Introduction to AWS Relational Database Service

Task 6: Creating security group

A security group acts as a virtual firewall for instances, controlling inbound and outbound traffic. Security groups operate at the instance network interface level, not the subnet level. Therefore, each instance can have its own firewall that controls traffic. If you do not specify a particular security group at launch time, the instance is automatically assigned to the default security group for the VPC.

In this task, you create a security groups for rds instance:

- Login into the AWS management console and navigate to **VPC** service.
- In the left navigation pane, choose **Security Groups**.
- Choose **Create security group** and configure it with the following details:
- **Security group name**: Enter rds-sg
- **Description**: Enter rds security group
- Select *your* ***vpc***

- For **Inbound rules**, choose **Add rule** and configure it with the following details:
- **Type** : MYSQL/Aurora
- **Source**: Select *0.0.0.0/0* (Anywhere)
- For **Tags - optional**, choose **Add new tag** and configure it with the following details:
- **Key**: Enter Nam
- **Value**: Enter rds-sg

Note: By adding tags, it will be easy to identify the subnets in the subnet list.

- At the bottom of the page, choose **Create security group**

This will allow communication with the RDS instances on port 3306.

Task 7: Launch web app instances and database resources, and deploy the application

Create the database in vpc subnets

Create a DB (database) subnet group so that the RDS will be deployed within the subnets you want to use.

- On the **Services** menu, type RDS in the search bar and choose RDS.
- In the left navigation pane, choose ***Subnet groups***.
- Choose **Create DB subnet group**
  and configure it with the following details:

- **Name**: Enter LabVPCRDSsubnetgr
- **Description**: Enter Subnet group for RDS
- Select your *VPC*
- **Availability Zones**: Select two Availability Zones
- **Subnets**: Choose a subnet from each availability zone.
- Choose **Create**

The DB subnet group has been created successfully. Now let's create the database.

- In the left navigation pane, choose **Databases**.

- Choose **Create database** and configure it with the following details:

- **Choose a database creation method**: *Standard Create*
- **Engine Options**:
    1. **Engine type**: Select *MySQL*
    2. **Templates:** *Free tier*
- **Templates**: Select ***Free tier***
- **Settings**:
    1. **DB Cluster identifier**: Enter LabVPCDBClust
    2. **Master username**: Leave the default as admir
    3. **Master password**: Enter testingrdscluster
    4. **Confirm password**: Enter testingrdscluster
- In the **Storage** Section
    1. Disable **Enable storage autoscaling**
- **Connectivity**:
    1. **Virtual private cloud(VPC)**: Select *your vpc*
- **Subnet group**: Select *LabVPCRDSsubnetgroup*
- **Public access**: Select ***No***
- **VPC security group**: Select *Choose existing*
- **Existing VPC security groups**: Select ***rds-sg*** and remove ***default***
- **Database port**: Select *3306*
- Choose **Additional Configuration**
- **Initial database name**: Enter Population
- Uncheck *automatic backups* in the **Backup** section.

The other values will be left to the default values selected.

- Choose **Create database**

This will create a writer instance in one Availability Zone and a reader instance in the other Availability Zone. It will take few minutes to complete. Choose the refresh button to see the status updated. You can proceed once you receive the *Successfully created database* message.

- Choose *LabVPCDBCluster* under **DB identifier** and then copy the *Endpoint name* under **Endpoints** for the *Writer* type and the *Reader* type. These will be needed when the application is deployed on the EC2 instances.

Note: The database may initially show two *Reader* types, but it will show the *Writer* type once the creation is complete.