

Highly Available and Scalable Web Server Infrastructure

Phase 1: Public EC2 for Web App and Database

Task 1.1: Setting Up EC2 for Web App and Database

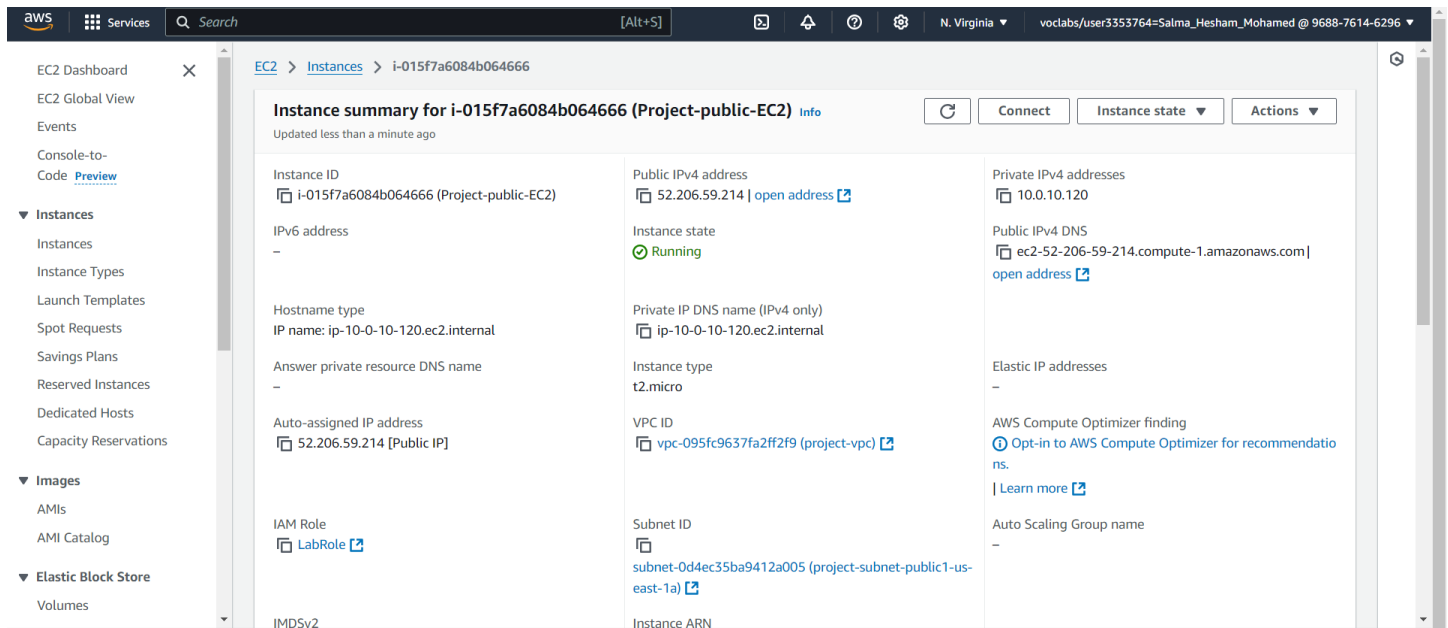
Description: A public EC2 instance was used to host both the web application and the database, serving as the foundational structure.

Steps:

1. Created an EC2 instance via AWS Management Console/CLI.
2. Configured the instance to host both the web app and the database.

Tools:

