

# **CLOUD INFRASTRUCTURE USING AWS**

**by:**  
**Renuka Pulavarthi**

<b>1.Introduction</b>	1
<b>2.Overview</b>	2
<b>3.STEPS</b>	3
<b>3.1. Creating a VPC</b>	3
<b>3.2. Creating Public and Private Subnets</b>	3
<b>3.3. Connecting an Internet Gateway to the VPC</b>	4
<b>3.4. Creating a Route Table</b>	5
<b>3.5. Connecting the Route Table to the Internet Gateway</b>	6
<b>3.6. Creating a NAT Gateway</b>	7
<b>3.7. Connecting the Default Route Table to the Private Subnet and NAT Gateway</b>	9
<b>3.8. Creating Simple Notification Service (SNS)</b>	10
<b>3.9. Creating an S3 Bucket</b>	10
<b>3.10. Creating Flow Logs for the VPC</b>	11
<b>3.11. Creating IAM Role (EC2-S3 Full Access)</b>	11
<b>3.12. Creating EFS</b>	12
<b>3.13. Creating Security Groups</b>	12
<b>3.14. Creating a Load Balancer</b>	13
<b>3.15. Launching EC2 Instances (Using Launch Template)</b>	13
<b>3.16. Auto Scaling Configuration</b>	14
<b>3.17. Creating a Bastion Server</b>	15
<b>3.18. Creating a Database Server</b>	15
<b>3.19. Configuring CloudWatch</b>	16
<b>3.20. Configuring CloudTrail</b>	17
<b>3.21. Creating Network ACL (NACL)</b>	17
<b>4.Testing</b>	18
<b>4.1. Testing Web Server</b>	18
<b>4.2. Testing Load Balancers</b>	18
<b>4.3. Testing Bastion Servers</b>	19
<b>5.Conclusion</b>	20

## 1. Introduction

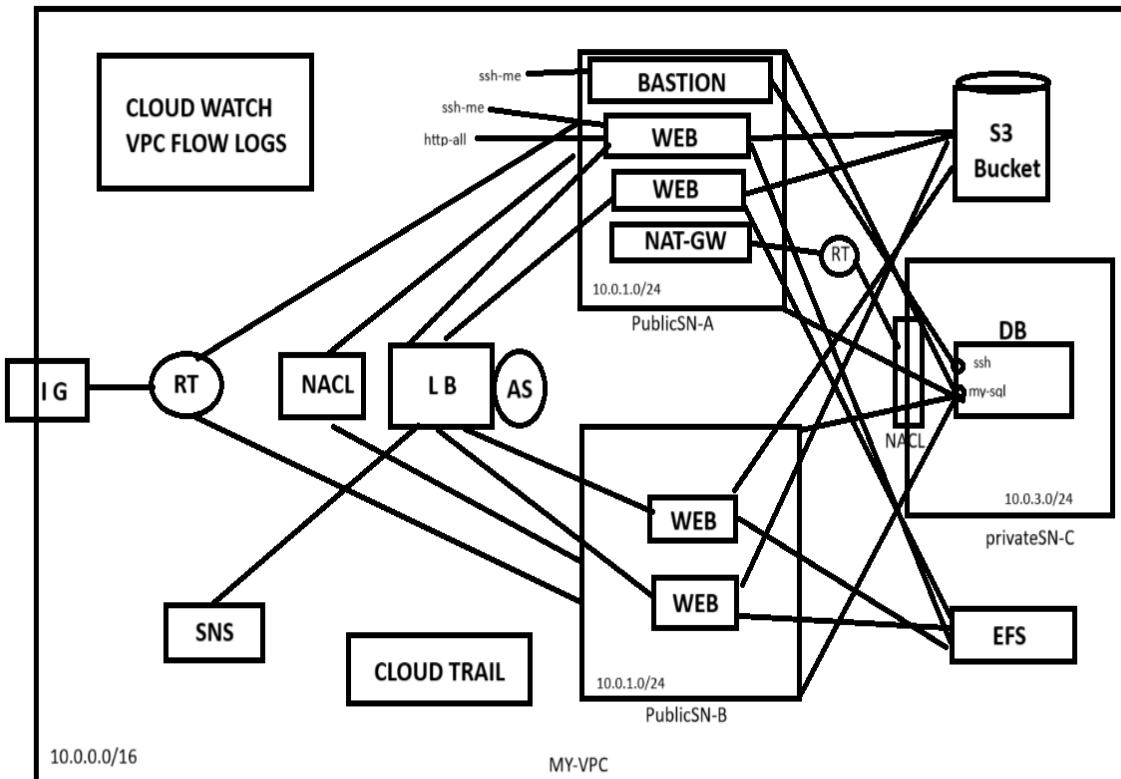
This document details the process of setting up an AWS infrastructure that includes one VPC with two public subnets and one private subnet. The architecture consists of multiple EC2 instances, a Bastion server, a NAT Gateway, a database server, an S3 bucket, EFS, SNS, CloudWatch, CloudTrail, a Load Balancer, Auto Scaling, NACL, a Route Table, and an Internet Gateway. Each component is connected and configured to ensure secure and efficient operations.



This infrastructure is designed to provide:

- **High availability and scalability** using Auto Scaling and Load Balancer.
- **Enhanced security** with Security Groups, NACLs, and Bastion Host.
- **Efficient storage and data management** through S3, EFS, and a dedicated Database Server.
- **Real-time monitoring and logging** with CloudWatch and CloudTrail.
- **Optimized network performance** using NAT Gateway, Route Tables, and Internet Gateway.

