Lab Guide: Amazon S3 Activities Using Python Boto3 (Windows & Linux/Mac)

Objective

This lab provides step-by-step instructions to perform multiple activities on Amazon S3 using Python Boto3, including:

1. Creating an S3 bucket.
2. Uploading a file to the bucket.
3. Listing bucket contents.
4. Downloading a file from the bucket.
5. Deleting a file from the bucket.
6. Enabling versioning.
7. Deleting the bucket.

Reminder: Ensure the AWS user executing these scripts has the necessary **S3 permissions** attached to their IAM role or user. The required permissions include:

```
□{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "s3:*",
            "Resource": "arn:aws:s3:::*"
        }
    ]
}
```

□Attach this policy or a more restrictive version to the IAM user before proceeding.

Important: S3 bucket names must be globally unique. When creating a bucket, ensure you modify the bucket name in the script to something unique (e.g., `my-boto3-s3-bucket-<yourname>`).

Prerequisites

1. **AWS Account**: Ensure you have access to an AWS account.
2. **AWS CLI Installed**: Download and install AWS CLI from [AWS CLI Download](#).
3. **AWS Credentials Configured**: Run `aws configure` and set up your AWS credentials.
4. **Python Installed**: Ensure you have Python 3.x installed.
5. **Boto3 Installed**: Run `pip install boto3` to install the AWS SDK for Python.

Part 1: Create an S3 Bucket

1. Create a Python script named `create_bucket.py`:

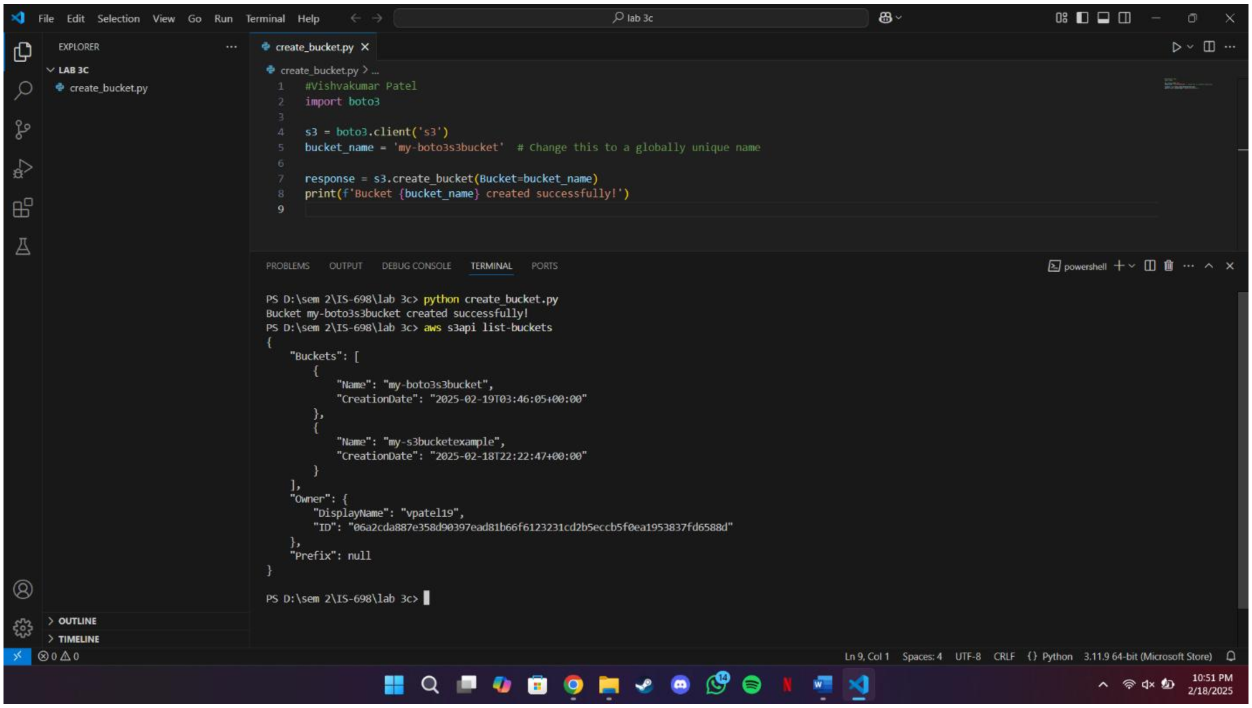
```
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3-s3-bucket-<yourname>'  # Change this to a globally unique name

response = s3.create_bucket(Bucket=bucket_name)

print(f'Bucket {bucket_name} created successfully!')
```

