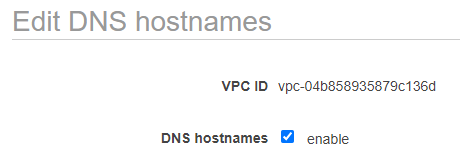
1. Select region N.Virginia
2. Create VPC

VPC name - **VPC-EC2-to-DB**

IPv4 CIDR – **10.0.0.0/16**

Once VCP is created, select the newly created VCP and make sure that from Actions button -> Edit DNS hostnames -> enable is checked



1. Create Subnets

Subnet name - **Subnet-EC2-Public**

IPv4 CIDR – **10.0.1.0/24**

Availability Zone **- us-east-1a**

Subnet name - **Subnet-DB-Private1**

IPv4 CIDR – **10.0.2.0/24**

Availability Zone **- us-east-1b**

Subnet name - **Subnet-DB-Private2**

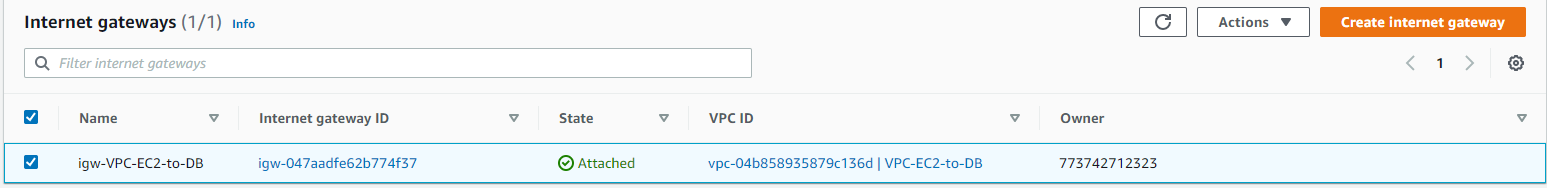
IPv4 CIDR – **10.0.3.0/24**

Availability Zone **- us-east-1c**

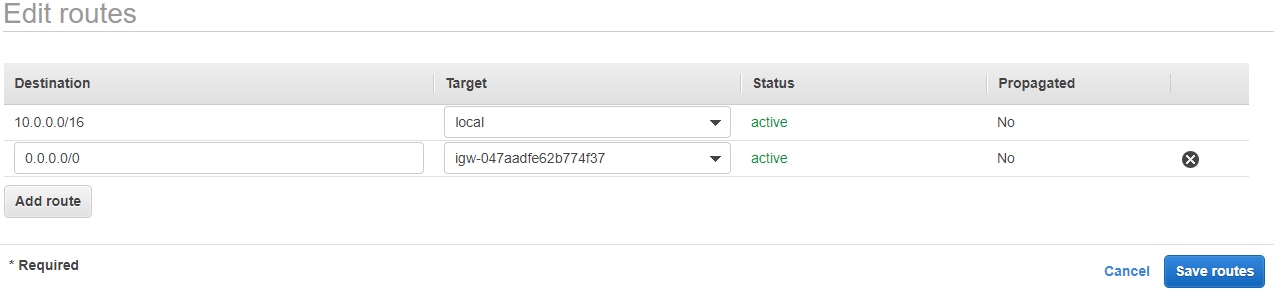
1. Cerate Internet Gateway

Name - **igw-VPC-EC2-to-DB**

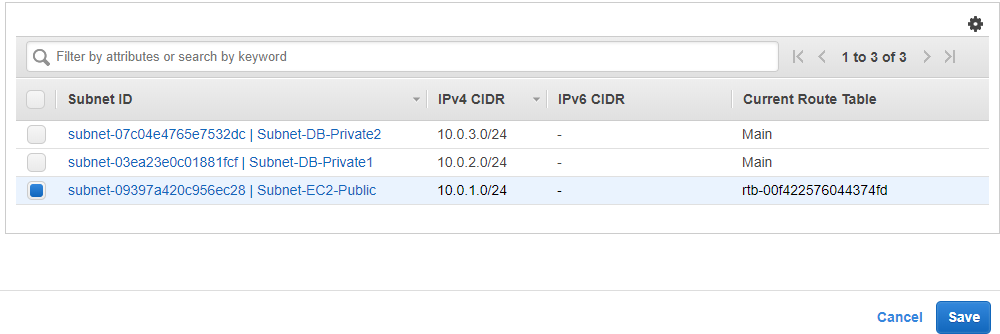
1. Attach created Gateway to **VPC-EC2-to-DB** VPC



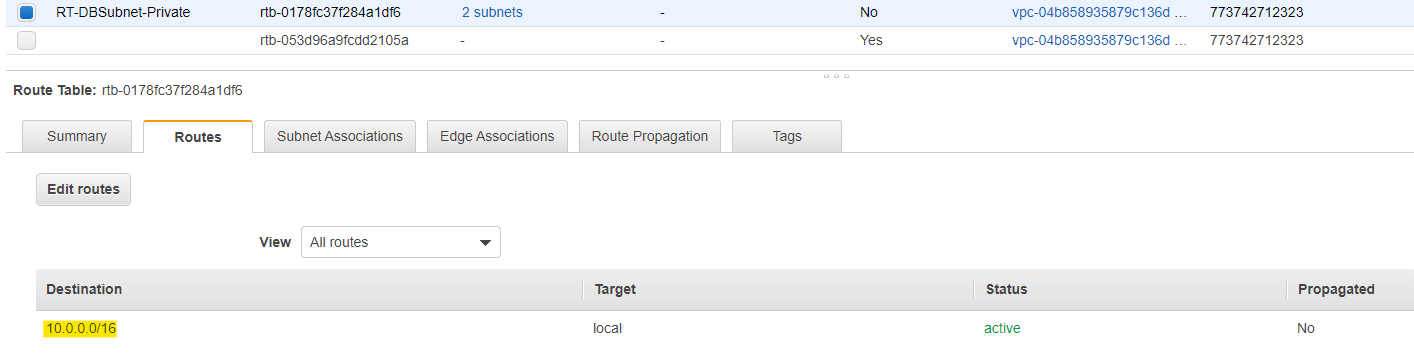
1. Create Route Table for Public Subnet
   1. Name - **RT-EC2Subnet-Public**
   2. From Tab Routes –> Edit routes –> Add route -> Destination **0.0.0.0/0** -> Target from dropdown menu select **Internet Gateway** -> Select newly created Gateway **igw-VPC-EC2-to-DB** -> Save routes