AWS EC2, AWS Management Console, AWS CLI



Phase 2: Building the Core Infrastructure

Task 2.1: Creating VPC and Subnets

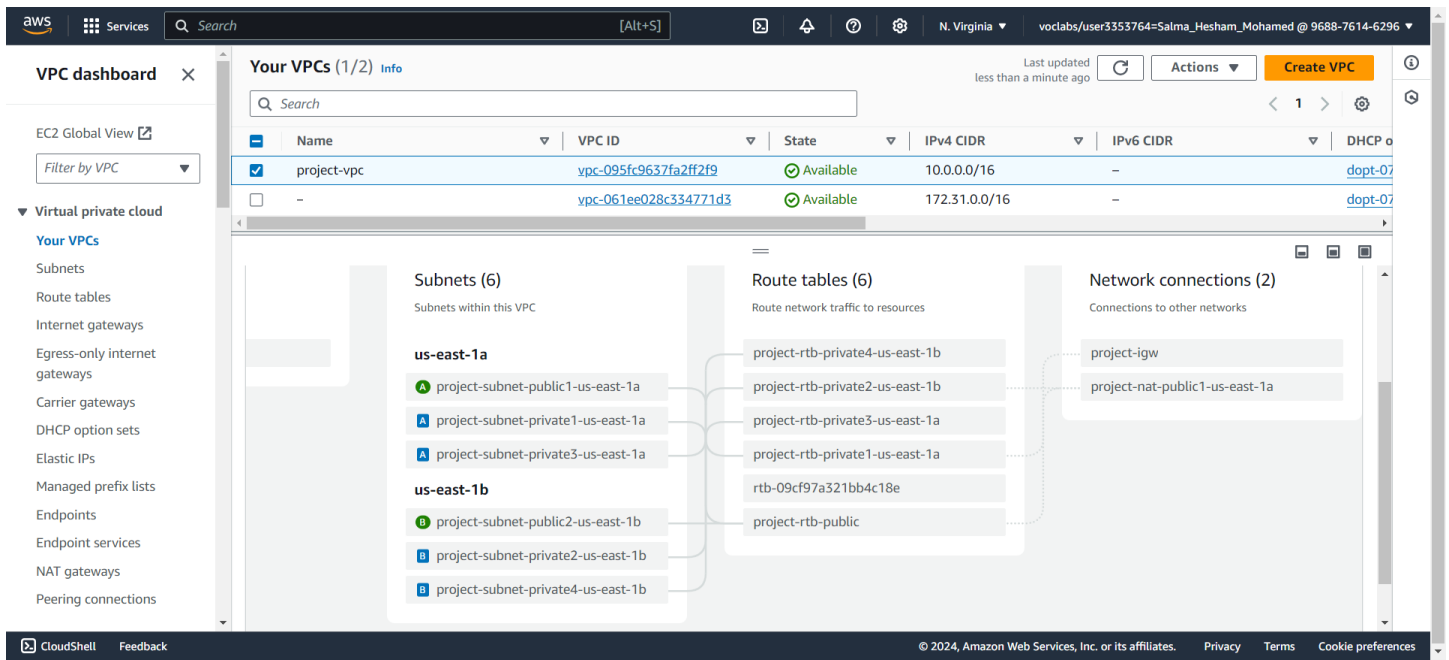
Description: Created a VPC with two public subnets and two private subnets for separating external and internal resources.

Steps:

1. Created a VPC via AWS Management Console/CLI.
2. Configured two public and two private subnets within the VPC.

Tools:

AWS VPC, AWS Management Console, AWS CLI



Task 2.2: Configuring Auto Scaling Group

Description: Configured Auto Scaling Group for automatic scaling of EC2 instances based on traffic demands.

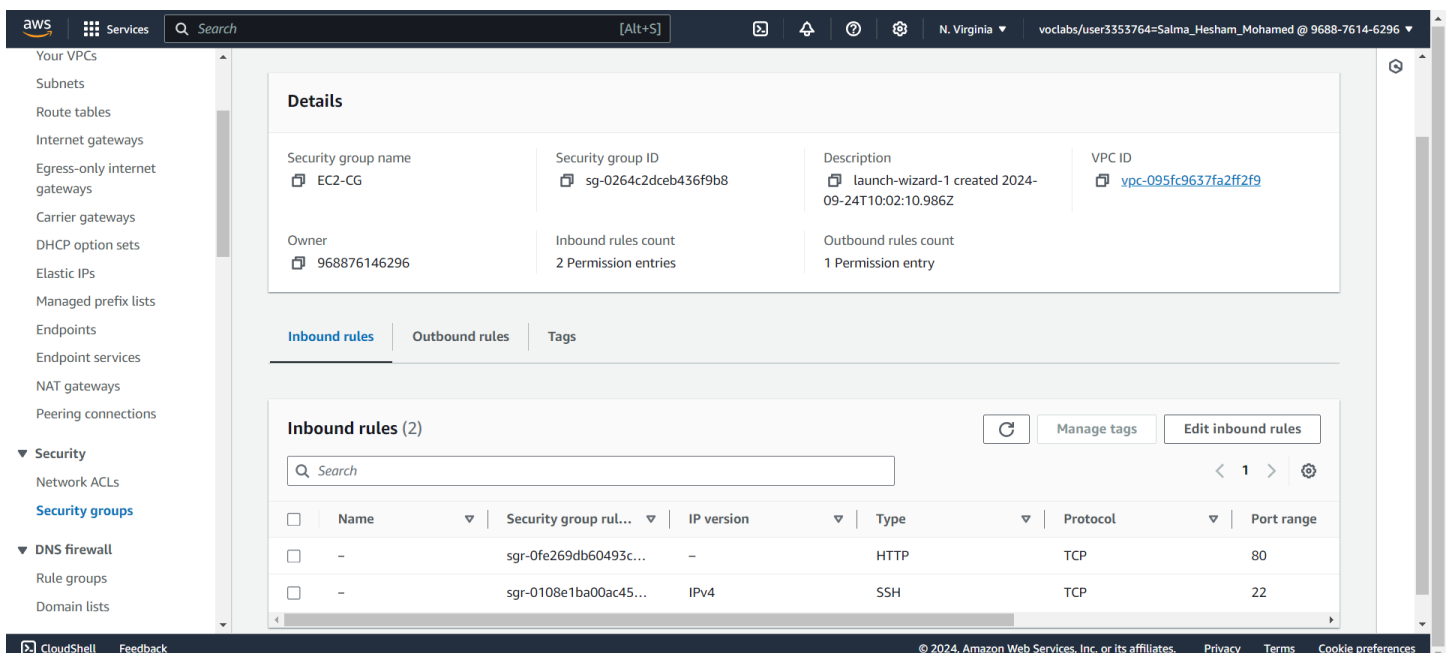
Steps:

1. Set up an Auto Scaling Group with desired: 2, min: 2, max: 4.
2. Deployed EC2 instances in public subnets to host the web app.

