**Task 1: Create a Security Group for the RDS DB Instance**

In this task, you will create a security group to allow your web server to access your RDS DB instance. The security group will be used when you launch the database instance.

1. In the AWS Management Console, select the **Services** menu, and then select **VPC** under **Networking & Content Delivery**.
2. In the left navigation pane, click **Security Groups**.
3. Click Create security group and then configure:
   * **Security group name:** DB Security Group
   * **Description:** Permit access from Web Security Group
   * **VPC:** *Lab VPC*

You will now add a rule to the security group to permit inbound database requests. The security group currently has no rules. You will add a rule to permit access from the *Web Security Group*.

1. In the **Inbound rules** section, click Add rule, then configure:
   * **Type:** *MySQL/Aurora (3306)*
   * **CIDR, IP, Security Group or Prefix List:** Type sg and then select *Web Security Group*.

This configures the Database security group to permit inbound traffic on port 3306 from any EC2 instance that is associated with the *Web Security Group*.

1. Scroll to the bottom of the screen, then click Create security group

You will use this security group when launching the Amazon RDS database.

**Task 2: Create a DB Subnet Group**

In this task, you will create a *DB subnet group* that is used to tell RDS which subnets can be used for the database. Each DB subnet group requires subnets in at least two Availability Zones.

1. In the AWS Management Console, select the **Services** menu, and then select **RDS** under **Database**.
2. In the left navigation pane, click **Subnet groups**.

If the navigation pane is not visible, click the menu icon in the top-left corner.

1. Click Create DB Subnet Group then configure:
   * **Name:** DB Subnet Group
   * **Description:** DB Subnet Group
   * **VPC ID:** *Lab VPC*
2. In the **Add subnets** section for *Availability zones*, click the , then:
   * Select the first Availability zone
   * Select the second Availability zone
3. For **Subnets**, click the , then:
   * For the first Availability zone, select *10.0.1.0/24*
   * For the second Availability zone, select *10.0.3.0/24*
4. Click Create

This adds Private Subnet 1 (10.0.1.0/24) and Private Subnet 2 (10.0.3.0/24). You will use this DB subnet group when creating the database in the next task.

**Task 3: Create an Amazon RDS DB Instance**

In this task, you will configure and launch a Multi-AZ Amazon RDS for MySQL database instance.

Amazon RDS ***Multi-AZ*** deployments provide enhanced availability and durability for Database (DB) instances, making them a natural fit for production database workloads. When you provision a Multi-AZ DB instance, Amazon RDS automatically creates a primary DB instance and synchronously replicates the data to a standby instance in a different Availability Zone (AZ).

1. In the left navigation pane, click **Databases**.
2. Click Create database

If you see **Switch to the new database creation flow** at the top of the screen, please click it.

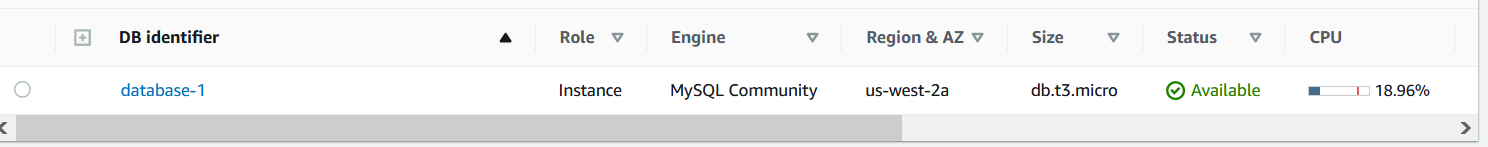
1. Select **MySQL**.
2. Under **Settings**, configure:
   * **DB instance identifier:** lab-db
   * **Master username:** main
   * **Master password:** lab-password
   * **Confirm password:** lab-password
3. Under **DB instance size**, configure:
   * Select **Burstable classes (includes t classes)**.
   * Select *db.t3.micro*. **Note**: If the *db.t3.micro* instance is not available in the Availability Zone, choose *db.t2.micro* instead.
4. Under **Storage**, configure:
   * Select *General Purpose (SSD)* under **Storage type**.
5. Under **Connectivity**, configure:
   * **Virtual Private Cloud (VPC):** *Lab VPC*
6. Under **VPC security group** select **Choose existing**
7. Under **Existing VPC security groups**
   * Use X to Remove *default*
   * Select **DB Security Group** to highlight it in blue
8. Expand **Additional configuration**, then configure:
   * **Initial database name:** lab
   * Uncheck **Enable automated backups**.
   * Uncheck **Enable Enhanced monitoring**.

