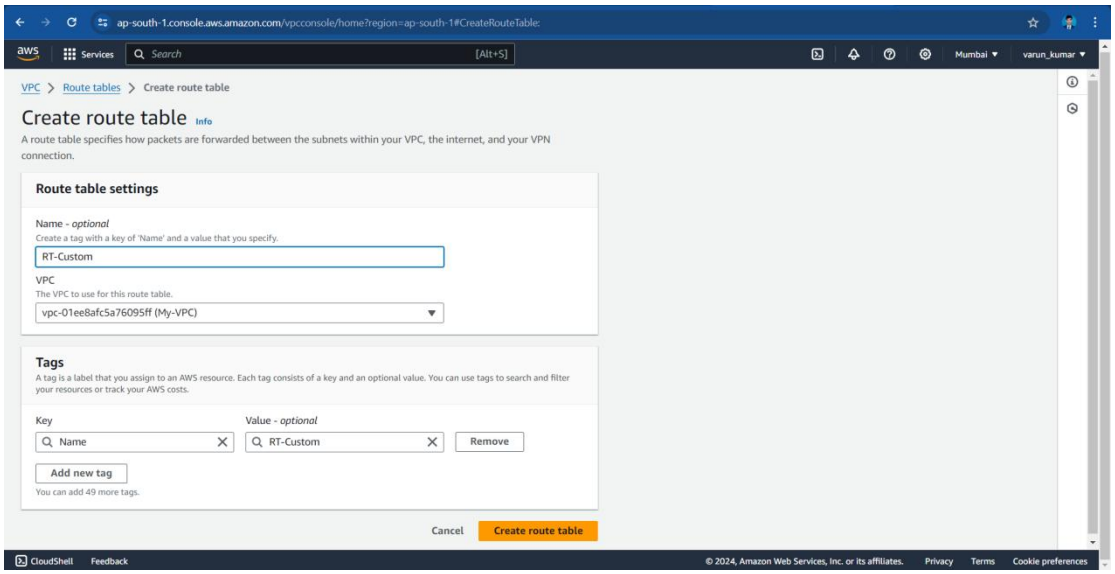


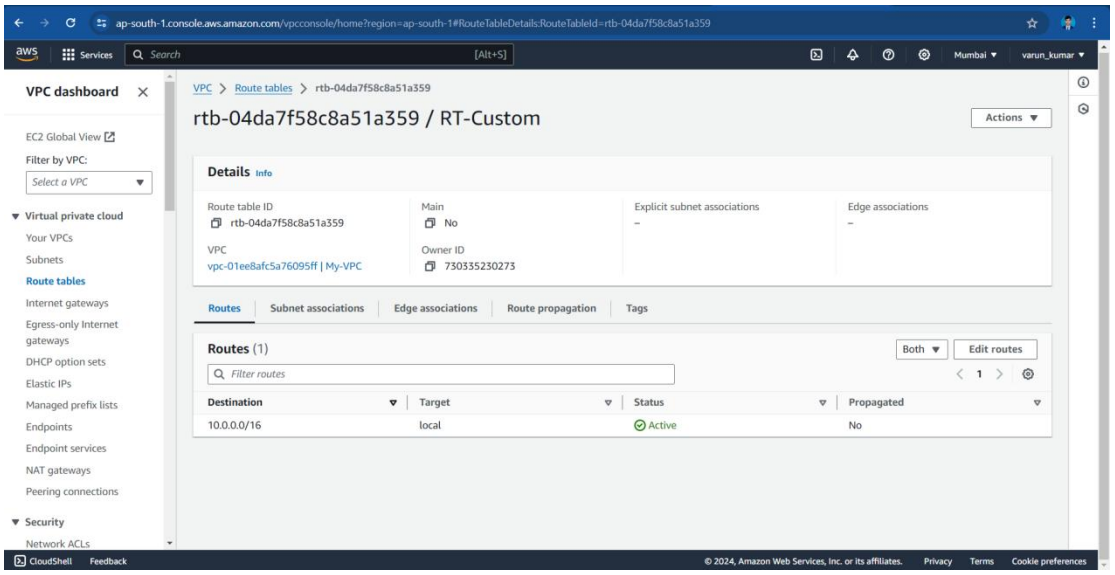
5.4 Creating Route Table

While creating our VPC a route table is created by default which is of type main.

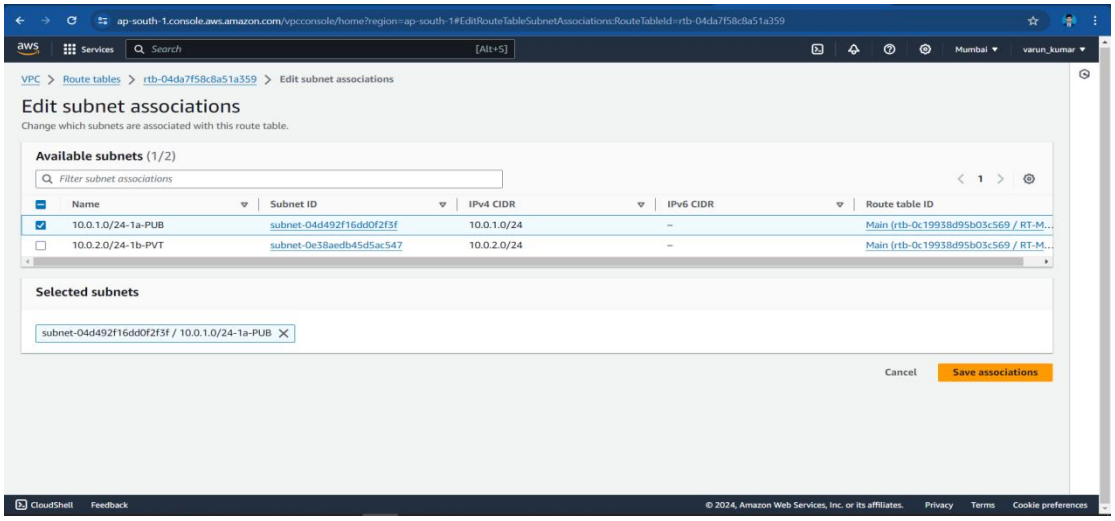


Now we will create a custom route table provide a name to the route table and the VPC you want to connect and click on create.

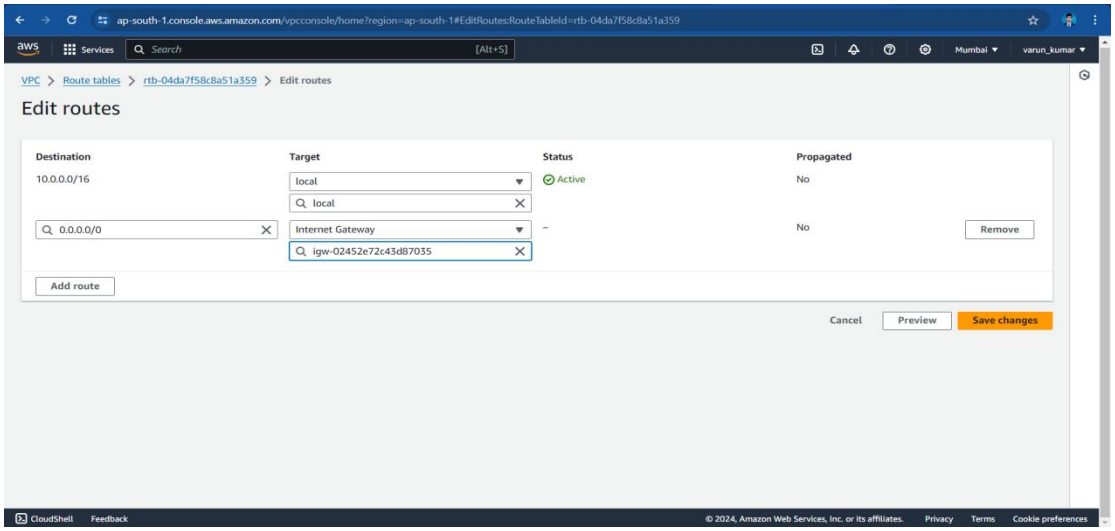




We will now attach the custom route table with the public subnet by editing the subnet associations.



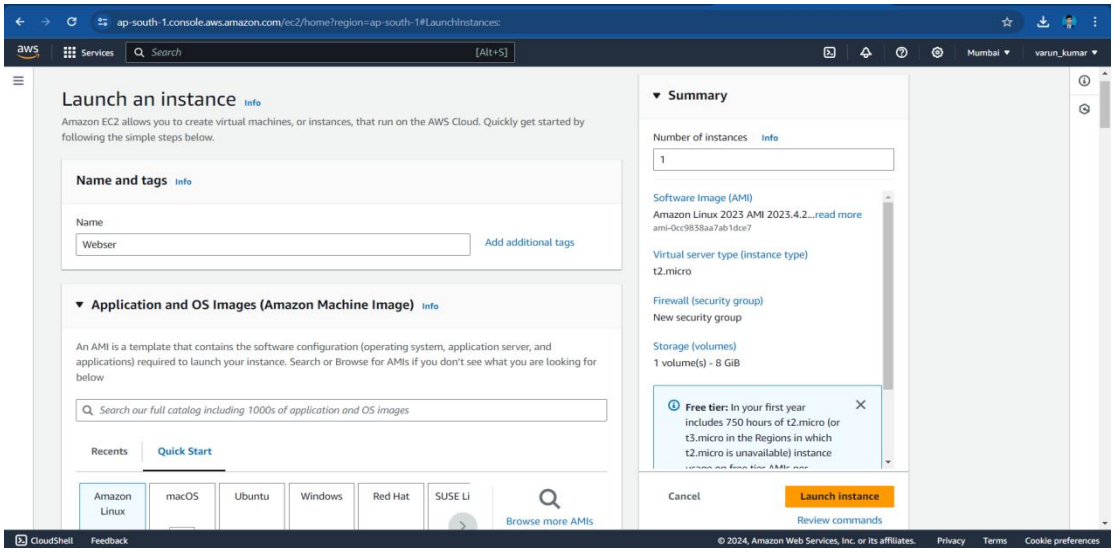
We will attach the internet gateway to our route table by editing the routes option in the route table.

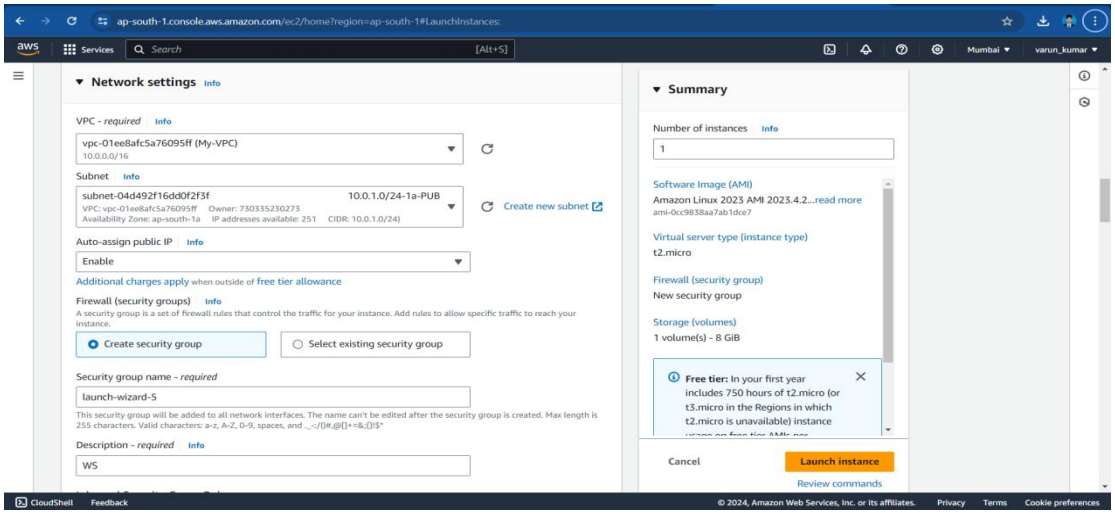


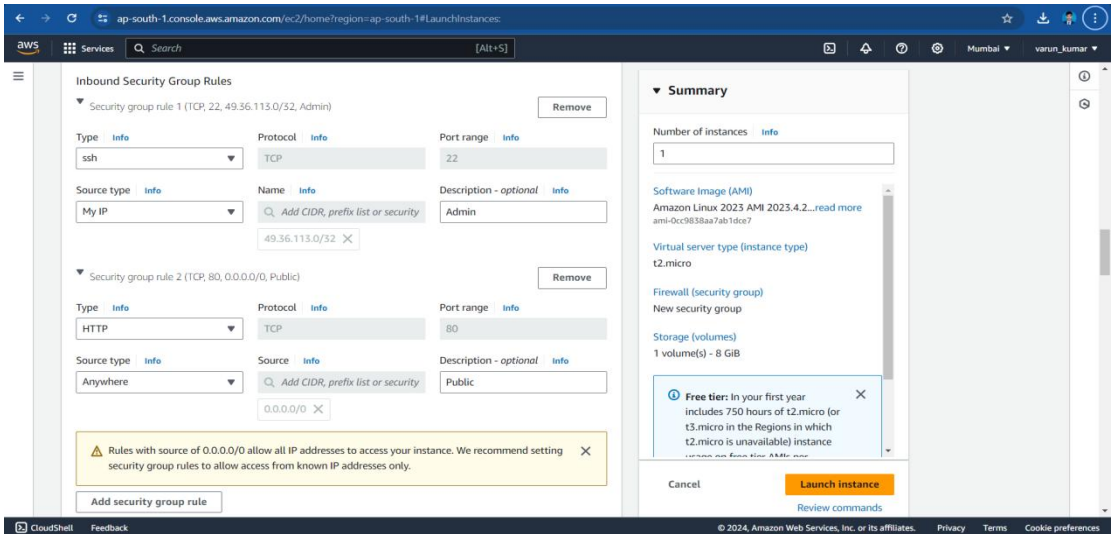
5.5 Creaing Public Instance

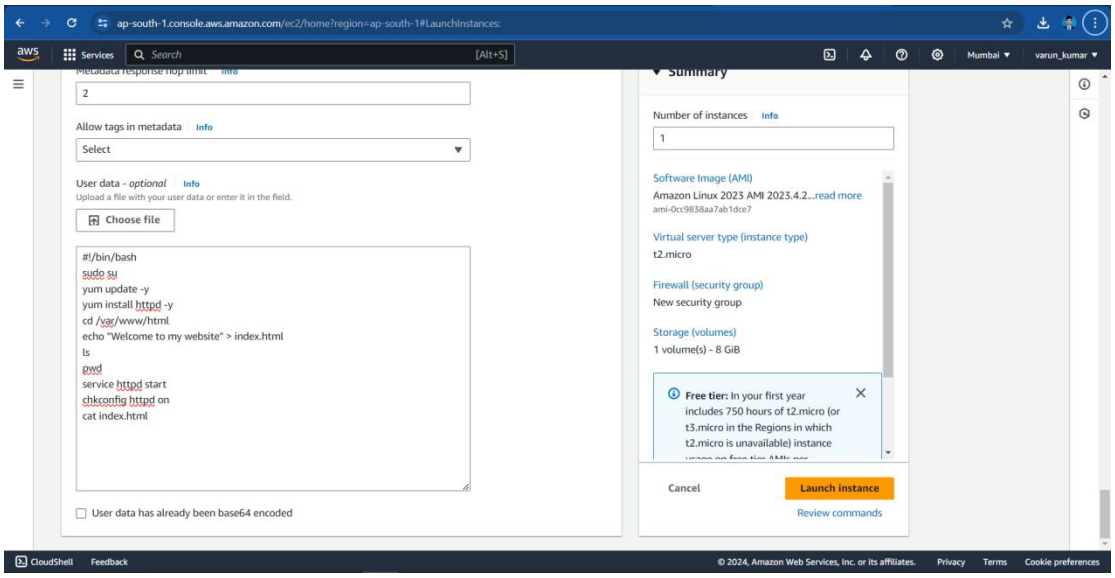
We will now create an instance and attach it on the public subnet on this instance we will host our web server.

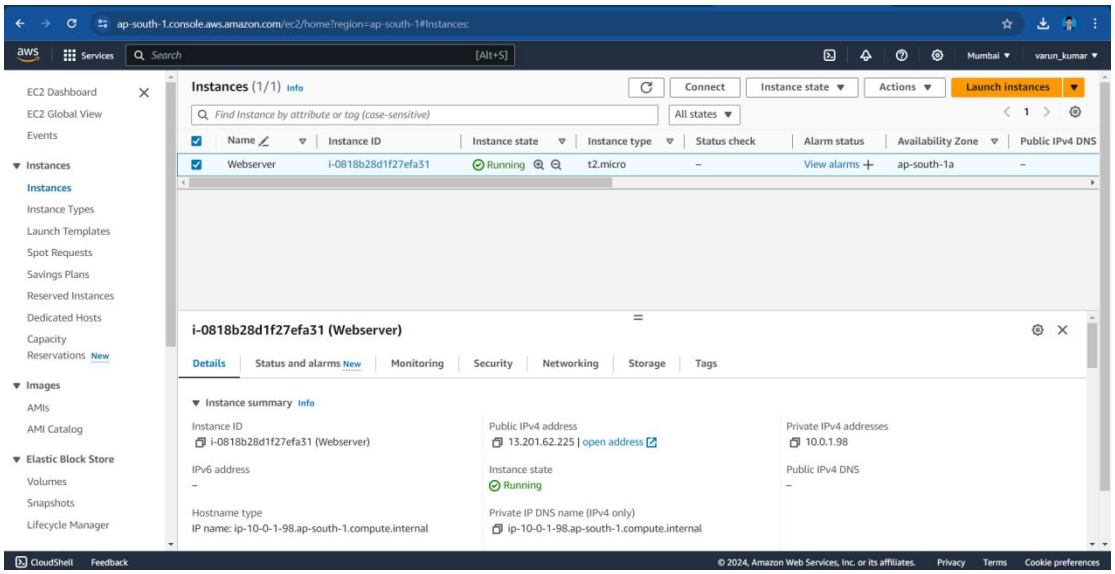
While creating the instance we will attach it to our created vpc we will add inbound security group rules for my IP and the internet and we will insert bin bash command in the user data in advance settings and click on create.









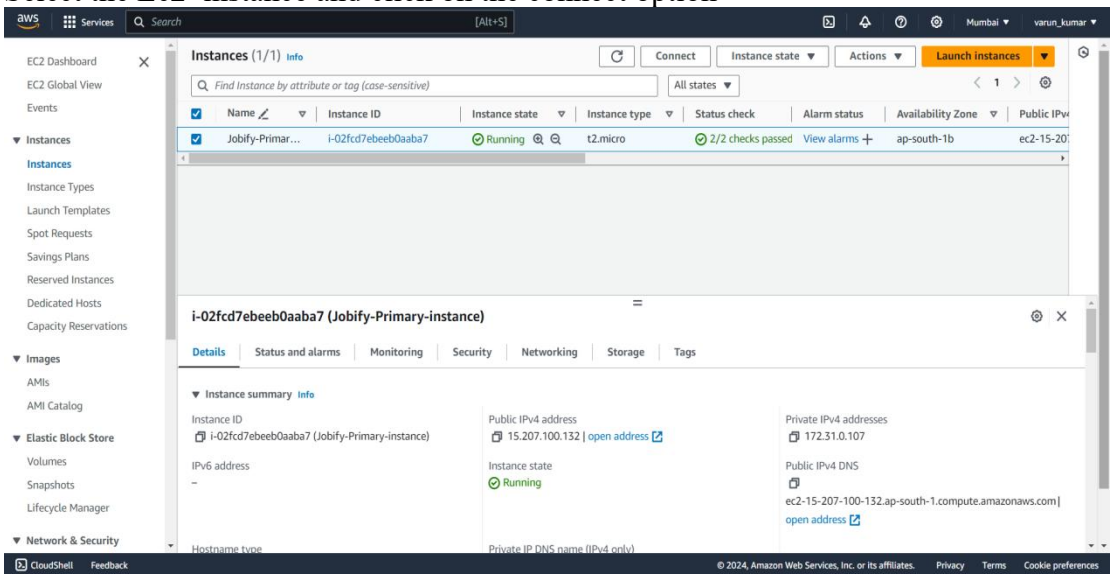


5.5.1 Checking the Static website running on instance

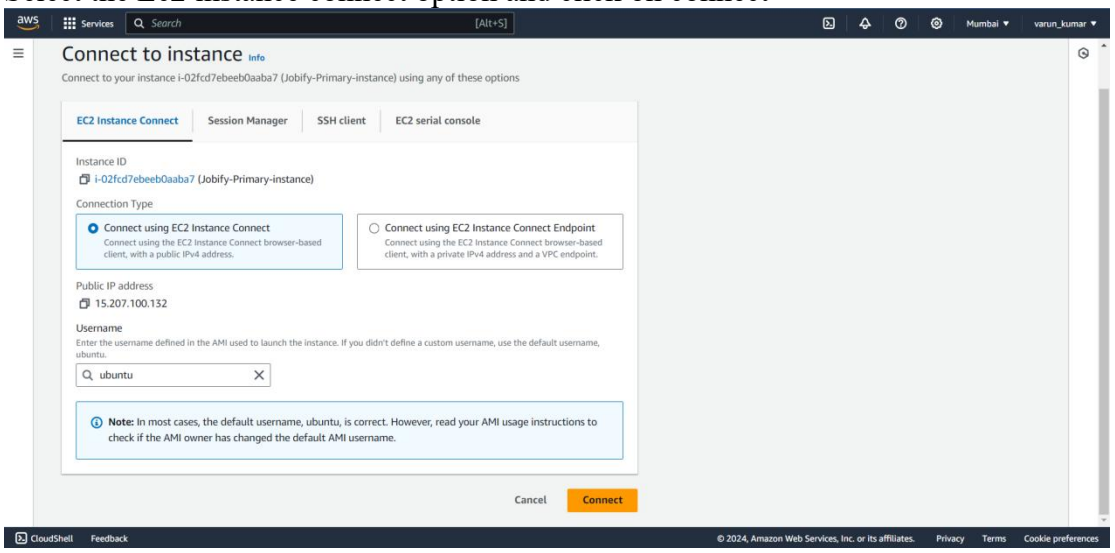
We will now check our public IPv4 address to confirm our website is running.

