Challenge Lab: Creating a Dynamic Website for the Café

Scenario

After the café launched the first version of its website, customers told the café staff how nice the website looks. However, in addition to the providing praise, customers often asked whether they could place online orders.

Sofía, Nikhil, Frank, and Martha discussed the situation. They agreed that their business strategy and decisions should focus on delighting their customers and providing them with the best possible café experience.

Lab overview and objectives

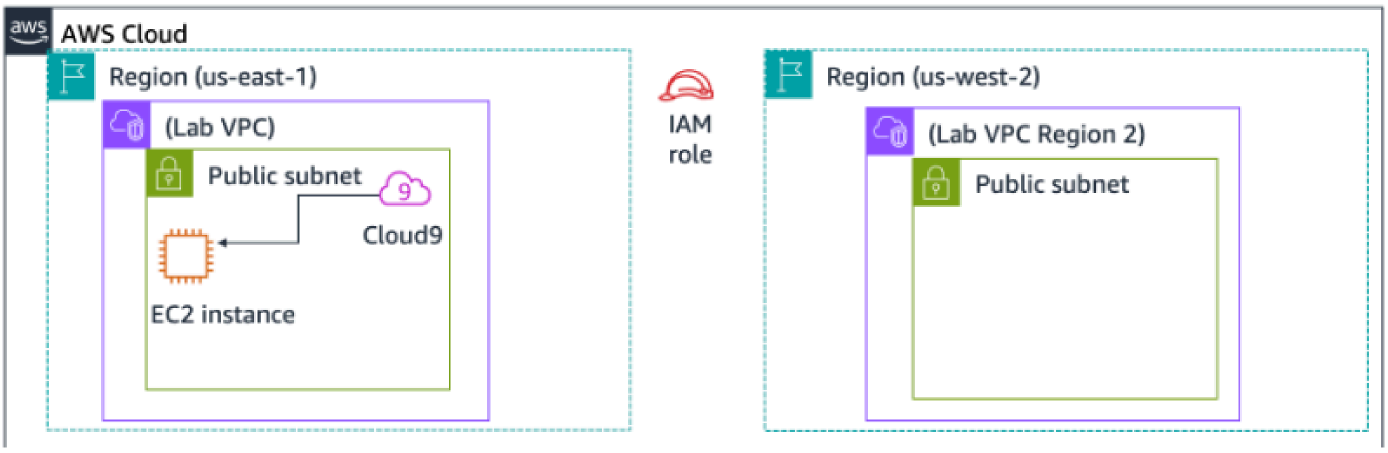
In this lab, you deploy an application on an Amazon Elastic Compute Cloud (Amazon EC2) instance. The application gives the café the ability to accept online orders. After testing that the application works as intended in the first AWS Region (the development environment), you then create an Amazon Machine Image (AMI) from the EC2 instance. You also deploy a second instance of the same application as the production environment in another AWS Region.

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

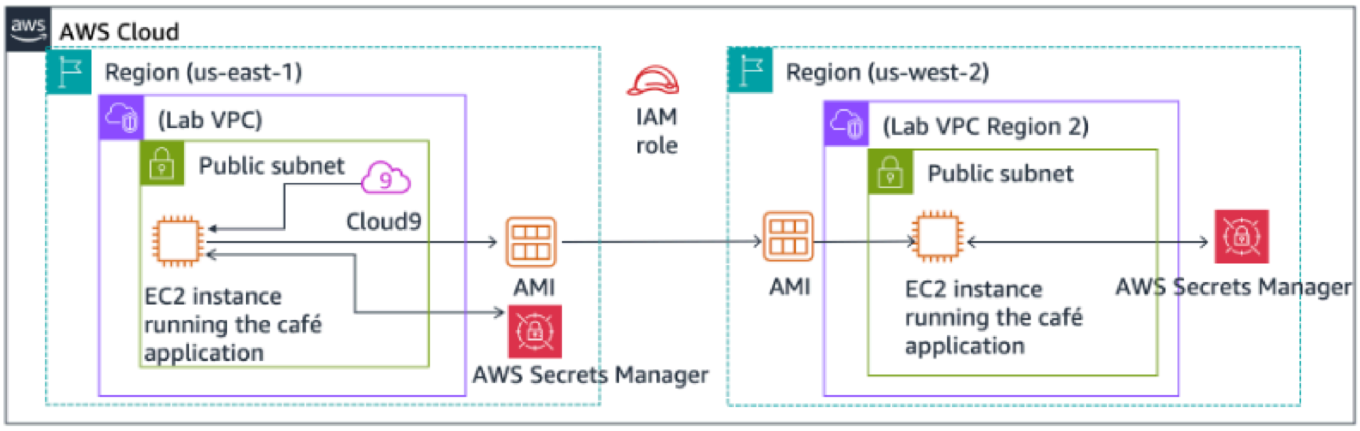
- Connect to the AWS Cloud9 integrated development environment (IDE) on an existing EC2 instance.
- Analyze the EC2 instance environment and confirm web server accessibility.
- Install a web application on an EC2 instance that also uses AWS Secrets Manager.
- Test the web application.
- Create an AMI.
- Deploy a second copy of the web application to another AWS Region.