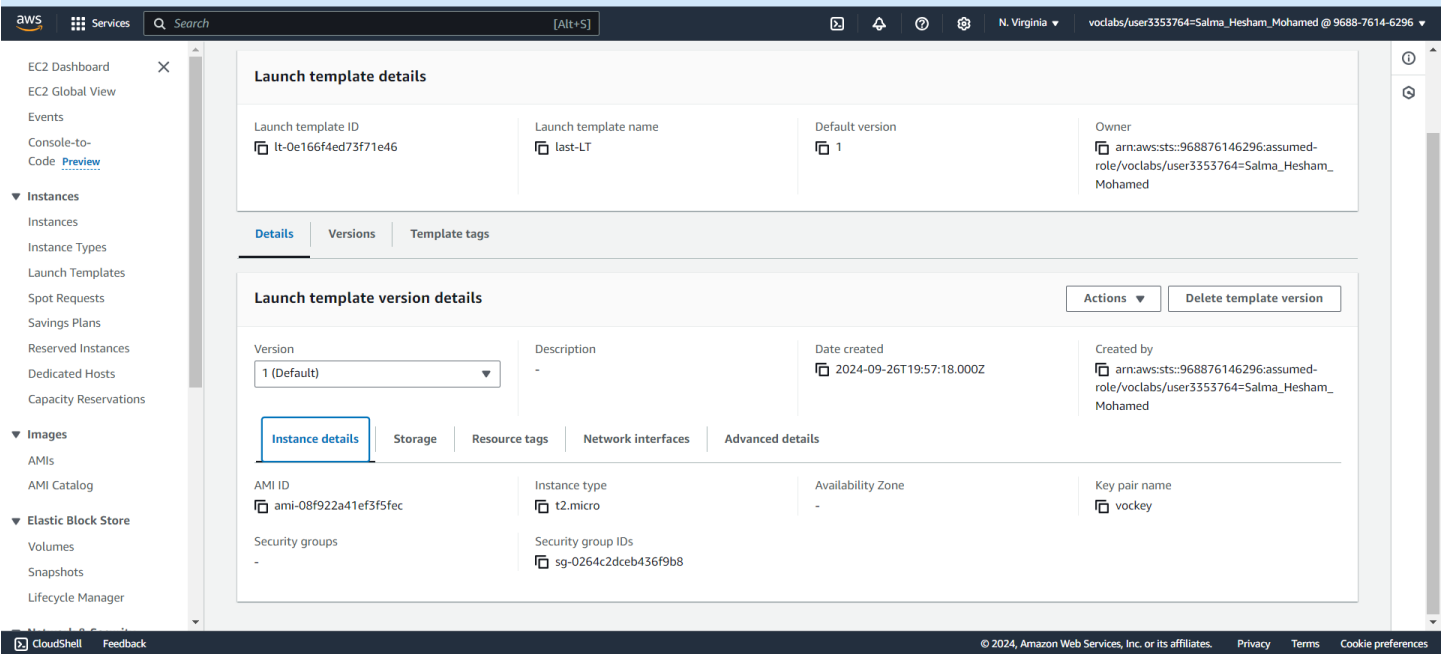
Tools:

