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	Rik KISNAH
	<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 marks
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

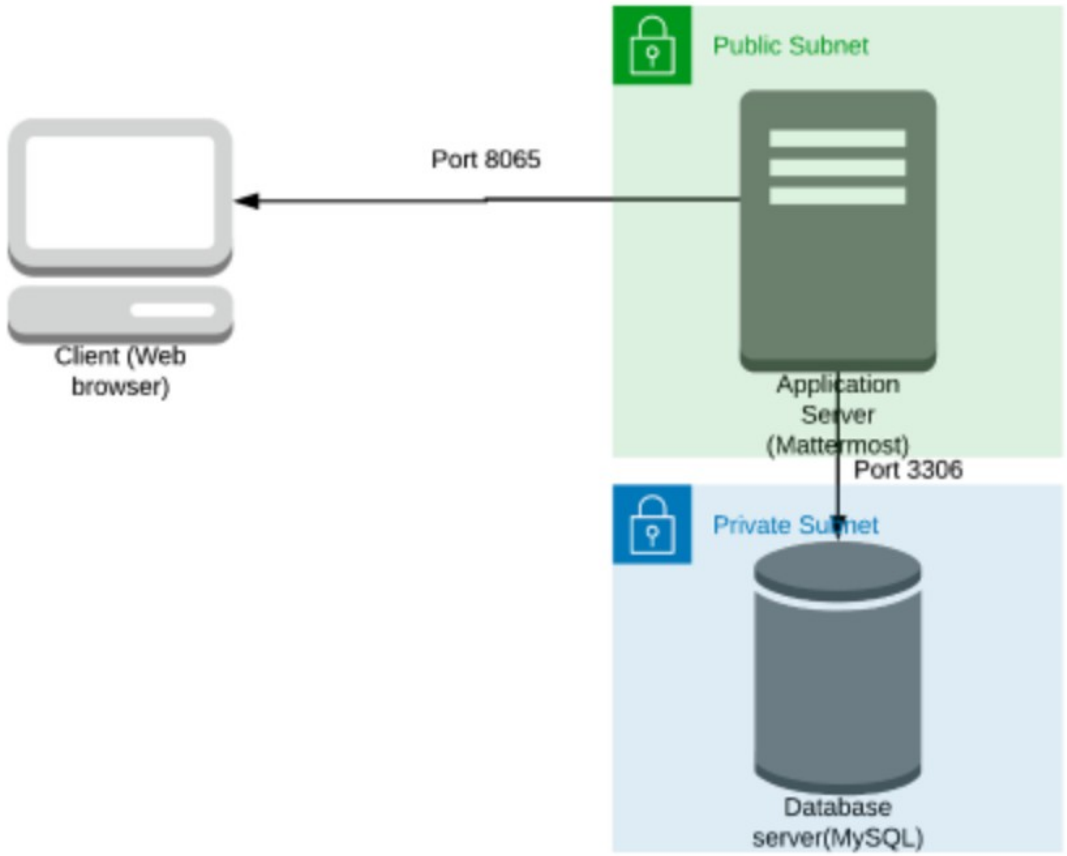
Resource Clean Up
Cloud is always pay per use model and all resources/services that we consume are chargeable. Cleaning up when you've completed your lab or project is always necessary. This is true whether you're doing a lab or implementing a project at your workplace.
After completing the lab, make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



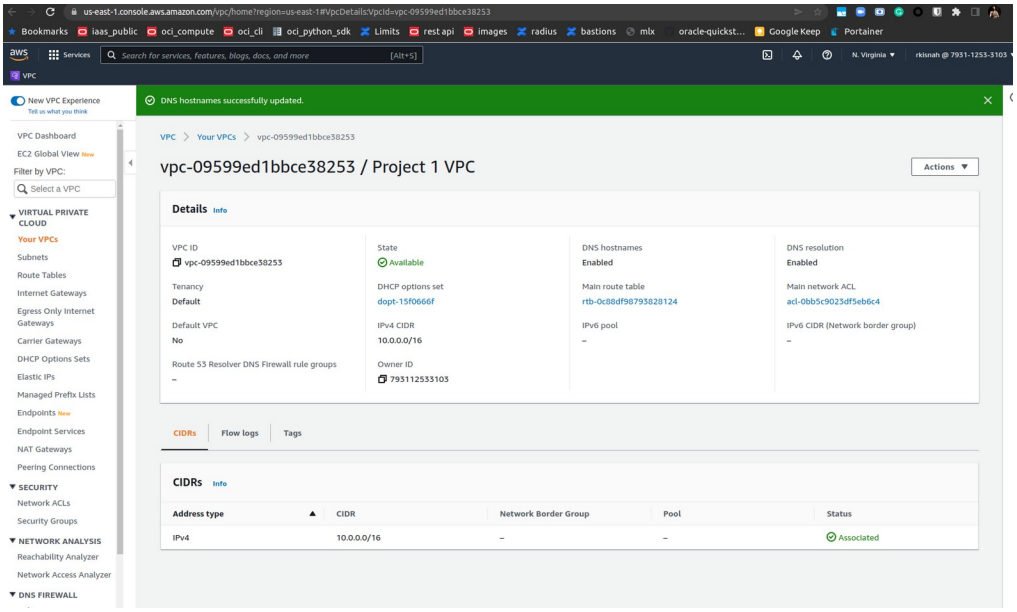
Architecture Implementation	
1	Implement 2 different subnets (one public and the other private) in a custom VPC
2	Install and configure MySQL on an Amazon Linux 2 instance on the private subnet using the instructions provided. (Hint: Use a bastion host and a NAT gateway)
3	Install and configure Mattermost 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

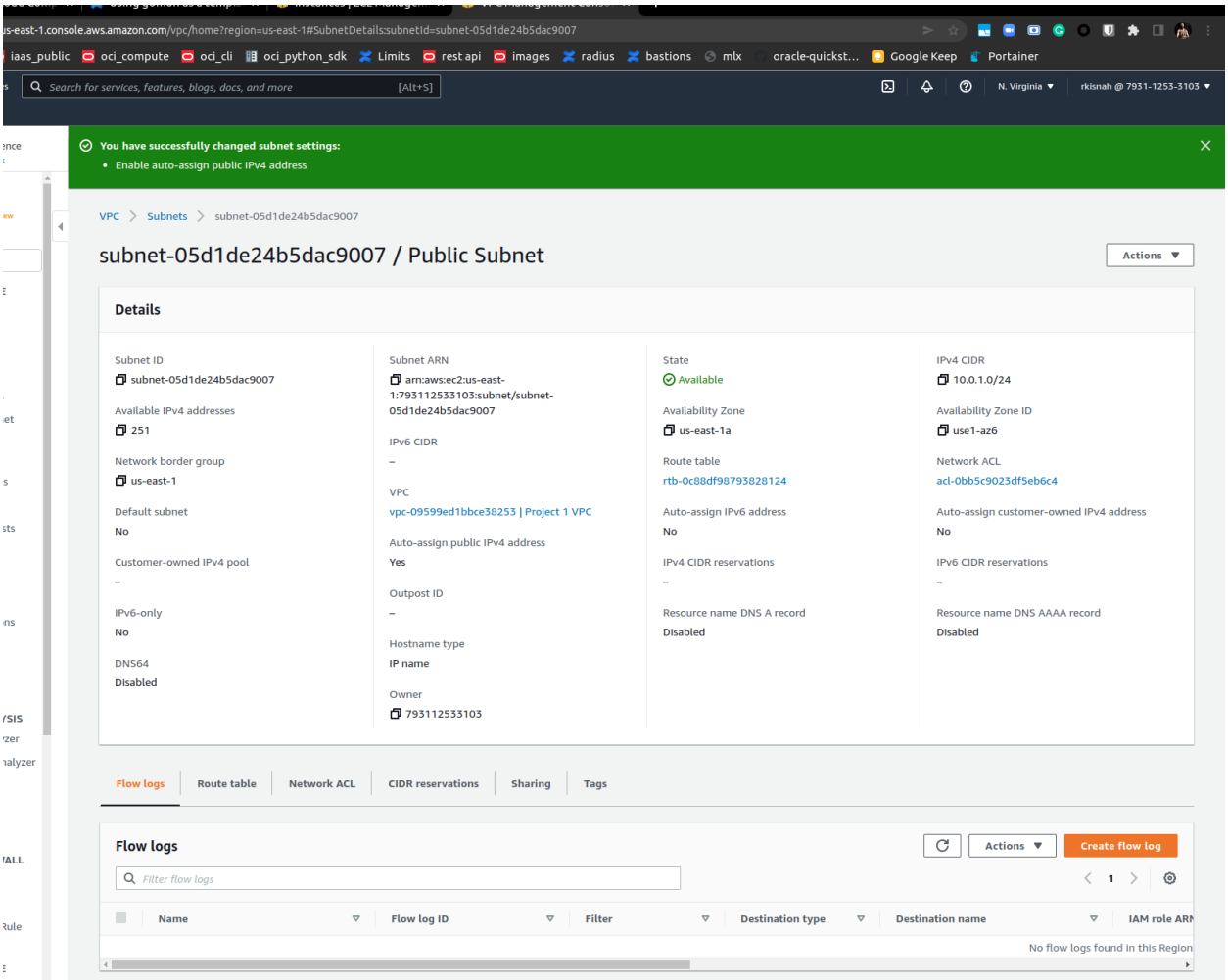
Step number	a	
Step name	Creation of VPC	
Instructions	1) Navigate to VPC using the Services button at the top of the screen 2) Select "Your VPCs" on the left side of the screen 3) Click on "Create VPC" 4) Enter the following fields : Name: Project 1 VPC IPv4 CIDR Block : 10.0.0.0/16 The rest of the 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	done
Expected screenshots	1) Created VPC with properties visible	

