aws training and certification

Lab 7: Capstone Lab

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

You are tasked with applying your new knowledge to solve several architectural challenges within a specific business case. First, you are given a list of requirements related to the design. Then, you perform a series of actions to deploy and configure the services needed to meet the requirements.

During this capstone lab, you have access to the following:

- Downloadable AWS CloudFormation templates and other lab files
- Task scenarios, descriptions, and requirements
- Optional step-by-step instructions, if needed

The task scenarios provide relevant background and help you understand how the requirements solve a real-world business problem. Use the templates and the requirements list to complete all of the tasks in the capstone. Now that you are familiar with concepts and services, this lab solidifies your knowledge through practice. In the real world, you encounter problems that are not well-defined or sequenced logically. By the end of this capstone, you should have a better understanding of how you can apply architectural best practices to real-world problems.

OBJECTIVES

After completing this lab, you should be able to do the following:

- Deploy a virtual network spread across multiple Availability Zones in a Region using a CloudFormation template.
- Deploy a highly available and fully managed relational database across those Availability Zones (AZ) using Amazon Relational Database Service (Amazon RDS).
- Use Amazon Elastic File System (Amazon EFS) to provision a shared storage layer across multiple Availability Zones for the application tier, powered by Network File System (NFS).
- Create a group of web servers that automatically scales in response to load variations to complete your application tier.

DURATION

This lab requires approximately *90* minutes to complete.

ICON KEY

Various icons are used throughout this lab to call attention to different types of instructions and notes. The following list explains the purpose for each icon:

- **Expected output:** A sample output that you can use to verify the output of a command or edited file.
- **Note:** A hint, tip, or important guidance.
- **Learn more:** Where to find more information.
- **Security:** An opportunity to incorporate security best practices.
- **Refresh:** A time when you might need to refresh a web browser page or list to show new information.
- **Copy edit:** A time when copying a command, script, or other text to a text editor (to edit specific variables within it) might be easier than editing directly in the command line or terminal.
- **Hint:** A hint to a question or challenge.

Start lab

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

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.

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:

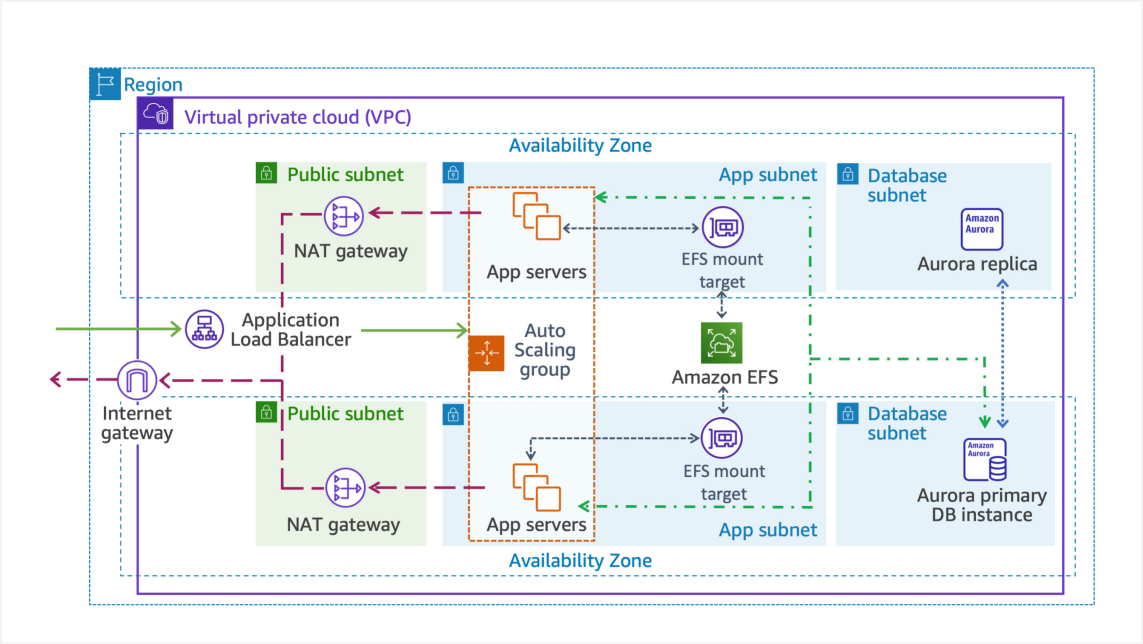
- Add the lab domain name to your pop-up or script blocker's allow list or turn it off.
- Refresh the page and try again.

LAB SCENARIO

Example Corp. creates marketing campaigns for small-sized to medium-sized businesses. They recently hired you to work with the engineering teams to build out a proof of concept for their business. To date, they have hosted their client-facing application in an on-premises data center, but they recently decided to move their operations to the cloud in an effort to save money and transform their business with a cloud-first approach. Some members of their team have cloud experience and recommended the AWS Cloud services to build their solution.

In addition, they decided to redesign their web portal. Customers use the portal to access their accounts, create marketing plans, and run data analysis on their marketing campaigns. They would like to have a working prototype in two weeks. You must design an architecture to support this application. Your solution must be fast, durable, scalable, and more cost-effective than their existing on-premises infrastructure.

The following image shows the final architecture of the designed solution:



AWS SERVICES NOT USED IN THIS LAB

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

Note: This lab includes two sets of instructions.

- **Section 1: High-level instructions** provide students with a limited set of high-level instructions, a list of requirements and configurations, and hints.
- **Section 2: Detailed instructions** provide students with step-by-step instructions for each task.

Students are strongly encouraged to attempt the lab only using the **high-level instructions** .

Section 1: High-level instructions

In this section, all of the tasks list the requirements and configurations that are required to build the solution. Try building the solution using this section only. You can always refer to **Section 2** for detailed instructions when you experience challenges.

Task 1: Reviewing and running a preconfigured CloudFormation template

In this task, you design and deploy a highly available and scalable deployment of the WordPress application stack. You use a CloudFormation template to deploy the networking resources required to support the application.

