Develop and Deploy Application for No SQL Operation

(LAB-M07-02)

Version Control	
Document	Develop and Deploy Web Application for Database CRUD Operation
Owner	Ahmad Majeed Zahoory
Version	2.1
Last Change	24th August 2023
Description of Change	Task steps updated

Lab duration: **60 minutes**

Lab scenario

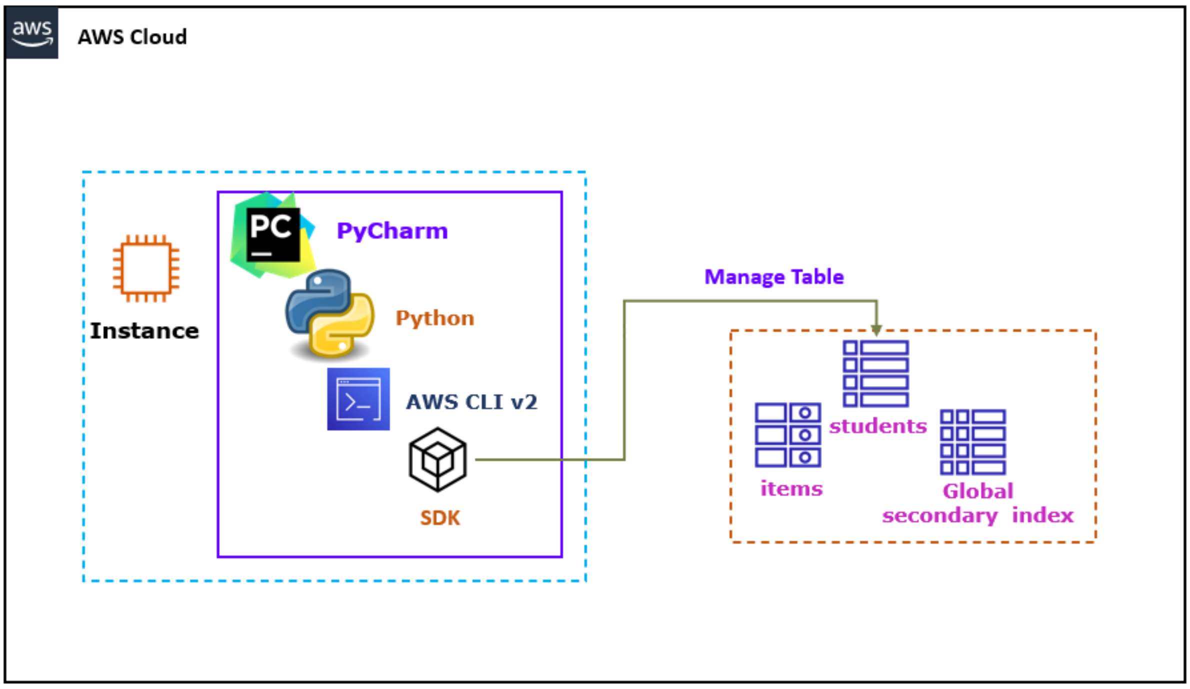
You are preparing to store data in AWS. As a development group, your team has decided to use Python to manage the data from AWS DynamoDB programmatically.

Objectives

After you complete this lab, you will be able to:

- Create DynamoDB Table.
- Create Items in DynamoDB Table.
- Read Item from DynamoDB Table.
- Update Items in DynamoDB Table.
- Delete Items in DynamoDB Table.
- Scan Items from DynamoDB Table.
- Query Items from DynamoDB Table.
- Create DynamoDB Table with GSI.
- Create DynamoDB Table with LSI.

Ahmad Majeed Zahoory



Ahmad Majeed

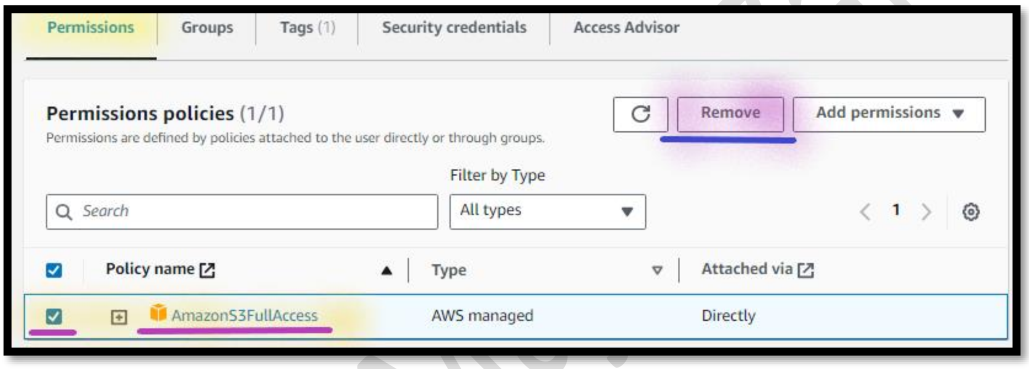
© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Task 1: Update IAM Role

In this task, you will update the AWS IAM role with permission to manage the DynamoDB.

Step 1: Update the IAM User Permission

1. In the **AWS Management Console**, on the **Services** menu, click **IAM**.
2. Select **Users**.
   a. Open the **Dev-User-YOUR NAME**.
      i. Select **Permissions**.
         a) Select **AmazonS3FullAccess**.
            1) Select **Remove**.

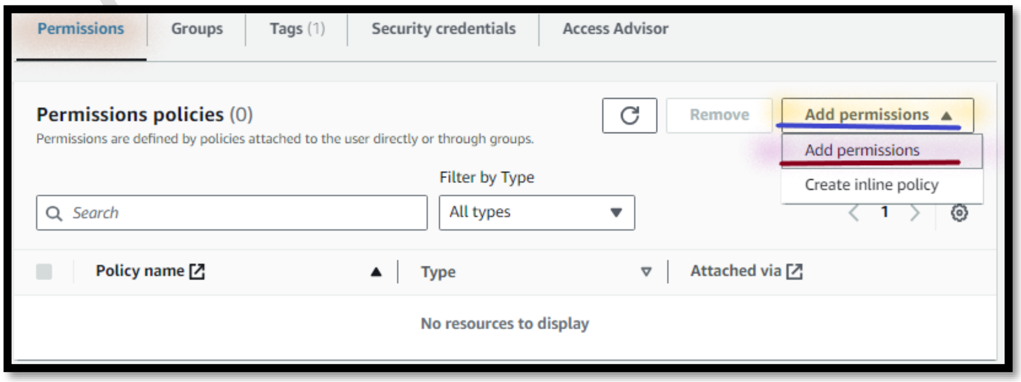


Permissions policies (1/1)
Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via
AmazonS3FullAccess	AWS managed	Directly

I. Select **Remove Policy**.

3. From the **Dev-User-YOUR NAME** console:
   a. Select **Permissions**.
      i. Select **Add permissions**.
         a) Select **Add permissions**.



Permissions policies (0)
Permissions are defined by policies attached to the user directly or through groups.

Policy name	Type	Attached via

No resources to display

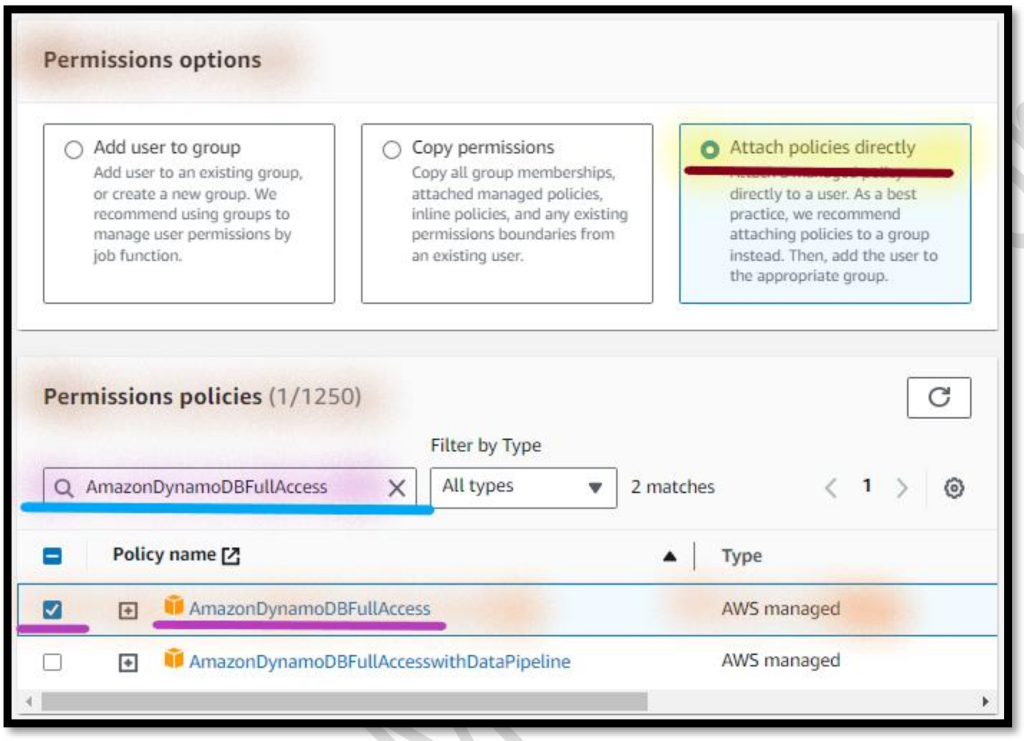
© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

b. In the **Add permissions** page:

i. **Permissions options**: Select **Attach policies directly**.

ii. **Permissions policies**:

a) Search and Select **AmazonDynamoDBFullAccess**.



Permissions options

Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.

Copy permissions
Copy all group memberships, attached managed policies, inline policies, and any existing permissions boundaries from an existing user.

Attach policies directly
directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

Permissions policies (1/1250)

Filter by Type

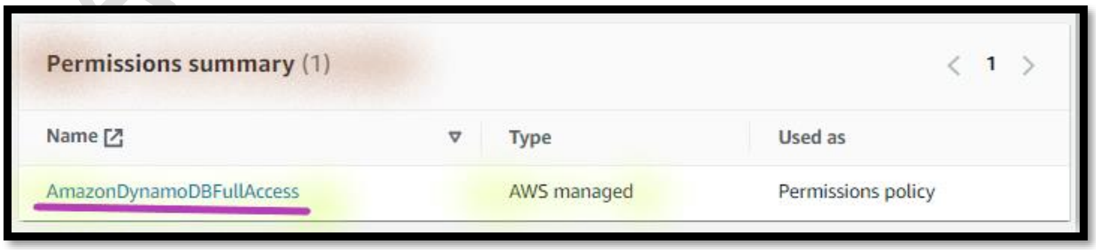
AmazonDynamoDBFullAccess | All types | 2 matches

	Policy name	Type
✓	AmazonDynamoDBFullAccess	AWS managed
	AmazonDynamoDBFullAccesswithDataPipeline	AWS managed

iii. Select **Next**.

c. In the **Review** page:

Note: You can see the **AmazonDynamoDBFullAccess** under the **Permissions summary**.



Permissions summary (1)

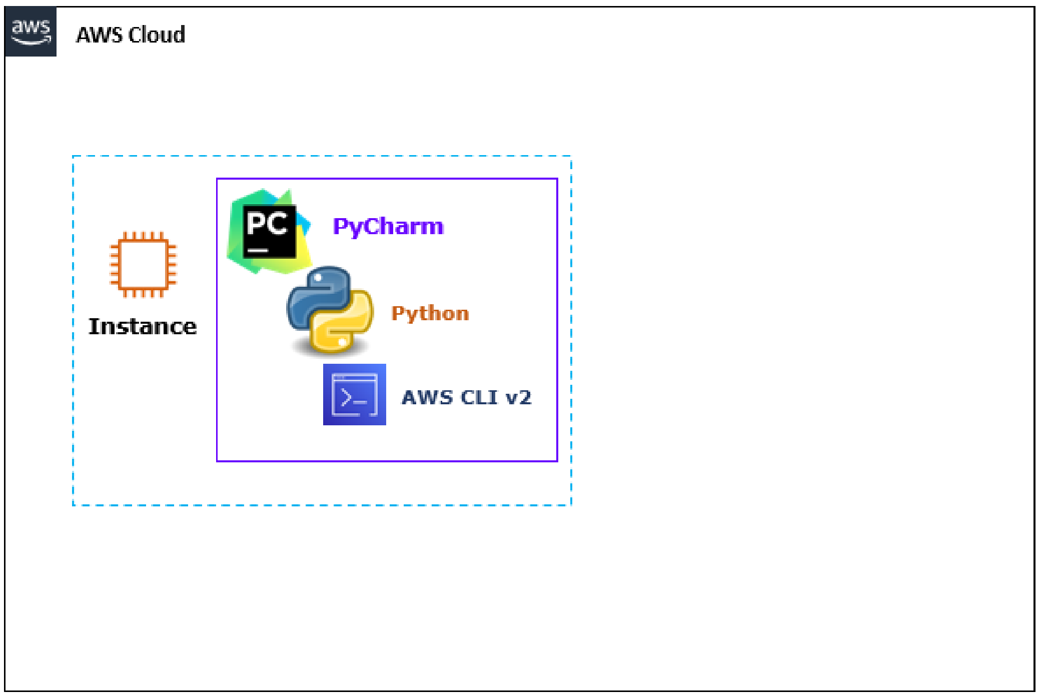
Name	Type	Used as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy

i. Select **Add permissions**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Task 2: Build Server for Development Environment

In this task, you will build the AWS Virtual machine to build development environment and install Python, PyCharm and AWS CLI.



Step 1: Create EC2 Instances

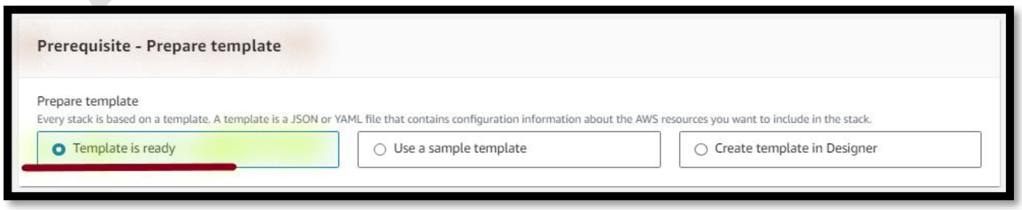
4. In the **AWS Management Console**, on the **Services** menu Search and Select **CloudFormation**.

5. Choose the **YOUR ALLOCATED REGION**, region list to the right of your account information on the navigation bar.

6. Select **Create stack** and configure:

   a. In the **Create stack** page:

      i. **Prepare template**: Select **Template is ready**.



Prerequisite - Prepare template

Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

[✓ Template is ready] Use a sample template Create template in Designer

      ii. **Template source**: Select **Upload a template file**.

      iii. **Choose file**: Click on **Choose file**.

         a) **Navigate** and **select** the **Dev-Instance**.yaml file.

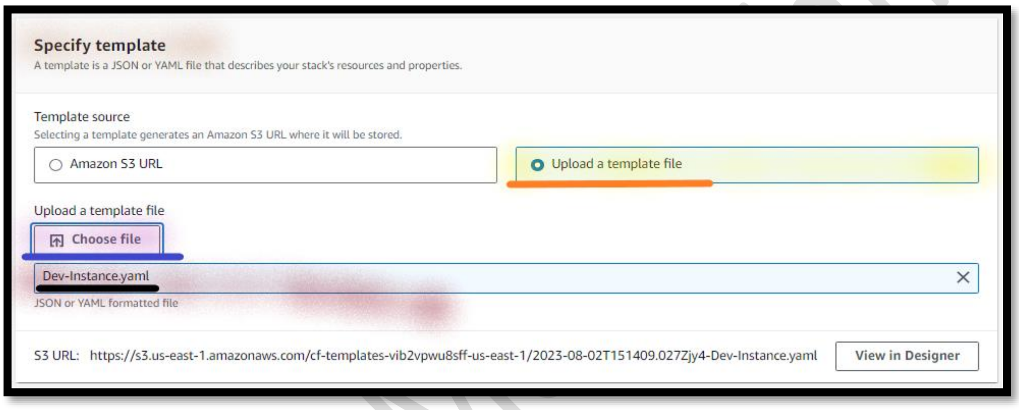
© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Note: **Dev-Instance**.yaml template is provided with the Lab manual.

Note: AWS template **performing** the **following** tasks:

1. Creating **Windows instances**.
2. Creating **t2.medium** instance (*2 vCPU and 4 GB*) [*This instance type attarct charges*].
3. Set the "**Administrator**" **password**.

Note: You can also use **t2.micro**, but the **performance will be low** to build development environment.



Specify template

A template is a JSON or YAML file that describes your stack's resources and properties.

Template source

Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL [✓ Upload a template file]

Upload a template file

Choose file

Dev-Instance.yaml

JSON or YAML formatted file

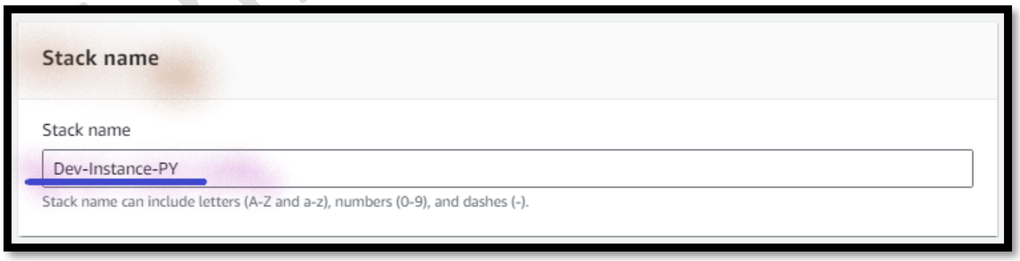
S3 URL: https://s3.us-east-1.amazonaws.com/cf-templates-vib2vpwu8sff-us-east-1/2023-08-02T151409.027Zjy4-Dev-Instance.yaml

View in Designer

iv. Select **Next**.

b. In the **Specify stack details** page:

i. **Stack name**: Write **Dev-Instance-PY**.



Stack name

Stack name

Dev-Instance-PY

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Note: Leave other details as default.

ii. Select **Next**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

c. In the **Configure stack options** page:

Note: Leave all the details as default.

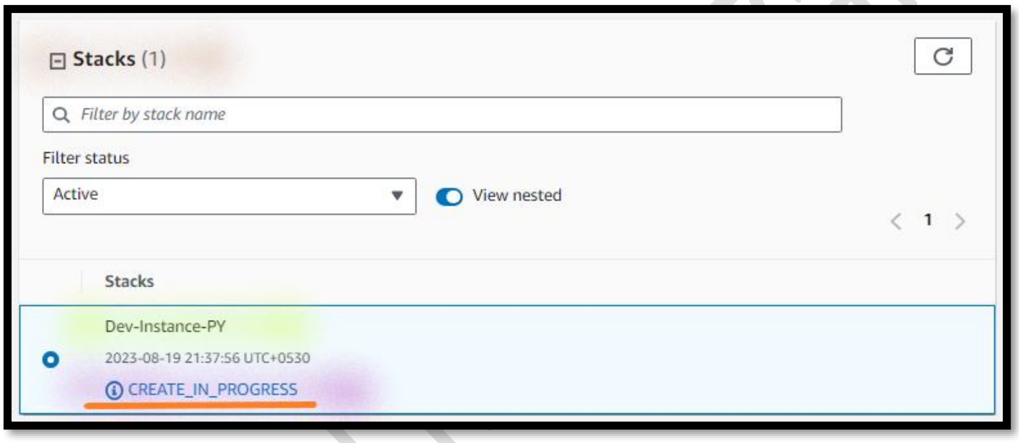
i. Select **Next**.

d. In the **Review Dev-Instance-PY** page:

Note: Review all the details.

i. Select **Submit**.

Note: You can see the **Stack** status as **CREATE_IN_PROGRESS**.