2. Run the script:

```
python create_bucket.py
```

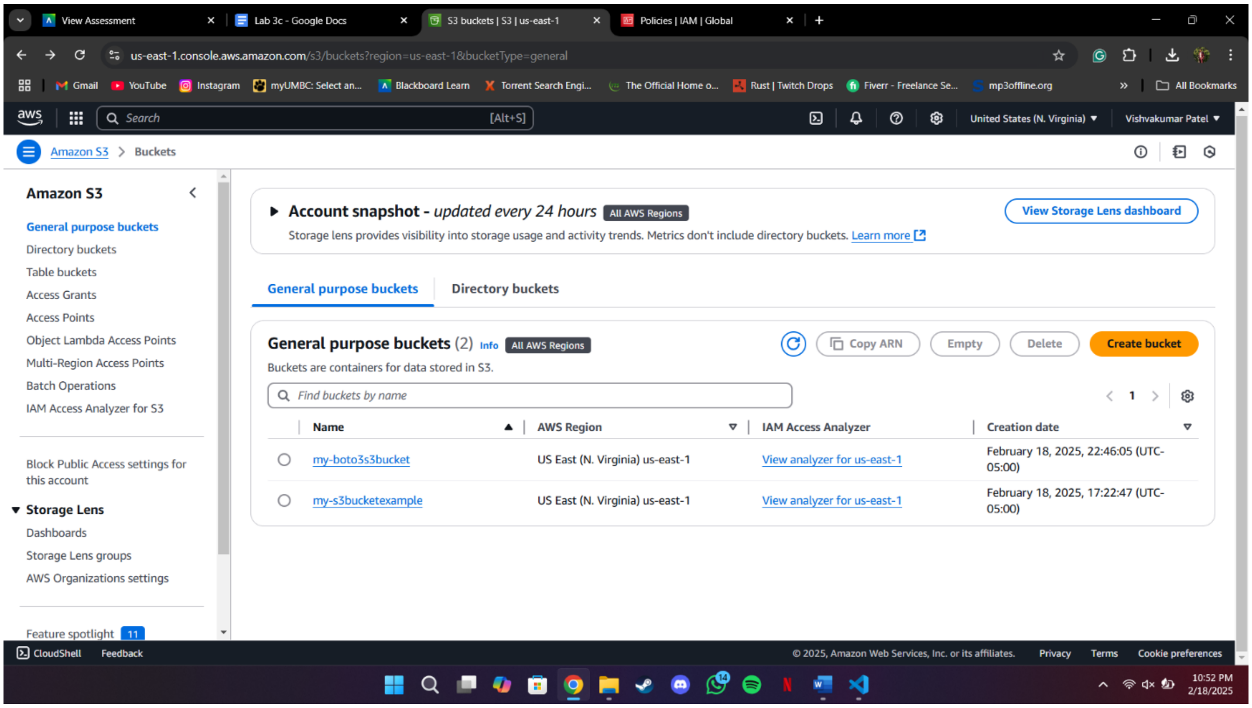
3. Verify the bucket creation using:

```
aws s3api list-buckets
```