<Insert Screenshot a(1) here>



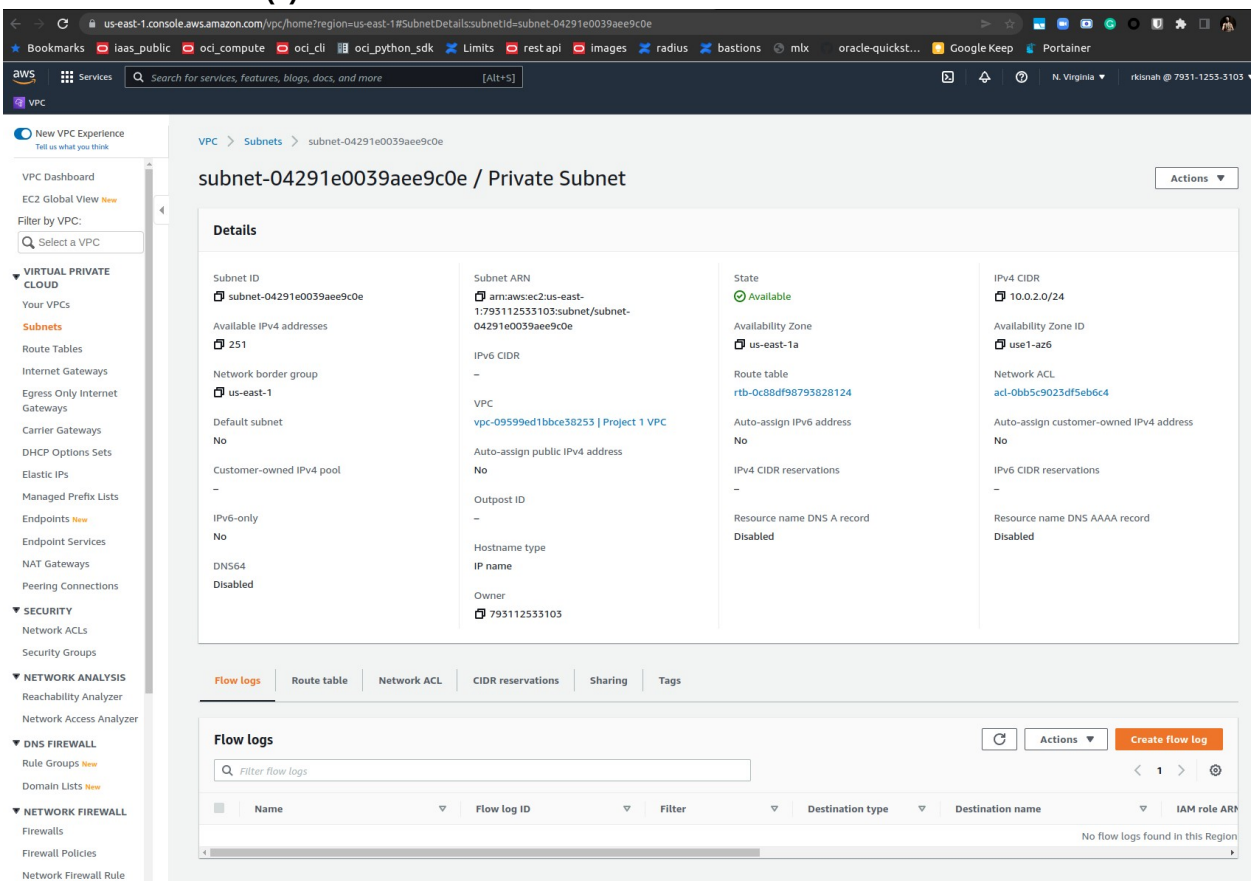
Step number	b
Step name	Creation of public subnet
Instructions	1) Navigate to VPC->Subnets 2) Click on "Create Subnet" 3) Enter the following fields Name tag : Public Subnet VPC : Select the Project 1 VPC IPv4 CIDR block : 10.0.1.0/24 The other options can be ignored 4) Click on Create 5) Once the subnet 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	1) 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	1) Subnet Creation screen

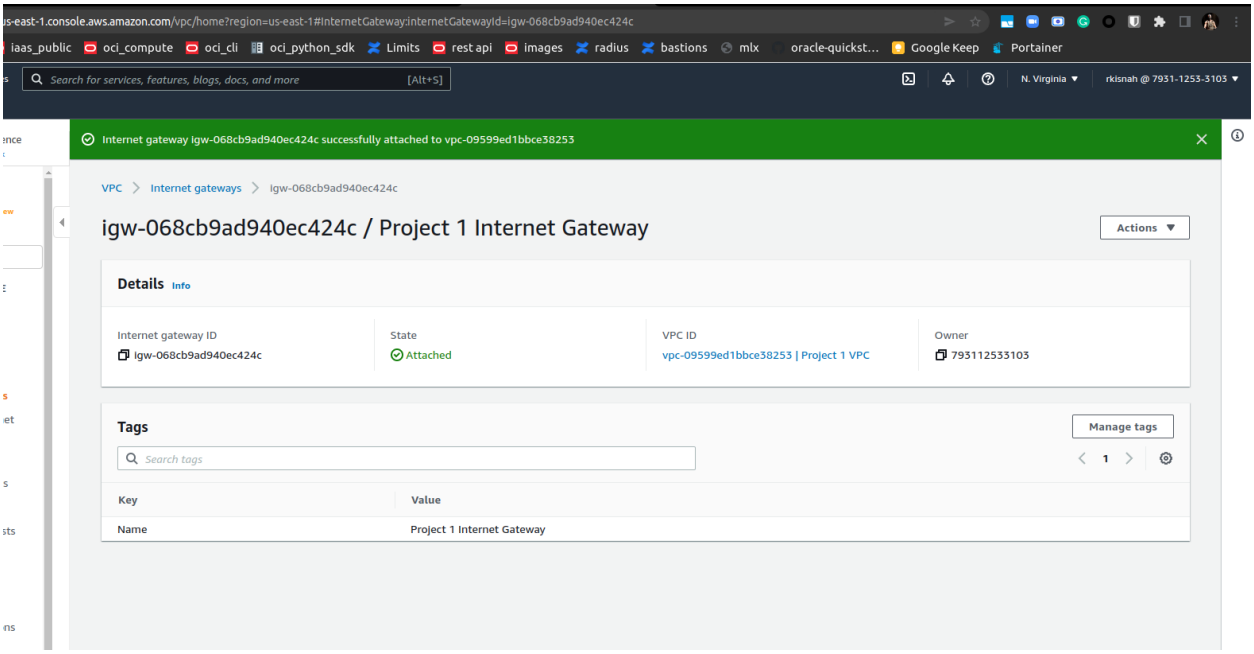
<Insert Screenshot c(1) here>



Step 2 : Internet Gateway and VPC

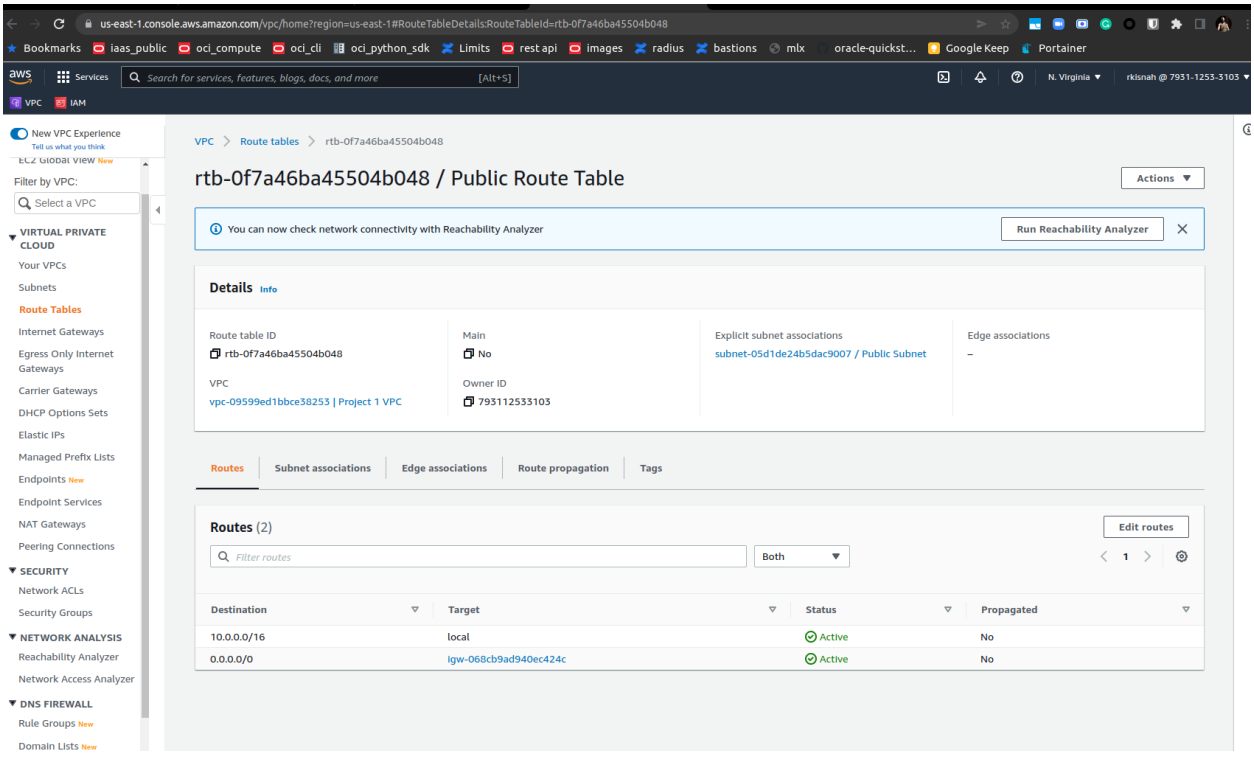
Step number	a
Step name	Creation and Configuration of Internet Gateway
Instructions	1) Navigate to VPCs->Internet Gateway 2) Click on "Create Internet Gateway" 3) Enter the name tag "Project 1 Internet Gateway" and click on "Create Internet Gateway" 4) After the gateway is created, select it and click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