Stacks (1)

Filter by stack name

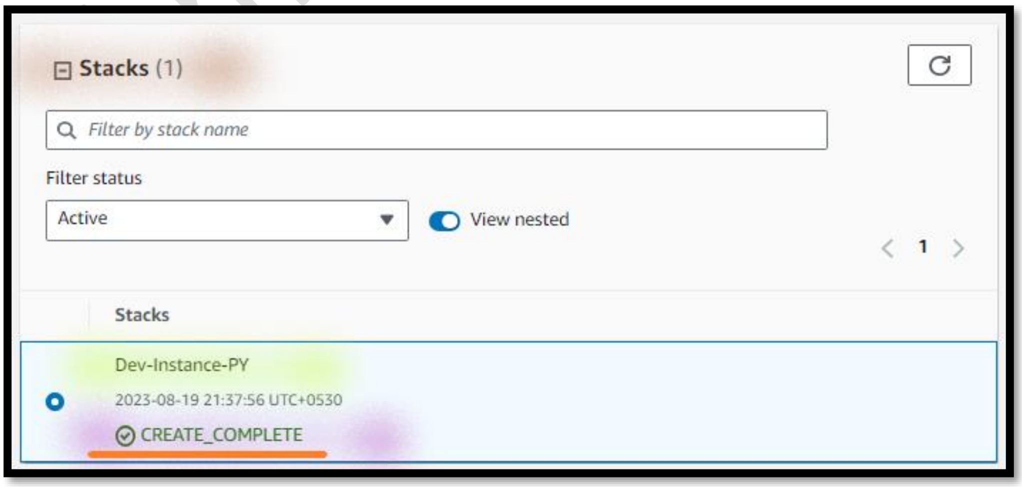
Filter status: Active

View nested

Stacks

Dev-Instance-PY
2023-08-19 21:37:56 UTC+0530
CREATE_IN_PROGRESS

Note: **Wait**, till you can see the **Stack** status as **CREATE_COMPLETE**. You can **Refresh** your screen


Stacks (1)

Filter by stack name

Filter status: Active

View nested

Stacks

Dev-Instance-PY
2023-08-19 21:37:56 UTC+0530
CREATE_COMPLETE

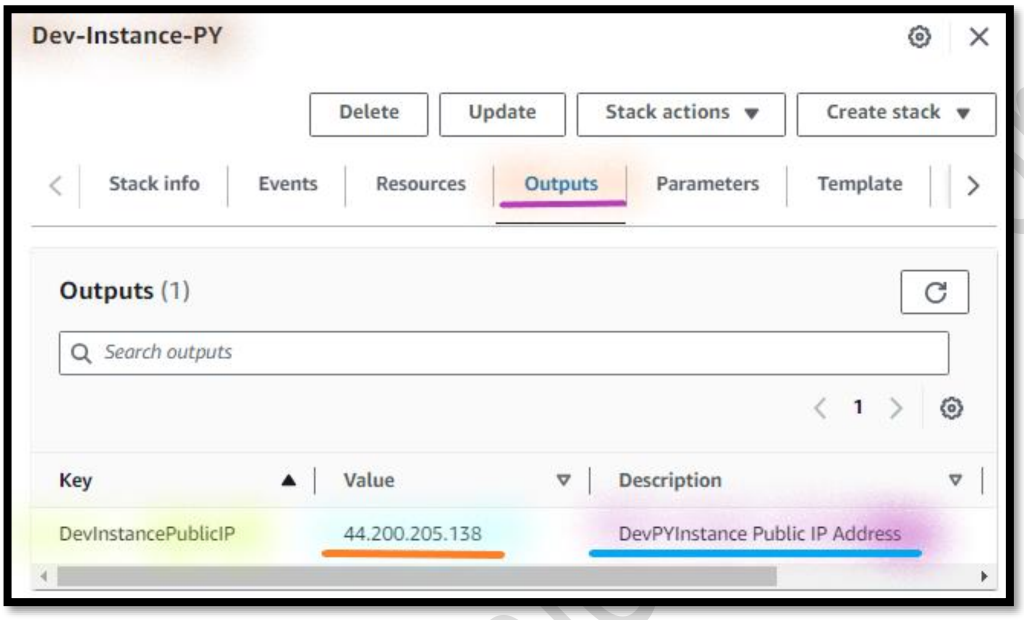
© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Step 2: View the Output

7. From the Dev-Instance-PY CloudFormation console:

   a. Select Outputs.

Note: **Copy** the **DevPYInstance Public IP** address in the **Notepad**.



Key	Value	Description
DevInstancePublicIP	44.200.205.138	DevPYInstance Public IP Address

Step 3: Connect to Instance

8. From the Local Desktop/ Laptop (Windows Desktop), right click on Start & Run.

   a. In the Open, write mstsc.

   b. Select Ok.

      i. From the Remote Desktop Connection:

         a) **Computer**: Write the Public IP Address of the DevPYInstance.

         b) Select Connect.

Note: You can **get the prompt** to enter the **Username** and **Password**.

Ahmad Mahmoody

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

1) **Username:** Write **Administrator**.

2) **Password**: Write **lab-password@123**.

3) Select **Ok**.

Step 4: Install the Python

9. From the **DevPYInstance** (*Windows Server 2022*).

   a. **Download** and **install** the ***Python*** for **Windows x64**.

Note: Use the below URL to download the **Python 3.11** for **Windows**.

https://bitbucket.org/ahmadzahoory/aws-sdk/downloads/python-3.11.4-amd64.exe

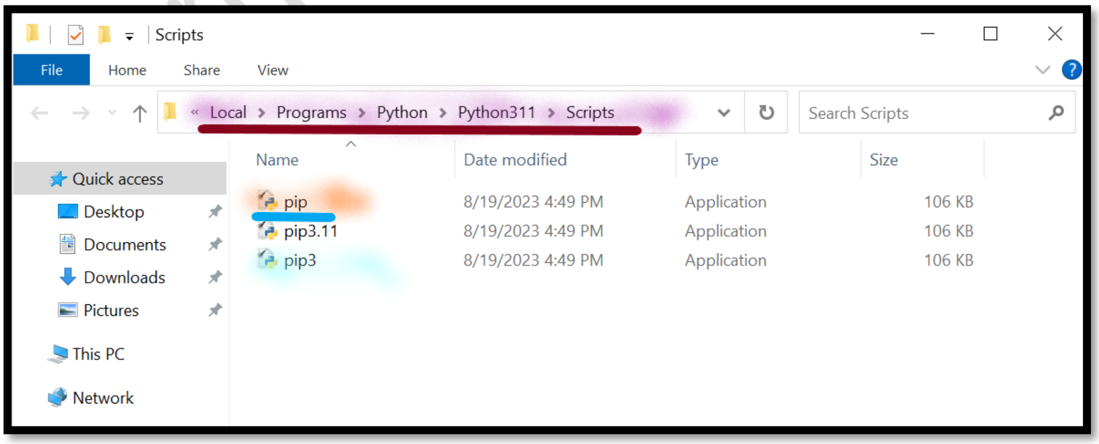
Note: **Wait**, till **Python** install **succesfully**.

Step 5: Configure the Environment

10. From the **DevPYInstance**, right click on **Start** & **Run**.

    a. In the **Open**, write **C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Scripts**.

Note: You can see the **PIP** executables.



Name	Date modified	Type	Size
pip	8/19/2023 4:49 PM	Application	106 KB
pip3.11	8/19/2023 4:49 PM	Application	106 KB
pip3	8/19/2023 4:49 PM	Application	106 KB

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

11. From the **DevPYInstance**, right click on **Start** & **Run**.

    a. In the **Open**, write **cmd**.

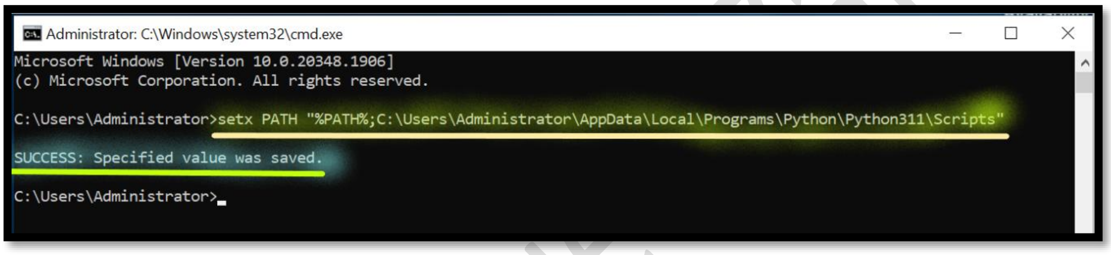
    b. Select **Ok**.

       i. From the **Command line interpreter**:

          a) **Execute** the ***below command*** to **set** the **Environment variables**:

```
setx PATH "%PATH%;C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Scripts"
```

Note: You can see the **Sucess** message.



```
Microsoft Windows [Version 10.0.20348.1906]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>setx PATH "%PATH%;C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Scripts"

SUCCESS: Specified value was saved.

C:\Users\Administrator>_
```

Step 6: Check the Python and Pip Version

12. From the **DevPYInstance**, right click on **Start** & **Run**.

    a. In the **Open**, write **cmd**.

    b. Select **Ok**.

       i. From the **Command line interpreter**:

          a) **Execute** the ***below command*** to **verify** the **Python version**:

```
py --version
```

Note: You can see the **Python** installed **version**.

Ahmad Majeed Zahoory

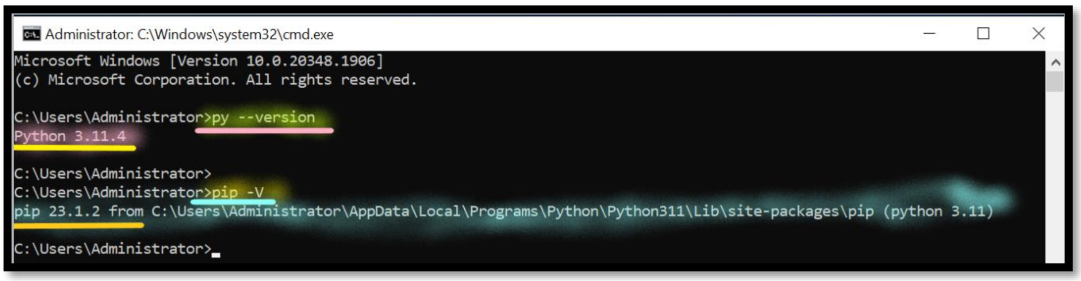
© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

ii. From the **Command line interpreter**:

a) **Execute** the ***below command*** to **verify** the **PIP version**:

```
pip -V
```

Note: You can see the **Pip** installed **version**.



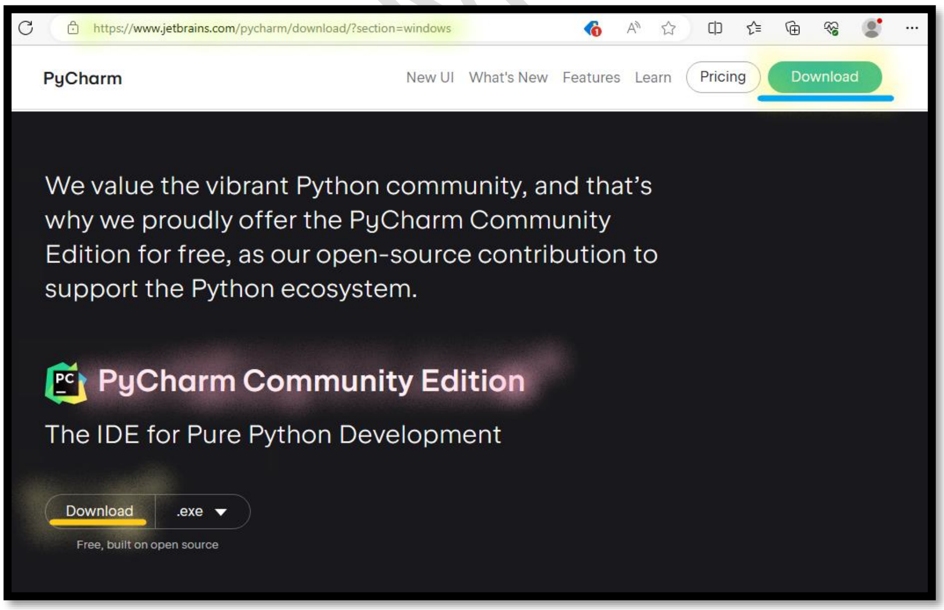
```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.20348.1906]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>py --version
Python 3.11.4

C:\Users\Administrator>
C:\Users\Administrator>pip -V
pip 23.1.2 from C:\Users\Administrator\AppData\Local\Programs\Python\Python311\Lib\site-packages\pip (python 3.11)

C:\Users\Administrator>_
```

Step 7: Install the PyCharm IDE

13. **Download** and **Install** the ***PyCharm IDE*** for **Community Edition**.

Note: Use the below URL to download the **PyCharm IDE**.

https://www.jetbrains.com/pycharm/



https://www.jetbrains.com/pycharm/download/?section=windows

We value the vibrant Python community, and that's why we proudly offer the PyCharm Community Edition for free, as our open-source contribution to support the Python ecosystem.

PyCharm Community Edition

The IDE for Pure Python Development

Note: **Wait**, till **PyChram IDE** install **succesfully**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Note: **Don't launch** the **PyCharm IDE**.

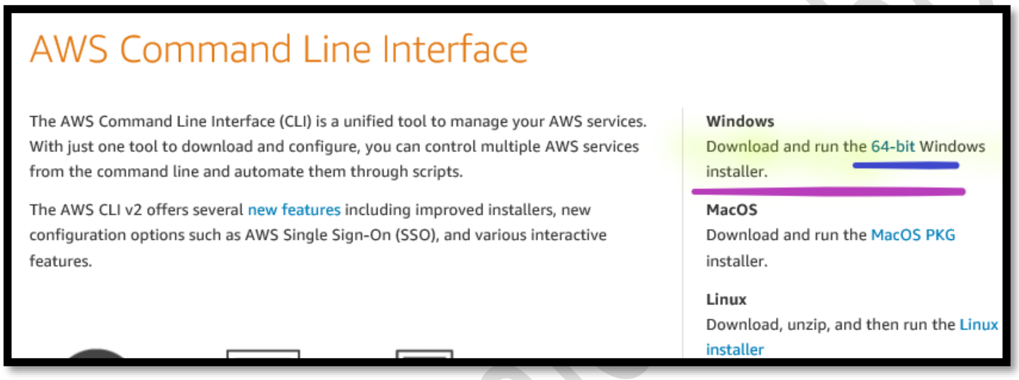
Step 8: Install the AWS CLI V2

14. From the **DevPYInstance**.

    a. **Download** and **install** the ***AWS CLI v2***.

Note: Use the below URL to download the **AW CLI v2**.

https://aws.amazon.com/cli/



AWS Command Line Interface

The AWS Command Line Interface (CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

The AWS CLI v2 offers several new features including improved installers, new configuration options such as AWS Single Sign-On (SSO), and various interactive features.

Windows
Download and run the 64-bit Windows installer.

MacOS
Download and run the MacOS PKG installer.

Linux
Download, unzip, and then run the Linux installer

Note: **Wait**, till **AWS CLI v2** install **succesfully**.

Check the AWS CLI Version

15. From the **DevPYInstance**, right click on **Start** & **Run**.

    a. In the **Open**, write **cmd**.

    b. Select **Ok**.

        i. From the **Command line interpreter**:

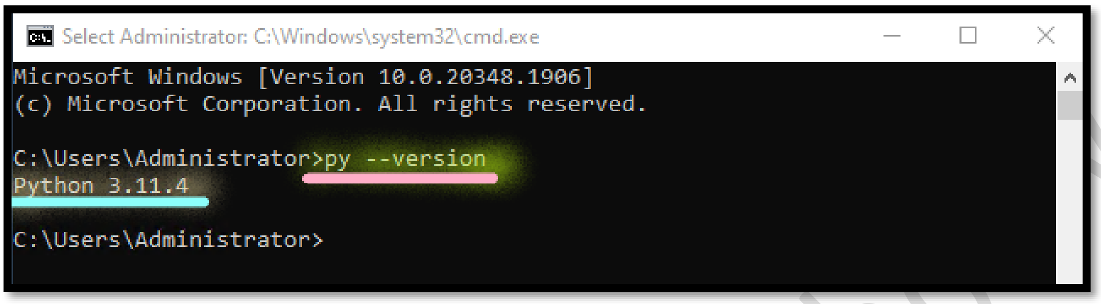
            a) **Execute** the ***below command*** to **verify** the **AWS version**.

```
aws --version
```

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Note: You can see the **AWS CLI** installed **version**.

Note: If you can see the "**'aws' is not recognized as an internal or external command**" message, **Restart** the **DevPYInstance**.



Select Administrator: C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.20348.1906]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Administrator>py --version
Python 3.11.4

C:\Users\Administrator>
```

Step 9: Configure the Credentials and Configuration

16. From the **DevPYInstance**, right click on **Start** & **Run**.
    a. In the **Open**, write **cmd**.
    b. Select **Ok**.
        i. From the **Command line interpreter**:
            a) **Execute** the ***below command*** to **configure** the **AWS credentials**.

```
aws configure
```

a) **AWS Access Key ID**: Type **Dev-User-YOUR NAME access key**, press **Enter** key to continue.

b) **AWS Secret Access Key**: Type **Dev-User-YOUR NAME secret access key**, press **Enter** key to continue.

Note: Copy the **access key** and **secret access key** of the IAM user **Dev-User** from **.csv file** which you have downloaded in the previous step.

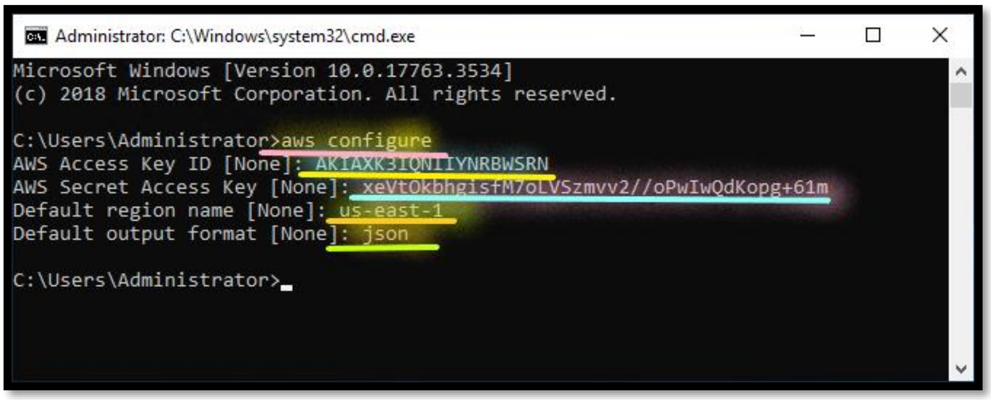
c) **Default region name**: Type **YOUR ALLOCATED REGION CODE**, press **Enter** key to continue.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Ahmed Majeed Zahoory

Note: **Refer** the **link** to know your **respective region Code**
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/using-regions-availability-zones.html

d) **Default output format**: Type **json**, press **Enter** key to continue.


Administrator: C:\Windows\system32\cmd.exe

```
Microsoft Windows [Version 10.0.17763.3534]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>aws configure
AWS Access Key ID [None]: AKIAXK3IQNIIYNRBWSRN
AWS Secret Access Key [None]: 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
Default region name [None]: us-east-1
Default output format [None]: json

C:\Users\Administrator>_
```

b) **Execute** the ***below command*** to **exit**.

