Declaration	
Questions in this exercise are intentionally complex and could be convoluted or confusing. This is by design and to simulate real life situations where customers seldom give crystal clear requirements and ask unambiguous questions.	
I have read the above statement and agree to these conditions	
I AGREE	MOHAMED EL HASSNAOUI
	<Enter your name above this line to indicate that you are in agreement>

Instructions
Every screenshot requested in this workbook is compulsory and carries 0.5 points.
Your AWS account ID must be clearly visible in every screenshot using the AWS console; missing id or using someone else's id is not permitted. Such cases will be considered as plagiarism and severe penalty will be imposed.
All screenshots must be in the order mentioned under "Expected Screenshots" for every step
DO NOT WAIT UNTIL THE LAST MINUTE. The program office will not extend the project submission deadline under any circumstances.
The file should be renamed in the format BATCH_FIRSTNAME_LASTNAME_PROJECT1. For example: PGPCCMAY18_VIJAY_DWIVEDI_PROJECT1.pdf

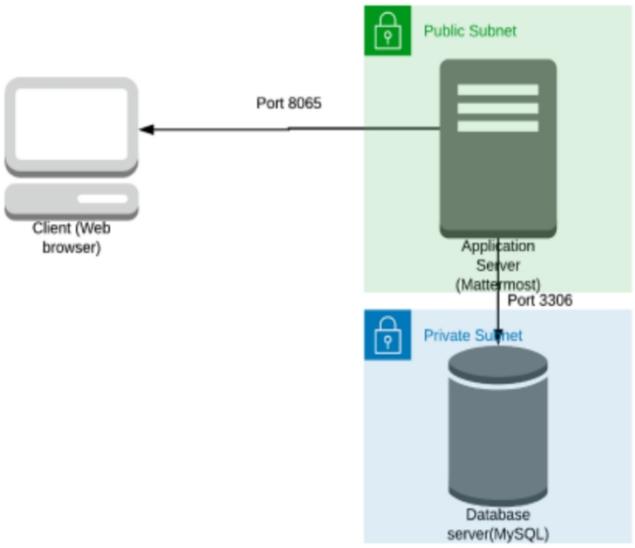
Resource Clean Up
Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.
After completing the lab, make sure to delete each resource created in reverse chronological order.

Scenario

Team communication and instant messaging solutions are an integral part of any business environment today. As of 2020, the total number of users of Slack and Microsoft Teams exceeded 20 million.

Some organizations might have compliance policies in place which do not allow them to use services managed by third parties. They will prefer solutions that can be managed and hosted on servers controlled by them. The same will extend to communication solutions as well.

Architecture diagram

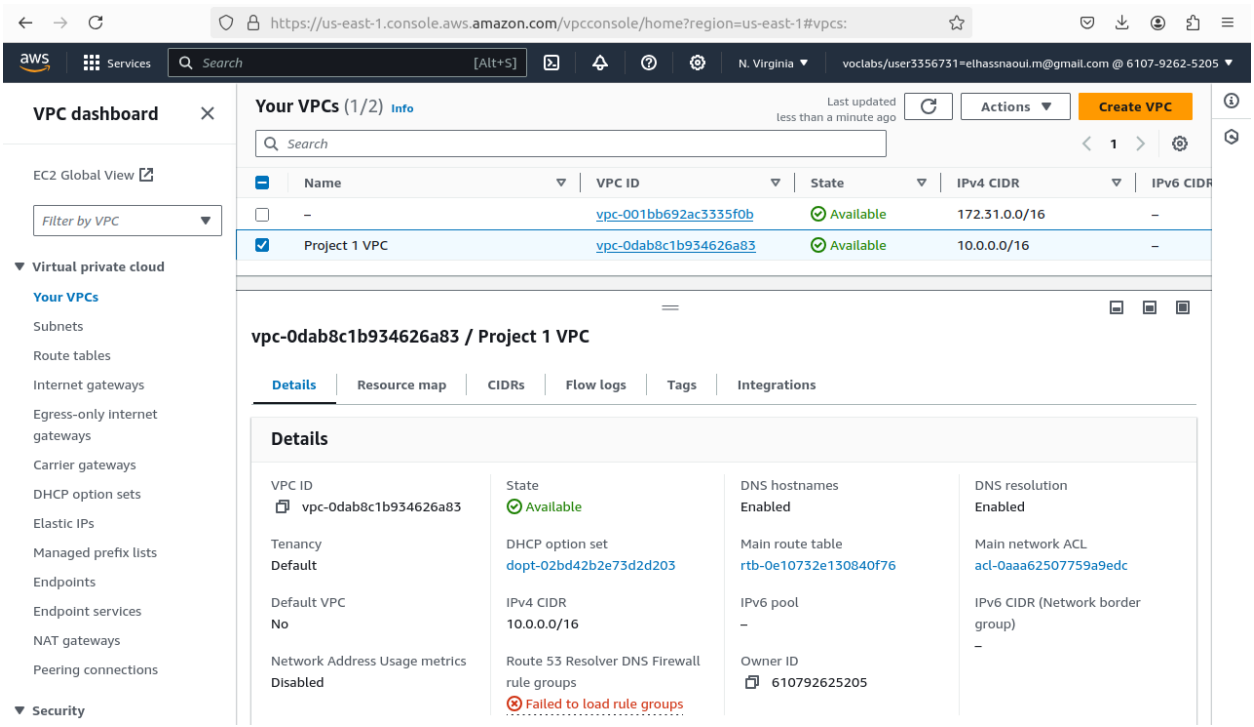


Architecture Implementation	
1	Implement 2 different subnets (one public and the other private) in a custom VPC
2	Install and configure MySQL on an Amazon Linux 2 instance on the private subnet using the instructions provided. (Hint: Use a bastion host and a NAT gateway)
3	Install and configure Mattermost on an Amazon Linux 2 instance on the public subnet using the provided instructions.
4	Configure the security groups to allow the ports as shown in the architecture.
5	Test the installation by accessing the IP of the public instance in a browser via the port 8065.

Step 1: VPC and Subnet Creation

Step number	a
Step name	Creation of VPC
Instructions	1) Navigate to VPC using the Services button at the top of the screen 2) Select "Your VPCs" on the left side of the screen 3) Click on "Create VPC" 4) Enter the following fields : Name: Project 1 VPC IPv4 CIDR Block : 10.0.0.0/16 The rest of the options can be ignored 5) Select "Create VPC" 6) Select the VPC and click on Actions->Edit DNS hostnames 7) Enable DNS hostnames and click on Save
Expected screenshots	Created VPC with properties visible

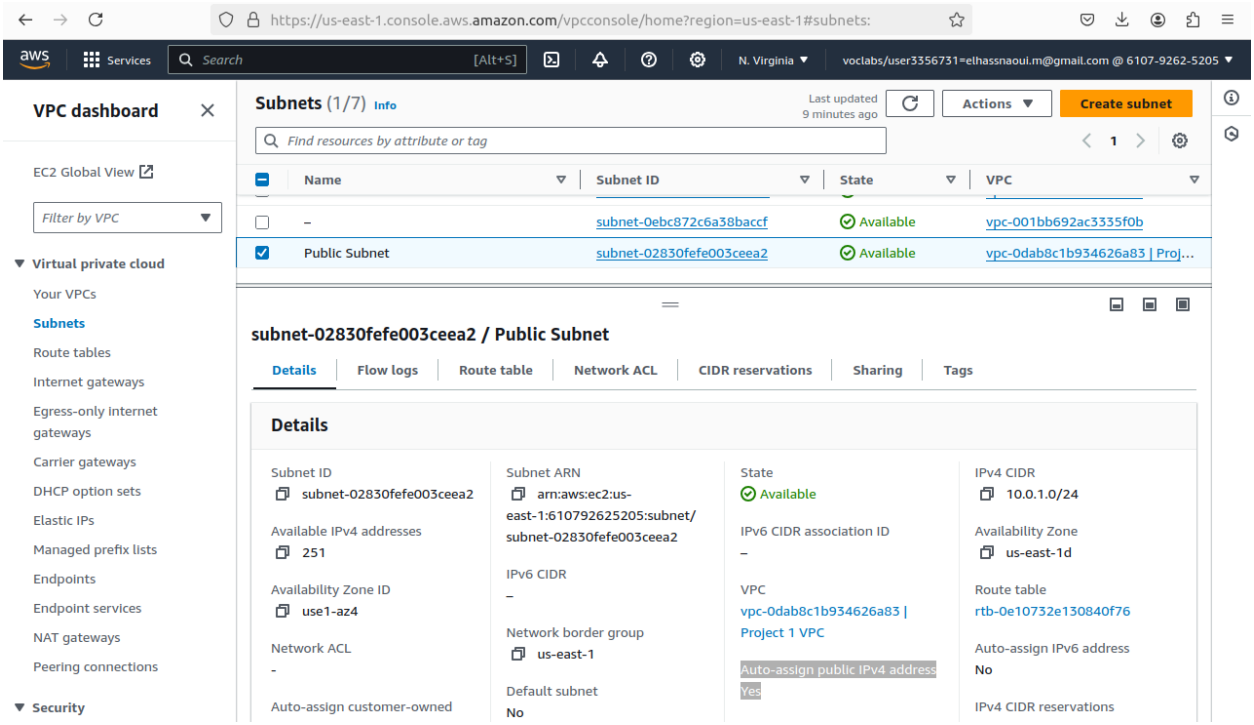
<Insert Screenshot a(1) here



>

Step number	b
Step name	Creation of public subnet
Instructions	1) Navigate to VPC->Subnets 2) Click on "Create Subnet" 3) Enter the following fields Name tag : Public Subnet VPC : Select the Project 1 VPC IPv4 CIDR block : 10.0.1.0/24 The other options can be ignored 4) Click on Create 5) Once the subnet has been created, select the subnet and click on Actions->Modify Auto-assign IP settings 6) Enable the option "Auto assign IPv4" and select Save
Expected screenshots	Subnet Creation screen

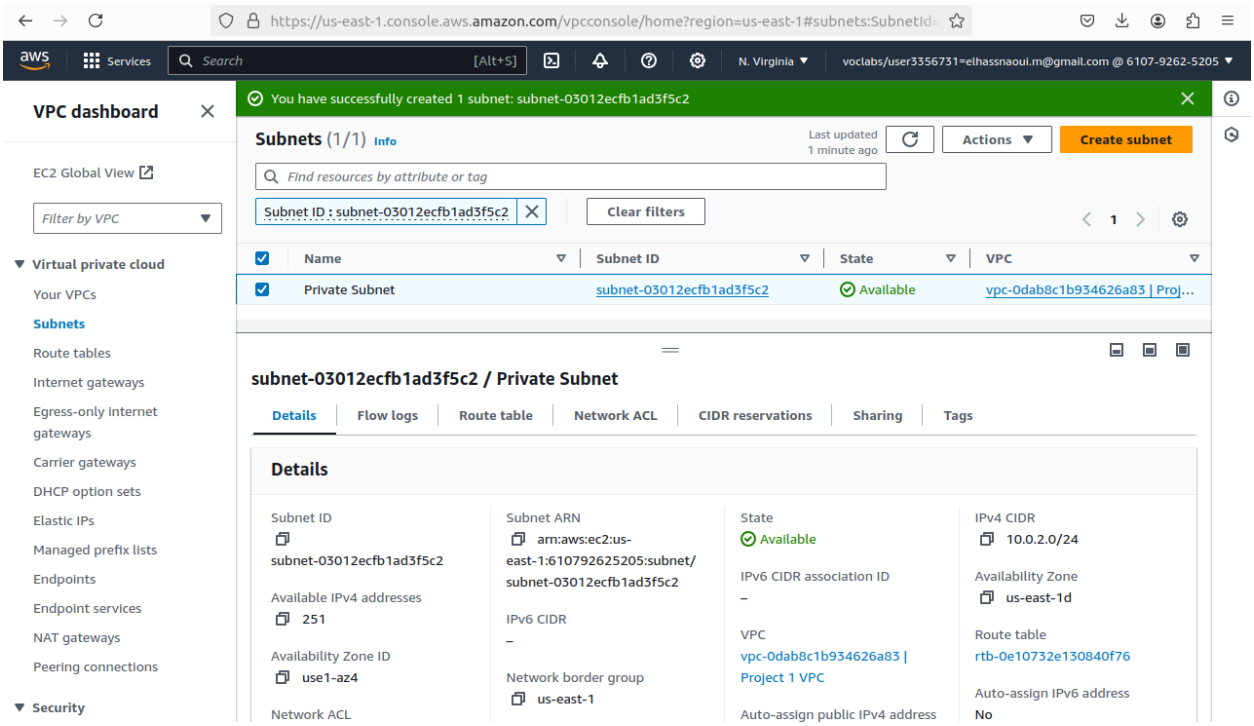
<Insert Screenshot b(1) here



>

Step number	c
Step name	Creation of private subnet
Instructions	1) Navigate to VPC->Subnets 2) Click on "Create Subnet" 3) Enter the following fields Name tag : Private Subnet VPC : Select the Project 1 VPC IPv4 CIDR block : 10.0.2.0/24 The other options can be ignored 4) Click on Create
Expected screenshots	Subnet Creation screen

