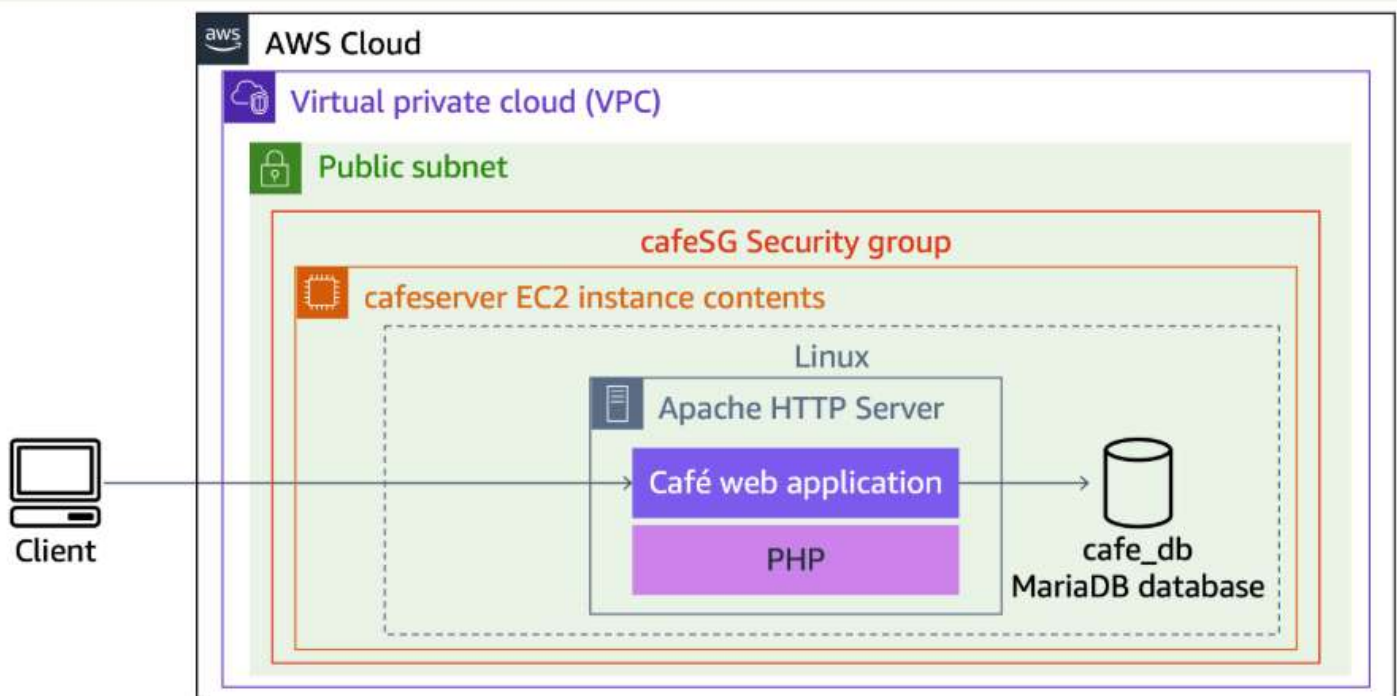# Troubleshooting the Creation of an EC2 Instance

Activity overview

In this activity, you use the AWS Command Line Interface (AWS CLI) to launch Amazon Elastic Compute Cloud (Amazon EC2) instances.

When you create the instance, you will reference a user data script to configure the instance to have an Apache web server, a MariaDB relational database (which is a fork of the MySQL relational database), and PHP running on the instance. Together, these software packages installed on a single machine are often referred to as a LAMP stack (Linux, Apache web server, MySQL, and PHP). Using a LAMP stack is a common way to create a website with a database backend on a single machine.

The same user data file will deploy website files and run database configuration scripts on the instance. The result will be an instance that hosts the Café Web Application.

The following diagram shows the architecture that you will create in this activity.

Objectives

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

- Launch an EC2 instance by using the AWS CLI.
- Troubleshoot AWS CLI commands and Amazon EC2 service settings by using basic troubleshooting tips and the open-source nmap utility.

Duration

This activity will require approximately **45 minutes** to complete.

Accessing the AWS Management Console

1. At the top of these instructions, choose | Start Lab | to launch your lab.

   A **Start Lab** panel opens displaying the lab status.

2. Wait until the message "Lab status: ready" appears, and then choose **X** to close the **Start Lab** panel.

3. Next to | Start Lab |, choose | AWS | to open the AWS Management Console on a new browser tab. The system automatically signs you in.

