PRACTICAL: 2

AIM: To understand the fundamentals of developing applications on AWS, explore AWS development tools, and learn how to use AWS SDKs, AWS CLI, and AWS CloudShell for cloud-based application development.

Scenario:

Frank and Martha are a married team who own and operate a small café business that sells desserts and coffee. Their daughter, Sofía, works at the café. Sofía is pursuing a degree in cloud computing at a local university in the evenings and on the weekends. She has Python development skills and is learning more about how to develop solutions in the cloud. Sofía is eager to start developing a web presence for the café. She thinks that before she starts coding, it would be a good idea to decide on a development environment for developing and running her code. She decides to explore at least two options that are available on AWS.

Lab overview and objectives:

In this lab, you will take on the role of Sofía. You will connect to an AWS CloudShell environment and explore its capabilities. You will also launch an instance of AWS Cloud9, connect to it, and explore the layout and functionality of its integrated development environment (IDE). In addition, you will use Amazon CodeWhisperer inside the AWS Cloud9 IDE to generate a Python script.

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

- Connect to AWS CloudShell and run AWS Command Line Interface (AWS CLI) commands and AWS SDK code from it.
- Create an AWS Cloud9 development environment and connect to the browser-based IDE.
- Copy files to and from Amazon Simple Storage Service (Amazon S3), CloudShell, and AWS Cloud9.
- Install the AWS SDK for Python (Boto3) on an AWS Cloud9 instance.
- Use the AWS Cloud9 development environment to create files and run code files.
- Use Amazon CodeWhisperer in AWS Cloud9 to generate code to interact with AWS services. •

THEORY:

1. **AWS CloudShell**
   AWS CloudShell is a browser-based shell environment provided by Amazon Web Services. It allows users to interact with AWS services using the AWS Command Line Interface (AWS CLI) and supported programming languages.

2. **AWS Cloud9 Integrated Development Environment (IDE)**
   AWS Cloud9 is a cloud-based Integrated Development Environment that supports multiple programming languages and includes tools for development, debugging, and deployment.

3. **Amazon Simple Storage Service (Amazon S3)**

Amazon S3 is a scalable object storage service designed for storing and retrieving large amounts of data.

4. AWS SDK for Python (Boto3)

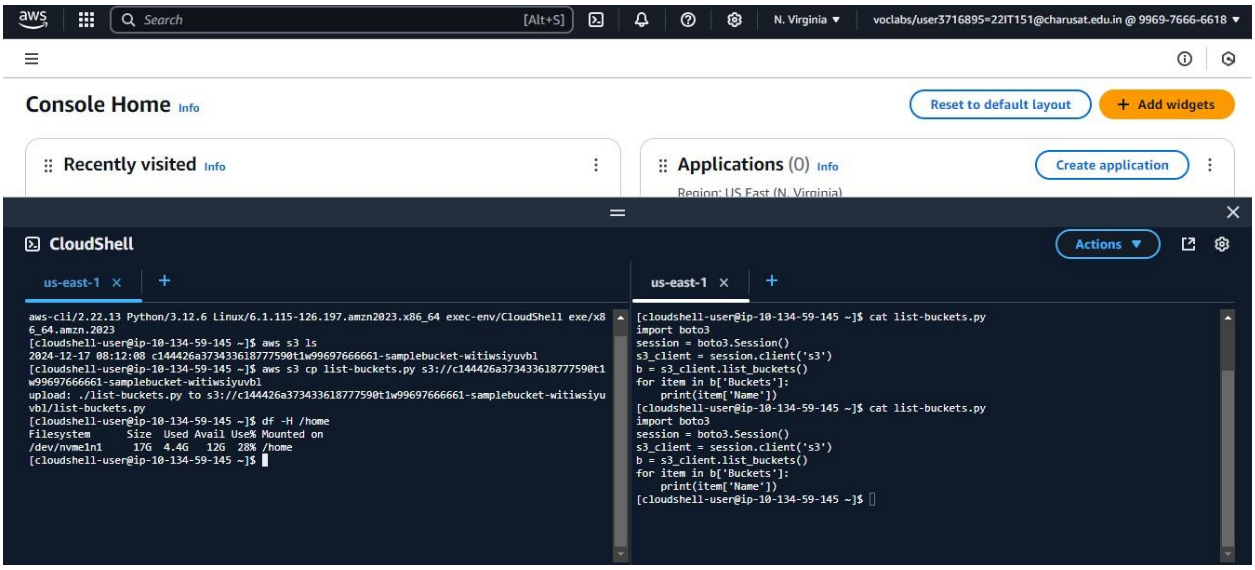
Boto3 is the official software development kit (SDK) for Python to interact with AWS services programmatically.