<Insert Screenshot c(1) here

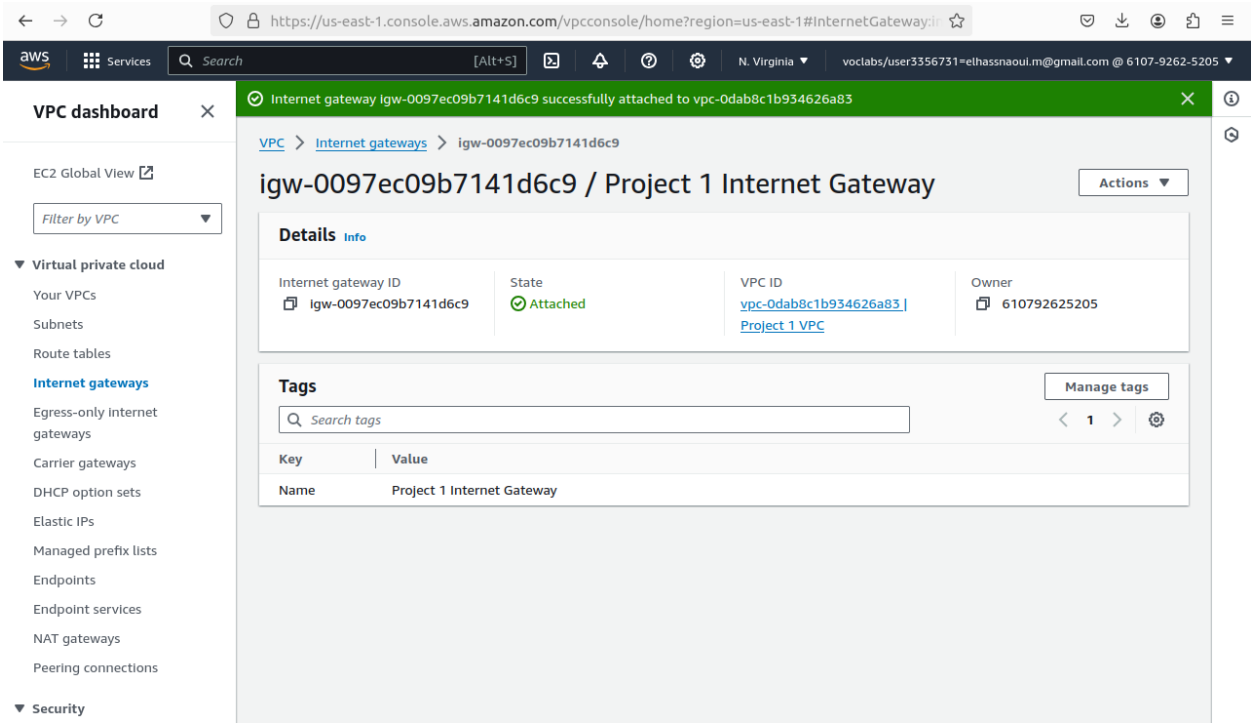


>

Step 2 : Internet Gateway and VPC

Step number	a
Step name	Creation and Configuration of Internet Gateway
Instructions	1) Navigate to VPCs->Internet Gateway 2) Click on "Create Internet Gateway" 3) Enter the name tag "Project 1 Internet Gateway" and click on "Create Internet Gateway" 4) After the gateway is created, select it and click on Actions->Attach to VPC 5) Select the Project 1 VPC and click on "Attach Internet Gateway"
Expected screenshots	Creation of Internet Gateway

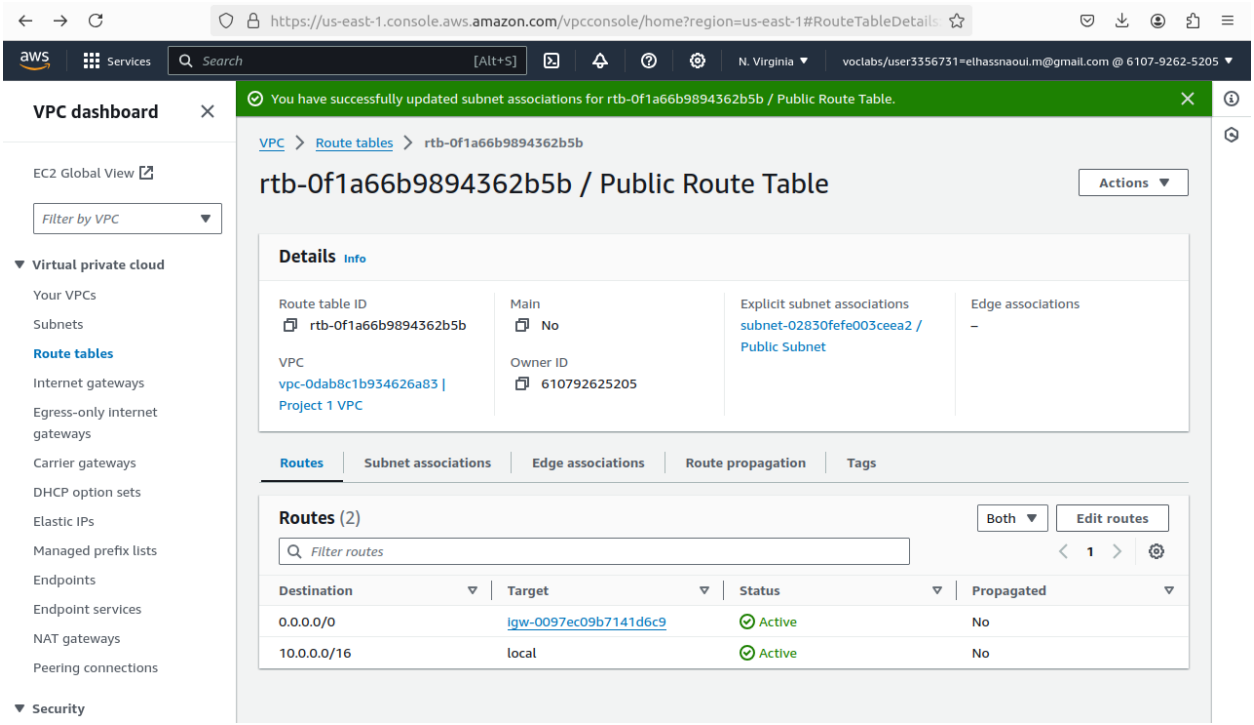
<Insert Screenshot a(1) here



>

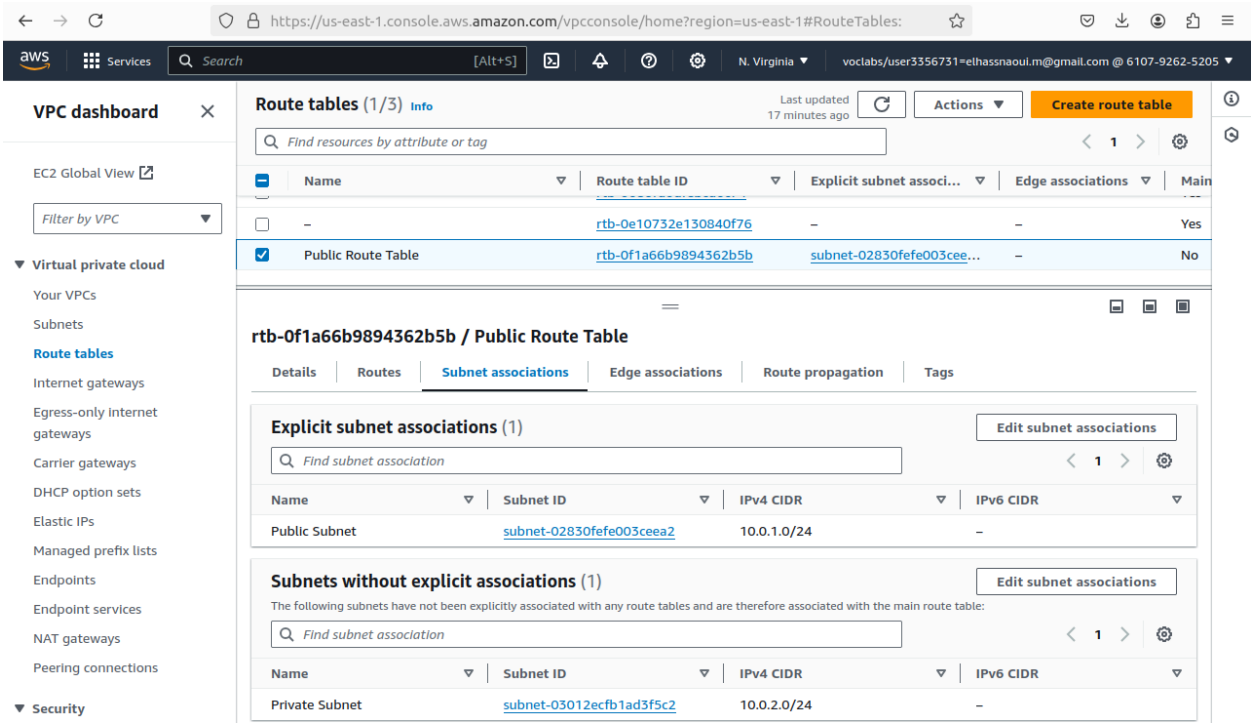
Step number	b
Step name	Creation of public route table
Instructions	1) Navigate to VPC -> Route Tables and click on Create Route table 2) Enter the name tag "Public Route Table", select the Project 1 VPC from the dropdown and click on Create 3) Once the route table is created, select it and select the Routes tab below the list of route tables 4) Click in Edit Routes and add the following route (Don't edit the existing one) - Destination : 0.0.0.0/0 - Target : Select Internet Gateway and the select the Project 1 Internet Gateway Click on Save Routes 5) Select the Subnet Associations tab and click on Edit Subnet Associations 6) Select the Public Subnet from the list and click on Save
Expected screenshots	1) Route list of the route table 2) Subnet Associations of the route table

<Insert Screenshot b(1) here - Route list of the route table



>

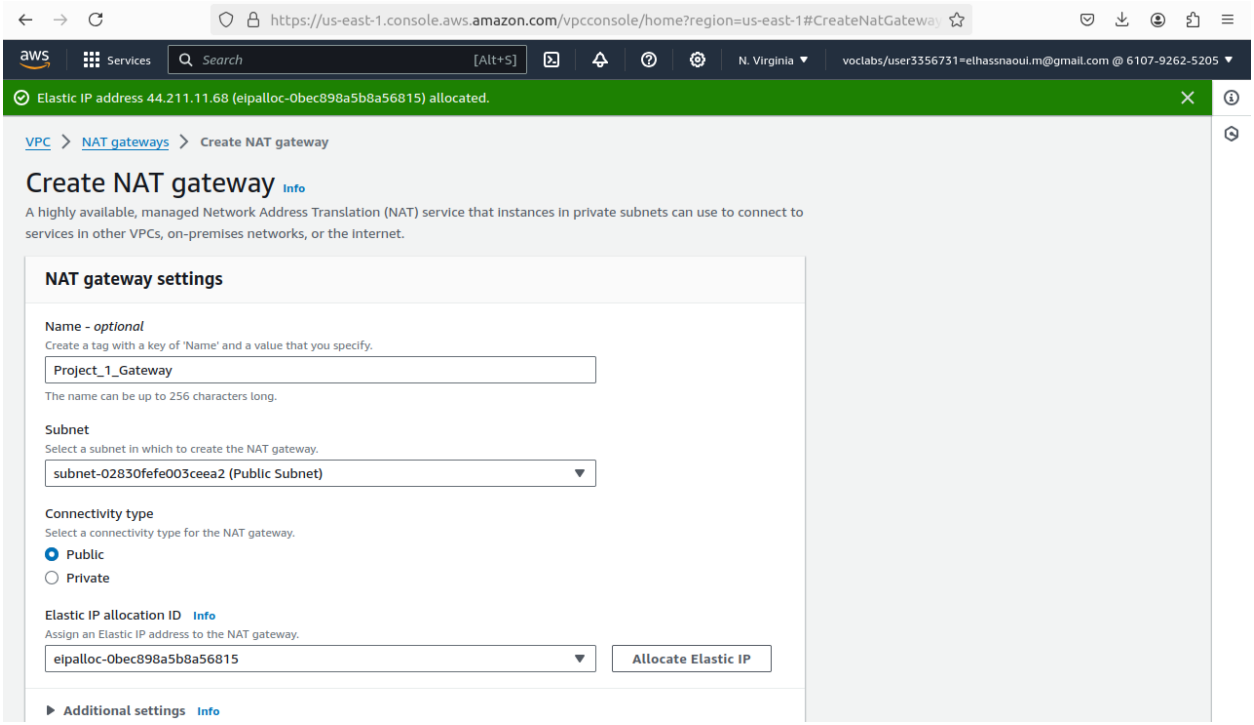
<Insert Screenshot b(2) here - Subnet Associations of the route table



