

**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**



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

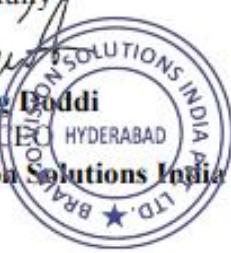
This is to certify that **Varun Kumar** has successfully completed his program with **BrainOvision Solutions Pvt. Ltd.** He was working on **AMAZON WEB SERVICES** and was actively & diligently involved in the projects and tasks assigned. During the span, we found him punctual and hardworking person.

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

Ganesh Nag Doddi
Founder & CEO HYDERABAD
Brainovision Solutions India Pvt Ltd



Dr. Buddha Chandrashekhar
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: Namayana 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, Madhagur, 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 the project.

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

Table of Contents.....	1
1. Introduction	
1.1 Synopsis of the Project.....	3
1.1.1 Company Profile.....	3
1.1.2 Objective.....	4
1.1.3 Scope.....	5
1.1.4 Problem Definition.....	6
1.1.5 Existing System.....	7
1.1.6 Proposed System.....	8
2. System Analysis.....	9
2.1 System Planning and Schedule.....	9
2.1.1 Gantt Chart.....	9
2.1.2 Pert Table.....	10
2.1.3 Waterfall Model.....	11
3. System Design.....	13
3.1 Technology We Use.....	13
3.1.1 Software Tools/Resources to be used.....	13
3.1.2 Selection of Technology/Specific Requirements.....	14
3.2 Detailed Life Cycle of the Project.....	14
4. Amazon Web Services.....	16
4.1 About AWS.....	16
4.2 EC2.....	16
4.3 VPC.....	17
4.4 IGW.....	18
4.5 Route Tables.....	19
4.6 Subnet.....	19
4.7 Servers.....	20
4.8 S3.....	21
4.9 IAM.....	22
4.10 Load Balancer.....	23
4.11 ASG.....	23
4.12 Bastion Host.....	24
4.13 NAT Gateway.....	25
4.14 NACL.....	26
5. Project Implementation and Screenshots.....	27
5.1 Creating a VPC.....	27

5.2	Creating Subnet.....	28
5.3	Creating Internet Gateway.....	30
5.4	Creating Route Table.....	31
5.5	Creating Public Instance.....	33
5.6	Creating an IAM role.....	40
5.7	Creating S3 Bucket using an IAM role.....	41
5.8	Creating Private Instance.....	44
5.9	Creating Jump Bastion Instance.....	45
5.10	Creating ELB and ASG.....	51
5.11	Creating NAT Gateway.....	55
5.12	Creating NACL for Security.....	58
6.	Future Enhancement.....	62
7.	Limitation.....	62
8.	Conclusion.....	63
9.	Reference.....	64

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

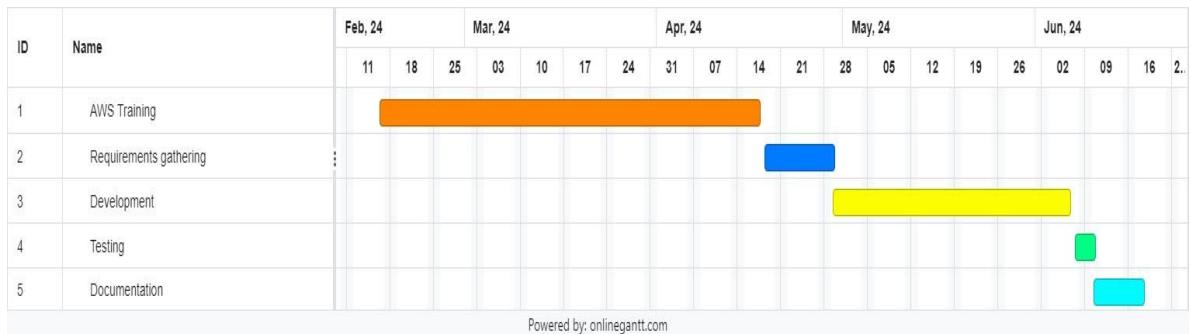
- The proposed system includes similar process to existing system with some different functionalities:
 1. VPC Creation
 2. IGW Creation
 3. Subnet Creation and Configuration
 4. EC2 instance Creation
 5. Security Configuration
 6. Route Tables Setup
 7. NAT Gateway Deployment
 8. Auto Scaling and Load 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

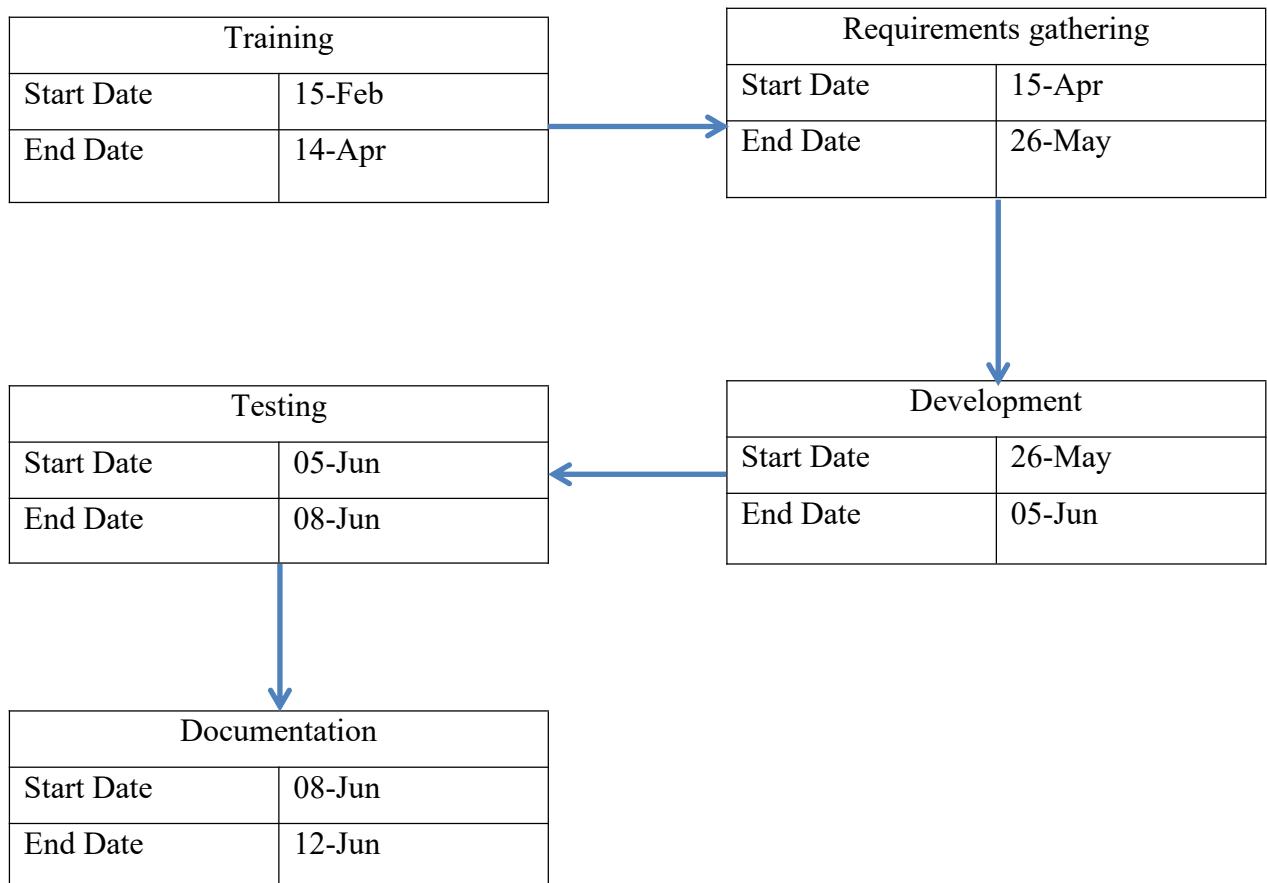
2.1 System Planning and Schedule

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.



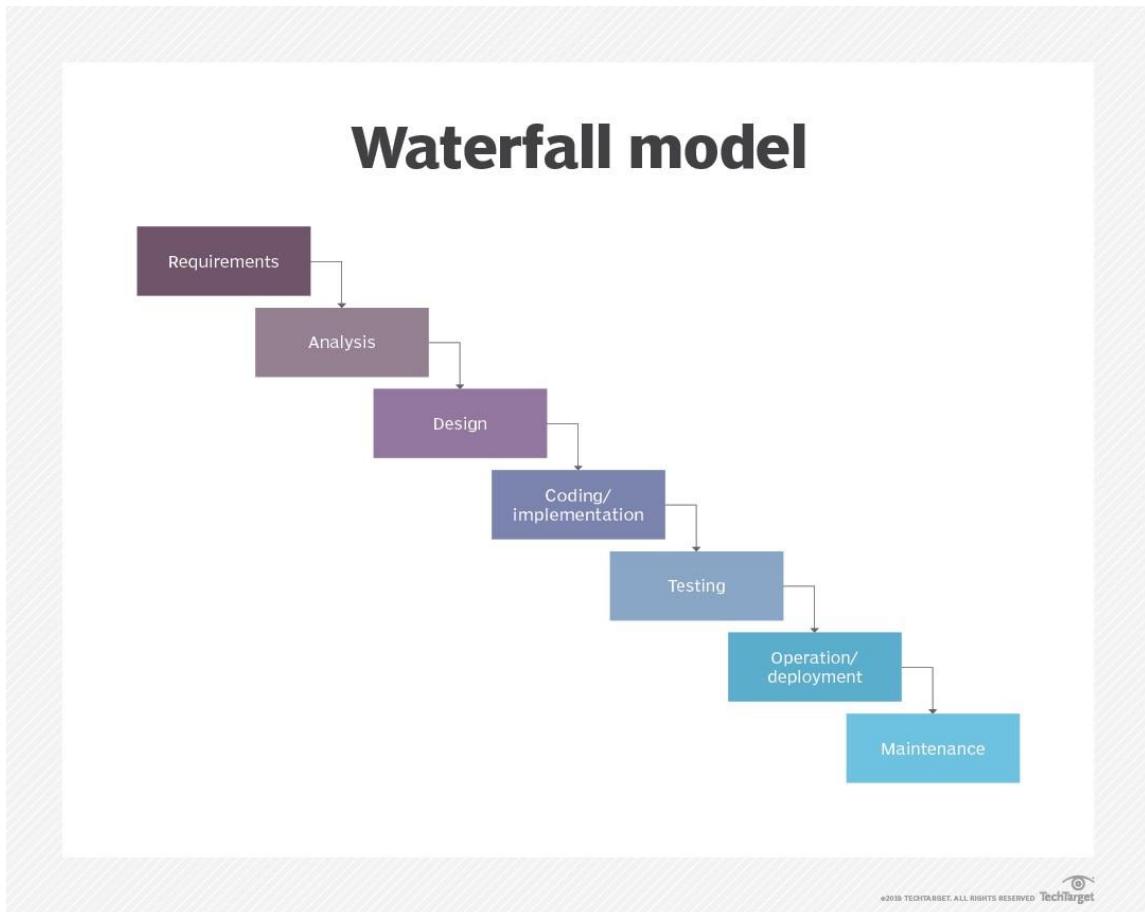
2.1.2 PERT TABLE



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.



e2020 TECHTARGET. ALL RIGHTS RESERVED 

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.
 - o) 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.

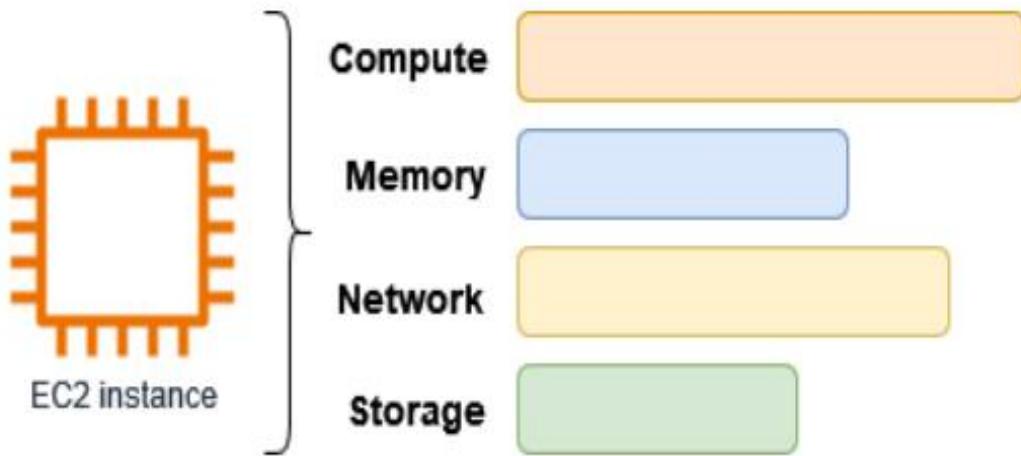


4.2 EC2

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.

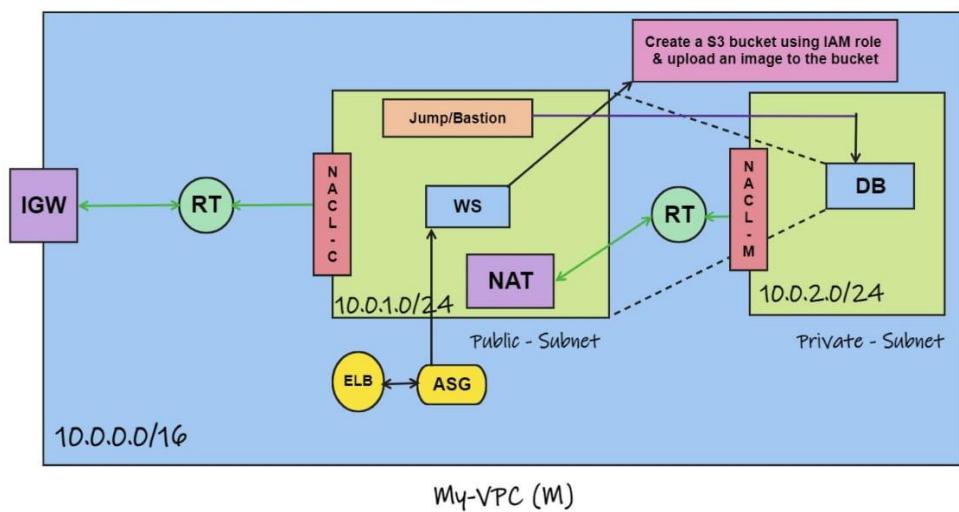
5. EC2 instances are deployed within a Virtual Private Cloud (VPC), which is a logically isolated section of the AWS cloud. This allows you to control the network settings, security, and access to your instances.