CODE:

```
aws s3 cp s3://<bucket-name>/list-buckets.py .
```

download the list-buckets.py file from Amazon S3 to the local storage on VS Code IDE

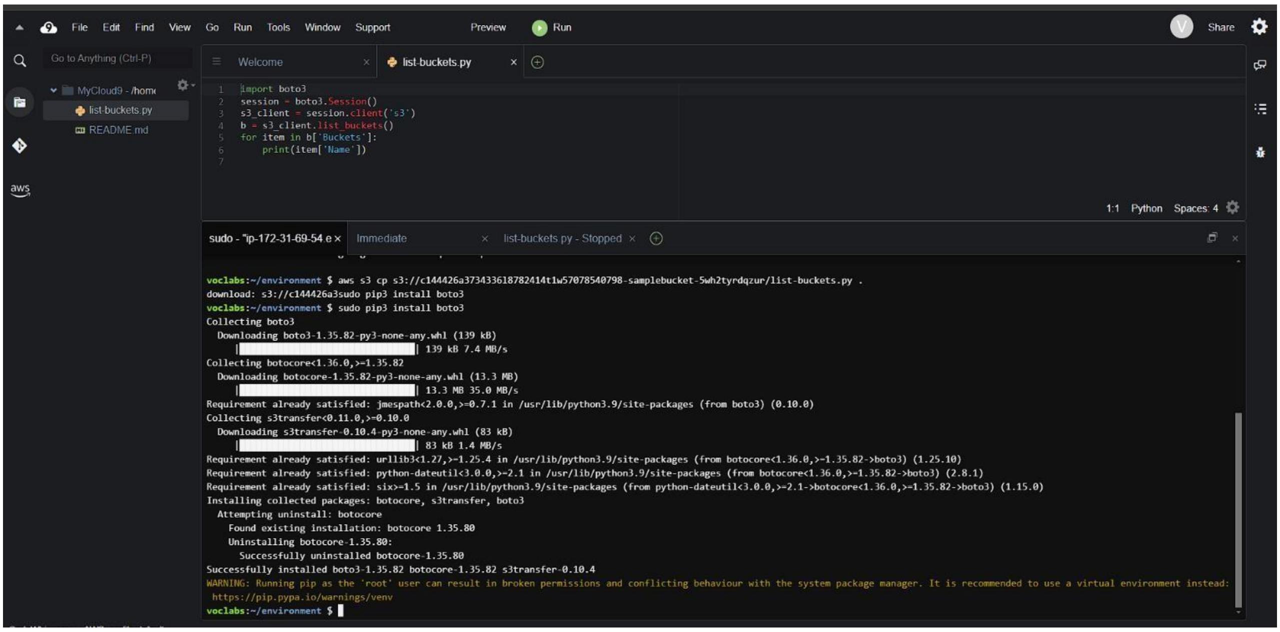
OUTPUT:



```
aws-cli/2.22.13 Python/3.12.6 Linux/6.1.115-126.197.amzn2023.x86_64 exec-env/CloudShell exe/x8
6_64.amzn.2023
[cloudshell-user@ip-10-134-59-145 ~]$ aws s3 ls
2024-12-17 08:12:08 c144426a373433618777590t1w99697666661-samplebucket-witiwsiyuvbl
[cloudshell-user@ip-10-134-59-145 ~]$ aws s3 cp list-buckets.py s3://c144426a373433618777590t1
w99697666661-samplebucket-witiwsiyuvbl
upload: ./list-buckets.py to s3://c144426a373433618777590t1w99697666661-samplebucket-witiwsiyu
vbl/list-buckets.py
[cloudshell-user@ip-10-134-59-145 ~]$ df -H /home
Filesystem      Size  Used Avail Use% Mounted on
/dev/nvme1n1     17G  4.4G   12G  28% /home
[cloudshell-user@ip-10-134-59-145 ~]$
```

```
[cloudshell-user@ip-10-134-59-145 ~]$ cat list-buckets.py
import boto3
session = boto3.Session()
s3_client = session.client('s3')
b = s3_client.list_buckets()
for item in b['Buckets']:
    print(item['Name'])
[cloudshell-user@ip-10-134-59-145 ~]$ cat list-buckets.py
import boto3
session = boto3.Session()
s3_client = session.client('s3')
b = s3_client.list_buckets()
for item in b['Buckets']:
    print(item['Name'])
[cloudshell-user@ip-10-134-59-145 ~]$
```