>

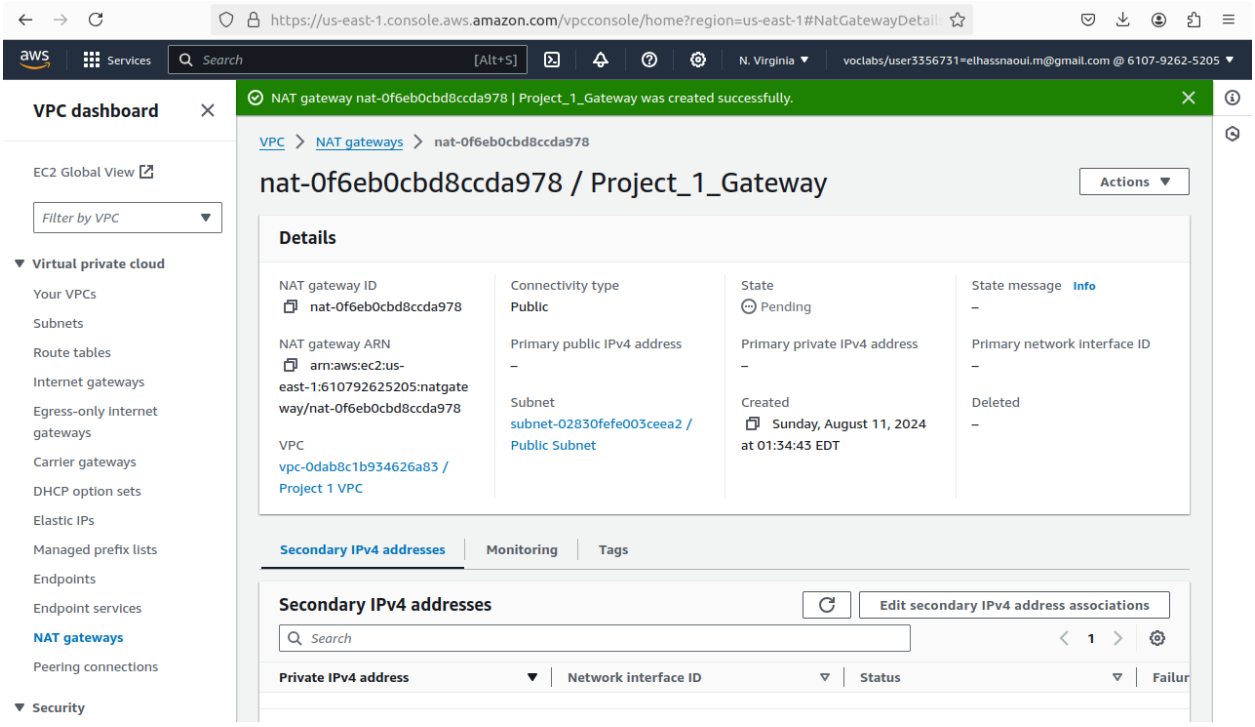
Step number	c
Step name	Creation of NAT gateway
Instructions	1) Navigate to VPC using the Services button at the top of the screen 2) Select NAT Gateway at the left side of the screen 3) Click on Create NAT Gateway - Deploy it in the public subnet - Connectivity type: Public - Allocate an elastic IP by clicking on “Allocate Elastic IP” 4) Click on “Create NAT Gateway” to create the gateway
Expected screenshots	1) NAT gateway creation details 2) Gateway after creation

<Insert Screenshot c(1) here - NAT gateway creation details



>

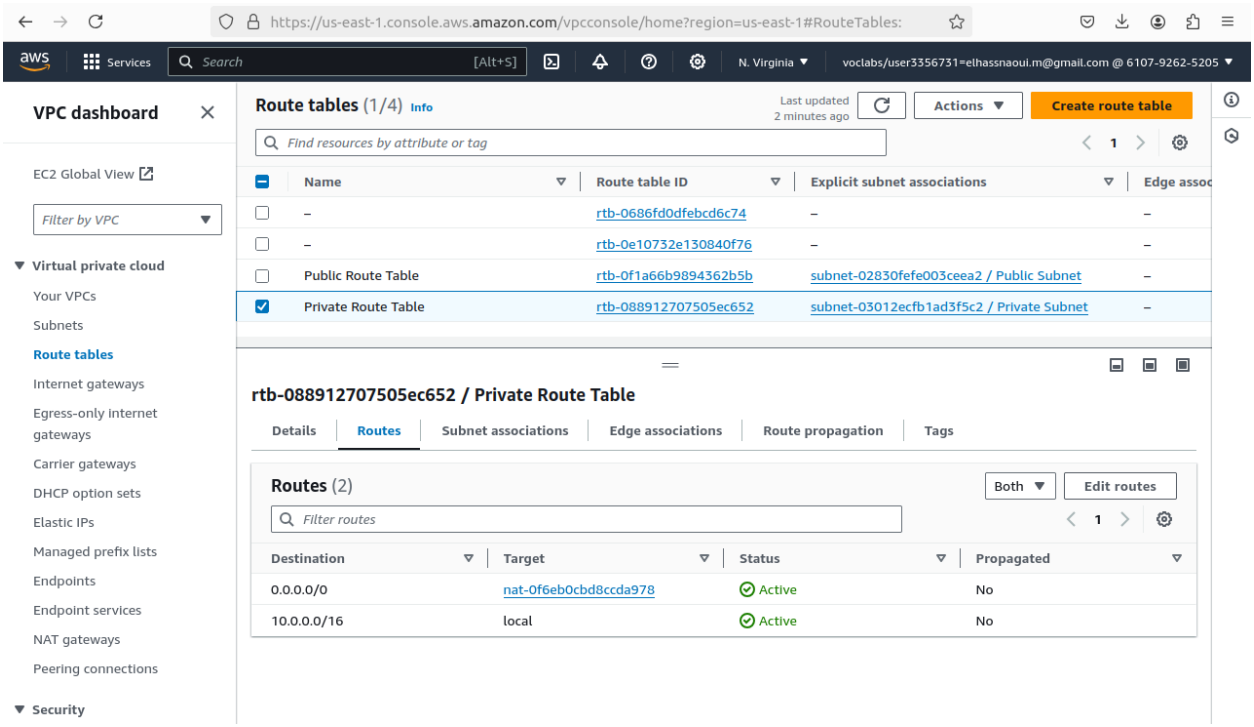
<Insert Screenshot c(2) here - Gateway after creation



>

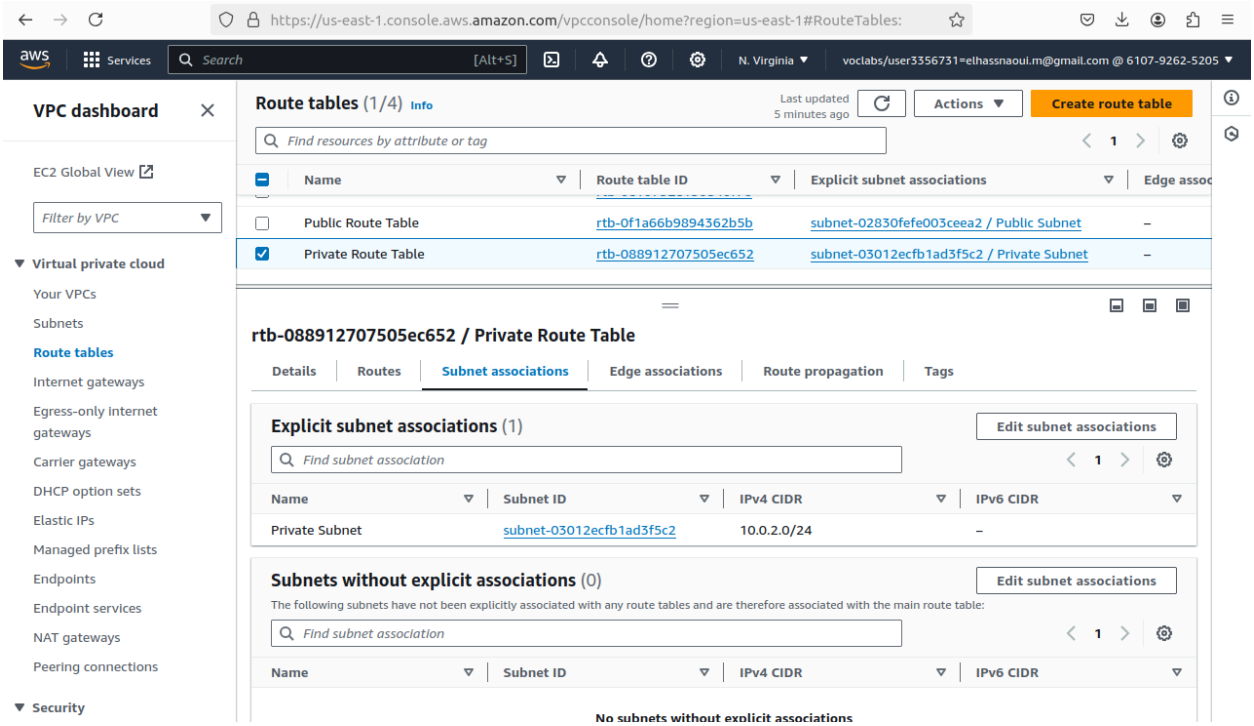
Step number	d
Step name	Creation of private route tables
Instructions	1) Navigate to VPC -> Route Tables and click on Create Route table 2) Enter the name tag "Private Route Table", select the Project 1 VPC from the dropdown and click on Create 3) Once the route table is created, select it and select the Routes tab below the list of route tables 4) Click in Edit Routes and add the following route (Don't edit the existing one) - Destination : 0.0.0.0/0 - Target: Select NAT Gateway and select the NAT Gateway created in the previous step Click on Save Routes 5) Select the Subnet Associations tab and click on Edit Subnet Associations 6) Select the private Subnet from the list and click on Save
Expected screenshots	1) Route list of the route table 2) Subnet association of the route table