5.5.2 Hosting a live website on EC2 instance

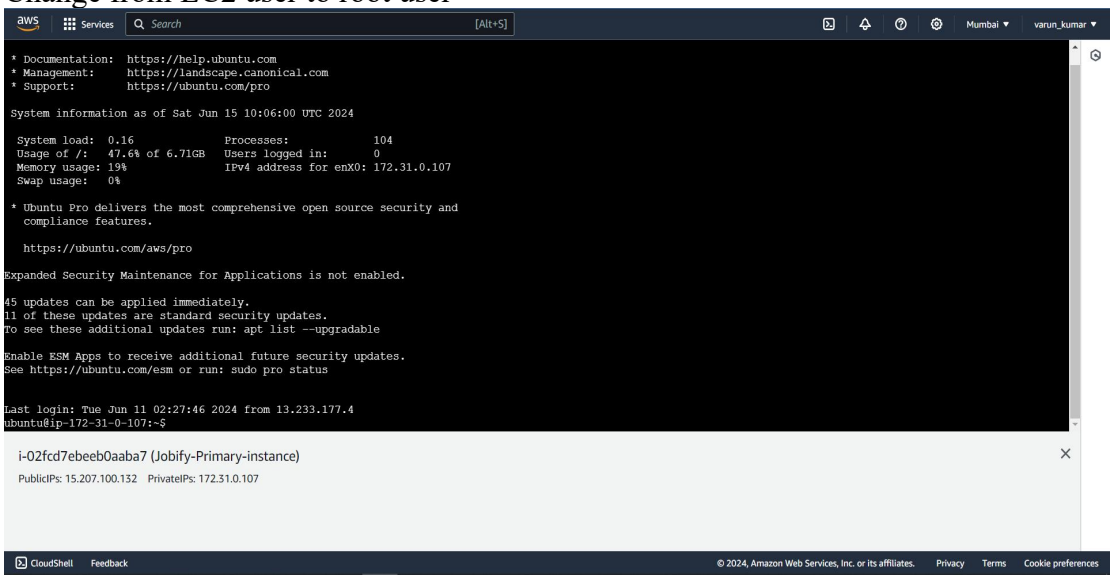
Select the Ec2 instance and click on the connect option



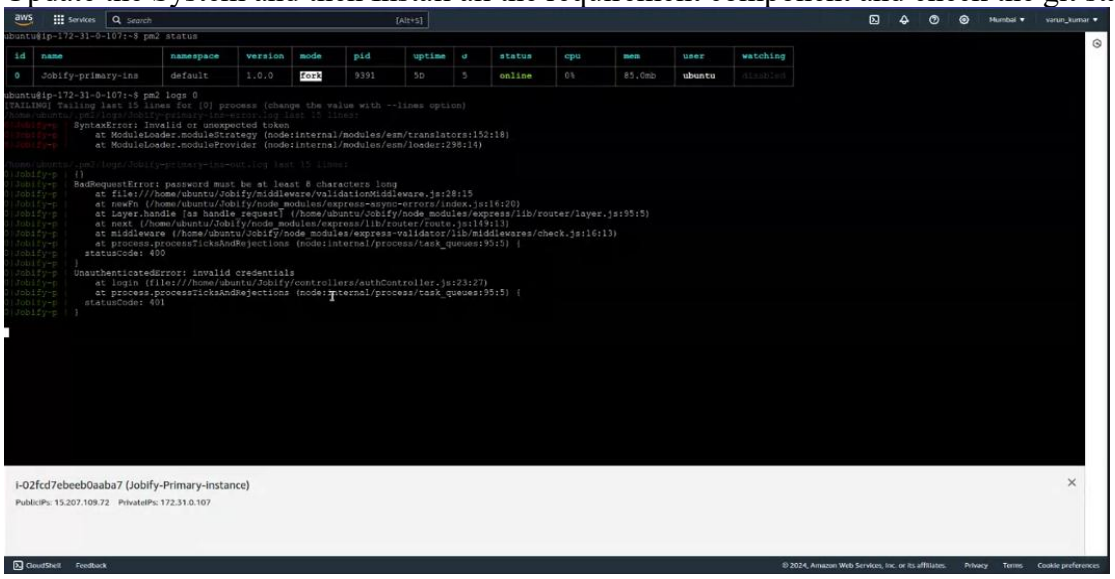
Select the Ec2 instance connect option and click on connect



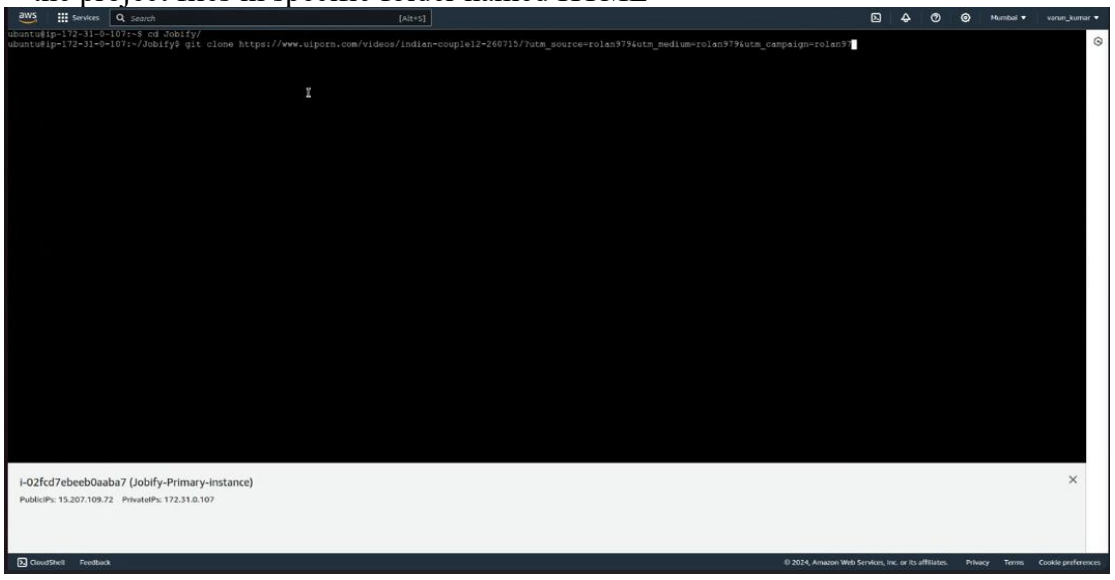
Change from EC2 user to root user



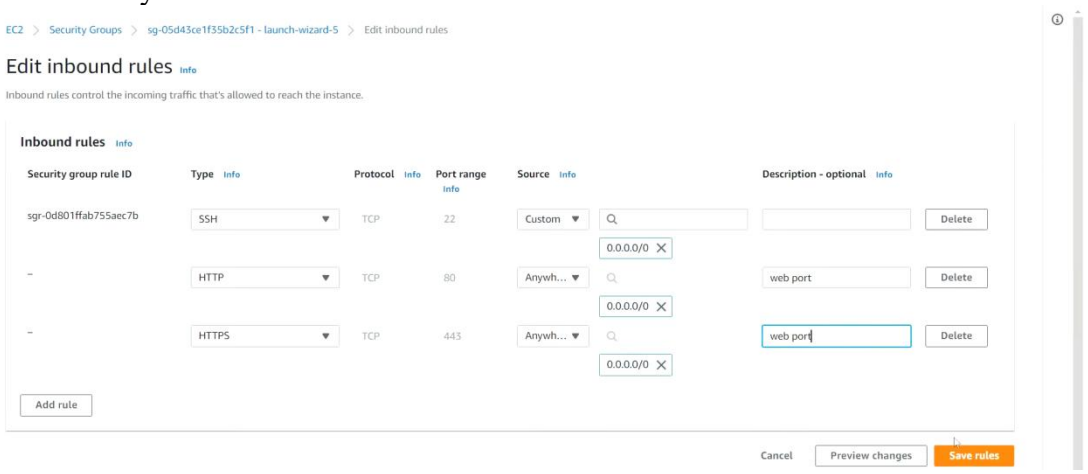
Update the System and then Install all the requirement component and check the git status.



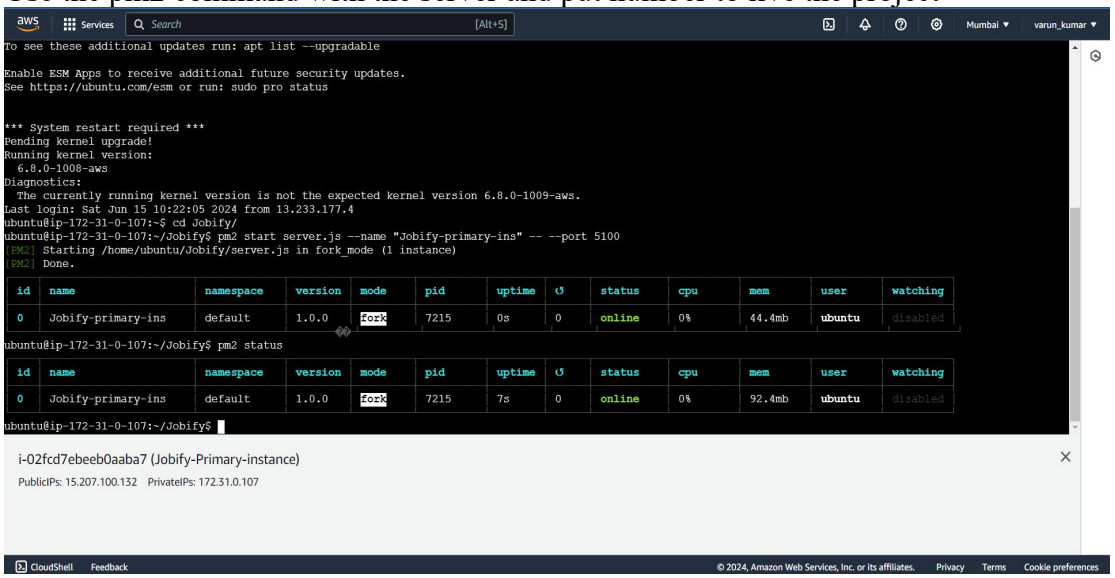
Create a directory and paste the project repository URL and then extract the project move the project files in specific folder named HTML



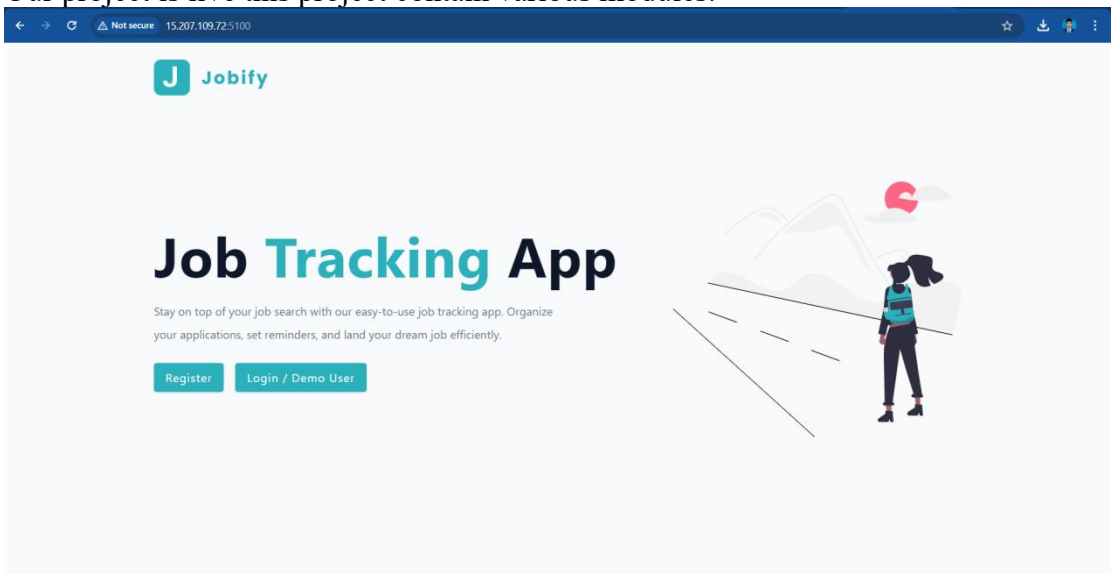
We have to change the inbound and outbound rules in the security group of the instance to allow everyone to access our website



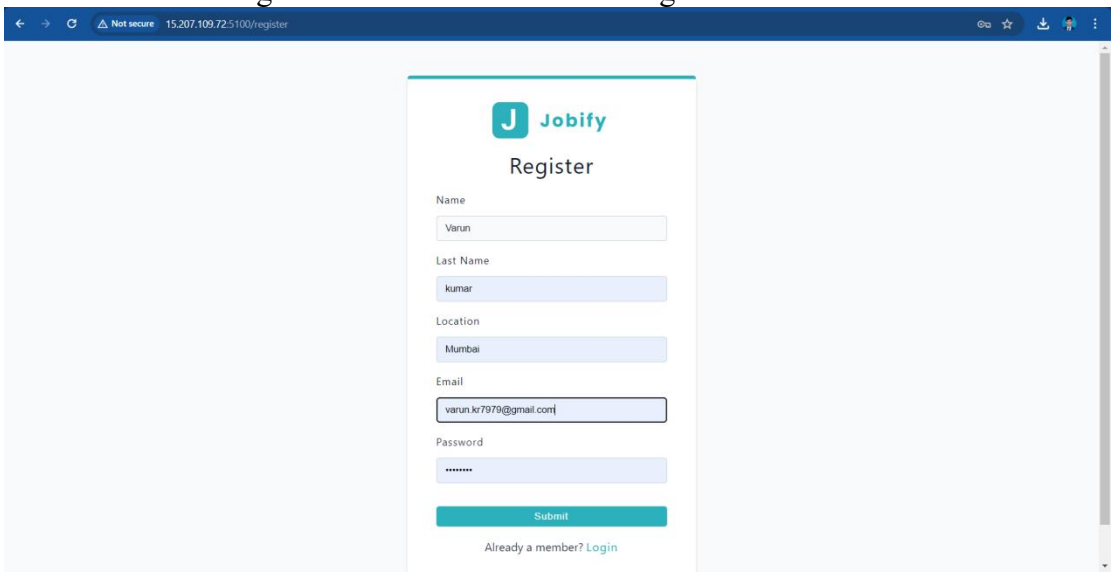
Use the pm2 command with the server and put number to live the project

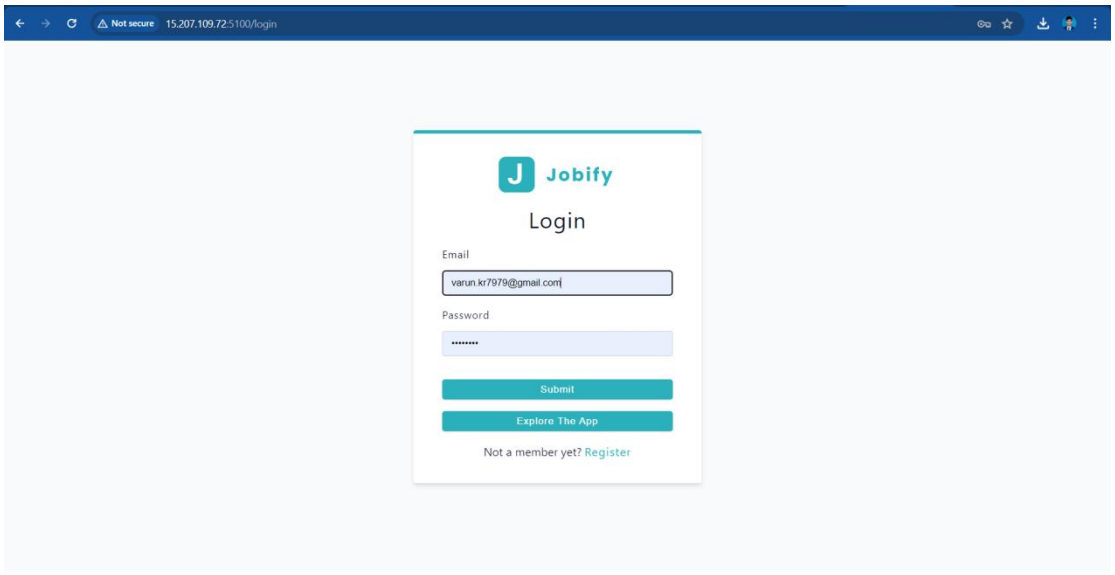


Our project is live this project contain various modules.

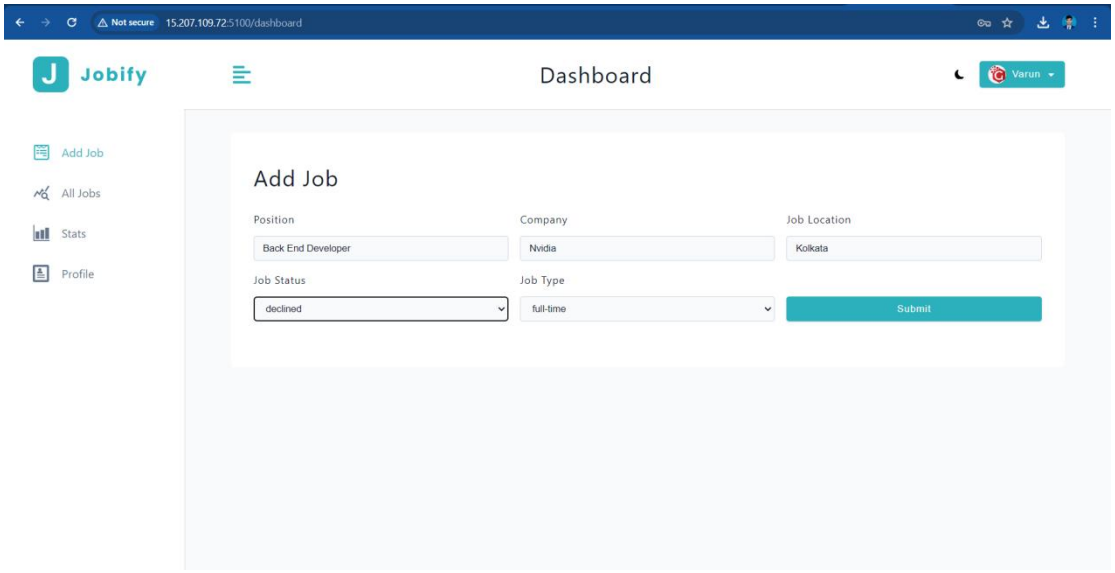


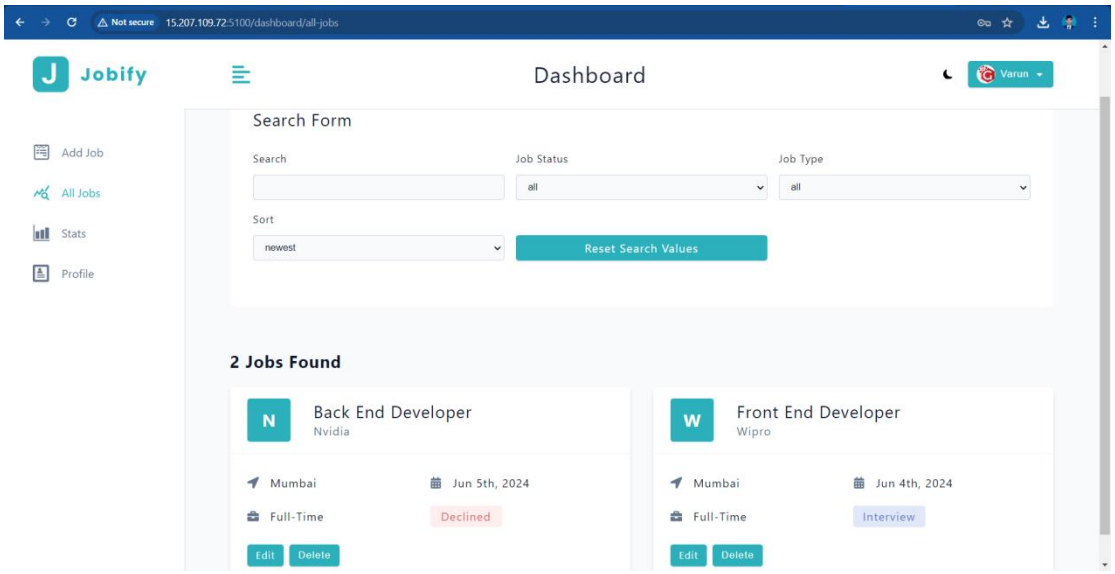
First we have the registered module and then the login module.

