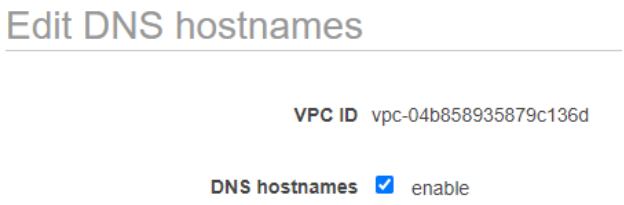
WS: Create VPC, link EC2 instance to DB within that cloud

Navigate to VPC- Your VPCs-Create VPC

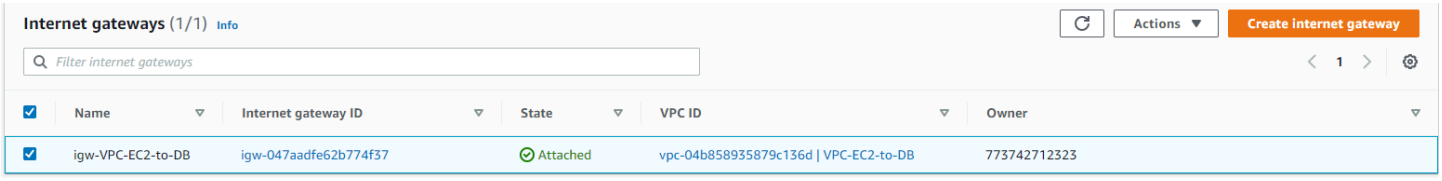
1. Select region N.Virginia
2. Create VPC
   VPC name - **VPC-EC2-to-DB**
   IPv4 CIDR – **10.0.0.0/16**

No other changes on this screen

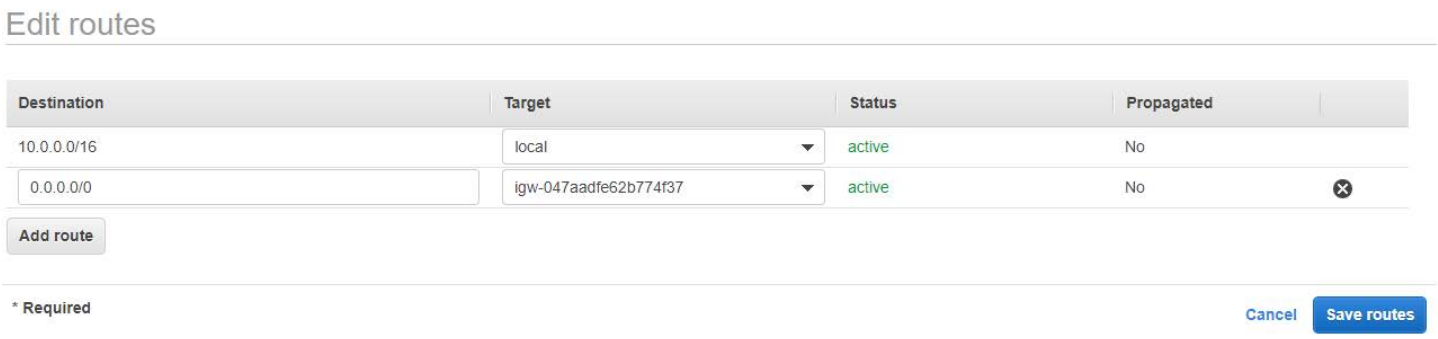
Once VCP is created, select the newly created VCP and make sure that from Actions button -> Edit DNS hostnames -> enable is checked



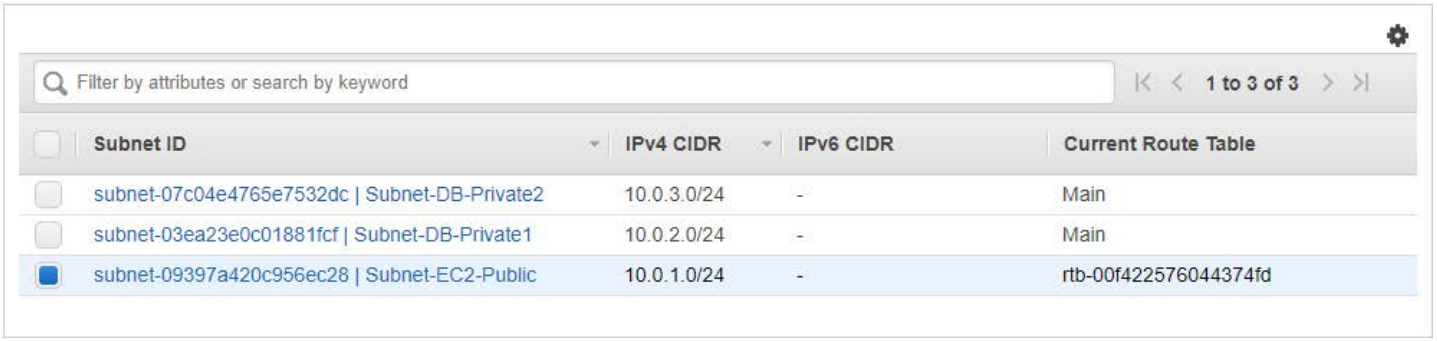
3. Create Subnets
   3.1.
   Subnet name - **Subnet-EC2-Public**
   IPv4 CIDR – **10.0.1.0/24**
   Availability Zone **- us-east-1a**
   3.2.
   Subnet name - **Subnet-DB-Private1**
   IPv4 CIDR – **10.0.2.0/24**
   Availability Zone **- us-east-1b**
   3.3.
   Subnet name - **Subnet-DB-Private2**
   IPv4 CIDR – **10.0.3.0/24**
   Availability Zone **- us-east-1c**
4. Cerate Internet Gateway
   Name - **igw-VPC-EC2-to-DB**
5. Attach created Gateway to **VPC-EC2-to-DB** VPC - Actions => Attach to VPC



