

# 160-[DF]-Lab - Build Your Database Server and Interact with Your DB Using an App

## Build Your DB Server and Interact With Your DB Using an App

This lab is designed to reinforce the concept of leveraging an AWS-managed database instance for solving relational database needs.

**Amazon Relational Database Service** (Amazon RDS) makes it easy to set up, operate, and scale a relational database in the cloud. It provides cost-efficient and resizable capacity while managing time-consuming database administration tasks, which allows you to focus on your applications and business. Amazon RDS provides you with six familiar database engines to choose from: Amazon Aurora, Oracle, Microsoft SQL Server, PostgreSQL, MySQL and MariaDB.

### Objectives

After completing this lab, you can:

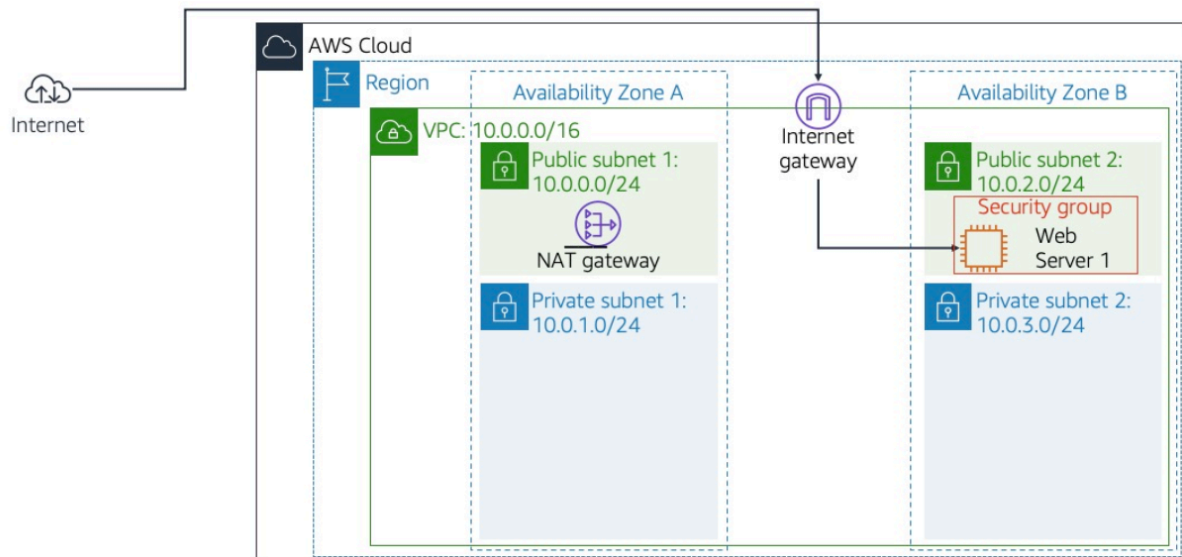
- Launch an Amazon RDS DB instance with high availability.
- Configure the DB instance to permit connections from your web server.
- Open a web application and interact with your database.

### Duration

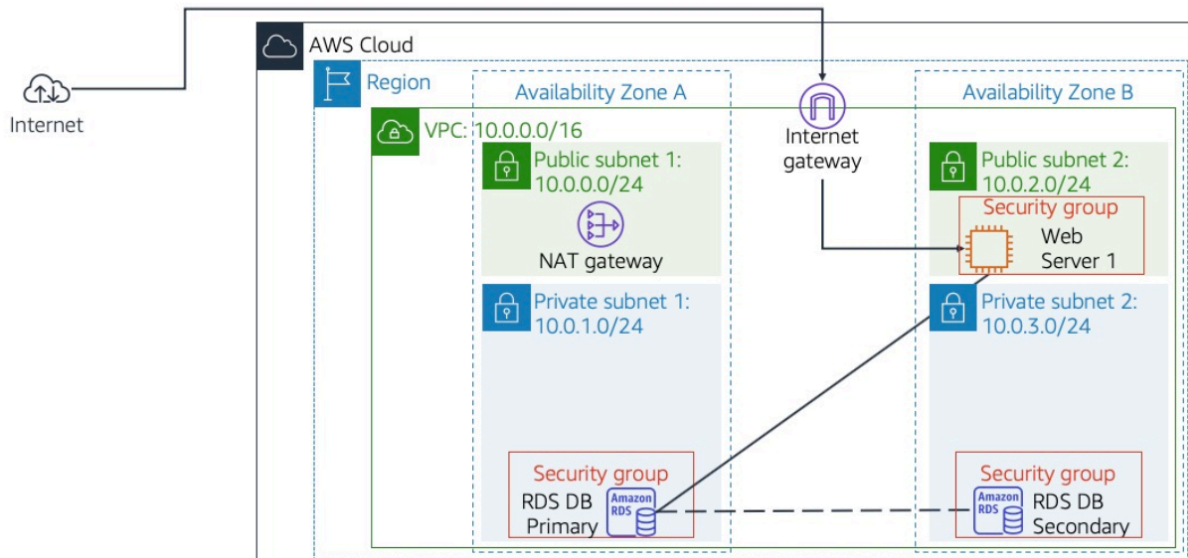
This lab takes approximately **45 minutes**.

