aws.qwiklabs.com

Architecting on AWS - Lab 3 - Create a database layer in your Amazon VPC infrastructure | Qwiklabs

Qwiklabs

19-24 minutes

Note: Do not include any personal, identifying, or confidential information into the lab environment. Information entered may be visible to others.

Corrections, feedback, or other questions? Contact us at *AWS Training and Certification*.

Lab overview

A backend database plays an important role in any environment and the security and access control to this critical resource is vital to any architecture. In this lab, you create an Amazon Aurora DB cluster to manage a MySQL database and an

Application Load Balancer (ALB). The AWS Security Pillar recommends keeping people away from data; as such, the database will be segregated from the front-end using the ALB. The ALB will route traffic to healthy EC2 instances that will host the front-end application to provide high availability and allow communication to the database to happen behind the ALB, in a private subnet.

Objectives

After completing this lab, you should be able to:

- Create an Amazon RDS database
- Create an Application Load Balancer
- Create an HTTP listener for the Application Load Balancer
- Create a target group
- Register targets with a target group
- Test the load balancer and the application connectivity to the database
- Review the Amazon RDS DB instance metadata via the console

Prerequisites

This lab requires:

- Access to a notebook computer with Wi-Fi and Microsoft Windows, macOS, or Linux (Ubuntu, SuSE, or Red Hat)
- An internet browser such as Chrome, Firefox, or Microsoft Edge
- A plaintext editor

Duration

This lab requires approximately **45** minutes to complete.

AWS Services Not Used in This Lab

AWS services not used in this lab are disabled in the lab environment. In addition, the capabilities of the services used in this lab are limited to what the lab requires. Expect errors when accessing other services or performing actions beyond those provided in this lab guide.

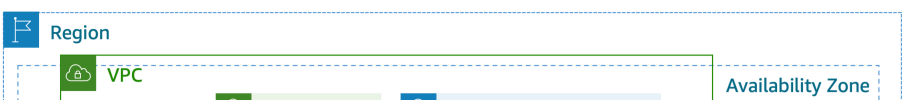
Scenario

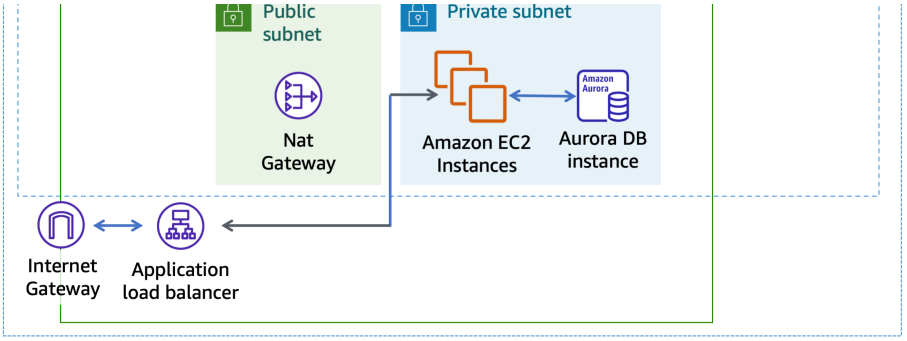
Your team has been tasked with prototyping an architecture for a new web-based application. To define your architecture, you need to have a better understanding of load balancers and managed databases such as Amazon RDS.

Lab Environment

The lab environment provides you with the following resources to get started: an Amazon Virtual Private Cloud (Amazon VPC), underlying necessary network structure, three Security groups to control inbound and outbound traffic, two Amazon EC2 instances in private subnet, and an associated Amazon EC2 instance profile. The instance profile contains the permissions necessary to allow the AWS Systems Manager Session Manager feature to access the Amazon EC2 instance.

The following diagram shows the expected architecture of the important lab resources you build and how they should be connected at the end of the lab.