<Insert Screenshot a(1) here >

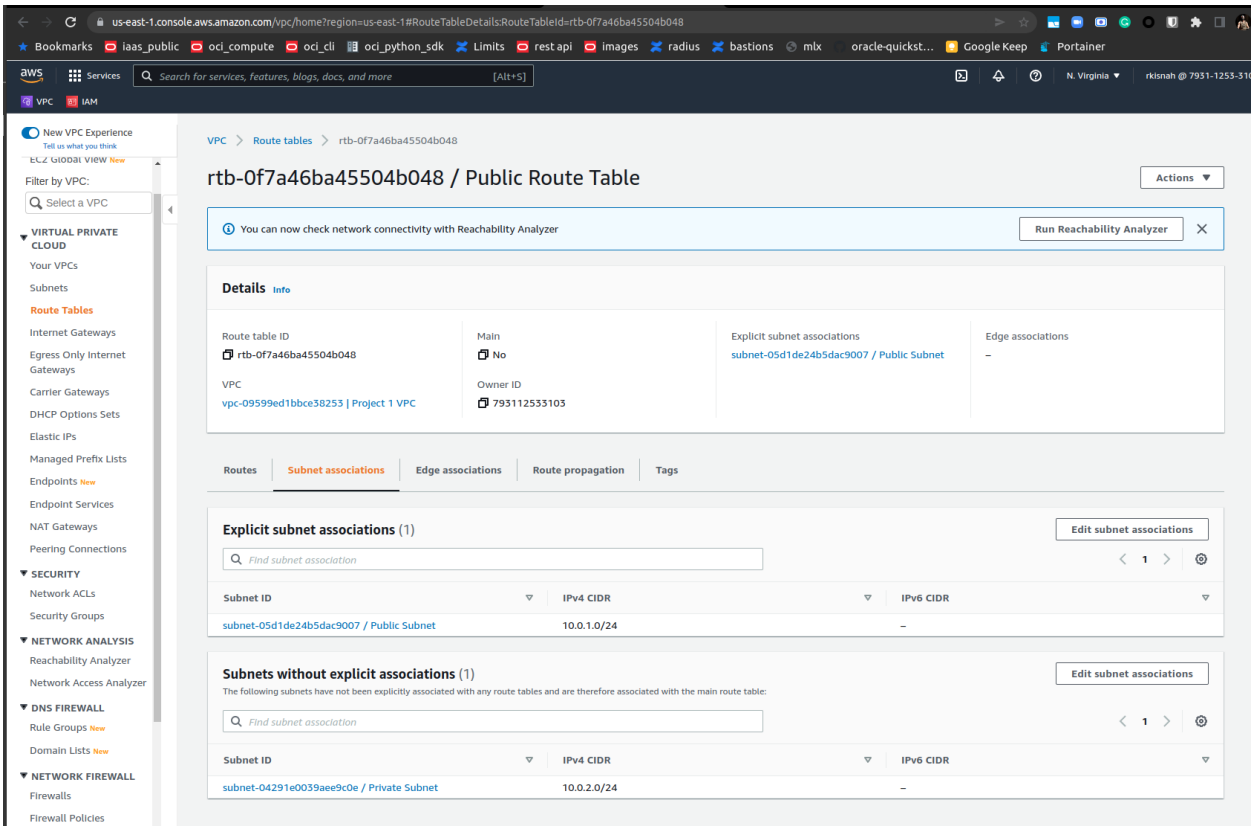


Step number	b
Step name	Creation of public route table
Instructions	1) Navigate to VPC -> Route Tables and click on Create Route table 2) Enter the name tag "Public Route Table", select the Project 1 VPC from the dropdown and click on Create 3) Once the route table is created, select it and select the 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>

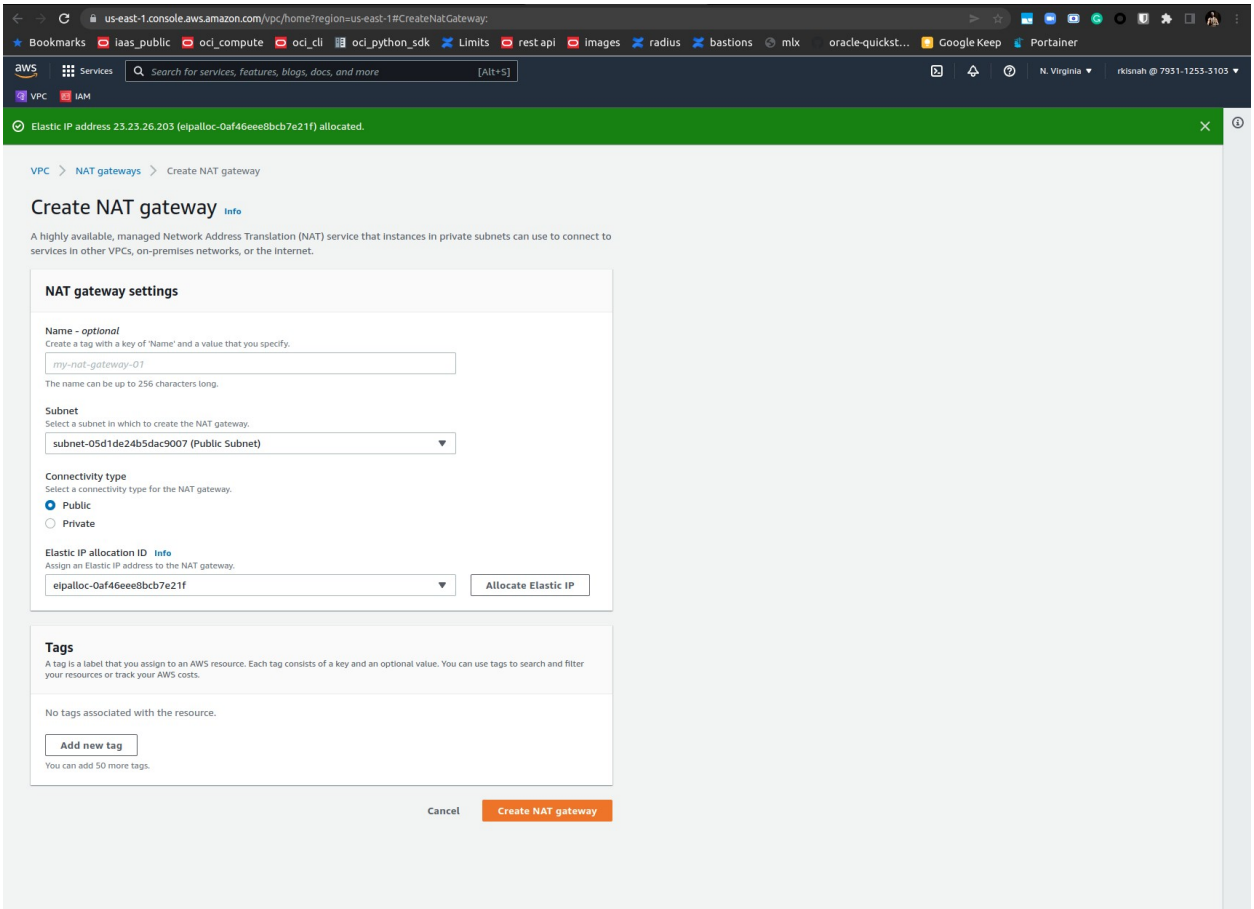


<Insert Screenshot b(2) here>

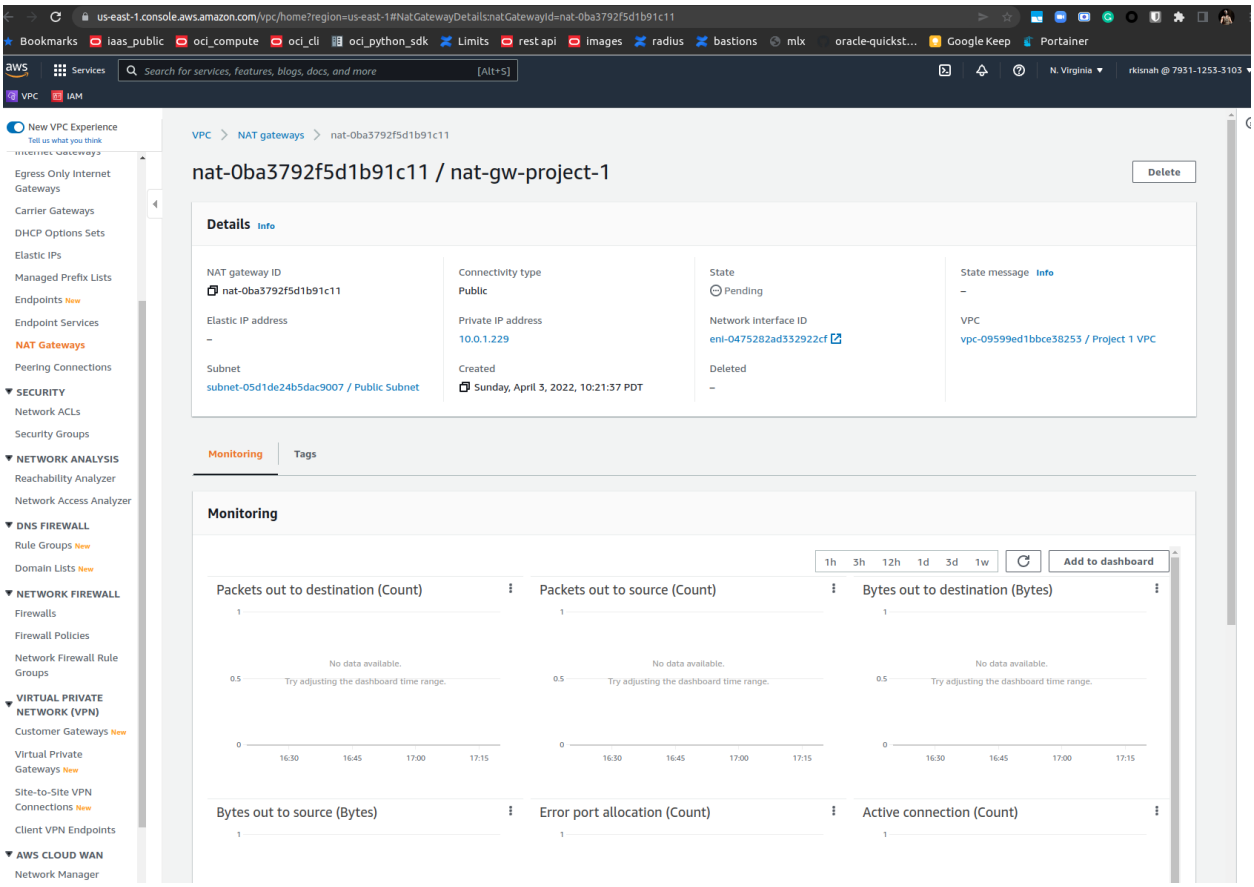


Step number	c
Step name	Creation of NAT gateway
Instructions	1) Navigate to VPC using the Services button at the top of the screen 2) Select NAT Gateway at the left side of the screen 3) Click on Create NAT Gateway - Deploy it in the public subnet - Connectivity type : Public - Allocate an elastic IP 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>