<Insert Screenshot for d(1) here - Route list of the route table



>

<Insert Screenshot for d(2) here - Subnet association of the route table

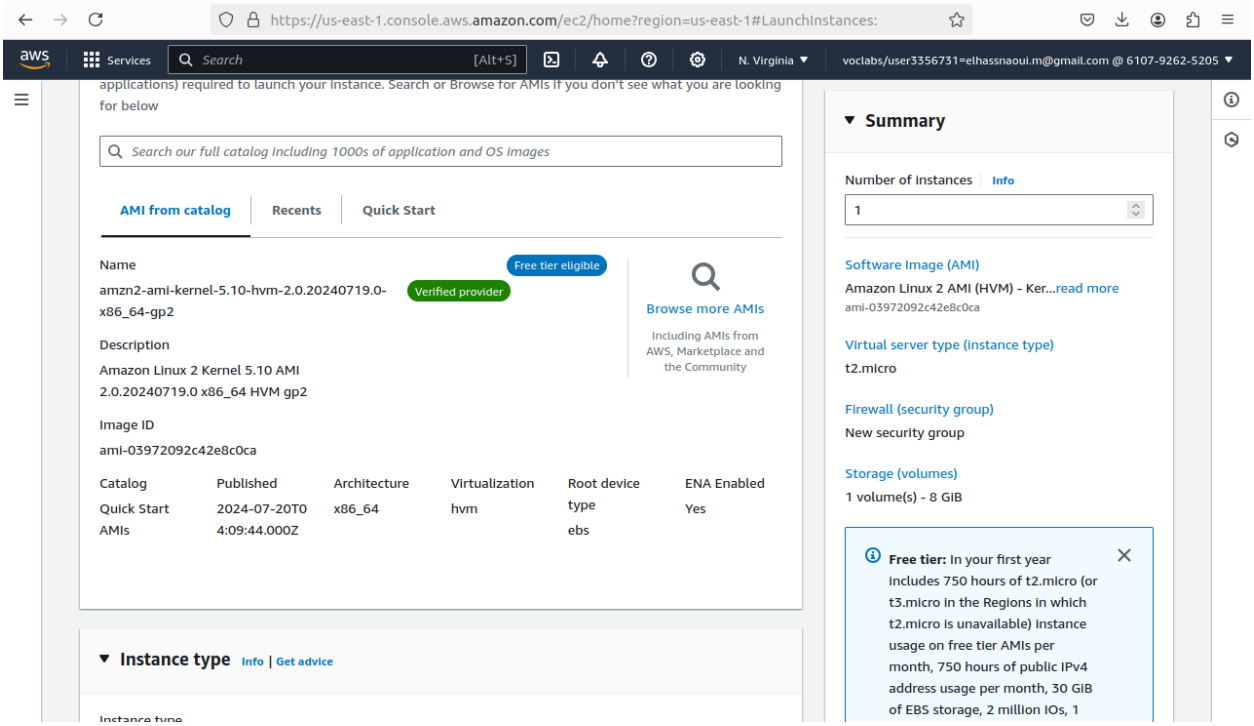


>

Step 3 : Creation of database and application servers

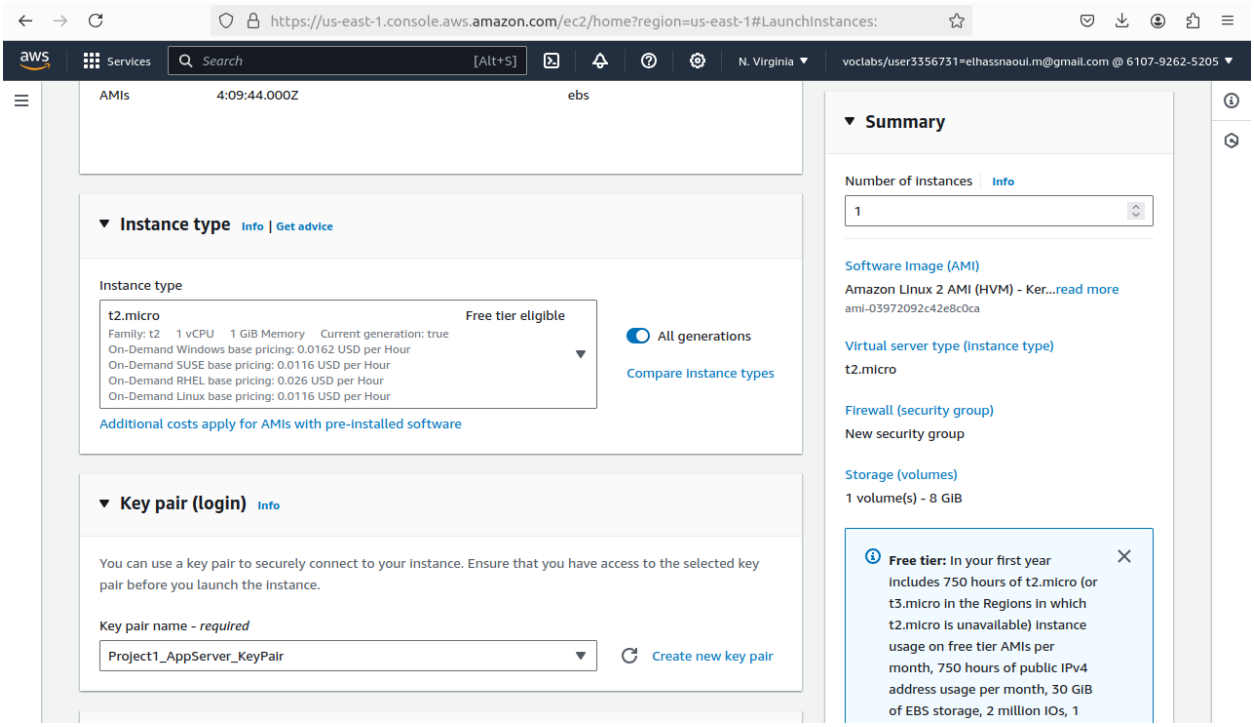
Step number	a
Step name	Creation of application server
Instructions	1) Navigate to EC2 using the Services button at the top of the screen 2) Select Instances at the left side of the screen 3) Click on Launch Instance - Select the AMI Amazon 2 Linux - Select the instance type t2.micro - Select Network as "Project 1 VPC" and subnet as "Public Subnet" - For the security group, open the ports 80,443, 22 and 8065 for source set to "Anywhere" 4) Launch the instance after creating a new pem file and downloading it
Expected screenshots	1) AMI used 2) Instance configuration screen 3) Security group rules 4) Instance after creation

<Insert screenshot a(1) here - AMI used



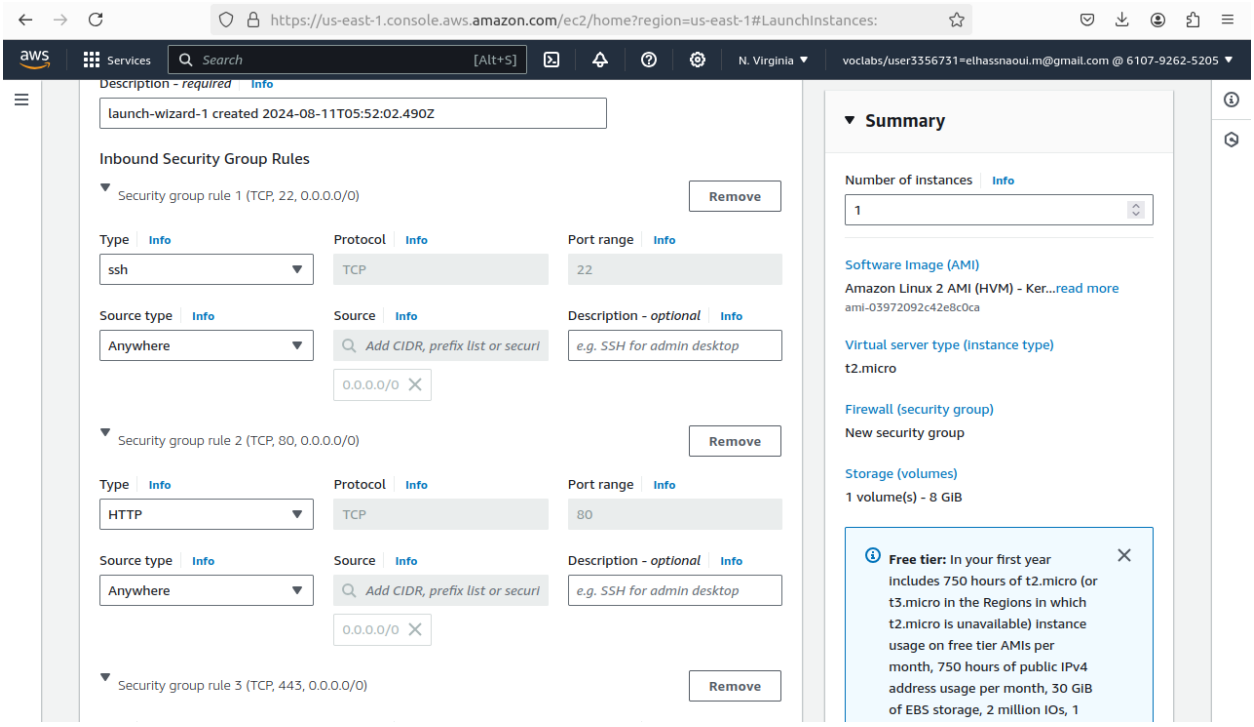
>

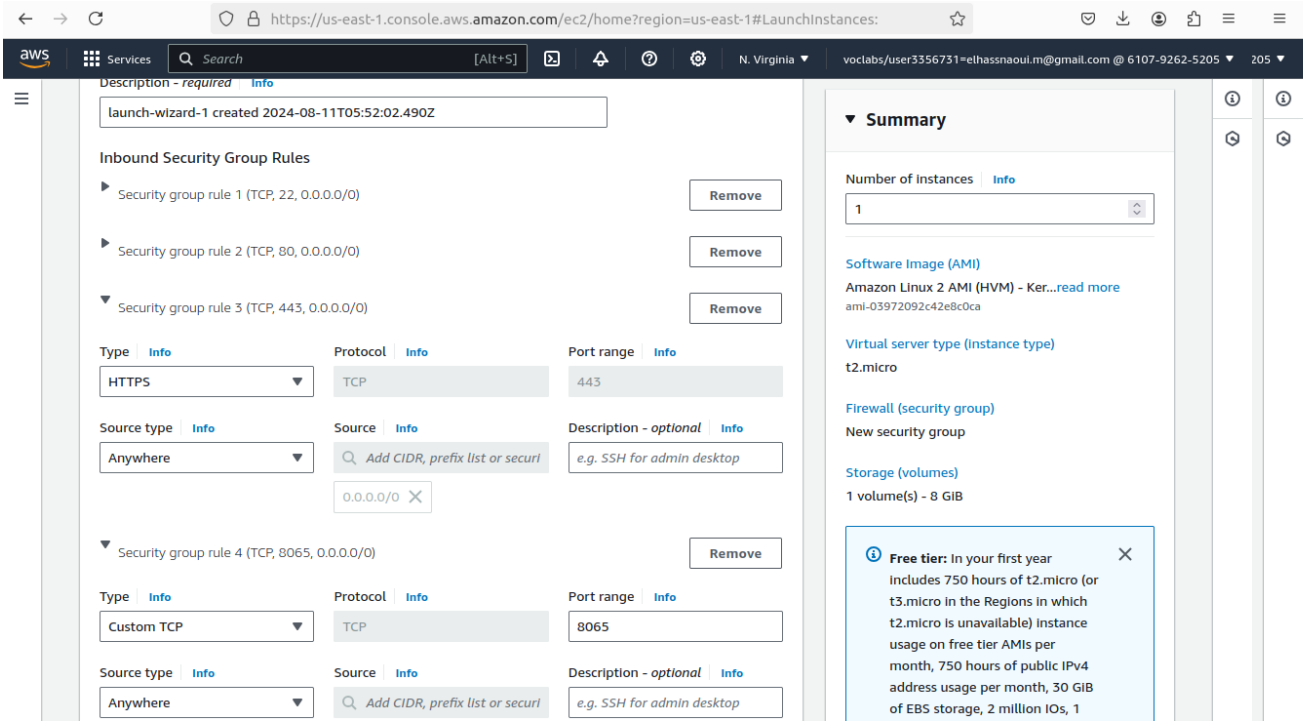
<Insert screenshot a(2) here - Instance configuration screen



