Cloud Computing Assignment I

SUBMITTED TO: Dr. Anil Negi

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

PATIALA-147001, PUNJAB

SUBMITTED BY:

NAME: RIYA **ROLL NUMBER: 101803256**

On April 10, 2021 **GROUP-3CO12**

STATEMENT: Write a python program to create a backup of any folder from your PC to the cloud/google drive.
Backing up the files from source to destination after a fixed time period (e.g. 24 hours). Also maintain the logs of a backup in a separate file (name it log.txt) in the following format: dd/mm/yyyy hh:mm:ss backup_status file_location/ issue (in case of backup failure)

Setting Up:

Create a new Google Cloud Platform (GCP) project

1) Open the Google Cloud Console.

2) Next to "Google Cloud Platform," click the Down arrow arrow_drop_down . A dialog listing current projects appears.

3) Click New Project and add all necessary information to create a new project.

Enable a Google Drive API

1. On your project dashboard in the top-left corner, click Menu > APIs & Services.
2. Click Enable APIs and Services. The Welcome to API Library page appears.
3. In the search field, enter Google *Drive API*.
4. Click Enable.

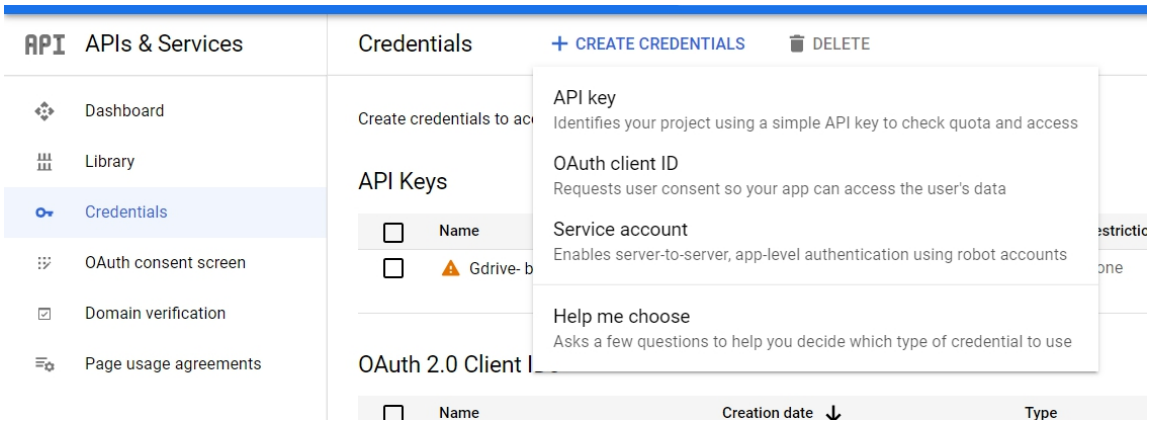
Configure the OAuth consent screen

1. Click APIs & Services > Credentials. The credential page for your project appears.
2. Click Configure Consent Screen.. Click Create. A second "OAuth consent screen" screen appears.
3. Fill out the form:
4. Click Back to Dashboard.