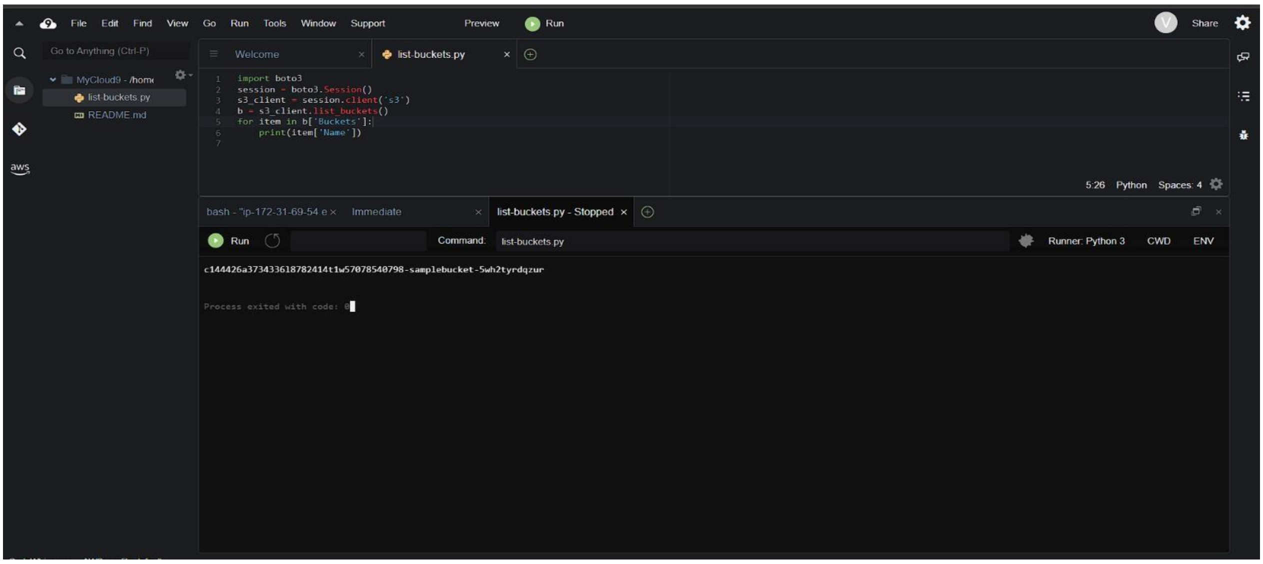
Figure 1: This picture shows that list-buckets.py file is successfully uploaded and shows output



```python
import boto3
session = boto3.Session()
s3_client = session.client('s3')
b = s3_client.list_buckets()
for item in b['Buckets']:
    print(item['Name'])
```

```
voclabs:~/environment $ aws s3 cp s3://c144426a373433618782414t1w57078540798-samplebucket-5wh2tyrdqzur/list-buckets.py .
download: s3://c144426a3sudo pip3 install boto3
voclabs:~/environment $ sudo pip3 install boto3
Collecting boto3
  Downloading boto3-1.35.82-py3-none-any.whl (139 kB)
     |████████████████████████████████| 139 kB 7.4 MB/s
Collecting botocore<1.36.0,>=1.35.82
  Downloading botocore-1.35.82-py3-none-any.whl (13.3 MB)
     |████████████████████████████████| 13.3 MB 35.0 MB/s
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in /usr/lib/python3.9/site-packages (from boto3) (0.10.0)
Collecting s3transfer<0.11.0,>=0.10.0
  Downloading s3transfer-0.10.4-py3-none-any.whl (83 kB)
     |████████████████████████████████| 83 kB 1.4 MB/s
Requirement already satisfied: urllib3<1.27,>=1.25.4 in /usr/lib/python3.9/site-packages (from botocore<1.36.0,>=1.35.82->boto3) (1.25.10)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/lib/python3.9/site-packages (from botocore<1.36.0,>=1.35.82->boto3) (2.8.1)
Requirement already satisfied: six>=1.5 in /usr/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.36.0,>=1.35.82->boto3) (1.15.0)
Installing collected packages: botocore, s3transfer, boto3
  Attempting uninstall: botocore
    Found existing installation: botocore 1.35.80
    Uninstalling botocore-1.35.80:
      Successfully uninstalled botocore-1.35.80
Successfully installed boto3-1.35.82 botocore-1.35.82 s3transfer-0.10.4
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual environment instead:
 https://pip.pypa.io/warnings/venv
voclabs:~/environment $
```