>

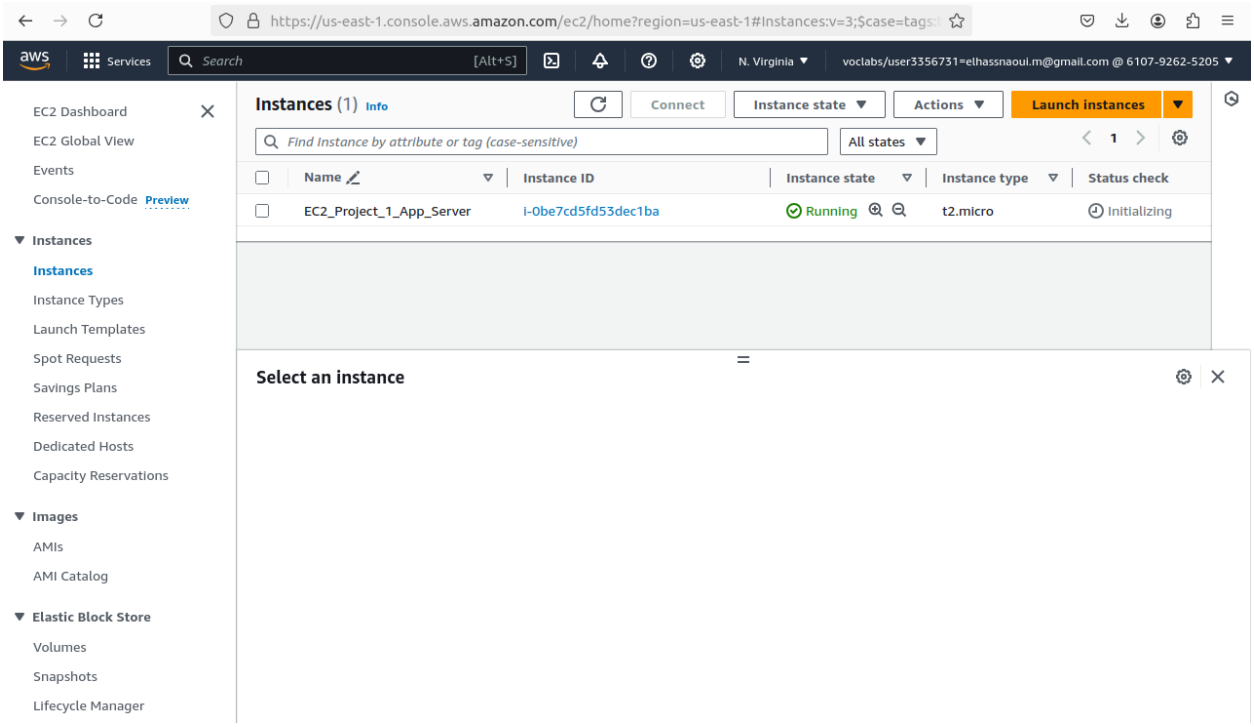
<Insert screenshot a(3) here - Security group rules





>

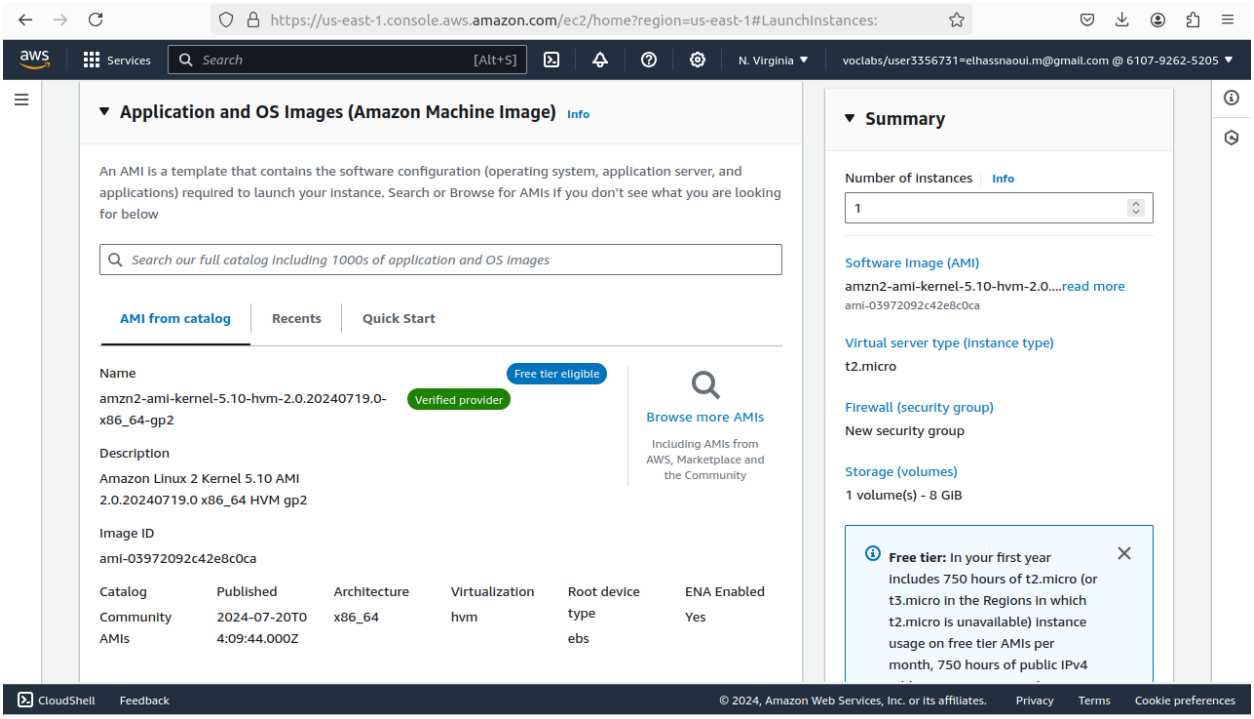
<Insert screenshot a(4) here - Instance after creation



>

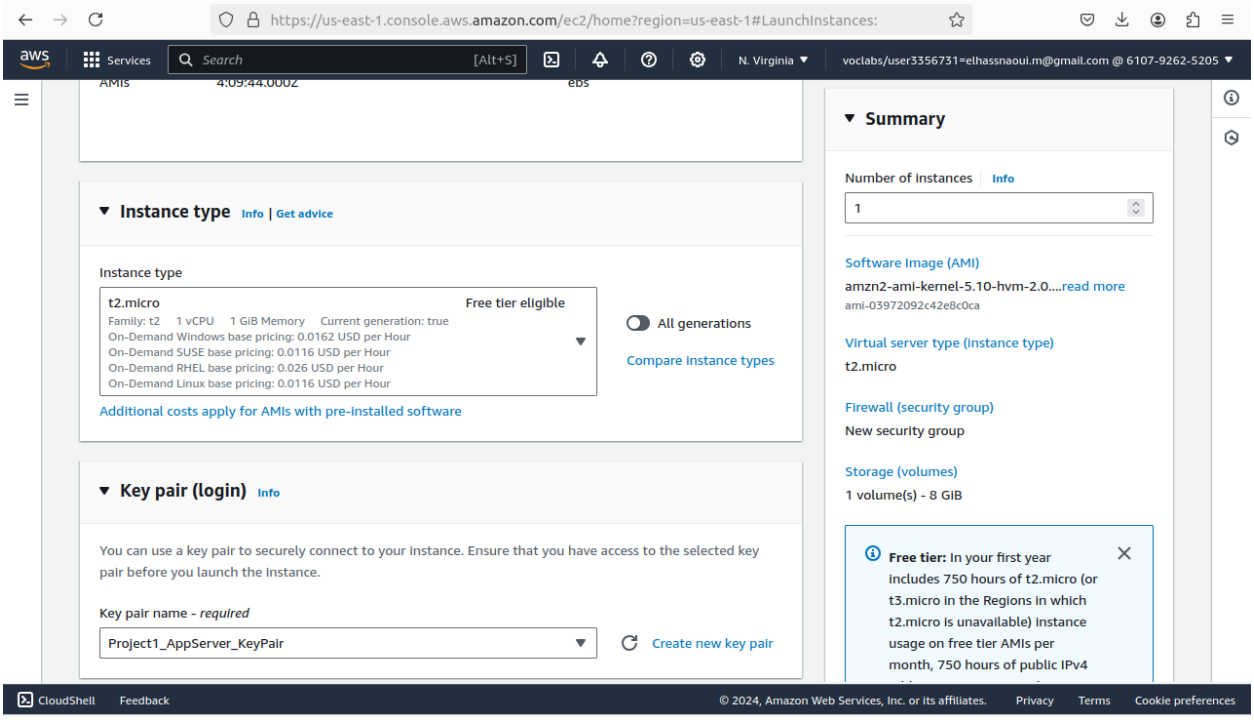
Step number	b
Step name	Creation of database server
Instructions	1) Navigate to EC2 using the Services button at the top of the screen 2) Select Instances at the left side of the screen 3) Click on Launch Instance - Select the AMI Amazon 2 Linux - Select the instance type t2.micro - Select Network as "Project 1 VPC" and subnet as "Private Subnet" - For the security group, open the ports 80, 443,22 and 3306 for source set to "Anywhere" 4) Launch the instance by selecting the same pem file created in the previous step
Expected screenshots	1) AMI used 2) Instance configuration screen 3) Security group rules 4) Instance after creation

<Insert screenshot b(1) here - AMI used



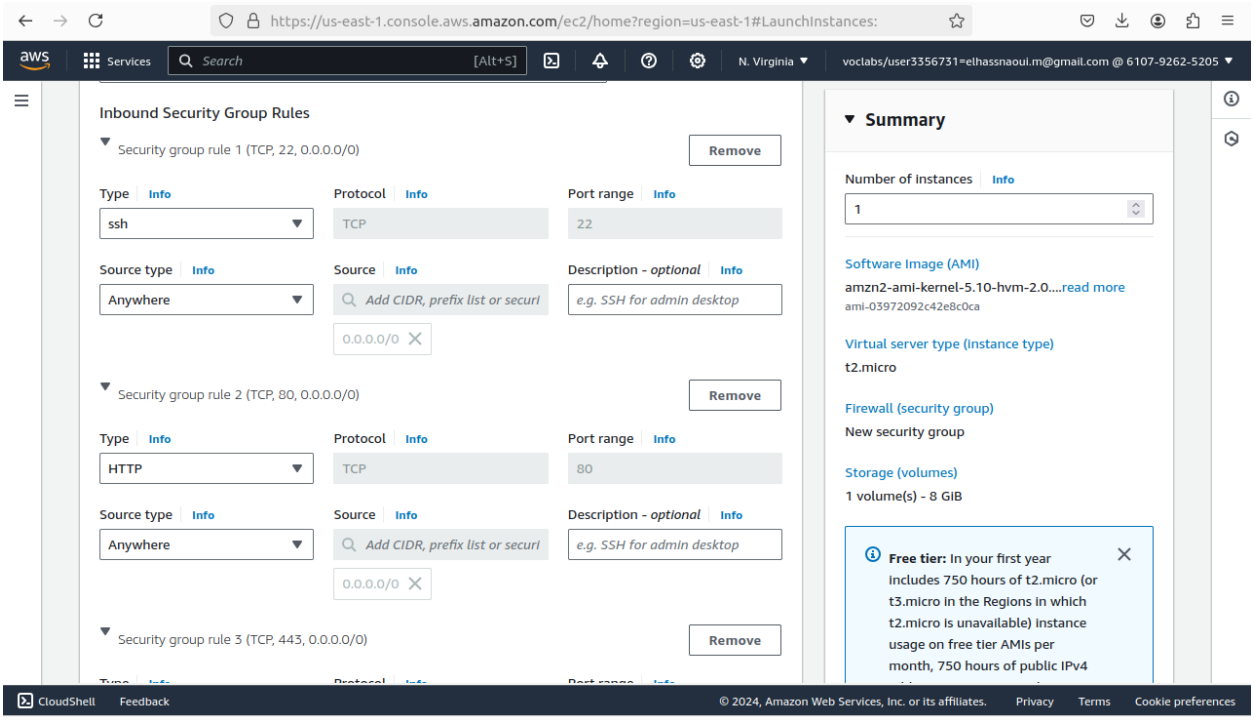
>

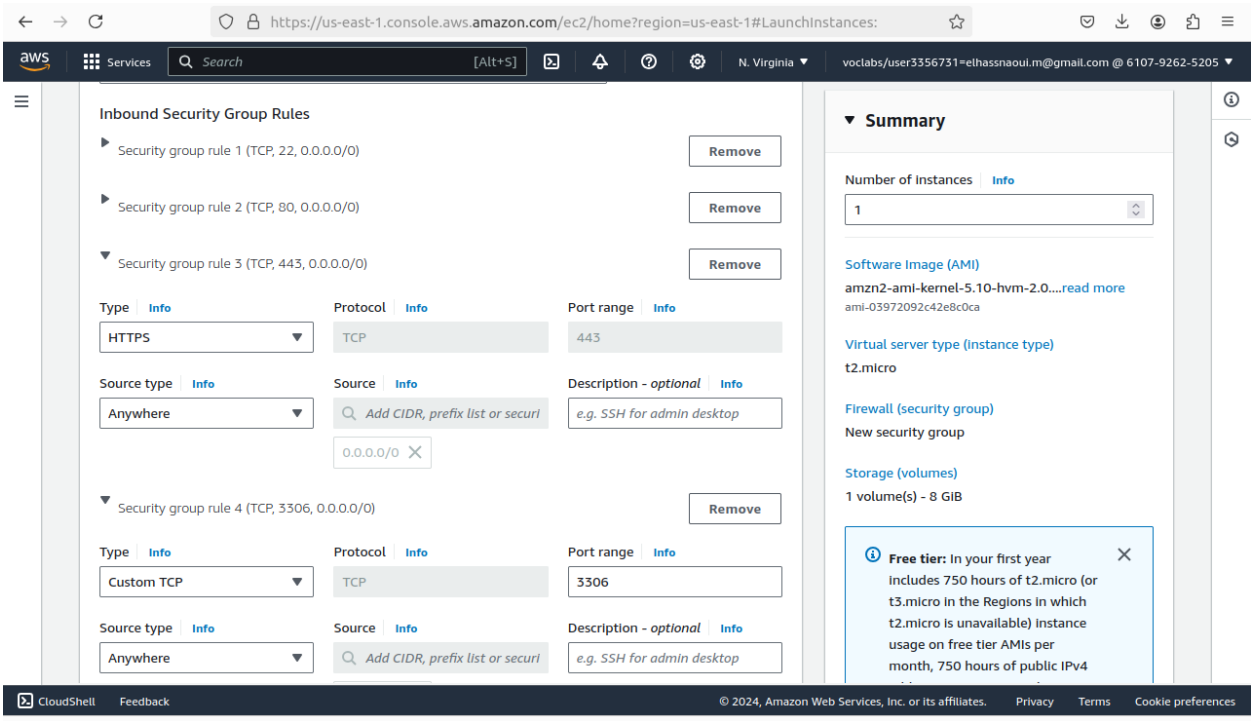
<Insert screenshot b(2) here - Instance configuration screen