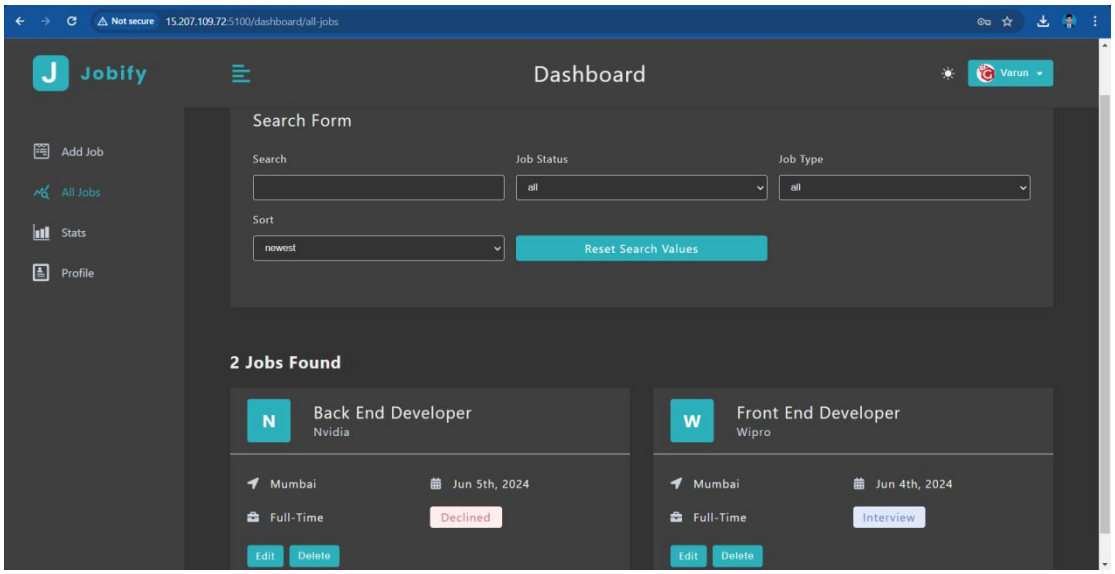


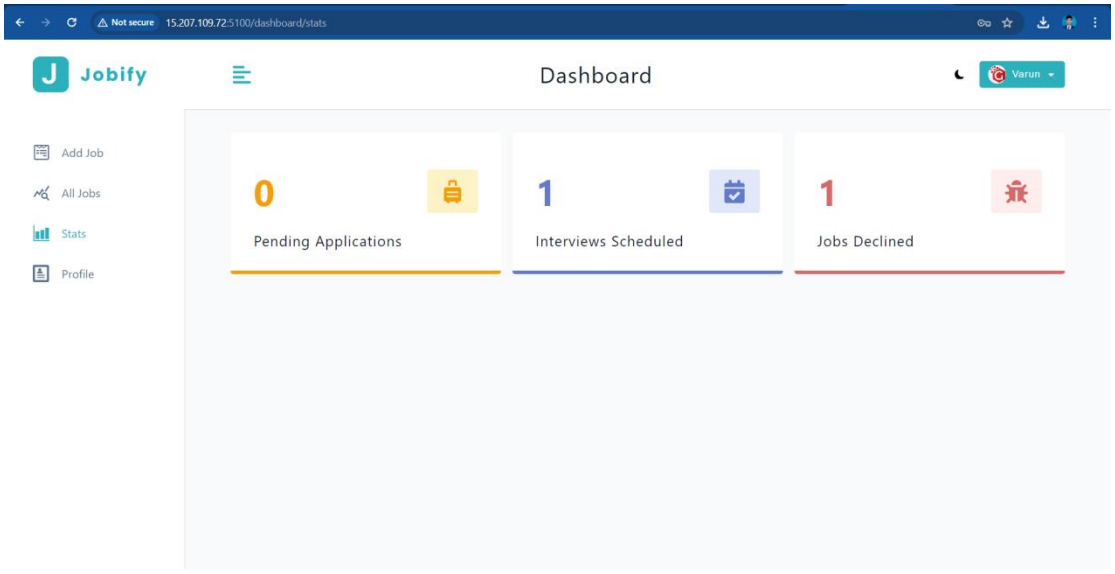


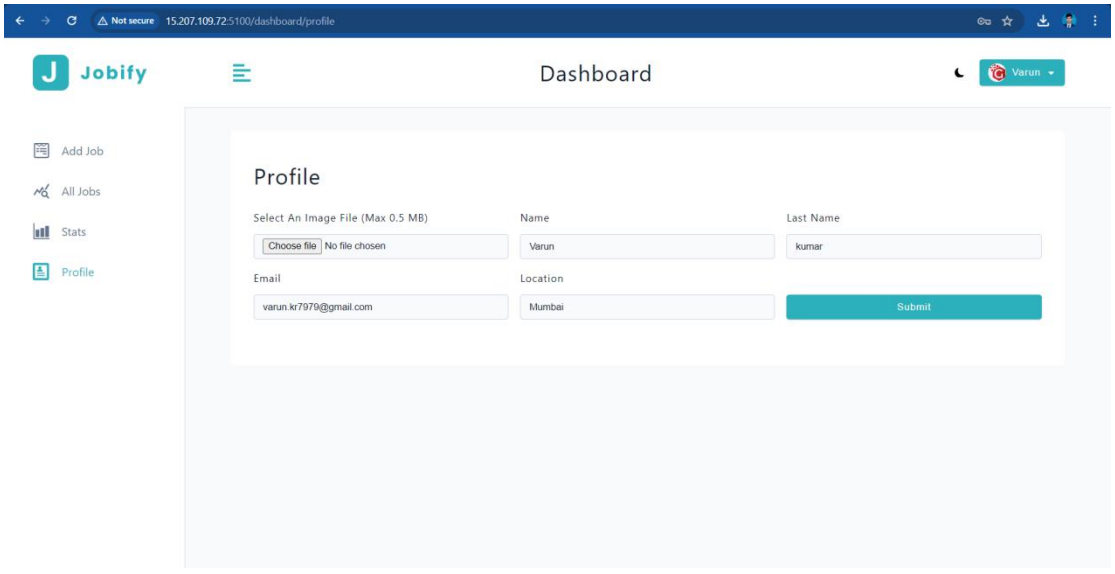
After login the website contain for more module here you can add your job profile one by one in add job module and in the all job module options you can see your all jobs in the stats module you can see the pending, interviewed and declined options and in the profile module you can save your personal information.






Not secure 15.207.109.72:5100/dashboard/all-jobs
Jobify
Dashboard
Varun
Add Job
All Jobs
Stats
Profile
Search Form
Search
Job Status
all
Job Type
all
Sort
newest
Reset Search Values
2 Jobs Found
N
Back End Developer
Nvidia
Mumbai
Jun 5th, 2024
Full-Time
Declined
Edit
Delete
W
Front End Developer
Wipro
Mumbai
Jun 4th, 2024
Full-Time
Interview
Edit
Delete

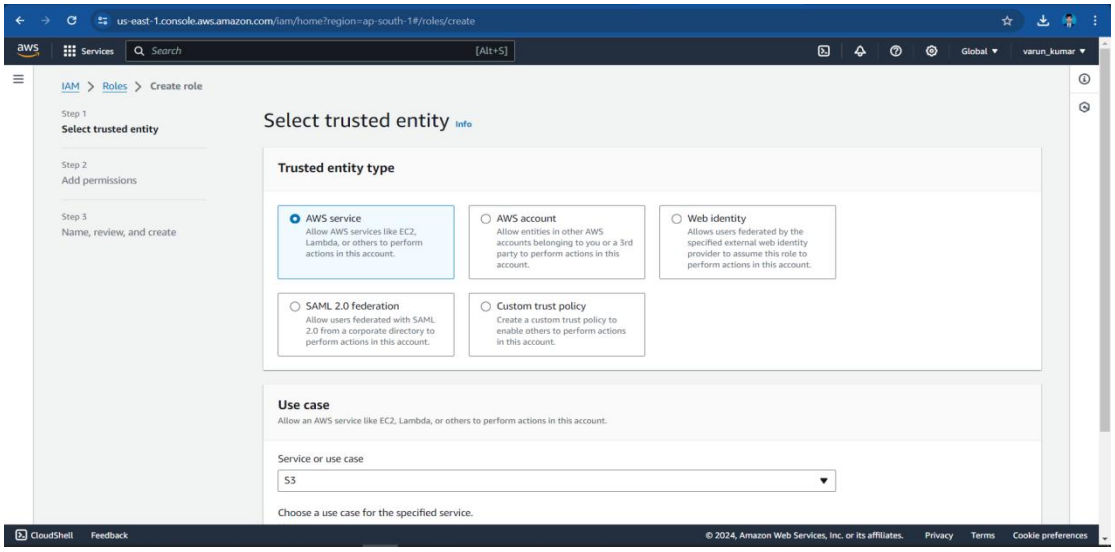

Not secure 15.207.109.72:5100/dashboard/stats
Jobify
Dashboard
Varun
Add Job
All Jobs
Stats
Profile
0
Pending Applications
1
Interviews Scheduled
1
Jobs Declined

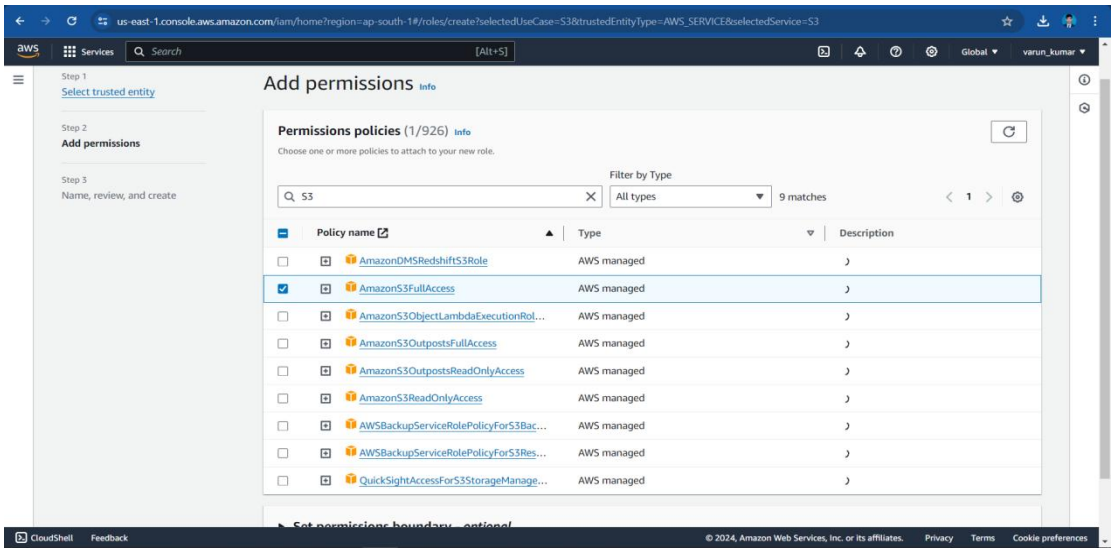

Not secure 15.207.109.72:5100/dashboard/profile
Jobify
Dashboard
Varun
Add Job
All Jobs
Stats
Profile
Profile
Select An Image File (Max 0.5 MB)
Choose file
No file chosen
Name
Varun
Last Name
kumar
Email
varun.kr7979@gmail.com
Location
Mumbai
Submit

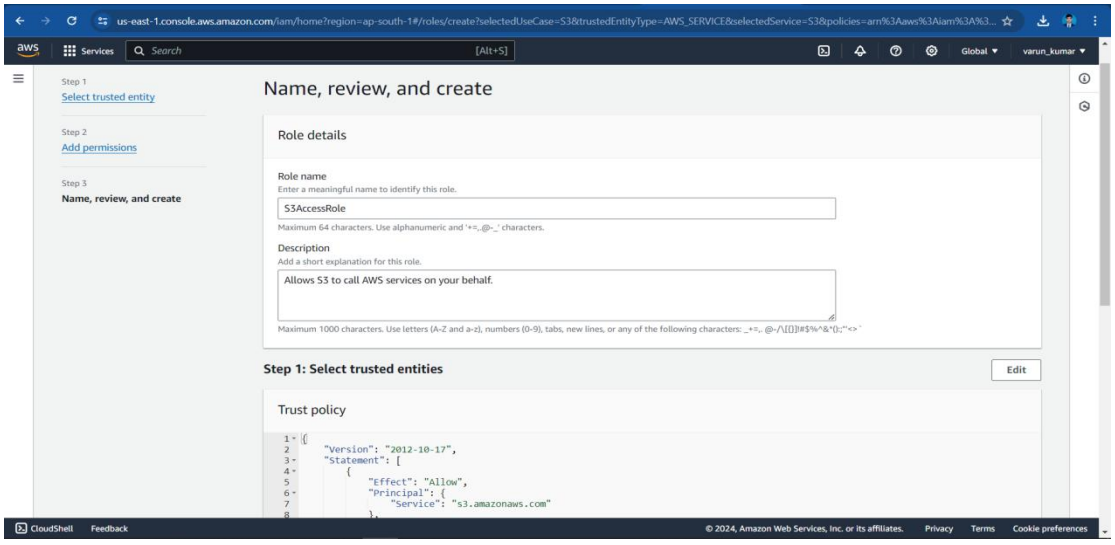
5.6 Creating an IAM role

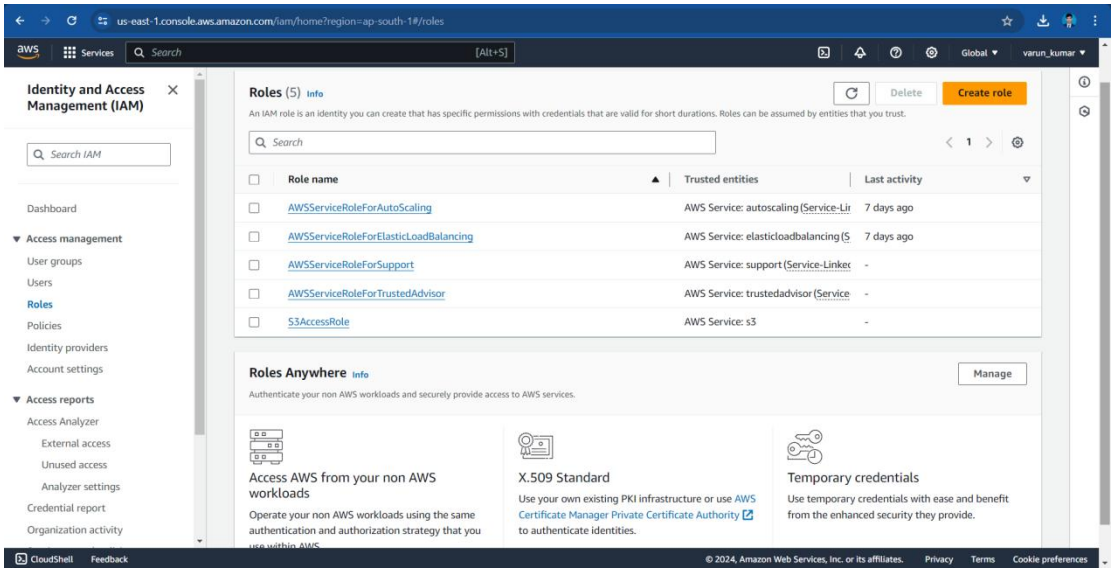
Now we will create IAM roles by going into IAM dashboard and clicking on the roles.we will click on create role.

first we have to select the trusted entity type then will will provide the Use case, then we have to provide the permission for the policies to be used and then we will provide a role name and click on create.



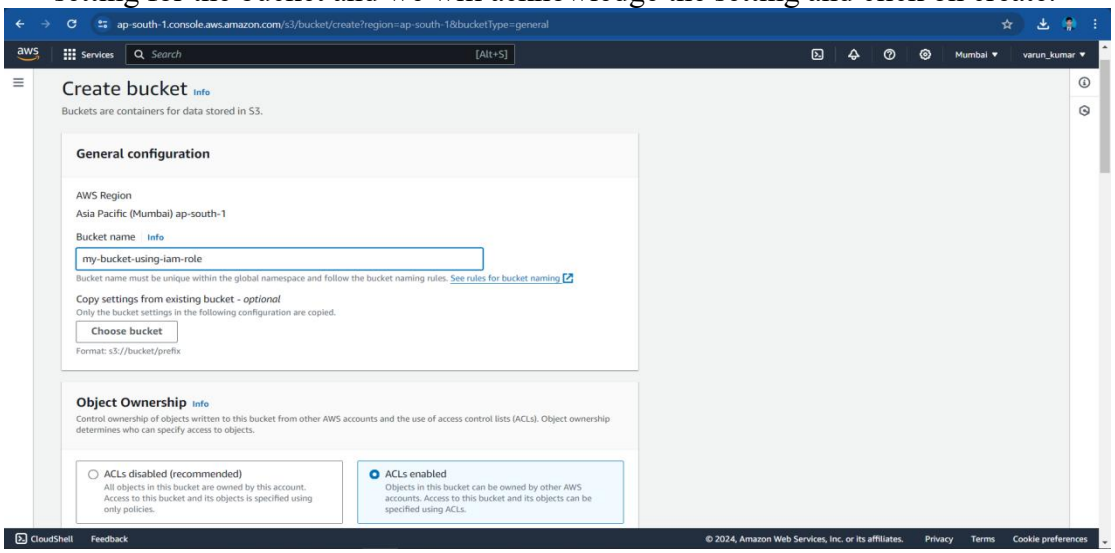


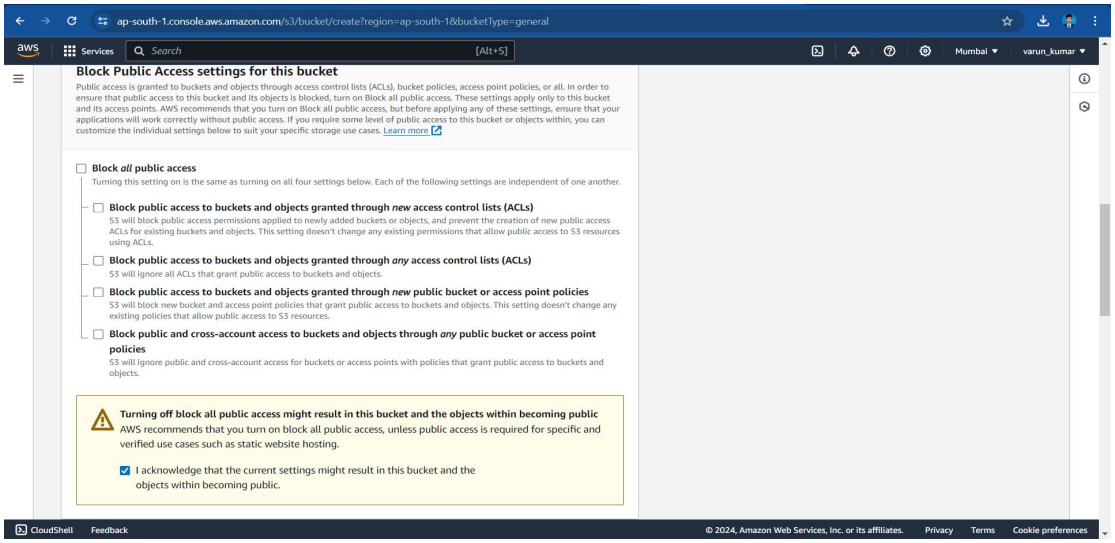




5.7 Creating a S3 bucket using an IAM role

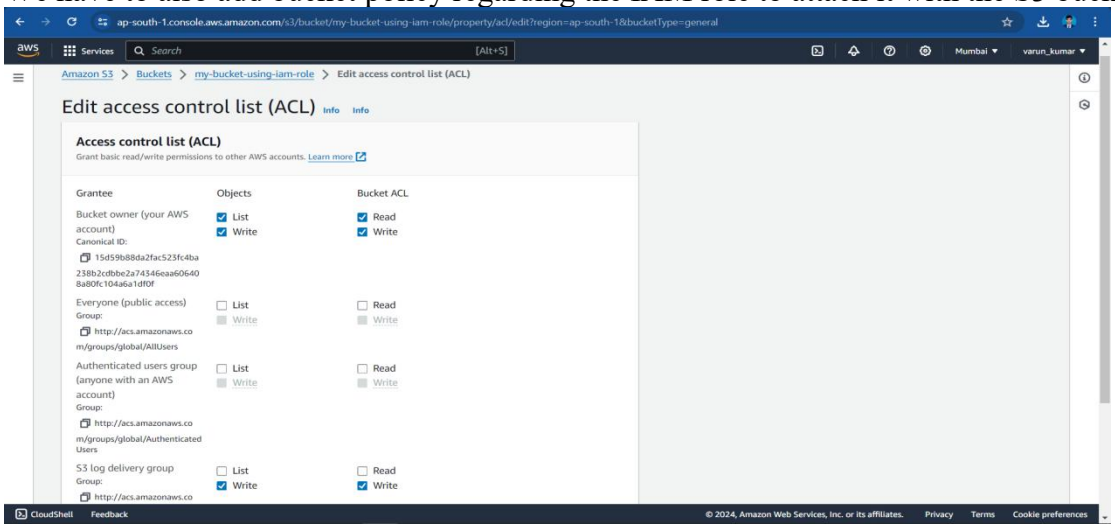
We will click on create bucket then we will provide a unique bucket name and we will select the ACLs enabled in the object ownership and we will unblock the public access setting for the bucket and we will acknowledge the setting and click on create.

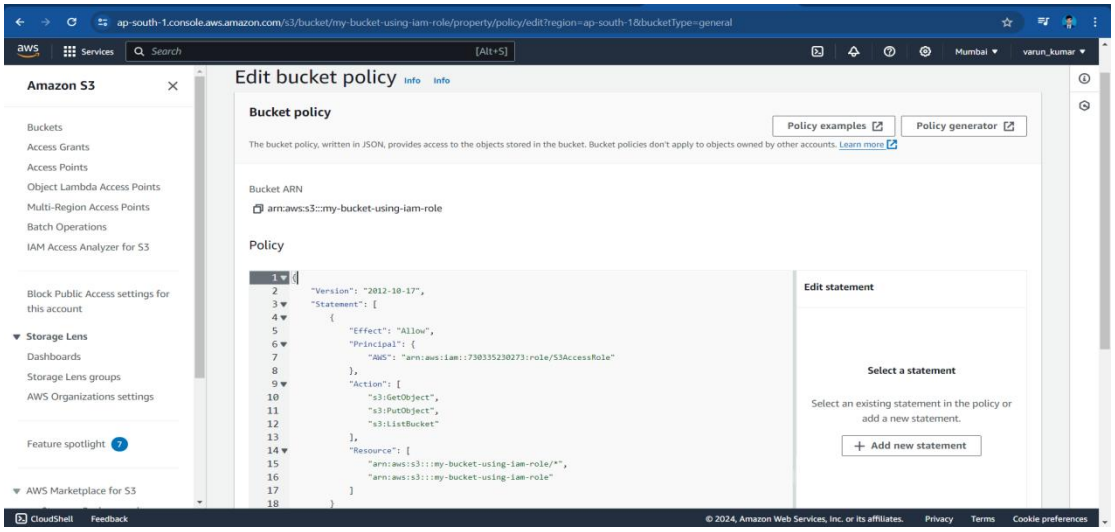




To attach the IAM role we have to edit the bucket access control list and allow the access to S3 log delivery group to write.

We have to also add bucket policy regarding the IAM role to attach it with the S3 bucket.

