ASSIGNMENT PYTHON TAS241

VIKAS K R

—---

I referred to this document for performing the assessment: https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3.html. It helped me with tasks such as making connections, copying, moving, creating, and uploading files.

AWS Account

Step 1: First we need to login with credentials.
Step 2: Create a bucket in any region i created eu region.

```
      REGION = 'eu-north-1'
```

Python part

Step 1: For this assignment i downloaded Flask because it is easy and lightweight by using

```
            Pip3 install Flask
```

Step 2: In order to interact with AWS from Python, you need the Boto3 module.

```
           Pip3 install boto3
```

Step 3: To connect with AWS, we need credentials, which we must obtain from AWS.

```
      s3 = boto3.client(
                       "s3",
                       aws_access_key_id=AWS_ACCESS_KEY,
                       aws_secret_access_key=AWS_SECRET_ACCESS_KEY,
                       region_name=REGION
                       )
```

Step 4: To fetch all the file and folder from the S3 bucket we have to use

Function 'list_objects_v2'

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3/client/list_objects_v2.html

I referred for reference

```
def list_s3_content():
    try:
        response = s3.list_objects_v2(Bucket=S3_BUCKET)
        contents = response.get('Contents', [])
        return contents

    except Exception as e:
        flash(str(e))
        return []

def get_folders(contents):
    folders = set()
    for item in contents:
        key = item['Key']
        if key.endswith('/'):
            folders.add(key)
    # return folders
    return sorted(folders)

@app.route('/')
def index():
    contents = list_s3_content()
    folders = get_folders(contents)
    # return render_template('index.html)
    return render_template('index.html', contents=contents,
folders=folders)
```

Step 5: **create_folder**

It retrieves the folder_name from the form, ensures the folder name ends with a /, and then uses s3.put_object to create the folder in the S3 bucket. Upon success or failure, a message is flashed, and the user is redirected to the homepage.

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3/client/put_object.html

```
@app.route('/create-folder', methods=['POST'])
def create_folder():
   folder_name = request.form['folder_name']
   if folder_name:
       # folder_name = folder_name.rstrip('/')
       folder_name = folder_name.rstrip('/') + '/'
       try:
           s3.put_object(Bucket=S3_BUCKET, Key=folder_name)
           flash('Folder Created')
           # print("Folder Created...")
       except Exception as e:
           flash(str(e))
   return redirect(url_for('index'))
```

Step 6: **delete_object**

It retrieves the key (object name) from the form and uses s3.delete_object to remove the object from the S3 bucket. Upon success or failure, a message is flashed, and the user is redirected to the homepage.

```
@app.route('/delete', methods=['POST'])
def delete_object():
   key = request.form['key']
   try:
       s3.delete_object(Bucket=S3_BUCKET, Key=key)
       # print("Deleted..")
       flash('Deleted Sucessfully')
   except Exception as e:
       flash(str(e))
   return redirect(url_for('index'))
```

Step 7: **move_copy_file**

It retrieves the source (src) and destination (dest) from the form. Depending on the selected action (move or copy), it either copies the file using s3.copy_object or moves it by copying and then deleting the source file. Success or failure messages are flashed, and the user is redirected to the homepage.