When you start the lab, some resources are already created for you in the AWS account:

At the end of this lab, your architecture should look like the following example:



At the end of this lab, your architecture should look like the following example:



Duration

This lab requires approximately **60 minutes** to complete.

A business request for the café: Preparing an EC2 instance to host a website (challenge #1)

The café wants to introduce online ordering for customers and give café staff the ability to view submitted orders. Their current website architecture, where the website is hosted on Amazon S3, does not support the new business requirements.

In the first part of this lab, you take on the role of Sofía. You configure an EC2 instance so that it is ready to host a website for the café.

Task 1: Analyzing the existing EC2 instance

In this task, you note details about an existing EC2 instance that has been created for you in the AWS account.

1. On the AWS Management Console, in the search box, enter and choose EC2 to open the Amazon EC2 console.
2. Choose **Instances**.
3. Notice the running instance named **aws-cloud9-CafeWebServer-...**. This EC2 instance was created when you started the lab.

Task 1.1: Answering questions about the instance

Your answers are evaluated when you choose the blue **Submit** button at the end of the lab.

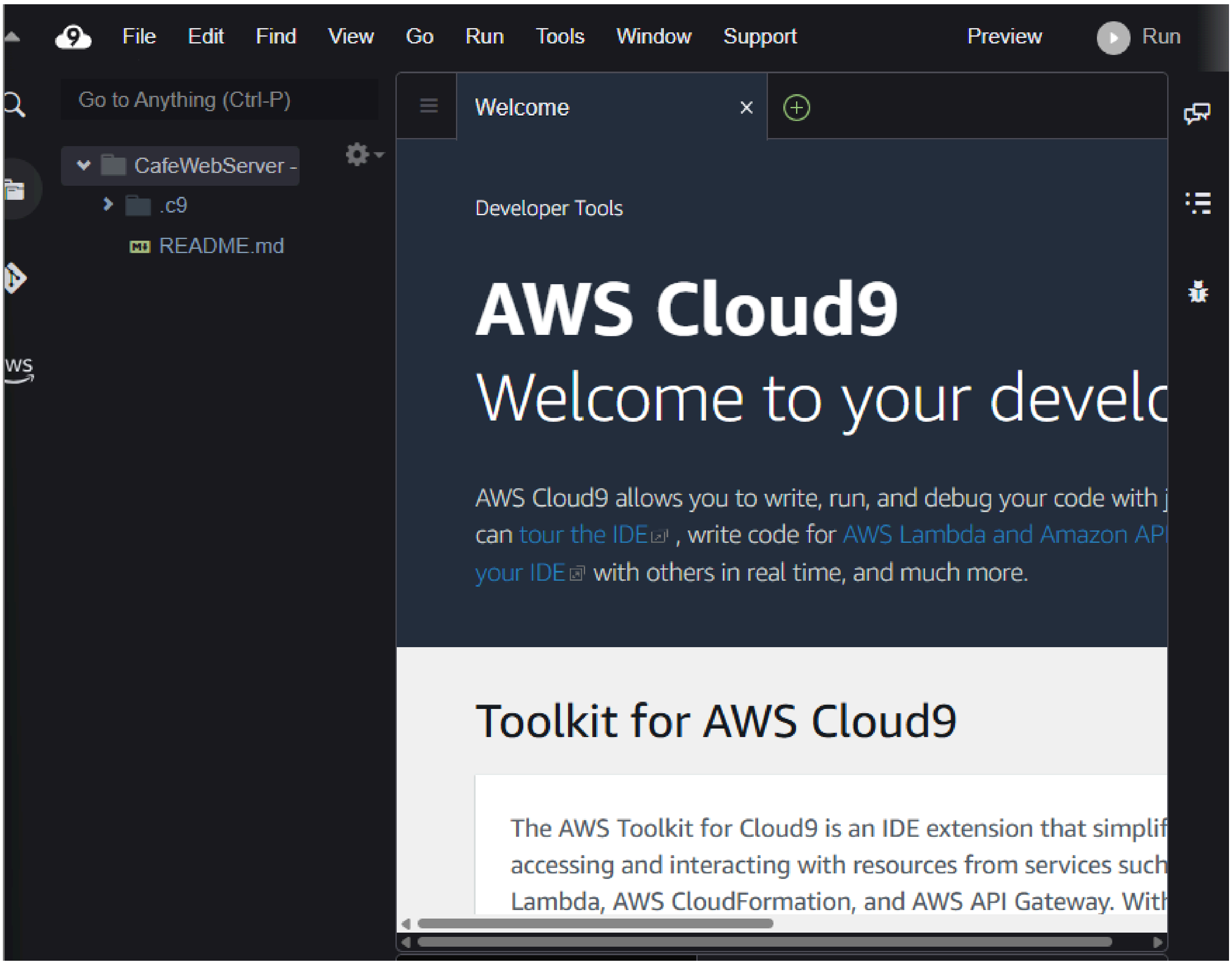
1. To access the questions in this lab, at the top of these instructions, choose **AWS Details**.
2. Choose the **Access the multiple choice questions** link.
3. On the page that you loaded, answer the first four questions about the **aws-cloud9-CafeWebServer-...** EC2 instance:
   - **Question 1**: Is the instance in a public subnet?
   - **Question 2**: Does the EC2 instance have an IPv4 Public IP address assigned to it?
   - **Question 3**: What inbound TCP port numbers are open for this instance?
   - **Question 4**: Does the EC2 instance have an AWS Identity and Access Management (IAM) role associated with it?
4. **Note**: Leave the questions webpage open in your browser tab. You return to it later in this lab.

