**VPC – Virtual Private Cloud Assignment**

**Documentation**

1……………… Introduction

2……………… The assignment: VPC – Virtual Private Cloud

3……………….Our solution- Step By Step guide

3.1………Create VPC

3.2……… Add subnets

3.3………. Add servers

3.4………... Create Security groups

3.5………. Verify the connection with the MySQL server

4…………………Bonus step

1. Introduction

**Creating a Secure VPC with MySQL Server and SSH Access: A Step-by-Step Guide + Bonus Automation Tasks.**

The task requires the creation of a VPC with multiple subnets, routing tables, and gateways. Additionally, two EC2 instances will be launched, one with a MySQL server installed and another with only SSH public access. Relevant security groups must also be created to secure the infrastructure.

The task also involves verifying that the MySQL server is accessible by the SQL client and can "talk" with the server both locally and remotely.

\* Bonus step: Automate the MySQL service installation on a public ec2.

**Overall**, this task aims to demonstrate an understanding of how to set up a secure and functional VPC with multiple subnets, gateways, and EC2 instances, as well as proficiency in configuring a MySQL server and using SQL clients to interact with it.

2) The assignment:

**VPC – Virtual Private Cloud Assignment**

On this assignment, you will have to provision and define VPC with some servers.

Use the Canvas sandbox environment to do so.

You should take screenshots of every resource and relevant definition steps. Order them in a doc or slide deck, in a coherent manner, with needed explanations in order to present your project.

1. Create VPC (without wizards – VPC only)

2. Add 4 subnets into it, in 2 different availability zones (2 in each AZ)

a. 2 private

b. 2 public

c. 2 routing tables, 1 public and 1 private, and assigned to respective subnets.

d. Create an internet gateway and attach it to a public subnet.

e. create NAT gateway attach it to private subnet.

3. On top of this infrastructure, you need to add two servers (pay attention to which subnet each one should be provisioned):

a. EC2 (Amazon Linux 2) with only SSH public access

• SQL client should be installed in addition to other relevant services

b. EC2 (Amazon Linux 2) with MySQL server not accessible publicly (check what are the

relevant requirements for running MySQL with minimal data)

4. Create relevant Security groups

5. Verify that after connecting to ec2 using ssh, and operating the SQL client, we can “talk” with the MySQL server. (connect and see connection succeeded message ) Steps in problem decomposition

1. Review how to create the infrastructure

a. VPC

b. Subnets

c. Route table

d. Internet Gateway

e. NAT

f. EC2

g. Security groups

2. Review how to install MySQL on

3. Learn how to configure security issues in MySQL server (users, passwords, local access, remote access)

4. Learn how to install SQL client (currently CLI one)

5. Learn how a client “talks” with the server locally (both on the same machine)

6. Learn how a client “talks” with the server remotely (pay attention to network, subnet, ports and other issues)

**Bonus steps**

Automate the MySQL service installation on each ec2

3) Our solution

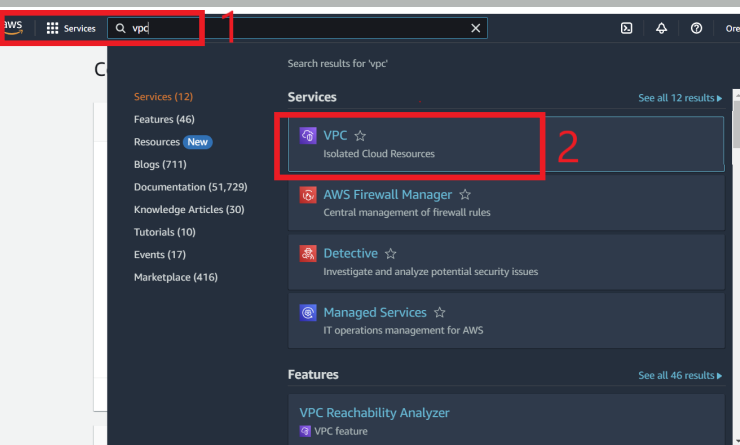
**Step-by-step guide**

1. Create VPC (without wizards – VPC only)

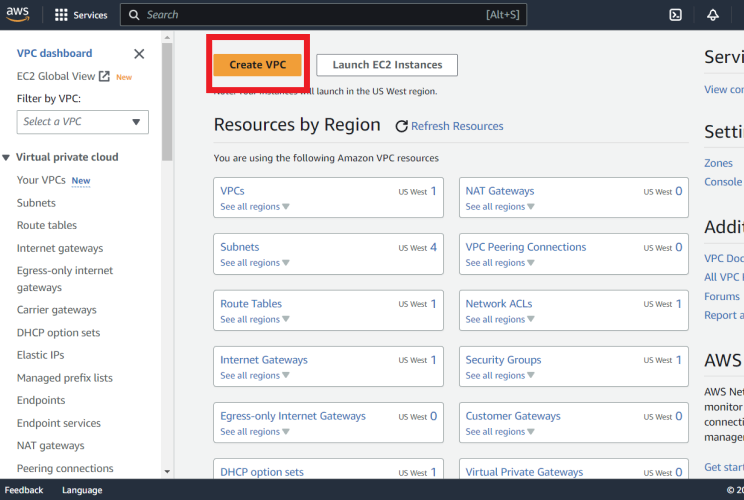
* Navigate to the VPC dashboard in the [AWS Management Console](https://labs.vocareum.com/main/main.php?m=clabide&mode=s&asnid=1404844&stepid=1404845&hideNavBar=1)
* click on the “Start Lab” button on the upper bar
* Wait till the red light next to AWS turns green, then click on it

(This will open the AWS Management Console in a new browser tab. The system will automatically log you in)

* Navigate to the VPC dashboard.



