Lab 3: Creating a Database Layer in Your Amazon VPC Infrastructure

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 database (DB) cluster to manage a MySQL database and an Application Load Balancer (ALB). The Amazon Web Services (AWS) Security pillar of the Well-Architected Framework recommends keeping people away from data; as such, the database is separated from the front end using the Application Load Balancer. The Application Load Balancer routes traffic to healthy Amazon Elastic Compute Cloud (Amazon EC2) instances that hosts the front-end application. This provides high availability and allow communication to the database to happen behind the Application Load Balancer in a private subnet.

Objectives

By the end of this lab, you will be able to do the following:

- Create an Amazon Relational Database Service (Amazon RDS) database instance.
- 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 using the console.
- Optional Task: Create an Amazon RDS read replica in a different AWS Region.

Prerequisites

This lab requires the following:

- 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

Start lab

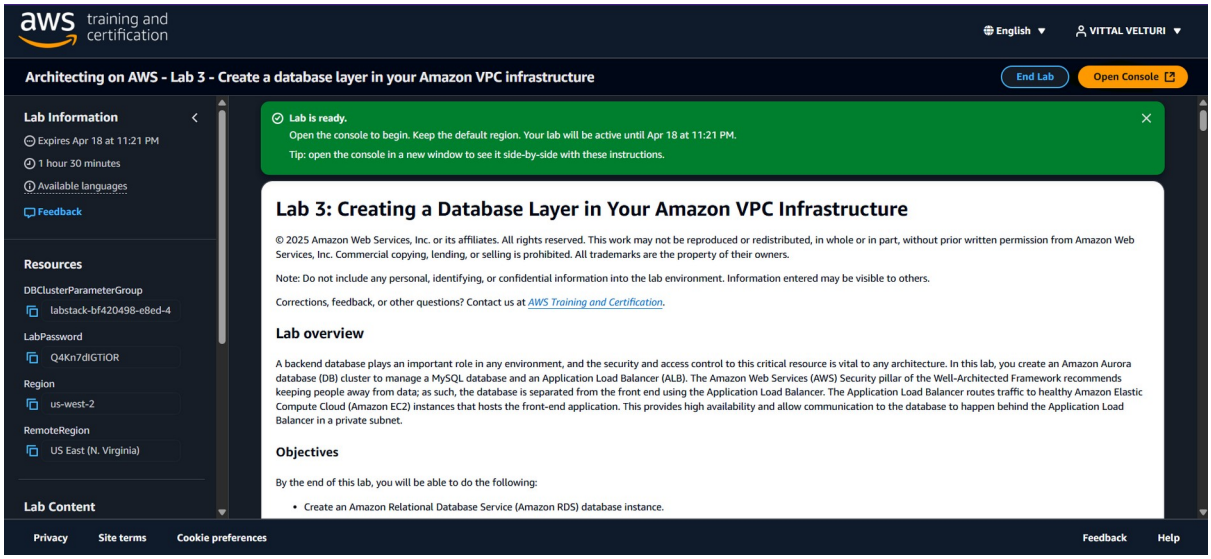
1. To launch the lab, at the top of the page, choose **Start Lab**.

   Caution: You must wait for the provisioned AWS services to be ready before you can continue.

2. To open the lab, choose **Open Console**.

   You are automatically signed in to the AWS Management Console in a new web browser tab.

Warning: Do not change the **Region** unless instructed.



Common sign-in errors

Error: You must first sign out

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

- Choose the **click here** link.
- Close your **Amazon Web Services Sign In** web browser tab and return to your initial lab page.
- Choose **Open Console** again.

Error: Choosing Start Lab has no effect

In some cases, certain pop-up or script blocker web browser extensions might prevent the **Start Lab** button from working as intended. If you experience an issue starting the lab:

- Add the lab domain name to your pop-up or script blocker's allow list or turn it off.
- Refresh the page and try again.

Resources

DBClusterParameterGroup: labstack-bf420498-e8ed-48d3-8727-cd8a9792f173-9qaiwkxz56u4hbnz8cqrbo-0-rdsdbclusterparametergroup-opdcurm5znhd

LabPassword: 85A4XUadNq4I

Region: us-west-2

RemoteRegion: US East (N. Virginia)

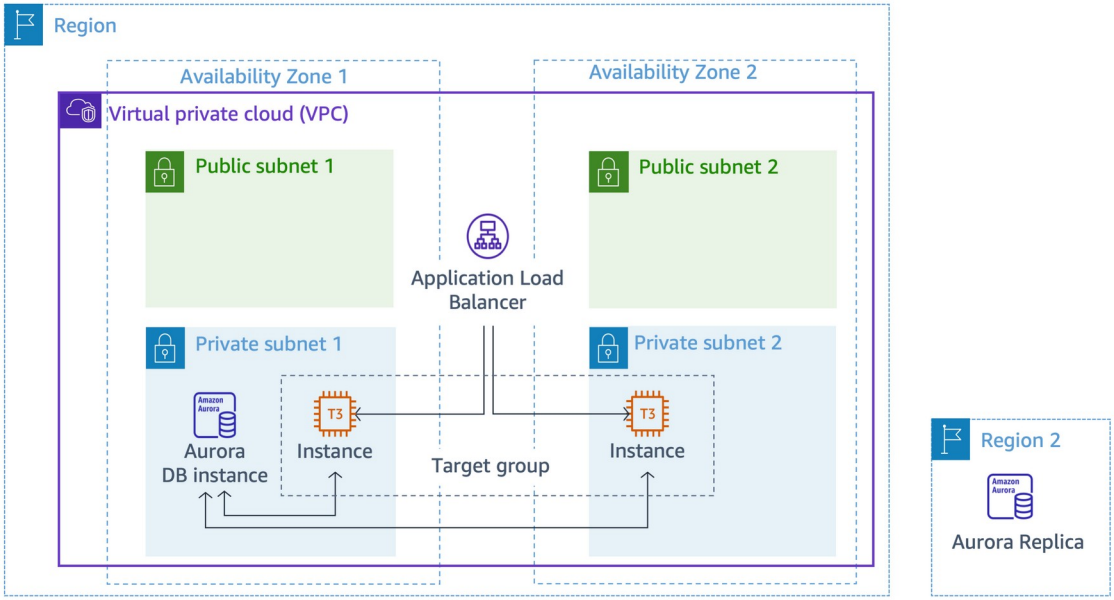
Scenario

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 EC2 instances in a private subnet, and an associated EC2 instance profile. The instance profile contains the permissions necessary to allow the AWS Systems Manager Session Manager feature to access the 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.



AWS services not used in this lab

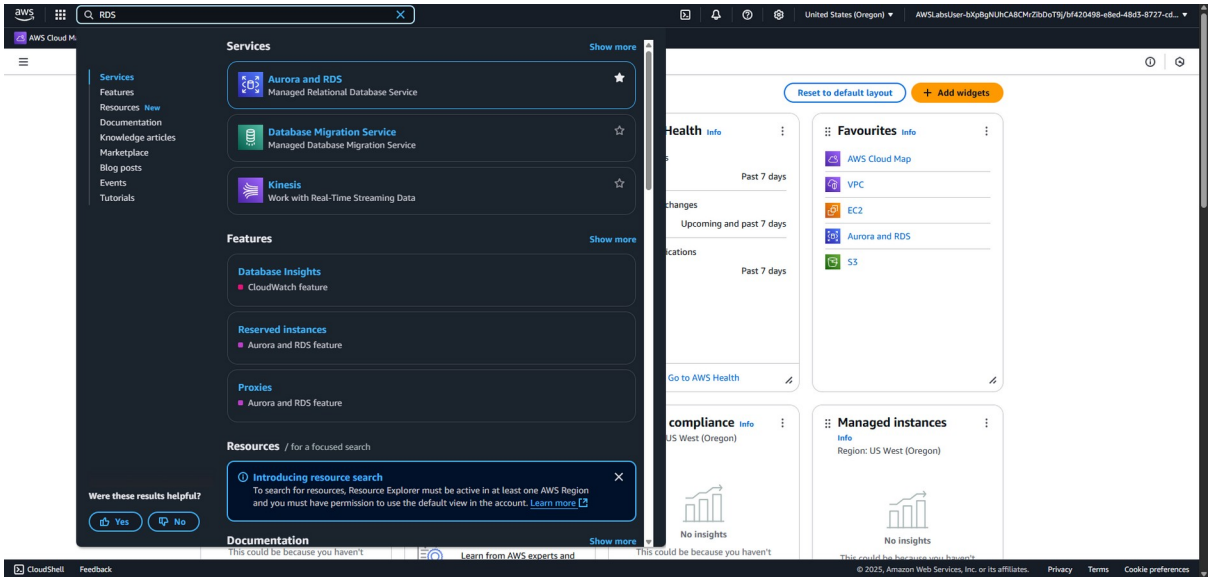
AWS services not used in this lab are turned off in the lab environment. In addition, the capabilities of the services used in this lab are limited to only what the lab requires. Expect to receive errors when accessing other services or performing actions beyond those provided in this lab guide.

Task 1: Create an Amazon RDS database

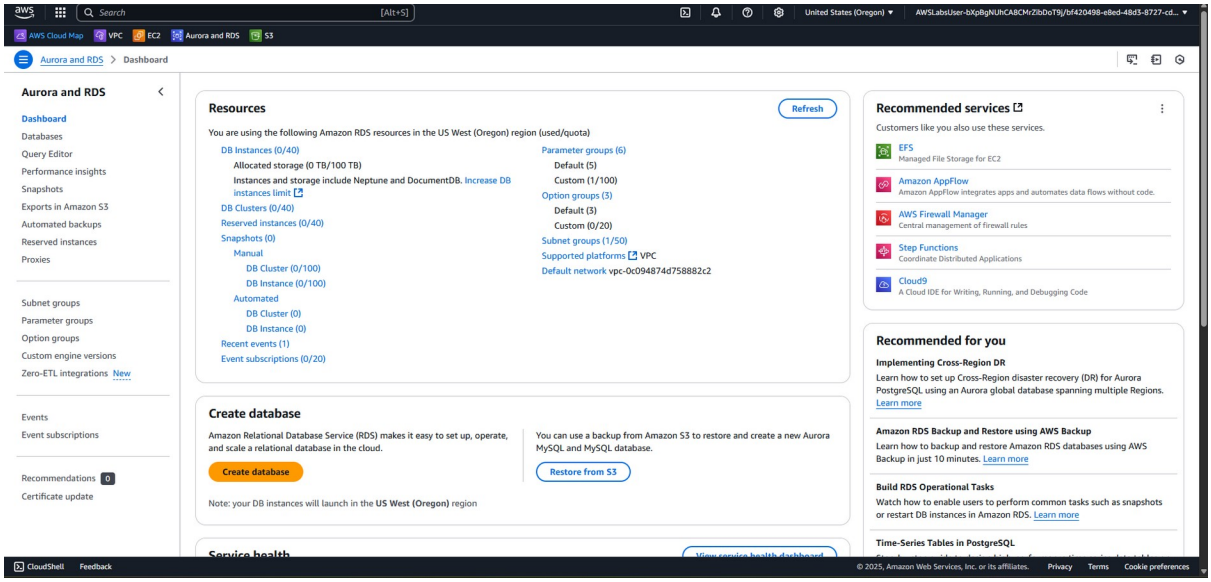
In this task, you create an Aurora DB cluster that is compatible with MySQL. An Aurora DB cluster consists of one or more DB instances and a cluster volume that manages the data for those DB instances.

Learn more: Amazon Aurora is a fully managed relational database engine that is compatible with MySQL and PostgreSQL. Aurora is part of the managed 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. For more information, see What is Amazon Aurora?.

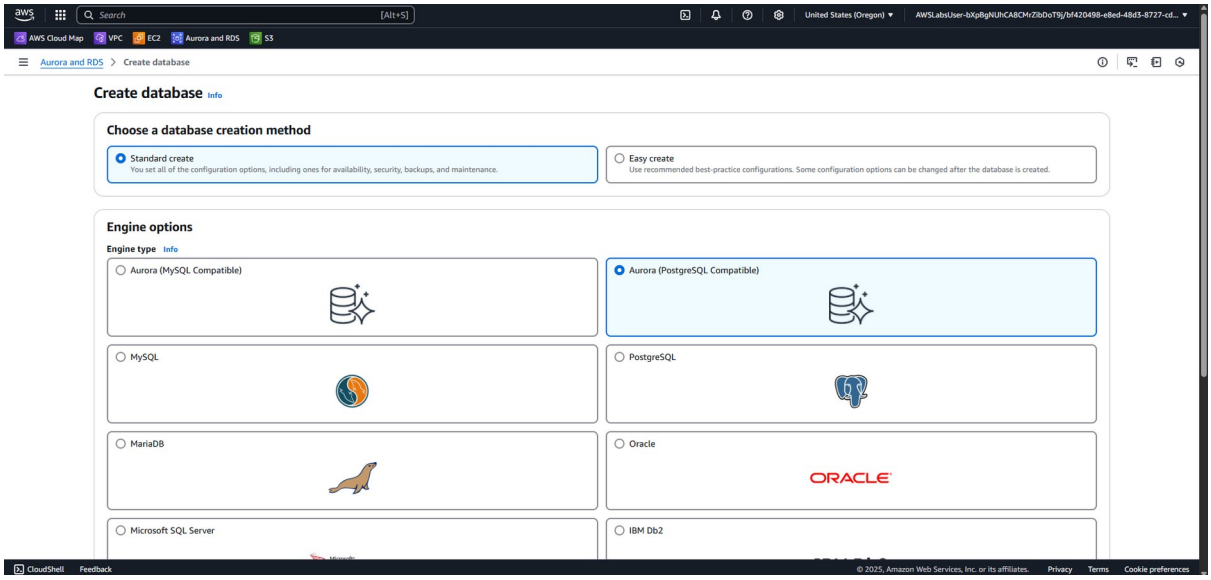
3. At the top of the AWS Management Console, in the search bar, search for and choose RDS.



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

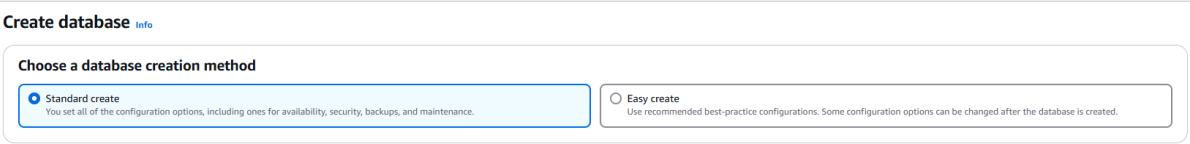


5. Choose **Create database**.



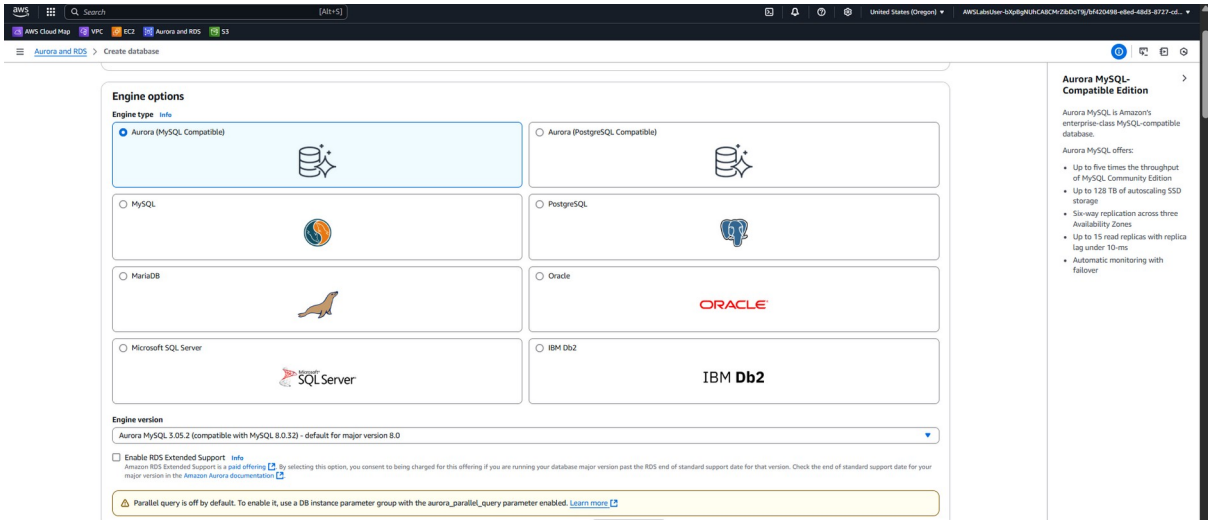
The **Create database** page is displayed.

6. In the **Choose a database creation method** section, select **Standard create**.

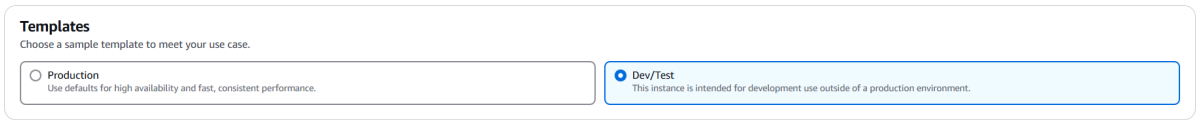


7. In the **Engine options** section, configure the following:

   - **Engine type:** Select **Aurora (MySQL Compatible)**.

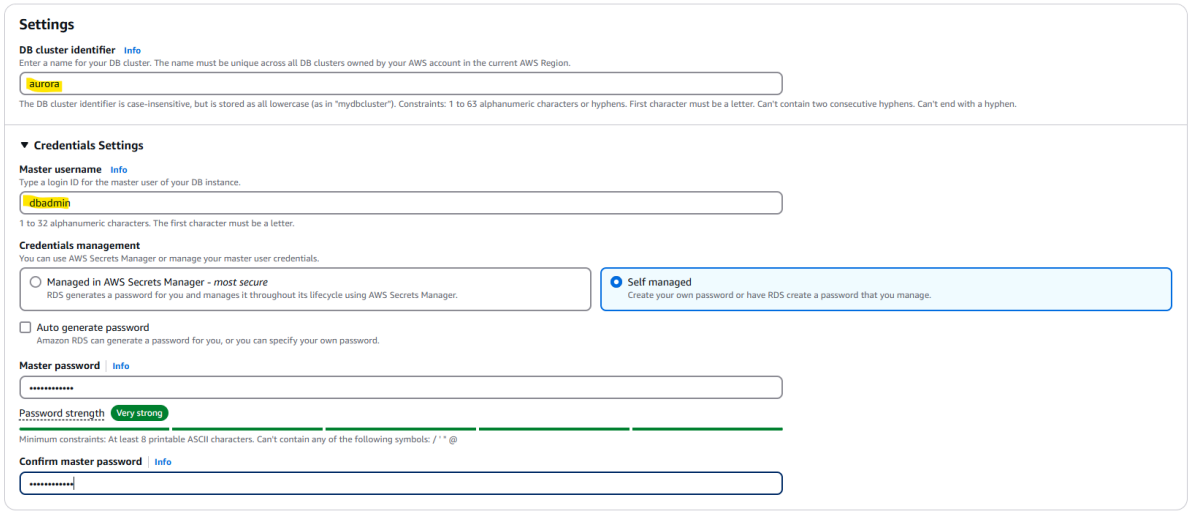


8. In the **Templates** section, select **Dev/Test**.



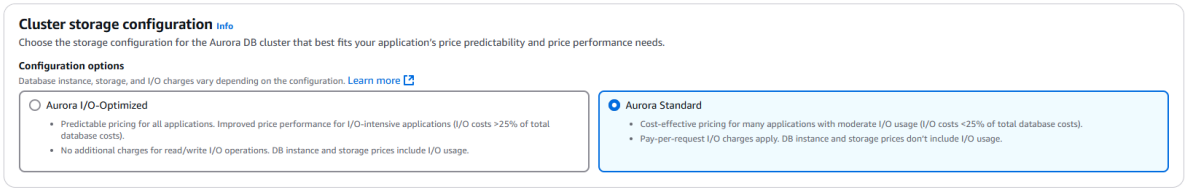
9. In the **Settings** section, configure the following:

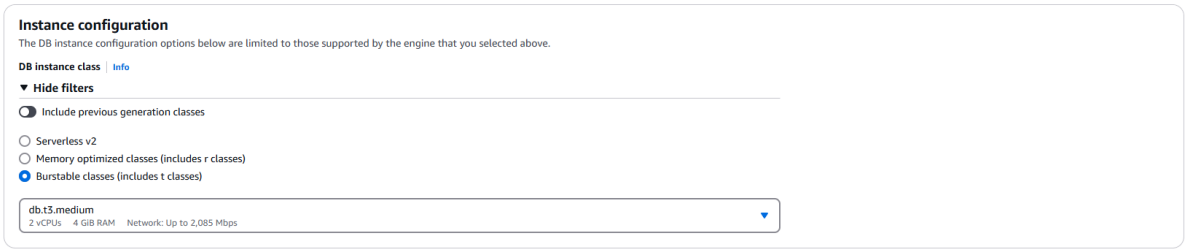
    - **DB cluster identifier:** Enter aurora.
    - **Master username:** Enter dbadmin.
    - Under **Credentials management** select **Self managed**
        - **Master password**: Paste the **LabPassword (85A4XUadNq4I)** value from the left side of these lab instructions.
        - **Confirm master password**: Paste the **LabPassword (85A4XUadNq4I)** value from the left side of these lab instructions.



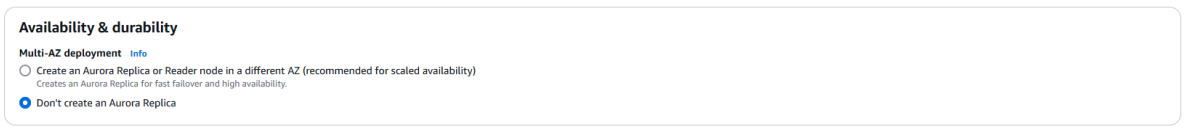
10. In the **Instance configuration** section, configure the following:

    - **DB instance class:** Select **Burstable classes (includes t classes)**.
    - From the dropdown menu, choose the **db.t3.medium** instance type.