File Edit Selection View Go Run Terminal Help
lab 3c
EXPLORER
LAB 3C
create_bucket.py
OUTLINE
TIMELINE
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
Ln 9, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 64-bit (Microsoft Store)
10:51 PM
2/18/2025

```python
#Vishvakumar Patel
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3s3bucket'  # Change this to a globally unique name

response = s3.create_bucket(Bucket=bucket_name)
print(f'Bucket {bucket_name} created successfully!')
```

```
PS D:\sem 2\IS-698\lab 3c> python create_bucket.py
Bucket my-boto3s3bucket created successfully!
PS D:\sem 2\IS-698\lab 3c> aws s3api list-buckets
{
    "Buckets": [
        {
            "Name": "my-boto3s3bucket",
            "CreationDate": "2025-02-19T03:46:05+00:00"
        },
        {
            "Name": "my-s3bucketexample",
            "CreationDate": "2025-02-18T22:22:47+00:00"
        }
    ],
    "Owner": {
        "DisplayName": "vpatel19",
        "ID": "06a2cda887e358d90397ead81b66f6123231cd2b5eccb5f0ea1953837fd6588d"
    },
    "Prefix": null
}

PS D:\sem 2\IS-698\lab 3c> 
```


View Assessment
Lab 3c - Google Docs
S3 buckets | S3 | us-east-1
Policies | IAM | Global
us-east-1.console.aws.amazon.com/s3/buckets?region=us-east-1&bucketType=general
Gmail YouTube Instagram myUMBC: Select an... Blackboard Learn Torrent Search Engi... The Official Home o... Rust | Twitch Drops Fiverr - Freelance Se... mp3offline.org All Bookmarks
Search [Alt+S]
United States (N. Virginia)
Vishvakumar Patel
Amazon S3 > Buckets
Amazon S3
General purpose buckets
Directory buckets
Table buckets
Access Grants
Access Points
Object Lambda Access Points
Multi-Region Access Points
Batch Operations
IAM Access Analyzer for S3
Block Public Access settings for this account
Storage Lens
Dashboards
Storage Lens groups
AWS Organizations settings
Feature spotlight 11
View Storage Lens dashboard
General purpose buckets
Directory buckets
Copy ARN
Empty
Delete
Create bucket
Find buckets by name
CloudShell Feedback
© 2025, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
10:52 PM
2/18/2025

Account snapshot - *updated every 24 hours* All AWS Regions

Storage lens provides visibility into storage usage and activity trends. Metrics don't include directory buckets. Learn more

General purpose buckets (2) Info All AWS Regions

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	my-boto3s3bucket	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 18, 2025, 22:46:05 (UTC-05:00)
○	my-s3bucketexample	US East (N. Virginia) us-east-1	View analyzer for us-east-1	February 18, 2025, 17:22:47 (UTC-05:00)