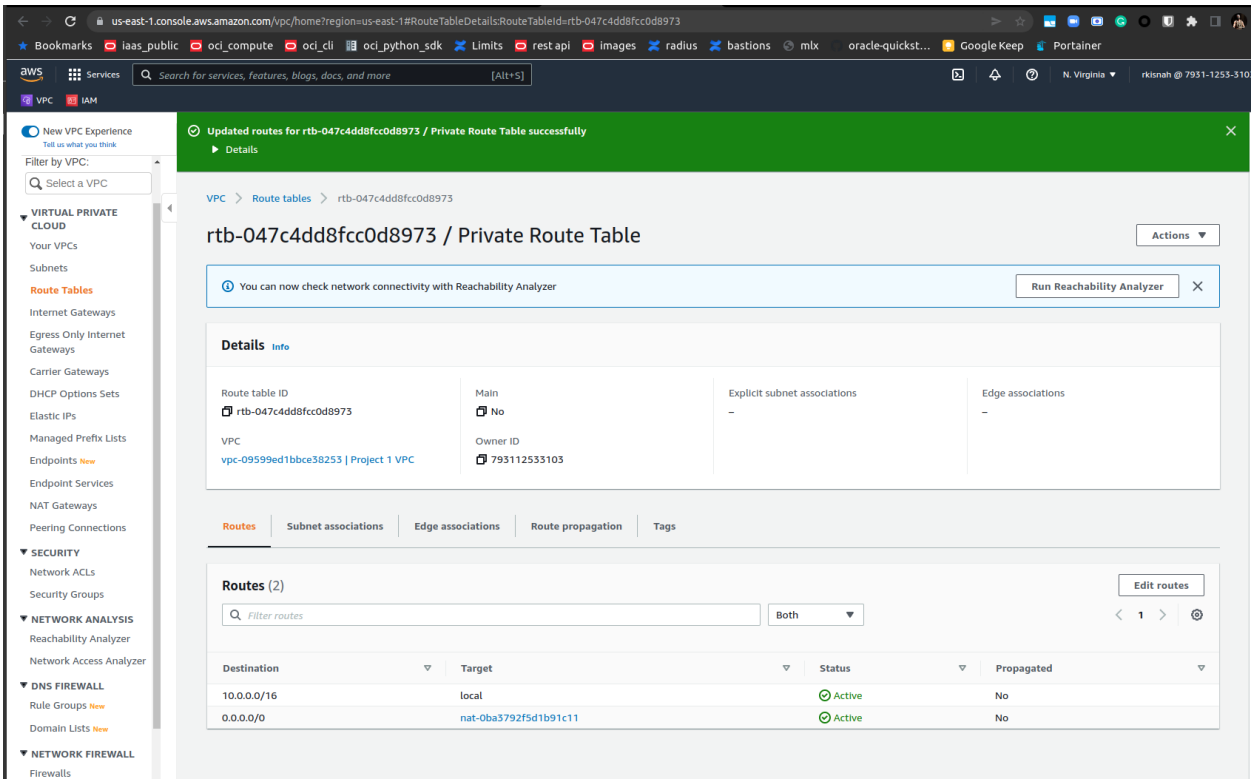


<Insert Screenshot c(2) here>

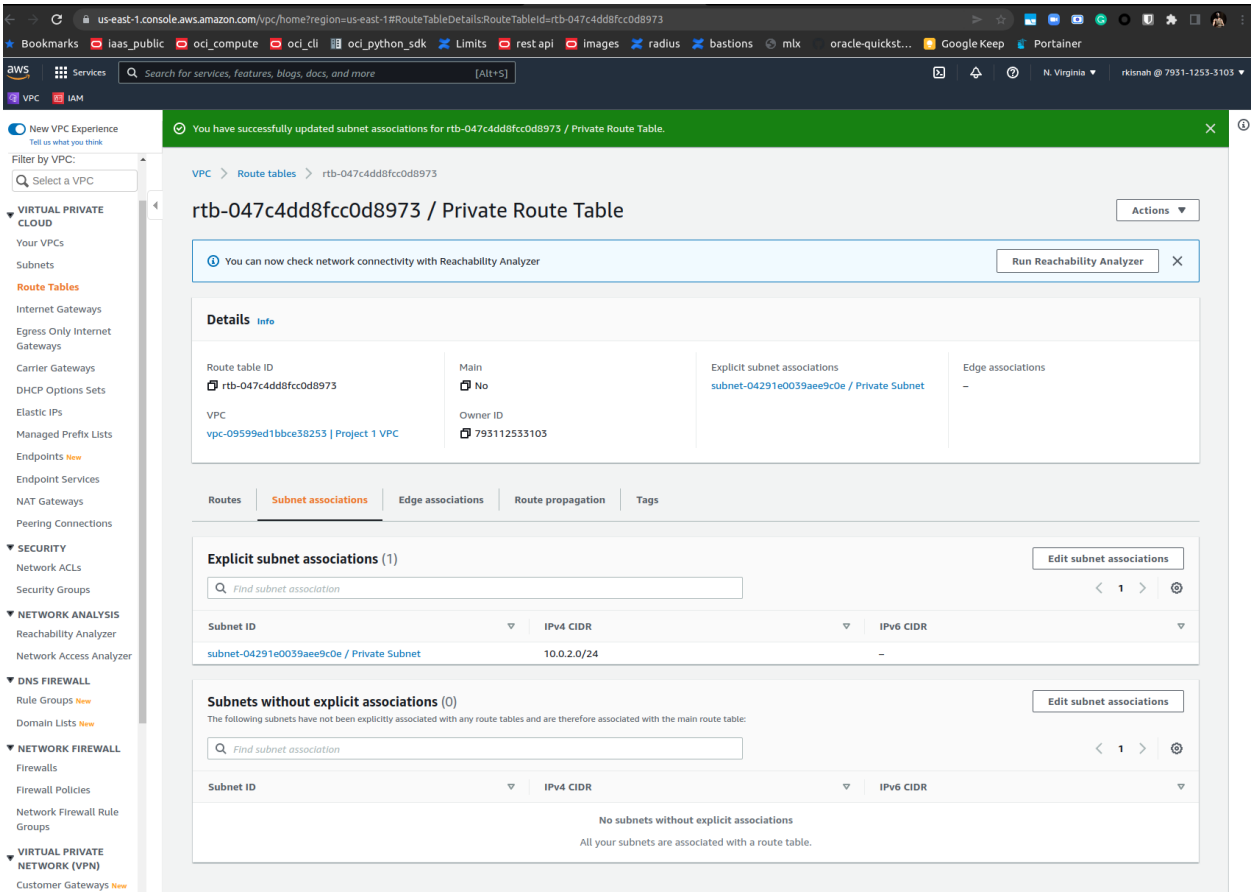


Step number	d
Step name	Creation of private route tables
Instructions	1) Navigate to VPC -> Route Tables and click on Create Route table 2) Enter the name tag "Private Route Table", select the Project 1 VPC from the dropdown and click on Create 3) Once the route table is created, select it and select 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 >