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

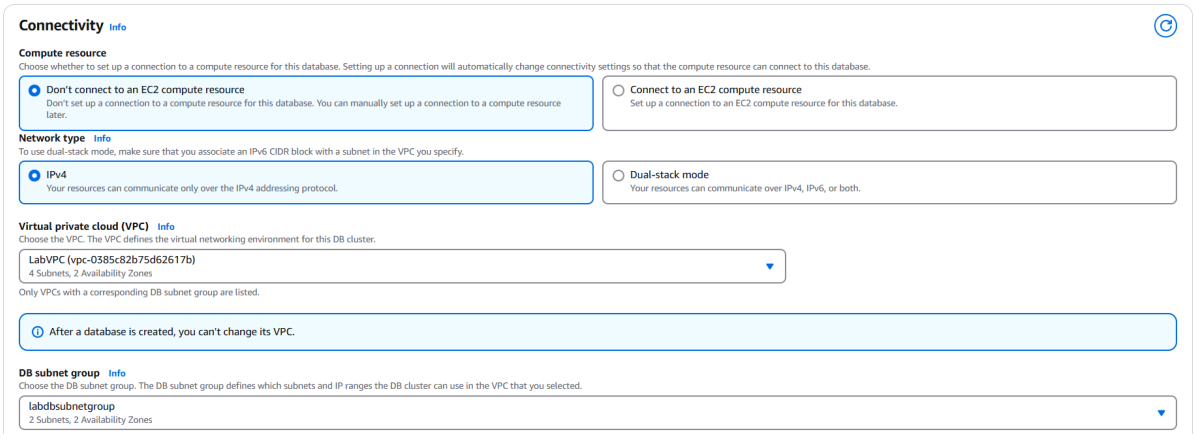


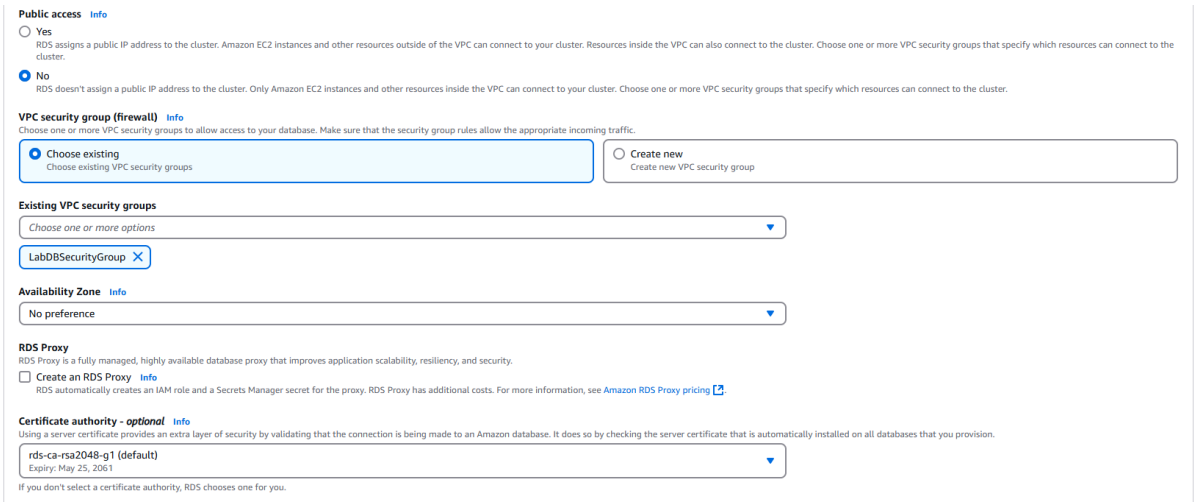
Learn more: AWS recommends that you distribute the primary instance and reader instances in your DB cluster over multiple Availability Zones to improve the availability of your DB cluster. That way, an issue that affects an entire Availability Zone doesn't cause an outage for your cluster. You can set up a Multi-AZ DB cluster by making a simple choice when you create the cluster. You can also convert an existing Aurora DB cluster into a Multi-AZ DB cluster by adding a new reader DB instance and specifying a different Availability Zone. For more information see High availability for Amazon Aurora.

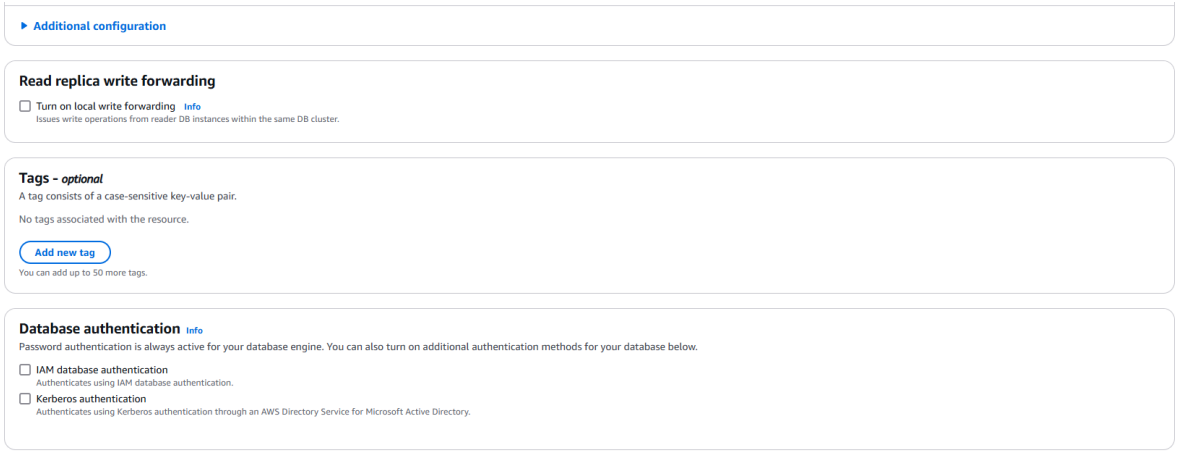
Note: Since this lab is about knowing the resources required to build a multi-tier architecture, you do not need to perform a Multi-AZ deployment. You learn how to deploy a Multi-AZ architecture in the next lab.

12. In the **Connectivity** section, configure the following:

    - **Virtual private cloud (VPC):** Select **LabVPC** from the dropdown menu.
    - **DB subnet group:** Select **labdbsubnetgroup** from the dropdown menu.
    - **Public access:** Select **No**.
    - **VPC security group (firewall):** Select **Choose existing**.
    - **Existing VPC security groups:**
        - To remove the **default** security group from the **Existing VPC security groups** field, select the **X**.
        - In the **Existing VPC security groups** dropdown menu, enter LabDBSecurityGroup to choose this option.



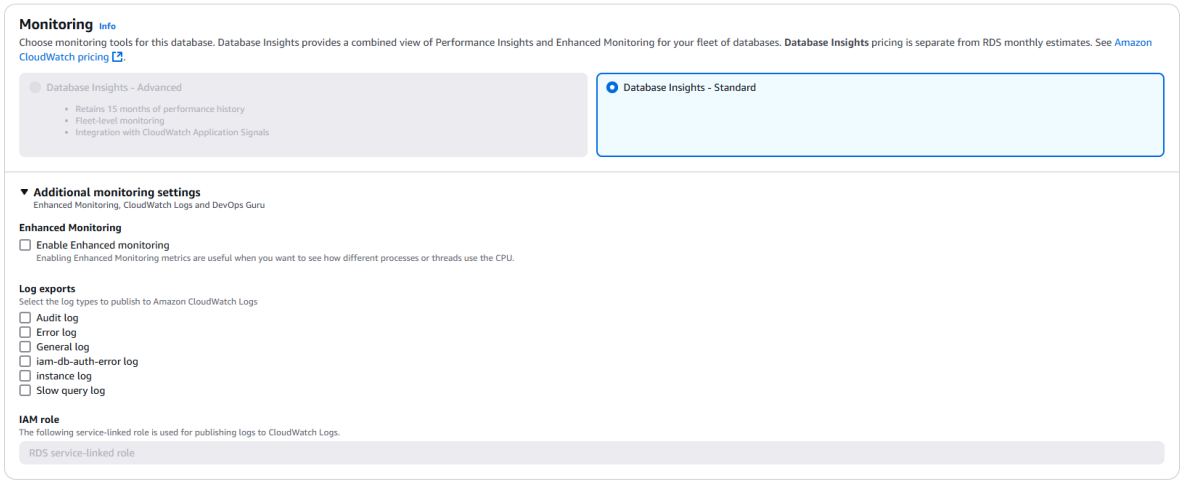




Learn more: *Subnets* are segments of an IP address range in an Amazon VPC that you designate to group your resources based on security and operational needs. A DB subnet group is a collection of subnets (typically private) that you create in an Amazon VPC and then designate for your DB instances. With a DB subnet group, you can specify an Amazon VPC when creating DB instances using the command line interface or API. If you use the console, you can just select the Amazon VPC and subnets you want to use. For more information, see Working with DB subnet groups.

Learn more: With Amazon VPC, you can 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. For more information, see Amazon VPC VPCs and Amazon RDS.

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

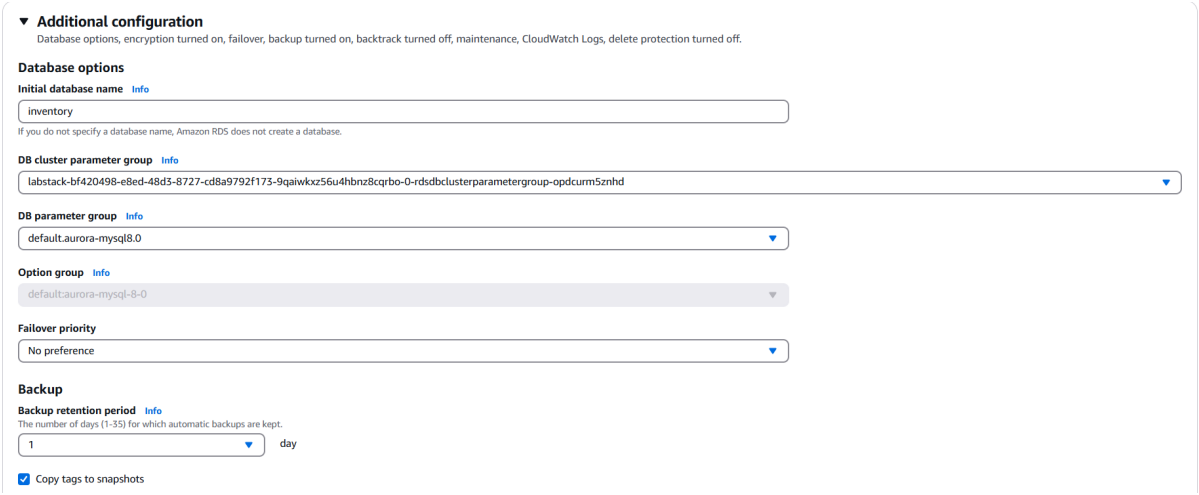


14. Expand the **Additional configuration** main section at the end of the page.

15. In the **Database options** section, configure the following:

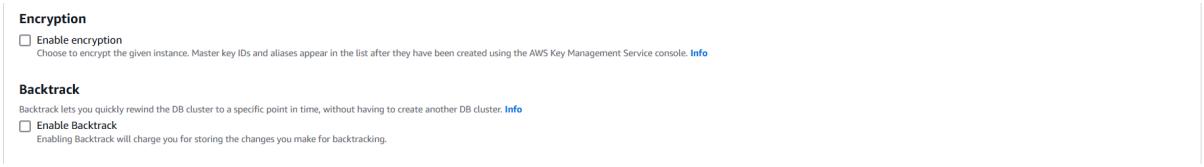
    - **Initial database name:** Enter inventory

DB cluster parameter group: Choose the value from the dropdown menu that matches the **DBClusterParameterGroup (labstack-bf420498-e8ed-48d3-8727-cd8a9792f173-9qaiwkxz56u4hbnz8cqrbo-0-rdsdbclusterparametergroup-opdcurm5znhd)** value from the left side of this page.



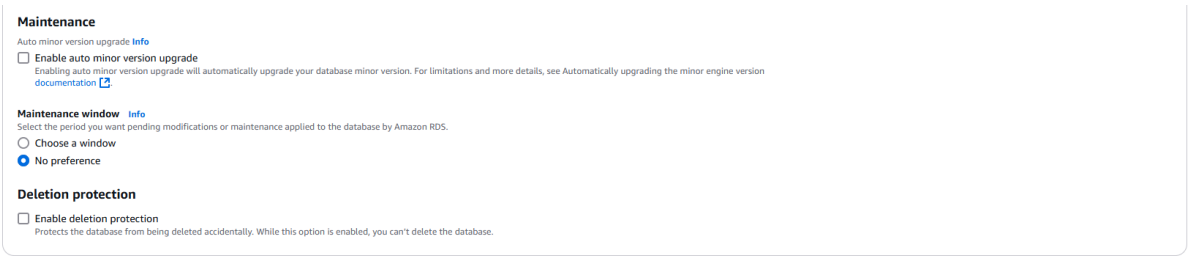
Caution: Ensure the correct value for **DB cluster parameter group** is selected from the dropdown menu. An incorrect value results in errors when building the database replicas.

16. In the **Encryption** section, unselect **Enable encryption**.



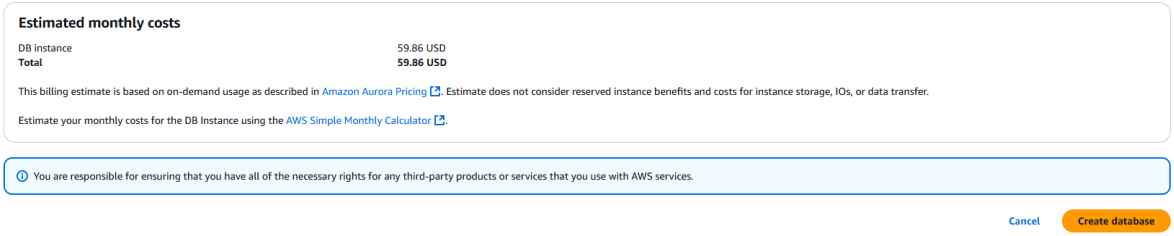
Learn more: You can encrypt your Amazon RDS instances and snapshots at rest by activating the encryption option for your Amazon RDS DB instance. Data that is encrypted at rest includes the underlying storage for a DB instance, its automated backups, read replicas, and snapshots. For more information, see Encrypting Amazon RDS resources.

17. In the **Maintenance** section, unselect **Enable auto minor version upgrade**.

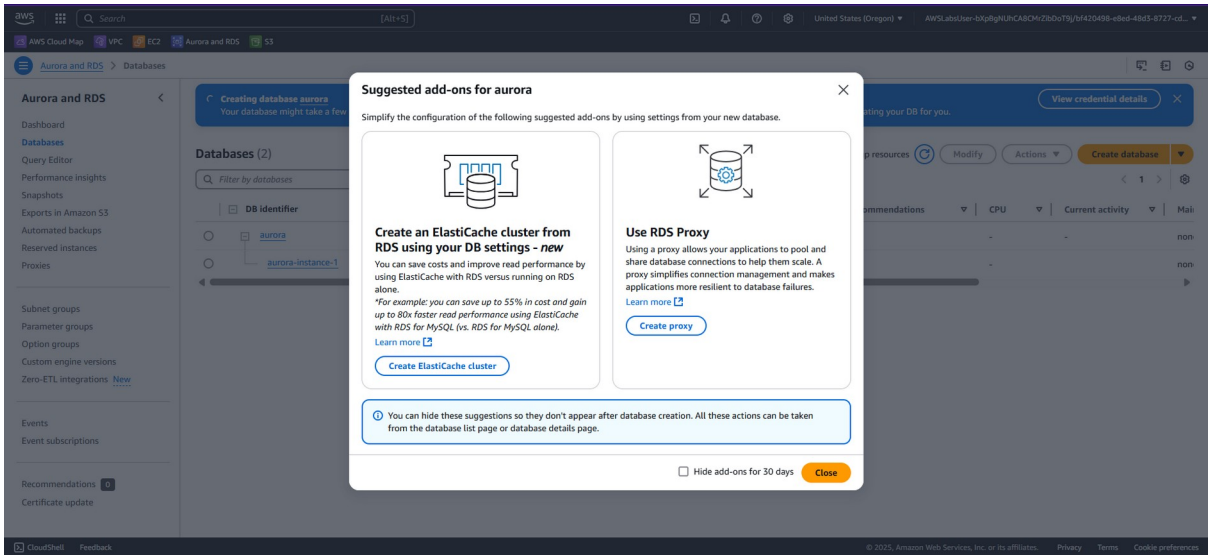


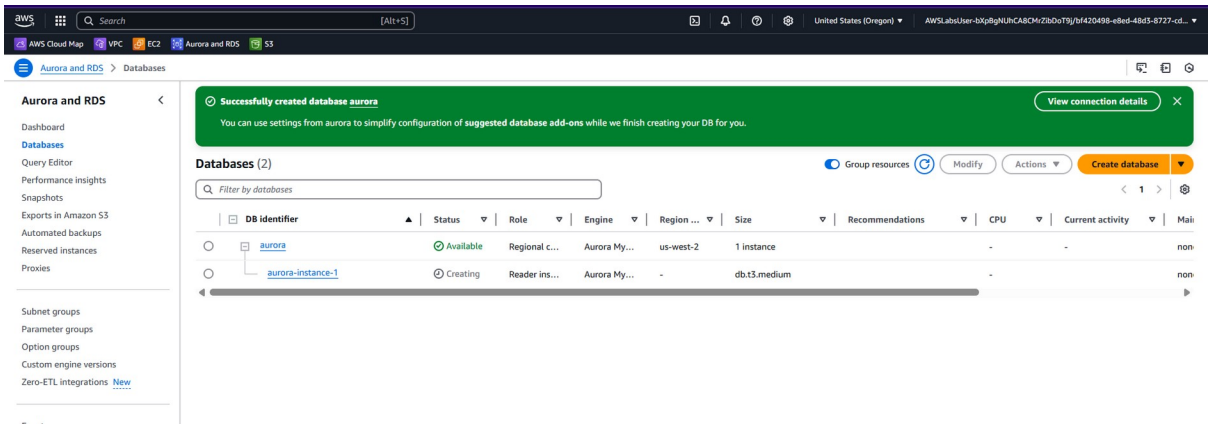
Note: Because the nature of this lab is short lived there is no need to set up a maintenance schedule for the database.

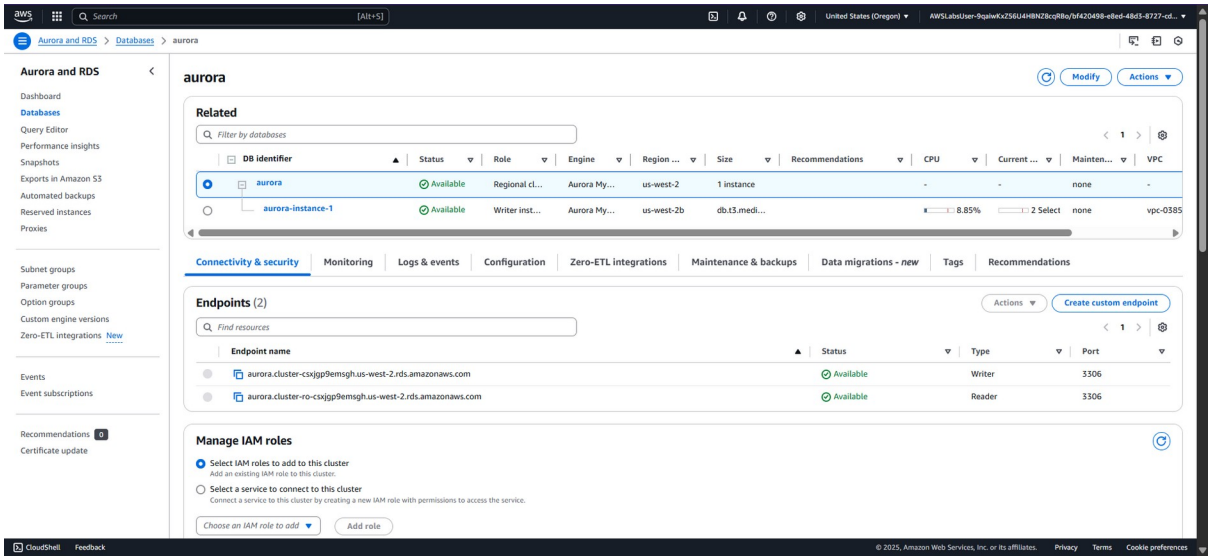
18. Scroll to the bottom of the screen, then choose **Create database**.



19. On the **Suggested add-ons for aurora** pop-up window, choose **Close**.







A **(Successfully created database aurora)** message is displayed on top of the screen.

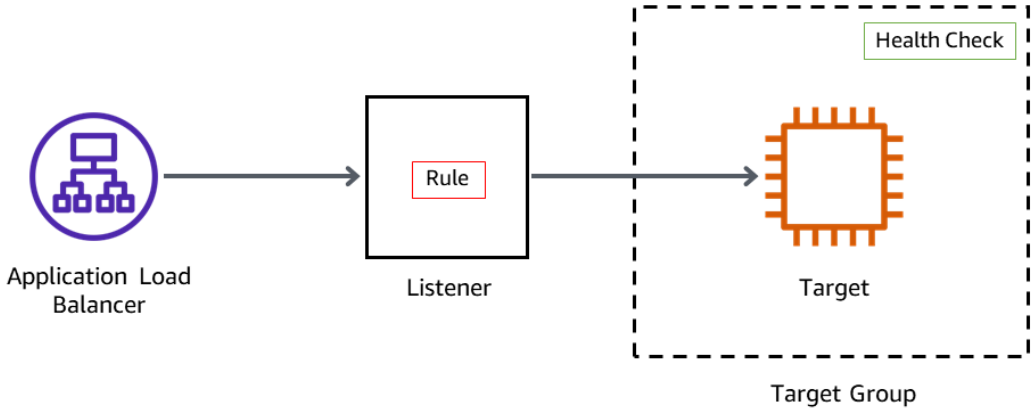
Your Aurora MySQL DB cluster is in the process of launching. The Amazon RDS database can take up to 5 minutes to launch. However, you can continue to the next task.

Congratulations! You have successfully created an Amazon RDS database.

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 EC2 instances as a target.

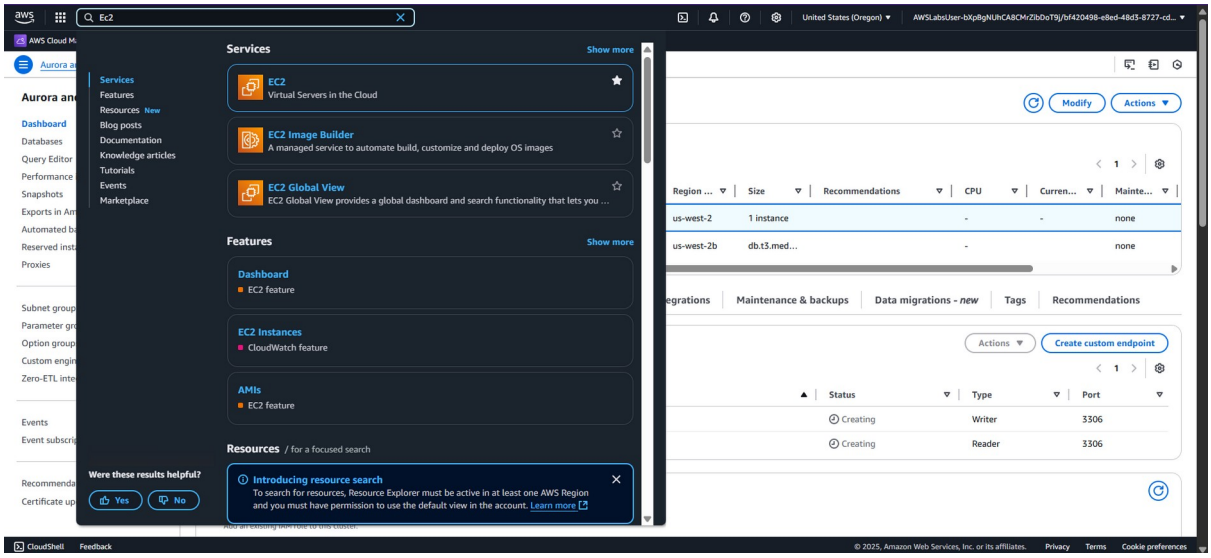
A load balancer serves as the single point of contact for clients. Clients send requests to the load balancer, and the load balancer sends them to targets, such as EC2 instances. To configure your load balancer, you create target groups and then register targets with your target groups.