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:

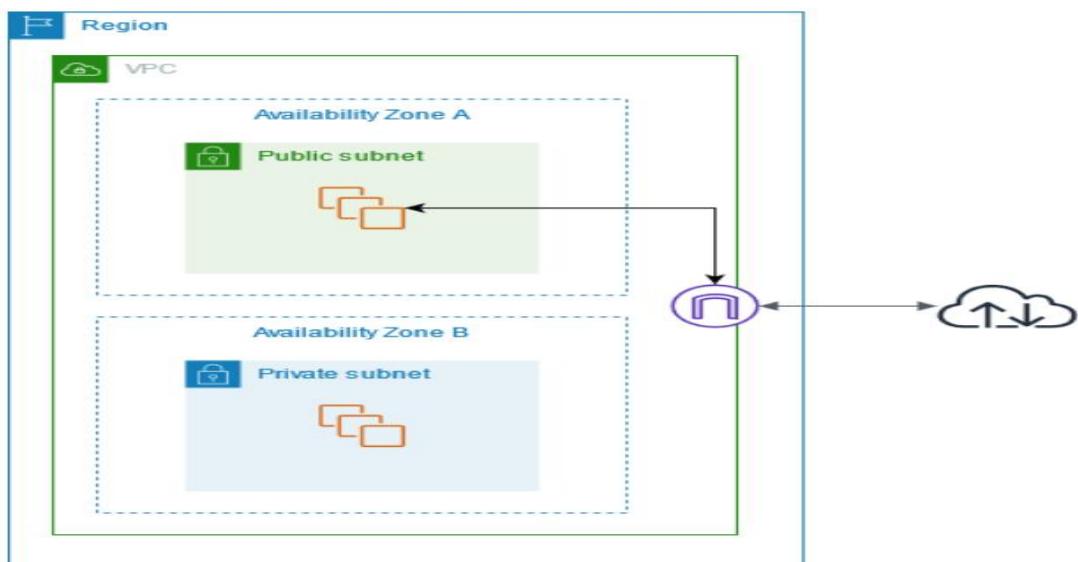
1. **Subnets:** Within the VPC, you can create multiple subnets. These subnets can be either public or private, depending on their routing configuration.
 - a. **Public Subnets:** Subnets with a route table that has a route to an Internet Gateway, allowing instances in that subnet to communicate with 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.



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:

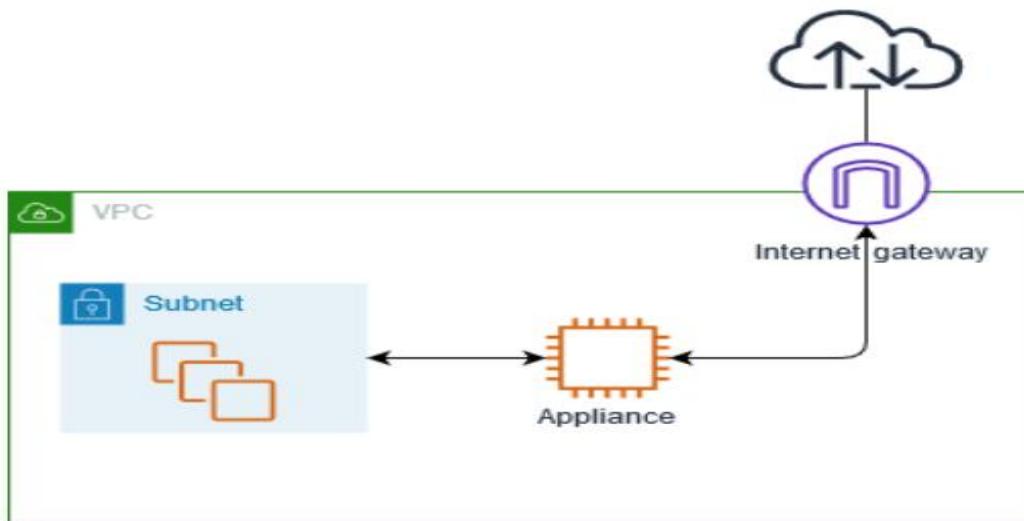
1. An IGW is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the internet.
2. When you create a VPC, it does not automatically come with an IGW. You need to create and attach an IGW to your VPC to enable internet access.
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

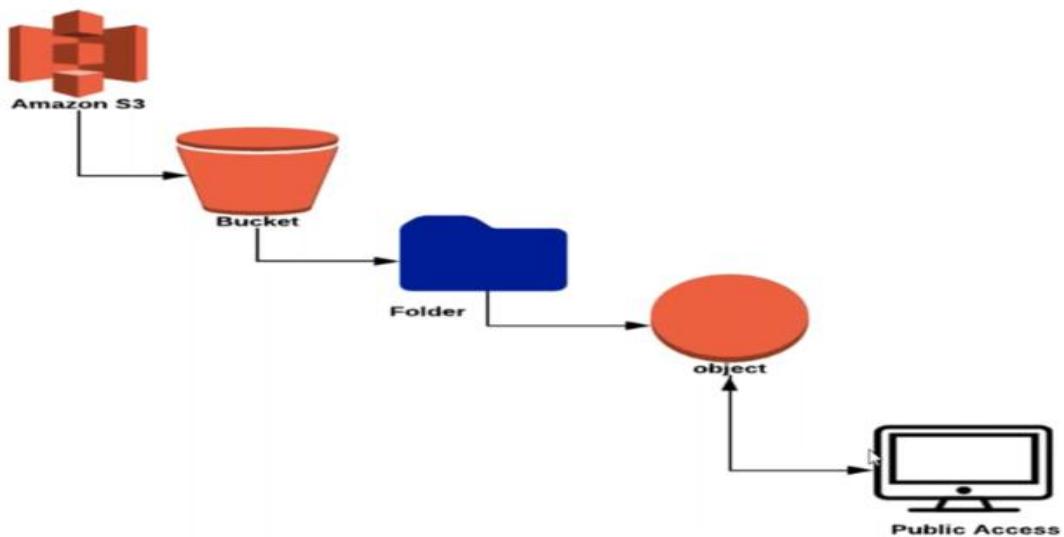
About Amazon S3 (Simple Storage Service) services:

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:

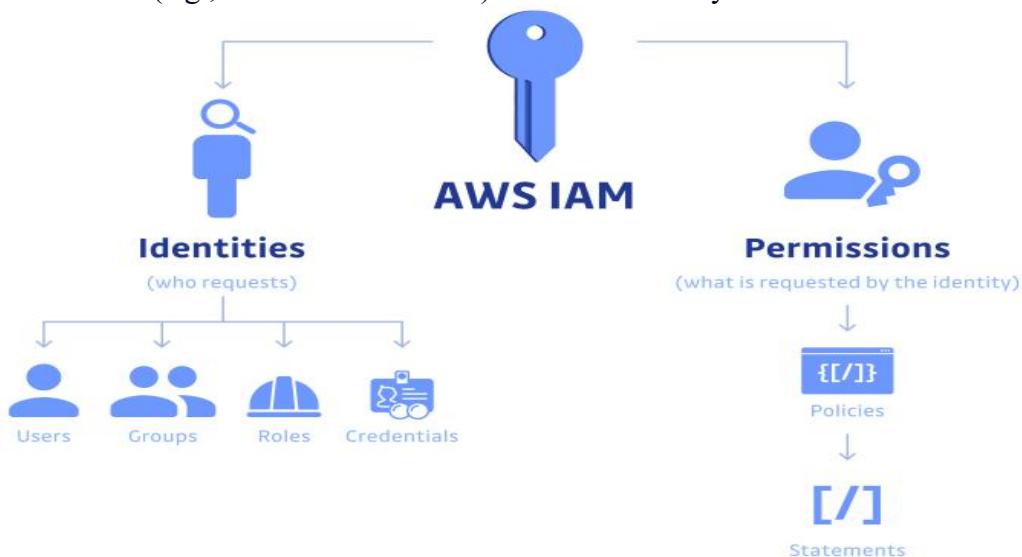
- a) Amazon S3 provides a simple web service interface to store and retrieve any amount of data from anywhere.
- b) It is highly scalable, and you only pay for what you use, allowing you to start small and grow as needed.
- c) S3 is designed to be highly flexible, allowing you to store any type and amount 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):

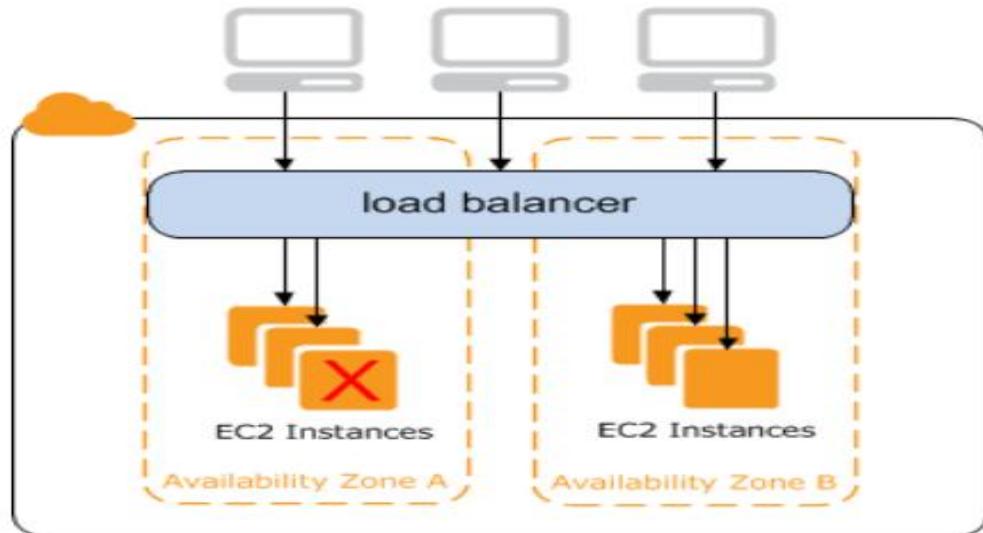
1. IAM is an AWS service that helps administrators securely control access to AWS resources. It allows you to manage user identities and their permissions.
2. IAM provides fine-grained access control across all of AWS. You can use IAM policies to manage permissions for your workforce and systems, ensuring 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).

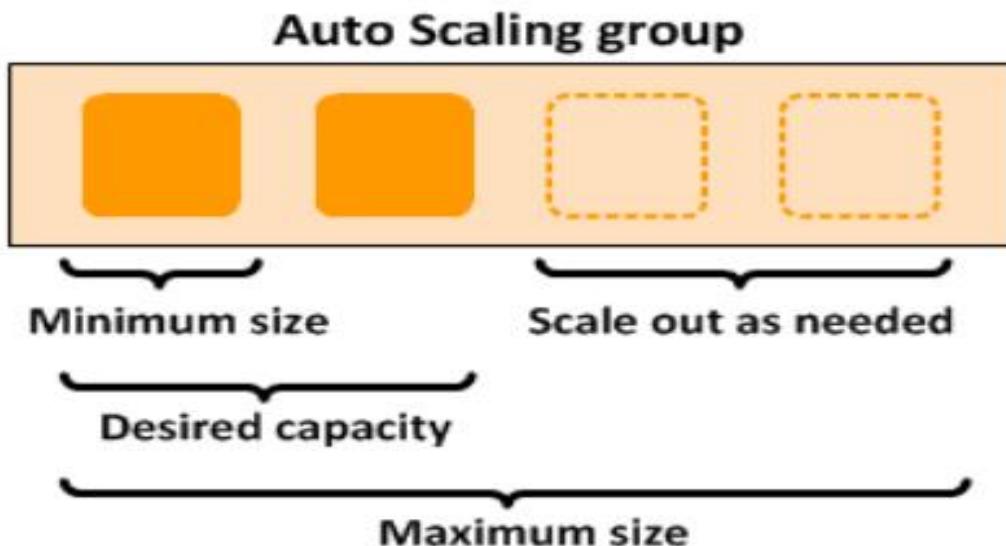


4.11 ASG

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.