Start Lab

1. At the top of your screen, launch your lab by choosing Start Lab

   This starts the process of provisioning your lab resources. An estimated amount of time to provision your lab resources is displayed. You must wait for your resources to be provisioned before continuing.

   If you are prompted for a token, use the one distributed to you (or credits you have purchased).

2. Open your lab by choosing Open Console

   This opens an AWS Management Console sign-in page.

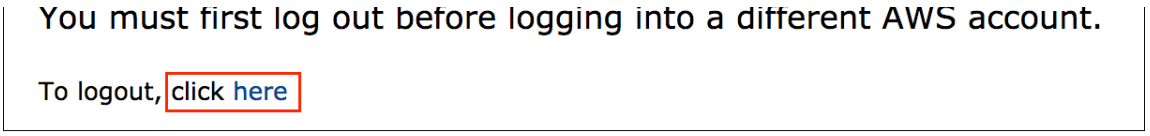
3. On the sign-in page, configure:

- **IAM user name:**
- **Password:** Paste the value of **Password** from the left side of the lab page
- Choose Sign In

  Do not change the Region unless instructed.

  Common Login Errors

  Error: You must first log out



If you see the message, **You must first log out before logging into a different AWS account:**

- Choose **here**
- Close your browser tab to return to your initial lab window
- Choose Open Console again

Task 1: Create an Amazon RDS database

In this task, you create an Amazon Aurora DB cluster that is compatible with MySQL. An Amazon Aurora DB cluster consists of one or more DB instances and a cluster volume that manages the data for those DB instances.

Amazon Aurora (Aurora) is a fully managed relational database engine that is compatible with MySQL and PostgreSQL. Aurora is part of the managed database service Amazon Relational Database Service (Amazon RDS). Amazon RDS is a web service that makes it easier to set up, operate, and scale a relational database in the cloud.

4. In the **AWS Management Console**, on the Services menu, choose **RDS**.

   Note: You can also locate a service in the AWS Management Console by searching for it by name in the unified search bar at the top-center of the page. The unified search bar is located to the right of the Services menu, and it is labeled:

   Search for services, features, marketplace products, and docs

5. In the left navigation pane, choose **Databases**.
6. Choose the Create database button.

7. In **Choose a database creation method**, choose *Standard Create*
8. In the **Engine options** section, configure:

- In **Engine type**, choose *Amazon Aurora*
- In **Edition**, choose *Amazon Aurora MySQL-Compatible Edition*
- In **Capacity type**, choose Provisioned

9. In **Templates**, choose *Dev/Test*
10. In the **Settings** section, configure:

- **DB cluster identifier:**
- **Master username:**
- **Master password:**
- **Confirm password:**

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

- **DB instance class:** choose *Burstable classes*
- *db.t3.small*

12. In the **Availability & durability** section, for **Multi-AZ deployment**, select *Don't create an Aurora Replica*

    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).

    Note Since this lab is all about knowing the resources required to build a multi-tier architecture, you do not need to perform a Multi-AZ deployment. You learn how to deploy a Multi-AZ architecture in the next lab.

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

- **Virtual Private Cloud (VPC):** *LabVPC*
- **Subnet group:** *labdbsubnetgroup*
- **Public access:** *No*
- **VPC security group:** *Choose existing*
- **Existing VPC security groups:**
- Remove the *default* security group
- Choose *LabDBSecurityGroup* from the drop-down menu

  Subnets are segments of an Amazon VPC's IP address range that you designate to group your resources based on security and operational needs. A DB subnet group is a collection of subnets (typically private) that you create in an Amazon VPC and then designate for your DB instances. A DB subnet group allows you to specify an Amazon VPC when creating DB instances using the CLI or API. If you use the console, you can just select the Amazon VPC and subnets you want to use.

  Amazon Virtual Private Cloud (Amazon VPC) enables you to launch AWS resources into a virtual network that you have defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

14. In the **Additional configuration** section:

- **Database port:** Leave the configuration at the default value.