## 2.Overview



Structure of the cloud infrastructure

The infrastructure consists of:

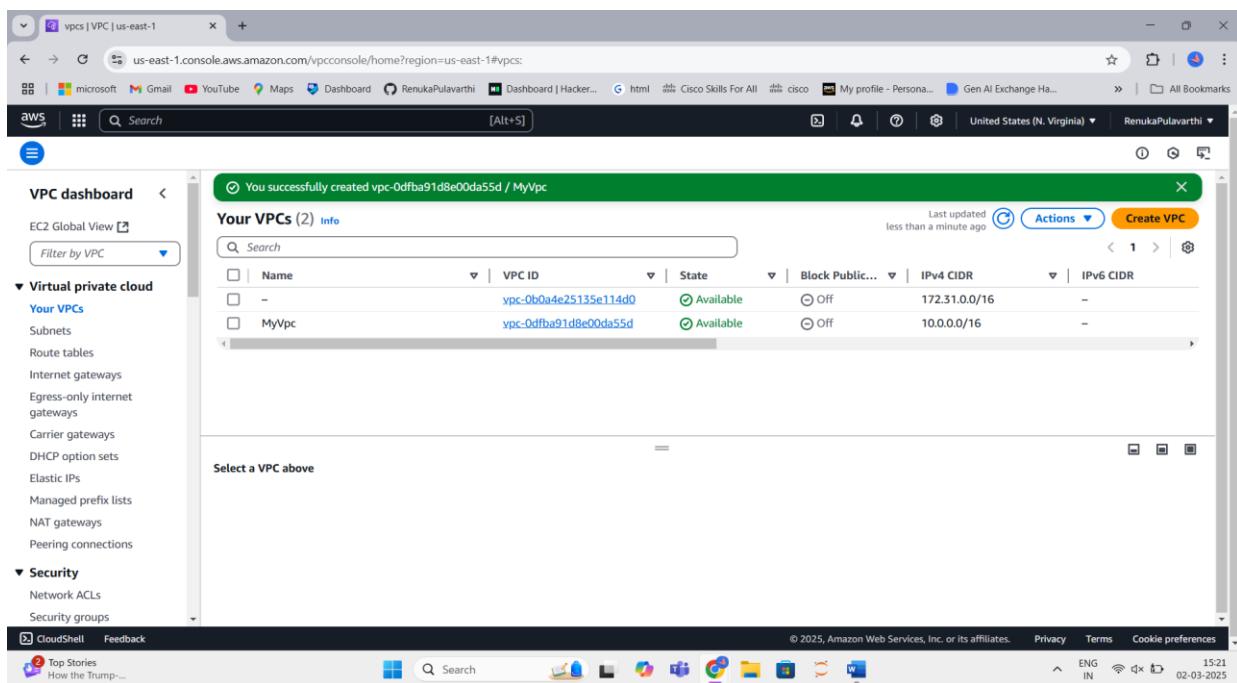
- **1 VPC**
- **2 Public Subnets**
  - **Public Subnet 1:** Contains 2 EC2 instances, a Bastion server, and a NAT Gateway.
  - **Public Subnet 2:** Contains 2 EC2 instances.
- **1 Private Subnet**
  - Hosts a Database Server.
- **Other AWS Resources:**
  - S3 Bucket for storage.
  - EFS for shared file storage.
  - SNS for notifications.
  - CloudWatch for monitoring.
  - CloudTrail for tracking AWS API activity.
  - Load Balancer (LB) connected to Auto Scaling (AS).
  - NACL and Route Table for network management.
  - Internet Gateway for internet access.

## 3 STEPS

### 3.1. Creating a VPC

**Purpose:** A Virtual Private Cloud (VPC) provides an isolated network environment for your AWS resources.

- Navigate to the **AWS VPC Console**.
- Click **Create VPC**.
- Choose **IPv4 CIDR block** (e.g., 10.0.0.0/16).
- Select **default tenancy**.
- Click **Create**.



### 3.2. Creating Public and Private Subnets

**Purpose:** Subnets divide the VPC into smaller network segments, enabling better organization and security.

- In the VPC dashboard, select **Subnets > Create Subnet**.
- Choose the created **VPC**.
- Define CIDR blocks:
  - Public Subnet A: 10.0.1.0/24
  - Public Subnet B: 10.0.2.0/24
  - Private Subnet C: 10.0.3.0/24
- Ensure public subnets have **auto-assign public IP enabled**.

You have successfully created 1 subnet: subnet-06620f9292adfd05a

Name	Subnet ID	State	VPC	Block Public...	IPv4 CIDR Range
-	subnet-0e17f0e06140ac8f5	Available	vpc-0b0a4e25135e114d0	Off	172.31.
-	subnet-0cb766dc8361e2cba	Available	vpc-0b0a4e25135e114d0	Off	172.31.
-	subnet-07333f0db252e5eda	Available	vpc-0b0a4e25135e114d0	Off	172.31.
PublicSN-A	subnet-02a1cc3fe24b79f58	Available	vpc-0dfba91d8e00da55d   MyVpc	Off	10.0.1.0/24
PublicSN-B	subnet-02b61147feac2a2f1	Available	vpc-0dfba91d8e00da55d   MyVpc	Off	10.0.2.0/24
PrivateSN-C	subnet-06620f9292adfd05a	Available	vpc-0dfba91d8e00da55d   MyVpc	Off	10.0.3.0/24

**Edit subnet settings**

**Subnet**

Subnet ID: subnet-02a1cc3fe24b79f58

Name: PublicSN-A

**Auto-assign IP settings**

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

Enable auto-assign customer-owned IPv4 address

**Resource-based name (RBN) settings**

Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

Enable resource name DNS A record on launch

Enable resource name DNS AAAA record on launch

**Hostname type**

Resource name

IP name

### 3.3. Connecting an Internet Gateway to the VPC

**Purpose:** Allows public subnets to access the internet.

- Navigate to **Internet Gateways**.
- Click **Create Internet Gateway**.
- Attach it to the **VPC**.