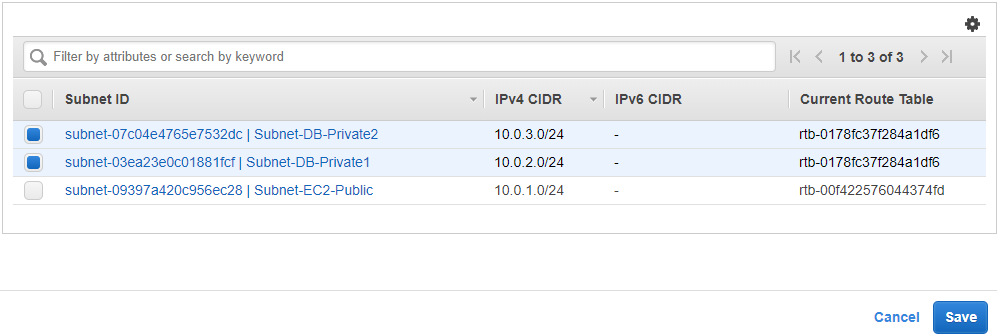
* 1. From Tab Subnet Associations -> Edit Subnet associations - > Select **Subnet-EC2-Public** subnet



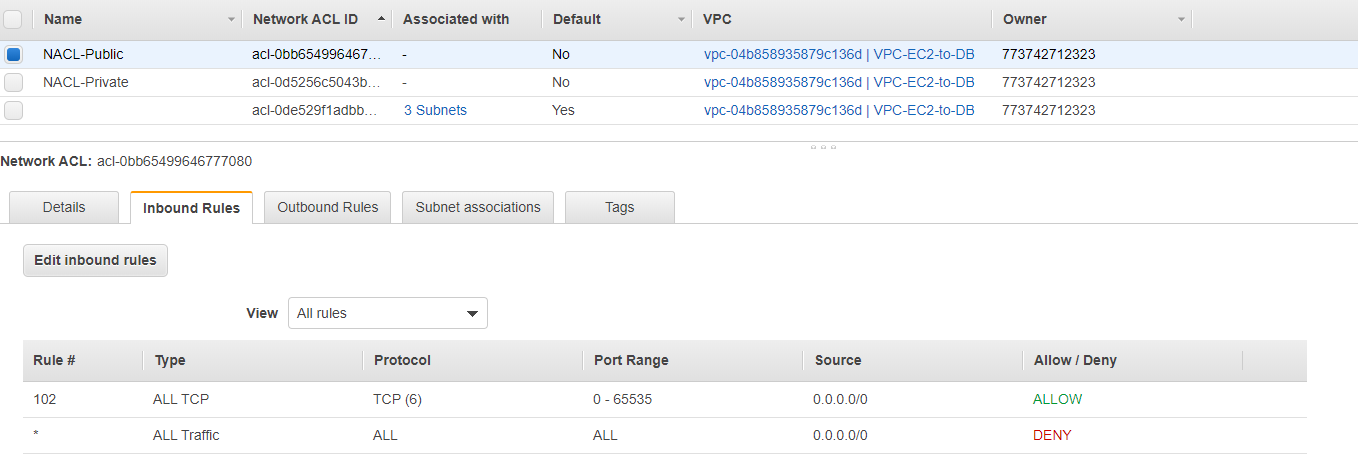
1. Create Route Table for Private Subnets
   1. Name - **RT-DBSubnet-Private**
   2. Tab Routes – should contain only VPC CIDR range



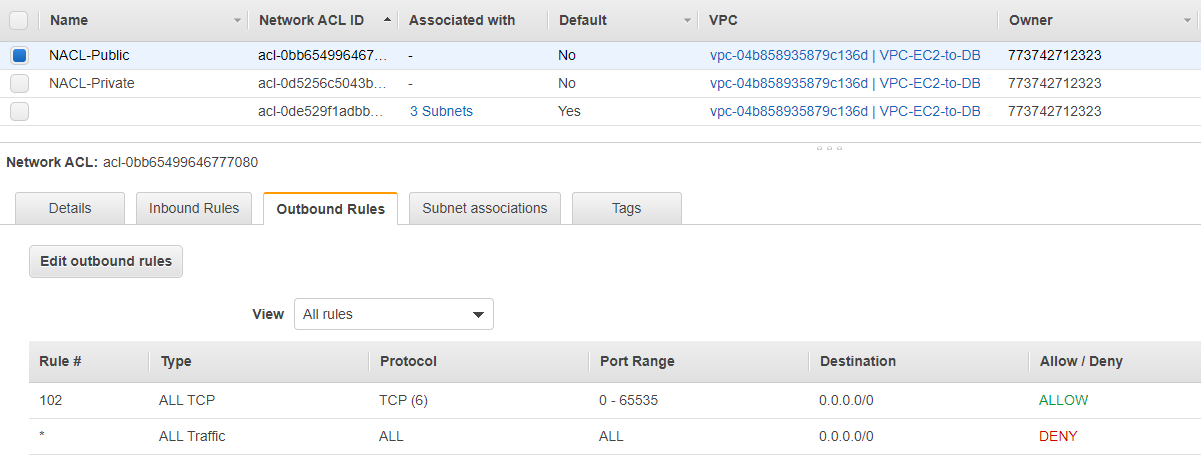
* 1. From Tab Subnet Associations -> Edit Subnet associations - > Select **Subnet-DB-Private1** and **Subnet-DB-Private2** subnets - > Save