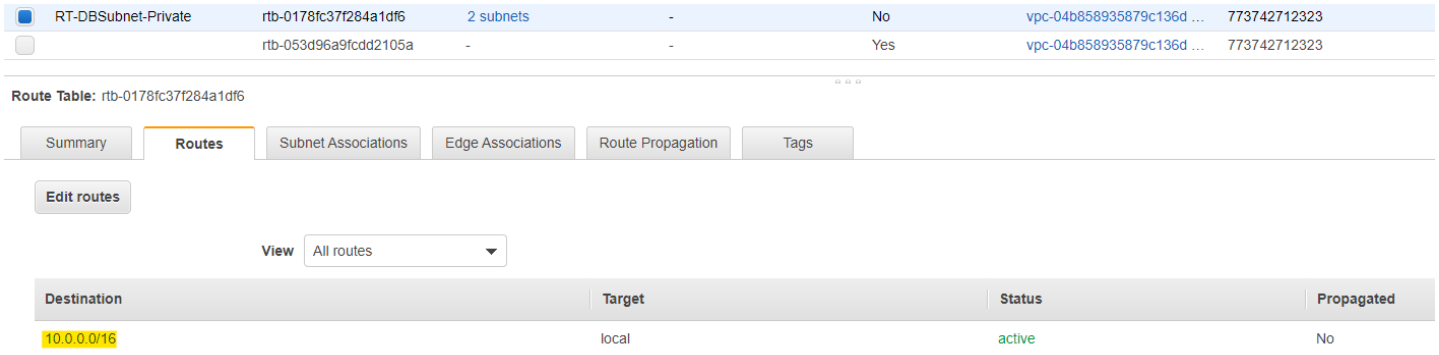
6. Create Route Table for Public Subnet Route Tables => Create Route Table
   6.1. Name - **RT-EC2Subnet-Public**
   6.2. From Tab Routes –> Edit routes –> Add route -> Destination **0.0.0.0/0** -> Target from dropdown menu select **Internet Gateway** -> Select newly created Gateway **igw-VPC-EC2-to-DB**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



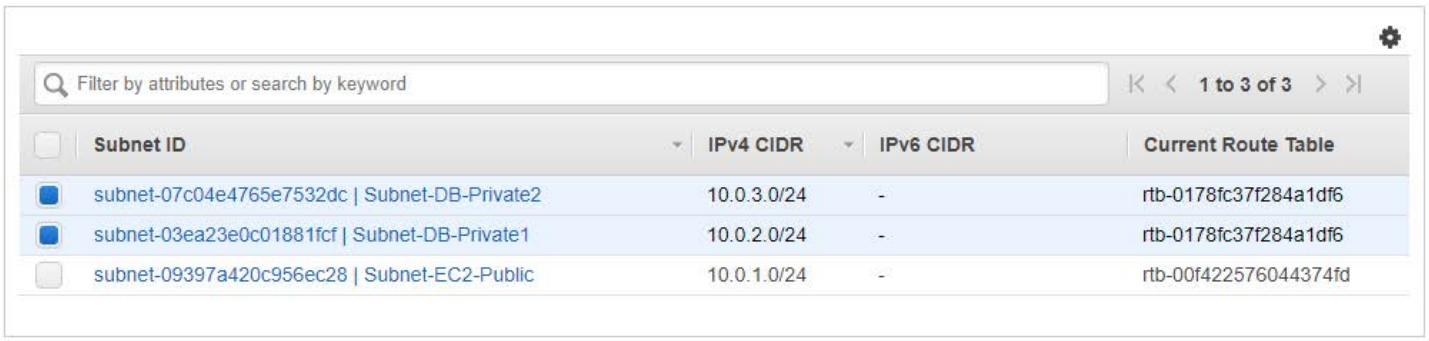
6.3. From Tab Subnet Associations -> Edit Subnet associations - > Select **Subnet-EC2-Public** subnet