The following internet gateway was created: igw-0289a28193e4efe97 - MyIGW. You can now attach to a VPC to enable the VPC to communicate with the Internet.

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-0181c125533d8746d	Attached	vpc-0b0a4e25135e114d0	010438465535
MyIGW	igw-0289a28193e4efe97	Detached	-	010438465535

Internet gateway igw-0289a28193e4efe97 successfully attached to vpc-0dfba91d8e00da55d

Internet gateway ID	State	VPC ID	Owner
igw-0289a28193e4efe97	Attached	vpc-0dfba91d8e00da55d   MyVpc	010438465535

**Tags**

Key	Value
Name	MyIGW

### 3.4. Creating a Route Table

**Purpose:** A route table controls the traffic routing between subnets and external resources.

- Go to **Route Tables** in the VPC dashboard.
- Click **Create Route Table** and associate it with the **VPC**.

The screenshot shows the AWS VPC Route Tables page. A success message at the top says "Route table rtb-019ae3324c1bc5813 | MtRT was created successfully." The main section is titled "rtb-019ae3324c1bc5813 / MtRT". It displays the "Details" tab with information about the route table ID (rtb-019ae3324c1bc5813), owner (MyVpc), and VPC (vpc-0dfba91d8e00da55d). Below this are tabs for "Routes", "Subnet associations", "Edge associations", "Route propagation", and "Tags". The "Routes" tab shows one route entry: Destination 10.0.0.0/16, Target local, Status Active, and Propagated No. The bottom of the page includes standard AWS navigation and status bars.

This screenshot shows the AWS VPC Route Tables page with two route tables listed: "rtb-06f8e70e8ee8aa70b" and "rtb-0b027ec596ee63838". Both route tables have "Main" set to Yes and are associated with the VPC "vpc-0b0a4e25135e114" and "vpc-0dfba91d8e00da55d". Below the route tables, a section titled "No subnet associations" indicates that no subnet associations have been explicitly defined. At the bottom, a table lists three subnets without explicit associations: PublicSN-A, PublicSN-B, and PrivateSN-C, each with their respective Subnet IDs and IPv4 CIDRs.

### 3.5. Connecting the Route Table to the Internet Gateway

**Purpose:** Enables internet access for resources in public subnets.

- **Edit the Route Table.**
- **Add a new route:**

- **Destination: 0.0.0.0/0**

- Target: Internet Gateway

- Associate the route table with public subnets.

The screenshot shows the AWS VPC console interface for managing route table associations. The URL in the browser is `us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#EditRouteTableSubnetAssociations:RouteTableId=rtb-019ae3324c1bc5813`. The page title is "Edit subnet associations". The breadcrumb navigation shows "VPC > Route tables > rtb-019ae3324c1bc5813 > Edit subnet associations".

**Available subnets (2/3)**

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> PublicSN-A	<a href="#">subnet-02a1cc3fe24b79f58</a>	10.0.1.0/24	-	Main (rtb-0b027ec596ee63838)
<input checked="" type="checkbox"/> PublicSN-B	<a href="#">subnet-02b61147feac2a2f1</a>	10.0.2.0/24	-	Main (rtb-0b027ec596ee63838)
<input type="checkbox"/> PrivateSN-C	<a href="#">subnet-06620f9292adfd05a</a>	10.0.3.0/24	-	Main (rtb-0b027ec596ee63838)

**Selected subnets**

[subnet-02a1cc3fe24b79f58 / PublicSN-A](#) [subnet-02b61147feac2a2f1 / PublicSN-B](#)

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

CloudShell Feedback © 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Top Stories Israel stops entr... ENG IN 15:31 02-03-2025

The screenshot shows the AWS VPC dashboard. On the left, there's a sidebar with navigation links like 'Virtual private cloud', 'Route tables', 'Internet gateways', 'Egress-only internet gateways', 'Carrier gateways', 'DHCP option sets', 'Elastic IPs', 'Managed prefix lists', 'NAT gateways', and 'Peering connections'. The main content area has a green header bar stating 'You have successfully updated subnet associations for rtb-019ae3324c1bc5813 / Mrt RT.' Below this is a table titled 'Route tables (1/3) Info' with one item listed:

Name	Route table ID	Explicit subnet associ...	Edge associations	Main	VPC
-	rtb-0b277ec596e63838	-	-	Yes	vpc-0dfba91d8e00da5!
Mrt RT	rtb-019ae3324c1bc5813	2 subnets	-	No	vpc-0dfba91d8e00da5!

Below the table, there's a detailed view for 'rtb-019ae3324c1bc5813 / Mrt RT' with tabs for 'Details', 'Routes', 'Subnet associations', 'Edge associations', 'Route propagation', and 'Tags'. The 'Details' tab is selected, showing information such as Route table ID (rtb-019ae3324c1bc5813), Main (No), Explicit subnet associations (2 subnets), and Edge associations (-). Other tabs show 'Routes' (empty), 'Subnet associations' (2 subnets), 'Edge associations' (-), 'Route propagation' (empty), and 'Tags' (empty).

### 3.6. Creating a NAT Gateway

**Purpose:** Allows private subnet instances to access the internet securely.

- Go to **NAT Gateway** in the VPC console.
- Select a public subnet.
- Assign an **Elastic IP**.
- Create the NAT Gateway.

