Python Assignment

TAS269 Harthik S A

Q. Implement s3 file manager using any python web framework(flask/django/...etc).

functions :
1. List content of s3.
2. Create/Delete folder + bucket .
3. Upload files to s3 + delete file from s3.
4. Copy/Move file withing s3.
Note:
1. Make sure your code is readable
2. Make sure your app is working properly
3. Need basic UI from which we can access app

Answer:

Create AWS Account

> Enter personal details
> Enter user detail
> Enter payment details
> Enter purpose

IAM Configuration

> Open aws
> Search 'Users'
> Create User
Provide user access to the AWS Management Console - optional // select this checkbox
> Are you providing console access to a person?
> Select Custom Password
> Add Password
> Permission Options :
> Select: Attach policies directly

> In policies options : (select AmazonS3FullAcess)
>click create user (user created)

To create credentials for accessing an AWS S3 bucket from a third-party application or an application running outside of the AWS environment, follow these steps:

1. **Go to the AWS Management Console**.
2. **Navigate to IAM (Identity and Access Management)**.
3. **Select "Users"** from the IAM dashboard.
4. **Choose the specific user** for whom you want to create access credentials.
5. **Click on "Create access key"**.
6. **Select "Application running outside AWS"**.
7. **Download the .CSV file** which contains the credentials details.

These credentials include an Access Key ID and Secret Access Key, which are needed to programmatically access AWS S3 from your application.

To set up your project for interacting with AWS S3 using Flask, follow these steps:

1. **Open VS Code** and **open your project directory**.

Install the necessary dependencies by running:
bash
Copy code
```
pip install flask boto3 flask-wtf
```

2. **Set up your AWS credentials**:
    - **Install the AWS CLI**.

Configure the AWS CLI to store your credentials locally by running:
```
>aws configure
```
Example configuration:

```
$ aws configure
AWS Access Key ID [None]: YOUR_ACCESS_KEY
AWS Secret Access Key [None]: YOUR_SECRET_KEY
```

```
Default region name [None]: YOUR_REGION (e.g., us-east-1, which is
globally active)
Default output format [None]: json
```