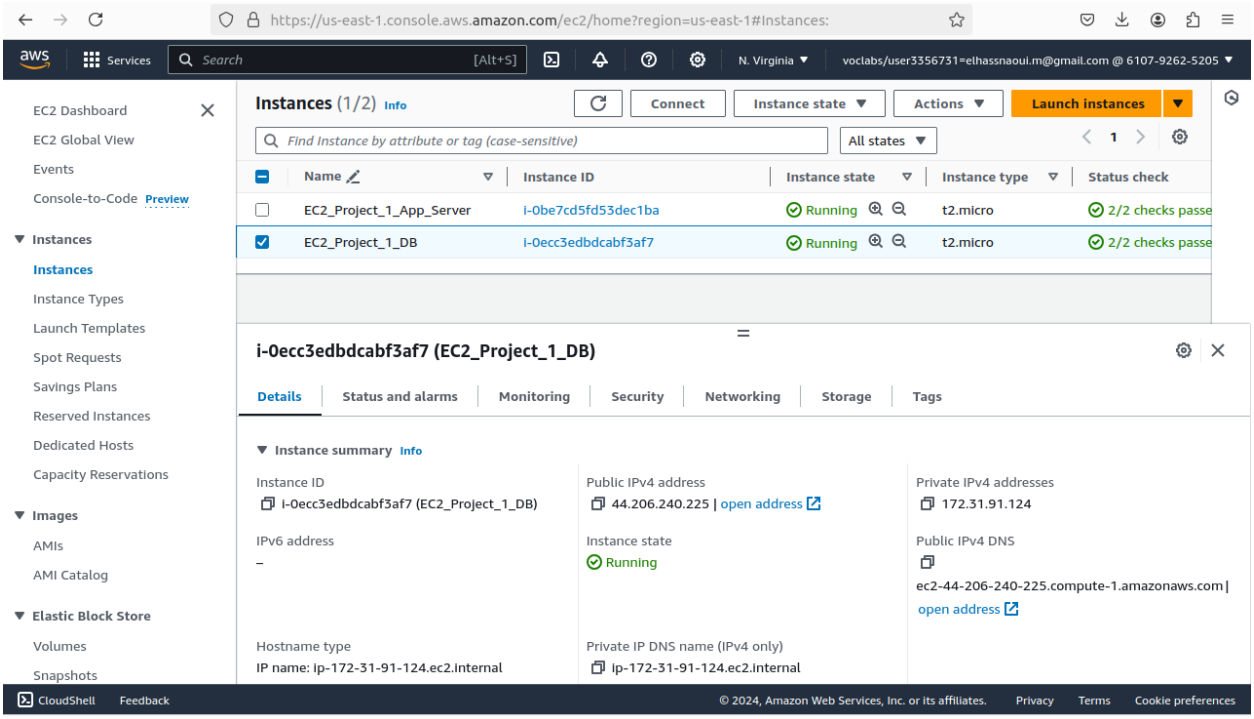
>

<Insert screenshot b(3) here - Security group rules>





<Insert screenshot b(4) here -Instance after creation



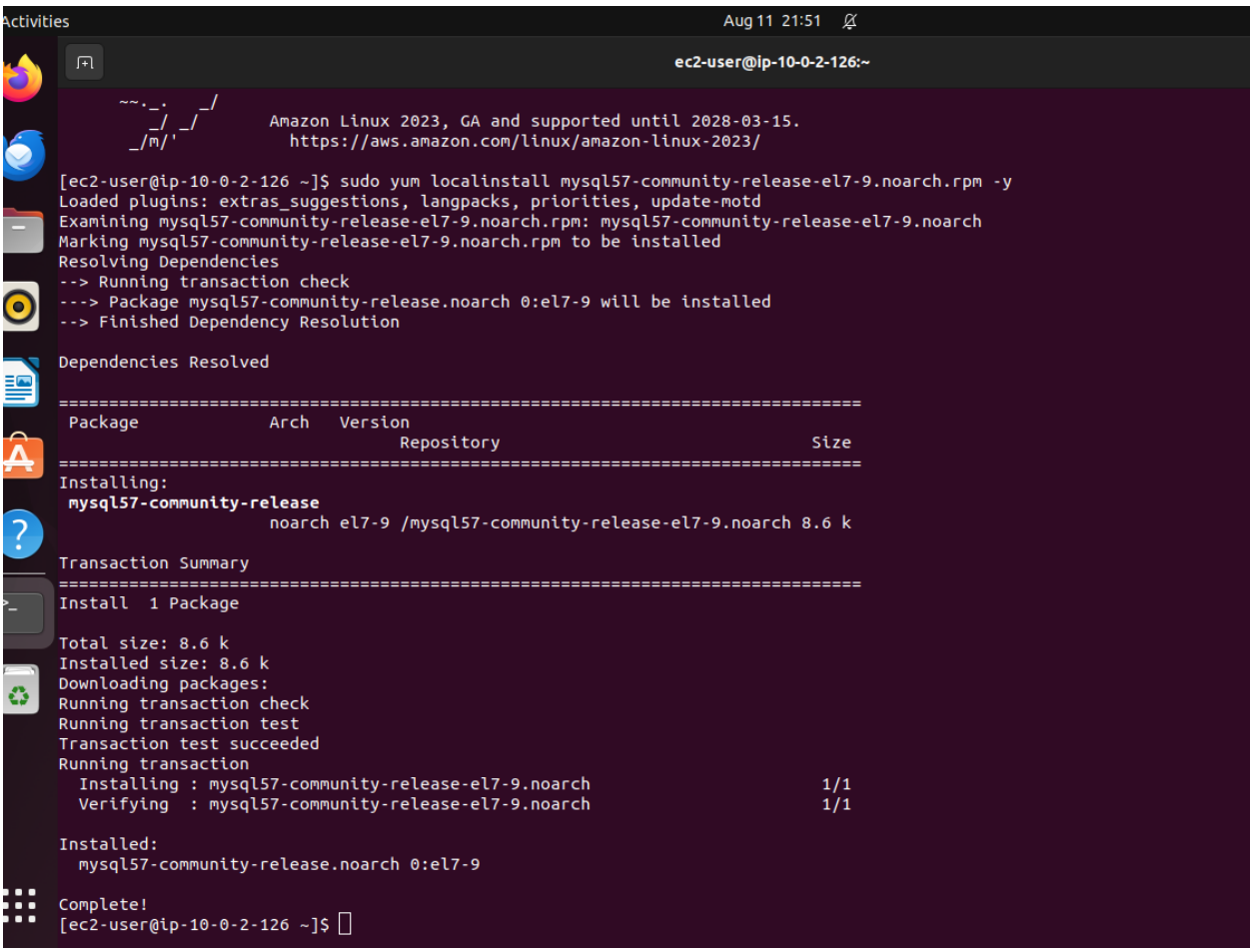
>

Step 4: Application and Database Installation and Testing

Step number	a
Step name	Installation and configuration of MySQL
Instructions	1) Copy the database pem file into the application server using the below command *scp -i YOUR_APP.pem YOUR_DB.pem ec2-user@YOUR_APP_PUBLIC_IP:/home/ec2-user* ***scp -i Project1_AppServer_KeyPair.pem Project1_AppServer_KeyPair.pem ec2-user@ec2-54-xxx-xx-xx.compute-1.amazonaws.com:/home/ec2-user*** 2) Log into the application server using SSH/Putty 3) From the application server, log into the database server using the pem file copied in step 1 and the private IP address of the database server with the following command *ssh -i YOUR_DB.pem ec2-user@YOUR_DB_PRIVATE_IP*

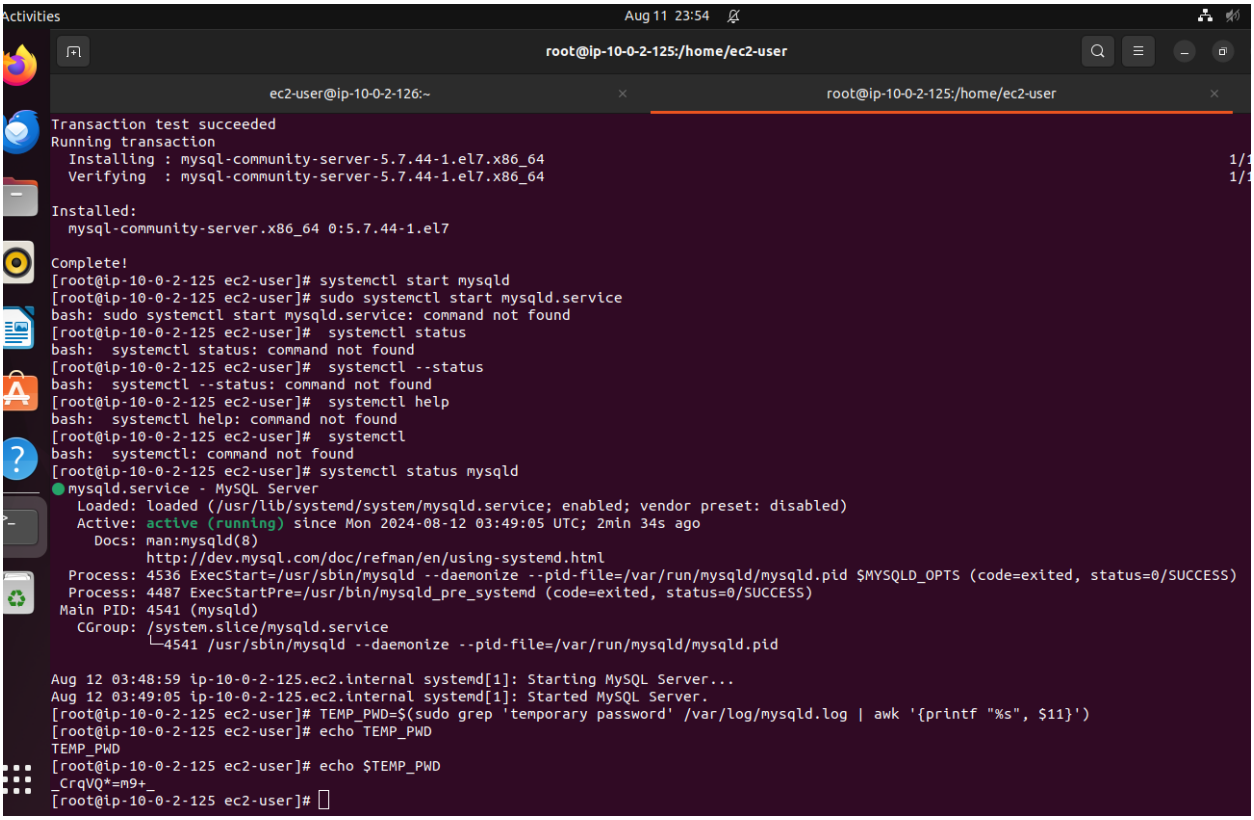
	4) Enter the following commands to install and configure MySQL on the database server *sudo yum update* *wget http://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm* *sudo yum localinstall mysql57-community-release-el7-9.noarch.rpm -y* *sudo yum install mysql-community-server -y –nogpgcheck* *sudo systemctl start mysqld.service* Run the below command to retrieve a temporary password for MySQL *TEMP_PWD=$(sudo grep 'temporary password' /var/log/mysqld.log | awk '{printf "%s", $11}')* Log in to MySQL with the below command *mysql -u root --password=$TEMP_PWD* Enter the below command after you login to MySQL. Do not change the password set in the below command. *ALTER USER 'root'@'localhost' IDENTIFIED BY 'Password42!';* Type 'exit' into the MySQL prompt and press Enter to exit out of the MySQL environment. Enter the below commands to complete the setup. Ignore any warning messages you receive. *wget https://d6opu47qoi4ee.cloudfront.net/install_mysql_linux.sh* *chmod 777 install_mysql_linux.sh* *sudo ./install_mysql_linux.sh* 5) Type *exit* to exit the database server and go back to the application server
Expected screenshots	1) Installation of MySQL 2) Retrieving the temporary password _CrqVQ*=m9+_ 3) Executing the provided script