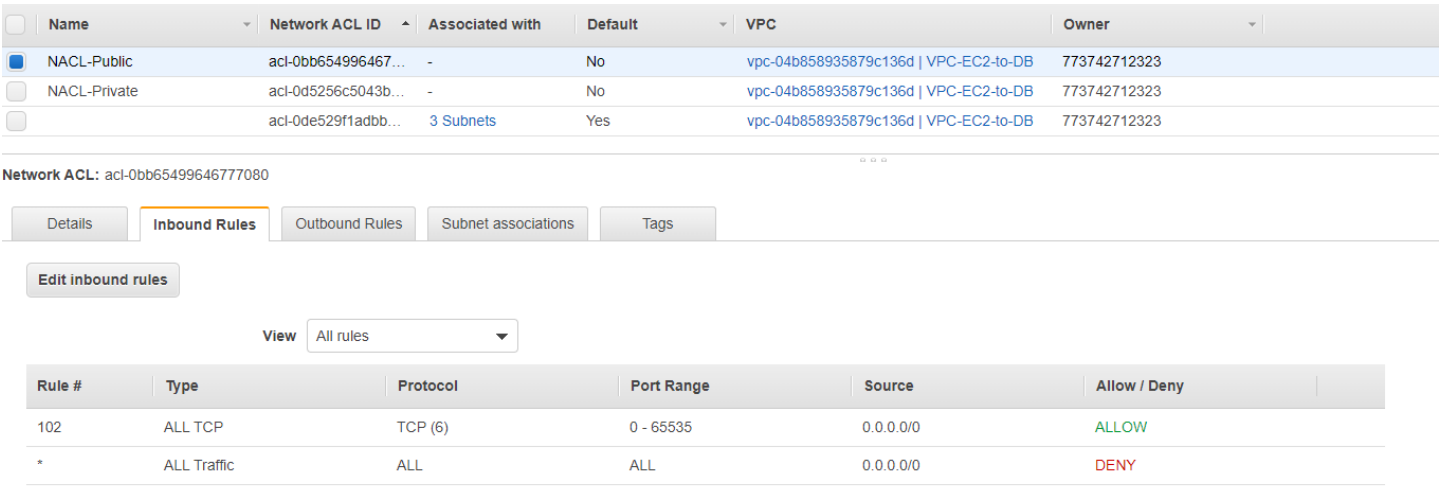
7. Create Route Table for Private Subnets
   7.1. Name - **RT-DBSubnet-Private**
   7.2. Tab Routes – should contain only VPC CIDR range



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



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



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

Name	Network ACL ID	Associated with	Default	VPC	Owner
NACL-Public	acl-0bb654996467…	-	No	vpc-04b858935879c136d | VPC-EC2-to-DB	773742712323
NACL-Private	acl-0d5256c5043b…	-	No	vpc-04b858935879c136d | VPC-EC2-to-DB	773742712323
	acl-0de529f1adbb…	3 Subnets	Yes	vpc-04b858935879c136d | VPC-EC2-to-DB	773742712323

Network ACL: acl-0bb65499646777080

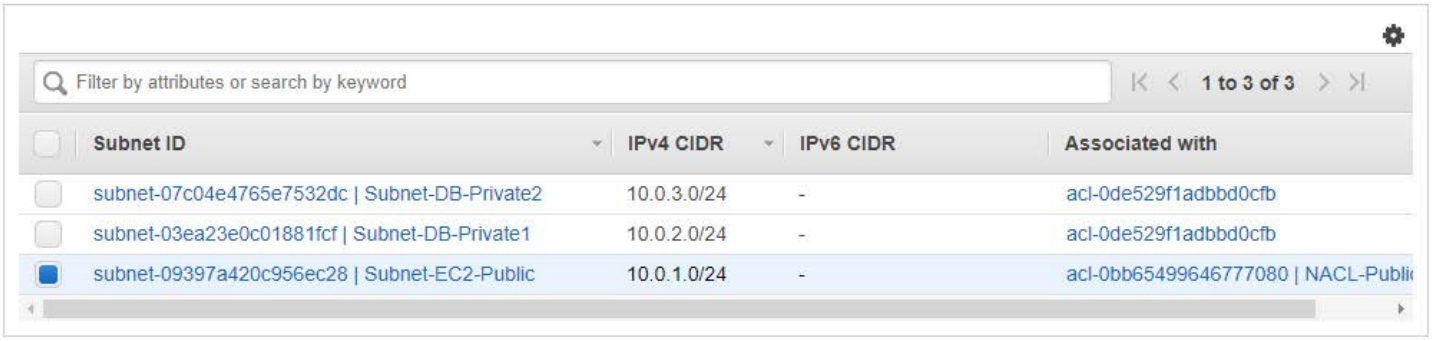
Details | Inbound Rules | **Outbound Rules** | Subnet associations | Tags

Edit outbound rules

View: All rules

Rule #	Type	Protocol	Port Range	Destination	Allow / Deny
102	ALL TCP	TCP (6)	0 - 65535	0.0.0.0/0	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

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



Cancel | **Edit**

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

NACL-Private	acl-0d5256c5043b…	2 Subnets	No	vpc-04b858935879c136d | VPC-EC2-to-DB	773742712323
	acl-0de529f1adbb…	-	Yes	vpc-04b858935879c136d | VPC-EC2-to-DB	773742712323