* Click on "Create VPC"



                Next step: **VPC settings**

* Choose VPC only
* Creates a tag with a key of 'Name' and a value that you specify
* Choose “no IPv6 CIDR block”



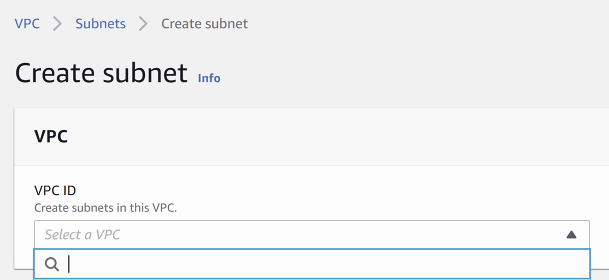
* select the IPv4 CIDR block  (for example, 10.0.0.0/16) for your VPC.
* Scroll down and click on “Create VPC” button.

* you successfully created VPC.



1. Add 4 subnets into it, in 2 different availability zones (2 in each AZ):

* At the sidebar, choose “Subnets”.
* Choose the “Create subnet” button.
* From the dropdown menu, choose “VPC ID” – the VPC we created.



* On **Subnet** **settings**:

Create 4 subnets in 2 different availability zones (2 in each AZ), 2 private, 2 public.

Example of creating a subnet:

* Create a tag with a key of 'Name' and a value that you specify.
* Choose availability zone
* Enter an IPv4 CIDR block for your subnet. We need 4 subnets.

\*When you create a subnet, you must specify the VPC that it belongs to and the IP address range for the subnet. In this case, you will create 4 subnets within the VPC you create.

**For example**, if you have chosen the VPC CIDR range as 10.0.0.0/16, you could create the following subnets:

AZ1-Public-Subnet: 10.0.1.0/24

AZ1-Private-Subnet: 10.0.2.0/24

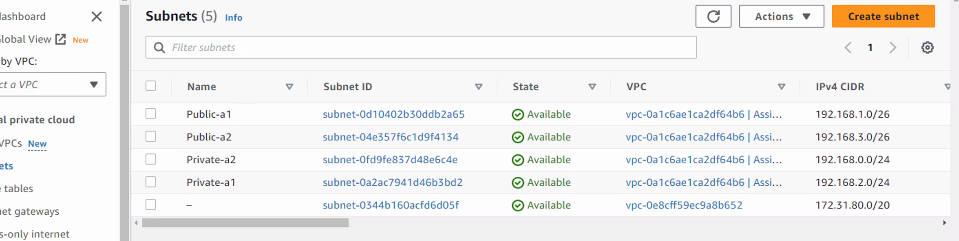
AZ2-Public-Subnet: 10.0.3.0/24

AZ2-Private-Subnet: 10.0.4.0/24ed in step 1, with 2 subnets in each of 2 different availability zones (AZs).

In this **example**, the first two subnets (AZ1-Public-Subnet and AZ1-Private-Subnet) would be created in Availability Zone 1, while the next two subnets (AZ2-Public-Subnet and AZ2-Private-Subnet) would be created in Availability Zone 2.

Make sure to choose appropriate IP address ranges for each subnet and select the correct AZ for each subnet during the creation process.

* press on the “Create” button.
* you have successfully created 4 subnets.



Create an internet gateway and attach it to a **public** subnet:

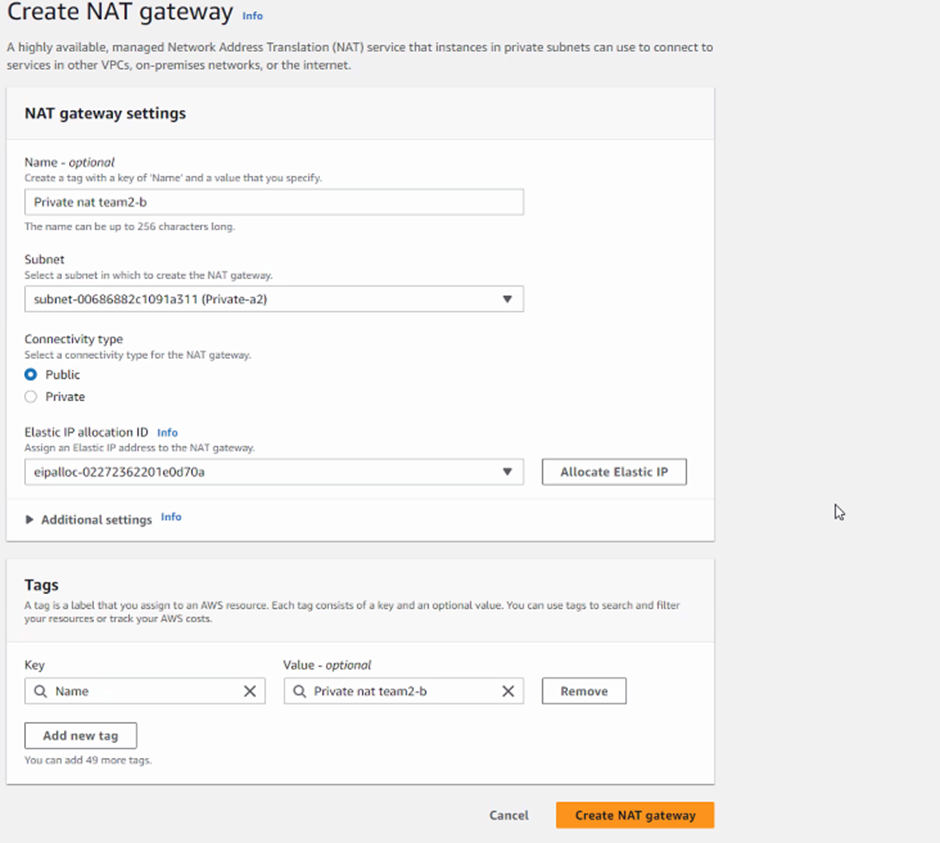
* Choose “Internet gateway” from the sidebar.
* Click on the “Create internet gateway**”** button.
* You created an internet gateway successfully.
* Press on the “attach to a VPC” button.
* Attach an internet gateway to a VPC, to enable the VPC to communicate with the internet. Specify the VPC to attach.
* click on the “Attach internet gateway” button.
* Internet gateway successfully attached to VPC