Task 2: Connecting to the IDE on the EC2 instance

AWS Cloud9 is service that can run on an EC2 instance. It provides an IDE that includes features such as a code editor, debugger, and terminal.

By using the AWS Cloud9 IDE, you don't need to download a key pair and connect to the EC2 instance by using PuTTY or similar Secure Shell (SSH) software. By using AWS Cloud9, you also don't need to use command line text-editing tools (such as vi or nano) to edit files on the Linux instance.

1. On the browser tab with these instructions, from the **AWS Details** window, copy the value for **Cloud9url**, and paste it into a new browser tab to open the AWS Cloud9 console.
2. **Note:** If you closed the **AWS Details** window, choose **AWS Details** to open it again.
3. You are now connected to the AWS Cloud9 IDE that is running on the EC2 instance that you observed earlier.
4. The IDE includes the following:
   - A bash terminal in the bottom-right panel.
   - A file browser in the left panel that shows files in the **/home/ec2-user/environment** directory on the instance.
   - A file editor in the upper-right panel. If you select a file in the file browser, such as the README.md file, it displays in the editor.



Developer Tools

AWS Cloud9

Welcome to your develc

AWS Cloud9 allows you to write, run, and debug your code with j
can tour the IDE , write code for AWS Lambda and Amazon API
your IDE with others in real time, and much more.

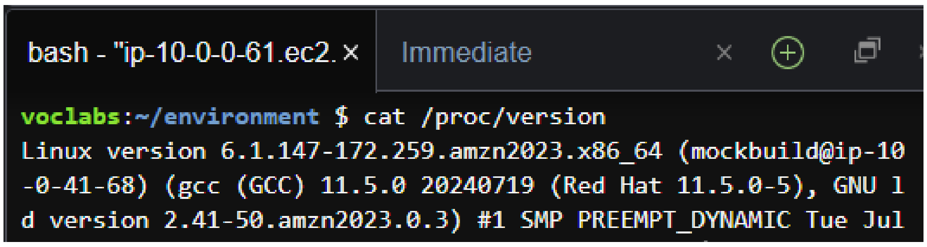
Toolkit for AWS Cloud9

The AWS Toolkit for Cloud9 is an IDE extension that simplif
accessing and interacting with resources from services such
Lambda, AWS CloudFormation, and AWS API Gateway. With

Task 3: Analyzing the LAMP stack environment and confirming that the web server is accessible

Recall that the objective of this challenge lab is to configure an EC2 instance to host the new dynamic website for the café. In this task, you analyze what is already installed.

1. To observe the operating system version, in the AWS Cloud9 bash terminal, run the following command:
2. cat /proc/version
3. Notice how the output indicates that it is an Amazon Linux instance roughly analogous to Red Hat 7.



```
voclabs:~/environment $ cat /proc/version
Linux version 6.1.147-172.259.amzn2023.x86_64 (mockbuild@ip-10
-0-41-68) (gcc (GCC) 11.5.0 20240719 (Red Hat 11.5.0-5), GNU l
d version 2.41-50.amzn2023.0.3) #1 SMP PREEMPT_DYNAMIC Tue Jul
```

1. To observe the web server, PHP details, and server state, run the following commands:

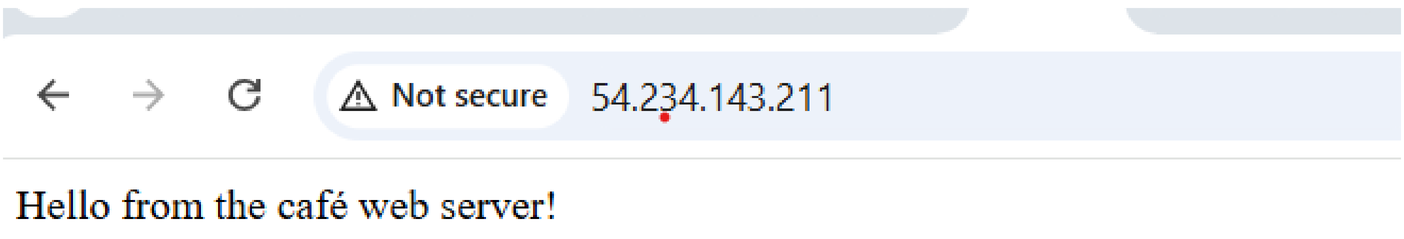
2. sudo httpd -v
3. service httpd status
4. php --version
5. The output should show the versions of the web server and that they are not currently running.
6. To start the web server, install the database, and set them to start automatically after any future EC2 instance restarts, run the following commands:
7. sudo chkconfig httpd on
8. sudo service httpd start
9. sudo service httpd status
10. #verify database
11. sudo mariadb --version
12. sudo systemctl enable mariadb
13. #
14. sudo chkconfig mariadb on
15. sudo service mariadb start
16. sudo service mariadb status

Note: After running these commands, if the terminal doesn't show a prompt, press Q.