```
exit
```

Step 10: Verify the Configuration

17. From the **DevJPYInstance**, right click on **Start** & **Run**.

   a. In the **Open**, write **C:\Users\Administrator**.

   b. Select **Ok**.

      i. From the **File explorer**:

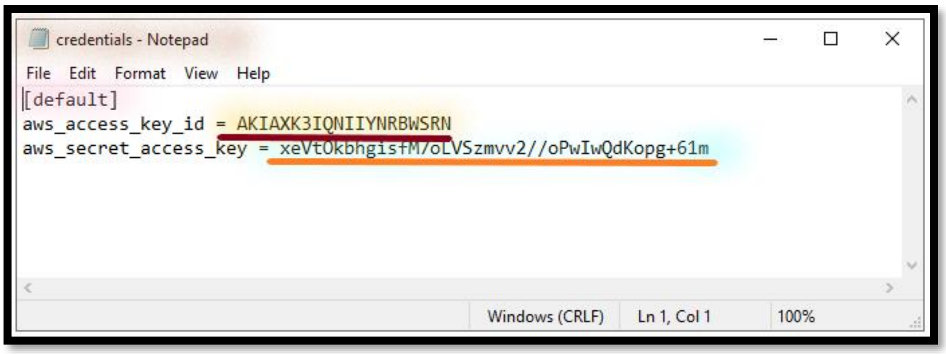
         a) **Open** the **.aws** folder.

         b) **Open** the **Credentials** file in **Notepad**.

Note: You can see the **access key** and **secret access key** details.

Ahmad Majeed Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.


credentials - Notepad

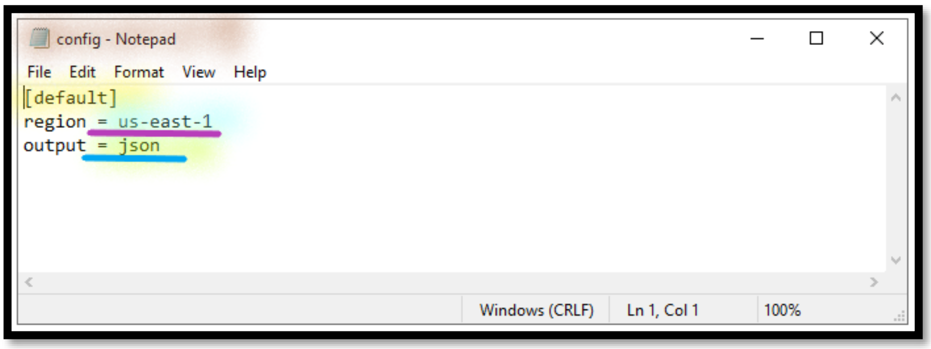
```
[default]
aws_access_key_id = AKIAXK3IQNIIYNRBWSRN
aws_secret_access_key = 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
```

1) Select **File**.

2) Select **Exit**.

b) **Open** the **Config** file in **Notepad**.

Note: You can see the **region** and **output** format details.


config - Notepad

```
[default]
region = us-east-1
output = json
```

1) Select **File**.

2) Select **Exit**.

c) **Close** the **File explorer**.

Ahmad Mohammad

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Task 3: Create Custom Image

In this task, you will create the Image of the Python server to use in the upcoming labs.

Step 1: Create Image

18. In the **AWS Management Console**, on the **Services** menu, search and Select **EC2**.

19. Choose the **YOUR ALLOCATED REGION**, region list to the right of your account information on the navigation bar.

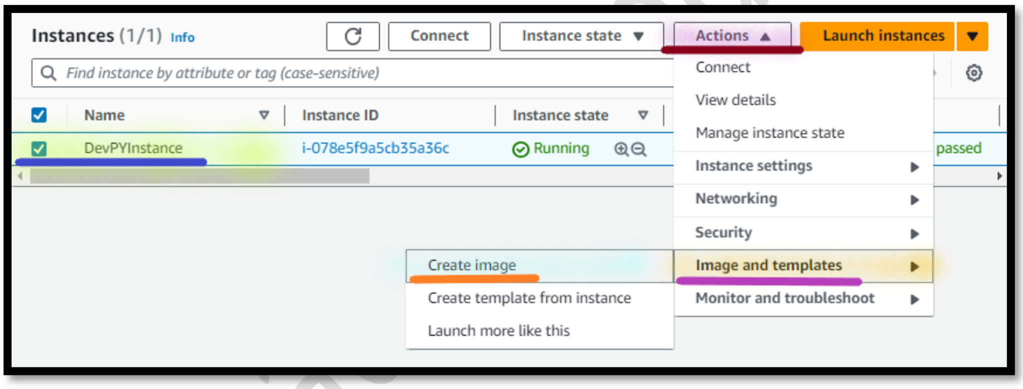
20. Select **Instances**.

    a. Select **DevPYInstance**.

        i. Select **Actions**.

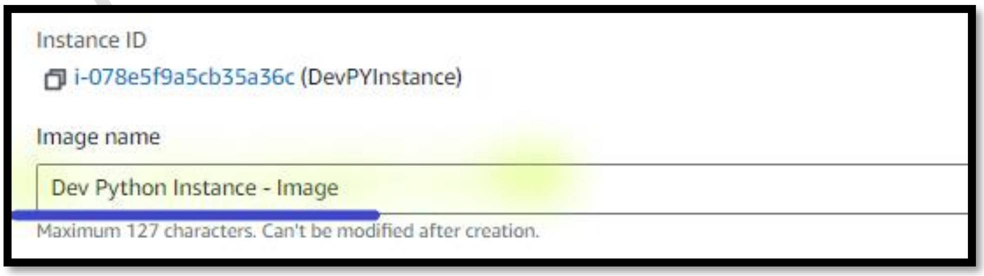
            a) Select **Image and templates**.

                I. Select **Create image**.



    b. From the **Create image** page:

        i. **Image name**: Write **Dev Python Instance - Image**.



Instance ID

i-078e5f9a5cb35a36c (DevPYInstance)

Image name

Dev Python Instance - Image

Maximum 127 characters. Can't be modified after creation.

Ahmad Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Note: Leave other details as default.

I. Select **Create image**.

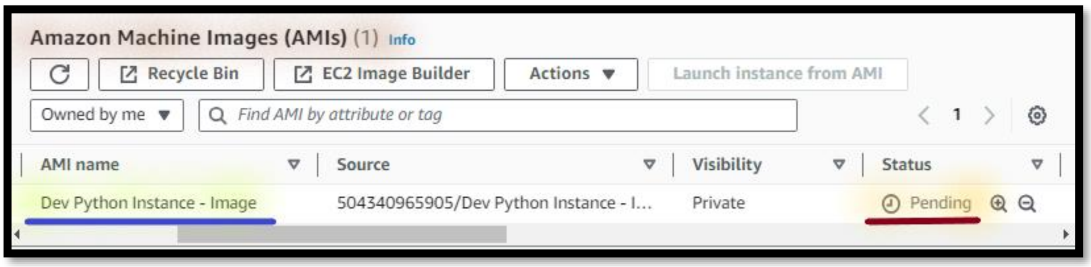
Note: DevPYInstance gets **Rebooted**.

Step 2: View the Custom Image

21. From the **EC2** console.

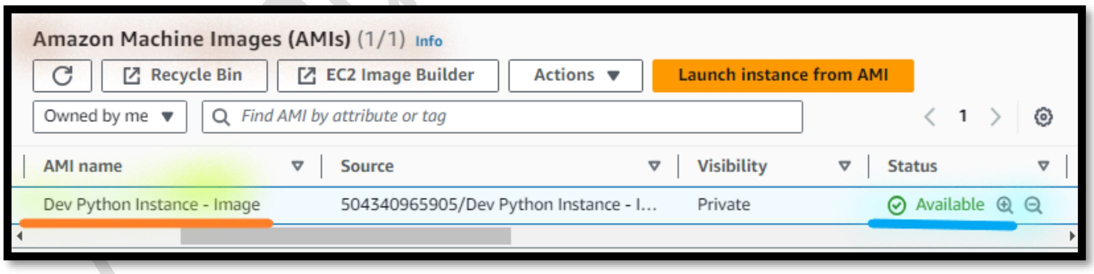
22. Select **AMIs**.

Note: You can see the **Dev Python Instance - Image** status as **Pending**.



AMI name	Source	Visibility	Status
Dev Python Instance - Image	504340965905/Dev Python Instance - I...	Private	Pending

Note: **Wait**, till **Dev Python Instance - Image** status as **Available**.



AMI name	Source	Visibility	Status
Dev Python Instance - Image	504340965905/Dev Python Instance - I...	Private	Available

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Task 4: Create Python Project for DynamoDB

In this task, you will create Python project to manage DynamoDB programmatically.

Step 1: Develop Python Code

23. **Unzip** the **LAB-07-02-Python-Code-A.zip**.

Note: **Lab-07-02-Python-Code-A**.zip code file is available with the **Lab manual**.

Note: **Review** the **code** after opening in the **Notepad**.

Step 2: Connect to Instance

24. From the **Local Desktop/ Laptop**, right click on **Start** & **Run**.

    c. In the **Open**, write **mstsc**.

    d. Select **Ok**.

        ii. **From** the **Remote Desktop Connection**:

            c) **Computer**: Write the **Public IP Address** of the **DevPYInstance**.

            d) Select **Connect**.

Note: You can **get the prompt** to enter the **Username** and **Password**.

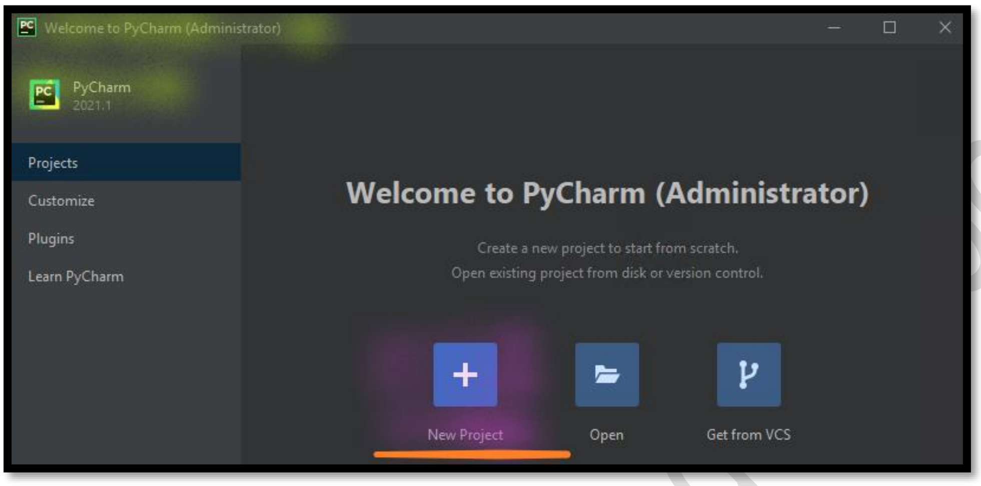
4) **Username:** Write **Administrator**.
5) **Password**: Write **lab-password@123**.
6) Select **Ok**.

Ahmad Majeed Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Step 3: Launch the PyCharm IDE

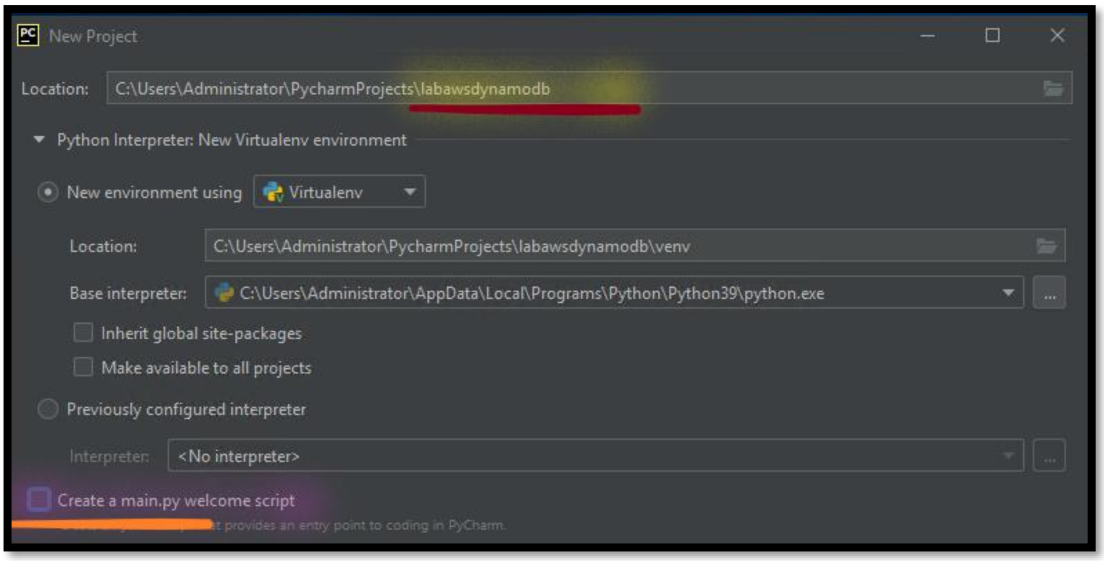
25. From the **DevPYInstance**.
    a. Open the **PyCharm**.
26. From the **PyCharm**:
    a. Select the **New Project**.



Welcome to PyCharm (Administrator)

Create a new project to start from scratch.
Open existing project from disk or version control.

    b. From the **New Project** section:
        i. **Location**: **Replace** the ***existing name*** (**pythonProject**) and write **labawsdynamodb**.
        ii. **Uncheck** the ***Create a main.py welcome script***.



Location: C:\Users\Administrator\PycharmProjects\labawsdynamodb

Python Interpreter: New Virtualenv environment

New environment using Virtualenv

Location: C:\Users\Administrator\PycharmProjects\labawsdynamodb\venv

Base interpreter: C:\Users\Administrator\AppData\Local\Programs\Python\Python39\python.exe

Inherit global site-packages

Make available to all projects

Previously configured interpreter

Interpreter: <No interpreter>

Create a main.py welcome script

...t provides an entry point to coding in PyCharm.

        iii. Select **Create**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

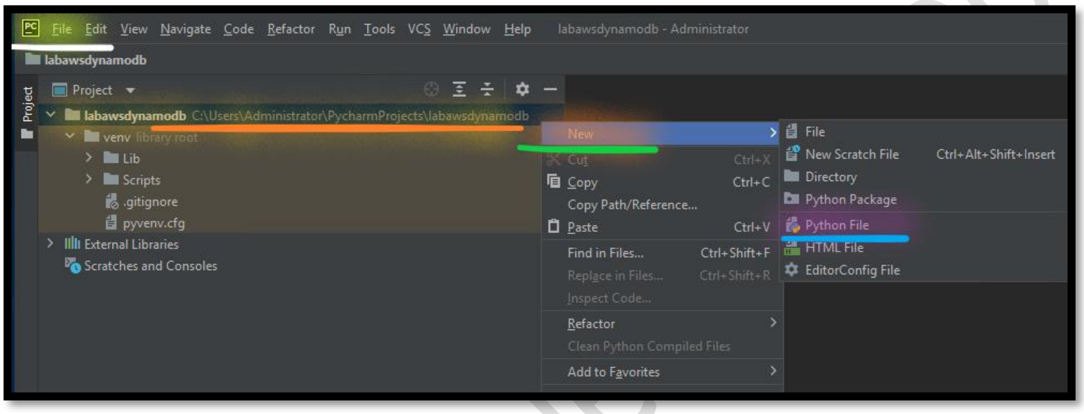
Note: **Wait**, till **virtual environment** gets **created**.

Step 4: Create the Files in the Python Project

27. **Expand** the ***labawsdynamodb*** Python project**.**

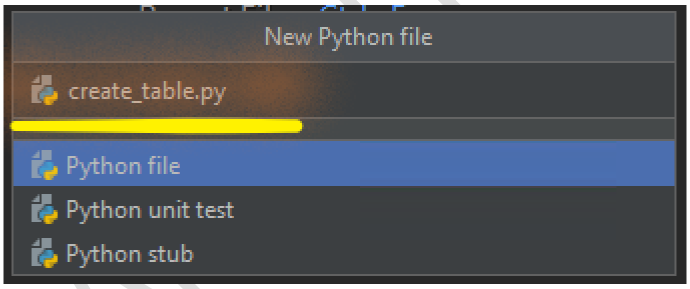
Create create_table.py File

a. **Right-click** on the **labawsdynamodb** Python project.
   i. Select **New**.
      a) Select **Python File**.



File Edit View Navigate Code Refactor Run Tools VCS Window Help labawsdynamodb - Administrator
labawsdynamodb
Project
labawsdynamodb C:\Users\Administrator\PycharmProjects\labawsdynamodb
venv library root
Lib
Scripts
.gitignore
pyvenv.cfg
External Libraries
Scratches and Consoles
New
Cut Ctrl+X
Copy Ctrl+C
Copy Path/Reference...
Paste Ctrl+V
Find in Files... Ctrl+Shift+F
Replace in Files... Ctrl+Shift+R
Inspect Code...
Refactor
Clean Python Compiled Files
Add to Favorites
File
New Scratch File Ctrl+Alt+Shift+Insert
Directory
Python Package
Python File
HTML File
EditorConfig File

b. In the **New python file** page:
   i. **File name**: Write **create_table.py**.



   ii. Select **Enter**.

Create create_items.py File

28. **Right-click** on the ***labawsdynamodb*** Python package.

a. Select **New**.
   i. Select **Python File**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

ii. In the **New python file** page:

a) **File name**: Write **create_items.py**.

b) Select **Enter**.

Create read_items.py File

29. **Right-click** on the ***labawsdynamodb*** Python package.

a. Select **New**.

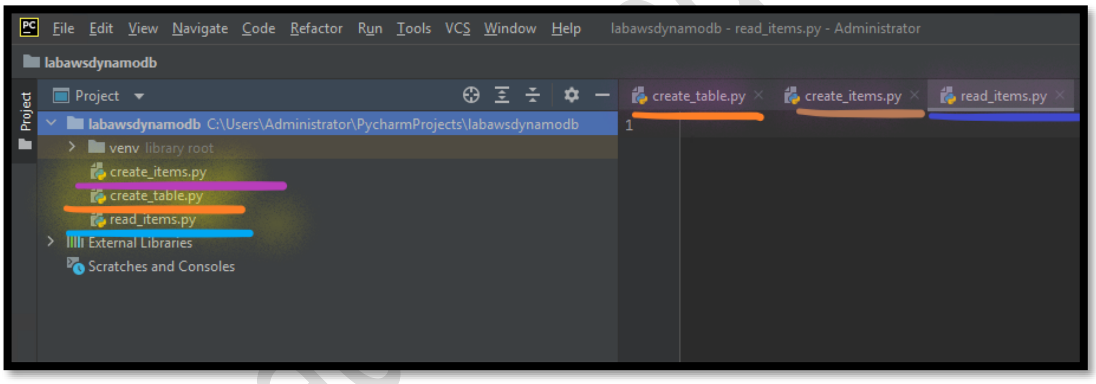
i. Select **Python File**.

ii. In the **New python file** page:

a) **File name**: Write **read_items.py**.

b) Select **Enter**.

Note: You can see the **create_table.py**, **create_items.py** and **read_items.py** under Python package.


File Edit View Navigate Code Refactor Run Tools VCS Window Help labawsdynamodb - read_items.py - Administrator
labawsdynamodb
Project
labawsdynamodb C:\Users\Administrator\PycharmProjects\labawsdynamodb
venv library root
create_items.py
create_table.py
read_items.py
External Libraries
Scratches and Consoles
create_table.py
create_items.py
read_items.py

Step 5: Update the Python Code

30. **Double-click** on the ***create_table.py*** Python file.

a. **Paste** the **Code** from ***create_table.py*** Python file.

b. **From** the **PyCharm IDE**.

i. Press **CTRL + S** (*to save*).

31. **Double-click** on the ***create_items.py*** Python file.

a. **Paste** the **Code** from ***create_items.py*** file.

b. **From** the **PyCharm IDE**.

i. Press **CTRL + S** (*to save*).

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

32. **Double-click** on the ***read_items.py*** Python file.

   a. **Paste** the **Code** from ***read_items.py*** file.

   b. **From** the **PyCharm IDE**.

      i. Press **CTRL + S** (*to save*).

Task 5: Manage AWS DynamoDB from PyCharm

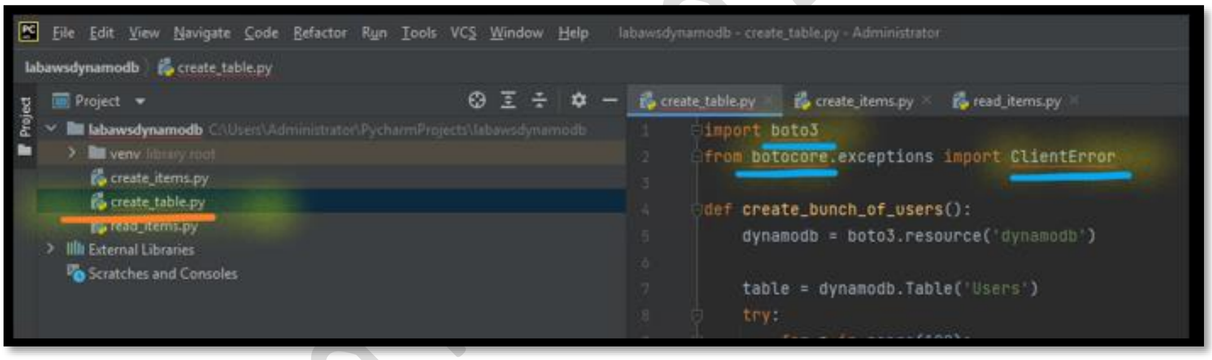
In this task, you will manage AWS DynamoDB from PyCharm using Python.

Step 1: Install Python SDK

33. **Expand** the ***labawsdynamodb*** Python project.

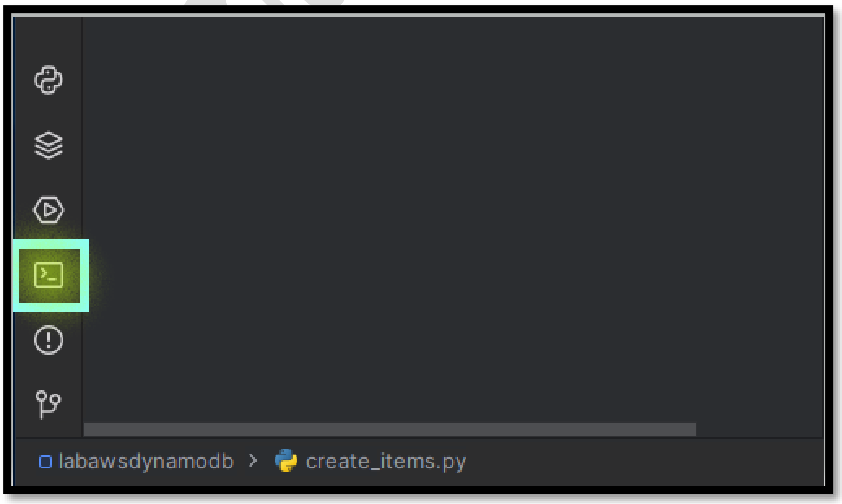
34. **Double-click** on the ***create_table.py*** Python file.

   a. It will show the ***error*** against **boto3**, **botocore** and **ClientError** references in the code lines with **Red colour underline**.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
labawsdynamodb - create_table.py - Administrator
labawsdynamodb
create_table.py
Project
labawsdynamodb C:\Users\Administrator\PycharmProjects\labawsdynamodb
venv library root
create_items.py
create_table.py
read_items.py
External Libraries
Scratches and Consoles
create_table.py
create_items.py
read_items.py