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

4. Arrange the AWS Management Console so that it appears alongside these instructions.

   Important: Do not change the lab Region unless specifically instructed to do so.

Task 1: Connecting to the CLI Host instance

In this task, you use EC2 Instance Connect to connect to the *CLI Host* EC2 instance that was created when the lab was provisioned. You will use this instance to run CLI commands.

5. On the **AWS Management Console**, in the **Search** bar, enter and choose `EC2` to open the **Amazon EC2 Management Console**.

6. In the navigation pane, choose **Instances**.

7. From the list of instances, select the ☐ **CLI Host** instance.

8. Above the list of instances, choose **Connect.**

9. On the **EC2 Instance Connect** tab, choose **Connect.**

   Note: If you prefer to use an SSH client to connect to the EC2 instance, see the guidance to Connect to Your Linux Instance.

Now that you are connected to the CLI Host instance, you can configure and use the AWS CLI to call AWS services.

Task 2: Configuring the AWS CLI

In this task, you configure the AWS CLI by providing the configuration parameters that were made available to you when the lab was provisioned. After configuration, you run CLI commands to interact with AWS services.

Note: Amazon Linux Amazon Machine Images (AMIs) already have the AWS CLI preinstalled.

10. To set up the AWS CLI profile with credentials, run the following command:

```
aws configure
```

   Tip: In EC2 Instance Connect, you might need to use the context (right-click) menu to paste values in the terminal.

11. When prompted, enter the following information:

   - **AWS Access Key ID:** At the top of these instructions, choose Details , and then choose Show . Copy the **AccessKey** value into the terminal window.
   - **AWS Secret Access Key:** Copy the **SecretKey** value from the lab instructions into the terminal window.
   - **Default region name:** Copy the **LabRegion** value from the lab instructions into the terminal window.
   - **Default output format:** Enter `json`

Task 3: Creating an EC2 instance by using the AWS CLI

In this task, you observe and run a *shell script*, which was provided to you, to create an EC2 LAMP instance by using AWS CLI commands. The script intentionally contains issues. Your challenge is to find the issues and resolve them. As you resolve each issue, you can run the script again to check that the issue was resolved.

Task 3.1: Observe the script details

First, create a backup of the script that you will edit in a later step.

12. To change to the directory where the script file exists and create a backup of it, run the following commands:

```
cd ~/sysops-activity-files/starters
cp create-lamp-instance-v2.sh create-lamp-instance.backup
```

Tip: It is a good practice to back up files before you modify them.

13. Open the *create-lamp-instance-v2.sh* script file in a command line text editor, such as VI.

Tip: To open the file in read-only mode in the VI editor, use the following command:

```
view create-lamp-instance-v2.sh
```

14. Analyze the contents of the script:

Tip: If you are using VI, you can display the line numbers by typing `:set number` and then pressing Enter.

- **Line 1:** This is a bash file, so the first line contains `#!/bin/bash`.
- **Lines 7–11:** The instance size is set to t3.small, which should be large enough to run the database and web server.
- **Lines 16–29:** The script invokes the AWS CLI describe-regions command to get a list of all AWS Regions. In each Region, the script queries for an existing VPC that is named *Cafe VPC*. After finding the VPC, the script captures the *VPC ID* and *Region*, and breaks out of the while loop. This is the VPC where the LAMP instance for the Café Web Application should be deployed.
- **Lines 31–55:**
    - The script invokes AWS CLI commands to look up the *subnet ID, key pair name,* and *AMI ID* values that are needed to create the EC2 instance.
    - Notice that line 32 ends with a backslash (\) character. This character can be used to wrap a single command on another line in the script file. This technique is used many times in the script to make it easier to read.
- **Lines 57–122:** The script cleans up the AWS account for situations where this script already ran in the account and is being run again. The script checks whether an instance that is named *cafeserver* already exists and whether a security group that includes *cafeSG* in its name already exists. If either resource is found, the script prompts you to delete them.
- **Lines 124–152:** The script creates a new security group with ports 22 and 80 open.

- **Lines 154–168:**
    - The script creates a new EC2 instance. Notice how this part of the script uses the values that were set in lines 8 and 10, and the values that were collected in lines 16–57.
    - Notice the reference to the user data file. You will review the details of this file in another step.
    - The entire call to create the instance is captured in a variable that is named *instanceDetails*. The contents of this variable are then echoed out to the terminal on line 177, and they are formatted for easier viewing by using a Python JSON tool.
- **Lines 179–188:** The *instanceId* value is parsed out of the *instanceDetails* variable. Then, a while loop checks every 10 seconds to see if a public IP address has been assigned to the instance. After the check succeeds, the public IP address is written to the terminal.