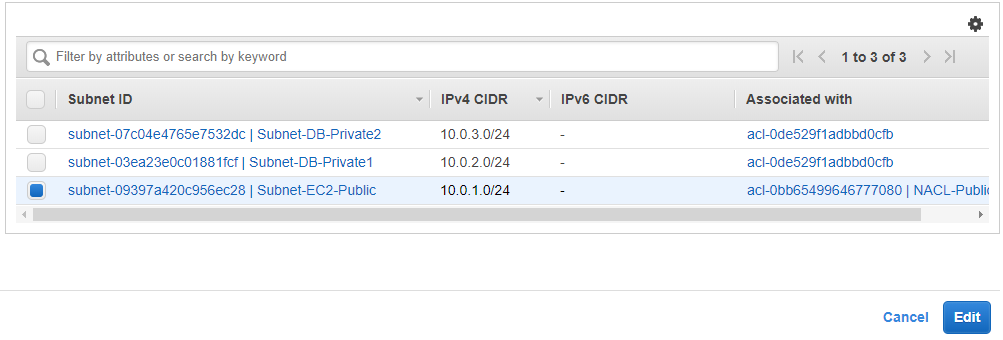
1. Create NACLs
   1. From menu Network ACLs -> Create network ACL -> Name **NACL-Public** -> Select from dropdown VPC **VPC-EC2-to-DB**
   2. From menu Network ACLs -> Create network ACL -> Name **NACL-Private** -> Select from dropdown VPC **VPC-EC2-to-DB**
2. Configure NACLs
   1. Select **NACL-Public** -> Inbound Rules tab -> Edit inbound rules -> Add rule -> Rule# 102 -> Port Range 0-65535 -> Save



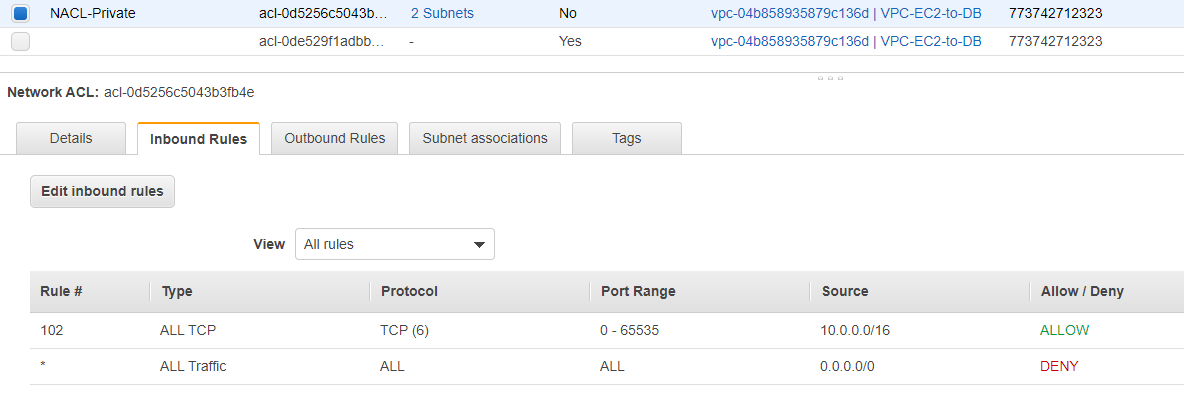
* 1. Select **NACL-Public** -> Outbound Rules tab -> Edit outbound rules -> Add rule -> Rule# 102 -> Port Range 0-65535 -> Save



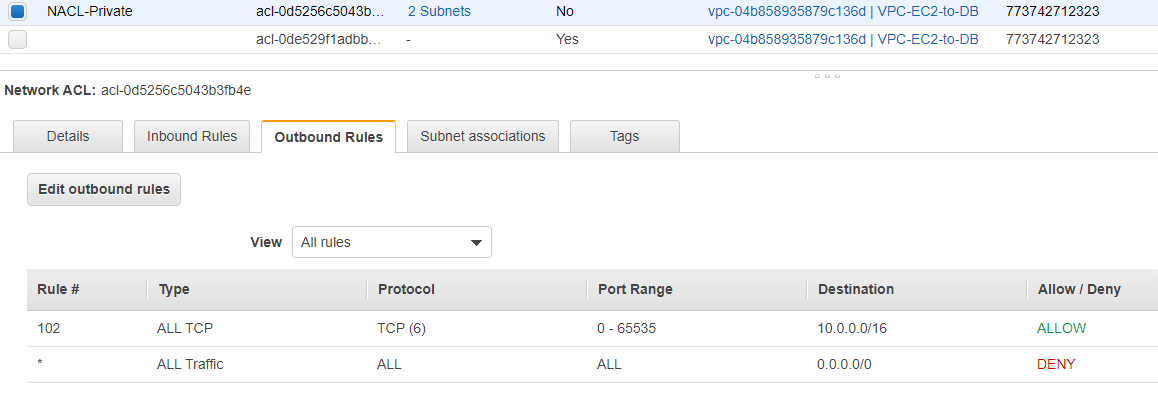
* 1. Select **NACL-Public** -> Subnet Associations tab -> Edit subnet association -> Select **Subnet-EC2-Public** -> Edit