**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**  
Select a connectivity type for the NAT gateway.  
 Public  
 Private

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

**Additional settings** Info

**VPC dashboard** <

**Virtual private cloud**

- Your VPCs
- Subnets
- Route tables**
- Internet gateways
- Egress-only internet gateways
- Carrier gateways
- DHCP option sets
- Elastic IPs
- Managed prefix lists
- NAT gateways
- Peering connections

**Security**

- Network ACLs
- Security groups

**rtb-019ae3324c1bc5813 / MtRT**

**Details** Info

<b>Route table ID</b> <input type="text" value="rtb-019ae3324c1bc5813"/>	<b>Main</b> <input type="text" value="No"/>	<b>Explicit subnet associations</b> <input type="text" value="2 subnets"/>	<b>Edge associations</b> <input type="text" value="-"/>
<b>VPC</b> <input type="text" value="vpc-0dfba91d8e00da55d   MyVpc"/>	<b>Owner ID</b> <input type="text" value="010438465535"/>		

**Routes** Info

Destination	Target	Status	Propagated
0.0.0.0/0	igw-0289a28193e4efe97	<input checked="" type="checkbox"/> Active	No
10.0.0.0/16	local	<input checked="" type="checkbox"/> Active	No

## 3.7. Connecting the Default Route Table to the Private Subnet and NAT Gateway

**Purpose:** Ensures instances in the private subnet can reach the internet via the NAT Gateway.

- Modify the **private route table**.
- Add a new route:
  - Destination: **0.0.0.0/0**
  - Target: **NAT Gateway**

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No
	nat-02932c33939599b3d		

**Edit routes**

Add route

Cancel Preview Save changes

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0b027ec596ee63838	Yes	subnet-06620f9292adfd05a / PrivateSN-C	-
VPC	Owner ID		
vpc-0dfba91d8e00da55d   MyVpc	010438465535		

Updated routes for rtb-0b027ec596ee63838 / DefaultRT successfully

Details

Actions

Routes (2)

Destination	Target	Status	Propagated
0.0.0.0/0	nat-02932c33939599b3d	Active	No
10.0.0.0/16	local	Active	No

## 3.8. Creating Simple Notification Service (SNS)

**Purpose:** Enables automated messaging and alerts for AWS services.

- Go to **SNS** in AWS Console.
- Click **Create Topic**.
- Define a name and create the topic.
- Create subscriptions as needed.

The screenshot shows the AWS SNS console with a subscription named "Subscription: c8f46312-efca-456e-b200-6eec54406a46". The "Details" section displays the ARN (arn:aws:sns:us-east-1:010438465535:MyTeam:c8f46312-efca-456e-b200-6eec54406a46), Endpoint (renukapulavarthi04@gmail.com), Topic (MyTeam), and Subscription Principal (arn:aws:iam::010438465535:root). The status is confirmed, and the protocol is EMAIL. A "Subscription filter policy" tab is visible at the bottom.

## 3.9. Creating an S3 Bucket

**Purpose:** Provides scalable storage for files and backups.

- Navigate to **S3** in AWS.
- Click **Create Bucket**.
- Define the name and region.
- Set appropriate permissions.

The screenshot shows the AWS S3 console with a message indicating a successful bucket creation: "Successfully created bucket 'mybucket-first548'". The "General purpose buckets" tab is selected, showing two buckets: "flowlogs-renukaa" and "mybucket-first548". Both buckets are located in "US East (N. Virginia) us-east-1". The "View Storage Lens dashboard" button is visible. A "Create bucket" button is located at the top right of the table.



## 3.10. Creating Flow Logs for the VPC

**Purpose:** Captures network traffic logs for security and monitoring.

- In the **VPC dashboard**, go to **Flow Logs**.
- Click **Create Flow Log**.
- Choose **VPC** and destination (**S3** or **CloudWatch**).

The screenshot shows the AWS S3 buckets interface. On the left, there's a sidebar for 'Amazon S3' with options like 'General purpose buckets', 'Directory buckets', etc. The main area displays an 'Account snapshot - updated every 24 hours' with a link to 'View Storage Lens dashboard'. Below it, there are tabs for 'General purpose buckets' and 'Directory buckets', with 'General purpose buckets' selected. It shows two buckets: 'flowlogs-renuka' and 'mybucket-first548', both created on March 2, 2025. A 'Create bucket' button is visible at the top right of the list.

## 3.11. Creating IAM Role (EC2-S3 Full Access)

**Purpose:** Grants EC2 instances permissions to interact with S3.

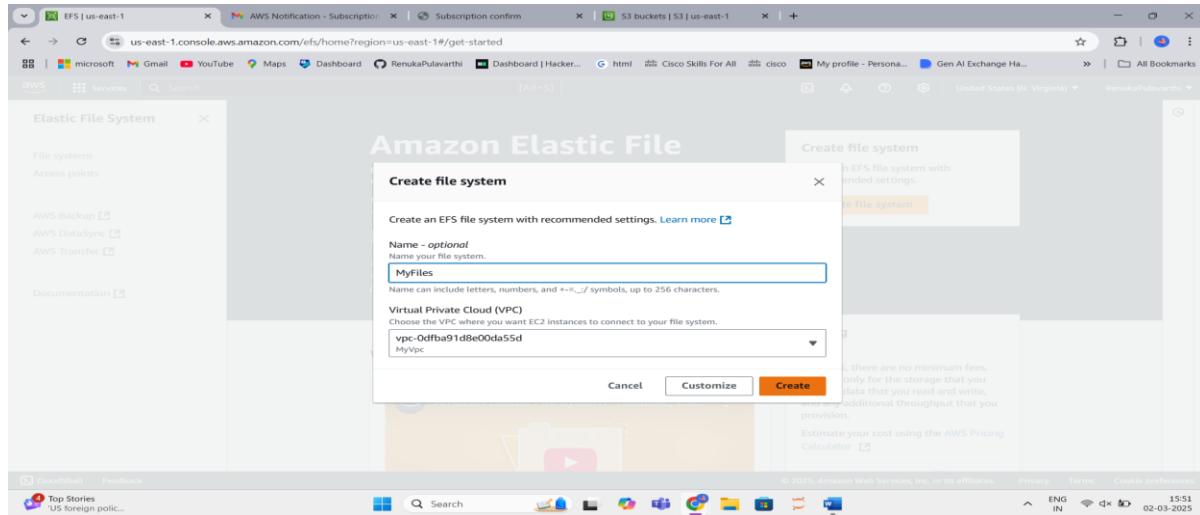
- Go to **IAM > Roles**.
- Click **Create Role**.
- Select **AWS Service > EC2**.
- Attach **AmazonS3FullAccess policy**.