Network ACL: acl-0d5256c5043b3fb4e

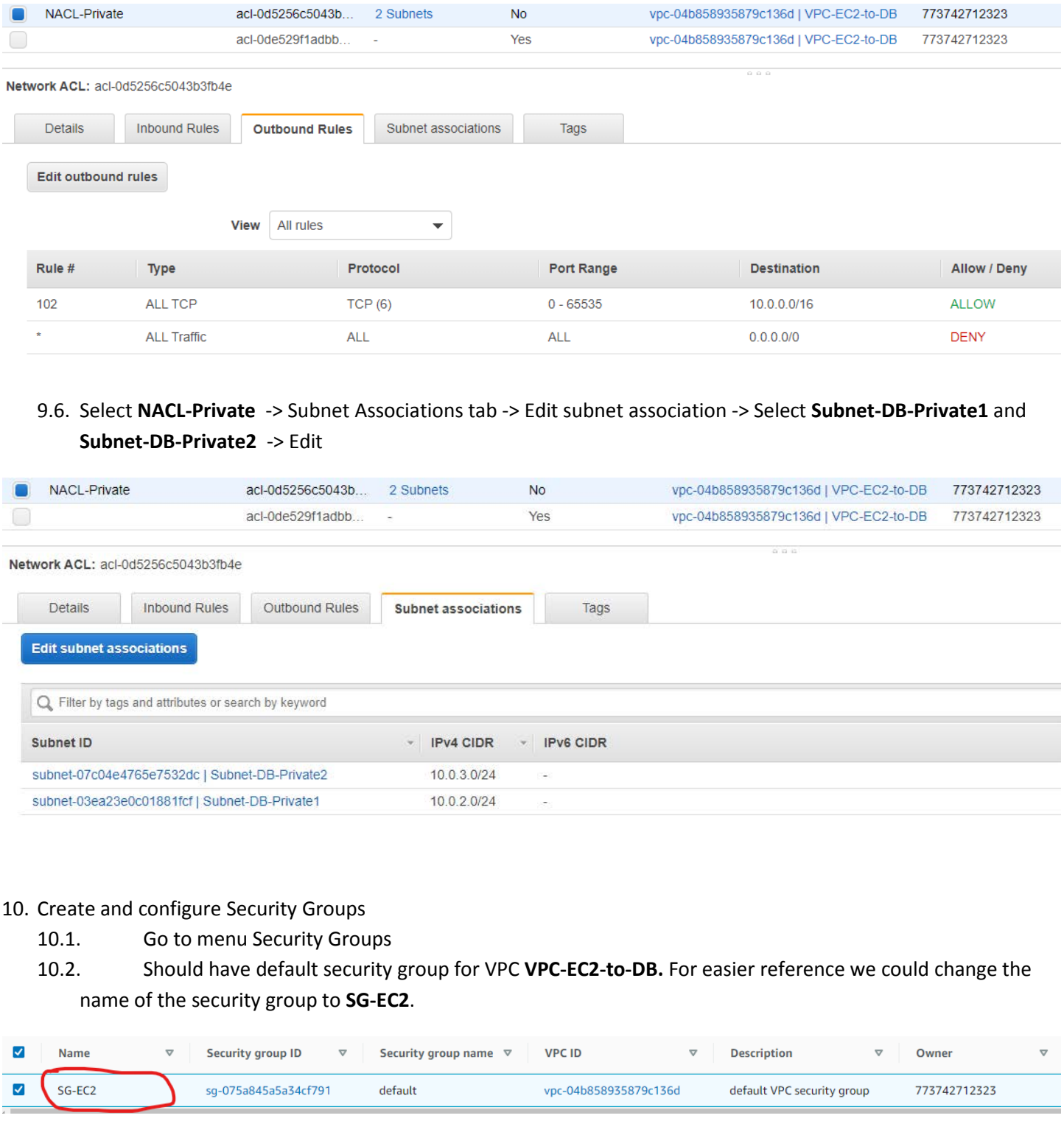
Details | **Inbound Rules** | Outbound Rules | Subnet associations | Tags

Edit inbound rules

View: All rules

Rule #	Type	Protocol	Port Range	Source	Allow / Deny
102	ALL TCP	TCP (6)	0 - 65535	10.0.0.0/16	ALLOW
*	ALL Traffic	ALL	ALL	0.0.0.0/0	DENY

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



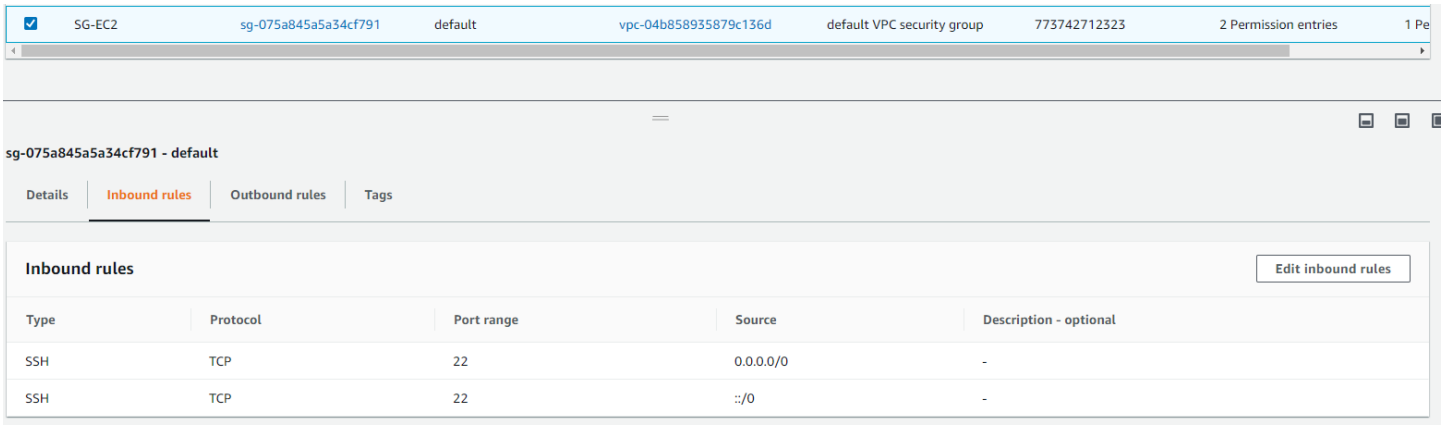
9.6. Select **NACL-Private** -> Subnet Associations tab -> Edit subnet association -> Select **Subnet-DB-Private1** and **Subnet-DB-Private2** -> Edit