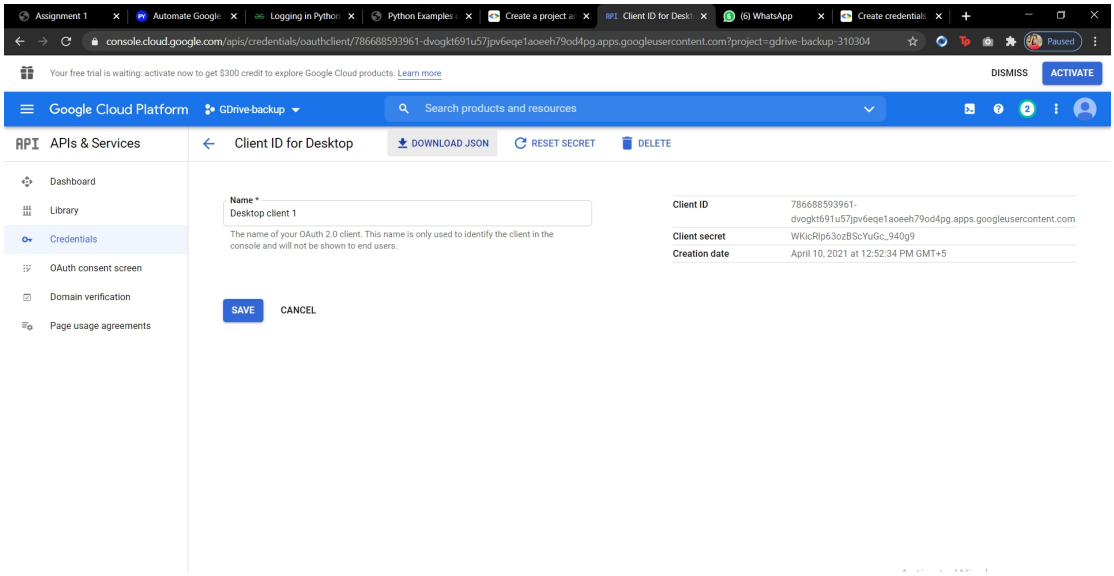
Create a credential

In the left-hand navigation, click Credentials. The "Credentials" page appears.

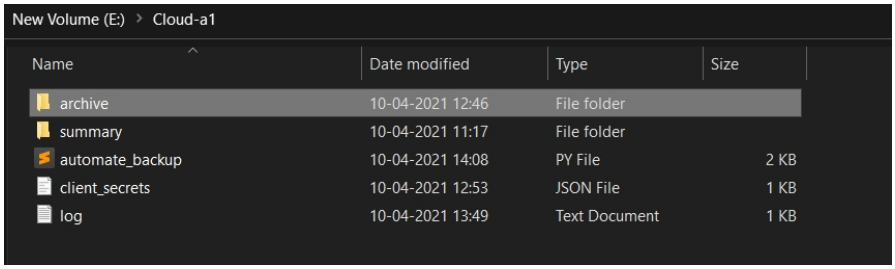
Click Create Credentials and select OAuth client ID. Fill the required fields of form.



Click on name of OAuth 2.0 Client ID.



With this, we are done setting up our google drive API and now to complete this process click on *DOWNLOAD JSON* button to download the client secrets JSON file.
Rename this file to client_secrets.json and move it to the working directory of this project.



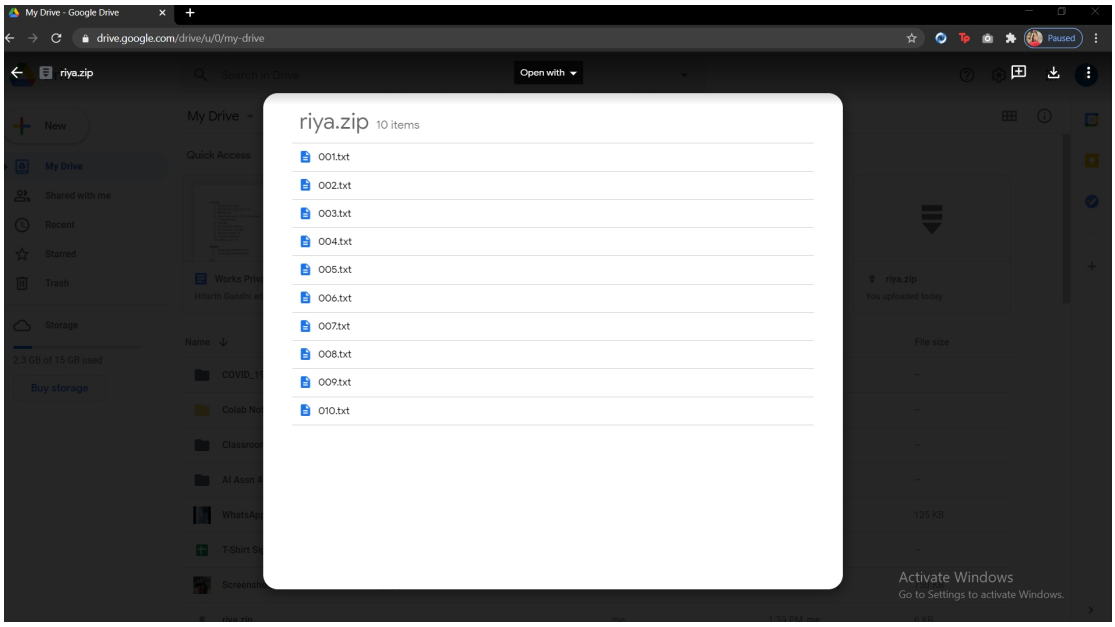
Coding:

```
from datetime import datetime
import shutil
import sys, os
from pydrive.auth import GoogleAuth
from pydrive.drive import GoogleDrive
import schedule
import logging

def create_zip(path, file_name):
    # use shutil to create a zip file
    try:
        shutil.make_archive(f"archive/{file_name}", 'zip', path)
        return True
    except FileNotFoundError as e:
            return False

def google_auth():
    gauth = GoogleAuth()
    # use local default browser for authentication
    gauth.LocalWebserverAuth()
    drive = GoogleDrive(gauth)
    return gauth, drive

def upload_backup(drive, path, file_name):

    # create a google drive file instance
    f = drive.CreateFile({'title': file_name})

    # set the path to zip file
    f.SetContentFile(os.path.join(path, file_name))

    # start logging
    logging.basicConfig(filename="log.log",format='%(asctime)s %(message)s')
    logger=logging.getLogger()
    logger.setLevel(logging.CRITICAL)
    try:
            f.Upload()
            logger.critical("backup_successful "+os.path.join(os.getcwd(),path, file_name))
    except Exception as e:
            logger.critical("backup_failed "+e.__class__)

    # f is set to none because of a vulnerability found in PyDrive
    f = None

def controller():
    # folder path to backup
    path = "./summary"
```