This will turn off backups, which is not normally recommended, but will make the database deploy faster for this lab.

1. Scroll to the bottom of the screen, then click Create database

Your database will now be launched.

1. Click **lab-db** (click the link itself).



You will now need to wait **approximately 4 minutes** for the database to be available. The deployment process is deploying a database in two different Availability zones.

While you are waiting, you might want to review the [Amazon RDS FAQs](https://aws.amazon.com/rds/faqs/) or grab a cup of coffee.

Graphical user interface, application

Description automatically generated

1. Wait until **Info** changes to **Modifying** or **Available**.
2. Scroll down to the **Connectivity & Security** section and copy the **Endpoint** field.

It will look similar to: *lab-db.cggq8lhnxvnv.us-west-2.rds.amazonaws.com*

1. Paste the Endpoint value into a text editor. You will use it later in the lab.

## Task 4: Using SSH to Connect to the Linux Host

### Windows Users: Using SSH to Connect

These instructions are specifically for Windows users. If you are using macOS or Linux, [skip to the next section](https://labs.vocareum.com/web/1827760/707856.0/ASNLIB/public/docs/lang/en-us/README.html#ssh-MACLinux).

1. Click the Details drop down menu above the instructions you are currently reading, and then click Show. A Credentials window will be presented.
2. Click the **Download PPK** button and save the **labsuser.ppk** file. Typically your browser will save it to the Downloads directory.
3. Exit the Details panel by clicking the **X**.
4. Download **PuTTY** to SSH into the Amazon EC2 instance. If you do not have PuTTY installed on your computer, [download it here](https://the.earth.li/~sgtatham/putty/latest/w64/putty.exe).
5. Open **putty.exe**
6. Configure PuTTY timeout to keep the PuTTY session open for a longer period of time:
   * Click **Connection**.
   * Set **Seconds between keepalives** to 30.
7. Configure your PuTTY session:
   * Click **Session**.
   * **Host Name (or IP address):** Paste the **IP address of the Linux Host instance** you saved in the Lab Details file earlier.
   * Back in PuTTY, in the **Connection** list, expand **SSH**
   * Click **Auth** (don't expand it).
   * Click **Browse**.
   * Browse to and select the **labsuser.ppk** file that you downloaded.
   * Click **Open** to select it.
   * Click **Open** again.
8. Click **Yes**, to trust and connect to the host.
9. When prompted **login as**, enter: ec2-user. This will connect you to the EC2 instance.

## Task 5: Accessing Database from BASH

1. mysql - -version



1. sudo yum install mysql  
   if   
     
     
   Text

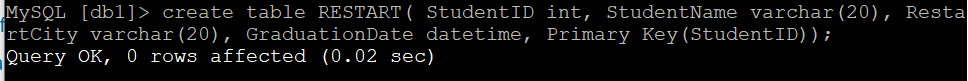
   Description automatically generated  
   y  
   Text

   Description automatically generated
2. launch db

CMD on CLI: mysql -h <db endpoint> -P <port number> -u <user> -p

Text

Description automatically generated

1. Enter password
2. Create database
3. Use database
4. Create table  
   CREATE TABLE RESTART (StudentID int, StudentName varchar(20), RestartCity varchar(20), GraduationDate (datetime));  
     
   
5. Show tables;

Text

Description automatically generated

1. Describe RESTART;  
   A screenshot of a computer

   Description automatically generated with medium confidence
2. Insert Into RESTART & SELECT \* FROM db1.RESTART;  
   Graphical user interface, text

   Description automatically generated
3. Insert Into CLOUD\_PRACTITIONER… & SELECT \* FROM db1. CLOUD\_PRACTITIONER;

Text

Description automatically generated

1. RIGHT JOIN  
     
   SELECT RESTART.StudentID, RESTART.StudentName, CLOUD\_PRACTITIONER.CertificationDate FROM RESTART RIGHT JOIN CLOUD\_PRACTITIONER ON RESTART.StudentID = CLOUD\_PRACTITIONER.StudentID;  
     
   Graphical user interface, text

   Description automatically generated

s