10. Create and configure Security Groups

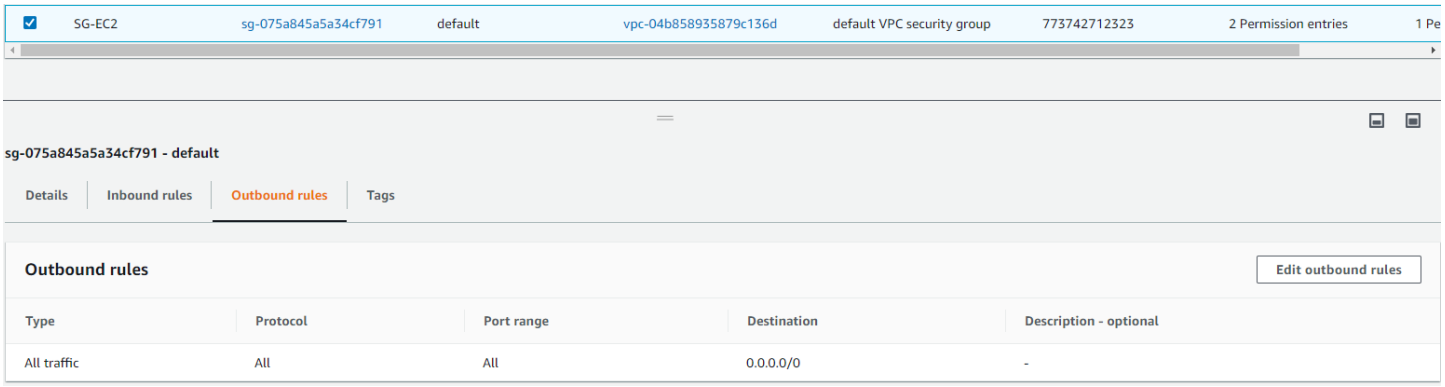
10.1. Go to menu Security Groups

10.2. 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**.