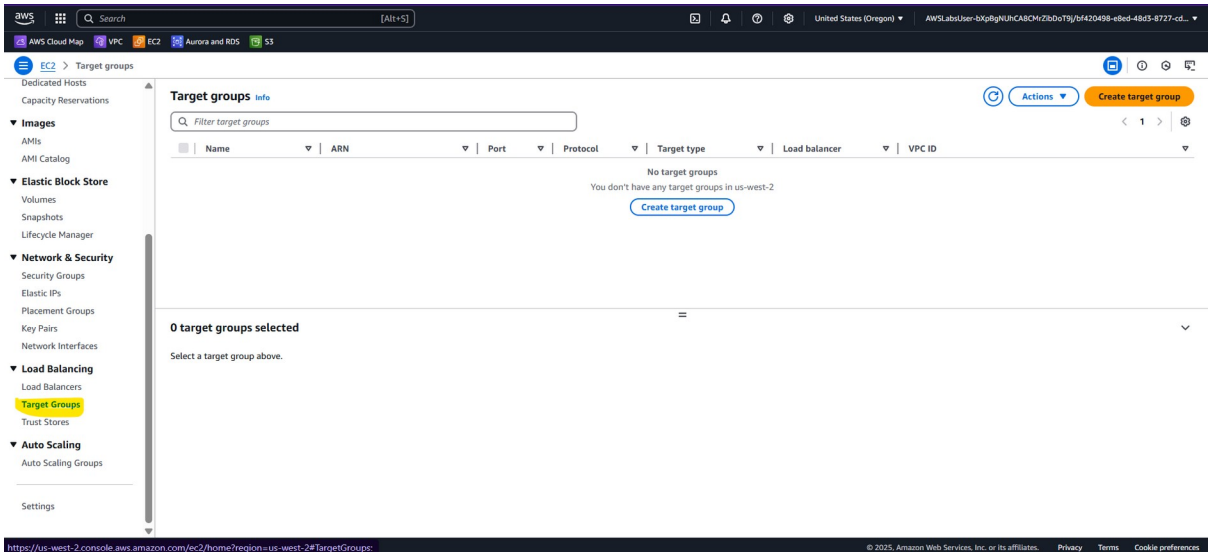
Task 2.1: Create a target group

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.

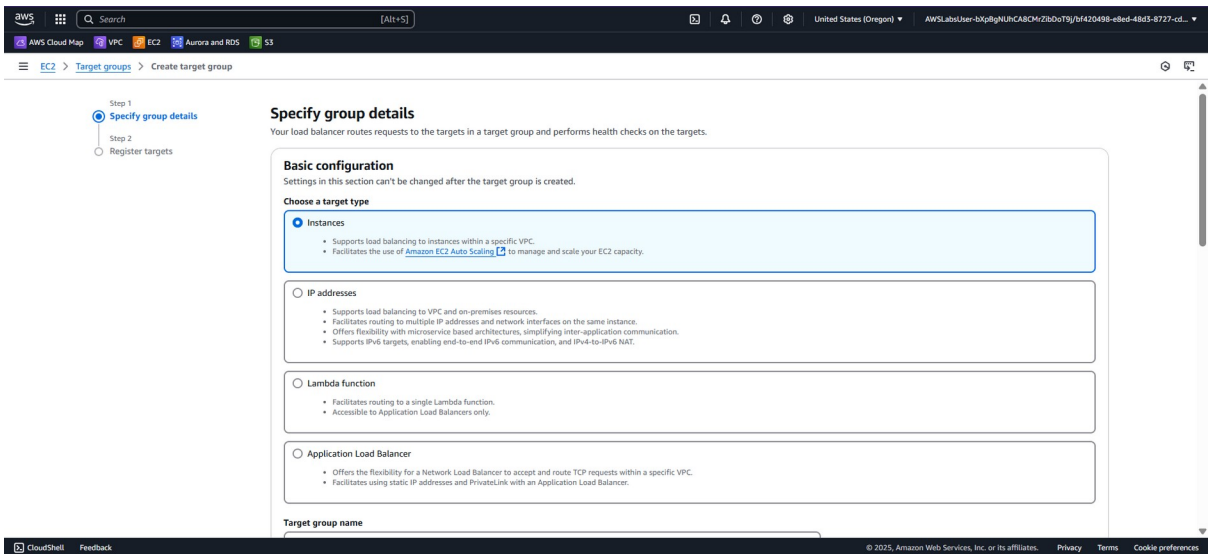
20. At the top of the console, in the search bar, search for and choose EC2.



21. In the left navigation pane, expand the **Load Balancing** section and choose **Target Groups**.



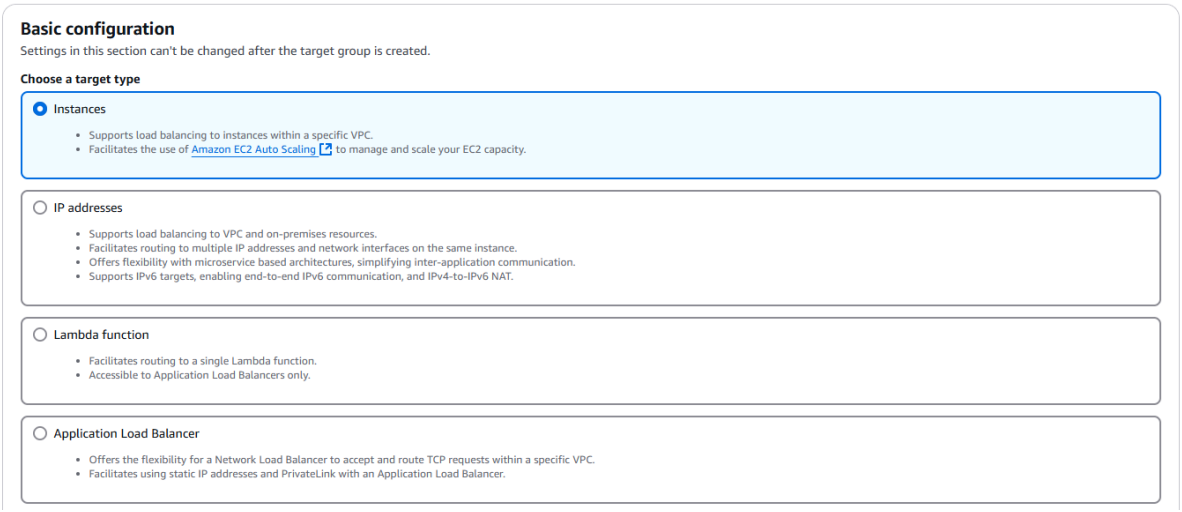
22. Choose **Create target group**.

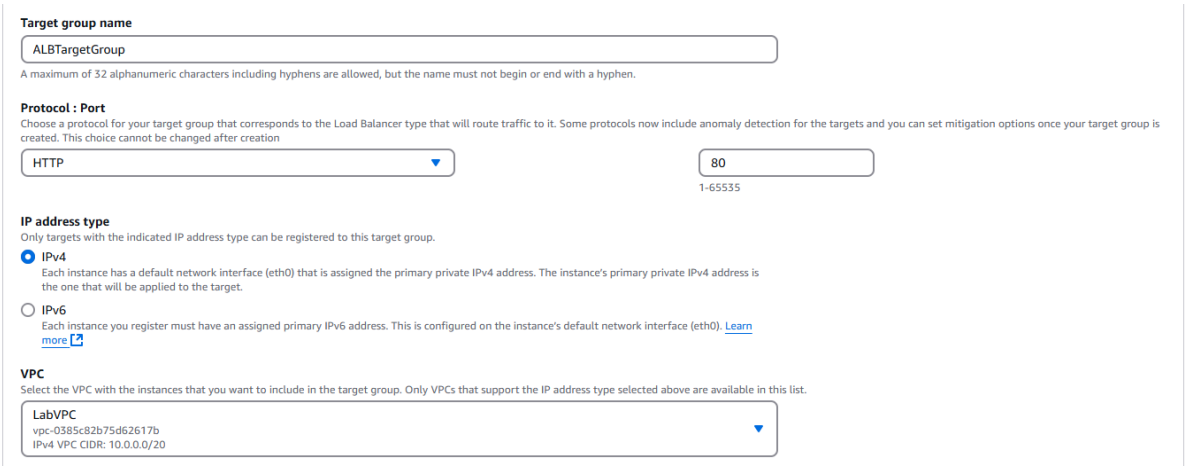


The **Specify group details** page is displayed.

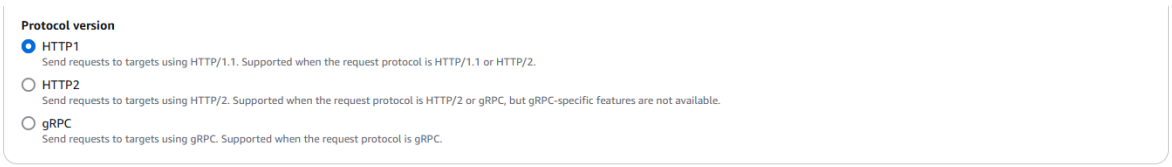
23. In the **Basic configuration** section, configure the following:

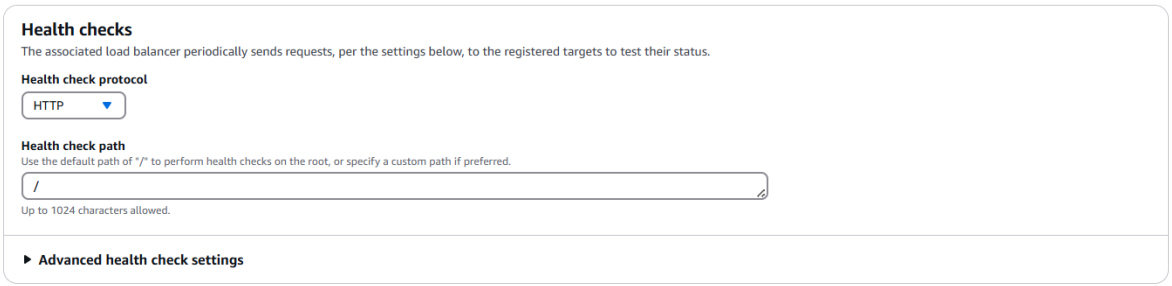
    - **Choose a target type**: Select **Instances**.
    - **Target group name**: Enter ALBTargetGroup.
    - **VPC**: Select **LabVPC** from the dropdown menu.

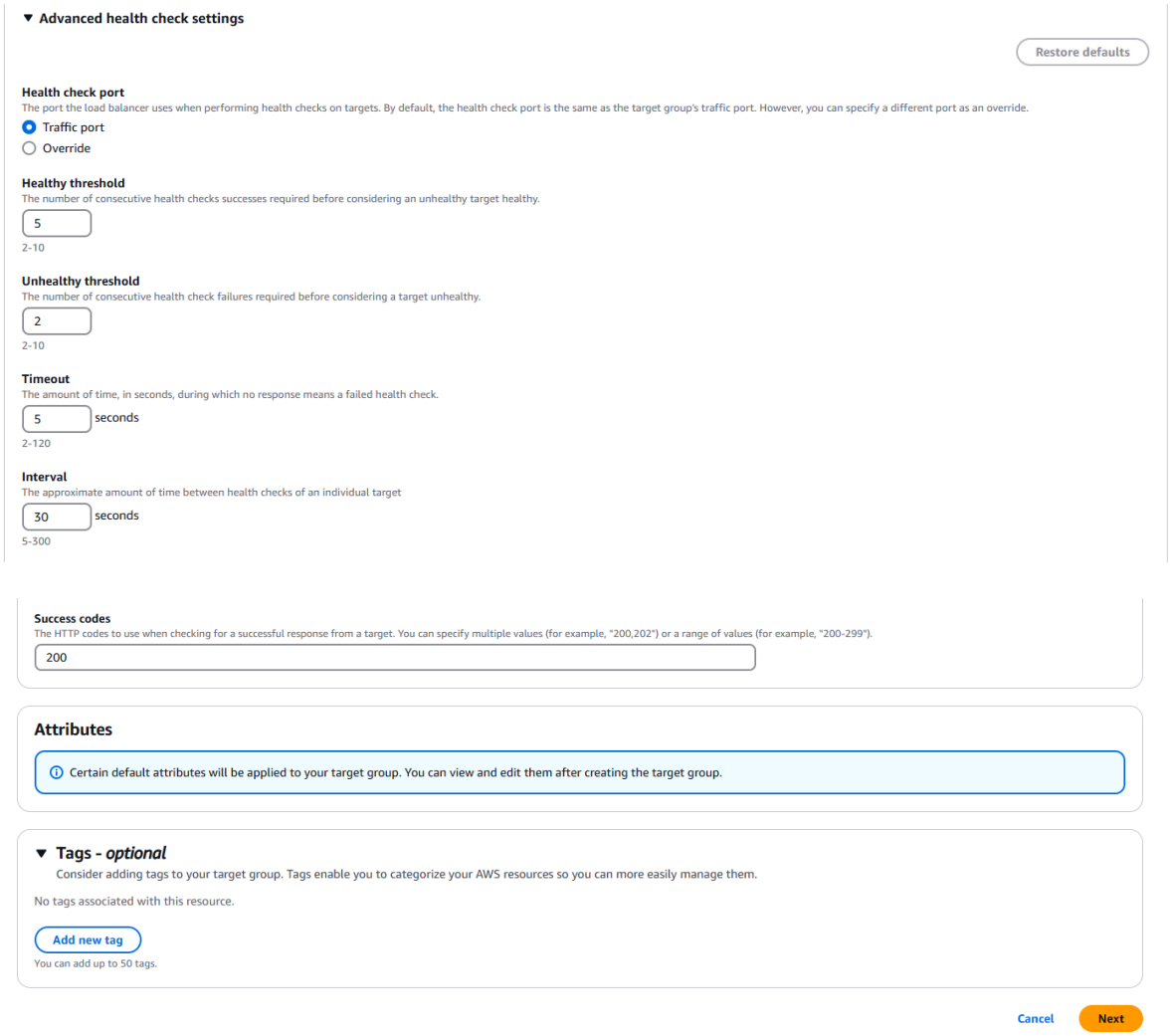




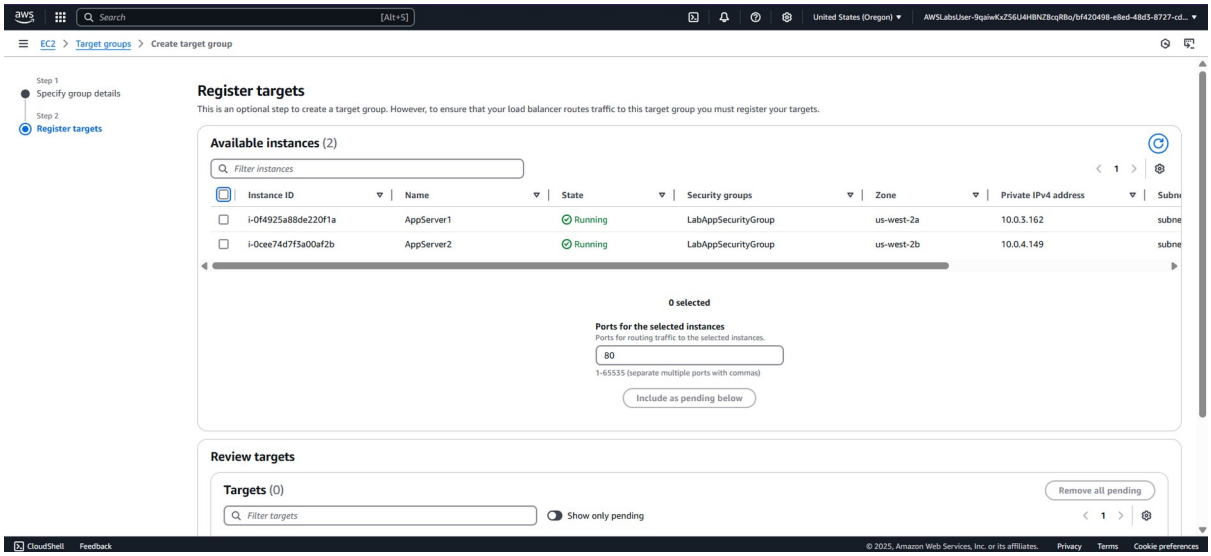
The remaining settings on the page can be left at their default values.







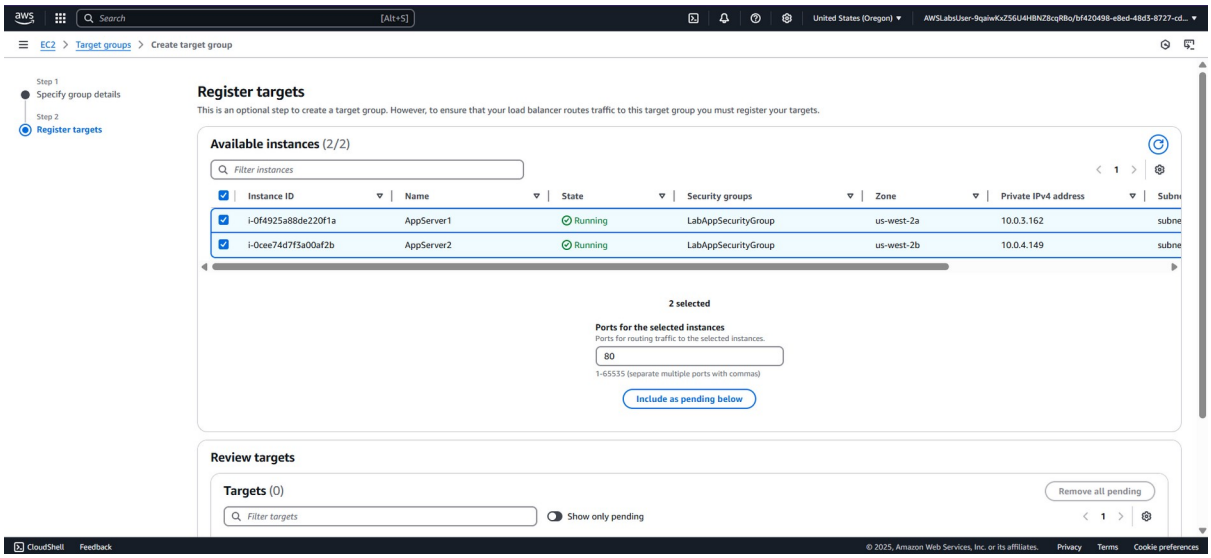
24. Choose **Next**.

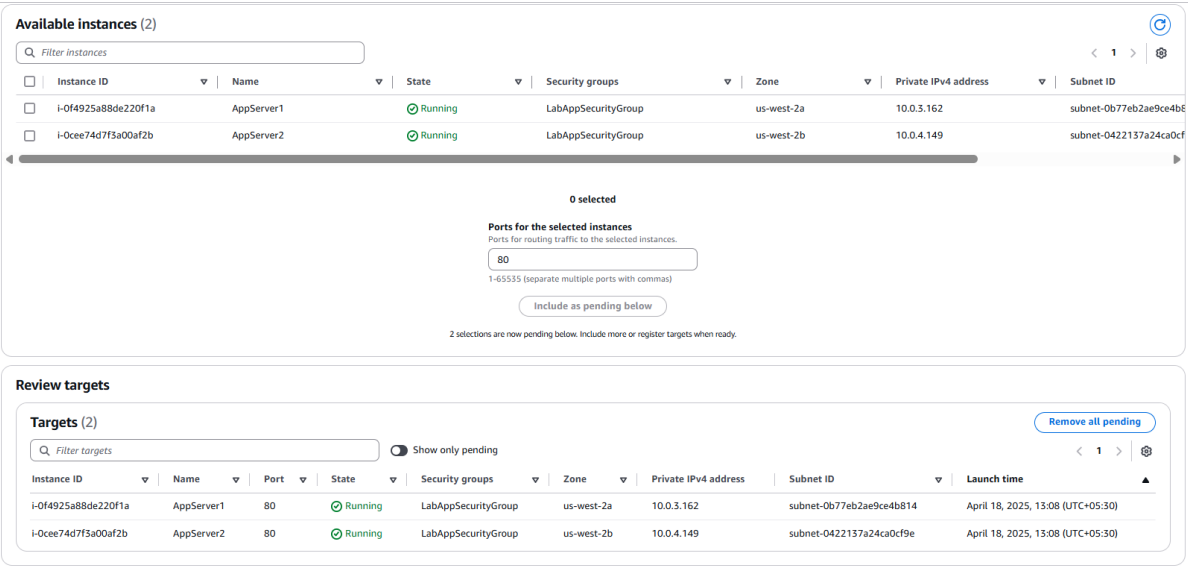


The **Register targets** page is displayed.

25. In the **Available instances** section, configure the following:

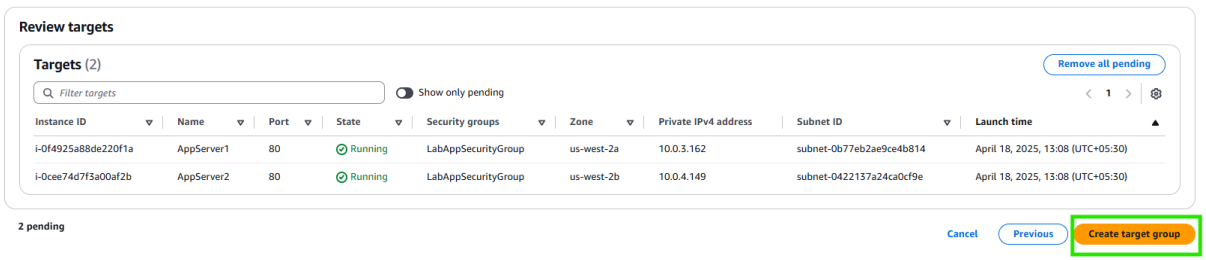
- Select the EC2 instance named **AppServer1** and **AppServer2**.
- Choose **Include as pending below**.

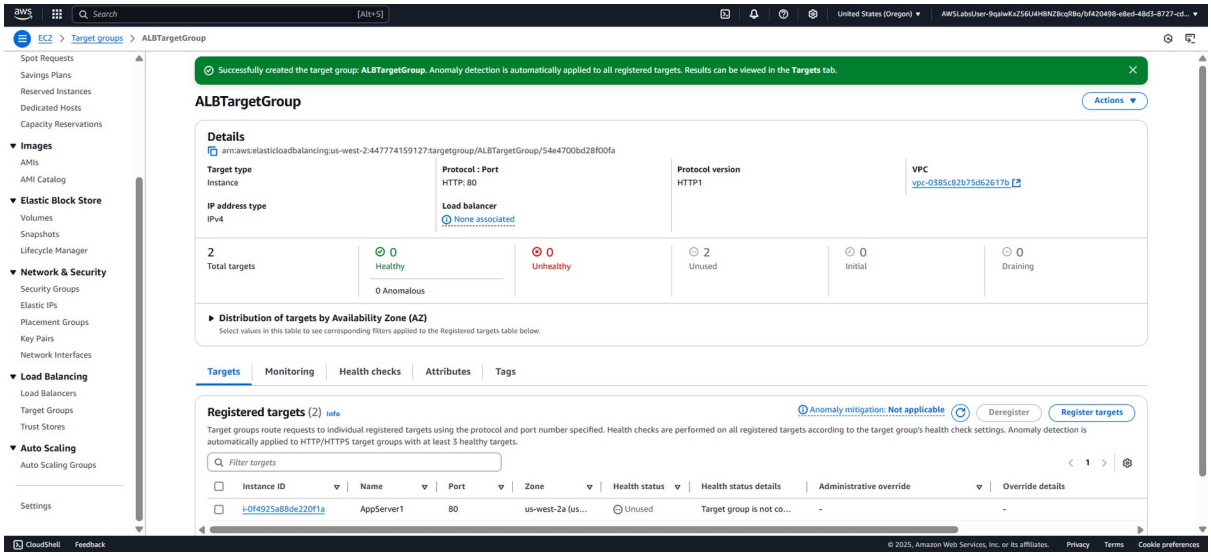




The instance appears under the **Targets** section of the page.