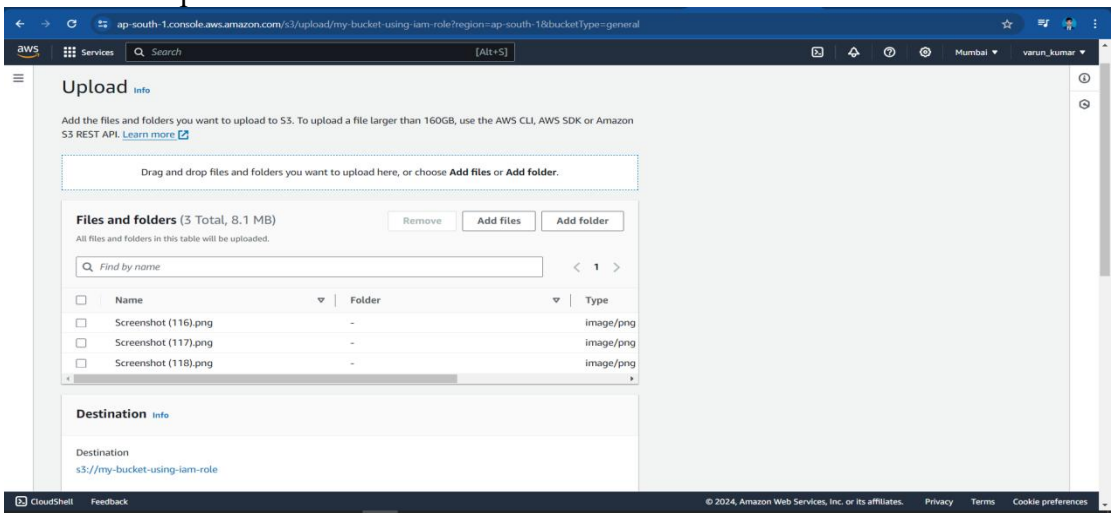


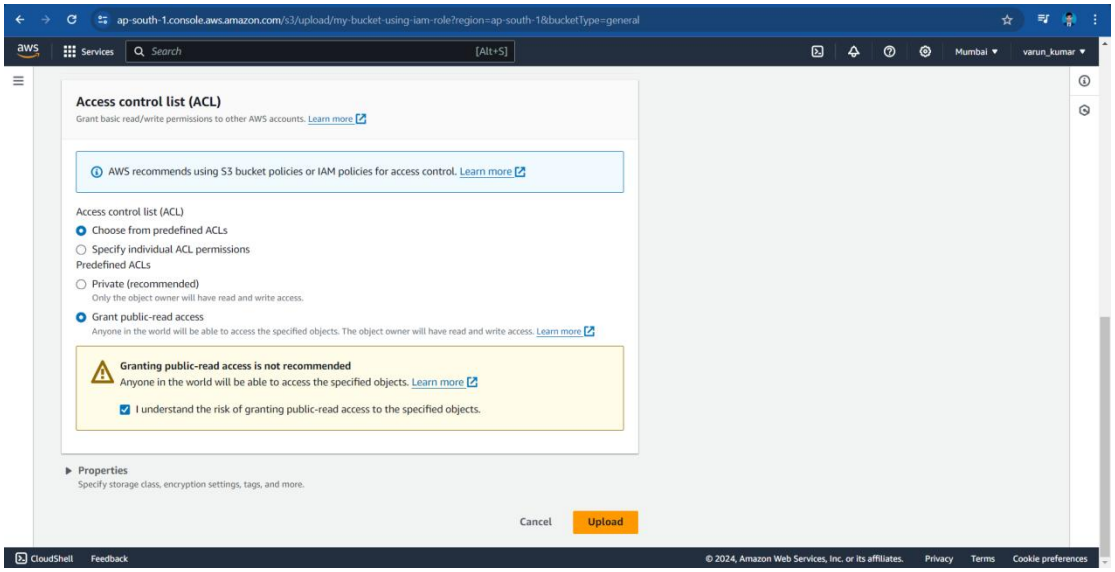


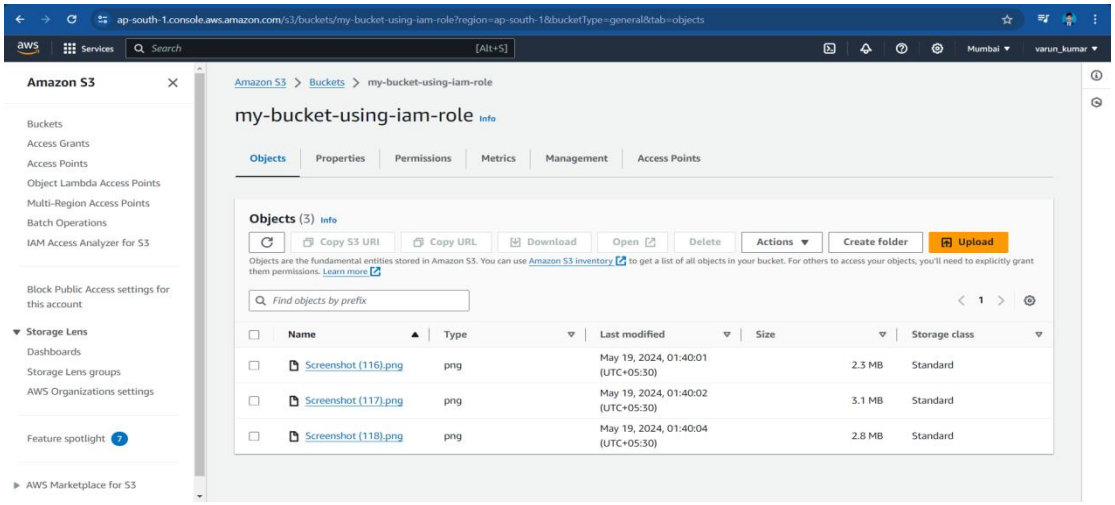
5.7.1 Uploading an object in the bucket

We will now upload the object on the bucket using the upload option.

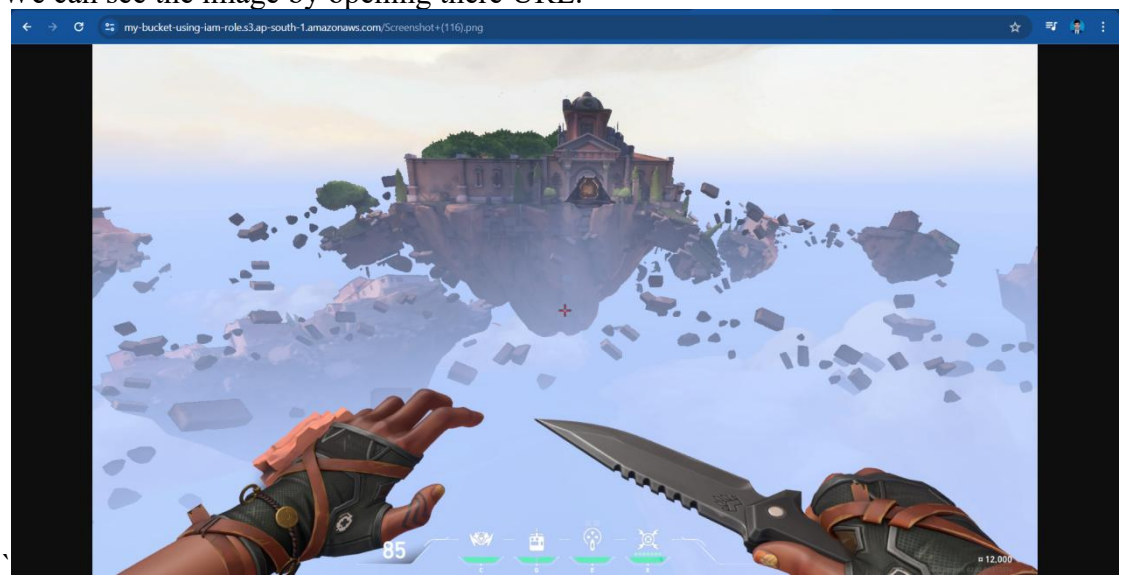
In upload option we have to first add files we want to upload then we will enabled the access control list to grant public read access and we will acknowledge the setting and click on upload.



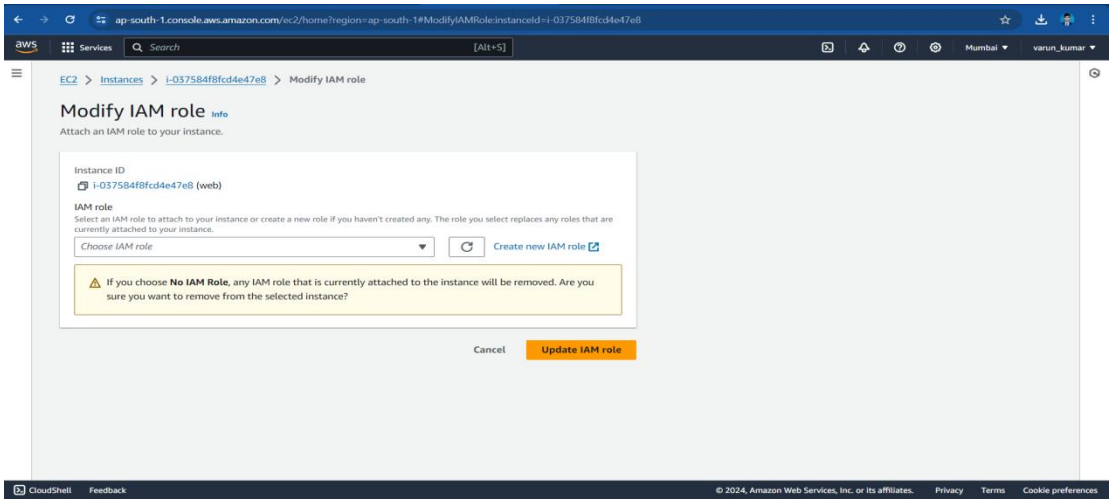




We can see the image by opening there URL.

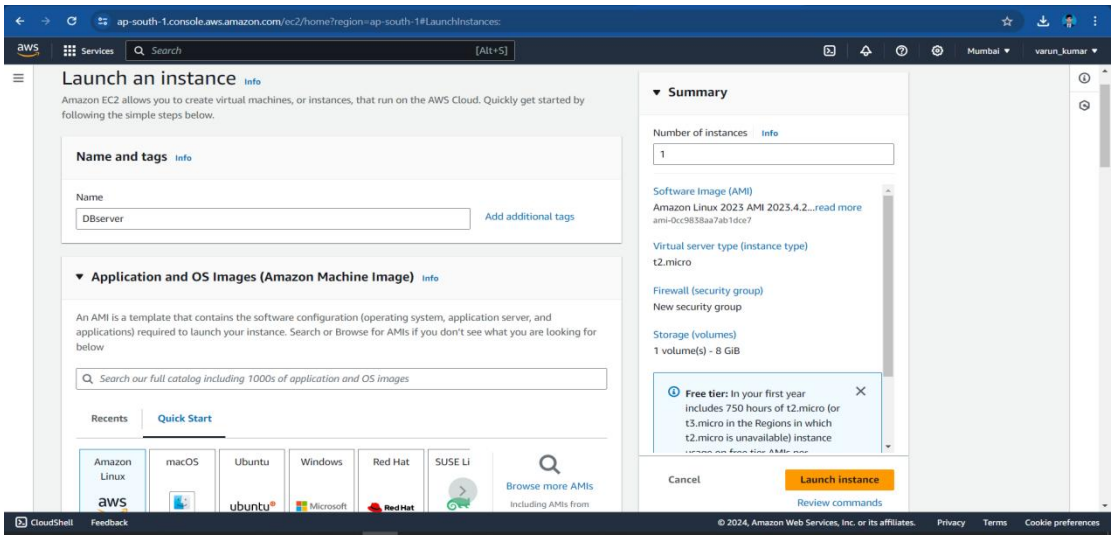


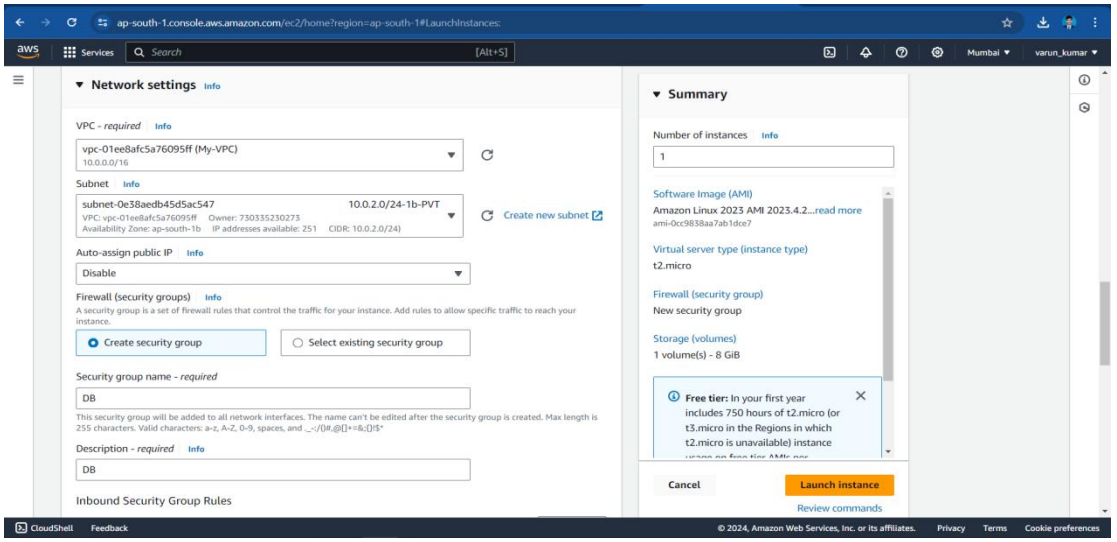
Now we will attach this IAM role with the Amazon S3 full access to our web server instance in the public subnet through the security option and we will choose to modify IAM rule and we will select the IAM role created and will will click on update to add the IAM role.

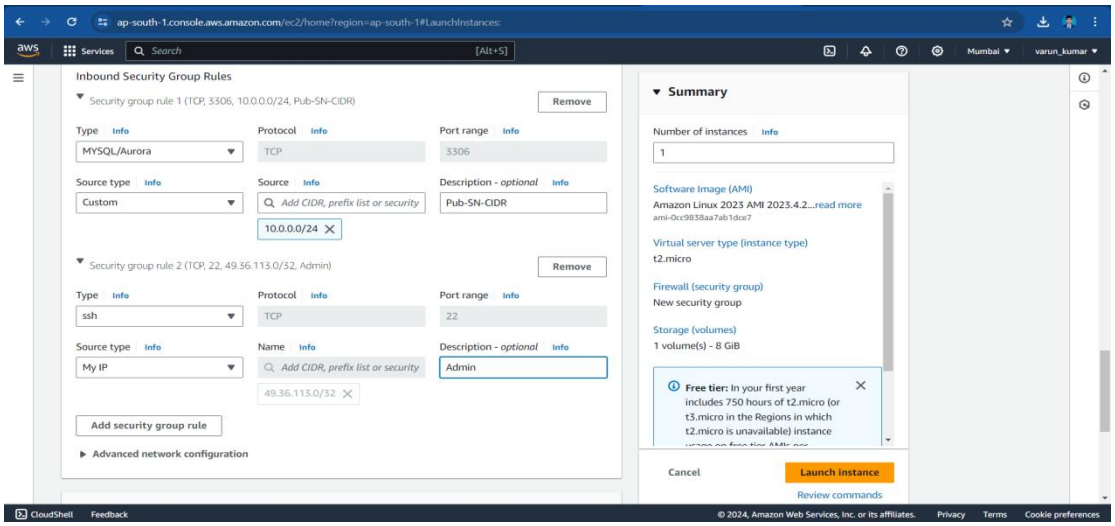


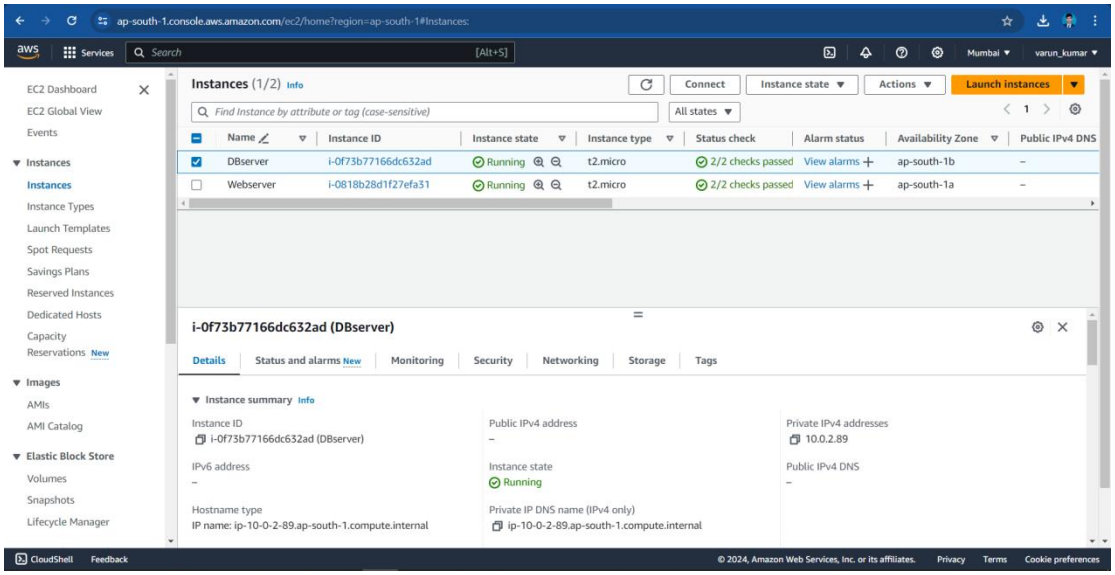
5.8 Creating Private Instance

We will now create an instance and attach it on the private subnet on this instance we will host our Database server and proceed with same configuration as public subnet instance and click on create.







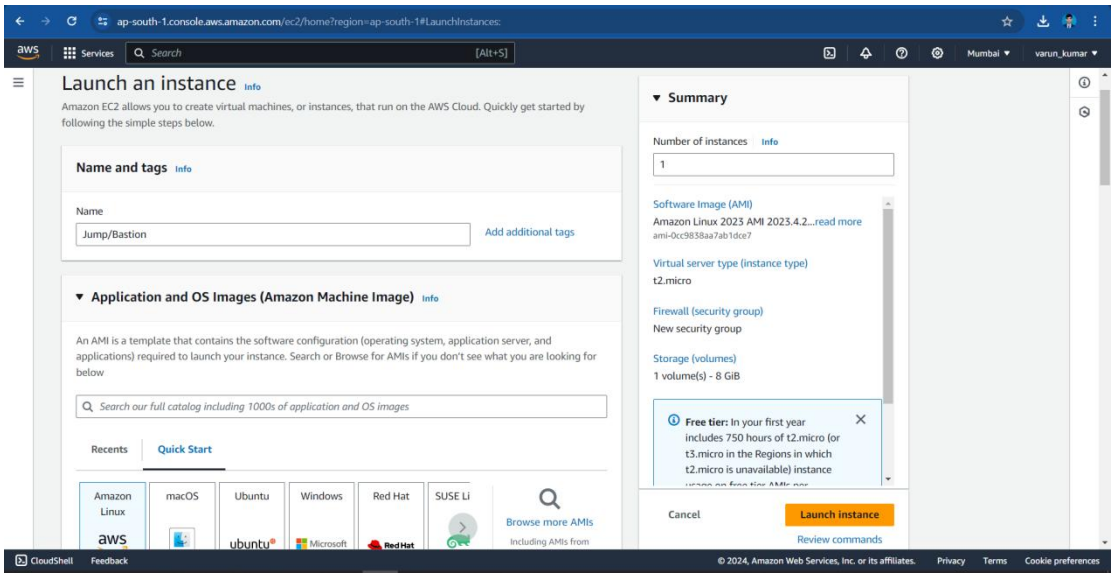


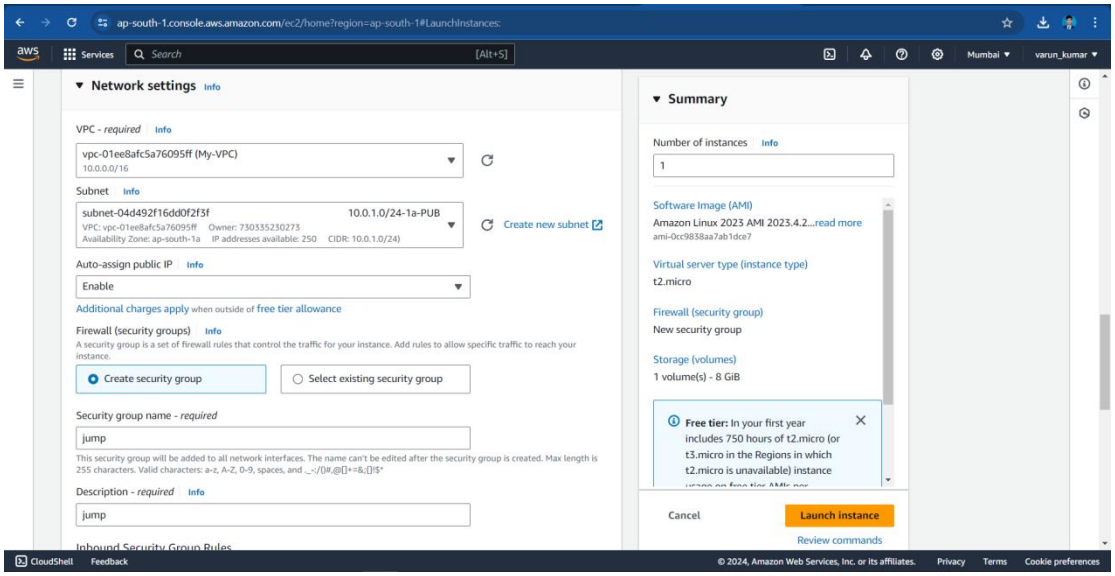
5.9 Creating Jump Bastion Instance

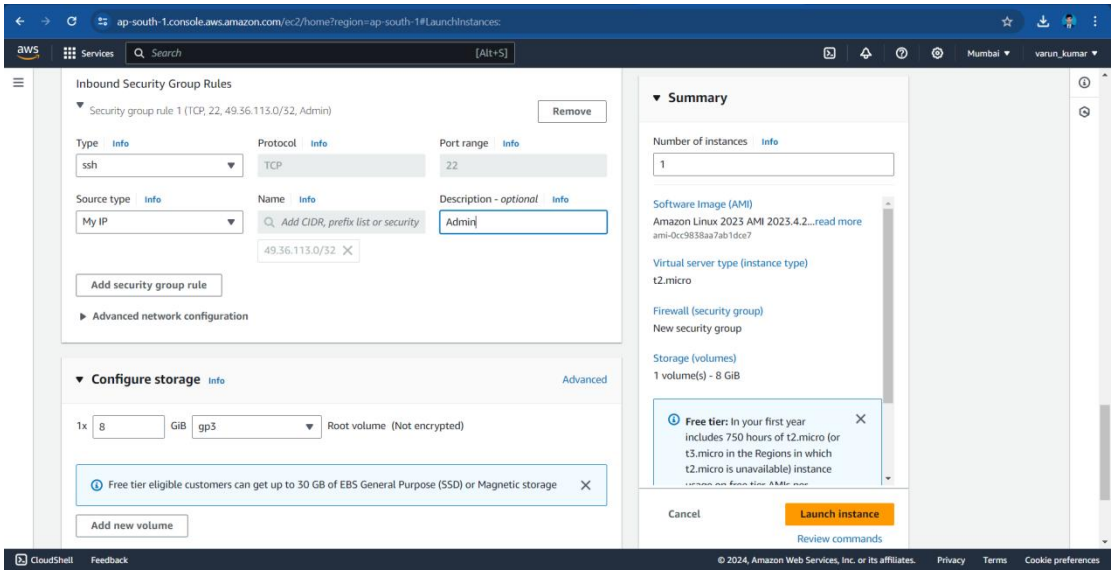
To connect a jump bastion instance to a database instance in AWS, you can follow these steps:

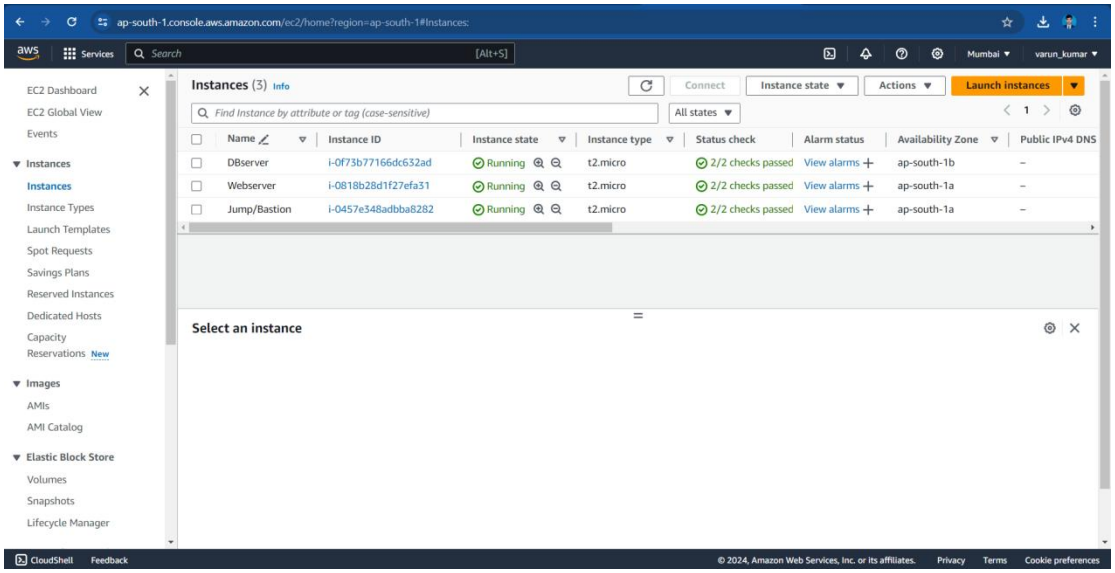
1. Launch a Jump Bastion Instance:
   - Go to the EC2 console in the AWS Management Console.
   - Click on "Launch Instances" to start the instance creation wizard.
   - Choose an appropriate Amazon Machine Image (AMI) for the jump bastion instance, such as an Amazon Linux or Ubuntu Server.
   - Select an instance type that suits your needs.
   - Configure the instance details, including network settings, security groups, and storage options.
   - Review your configuration and launch the instance.
2. Configure Security Groups:
   - While launching the instance or after the launch, configure the security groups for both the jump bastion instance and the database instance.
   - Create a new security group or choose an existing one for each instance.
   - Define inbound and outbound rules to allow the necessary network traffic.
   - For the jump bastion instance, allow SSH access (port 22) from your IP address or a specific range of IP addresses.
   - For the database instance, allow the required ports for database connectivity (e.g., port 3306 for MySQL, port 5432 for PostgreSQL).