1. To configure the EC2 instance so that you can use the AWS Cloud9 editor to edit web server files, run the following commands:
2. ln -s /var/www/ /home/ec2-user/environment
3. sudo chown ec2-user:ec2-user /var/www/html
4. Notice that the AWS Cloud9 file browser currently does not display the Apache web server default web directory.
5. The first command creates a symlink from the default AWS Cloud9 editor workspace to the **/var/www** directory that contains your web server files.
6. The second command changes ownership of the **html** subdirectory so that the **ec2-user** (which you are logged in as) can edit and create new files in it.
7. Next, you create a test webpage.
8. In the file browser, expand the **CafeWebServer > www** directory, and choose the **html** directory.
9. Choose **File** > **New File**.
10. In the text editor tab, paste the following line:
11. <html>Hello from the café web server!</html>
12. Choose **File** > **Save**, and save the file in the **html** directory as index.html.
13. Make the website accessible from the internet.
14. In this step, you need to verify and update the configurations that make the webpages (which are hosted on the web server) accessible from the internet.
15. Expand each tip for help solving the challenge.

Tip #1 In the Amazon EC2 console, locate the public IPv4 address of the EC2 instance. In a new browser tab, enter `http://<public-ip>` and replace ** with the public IPv4 address, and open the page. Does the message that you entered into index.html file load in the browser?

Tip #2 To allow inbound HTTP traffic on TCP port 80 from anywhere, update the **Source** in the security group of the EC2 instance as needed.



Hello from the café web server!

New business requirement: Installing a dynamic website application on the EC2 instance (challenge #2)

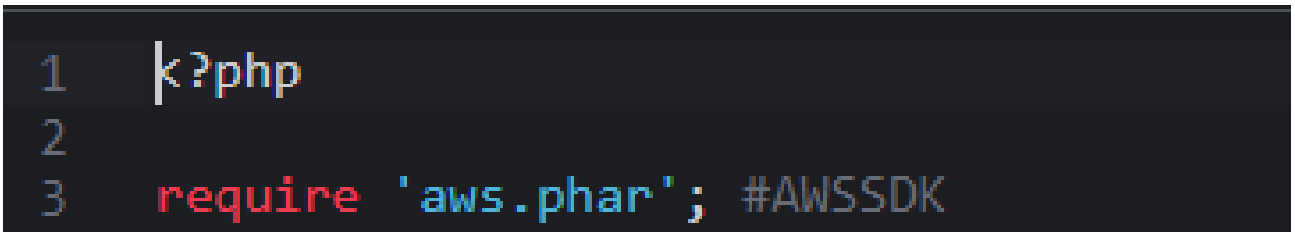
In the previous challenge, you configured the EC2 instance. You now know that PHP is installed and that the application environment has a running relational database. Also, the environment has a running web server that can be accessed from the internet. You now have the basic setup for hosting a dynamic website for the café.

In the second part of this lab, you take on the role of Sofía and install the café application on the EC2 instance.

Task 4: Installing the café application

1. To download and extract the web server application files, run the following commands:
2. cd ~/environment
3. wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACACAD-3-113230/03-lab-mod5-challenge-EC2/s3/setup.zip
4. unzip setup.zip
5. wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACACAD-3-113230/03-lab-mod5-challenge-EC2/s3/db.zip
6. unzip db.zip
7. wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACACAD-3-113230/03-lab-mod5-challenge-EC2/s3/cafe.zip
8. unzip cafe.zip -d /var/www/html/
9. cd /var/www/html/cafe/
10. wget https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.zip
11. wget https://docs.aws.amazon.com/aws-sdk-php/v3/download/aws.phar
12. unzip aws -d /var/www/html/cafe/
13. chmod -R +r /var/www/html/cafe/
14. Notice how the file browser now shows the .zip files that you downloaded.
15. You also extracted these archive files, which created the **cafe**, **db**, and **setup** directories in your work environment.

16. Next, you observe how the application is designed to work.
17. To open the **www/html/cafe/Psr/index.php** source code file in the AWS Cloud9 editor, select (double-click) the file.
18. Notice that this file has HTML code in it, but it also contains sections that are enclosed in elements. These elements make calls to other systems and resources. For example, on line 18, you see that the PHP code references a file named **getAppParameters.php**.
19. Open the **getAppParameters.php** file in the code editor.
    - Notice on line 3 of this file that the **AWSSDK** is invoked.