AWS EC2, Auto Scaling Group, AWS Management Console, AWS CLI

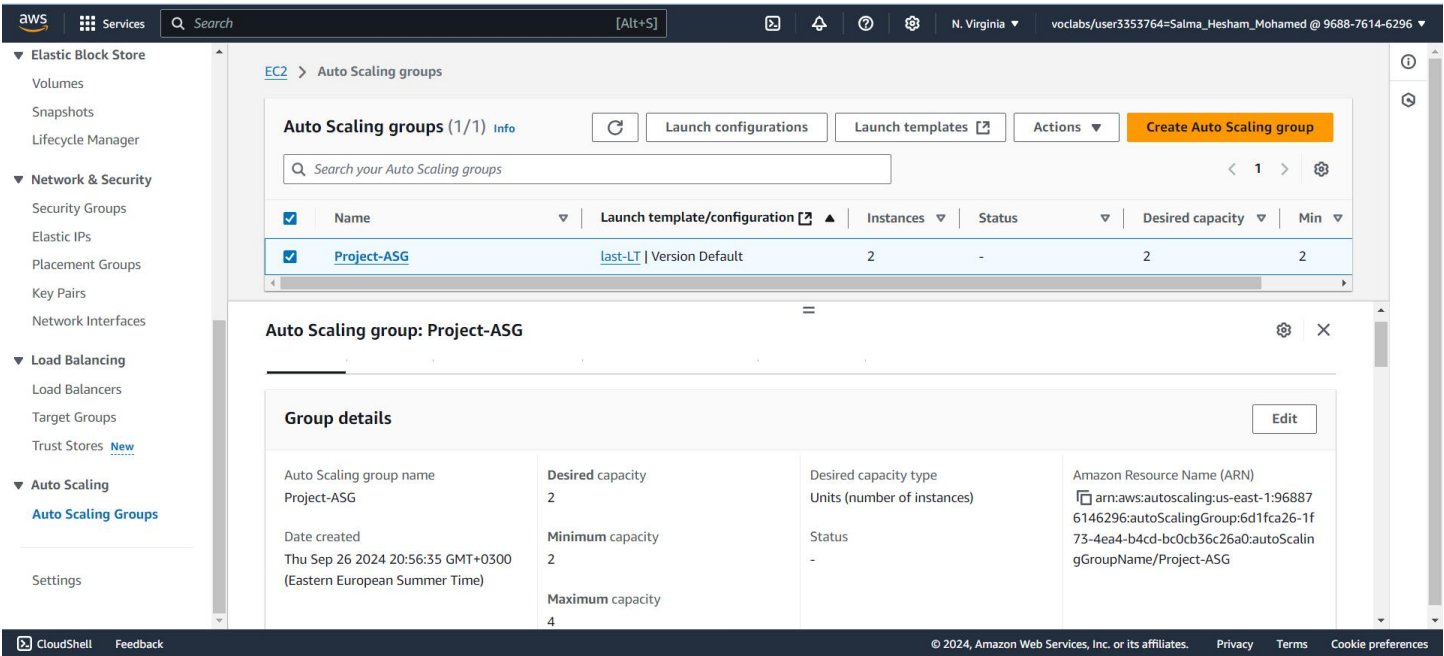
Configuring EC2 SG



Configuring the Launch Templet



Configuring the Auto Scaling Group



Task 2.3: Setting Up Application Load Balancer (ALB)

Description: Configured an ALB for distributing incoming traffic across multiple EC2 instances.

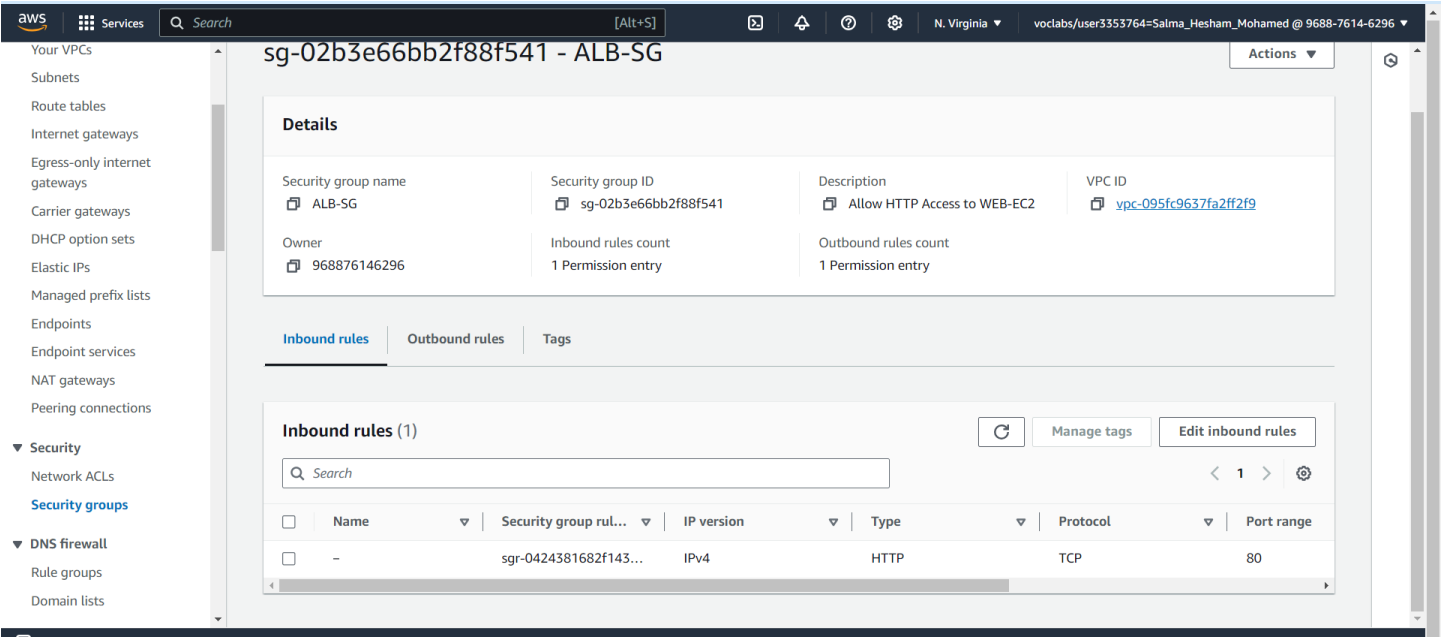
Steps:

- 1. Created an ALB and assigned it to the public subnets.
- 2. Configured security groups for ALB to allow HTTP/HTTPS traffic.