3. Connect to the Jump Bastion Instance:
   - Retrieve the public IP address or DNS name of the jump bastion instance from the EC2 console.
   - Use an SSH client (e.g., Terminal on macOS/Linux, PuTTY on Windows) to connect to the jump bastion instance.
   - Provide the necessary credentials (SSH key pair or username/password) to establish the connection.
4. Connect to the Database Instance:
   - From the jump bastion instance, you can establish a secure connection to the database instance.
   - Use the appropriate client or command-line tool for your database (e.g., MySQL Workbench for MySQL, PostgreSQL).
   - Provide the necessary connection details such as the database host, port, username, and password.
   - Connect to the database instance.

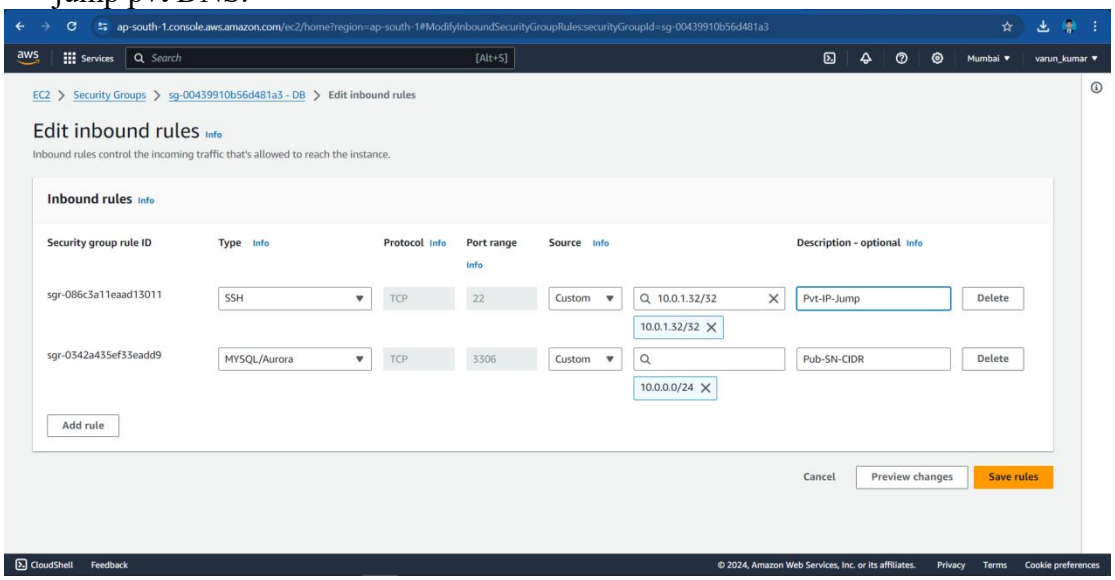


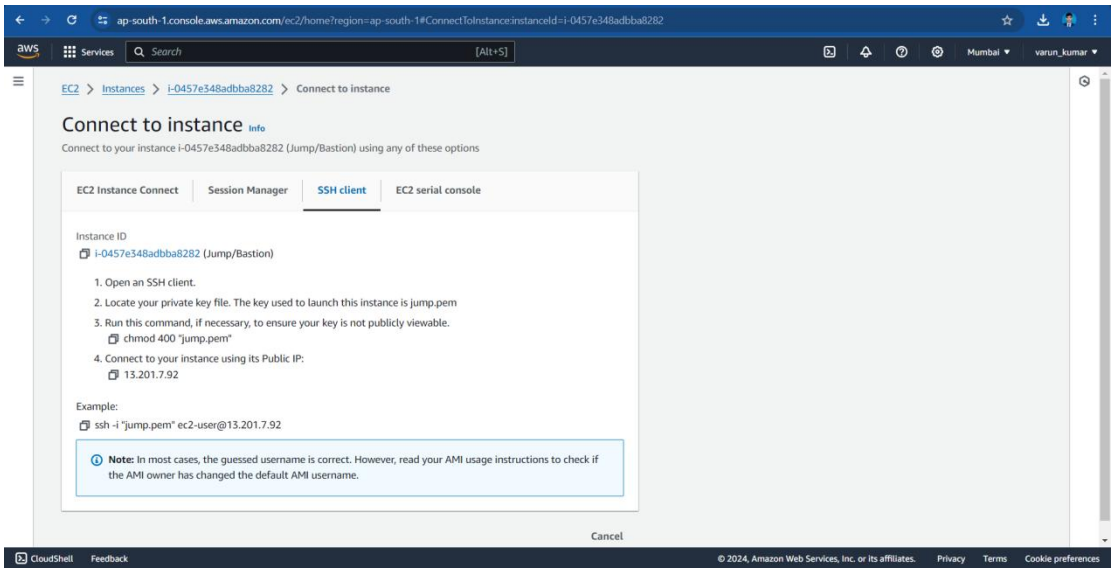


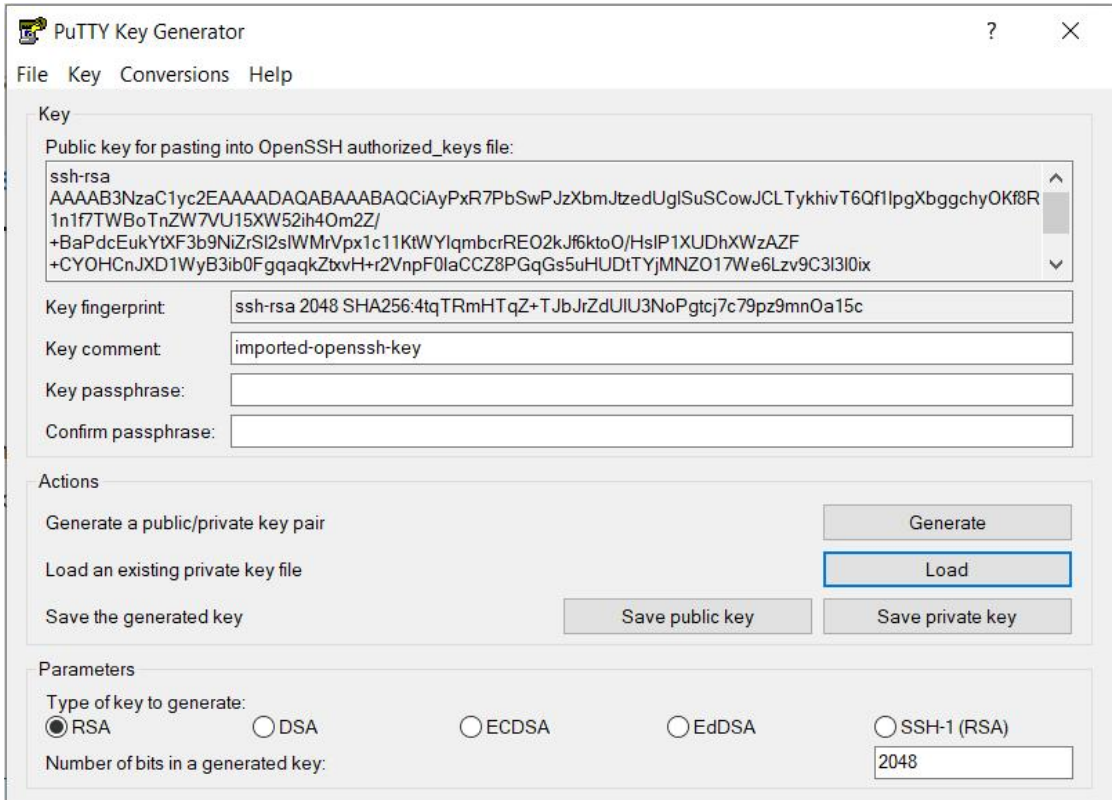


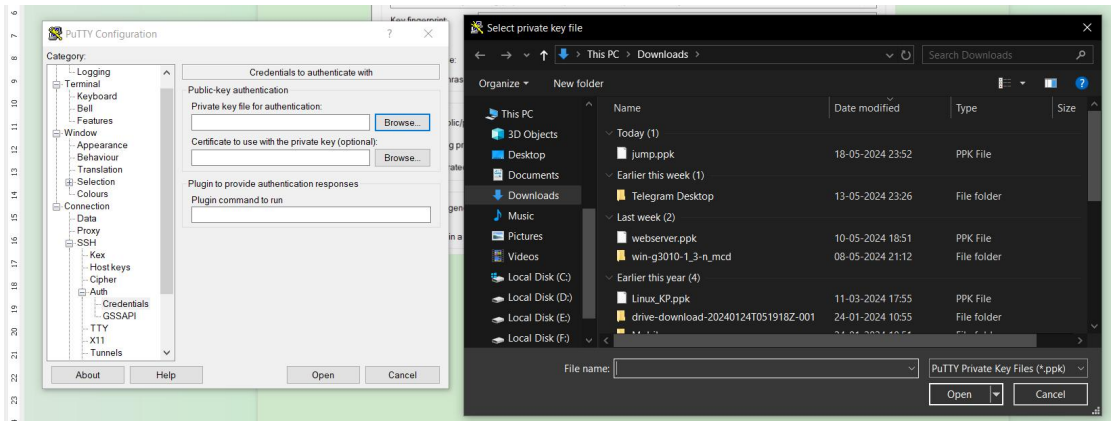


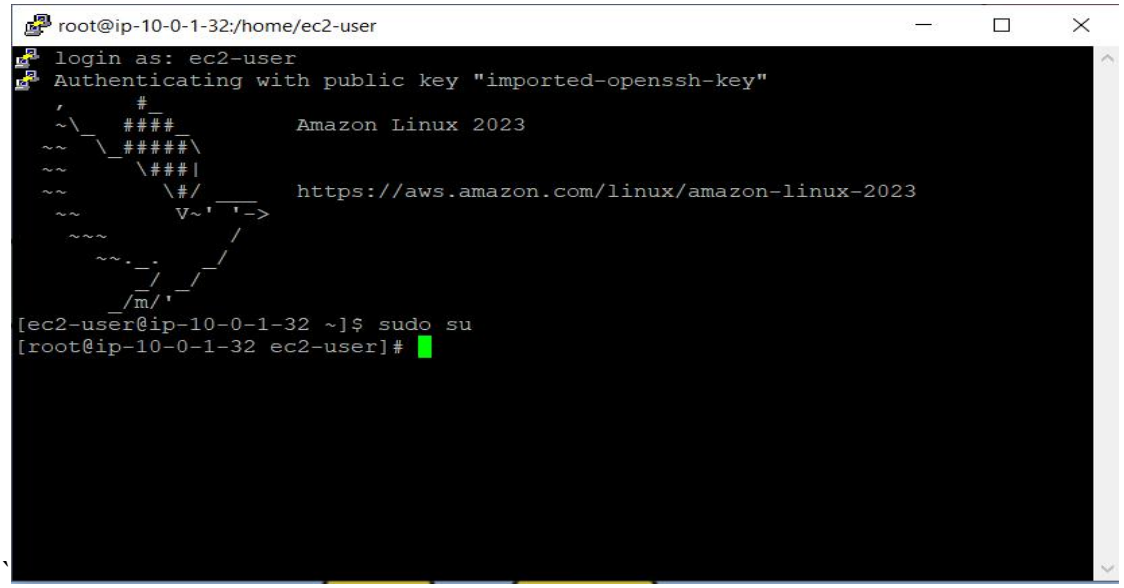
We copied the pvt DNS of jump to the Db security inbound rules and edit the ssh source to jump pvt DNS.



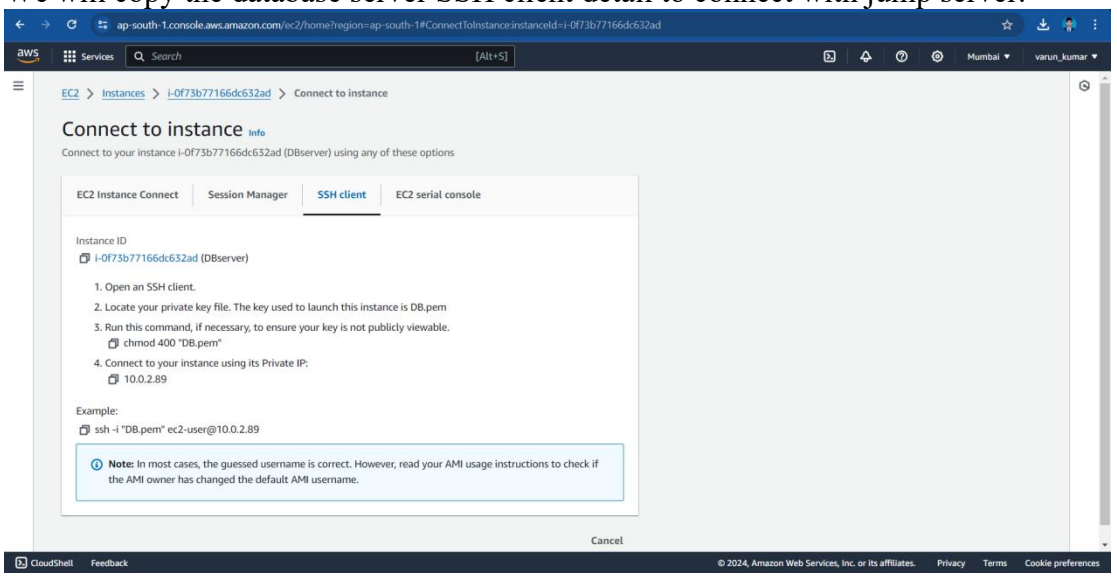

ap-south-1.console.aws.amazon.com/ec2/home?region=ap-south-1#ConnectToInstance:instanceId=i-0457e348adbba8282
Services
Search
[Alt+S]
Mumbai
varun_kumar
EC2 > Instances > i-0457e348adbba8282 > Connect to instance
Connect to instance Info
Connect to your instance i-0457e348adbba8282 (Jump/Bastion) using any of these options
EC2 Instance Connect
Session Manager
SSH client
EC2 serial console
Instance ID
i-0457e348adbba8282 (Jump/Bastion)
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is jump.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
chmod 400 "jump.pem"
4. Connect to your instance using its Public IP:
13.201.7.92
Example:
ssh -i "jump.pem" ec2-user@13.201.7.92
Note: In most cases, the guessed username is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username.
Cancel
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


PuTTY Key Generator
File Key Conversions Help
Key
Public key for pasting into OpenSSH authorized_keys file:
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQCiAyPxR7PbSwPJzXbmJtzedUglSuSCowJCLTykhivT6Qf1IpgXbggchyOKf8R
1n1f7TWBoTnZW7VU15XW52ih4Om2Z/
+BaPdcEukYtXF3b9NiZrSl2slWMrVpx1c11KtWYlqmbcrREO2kJf6ktoO/HslP1XUDhXWzAZF
+CYOHCnJXD1WyB3ib0FgqaqkZtxvH+r2VnpF0laCCZ8PGqGs5uHUDtTYjMNZO17We6Lzv9C3l3l0ix
Key fingerprint:
ssh-rsa 2048 SHA256:4tqTRmHTqZ+TJbJrZdUlU3NoPgtcj7c79pz9mnOa15c
Key comment:
imported-openssh-key
Key passphrase:
Confirm passphrase:
Actions
Generate a public/private key pair
Generate
Load an existing private key file
Load
Save the generated key
Save public key
Save private key
Parameters
Type of key to generate:
RSA
DSA
ECDSA
EdDSA
SSH-1 (RSA)
Number of bits in a generated key:
2048

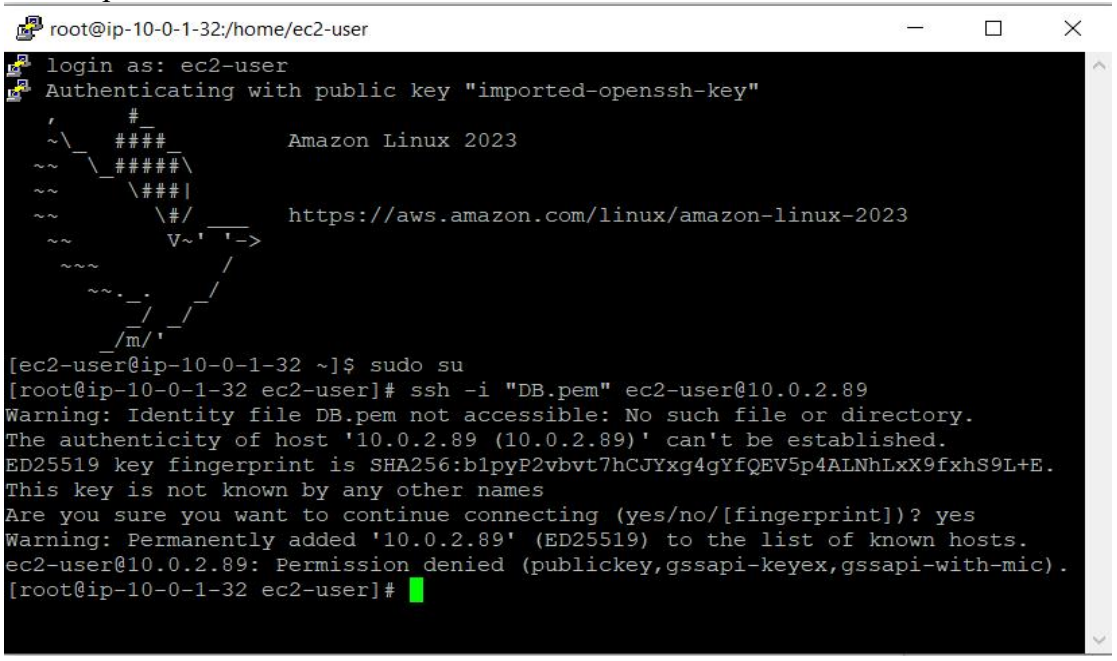

PuTTY Configuration
Category:
Logging
Terminal
Keyboard
Bell
Features
Window
Appearance
Behaviour
Translation
Selection
Colours
Connection
Data
Proxy
SSH
Kex
Host keys
Cipher
Auth
Credentials
GSSAPI
TTY
X11
Tunnels
Credentials to authenticate with
Public-key authentication
Private key file for authentication:
Browse...
Certificate to use with the private key (optional):
Browse...
Plugin to provide authentication responses
Plugin command to run
About
Help
Open
Cancel
Select private key file
This PC > Downloads
Search Downloads
Organize
New folder
This PC
3D Objects
Desktop
Documents
Downloads
Music
Pictures
Videos
Local Disk (C:)
Local Disk (D:)
Local Disk (E:)
Local Disk (F:)
Name
Date modified
Type
Size
Today (1)
jump.ppk
18-05-2024 23:52
PPK File
Earlier this week (1)
Telegram Desktop
13-05-2024 23:26
File folder
Last week (2)
webserver.ppk
10-05-2024 18:51
PPK File
win-g3010-1_3-n_mcd
08-05-2024 21:12
File folder
Earlier this year (4)
Linux_KP.ppk
11-03-2024 17:55
PPK File
drive-download-20240124T051918Z-001
24-01-2024 10:55
File folder
File name:
PuTTY Private Key Files (*.ppk)
Open
Cancel



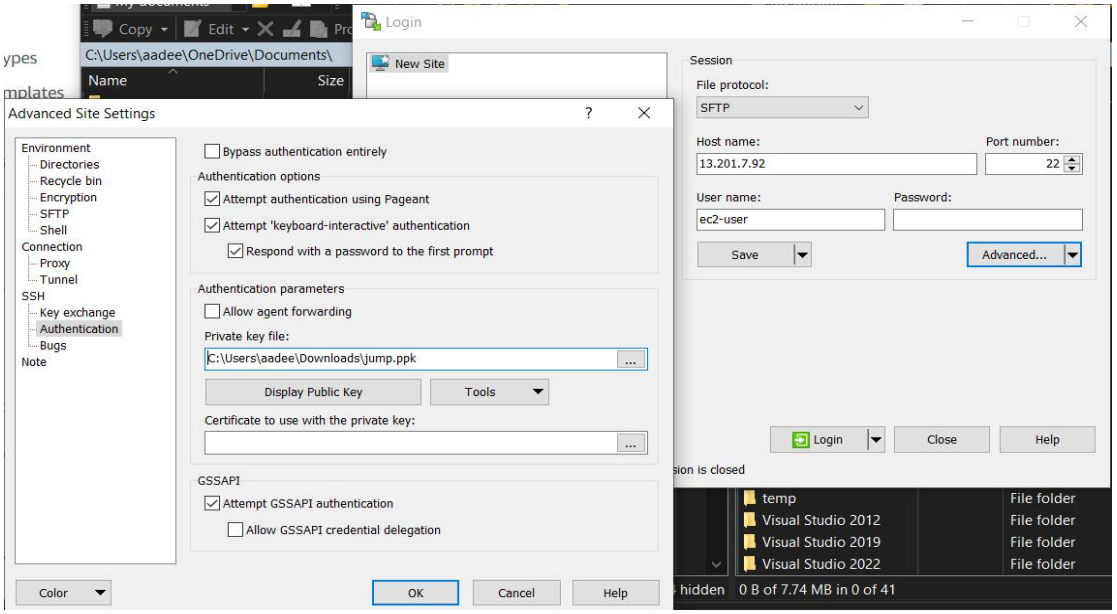
We will copy the database server SSH client detail to connect with jump server.