create NAT gateway and attach it to **private** subnet:

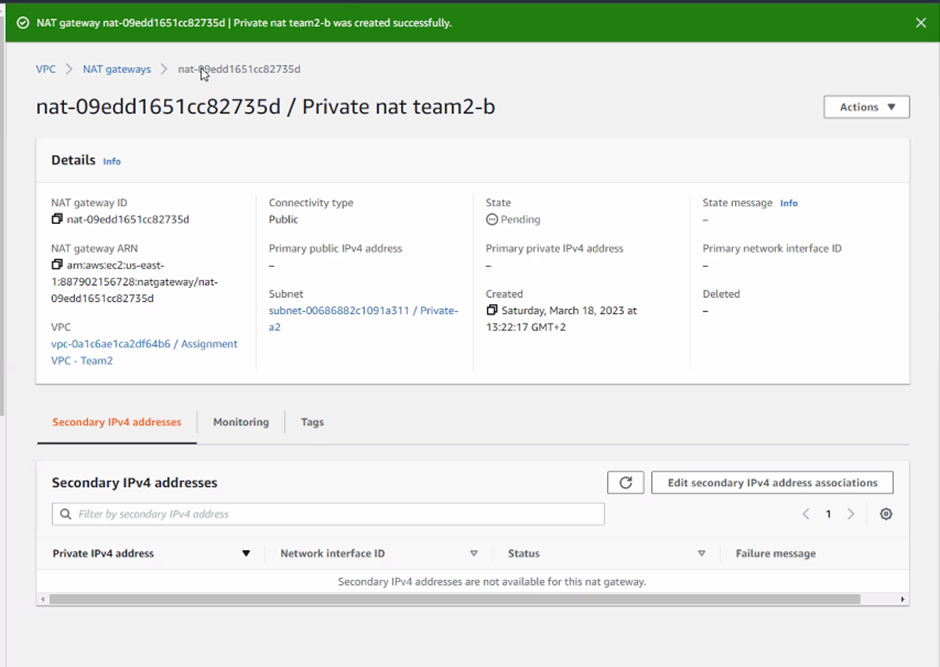
Select the subnet in which to create the NAT gateway

(Select the private subnet).

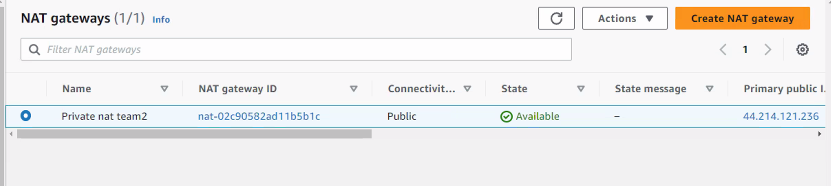
* Assign an Elastic IP address to the NAT gateway, and click on the “Allocate Elastic IP” button.
* Click on the “Create NAT gateway” button.



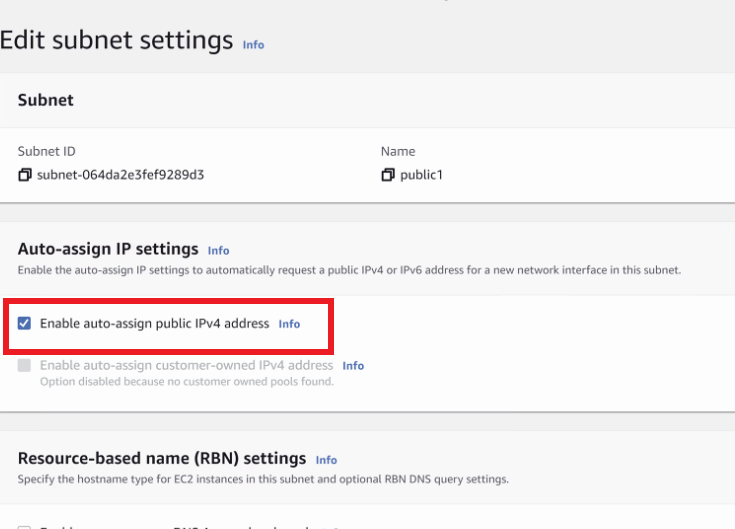
* NAT gateway was created successfully:

****

* Go to “Action”, Edit subnet settings:

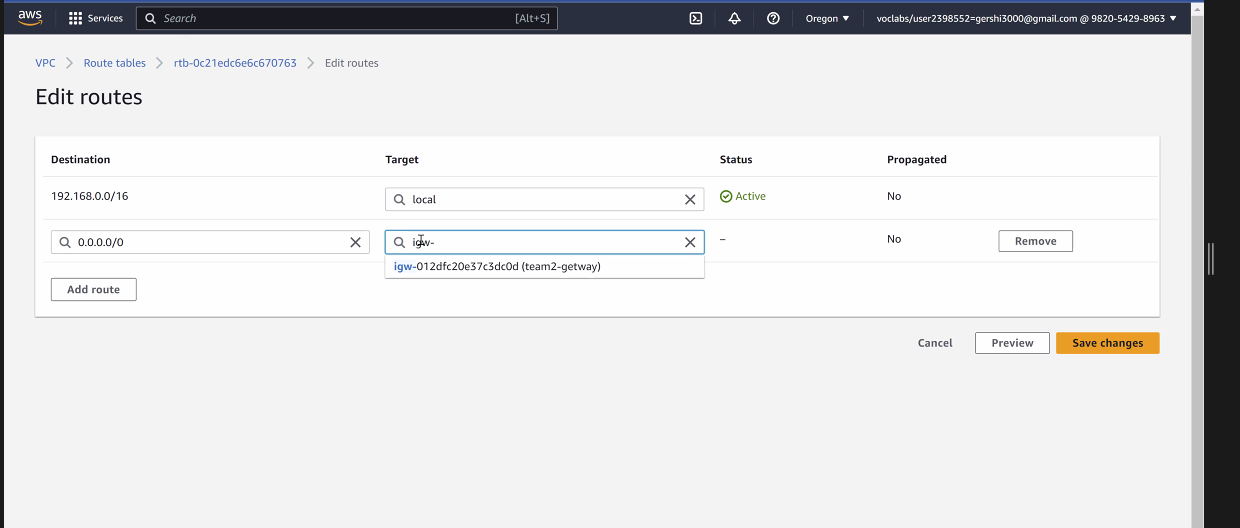
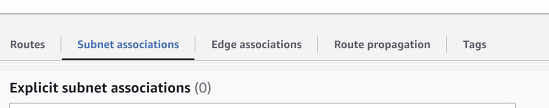


**Enable** the auto-assign IP settings to automatically request a **public** IPv4 or IPv6 address for a new network interface in this subnet:



* Click on the “Save” button.

Create: 2 routing tables, 1 public and 1 private, and assign to respective subnets:

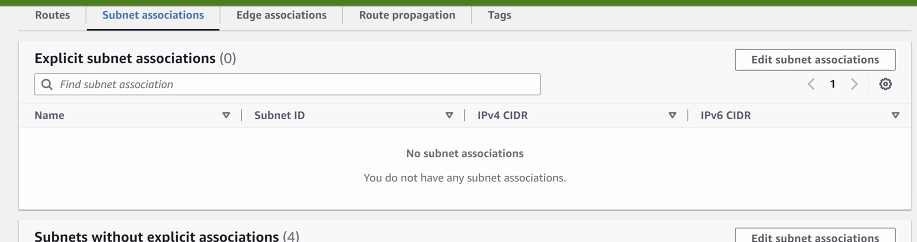
* On the sidebar, choose “Route Table”
* click on “Create route table”.
* Choose the “Route” tag
* click on “Edit routes”
* Add routes:
* Under “Target”, Start typing “igw” and choose  the gateway:
* Click on “Save changes.”
* Updated routes successfully
* On “Subnet associations” tag, you can see all the subnets that haven’t been explicitly associated with any route tables and are therefore associated with the public and private route table respectively:

Now, we will create 2 routing tables:

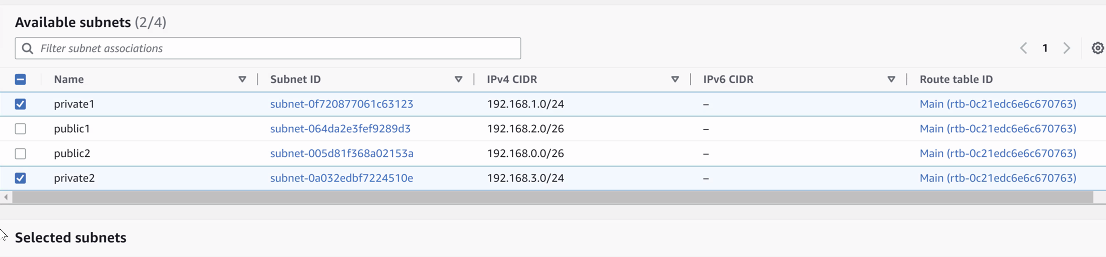
* On the sidebar, choose “route table”
* Press on the “Create route table” button on the upper right corner.
* Create a tag with a key of 'Name' and a value that you specify, and choose the VPC to use for this route table
* Click on the “Create route table” button.
* Route table was created successfully.



