### **Accessing the AWS Management Console**

## **At the top of these instructions, choose Start Lab to launch your lab.**

## **A Start Lab panel opens and displays the lab status.**

## **Wait until you see the message Lab status: ready, and then close the Start Lab panel by choosing the X.**

## 

### **Task 1: Preparing the Lab**

## **Connect to the AWS Cloud9 IDE:**

## **From the Services menu, search for and select Cloud9.**

## **Locate the existing IDE named Cloud9 Instance.**

## **In the Cloud9 Instance pane, choose Open IDE.**

## **Download and extract required files:**

## **Run the following command in the terminal: wget https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/CUR-TF-200-ACCDEV-2-91558/03-lab-dynamo/code.zip -P /home/ec2-user/environment**

## **Extract the file: unzip code.zip**

## **Upgrade Python and AWS CLI versions:**

## **Run the following commands to set permissions and execute the script: chmod +x ./resources/setup.sh && ./resources/setup.sh**

## 

### **Task 2: Creating a DynamoDB Table Using the SDK for Python**

## **Open the DynamoDB console via Services > DynamoDB.**

## **From the DynamoDB navigation pane, choose Tables.**

## **Edit the script to create the table:**

## **In the AWS Cloud9 IDE, expand the python\_3 directory.**

## **Open the create\_table.py script.**

## **Replace <FMI\_1> with the table name: FoodProducts**

## **Save the file: File > Save.**

## **Run the script:**

## **Navigate to the python\_3 directory and execute: cd python\_3**

## **python3 create\_table.py**

### **Task 3: Working with DynamoDB Data – Understanding Condition Expressions**

## **Insert a new record:**

## **In the AWS Cloud9 IDE, expand the resources folder.**

## **Open the not\_an\_existing\_product.json file.**

## **Run the following command to insert the record: aws dynamodb put-item \**

## **--table-name FoodProducts \**

## **--item file://../resources/not\_an\_existing\_product.json \**

## **--region us-east-1**

## **View the new record:**

## **In the DynamoDB console, go to Item explorer.**

## **Confirm that FoodProducts is selected, choose Scan, and click Run.**

## **Review the returned items.**

## 

### **Task 4: Adding and Modifying a Single Item Using the SDK**

## **In the AWS Cloud9 IDE, open the conditional\_put.py script in the python\_3 directory.**

## **Replace the <FMI> placeholders as directed.**

## **Save the file: File > Save.**

## 

### **Task 5: Adding Multiple Items Using the SDK and Batch Processing**

## **In the DynamoDB Item explorer, refresh the view by choosing Run.**

## **Delete all records:**

## **Select all table records.**

## **From the Actions menu, choose Delete item(s).**

## **In the confirmation box, type Delete and choose Delete items.**

## **Update the batch load script:**

## **Open the test\_batch\_put.py script in the python\_3 directory.**

## **Replace <FMI\_1> with FoodProducts and <FMI\_2> with the primary key name product\_name.**

## **Save the file.**

## **Run the script:**

## **Execute: python3 test\_batch\_put.py**

## **Modify batch\_put.py:**

## **Replace <FMI> with FoodProducts.**

## **Save the file and run it.**

## 

### **Task 6: Querying the Table Using the SDK**

## **Open the get\_all\_items.py script in the python\_3 directory.**

## **Replace <FMI\_1> with FoodProducts.**

## **Save the file and run: python3 get\_all\_items.py**

## **Update get\_one\_item.py:**

## **Replace <FMI\_1> with the table's primary key name.**

## **Save the file and run: python3 get\_one\_item.py**

## 

### **Task 7: Adding a Global Secondary Index to the Table**

## **Update add\_gsi.py:**

## **Replace <FMI\_1> with the KeyType of HASH.**

## **Save the file and run: python3 add\_gsi.py**

## **Monitor index status:**

## **In the DynamoDB console, select Tables > FoodProducts > Indexes tab.**

## **Wait until the status changes from Creating to Active.**

## **Update scan\_with\_filter.py:**

## **Replace <FMI\_1> with special\_GSI and <FMI\_2> with tags.**

## **Save the file and run: python3 scan\_with\_filter.py**

## 

### **Submitting Your Work**