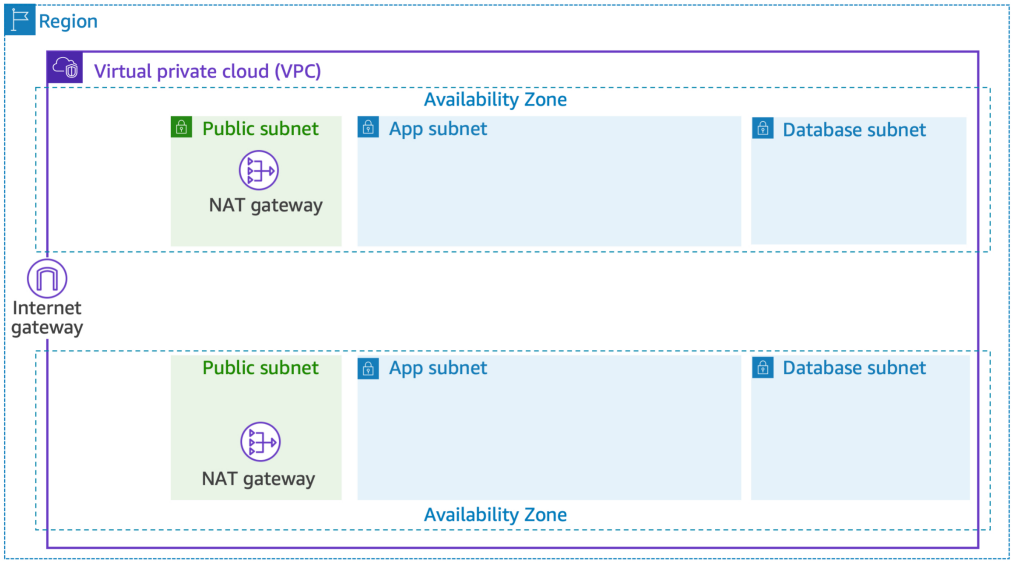
TASK 1: SCENARIO

The company wants to deploy a new web application on WordPress on their existing web hosting account. The first step is to gather the requirements for the architecture design. To help you get started, the network team gave you a list of requirements to ensure there are no issues with the existing landscape. This includes the Classless Inter-Domain Routing (CIDR) range for the Amazon Virtual Private Cloud (Amazon VPC). Using this address range, build a VPC with two public subnets, two private application subnets, and two private database subnets. Make sure you attach an internet gateway to the VPC with a NAT gateway for routing traffic. They also requested two Elastic IP addresses. Ensure that you associate the following:

- The public subnet with the internet gateway
- The private application subnet and private database subnet with the NAT gateway

The network team also provided the route tables needed with their subnet associations.

Your cloud engineer has transferred the information request to a CloudFormation template and set up the security groups and outputs needed for future deployments. Please review the CloudFormation template with your cloud engineer, deploy the template, and check back with the network team to validate that the build meets all of their requirements.



TASK 1: SUB-TASKS

- Download and review the template file to understand the infrastructure being deployed.
- Upload the template file to create the CloudFormation stack.
- View created resources from the console.

TASK 1: REQUIREMENTS AND CONFIGURATION

- Open the context (right-click) menu on this Task1.yaml link and choose the option to save the CloudFormation template to your computer.
- Open the downloaded file in a text editor (not a word processor).
- Upload the template file in CloudFormation.
- Use the default values in this task.

Note: You can save this CloudFormation template for future use. If you choose to build it on your AWS account in the future, tailor the configuration to your requirements.

TASK 1: PROCEED TO THE AWS MANAGEMENT CONSOLE

Now that you have your first task requirements, go ahead and build. Instructions are available in **Section 2** if you experience challenges, but it's recommended you try to build first without them.

Congratulations! You learned to configure the stack and created all of the resources using the provided CloudFormation template.

Task 2: Creating an Amazon RDS database

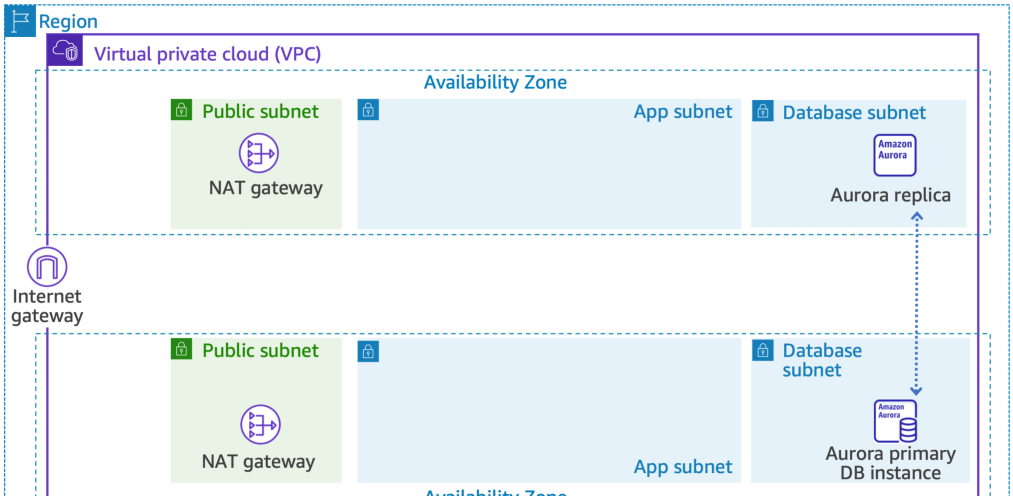
In this task, you create an Amazon Aurora DB instance and the subnet group to support it.

TASK 2: SCENARIO

The network team verified the previous deployment requirements and everything looks good. The next phase of the project is to establish a secure backend database to give the database administrator (DBA) access for any migration operations before go-live.

The current database requires a lot of administrative overhead and the business has agreed to move to a managed database service. They want their architecture to be highly available, so you recommend that they set up a Multi-AZ RDS Aurora database layer. After reviewing the proposed design, the DBA has outlined the database requirements. Because of previous performance issues, the database does not require encryption, and you agree that this is the best choice.

