**Accessing the AWS Management Console**

1. At the top of these instructions, choose **Start Lab** to launch your lab.
2. A **Start Lab** panel opens and displays the lab status.

### **Task 1: Preparing the Development Environment**

#### **Connect to AWS Cloud9 IDE**

1. In the **AWS Management Console**, open **Cloud9** from the **Services** menu.
2. Open the **Cloud9 Instance** IDE.

#### **Download and Extract Files**

Run the following command in the Cloud9 terminal:  
wgethttps://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2-91558/04-lab-api/code.zip -P /home/ec2-user/environment

Extract the file:  
unzip code.zip

#### **Run the Setup Script**

Run the following commands to upgrade Python, the AWS CLI, and set up the café website:  
chmod +x resources/setup.sh && resources/setup.sh

When prompted, visit **whatismyip.com** to find your IP address and enter the IPv4 address.

#### **Verify Environment Setup**

Check the AWS CLI version:  
aws --version

Ensure version 2 is installed.

Verify the **SDK for Python** is installed:  
pip show boto3

#### **Verify the Café Website**

1. In the **Amazon S3 Console**, navigate to your bucket, select **Objects**, and refresh if needed.
2. Choose the index.html file and copy its **Object URL** (e.g., https://<bucket-name>.s3.amazonaws.com/index.html).
3. Paste the URL in a browser to confirm the website loads correctly.

### **Task 2: Creating the First API Endpoint (GET)**

#### **Define the Products API**

1. In Cloud9, open python\_3/create\_products\_api.py.
2. Replace (fill me in) on line 3 with the correct API Gateway client value (refer to the Python SDK documentation).
3. Analyze the script:
   * Lines 5–24 create the **ProductsApi** REST API and a products resource.
   * Lines 28–33 define the GET method request.

Save the file and run the script:  
cd python\_3

python3 create\_products\_api.py

#### **Test the API in API Gateway**

1. Open the **API Gateway Console** and select the **ProductsApi** you just created.
2. Choose the **GET** method under the products resource.
3. Click the **TEST** link, scroll down, and click **Test**.

### **Task 3: Creating the Second API Endpoint (GET)**

#### **Define the /on\_offer Resource**

1. In Cloud9, open python\_3/create\_on\_offer\_api.py.
2. Replace <FMI\_1> and <FMI\_2> with the appropriate **api\_id** and **parent\_id**.

Save and run the script:  
python3 create\_on\_offer\_api.py

#### **Test the /on\_offer Resource**

1. Open **API Gateway Console** > **ProductsApi** > **/on\_offer > GET**.
2. Use the **TEST** link to verify the mock response.

### **Task 4: Creating the Third API Endpoint (POST)**

#### **Define the /create\_report Resource**

#### Open python\_3/create\_report\_api.py.

Replace <FMI\_1> with the **api\_id** (use the CLI if needed):  
aws apigateway get-rest-apis --query items[0].id --output text

Save and run the script:  
python3 create\_report\_api.py

#### **Test the /create\_report Resource**

1. Open **API Gateway Console** > **ProductsApi** > **/create\_report > POST**.

### **Task 5: Deploying the API**

1. In **API Gateway Console**, select the root / under **Resources**.
2. Choose **Actions > Deploy API** and fill in:
   * Deployment stage: **[New Stage]**
   * Stage name: **prod**
3. Click **Deploy** and copy the **Invoke URL**.

### **Task 6: Updating the Website**

#### **Update the Config.js File**

1. In Cloud9, open resources/website/config.js.
2. Replace null with the **Invoke URL** (surrounded by quotes).
3. Save the file.

#### **Update the Config File on S3**

#### Open python\_3/update\_config.py.

Replace <FMI\_1> with your bucket name (use the CLI to find it):  
aws s3 ls

Save and run the script:  
python update\_config.py

#### **Test the Website**

1. Refresh the **Café website** in your browser.
2. Verify the website now uses the API:

### **Submitting Your Work**