Part 2: Upload a File to the Bucket

1. Create a Python script named `upload_file.py`:

□

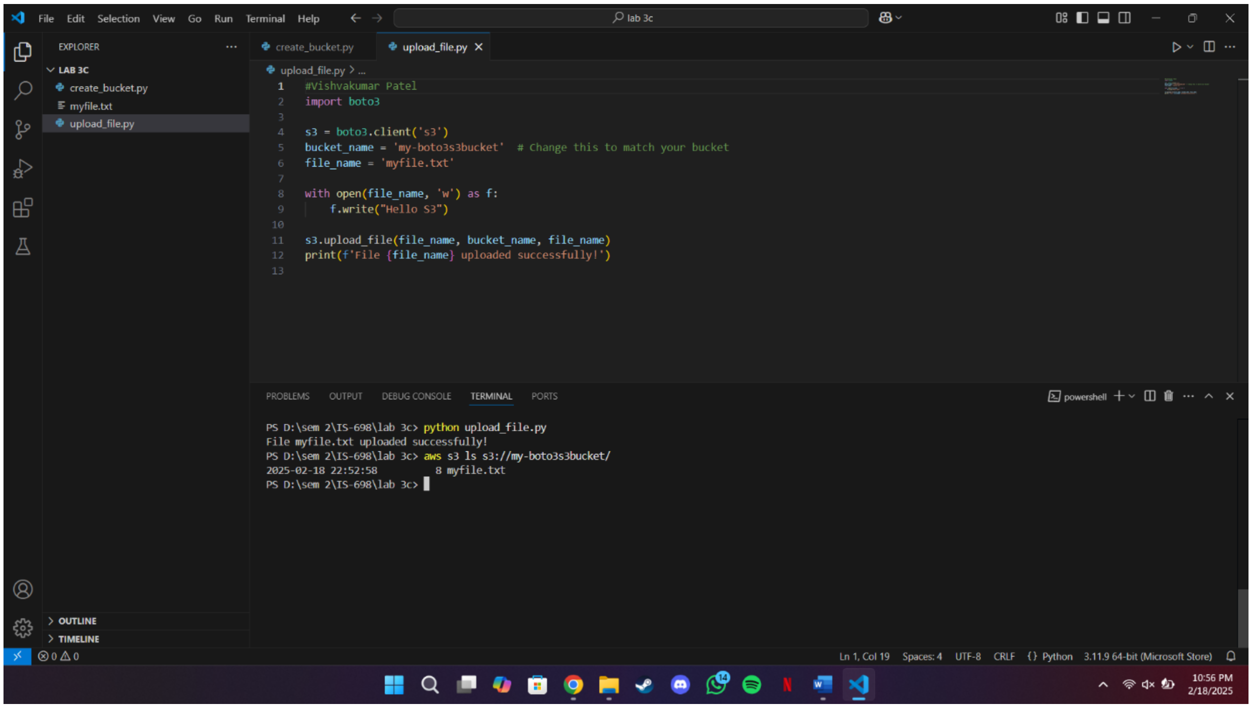
```
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3-s3-bucket-<yourname>'  # Change this to match your bucket
file_name = 'myfile.txt'

with open(file_name, 'w') as f:
    f.write("Hello S3")

s3.upload_file(file_name, bucket_name, file_name)

print(f'File {file_name} uploaded successfully!')
```

2. □Run the script:

```
□python upload_file.py
```

3. □Confirm the upload:

```
□aws s3 ls s3://my-boto3-s3-bucket-<yourname>/
```


File Edit Selection View Go Run Terminal Help
lab 3c
EXPLORER
LAB 3C
create_bucket.py
myfile.txt
upload_file.py
create_bucket.py
upload_file.py
upload_file.py > ...
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell
OUTLINE
TIMELINE
Ln 1, Col 19 Spaces: 4 UTF-8 CRLF {} Python 3.11.9 64-bit (Microsoft Store)
10:56 PM
2/18/2025

```
#Vishvakumar Patel
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3s3bucket'  # Change this to match your bucket
file_name = 'myfile.txt'

with open(file_name, 'w') as f:
    f.write("Hello S3")

s3.upload_file(file_name, bucket_name, file_name)
print(f'File {file_name} uploaded successfully!')
```

```
PS D:\sem 2\IS-698\lab 3c> python upload_file.py
File myfile.txt uploaded successfully!
PS D:\sem 2\IS-698\lab 3c> aws s3 ls s3://my-boto3s3bucket/
2025-02-18 22:52:58          8 myfile.txt
PS D:\sem 2\IS-698\lab 3c>
```

Part 3: Download a File from the Bucket

1. Create a Python script named `download_file.py`:

□

```
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3-s3-bucket-<yourname>'
file_name = 'myfile.txt'
download_name = 'downloaded-file.txt'

s3.download_file(bucket_name, file_name, download_name)

print(f'File {file_name} downloaded successfully as {download_name}!')
```

2. □Run the script:

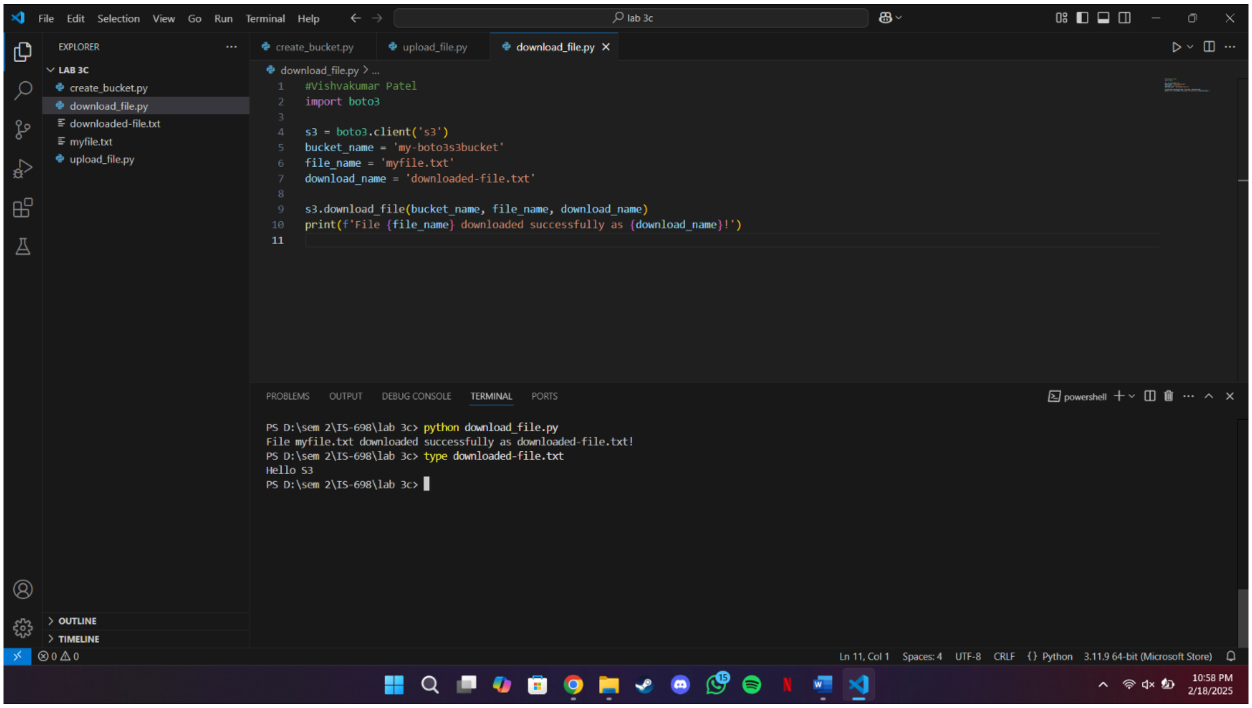
□

```
python download_file.py
```

3. □Verify the file contents:

□

```
cat downloaded-file.txt   # Linux/Mac

type downloaded-file.txt   # Windows
```



```python
#Vishvakumar Patel
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3s3bucket'
file_name = 'myfile.txt'
download_name = 'downloaded-file.txt'

s3.download_file(bucket_name, file_name, download_name)
print(f'File {file_name} downloaded successfully as {download_name}!')
```

```
PS D:\sem 2\IS-698\lab 3c> python download_file.py
File myfile.txt downloaded successfully as downloaded-file.txt!
PS D:\sem 2\IS-698\lab 3c> type downloaded-file.txt
Hello S3
PS D:\sem 2\IS-698\lab 3c>
```