Figure 2: Downloading boto3 module and run list-buckets.py



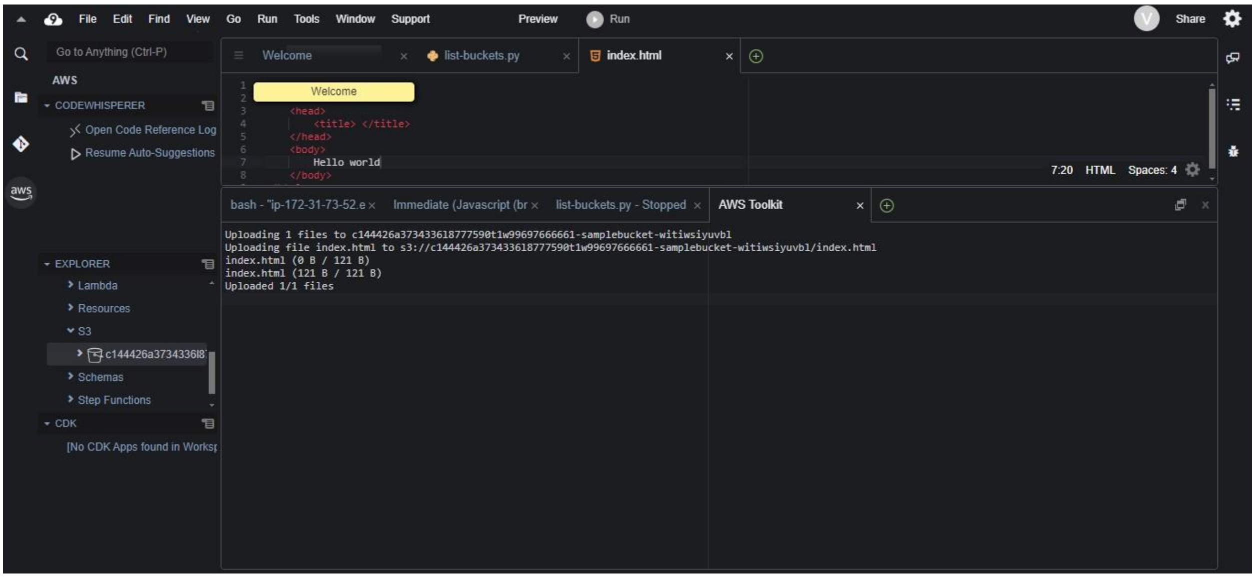
```python
import boto3
session = boto3.Session()
s3_client = session.client('s3')
b = s3_client.list_buckets()
for item in b['Buckets']:
    print(item['Name'])
```

Command: list-buckets.py

```
c144426a373433618782414t1w57078540798-samplebucket-5wh2tyrdqzur


Process exited with code: 0
```