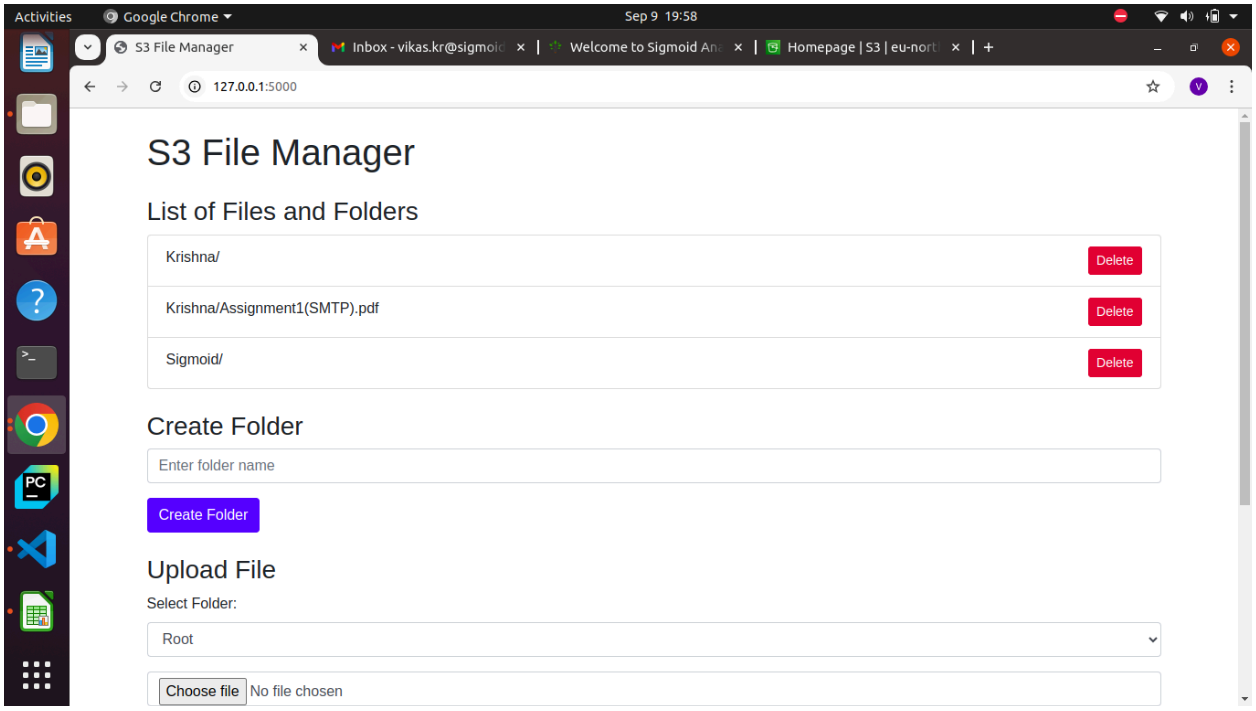
```
@app.route('/move-copy', methods=['POST'])
def move_copy_file():
   src = request.form['src']
   dest = request.form['dest']
   action = request.form['action']

   try:
       if not src or not dest:
           flash('Source and Destination keys must be provided')
           return redirect(url_for('index'))
       s3.copy_object(Bucket=S3_BUCKET, CopySource={'Bucket': S3_BUCKET,
'Key': src}, Key=dest)
       if action == 'move':
           # print(action)
           s3.delete_object(Bucket=S3_BUCKET, Key=src)

       flash(f'File {action} successfully from {src} to {dest}')
   except Exception as e:
       flash(f"Error: {str(e)}")
   return redirect(url_for('index'))
```

https://boto3.amazonaws.com/v1/documentation/api/latest/reference/services/s3/client/copy_object.html

OUTPUT:



S3 File Manager

List of Files and Folders

Krishna/	Delete
Krishna/Assignment1(SMTP).pdf	Delete
Sigmoid/	Delete

Create Folder

Enter folder name

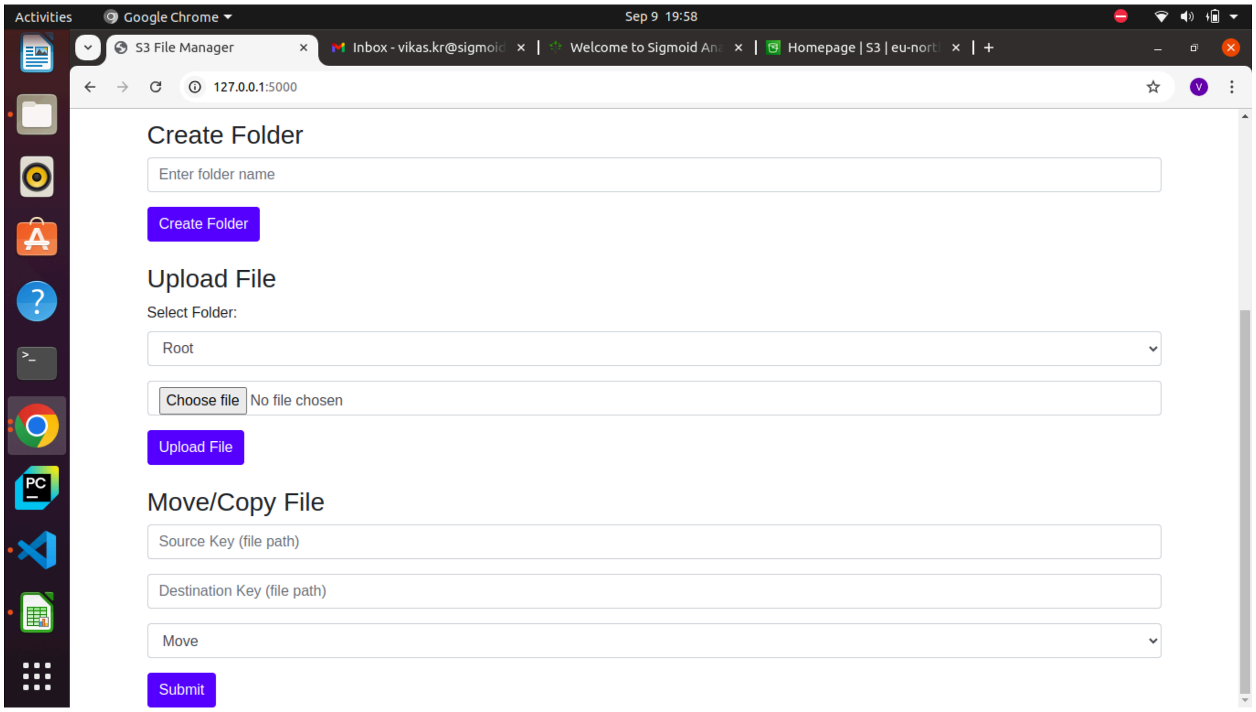
Create Folder

Upload File

Select Folder:

Root

Choose file No file chosen



Create Folder

Enter folder name

Create Folder

Upload File

Select Folder:

Root

Choose file No file chosen

Upload File

Move/Copy File

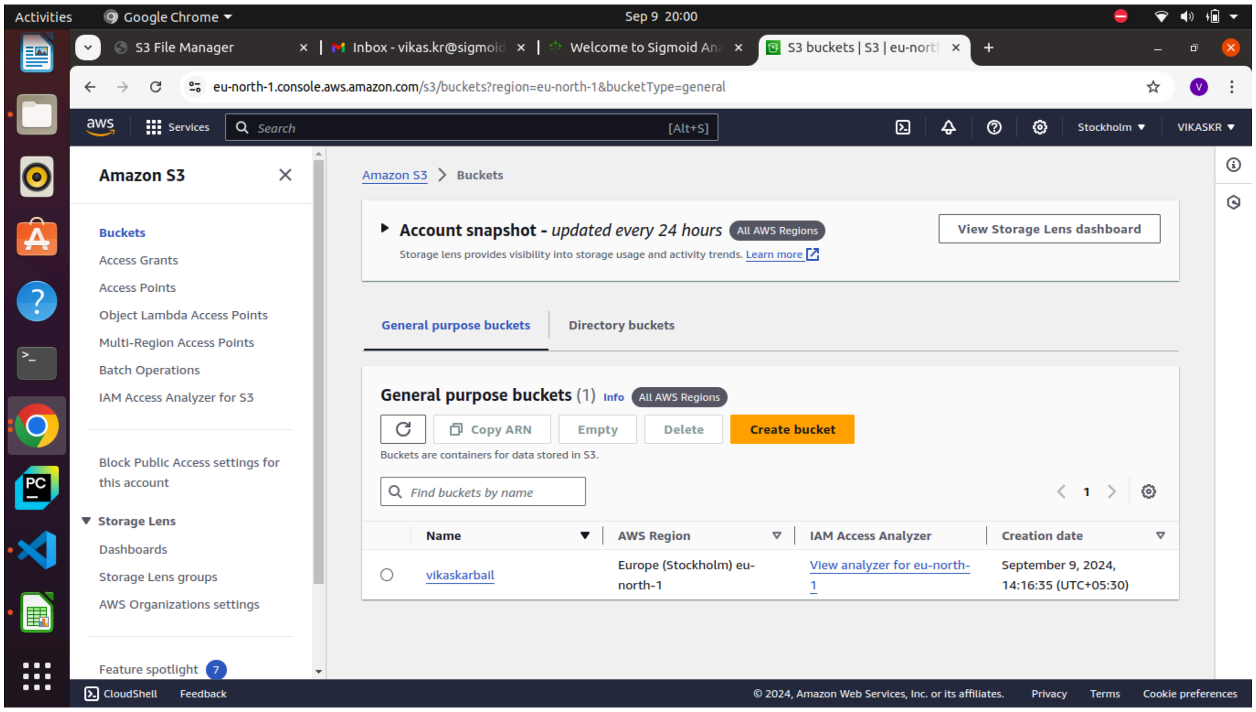
Source Key (file path)

Destination Key (file path)

Move

Submit

AWS Bucket



Storage lens provides visibility into storage usage and activity trends. Learn more

General purpose buckets (1) All AWS Regions

Buckets are containers for data stored in S3.

	Name	AWS Region	IAM Access Analyzer	Creation date
○	vikaskarbail	Europe (Stockholm) eu-north-1	View analyzer for eu-north-1	September 9, 2024, 14:16:35 (UTC+05:30)