Part 4: Delete a File from the Bucket

1. Create a Python script named `delete_file.py`:

□

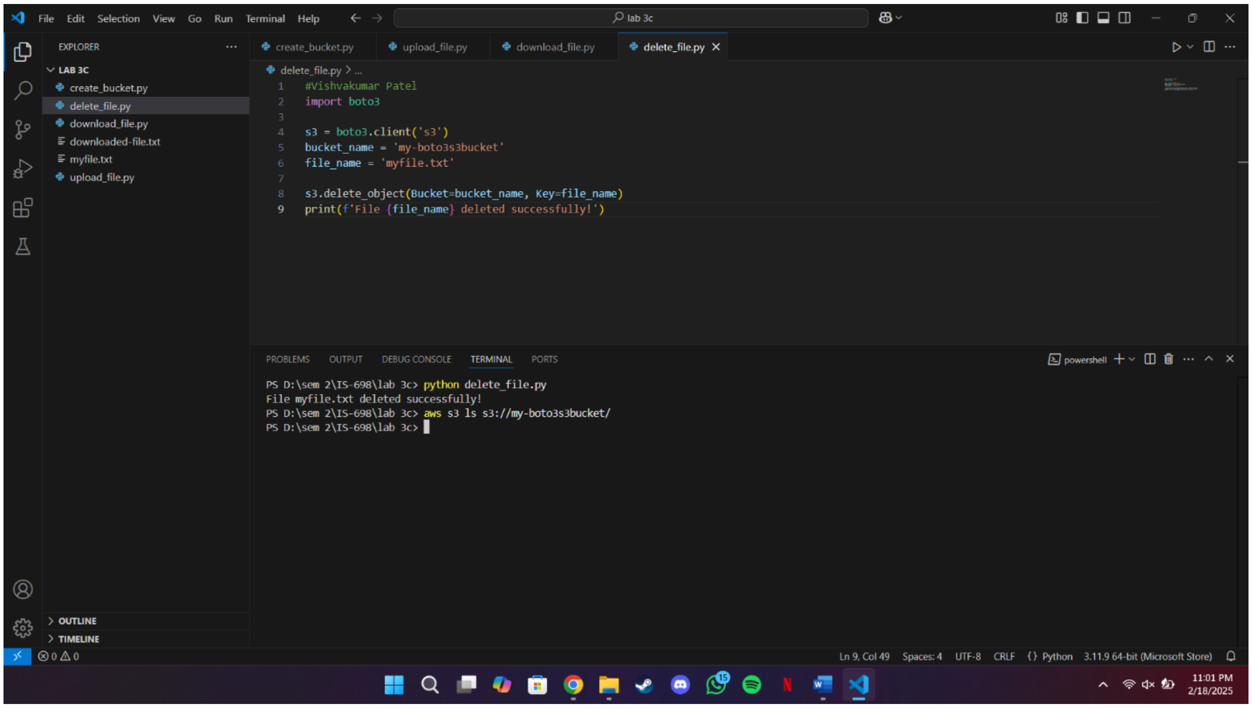
```python
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3-s3-bucket-<yourname>'
file_name = 'myfile.txt'

s3.delete_object(Bucket=bucket_name, Key=file_name)

print(f'File {file_name} deleted successfully!')
```

2. □Run the script:

```
□python delete_file.py
```

3. □Verify the file is deleted:

```
□aws s3 ls s3://my-boto3-s3-bucket-<yourname>/
```



```python
#Vishvakumar Patel
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3s3bucket'
file_name = 'myfile.txt'

s3.delete_object(Bucket=bucket_name, Key=file_name)
print(f'File {file_name} deleted successfully!')
```

```
PS D:\sem 2\IS-698\lab 3c> python delete_file.py
File myfile.txt deleted successfully!
PS D:\sem 2\IS-698\lab 3c> aws s3 ls s3://my-boto3s3bucket/
PS D:\sem 2\IS-698\lab 3c>
```