The screenshot shows the AWS IAM Roles interface. The left sidebar includes 'Identity and Access Management (IAM)' and 'Access management' sections. The main area shows a success message 'Role Ec2\_s3FullAccess created.' and a table of existing roles. One role, 'Ec2\_s3FullAccess', is highlighted. At the bottom, there's a section titled 'Roles Anywhere' with a note about authenticating non-AWS workloads and a 'Temporary credentials' button.

## 3.12. Creating EFS

**Purpose:** Provides scalable and shared file storage for multiple EC2 instances.

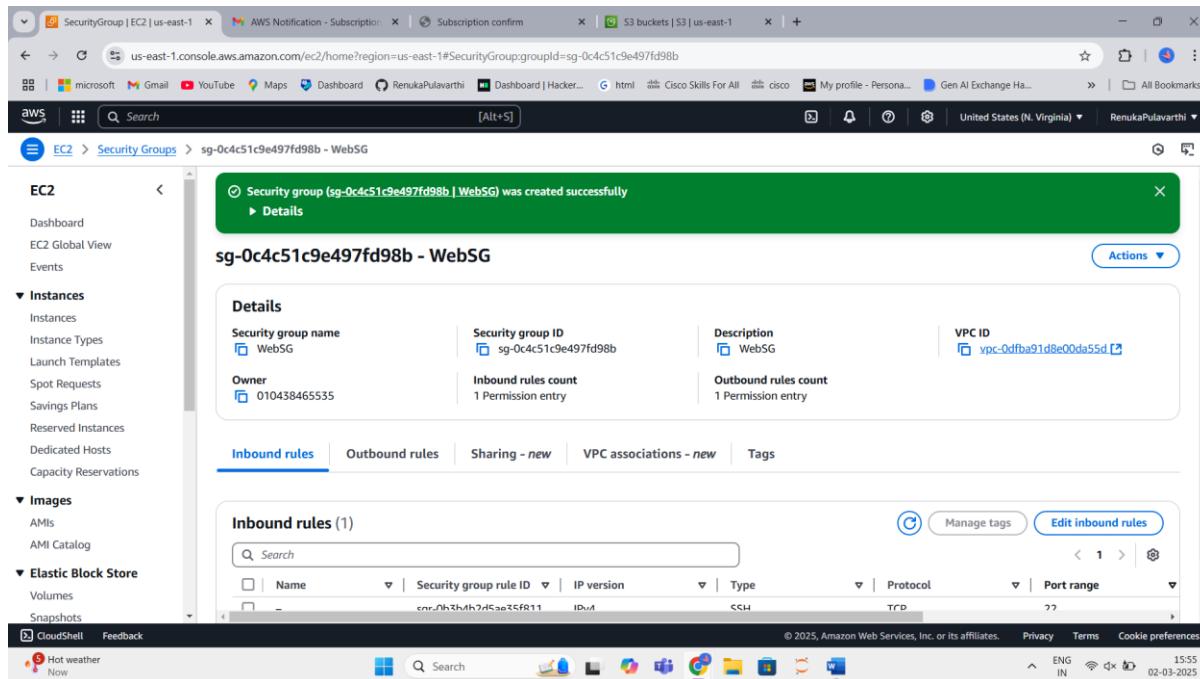
- Navigate to **EFS** in AWS.
- Click **Create File System**.
- Associate with **VPC** and define permissions.



## 3.13. Creating Security Groups

**Purpose:** Controls access to AWS resources based on IP and port rules.

- Go to **Security Groups**.
- Create separate **security groups** for EC2, database, and load balancer with appropriate inbound/outbound rules.



### 3.14. Creating a Load Balancer

**Purpose:** Distributes traffic evenly across multiple EC2 instances.

- Navigate to **EC2 > Load Balancers**.
- Click **Create Load Balancer**.
- Select type (**Application/Network**).
- Assign **subnets, security groups, and target groups**.

The screenshot shows the AWS Cloud Console interface for the EC2 service. On the left, there is a navigation sidebar with sections like 'Images', 'Elastic Block Store', 'Network & Security', 'Load Balancing' (which is currently selected), and 'Auto Scaling'. The main content area displays a table titled 'Load balancers (1)'. The table has columns for Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. A single row is shown for 'MyloadBalancer' with the following details: MyloadBalancer-23999987..., -, vpc-0dfba91d8e00da55d, 2 Availability Zones, classic, March 2, 2025. Above the table, a green success message box states: 'Successfully created load balancer: MyloadBalancer. It might take a few minutes for your load balancer to be fully set up and ready to route traffic. Targets will also take a few minutes to complete the registration process and pass initial health checks.' At the bottom of the page, there are links for 'CloudShell', 'Feedback', and social sharing options, along with the standard AWS footer information.