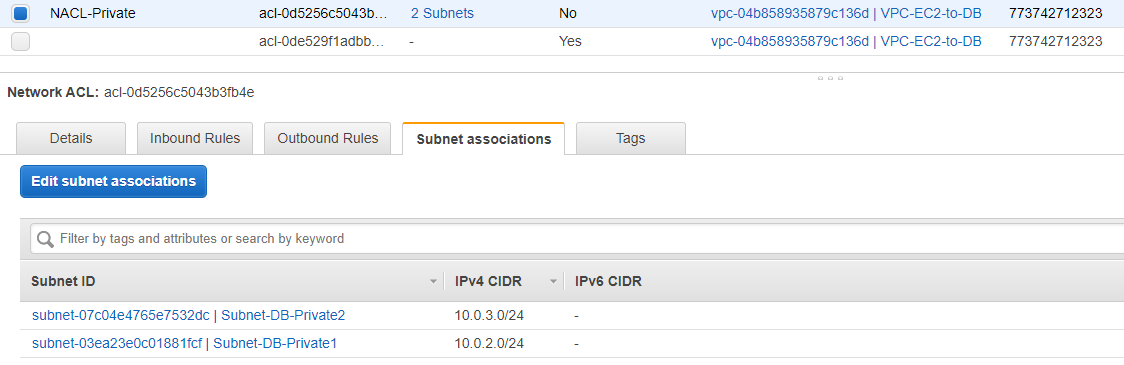
* 1. Select **NACL-Private** -> Inbound Rules tab -> Edit inbound rules -> Add rule -> Rule# 102 -> Port Range 0-65535 -> Source 10.0.0.0/16 -> Save



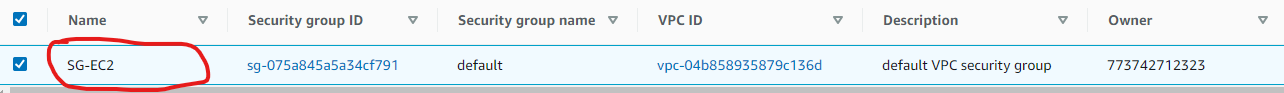
* 1. Select **NACL-Private** -> Outbound Rules tab -> Edit outbound rules -> Add rule -> Rule# 102 -> Port Range 0-65535 -> Source 10.0.0.0/16 -> Save



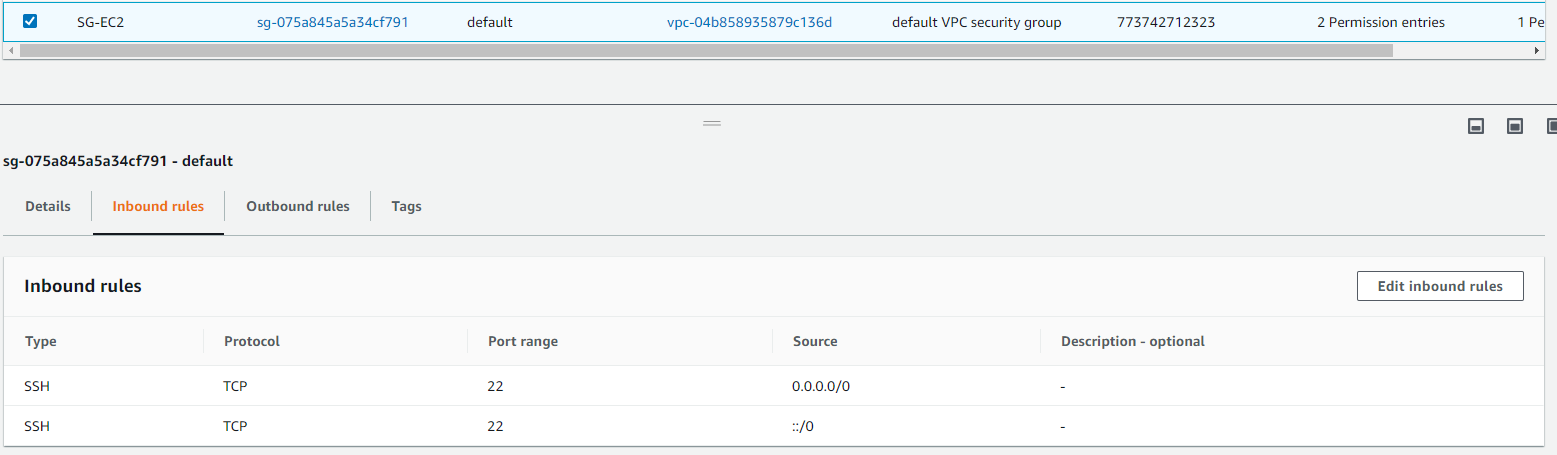
* 1. Select **NACL-Private** -> Subnet Associations tab -> Edit subnet association -> Select **Subnet-DB-Private1** and **Subnet-DB-Private2** -> Edit



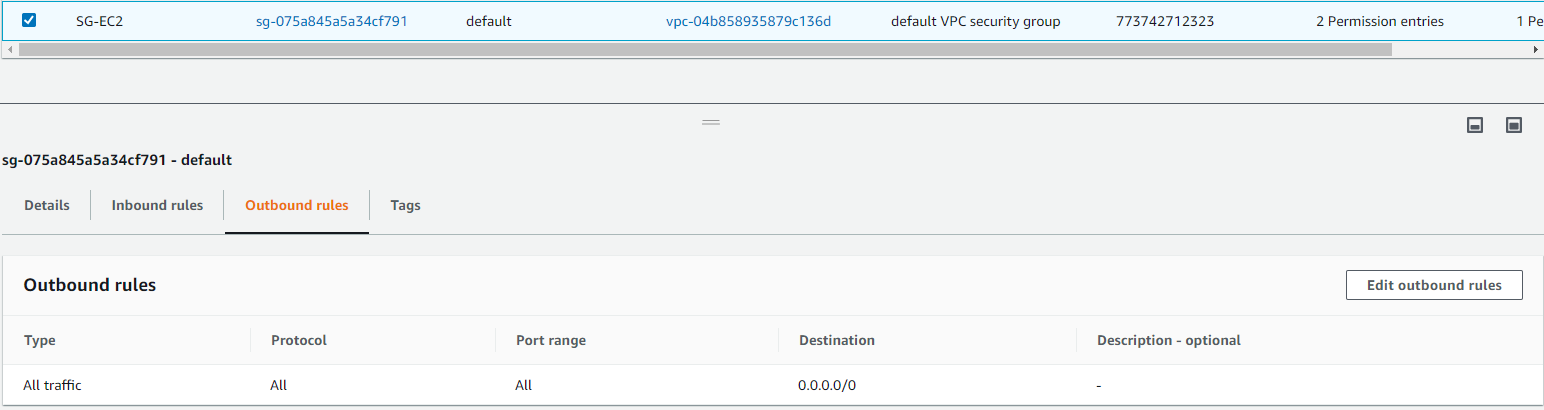
1. Create and configure Security Groups
   1. Go to menu Security Groups
   2. Should have default security group for VPC **VPC-EC2-to-DB.** For easier reference we could change the name of the security group to **SG-EC2**.