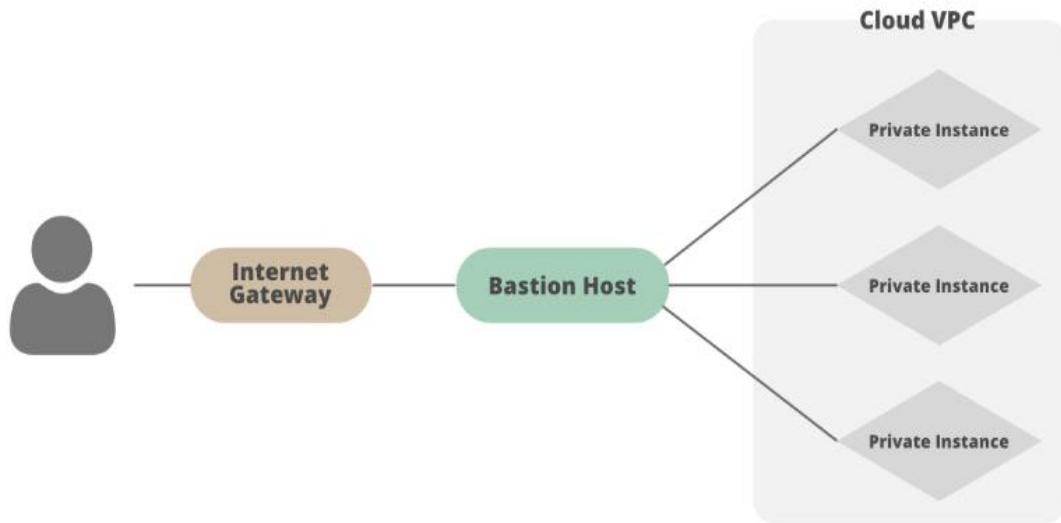


4.12 Bastion host

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:

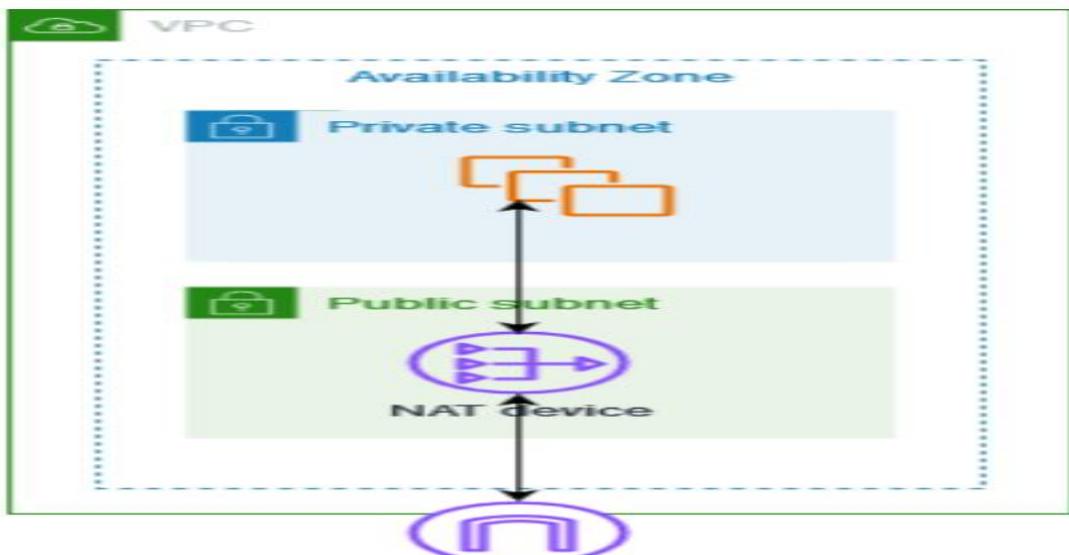
1. Bastion hosts are typically launched in a public subnet, allowing them to be accessed from the internet. This provides a secure entry point to connect to resources in private subnets.
2. To connect to a private EC2 instance from outside the VPC, you would first connect to the bastion host using SSH 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:

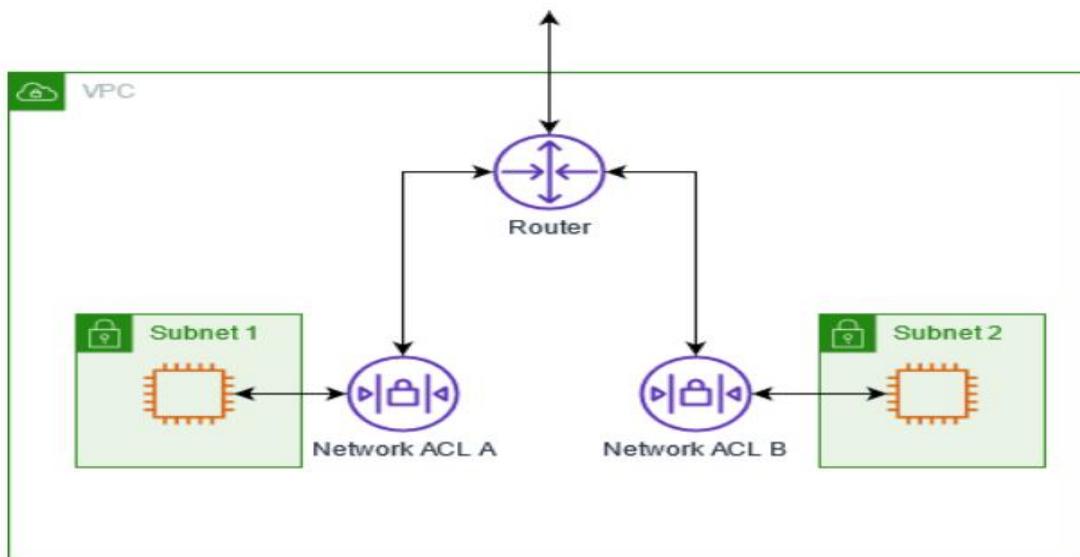
1. A NAT (Network Address Translation) Gateway is a managed service provided by AWS that enables instances in a private subnet to connect to the internet or other AWS services, while preventing the internet from initiating a connection with those instances.
2. The NAT Gateway performs both address translation 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:

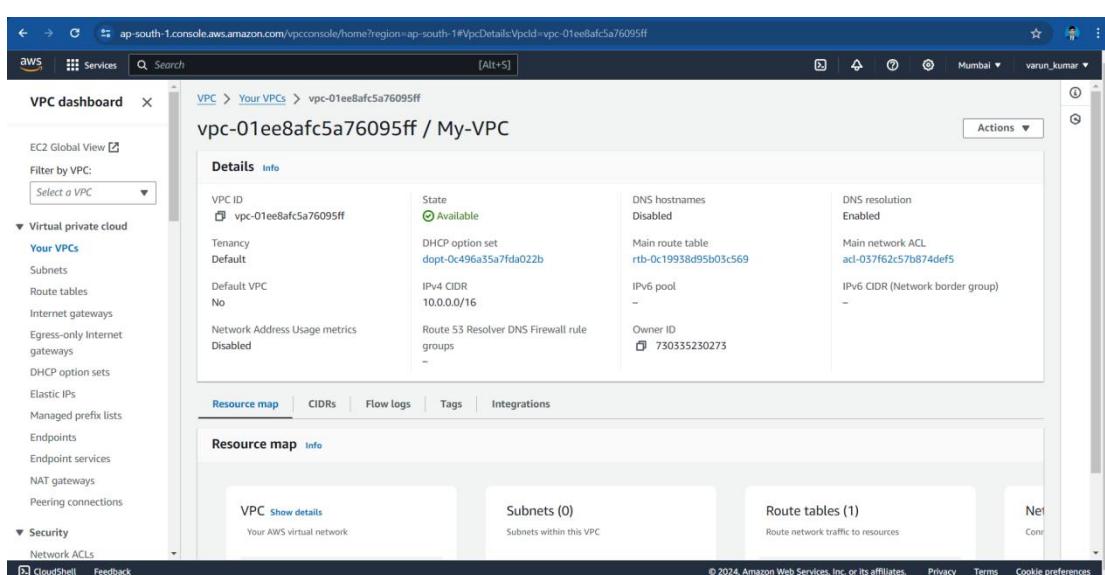
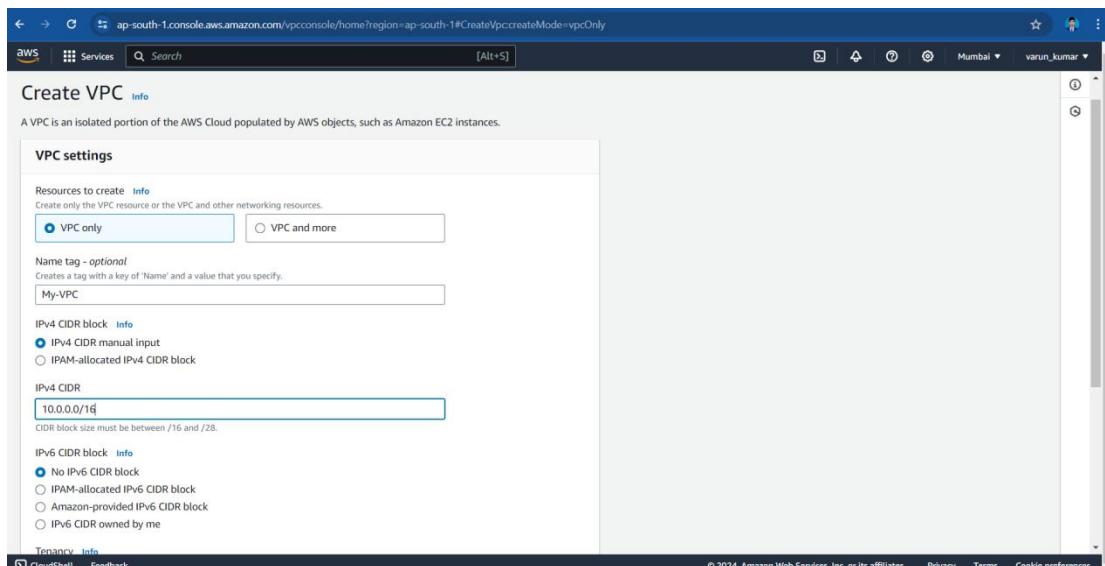
1. NACLs are an optional additional layer of security for your VPC, complementing the security group rules applied at the instance level.
2. NACLs have separate inbound and outbound rules, and each rule can either allow or deny traffic based on IP addresses, ports, and protocols.
3. NACL rules are evaluated 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

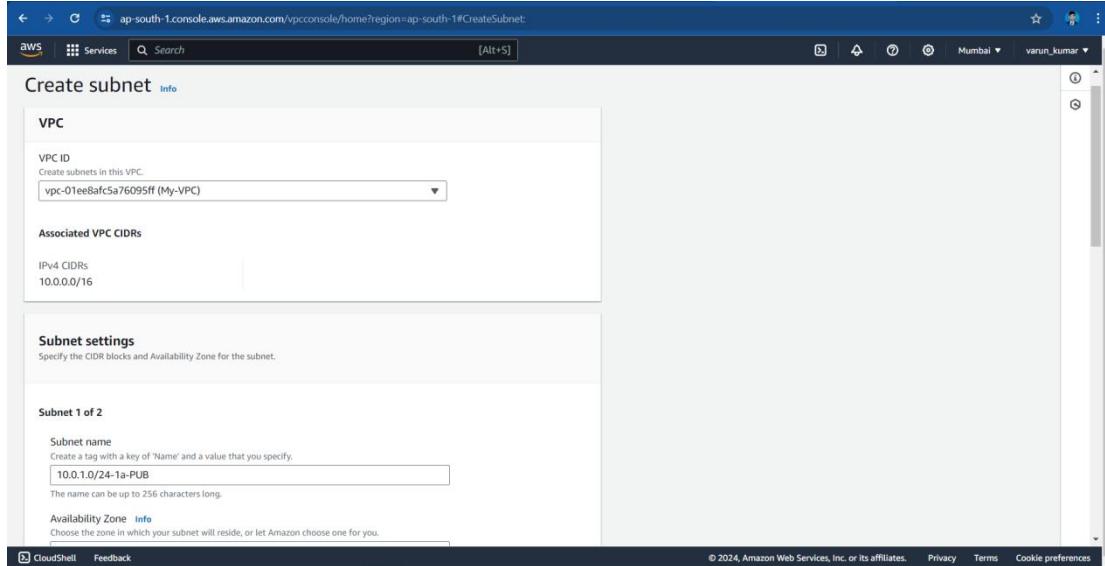
5.1 Creating a VPC (Virtual Private Cloud):

- 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.



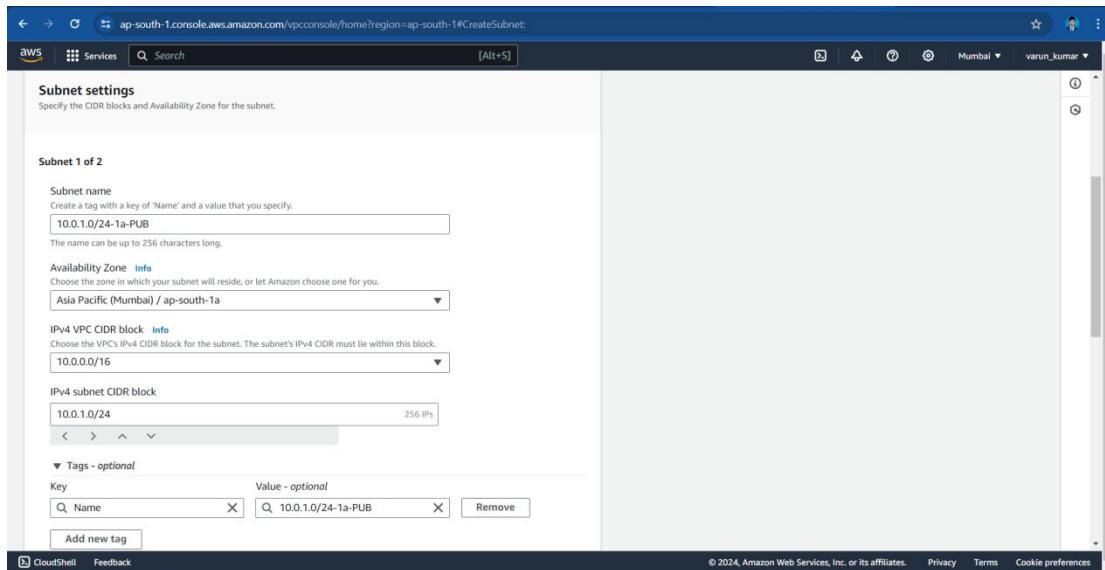
5.2 Creating Subnet

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.

- 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.

Subnet 2 of 2

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone **Info**
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 VPC CIDR block **Info**
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

IPv4 subnet CIDR block
 256 IPs

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="10.0.2.0/24-1b-PVT"/>

Add new tag
You can add 49 more tags.

Remove

Add new subnet

Subnets (2) **Info**

Name	Subnet ID	State	VPC	IPv4 CIDR
10.0.1.0/24-1a-PUB	subnet-04d492f16dd0f2f3f	Available	vpc-01ee8afc5a76095ff My-VPC	10.0.1.0/24
10.0.2.0/24-1b-PVT	subnet-0e38aedb45d5ac547	Available	vpc-01ee8afc5a76095ff My-VPC	10.0.2.0/24

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

Edit subnet settings **Info**

Subnet

Subnet ID <input type="text" value="subnet-04d492f16dd0f2f3f"/>	Name <input type="text" value="10.0.1.0/24-1a-PUB"/>
--	---

Auto-assign IP settings **Info**
Enable AWS to automatically assign a public IPv4 or IPv6 address to a new primary network interface for an instance in this subnet.

Enable auto-assign public IPv4 address **Info**

Enable auto-assign customer-owned IPv4 address **Info**
Option disabled because no customer owned pools found.