Part 5: Enable Versioning on the Bucket

1. Create a Python script named `enable_versioning.py`:

□

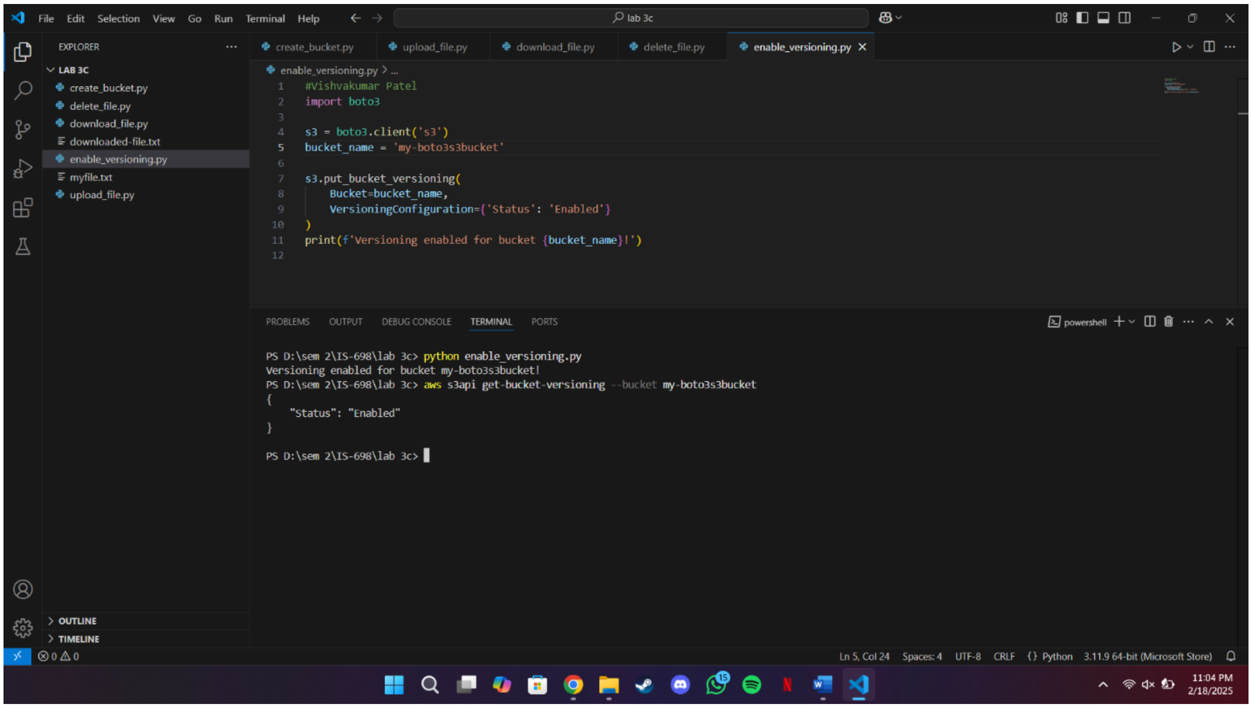
```python
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3-s3-bucket-<yourname>'

s3.put_bucket_versioning(
    Bucket=bucket_name,
    VersioningConfiguration={'Status': 'Enabled'}
)

print(f'Versioning enabled for bucket {bucket_name}!')
```

2. □Run the script:

```
□python enable_versioning.py
```

3. □Confirm versioning:

```
□aws s3api get-bucket-versioning --bucket my-boto3-s3-bucket-<yourname>
```



```python
#Vishvakumar Patel
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3s3bucket'

s3.put_bucket_versioning(
    Bucket=bucket_name,
    VersioningConfiguration={'Status': 'Enabled'}
)
print(f'Versioning enabled for bucket {bucket_name}!')

```

```
PS D:\sem 2\IS-698\lab 3c> python enable_versioning.py
Versioning enabled for bucket my-boto3s3bucket!
PS D:\sem 2\IS-698\lab 3c> aws s3api get-bucket-versioning --bucket my-boto3s3bucket
{
    "Status": "Enabled"
}

PS D:\sem 2\IS-698\lab 3c> 
```