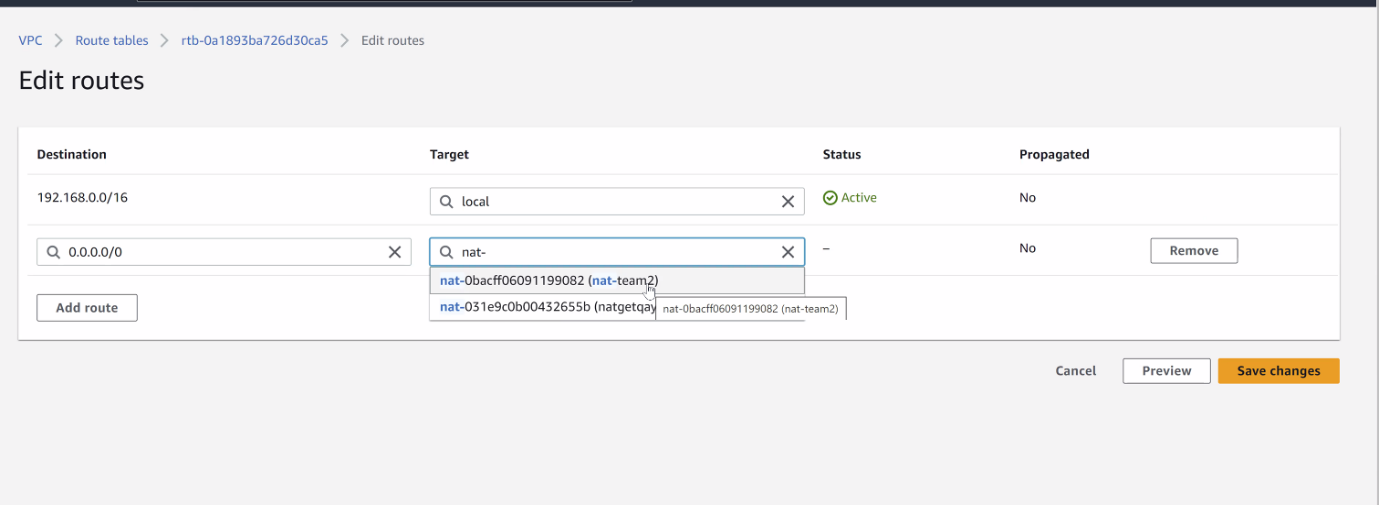
* Edit subnet associations:



* Choose the private subnet that you have created:



* Click on the “Save” button.”
* You have successfully updated subnet association for rtb /**private** -RT
* On the upper right corner, press “route table”
* click on “Create a route table”
* Click on “Edit routes” button



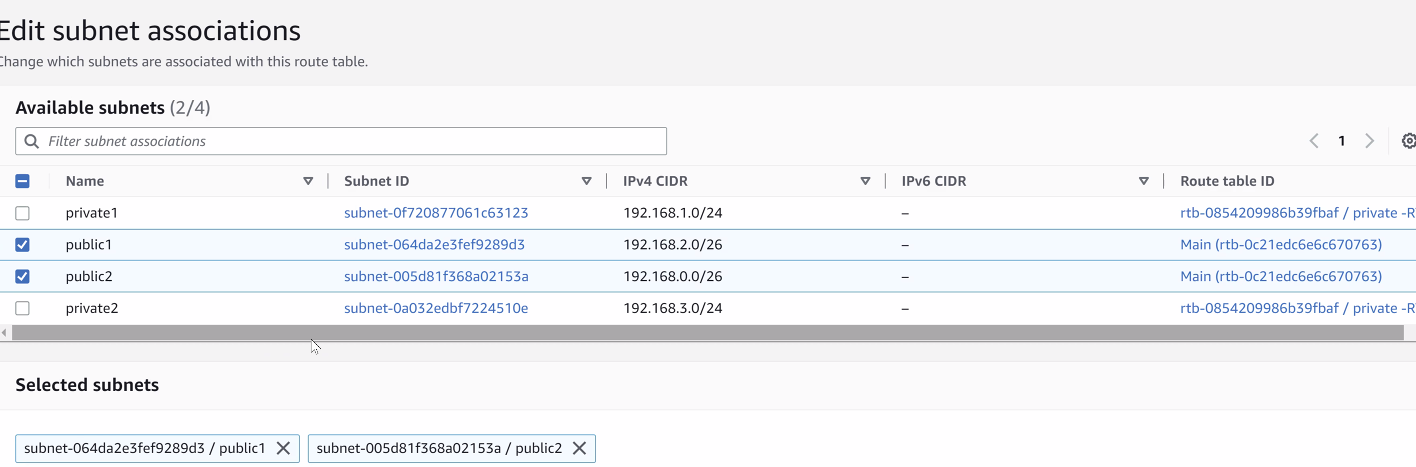
* click on the “Add route” button.

