

# Creating an Amazon RDS Database

## Lab overview

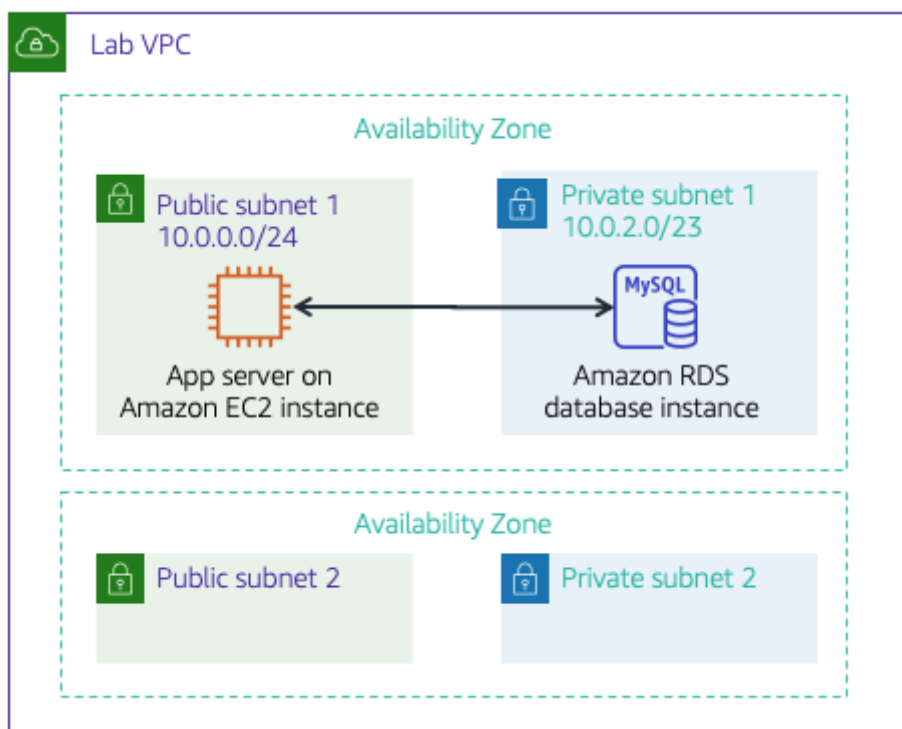
Traditionally, creating a database can be a complex process that requires either a database administrator or a systems administrator. In the cloud, you can simplify this process by using Amazon Relational Database Service (Amazon RDS).

## Objectives

After completing this lab, you should be able to:

- Launch a database using Amazon RDS
- Configure a web application to connect to the database instance

At the end of this lab, your architecture will look like the following example:



## Task 1: Creating an Amazon RDS database

In this task, you create a MySQL database in your virtual private cloud (VPC). MySQL is a popular open-source relational database management system (RDBMS), so there are no software licensing fees.

**Windows Users:** Use Chrome or Firefox as your web browser for this lab. The lab instructions are not compatible with Internet Explorer because of a difference in the Amazon RDS console.

5. On the **Services** menu, choose **RDS**.
6. Choose **Create database**
7. Under **Engine options**, select **MySQL**.

The options include several use cases, ranging from enterprise-class databases to Dev/Test systems. In the options, you might notice Amazon Aurora. Aurora is a MySQL-compatible system that was re-architected for the cloud. If your company uses large-scale MySQL or PostgreSQL databases, Aurora can provide enhanced performance.

8. In the **Templates** section, select **Dev/Test**.

You can now select a database configuration, including software version, instance class, storage, and login settings. The Multi-AZ deployment option automatically creates a replica of the database in a second Availability Zone for high availability.

9. In **Availability and durability** section, choose **Single DB instance**.

10. In the **Settings** section, configure the following options:

- **DB instance identifier:** inventory-db
- **Master username:** admin
- **Master password:** lab-password
- **Confirm password:** lab-password

11. In the **DB instance class** section, configure the following options:

- Select **Burstable classes (includes t classes)**.
- Select **db.t3.micro**.

12. In the **Storage** section, configure the following options:

- **Storage type:** Select **General Purpose SSD (gp2)**.
- **Allocated storage:** Enter **20 GiB**.

13. In the **Connectivity** section, configure the following option:

- **Virtual private cloud (VPC):** Lab VPC

14. In the **Connectivity** section, for **Existing VPC security groups**, choose the **X** on **default** to remove this security group. Then choose the dropdown list, and select **DB-SG** to add it.

15. Scroll to the **Monitoring** section, and clear (deselect) the **Enable Enhanced monitoring** option.

16. Scroll to the **Additional configuration** section, and choose to expand it.

17. For **Initial database name**, enter inventory

This is the logical name of the database that the application will use.

You can review the many other options displayed on the page, but leave them set to their default values. Options include automatic backups, the ability to export log files, and automatic version upgrades. The ability to activate these features with check boxes demonstrates the power of using a fully managed database solution instead of installing, backing up, and maintaining the database yourself.

18. At the bottom of the page, choose **Create database**

You should receive this message: **Creating database inventory-db**.

If you receive an error message that mentions **rds-monitoring-role**, confirm that you have cleared the **Enable Enhanced monitoring** option in the previous step, and then try again.

Before you continue to the next task, the database instance status must be **Available**. This process could take several minutes.

## Task 2: Configuring web application communication with a database instance

This lab automatically deployed an Amazon Elastic Compute Cloud (Amazon EC2) instance with a running web application. You must use the IP address of the instance to connect to the application.

19. On the **Services** menu, choose **EC2**.
20. In the left navigation pane, choose **Instances**.

In the center pane, there should be a running instance that is named **App Server**.

21. Select the check box for the **App Server** instance.
22. In the **Details** tab, copy the **Public IPv4 address** to your clipboard.

**Tip:** If you hover over the IP address, a copy icon appears. To copy the displayed value, choose the icon.

23. Open a new web browser tab, paste the IP address into the address bar, and then press Enter.

The web application should appear. It does not display much information because the application is not yet connected to the database.

24. Choose **Settings**.

You can now configure the application to use the Amazon RDS database instance that you created earlier. You first retrieve the database endpoint so that the application knows how to connect to a database.

25. Return to the AWS Management Console, but do not close the application tab. (You will return to it soon.)
26. On the **Services** menu, choose **RDS**.
27. In the left navigation pane, choose **Databases**.
28. Choose **inventory-db**.
29. Scroll to the **Connectivity & security** section, and copy the **Endpoint** to your clipboard.

It should look similar to this example: **inventory-db.crwxbgqad61a.rds.amazonaws.com**

30. Return to the browser tab with the inventory application, and enter the following values:
  - For **Endpoint**, paste the endpoint you copied earlier.
  - For **Database**, enter **inventory**
  - For **Username**, enter **admin**
  - For **Password**, enter **lab-password**
  - Choose **Save**.

The application will now connect to the database, load some initial data, and display information.

31. You can use the web application to Add inventory, edit, and delete inventory information.

The inventory information is stored in the Amazon RDS MySQL database that you created earlier in the lab. This means that in the event of any failure in the application server, you will not lose any data. It also means that multiple application servers can access the same data.

32. Insert new records into the table. Ensure that the table has **5** or more inventory records before submitting your work.