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

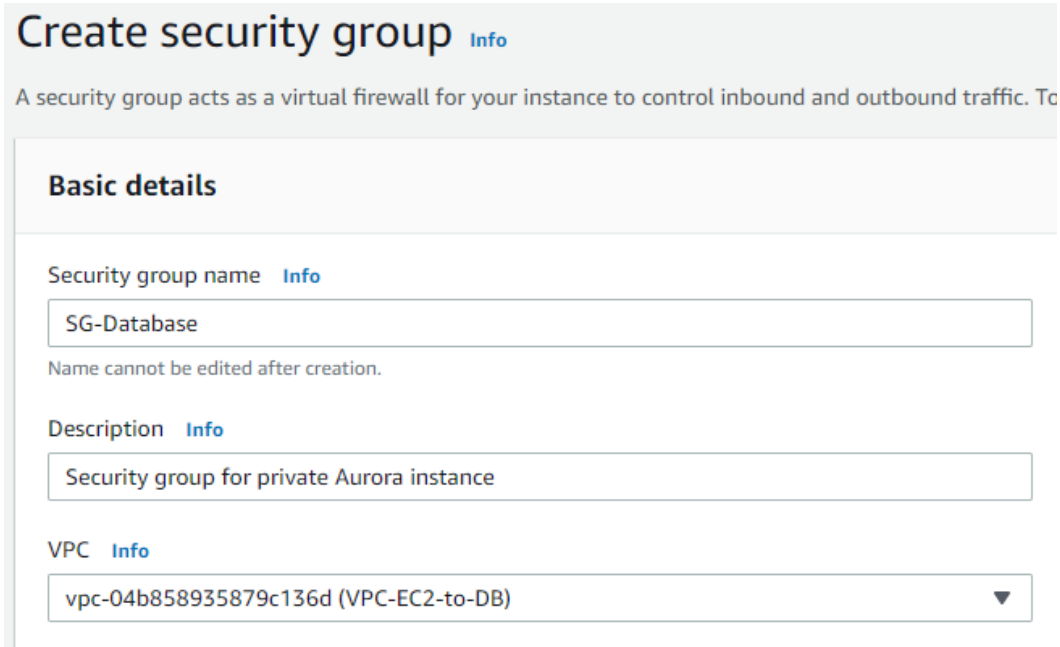


10.4. Under tab Outbound rules we should have only All Traffic

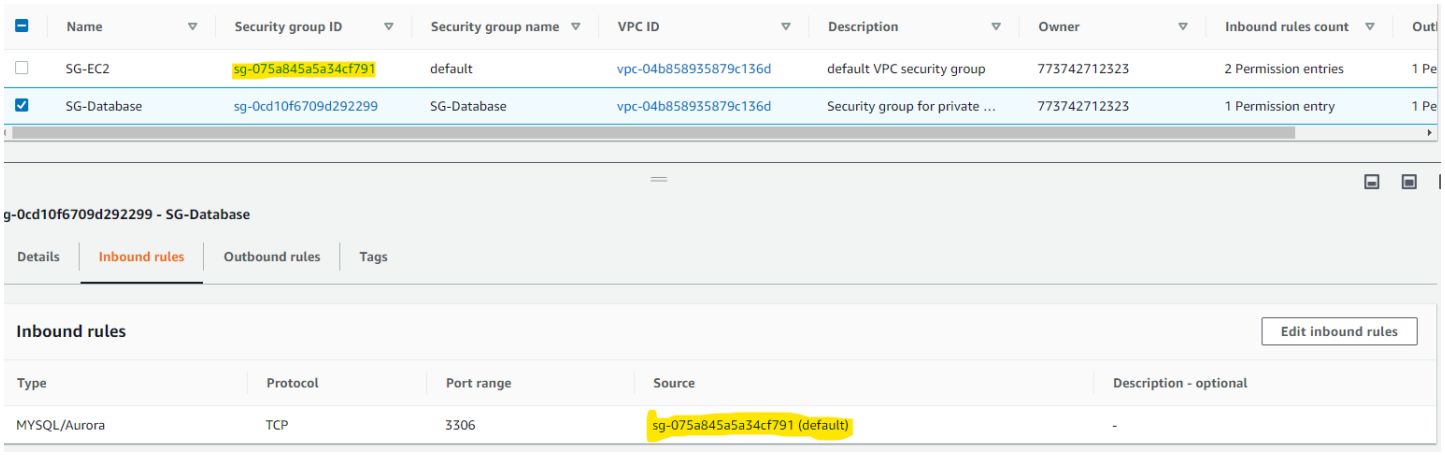


10.5. Create Security group for private Database source.

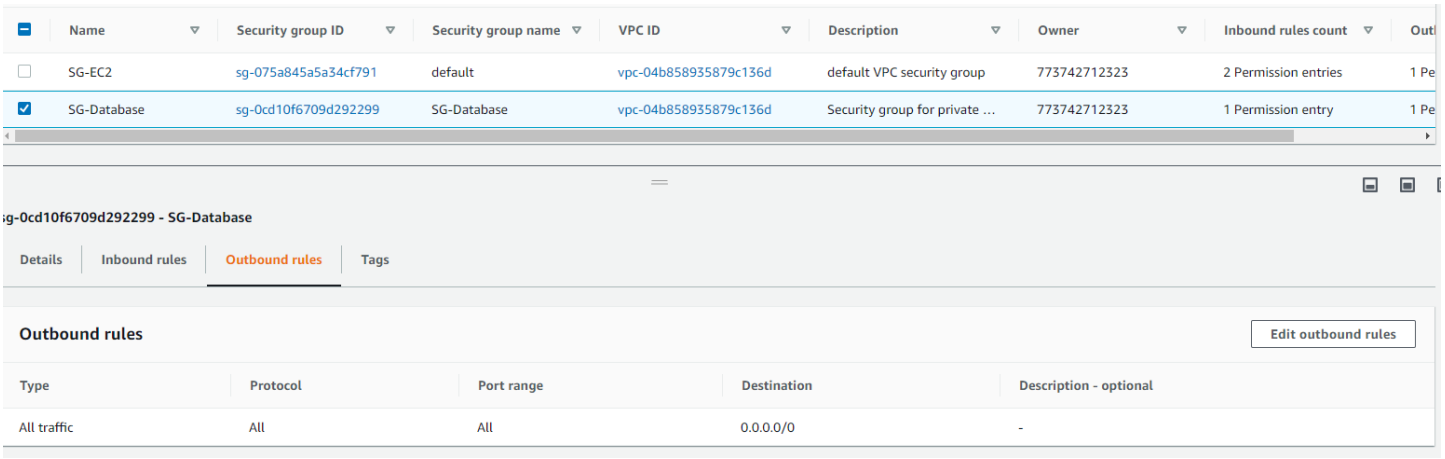
Create security group -> Name **SG-Database** -> Description **Security group for private Aurora instance** -> select VPC from dropdown **VPC-EC2-to-DB**



10.6. 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)



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



11. Launch EC2 Instance
    - 11.1. Select **Amazon Linux 2 AMI (HVM), SSD Volume Type**
    - 11.2. Next select t2.micro
    - 11.3. Configure Instance.

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

      Subnet: **Subnet-EC2-Public**

      Auto-assign Public IP: **Enable**
    - 11.4. Next Add Storage – Leave default values
    - 11.5. Add tags optional. **Name – EC2Bastion**
    - 11.6. Configure Security Groups

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

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server HTTP and HTTPS ports. You can create a new security group or select from an existing one below. Learn more about Amazon EC2 security groups.

Assign a security group: ○ Create a **new** security group

● Select an **existing** security group

Security Group ID	Name	Description
sg-075a845a5a34cf791	default	default VPC security group
sg-0cd10f6709d292299	SG-Database	Security group for private Aurora instance

Inbound rules for sg-075a845a5a34cf791 (Selected security groups: sg-075a845a5a34cf791)

Type	Protocol	Port Range	Source
SSH	TCP	22	0.0.0.0/0
SSH	TCP	22	::/0

11.7. 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.