Resource-based name (RBN) settings **Info**
Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

Enable resource name DNS A record on launch **Info**

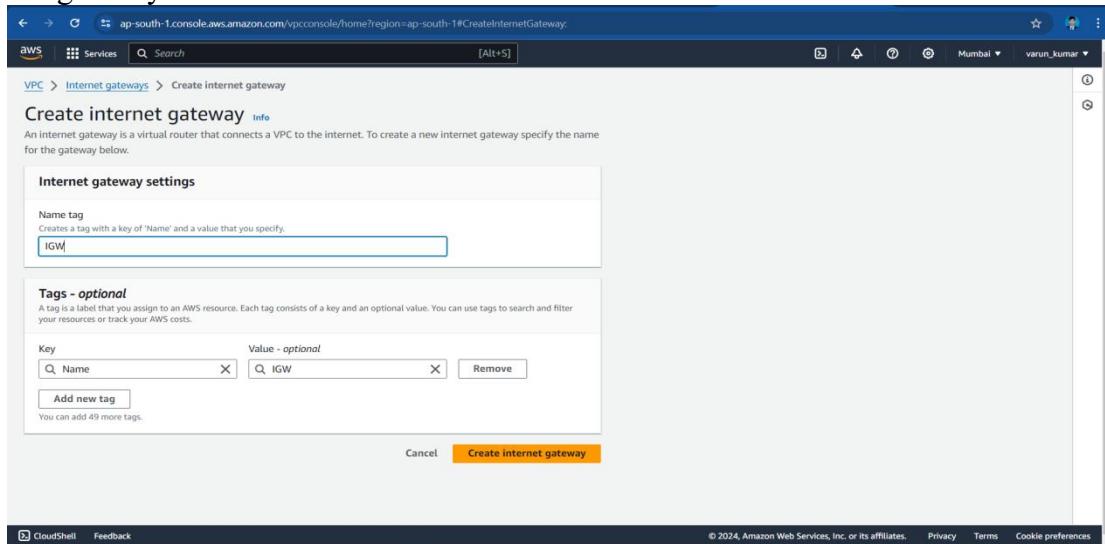
Enable resource name DNS AAAA record on launch **Info**

Hostname type **Info**

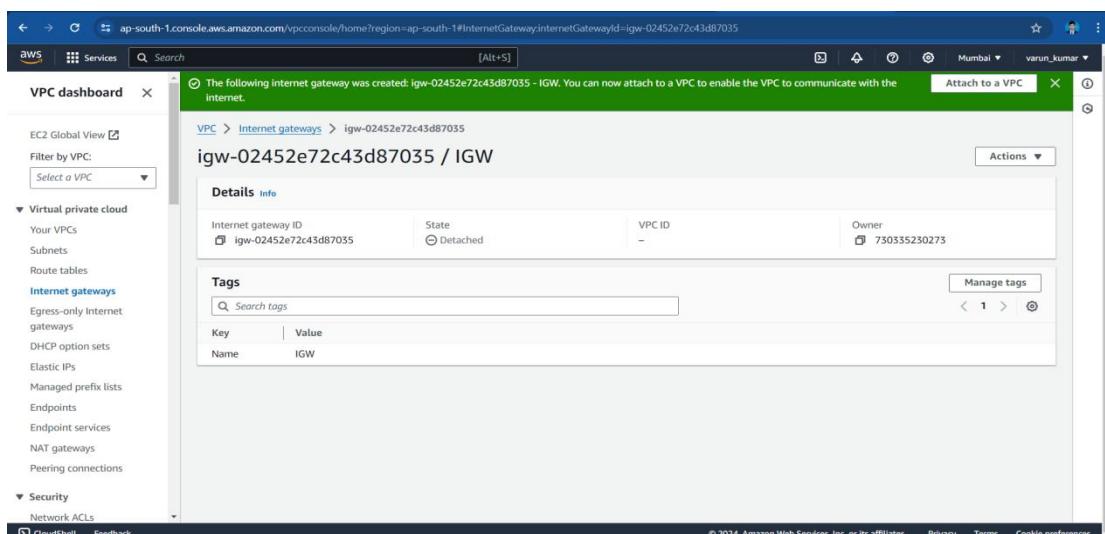
Resource name
 IP name

5.3 Creating Internet Gateway

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



The screenshot shows the 'Create internet gateway' wizard. In the 'Internet gateway settings' section, the 'Name tag' field contains 'IGW'. Below it, the 'Tags - optional' section shows a single tag with Key 'Name' and Value 'IGW'. At the bottom right, the 'Create internet gateway' button is highlighted in orange.

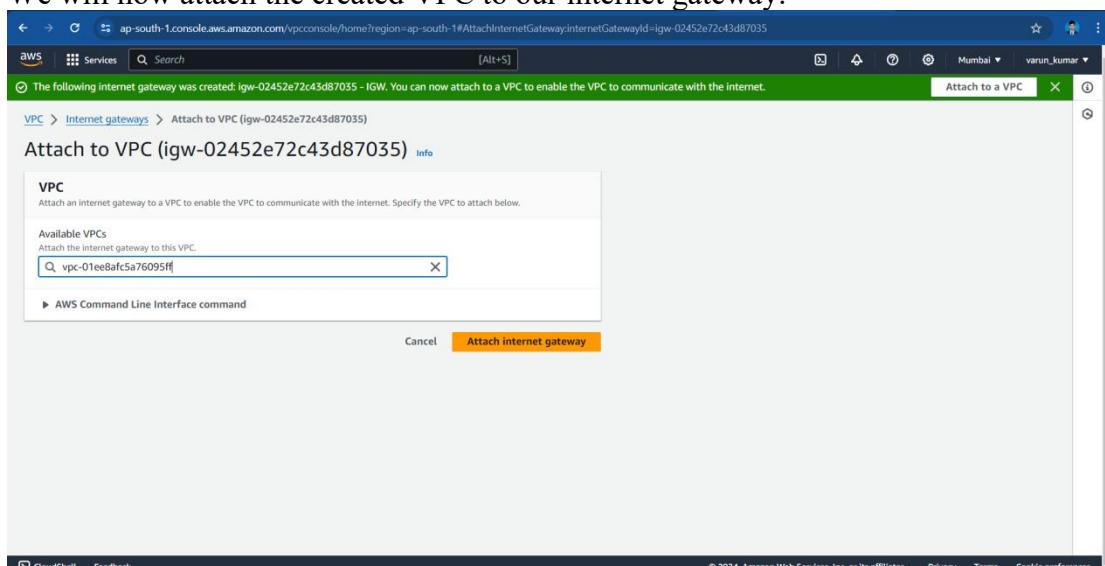


The screenshot shows the VPC dashboard with the newly created Internet Gateway 'igw-02452e72c43d87035 / IGW'. The 'Details' tab displays the following information:

- Internet gateway ID: igw-02452e72c43d87035
- State: Detached
- VPC ID: None
- Owner: 730335230273

The 'Tags' section shows one tag: Name: IGW. An 'Attach to a VPC' button is located at the top right of the gateway's card.

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



The screenshot shows the 'Attach to VPC' wizard. In the 'Available VPCs' section, the VPC 'vpc-01e8afc5a76095f' is selected. At the bottom right, the 'Attach internet gateway' button is highlighted in orange.

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.

Route table ID: rtb-04da7f58c8a51a359

VPC: vpc-01ee8afc5a76095ff | My-VPC

Routes (1):

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

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

Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
10.0.1.0/24-1a-PUB	subnet-04d492f16dd0f2f3f	10.0.1.0/24	-	Main (rtb-0c19938d95b03c569 / RT-M...)
10.0.2.0/24-1b-PVT	subnet-0e38aedb45d5ac547	10.0.2.0/24	-	Main (rtb-0c19938d95b03c569 / RT-M...)

Selected subnets

subnet-04d492f16dd0f2f3f / 10.0.1.0/24-1a-PUB	X
---	---

Save associations

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

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway igw-02452e72c43db7035	-	No

Add route

Save changes

5.5 Creating Public Instance

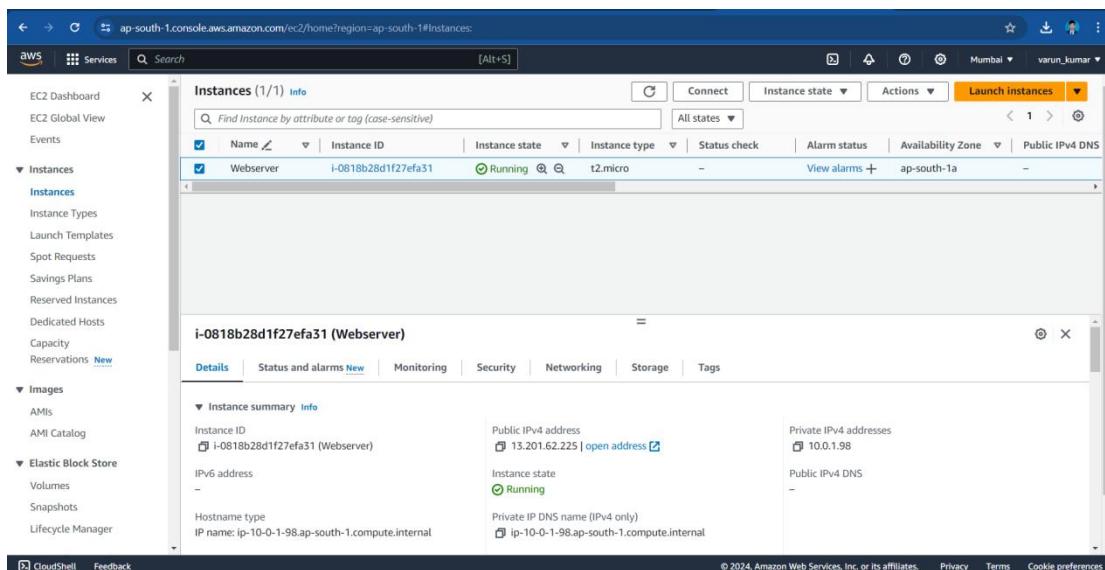
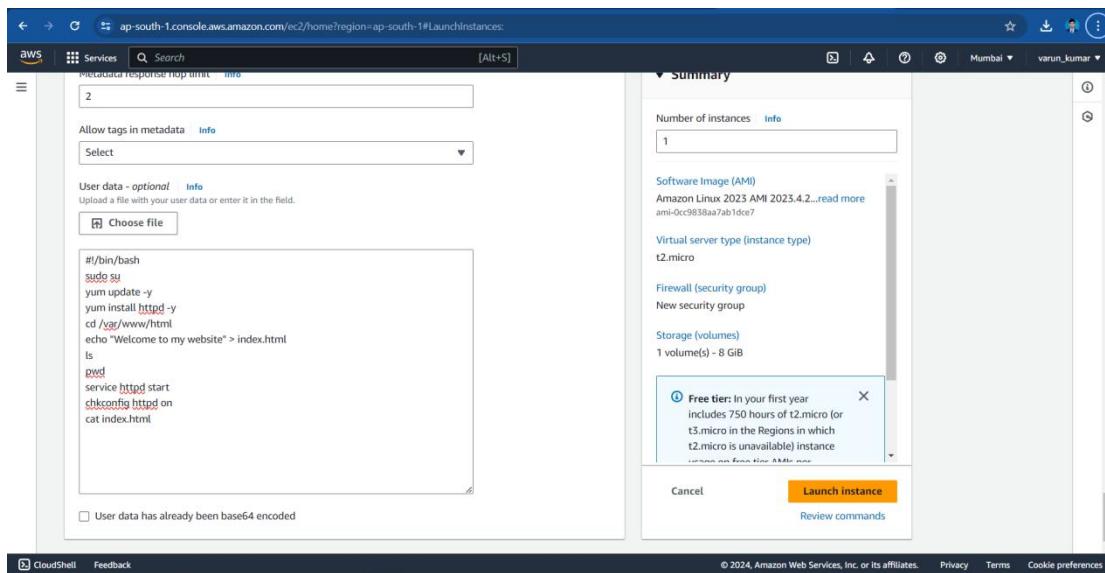
We will now create an instance and attach it on the public subnet on this instance we will host our web server.

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.

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' section, the name is set to 'Webser'. Under 'Application and OS Images (Amazon Machine Image)', a search bar is present, and the 'Quick Start' tab is selected. Below it, there are tabs for 'Amazon Linux', 'macOS', 'Ubuntu', 'Windows', 'Red Hat', and 'SUSE Linux'. The 'Amazon Linux' tab is highlighted. On the right, the 'Summary' section shows 1 instance, using 'Amazon Linux 2023 AMI 2023.4.2...', 't2.micro' instance type, and a new security group. A tooltip for the 'Free tier' is visible. At the bottom, there are 'Cancel', 'Launch instance', and 'Review commands' buttons.

The screenshot shows the 'Network settings' section of the wizard. It includes fields for VPC (selected: 'vpc-01ee8afc5a76095ff (My-VPC)'), Subnet (selected: 'subnet-04d492f16dd0f2f3'), and Auto-assign public IP (set to 'Enable'). Below these, there are sections for Firewall (security groups) and Security group name. A tooltip for the 'Free tier' is visible. The 'Summary' section on the right shows 1 instance, using 'Amazon Linux 2023 AMI 2023.4.2...', 't2.micro' instance type, and a new security group. At the bottom, there are 'Cancel', 'Launch instance', and 'Review commands' buttons.

The screenshot shows the 'Inbound Security Group Rules' section. It lists two rules: 'Security group rule 1 (TCP, 22, 49.36.113.0/32, Admin)' and 'Security group rule 2 (TCP, 80, 0.0.0.0/0, Public)'. Rule 1 has 'Type' as 'ssh', 'Protocol' as 'TCP', and port '22'. Rule 2 has 'Type' as 'HTTP', 'Protocol' as 'TCP', and port '80'. A warning message at the bottom of this section states: '⚠️ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.' The 'Summary' section on the right shows 1 instance, using 'Amazon Linux 2023 AMI 2023.4.2...', 't2.micro' instance type, and a new security group. A tooltip for the 'Free tier' is visible. At the bottom, there are 'Cancel', 'Launch instance', and 'Review commands' buttons.



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

The screenshot shows the AWS EC2 Instances page. A single instance, "Jobify-Primary-instance" (Instance ID: i-02fc7eb0aaba7), is listed as "Running". The instance type is t2.micro, and it has 2/2 checks passed. It is located in the ap-south-1b availability zone with a public IPv4 of 15.207.100.132 and a private IPv4 of 172.31.0.107. The instance has a public IPv6 address of fe80::1%enx0. The Networking tab shows the instance has a private IP DNS name (IPv4 only) of ec2-15-207-100-132.ap-south-1.compute.amazonaws.com and a public IP DNS name of 15.207.100.132.

