

Activity - Troubleshoot Creating an EC2 Instance Using the AWS CLI

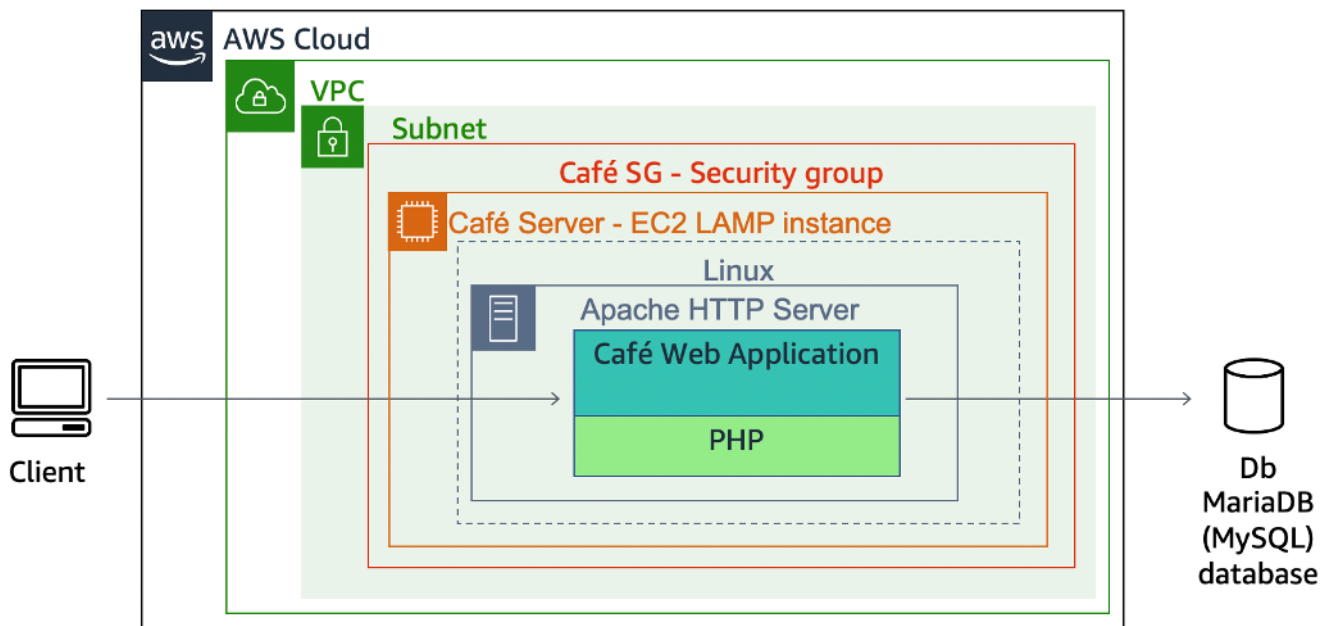
Activity overview

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

When you create the instance, the *user-data* file you reference will configure the instance to have an Apache web server, a MariaDB relational database (a fork of the MySQL relational database), and PHP running on it. Together, these software packages installed on a single machine are often referred to as a LAMP stack (**L**inux, **A**pache web server, **M**ySQL, and **P**HP). LAMP stacks are a common way to create a website, which also has a database backend deployment, on a single machine.

The same user-data file will deploy website files on the instance and run SQL scripts on the instance. The result will be an instance that hosts an updated version of the Café website. The new version of the website will support online customer orders.

This diagram summarizes the activities that you will complete in this activity.



Activity objectives

After completing this activity, you will be able to:

- **Launch** an Amazon EC2 instance using the AWS CLI
- **Troubleshoot** AWS CLI commands and Amazon EC2 service settings

Business case relevance

A new business requirement for Café—Online orders



Martha and Frank were really impressed by the work that Nikhil and Sofia did when the first version of the cafe's website went live. Customers also expressed their appreciation because they finally had an online resource that provided information about the café. However, many customers who come to the café have said that they would like to be able to place orders online.

It seems that the work of a cloud operations professional is never complete! In this activity, you will again take on the roles of Nikhil and Sofia, and work on upgrading the website so that it takes online orders.

Activity steps

Duration: This activity requires approximately **45 minutes** to complete.

Accessing the AWS Management Console

1. At the top of these instructions, click to launch your lab.

A Start Lab panel opens displaying the lab status.

2. Wait until you see the message "**Lab status: ready**", then click the **X** to close the Start Lab panel.

3. At the top of these instructions, click

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

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

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

Leave this browser tab open. You will return to it later in this activity.

Task 1: Connect to an Amazon Linux EC2 instance by using SSH


In this task, you will connect to an existing Amazon Linux EC2 instance that already has the AWS CLI installed on it.

Windows users should follow Task 1.1. Both macOS and Linux users should follow Task 1.2.

[macOS/Linux users—click here for login instructions](#)

Task 1.1: Windows SSH

Windows Users: Using SSH to Connect

 These instructions are for Windows users only.