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS (IPv4)	IPv4 Public IP	IPv6 IPs	Key Name	Monitoring
EC2Bastion	i-0682618145579ab...	t2.micro	us-east-1a	running	Initializing	None	ec2-52-87-171-221.co...	52.87.171.221	-	EC2toDB	disabled

Instance: i-0682618145579ab54 (EC2Bastion) **Public DNS:** ec2-52-87-171-221.compute-1.amazonaws.com

Description | Status Checks | Monitoring | Tags

Instance ID	i-0682618145579ab54	Public DNS (IPv4)	ec2-52-87-171-221.compute-1.amazonaws.com
Instance state	running	IPv4 Public IP	52.87.171.221
Instance type	t2.micro	IPv6 IPs	-
Finding	Opt-in to AWS Compute Optimizer for recommendations. Learn more	Elastic IPs	
Private DNS	ip-10-0-1-132.ec2.internal	Availability zone	us-east-1a
Private IPs	10.0.1.132	Security groups	default. view inbound rules. view outbound rules
Secondary private IPs		Scheduled events	No scheduled events
VPC ID	vpc-04b858935879c136d (VPC-EC2-to-DB)	AMI ID	amzn2-ami-hvm-2.0.20200722.0-x86_64-gp2 (ami-02354e95b39ca8dec)
Subnet ID	subnet-09397a420c956ec28 (Subnet-EC2-Public)	Platform details	Linux/UNIX
Network interfaces	eth0	Usage operation	RunInstances
IAM role	-	Source/dest. check	True
Key pair name	EC2toDB	T2/T3 Unlimited	Disabled
Owner	773742712323	EBS-optimized	False
Launch time	August 7, 2020 at 2:07:59 PM UTC+3 (less than one hour)	Root device type	ebs
Termination protection	False	Root device	/dev/xvda
Lifecycle	normal	Block devices	/dev/xvda
Monitoring	basic	Elastic Graphics ID	-
Alarm status	None	Elastic Inference accelerator ID	-
Kernel ID	-	Capacity Reservation	-
RAM disk ID	-	Capacity Reservation Settings	Open
		Outpost Arn	-
Placement group	-		
Partition number	-		

12. Go to RDS service
13. From the left pane go to **Subnet groups** menu.
14. 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)

Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.

Aurora Subnet Group

Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description

Test Subnet group for Aurora DB

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.

VPC-EC2-to-DB (vpc-04b858935879c136d)

Add subnets

Availability Zones
Choose the Availability Zones that include the subnets you want to add.

Choose an availability zone

us-east-1a ✕ us-east-1b ✕ us-east-1c ✕ us-east-1d ✕

us-east-1e ✕ us-east-1f ✕

Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.

Select subnets

subnet-07c04e4765e7532dc (10.0.3.0/24) ✕

subnet-03ea23e0c01881fcf (10.0.2.0/24) ✕

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1c	subnet-07c04e4765e7532dc	10.0.3.0/24
us-east-1b	subnet-03ea23e0c01881fcf	10.0.2.0/24

Cancel **Create**

15. Go to Database menu from the left pane and create new Database.
16. Database configuration. Change only

    Templates: **DEV/Test**

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

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

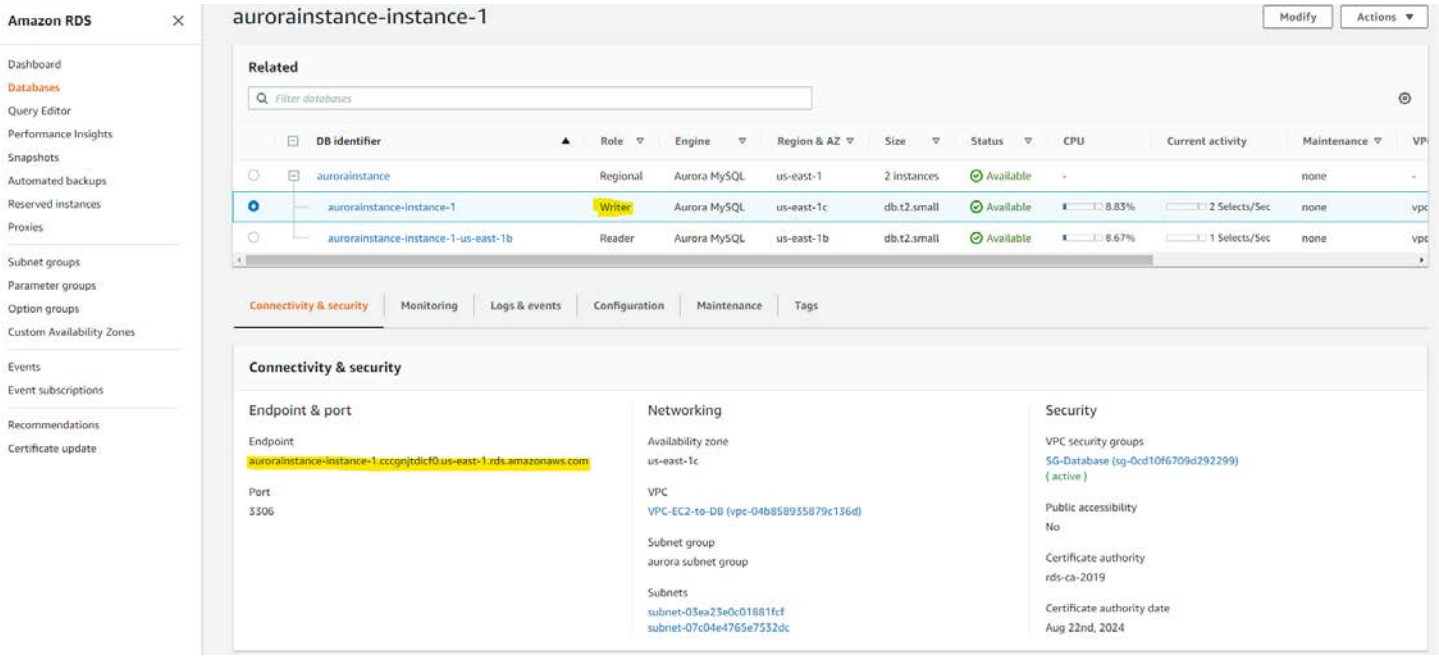
    DB Instance size: **Burstable classes (includes t classes)**