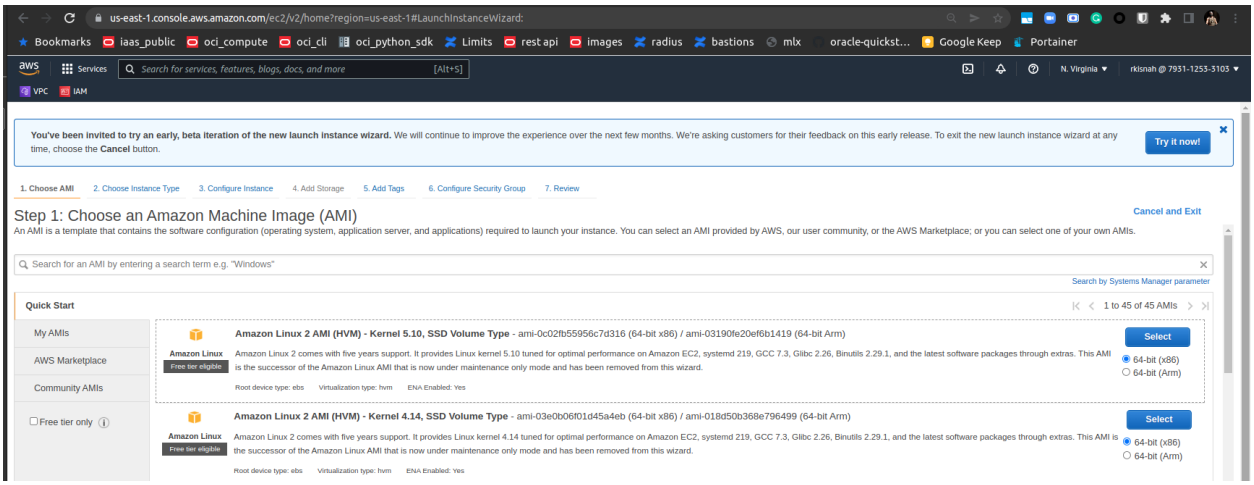
<Insert Screenshot for d(2) here>



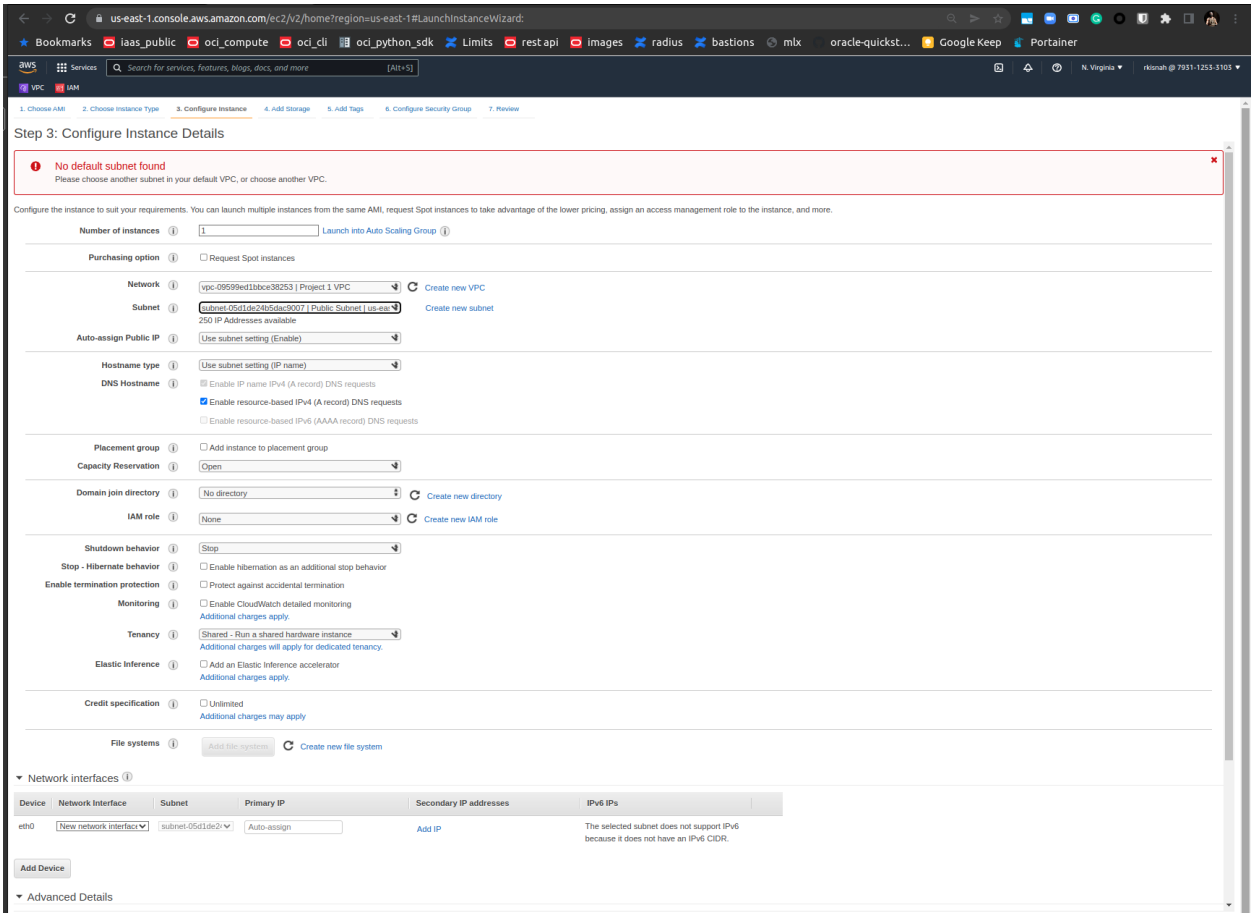
Step 3 : Creation of database and application servers