The existing monitoring solution polls data every 10 minutes. The engineering team doesn't have room in the budget for additional features; therefore, enhanced monitoring is not required.



TASK 2: SUB-TASKS

- Create a new DB subnet group.
- Create a new Aurora database.
- Copy the database metadata.

TASK 2: REQUIREMENTS AND CONFIGURATION

DB subnet group

- As your first step, create a DB subnet group from the Amazon RDS console.
- Place the subnet group in the **Lab VPC** .
- Choose Availability Zone A and Availability Zone B.
- Choose the subnet with 10.0.4.0/24 CIDR block for the DB Subnet in Availability Zone A.
- Choose the subnet with 10.0.5.0/24 CIDR block for the DB Subnet in Availability Zone B.

Amazon Aurora database

- The database instances need to be **Amazon Aurora with MySql compatible** and **burstable-performance** DB instance.
- The database instance size needs to be **db.t3.medium** .
- The database layer needs to be deployed into the **LabVPC** .
- The subnet group needs to be configured with the **Database subnets** only.
- The database needs to be configured with the **RDS Security Group** set up by your security team only.
- Make sure to provide an **Initial database name** , such as `WPDatabase` .
- Make sure to provide a **DB cluster identifier** , such as `MyDBCluster` .
- Turn off **Encryption** .
- Turn off **Enhanced monitoring** .
- Turn off **Enable auto minor version upgrade** .
- Turn off **Enable deletion protection** .

Note: Take note of the following because you need them later:

- **Master username** and **Master password**
- **Initial database name**
- **Writer endpoint**

Note If you notice the error “Failed to turn on DevOps Guru for mydbcluster-instance-x because of missing permissions” you can safely ignore the error.

TASK 2: PROCEED TO THE AWS MANAGEMENT CONSOLE

Now that you have the requirements, go ahead and build. Instructions are available in **Section 2** if you experience challenges, but it’s recommended you try to build first without them.

Congratulations! You have created the Amazon Aurora database.

Task 3: Creating an Amazon EFS file system

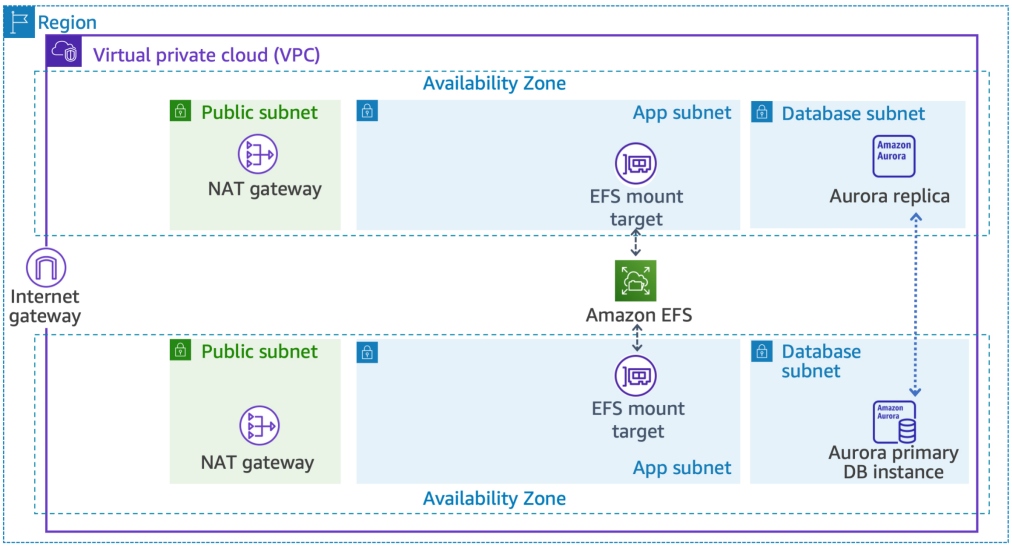
In this task, you create and configure a custom Amazon EFS file system.

TASK 3: SCENARIO

Example Corp. is having issues with the lead time on ordering new hardware. This is slowing down their ability to onboard new customers and expand the business. The SysOps team has a request for a storage solution built for the cloud. They need to be able to confirm that the backup policies and encryption settings meet their internal and regulatory compliance requirements. Managing time, cost, and compliance gives Example Corp. a competitive advantage.

You recommend Amazon EFS to the business as a simple, serverless, set-and-forget, elastic file system. With Amazon EFS, they can share data securely, without provisioning or managing storage.

Your task is to create an Amazon EFS file system that meets the SysOps team’s requirements.



TASK 3: SUB-TASKS

- Create a new EFS file system.
- Copy the EFS file system metadata.

TASK 3: REQUIREMENTS AND CONFIGURATION

Customize the creation of an Amazon EFS file system:

- It must have **Regional** availability and durability.
- It needs to be **General Purpose** performance to control costs.
- It needs to have **Bursting** throughput to scale the system size.
- Turn off **Automatic backups** and **Encryption** of data at rest.
- EFS needs to be deployed in **Lab VPC** .
- Select **Availability Zone ending in "a"** and **Availability Zone ending in "b"** .
- Assign **AppSubnet1** and **AppSubnet2** as subnet IDs.
- Remove the default security group and select **EFSMountTargetSecurityGroup** .
- It does not need a file system policy created at this time.

Note: Take note of the **File system ID** . You need it later.

TASK 3: PROCEED TO THE AWS MANAGEMENT CONSOLE

Now that you have the requirements, go ahead and build. Instructions are available in **Section 2** if you experience challenges, but it's recommended you try to build first without them.