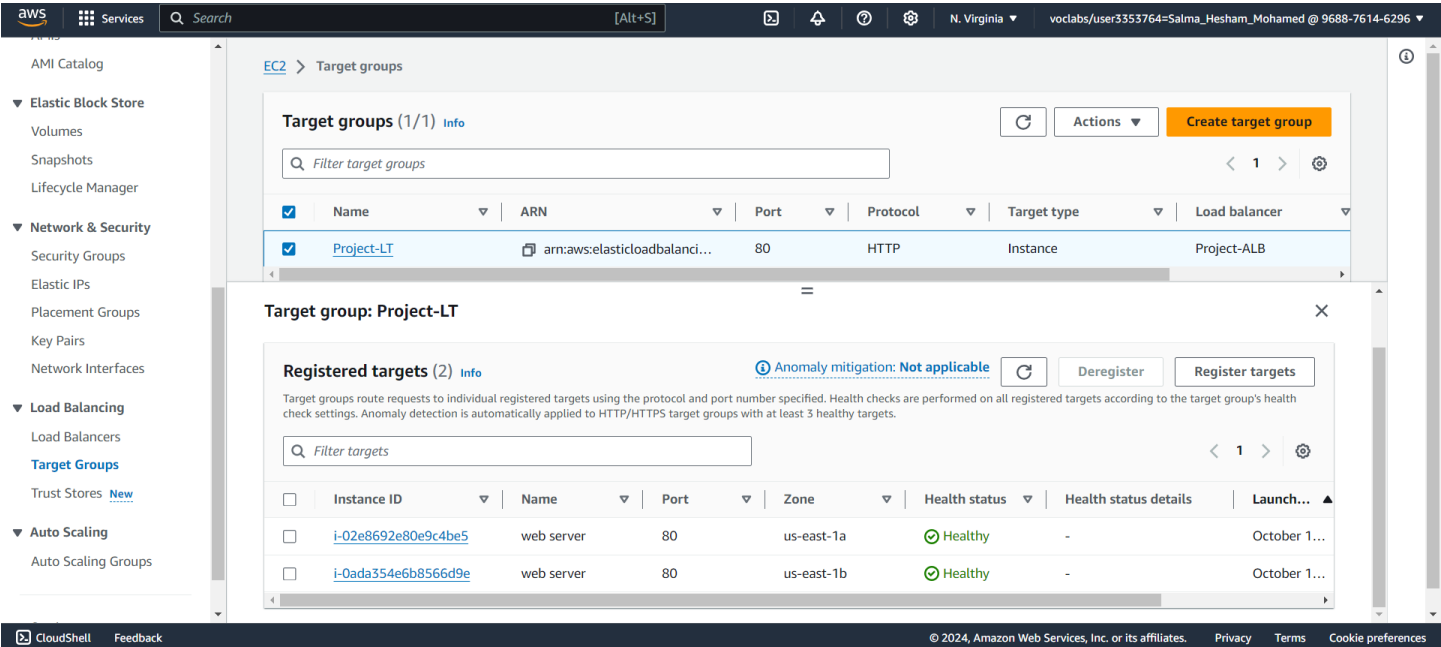
Tools:

AWS ALB, AWS Management Console

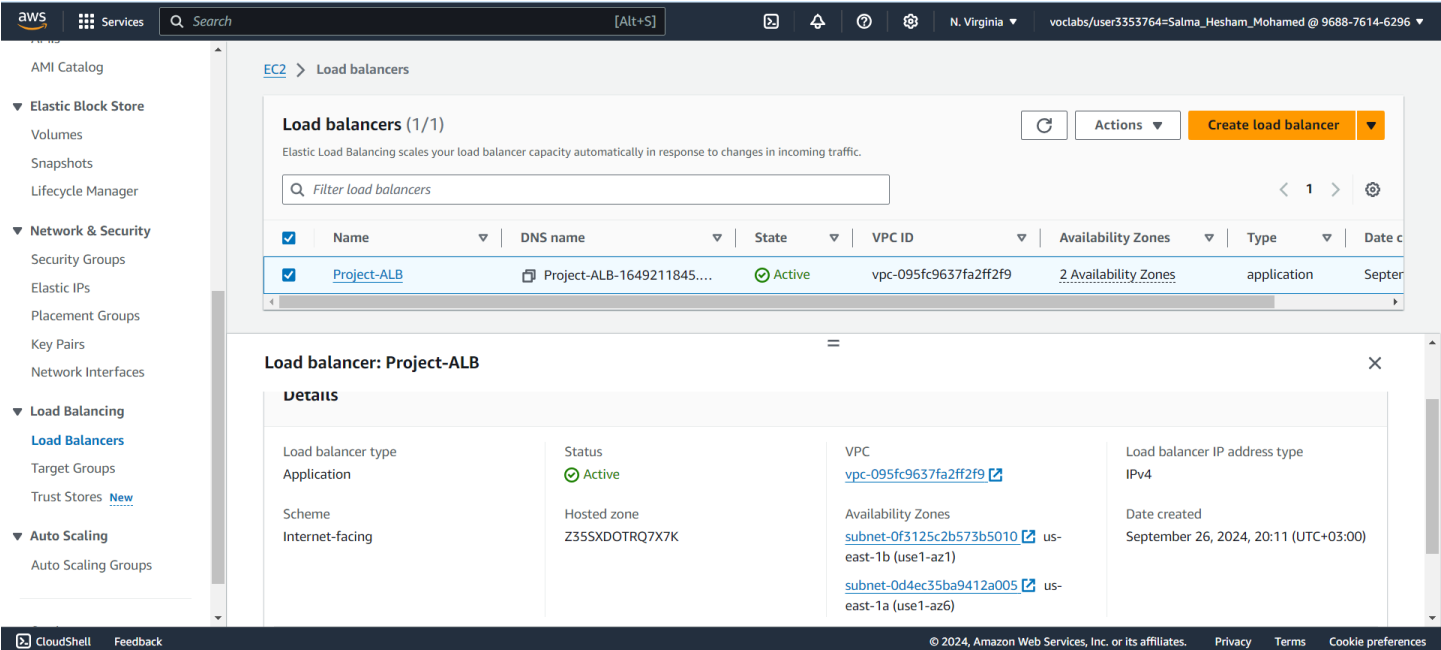
Configuring SG for Application Load Balancer