```
import boto3
from botocore.exceptions import ClientError

def create_bunch_of_users():
    dynamodb = boto3.resource('dynamodb')

    table = dynamodb.Table('Users')
    try:
```

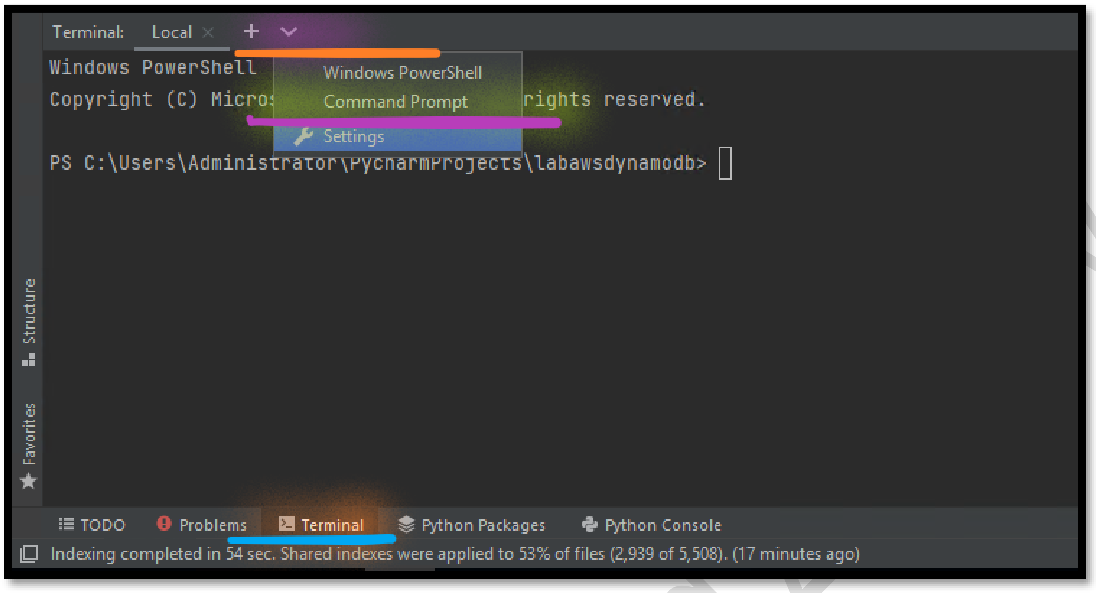
35. ***Go below*** in the console, click on the **Terminal**.



© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

a. From the **Terminal**:

i. Dropdown (*see the screenshot*) and select **Command prompt**.



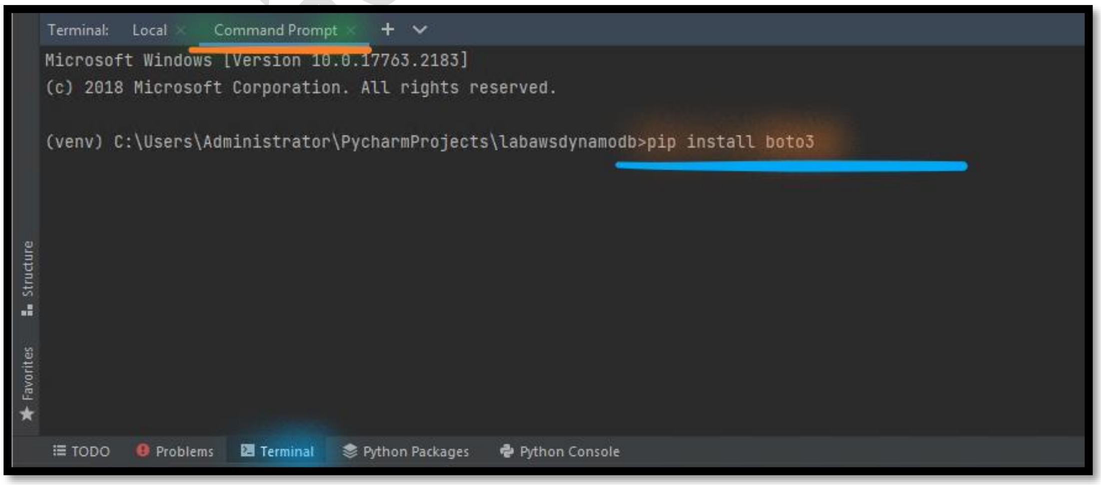
```
Windows PowerShell
Copyright (C) Micros                              rights reserved.

PS C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

Indexing completed in 54 sec. Shared indexes were applied to 53% of files (2,939 of 5,508). (17 minutes ago)

b. From the **Terminal**:

i. **Type pip install boto3**.

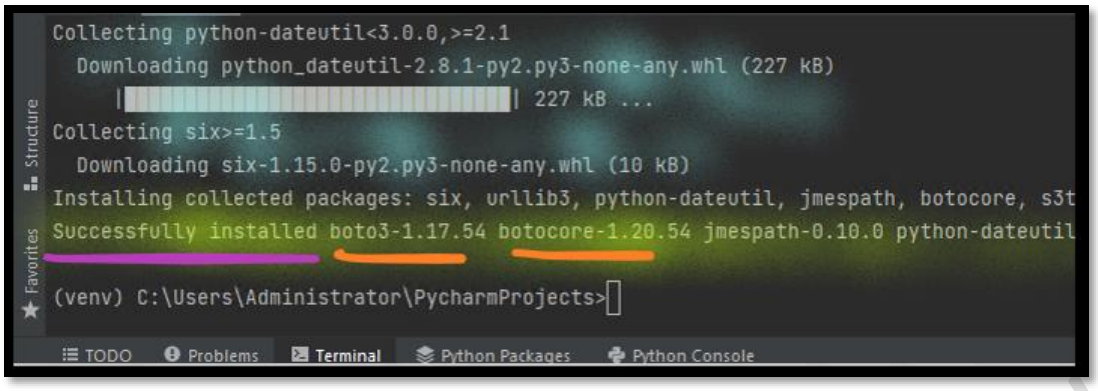
Note: **Boto3** is the Amazon Web Services (AWS) Software Development Kit (SDK) for Python.



```
Microsoft Windows [Version 10.0.17763.2183]
(c) 2018 Microsoft Corporation. All rights reserved.

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>pip install boto3
```

Note: **Wait**, till **boto3 sdk** install **succesfully**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.



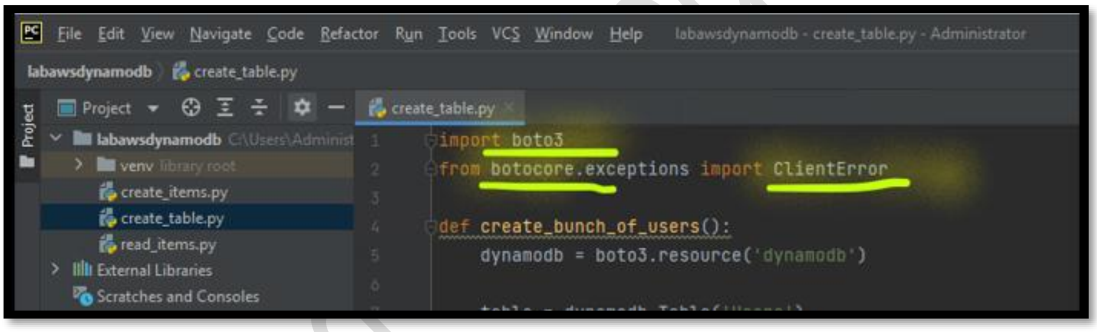
```
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.8.1-py2.py3-none-any.whl (227 kB)
     |████████████████████████████████| 227 kB ...
Collecting six>=1.5
  Downloading six-1.15.0-py2.py3-none-any.whl (10 kB)
Installing collected packages: six, urllib3, python-dateutil, jmespath, botocore, s3t
Successfully installed boto3-1.17.54 botocore-1.20.54 jmespath-0.10.0 python-dateutil

(venv) C:\Users\Administrator\PycharmProjects>
```

Note: **Wait**, till **indexing** gets **Completed**.

36.**Expand** the ***labawsdynamodb*** Python project.

37.**Double-click** on the ***create_table.py*** Python file.

a. **Click** on the **boto3**, **botocore** and **ClientError** references in the code lines. ***Red colour underline*** is ***resolved now***.



```
import boto3
from botocore.exceptions import ClientError

def create_bunch_of_users():
    dynamodb = boto3.resource('dynamodb')
```

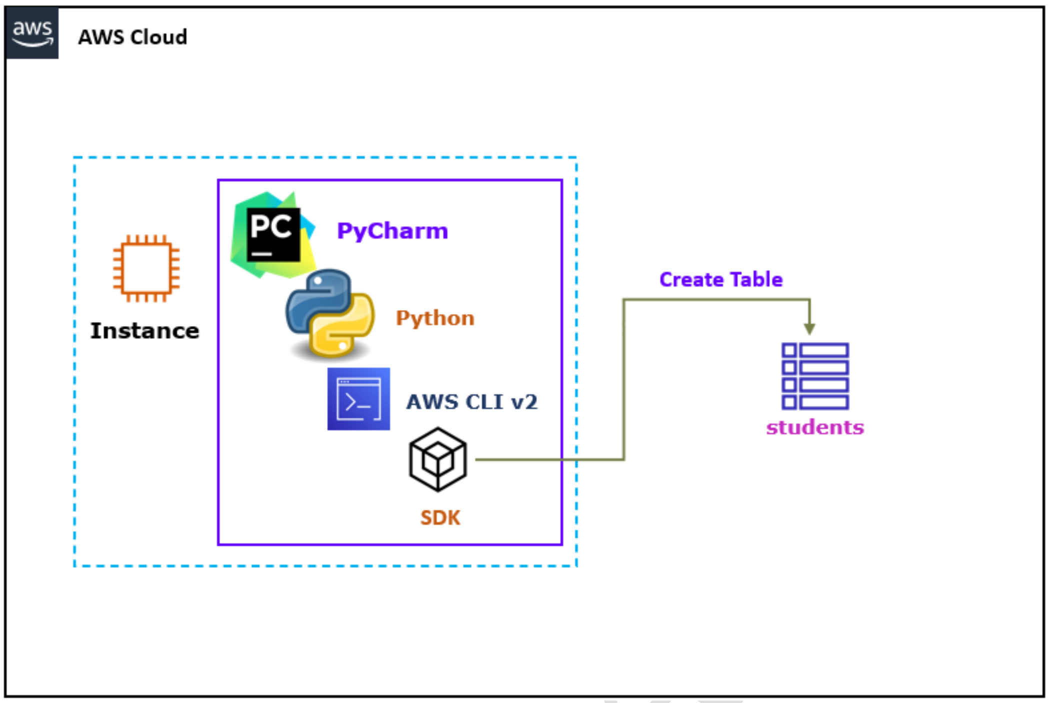
File Edit View Navigate Code Refactor Run Tools VCS Window Help
labawsdynamodb - create_table.py - Administrator
labawsdynamodb
create_table.py
Project
labawsdynamodb C:\Users\Administ
venv library root
create_items.py
create_table.py
read_items.py
External Libraries
Scratches and Consoles

Ahmad Zahoor

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Step 2: Create DynamoDB Table

In this task, you will create DynamoDB table from PyCharm using Python.



38. **Expand** the ***labawsdynamodb*** Python Project.

39. **Double-click** on the ***create_table.py*** Python file.

Info: **Examine the code:**

1. **Creating** the **DynamoDB table**:
    a. **Table name**: '**students**'.
    b. **Partition key** (or Hash key): '**id**' as number.
    c. **Sort key** (or Range key): '**username**' as string.

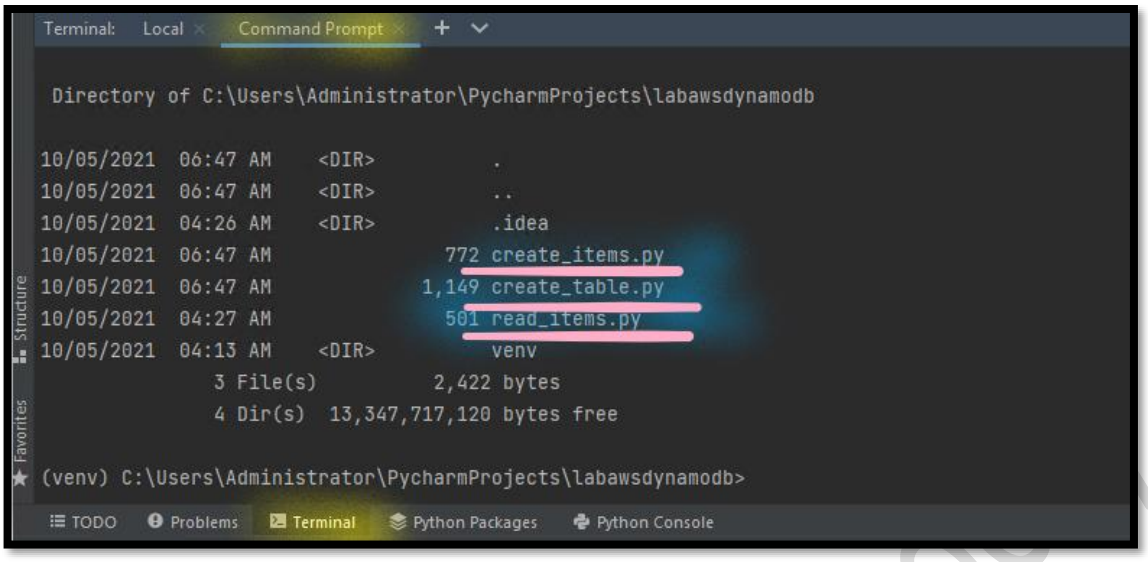
a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **Dir**.

Note: You can see the **create_table.py**, **create_items.py** and **read_items.py**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