- ○

3. **Create an `app.py` file** in your project directory. This file will contain the programming logic to connect with the AWS S3 bucket and enable the upload and retrieval of files and folders from Amazon S3.

```
from flask import Flask, request, redirect, render_template, url_for

import boto3

from botocore.exceptions import NoCredentialsError, ClientError



app = Flask(__name__)

app.secret_key = efXUmgAxfHgIM2tW1YzzBmkf2SbqhDhWR5rHayVV '  # Replace with your actual secret key



# Initialize S3 client with your AWS credentials

s3 = boto3.client(

    's3',

    aws_access_key_id=' AKIA356SJQ2M3GXXPR6U ',  # Replace with your actual access key id

     aws_secret_access_key='efXUmgAxfHgIM2tW1YzzBmkf2SbqhDhWR5rHayVV ',

    region_name='us-east-1'  # Replace with your region if needed

)



# BUCKET_NAME = 'harthik-first-bucket-99777'



@app.route('/')
```

```
def list_bucket_contents():
    try:
        response = s3.list_objects_v2(Bucket=BUCKET_NAME)
        contents = response.get('Contents', [])
    except NoCredentialsError:
        return "Credentials not available", 403
    except ClientError as e:
        return str(e), 400
    return render_template('index.html', contents=contents)


@app.route('/upload', methods=['POST'])
def upload_file():
    if 'file' not in request.files or not request.form['folder_name']:
        return redirect('/')
    file = request.files['file']
    folder_name = request.form['folder_name']
    if file.filename == '':
        return redirect('/')
    try:
        s3.upload_fileobj(file, BUCKET_NAME, folder_name + '/' + file.filename)
    except NoCredentialsError:
        return "Credentials not available", 403
    except ClientError as e:
        return str(e), 400
    return redirect('/')
```

```
@app.route('/delete_file/<file_key>', methods=['POST'])
def delete_file(file_key):
    try:
        s3.delete_object(Bucket=BUCKET_NAME, Key=file_key)
    except ClientError as e:
        return str(e), 400
    return redirect('/')


@app.route('/copy_file', methods=['POST'])
def copy_file():
    src_key = request.form['src_key']
    dest_key = request.form['dest_key']
    try:
        copy_source = {'Bucket': BUCKET_NAME, 'Key': src_key}
        s3.copy_object(CopySource=copy_source, Bucket=BUCKET_NAME, Key=dest_key)
    except ClientError as e:
        return str(e), 400
    return redirect('/')


@app.route('/move_file', methods=['POST'])
def move_file():
    src_key = request.form['src_key']
    dest_key = request.form['dest_key']
    try:
        copy_source = {'Bucket': BUCKET_NAME, 'Key': src_key}
        s3.copy_object(CopySource=copy_source, Bucket=BUCKET_NAME, Key=dest_key)
        s3.delete_object(Bucket=BUCKET_NAME, Key=src_key)
    except ClientError as e:
        return str(e), 400
    return redirect('/')

@app.route('/create_folder', methods=['POST'])
def create_folder():
    folder_name = request.form['folder_name']
    try:
        s3.put_object(Bucket=BUCKET_NAME, Key=folder_name + '/')
    except ClientError as e:
        return str(e), 400
    return redirect('/')

@app.route('/delete_folder', methods=['POST'])
def delete_folder():
    folder_name = request.form['folder_name']
    try:
        # Delete all objects with the folder prefix
        response = s3.list_objects_v2(Bucket=BUCKET_NAME, Prefix=folder_name + '/')
        for obj in response.get('Contents', []):
            s3.delete_object(Bucket=BUCKET_NAME, Key=obj['Key'])
        # Delete the folder itself
        s3.delete_object(Bucket=BUCKET_NAME, Key=folder_name + '/')