Part 6: Delete the Bucket

1. Create a Python script named `delete_bucket.py`:

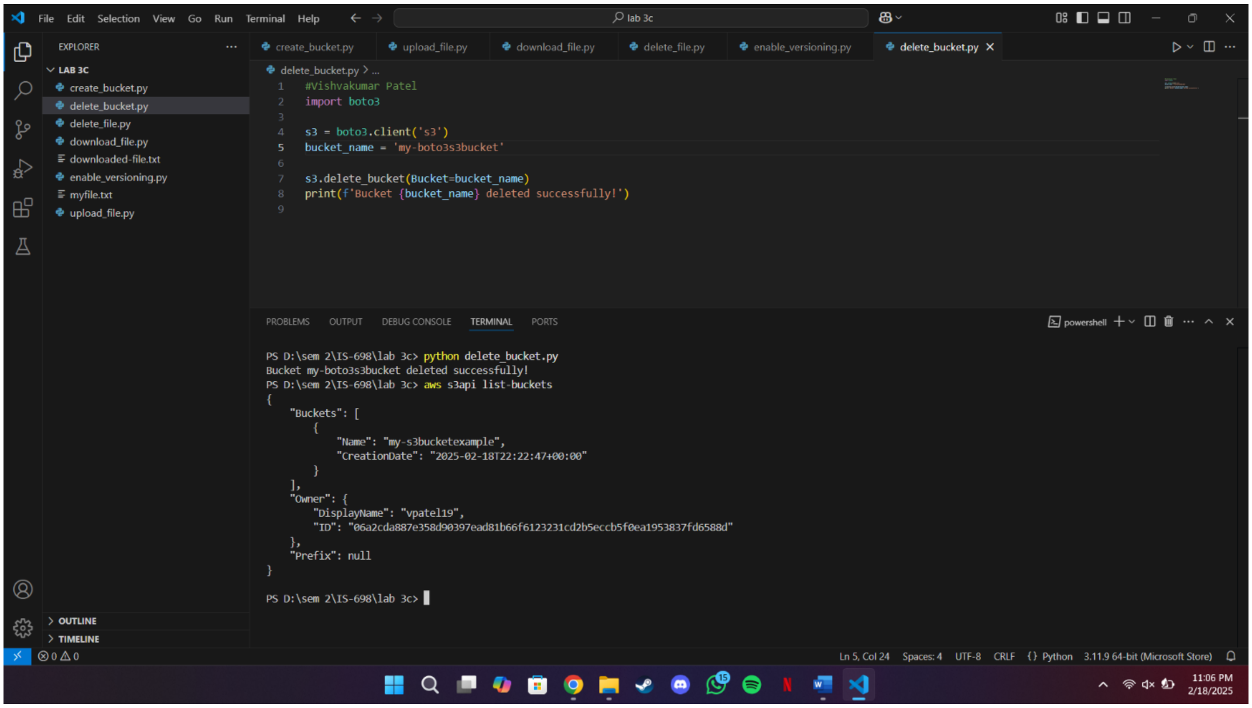
```
□
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3-s3-bucket-<yourname>'

s3.delete_bucket(Bucket=bucket_name)

print(f'Bucket {bucket_name} deleted successfully!')
```

2. □Run the script:

```
□python delete_bucket.py
```

3. □Verify bucket deletion:

```
aws s3api list-buckets
```



```python
#Vishvakumar Patel
import boto3

s3 = boto3.client('s3')
bucket_name = 'my-boto3s3bucket'

s3.delete_bucket(Bucket=bucket_name)
print(f'Bucket {bucket_name} deleted successfully!')
```

```
PS D:\sem 2\IS-698\lab 3c> python delete_bucket.py
Bucket my-boto3s3bucket deleted successfully!
PS D:\sem 2\IS-698\lab 3c> aws s3api list-buckets
{
    "Buckets": [
        {
            "Name": "my-s3bucketexample",
            "CreationDate": "2025-02-18T22:22:47+00:00"
        }
    ],
    "Owner": {
        "DisplayName": "vpatel19",
        "ID": "06a2cda887e358d90397ead81b66f6123231cd2b5eccb5f0ea1953837fd6588d"
    },
    "Prefix": null
}

PS D:\sem 2\IS-698\lab 3c>
```

Deliverables

1. **Screenshots of:**
   - Bucket creation.
   - File upload and retrieval.
   - File deletion.
   - Versioning enabled and file versions listed.
   - Bucket deletion confirmation.
2. **Summary Report:**
   - Describe each step performed.
   - Mention any challenges faced and solutions applied.
3. **GitHub Submission:**
   - Upload all Python scripts to a **GitHub repository**.
   - Ensure **AWS credentials or sensitive information are removed**.
   - Share the repository link with your submission.