Enter:

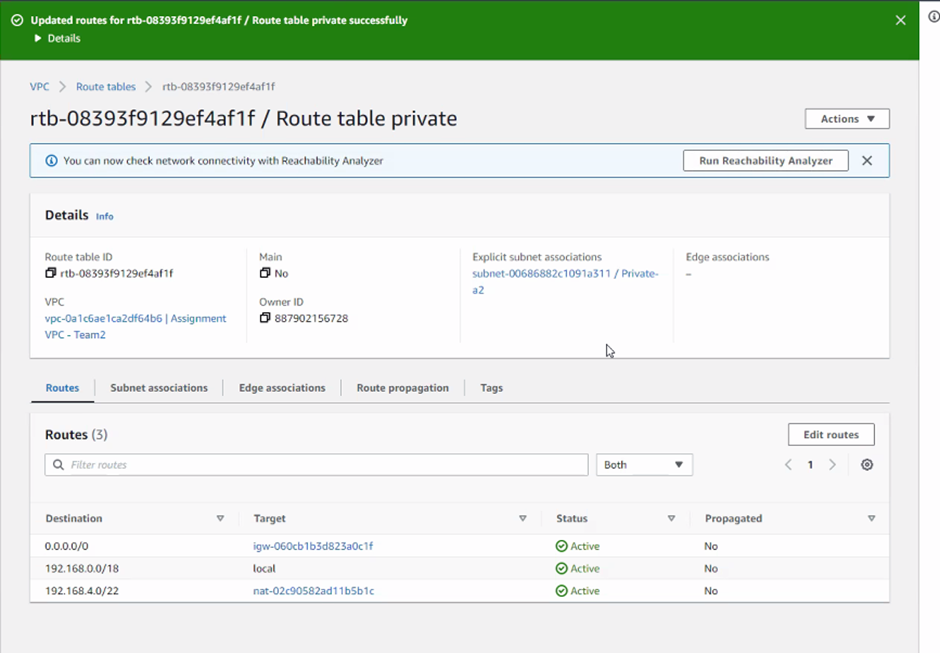
Destination: 0.0.0.0/0

Target: Select the Internet Gateway you created

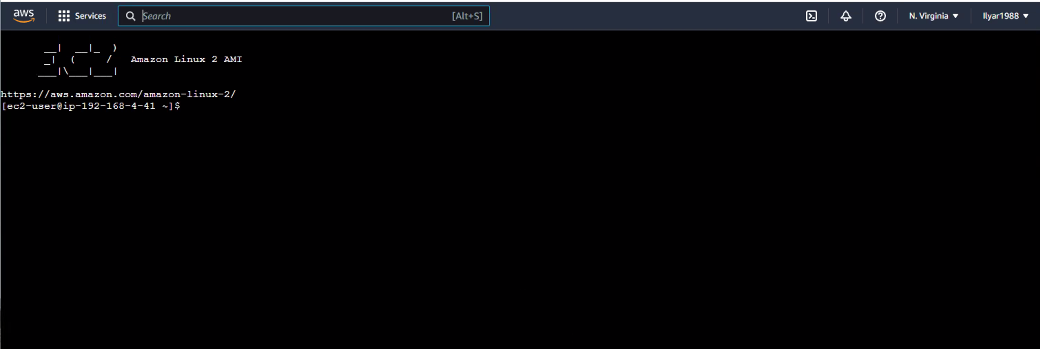
* Click on the “Save route” button.
* click on “Edit subnet associations”
* Choose the private subnet that you have created:



* Click on the “Save” button.
* You have successfully updated subnet associations for ….. / **public**.
* Create route table (for the ec2 **private**):



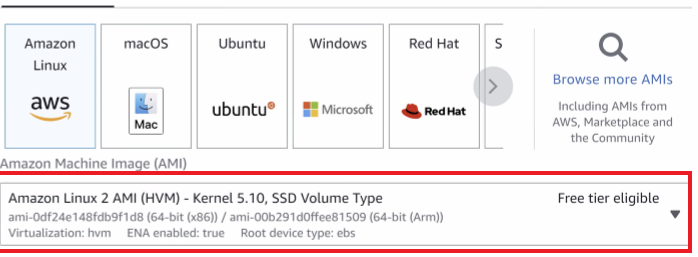
successfully connected (private):

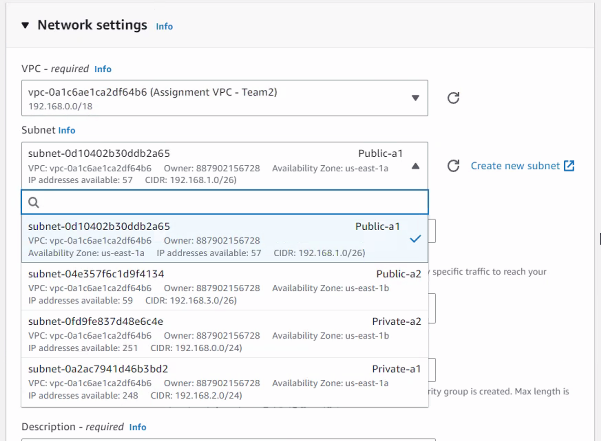


1. Add two servers

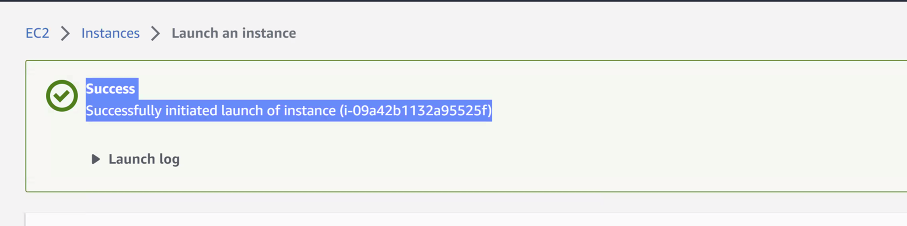
A. **EC2 (Amazon Linux 2) with only SSH public access**