<Insert screenshot a(1) here - Installation of MySQL



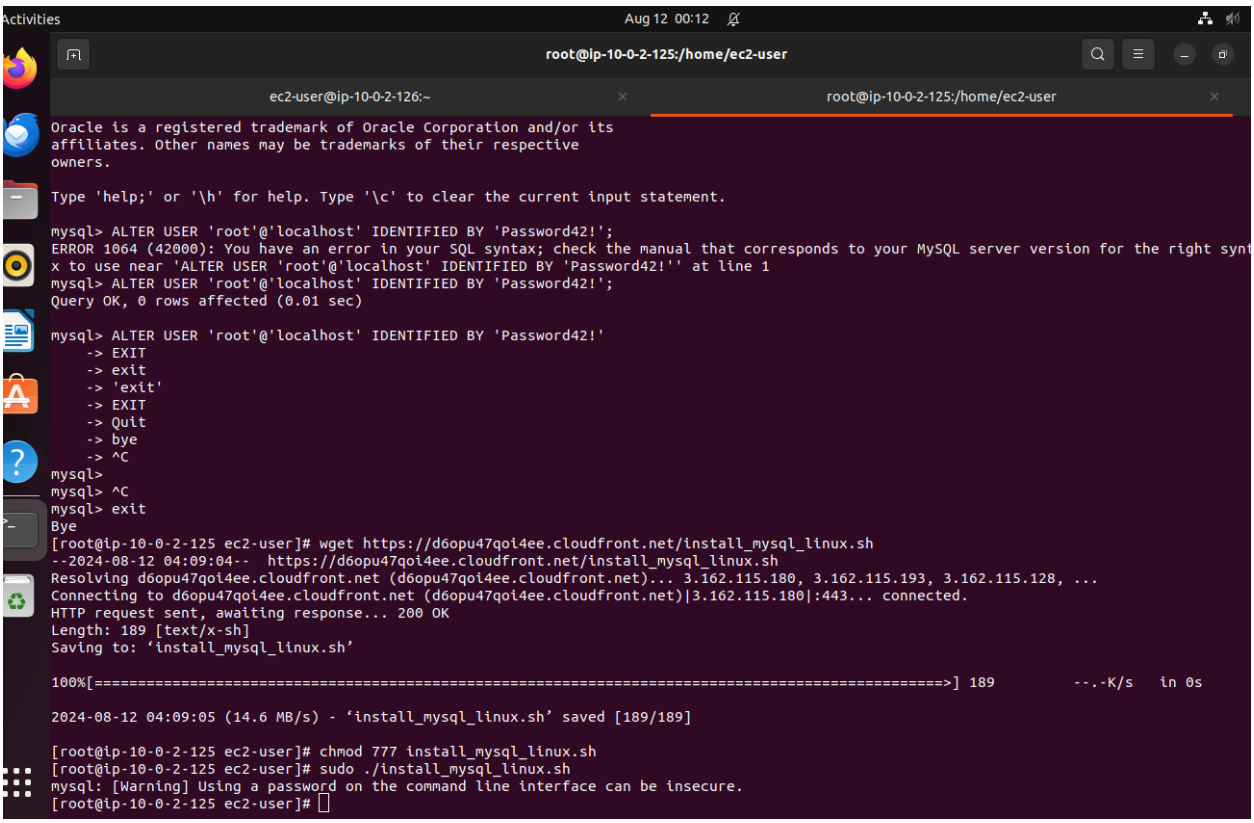
>

<Insert screenshot a(2) here - Retrieving the temporary password



>

<Insert screenshot a(3) here -Executing the provided script

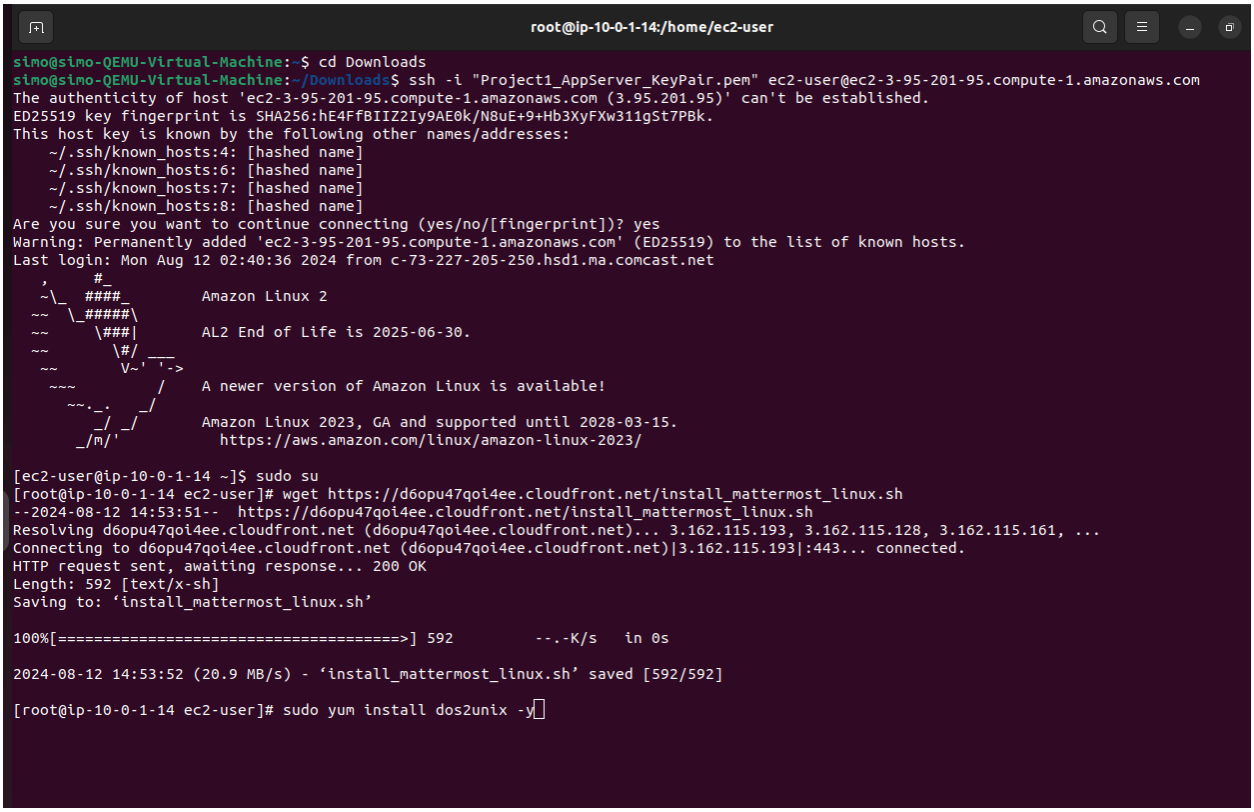


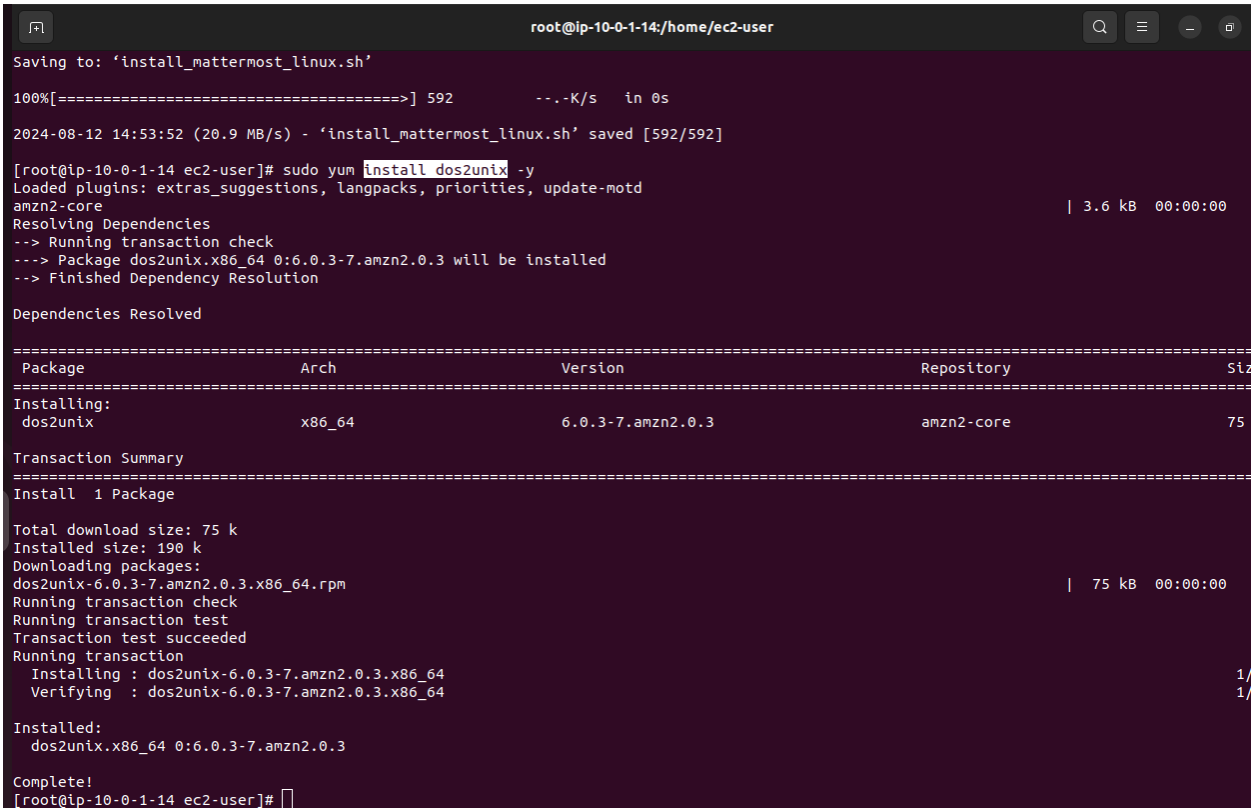
>

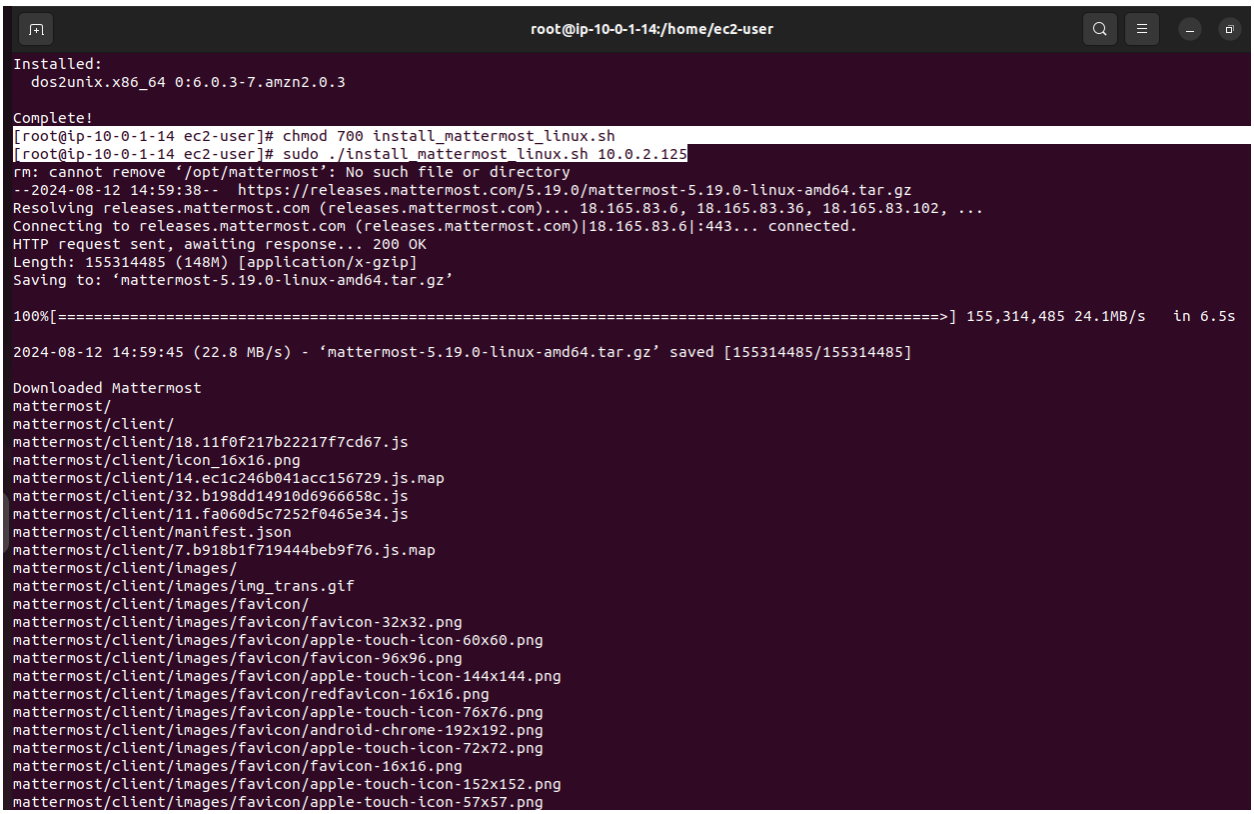
Step number	b
Step name	Installation and configuration of Mattermost
Instructions	1) Enter the following commands after logging into the application server via SSH to install and configure Mattermost. *wget https://d6opu47qoi4ee.cloudfront.net/install_mattermost_linux.sh* *sudo yum install dos2unix -y* *sudo dos2unix install_mattermost_linux.sh* *chmod 700 install_mattermost_linux.sh* *sudo ./install_mattermost_linux.sh YOUR_DB_PRIVATE_IP* *sudo chown -R mattermost:mattermost /opt/mattermost* *sudo chmod -R g+w /opt/mattermost* *cd /opt/mattermost* *sudo -u mattermost ./bin/mattermost*

	2) Check whether the server has been successfully deployed by navigating to the following URL in your web browser. The web page might take a couple of minutes to load. http://YOUR_APP_PUBLIC_IP:8065
Expected screenshots	1) Executing the script 2) Starting the Mattermost server 3) Accessing the application via web browser

<Insert screenshot b(1) here - Executing the script

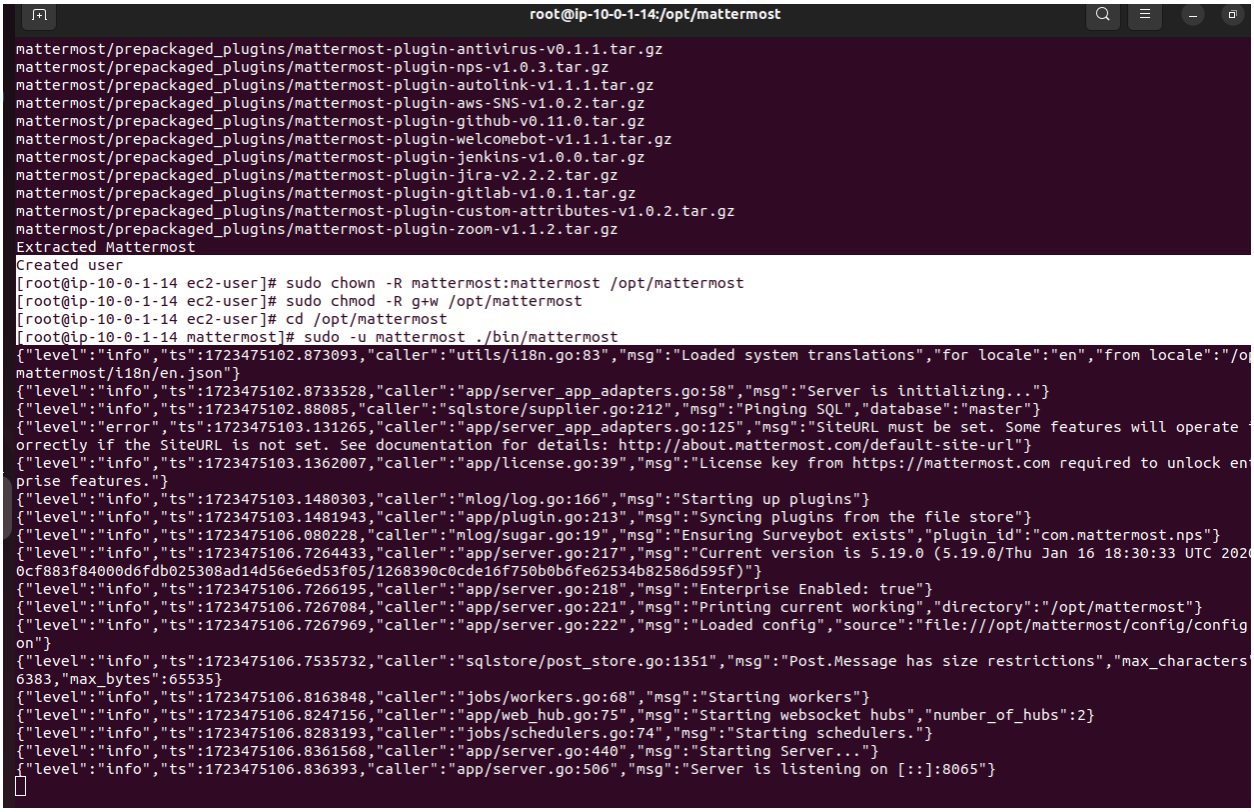






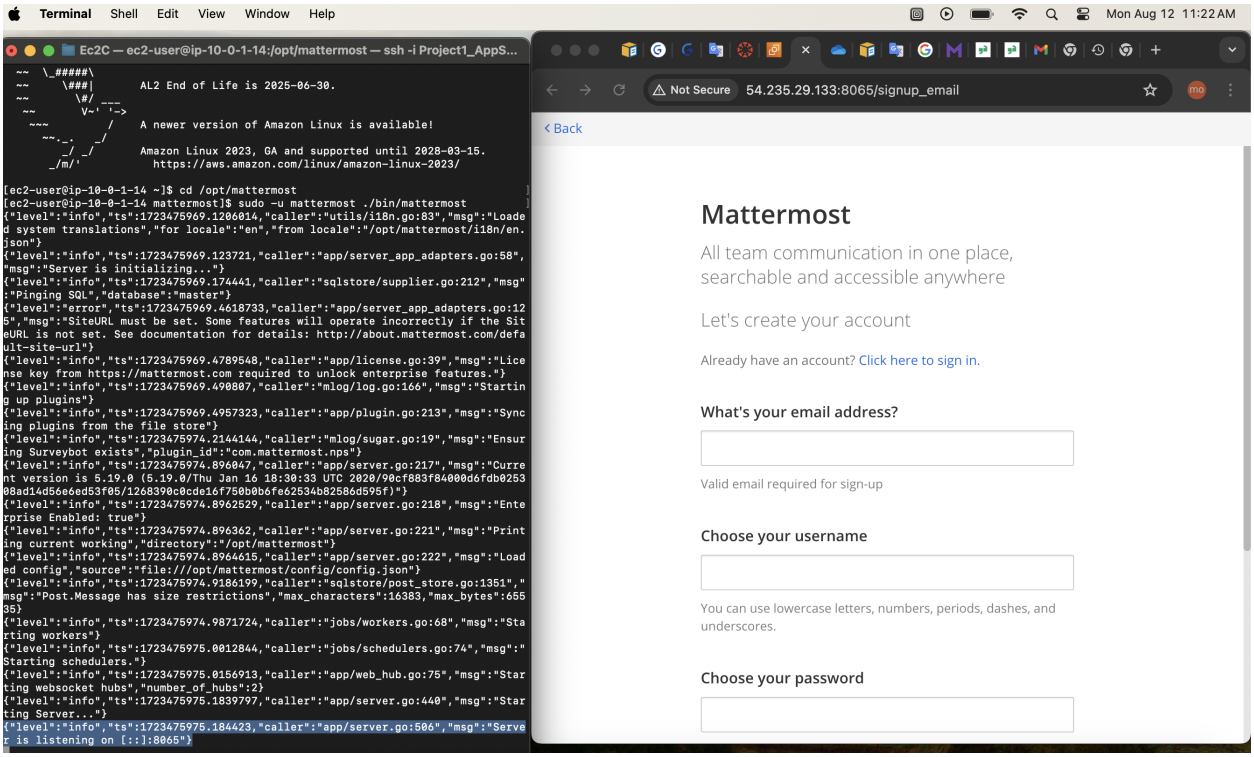
>

<Insert screenshot b(2) here - Starting the Mattermost server



>

<Insert screenshot b(3) here - Accessing the application via web browser



>

Step 5: Answer the following questions

Q1 What is the default setting for DNS hostnames when a new VPC is created?

a) Enabled

b) Disabled

c) Can be set during VPC creation

d) Depends on the region used

Enter your answer here: Disabled

Q2 What is the term used for the machine when we use it to log into the database server?

a) Bastion Host

b) NAT Gateway

c) Tunnel Interface

d) SSH Gateway

Enter your answer here: Bastion Host

Q3 The database server security group in this exercise has to keep port 3306 open. Which protocol uses this port to communicate?

a) HTTPS

b) RDP

c) TCP

d) SCP

Enter your answer here | TCP

Q4 Which port is being used by Mattermost to communicate with the client application

a) 8080

b) 80

c) 443

d) 8065

Enter your answer here | 8065

Q5 Which of the following is a reason why we cannot set the CIDR block for the public subnet to 10.0.2.0/16, assuming the values for the other CIDR blocks are the same as mentioned in the instructions?