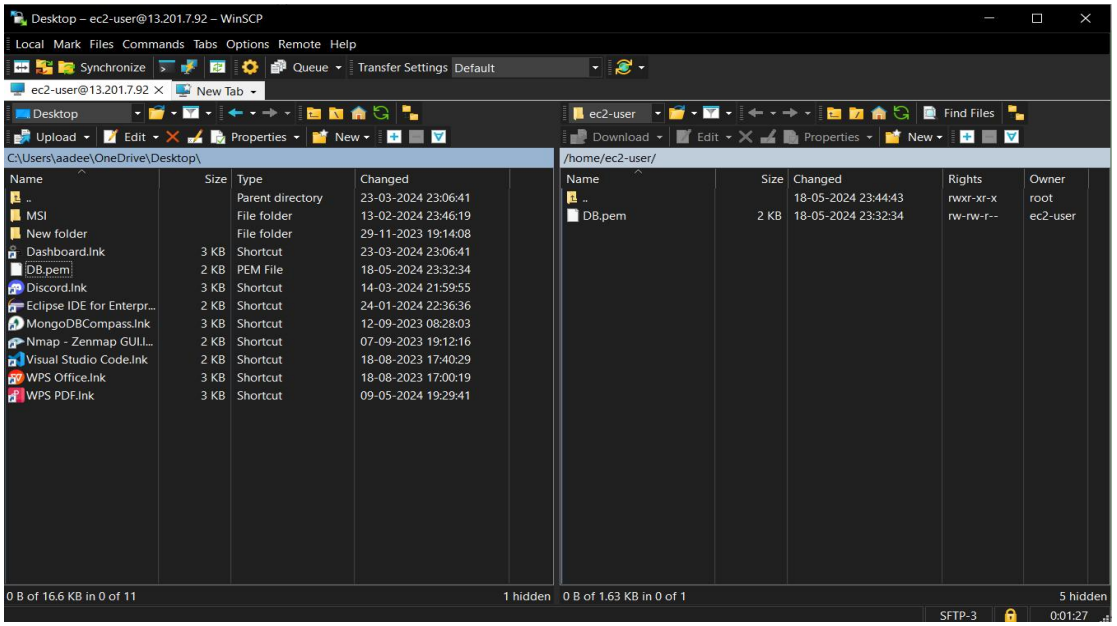


As the .pem file of the database server is not in the root folder the access will be denied.

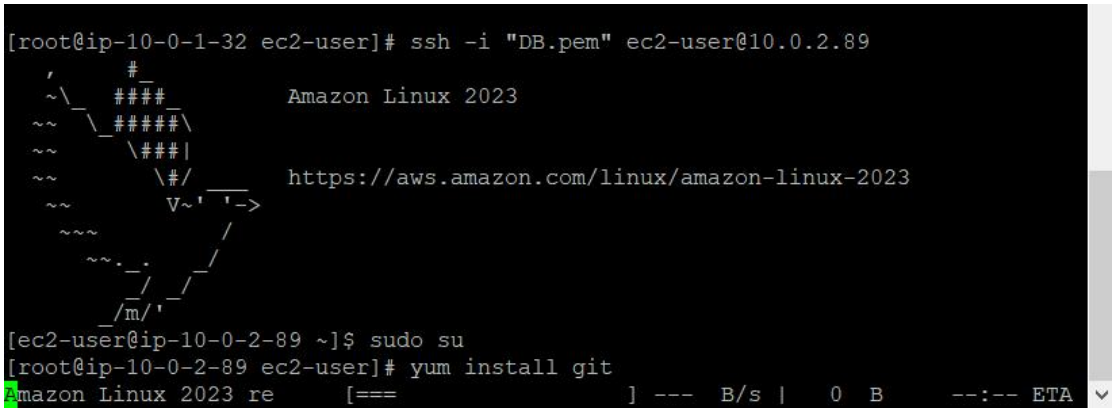


We will use winSCP application to to move the .pem file to the root folder. We will provide our host name and username with password to login in the application and do our task.





The connection with the database server has been made with the jump server but we can see there is no internet connection in the database server.



5.10 Creating ELB AND ASG

Create an Elastic Load Balancer:

- Go to the EC2 console.
- Click on "Load Balancers" in the sidebar.
- Click on "Create Load Balancer" and select the type of load balancer you want to create (e.g., Application Load Balancer, Network Load Balancer, or Classic Load Balancer).
- Configure the load balancer's settings, such as listeners, target groups, security groups, and availability zones.
- Save your load balancer.

Create an Auto Scaling Group:

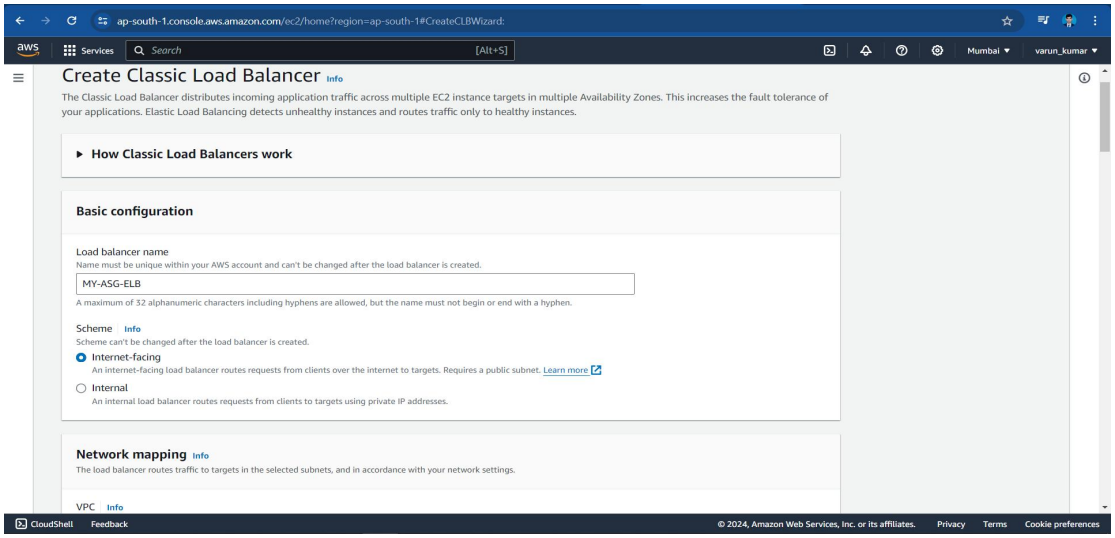
- In the EC2 Auto Scaling console, click on "Auto Scaling Groups" in the sidebar.
- Click on "Create an Auto Scaling group" and follow the wizard.
- Select the launch configuration you created in the previous step.
- Configure the desired minimum, maximum, and desired capacity of instances.
- Configure any additional settings such as scaling policies and tags.
- Save your Auto Scaling group.

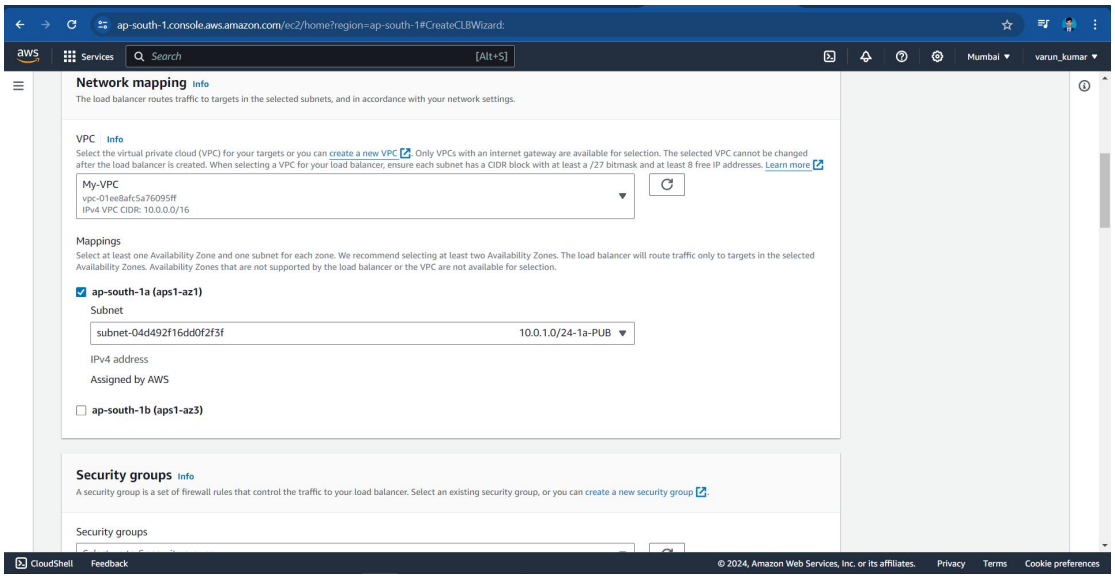
Configure the Auto Scaling Group with the Load Balancer:

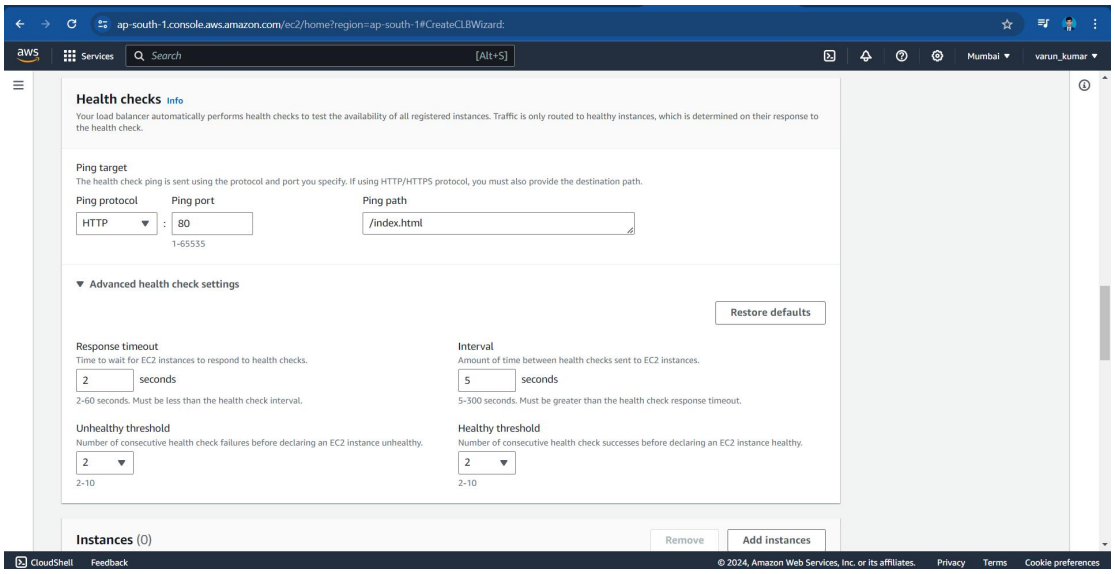
- In the EC2 Auto Scaling console, select your Auto Scaling group.
- Click on the "Edit" button in the "Details" tab.
- In the "Load balancing" section, select "Enable" for "Classic Load Balancer" or "Target groups" for "Application Load Balancer" or "Network Load Balancer."
- Select the load balancer and target groups you created in the previous step.
- Save your changes.

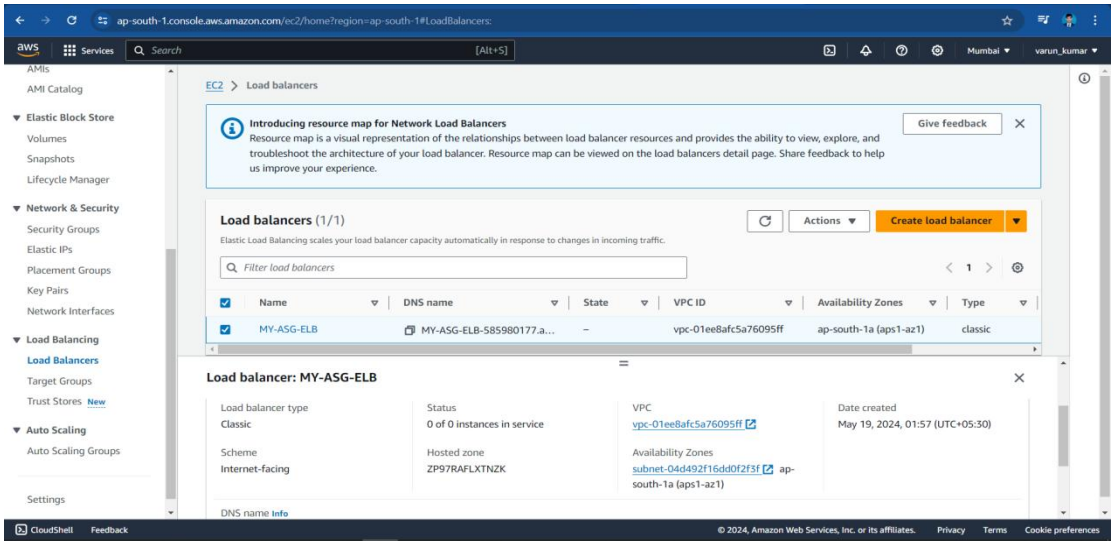
After following the above steps, we can connect to the webserver.

While creating classic load balancer we will have to choose the VPC we have created and we have to add the load balancer to the public subnet so we will select the public subnet. in the health checks option we will set the time out interval and threshold input and click on create.





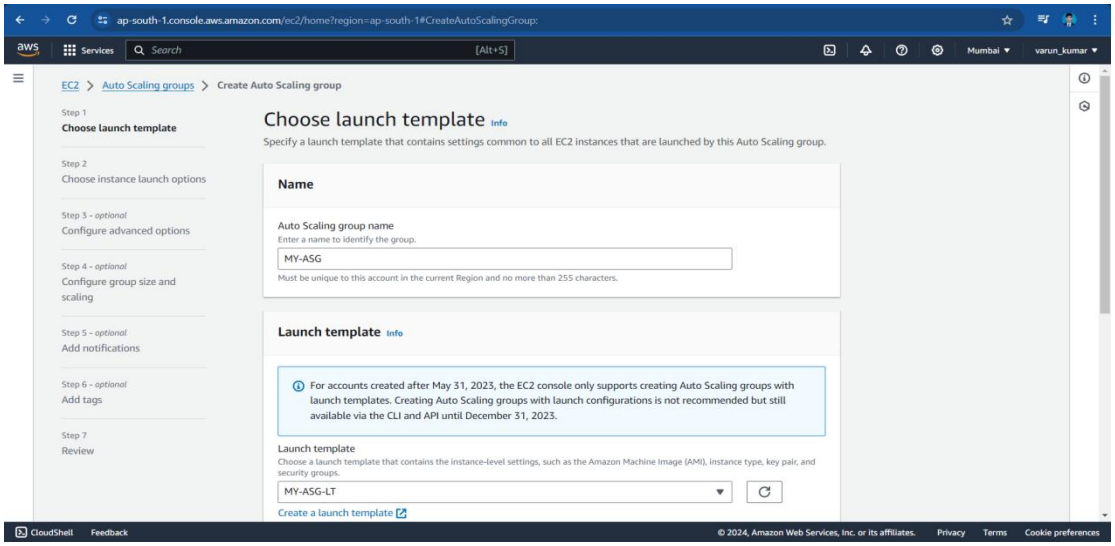


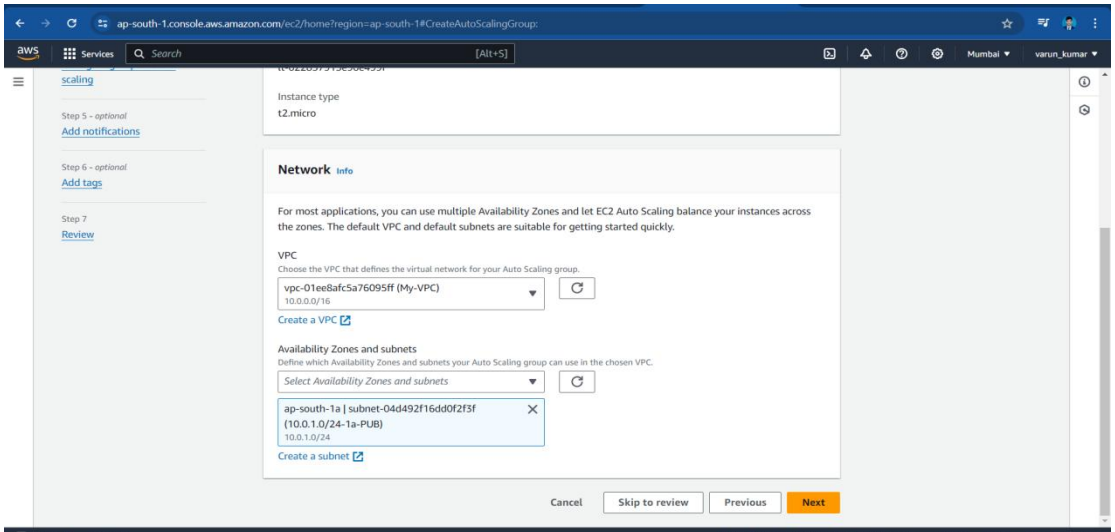


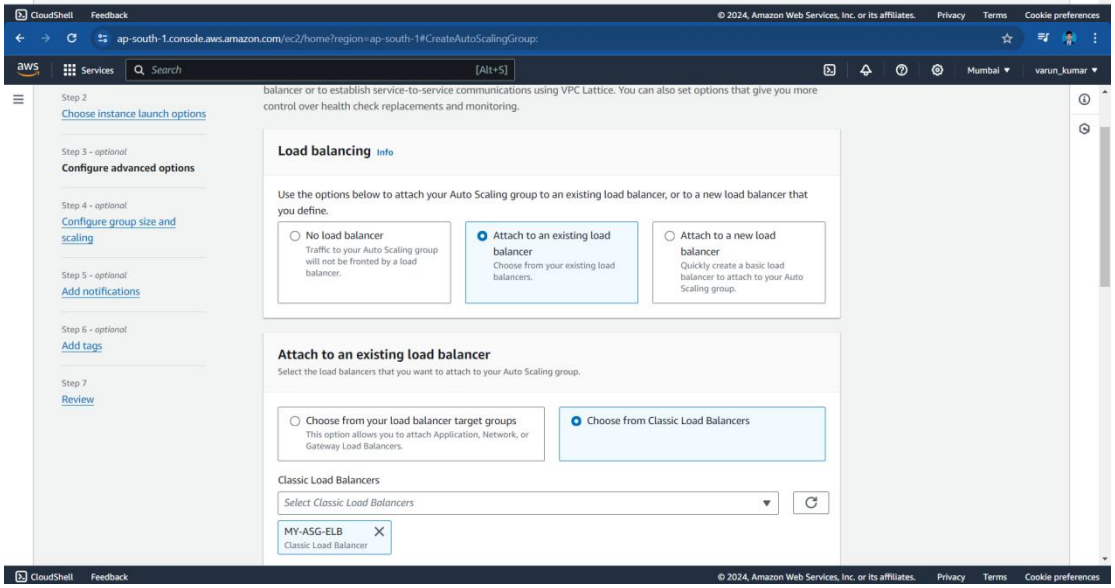
While creating auto scaling groups we will choose our created launch template and the availability zone in which we want to attach the auto scaling groups and the subnet and the vpc.

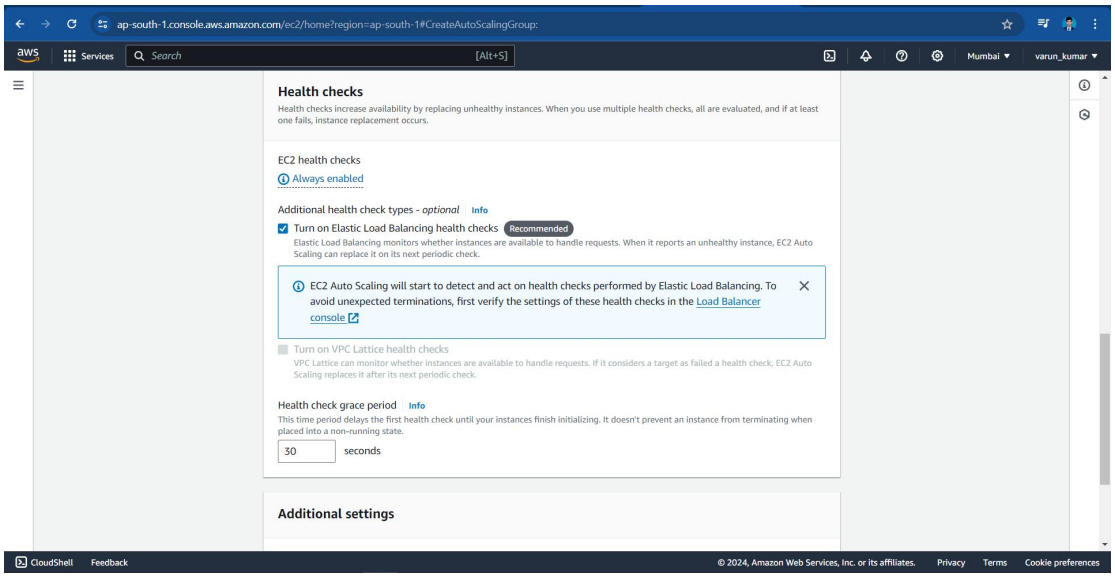
we will attach the created load balancer to the auto scaling group.