* 1. Go to Inbound rules tab -> Edit inbound rule -> select Type SSH -> Source Anywhere

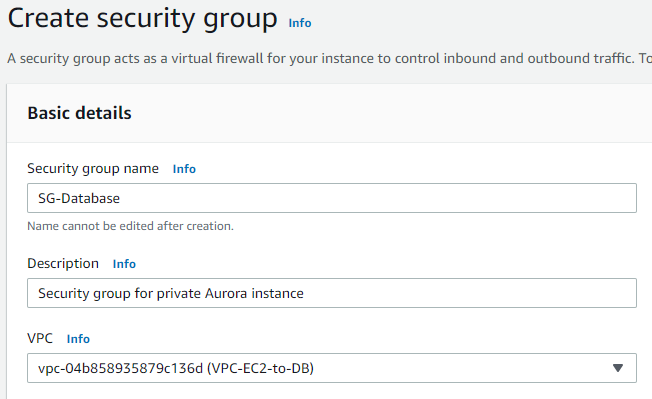


* 1. Under tab Outbound rules we should have only All Traffic

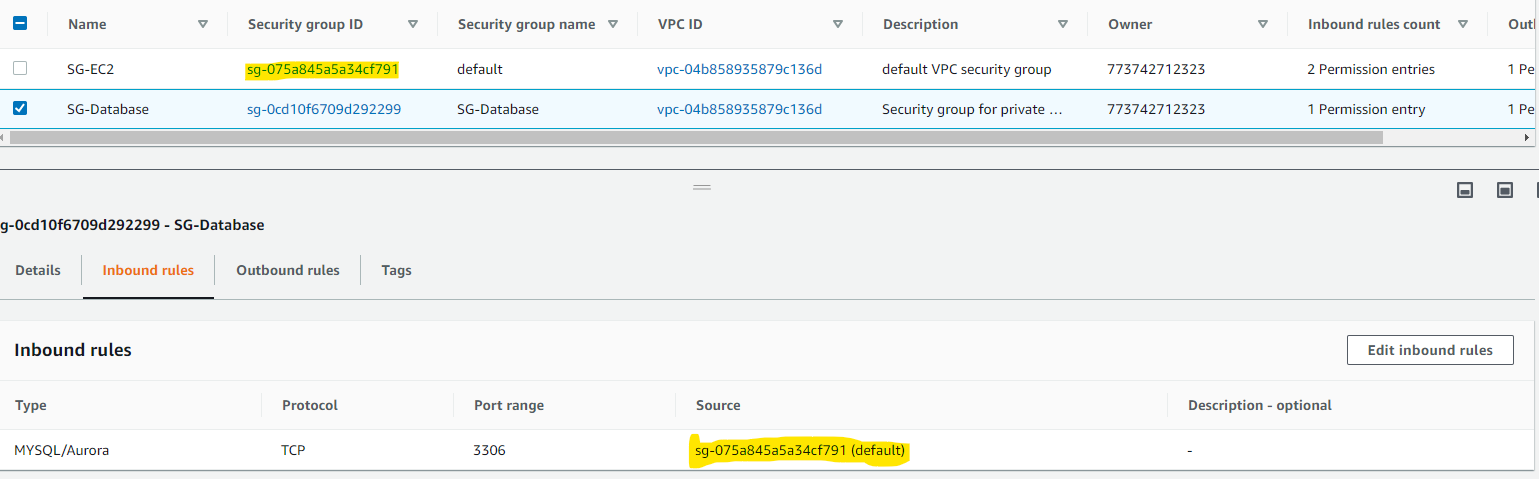


* 1. Create Security group for private Database source.