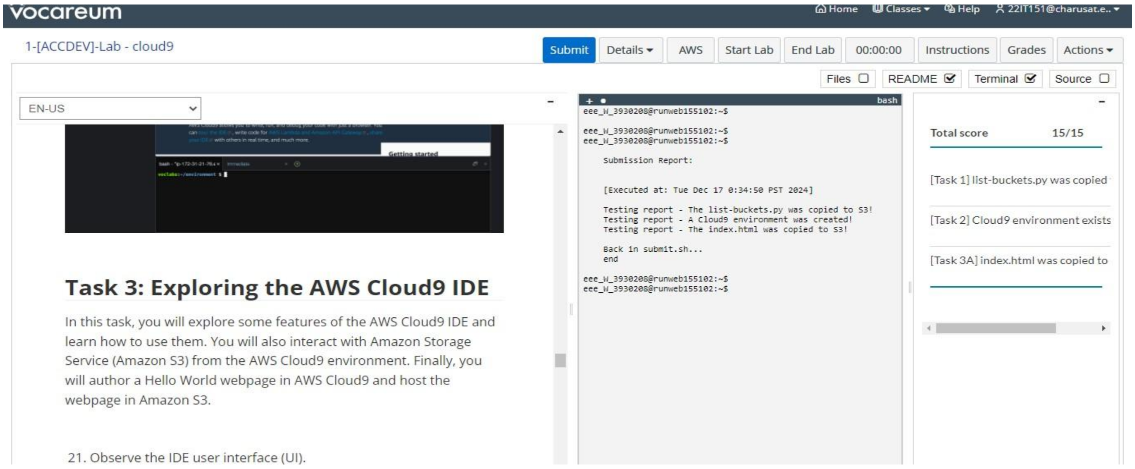
Figure 3: After installing module it successfully shows s3 bucket name



```
<head>
    <title> </title>
</head>
<body>
    Hello world
</body>
```

```
Uploading 1 files to c144426a373433618777590t1w99697666661-samplebucket-witiwsiyuvbl
Uploading file index.html to s3://c144426a373433618777590t1w99697666661-samplebucket-witiwsiyuvbl/index.html
index.html (0 B / 121 B)
index.html (121 B / 121 B)
Uploaded 1/1 files
```

Figure 4: This file is uploaded from AWS Cloud9 instance to s3 bucket



Task 3: Exploring the AWS Cloud9 IDE

In this task, you will explore some features of the AWS Cloud9 IDE and learn how to use them. You will also interact with Amazon Storage Service (Amazon S3) from the AWS Cloud9 environment. Finally, you will author a Hello World webpage in AWS Cloud9 and host the webpage in Amazon S3.

21. Observe the IDE user interface (UI).

```
eee_W_3930208@runweb155102:~$

eee_W_3930208@runweb155102:~$
eee_W_3930208@runweb155102:~$

    Submission Report:


    [Executed at: Tue Dec 17 0:34:50 PST 2024]

    Testing report - The list-buckets.py was copied to S3!
    Testing report - A Cloud9 environment was created!
    Testing report - The index.html was copied to S3!

    Back in submit.sh...
    end

eee_W_3930208@runweb155102:~$
eee_W_3930208@runweb155102:~$
```

Total score	15/15
[Task 1] list-buckets.py was copied	
[Task 2] Cloud9 environment exists	
[Task 3A] index.html was copied to	

Figure 5: Lab Completed

LATEST APPLICATIONS:

1. **Data Engineering**: Automate ETL pipelines using Boto3 and S3.
2. **DevOps**: Deploy CI/CD pipelines and monitor infrastructure with AWS Cloud9 and Code Whisperer.
3. **Machine Learning**: Use Cloud9 to develop machine learning workflows leveraging AWS SageMaker.
4. **IoT Development**: Develop and test IoT applications with the AWS SDK in a secure environment.
5. **Game Development**: Use AWS Cloud9 to integrate with AWS GameLift for game server hosting.

LEARNING OUTCOME:

- Understood the **Systems Development Lifecycle (SDLC)** in AWS cloud development.
- Learned how to install and use the **AWS CLI** for managing AWS services.
- Explored **AWS SDKs** and their role in cloud-based application development.
- Gained knowledge of **Amazon Q Developer** and how AI assists developers.
- Practiced working with **AWS CloudShell** and IDEs for cloud development.

Faculty Sign **Grade**