* On the AWS console, choose “EC2”
* Click on the "Launch Instance" button.
* Choose the Amazon Linux 2 AMI:



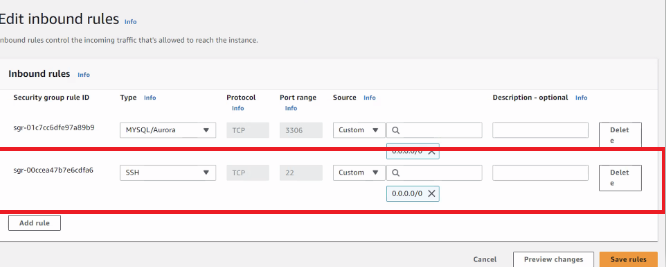
* Create new key pair
* in “Network settings,” select the VPC you created.
* Choose subnets: (**public**)

* In the "Auto-assign Public IP" dropdown menu, select "Enable"
* click on the "Launch instance" button.
* you successfully initiated the launch of instance (......)

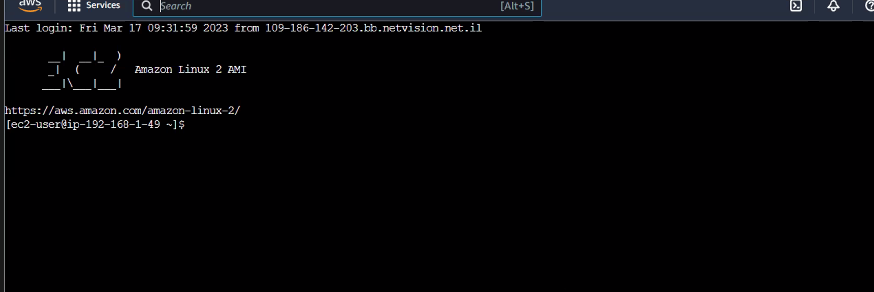


1. Create Security groups

* On the EC2 Dashboard Select "Security Groups" from the left-hand menu.
* Click "Create Security Group" and enter a name and description for the security group.
* Create a new security group with inbound rules that only allow SSH traffic from a specific IP address or range:

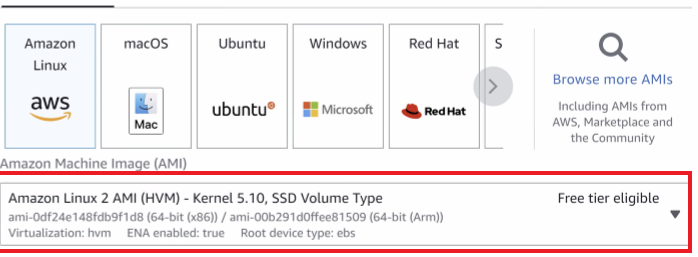


successfully connected:

****

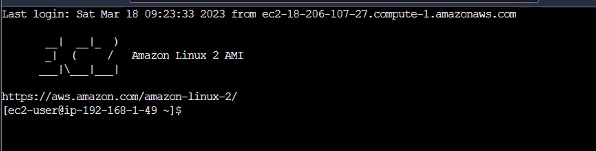
B. **EC2 (Amazon Linux 2) with MySQL server not accessible publicly**

* On the AWS console, choose “EC2”
* Click on the "Launch Instance" button.
* Choose the Amazon Linux 2 AMI:



* Create new key pair
* in “Network settings”,  select the VPC you created.
* Choose subnets: (**private**)
* Go to “instance”
* Click on the “connect” button

successfully connected:



* Execute the following command to enable MySQL yum repository:



* Since the MySQL yum repository has multiple repositories versions, you need to disable all repositories in mysql repo file:



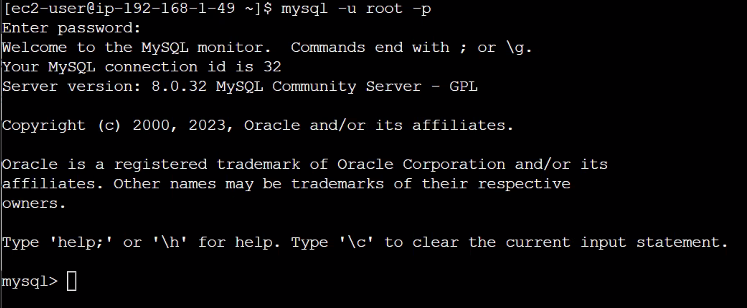
* Install MySql server:

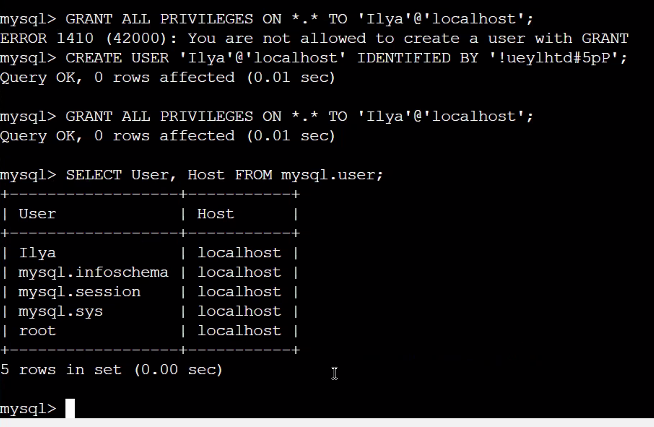


* Start mysql service:
* To show the password of the root user account, you use this command:

grep "A temporary password" /var/log/mysqld.log

* Execute mysql\_secure\_installation to secure MySQL server.
* Restart the mysql service.
* Use this command to connect to MySQL server:



* Add user with permissions to connect to MySql server remotely.