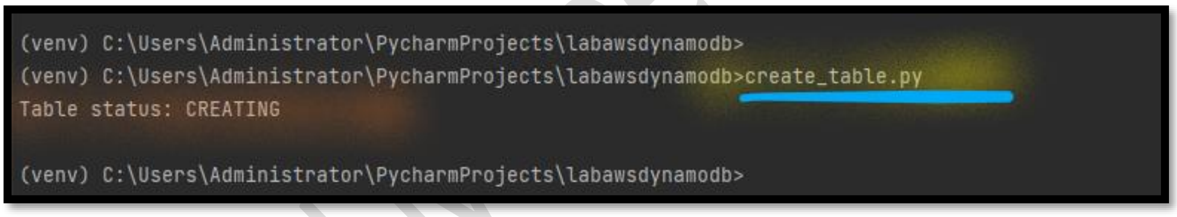

Terminal: Local Command Prompt
TODO Problems Terminal Python Packages Python Console
Structure Favorites

```
 Directory of C:\Users\Administrator\PycharmProjects\labawsdynamodb

10/05/2021  06:47 AM    <DIR>          .
10/05/2021  06:47 AM    <DIR>          ..
10/05/2021  04:26 AM    <DIR>          .idea
10/05/2021  06:47 AM               772 create_items.py
10/05/2021  06:47 AM             1,149 create_table.py
10/05/2021  04:27 AM               501 read_items.py
10/05/2021  04:13 AM    <DIR>          venv
               3 File(s)          2,422 bytes
               4 Dir(s)  13,347,717,120 bytes free

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

ii. From the **Terminal** (***command prompt***):

a) **Type** **python create_table.py**.

Note: If table created succesfully, you will see the "**Creating**" message.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>create_table.py
Table status: CREATING

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

Note: **Go to next task**, but **Don't close** the **DevPYInstance** console.

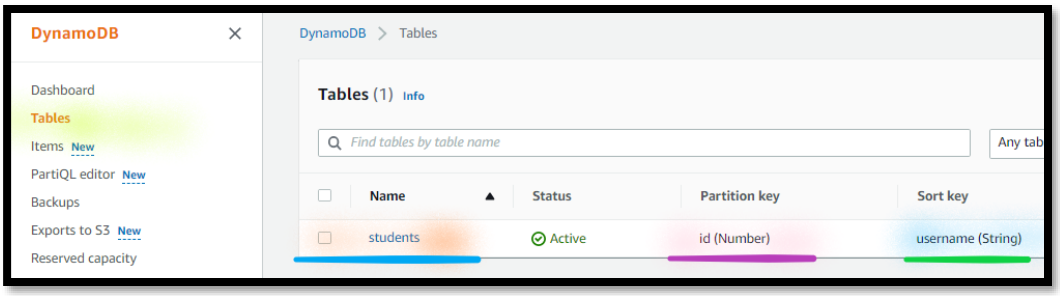
Verify DynamoDB Table from AWS Console

40. In the **AWS Management Console**, on the **Services** menu, click **DynamoDB**.

41. Select **Tables**.

Note: You can see the "**students**" table and its **schema** details.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.


DynamoDB

- Dashboard
- Tables
- Items New
- PartiQL editor New
- Backups
- Exports to S3 New
- Reserved capacity

DynamoDB > Tables

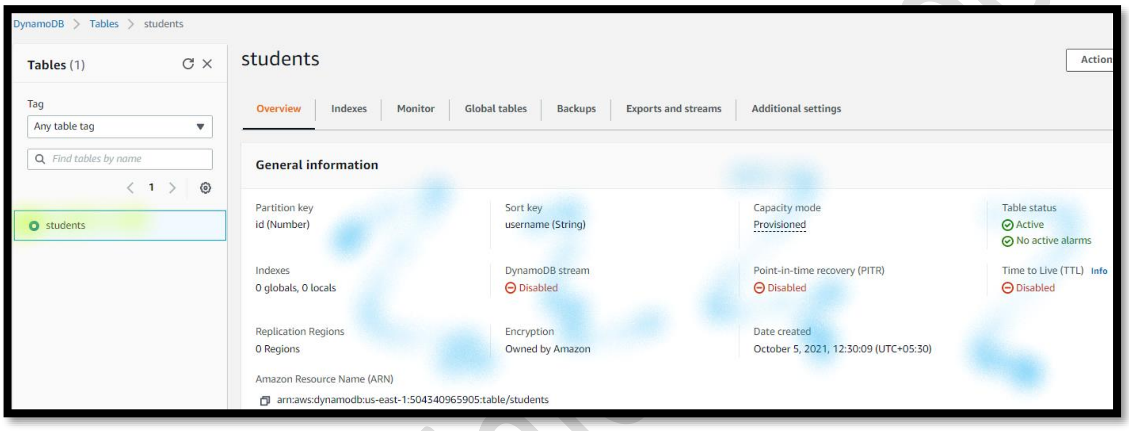
Tables (1) Info

Find tables by table name | Any tab

Name	Status	Partition key	Sort key
students	Active	id (Number)	username (String)

a. Open **students** table.

Note: You can view the **details** of the students table.


DynamoDB > Tables > students

Tables (1)

Tag: Any table tag

Find tables by name

students

students

Overview | Indexes | Monitor | Global tables | Backups | Exports and streams | Additional settings

General information

Partition key	Sort key	Capacity mode	Table status
id (Number)	username (String)	Provisioned	Active No active alarms
Indexes	**DynamoDB stream**	**Point-in-time recovery (PITR)**	**Time to Live (TTL)**
0 globals, 0 locals	Disabled	Disabled	Disabled
Replication Regions	**Encryption**	**Date created**	
0 Regions	Owned by Amazon	October 5, 2021, 12:30:09 (UTC+05:30)	

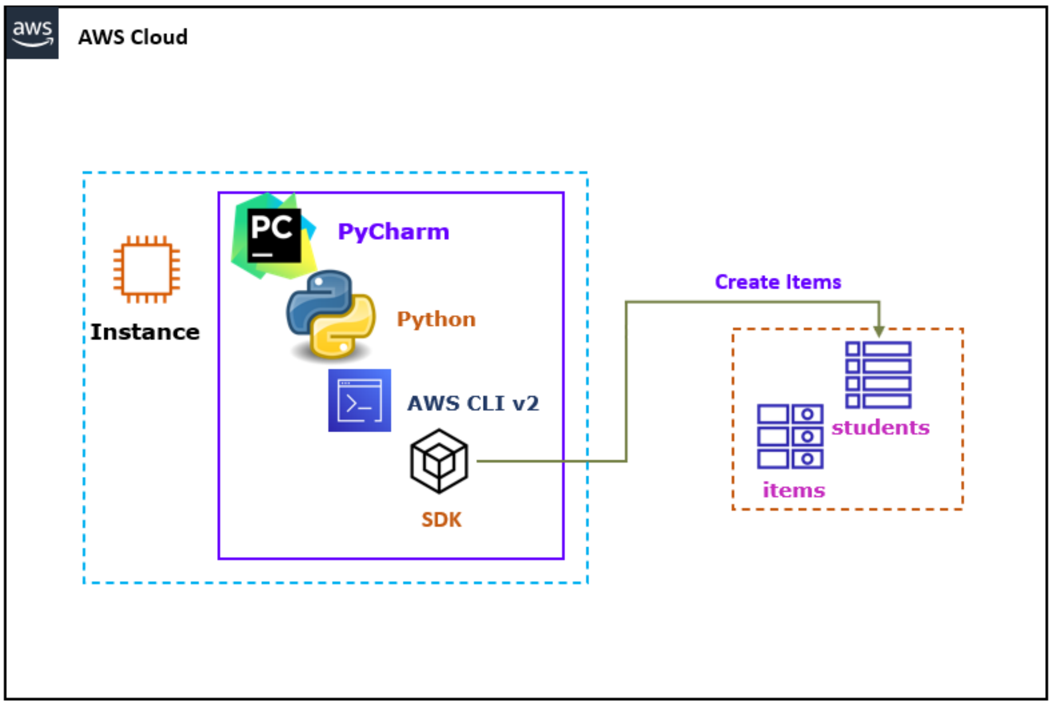
Amazon Resource Name (ARN)
arn:aws:dynamodb:us-east-1:504340965905:table/students

Ahmad Maje

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Step 3: Create Items in DynamoDB Table

In this task, you will create new Items using loop in DynamoDB table from PyCharm using Python.



42. **Return** to the **DevPYInstance**.

43. **Expand** the ***labawsdynamodb*** Python Project.

44. **Double-click** on the ***create_items.py*** Python file.

Info: **Examine the code:**

1. **Creating** the **Items** in **DynamoDB Table** name: '**students**'.
   a. **Adding** the **items** using loop.
   b. **Adding** the **username** from **Ahmad0** to **Ahmad9** using loop.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python create_items.py**.

Note: If items added succesfully, you will see the "**Created succesfully**" message.

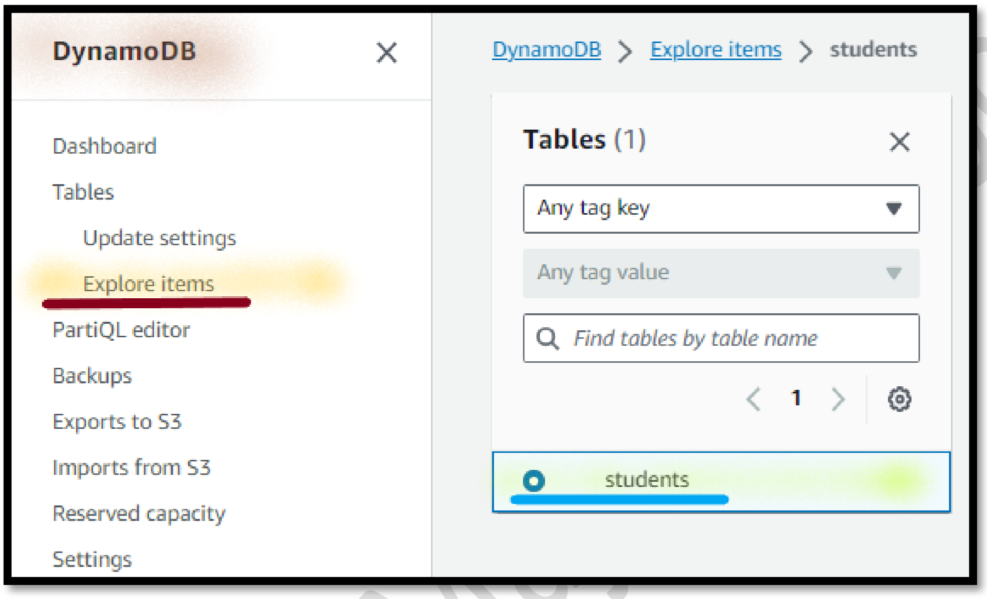
© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Note: **Go to next task**, but **Don't close** the **DevPYInstance** console.

Verify Added Items in DynamoDB Table from AWS Console

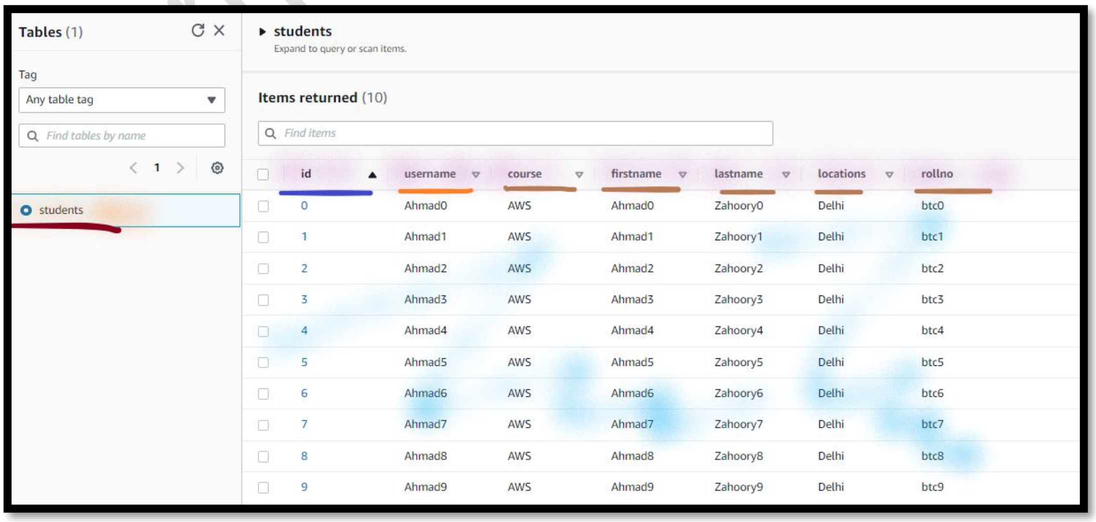
45. In the **AWS Management Console**, on the **Services** menu, click **DynamoDB**.

46. Select **Explore Items**.



a. Select **students** table.

Note: You can see the added **items** in **students** table.



Items returned (10)

id	username	course	firstname	lastname	locations	rollno
0	Ahmad0	AWS	Ahmad0	Zahoory0	Delhi	btc0
1	Ahmad1	AWS	Ahmad1	Zahoory1	Delhi	btc1
2	Ahmad2	AWS	Ahmad2	Zahoory2	Delhi	btc2
3	Ahmad3	AWS	Ahmad3	Zahoory3	Delhi	btc3
4	Ahmad4	AWS	Ahmad4	Zahoory4	Delhi	btc4
5	Ahmad5	AWS	Ahmad5	Zahoory5	Delhi	btc5
6	Ahmad6	AWS	Ahmad6	Zahoory6	Delhi	btc6
7	Ahmad7	AWS	Ahmad7	Zahoory7	Delhi	btc7
8	Ahmad8	AWS	Ahmad8	Zahoory8	Delhi	btc8
9	Ahmad9	AWS	Ahmad9	Zahoory9	Delhi	btc9

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Add Items (additional) in DynamoDB Table

47. **Return** to the **DevPYInstance**.

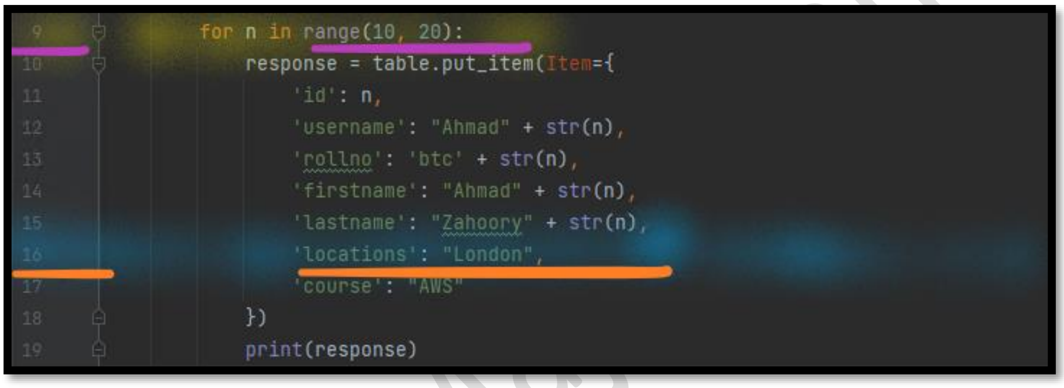
48. **Expand** the ***labawsdynamodb*** Python project.

49. **Double-click** on the ***create_items.py*** Python file.

    a. **Replace** the **Range function** in **Row 9** from **10** to **10, 20**.

Info: Adding the **username** (*sort key*) from **Ahmad10** to **Ahmad19** using loop.

    b. **Replace** the **Locations value** in **Row 16** to **London**.



```
    for n in range(10, 20):
        response = table.put_item(Item={
            'id': n,
            'username': "Ahmad" + str(n),
            'rollno': 'btc' + str(n),
            'firstname': "Ahmad" + str(n),
            'lastname': "Zahoory" + str(n),
            'locations': "London",
            'course': "AWS"
        })
        print(response)
```

    c. **From** the **PyCharm IDE**.

        i. Select **CTRL + S**.

    d. ***Go below*** in the console, click on the **Terminal**.

        i. From the **Terminal** (***command prompt***):

            a) **Type** **python create_items.py**.

Note: If items added succesfully,you will see the response "**Created succesfully**" message.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Ahmad Majeed Zahoory

Add Items (additional) in DynamoDB Table

50. **From** the ***create_items.py*** Python file.

    a. **Replace** the **Range function** in **Row 9** from **10, 20** to **20, 30**.

Info: Adding the **username** (*sort key*) from **Ahmad20** to **Ahmad29** using loop.

    b. **Replace** the **Locations value** in **Row 16** to **Paris**.

    c. **From** the **PyCharm IDE**.

        i. Select **CTRL + S**.

    d. ***Go below*** in the console, click on the **Terminal**.

        i. From the **Terminal** (***command prompt***):

            a) **Type** **python create_items.py**.

Note: If items added succesfully,you will see the response "**Created succesfully**" message.

Add Items (additional) in DynamoDB Table

51. **From** the ***create_items.py*** Python file.

    a. **Replace** the **Range function** in **Row 9** from **20, 30** to **30, 40**.

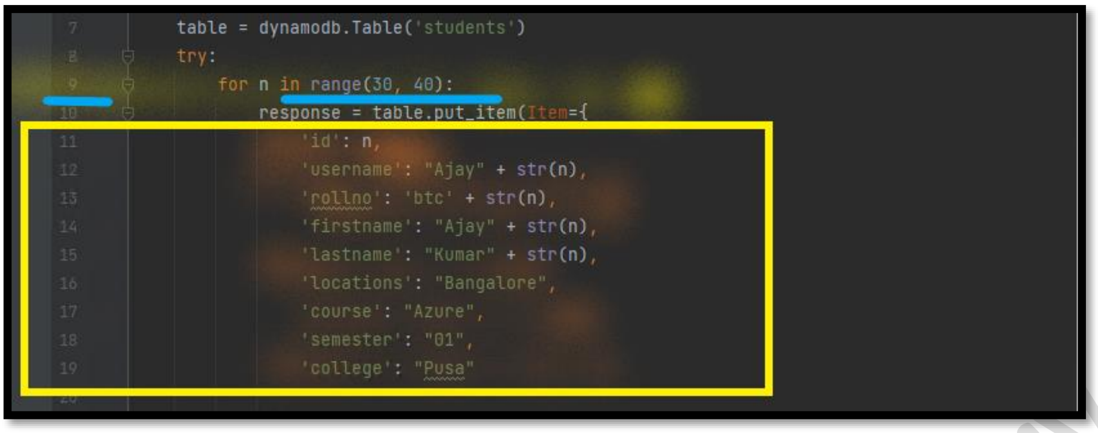
Info: Adding the **username** (*sort key*) from **Ahmad30** to **Ahmad39** using loop.

    b. **Replace** the following **Values** for Items **between** **Row 11** to **Row 19**:

```
'id': n,
'username': "Ajay" + str(n),
'rollno': 'btc' + str(n),
'firstname': "Ajay" + str(n),
'lastname': "Kumar" + str(n),
'locations': "Bangalore",
'course': "Azure",
'semester': "01",
'college': "Pusa"
```

Ahmad Malek Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.



```
    table = dynamodb.Table('students')
    try:
        for n in range(30, 40):
            response = table.put_item(Item={
                'id': n,
                'username': "Ajay" + str(n),
                'rollno': 'btc' + str(n),
                'firstname': "Ajay" + str(n),
                'lastname': "Kumar" + str(n),
                'locations': "Bangalore",
                'course': "Azure",
                'semester': "01",
                'college': "Pusa"
```

c. **From** the **PyCharm IDE**.

i. Select **CTRL + S**.

d. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python create_items.py**.

Note: If items added succesfully,you will see the response "**Created succesfully**" message.

Add Items (additional) in DynamoDB Table

52. **From** the ***create_items.py*** Python file.

a. **Replace** the **Range function** in **Row 9** from **30, 40** to **40, 50**.

Info: Adding the **username** (*sort key*) from **Ahmad40** to **Ahmad49** using loop.

b. **Replace** the following **Values** for Items **between** **Row 11** to **Row 19**:

```
'id': n,
'username': "Eric" + str(n),
'rollno': 'btc' + str(n),
'firstname': "Eric" + str(n),
'lastname': "Layman" + str(n),
'locations': "Chicago",
'course': "GCP",
'level': "intermediate",
'fees': "paid"
```

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

c. **From** the **PyCharm IDE**.

i. Select **CTRL + S**.

d. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** *(**command prompt**)*:

a) **Type** **python create_items.py**.

Note: If items added succesfully,you will see the response "**Created succesfully**" message.

Note: **Go to next task**, but **Don't close** the **DevPYInstance** console.

Verify Added Items in DynamoDB Table from AWS Console

53. In the **AWS Management Console**, on the **Services** menu, click **DynamoDB**.

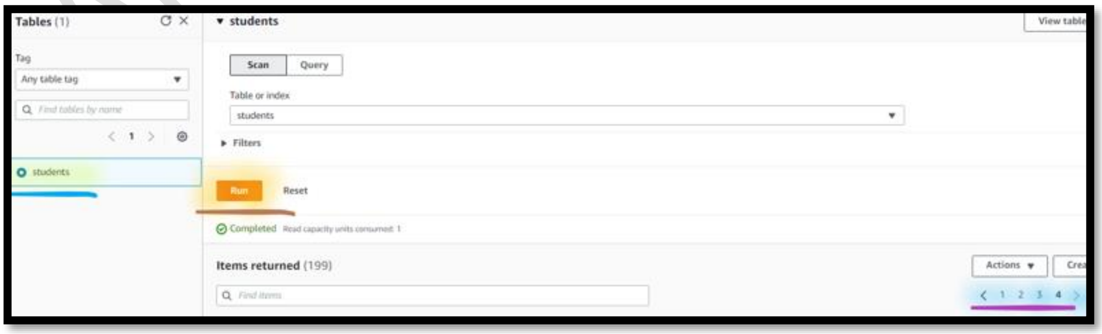
54. Select **Explore Items**.

a. Select the **students** table.

Note: You can see the added **items** in **students** table.

Note: If you are **not able to view** the **newly added items**, Select the **Run**.

Note: If you are **not able to view** the **newly added items**, Select the **Page Number**.

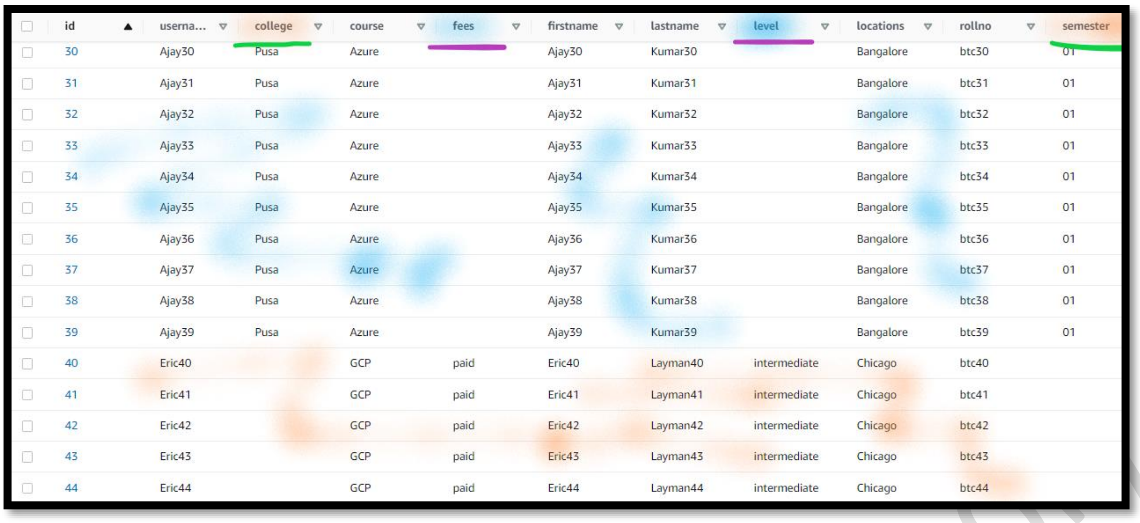


Completed Read capacity units consumed: 1

Items returned (199)

Find items

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.



id	userna...	college	course	fees	firstname	lastname	level	locations	rollno	semester
30	Ajay30	Pusa	Azure		Ajay30	Kumar30		Bangalore	btc30	01
31	Ajay31	Pusa	Azure		Ajay31	Kumar31		Bangalore	btc31	01
32	Ajay32	Pusa	Azure		Ajay32	Kumar32		Bangalore	btc32	01
33	Ajay33	Pusa	Azure		Ajay33	Kumar33		Bangalore	btc33	01
34	Ajay34	Pusa	Azure		Ajay34	Kumar34		Bangalore	btc34	01
35	Ajay35	Pusa	Azure		Ajay35	Kumar35		Bangalore	btc35	01
36	Ajay36	Pusa	Azure		Ajay36	Kumar36		Bangalore	btc36	01
37	Ajay37	Pusa	Azure		Ajay37	Kumar37		Bangalore	btc37	01
38	Ajay38	Pusa	Azure		Ajay38	Kumar38		Bangalore	btc38	01
39	Ajay39	Pusa	Azure		Ajay39	Kumar39		Bangalore	btc39	01
40	Eric40		GCP	paid	Eric40	Layman40	intermediate	Chicago	btc40	
41	Eric41		GCP	paid	Eric41	Layman41	intermediate	Chicago	btc41	
42	Eric42		GCP	paid	Eric42	Layman42	intermediate	Chicago	btc42	
43	Eric43		GCP	paid	Eric43	Layman43	intermediate	Chicago	btc43	
44	Eric44		GCP	paid	Eric44	Layman44	intermediate	Chicago	btc44	

Step 4: Read Items from DynamoDB Table

55. **Return** to the **DevPYInstance**.

56. **Expand** the ***abawsdynamodb*** Python project.

57. **Double-click** on the ***read_items.py*** Python file.

Info: **Examine the code:**

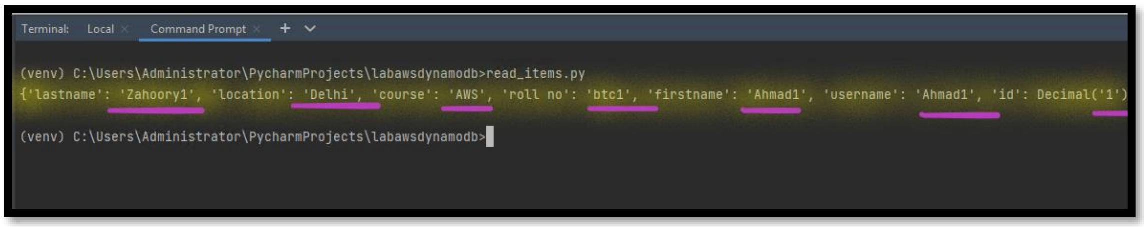
1. **Reading** the **items** from **DynamoDB Table** name: '**students**'.
   a. **Reading** the **items** with **Partition key** as '**1**' and **Sort key** as '**Ahmad1**'.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python read_items.py**.

Note: If able to read items succesfully, you will see the **items details**.



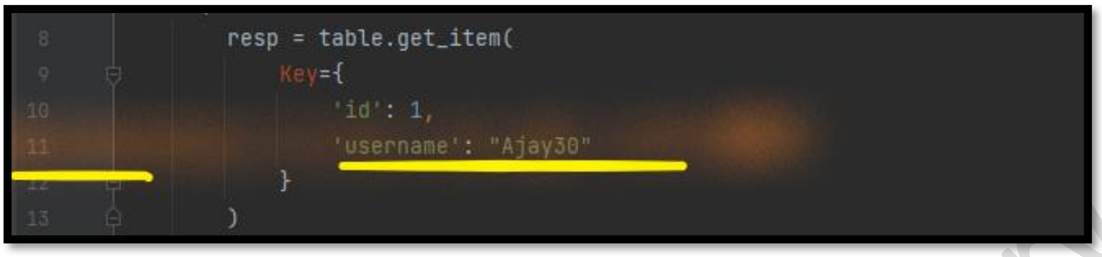
```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>read_items.py
{'lastname': 'Zahoory1', 'location': 'Delhi', 'course': 'AWS', 'roll no': 'btc1', 'firstname': 'Ahmad1', 'username': 'Ahmad1', 'id': Decimal('1')