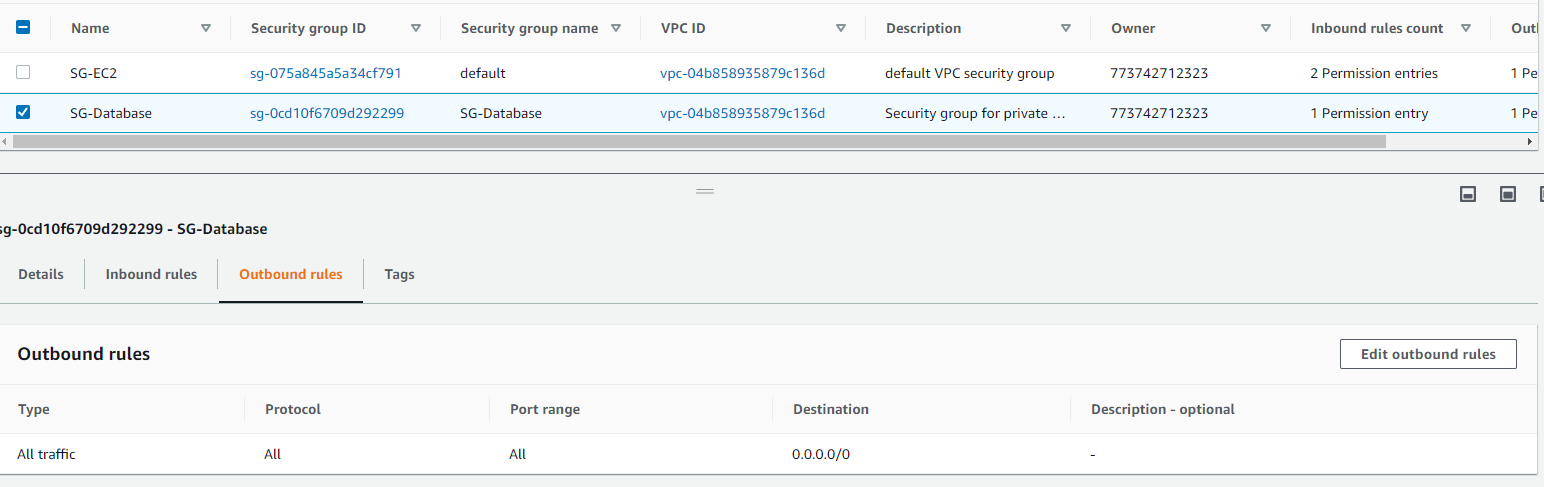
Create security group -> Name **SG-Database** -> Description **Security group for private Aurora instance** -> select VPC from dropdown **VPC-EC2-to-DB**

****

* 1. Go to Inbound rules tab -> Edit inbound rules -> Add rule -> for Type select MYSQL/Aurora -> for Source select ID of the default Security group (SG-EC2)



* 1. In Outbound rules tab we should have only All traffic rule



1. Launch EC2 Instance
   1. Select **Amazon Linux 2 AMI (HVM), SSD Volume Type**
   2. Next select t2.micro
   3. Configure Instance.

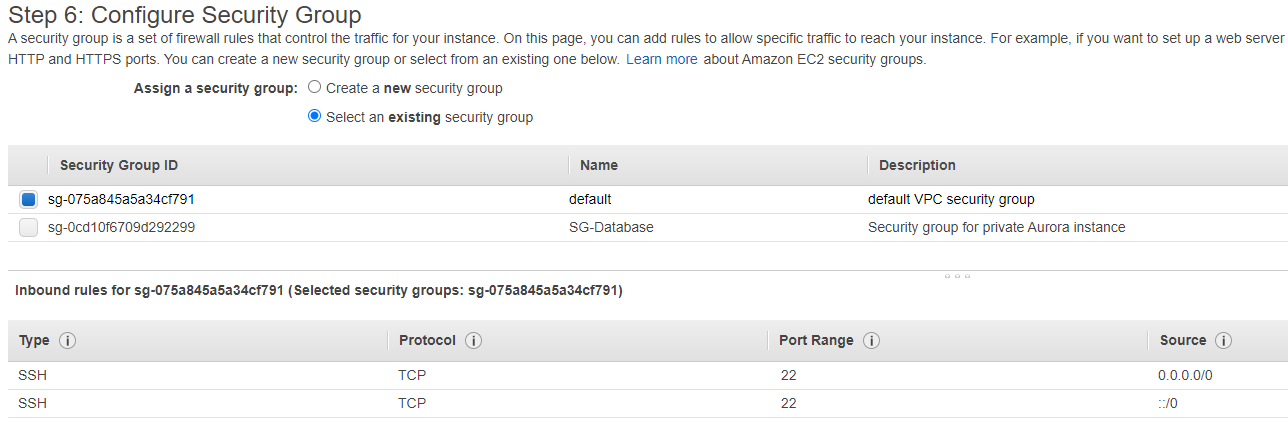
Network: **VPC-EC2-to-DB**

Subnet: **Subnet-EC2-Public**

Auto-assign Public IP: **Enable**

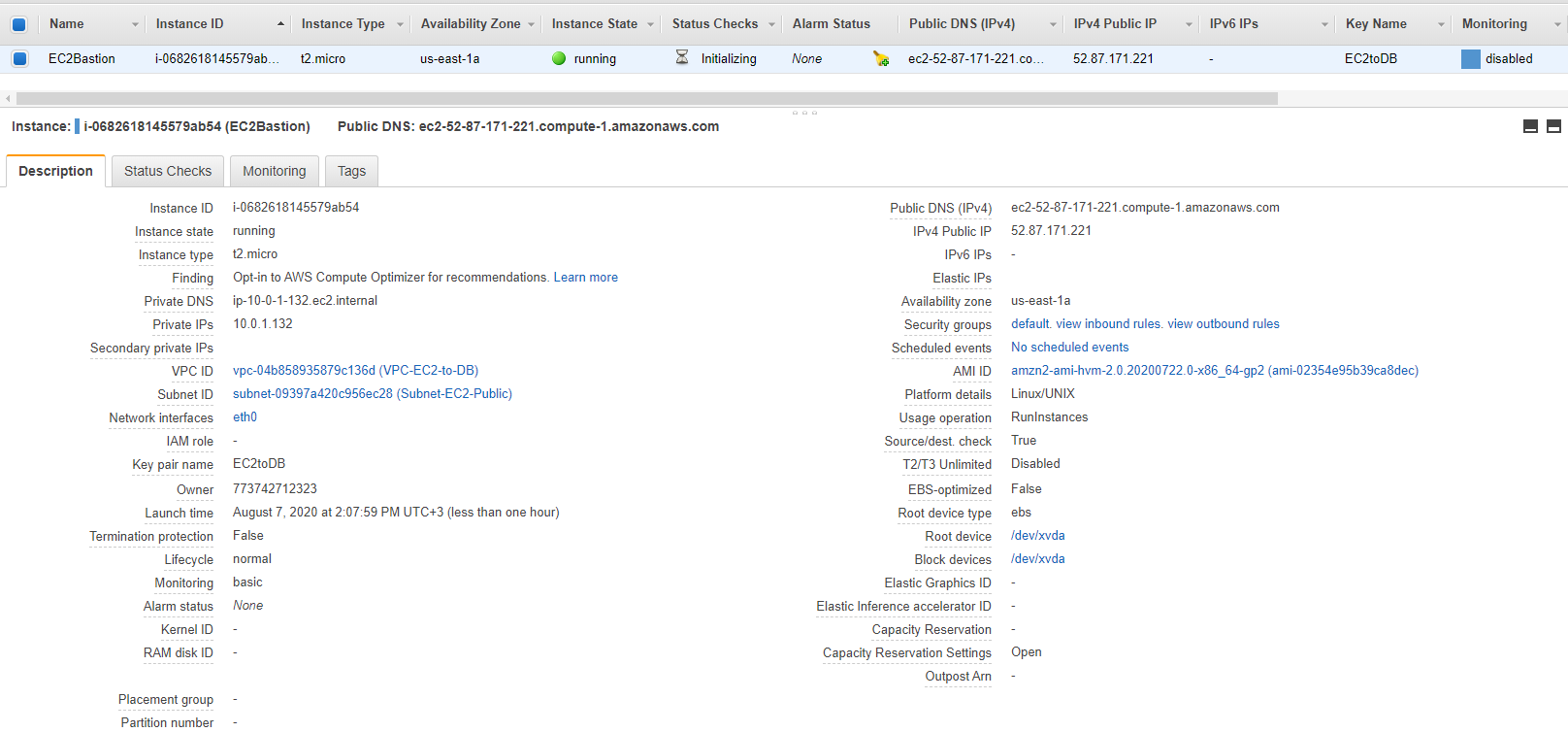
* 1. Next Add Storage – Leave default values
  2. Add tags optional. **Name** – **EC2Bastion**
  3. Configure Security Groups