    except ClientError as e:
        return str(e), 400
    return redirect('/')


if __name__ == '__main__':
    app.run(debug=True)
```

In your project directory, **create a directory named `templates`**. Inside this directory, **create a file named `index.html`**. This file will serve as the dashboard where users can:

- **Create a folder** in the AWS S3 bucket.
- **Delete a folder** from the AWS S3 bucket.
- **Upload a file** to the AWS S3 bucket by selecting a folder name and creating the file.
- **Delete a file** from the AWS S3 bucket.
- **Copy a file** to a destination within the AWS S3 bucket.
- **Move a file** to a destination within the AWS S3 bucket.

```html
<!DOCTYPE html>

<html>

<head>

   <title>S3 File Manager</title>

   <style>

       body {

           font-family: Arial, sans-serif;

           background-color: #f4f4f9;

           color: #333;
```

```css
        margin: 0;

        padding: 0;

    }

    h1, h2 {

        color: #0073e6;

        text-align: center;

    }

    h1 {

        margin-top: 20px;

    }

    form {

        margin: 10px 0;

    }

    ul {

        list-style-type: none;

        padding: 0;

    }

    li {

        background-color: #ffffff;

        border: 1px solid #ddd;

        border-radius: 5px;

        margin: 5px 0;

        padding: 10px;

        display: flex;
    align-items: center;
    justify-content: space-between;
}
button {
    background-color: #0073e6;
    color: #ffffff;
    border: none;
    border-radius: 5px;
    padding: 5px 10px;
    cursor: pointer;
    margin: 0 5px;
}
button:hover {
    background-color: #005bb5;
}
input[type="text"] {
    padding: 5px;
    border: 1px solid #ddd;
    border-radius: 5px;
    margin-right: 5px;
}
input[type="file"] {
    margin: 5px 0;
}
```

```
       form input[type="text"], form input[type="file"] {

           width: 200px;

       }

       form {

           display: flex;

           flex-direction: column;

           align-items: center;

       }

   </style>

</head>

<body>

   <h1>List of Files in Bucket</h1>

   <ul>

       {% for item in contents %}

       <li>

           {{ item.Key }}

           <div>

               <form action="{{ url_for('delete_file', file_key=item.Key) }}"
method="post" style="display:inline;">

                   <button type="submit">Delete File</button>

               </form>

               <form action="{{ url_for('copy_file') }}" method="post"
style="display:inline;">

                   <input type="hidden" name="src_key" value="{{ item.Key
}}">
```

```
                   <input type="text" name="dest_key"
placeholder="Destination Key" required>

                   <button type="submit">Copy File</button>

               </form>

               <form action="{{ url_for('move_file') }}" method="post"
style="display:inline;">

                   <input type="hidden" name="src_key" value="{{ item.Key
}}">

                   <input type="text" name="dest_key"
placeholder="Destination Key" required>

                   <button type="submit">Move File</button>

               </form>

           </div>

       </li>

       {% endfor %}

   </ul>



   <h2>Upload a File</h2>

   <form action="/upload" method="post" enctype="multipart/form-data">

       <input type="file" name="file" required>

       <input type="text" name="folder_name" placeholder="Folder Name"
required>

       <button type="submit">Upload</button>

   </form>



   <h2>Create a Folder</h2>

   <form action="/create_folder" method="post">
```