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Read Other Items from DynamoDB Table

58. **Double-click** on the ***read_items.py*** Python file.

    a. **Replace** the **Username** in **Row 11** from **Ahmad1** to **Ajay30**.



```
resp = table.get_item(
    Key={
        'id': 1,
        'username': "Ajay30"
    }
)
```

    b. **From** the **PyCharm IDE**.

        i. Select **CTRL + S**.

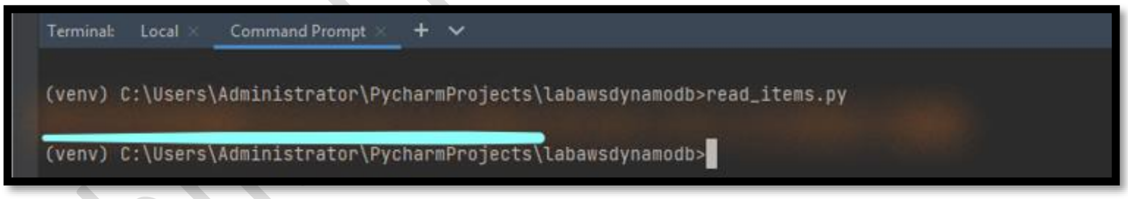
    c. ***Go below*** in the console, click on the **Terminal**.

        i. From the **Terminal** (***command prompt***):

            a) **Type** **python read_items.py**.

Note: If able to read items succesfully, you will see the **items details**.

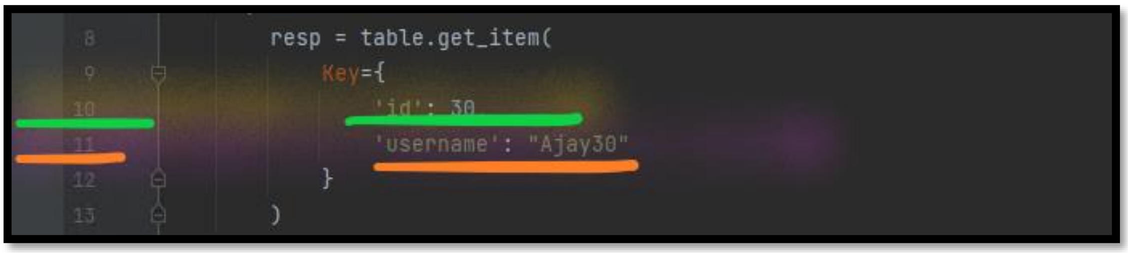
Note: You **can't see any response** in the **Terminal**. **Reason** you are **not passing** the **Correct Partition Key values** against Sort Key values.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>read_items.py

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

59. **Double-click** on the ***read_items.py*** Python file.

    a. **Replace** the **id** in **Row 10** from **1** to **30**.



```
resp = table.get_item(
    Key={
        'id': 30
        'username': "Ajay30"
    }
)
```

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

b. **From** the **PyCharm IDE**.

i. Select **CTRL + S**.

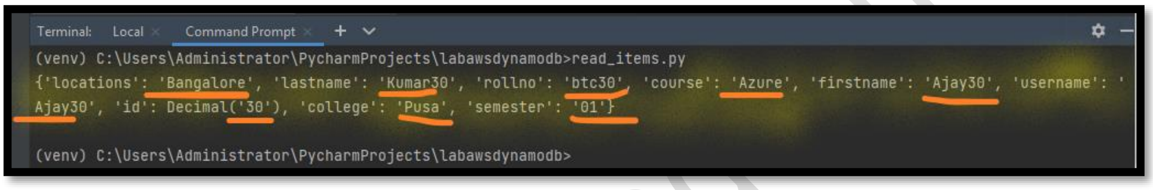
c. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python read_items.py**.

Note: If able to read items succesfully, you will see the **items details**.

Note: You **can see the response** in the **Terminal**. Reason you are passing the **Correct Partition Key value** and Sort Key values.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>read_items.py
{'locations': 'Bangalore', 'lastname': 'Kumar30', 'rollno': 'btc30', 'course': 'Azure', 'firstname': 'Ajay30', 'username': '
Ajay30', 'id': Decimal('30'), 'college': 'Pusa', 'semester': '01'}

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

Task 6: Manage AWS DynamoDB from PyCharm [Additional]

In this task, you will manage additional operations for AWS DynamoDB from PyCharm using Python.

Step 1: Develop Python code for Other Operations

60. **Unzip** the **LAB-07-02-Python-Code-B.zip**.

Note: **Lab-07-02-Python-Code-B.zip** code file is available with **the Lab manual**.

Note: **Review** the **Code** after opening in the **Notepad**.

61. **From** the **DevPYInstance**, right click on **Start** & **Run**.

a. In the **Open**, write **C:\Users\Administrator**, press **Ok**.

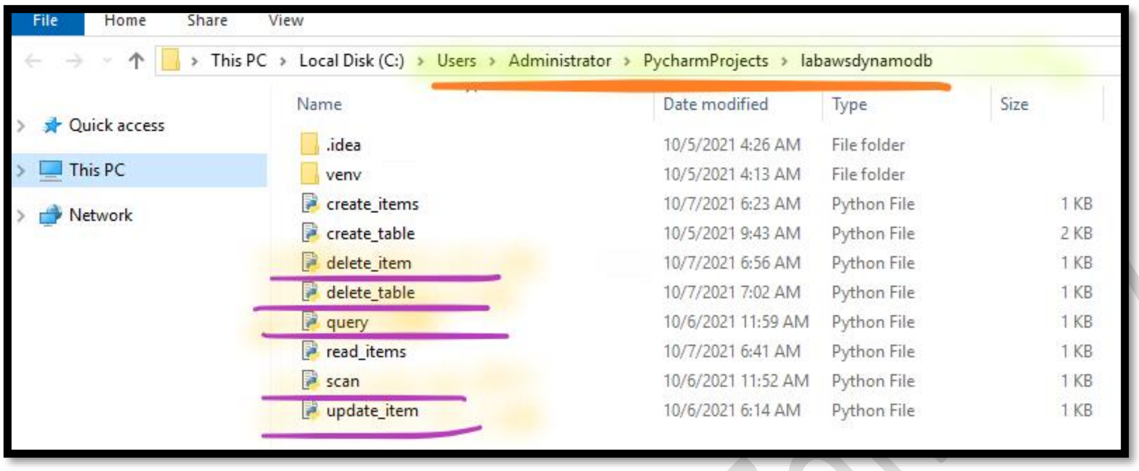
i. Open the **PycharmProjects** folder.

a) Open the **labawsdynamodb** folder.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Ahmad Majeed Khan

I. **Copy** the **update_item.py**, **scan.py**, **query.py**, **delete_item.py** and **delete_table.py** ***python files*** in the **labawsdynamodb** folder.

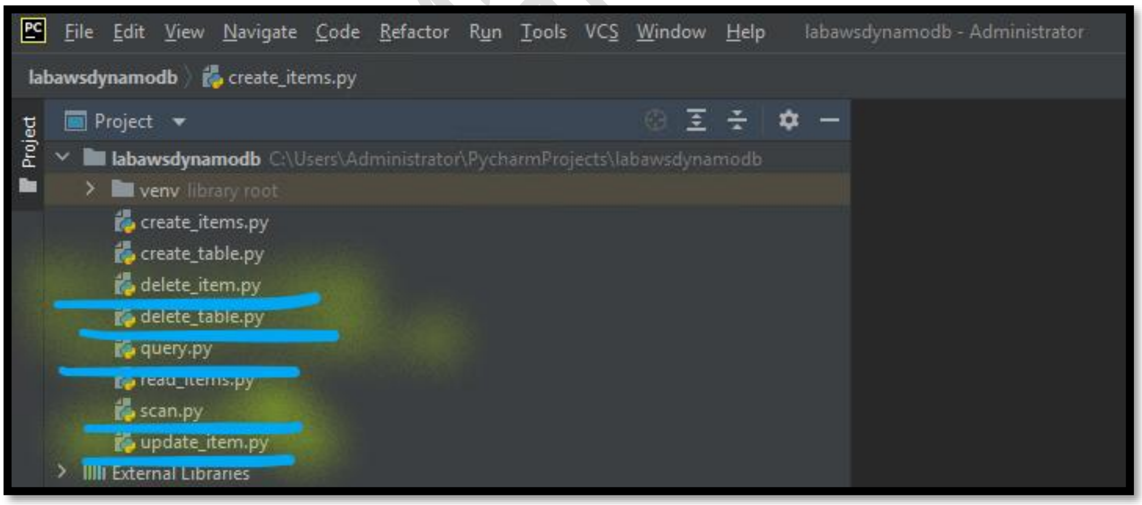

File
Home
Share
View
This PC > Local Disk (C:) > Users > Administrator > PycharmProjects > labawsdynamodb
Quick access
This PC
Network

Name	Date modified	Type	Size
.idea	10/5/2021 4:26 AM	File folder	
venv	10/5/2021 4:13 AM	File folder	
create_items	10/7/2021 6:23 AM	Python File	1 KB
create_table	10/5/2021 9:43 AM	Python File	2 KB
delete_item	10/7/2021 6:56 AM	Python File	1 KB
delete_table	10/7/2021 7:02 AM	Python File	1 KB
query	10/6/2021 11:59 AM	Python File	1 KB
read_items	10/7/2021 6:41 AM	Python File	1 KB
scan	10/6/2021 11:52 AM	Python File	1 KB
update_item	10/6/2021 6:14 AM	Python File	1 KB

62. **Return** to the **PyCharm IDE**.

a. **Expand** the ***labawsdynamodb*** Python project.

Note: You can see the **update_item.py**, **scan.py**, **query.py**, **delete_item.py** and **delete_table.py** Python files.


File Edit View Navigate Code Refactor Run Tools VCS Window Help labawsdynamodb - Administrator
labawsdynamodb create_items.py
Project

```
labawsdynamodb C:\Users\Administrator\PycharmProjects\labawsdynamodb
  venv library root
  create_items.py
  create_table.py
  delete_item.py
  delete_table.py
  query.py
  read_items.py
  scan.py
  update_item.py
External Libraries
```

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Step 2: Update Items in DynamoDB Table

63. **Double-click** on the ***update_items.py*** Python file.

Info: **Examine the code:**

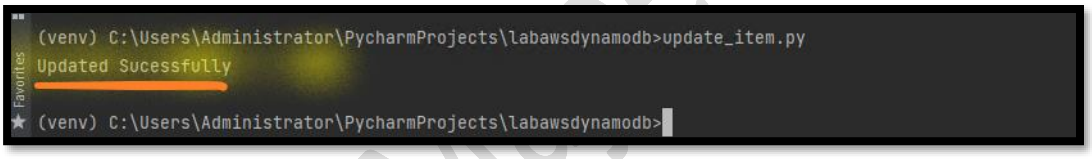
1. **Updating** the **Location attribute** of the existing items in **DynamoDB Table** name: '**students**'.
    a. **Updating** the **attribute** based on **Partition key** as '**1**' and **Sort key** as '**Ahmad1**'.
    b. **Updating** the **Location** to '**Hyderabad**'.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python update_item.py**.

Note: If able to update item succesfully, you will see the response "**Updated Sucessfully**".



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>update_item.py
Updated Sucessfully

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

Read the Updated Item from DynamoDB Table

64. **Double-click** on the ***update_items.py*** Python file.

a. **Replace** the **Values** for Items in **Row 10** to update the **id** to **1**.

b. **Replace** the **Values** for Items in **Row 11** to update the **username** to **Ahmad1**.

c. **From** the **PyCharm IDE**.

i. Select **CTRL + S**.

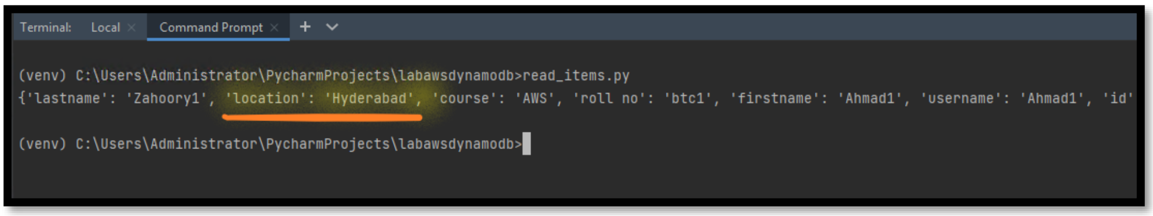
d. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python read_items.py**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Note: View the **Changes** (locations) after update.



```
Terminal: Local × Command Prompt × + ∨

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>read_items.py
{'lastname': 'Zahoory1', 'location': 'Hyderabad', 'course': 'AWS', 'roll no': 'btc1', 'firstname': 'Ahmad1', 'username': 'Ahmad1', 'id'

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

Update the Code to Update Multiple Attributes

Info: **Examine the code:**

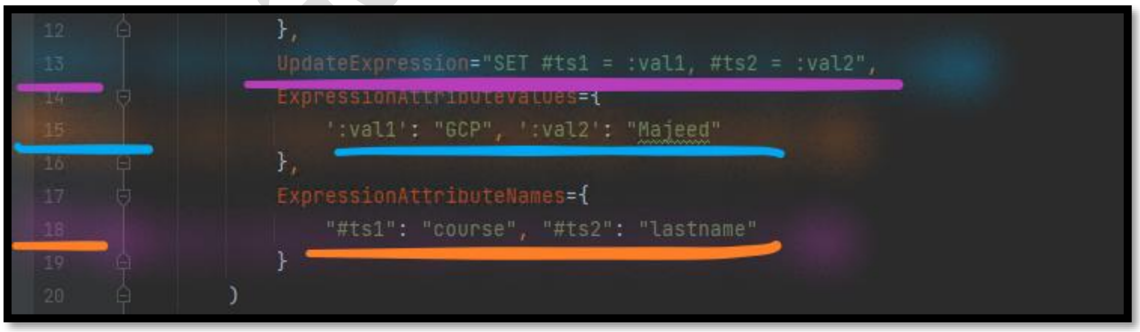
1. **Updating** the **Location attribute** of the existing items in **DynamoDB Table** name: '**students**'.
   a. **Updating** the **Location** to '**Hyderabad**'.

65. **Double-click** on the ***update_item.py*** Python file.

a. **Replace** the **Values** in **Row 13** from **UpdateExpression="SET #ts = :val1",** to **UpdateExpression="SET #ts1 = :val1, #ts2 = :val2",**

b. **Replace** the **Values** in **Row 15** from **':val1': "Hyderabad"** to **':val1': "GCP", ':val2': "Majeed"**

c. **Replace** the **Values** in **Row 18** from **"#ts": "locations"** to **"#ts1": "course", "#ts2": "lastname"**