26. Choose **Create target group**.



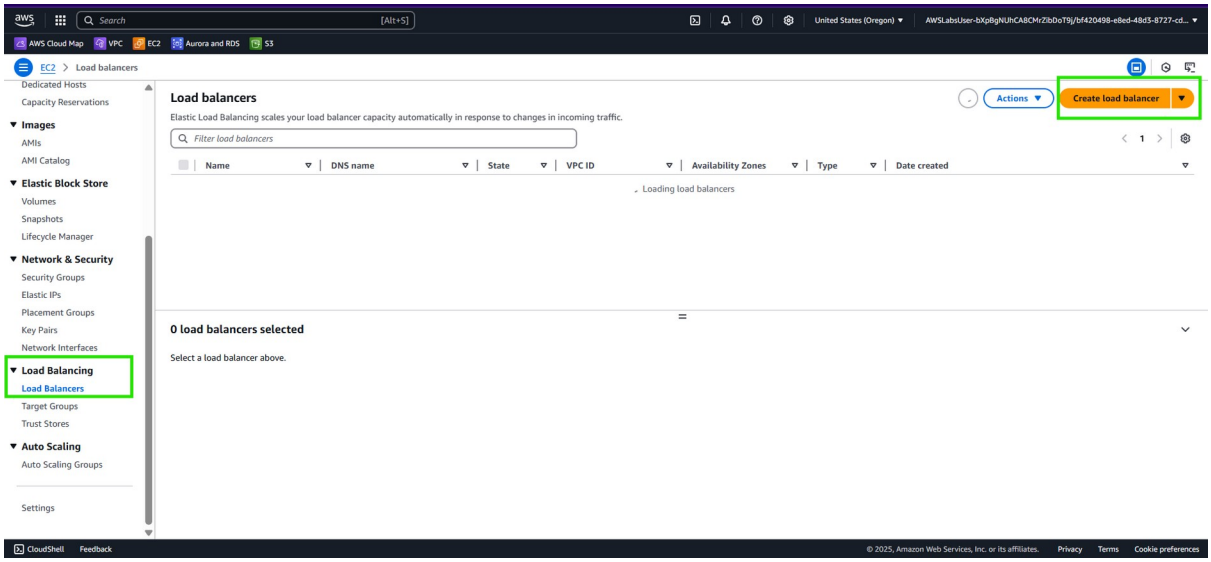


A (Successfully created target group: ALBTargetGroup) message is displayed on top of the screen.

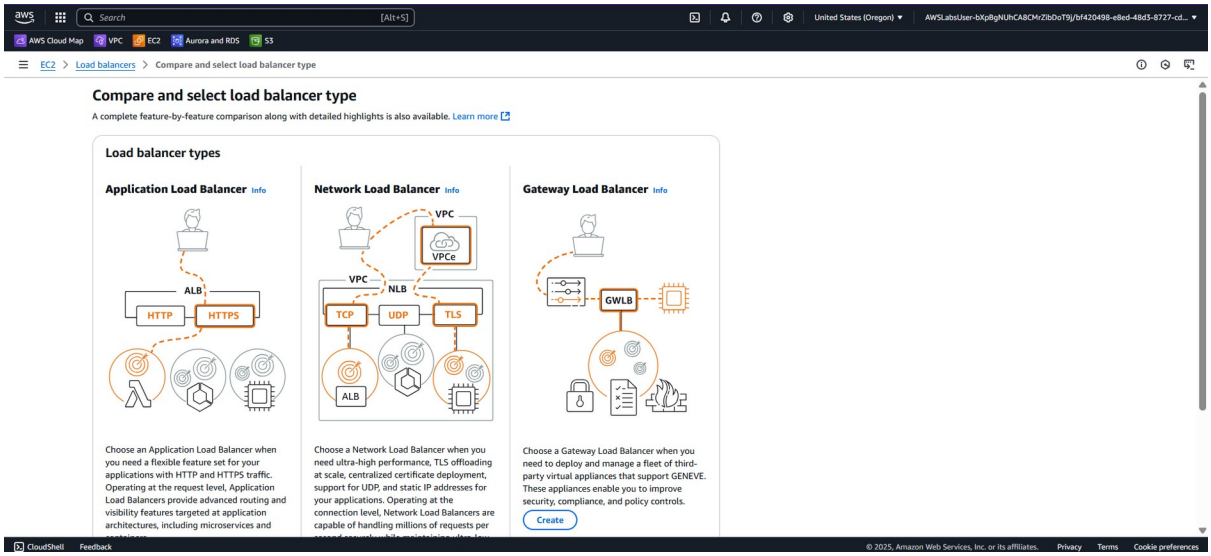
Task 2.2: Create an Application Load Balancer

In this task, you create an Application Load Balancer. 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.

27. In the left navigation pane, expand the **Load Balancing** section and choose **Load Balancers**.

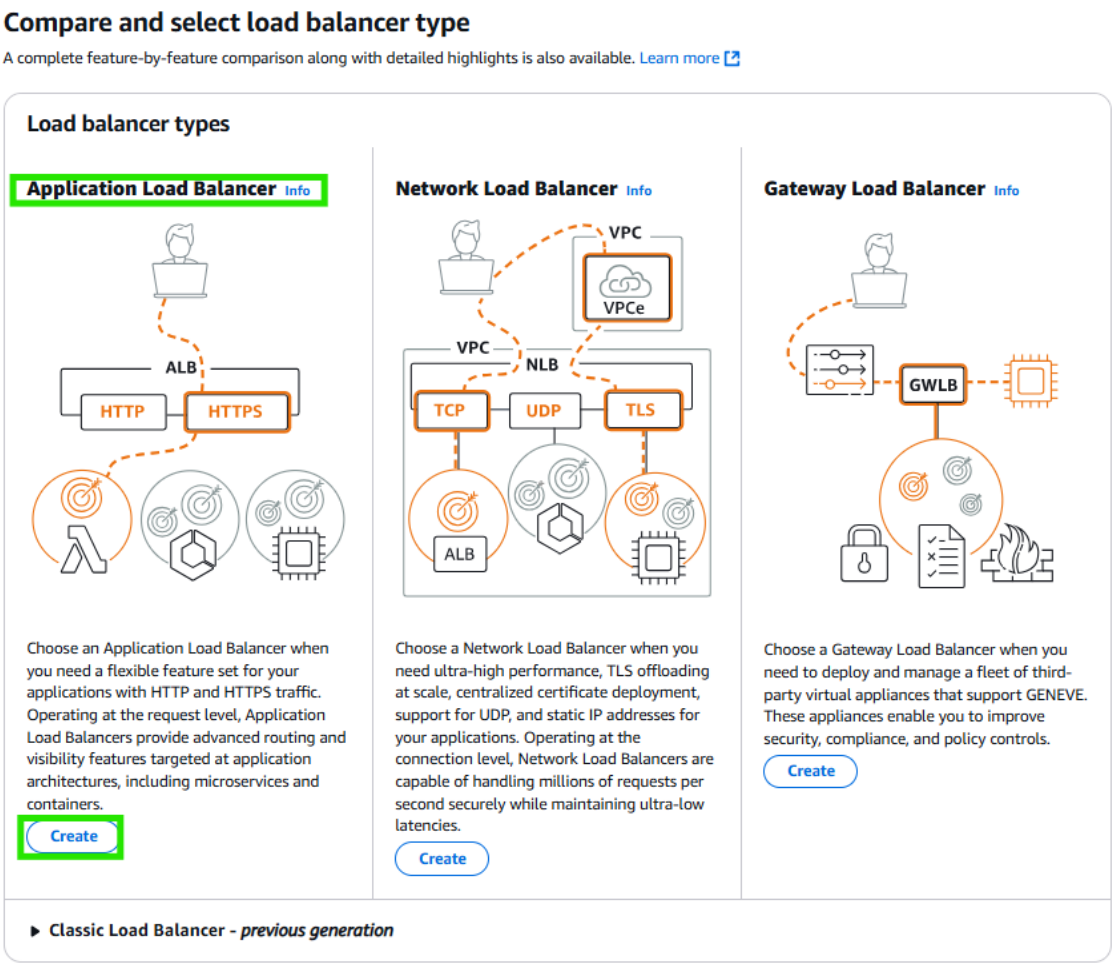


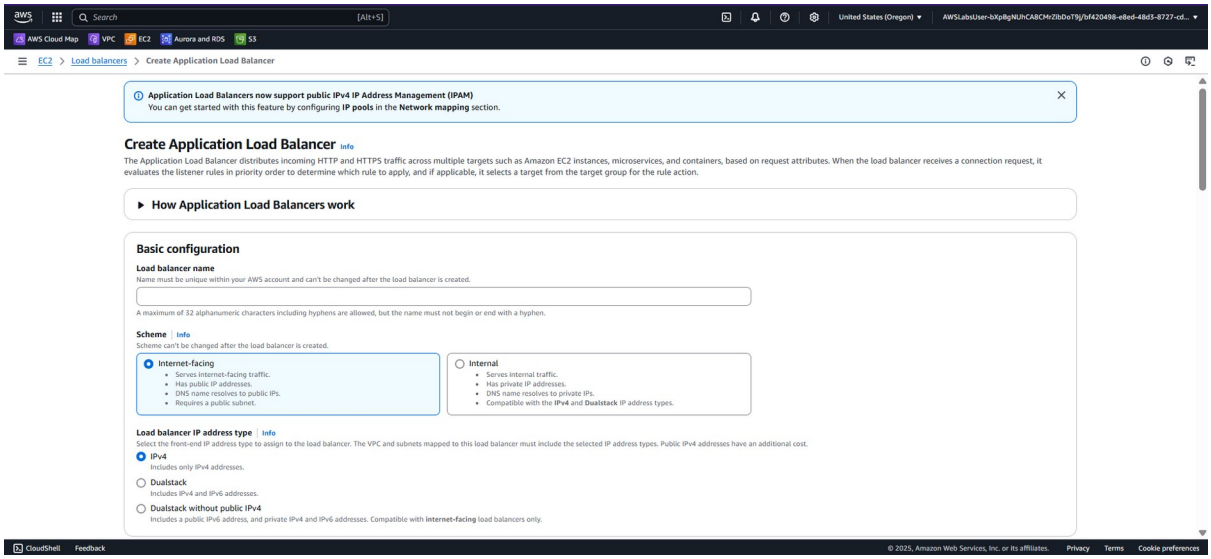
28. Choose **Create load balancer**.



Compare and select load balancer type page is displayed.

29. In the **Load balancer types** section, for **Application Load Balancer** card, choose **Create**.

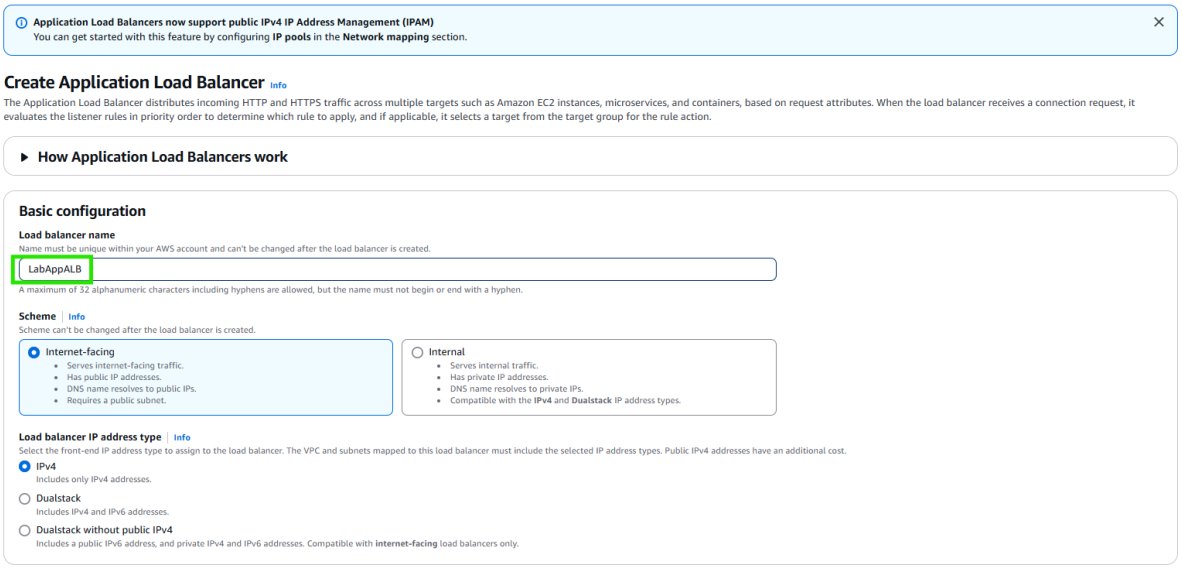




The **Create Application Load Balancer** page is displayed.

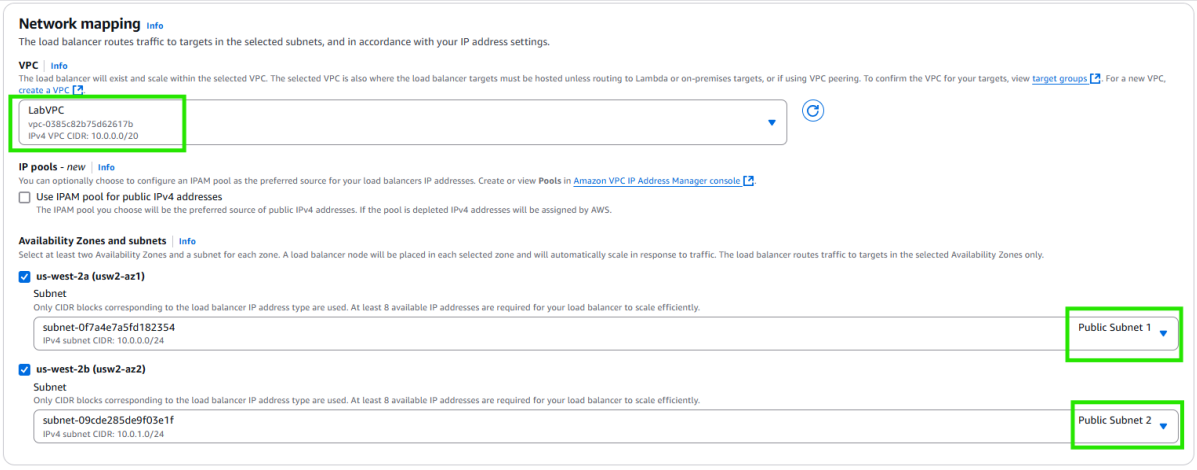
30. In the **Basic configuration** section, configure the following:

    - **Load balancer name**: Enter LabAppALB.



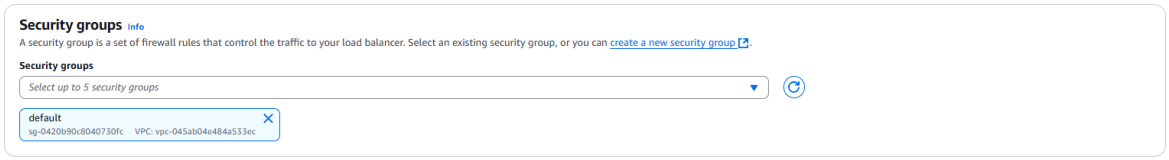
31. In the **Network mapping** section, configure the following:

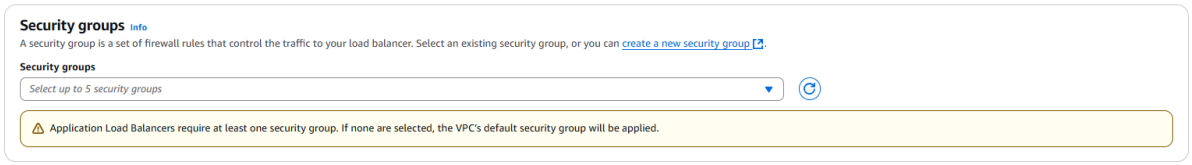
    - **VPC**: Select **LabVPC** from the dropdown menu.
    - **Mappings**:
        - Select the check box for the first Availability Zone listed, and select **PublicSubnet1** from the Subnet list dropdown menu.
        - Select the check box for the second Availability Zone listed, and select **PublicSubnet2** from the Subnet list dropdown menu.



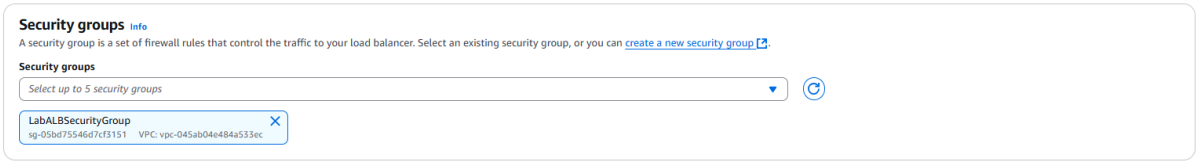
32. In the **Security groups** section, configure the following:

    - Select the **X** to remove the default security group.



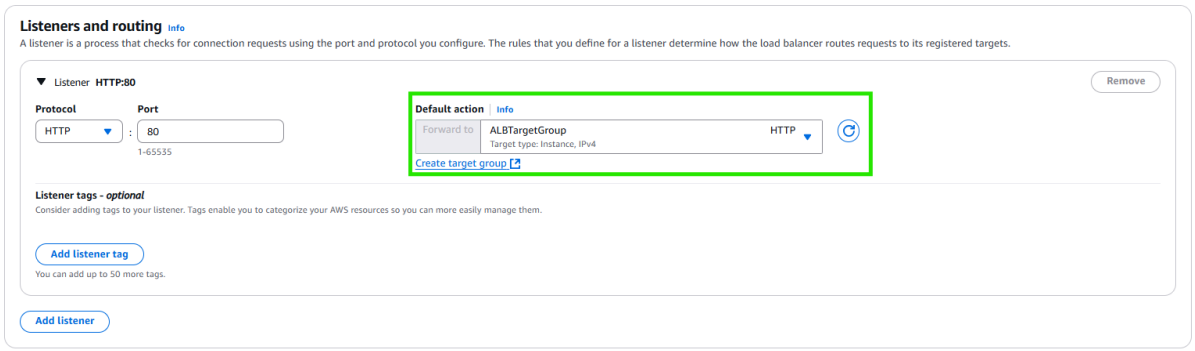


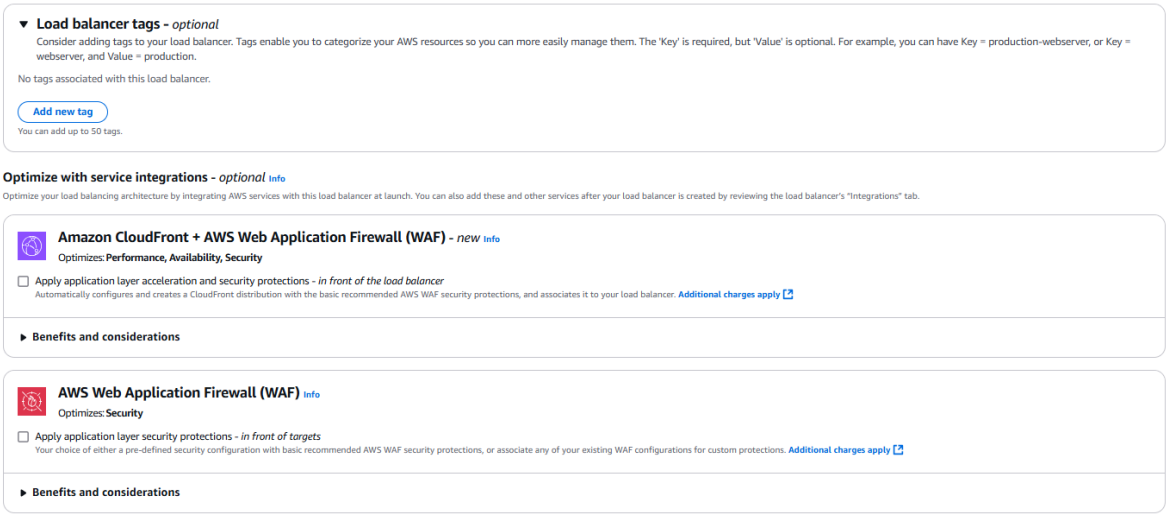
    - Select **LabALBSecurityGroup** from the dropdown menu.

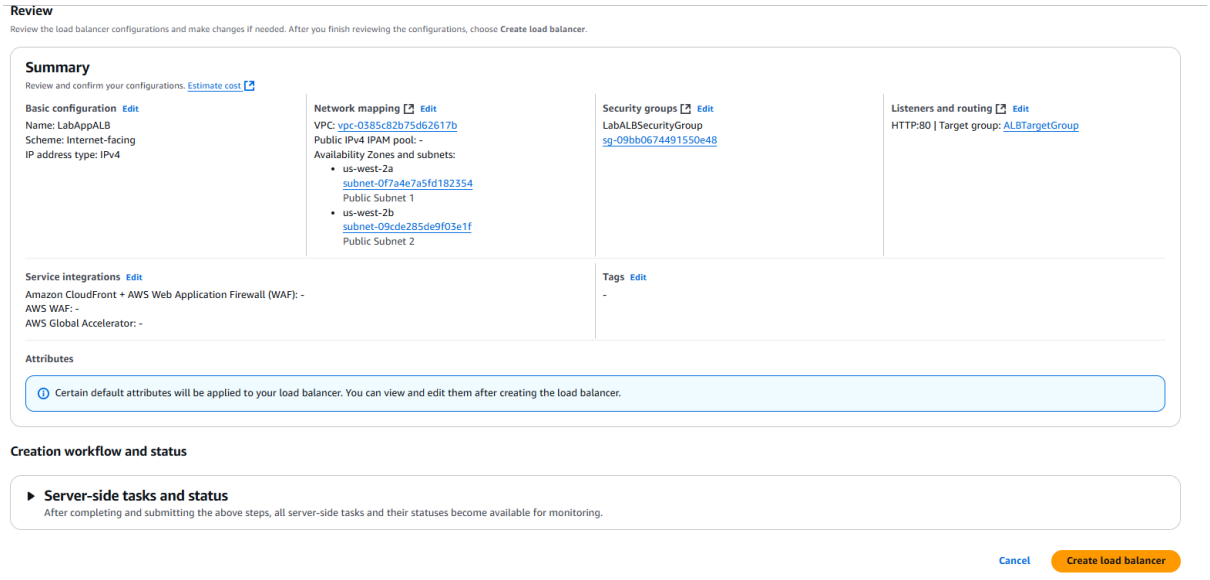


33. In the **Listeners and routing** section, configure the following:

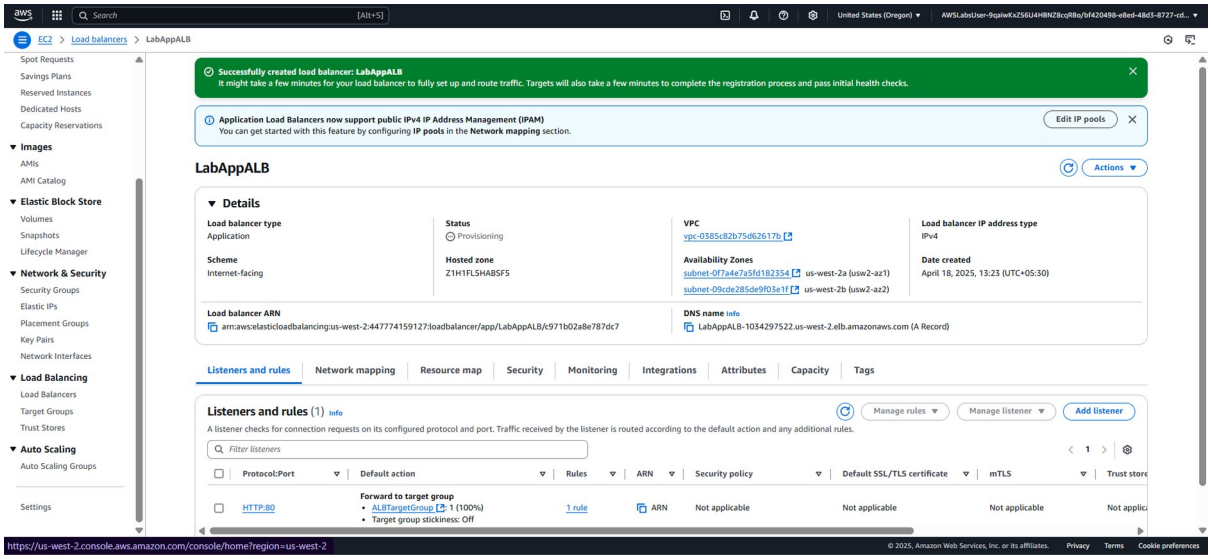
    - For **Listener HTTP:80**: From the Default action dropdown menu, select **ALBTargetGroup**.