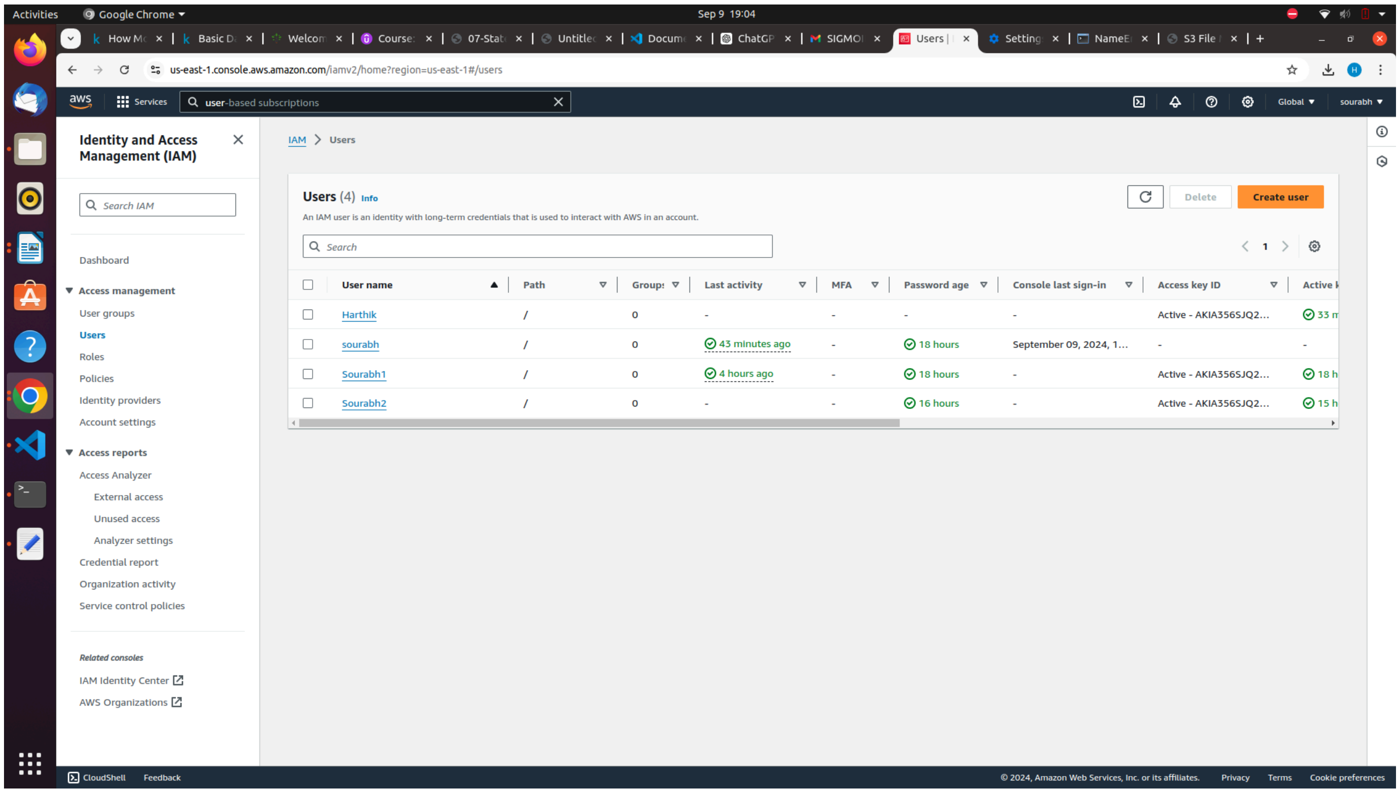
```
       <input type="text" name="folder_name" placeholder="Folder Name"
required>

       <button type="submit">Create Folder</button>

   </form>



   <h2>Delete a Folder</h2>

   <form action="/delete_folder" method="post">

       <input type="text" name="folder_name" placeholder="Folder Name"
required>

       <button type="submit">Delete Folder</button>

   </form>

</body>

</html>
```