```
12        },
13        UpdateExpression="SET #ts1 = :val1, #ts2 = :val2",
14        ExpressionAttributeValues={
15            ':val1': "GCP", ':val2': "Majeed"
16        },
17        ExpressionAttributeNames={
18            "#ts1": "course", "#ts2": "lastname"
19        }
20    )
```

d. **From** the **PyCharm IDE**.

   i. Select **CTRL + S**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

e. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python update_item.py**.

Note: If able to update item succesfully, you will see the response "**Updated Sucessfully**".

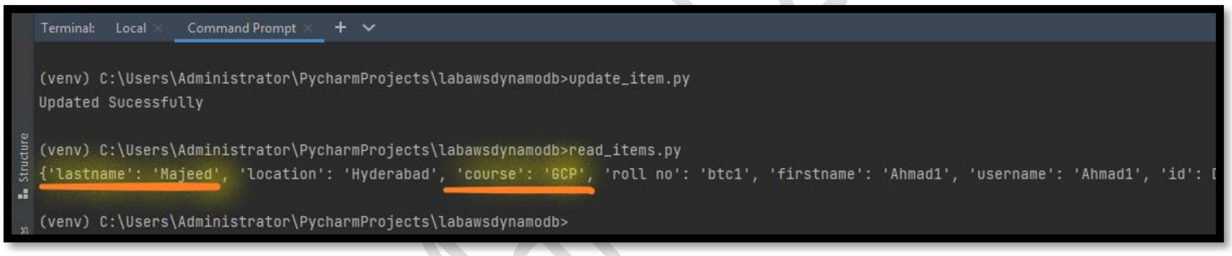
Read the Updated Item from DynamoDB Table

f. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python read_items.py**.

Note: View the **Changes** after update.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>update_item.py
Updated Sucessfully

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>read_items.py
{'lastname': 'Majeed', 'location': 'Hyderabad', 'course': 'GCP', 'roll no': 'btc1', 'firstname': 'Ahmad1', 'username': 'Ahmad1', 'id': I

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

Step 3: Scan Items from DynamoDB Table

66. **Double-click** on the ***scan.py*** Python file.

Info: **Examine the code:**

1. **Scan** the **items** from **DynamoDB Table** name: '**students**'.

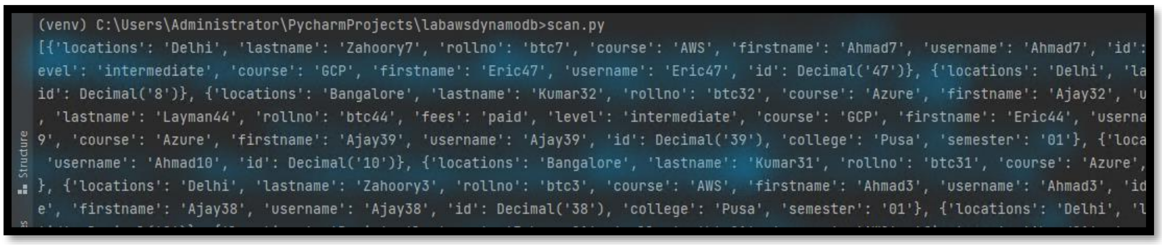
a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python scan.py**.

Note: You can see **all** the **items** in the Terminal.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>scan.py
[{'locations': 'Delhi', 'lastname': 'Zahoory7', 'rollno': 'btc7', 'course': 'AWS', 'firstname': 'Ahmad7', 'username': 'Ahmad7', 'id':
evel': 'intermediate', 'course': 'GCP', 'firstname': 'Eric47', 'username': 'Eric47', 'id': Decimal('47')}, {'locations': 'Delhi', 'la
id': Decimal('8')}, {'locations': 'Bangalore', 'lastname': 'Kumar32', 'rollno': 'btc32', 'course': 'Azure', 'firstname': 'Ajay32', 'u
, 'lastname': 'Layman44', 'rollno': 'btc44', 'fees': 'paid', 'level': 'intermediate', 'course': 'GCP', 'firstname': 'Eric44', 'userna
9', 'course': 'Azure', 'firstname': 'Ajay39', 'username': 'Ajay39', 'id': Decimal('39'), 'college': 'Pusa', 'semester': '01'}, {'loca
 'username': 'Ahmad10', 'id': Decimal('10')}, {'locations': 'Bangalore', 'lastname': 'Kumar31', 'rollno': 'btc31', 'course': 'Azure',
}, {'locations': 'Delhi', 'lastname': 'Zahoory3', 'rollno': 'btc3', 'course': 'AWS', 'firstname': 'Ahmad3', 'username': 'Ahmad3', 'id
e', 'firstname': 'Ajay38', 'username': 'Ajay38', 'id': Decimal('38'), 'college': 'Pusa', 'semester': '01'}, {'locations': 'Delhi', 'l
```

Scan with Filtering

67. **Double-click** on the ***scan.py*** Python file.

   a. **Add** the **Following** in **Row 7**:

```
course = "GCP"
```

   b. **Add** the **Following** in **Row 9 and 10**:

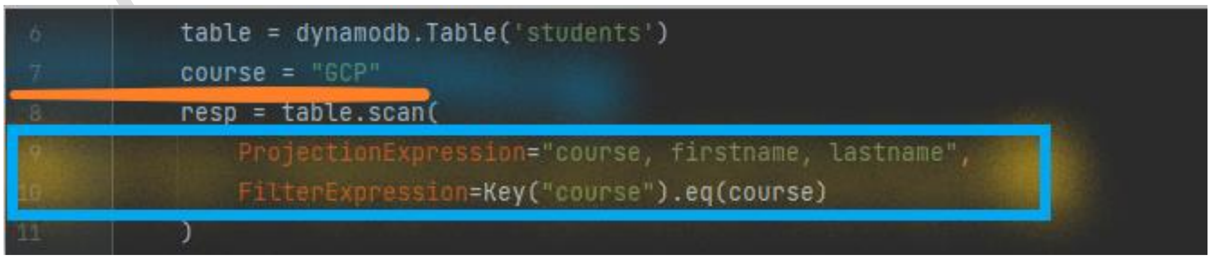
```
ProjectionExpression="course, firstname, lastname",
FilterExpression=Key("course").eq(course)
```

Info: **Examine the code:**

1. **Scan** the **items** from **DynamoDB Table** which have attribute '**Course**' as '**GCP**'.
   a. **Display** the **Result** with attributes **Course**, **Firstname** and **Lastname** only.

Info:

1. **ProjectionExpression** specifies the **attributes** you want in the scan result.
2. **FilterExpression** specifies a **condition** that returns only items that satisfy the condition.



```
    table = dynamodb.Table('students')
    course = "GCP"
    resp = table.scan(
        ProjectionExpression="course, firstname, lastname",
        FilterExpression=Key("course").eq(course)
    )
```

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

c. **From** the **PyCharm IDE**.

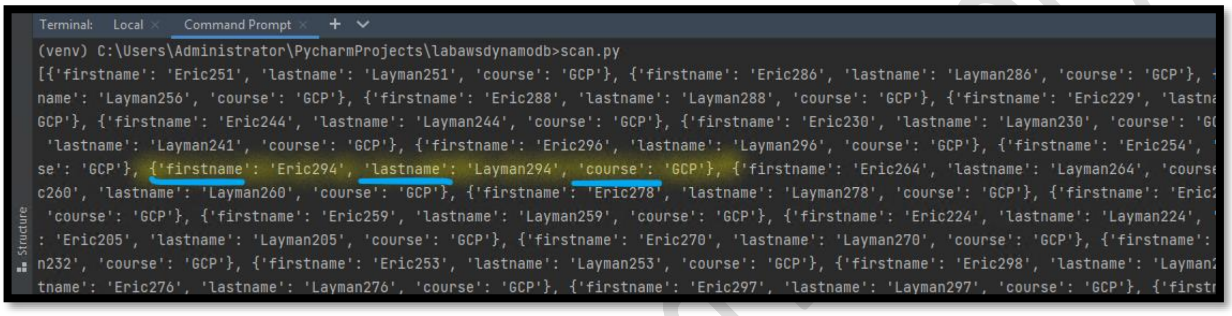
i. Select **CTRL + S**.

d. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python scan.py**.

Note: You can see the **items** as per the **filtering**, where **attribute course** is equal to **value GCP**.


Terminal: Local × Command Prompt × + ⌄

```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>scan.py
[{'firstname': 'Eric251', 'lastname': 'Layman251', 'course': 'GCP'}, {'firstname': 'Eric286', 'lastname': 'Layman286', 'course': 'GCP'},
name': 'Layman256', 'course': 'GCP'}, {'firstname': 'Eric288', 'lastname': 'Layman288', 'course': 'GCP'}, {'firstname': 'Eric229', 'lastna
GCP'}, {'firstname': 'Eric244', 'lastname': 'Layman244', 'course': 'GCP'}, {'firstname': 'Eric230', 'lastname': 'Layman230', 'course': 'GC
 'lastname': 'Layman241', 'course': 'GCP'}, {'firstname': 'Eric296', 'lastname': 'Layman296', 'course': 'GCP'}, {'firstname': 'Eric254',
se': 'GCP'}, {'firstname': 'Eric294', 'lastname': 'Layman294', 'course': 'GCP'}, {'firstname': 'Eric264', 'lastname': 'Layman264', 'course
c260', 'lastname': 'Layman260', 'course': 'GCP'}, {'firstname': 'Eric278', 'lastname': 'Layman278', 'course': 'GCP'}, {'firstname': 'Eric
 'course': 'GCP'}, {'firstname': 'Eric259', 'lastname': 'Layman259', 'course': 'GCP'}, {'firstname': 'Eric224', 'lastname': 'Layman224',
: 'Eric205', 'lastname': 'Layman205', 'course': 'GCP'}, {'firstname': 'Eric270', 'lastname': 'Layman270', 'course': 'GCP'}, {'firstname':
n232', 'course': 'GCP'}, {'firstname': 'Eric253', 'lastname': 'Layman253', 'course': 'GCP'}, {'firstname': 'Eric298', 'lastname': 'Layman
tname': 'Eric276', 'lastname': 'Layman276', 'course': 'GCP'}, {'firstname': 'Eric297', 'lastname': 'Layman297', 'course': 'GCP'}, {'firstr
```

Scan with Multiple Filtering

68. **Double-click** on the ***scan.py*** Python file.

a. **Update** the **Following** from course = "GCP" in **Row 7** to:

```
course = "GCP"; locations = "Hyderabad"
```

b. **Update** the **Following** from

ProjectionExpression="course, firstname, lastname", FilterExpression=Key("course").eq(course)

in **Row 9 and 10** to:

```
ProjectionExpression="course, firstname, lastname, locations",
FilterExpression=Key("course").eq(course) and Key("locations").eq(locations)
```

Info: **Examine the code:**

1. **Scan** the **items** from **DynamoDB Table** which have attribute '**Course**' as '**GCP**' and '**Location**' as '**Hyderabad**.

   a. **Display** the **Result** with attributes **Course**, **Firstname**, **Lastname** and **Locations** only.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

c. **From** the **PyCharm IDE**.

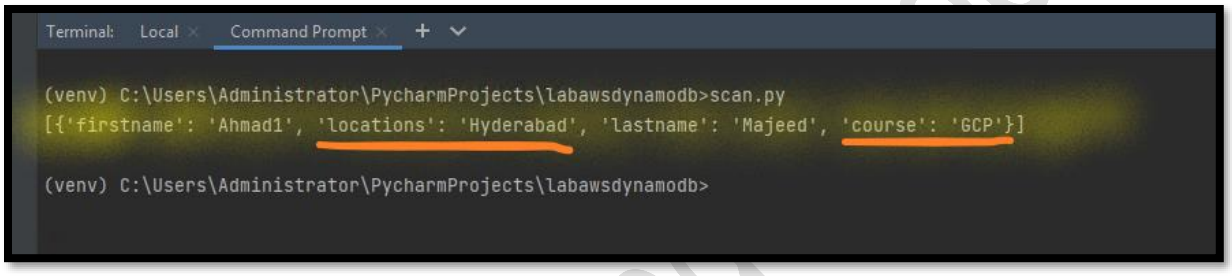
i. Select **CTRL + S**.

d. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python scan.py**.

Note: You can see the **items** as per the **filtering**, where **attribute course** is equal to **value GCP** **and** (Conditional Operator **AND**) **attribute locations** is equal to **value Hyderabad**.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>scan.py
[{'firstname': 'Ahmad1', 'locations': 'Hyderabad', 'lastname': 'Majeed', 'course': 'GCP'}]

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>
```

Step 8: Query Items from DynamoDB Table

69.**Double-click** on the ***query.py*** Python file.

Info: **Examine the code:**

1. **Query** the **item**s from **DynamoDB Table** name: '**students**'.
   a. **Querying** with the **attribute** based on **Partition key** as '**1**' and **Sort key** as '**Ahmad1**'.

Info: **KeyConditions** are the selection criteria for a Query operation. For a query on a table, you can have **key conditions** only on the table **primary key** attributes and **Filter Expression** can only contain non-primary key attributes.

a. **From** the **PyCharm IDE**.

i. Select **CTRL + S**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

b. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python query.py**.

Note: You can see the **items** as per the **filtering**, where **id** (**primary key**) equal to **1** and filtering on **firstname** (**non-primary key**) equal to **Ahmad1**.

Note: You can see the one **item** as **Primary key** is **unique** (**numbers**) for each items.

Step 9: Delete Item from DynamoDB Table

70. **Double-click** on the Python file ***delete_item.py***.

Info: **Examine the code:**

1. **Query** the **item**s from **DynamoDB Table** name: '**students**'.
   a. **Deleting** with the **attribute** based on **Partition key** as '**2**' and **Sort key** as '**Ahmad2**'.

a. **From** the **PyCharm IDE**.

i. Select **CTRL + S**.

b. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python delete_item.py**.

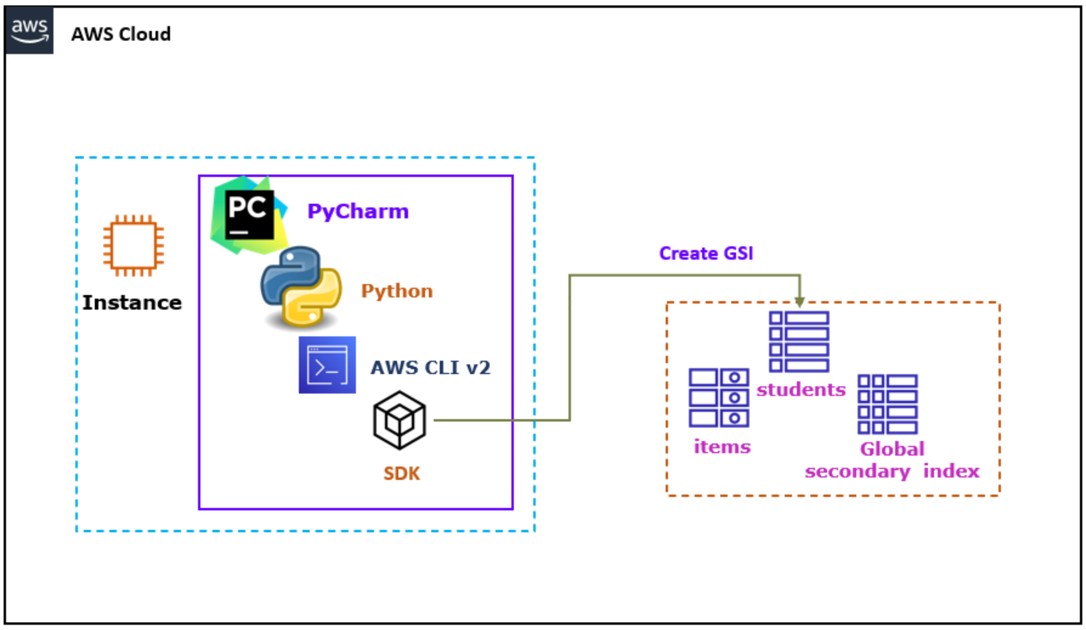
Note: If items added succesfully, you will see the response "**Deleted succesfully**".

Ahmad Majeed Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Task 7: Perform DynamoDB Index [GSI] Operation

In this task, you will create a Global secondary index in existing DynamoDB table.



Step 1: Create the Directory

71. **Expand** the ***labawsdynamodb*** Python Project.
    a. **Right-click** on the ***labawsdynamodb*** Python project.
        i. Select **New**.
            a) Select **Directory**.
                1) In the **New Directory** page:
                    I. **File name**: Write **gsi**.
                        1. Select **Enter**.

Step 2: Develop Python code for Index (GSI) Operations

72. **Unzip** the **LAB-07-02-Python-Code-C.zip**.

Note: **Lab-07-02-Python-Code-C.zip** code file is available with **the Lab manual**.

Note: **Review** the **Code** after opening in the **Notepad**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Ahmad Majeed

73.**From** the **DevPYInstance**, right click on **Start** & **Run**.

a. In the **Open**, write **C:\Users\Administrator**, press **Ok**.

i. Open the **PycharmProjects** folder.

a) Open the **labawsdynamodb** folder.

1) Open the **gsi** folder.

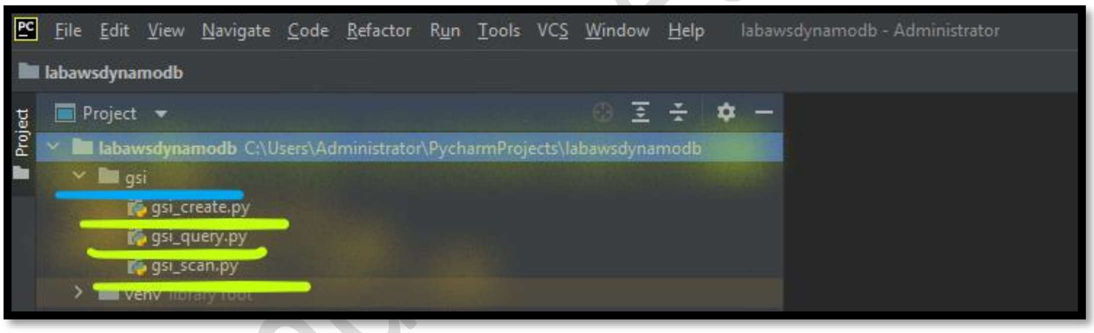
I. **Copy** the **gsi_create.py**, **gsi_scan.py** and **gsi_query.py** Python files in the **gsi** folder.

74.**Return** to the **DevPYInstance**.

a. **Expand** the ***labawsdynamodb*** Python Project.

i. **Expand** the ***gsi*** Directory.

Note: You can see the **gsi_create.py**, **gsi_scan.py** and **gsi_query.py** Python files.


File Edit View Navigate Code Refactor Run Tools VCS Window Help
labawsdynamodb - Administrator
labawsdynamodb
Project
labawsdynamodb C:\Users\Administrator\PycharmProjects\labawsdynamodb
gsi
gsi_create.py
gsi_query.py
gsi_scan.py
venv library root

b. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

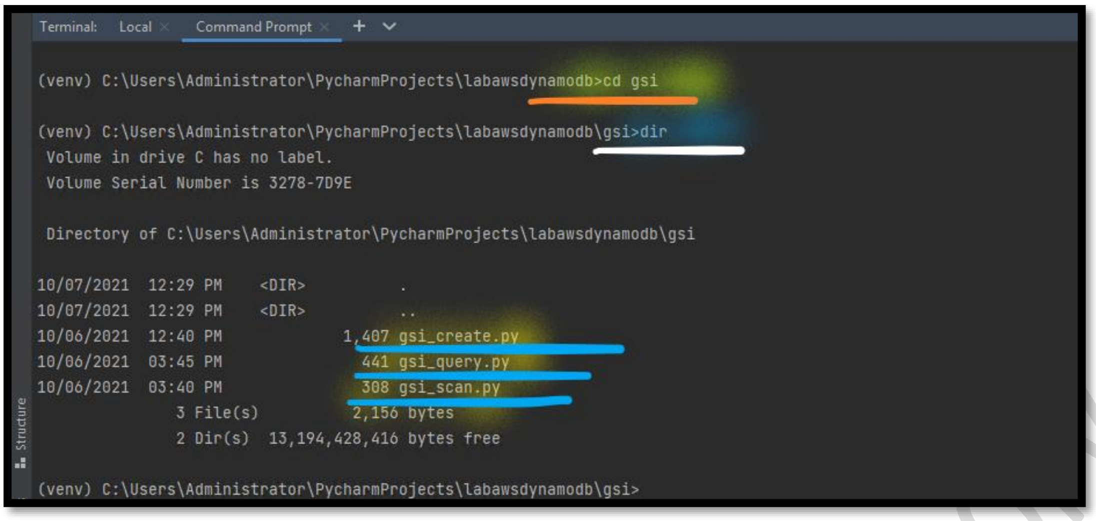
a) **Type** **cd gsi**.

b) **Type** **dir**.

Note: You can see the **gsi_create.py**, **gsi_scan.py** and **gsi_query.py** Python files.

Ahmad Mohammad Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb>cd gsi

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb\gsi>dir
 Volume in drive C has no label.
 Volume Serial Number is 3278-7D9E

 Directory of C:\Users\Administrator\PycharmProjects\labawsdynamodb\gsi

10/07/2021  12:29 PM    <DIR>          .
10/07/2021  12:29 PM    <DIR>          ..
10/06/2021  12:40 PM             1,407 gsi_create.py
10/06/2021  03:45 PM               441 gsi_query.py
10/06/2021  03:40 PM               308 gsi_scan.py
               3 File(s)          2,156 bytes
               2 Dir(s)  13,194,428,416 bytes free

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb\gsi>
```

Step 3: Create Global Secondary Index (GSI)

75. **Expand** the ***labawsdynamodb*** Python Project.