Select an existing security group - > check the default Security group



* 1. Next create new Key Pair. Name it EC2toDB -> Download the kay on your local storage -> Launch the instance

We should have something like this when instance is running. Copy the **IPv4 Public IP** and save it. We are going to use it in MySQL Workbench connection configuration at later point.



1. Go to RDS service
2. From the left pane go to **Subnet groups** menu.
3. Click Create DB Subnet Group and configure it.

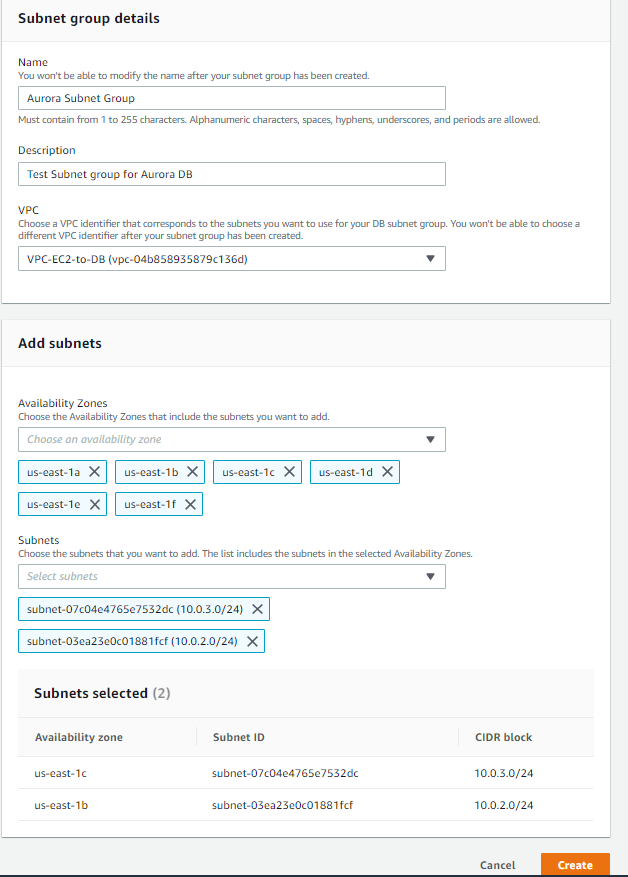
Name: **Aurora Subnet Group**

Description (optional): **Test Subnet group for Aurora DB**

VPC: select from dropdown menu our **VPC-EC2-to-DB**

Availability Zones: Select all zones from the dropdown menu

Subnets: Select the 2 subnets which are private (10.0.2.0/24 and 10.0.3.0/24)