Select the Ec2 instance connect option and click on connect

The screenshot shows the "Connect to instance" dialog for the instance i-02fc7eb0aaba7. It offers two main connection methods: "EC2 Instance Connect" (selected) and "Session Manager". Under "EC2 Instance Connect", there are two options: "Connect using EC2 Instance Connect" (selected) and "Connect using EC2 Instance Connect Endpoint". The "Connect using EC2 Instance Connect" option is described as connecting using the browser-based client with a public IPv4 address. The "Connect using EC2 Instance Connect Endpoint" option is described as connecting using the browser-based client with a private IPv4 address and a VPC endpoint. Below these, the "Public IP address" is listed as 15.207.100.132. The "Username" field contains "ubuntu". A note at the bottom states: "Note: In most cases, the default username, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI username." At the bottom right are "Cancel" and "Connect" buttons.

Change from EC2 user to root user

The screenshot shows a terminal session on the EC2 instance. The user has switched from the default "ubuntu" user to the "root" user. The terminal output includes system documentation links for Ubuntu, system load statistics, memory usage, swap usage, and a note about Ubuntu Pro. It also mentions expanded security maintenance and ESM Apps. The last login information is shown, followed by the command "sudo su" which switches to the root user. The terminal prompt changes to "#". The session ends with the command "exit" to return to the previous user.

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

```

pm2 status
[output]
0 jobify-primary-ins default 1.0.0 fork 9391 50 5 online 0% 85.0mb ubuntu classified
[output]
tail -15 /tmp/jobify-primey-logs-out.log
[output]
(node:10) SyntaxError: Invalid or unexpected token
at ModuleLeader.moduleStrategy (node:internal/modules/esm/translators:152:18)
at ModuleLeader.moduleProvider (node:internal/modules/esm/loader:298:17)
[output]
SyntaxError: Invalid or unexpected token
at file:///home/ubuntu/Jobify/node_modules/express-sync-errors/index.js:16:20
[output]
at newFn (/home/ubuntu/Jobify/node_modules/express/lib/router/route.js:119:13)
[output]
at next (/home/ubuntu/Jobify/node_modules/express/lib/router/route.js:119:13)
[output]
at middleware (/home/ubuntu/Jobify/node_modules/express-validator/lib/middlewares/check.js:16:13)
[output]
at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
[output]
statusCode: 400
[output]
(node:10) Unauthenticated user provided invalid credentials
[output]
at main (/file:///home/ubuntu/Jobify/controllers/authController.js:23:27)
[output]
at process.processTicksAndRejections (node:internal/process/task_queues:95:5)
[output]
statusCode: 401
[output]

```

i-02fc7ebeeb0aab7 (Jobify-Primary-instance)
PublicIPs: 15.207.109.72 PrivateIPs: 172.31.0.107

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

```

git clone https://www.vipora.com/videos/indian-coupling-2607157utm_source=glas794utm_medium=colon794utm_campaign=colon5

```

i-02fc7ebeeb0aab7 (Jobify-Primary-instance)
PublicIPs: 15.207.109.72 PrivateIPs: 172.31.0.107

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

[EC2](#) > [Security Groups](#) > sg-05d43ce1f35b2c5f1 - launch-wizard-5 > Edit inbound rules

Edit inbound rules [Info](#)

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

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-0d801ffab755aec7b	SSH	TCP	22	Custom	0.0.0.0/0
-	HTTP	TCP	80	Anywhere	web port
-	HTTPS	TCP	443	Anywhere	web port

[Add rule](#) [Cancel](#) [Preview changes](#) [Save rules](#)

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

```
aws Services Search [Alt+S] Mumbai varun.kumar
To see these additional updates run: apt list --upgradable
Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Pending kernel upgrade!
Running kernel version:
 6.8.0-1008-aws
Diagnostics:
  The currently running kernel version is not the expected kernel version 6.8.0-1009-aws.
Last login: Sat Jun 15 10:22:05 2024 from 13.233.177.4
ubuntu@ip-172-31-0-107:~ cd Jobify/
ubuntu@ip-172-31-0-107:~/Jobify$ pm2 start server.js --name "Jobify-primary-ins" -- --port 5100
[PM2] Starting /home/ubuntu/Jobify/server.js in fork_mode (1 instance)
[PM2] Done.

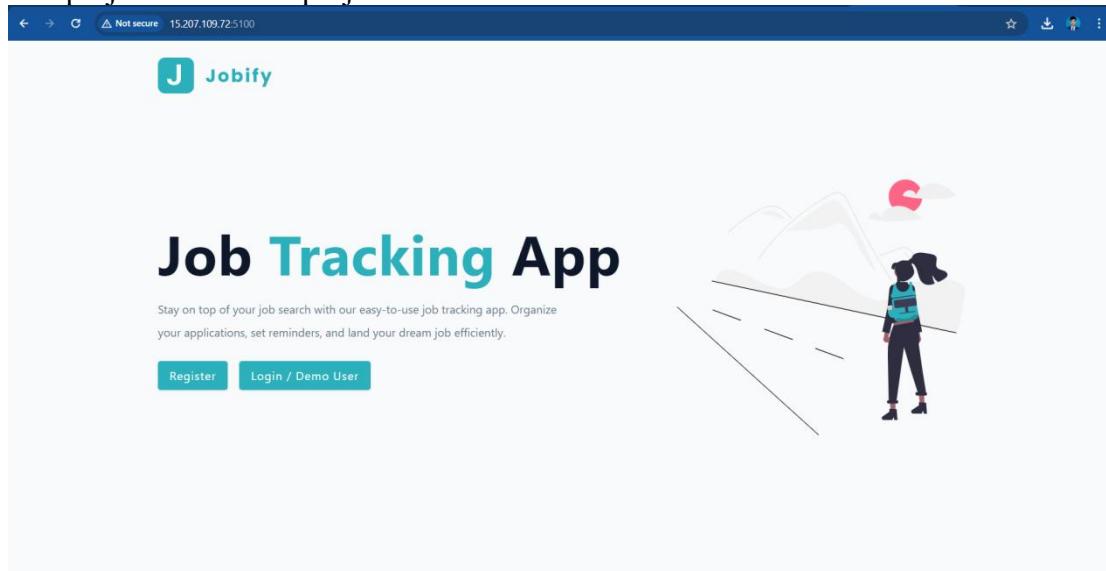
  id  name          namespace  version  mode   pid  uptime  ⚡  status    cpu     mem     user      watching
  0   Jobify-primary-ins  default   1.0.0    fork  7215  0s      0  online   0%   44.4mb  ubuntu  disabled

ubuntu@ip-172-31-0-107:~/Jobify$ pm2 status
  id  name          namespace  version  mode   pid  uptime  ⚡  status    cpu     mem     user      watching
  0   Jobify-primary-ins  default   1.0.0    fork  7215  7s      0  online   0%   92.4mb  ubuntu  disabled

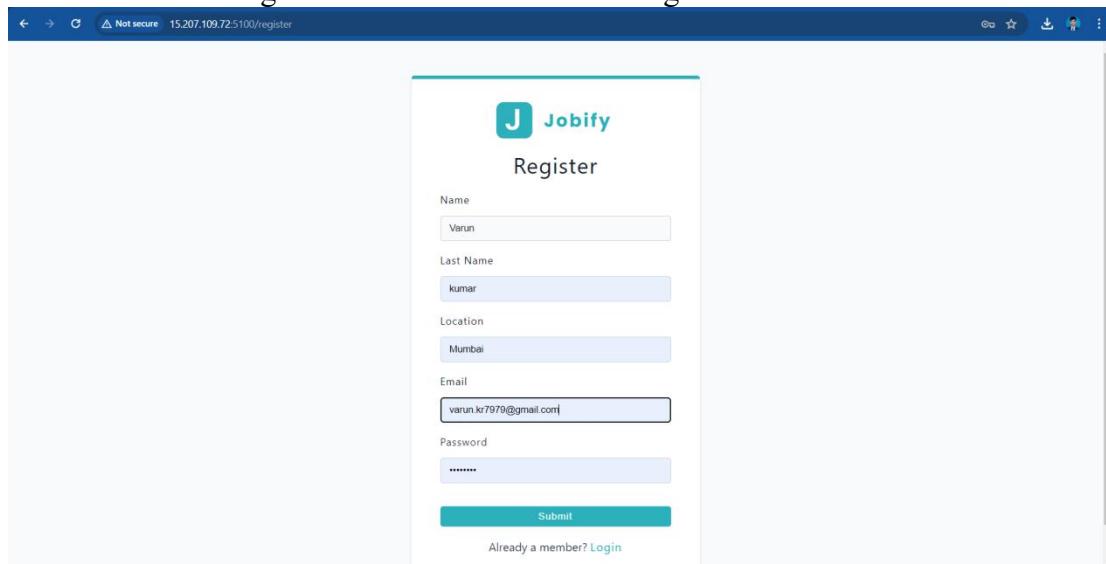
ubuntu@ip-172-31-0-107:~/Jobify$ i-02fcd7eb0aabb7 (Jobify-Primary-instance)
Public IPs: 15.207.100.132 Private IPs: 172.31.0.107

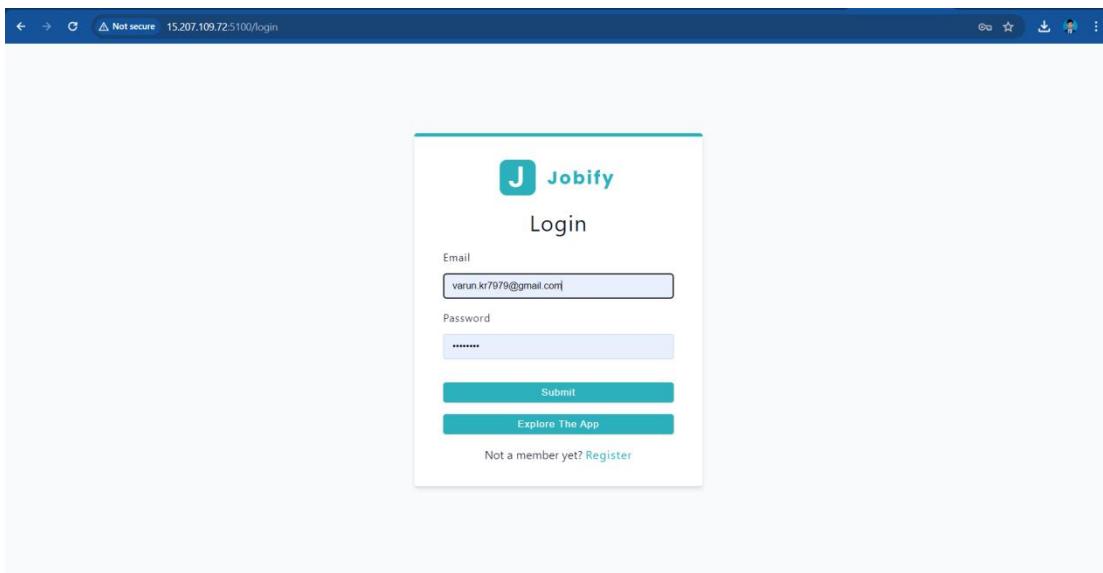
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
```

Our project is live this project contain various modules.

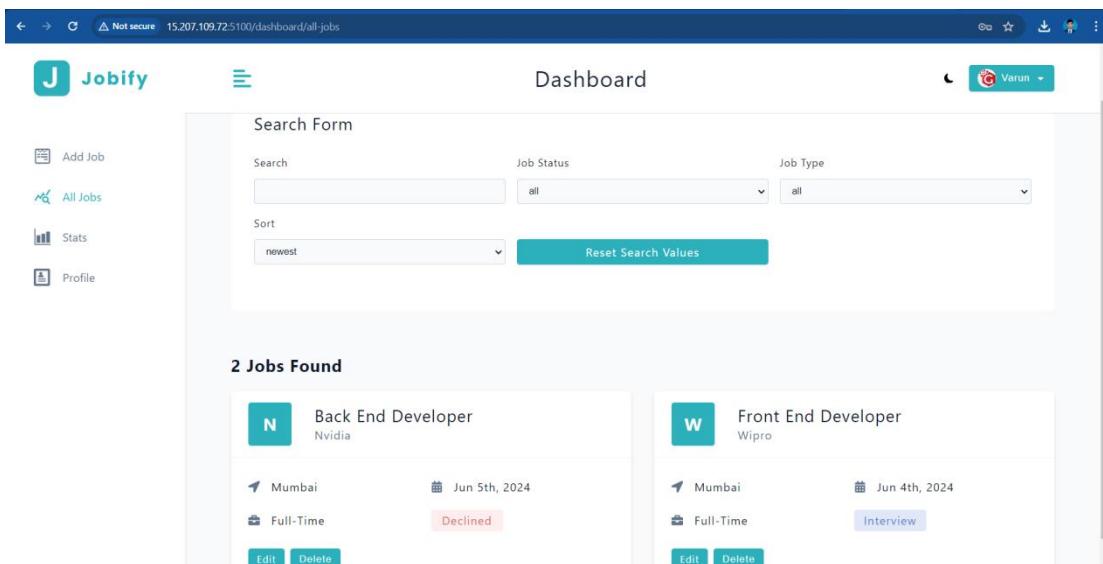
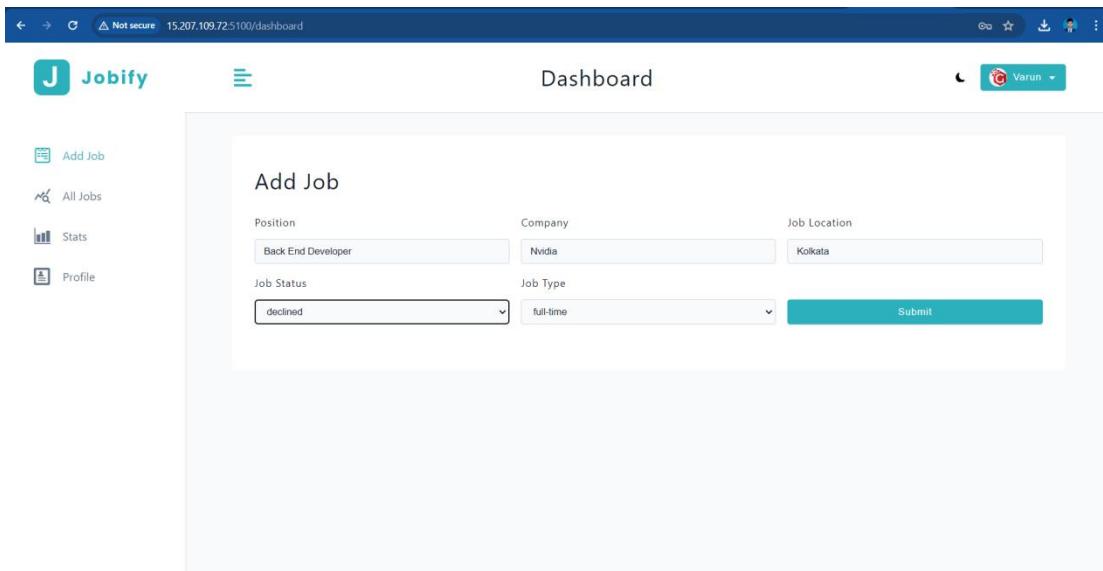


First we have the registered module and then the login module.