Configuring Target Groups for Application Load Balancer



Configuring the Application Load Balancer



Phase 3: Database Setup and Security

Task 3.1: Configuring MySQL RDS in Private Subnets

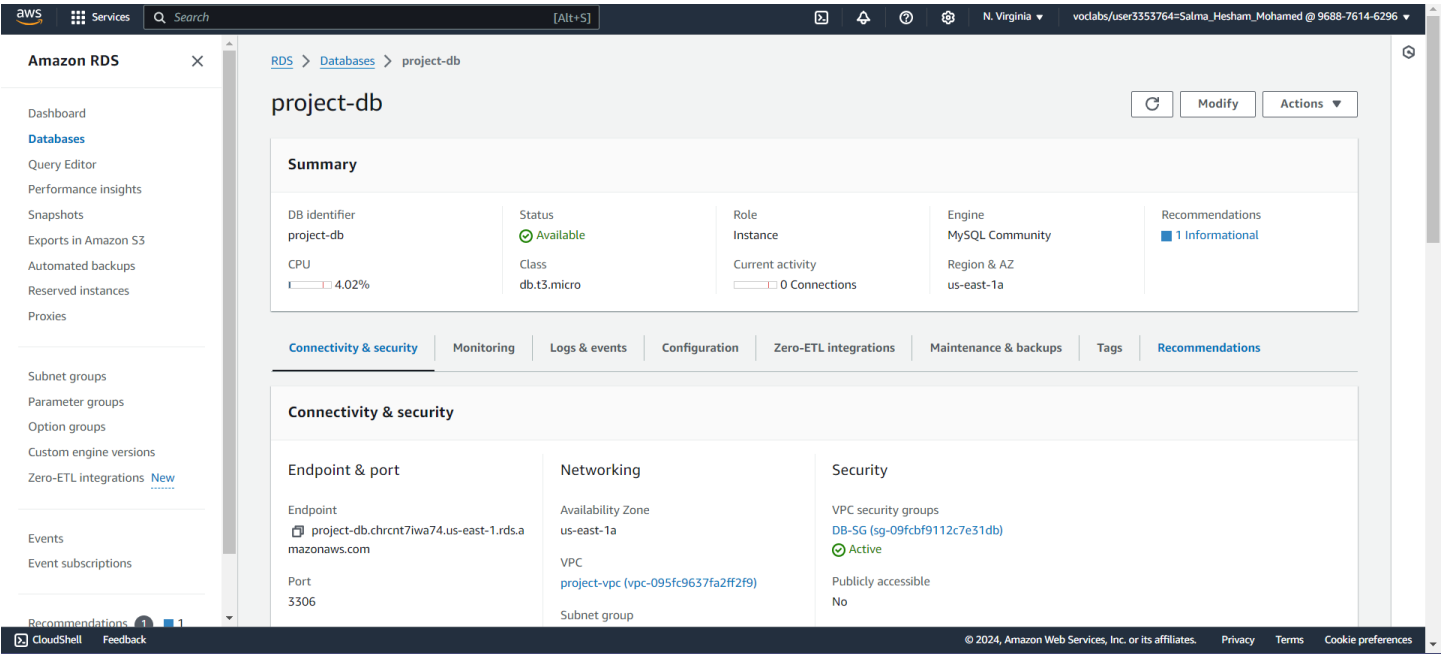
Description: Set up MySQL RDS with one primary and one standby instance in private subnets, ensuring data security.

Steps:

1. Created MySQL RDS in private subnets with multi-AZ deployment.
2. Stored database credentials in AWS Secrets Manager and restricted access.