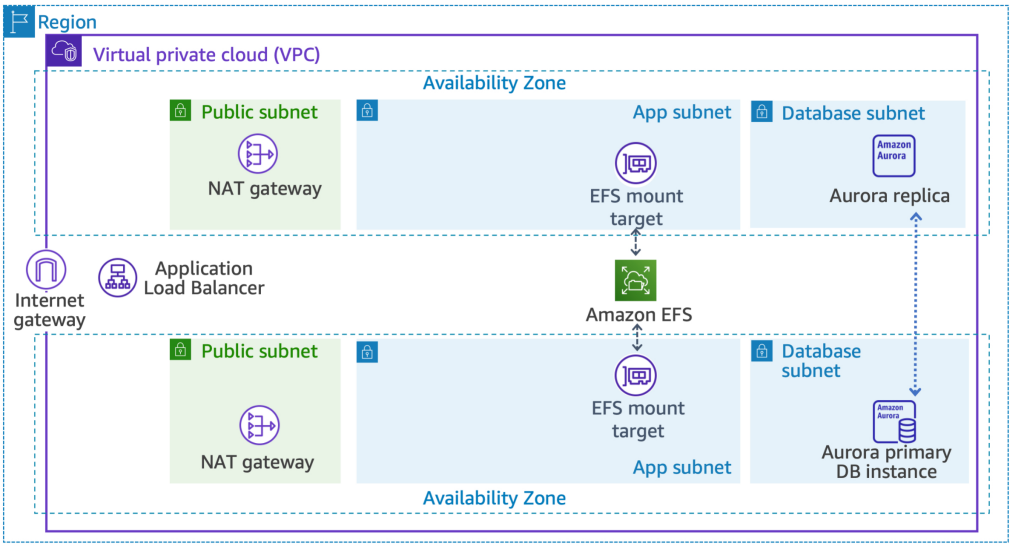
Congratulations! You have created the Amazon EFS file system.

Task 4: Creating an Application Load Balancer

In this task, you create the Application Load Balancer and a target group.

TASK 4: SCENARIO

The SysOps team is excited about the new Amazon EFS configuration and eager to address their next pain point. The current application experiences frequent outages because of variable, unexpected traffic loads. The SysOps team wants to know if AWS has a service to address this issue that can be used at the application layer (Layer 7). On further investigation, you recognize the need for an Application Load Balancer for the application servers in the private subnet. Your task is to deploy and configure an Application Load Balancer with a health check, and register the required targets.



TASK 4: SUB-TASKS

- Create an Application Load Balancer.
- Copy the Application Load Balancer metadata.

TASK 4: REQUIREMENTS AND CONFIGURATION

The Application Load Balancer **Target group** should be configured to use the *LabVPC* for load balancing instances with the following properties:

- **Health check path** : `/wp-login.php`
- **Advanced health check settings** :
  - **Healthy threshold** : `2`
  - **Unhealthy threshold** : `10`
  - **Timeout** : `50`
  - **Interval** : `60`
- Leave all other settings at their default values.

The **Application Load Balancer** needs to be configured in the *LabVPC* and in the *public subnets* with the following properties:

- **Subnets** : Select all Public subnets.
- **SecurityGroups** : Choose **AppInstanceSecurityGroup** .
- **Listener** : Forward to the created target group.

Note: Take note of the Application Load Balancer's **DNS name** . You need it later.

TASK 4: PROCEED TO THE AWS MANAGEMENT CONSOLE

Now that you have the requirements, go ahead and build. Instructions are available in **Section 2** if you experience challenges, but it's recommended you try to build first without them.

Congratulations! You have created the target group and an Application Load Balancer.

Task 5: Creating a launch template using CloudFormation

In this task, you use a CloudFormation template to deploy the WordPress user data within an Amazon Elastic Compute Cloud (Amazon EC2) Auto Scaling launch template. The template includes the Amazon EFS mount points and the Aurora configurations.

TASK 5: SCENARIO

Up to this point, you created the underlying network resources, the database, an Amazon EFS file system, and an Application Load Balancer. It's time to put it all together. Example Corp. already has a WordPress account, so the cloud engineer uses the settings and configuration from their existing environment to create a CloudFormation template. This includes all of the new resources you provisioned to set up a launch template.

Review the CloudFormation template and create all of the necessary resources. Pay special attention to the user data script. Check for any errors in the deployment and address those, if needed. After that is completed, you can create the application servers to support traffic to the new environment.

TASK 5: SUB-TASKS

- Obtain and review the CloudFormation template.
- Upload the template to create the CloudFormation stack.
- View created resources from the console.
- View the created launch template.

TASK 5: REQUIREMENTS AND CONFIGURATION

- Open the context (right-click) menu on this Task5.yaml link, and choose the option to save the CloudFormation template to your computer.

Note: You can save this CloudFormation template for future use. If you choose to build it on your AWS account in the future, tailor the configuration to your requirements.

- Open the downloaded file in a text editor (not a word processor).
- Upload the template file in CloudFormation.
- Using the values saved from previous tasks, update the following Parameters values at the time of stack creation:
  - **DB Name** : Paste the *initial database name* (from Task 2). **Note:** Make sure that you paste the *initial database name*, not the cluster name.
  - **Database endpoint** : Paste the *Writer endpoint* (from Task 2).
  - **Database User Name** : Paste the *Master username* (from Task 2).
  - **Database Password** : Paste the *Master password* (from Task 2).
  - **WordPress admin username** : Defaults to `wpadmin` .
  - **WordPress admin password** : Paste the **LabPassword** value from the left side of these lab instructions.
  - **WordPress admin email address** : Input a valid email address.
  - **Instance Type** : Defaults to `t3.medium` .
  - **ALBDnsName** : Paste the *DNS name* value (from Task 4).
  - **LatestAL2AmiId** : Leave the default value.
  - **WPElasticFileSystemID** : Paste the *File system ID* value (from Task 3).

TASK 5: PROCEED TO THE AWS MANAGEMENT CONSOLE

Now you have the requirements, go ahead and build the next phase of the project. Instructions are available in **Section 2** if you experience challenges, but it's recommended you try to build first without them.

Congratulations! You have created the stack using the provided CloudFormation template.