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

Search Form

Add Job All Jobs Stats Profile

2 Jobs Found

N Back End Developer Nvidia

Mumbai Jun 5th, 2024

Full-Time Declined

W Front End Developer Wipro

Mumbai Jun 4th, 2024

Full-Time Interview

Edit Delete 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)

Name: Varun

Last Name: kumar

Email: varun.kr797@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.

The screenshot shows the 'Select trusted entity' step of the IAM role creation process. It includes a sidebar with navigation links for Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area has two sections: 'Trusted entity type' and 'Use case'. In 'Trusted entity type', the 'AWS service' option is selected, with a sub-note about allowing actions in the account. Other options include 'AWS account', 'Web identity', 'SAML 2.0 federation', and 'Custom trust policy'. In the 'Use case' section, a dropdown menu shows 'S3' as the selected service. At the bottom, there's a note to choose a use case for the specified service.

The screenshot shows the 'Add permissions' step of the IAM role creation process. It includes a sidebar with navigation links for Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area displays a list of 'Permissions policies (1/926)' with a search bar and a filter for 'All types'. One policy, 'AmazonS3FullAccess', is selected and highlighted. Other policies listed include 'AmazonDMSRedshiftS3Role', 'AmazonS3ObjectLambdaExecutionRole', 'AmazonS3OutpostsFullAccess', 'AmazonS3OutpostsReadOnlyAccess', 'AmazonS3ReadOnlyAccess', 'AWSBackupServiceRolePolicyForS3Backup', 'AWSBackupServiceRolePolicyForS3Recovery', and 'QuickSightAccessForS3StorageManagement'. At the bottom, there's a note to set a permissions boundary (optional).

The screenshot shows the 'Name, review, and create' step of the IAM role creation process. It includes a sidebar with navigation links for Step 1 (Select trusted entity), Step 2 (Add permissions), and Step 3 (Name, review, and create). The main content area has a 'Role details' section where the 'Role name' is set to 'S3AccessRole'. Below it is a 'Description' field containing the text 'Allows S3 to call AWS services on your behalf.'. Under 'Step 1: Select trusted entities', there's a 'Trust policy' section showing a JSON-based policy document:

```
1+ [ {  
2+     "Version": "2012-10-17",  
3+     "Statement": [  
4+         {  
5+             "Effect": "Allow",  
6+             "Principal": "*",  
7+             "Service": "s3.amazonaws.com"  
8+         },  
9+     ]  
10+ }
```

The screenshot shows the AWS IAM Roles page. On the left, there's a sidebar with navigation links like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management', 'Roles', 'Policies', 'Identity providers', 'Account settings', 'Access reports', 'Credential report', and 'Organization activity'. The main content area has a title 'Roles (5) Info' and a table showing five roles: 'AWSServiceRoleForAutoScaling', 'AWSServiceRoleForElasticLoadBalancing', 'AWSServiceRoleForSupport', 'AWSServiceRoleForTrustedAdvisor', and 'S3AccessRole'. Each role entry includes a 'Role name' link, 'Trusted entities' (listing AWS services), and 'Last activity' (e.g., '7 days ago'). Below this is a section titled 'Roles Anywhere' with three options: 'Access from your non AWS workloads' (using X.509 Standard), 'Temporary credentials' (using Certificate Manager Private Certificate Authority), and 'Operate your non AWS workloads using the same authentication and authorization strategy that you use within AWS'.

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.

The screenshot shows the 'Create bucket' page. In the 'General configuration' section, the 'Bucket name' field is filled with 'my-bucket-using-iam-role'. Below it, there's a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button. In the 'Object Ownership' section, there are two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled' (which is described as allowing other AWS accounts to own objects). At the bottom, there's a note about bucket naming rules and a 'See rules for bucket naming' link.

The screenshot shows the 'Block Public Access settings for this bucket' page. It lists several settings under 'Block all public access': 'Block public access to buckets and objects granted through new access control lists (ACLS)', 'Block public access to buckets and objects granted through any access control lists (ACLS)', 'Block public access to buckets and objects granted through new public bucket or access point policies', and 'Block public and cross-account access to buckets and objects through any public bucket or access point policies'. A warning message states: 'Turning off block all public access might result in this bucket and the objects within becoming public. AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.' A checkbox at the bottom accepts this acknowledgement.

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.

Grantee	Objects	Bucket ACL
Bucket owner (your AWS account)	<input checked="" type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input checked="" type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Everyone (public access)	<input type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input type="checkbox"/> Read <input checked="" type="checkbox"/> Write
Authenticated users group (anyone with an AWS account)	<input type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input type="checkbox"/> Read <input checked="" type="checkbox"/> Write
S3 log delivery group	<input type="checkbox"/> List <input checked="" type="checkbox"/> Write	<input type="checkbox"/> Read <input checked="" type="checkbox"/> Write

```

1 "Version": "2012-10-17",
2 "Statement": [
3     {
4         "Effect": "Allow",
5         "Principal": "*",
6         "Action": [
7             "s3:GetObject",
8             "s3:PutObject",
9             "s3>ListBucket"
10        ],
11        "Resource": [
12            "arn:aws:s3:::my-bucket-using-iam-role/*",
13            "arn:aws:s3:::my-bucket-using-iam-role"
14        ]
15    }
16]
17
18
  
```

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.

Access control list (ACL)

Grant basic read/write permissions to other AWS accounts. [Learn more](#)

AWS recommends using S3 bucket policies or IAM policies for access control. [Learn more](#)

Access control list (ACL)

- Choose from predefined ACLs
- Specify individual ACL permissions

Predefined ACLs

- Private (recommended)
Only the object owner will have read and write access.
- Grant public-read access
Anyone in the world will be able to access the specified objects. The object owner will have read and write access. [Learn more](#)

Granting public-read access is not recommended

Anyone in the world will be able to access the specified objects. [Learn more](#)

I understand the risk of granting public-read access to the specified objects.

Properties

Specify storage class, encryption settings, tags, and more.

Cancel **Upload**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Amazon S3

Buckets Access Grants Access Points Object Lambda Access Points Multi-Region Access Points Batch Operations IAM Access Analyzer for S3 Block Public Access settings for this account Storage Lens Dashboards Storage Lens groups AWS Organizations settings Feature spotlight

Amazon S3 > Buckets > my-bucket-using-iam-role

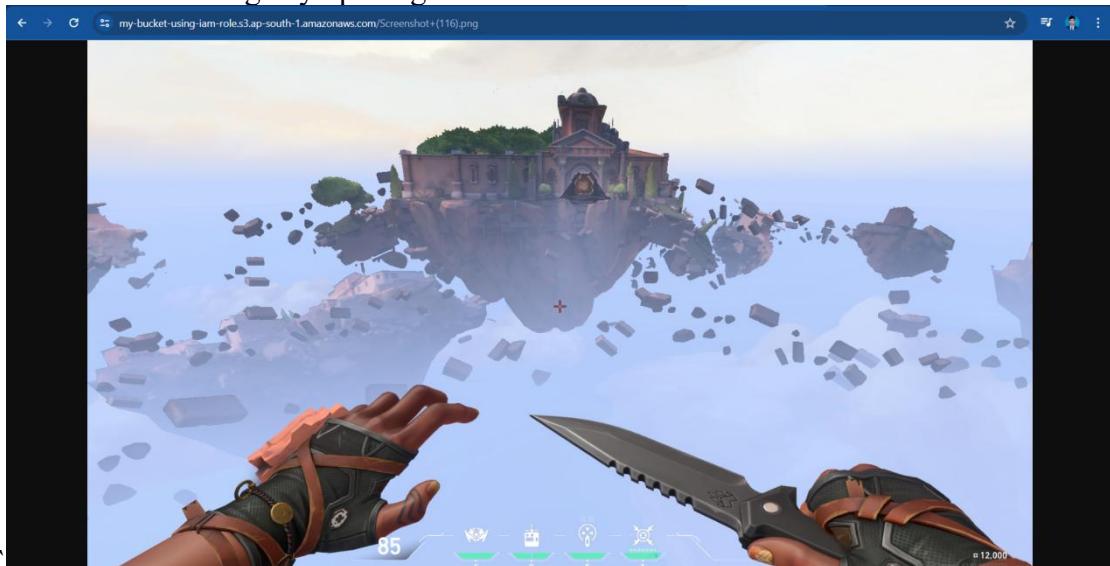
my-bucket-using-iam-role [Info](#)

Objects | Properties | Permissions | Metrics | Management | Access Points

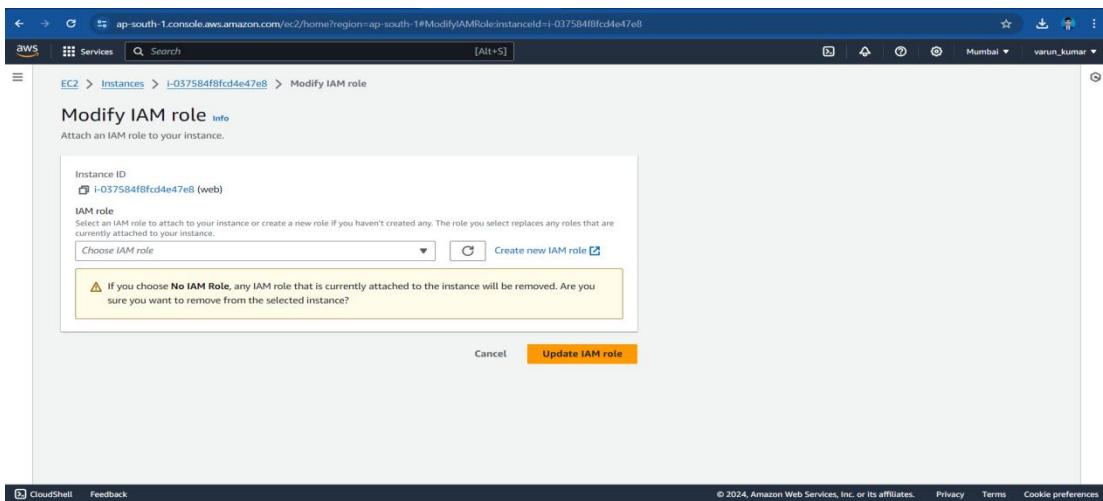
Objects (3) [Info](#)

<input type="checkbox"/>	Name	Type	Last modified	Size	Storage class
<input type="checkbox"/>	Screenshot (116).png	png	May 19, 2024, 01:40:01 (UTC+05:30)	2.3 MB	Standard
<input type="checkbox"/>	Screenshot (117).png	png	May 19, 2024, 01:40:02 (UTC+05:30)	3.1 MB	Standard
<input type="checkbox"/>	Screenshot (118).png	png	May 19, 2024, 01:40:04 (UTC+05:30)	2.8 MB	Standard

We can see the image by opening there URL.

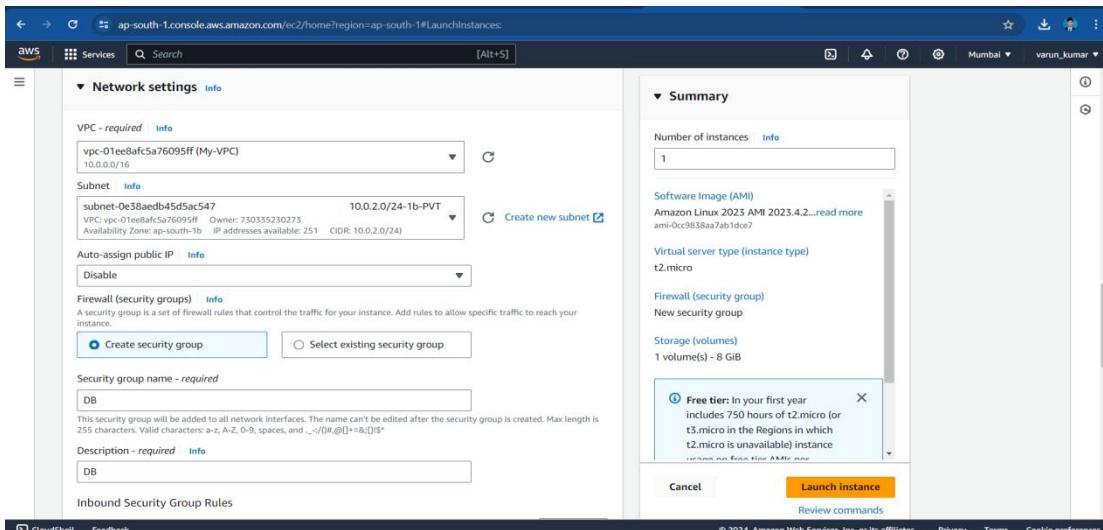
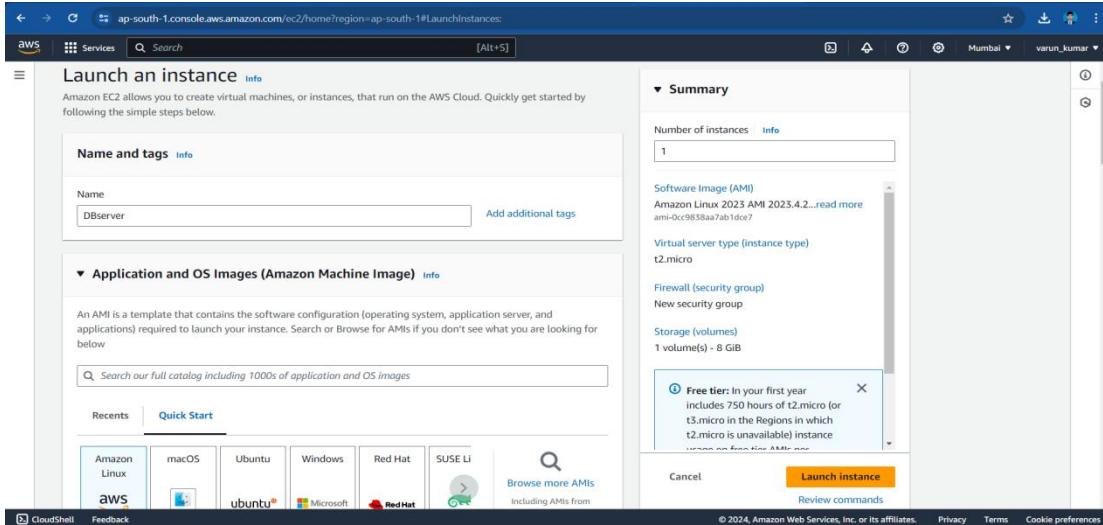


Now we will attach this IAM role with the Amazon S3 full access to our web server instance in the public subnet through the security option and we will choose to modify IAM rule and we will select the IAM role created and will click on update to add the IAM role.