Run File:

Run the Flask Application:

python3 app.py

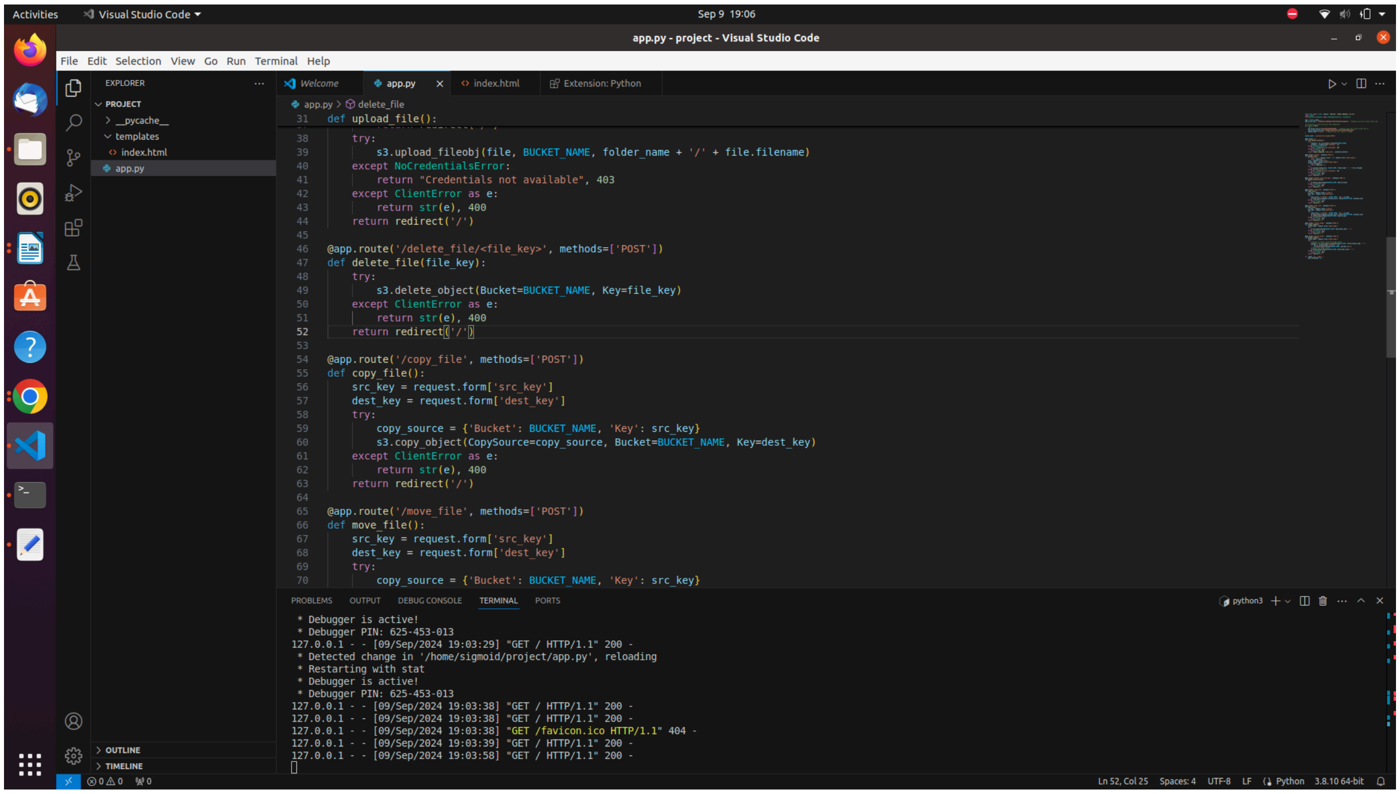
Access the Application: Open your browser and navigate to http://127.0.0.1:5000/.