Tools:

AWS RDS, AWS Secrets Manager, AWS Management Console



Task 3.2: Configuring Database Security Group

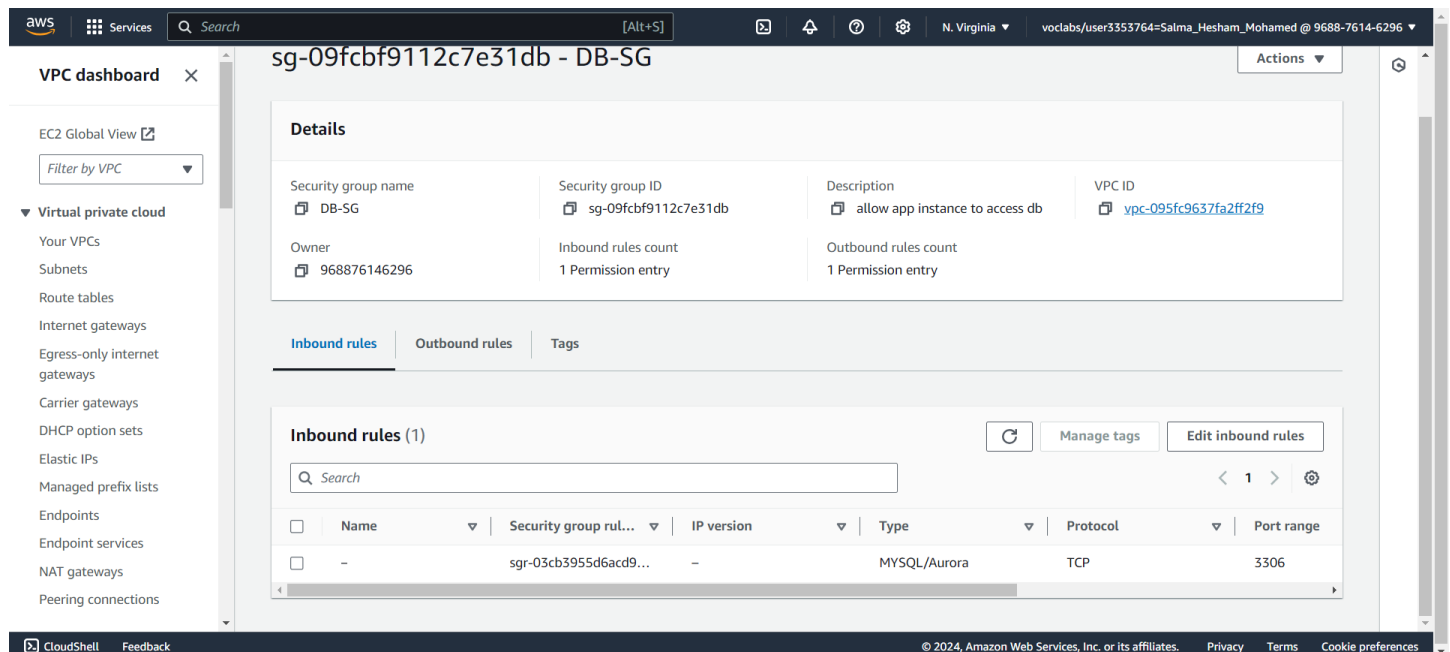
Description: Created a security group to allow only MySQL traffic from the EC2 web application to the database.

Steps:

1. Created a security group allowing MySQL traffic on port 3306 from the web app.
2. Applied the security group to the RDS instance.

Tools:

AWS Security Groups, AWS Management Console



Phase 4: Data Migration

Task 4.1: Migrating Data from EC2 to RDS

Description: Migrated the database from the EC2 instance to the MySQL RDS instance, ensuring data remains in the private subnets.

Steps:

1. Created a dump of the database from the EC2 instance.
2. Imported the dump into the MySQL RDS instance.

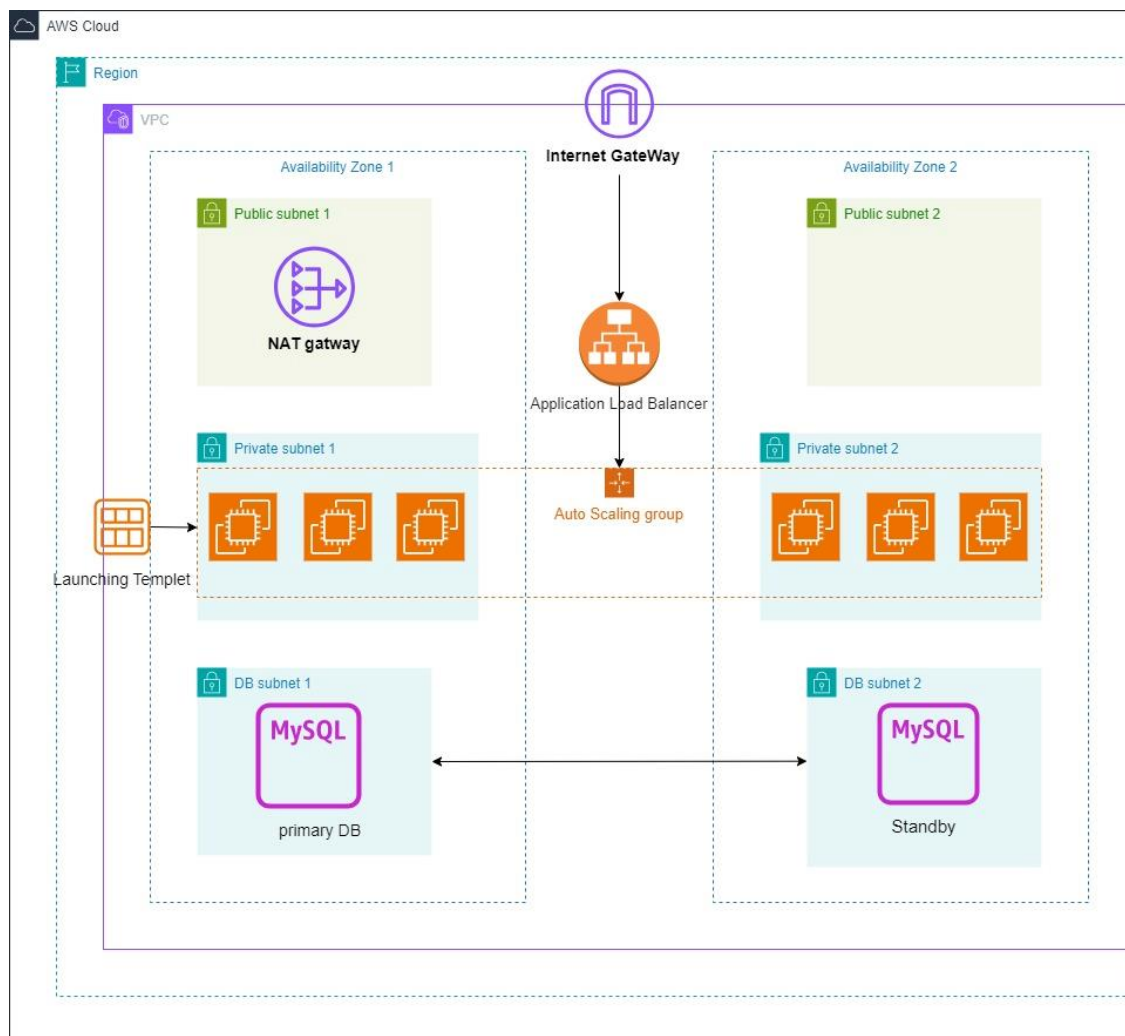
Tools:

AWS EC2, AWS RDS, MySQL

Test the deployment: Verify that the web app is functional by adding, deleting, and modifying records



Create an architectural diagram





Contact your AWS representative: [Contact Sales](#)

Export Date: 09/30/2024

Language: English

[Estimate url](#)

Estimate summary

Upfront cost 0.00 USD	Monthly cost 201.40 USD	Total 12 months cost 2,416.80 USD Includes upfront cost
---------------------------------	-----------------------------------	--

Detailed Estimate

Name	Group	Region	Upfront cost	Monthly cost
Amazon RDS for MySQL	-	US East (N. Virginia)	0.00 USD	128.34 USD
Status	-			
Description:	using Amazon RDS with MySQL to manage my database in a scalable, secure, and automated way.			
Config summary	Storage amount (20 GB), Storage for each RDS instance (General Purpose SSD (gp2)), Nodes (2), Instance type (db.t2.medium), Utilization (On-Demand only) (60 % Utilized/Month), Deployment option (Multi-AZ), Pricing strategy (OnDemand)			
Name	Group	Region	Upfront cost	Monthly cost
Elastic Load Balancing	-	US East (N. Virginia)	0.00 USD	28.11 USD
Status	-			
Description:	using an Application Load Balancer (ALB) to distribute incoming traffic across multiple instances of my web application.			
Config summary	Number of Application Load Balancers (1)			
Name	Group	Region	Upfront cost	Monthly cost
AWS Secrets Manager	-	US East (N. Virginia)	0.00 USD	0.40 USD
Status	-			
Description:	using AWS Systems Manager (SSM) to manage and automate tasks across my AWS infrastructure			
Config summary	Number of secrets (1), Average duration of each secret (30 days), Number of API calls (1 per hour)			
Name	Group	Region	Upfront cost	Monthly cost
Amazon EC2	-	US East (N. Virginia)	0.00 USD	11.47 USD
Status	-			
Description:	sing Amazon EC2 to host a web application, and I've opted for a 3-year EC2 Savings Plan. This plan allows me to save money by committing to a consistent usage level for 3 years, making the hosting costs more predictable and significantly lower compared to on-demand pricing.			
Config summary	Tenancy (Shared Instances), Operating system (Linux), Workload (Consistent, Number of instances: 2), Advance EC2 instance (t2.micro), Pricing strategy (Compute Savings Plans 3yr No Upfront), Enable monitoring (disabled), EBS Storage amount (15 GB), DT Inbound: Not selected (0 TB per month), DT Outbound: Not selected (0 TB per month), DT Intra-Region: (0 TB per month)			
Name	Group	Region	Upfront cost	Monthly cost
Amazon Virtual Private Cloud (VPC)	-	US East (N. Virginia)	0.00 USD	33.08 USD
Status	-			
Description:	using a NAT Gateway to enable my private instances in a VPC to securely access the internet without exposing them to inbound traffic.			
Config summary	Number of NAT Gateways (1)			

Acknowledgement

AWS Pricing Calculator provides only an estimate of your AWS fees and doesn't include any taxes that might apply. Your actual fees depend on a variety of factors, including your actual usage of AWS services. [Learn more](#)