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.

The screenshot shows the 'Launch an instance' wizard in the AWS Management Console. In the 'Name and tags' section, the instance is named 'Jump/Bastion'. Under 'Application and OS Images (Amazon Machine Image)', the 'Quick Start' tab is selected, showing recent AMIs like Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE Linux. A search bar allows finding specific AMIs. The 'Summary' panel on the right shows a configuration for 1 instance with the following details:

- Software Image (AMI): Amazon Linux 2023 AMI 2023.4.2... (ami-0c9858aa7ab1dcf7)
- Virtual server type (instance type): t2.micro
- Firewall (security group): New security group
- Storage (volumes): 1 volume(s) - 8 GiB

A tooltip for the 'Free tier' is visible, stating: 'In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage per month per AMI'. The 'Launch instance' button is highlighted in orange at the bottom right.

The screenshot shows the 'Network settings' step of the 'Launch an instance' wizard. It configures a VPC and subnet. The 'VPC - required' dropdown is set to 'vpc-01ee8afc5a76095ff (My-VPC)'. The 'Subnet' dropdown is set to 'subnet-04d492f16dd0f2f3f'. Under 'Auto-assign public IP', 'Enable' is selected. In the 'Firewall (security groups)' section, there is a 'Create new security group' button. The 'Security group name - required' field is filled with 'jump'. The 'Description - required' field also contains 'jump'. The 'Summary' panel on the right shows the same configuration as the previous screenshot, including the 'Free tier' tooltip. The 'Launch instance' button is again highlighted in orange.

The screenshot shows the 'Launch instances' step of the AWS EC2 wizard. On the left, under 'Inbound Security Group Rules', a single rule is defined: 'Security group rule 1 (TCP, 22, 49.36.113.0/32, Admin)'. The 'Type' is 'ssh', 'Protocol' is 'TCP', and 'Port range' is '22'. The 'Source type' is 'My IP'. The 'Name' is 'Admin' and the 'Description - optional' is 'Admin'. A note below says '49.36.113.0/32'. Below this is a button 'Add security group rule' and a link 'Advanced network configuration'. Under 'Configure storage', it shows '1x 8 GiB gp3 Root volume (Not encrypted)'. A note says 'Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage'. A button 'Add new volume' is available. On the right, the 'Summary' section shows 'Number of instances: 1', 'Software image (AMI): Amazon Linux 2023 AMI 2023.4.2...', 'Virtual server type (instance type): t2.micro', 'Firewall (security group): New security group', and 'Storage (volumes): 1 volume(s) - 8 GiB'. A tooltip for the 'Free tier' note is visible. At the bottom are 'Cancel', 'Launch instance' (which is highlighted in orange), and 'Review commands'.

The screenshot shows the 'Instances' page in the AWS EC2 dashboard. The sidebar includes links like EC2 Dashboard, EC2 Global View, Events, Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, Reservations, Images, AMIs, AMI Catalog, and Elastic Block Store. The main area displays a table of three instances: DBserver, Webserver, and Jump/Bastion, all running on t2.micro instances. The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IPv4 DNS. A modal window titled 'Select an instance' is open at the bottom, listing the same three instances. At the bottom of the page are 'CloudShell' and 'Feedback' buttons, along with copyright information for 2024 and links for Privacy, Terms, and Cookie preferences.

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

The screenshot shows the 'Edit inbound rules' page for a specific security group. The top navigation bar includes 'EC2 > Security Groups > sg-00439910b56d481a3 - DB > Edit inbound rules'. The main area is titled 'Edit inbound rules' and contains a table of two existing rules:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-086c3a11eaad13011	SSH	TCP	22	Custom	10.0.1.32/32 Pvt-IP-Jump
sgr-0342a435ef33eadd9	MySQL/Aurora	TCP	3306	Custom	Q Pub-SN-CIDR 10.0.0.0/24

Buttons at the bottom include 'Add rule', 'Cancel', 'Preview changes', and 'Save rules' (highlighted in orange). The footer includes 'CloudShell', 'Feedback', and standard links for 2024, Privacy, Terms, and Cookie preferences.

The screenshot shows the AWS EC2 Connect to instance page. The instance ID is i-0457e348adba8282 (Jump/Bastion). The SSH client tab is selected. The page provides instructions for connecting via an SSH client, including steps to open the client, locate the private key file (jump.pem), run chmod 400 on it, and connect using the Public IP (13.201.7.92). It also includes an example command: ssh -i "jump.pem" ec2-user@13.201.7.92. A note states that the guessed username is correct but to check if the AMI owner has changed the default AMI username.

The screenshot shows the PuTTY Key Generator application. In the 'Key' section, a public key is displayed for pasting into an OpenSSH authorized_keys file. The key type is ssh-rsa, with a long alphanumeric string. Below it, the 'Key fingerprint' is listed as ssh-rsa 2048 SHA256:4tqTRmHTqZ+TJbJrZdUIU3NoPtgcj7c79pz9mnOa15c. The 'Key comment' is set to imported-openssh-key. The 'Actions' section contains buttons for 'Generate' (disabled), 'Load' (highlighted in blue), 'Save public key', and 'Save private key'. In the 'Parameters' section, the 'Type of key to generate' is set to RSA, and the 'Number of bits in a generated key:' is set to 2048.

The screenshot shows the PuTTY Configuration dialog and a separate 'Select private key file' dialog. The configuration dialog shows the 'SSH' category selected. The 'Private key file for authentication' field is highlighted with a blue border. The 'Select private key file' dialog shows a list of files in the Downloads folder, including jump.pem (PPK File) and Telegram Desktop (File folder). The file name entry field is empty, and there are 'Open' and 'Cancel' buttons at the bottom.

```
[root@ip-10-0-1-32 ~]$ sudo su
[root@ip-10-0-1-32 ec2-user]#
```

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

EC2 > Instances > i-0f73b77166dc632ad > Connect to instance

Connect to instance Info

Connect to your instance i-0f73b77166dc632ad (DBserver) using any of these options

[EC2 Instance Connect](#) | [Session Manager](#) | [SSH client](#) [EC2 serial console](#)

Instance ID
[i-0f73b77166dc632ad \(DBserver\)](#)

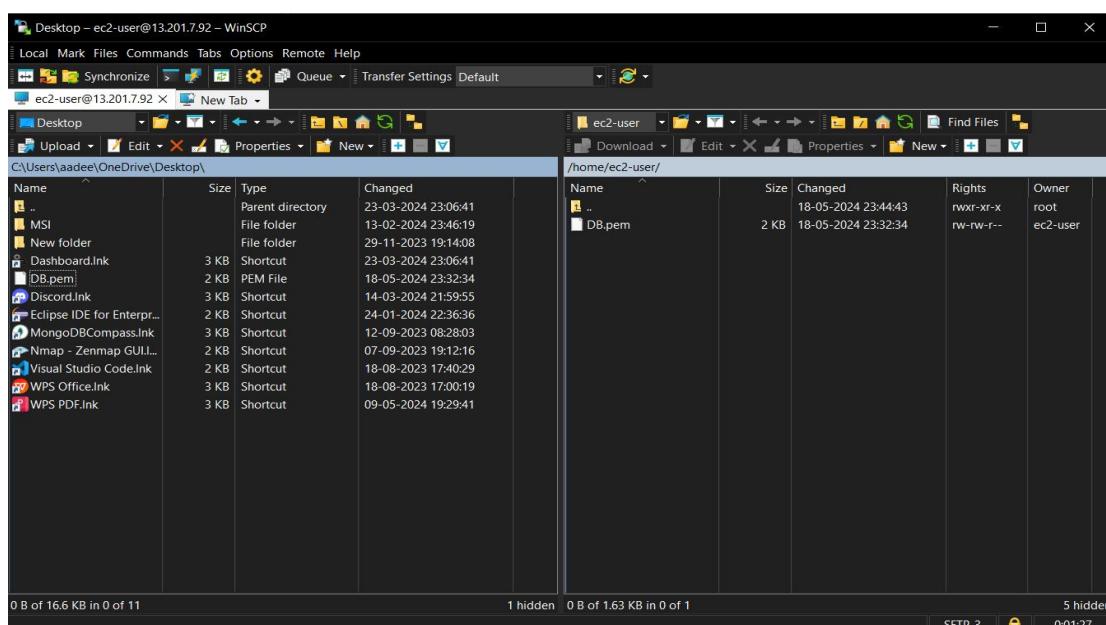
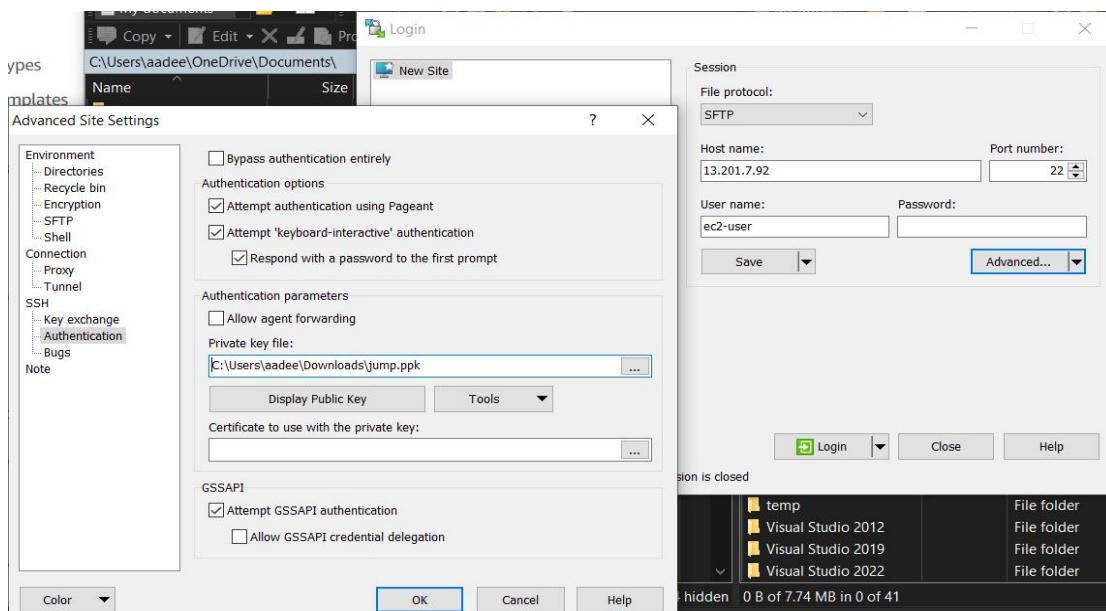
1. Open an SSH client.
2. Locate your private key file. The key used to launch this instance is DB.pem
3. Run this command, if necessary, to ensure your key is not publicly viewable.
 chmod 400 "DB.pem"
4. Connect to your instance using its Private IP:
 10.0.2.89

Example:
 ssh -i "DB.pem" ec2-user@10.0.2.89

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.

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 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.

```
[root@ip-10-0-1-32 ec2-user]# ssh -i "DB.pem" ec2-user@10.0.2.89
      #
      ##_
      Amazon Linux 2023
      _\##_\
      \##_|
      \|##|_
      \#/   https://aws.amazon.com/linux/amazon-linux-2023
      V~'-'>
      /_
      /_
      /m/,'_
[ec2-user@ip-10-0-2-89 ~]$ sudo su
[root@ip-10-0-2-89 ec2-user]# yum install git
Amazon Linux 2023 re      [===[          ] --- B/s | 0 B
```

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:

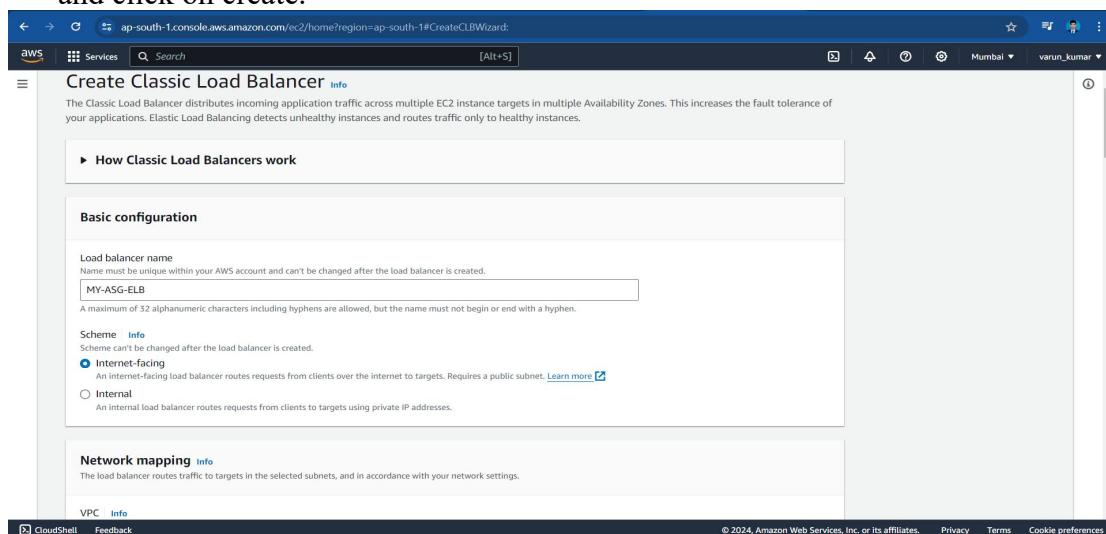
- In the EC2 Auto Scaling console, click on "Auto Scaling Groups" in the sidebar.
- Click on "Create an Auto Scaling group" and follow the wizard.
- Select the launch configuration you created in the previous step.
- Configure the desired minimum, maximum, and desired capacity of instances.
- Configure 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.