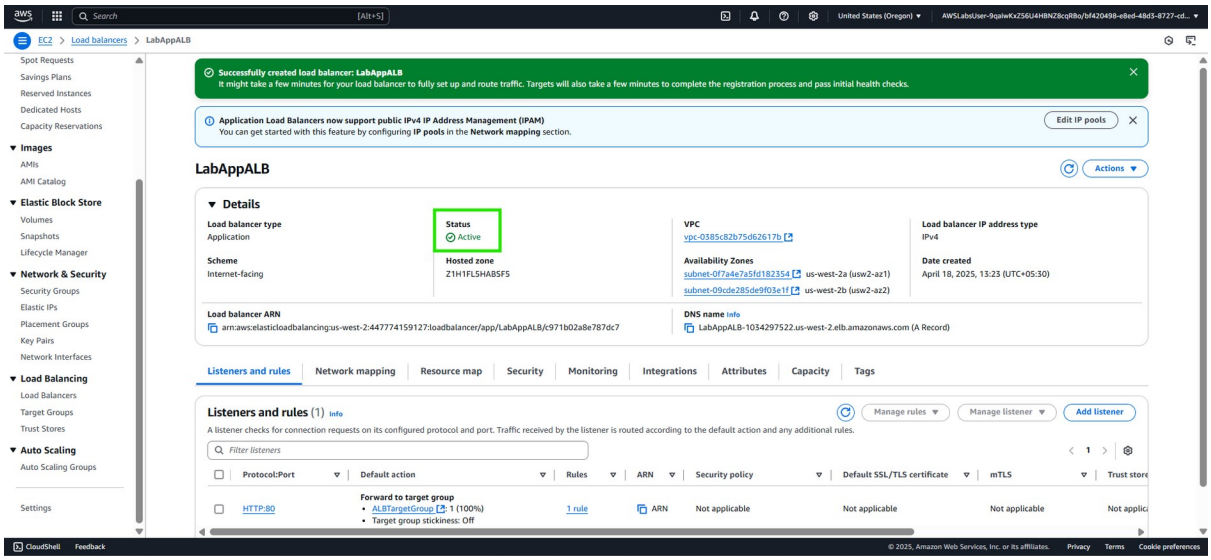


34. Choose **Create load balancer**.



A **(Successfully created load balancer: LabAppALB)** message is displayed on top of the screen.

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



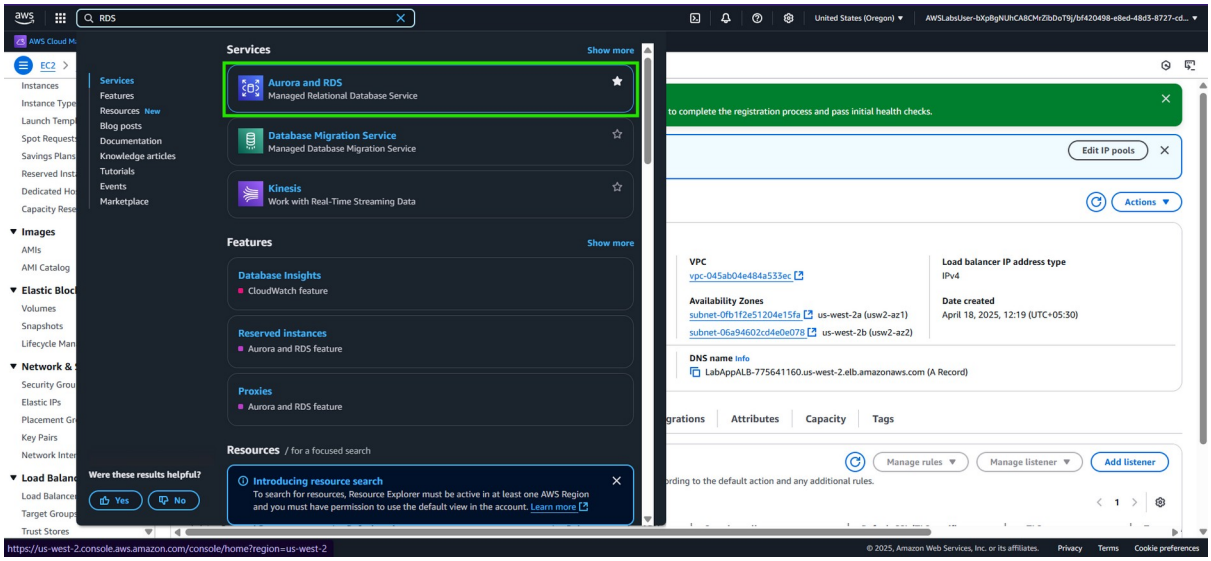
In this task, you created an Application Load Balancer and you added EC2 instances as a target to the load balancer. This task provides a demonstration on how to register a target with a load balancer. In addition to individual EC2 instances, Auto Scaling groups can also be registered as targets for 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, EC2 instances that are ended by the Auto Scaling groups are automatically unregistered from the load balancer. Using Auto Scaling groups with a load balancer is demonstrated in the next lab.

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

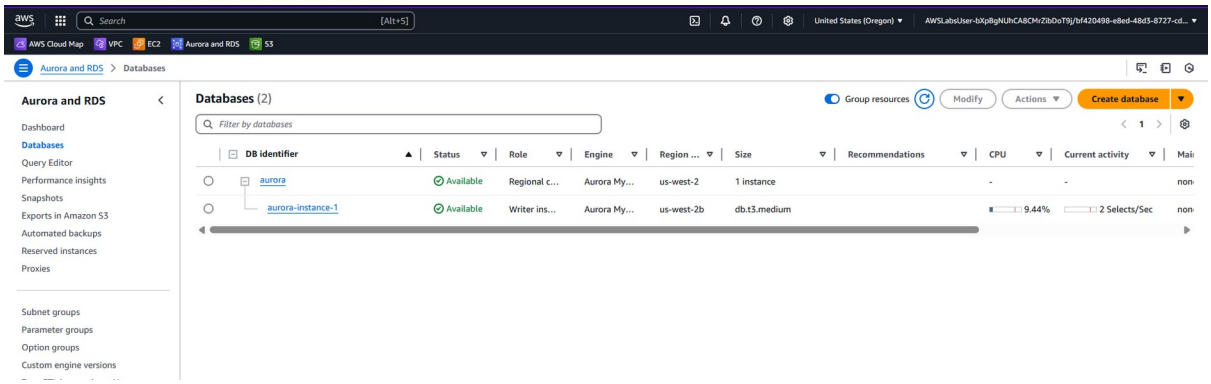
Task 3: Review the Amazon RDS DB instance metadata through the console

In this task, you navigate through the Amazon RDS console to ensure the instance created in Task 1 has completed and is active. You explore the 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.

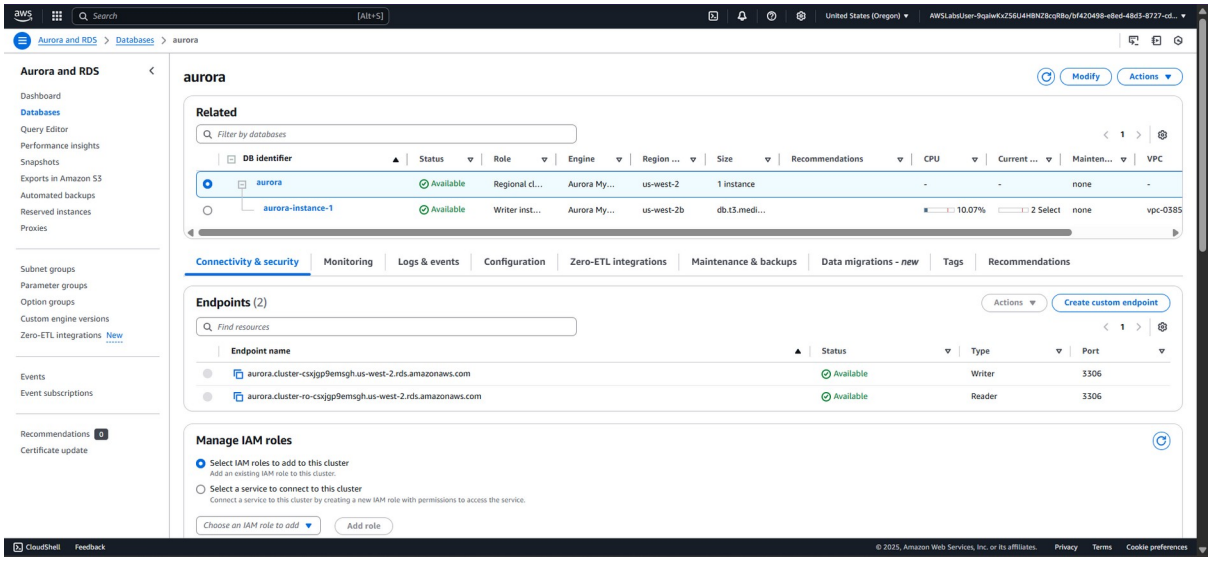
35. At the top of the console, in the search bar, search for and choose RDS.



36. In the navigation pane, choose **Databases**.



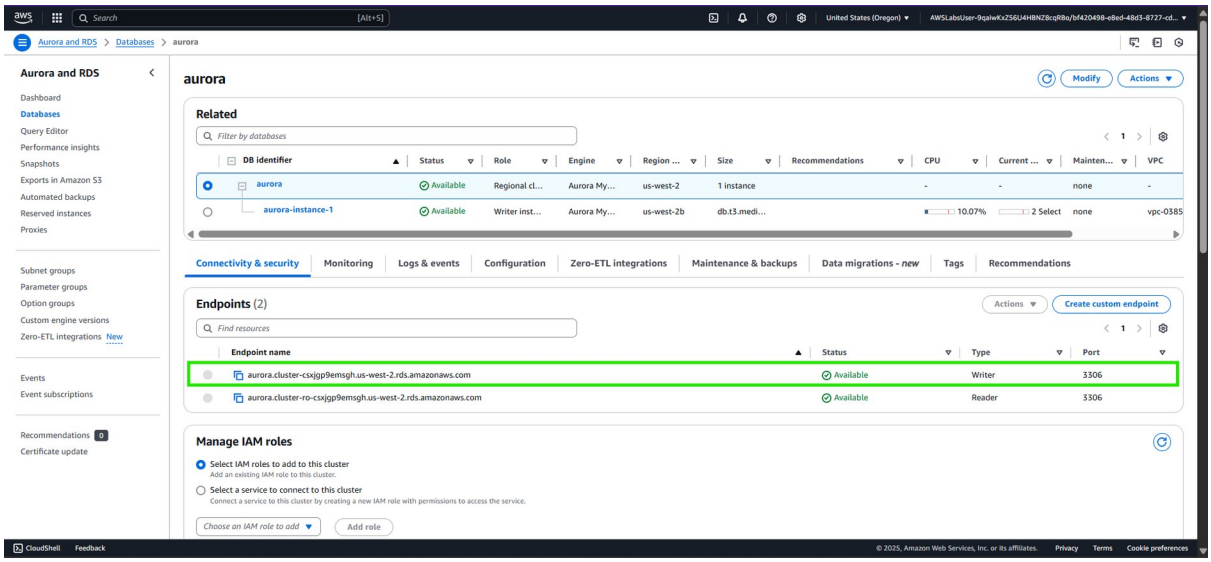
37. From the list of DB identifiers, select the hyperlink for the cluster named **aurora**.



A page with details about the database is displayed.

38. 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.

39. Copy and paste the **Endpoint name** of the **writer instance** value to a notepad. You need this value later in the lab.

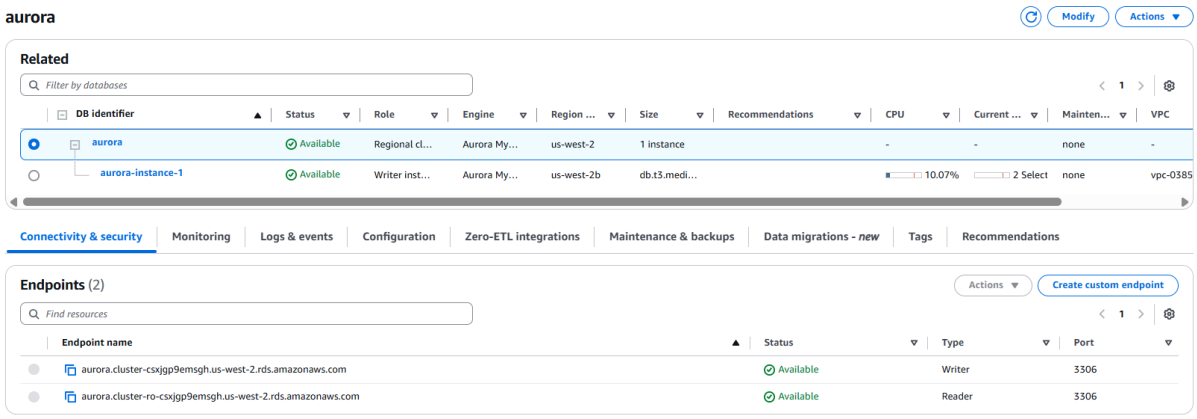


It should look similar to *aurora.cluster-crwxbgqad61a.us-west-2.rds.amazonaws.com*.

End point: *aurora.cluster-csxjgp9emsgh.us-west-2.rds.amazonaws.com*
Port*: 3306*

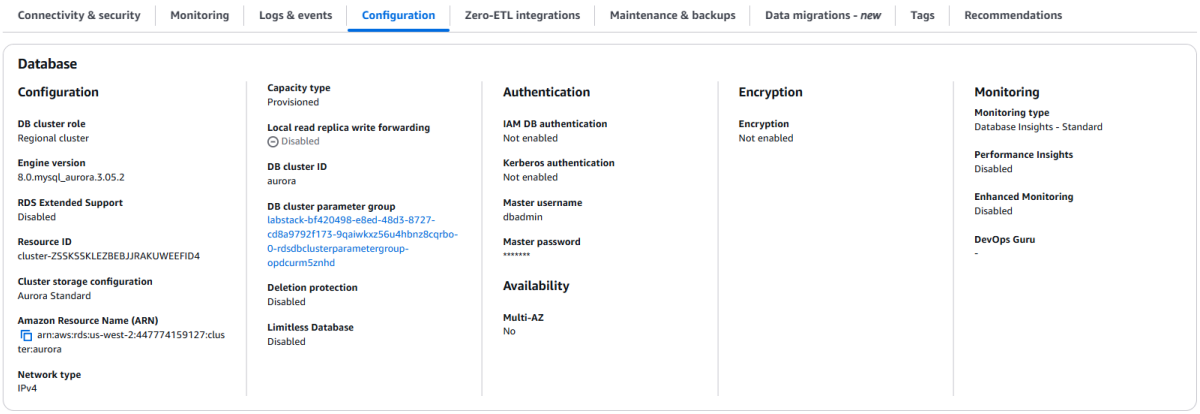
End point*: aurora.cluster-ro-csxjgp9emsgh.us-west-2.rds.amazonaws.com*
Port*: 3306*

Tip: To copy the **writer instance** endpoint, hover on it and choose the copy icon.



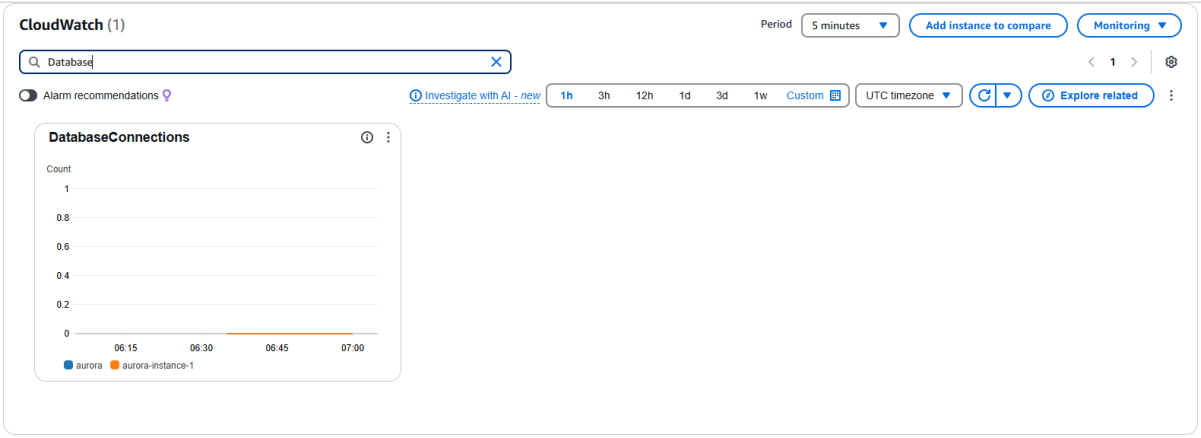
Notice that the status for the **endpoints** is Available.

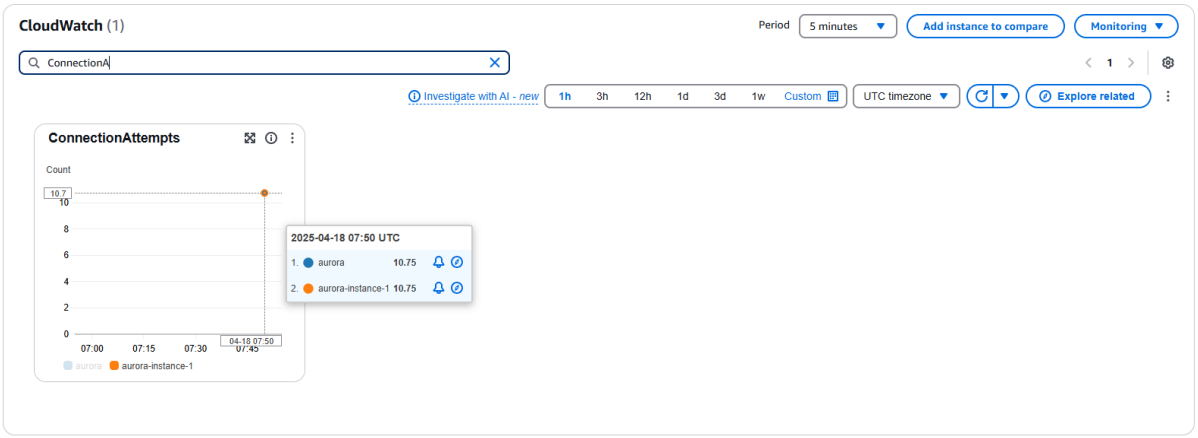
40. On the **Configuration** tab, you can find details regarding how the database is currently configured.



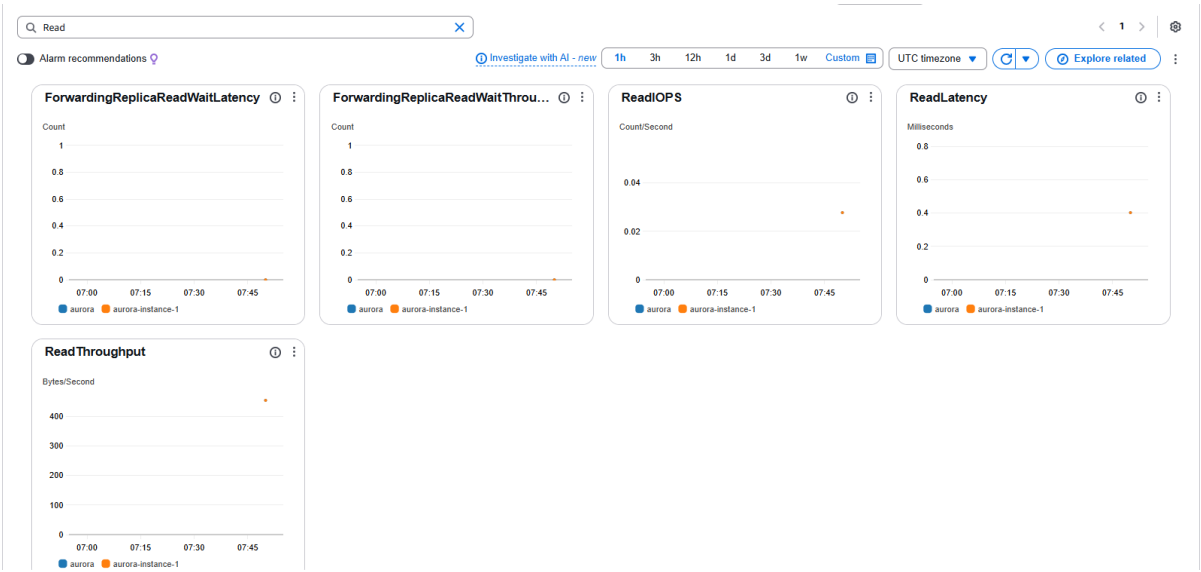
41. On the **Monitoring** tab, you can monitor metrics for the following items of your database:

    - The number of connections to a database instance

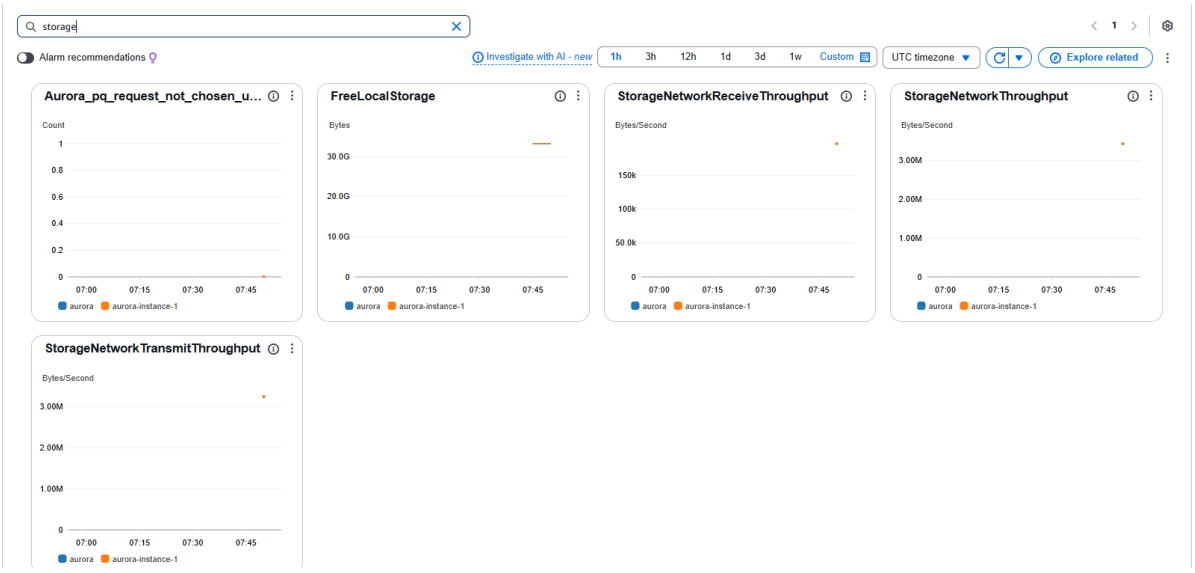




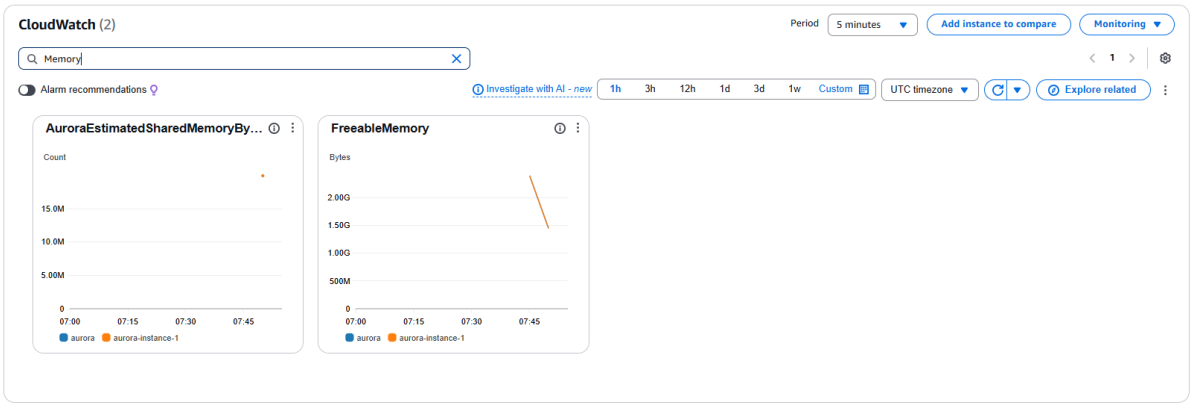
- The amount of read and write operations to a database instance

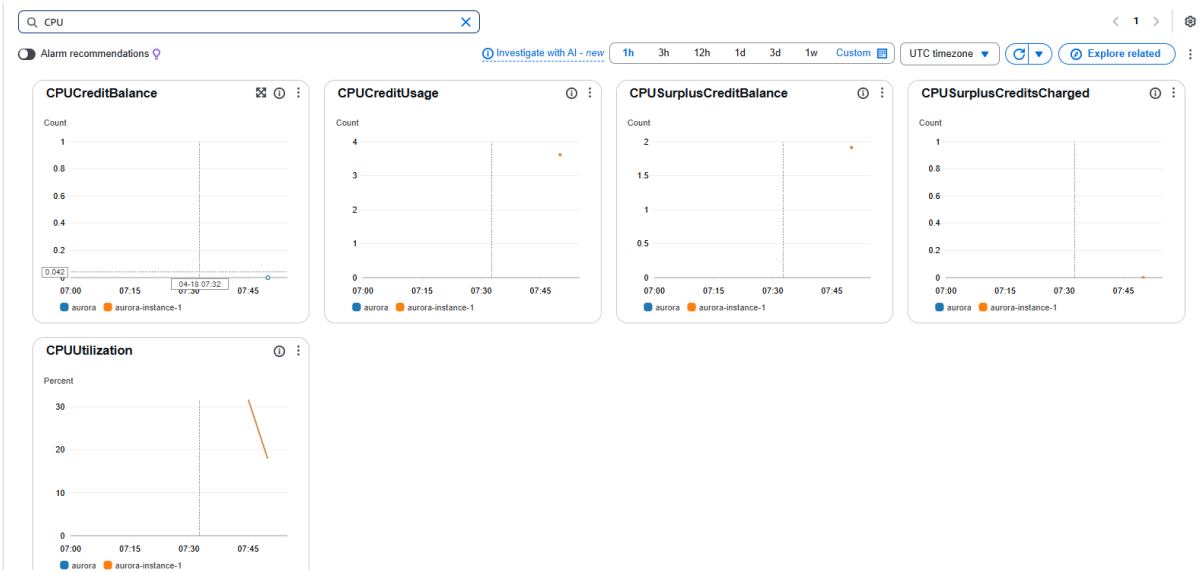


- The amount of storage that a database instance is currently using

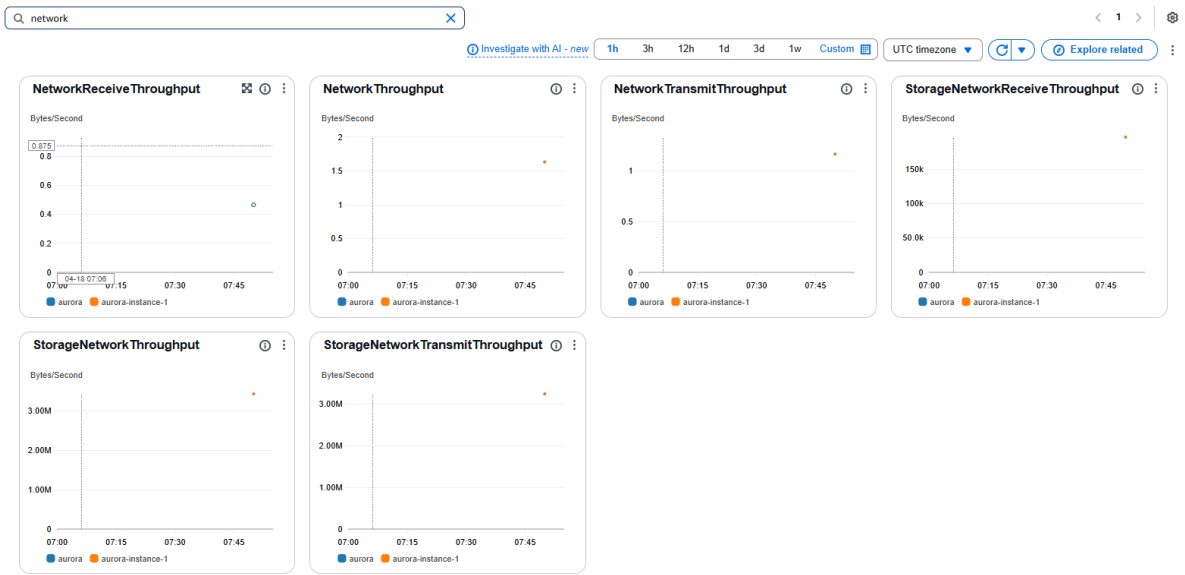


- The amount of memory and CPU being used for a database instance

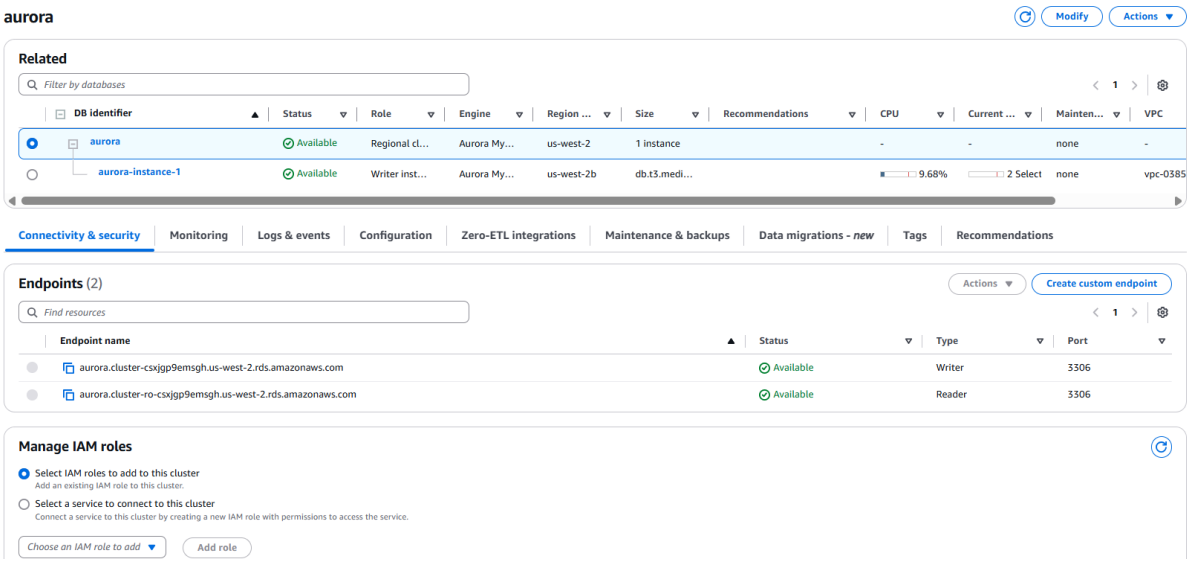




- The amount of network traffic to and from a database instance



WARNING: Wait for the *Status* of the **aurora DB instance** to show as Available before continuing to the next task.

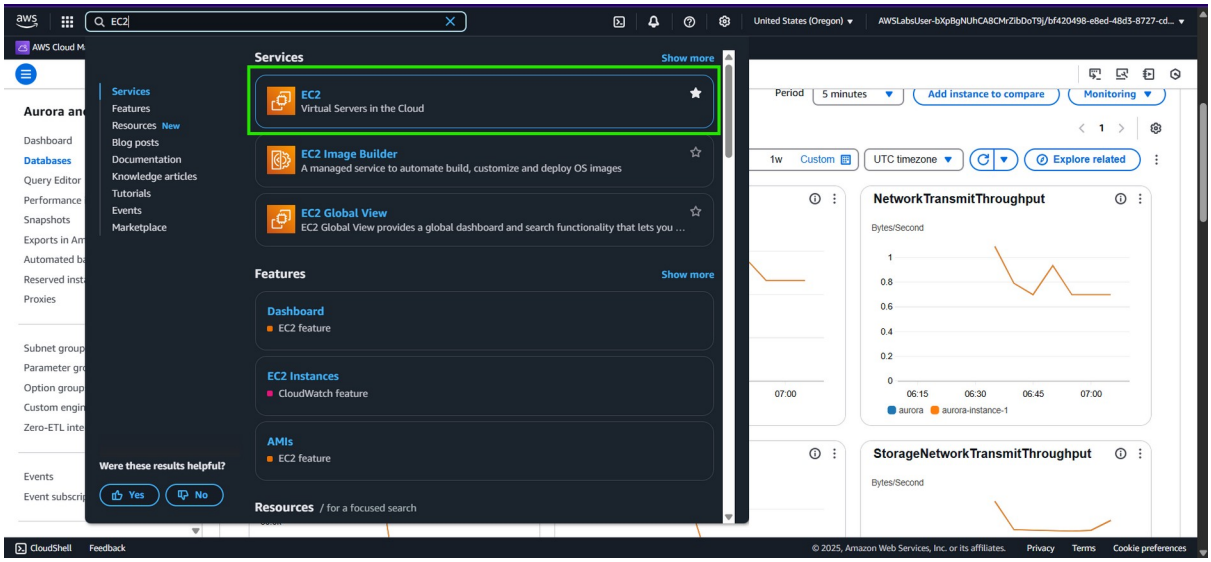


Congratulations! You have successfully reviewed the Amazon RDS DB instance metadata through the console.

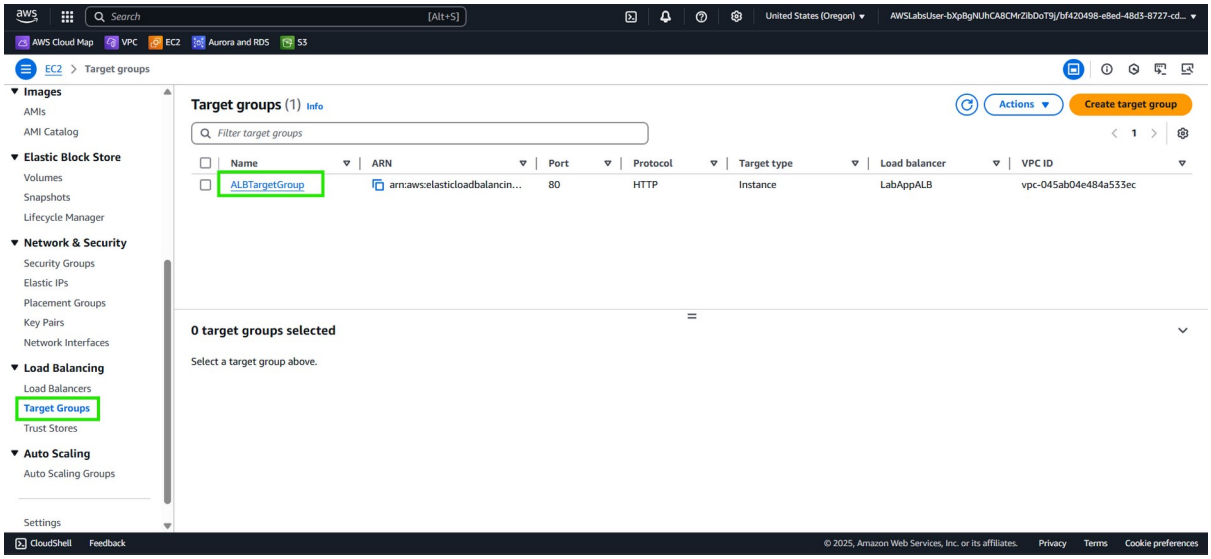
Task 4: Test the application connectivity to the database

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

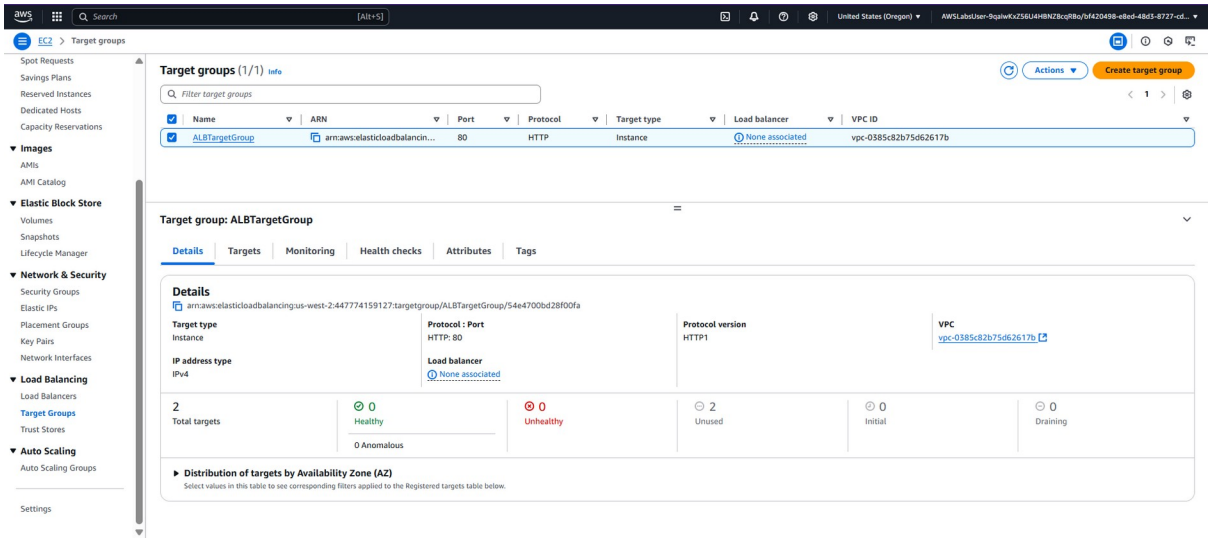
42. At the top of the console, in the search bar, search for and choose EC2.



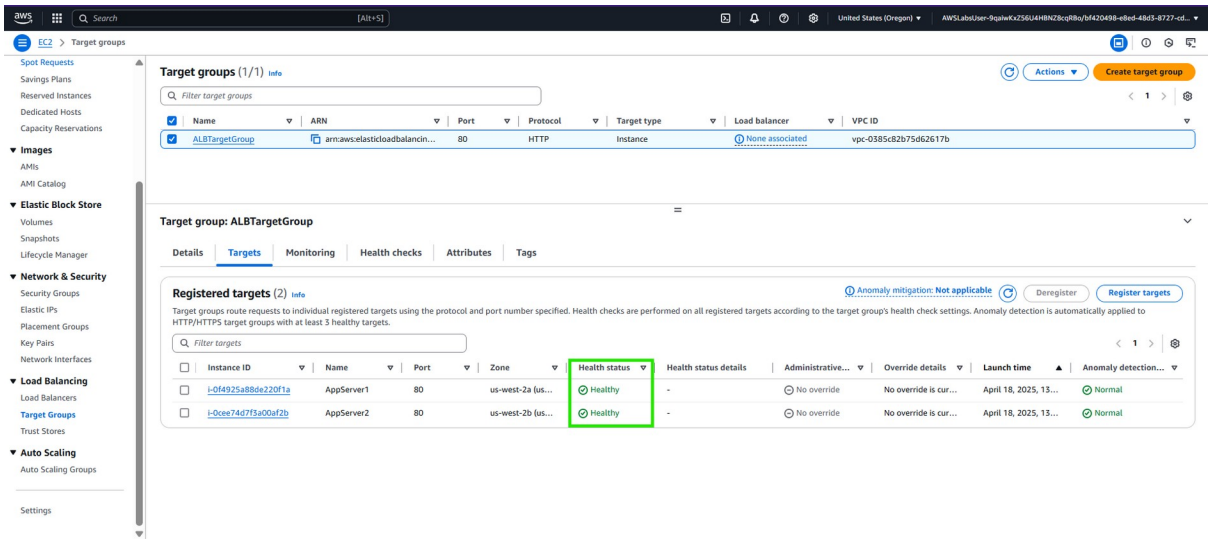
43. In the left navigation pane, choose **Target Groups**.



44. Select **ALBTargetGroup**.

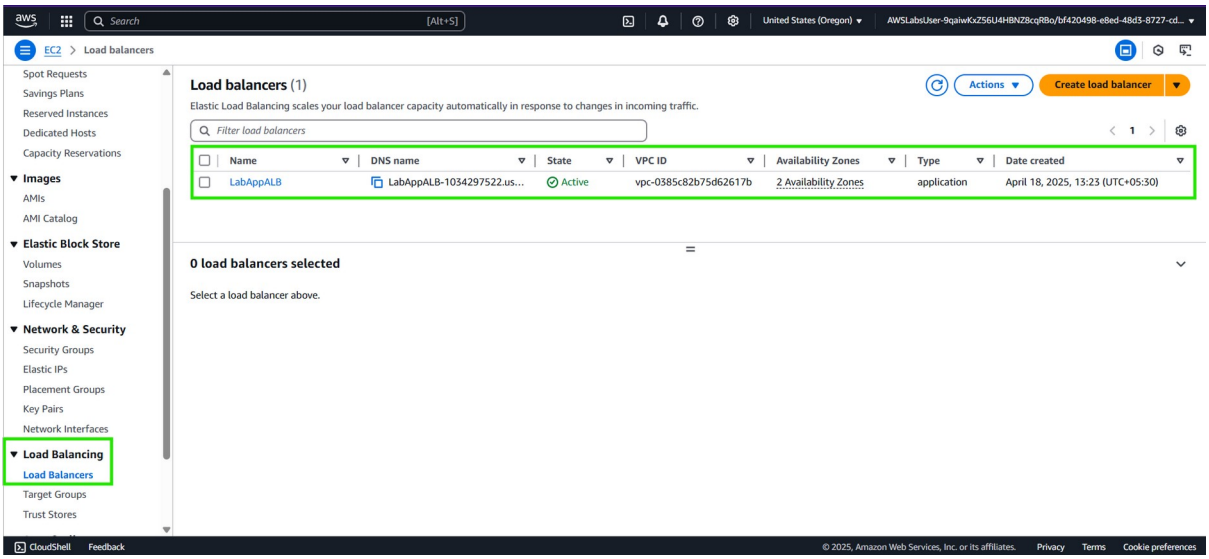


45. In the **Targets** tab, wait until the instance status is displayed as healthy.



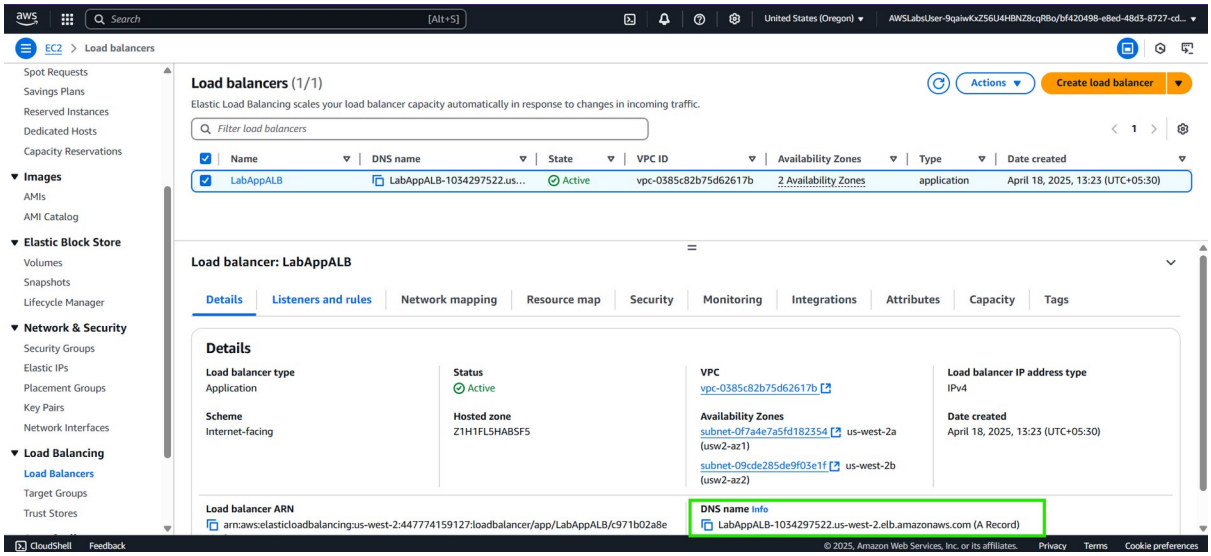
Learn more: Elastic Load Balancing periodically tests the ping path on your web server instance to determine health. A 200 HTTP response code indicates a healthy status, and any other response code indicates an unhealthy status. If an instance is unhealthy and continues in that state for a successive number of checks (unhealthy threshold), the load balancer removes it from service until it recovers. For more information, see Health checks for your target groups.

46. In the left navigation pane, choose **Load Balancers**.



The **Load balancers** page is displayed.