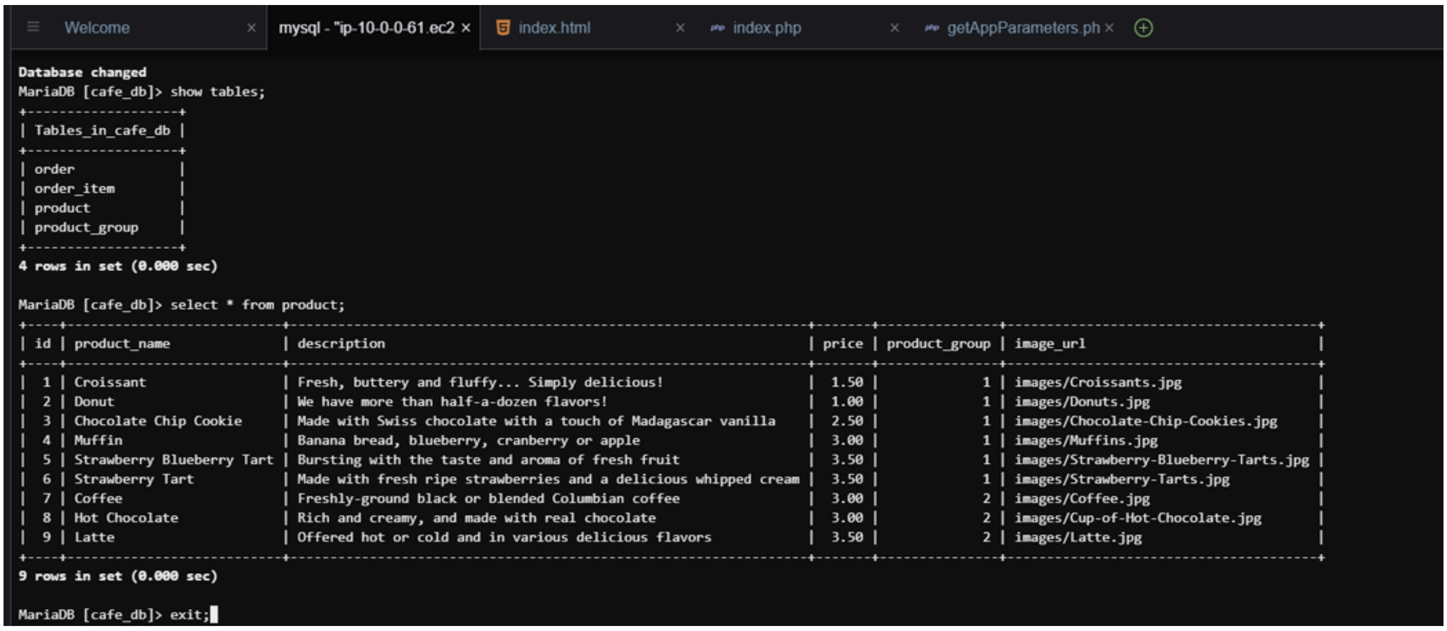
```
1   <?php
2
3   require 'aws.phar'; #AWSSDK
```

    - In the subsequent sections, the web application creates a client that connects to Secrets Manager. The application then retrieves seven parameters from Secrets Manager. Those parameters have not been created in Secrets Manager yet, but you do that next.

1. To configure the application parameters, in the bash terminal, run the following commands:
2. cd
3. cd environment/setup/
4. ./set-app-parameters.sh
5. The shell script that you just ran issues AWS Command Line Interface (AWS CLI) commands. These commands add the secrets that the application will use from Secrets Manager.
6. On the AWS Management Console, in the search box, enter and choose Secrets Manager to open the Secrets Manager console.
7. From the panel on the left, choose **Secrets**.
8. There are now seven parameters stored as secrets.
9. **Note:** It might take a few minutes for these parameters to be stored as secrets. If you don't see the secrets, wait a few minutes, and choose **Refresh**.
10. The café application's PHP code references these values (for example, so that it can retrieve the connection information for the MySQL database).
11. Choose the **/cafe/dbPassword** parameter.
12. Choose **Retrieve secret value**, and copy this value to your clipboard.
13. You use this value in a moment.
14. To configure the MySQL database to support the café application, in the AWS Cloud9 bash terminal, run the following commands:
15. cd ../db/
16. ./set-root-password.sh
17. ./create-db.sh

Next, you observe the database tables that were created.

1. To connect the terminal-based MySQL client to the database, run the following command:
2. `mysql -u admin -p`
3. When you are prompted for the database password, paste the **dbPassword** parameter value that you copied.
4. You should now see a **mysql>** prompt, which indicates that you are now connected to the MySQL database that runs on this EC2 instance.
5. To observe the contents of the database (specifically, the tables that support the café web application), run the following commands:
6. `show databases;`
7. `use cafe_db;`
8. `show tables;`
9. `select * from product;`
10. `exit;`