### 3.15. Launching EC2 Instances (Using Launch Template)

**Purpose:** Provides reusable configurations for instance deployment.

- Go to **EC2 > Launch Templates**.
- Click **Create Template**.
- Define **AMI, instance type, security group, and IAM role**.

The screenshot shows the AWS EC2 Launch Templates page. The left sidebar is collapsed. The main content area displays a table titled "Launch Templates (1) Info". The table has columns for Launch Template ID, Launch Template Name, Default Version, Latest Version, Create Time, and Created By. One entry is listed: "lt-08ee9f48e387595c3" with "MyTemp" as the name, version 1, latest version 1, created on 2025-03-02T10:36:48.000Z, and created by "arn:aws:siam::010438". Below the table, a section titled "Select a launch template" is visible.

### 3.16. Auto Scaling Configuration

**Purpose:** Automatically adjusts the number of running instances based on demand.

- Navigate to **EC2 > Auto Scaling Groups**.
- Click **Create Auto Scaling Group**.
- Choose the launch template.
- Set desired instance count and scaling policies.

The screenshot shows the AWS Auto Scaling groups page. The left sidebar is collapsed. The main content area displays a table titled "Auto Scaling groups (1) Info". The table has columns for Name, Launch template/configuration, Instances, Status, Desired capacity, Min, Max, and Availability Zones. One entry is listed: "MyASG" with "MyTemp | Version Default" as the launch template, 4 instances, and the status "-". The desired capacity is 4, min is 4, max is 10, and the availability zones are "us-east-1a, us-east-1b". Below the table, a message states "0 Auto Scaling groups selected".

**Instances (4) Info**

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone	Public IPv4
WebServer	i-0bd671da46f126b50	Running	t2.micro	Initializing	<a href="#">View alarms</a>	us-east-1b	ec2-5-231-15-
WebServer	i-0240f56e11abcf49	Running	t2.micro	Initializing	<a href="#">View alarms</a>	us-east-1b	ec2-44-223-1l
WebServer	i-0af9299ae46d72034	Running	t2.micro	Initializing	<a href="#">View alarms</a>	us-east-1a	ec2-23-20-18
WebServer	i-06770ec458d2caffb	Running	t2.micro	Initializing	<a href="#">View alarms</a>	us-east-1a	ec2-54-166-2

Select an instance

### 3.17. Creating a Bastion Server

**Purpose:** Securely access instances in the private subnet.

- Launch an **EC2 instance** in a **public subnet**.
- Allow **SSH access** from trusted IPs.
- This acts as a secure gateway to the private network.

### 3.18. Creating a Database Server

**Purpose:** Provides backend data storage for applications.

- Launch an **RDS instance** or an **EC2 database server** in the **private subnet**.
- Configure security group for restricted access.

### 3.19. Configuring CloudWatch

**Purpose:** Enables monitoring and alerting for AWS resources.

- Navigate to **CloudWatch**.
- Set up **logs, metrics, and alarms** for resources.

## 3.20. Configuring CloudTrail

**Purpose:** Records AWS API calls for security and compliance.

- Go to **CloudTrail**.
- Click **Create Trail**.
- Configure logging for **AWS API events**.

The screenshot shows the AWS CloudTrail Event history interface. On the left, there's a navigation sidebar with options like Dashboard, Event history, Insights, Lake (with sub-options like Dashboard, Query, Event data stores, Integrations, Trails), Settings, Pricing, Documentation, Forums, and FAQs. The main area is titled "Event history (50+)" and displays a table of recent events. The table has columns for Event name, Event time, User name, Event source, Resource type, and Resource name. The events listed are: DeleteTrail, StartLogging, CreateTrail, PutEventSelectors, PutBucketEncryption, and PutBucketPolicy, all performed by root user on March 02, 2025, at various times. There are buttons for "Download events" and "Create Athena table". A message banner at the top right says "What's new: Strengthen your data perimeter and implement better detective controls for your VPC endpoints by enabling Network activity events on your Trail or CloudTrail Lake." with a "Learn more" link.

## 3.21. Creating Network ACL (NACL)

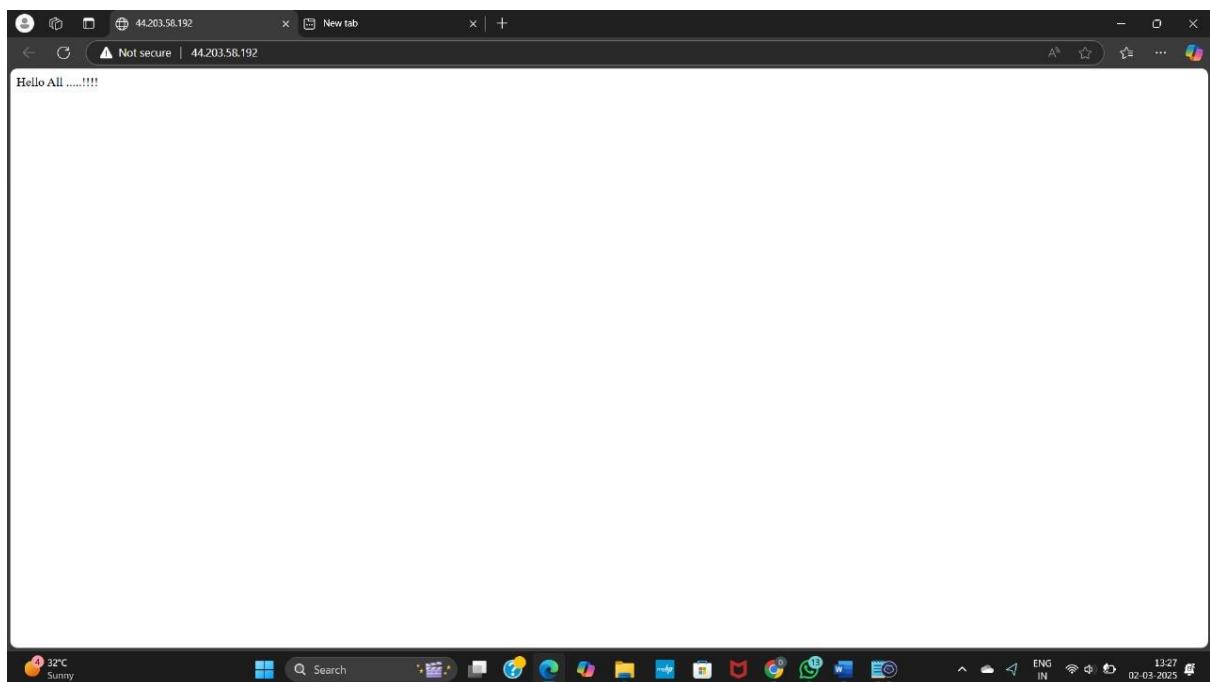
**Purpose:** Controls traffic at the subnet level.