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



19. MySQL Workbench configuration. Can be downloaded from here.

    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)

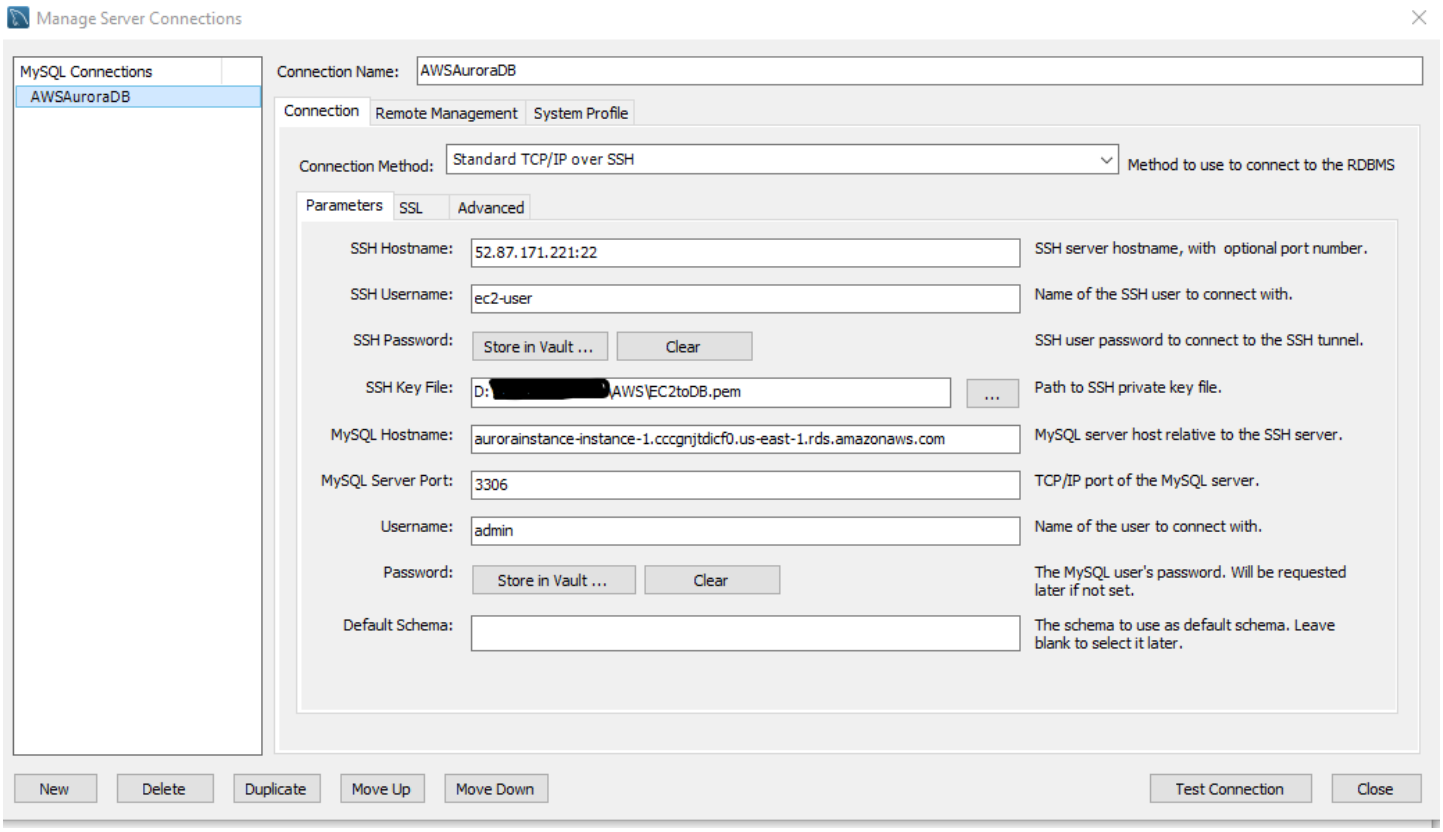
    SSH Username: **ec2-user**

    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



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