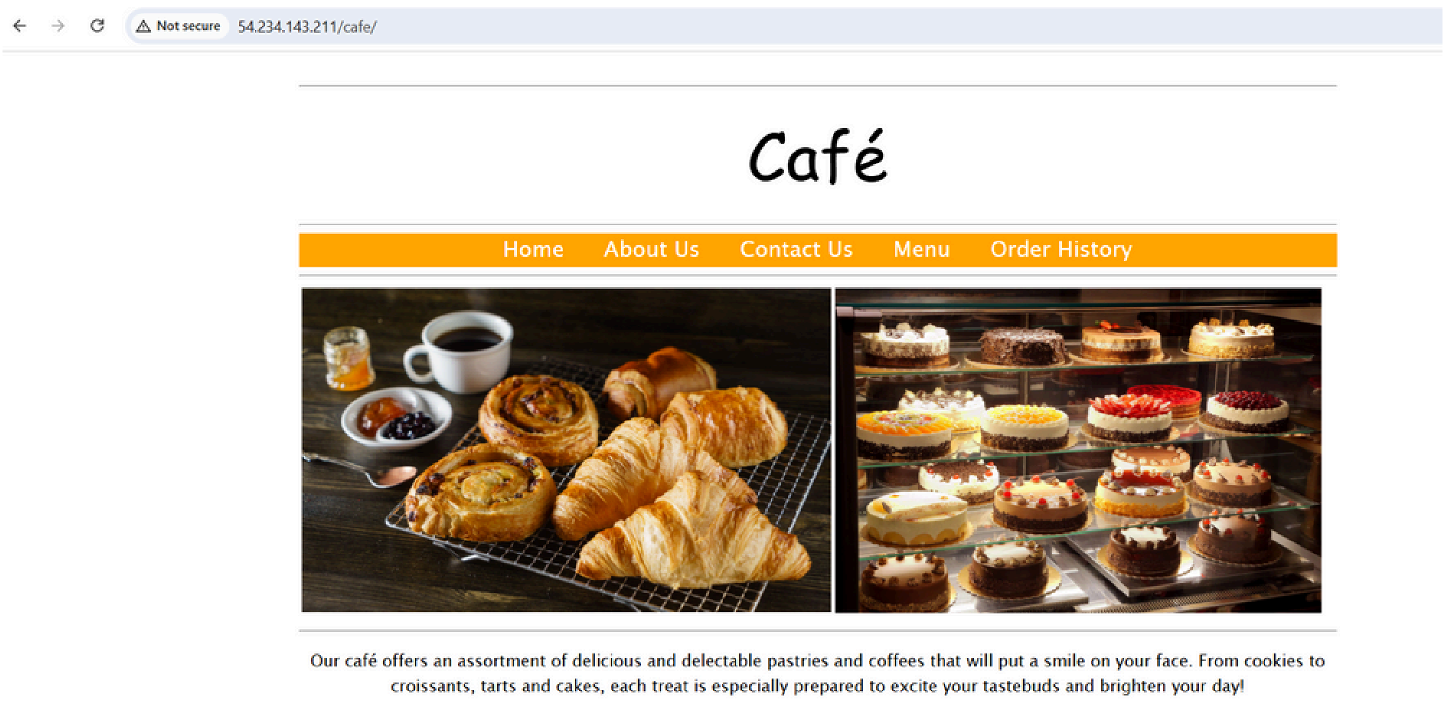
```
Database changed
MariaDB [cafe_db]> show tables;
+-------------------+
| Tables_in_cafe_db |
+-------------------+
| order             |
| order_item        |
| product           |
| product_group     |
+-------------------+
4 rows in set (0.000 sec)

MariaDB [cafe_db]> select * from product;
+----+---------------------------+-----------------------------------------------------------------+-------+---------------+---------------------------------------+
| id | product_name              | description                                                     | price | product_group | image_url                             |
+----+---------------------------+-----------------------------------------------------------------+-------+---------------+---------------------------------------+
|  1 | Croissant                 | Fresh, buttery and fluffy... Simply delicious!                  |  1.50 |             1 | images/Croissants.jpg                 |
|  2 | Donut                     | We have more than half-a-dozen flavors!                         |  1.00 |             1 | images/Donuts.jpg                     |
|  3 | Chocolate Chip Cookie     | Made with Swiss chocolate with a touch of Madagascar vanilla    |  2.50 |             1 | images/Chocolate-Chip-Cookies.jpg     |
|  4 | Muffin                    | Banana bread, blueberry, cranberry or apple                     |  3.00 |             1 | images/Muffins.jpg                    |
|  5 | Strawberry Blueberry Tart | Bursting with the taste and aroma of fresh fruit                |  3.50 |             1 | images/Strawberry-Blueberry-Tarts.jpg |
|  6 | Strawberry Tart           | Made with fresh ripe strawberries and a delicious whipped cream |  3.50 |             1 | images/Strawberry-Tarts.jpg           |
|  7 | Coffee                    | Freshly-ground black or blended Columbian coffee                |  3.00 |             2 | images/Coffee.jpg                     |
|  8 | Hot Chocolate             | Rich and creamy, and made with real chocolate                   |  3.00 |             2 | images/Cup-of-Hot-Chocolate.jpg       |
|  9 | Latte                     | Offered hot or cold and in various delicious flavors            |  3.50 |             2 | images/Latte.jpg                      |
+----+---------------------------+-----------------------------------------------------------------+-------+---------------+---------------------------------------+
9 rows in set (0.000 sec)

MariaDB [cafe_db]> exit;
```