a) CIDR block overlaps with existing block

b) CIDR block is not a valid CIDR

c) CIDR block does not fall within the VPC

d) There is no reason, this is a perfectly valid CIDR

Enter your answer here | CIDR block overlaps with existing block

Q6 Assume that you have been asked to create 3 EC2 instances - application server, the database server and NAT instance. Each of these instances have their own security groups with a set of ports to be kept open. One of these ports is entirely unnecessary for the given architecture to function. Which of the following could it be?

a) Port 22 on the NAT instances

b) Port 3306 on the database server

c) Port 3306 on the application server

d) Port 22 on the application server

Enter your answer here | Port 3306 on the application server

Q7 How are we going to increase the security of the Mattermost server to ensure the users are from a specific organization and the traffic is originating from a known IP address?

By Enabling or configuring the following:

Security Groups:
Define inbound and outbound traffic rules for EC2 instances.
Allow traffic only from specific IP addresses or security groups.

Network ACLs (Network Access Control Lists):
Control traffic at the subnet level for additional security.
Deny all traffic by default and allow only necessary traffic.

AWS WAF (Web Application Firewall):
Protect web applications from common web exploits.
Filter traffic based on IP addresses, country, and other criteria.

AWS Shield:
Protect against DDoS attacks.

CloudWatch:
Monitor VPC flow logs to track network traffic.
Set up alarms for suspicious activity.

IP Whitelisting: Only allow access from specific IP addresses or IP ranges.

VPN or Proxy: Require users to access resources through a VPN or proxy for added security.

IAM Roles with Temporary Credentials: Limit the lifetime of credentials for enhanced security (required new credentials every 3 months).

Q8 How do we achieve elasticity for the Mattermost server?

Auto Scaling Group (ASG)
The ASG automatically scales the number of Mattermost instances based on load.
Load Balancing
The ALB distributes traffic across healthy instances.
Monitoring and Alerting:
- Amazon CloudWatch to monitor Mattermost server performance and set up alarms for critical metrics.
- Integrate with CloudWatch Events to trigger actions based on alarm states.
DR plan: implement a Disaster Recovery plan in case of any disater

Grade distribution	
MCQs	6 (1 point each)
Subjective questions	10 (5 points each)
Implementation screenshots	24 points (1 point each)
Total	40 points