- Navigate to **VPC > Network ACLs**.
- Create and associate it with subnets.
- Define rules for **inbound** and **outbound** traffic.

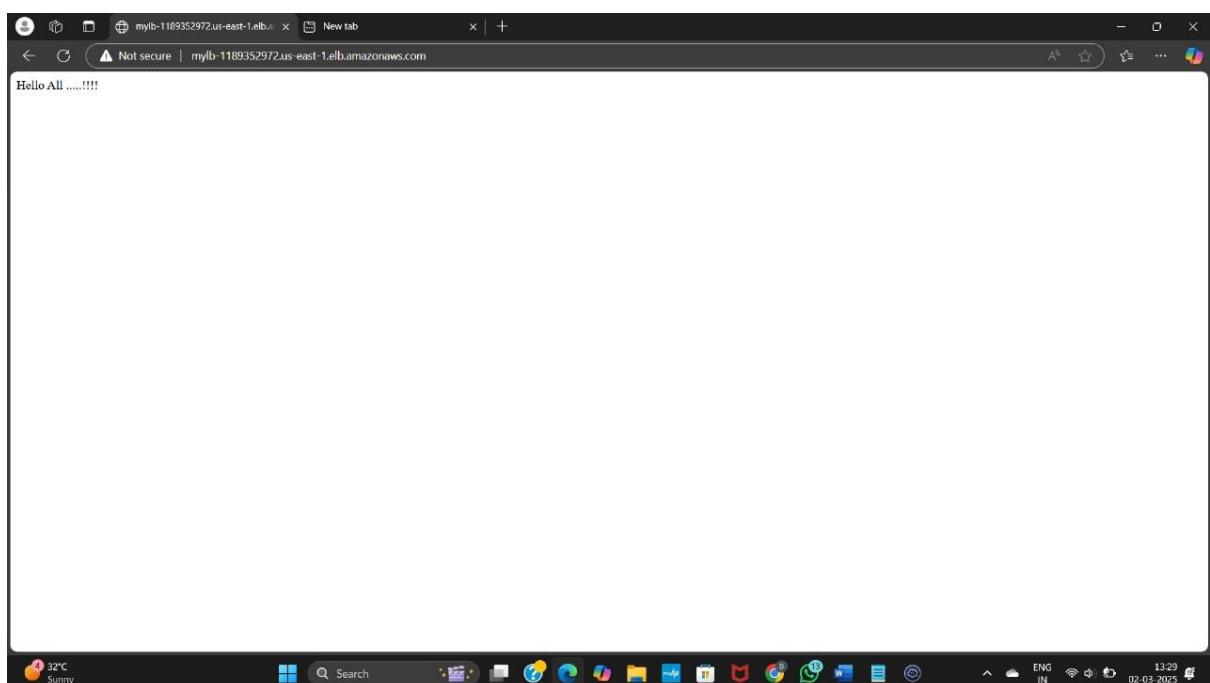
The screenshot shows the AWS VPC dashboard. The left sidebar includes sections for EC2 Global View, Virtual private cloud (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, NAT gateways, Peering connections), Security (Network ACLs, Security groups), and PrivateLink and. The main area shows a table of Network ACLs with one row selected: "publicNACL" (acl-0d35d40c632818a82). The table columns are Name, Network ACL ID, Associated with, Default, VPC ID, and Inbound. The "publicNACL" row has a checked checkbox in the first column. Below the table, a detailed view for "acl-0d35d40c632818a82 / publicNACL" is shown with tabs for Details, Inbound rules, Outbound rules, Subnet associations, and Tags. The Details tab shows the Network ACL ID as acl-0d35d40c632818a82, Associated with 2 Subnets, Default No, and VPC ID vpc-04309df8fcccc9466 / MyVpc.