1. Go to Database menu from the left pane and create new Database.
2. Database configuration. Change only

Templates: **DEV/Test**

Settings -> DB cluster identifier: **Aurorainstance**

Master password: Type your own password and save (remember) it

DB Instance size: **Burstable classes (includes t classes)**

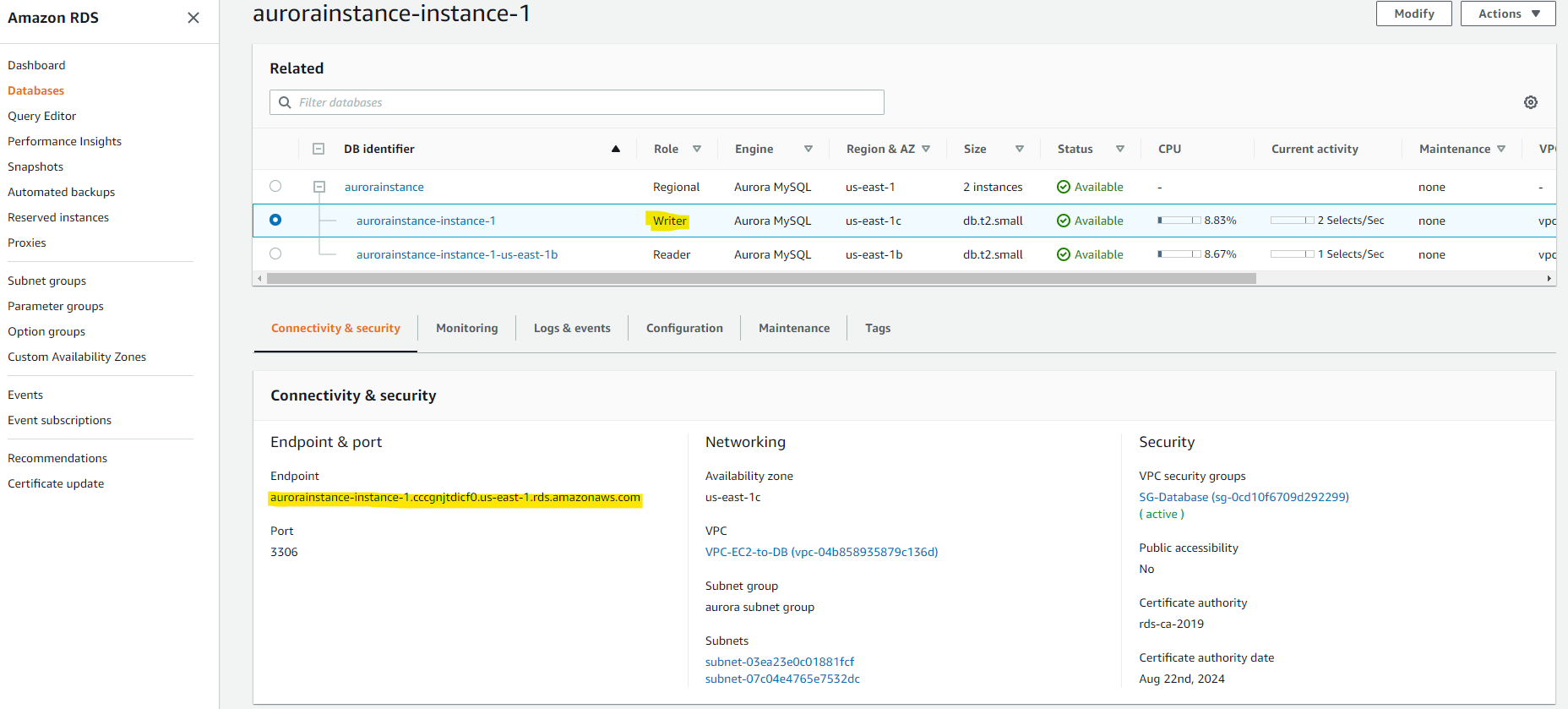
Availability and Durability: **Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)**

Connectivity -> VPC: Make sure that your VPC is selected

Connectivity -> Additional connectivity configuration -> Existing VPC security groups: Remove default and from the menu select **SG-Database (**the Security group that we created earlier for our database source**)**

Additional Configuration -> Database options -> Initial database name: **TestAuroraDB** (or name it how you prefer)

1. Click **Create Database** and wait AWS to create the database. In a few minutes Aurora DB should be active.
2. Select the instance where role is Writer and copy the Endpoint somewhere. We are going to use it in MySQL Workbench connection properties.



1. MySQL Workbench configuration. Can be downloaded from [here](https://www.mysql.com/products/workbench/).

Connection name: **AWSAuroraDB** (or what you prefer)

Connection Method: from dropdown select **Standard TCP/IP over SSH**

SSH Hostname: **52.87.171.221**:22 (the Public IP of your EC2 instance)

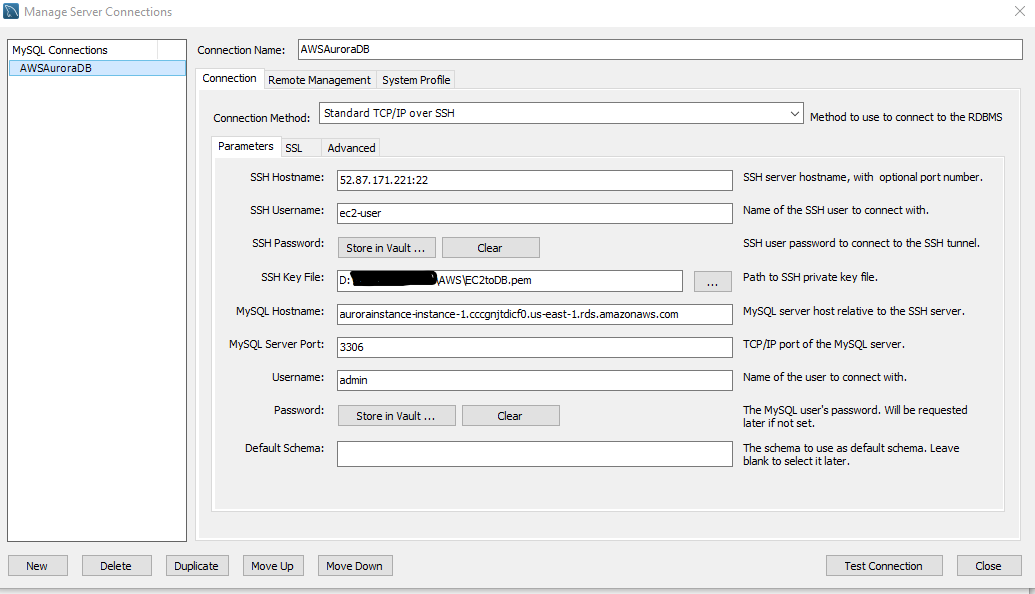
SSH Username: **ec2-user**

SSH Key File: navigate where you have saved the .pem file when you launched the EC2 instance earlier. Select it.

MySQL Hostname: the Endpoint of your Aurora “Writer” instance that you have copied earlier

Username: The username when you created Aurora Database from AWS RDS service. (admin)

Password: your password for the above user



1. From MySQL Workbench click on the newly created AWSAuroraDB connection and test whether you have access to your Aurora AWS database.