Task 6: Creating the application servers by configuring an Auto Scaling group and a scaling policy

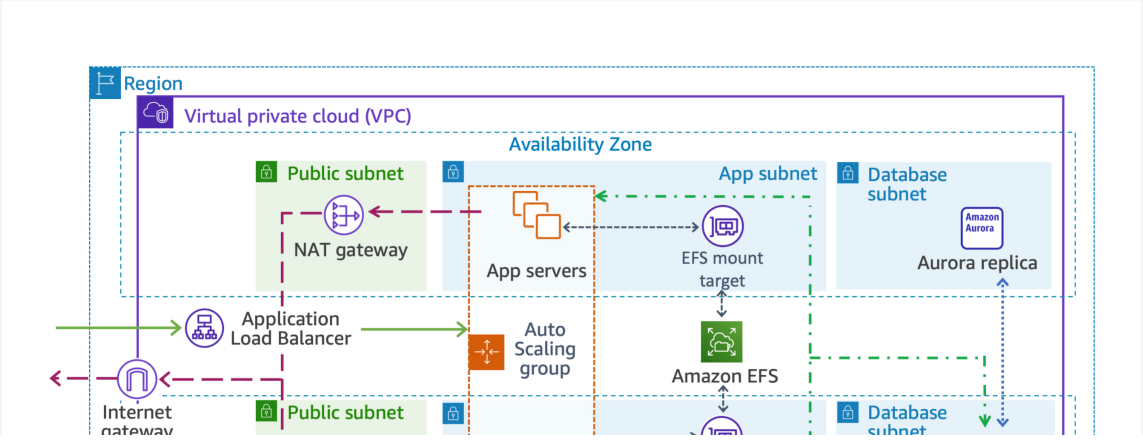
In this task, you create the WordPress application servers by configuring an Auto Scaling group and a scaling policy.

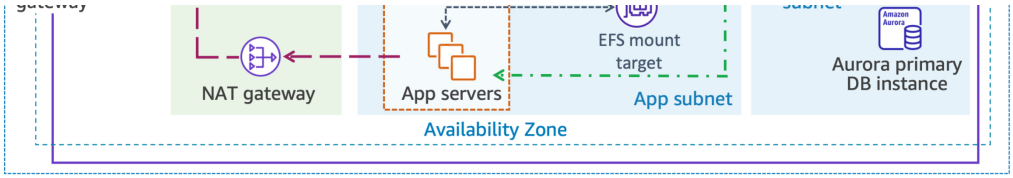
TASK 6: SCENARIO

Now that your template has been deployed, it's time to create the application servers and auto scaling mechanism. In the previous task, you chose to implement auto scaling for the app servers to meet the scaling requirements of the project plan.

Create an Auto Scaling group and scaling policy. Verify that the instance status is healthy, and test the load balancer availability. Hand the environment over to the engineering teams to validate full functionality when the unit test is complete, and ask them for feedback.

When the team is satisfied, ask them to migrate the WordPress site to the AWS environment and test the app functionality. A common practice is to introduce examples of failure. If the app is working as intended, you would introduce some examples of failure, for example, delete an app server or roll back the database to a recent backup. This step is not a part of this lab.





TASK 6: SUB-TASKS

- Create an Auto Scaling group.
- Verify the target groups and test the application server through the load balancer.
- Log in to WordPress and explore.

TASK 6: REQUIREMENTS AND CONFIGURATION

- Use the **Launch template** from the previous task.
- Use the **LabVPC** for network settings.
- Use the Application subnets in both Availability Zones.
- Use the existing load balancer and target groups created earlier.
- Use the Elastic Load Balancing (ELB) health check and set the health check grace period to **300** seconds.
- Turn on **Enable group metrics collection within CloudWatch** .
- Configure the group size:
  - **Desired capacity** : 2
  - **Minimum capacity** : 2
  - **Maximum capacity** : 4
- Configure **Scaling policies** with **Target tracking scaling policy** and use the defaults.
- Notifications are out of scope for the lab and do not need to be configured.
- Verify that the **Auto Scaling group** has launched your EC2 instances.
- Test the application using the Application Load Balancer DNS name.

TASK 6: PROCEED TO THE AWS MANAGEMENT CONSOLE

Now that you have the final task requirements, go ahead and build to complete your project! Instructions are available in **Section 2** if you experience challenges, but it's recommended you try to build first without them.

Congratulations! You have created the AWS Auto Scaling group and successfully launched the WordPress application.

Section 2: Detailed instructions

In this section, all of the tasks list step-by-step instructions for reference if you experience challenges while building the solution on your own. Have you tried **Section 1** to build the solution first?

Task 1 instructions: Reviewing and running a preconfigured CloudFormation template

TASK 1.1: NAVIGATING TO THE CLOUDFORMATION CONSOLE

3. At the top of the AWS Management Console, in the search box, search for and choose `CloudFormation` .

TASK 1.2: OBTAINING AND REVIEWING THE CLOUDFORMATION TEMPLATE

4. Open the context (right-click) menu on this Task1.yaml link, and choose the option to save the CloudFormation template to your computer.

5. Open the downloaded file in a text editor (not a word processor).

6. Review the CloudFormation template.

7. Predict what resources are created by this template.

TASK 1.3: CREATING THE CLOUDFORMATION STACK

8. Choose **Create stack** .

Note: If the console starts you on the Stacks page instead of the Amazon CloudFormation landing page, then you can get to the Create stack page in two steps.

- Choose the Create stack dropdown menu.
- Choose With new resources (standard) .

The **Create stack** page is displayed.

9. Configure the following:

- Select Template is ready.
- Select Amazon S3 URL.
- Copy the **Task1TemplateUrl** value from the left side of these lab instructions and paste it in the **Amazon S3 URL** text box.
- Choose **Next** .

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

10. **Stack name** : Enter `VPCStack` .

11. **Parameters** : Keep the default values.

12. Choose **Next** .

The **Configure stack options** page is displayed. You can use this page to specify additional parameters. You can browse the page, but leave settings at the default values.

13. Choose **Next** .

The **Review VPCStack** page is displayed. This page is a summary of all settings.