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

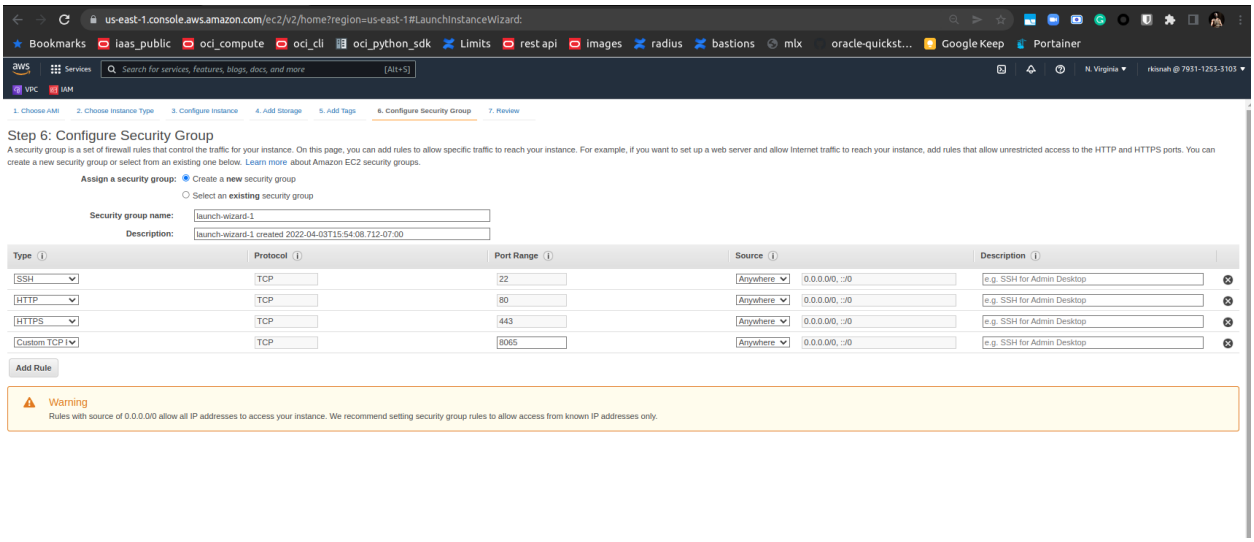
<Insert screenshot a(1) here>



<Insert screenshot a(2) here>

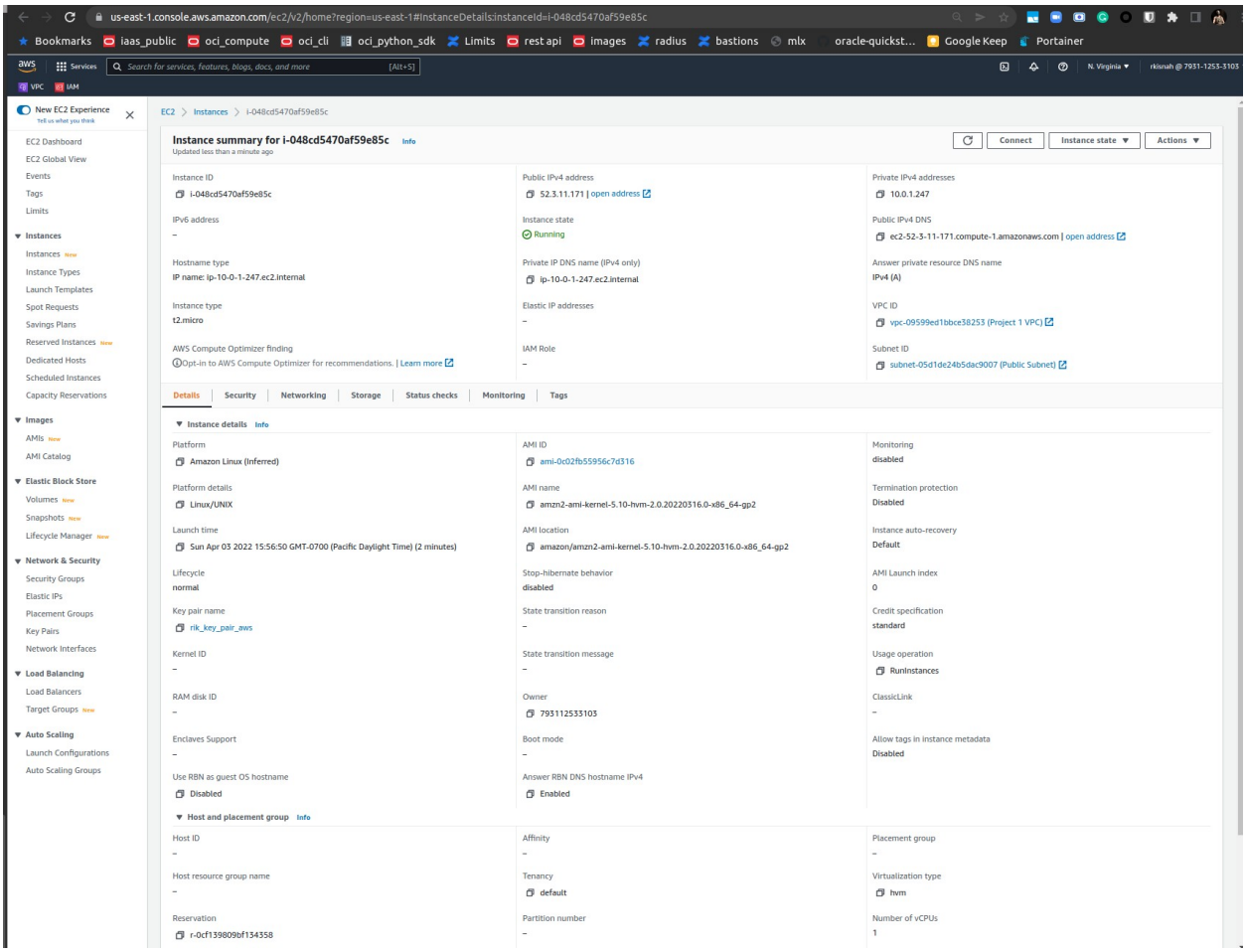


<Insert screenshot a(3) here>



<Insert screenshot a(4) here>

<Insert screenshot a(4) here>


us-east-1.console.aws.amazon.com/ec2/v2/home?region=us-east-1#InstanceDetails:instanceId=i-048cd5470af59e85c
Bookmarks iaas_public oci_compute oci_cli oci_python_sdk Limits rest api images radius bastions mlx oracle-quickst... Google Keep Portainer
Services Search for services, features, blogs, docs, and more [Alt+S] N. Virginia rkisnah @ 7931-1253-3103
VPC IAM
New EC2 Experience
Tell us what you think
EC2 Dashboard
EC2 Global View
Events
Tags
Limits
Instances
Instances New
Instance Types
Launch Templates
Spot Requests
Savings Plans
Reserved Instances New
Dedicated Hosts
Scheduled Instances
Capacity Reservations
Images
AMIs New
AMI Catalog
Elastic Block Store
Volumes New
Snapshots New
Lifecycle Manager New
Network & Security
Security Groups
Elastic IPs
Placement Groups
Key Pairs
Network Interfaces
Load Balancing
Load Balancers
Target Groups New
Auto Scaling
Launch Configurations
Auto Scaling Groups
EC2 > Instances > i-048cd5470af59e85c
Instance summary for i-048cd5470af59e85c Info
Updated less than a minute ago
Connect Instance state Actions
Instance ID
i-048cd5470af59e85c
Public IPv4 address
52.3.11.171 | open address
Private IPv4 addresses
10.0.1.247
IPv6 address
–
Instance state
Running
Public IPv4 DNS
ec2-52-3-11-171.compute-1.amazonaws.com | open address
Hostname type
IP name: ip-10-0-1-247.ec2.internal
Private IP DNS name (IPv4 only)
ip-10-0-1-247.ec2.internal
Answer private resource DNS name
IPv4 (A)
Instance type
t2.micro
Elastic IP addresses
–
VPC ID
vpc-09599ed1bbce38253 (Project 1 VPC)
AWS Compute Optimizer finding
Opt-in to AWS Compute Optimizer for recommendations. | Learn more
IAM Role
–
Subnet ID
subnet-05d1de24b5dac9007 (Public Subnet)
Details Security Networking Storage Status checks Monitoring Tags
Instance details Info
Platform
Amazon Linux (Inferred)
AMI ID
ami-0c02fb55956c7d316
Monitoring
disabled
Platform details
Linux/UNIX
AMI name
amzn2-ami-kernel-5.10-hvm-2.0.20220316.0-x86_64-gp2
Termination protection
Disabled
Launch time
Sun Apr 03 2022 15:56:50 GMT-0700 (Pacific Daylight Time) (2 minutes)
AMI location
amazon/amzn2-ami-kernel-5.10-hvm-2.0.20220316.0-x86_64-gp2
Instance auto-recovery
Default
Lifecycle
normal
Stop-hibernate behavior
disabled
AMI Launch index
0
Key pair name
rik_key_pair_aws
State transition reason
–
Credit specification
standard
Kernel ID
–
State transition message
–
Usage operation
RunInstances
RAM disk ID
–
Owner
793112533103
ClassicLink
–
Enclaves Support
–
Boot mode
–
Allow tags in instance metadata
Disabled
Use RBN as guest OS hostname
Disabled
Answer RBN DNS hostname IPv4
Enabled
Host and placement group Info
Host ID
–
Affinity
–
Placement group
–
Host resource group name
–
Tenancy
default
Virtualization type
hvm
Reservation
r-0cf139809bf134358
Partition number
–
Number of vCPUs
1

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>
<Insert screenshot b(2) here>
<Insert screenshot b(3) here>
<Insert screenshot b(4) here>

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 <application server pem file> <database server pem file > ec2-user@<application server public IP>:/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 1and the private IP address of the database server with the following command *ssh -i <database server pem file> ec2-user@<private IP of database server>* 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 *sudo grep 'temporary password' /var/log/mysqld.log | rev | cut -d" " -f1 | rev | tr -d "."* Log in to MySQL with the below command and enter the above password when prompted *mysql -u root -p* Enter the below command after you login to MySQL *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 3) Executing the provided script

<Insert screenshot a(1) here> - install mysql

the instructions are wrong outdated. Attached is the terminal text output.

```
ec2-user@ip-10-0-2-215 ~]$ mkdir downloads
```

```
[ec2-user@ip-10-0-2-215 ~]$ cd downloads/

[ec2-user@ip-10-0-2-215 downloads]$ wget http://dev.mysql.com/get/mysql57-
community-release-el7-9.noarch.rpm
-bash: wget http://dev.mysql.com/get/mysql57-community-release-el7-
9.noarch.rpm: No such file or directory
[ec2-user@ip-10-0-2-215 downloads]$ #wtf right??? instructions are dud



# Using dr google good ref: https://tecadmin.net/install-mysql-5-7-centos-
rhel/

[ec2-user@ip-10-0-2-215 downloads]$ sudo yum localinstall
https://dev.mysql.com/get/mysql57-community-release-el7-9.noarch.rpm
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
mysql57-community-release-el7-9.noarch.rpm                              | 9.0 kB
00:00:00
Examining /var/tmp/yum-root-jwDpAh/mysql57-community-release-el7-9.noarch.rpm:
mysql57-community-release-el7-9.noarch
Marking /var/tmp/yum-root-jwDpAh/mysql57-community-release-el7-9.noarch.rpm to
be installed
Resolving Dependencies
--> Running transaction check
---> Package mysql57-community-release.noarch 0:el7-9 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
===============
 Package                    Arch    Version   Repository
Size
================================================================================
===============
Installing:
 mysql57-community-release  noarch  el7-9     /mysql57-community-release-el7-
9.noarch  8.6 k

Transaction Summary
================================================================================
===============
Install  1 Package

Total size: 8.6 k
Installed size: 8.6 k
Is this ok [y/d/N]: y
Downloading packages:
Running transaction check
```

```
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mysql57-community-release-el7-9.noarch
1/1
  Verifying  : mysql57-community-release-el7-9.noarch
1/1

Installed:
  mysql57-community-release.noarch 0:el7-9

Complete!

[ec2-user@ip-10-0-2-215 downloads]$ yum install mysql-community-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
You need to be root to perform this command.
[ec2-user@ip-10-0-2-215 downloads]$ sudo yum install mysql-community-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                                               | 3.7 kB
00:00:00
49 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package mysql-community-server.x86_64 0:5.7.37-1.el7 will be installed
--> Processing Dependency: mysql-community-common(x86-64) = 5.7.37-1.el7 for
package: mysql-community-server-5.7.37-1.el7.x86_64
--> Processing Dependency: mysql-community-client(x86-64) >= 5.7.9 for
package: mysql-community-server-5.7.37-1.el7.x86_64
--> Running transaction check
---> Package mysql-community-client.x86_64 0:5.7.37-1.el7 will be installed
--> Processing Dependency: mysql-community-libs(x86-64) >= 5.7.9 for package:
mysql-community-client-5.7.37-1.el7.x86_64
--> Processing Dependency: libncurses.so.5()(64bit) for package: mysql-
community-client-5.7.37-1.el7.x86_64
--> Processing Dependency: libtinfo.so.5()(64bit) for package: mysql-
community-client-5.7.37-1.el7.x86_64
---> Package mysql-community-common.x86_64 0:5.7.37-1.el7 will be installed
--> Running transaction check
---> Package mariadb-libs.x86_64 1:5.5.68-1.amzn2 will be obsoleted
--> Processing Dependency: libmysqlclient.so.18()(64bit) for package:
2:postfix-2.10.1-6.amzn2.0.3.x86_64
--> Processing Dependency: libmysqlclient.so.18(libmysqlclient_18)(64bit) for
package: 2:postfix-2.10.1-6.amzn2.0.3.x86_64
---> Package mysql-community-libs.x86_64 0:5.7.37-1.el7 will be obsoleting
---> Package ncurses-compat-libs.x86_64 0:6.0-8.20170212.amzn2.1.3 will be
installed
--> Running transaction check
---> Package mysql-community-libs-compat.x86_64 0:5.7.37-1.el7 will be
obsoleting
```

```
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
===============
 Package                        Arch     Version                      Repository
Size
================================================================================
===============
Installing:
 mysql-community-libs           x86_64   5.7.37-1.el7                 mysql57-
community   2.4 M
     replacing  mariadb-libs.x86_64 1:5.5.68-1.amzn2
 mysql-community-libs-compat    x86_64   5.7.37-1.el7                 mysql57-
community   1.2 M
     replacing  mariadb-libs.x86_64 1:5.5.68-1.amzn2
 mysql-community-server         x86_64   5.7.37-1.el7                 mysql57-
community   174 M
Installing for dependencies:
 mysql-community-client         x86_64   5.7.37-1.el7                 mysql57-
community    25 M
 mysql-community-common         x86_64   5.7.37-1.el7                 mysql57-
community   311 k
 ncurses-compat-libs            x86_64   6.0-8.20170212.amzn2.1.3     amzn2-core
308 k

Transaction Summary
================================================================================
===============
Install  3 Packages (+3 Dependent packages)

Total download size: 203 M
Is this ok [y/d/N]: y
Downloading packages:
warning: /var/cache/yum/x86_64/2/mysql57-community/packages/mysql-community-
common-5.7.37-1.el7.x86_64.rpm: Header V4 RSA/SHA256 Signature, key ID
3a79bd29: NOKEY
Public key for mysql-community-common-5.7.37-1.el7.x86_64.rpm is not installed
(1/6): mysql-community-common-5.7.37-1.el7.x86_64.rpm                 | 311 kB
00:00:00
(2/6): mysql-community-libs-5.7.37-1.el7.x86_64.rpm                   | 2.4 MB
00:00:00
(3/6): mysql-community-libs-compat-5.7.37-1.el7.x86_64.rpm            | 1.2 MB
00:00:00
(4/6): mysql-community-client-5.7.37-1.el7.x86_64.rpm                 |  25 MB
00:00:00
(5/6): ncurses-compat-libs-6.0-8.20170212.amzn2.1.3.x86_64.rpm        | 308 kB
00:00:01
```

```
(6/6): mysql-community-server-5.7.37-1.el7.x86_64.rpm                 | 174 MB
00:00:02
--------------------------------------------------------------------------------
---------------
Total                                                        70 MB/s | 203 MB
00:00:02
Retrieving key from file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
Importing GPG key 0x5072E1F5:
 Userid     : "MySQL Release Engineering <mysql-build@oss.oracle.com>"
 Fingerprint: a4a9 4068 76fc bd3c 4567 70c8 8c71 8d3b 5072 e1f5
 Package    : mysql57-community-release-el7-9.noarch (installed)
 From       : /etc/pki/rpm-gpg/RPM-GPG-KEY-mysql
Is this ok [y/N]: y


Public key for mysql-community-libs-compat-5.7.37-1.el7.x86_64.rpm is not
installed


 Failing package is: mysql-community-libs-compat-5.7.37-1.el7.x86_64
 GPG Keys are configured as: file:///etc/pki/rpm-gpg/RPM-GPG-KEY-mysql

[ec2-user@ip-10-0-2-215 downloads]$ sudo yum install mysql-community-server -y
--nogpgcheck
-bash: sudo yum install mysql-community-server: command not found
[ec2-user@ip-10-0-2-215 downloads]$ ^Cdo yum install mysql-community-server -y
--nogpgcheck
[ec2-user@ip-10-0-2-215 downloads]$ mysq^C
[ec2-user@ip-10-0-2-215 downloads]$ yum install mysql-community-server
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
You need to be root to perform this command.
[ec2-user@ip-10-0-2-215 downloads]$ sudo yum install mysql-community-server -y
--nogpgcheck
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
49 packages excluded due to repository priority protections
Resolving Dependencies
--> Running transaction check
---> Package mysql-community-server.x86_64 0:5.7.37-1.el7 will be installed
--> Processing Dependency: mysql-community-common(x86-64) = 5.7.37-1.el7 for
package: mysql-community-server-5.7.37-1.el7.x86_64
--> Processing Dependency: mysql-community-client(x86-64) >= 5.7.9 for
package: mysql-community-server-5.7.37-1.el7.x86_64
--> Running transaction check
---> Package mysql-community-client.x86_64 0:5.7.37-1.el7 will be installed
--> Processing Dependency: mysql-community-libs(x86-64) >= 5.7.9 for package:
mysql-community-client-5.7.37-1.el7.x86_64
--> Processing Dependency: libncurses.so.5()(64bit) for package: mysql-
community-client-5.7.37-1.el7.x86_64
```

```
--> Processing Dependency: libtinfo.so.5()(64bit) for package: mysql-
community-client-5.7.37-1.el7.x86_64
---> Package mysql-community-common.x86_64 0:5.7.37-1.el7 will be installed
--> Running transaction check
---> Package mariadb-libs.x86_64 1:5.5.68-1.amzn2 will be obsoleted
--> Processing Dependency: libmysqlclient.so.18()(64bit) for package:
2:postfix-2.10.1-6.amzn2.0.3.x86_64
--> Processing Dependency: libmysqlclient.so.18(libmysqlclient_18)(64bit) for
package: 2:postfix-2.10.1-6.amzn2.0.3.x86_64
---> Package mysql-community-libs.x86_64 0:5.7.37-1.el7 will be obsoleting
---> Package ncurses-compat-libs.x86_64 0:6.0-8.20170212.amzn2.1.3 will be
installed
--> Running transaction check
---> Package mysql-community-libs-compat.x86_64 0:5.7.37-1.el7 will be
obsoleting
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
===============
 Package                        Arch     Version                      Repository
Size
================================================================================
===============
Installing:
 mysql-community-libs           x86_64   5.7.37-1.el7                 mysql57-
community   2.4 M
     replacing  mariadb-libs.x86_64 1:5.5.68-1.amzn2
 mysql-community-libs-compat    x86_64   5.7.37-1.el7                 mysql57-
community   1.2 M
     replacing  mariadb-libs.x86_64 1:5.5.68-1.amzn2
 mysql-community-server         x86_64   5.7.37-1.el7                 mysql57-
community   174 M
Installing for dependencies:
 mysql-community-client         x86_64   5.7.37-1.el7                 mysql57-
community    25 M
 mysql-community-common         x86_64   5.7.37-1.el7                 mysql57-
community   311 k
 ncurses-compat-libs            x86_64   6.0-8.20170212.amzn2.1.3     amzn2-core
308 k

Transaction Summary
================================================================================
===============
Install  3 Packages (+3 Dependent packages)

Total size: 203 M
Downloading packages:
```

```
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : mysql-community-common-5.7.37-1.el7.x86_64
1/7
  Installing : mysql-community-libs-5.7.37-1.el7.x86_64
2/7
  Installing : ncurses-compat-libs-6.0-8.20170212.amzn2.1.3.x86_64
3/7
  Installing : mysql-community-client-5.7.37-1.el7.x86_64
4/7
  Installing : mysql-community-server-5.7.37-1.el7.x86_64
5/7
  Installing : mysql-community-libs-compat-5.7.37-1.el7.x86_64
6/7
  Erasing    : 1:mariadb-libs-5.5.68-1.amzn2.x86_64
7/7
  Verifying  : ncurses-compat-libs-6.0-8.20170212.amzn2.1.3.x86_64
1/7
  Verifying  : mysql-community-libs-compat-5.7.37-1.el7.x86_64
2/7
  Verifying  : mysql-community-libs-5.7.37-1.el7.x86_64
3/7
  Verifying  : mysql-community-common-5.7.37-1.el7.x86_64
4/7
  Verifying  : mysql-community-server-5.7.37-1.el7.x86_64
5/7
  Verifying  : mysql-community-client-5.7.37-1.el7.x86_64
6/7
  Verifying  : 1:mariadb-libs-5.5.68-1.amzn2.x86_64
7/7

Installed:
  mysql-community-libs.x86_64 0:5.7.37-1.el7
  mysql-community-libs-compat.x86_64 0:5.7.37-1.el7
  mysql-community-server.x86_64 0:5.7.37-1.el7

Dependency Installed:
  mysql-community-client.x86_64 0:5.7.37-1.el7
  mysql-community-common.x86_64 0:5.7.37-1.el7
  ncurses-compat-libs.x86_64 0:6.0-8.20170212.amzn2.1.3

Replaced:
  mariadb-libs.x86_64 1:5.5.68-1.amzn2

Complete!
```

```
[root@ip-10-0-2-215 log]# systemctl start mysqld

[root@ip-10-0-2-215 downloads]# sudo systemctl status mysqld
● mysqld.service - MySQL Server
   Loaded: loaded (/usr/lib/systemd/system/mysqld.service; enabled; vendor
preset: disabled)
   Active: active (running) since Sun 2022-04-03 23:19:17 UTC; 9min ago
     Docs: man:mysqld(8)
           http://dev.mysql.com/doc/refman/en/using-systemd.html
  Process: 3713 ExecStart=/usr/sbin/mysqld --daemonize
--pid-file=/var/run/mysqld/mysqld.pid $MYSQLD_OPTS (code=exited,
status=0/SUCCESS)
  Process: 3664 ExecStartPre=/usr/bin/mysqld_pre_systemd (code=exited,
status=0/SUCCESS)
 Main PID: 3717 (mysqld)
   CGroup: /system.slice/mysqld.service
           └─3717 /usr/sbin/mysqld --daemonize
--pid-file=/var/run/mysqld/mysqld.pid

Apr 03 23:19:11 ip-10-0-2-215.ec2.internal systemd[1]: Starting MySQL
Server...
Apr 03 23:19:17 ip-10-0-2-215.ec2.internal systemd[1]: Started MySQL Server.
[root@ip-10-0-2-215 downloads]#
```