47. Copy the **DNS name** and paste the value in a new browser tab to invoke the load balancer.



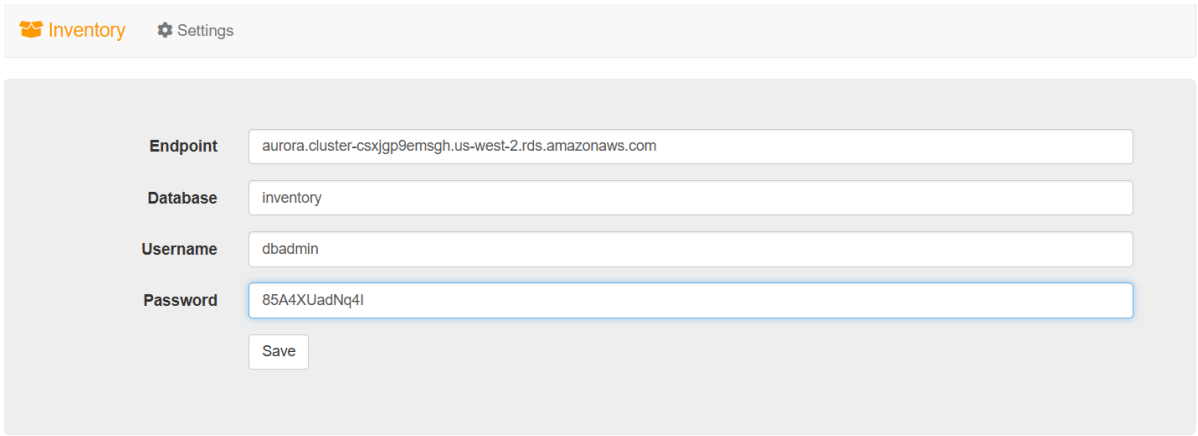
Tip: To copy the *DNS name*, hover on it and select the copy icon.

DNS name: *LabAppALB-1034297522.us-west-2.elb.amazonaws.com*

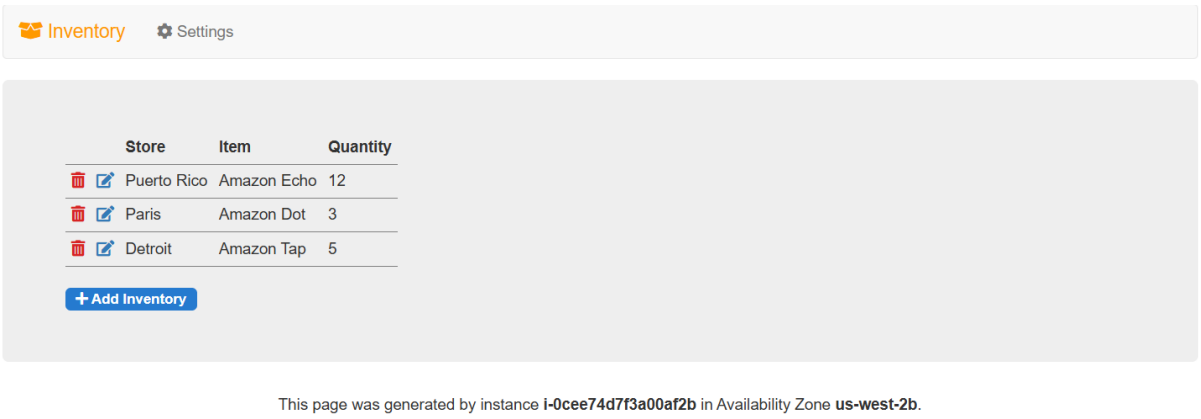
Expected output: A web page like this is displayed.

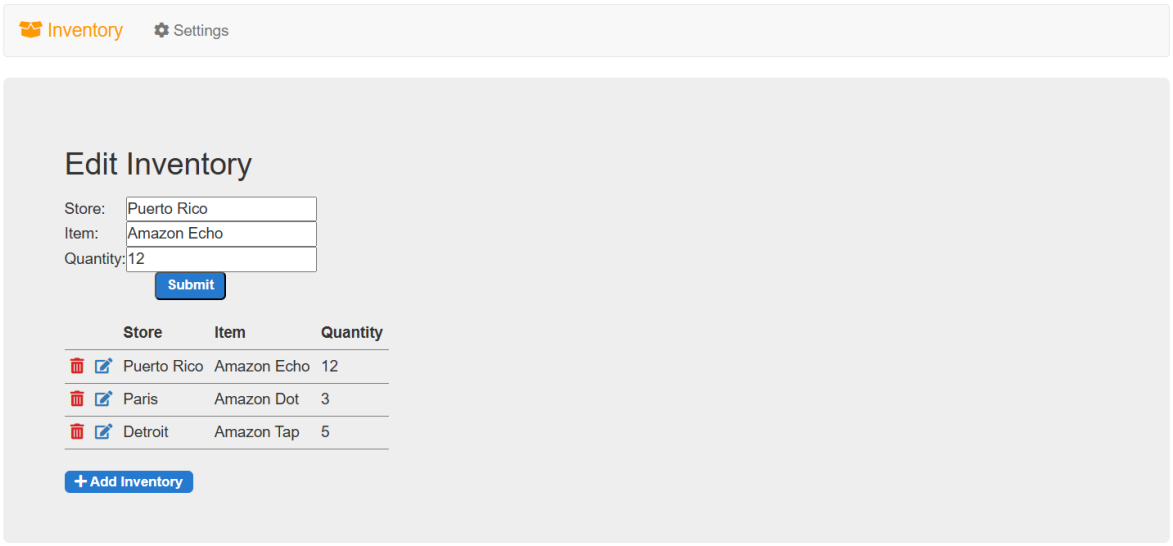
48. Choose the **Settings** tab and then configure the following:

    - **Endpoint:** Paste the *writer instance endpoint* you copied earlier. (aurora.cluster-csxjgp9emsgh.us-west-2.rds.amazonaws.com)
    - **Database:** Enter inventory.
    - **Username:** Enter dbadmin.
    - **Password:** Paste the **LabPassword** (**85A4XUadNq4I**) value from the left side of these lab instructions.



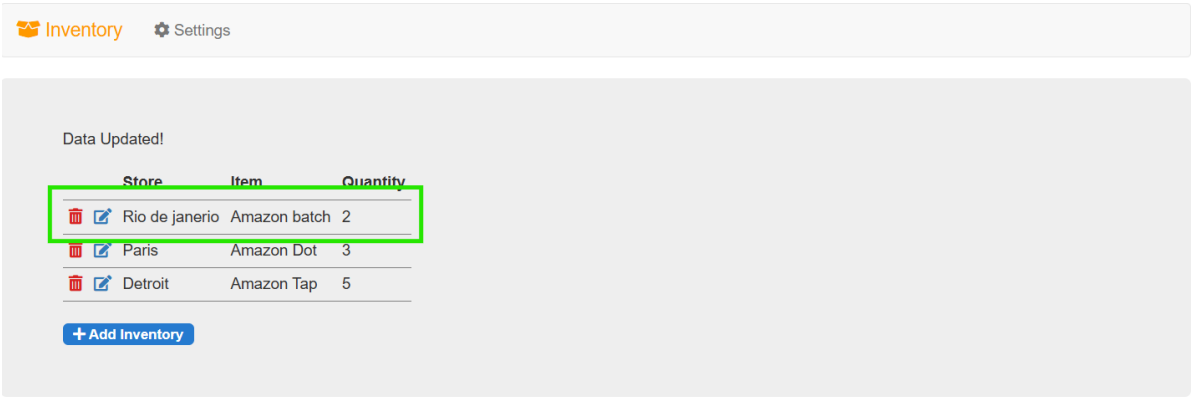
49. Choose **Save**



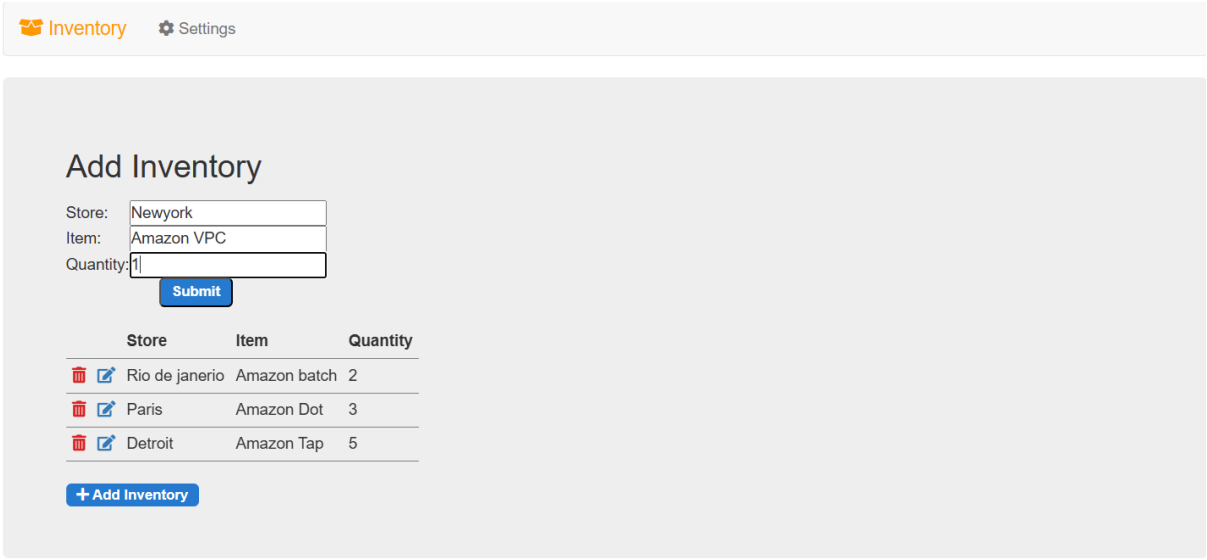


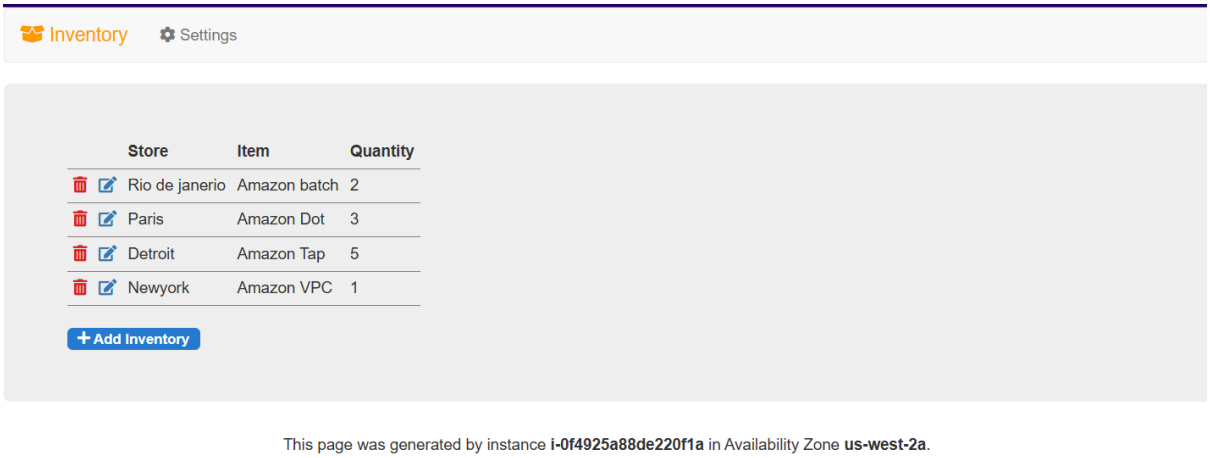
This page was generated by instance **i-0f4925a88de220f1a** in Availability Zone **us-west-2a**.

This page was generated by instance **i-0f4925a88de220f1a** in Availability Zone **us-west-2a**.



This page was generated by instance **i-0f4925a88de220f1a** in Availability Zone **us-west-2a**.





The application connects to the database, loads some initial data, and displays information. With this application, you can 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 won't be lost. It also means that multiple application servers can access the same data.

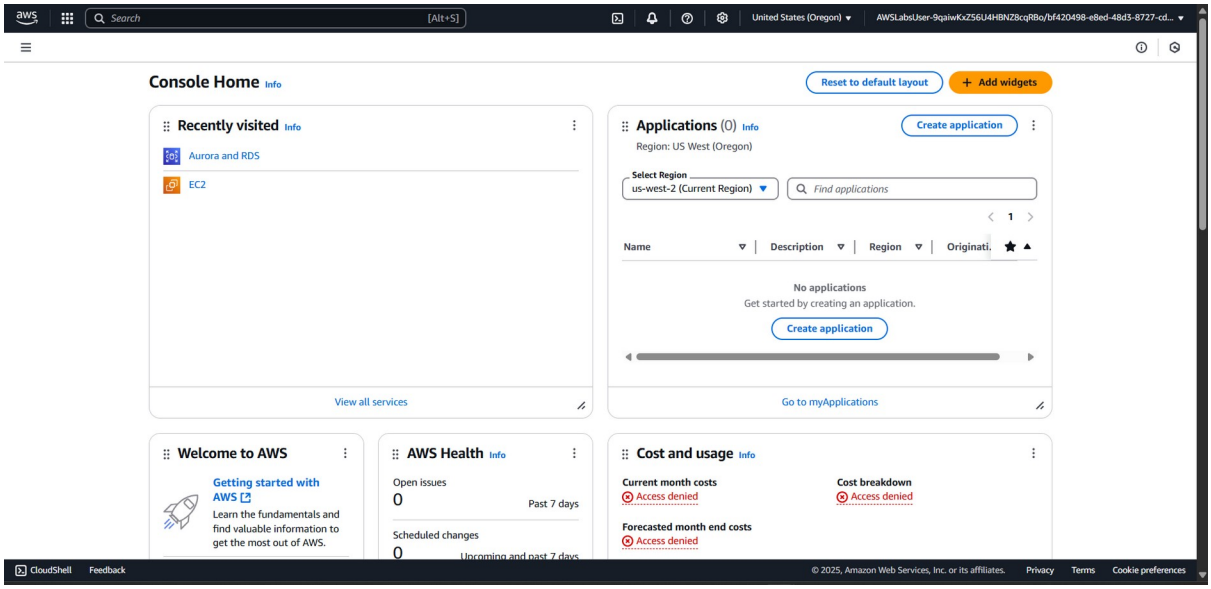
Congratulations! You have successfully accessed the web application installed on the EC2 instance through the load balancer.

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

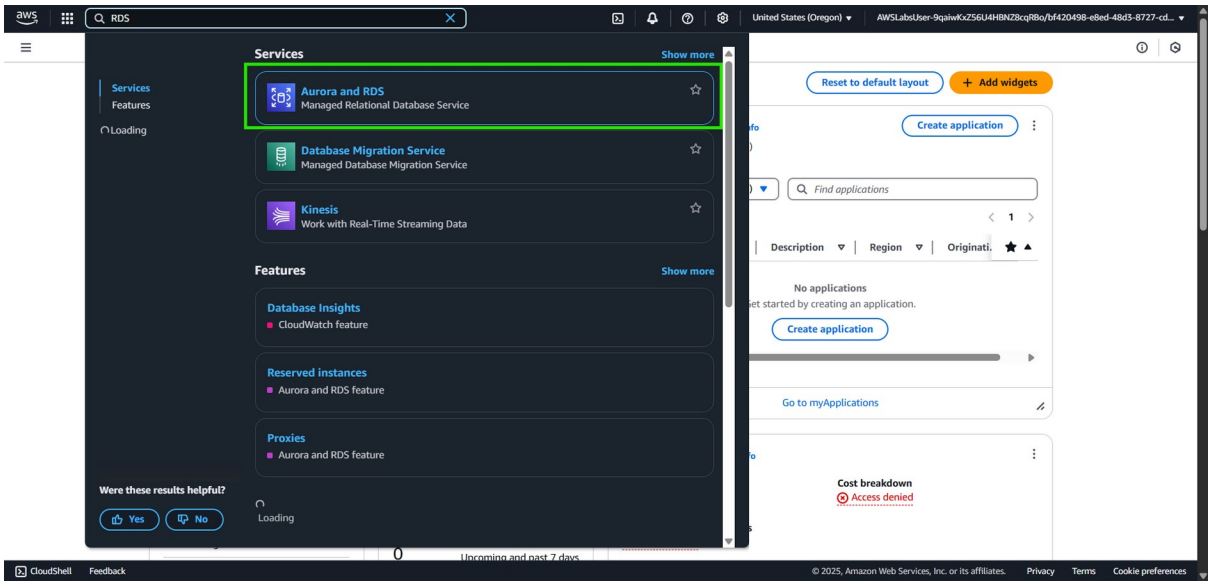
In this challenge task, you create a cross-Region read replica from the source DB instance. You create a read replica in a different AWS Region to improve your disaster recovery capabilities, scale read operations into an AWS Region closer to your users, and to make it easier to migrate from a data center 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 can complete this task or skip to the end of the lab here.

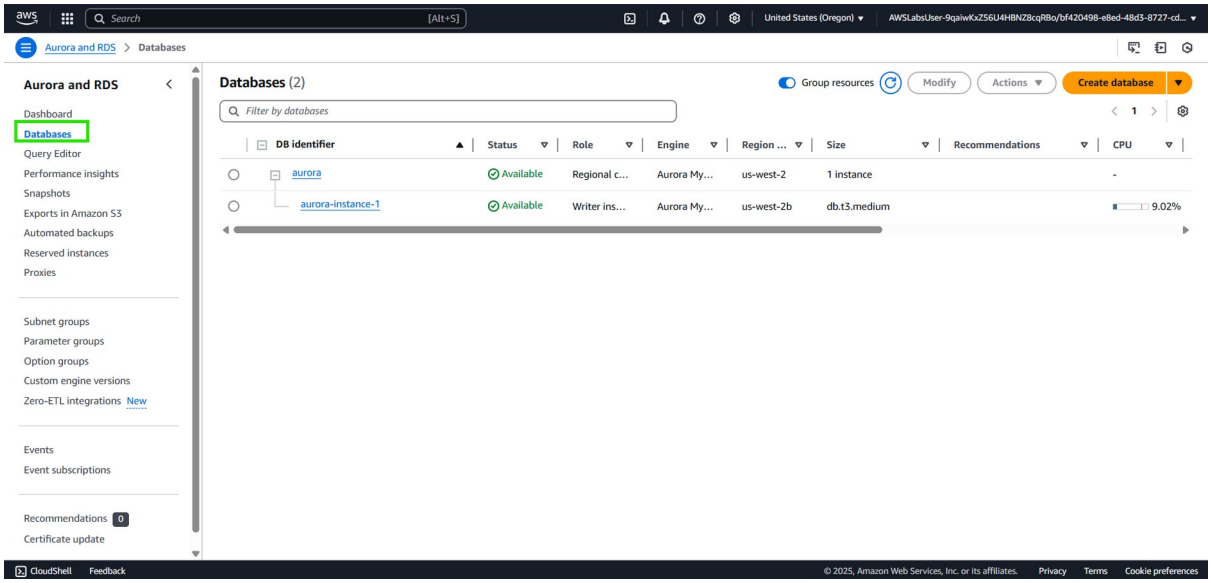
50. Switch back to the browser tab open to the AWS Management Console.



51. At the top of the console, in the search bar, search for and choose RDS.

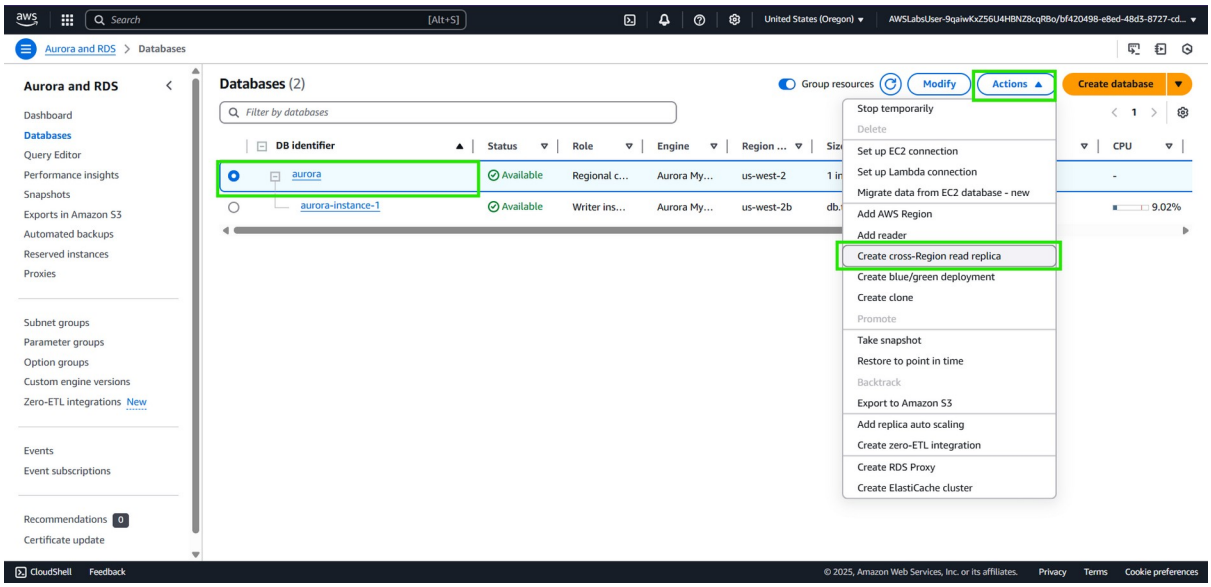


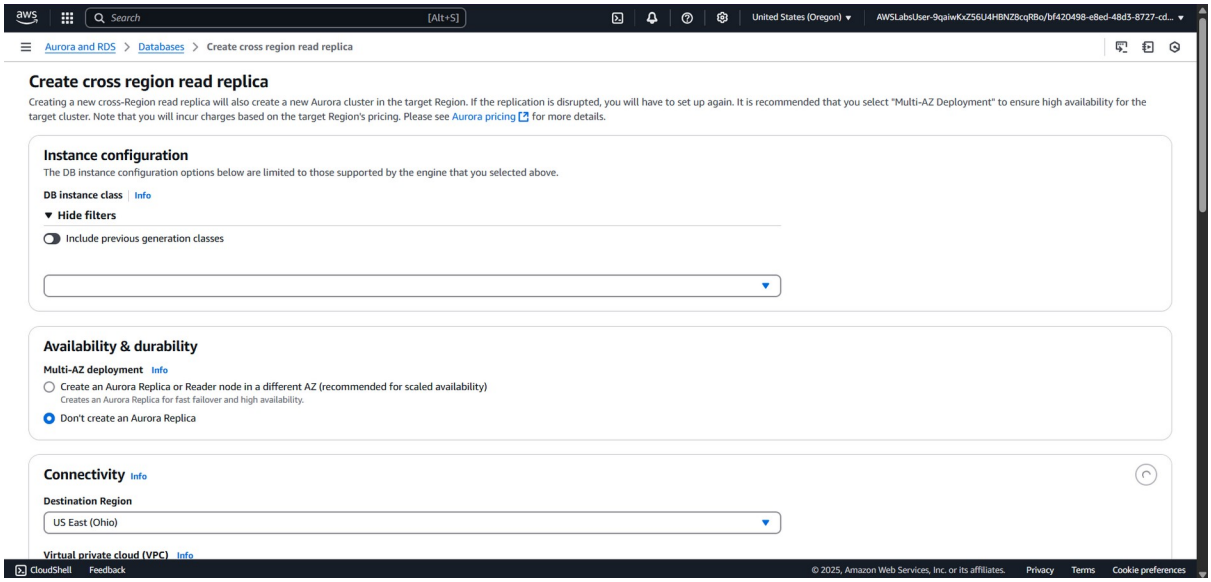
52. In the left navigation pane, choose **Databases**.