1. To update the time zone configuration in PHP, in the bash terminal, run the following commands:
2. `sudo sed -i "2i date.timezone = \"America/New_York\" " /etc/php.ini`
3. `sudo service httpd restart`
4. The first command configures the time zone in the PHP software.
5. The second command restarts the web server so that the web server notices the configuration update.
6. To test whether the café website is working and can be accessed from the internet, in a new browser tab, enter http://<public-ip>/cafe and replace *<public-ip>* with the public IPv4 address of the EC2 instance.
7. Resolve the issue with the website.
8. In this step, you need to figure out how to make the café website function correctly.
9. Here's a list of what does work:
   - The test page at **http://<public-ip>/** loads, so you know that the web server works and is accessible from the internet.

- You also know that the MySQL database is running and contains tables and data to support the application.



Our café offers an assortment of delicious and delectable pastries and coffees that will put a smile on your face. From cookies to croissants, tarts and cakes, each treat is especially prepared to excite your tastebuds and brighten your day!

When you think you have fixed the issue, load the http://<public-ip>/cafe page again. Does it load completely so that you can see the café menu items? If so, congratulations!

Note: If you still can't solve the issue, you might find it helpful to run the grading script as documented in the **Submitting your work** section at the end of these lab instructions. The submission report that is generated can provide additional tips for parts of the lab that you didn't complete successfully. You can submit your work as many times as you like. Only the score that you achieve on the last submission will be retained.

Task 5: Testing the web application

In this task, you test placing an order.

1. In the browser tab where you have the http://<public-ip>/cafe page open, choose **Menu**.
2. Submit an order for at least one of the menu items displayed.
3. **Note:** You might need to scroll down to find the **Submit Order** button.
4. Return to the menu page and place another order.
5. Go to the **Order History** page to see the order details for all the orders that you placed.

New business requirement: Creating development and production websites in different AWS Regions (challenge #3)

Everyone at the café is impressed with the new dynamic website that Sofía created. Customers are delighted that they can now place online orders and schedule dessert items for pickup. Customer satisfaction has increased because of reduced wait times.

However, another business requirement emerges, along with the praise. Martha and Frank would like to have two café websites:

- One website that can be used as a development environment to mock up new features and web designs before they are released to customers
- A separate website that hosts the production environment that customers use

Sofía discusses the new requirement with Mateo, an AWS systems administrator and engineer, when he comes into the café one morning for his coffee. He suggests that, ideally, the two environments would exist in different AWS Regions. Such a design would have the added benefit of providing more robust disaster recovery (DR) in the unlikely scenario when an AWS Region becomes temporarily unavailable.

Sofía is now very busy! As she accomplishes more impressive work, her skills become more in-demand.

Task 6: Creating an AMI and launching another EC2 instance

Because the café website already runs well on an existing EC2 instance, Sofía decides to duplicate it by creating an AMI from it. She then launches a new instance from the new AMI.

You continue to take on the role of Sofía for this task. Before you create an AMI out of this instance, you should create a new key pair, which might be important to have later in this lab.

1. To set a static internal hostname and create a new key pair on the EC2 instance, in the bash terminal, run the following commands, ensure that you are on prompt *voclabs:~/environment $*, if not , used cd ..
2. `sudo hostname cafeserver`
3. `ssh-keygen -t rsa -f ~/.ssh/id_rsa`
4. For the two times that you are prompted for a passphrase, press Enter.
5. To make the new key available to the SSH utilities, in the bash terminal, run the following command:
6. `cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys`
7. In the AWS Management Console, browse to the Amazon EC2 console, and then select the instance.
8. Choose **Actions** > **Images and templates** > **Create image**.
9. **Tip**: Leave the **Create Image** dialog open in the browser tab while you answer some questions about AMIs.

Answering questions about AMIs

Your answers are evaluated when you choose the **Submit** button at the end of the lab.

1. Return to the browser tab that has the questions for this lab. You accessed this tab earlier.
2. If you need to find the page again, follow these steps:

  - At the top of these instructions, choose **AWS Details**.
  - Choose the **Access the multiple choice questions** link.
3. On the page, submit answers to the following questions:
   - **Question 5**: When you create an AMI from an instance, will the instance be rebooted?
   - **Question 6**: In what ways can you modify the root volume properties when you create an AMI from an instance?
   - **Question 7**: Can you add more volumes to an AMI that you create from an instance that only has one volume?

Next, you configure the new AMI.

1. On the Amazon EC2 console, in the **Create Image** page, for **Image name**, enter CafeServer.
2. Choose **Create Image**.
3. From the navigation menu, choose **AMIs**, and wait until the image status becomes *Available*.

On the navigation menu, you might need to expand **Images** to find **AMIs**.

The process typically takes about 2 minutes.

1. Create an AMI in another AWS Region.
2. In this step, your objective is to create a new EC2 instance from the AMI that you just captured. However, you must create the new instance in the Oregon (us-west-2) AWS Region.