<Insert screenshot a(2) here> - retrieve temp pwd

```
root@ip-10-0-2-215 log]# vim mysqld.log
[root@ip-10-0-2-215 log]# grep 'A temporary password' /var/log/mysqld.log |tail -1
2022-04-03T23:19:14.591543Z 1 [Note] A temporary password is generated for root@localhost: lH%*,#jwy70_
[root@ip-10-0-2-215 log]# ^C
[root@ip-10-0-2-215 log]# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.7.37

Copyright (c) 2000, 2022, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'Password42!';
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near
'ALTER USER 'root'@'localhost' IDENTIFIED BY 'Password42!'' at line 1
mysql> ^DBye
[root@ip-10-0-2-215 log]# /usr/bin/mysql_secure_installation

Securing the MySQL server deployment.

Enter password for user root:

The existing password for the user account root has expired. Please set a new password.

New password:

Re-enter new password:
The 'validate_password' plugin is installed on the server.
The subsequent steps will run with the existing configuration
of the plugin.
Using existing password for root.

Estimated strength of the password: 100
```

Change the password for root ? ((Press y|Y for Yes, any other key for No) : y

New password:

Re-enter new password:

Estimated strength of the password: 100
Do you wish to continue with the password provided?(Press y|Y for Yes, any other key for No) : Y
By default, a MySQL installation has an anonymous user,
allowing anyone to log into MySQL without having to have
a user account created for them. This is intended only for
testing, and to make the installation go a bit smoother.
You should remove them before moving into a production
environment.

Remove anonymous users? (Press y|Y for Yes, any other key for No) :

 ... skipping.

Normally, root should only be allowed to connect from
'localhost'. This ensures that someone cannot guess at
the root password from the network.

Disallow root login remotely? (Press y|Y for Yes, any other key for No) :

 ... skipping.
By default, MySQL comes with a database named 'test' that
anyone can access. This is also intended only for testing,
and should be removed before moving into a production
environment.

Remove test database and access to it? (Press y|Y for Yes, any other key for No) :

 ... skipping.
Reloading the privilege tables will ensure that all changes
made so far will take effect immediately.

Reload privilege tables now? (Press y|Y for Yes, any other key for No) :

 ... skipping.
All done!
[root@ip-10-0-2-215 log]# mysql -h localhost -u root -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 6
Server version: 5.7.37 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> exit
Bye
[root@ip-10-0-2-215 log]# cd

<Insert screenshot a(3) here> - execute provided script

```
[ec2-user@ip-10-0-2-215 downloads]$ wget wget https://d6opu47qoi4ee.cloudfront.net/install_mysql_linux.sh
--2022-04-03 23:23:09--  http://wget/
Resolving wget (wget)... failed: Name or service not known.
wget: unable to resolve host address ‘wget’
--2022-04-03 23:23:09--  https://d6opu47qoi4ee.cloudfront.net/install_mysql_linux.sh
Resolving d6opu47qoi4ee.cloudfront.net (d6opu47qoi4ee.cloudfront.net)... 13.32.204.92, 13.32.204.110, 13.32.204.150, ...
Connecting to d6opu47qoi4ee.cloudfront.net (d6opu47qoi4ee.cloudfront.net)|13.32.204.92|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 189 [text/x-sh]
Saving to: ‘install_mysql_linux.sh’

100%[======================================================>] 189         --.-K/s   in 0s

2022-04-03 23:23:09 (27.1 MB/s) - ‘install_mysql_linux.sh’ saved [189/189]

FINISHED --2022-04-03 23:23:09--
Total wall clock time: 0.06s
Downloaded: 1 files, 189 in 0s (27.1 MB/s)
[ec2-user@ip-10-0-2-215 downloads]$ ls
install_mysql_linux.sh
[ec2-user@ip-10-0-2-215 downloads]$ chmod +x install_mysql_linux.sh
[ec2-user@ip-10-0-2-215 downloads]$ chmod 777 install_mysql_linux.sh
[ec2-user@ip-10-0-2-215 downloads]$ sudo ./install_mysql_linux.sh
mysql: [Warning] Using a password on the command line interface can be insecure.
[ec2-user@ip-10-0-2-215 downloads]$ cat install_mysql_linux.sh
mysql -u root -pPassword42! <<-EOF
CREATE USER 'mmuser'@'%' IDENTIFIED BY 'Mostest42!';
CREATE DATABASE mattermost_test;
GRANT ALL PRIVILEGES ON mattermost_test.* TO 'mmuser'@'%';
EOF
[ec2-user@ip-10-0-2-215 downloads]$ ^C
[ec2-user@ip-10-0-2-215 downloads]$ ^C
[ec2-user@ip-10-0-2-215 downloads]$ ls
```