53. Select **aurora** DB instance as the source for a read replica.

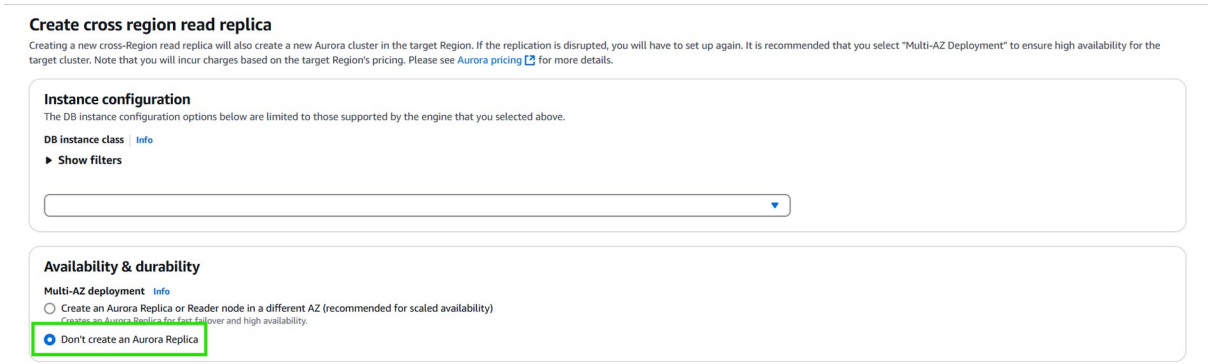
54. Choose Actions and select **Create cross-Region read replica**.





The **Create cross region read replica** page is displayed.

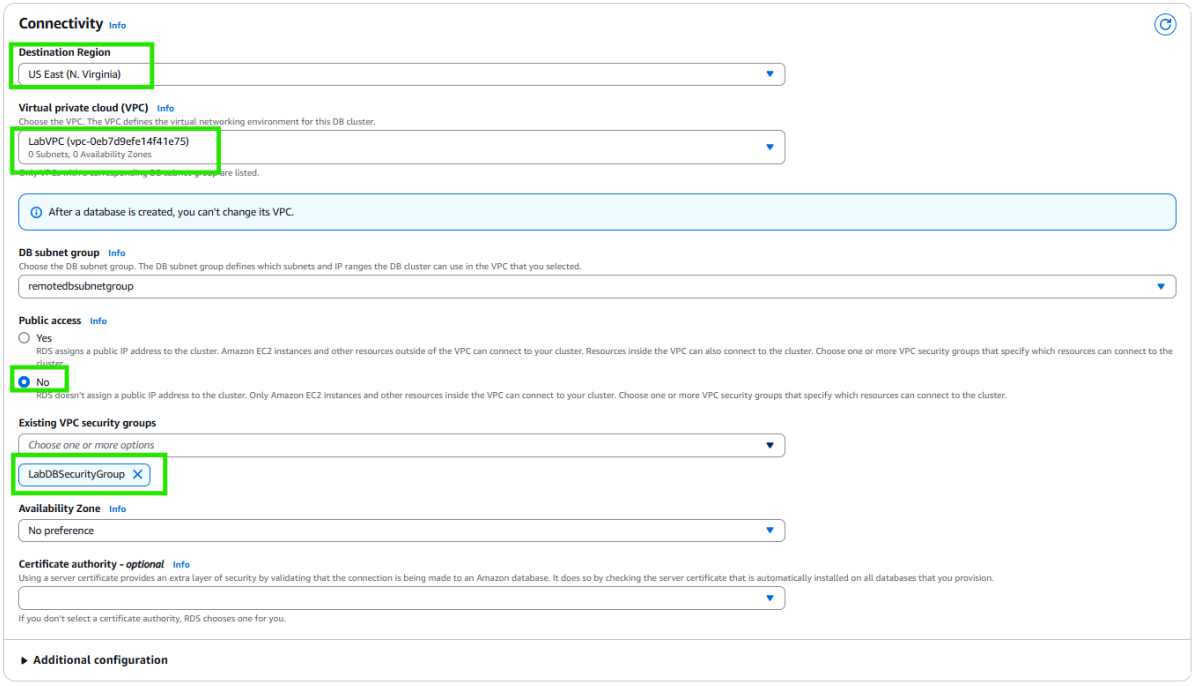
For **Multi-AZ deployment**: Select **Don't create an Aurora Replica**.



The remaining settings in this section can be left at their default values.

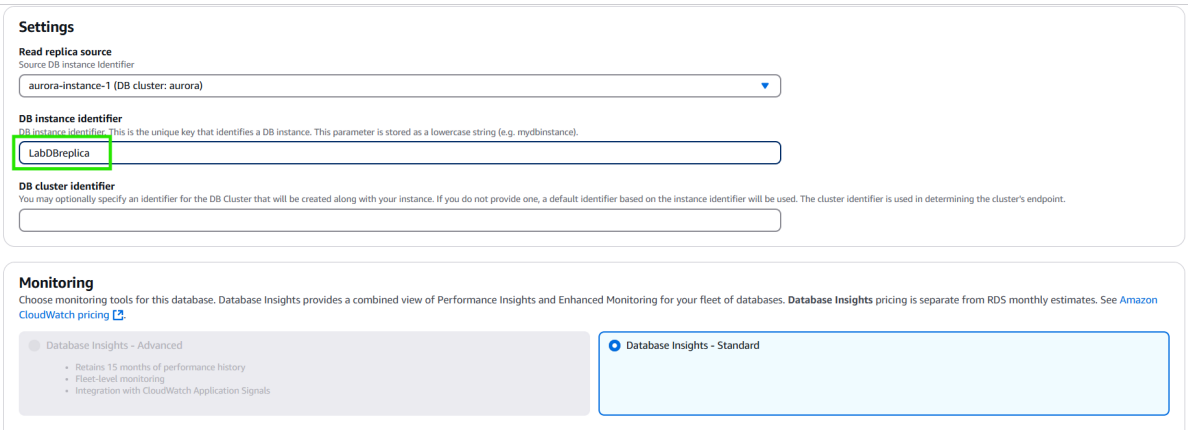
55. In the **Connectivity** section, configure the following:

    - **Destination Region:** From the dropdown menu, select the region that matches the **RemoteRegion (US East (N. Virginia))** value from the lab instructions.
    - **Virtual private cloud (VPC):** *LabVPC*
    - **Public access:** Select **No**.
    - For **Existing VPC security groups:**
        - To remove the *default* security group, select the **X**.
        - From the dropdown menu, enter LabDBSecurityGroup to choose this option. The remaining settings in this section can be left at their default values.

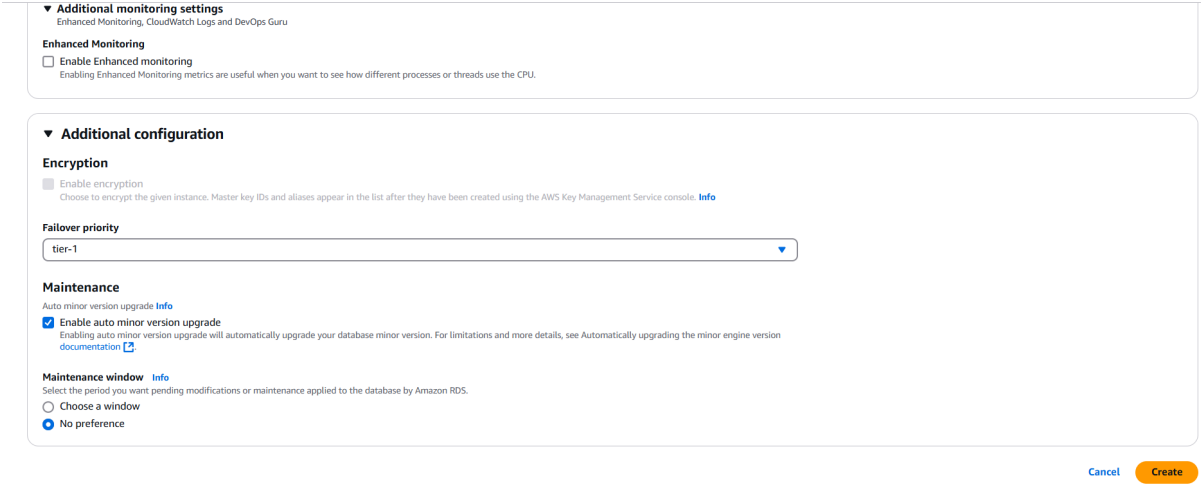


56. In the **Settings** section, configure the following:

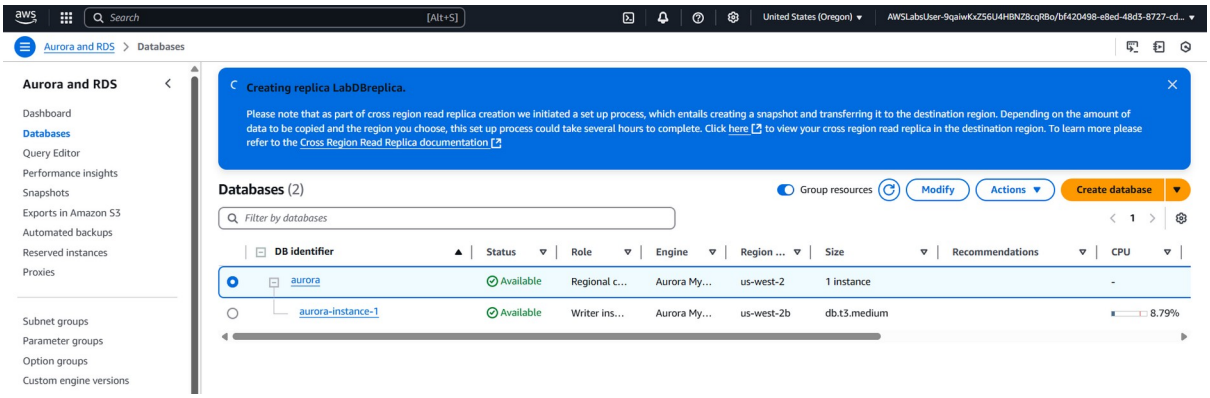
    - **DB instance identifier:** Enter LabDBreplica.



The remaining settings in this section can be left at their default values.

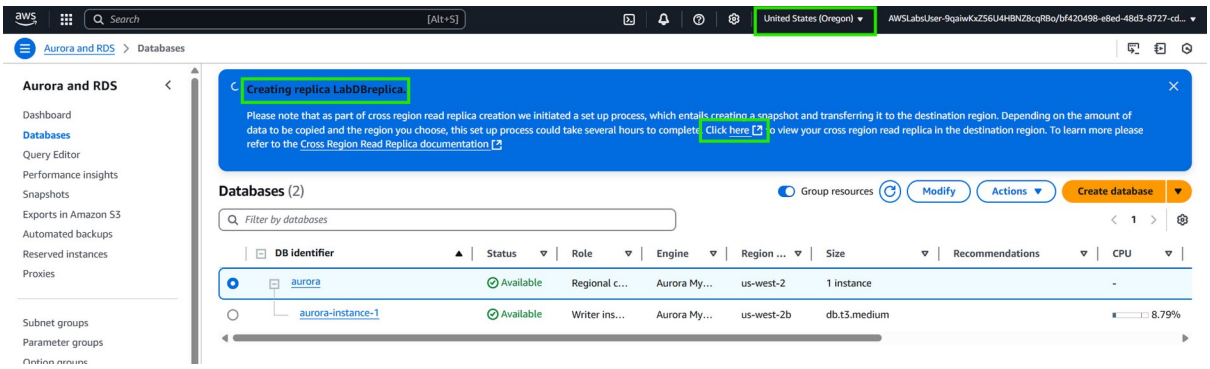


57. Choose **Create**.

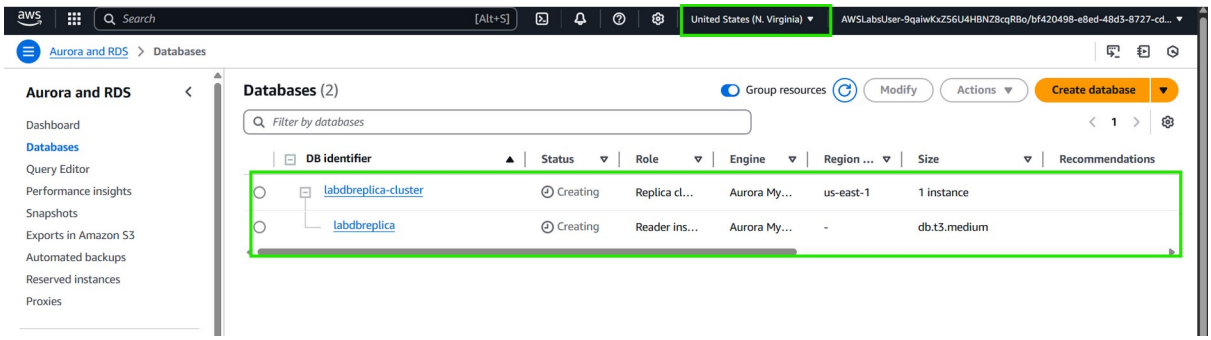


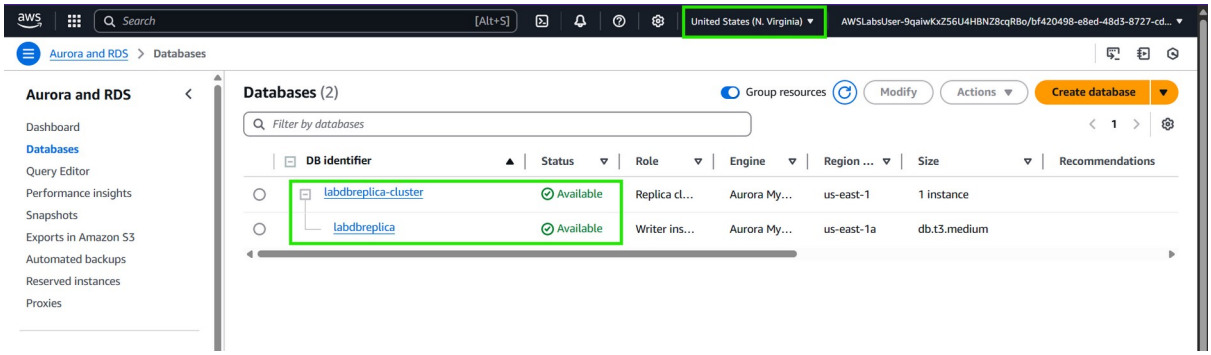
A **(Creating replica LabDBreplica.)** message is displayed on the screen.

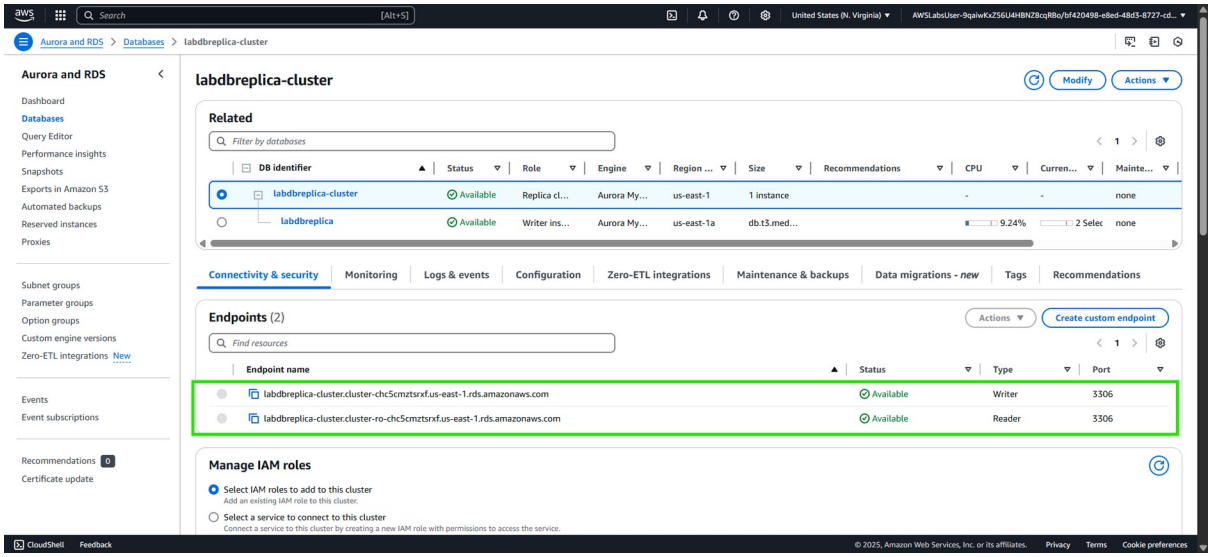
58. To review the cross-Region read replica in the destination region, choose the hyperlink as present in the below image.



59. Otherwise, choose **Close**.







Congratulations! You have successfully completed the optional task and started the creation of a cross-Region read replica for the Amazon RDS database.

Conclusion

Congratulations! You have now successfully completed the following:

- Created an Amazon RDS DB instance.
- Created an Application Load Balancer.
- Created an HTTP listener for the Application Load Balancer.
- Created a target group.
- Registered targets with a target group.
- Tested the load balancer and the application connectivity to the database.

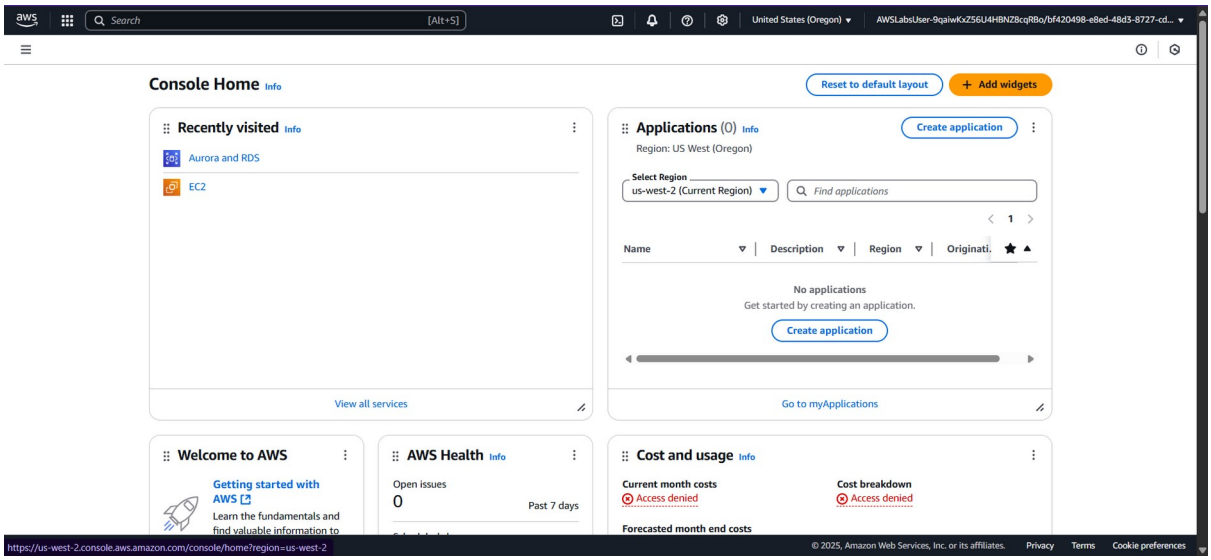
- Reviewed the Amazon RDS DB instance metadata using the console.

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.

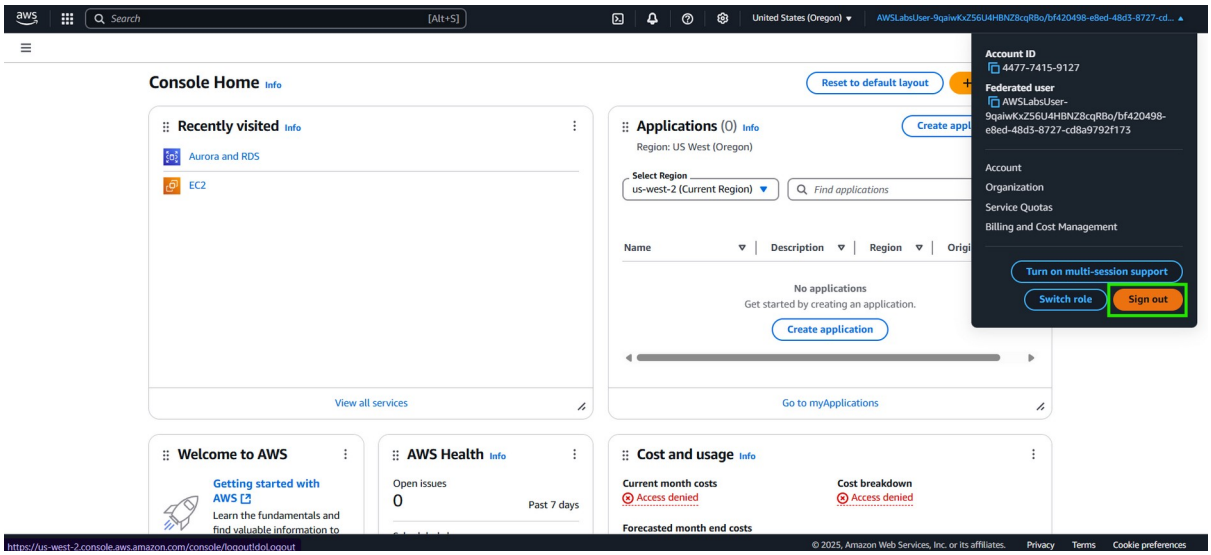
End lab

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

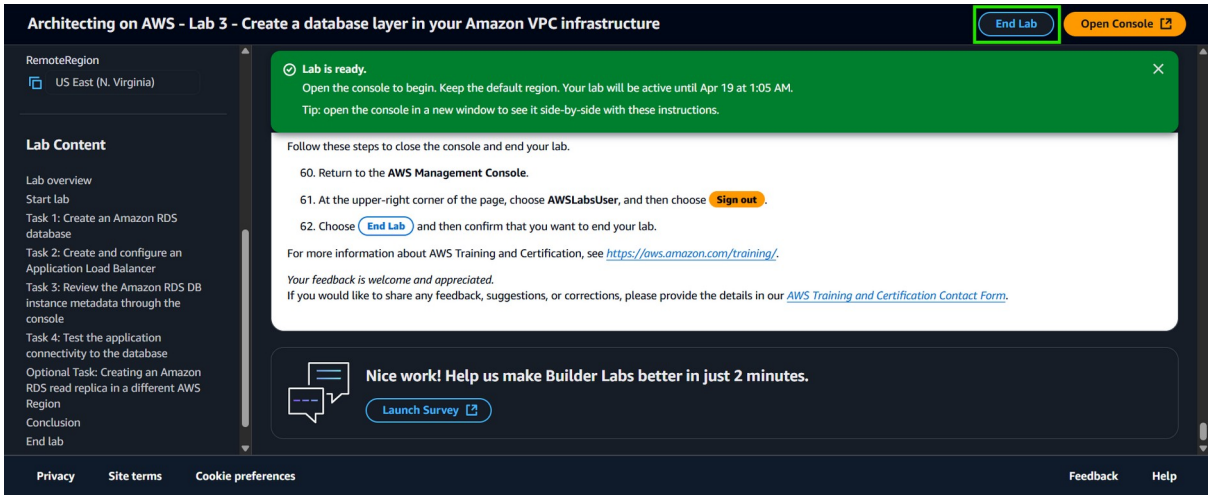
60. Return to the **AWS Management Console**.



61. At the upper-right corner of the page, choose **AWSLabsUser**, and then choose **Sign out**.



62. Choose **End Lab** and then confirm that you want to end your lab.



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*.