15. Expand the **Additional configuration** main section at the end of the page.

16. In the **Database options** section, configure:

- **Initial database name:**

- **DB cluster parameter group:** select the value from the drop-down menu that matches to the **DBClusterParameterGroup** value from the left side of the page

  Caution Ensure the correct value for **DB cluster parameter group** is selected from the drop-down menu. An incorrect value results in errors when building the database replicas.

17. In the **Encryption** section, de-select **Enable encryption**.

    You can encrypt your Amazon RDS instances and snapshots at rest by enabling the encryption option for your Amazon RDS DB instance. Data that is encrypted at rest includes the underlying storage for a DB instance, its automated backups, Read Replicas, and snapshots.

18. In the **Monitoring** section, de-select **Enable Enhanced monitoring**

19. In the **Maintenance** section, de-select **Enable auto minor version upgrade**.

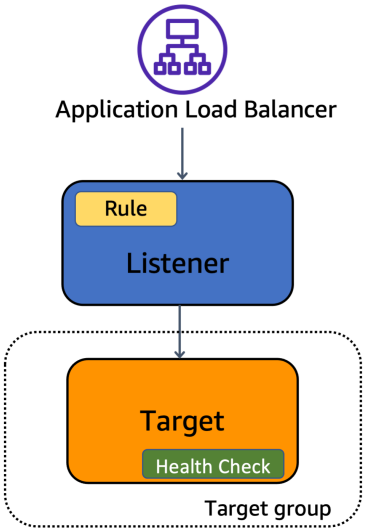
20. Scroll to the bottom of the screen, then choose the Create database button.

    Your Aurora MySQL DB cluster is in the process of launching. The Amazon RDS database can take up to 5 minutes to launch. However, you can continue to the next task.

Task 2: Create and configure an Application Load Balancer

In this task, you create an Application Load Balancer in the public subnets to access the application from a browser. You navigate to the Amazon EC2 console and create an Application Load Balancer into the existing Amazon VPC infrastructure and add the private Amazon EC2 instances as a target.

A load balancer serves as the single point of contact for clients. Clients send requests to the load balancer, and the load balancer sends them to targets, such as EC2 instances. To configure your load balancer, you create target groups, and then register targets with your target groups.



Task 2.1 : Create a target group

In this task, you create a target group and register your targets with the target group. By default, the load balancer sends requests to registered targets using the port and protocol that you specified for the target group.

21. In the **AWS Management Console**, on the Services menu, choose **EC2**.
22. In the left navigation pane, choose **Target groups**.
23. Choose Create target group button.

    The **Specify group details** page is displayed.
24. In the **Basic configuration** section, configure the following:

- For **Choose a target type**: Select *Instances*
- For **Target group name**: Enter

- For **VPC**: Select *LabVPC*

  The remaining settings on the page can be left at their default values.

25. Choose the Next button.

    The **Register targets** page is displayed.

26. In the **Available Instances** section, configure the following:

- Select the Amazon EC2 instance named **AppServer1** and **AppServer2**
- Choose the **Include as pending below** button.

  The instance appears under the **Targets** section of the page.

27. Choose Create target group button.
28. A message like the following is displayed:

- Successfully created target group: ALBTargetGroup

  Task 2.2 : Create an Application Load Balancer

  In this task, you create an Application Load Balancer and to do that you must first provide basic configuration information for your load balancer, such as a name, scheme, and IP address type. Then, you provide information about your network, and one or more listeners.

29. In the left navigation pane, choose **Load Balancers**.
30. Choose the **Create Load Balancer** button.
31. Choose the **Create** button in the **Application Load Balancer** section.

    The **Create Application Load Balancer** page is displayed.