14. At the bottom of the page, choose **Submit** .

The **stack details** page is displayed.

The stack enters the CREATE_IN_PROGRESS status.

15. Choose the **Stack info** tab.

16. Occasionally choose the **Overview** refresh .

17. Wait for the status to change to CREATE_COMPLETE .

Note: This stack can take up to 5 minutes to deploy the resources.

TASK 1.4: VIEWING CREATED RESOURCES FROM THE CONSOLE

18. Choose the **Resources** tab.

The list shows the resources that are being created. CloudFormation determines the optimal order for resources to be created, such as creating the VPC before the subnet.

19. Review the resources that were deployed in the stack.

20. Choose the **Events** tab and scroll through the list.

The list shows (in reverse order) the activities performed by CloudFormation, such as starting to create a resource and then completing the resource creation. Any errors encountered during the creation of the stack are listed in this tab.

21. Choose the **Outputs** tab.

22. Review the key-value pairs in the Outputs section. These values might be useful in later lab tasks.

Congratulations! You have learned to configure the stack and created all of the resources using the provided CloudFormation template.

Task 2 instructions: Creating an Amazon RDS database

TASK 2.1: NAVIGATING TO THE AMAZON RDS CONSOLE

23. At the top of the AWS Management Console, in the search box, search for and choose `RDS` .

The **Amazon RDS console** page is displayed.

TASK 2.2: CREATING A NEW DB SUBNET GROUP

24. In the left navigation pane, choose **Subnet groups** .

25. Choose **Create DB subnet group** .

The **Create DB subnet group** page is displayed.

26. In the **Subnet group details** section, configure the following:

- **Name** : Enter `AuroraSubnetGroup` .
- **Description** : Enter `A 2 AZ subnet group for my database` .
- **VPC** : Select **LabVPC** from the dropdown menu.

27. In the **Add subnets** section, configure the following:

- From the **Availability Zones** dropdown menu:
  - Select the Availability Zone ending in **a** .
  - Select the Availability Zone ending in **b** .
- From the **Subnets** dropdown menu:
  - Select the subnet with the CIDR block 10.0.5.0/24 from the Availability Zone ending in **b** .
  - Select the subnet with the CIDR block 10.0.4.0/24 from the Availability Zone ending in **a** .

28. Choose **Create** .

A Successfully created AuroraSubnetGroup. View subnet group message is displayed on top of the screen.

TASK 2.3: CREATING A NEW AMAZON AURORA DATABASE

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

30. Choose **Create database** .

The **Create database** page is displayed.

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

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

- In **Engine type** , select **Aurora(MySQL Compatible)** .

33. In the **Templates** section, select **Production** .

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

- **DB cluster identifier** : Enter `MyDBCluster` .
- **Master username** : Enter `admin` .
- **Master password** : Paste the **LabPassword** value from the left side of these lab instructions.
- **Confirm master password** : Paste the **LabPassword** value from the left side of these lab instructions.

Note: Remember your credentials.

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

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

36. In the **Availability & durability** section, for **Multi-AZ deployment** , select **Create an Aurora Replica or Reader node in a different AZ (recommended for scaled availability)** .

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

- **Virtual private cloud (VPC)** : Select **LabVPC** from the dropdown menu.
- **DB subnet group** : Select **aurorasubnetgroup** from the dropdown menu.
- **Public access** : Select **No** .
- **VPC security group (firewall)** : Select **Choose existing** .
- **Existing VPC security groups** :

- Select **xxxxx-RDSSecurityGroup-xxxxx** from the dropdown menu.
- To remove the default security group, choose the **X** .

- Expand the ▸ **Additional configuration** section and configure the following:
  - **Database port** : Leave the configuration at the default value.

38. In the **Monitoring** section, deselect ☐ **Enable Enhanced monitoring** .

39. Scroll to the bottom of the page and expand the main ▸ **Additional configuration** section.

- In the **Database options** section, configure the following:
  - **Initial database name** : Enter `WPDatabase` .
- In the **Encryption** section, deselect ☐ **Enable encryption** .
- In the **Maintenance** section, deselect ☐ **Enable auto minor version upgrade** .
- In the **Deletion protection** section, deselect ☐ **Enable deletion protection** .

40. Scroll to the bottom of the screen and choose **Create database** .

41. On the **Suggested add-ons for mydbcluster** pop-up window, choose **Close** .

Note: Your Aurora MySQL DB cluster is in the process of launching. The cluster you configured consists of two instances, each in a different Availability Zone. The Amazon Aurora DB cluster can take up to 5 minutes to launch. Wait for the mydbcluster status to change to *Available*. You do not have to wait for the availability of the instances to continue.

A ✓ Successfully created database mydbcluster message is displayed on top of the screen.

42. Choose View connection details displayed on the success message border to save the connection details of your **mydbcluster** database to a text editor.

Note If you notice the error "Failed to turn on DevOps Guru for mydbcluster-instance-x because of missing permissions" you can safely ignore the error.

TASK 2.4: COPYING DATABASE METADATA

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

44. Choose the **mydbcluster** link.

45. Choose the **Connectivity & security** tab.

46. Copy the **endpoint** value for the **Writer** instance to a text editor.

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

47. Choose the **Configuration** tab.

48. Copy the **Master username** value to a text editor.

49. For **Master password** : Use the **LabPassword** value from the left side of these lab instructions.

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

51. Choose the **mydbcluster-instance-x** writer instance link.

52. Choose the **Configuration** tab.

53. Copy the **DB name** value to a text editor.

Congratulations! You have created the Amazon Aurora database.

Task 3 instructions: Creating an Amazon EFS file system

TASK 3.1: NAVIGATING TO THE AMAZON EFS CONSOLE

54. At the top of the AWS Management Console, in the search box, search for and choose `EFS` .

The **Amazon Elastic File System console** page is displayed.

TASK 3.2: CREATE A NEW FILE SYSTEM

55. Choose **Create file system** .

The **Create file system** page is displayed.