Users (4) Info

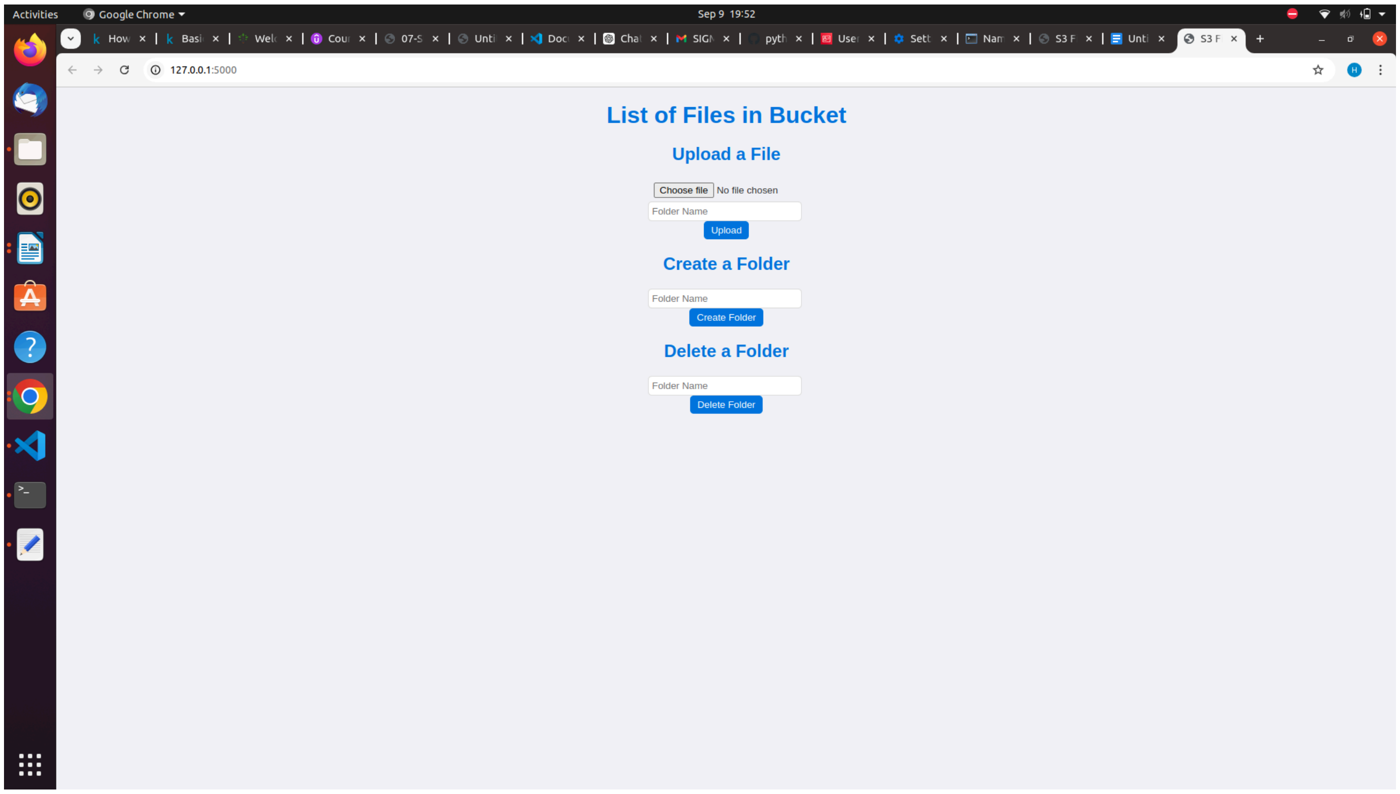
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Path	Group:	Last activity	MFA	Password age	Console last sign-in	Access key ID	Active k
Harthik	/	0	-	-	-	-	Active - AKIA356SJQ2...	✓ 33 m
sourabh	/	0	✓ 43 minutes ago	-	✓ 18 hours	September 09, 2024, 1...	-	-
Sourabh1	/	0	✓ 4 hours ago	-	✓ 18 hours	-	Active - AKIA356SJQ2...	✓ 18 h
Sourabh2	/	0	-	-	✓ 16 hours	-	Active - AKIA356SJQ2...	✓ 15 h



```python
def upload_file():
    try:
        s3.upload_fileobj(file, BUCKET_NAME, folder_name + '/' + file.filename)
    except NoCredentialsError:
        return "Credentials not available", 403
    except ClientError as e:
        return str(e), 400
    return redirect('/')

@app.route('/delete_file/<file_key>', methods=['POST'])
def delete_file(file_key):
    try:
        s3.delete_object(Bucket=BUCKET_NAME, Key=file_key)
    except ClientError as e:
        return str(e), 400
    return redirect('/')

@app.route('/copy_file', methods=['POST'])
def copy_file():
    src_key = request.form['src_key']
    dest_key = request.form['dest_key']
    try:
        copy_source = {'Bucket': BUCKET_NAME, 'Key': src_key}
        s3.copy_object(CopySource=copy_source, Bucket=BUCKET_NAME, Key=dest_key)
    except ClientError as e:
        return str(e), 400
    return redirect('/')

@app.route('/move_file', methods=['POST'])
def move_file():
    src_key = request.form['src_key']
    dest_key = request.form['dest_key']
    try:
        copy_source = {'Bucket': BUCKET_NAME, 'Key': src_key}
```

```
 * Debugger is active!
 * Debugger PIN: 625-453-013
127.0.0.1 - - [09/Sep/2024 19:03:29] "GET / HTTP/1.1" 200 -
 * Detected change in '/home/sigmoid/project/app.py', reloading
 * Restarting with stat
 * Debugger is active!
 * Debugger PIN: 625-453-013
127.0.0.1 - - [09/Sep/2024 19:03:38] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Sep/2024 19:03:38] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Sep/2024 19:03:38] "GET /favicon.ico HTTP/1.1" 404 -
127.0.0.1 - - [09/Sep/2024 19:03:39] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [09/Sep/2024 19:03:58] "GET / HTTP/1.1" 200 -
```



127.0.0.1:5000

List of Files in Bucket

Upload a File

Choose file No file chosen

Folder Name

Upload

Create a Folder

Folder Name

Create Folder

Delete a Folder

Folder Name

Delete Folder