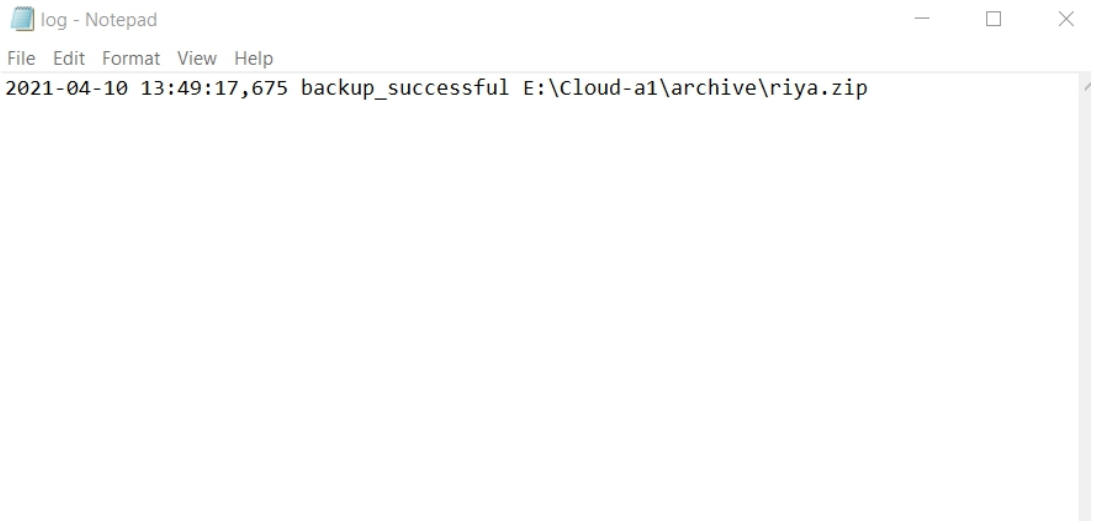
```
    now = datetime.now()

    # if zip creation fails then abort execution
    if  not create_zip(path, "riya"):
            sys.exit(0)
    auth, drive = google_auth()
    upload_backup(drive, r"archive", "riya"+'.zip')

if __name__=="__main__":
    # set 12:00 am as time to trigger controller check for pending tasks and execute if
any
    # controller()
    schedule.every().day.at("00:00").do(controller)
    while True:
        schedule.run_pending()
```

EXPLANATION

1. Install *PyDrive* library and schedule library.
2. Import all necessary libraries as shown in code..
3. First, we will write a function create_zip() that creates a zip file of our backup folder and save it with a custom name. All zips will be store in *archive* folder, this also helps in maintaining a local backup. If zip creation fails it will return False.
4. Secondly, we will write a function google_auth() to start authentication with google drive API. As this function is called you default browser will open and ask for user permissions to access your drive contents.
5. Next, it is time to write a function that will take a file name and path of the zip file and upload it to google drive. Upload_backup(). **Logger** object is created to keep logs of whether backup was successful or not.
6. And finish it of by setting up a scheduler that will call controller() everyday at 12:00 am. For testing purposes, you can remove the scheduler part and just run controller to see if backup is taking place
7. Run code (python automate_backup.py) in my case.
8. You will be asked for google authentication. Enter details to make it work.



Backup created at myDrive



Logs.txt