56. Choose Customize .

The **File system settings** page is displayed.

57. In the **General** section, configure the following:

- **Name -** *optional* : Enter `myWPEFS` .
- Deselect ☐ **Enable automatic backups** .
- Deselect ☐ **Enable encryption of data at rest** .
- In the **Tags - optional** section, configure the following:
  - **Tag key** : Enter `Name` .
  - **Tag value –** *optional* : Enter `myWPEFS` .
  - Leave all other settings at their default value.

58. Choose **Next** .

The **Network access** page is displayed.

59. From the **Virtual Private Cloud (VPC)** dropdown menu, select `LabVPC` .

60. For **Mount targets** , configure the following:

- **Availability zone** : Select the **Availability Zone ending in "a"** from the dropdown menu.
- **Subnet ID** : Select **AppSubnet1** from the dropdown menu.
- **Security groups** : Select **EFSMountTargetSecurityGroup** from the dropdown menu.
- To remove the **default** Security group, choose the **X** .
- **Availability zone** : Select **Availability Zone ending in "b"** from the dropdown menu.
- **Subnet ID** : Select **AppSubnet2** from the dropdown menu.
- **Security groups** : Select **EFSMountTargetSecurityGroup** from the dropdown menu.
- To remove the **default** Security group, choose the **X** .

61. Choose **Next** .

The **File system policy –** *optional* page is displayed.

Note: Configuring this page is not necessary in this lab.

62. Choose **Next** .

The **Review and create** page is displayed.

63. Scroll to the bottom of the page, and choose **Create** .

A **Success! File system (fs-xxxxxxx) is available.** message is displayed on top of the screen.

The file system state displays as *Available* after several minutes.

TASK 3.3: COPYING AMAZON EFS METADATA

64. In the left navigation pane, select **File systems** .

65. Copy the **File system ID** generated for *myWPEFS* to a text editor. It has a format like *fs-a1234567*.

Congratulations, you have created the Amazon EFS.

Task 4 instructions: Creating an Application Load Balancer

TASK 4.1: NAVIGATING TO THE AMAZON EC2 CONSOLE

66. At the top of the AWS Management Console, in the search box, search for and choose `EC2` .

TASK 4.2: CREATING A TARGET GROUP

67. In the left navigation pane, under the **Load Balancing** section, choose **Target Groups** .

68. Choose **Create target group** .

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

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

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

70. In the **Health checks** section, configure the following:

- For **Health check path** : Enter `/wp-login.php` .
- Expand the **Advanced health check settings** section and configure the following:
  - **Healthy threshold** : Enter `2` .
  - **Unhealthy threshold** : Enter `10` .
  - **Timeout** : Enter `50` .
  - **Interval** : Enter `60` .

Leave the remaining settings on the page at their default values.

71. Choose **Next** .

The **Register targets** page is displayed. There are no targets to register currently.

72. Scroll to the bottom of the page and choose **Create target group** .

A **Successfully created target group: myWPTargetGroup** message is displayed on top of the screen.

TASK 4.3: CREATING AN APPLICATION LOAD BALANCER

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

74. Choose **Create load balancer** .

The **Select load balancer type** page is displayed.

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

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

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

- For **Load balancer name** : Enter `myWPAppALB` .

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

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

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

- From the **Security groups** dropdown menu, select **AppInstanceSecurityGroup** .
- To remove the **default** security group, choose the **X** .

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

- For **Listener HTTP:80** : from the **Default action** dropdown menu, select **myWPTargetGroup** .

80. Scroll to the bottom of the page and choose **Create load balancer** .

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

81. Choose **View load balancer** .

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

82. Copy the **myWPAppALB** load balancer **DNS name** to a text editor.

Congratulations, you have created the target group and an Application Load Balancer.

Task 5 instructions: Creating a launch template using CloudFormation

TASK 5.1: NAVIGATING TO THE CLOUDFORMATION CONSOLE

83. At the top of the AWS Management Console, in the search box, search for and choose `CloudFormation`.

TASK 5.2: OBTAINING AND REVIEWING THE CLOUDFORMATION TEMPLATE

84. Open the context (right-click) menu on this Task5.yaml link, and choose the option to save the CloudFormation template to your computer.

85. Open the downloaded file in a text editor (not a word processor).

86. Review the CloudFormation template.

87. Predict what resources are created by this template.

TASK 5.3: CREATING THE CLOUDFORMATION STACK

88. Choose **Create stack**.

Note: If the console starts you on the Stacks page instead of the Amazon CloudFormation landing page, then you can get to the Create stack page in two steps.

- Choose the Create stack dropdown menu.
- Choose **With new resources (standard)**.

The **Create stack** page is displayed.

89. Configure the following:

- Select Template is ready.
- Select Amazon S3 URL.
- Copy the **Task5TemplateUrl** value from the left side of these lab instructions and paste it in the **Amazon S3 URL** text box.
- Choose **Next**.

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

90. Set the **Stack name** as `WPLaunchConfigStack`.

91. Configure the following **parameters**:

- **DB name**: Paste the **initial database name** you copied in Task 2.
  Note: Make sure that you paste the *initial database name*, not the cluster name.
- **Database endpoint**: Paste the **writer endpoint** you copied in Task 2.
- **Database User Name**: Paste the **Master username** you copied in Task 2.
- **Database Password**: Paste the **Master password** you copied in Task 2.
- **WordPress admin username**: Defaults to `wpadmin`.
- **WordPress admin password**: Paste the **LabPassword** value from the left side of these lab instructions.
- **WordPress admin email address**: Input a valid email address.
- **Instance Type**: Leave the default value of **t3.medium**.
- **ALBDnsName**: Paste the **DNS name** value you copied in Task 4.
- **LatestAL2AmiId**: Leave the default value.
- **WPElasticFileSystemID**: Paste the **File system ID** value you copied in Task 3.

92. Choose **Next**.

The **Configure stack options** page is displayed. You can use this page to specify additional parameters. You can browse the page, but leave settings at their default values.