The screenshots illustrate the configuration steps for creating a Load Balancer:

- Network mapping**: Selects the VPC (My-VPC) and a subnet (subnet-04d492f16dd0f2f3f) with an IPv4 address range of 10.0.1.0/24.
- Health checks**: Configures the ping target (Ping protocol: HTTP, Ping port: 80, Ping path: /index.html) and advanced health check settings (Response timeout: 2 seconds, Interval: 5 seconds, Healthy threshold: 2).
- Load balancers**: Shows the created load balancer 'MY-ASG-ELB' with details like Type: classic, Status: 0 instances in service, and DNS name: MY-ASG-ELB-585980177.a...

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.

Choose launch template

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

Name

Auto Scaling group name
Enter a name to identify the group.
MY-ASG

Must be unique to this account in the current Region and no more than 255 characters.

Launch template

Choose a launch template that contains the instance-level settings, such as the Amazon Machine Image (AMI), instance type, key pair, and security groups.
MY-ASG-LT

Create a launch template

scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Instance type
t2.micro

Network

For most applications, you can use multiple Availability Zones and let EC2 Auto Scaling balance your instances across the zones. The default VPC and default subnets are suitable for getting started quickly.

VPC
Choose the VPC that defines the virtual network for your Auto Scaling group.
vpc-01eeafca5a76095f (My-VPC)

Availability Zones and subnets
Define which Availability Zones and subnets your Auto Scaling group can use in the chosen VPC.
Select Availability Zones and subnets

ap-south-1a | subnet-04d492f16dd0f2f3f

Create a subnet

Cancel Skip to review Previous **Next**

**Step 2
Choose instance launch options**

Step 3 - optional
Configure advanced options

Step 4 - optional
Configure group size and scaling

Step 5 - optional
Add notifications

Step 6 - optional
Add tags

Step 7
Review

Load balancing

Use the options below to attach your Auto Scaling group to an existing load balancer, or to a new load balancer that you define.

No load balancer
Traffic to your Auto Scaling group will not be fronted by a load balancer.

Attach to an existing load balancer
Choose from your existing load balancers.

Attach to a new load balancer
Quickly create a basic load balancer to attach to your Auto Scaling group.

Attach to an existing load balancer

Select the load balancers that you want to attach to your Auto Scaling group.

Choose from your load balancer target groups
This option allows you to attach Application, Network, or Gateway Load Balancers.

Choose from Classic Load Balancers

Classic Load Balancers

Select Classic Load Balancers

MY-ASG-ELB

Classic Load Balancer

The screenshot shows the 'Create AutoScalingGroup' step 3 of 7. Under 'Health checks', 'EC2 health checks' are enabled. A note says: 'EC2 Auto Scaling will start to detect and act on health checks performed by Elastic Load Balancing. To avoid unexpected terminations, first verify the settings of these health checks in the Load Balancer console.' Below it, 'Turn on VPC Lattice health checks' is also listed.

The screenshot shows the 'Create AutoScalingGroup' step 4 of 7. Under 'Group size', 'Desired capacity type' is set to 'Units (number of instances)' with a value of 3. Under 'Scaling', 'Min desired capacity' is 3 and 'Max desired capacity' is 10. Both are set to 'Equal or less than desired capacity'.

The screenshot shows the 'Auto Scaling groups' details for 'MY-ASG'. It lists the group name, launch template ('MY-ASG-LT Version Default'), and instance counts (3). The 'Edit' button is visible at the top right of the group details section.

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.

Load balancers (1/1)

Name	DNS name	State	VPC ID	Availability Zones	Type	Date
MY-ASG-ELB	MY-ASG-ELB-585980177.a...	-	vpc-01ee8afc5a7609ff	ap-south-1a (aps1-az1)	classic	May

Instances (6) Info

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IP
DBserver	i-0f73b77166dc652ad	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1b	-
WebServer	i-0818928d1f27ef631	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-
Jump/Bastion	i-0457e348afdbba8282	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-
WebServer	i-00665710e8650c483	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-
WebServer	i-07d96aaeaa4d9e480	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-
WebServer	i-04bdd43908fa6463f	Running	t2.micro	2/2 checks passed	View alarms +	ap-south-1a	-

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.

 The name can be up to 256 characters long.

Subnet
 Select a subnet in which to create the NAT gateway.

Connectivity type
 Public
 Private

Elastic IP allocation ID Info
 Assign an Elastic IP address to the NAT gateway.

[► 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.

[CloudShell](#) [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#NatGatewayDetails:natGatewayId=nat-035feac9432f0f855

VPC dashboard X

[VPC](#) > [NAT gateways](#) > nat-035feac9432f0f855 / NAT

[Actions ▾](#)

NAT gateway ID	Connectivity type	State	State message <small>Info</small>
nat-035feac9432f0f855	Public	Pending	-
NAT gateway ARN	Primary public IPv4 address	Primary private IPv4 address	Primary network interface ID
arn:aws:ec2:ap-south-1:730335230273:natgateway/nat-035feac9432f0f855	-	-	-
VPC	Subnet	Created	Deleted
vpc-01ee8afc5a76095ff / My-VPC	subnet-04d492f16dd0f2f3f / 10.0.1.0/24-1a-PUB	Sunday 19 May 2024 at 02:10:14 GMT+5:30	-

[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.			

[CloudShell](#) [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

ap-south-1.console.aws.amazon.com/vpcconsole/home?region=ap-south-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-0c19938d95b03c569

VPC X

[Route tables](#) > [rtb-0c19938d95b03c569](#) > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

<input type="checkbox"/> Name	<input type="checkbox"/> Subnet ID	<input type="checkbox"/> IPv4 CIDR	<input type="checkbox"/> IPv6 CIDR	<input type="checkbox"/> Route table ID
<input type="checkbox"/> 10.0.1.0/24-1a-PUB	subnet-04d492f16dd0f2f3f	10.0.1.0/24	-	rtb-04da7f58c8a51a359 / RT-Custom
<input checked="" type="checkbox"/> 10.0.2.0/24-1b-PVT	subnet-0e38aedb45d5ac547	10.0.2.0/24	-	Main (rtb-0c19938d95b03c569 / RT-Main)

Selected subnets

[subnet-0e38aedb45d5ac547 / 10.0.2.0/24-1b-PVT X](#)

[Cancel](#) [Save associations](#)

[CloudShell](#) [Feedback](#)

© 2024, Amazon Web Services, Inc. or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

VPC dashboard

Route tables (1/3) Info

Name	Route table ID	Explicit subnet associations	Main	VPC
RT-Custom	rtb-04da7f58c8a51a359	subnet-04d492f16dd0f2f5f...	No	vpc-01ee8afc5a76095ff
-	rtb-0205b4d506f710fb7	-	Yes	vpc-01463a85bb642e70
RT-Main	rtb-0c19938d95b03c569	subnet-0e38aedb45d5ac54...	Yes	vpc-01ee8afc5a76095ff

rtb-0c19938d95b03c569 / RT-Main

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

Details

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0c19938d95b03c569	Yes	subnet-0e38aedb45d5ac547 / 10.0.0.0/24-1b-PVT	-
VPC	Owner ID		

VPC > Route tables > rtb-0c19938d95b03c569 > Edit routes

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No

Add route

Cancel | Preview | Save changes

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

```
m/ [root@ip-10-0-2-89 ~]# Last login: Sat May 18 18:42:35 2024 from 10.0.1.32
[root@ip-10-0-2-89 ~]# sudo su
[root@ip-10-0-2-89 ~]# yum install git -y
Amazon Linux 2023 repository           49 MB/s | 24 MB     00:00
Amazon Linux 2023 Kernel Livepatch repository 914 KB/s | 165 KB   00:00
Dependencies resolved.

Transaction Summary
Install 8 Packages

Total download size: 7.1 M
Installed size: 34 M
Downloading Packages:
(1/8): git-2.40.1-1.amzn2023.0.2.x86_64.rpm    718 kB/s | 5.4 kB     00:00
(2/8): perl-Error-0.17029-5.amzn2023.0.2.noarch 2.3 MB/s | 41 kB     00:00
(3/8): perl-File-Find 1.37-477.amzn2023.0.6.noa 1.5 MB/s | 26 kB     00:00
(4/8): git-core-doc-2.40.1-1.amzn2023.0.2.noarc 19 MB/s | 2.6 MB   00:00
(5/8): perl-Git-2.40.1-1.amzn2023.0.2.noarch 1.1 MB/s | 42 kB     00:00
(6/8): perl-lib-0.65-477.amzn2023.0.6.x86_64.rpm 831 kB/s | 15 kB     00:00
(7/8): perl-TermReadKey-2.38.9.amzn2023.0.2.x86_64 1.1 MB/s | 36 kB     00:00
(8/8): git-core-2.40.1-1.amzn2023.0.2.x86_64.rpm 15 MB/s | 4.3 MB   00:00
Total                                         20 MB/s | 7.1 MB  00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
Preparing : 1/1
Installing : git-core-2.40.1-1.amzn2023.0.2.x86_64 1/8
Installing : perl-Error-0.17029-5.amzn2023.0.2.noarch 2/8
Installing : perl-File-Find 1.37-477.amzn2023.0.6.x86_64 3/8
Installing : git-core-doc-2.40.1-1.amzn2023.0.2.noarc 4/8
Installing : perl-Git-2.40.1-1.amzn2023.0.2.noarch 5/8
Installing : perl-lib-0.65-477.amzn2023.0.6.x86_64 6/8
Installing : perl-TermReadKey-2.38.9.amzn2023.0.2.x86_64 7/8
Installing : perl-File-Find-1.37-477.amzn2023.0.6.noarch 8/8
```

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.

Network ACLs (1/2) info

Name	Network ACL ID	Associated with	Default	VPC ID
acl-0bf69d62978c8b306	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

Details

Network ACL ID acl-037f62c57b874def5	Associated with 2 Subnets	Default Yes	VPC ID vpc-01ee8afc5a76095ff / My-VPC
Owner 730355230273			

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.

VPC
VPC to use for this network ACL.

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 <input type="text" value="Name"/>	Value - <i>optional</i> <input type="text" value="NACL-Custom"/>	<input type="button" value="Remove tag"/>
<input type="button" value="Add tag"/>		You can add 49 more tags

Edit subnet associations info

Change which subnets are associated with this network ACL.

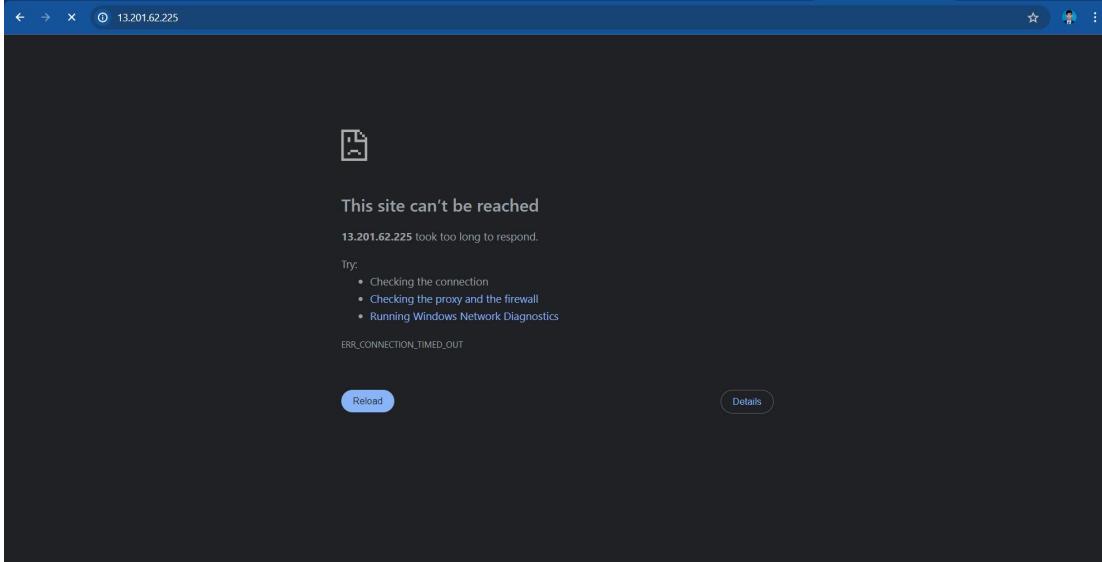
Available subnets (1/2)

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 <input type="button" value="X"/>
--

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



Inbound rules [Info](#)

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

Security group rule ID	Type Info	Protocol Info	Port range Info	Source Info	Description - optional Info
sgr-0bc298a30f5098c32	HTTP	TCP	80	Custom	0.0.0.0/0 Public
sgr-095f0934cbf01c0b1	SSH	TCP	22	Custom	49.36.113.0/32 Admin

Add rule

⚠ Rules with source of 0.0.0.0 or ::/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Cancel [Preview changes](#) [Save rules](#)

VPC > Network ACLs > acl-02d79d3e6fa15d766 / NACL-Custom > Edit inbound rules

Edit inbound rules [Info](#)

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

Rule number Info	Type Info	Protocol Info	Port range Info	Source Info	Allow/Deny Info
100	SSH (22)	TCP (6)	22	49.36.113.0/32	Allow
200	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
300	Custom TCP	TCP (6)	1024-65535	0.0.0.0/0	Allow
*	All traffic	All	All	0.0.0.0/0	Deny

Add new rule [Sort by rule number](#)

Cancel [Preview changes](#) [Save changes](#)

Edit outbound rules Info

Outbound rules control the outgoing traffic that's allowed to leave the VPC.

Rule number	Type info	Protocol info	Port range info	Destination info	Allow/Deny info
100	SSH (22)	TCP (6)	22	49.36.113.0/32	Allow
200	HTTP (80)	TCP (6)	80	0.0.0.0/0	Allow
300	Custom TCP	TCP (6)	1024-65535	0.0.0.0/0	Allow
*	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



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. The study 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 understanding of 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://en.wikipedia.org/wiki/Server_(computing))

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

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

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