32. In the **Basic Configuration** section, configure the following:

- For **Load balancer name**: Enter

33. In the **Network mapping** section, configure the following:
    - **VPC**: Select
    - **Mappings**:
    - Select the check box for the first Availability Zone listed, and select from the Subnet list.
    - Select the check box for the second Availability Zone listed, and select from the Subnet list.
34. In the **Security groups** section, configure:
    - Remove the *default* security group
    - Select *LabALBSecurityGroup* from the drop-down menu
35. In the **Listeners and routing** section, configure the following:
    - For **Listener HTTP:80**: Select *ALBTargetGroup* from the Default action drop-down list
36. Choose the Create load balancer button.
37. A message like the following is displayed:
    - Successfully created load balancer:LabAppALB

    Load balancer **LabAppALB** was successfully created
38. Choose the **View load balancers** button.

    The load balancer is in the *provisioning* state for few minutes and then changes to *active*.

    In this task, you created an application load balancer and you added Amazon EC2 instances as a target to the load balancer. This task provides a demonstration on how to register a target with a load balancer. In addition to individual Amazon EC2 instances, Auto Scaling Groups can also be registered as

targets for the load balancer. When you use Auto Scaling Groups as targets for load balancing, the instances that are launched by the Auto Scaling group are automatically registered with the load balancer. Likewise, Amazon EC2 instances that are terminated by the Auto Scaling Groups are automatically de-registered from the load balancer. Using Auto Scaling Groups with a load balancer is demonstrated in the *next lab*.

You have successfully created a load balancer, created target groups, and registered the EC2 instances with the target group.

Task 3: Review the Amazon RDS DB instance metadata via the console

In this task, we navigate through the Amazon RDS console to ensure the instance created in Task 1 has completed and is active. You explore the AWS Management Console to learn how to find the connection information for a DB instance. The connection information for a DB instance includes its endpoint, port, and a valid database user.

39. In the **AWS Management Console**, on the Services menu, choose **RDS**.
40. In the navigation pane, choose **Databases**.
41. Select the link for the cluster named **aurora** from the list of DB identifiers.

    A page with details about the database are displayed.
42. On the **Connectivity & security** tab, you can find the endpoint, and port number for the database cluster. In general, you need the endpoints and the port number to connect to the database.
43. Copy and paste the **Endpoint** of the **writer** instance value to a notepad, you need this value later in the lab.

It should look similar to:
aurora.crwxbgqad61a.rds.amazonaws.com

Tip To copy the **writer** instance *Endpoint*, hover on it and select the copy icon.

Notice that the status for the endpoints is **Available**.

44. On the **Configuration** tab, you can find details regarding how the database is currently configured.
45. On the **Monitoring** tab, you can monitor metrics for the following items of your database:

- The number of connections to a database instance.
- The amount of read and write operations to a database instance.
- The amount of storage that a database instance is currently using.
- The amount of memory and CPU being used for a database instance.
- The amount of network traffic to and from a database instance.

Task 4: Test the application connectivity to the database

In this task, you identify the Application Load Balancer URL and run a simple HTTP request through the load balancer to launch the web application installed on the Amazon EC2 instances and test the application connectivity to the database.

46. In the **AWS Management Console**, on the Services menu, choose **EC2**.
47. In the left navigation pane, choose **Target Groups**.

48. Choose **ALBTargetGroup**.
49. In the **Targets** tab, wait until the instance status is displayed as healthy.

    The ELB periodically tests the ping path on your web server instance to determine health: a 200 HTTP response code indicates a healthy status, and any other response code indicates an unhealthy status. If an instance is unhealthy and continues in that state for a successive number of checks (unhealthy threshold), the load balancer removes it from service until it recovers.
50. In the left navigation pane, choose **Load Balancers**.
51. Select the check box for the load balancer named **LabAppALB**.
52. In the **Description** tab, copy the **DNS name** and paste the value in a new browser tab to invoke the load balancer.

    Tip To copy the *DNS name*, hover on it and select the copy icon.