93. Choose **Next**.

The **Review WPLaunchConfigStack** page is displayed. This page is a summary of all settings.

94. Scroll to the bottom of the page and choose **Submit**.

The **stack details** page is displayed.

The stack enters the CREATE_IN_PROGRESS status.

95. Choose the **Stack info** tab.

96. Occasionally choose the **Overview** refresh.

97. Wait for the stack status to change to CREATE_COMPLETE.

Note: This stack can take up to 5 minutes to deploy the resources.

TASK 5.4: VIEWING CREATED RESOURCES FROM THE CONSOLE

98. Choose the **Resources** tab.

The list shows the resources that are created.

Congratulations, you have created the stack using the provided CloudFormation template.

Task 6 instructions: Creating the application servers by configuring an Auto Scaling group and a scaling policy

TASK 6.1: CREATING AN AUTO SCALING GROUP

99. At the top of the AWS Management Console, in the search box, search for and choose `EC2`.

100. In the left navigation pane, under the **Auto Scaling** section, choose **Auto Scaling Groups**.

101. Choose **Create Auto Scaling group**.

The **Choose launch template or configuration** page is displayed.

102. Configure the following:

- **Auto Scaling group name**: Enter `WP-ASG`.
- **Launch template**: Select the launch template that you created in Task 5.

103. Choose **Next** .

The **Choose instance launch options** page is displayed.

104. In the **Network** section, configure the following:

- **VPC** : Select **LabVPC** from the dropdown menu.
- **Availability Zones and subnets** : Select **AppSubnet1** and **AppSubnet2** from the dropdown menu.

105. Choose **Next** .

The **Configure advanced options -** *optional* page is displayed.

106. On the **Configure advanced options -** *optional* page, configure the following:

- Select **Attach to an existing load balancer** .
- Select **Choose from your load balancer target groups** .
- From the **Existing load balancer target groups** dropdown menu, select **myWPTargetGroup | HTTP** .
- For **Additional health check types -** *optional* : Select **Turn on Elastic Load Balancing health checks** .
- **Health check grace period** : Leave at the default value of 300 or more.
- **Monitoring** : Select **Enable group metrics collection within CloudWatch** .

107. Choose **Next** .

The **Configure group size and scaling policies -** *optional* page is displayed.

108. On the **Configure group size and scaling policies -** *optional* page, configure the following:

- In the **Group size** section:
  - **Desired capacity:** Enter `2` .
  - **Minimum capacity:** Enter `2` .
  - **Maximum capacity:** Enter `4` .
- In the **Scaling policies -** *optional* section:
  - Select **Target tracking scaling policy** .

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

109. Choose **Next** .

The **Add notifications -** *optional* page is displayed.

110. Choose **Next** .

The **Add tags -** *optional* page is displayed.

111. Choose **Add tag** and then configure the following:

- **Key** : Enter `Name` .
- **Value - optional** : Enter `WP-App` .

112. Choose **Next** .

The **Review** page is displayed.

113. Review the Auto Scaling group configuration for accuracy, and then at the bottom of the page, choose **Create Auto Scaling group** .

A WP-ASG, 1 Scaling policy created successfully. Group metrics collection is enabled. message is displayed on top of the screen.

The **Auto Scaling groups** page is displayed.

Now that you have created your Auto Scaling group, you can verify that the group has launched your EC2 instances.

114. Choose the Auto Scaling group **WP-ASG** link.

115. To review information about the Auto Scaling group, examine the **Group details** section.

116. Choose the **Activity** tab.

The **Activity history** section maintains a record of events that have occurred in your Auto Scaling group. The Status column contains the current status of your instances. When your instances are launching, the status column shows *PreInService*. The status changes to *Successful* after an instance is launched.

117. Choose the **Instance management** tab.

Your Auto Scaling group has launched two Amazon EC2 instances and they are in the *InService* lifecycle state. The Health Status column shows the result of the Amazon EC2 instance health check on your instances.

If your instances have not reached the *InService* state yet, you need to wait a few minutes. You can choose the refresh button to retrieve the current lifecycle state of your instances.

118. Choose the **Monitoring** tab. Here, you can review monitoring-related information for your Auto Scaling group.

This page provides information about activity in your Auto Scaling group, as well as the usage and health status of your instances. The **Auto Scaling** tab displays Amazon CloudWatch metrics about your Auto Scaling group, while the **EC2** tab displays metrics for the Amazon EC2 instances managed by the Auto Scaling group.

TASK 6.2: VERIFYING THE TARGET GROUPS ARE HEALTHY

119. Expand the navigation menu by choosing the menu icon ☰ in the upper-left corner.

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

121. Choose the **myWPTargetGroup** link.

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

Note: It can take up to 5 minutes for the health checks to show as healthy. Wait for the **Health status** to display *healthy* before continuing.

TASK 6.3: LOGGING IN TO WORDPRESS WEB APPLICATION

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

124. Copy the **myWPAppALB** load balancer **DNS name** to a text editor and append the value `/wp-login.php` to the end of the DNS name to complete your WordPress application URL.

Expected output: Example of a completed WordPress application URL:

myWPAppELB-4e009e86b4f704cc.elb.us-west-2.amazonaws.com/wp-login.php

125. Paste the WordPress application URL value into a new browser tab.

The **WordPress login** page is displayed.

126. Enter the following:

- Username or Email Address: Enter `wpadmin` .
- Password: Paste the **LabPassword** value from the left side of these lab instructions

127. Choose the **Log in** button.

Congratulations, you have created the AWS Auto Scaling group and successfully launched the WordPress application.

Conclusion

Congratulations! You now have successfully:

- Deployed a virtual network spread across multiple Availability Zones in a Region using a CloudFormation template.
- Deployed a highly available and fully managed relational database across those Availability Zones using Amazon RDS.
- Used Amazon EFS to provision a shared storage layer across multiple Availability Zones for the application tier, powered by NFS.
- Created a group of web servers that automatically scales in response to load variations to complete your application tier.

End lab

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

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

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

130. 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* .