Expand each tip for help solving the challenge.

Copy the AMI ID of the image that you just created. Then, try to find it in the Oregon (us-west-2) Region.Select the AMI you that you created in the AWS Region where you created it. Next, choose the **Actions** menu. Do any actions seem like they could help you make the AMI available in the Oregon (us-west-2) Region? Choose the appropriate action. After you initiate it, the action might take up to 5 minutes to complete. Choose the refresh icon occasionally to check when it has completed.

Next, you create the new café instance from your AMI.

1. To create the new café instance from your AMI, make sure you are in the Oregon Region, and configure the following options:
   - For **Name and tags**, choose **Add additional tags**, and configure the following options:
     - For **Key**, enter Name.
     - For **Value**, enter ProdCafeServer.
   - For **Instance type**, choose **t2.small**.
   - For **Key pair (login)**, choose **Proceed without a key pair**. The key pair that you created earlier in this lab should work to connect to it if necessary.
   - For **Network settings**, choose **Edit**, and configure the following options:
     - For **VPC - *required***, choose **Lab VPC Region 2**.

        - For **Subnet**, choose **Public Subnet**.
        - For Security **Security group name**, enter cafeSG, and configure the following options:
            - Set TCP port **22** so that it is open to anywhere.
            - Set TCP port **80** so that it is also open to anywhere.
    - In the **Advanced details** section, for **IAM instance profile**, choose **CafeRole**.
2. Choose **Launch instance**.

Wait for the new instance to have a **Public IPv4 DNS** value assigned to it even if the status of the instance is still not *Available*.

1. Copy the **Public IPv4 DNS** value. You use it soon.

Next, you create the needed Secrets Manager secrets in the new AWS Region.

1. Return to the AWS Cloud9 IDE in the **N. Virginia (us-east-1)** Region, and open the **CafeWebServer/setup/set-app-parameters.sh** file in the text editor.
2. Edit line 15 of the file to match this setting:
3. `region="us-west-2"`
4. Edit line 21 to match this setting (where *<public-dns-of-ProdCafeServer-instance>* is the DNS of the **ProdCafeServer** instance):
5. `publicDNS="<public-dns-of-ProdCafeServer-instance>"`

Note: The line should still contain the quotation marks, but it should not contain the brackets (< >).

This example shows what line 15 should look like and how line 21 should be formatted. However, the value of your public DNS will be different.

1. Choose **File** > **Save**.
2. To run this script, go to the top of the IDE, and choose the **Run** button.

In the bash terminal below the text editor, you should see output that's formatted in JSON. This output indicates that the parameters script ran successfully.

If the script encountered an issue, review this information for troubleshooting tips.

Note: By changing the AWS Region details and running this script again, you create the same parameters that you created earlier in the us-east-1 Region of Secrets Manager. However, this time, you created these parameters in the Oregon Region.

Task 7: Verifying the new café instance

1. Return to the EC2 Console in the **Oregon** Region, and verify that the new **ProdCafeServer** instance is running.
2. Copy the **Public IPv4 address**, and load it in a web browser.
3. The *Hello from the cafe web server!* message should display.

4. Load the http://<public-ip>/cafe/ URL in a browser tab.
5. The entire café website should display.
6. Load the **Menu** page.
7. The full **Menu** page should load, and the order-placing functionality should work.
8. Place an order to verify that the website is working as intended.
9. If you encountered issues loading the **Menu** page, try these troubleshooting tips.
10. The grading script can provide additional tips for parts of the lab that you didn't complete successfully. You can submit your work as many times as you like. Only the score that you achieve on the last submission will be retained.
11. If you want to connect to the new EC2 instance in the Oregon (us-west-2) Region to do some troubleshooting, run the following command from the AWS Cloud9 IDE in us-east-1:
12. `ssh -i ~/.ssh/id_rsa ec2-user@<public-ip-of-ProdCafeServer>`
13. Note that *<public-ip-of-ProdCafeServer>* is the public IP address of the **ProdCafeServer** instance.

Update from the café

Sofía is now a hero at the cafe. She created a dynamic website, and she also created a duplicate version of the same website that runs in a second AWS Region.

Sofía decides to designate the first EC2 instance that she created—the one in the us-east-1 Region—as the development instance. The second instance she created—the one in the Oregon (the us-west-2) Region—is the production instance.

This way, Sofía and any other application developers can test application enhancements on the development site without affecting the production site. Then, when the developers decide that the enhancements look good and they have fully tested them, they can migrate the code to the production site.

Sofía explained to Frank and Martha what she had done. They were pleased to know that the website can now take online orders. They were also glad to hear that they can now test new enhancements to the website without immediately exposing those changes to customers.

Conclusion

Congratulations, you have successfully done the following:

- Connected to the AWS Cloud9 IDE on an existing EC2 instance
- Analyzed the EC2 instance environment and confirmed web server accessibility
- Installed a web application on an EC2 instance that also uses Secrets Manager
- Tested the web application
- Created an AMI
- Deployed a second copy of the web application to another AWS Region