    A web page like this is displayed:

53. Choose the **Settings** tab then configure:

- **Endpoint:** Paste the *writer instance endpoint* you copied earlier

- **Database:**
- **Username:**
- **Password:**

54. Choose the **Save** button.

    The application connects to the database, loads some initial data, and displays information. This application allows you to add, edit, or delete an item from a store's inventory.

    The inventory information is stored in the Amazon RDS MySQL compatible database you created earlier in the lab. This means that, if the web application server fails, the data will not be lost. It also means that multiple application servers could access the same data.

    You have successfully accessed the web application installed on the Amazon EC2 instance through the load balancer.

Optional Task: Create an Amazon RDS read replica in a different AWS Region

In this challenge task, you create a cross-region read replica from the source DB instance. You create a read replica in a different AWS Region to improve your disaster recovery capabilities, scale read operations into an AWS Region closer to your users, and to make it easier to migrate from a data center in one AWS Region to a data center in another AWS Region.

Note This challenge task is **optional** and is provided in case you have lab time remaining. You may complete this task or skip to the end of the lab here.

55. In the **AWS Management Console**, on the Services menu, choose **RDS**.

56. In the left navigation pane, choose **Databases**.
57. Choose **aurora** DB instance as the source for a read replica.
58. Choose the Actions button and select **Create cross-region read replica**.
59. In **Instance specifications** section, configure:
    - For **Multi-AZ deployment**, select No
    - The remaining settings in this section can be left at their default values.
60. In **Network & Security** section, configure:
    - For **Destination region:** select the region that matches to **RemoteRegion** value from the lab instructions.
    - For **Publicly accessible:** *No*
    - For **VPC security groups:**
    - Remove the *default* security group
    - Select *LabDBSecurityGroup* from the drop-down menu
    - The remaining settings in this section can be left at their default values.
61. In **Settings** section, configure:
    - For DB instance identifier, input
    - The remaining settings in this section can be left at their default values.
62. In **Performance Insights** section, configure:
    - select *Disable Performance Insights*
    - The remaining settings in this section can be left at their default values.

    Note The remaining settings on this page can be left at their

default values.

63. Choose the Create button.

    A message is displayed on the page with the message:

    Your Read Replica creation has been initiated.

64. Choose the hyperlink on the same page labelled *here* to review the cross region read replica in the destination region.

65. Otherwise, choose the Close button.

    You have successfully completed the optional task and started the creation of cross-region replica for the Amazon RDS database.

Conclusion

In this lab, you learned how to:

- Create an Amazon RDS DB instance
- Create an Application Load Balancer
- Create an HTTP listener for your Application Load Balancer
- Create a target group
- Register targets with a target group
- Test the load balancer and the application connectivity to the database
- Review the Amazon RDS instance metadata via the console

In this lab, you learned how to deploy various resources needed for a prototype web application in your Amazon VPC. However, the architecture that was created in this lab does not meet AWS Cloud best practices, because it is not an elastic, durable, highly available design. By relying on only a single Availability Zone in the architecture, there is a single point of failure. You learn how-

to configure your architecture for redundancy, failover and high availability in the next lab.

Lab Complete

Congratulations! You have completed the lab.

End Lab

Follow these steps to close the console, end your lab, and evaluate the experience.

66. Return to the AWS Management Console.
67. On the navigation bar, choose **awsstudent@<AccountNumber>**, and then choose **Sign Out**.
68. Choose End Lab
69. Choose OK
70. (Optional):
    - Select the applicable number of stars
    - Type a comment
    - Choose **Submit**
    - 1 star = Very dissatisfied
    - 2 stars = Dissatisfied
    - 3 stars = Neutral
    - 4 stars = Satisfied
    - 5 stars = Very satisfied

    You may close the window if you don't want to provide feedback.

    For more information about AWS Training and Certification, see *https://aws.amazon.com/training/*.

Your feedback is welcome and appreciated.

If you would like to share any feedback, suggestions, or corrections, please provide the details in our *AWS Training and Certification Contact Form*.