1. Verify the connection with MySQL server

* Install MySql client

yum install mysql

* Check the installed version.

mysql --version

* change the encryption of the password in public ec2.

ALTER USER 'Ilya' IDENTIFIED WITH mysql\_native\_password BY 'password';

* Connect to private ec2
* connect to the MySQL server with the username 'Ilya' and the password that was set in the previous step

mysql -u Ilya -p -h 52.90.151.106

4. Bonus step

Automate the MySQL service installation on a public ec2

* Create new instance.

Graphical user interface, text, application, email

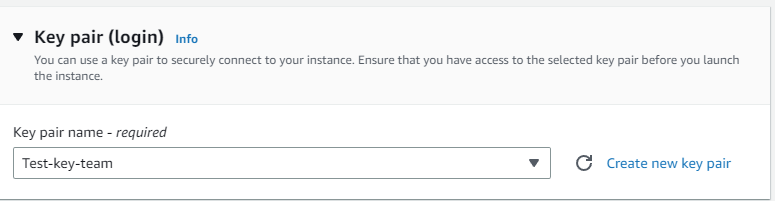
Description automatically generated

* Select Amazon Linux 2 AMI

Graphical user interface, text, email

Description automatically generated

* Select exciting key pair.

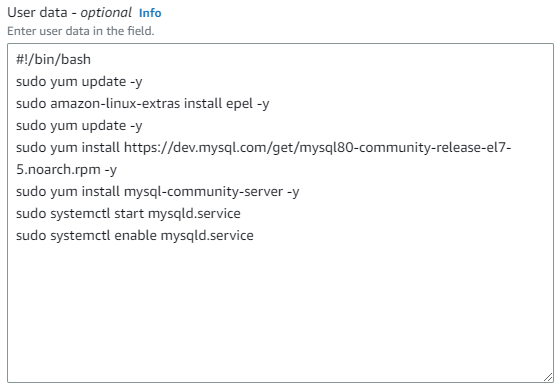


* Select Assignment VPC - Team2.
* Select public subnet.
* existing security group and high availability zone us-east-1b.

Graphical user interface, text, application, email

Description automatically generated

* Click on Advanced details option and enter the following script in the User data field:

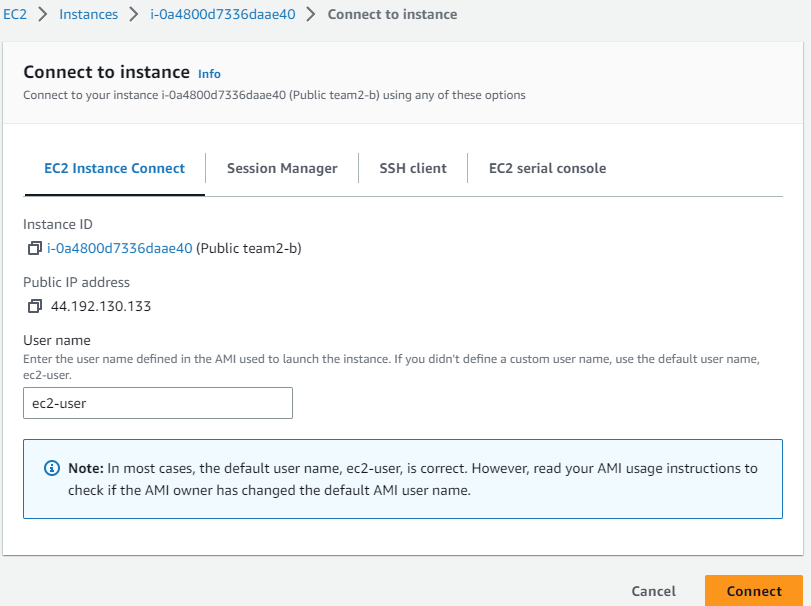


* Click on Launch instance button to create the instance.

A picture containing icon

Description automatically generated

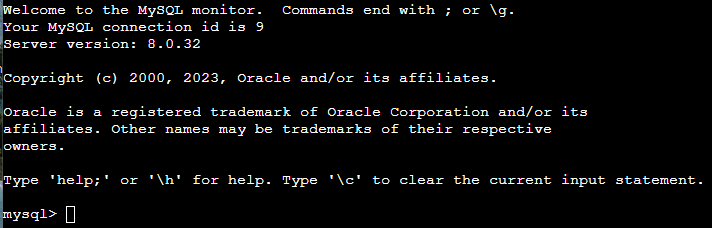
Connect to ec2 from AWS console.



Enter the following command in the ec2 instance and connect to SQL server.



The connection to the MySQL was successful as expected.



***Providing credit to:***

***Roni gershon.***

***Ilya rahmilevich***