76. **Expand** the ***gsi*** Directory.

77. **Double-click** on the ***gsi_create.py*** Python file.

Info: **Examine the code:**

1. **Creating** the DynamoDB **Global Secondary Index**:
   a. **Index name**: '**students2**'.
   b. **Partition key** (or Hash key): '**locations**' as string **instead** of '**id**'.
   c. **Sort key** (or Range key): '**course**' as string **instead** of . '**username**'.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python gsi_create.py**.

Note: If table created succesfully, you will see the response "**Created succesfully**".

Note: **Go to next task**, but **Don't close** the **DevPYInstance Console**.

Ahmad Zahoory

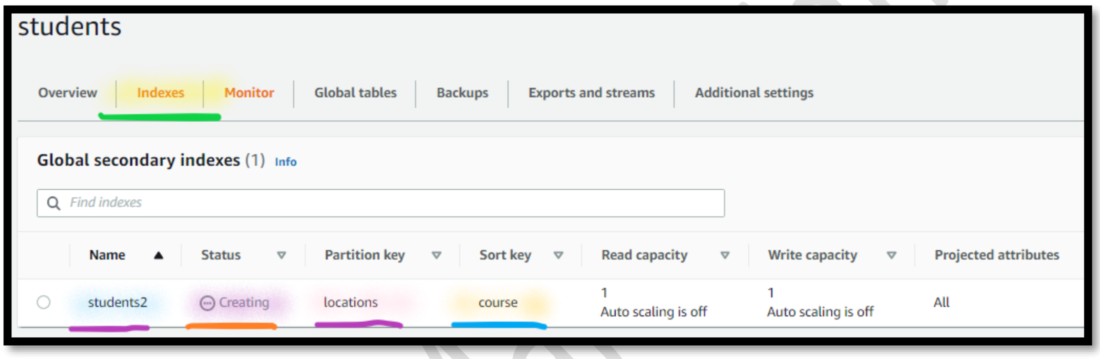
© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Verify the DynamoDB GSI from AWS Console

78. In the **AWS Management Console**, on the **Services** menu, click **DynamoDB**.
79. Select **Tables**.
    a. Open **students** table.
        i. Open the **Indexes**.

Note: You can see the "**students2**" index in creating State.

Note: You can see the new **Partition key** and **Sort key**.



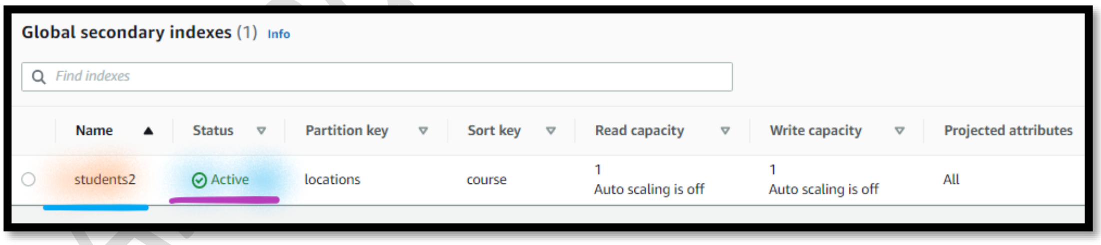
students

Overview | Indexes | Monitor | Global tables | Backups | Exports and streams | Additional settings

Global secondary indexes (1) Info

	Name	Status	Partition key	Sort key	Read capacity	Write capacity	Projected attributes
○	students2	Creating	locations	course	1 Auto scaling is off	1 Auto scaling is off	All

Note: **Wait**, till Index gets created and **Status** is **Active**.



Global secondary indexes (1) Info

	Name	Status	Partition key	Sort key	Read capacity	Write capacity	Projected attributes
○	students2	Active	locations	course	1 Auto scaling is off	1 Auto scaling is off	All

Step 4: Scan Items from DynamoDB Index (GSI)

80. **Return** to the **DevPYInstance**.
81. **Double-click** on the ***gsi_scan.py*** Python file.

Info: **Examine the code:**

1. **Scan** the **items** from **DynamoDB Table Index**: '**students2**'.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python gsi_scan.py**.

Note: You can see **all** the **items** in the Terminal.

Scan with Multiple Filtering

82. **Double-click** on the ***gsi_scan.py*** Python file.

a. **Add** the **Following** in **Row 7**:

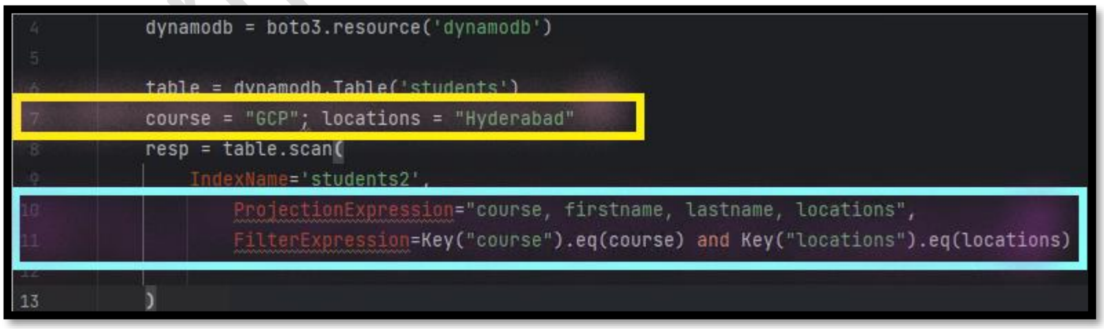
```
course = "GCP"; locations = "Hyderabad"
```

b. **Add** the **Following** in **Row 10 and 11**:

```
ProjectionExpression="course, firstname, lastname, locations",
FilterExpression=Key("course").eq(course) and Key("locations").eq(locations)
```

Info: **Examine the code:**

1. **Scan** the **items** from **DynamoDB Table Index**: '**students2**', which have attribute '**Course**' as '**GCP**' and '**Location**' as '**Hyderabad**.
   a. **Display** the **Result** with attributes **Course**, **Firstname**, **Lastname** and **Locations** only.



```
4  dynamodb = boto3.resource('dynamodb')
5
6  table = dynamodb.Table('students')
7  course = "GCP"; locations = "Hyderabad"
8  resp = table.scan(
9      IndexName='students2',
10         ProjectionExpression="course, firstname, lastname, locations",
11         FilterExpression=Key("course").eq(course) and Key("locations").eq(locations)
12
13 )
```

c. **From** the **PyCharm IDE**.

i. Select **CTRL + S**.

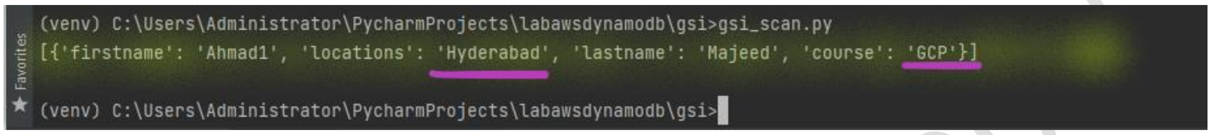
© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

d. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python gsi_scan.py**.

Note: You can see the **items** as per the **filtering**, where **attribute course** is equal to **value GCP** **and** (Conditional Operator **AND**) **attribute locations** is equal to **value Hyderabad**.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb\gsi>gsi_scan.py
[{'firstname': 'Ahmad1', 'locations': 'Hyderabad', 'lastname': 'Majeed', 'course': 'GCP'}]

(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb\gsi>
```

Scan with Multiple Filtering

83. **Double-click** on the ***gsi_scan.py*** Python file.

a. **Update** the **Locations value** in **Row 7** from **Hyderabad** to **Chicago**.

b. **From** the **PyCharm IDE**.

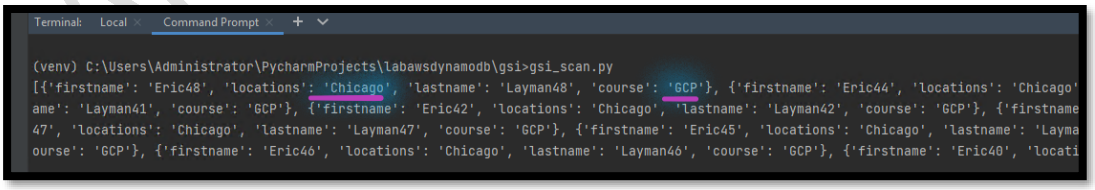
i. Select **CTRL + S**.

c. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python gsi_scan.py**.

Note: You can see the **items** as per the **filtering**, where **attribute course** is equal to **value GCP** **and** (Conditional Operator **AND**) **attribute locations** is equal to **value Chicago**.



```
(venv) C:\Users\Administrator\PycharmProjects\labawsdynamodb\gsi>gsi_scan.py
[{'firstname': 'Eric48', 'locations': 'Chicago', 'lastname': 'Layman48', 'course': 'GCP'}, {'firstname': 'Eric44', 'locations': 'Chicago'
ame': 'Layman41', 'course': 'GCP'}, {'firstname': 'Eric42', 'locations': 'Chicago', 'lastname': 'Layman42', 'course': 'GCP'}, {'firstname
47', 'locations': 'Chicago', 'lastname': 'Layman47', 'course': 'GCP'}, {'firstname': 'Eric45', 'locations': 'Chicago', 'lastname': 'Layma
ourse': 'GCP'}, {'firstname': 'Eric46', 'locations': 'Chicago', 'lastname': 'Layman46', 'course': 'GCP'}, {'firstname': 'Eric40', 'locati
```

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Step 5: Query Items from DynamoDB Index (GSI)

84. **Double-click** on the ***gsi_query.py*** Python file.

Info: **Examine the code:**

1. **Query** the **items** from **DynamoDB Table index**: '**students2**'.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python gsi_query.py**.

Note: You can see the **items** as per the **filtering**, where **locations** (**primary key**) equal to **Chicago** and filtering on **fees** (**non-primary key**) equal to **paid**.

Note: You can see **multiple item** as **primary key** is **same** for **multiple items**.

Step 6: Delete DynamoDB Table

85. **Double-click** on the ***delete_table.py*** Python file.

Info: **Examine the code:**

1. **Delete** the **DynamoDB Table** name: '**students**'.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **cd..**.

b) **Type** **python delete_table.py**.

Note: If table deleted succesfully, you will **not see** any **response**.

Ahmad Majeed Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Note: **Go to next task**, but **Don't close** the **DevPYInstance Console**.

Verify the DynamoDB Table from AWS Console

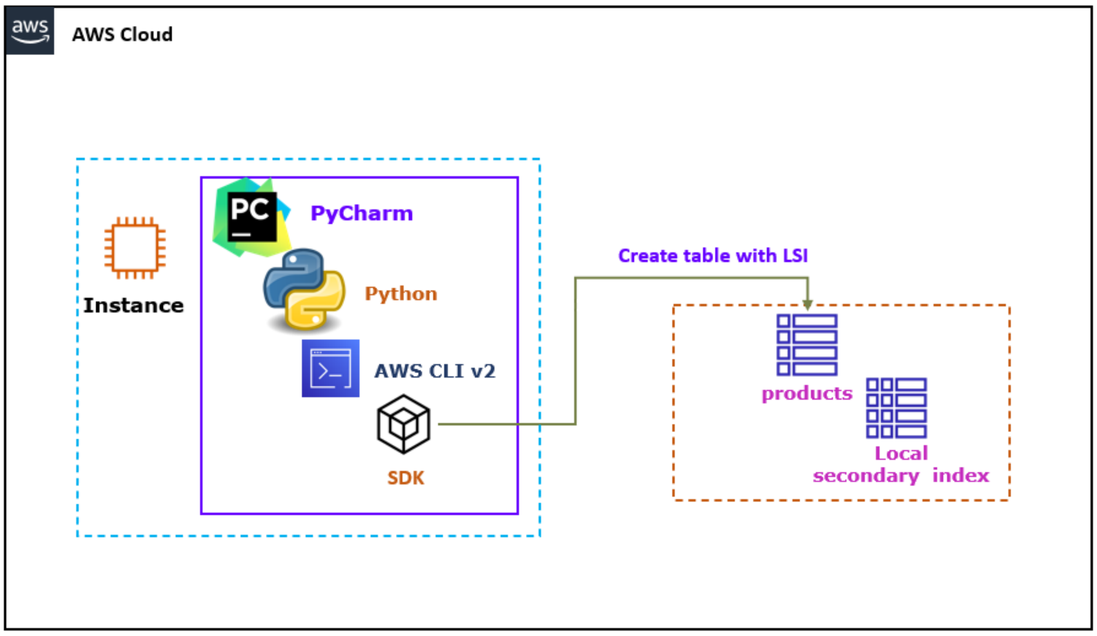
86. In the **AWS Management Console**, on the **Services** menu, click **DynamoDB**.

87. Select **Tables**.

Note: You can see the "**students**" table is deleted.

Task 4: Perform DynamoDB Index [LSI] Operation

In this task, you will create DynamoDB table with Local secondary index.



Step 1: Create the Directory

88. **Return** to the **DevPYInstance**.

89. **Expand** the ***labawsdynamodb*** Python project.

    a. **Right-click** on the ***labawsdynamodb*** Python project.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

i. Select **New**.

a) Select **Directory**.

1) In the **New Directory** page:

I. **File name**: Write **lsi**.

1. Select **Enter**.

Step 2: Develop Python code for Index (LSI) Operations

90. **Unzip** the **LAB-07-02-Python-Code-D.zip**.

Note: **Lab-07-02-Python-Code-D.zip** code file is available with the **Lab manual**.

Note: **Review** the **Code** after opening in the **Notepad**.

91. **From** the **DevPYInstance**, right click on **Start** & **Run**.

a. In the **Open**, write **C:\Users\Administrator**, press **Ok**.

i. Open the **PycharmProjects** folder.

a) Open the **labawsdynamodb** folder.

1) Open the **lsi** folder.

I. **Copy** the **lsi_create.py**, **lsi_put.py** and **lsi_query.py** Python files in the **lsi** folder.

92. **Expand** the ***labawsdynamodb*** Python Project.

a. **Expand** the ***lsi*** Directory.

Note: You can see the **lsi_create.py**, **lsi_put.py** and **lsi_query.py** Python files.

Ahmad Majeed Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

b. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **cd lsi**.

b) **Type** **dir**.

Note: You can see the **lsi_create.py**, **lsi_put.py** and **lsi_query.py** Python files.

Step 3: Create the Local Secondary Index (LSI)

93. **Expand** the ***labawsdynamodb*** Python project.

94. **Expand** the ***lsi*** Directory.

95. **Double-click** on the ***lsi_create.py*** Python file.

Info: **Examine the code:**

1. **Creating** the **DynamoDB Table**, name: '**products**'.
   a. **Partition key** (or Hash key): '**product_id**'.
   b. **Sort key** (or Range key): '**product_name**'.
2. **Creating** the DynamoDB **Local Secondary Index**:
   a. **Index name**: '**store**'.
   b. **Partition key** (or Hash key): '**product_id**'.
   c. **Sort key** (or Range key): '**store_location**' instead of **product_name**.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python lsi_create.py**.

Note: If table created succesfully, you will see the response "**Creating**".

Note: **Go to next task,** but **Don't close** the **DevPYInstance** Console.

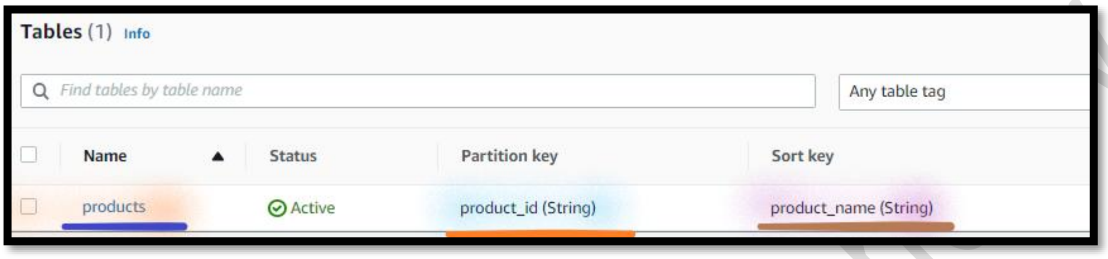
Ahmad Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Verify the DynamoDB LSI from AWS Console

96. In the **AWS Management Console**, on the **Services** menu, click **DynamoDB**.
97. Select **Tables**.

Note: You can see the "**products**" table.



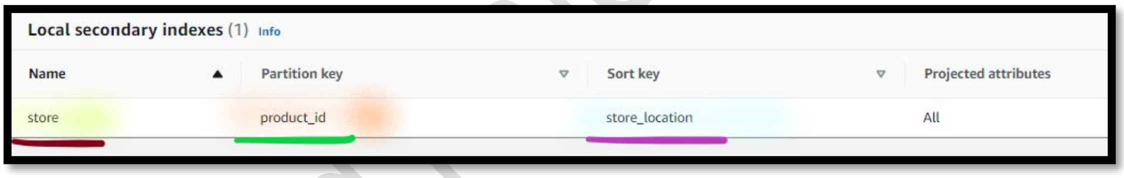
Tables (1) Info

Name	Status	Partition key	Sort key
products	Active	product_id (String)	product_name (String)

a. Open the **products**.

i. Open the **Indexes**.

Note: You can see the "**store**" index.



Local secondary indexes (1) Info

Name	Partition key	Sort key	Projected attributes
store	product_id	store_location	All

Step 4: Create Items in DynamoDB Table

98. **Return** to the **DevPYInstance**.
99. **Double-click** on the ***lsi_put.py*** Python file.

Info: **Examine the code:**

1. **Creating** the **Items** in **DynamoDB Table** name: '**products**'.
   a. **Adding Items**.

a. ***Go below*** in the console, click on the **Terminal**.

i. From the **Terminal** (***command prompt***):

a) **Type** **python lsi_put.py**.

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

Note: **Go to next task,** but **Don't close** the **DevPYInstance** Console.

Verify the Added Items in DynamoDB Table from AWS Console

100. In the **AWS Management Console**, on the **Services** menu, click **DynamoDB**.
101. Select **Explore items**.
    a. Open the **products** table.

Note: You can see the added **items** in **products** table.

Step 5: Query Items from DynamoDB Index (LSI)

102. **Double-click** on the ***lsi_query.py*** Python file.

Info: **Examine the code:**
1. **Query** the **items** from DynamoDB Table **index**: '**store**'.

    a. ***Go below*** in the console, click on the **Terminal**.
        i. From the **Terminal** (***command prompt***):
            a) **Type** **python lsi_query.py**.

Note: You can see the **items** as per the **filtering**, where **product_id** (**primary key**) equal to **p_1** and **store_location** (**sort_key**) equal to **Delhi**.

Note: You can see the **one item** as **primary key** is **unique** (**p_1**) for each items.

Ahmad Majeed Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.

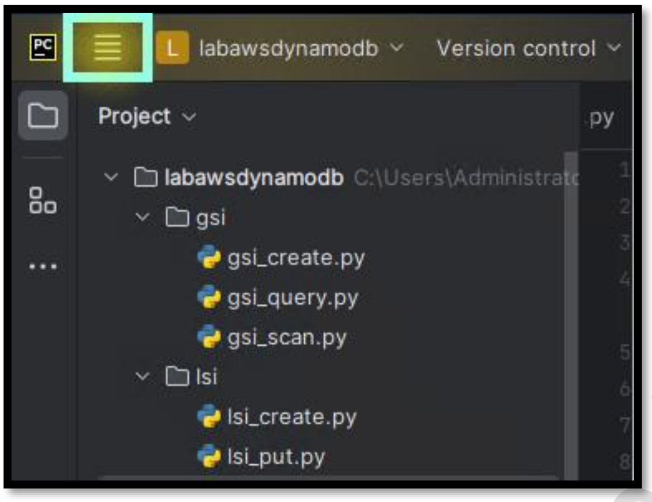
Task 5: Close the Project

In this task, you will close the Python project.

Step 1: Close the Project

103. **From** the **PyCharm IDE**.

  a. Select the **Menu**.



    i. Select **File**.

      a) Select **Close project**.

Note

- Do not delete any resources you deployed in this lab.
- You will be using them in the next lab of this module.

Ahmad Majeed Zahoory

© No part of this manual, may be reproduced in whole or in part in any manner without the permission of the copyright owner.