### Scenario

You start with the following infrastructure:



At the end of the lab, this is the infrastructure:



## Accessing the AWS Management Console

1. At the top right corner of these instructions, choose ► **Start Lab**

**Troubleshooting tip:** If you get an "Access Denied" error, close the error box and choose ► **Start Lab** again.

2. The lab status can be interpreted as follows.

- A red circle next to **AWS** ● at the top left corner of this page indicates the lab has not been started.
- A yellow circle next to **AWS** ● at the top left corner of this page indicates the lab is starting.
- A green circle next to **AWS** ● at the top left corner of this page indicates the lab is ready.

Please wait for the lab to be ready, before proceeding.

3. At the top of these instructions, choose the green circle next to **AWS** ●

This will open the AWS Management Console in a new browser tab. The system will automatically log you in.

**Tip:** If a new browser tab does not open, there will typically be a banner or icon at the top of your browser indicating that your browser is preventing the site from opening pop-up windows. Click on the banner or icon and choose "Allow pop ups."

4. Arrange the AWS Management Console tab so that it displays along side these instructions. Ideally, you will be able to see both browser tabs at the same time, to make it easier to follow the lab steps.


**⚠ Do not change the lab region unless specifically instructed to do so.**

---

## 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.

5. In the AWS Management Console, select the  **Services** menu, and then select **VPC** under **Networking & Content Delivery**.
6. In the left navigation pane, click **Security Groups**.
7. 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*.

8. In the **Inbound rules** section, click **Add rule**, then configure:
  - **Type:** `MySQL/Aurora (3306)`
  - **Source:** Type `sg` in the search field 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*.

---

9. 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.

10. In the AWS Management Console, select the  **Services** menu, and then select **RDS** under **Database**.
11. In the left navigation pane, click **Subnet groups**.
  - ⚠ If the navigation pane is not visible, click the  menu icon in the top-left corner.
12. Click **Create DB Subnet Group** then configure:
  - **Name:** `DB Subnet Group`
  - **Description:** `DB Subnet Group`
  - **VPC ID:** `Lab VPC`
13. In the **Add subnets** section for *Availability zones*, click the ▼, then:
  - Select ☒ the first Availability zone
  - Select ☒ the second Availability zone
14. 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`

15. 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).

16. In the left navigation pane, click **Databases**.

17. Click **Create database**

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

18. Choose **Create database**, then choose **Standard create**.

19. Under the **Engine options** section, for **Engine type**, choose **MySQL**.

20. For **Engine version**, choose the latest version.

21. For **Templates**, choose **Dev/Test**.

22. For **Availability and durability**, choose **Multi-AZ DB Instance**.

23. Under **Settings**, configure the following:

- **DB instance identifier:** `lab-db`
- **Master username:** `main`
- **Master password:** `lab-password`

24. Under **Instance configuration**, configure the following for **DB instance class**:

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

25. Under **Storage**, configure:

- Select *General Purpose (SSD)* under **Storage type**.

26. Under **Connectivity**, configure:

- **Virtual Private Cloud (VPC):** `Lab VPC`

27. Under **VPC security group** select ☒ **Choose existing**

28. Under **Existing VPC security groups**

- Use X to Remove *default*.
- Select **DB Security Group** to highlight it in blue.

29. Under **Monitoring**, expand **Additional configuration** and then configure the following:

- For **Enhanced Monitoring**, uncheck ☐ **Enable Enhanced monitoring**.

30. Scroll down to the **Additional configuration** section and expand this option. Then configure:

- **Initial database name:** `lab`
- Under **Backup**, uncheck ☐ **Enable automated backups**.

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

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

Your database will now be launched.

32. 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 database in two different Availability zones.

**Note:** If you are prompted with the **Suggested add-ons for lab-db** window, choose Close

**i** While you are waiting, you might want to review the [Amazon RDS FAQs](#) or grab a cup of coffee.

33. Wait until the **Status** changes to **Modifying** or **Available**.

34. Scroll down to the **Connectivity & Security** section and copy the **Endpoint** field.

It will look similar to: *lab-db.cgqg8lhnxvnx.us-west-2.rds.amazonaws.com*

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

## Task 4: Interact with Your Database



In this task, you will open a web application running on your web server and configure it to use the database.

36. Copy the **WebServer IP address** by selecting **i AWS Details** above these instructions you are currently reading.

37. Open a new web browser tab, paste the *WebServer* IP address and press Enter.

The web application will be displayed, showing information about the EC2 instance.

38. At the top of the web application page, click the **RDS** link.

 Load Test RDS 

Meta-Data	Value
InstanceId	<i>i-07ae9f8b551a3d166</i>
Availability Zone	<i>us-west-2b</i>

*Figure: A picture displaying the web application interface.*

You will now configure the application to connect to your database.

39. Configure the following settings:

- **Endpoint:** Paste the Endpoint you copied to a text editor earlier
- **Database:** `lab`
- **Username:** `main`
- **Password:** `lab-password`
- Click **Submit**

A message will appear explaining that the application is running a command to copy information to the database. After a few seconds the application will display an **Address Book**.

The Address Book application is using the RDS database to store information.

40. Test the web application by adding, editing and removing contacts.

The data is being persisted to the database and is automatically replicating to the second Availability Zone.

## Lab Complete

---

🎉 Congratulations! You have completed the lab.

41. Choose **End Lab** at the top of this page, and then select **Yes** to confirm that you want to end the lab.

A panel indicates that *DELETE has been initiated...* You may close this message box now.

42. A message *Ended AWS Lab Successfully* is briefly displayed, indicating that the lab has ended.