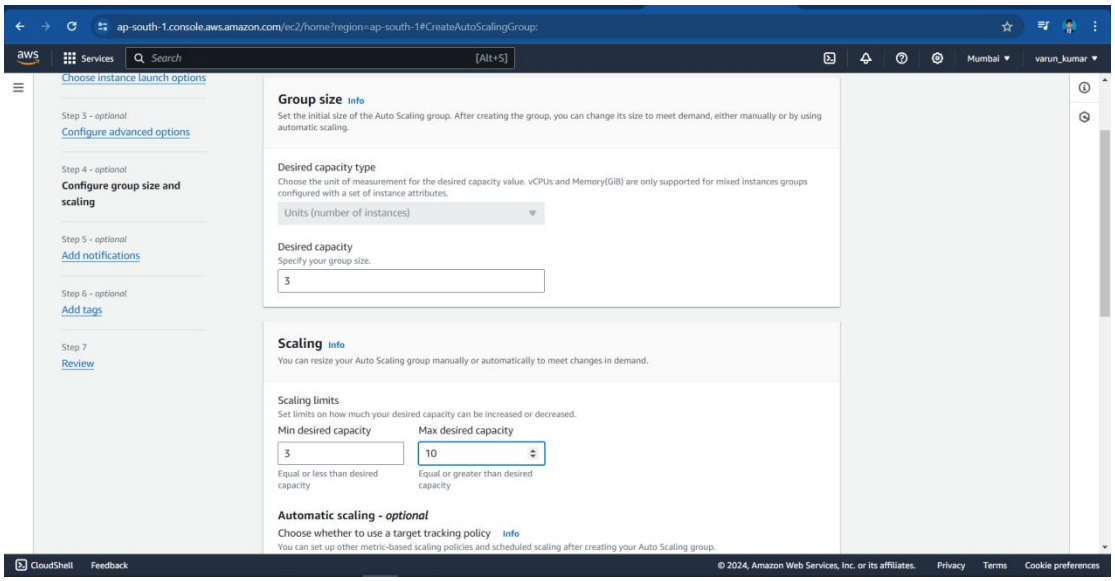
we will turn on the load balancing health check option.we will provide the desired, minimum desired and maximum desired capacity and click on create.

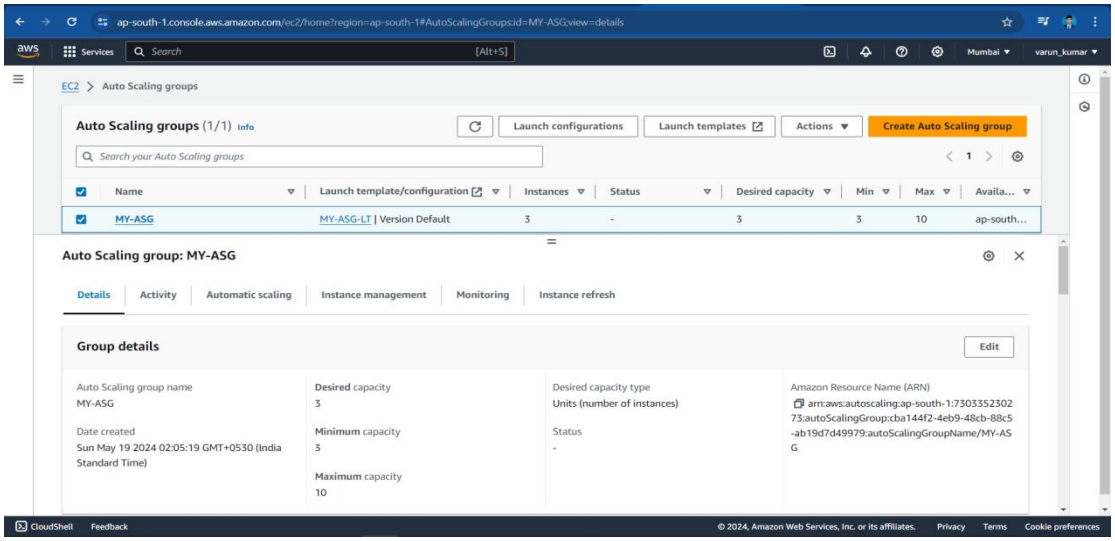






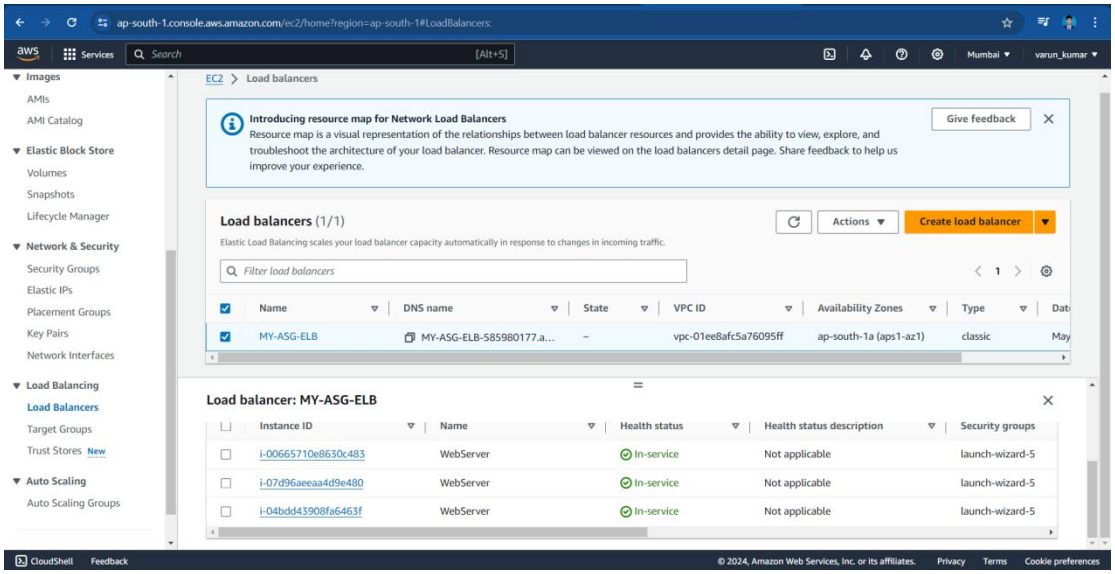


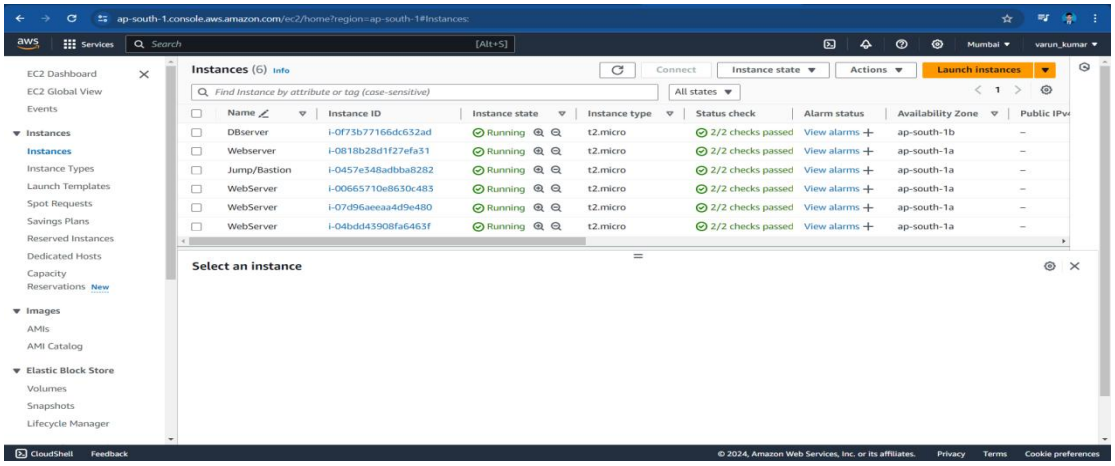




Now we can see that the load balancer will only allow the web server which have pass health check status.

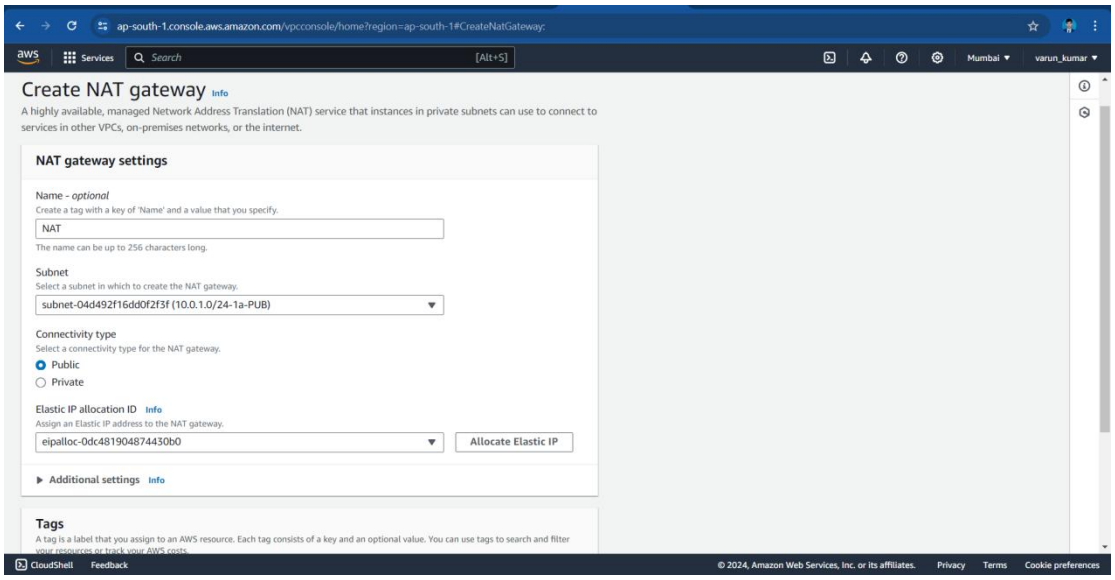
If the web server is crashed or some error has been occurred due to which web server is terminated ,the auto scaling group will create a new web server within the provided period.

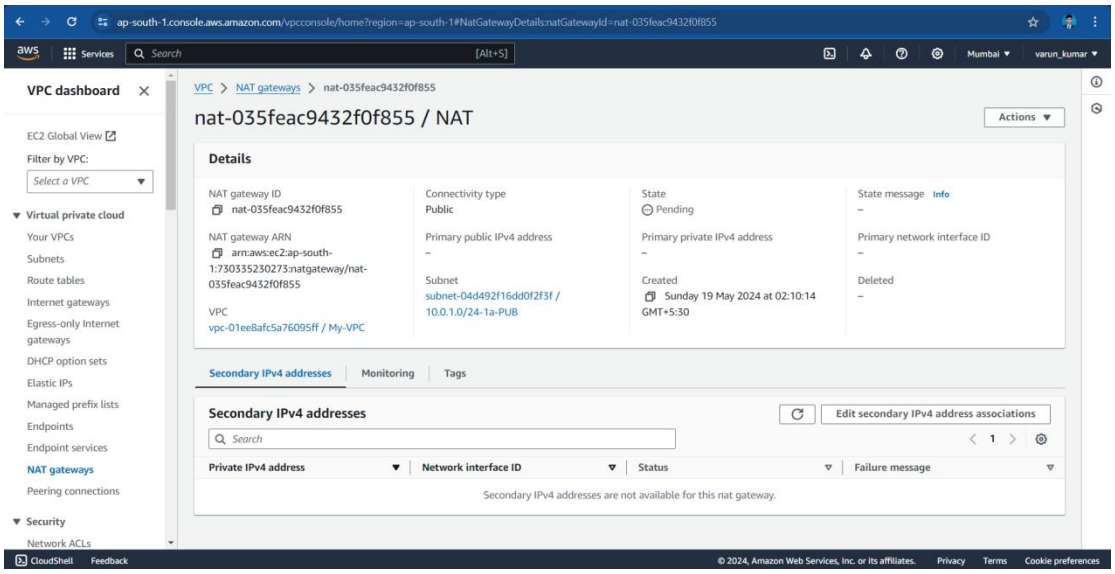


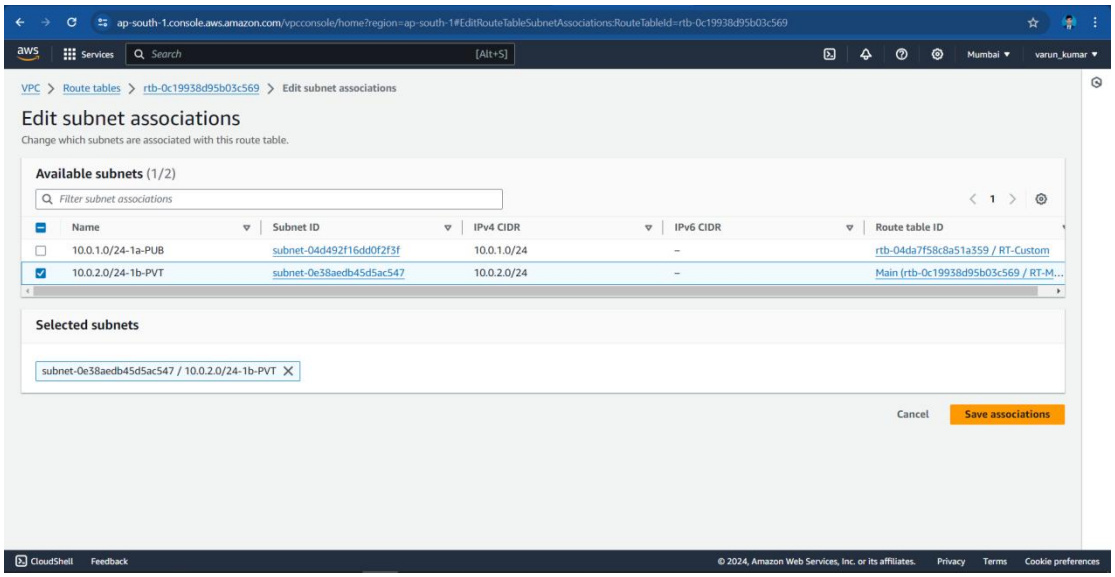


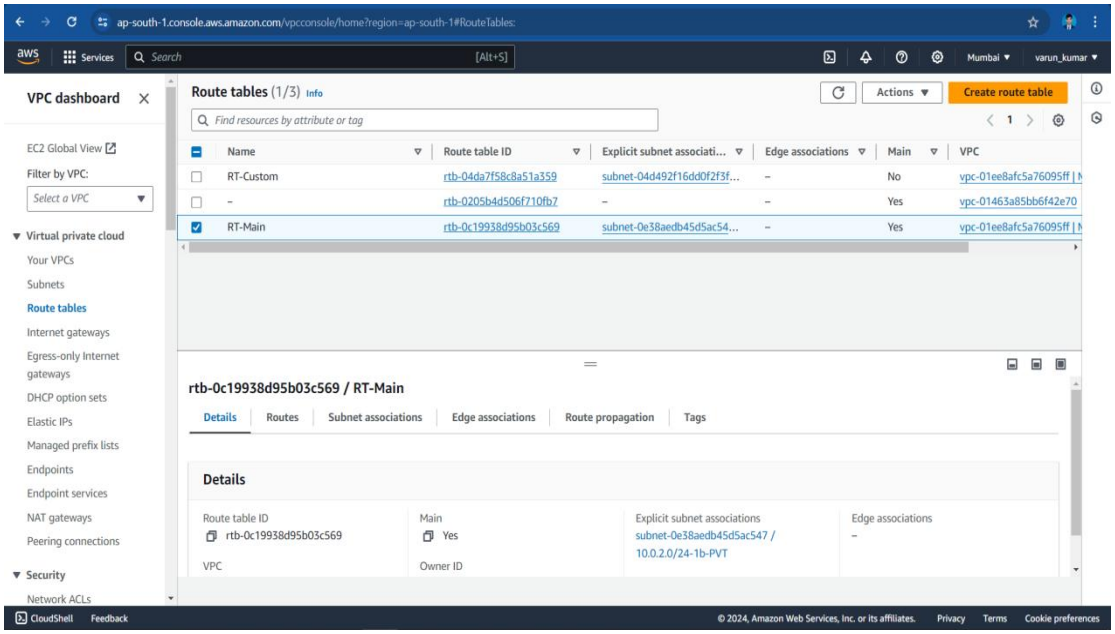
5.11 Creating NAT GATEWAY

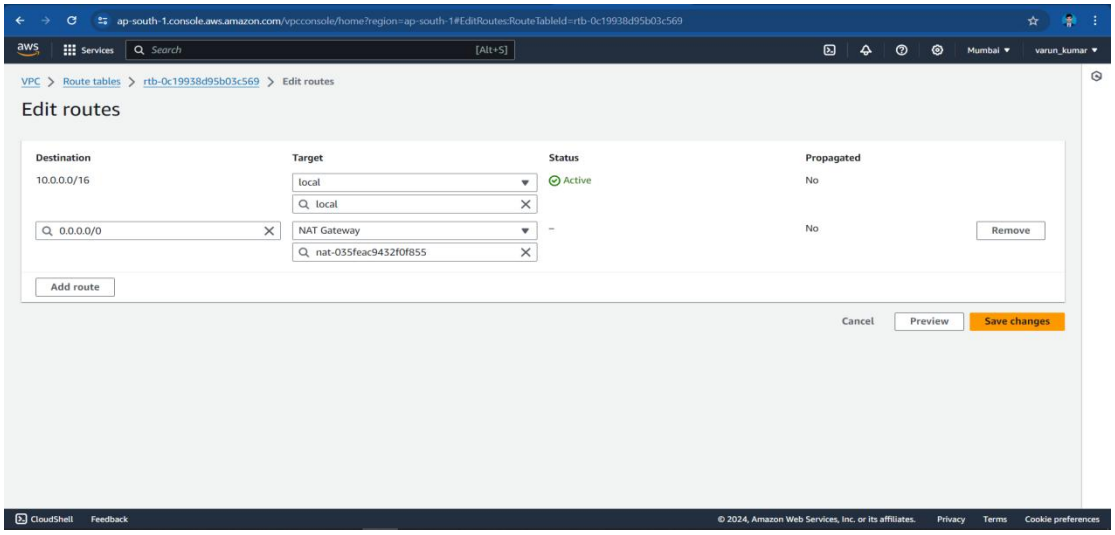
- Click on the "Create NAT Gateway" button to create a new NAT gateway.
- Configure the NAT gateway settings:
  - Select the subnet in which you want to create the NAT gateway. The subnet must be a public subnet, meaning it should have a route to an internet gateway.
  - Choose an existing Elastic IP address or allocate a new one to associate with the NAT gateway. The Elastic IP address serves as a public IP address for the NAT gateway.
- Verify the configuration details for the NAT gateway, including the selected subnet and Elastic IP address
- Click on the "Create NAT Gateway" button to create the NAT gateway. The creation process may take a few moments.
- Update route tables: After the NAT gateway is created, you need to update the route tables to direct the outbound traffic from private subnets to the NAT gateway.
  - Go to the "Route Tables" section in the VPC Dashboard.
  - Select the route table associated with the private subnets that need access to the internet via the NAT gateway.
  - Add a new route with a destination of "0.0.0.0/0" (or the desired IP range) and set the target as the newly created NAT gateway.
- Test the connectivity by launching an instance in a private subnet and ensuring it can access the internet through the NAT gateway.


ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateNatGateway:
Create NAT gateway Info
A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.
NAT gateway settings
Name - optional
Create a tag with a key of 'Name' and a value that you specify.
NAT
The name can be up to 256 characters long.
Subnet
Select a subnet in which to create the NAT gateway.
subnet-04d492f16dd0f2f3f (10.0.1.0/24-1a-PUB)
Connectivity type
Select a connectivity type for the NAT gateway.
Public
Private
Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-0dc481904874430b0
Allocate Elastic IP
Additional settings Info
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

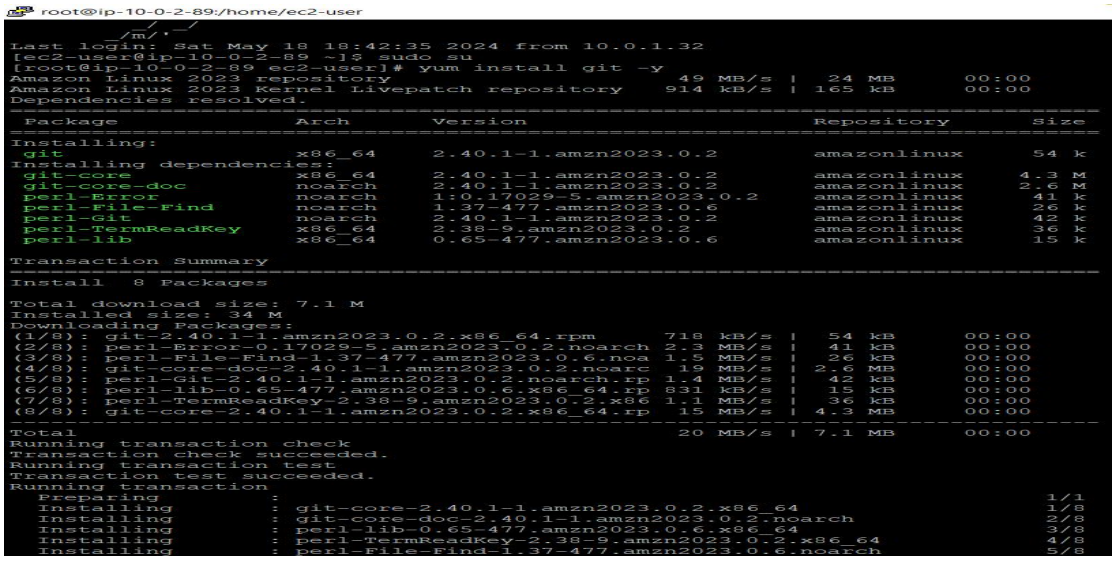

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#NatGatewayDetails:natGatewayId=nat-035feac9432f0f855
VPC dashboard
EC2 Global View
Filter by VPC:
Select a VPC
Virtual private cloud
Your VPCs
Subnets
Route tables
Internet gateways
Egress-only Internet gateways
DHCP option sets
Elastic IPs
Managed prefix lists
Endpoints
Endpoint services
NAT gateways
Peering connections
Security
Network ACLs
VPC > NAT gateways > nat-035feac9432f0f855
nat-035feac9432f0f855 / NAT
Actions
Details
NAT gateway ID
nat-035feac9432f0f855
Connectivity type
Public
State
Pending
State message Info
NAT gateway ARN
arn:aws:ec2:ap-south-1:730335230273:natgateway/nat-035feac9432f0f855
Primary public IPv4 address
Primary private IPv4 address
Primary network interface ID
VPC
vpc-01ee8afc5a76095ff / My-VPC
Subnet
subnet-04d492f16dd0f2f3f / 10.0.1.0/24-1a-PUB
Created
Sunday 19 May 2024 at 02:10:14 GMT+5:30
Deleted
Secondary IPv4 addresses
Monitoring
Tags
Secondary IPv4 addresses
Edit secondary IPv4 address associations
Private IPv4 address
Network interface ID
Status
Failure message
Secondary IPv4 addresses are not available for this nat gateway.


ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-0c19938d95b03c569
VPC > Route tables > rtb-0c19938d95b03c569 > Edit subnet associations
Edit subnet associations
Change which subnets are associated with this route table.
Available subnets (1/2)
Filter subnet associations
Name
Subnet ID
IPv4 CIDR
IPv6 CIDR
Route table ID
10.0.1.0/24-1a-PUB
subnet-04d492f16dd0f2f3f
10.0.1.0/24
rtb-04da7f58c8a51a359 / RT-Custom
10.0.2.0/24-1b-PVT
subnet-0e38aedb45d5ac547
10.0.2.0/24
Main (rtb-0c19938d95b03c569 / RT-M...
Selected subnets
subnet-0e38aedb45d5ac547 / 10.0.2.0/24-1b-PVT
Cancel
Save associations





After the configuration of NAT gateway through public subnet to private subnet we will have the access of internet in the database server.



5.12 Creating NACL for Security

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.

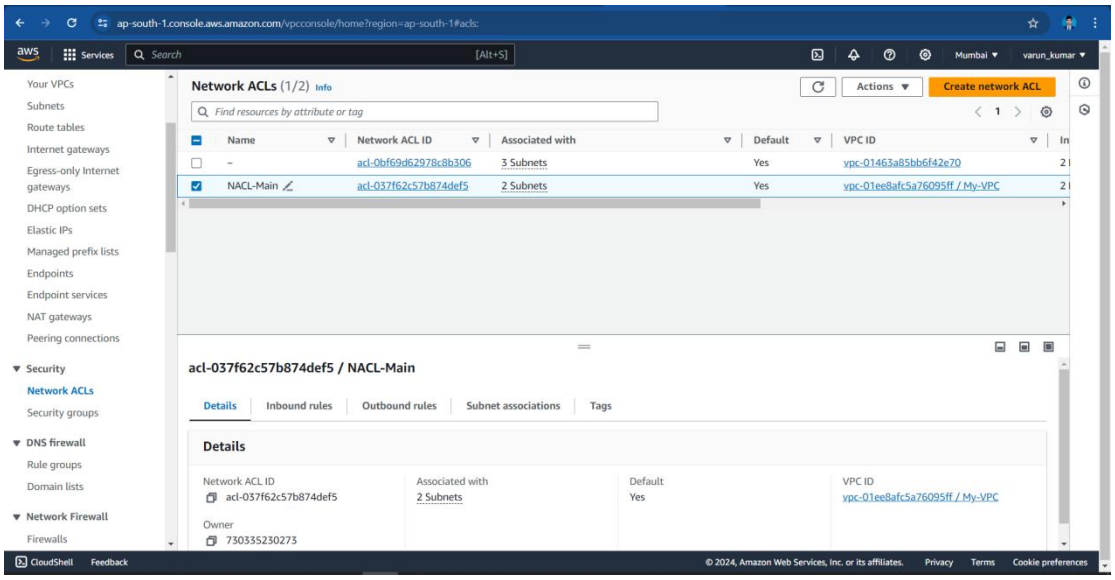
3. Associate the NACL with Subnets:
   - Click on the "Subnet Associations" tab in the NACL configuration.
   - Click on "Edit" and select the subnets you want to associate with the NACL.
   - You can choose to associate multiple subnets with the same NACL.
   - Make sure to confirm your changes to associate the NACL with the selected subnets.

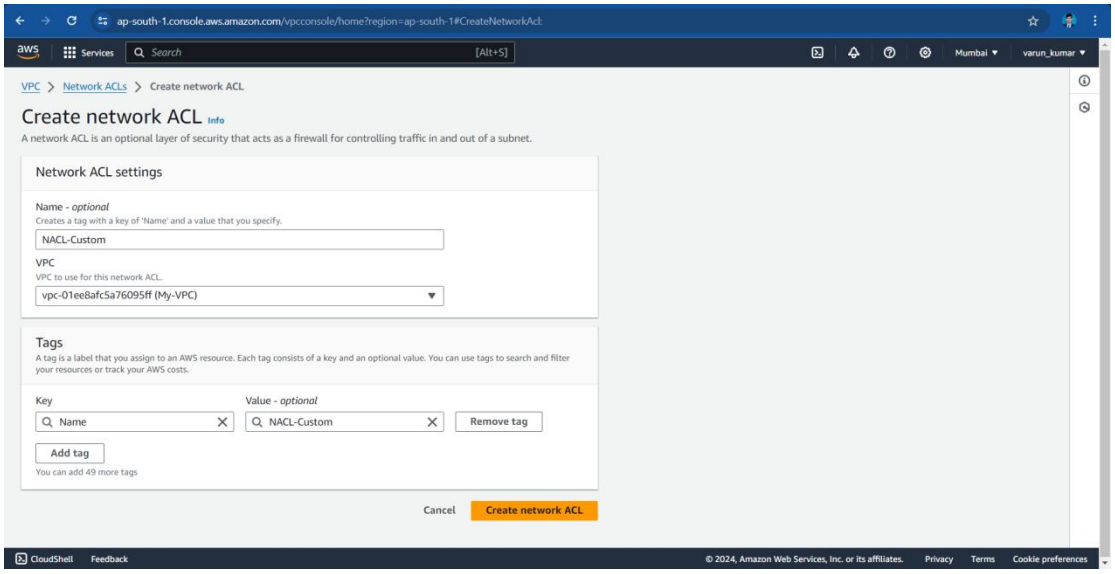
4. Configure Inbound and Outbound Rules:
   - Select the newly created NACL and click on the "Inbound Rules" or "Outbound Rules" tab.
   - Click on "Edit" and add or modify the rules according to your requirements.
   - Rules can be based on IP addresses, port ranges, protocols, and whether to allow or deny traffic.
   - Rules are evaluated in the order they appear, so ensure they are ordered correctly.

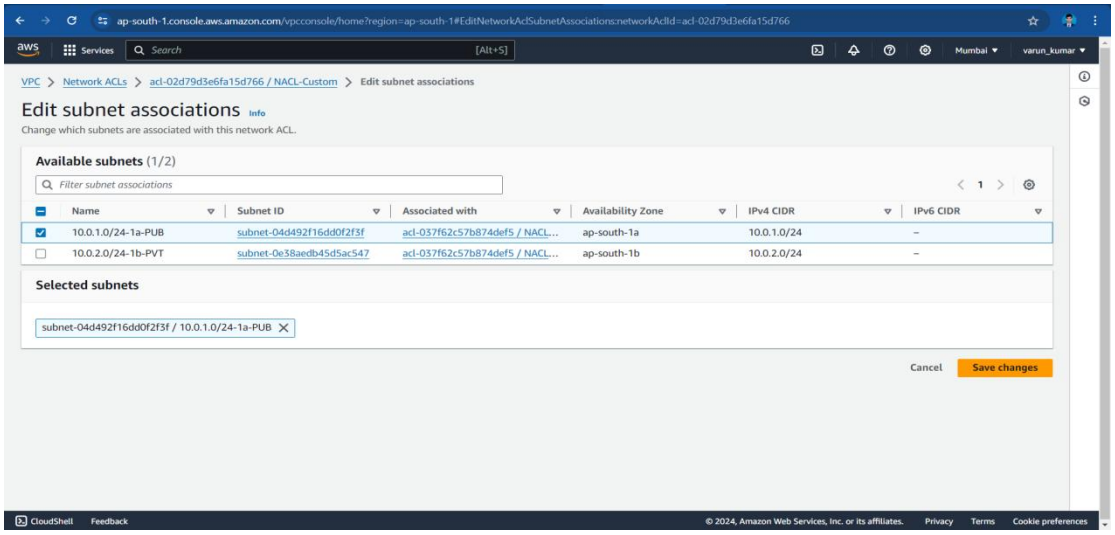
4. Associate the NACL with Subnets:
   - Click on the "Subnet Associations" tab in the NACL configuration.
   - Click on "Edit" and select the subnets you want to associate with the NACL.
   - You can choose to associate multiple subnets with the same NACL.
   - Make sure to confirm your changes to associate 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.

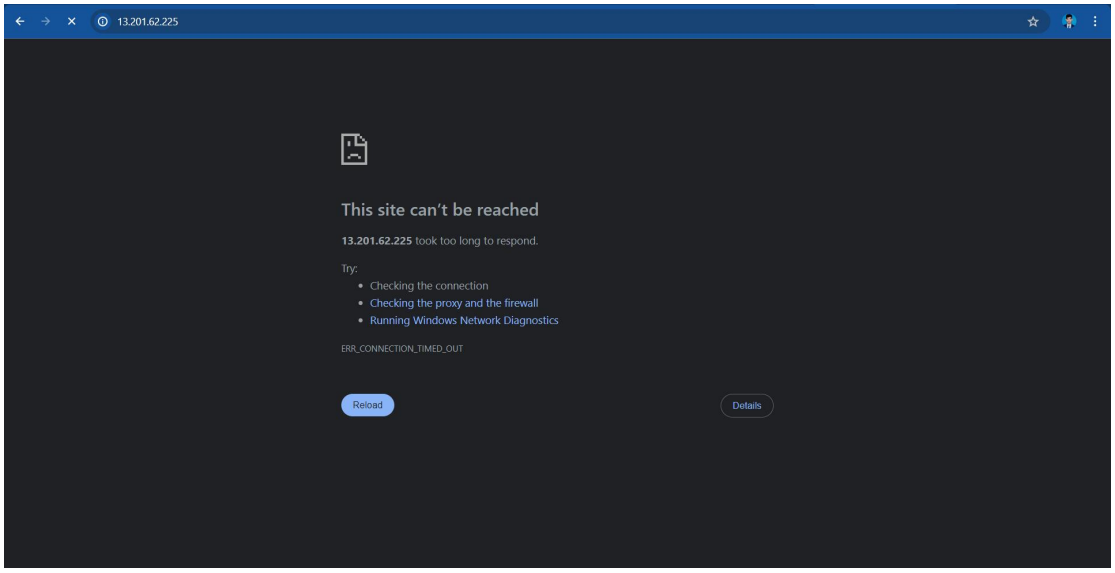
By following these steps, you can create and attach a Network Access Control List (NACL) to subnets in AWS. NACLs provide an additional layer of security by controlling inbound and outbound traffic at the subnet level, allowing you to define fine-grained rules for network traffic in your VPC.

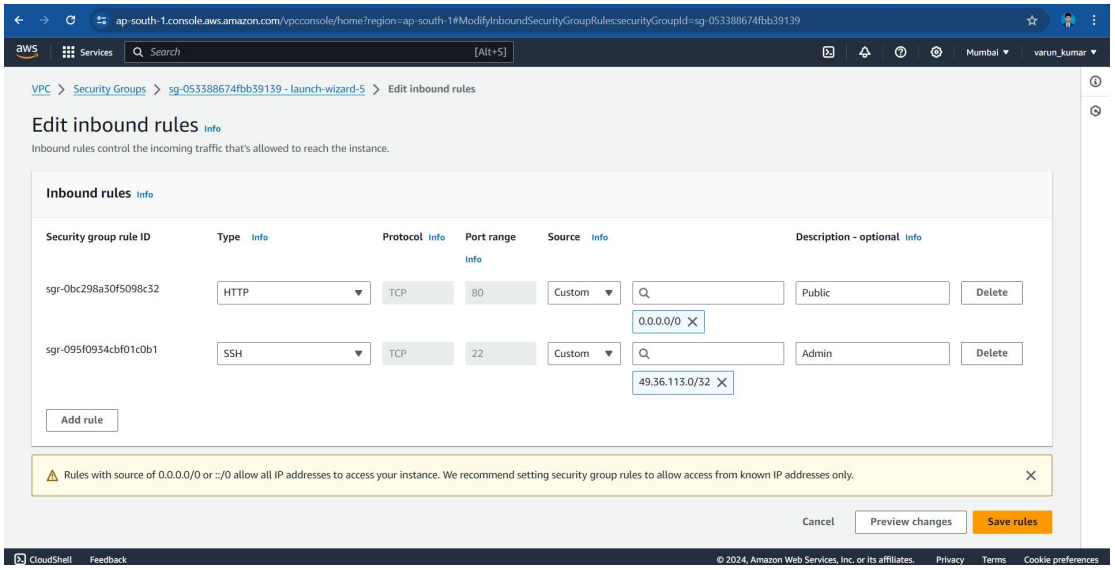

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#acls:
Network ACLs (1/2) Info
Actions
Create network ACL
Name
Network ACL ID
Associated with
Default
VPC ID
acl-0bf69d62978c8b306
3 Subnets
Yes
vpc-01463a85bb6f42e70
NACL-Main
acl-037f62c57b874def5
2 Subnets
Yes
vpc-01ee8afc5a76095ff / My-VPC
acl-037f62c57b874def5 / NACL-Main
Details
Inbound rules
Outbound rules
Subnet associations
Tags
Network ACL ID
acl-037f62c57b874def5
Associated with
2 Subnets
Default
Yes
VPC ID
vpc-01ee8afc5a76095ff / My-VPC
Owner
730335230273

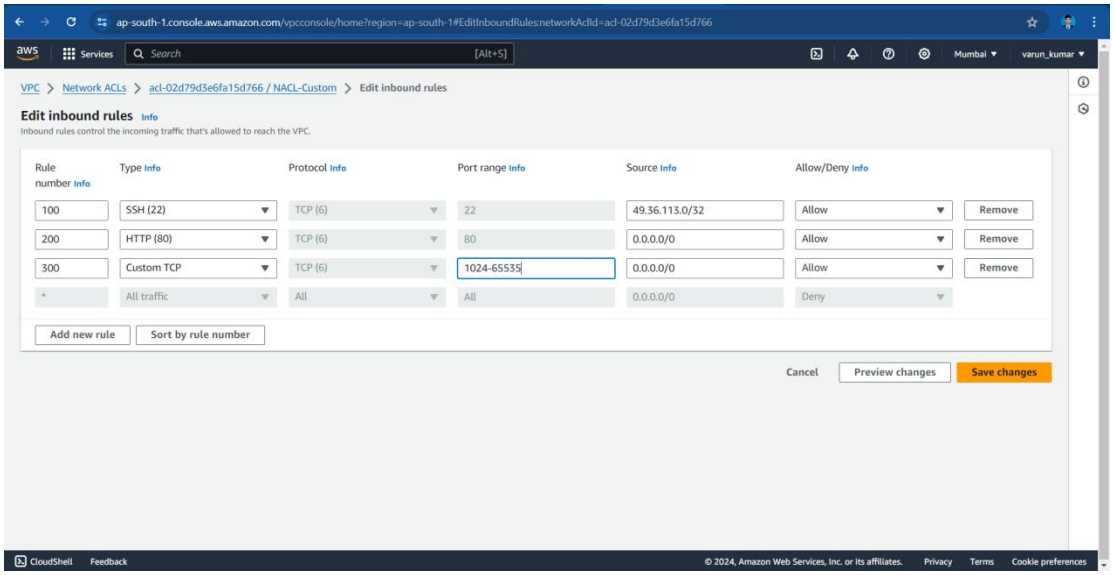

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#CreateNetworkAcl:
VPC > Network ACLs > Create network ACL
Create network ACL Info
A network ACL is an optional layer of security that acts as a firewall for controlling traffic in and out of a subnet.
Network ACL settings
Name - optional
Creates a tag with a key of 'Name' and a value that you specify.
NACL-Custom
VPC
VPC to use for this network ACL.
vpc-01ee8afc5a76095ff (My-VPC)
Tags
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.
Key
Value - optional
Name
NACL-Custom
Remove tag
Add tag
You can add 49 more tags
Cancel
Create network ACL

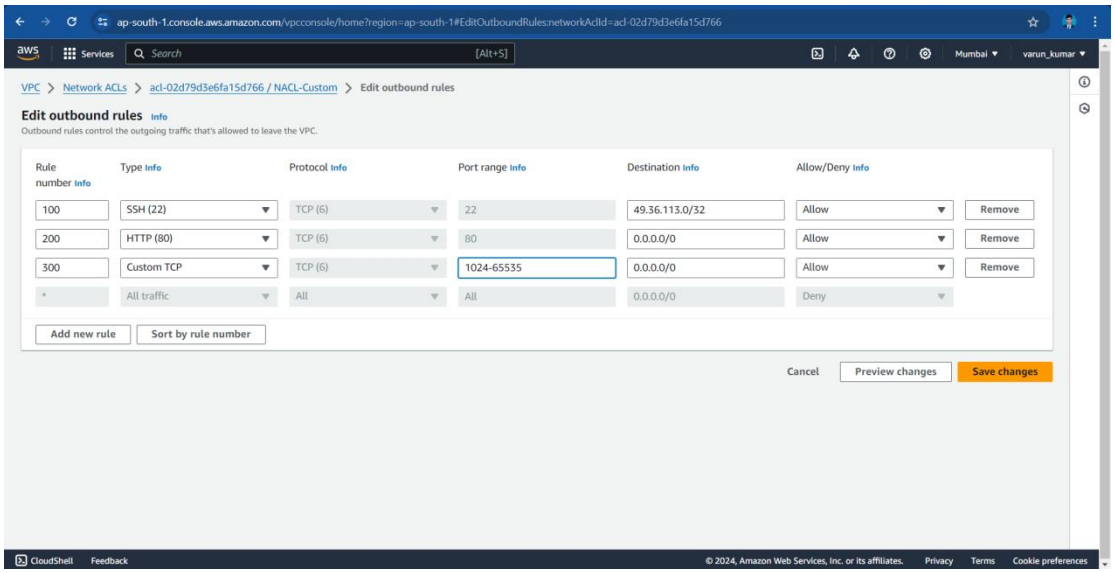

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditNetworkAclSubnetAssociations:networkAclId=acl-02d79d3e6fa15d766
VPC > Network ACLs > acl-02d79d3e6fa15d766 / NACL-Custom > Edit subnet associations
Edit subnet associations Info
Change which subnets are associated with this network ACL.
Available subnets (1/2)
Filter subnet associations
Name
Subnet ID
Associated with
Availability Zone
IPv4 CIDR
IPv6 CIDR
10.0.1.0/24-1a-PUB
subnet-04d492f16dd0f2f3f
acl-037f62c57b874def5 / NACL...
ap-south-1a
10.0.1.0/24
10.0.2.0/24-1b-PVT
subnet-0e38aedb45d5ac547
acl-037f62c57b874def5 / NACL...
ap-south-1b
10.0.2.0/24
Selected subnets
subnet-04d492f16dd0f2f3f / 10.0.1.0/24-1a-PUB
Cancel
Save changes

After changing the setting we can see that our web server has stop working.








ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditOutboundRules:networkAclId=acl-02d79d3e6fa15d766
Services
Search
[Alt+S]
Mumbai
varun_kumar
VPC > Network ACLs > acl-02d79d3e6fa15d766 / NACL-Custom > Edit outbound rules
Edit outbound rules Info
Outbound rules control the outgoing traffic that's allowed to leave the VPC.
Rule number Info
Type Info
Protocol Info
Port range Info
Destination Info
Allow/Deny Info
100
SSH (22)
TCP (6)
22
49.36.113.0/32
Allow
Remove
200
HTTP (80)
TCP (6)
80
0.0.0.0/0
Allow
Remove
300
Custom TCP
TCP (6)
1024-65535
0.0.0.0/0
Allow
Remove
*
All traffic
All
All
0.0.0.0/0
Deny
Add new rule
Sort by rule number
Cancel
Preview changes
Save changes
CloudShell
Feedback
© 2024, Amazon Web Services, Inc. or its affiliates.
Privacy
Terms
Cookie preferences


Not secure
13.201.62.225
Welcome to my website


13.201.62.225
Welcome to my website

6.Future Enhancement

- Expand the VPC by adding new subnets to accommodate more instances and services as needed.
- Implement tools and practices to ensure ongoing compliance with evolving regulatory standards.
- Use AI and machine learning algorithms for real-time threat detection and response to enhance security.
- Regularly collect and incorporate user feedback to continuously improve and update the infrastructure and services provided.
- Utilize AWS Cost Explorer and Trusted Advisor to continuously monitor and optimize costs.

7.Limitation

- Some older software versions and tools may not be fully compatible with the latest AWS services, necessitating updates or replacements.
- Continuous data backup is essential to prevent data loss; failure to implement backup strategies can result in significant data loss.
- Requires knowledge and experience with AWS infrastructure and services for efficient setup and management.
- Periodic manual intervention may be required to refresh and update data to reflect the latest changes.
- Full functionality requires up-to-date AWS services; disruptions or outages in AWS can impact access and performance.

8.Conclusion

In this project we have studied in detail about cloud computing its different types, uses& advantages. The Cloud computing technology is a growing at rapid pace; it will have a major impact on future. Future trends of Cloud computing has shown a positive review. Thestudy has been majorly focused on Amazon Web Services.

In this report we have done a detailed analysis of Amazon Web Services and its various components, and its services that we can use in Amazon Web Services.

The infrastructure set-up done for Compass Global has given me major understandingof Amazon Web Services which will definitely help me in the further growth in career. During executing the project the problems, challenges and various other factors have provided me a general insight of how the industry works. The help my colleagues and project guide has provided me during completing the project has developed me into a team person and also enhanced my leadership qualities

9. Reference

https://docs.aws.amazon.com/iam/?nc2=h_ql_doc_iam

https://www.geeksforgeeks.org/web-servers-work/

https://docs.aws.amazon.com/ec2/?nc2=h_ql_doc_ec2

https://en.wikipedia.org/wiki/Server_(computing)

https://docs.aws.amazon.com/s3/?icmpid=docs_homepage_featuredsvcs

https://www.javatpoint.com/cloud-service-models

https://docs.aws.amazon.com/vpc/?icmpid=docs_homepage_featuredsvcs