If you are using macOS or Linux, [skip to the next section](#).

5. Read through the three bullet points in this step before you start to complete the actions, because you will not be able to see these instructions when the Details panel is open.

- Click on the drop down menu above these instructions you are currently reading, and then click . A Credentials window will open.
- Click on the **Download PPK** button and save the **labsuser.ppk** file. Typically your browser will save it to the Downloads directory.
- Then exit the Details panel by clicking on the **X**.

6. Download needed software.

- You will use **PuTTY** to SSH to Amazon EC2 instances. If you do not have PuTTY installed on your computer, [download it here](#).

7. Open **putty.exe**

8. Configure PuTTY to not timeout:

- Click **Connection**
- Set **Seconds between keepalives** to

This allows you to keep the PuTTY session open for a longer period of time.

9. Configure your PuTTY session:

- Click **Session**
- Host Name (or IP address):** Copy and paste the **IPv4 Public IP address** for the instance. To find it, return to the EC2 Console and click on **Instances**. Check the box next to the instance you want to connect to and in the *Description* tab copy the **IPv4 Public IP** value.
- Back in PuTTY, in the **Connection** list, expand **SSH**
- Click **Auth** (don't expand it)
- Click **Browse**
- Browse to and select the lab#.ppk file that you downloaded
- Click **Open** to select it
- Click **Open**

10. Click **Yes**, to trust the host and connect to it.

11. When prompted **login as**, enter:

This will connect you to the EC2 instance.

12. [Windows Users: Click here to skip ahead to the next task.](#)

Task 1.2: macOS/Linux SSH

These instructions are for Mac/Linux users only. If you are a Windows user, [skip ahead to the next task](#).

13. Read through the three bullet points in this step before you start to complete the actions, because you will not be able see these instructions when the Details panel is open.

- Click on the `Details` drop down menu above these instructions you are currently reading, and then click `Show`. A Credentials window will open.
- Click on the **Download PEM** button and save the **labsuser.pem** file.
- Then exit the Details panel by clicking on the **X**.

14. Open a terminal window, and change directory `cd` to the directory where the labsuser.pem file was downloaded.

For example, run this command, if it was saved to your Downloads directory:

```
cd ~/Downloads
```

15. Change the permissions on the key to be read only, by running this command:

```
chmod 400 labsuser.pem
```

16. Return to the AWS Management Console, and in the EC2 service, click on **Instances**. Check the box next to the instance you want to connect to.

17. In the *Description* tab, copy the **IPv4 Public IP** value.

18. Return to the terminal window and run this command (replace **<public-ip>** with the actual public IP address you copied):

```
ssh -i labsuser.pem ec2-user@<public-ip>
```

19. Type `yes` when prompted to allow a first connection to this remote SSH server.

Because you are using a key pair for authentication, you will not be prompted for a password.

Task 2: Configure the AWS CLI

NOTE: Unlike some other Linux distributions that are available on AWS, Amazon Linux instances already have the AWS CLI pre-installed.

20. Update the AWS CLI software with the credentials.

```
aws configure
```

21. At the prompts, enter the following information:

- **AWS Access Key ID:** Click on the `Details` drop down menu above these instructions, and then click `Show`. Copy the **AccessKey** value and paste it into the terminal window.
- **AWS Secret Access Key:** Copy and paste the **SecretKey** value from the same Credentials screen.
- **Default region name:** Click on the `Details` drop down menu above these instructions, and then click `Show`. Copy the **LabRegion** value and paste it into the terminal window.
- **Default output format:** `json`

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

Task 3.1: Observe the script details

22. Change to the directory where the script file you will be editing exists, and create a backup of it as well, by running these commands:

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

Tip: It is always a good practice to backup files before you start making modifications to them.

23. Open the create-lamp-instance-v2.sh script file in your favorite command-line text editor (such as VI).

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

24. 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 file 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, it queries for an existing VPC that has the name `Cafe VPC`. When it finds it, it captures the **vpc** ID and **region** where the Café LAMP instance should be deployed and breaks out of the while loop.

Lines 31–57:

- The script invokes AWS CLI commands to look up the **Subnet ID**, **Keypair name**, and **AMI ID** values that will be needed to create an EC2 instance.
- Notice on line 32 that the line ends with a backslash (`\`) character. This character can be used to wrap a single command on to another line in the script file. You will find that this technique is used many times in this script to make it easier to read.

Lines 59–124:

- This part of the script cleans up the AWS account for situations where this script already ran on your AWS account, and it is now being run again. The script checks if an instance named *cafeserver* already exists, and if a security group that includes *cafeSG* in its name already exists. If either resource is found, the script prompts you to delete them.

Lines 126–154:

- The script creates a new security group with ports 22 and 80 open.

Lines 156–168:

- The script creates a new EC2 instance.
- Notice how the values set in lines 8 and 10 and the values collected in lines 16–57 are used in this AWS CLI call.
- Also, notice the reference to a user-data script.

NOTE: You will be prompted to look at the details of the user-data file in a moment.