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 <private IP of MySQL server>* Example : sudo *./install_mattermost_linux* 173.65.34.7 *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. <public IP of the application server>:8065
Expected screenshots	1) Executing the script 2) Starting the Mattermost server 3) Accessing the application via web browser

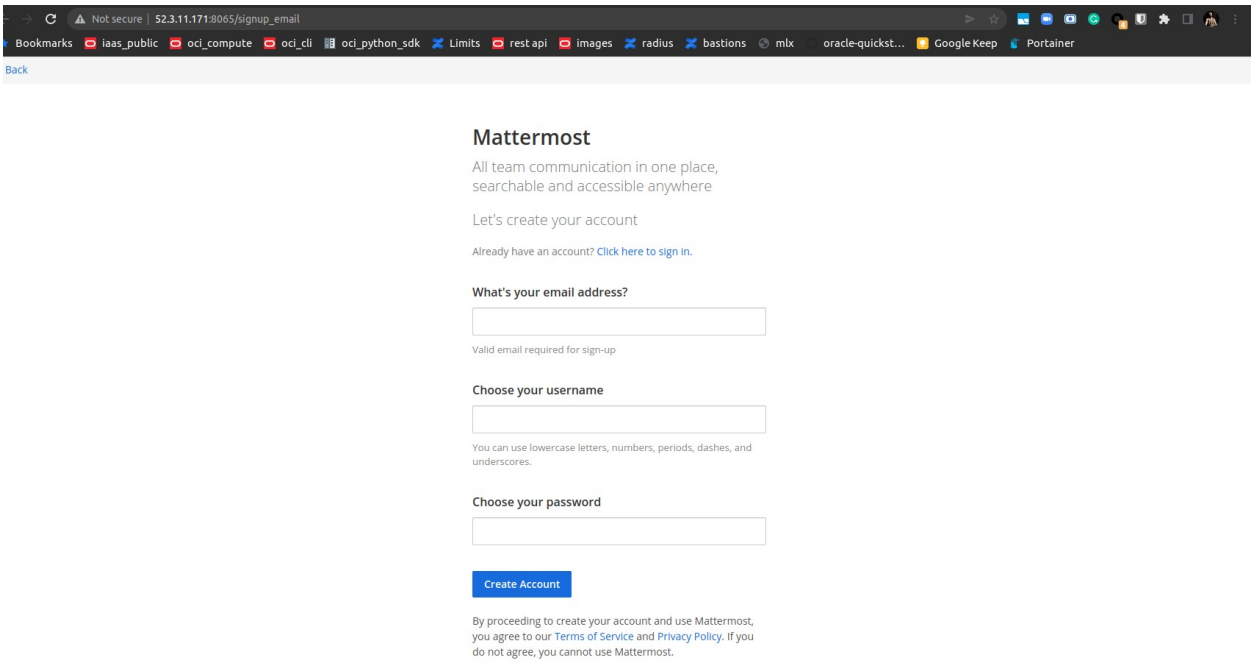
<Insert screenshot b(1) here> execute script

```
[ec2-user@ip-10-0-1-247 .ssh]$ mkdir download
[ec2-user@ip-10-0-1-247 .ssh]$ cd download/
[ec2-user@ip-10-0-1-247 download]$ wget https://d6opu47qoi4ee.cloudfront.net/install_mattermost_linux.sh
--2022-04-03 23:52:39--  https://d6opu47qoi4ee.cloudfront.net/install_mattermost_linux.sh
Resolving d6opu47qoi4ee.cloudfront.net (d6opu47qoi4ee.cloudfront.net)... 99.84.218.168, 99.84.218.37, 99.84.218.70, ...
Connecting to d6opu47qoi4ee.cloudfront.net (d6opu47qoi4ee.cloudfront.net)|99.84.218.168|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 592 [text/x-sh]
Saving to: ‘install_mattermost_linux.sh’

100%[==========================================================>] 592         --.-K/s   in 0s

2022-04-03 23:52:39 (77.9 MB/s) - ‘install_mattermost_linux.sh’ saved [592/592]

[ec2-user@ip-10-0-1-247 download]$ sudo yum -y install dos2unix -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core                                               | 3.7 kB  00:00:00
Resolving Dependencies
--> Running transaction check
---> Package dos2unix.x86_64 0:6.0.3-7.amzn2.0.3 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================================
 Package            Arch             Version                    Repository            Size
================================================================================================================
Installing:
 dos2unix           x86_64           6.0.3-7.amzn2.0.3          amzn2-core            75 k

Transaction Summary
================================================================================================================
Install  1 Package

Total download size: 75 k
Installed size: 190 k
Downloading packages:
dos2unix-6.0.3-7.amzn2.0.3.x86_64.rpm                    |  75 kB  00:00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Installing : dos2unix-6.0.3-7.amzn2.0.3.x86_64                                1/1
  Verifying  : dos2unix-6.0.3-7.amzn2.0.3.x86_64                                1/1

Installed:
  dos2unix.x86_64 0:6.0.3-7.amzn2.0.3

Complete!
[ec2-user@ip-10-0-1-247 download]$ sudo dos2unix install_mattermost_linux.sh
dos2unix: converting file install_mattermost_linux.sh to Unix format ...
[ec2-user@ip-10-0-1-247 download]$ chmod 700 install_mattermost_linux.sh
[ec2-user@ip-10-0-1-247 download]$ sudo ./install_mattermost_linux.sh 10.0.2.215
rm: cannot remove ‘/opt/mattermost’: No such file or directory
--2022-04-03 23:54:06--  https://releases.mattermost.com/5.19.0/mattermost-5.19.0-linux-amd64.tar.gz
Resolving releases.mattermost.com (releases.mattermost.com)... 13.32.207.77, 13.32.207.97, 13.32.207.65, ...
Connecting to releases.mattermost.com (releases.mattermost.com)|13.32.207.77|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 155314485 (148M) [application/x-gzip]
Saving to: ‘mattermost-5.19.0-linux-amd64.tar.gz’

100%[==========================================================>] 155,314,485 34.9MB/s   in 4.2s

2022-04-03 23:54:10 (35.5 MB/s) - ‘mattermost-5.19.0-linux-amd64.tar.gz’ saved [155314485/155314485]

Downloaded Mattermost
mattermost/
mattermost/client/
mattermost/client/18.11f0f217b22217f7cd67.js
mattermost/client/icon_16x16.png
```

<Insert screenshot b(2) here> - start mattermost server

ec2-user@ip-10-0-1-247 download]$ sudo chown -R mattermost:mattermost /opt/mattermost
[ec2-user@ip-10-0-1-247 download]$ sudo chmod -R g+w /opt/mattermost
[ec2-user@ip-10-0-1-247 download]$ cd /opt/mattermost
[ec2-user@ip-10-0-1-247 mattermost]$ sudo -u mattermost ./bin/mattermost
{"level":"info","ts":1649030086.0054836,"caller":"utils/i18n.go:83","msg":"Loaded system translations","for locale":"en","from locale":"/opt/mattermost/i18n/en.json"}
{"level":"info","ts":1649030086.0057468,"caller":"app/server_app_adapters.go:58","msg":"Server is initializing..."}
{"level":"info","ts":1649030086.0143862,"caller":"sqlstore/supplier.go:212","msg":"Pinging SQL","database":"master"}
{"level":"info","ts":1649030086.5916622,"caller":"sqlstore/upgrade.go:110","msg":"The database schema version has been set","version":"5.19.0"}
{"level":"error","ts":1649030088.25941,"caller":"app/server_app_adapters.go:125","msg":"SiteURL must be set. Some features will operate incorrectly if the SiteURL is not set. See documentation for details: http://about.mattermost.com/default-site-url"}
{"level":"info","ts":1649030088.262597,"caller":"app/license.go:39","msg":"License key from https://mattermost.com required to unlock enterprise features."}
{"level":"info","ts":1649030088.2635503,"caller":"app/migrations.go:26","msg":"Migrating roles to database."}
{"level":"info","ts":1649030088.3410423,"caller":"sqlstore/post_store.go:1351","msg":"Post.Message has size restrictions","max_characters":16383,"max_bytes":65535}
{"level":"info","ts":1649030088.345663,"caller":"app/migrations.go:102","msg":"Migrating emojis config t

<Insert screenshot b(3) here> - access app via web browser



Step 5: Answer the following questions

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 b)

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 a)

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 c

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 b

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 | c

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 those ports is entirely unnecessary for the given architecture to function. Which of the ports given in the option below could it be?

a) Port 22 on the NAT instances

b) Port 3306 on the database server

c) Port 443 on the NAT instance

d) Port 22 on the application server

Enter your answer here | A

Q7 Describe the steps you would take to increase security of the servers you have deployed so that they are not reachable from external sources

Option 1 Add a WAF in front the application server and use ACLS or country zone filtering

Option 2 Uses AWS Cognito to authenticate connection pools of allowed users

Q8 Describe the steps required to deploy the given application in an autoscaling environment

AS – can only apply the application server ; for the db it is more involved as we need to sync the data. (as the question was open-ended – i went with application server)

Use this blueprint
https://docs.aws.amazon.com/autoscaling/ec2/userguide/GettingStartedTutorial.html

Step 1. Make a snapshot of the current app server
Step 2. Create a Lauch template from Step 1 snapshot
Step 3 Create a single isntance ASG (Auto scaling group)
Step 4. Scale size of ASG via CPU or memory – go with horizontal scaling i.e add more hosts of the same type

Max marks **15**

Grades distribution	
MCQs	6 (1 mark each)
Subjective questions	10 marks (5+5)
Implementation screenshots	24 marks (1 marks each)
Total	40 marks