15. Exit the text editor.

Tip: If you are using VI, enter `:q!`

16. To display the contents of the user data script, run the following command:

```
cat create-lamp-instance-userdata-v2.txt
```

Notice how the user data script runs a series of commands on the instance after it is launched. These commands will install a web server, PHP, and a database server.

Task 3.2: Try to run the script

17. Now that you have an idea what the shell script is designed to do, try to run it:

```
./create-lamp-instance-v2.sh
```

The script fails and exits without successfully completing. *This behavior is expected.*

Task 3.3: Troubleshoot issues

Issue #1

The terminal output displays the following message: "An error occurred (InvalidAMIID.NotFound) when calling the RunInstances operation: The image id '[ami-xxxxxxxxxx]' does not exist".

Tips to help you resolve issue #1:

- Locate the line in the script that led to this error. Does anything look incorrect in the script?
- Was the correct Region value used for the run-instances command?
- After you identify the issue, update the script to fix the issue, and run the script again. If you are prompted to delete any existing security groups or instances that were created when the script ran previously, always reply with `Y`. Is the error resolved?

After you fix the issue, the run-instances command succeeds, and a public IPv4 address is assigned to the new instance.

Try to connect to the webpage

In a browser, navigate to the following address. Replace *<public-ip>* with the Public IPv4 address of the new instance that you created: `http://<public-ip>`

The attempt fails. You need to resolve issue #2.

Issue #2

The run-instances command succeeded, and a public IP address was assigned to the new instance. However, you cannot load the test webpage.

18. Connect to the new *LAMP instance* by using EC2 Instance Connect, which is the method that you used to connect to the CLI Host instance.

Tips to help you resolve issue #2:

- The web server runs on TCP port 80. Is it open?
- Can you verify that the web server service is running? The web server service name is *httpd*.

- Can you verify that the web server service is running? The web server service name is *httpd*.

19. In the terminal window for the *CLI Host* instance, run the following command to install nmap, which is a port scanning tool:

```
sudo yum install -y nmap
```

20. Next, run the following command. Replace *<public-ip>* with the actual public IPv4 address of your *LAMP instance*:

```
nmap -Pn <public-ip>
```

The output from this command shows which ports are accessible. Do the results match what you expected? Do you know what needs to be changed based on the output?

Test whether the user data script ran

21. After you identify and resolve the issue, in a browser, navigate to the following address. Replace *<public-ip>* with the Public IPv4 address of the new instance that you created: `http://<public-ip>`

If you resolved issue #2 successfully, you should see the following message: "Hello From Your Web Server!"

22. Check the log file that shows whether the user data script ran as expected.

In the terminal window for the *LAMP instance*, run the following command to see the log file entries as they are written:

```
sudo tail -f /var/log/cloud-init-output.log
```

On an Amazon Linux instance, the *cloud-init* service runs the commands in the user data file.

Observe the log file entries. Notice the message that is related to the installation of MariaDB and PHP. There should be no error messages.

You should also see messages related to files for the Café Web Application that were downloaded and extracted to this instance, such as "Create Database script completed".

When you are finished with your observations, to exit the tail utility, enter `Ctrl-C`

Tip: To view the entire log file, run `sudo cat /var/log/cloud-init-output.log`

Task 4: Verifying the functionality of the website

23. Verify that the website is deployed.

    In a browser, navigate to the following address. Replace *<public-ip>/cafe* with the Public IPv4 address of the new instance that you created: `http://<public-ip>/cafe`

    If you are successful, you should see the home page for the café website. Congratulations!

24. Test whether you can order items through the website.

- Choose the **Menu** link. A new page loads at http://<public-ip>/cafe/menu.php.
- Choose a few desserts to order, and then choose **Submit Order**. The Order Confirmation page displays with line-item details.
- Place another order for different items. Then, choose the **Order History** page. The details of both orders were captured.

    Note: The order details are being captured and stored in the database that is running on the LAMP instance that you launched.

Conclusion

Congratulations! You now have successfully done the following:

- Launched an EC2 instance by using the AWS CLI.
- Troubleshot AWS CLI commands and Amazon EC2 service settings by using basic troubleshooting tips and the open-source nmap utility.

Lab complete

Congratulations! You have completed the activity.

25. At the top of this page, choose | End Lab | and then choose Yes to confirm that you want to end the activity.

A panel appears indicating that "You may close this message box now. Lab resources are terminating."

26. To close the **End Lab** panel, choose the **X** in the upper-right corner.