25. While you are still in the bash script, look again at **lines 157–168**:

- 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 JavaScript Object Notation (JSON) tool.

Lines 179–188:

- The **instanceId** is parsed out of the `instanceDetails`, and then a while loop checks every 10 seconds to see if a public IP address has been assigned to the instance. When the check succeeds, the public IP address is written to the terminal.

26. Exit the text editor without making any changes (if you are using VI, type the `:q!` command).

27. Display the contents of the user-data script by running this 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 running the script

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

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

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

Issues were intentionally left in this script. *Your challenge is to find the issues and resolve them.* You can always run the script again after you make a change to see if it resolves the issue.

Task 3.3 Troubleshoot issues

Issue #1

The terminal output states that the call to `run-instances` fails with the message: *"An error occurred (InvalidAMIID.NotFound) when calling the RunInstances operation: The image id '[ami-xxxxxxxxxx]' does not exist"*

Tips to help you resolve the issue #1:

- Locate the line in the bash script that led to this error. Does anything look incorrect in the script?
- Is the region value used for the `run-instances` command correct?
- After you think you have found the issue and have adjusted the script accordingly, run it again (always reply with `Y` when you are prompted to delete any existing security groups or instances that were created the last time the script ran). Did the error get resolved?

Issue #2

The call to `run-instances` succeeds, and a public IP address is assigned to the new instance. However, when you try to use SSH to connect to the instance with the key pair file, you cannot connect (*Permission denied* error).

Tips to help you resolve the issue #2:

- It is an Amazon Linux 2 instance. The default user name is **ec2-user**.
- Are the permissions on the key pair file set correctly (e.g., `chmod 400`)? What key pair file are you using? Is that the key pair file that the instance expects you to connect with?
- Remember to use the [AWS CLI Command Reference](#) as an important source of information.
- After you think you have found the issue and have adjusted the script accordingly, run it again.
 - Are you able to connect to the instance by using SSH? Make the SSH attempt from your laptop, where you have the key pair already downloaded. Remember that it takes up to 5 minutes for the instance to fully boot.
 - You should be able to connect by using SSH. Leave this SSH connection open.

Try connecting to the web page

In a browser, load `http://<public-ip>`, where `<public-ip>` is the actual IPv4 Public IP of the instance.

The attempt fails. You will need to resolve issue #3.

Issue #3

The instance is created and you can establish an SSH connection to it successfully, but you cannot load the test webpage.

Tips to help you resolve the issue #3:

- The web server runs on TCP port 80.
- If you use SSH to connect to the instance, can you verify that the web server service is running? The web server service name is **httpd**.

29. While you are connected to the instance via SSH, run this command to install nmap, which is a port scanning tool:

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

30. Next, run this command (where `<public-ip>` is the actual public IP address of your LAMP instance):

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

- The results that are returned by this command show which ports are accessible. Do the results match what you expected?

Test the webpage again and verify that the user-data script ran

31. After you think you have resolved the third issue, in a browser, load `http://<public-ip>` where `<public-ip>` is the actual IPv4 Public IP address of the instance.

If you resolved Issue #3, this time you should see the following message: *Hello From Your Web Server!*

32. Check the log file that shows if the user-data script command ran as expected.

In the terminal window where you have an active SSH connection to the LAMP instance, run the following command in order to see the log files entries as they are written:

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

On an Amazon Linux instance, the user-data file commands are run by the **cloud-init** service.

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 cafe files that were downloaded and extracted onto this instance, including the message: *Create Database script completed.*

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

Tip: if you want to view the entire log file, use `sudo cat /var/log/cloud-init-output.log`

Task 4: Verify the functionality of the new website

33. Verify that the new website is now deployed.

Load `http://<public-ip>/cafe/` in the web browser (where `<public-ip>` is the actual IPv4 Public IP address).

If you are successful, you should see an improved version of the Café website. Congratulations!

34. Test placing orders.

- Click the **Menu** link. A new page should load at `http://<public-ip>/cafe/menu.php`
- Pick your favorite desserts by setting a quantity on a few of them, then scroll to the bottom and click **Submit Order**.
- The **Order Confirmation** page should display, with line-item details.
- Place another order for different items, then click the **Order History** page. You should see that the details of both orders were captured.

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

Update from Café



Nikhil and Sofia are really becoming heroes around the Café! Customers are noticing that they can now place orders online. Nikhil and Sofia also really appreciated the assistance from Mateo when it came to troubleshooting their deployment script.

Meanwhile, Martha and Frank have noticed another benefit of the new website: every evening, they can check the orders that are placed online for pickup the next day. Because Martha and Frank can see these orders in advance, they have a better idea of which desserts they must bake more of the next morning so they do not run out of them. Martha and Frank are really starting to see the benefits of creating an online component to their business!

Lab Complete

Congratulations! You have completed the activity.

35. Click at the top of this page and then click **Yes** to confirm that you want to end the activity.

A panel will appear, indicating that "DELETE has been initiated... You may close this message box now."

36. Click the **X** in the top right corner to close the panel.