#working

import os

from google.cloud import storage

from datetime import timedelta

import uuid

# Use service account key

# os.environ["GOOGLE\_APPLICATION\_CREDENTIALS"] = "deepanshu-storage.json"

os.environ["GOOGLE\_APPLICATION\_CREDENTIALS"] = "C:/Users/user121/Downloads/deepanshu-storage.json"

def create\_html\_file():

    html\_content = "<h1>Hello from Deepanshu</h1>"

    file\_path = "index.html"

    with open(file\_path, "w") as f:

        f.write(html\_content)

    print(f"Created local file: {file\_path}")

    return file\_path

def create\_private\_bucket(bucket\_name):

    storage\_client = storage.Client()

    bucket = storage\_client.bucket(bucket\_name)

    new\_bucket = storage\_client.create\_bucket(bucket, location="US")

    new\_bucket.iam\_configuration.uniform\_bucket\_level\_access\_enabled = True

    new\_bucket.patch()

    print(f"Private bucket {bucket.name} created.")

    return new\_bucket

def upload\_file(bucket, source\_file\_path, destination\_blob\_name):

    blob = bucket.blob(destination\_blob\_name)

    blob.upload\_from\_filename(source\_file\_path)

    print(f"File {source\_file\_path} uploaded as {destination\_blob\_name}.")

def generate\_signed\_url(bucket\_name, blob\_name, expiration\_minutes=15):

    storage\_client = storage.Client()

    bucket = storage\_client.bucket(bucket\_name)

    blob = bucket.blob(blob\_name)

    url = blob.generate\_signed\_url(

        version="v4",

        expiration=timedelta(minutes=expiration\_minutes),

        method="GET"

    )

    print(f"\nAccess it securely using this signed URL (valid for {expiration\_minutes} minutes):\n{url}")

    return url

if \_\_name\_\_ == "\_\_main\_\_":

    bucket\_name = f"private-bucket-{uuid.uuid4().hex[:6]}"

    blob\_name = "index.html"

    file\_path = create\_html\_file()

    bucket = create\_private\_bucket(bucket\_name)

    upload\_file(bucket, file\_path, blob\_name)

    generate\_signed\_url(bucket.name, blob\_name)

#private bucket

Python storage.py

**1. Create a Service Account with Access to Storage**

1. Go to: Google Cloud Console - IAM & Admin > Service Accounts
2. Click **Create Service Account**
3. Name it like signed-url-service
4. Assign role: Storage Admin
5. Click **Done**
6. Click the new service account → **Keys** tab → **Add Key → JSON**
7. Save the JSON file as service-account-key.json (put it in your script directory)

To set the GOOGLE\_APPLICATION\_CREDENTIALS environment variable for your service account key file located in the Downloads folder, follow these steps:

1. **Locate the correct path**: Your deepanshu-storage.json file is located in C:\Users\user121\Downloads\.
2. **Set the environment variable**:

In Git Bash, run the following command:

bash

CopyEdit

export GOOGLE\_APPLICATION\_CREDENTIALS="/c/Users/user121/Downloads/deepanshu-storage.json"

This uses the Unix-style path format, where C:\ is converted to /c/, and the rest of the path follows the same pattern.

1. **Verify that the environment variable is set**:

You can check if the environment variable is set correctly by running:

bash

CopyEdit

echo $GOOGLE\_APPLICATION\_CREDENTIALS

This should output the path to your JSON key file.

1. **Run your script**:

Now, you should be able to run your Python script, and it will authenticate using the service account specified in deepanshu-storage.json.

 **Verify the file path**: Make sure the file deepanshu-storage.json is in the correct directory and that the path is correct. If the file is in the Downloads folder, use the full path.

 **Update the environment variable for correct path**: Instead of setting the environment variable using a relative path, use the full path to the file in the script like this:

python

CopyEdit

os.environ["GOOGLE\_APPLICATION\_CREDENTIALS"] = "C:/Users/user121/Downloads/deepanshu-storage.json"

 **Alternatively, set the path via the terminal**: If you are running the script from the terminal, you can export the environment variable directly like this:

In Git Bash:

bash

CopyEdit

export GOOGLE\_APPLICATION\_CREDENTIALS="/c/Users/user121/Downloads/deepanshu-storage.json"

user121@DESKTOP-JFJ21I5 MINGW64 /d/agcp

$ python private\_bucket.py

Created local file: index.html

Private bucket private-bucket-44de2e created.

File index.html uploaded as index.html.

Access it securely using this signed URL (valid for 15 minutes):

https://storage.googleapis.com/private-bucket-44de2e/index.html?X-Goog-Algorithm=GOOG4-RSA-SHA256&X-Goog-Credential=signer-url-service%40manifest-zephyr-457115-v2.iam.gserviceaccount.com%2F202

50420%2Fauto%2Fstorage%2Fgoog4\_request&X-Goog-Date=20250420T054632Z&X-Goog-Expires=900&X-Goog-SignedHeaders=host&X-Goog-Signature=25b93e62ab1417f8eb16af2495048a46e213fa5207e03e1c405080face4ec9

7b29b44aa9c04401dbe2c555831037d368ae6202a5507b71a3546d4f628c6c0e3c5f2c75690ba2de53f9bc10cde14b042713cdbaecb3e22b2c66d989006fcbb1d23ebd7a94c99ed200961b785dec0db71400e770af837d0ded01e37717262110

95d6d2b40753df99f18c81f852302a6eb83842a03f4f48fb1c721b74df0b2115c880f0d2913c166d6d90070a2af06882bf9f987c06565c9c8631e9eee3c71b06a15f67e5c6a1074a79706641f155f36cef6b30dffd8c9f62785b8891df6ec099

3d1be04ac5dd1d9999aec592309bebc7a27a86f19fc8da20a43019dd3d5bb03551