## 4.Testing

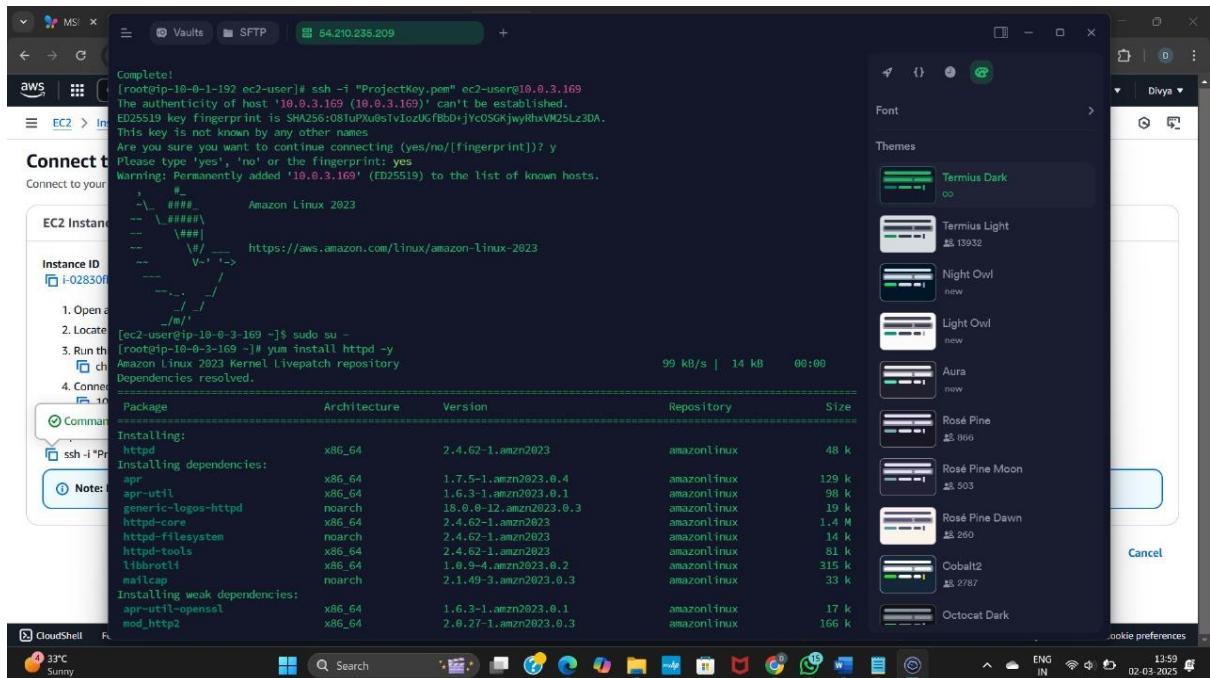
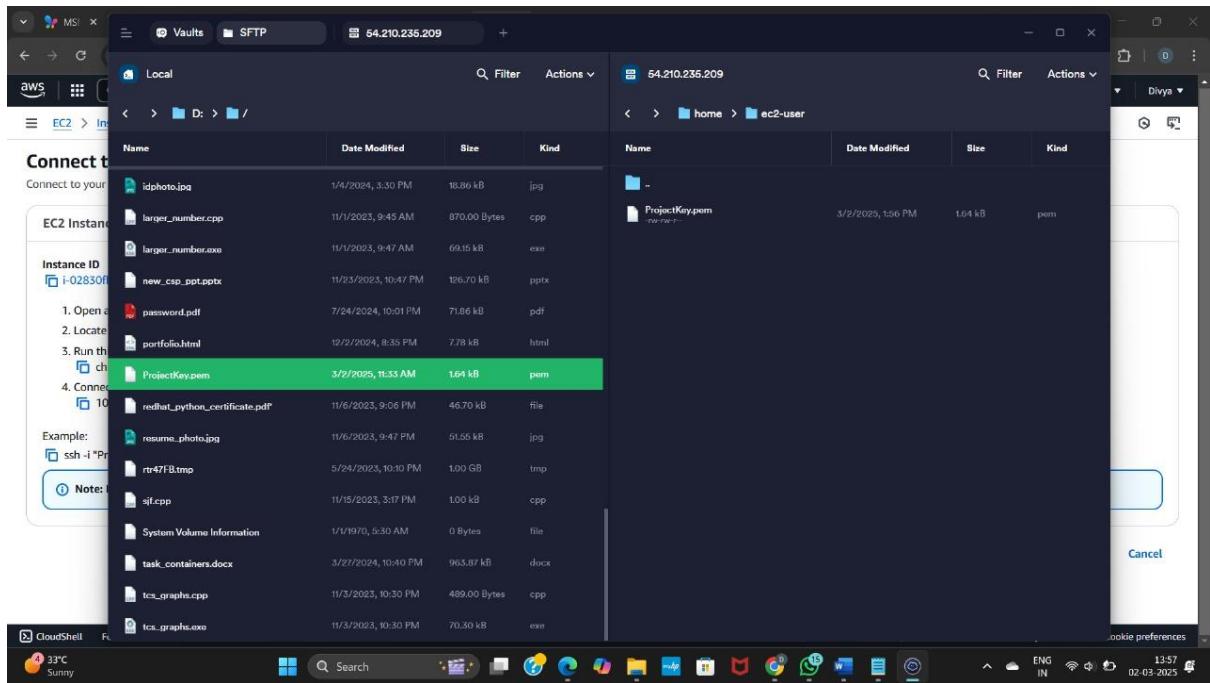
### 4.1. Testing Web Server



### 4.2. Testing Load Balancers



## 4.3. Testing Bastion Servers



## 5.Conclusion

Building a well-architected AWS infrastructure requires careful planning and execution to ensure security, scalability, and high availability. This guide provides a structured approach to creating a robust cloud environment with key components such as VPC, subnets, EC2 instances, load balancing, auto-scaling, and monitoring solutions.

By implementing these steps, organizations can achieve a highly efficient and secure cloud setup, enabling seamless operations while optimizing costs. The combination of CloudWatch, CloudTrail, and IAM policies enhances visibility and control over resources, while services like S3, EFS, and SNS improve data management and communication.

With this AWS infrastructure in place, businesses can focus on innovation and growth while leveraging the power of cloud computing to drive efficiency and reliability. Whether for hosting applications, databases, or enterprise solutions, this setup provides a solid foundation for cloud-based success.