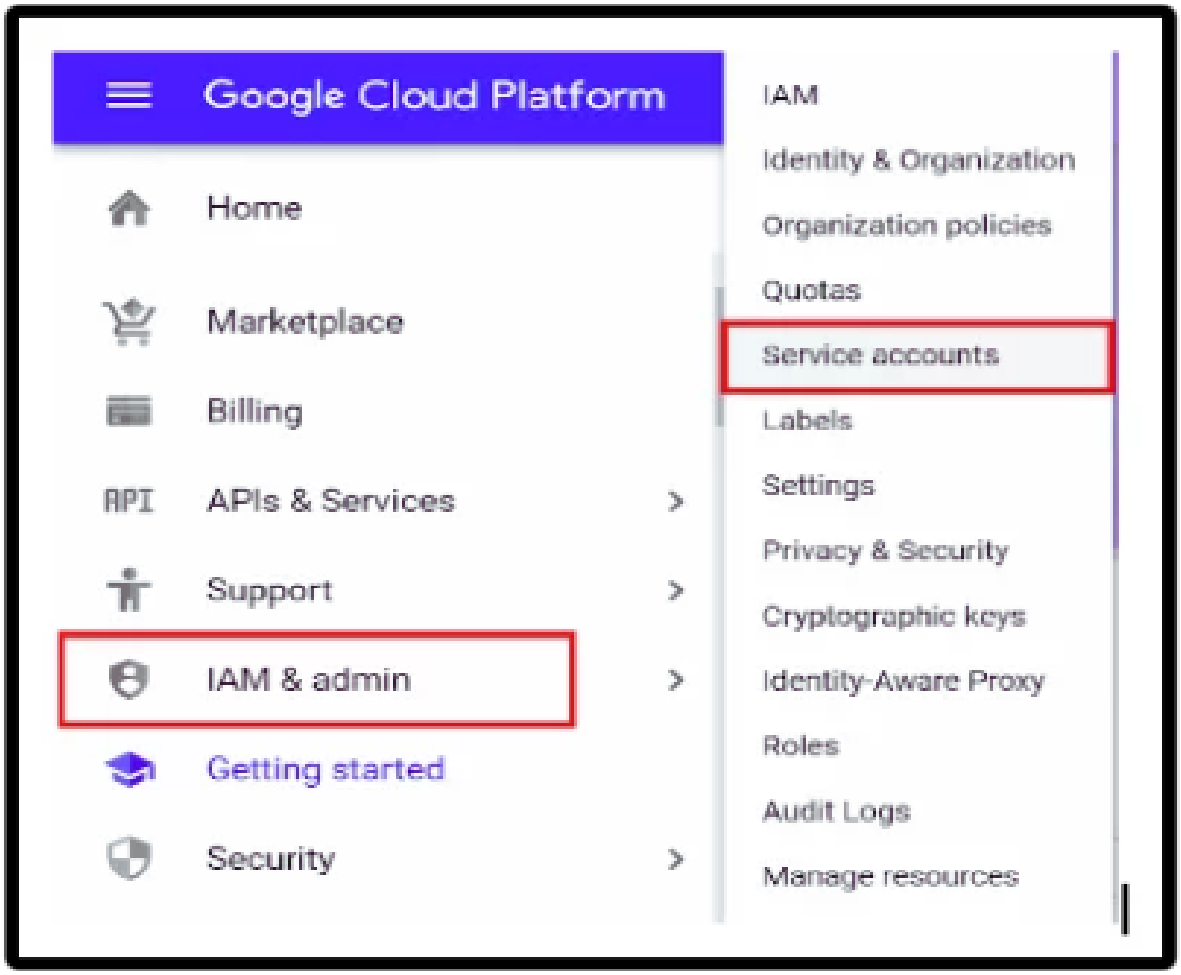
WSO2 Identity Server

A GUIDE TO DOWNLOAD FILES FROM GOOGLE DRIVE WITH OAUTH 2.0 USING A SERVICE ACCOUNT KEY



This guide will walk you through the process of downloading a file from Google Drive using a service account key.

A service account is a Google account that represents your application instead of an individual user. It allows your application to access Google APIs on its own behalf.

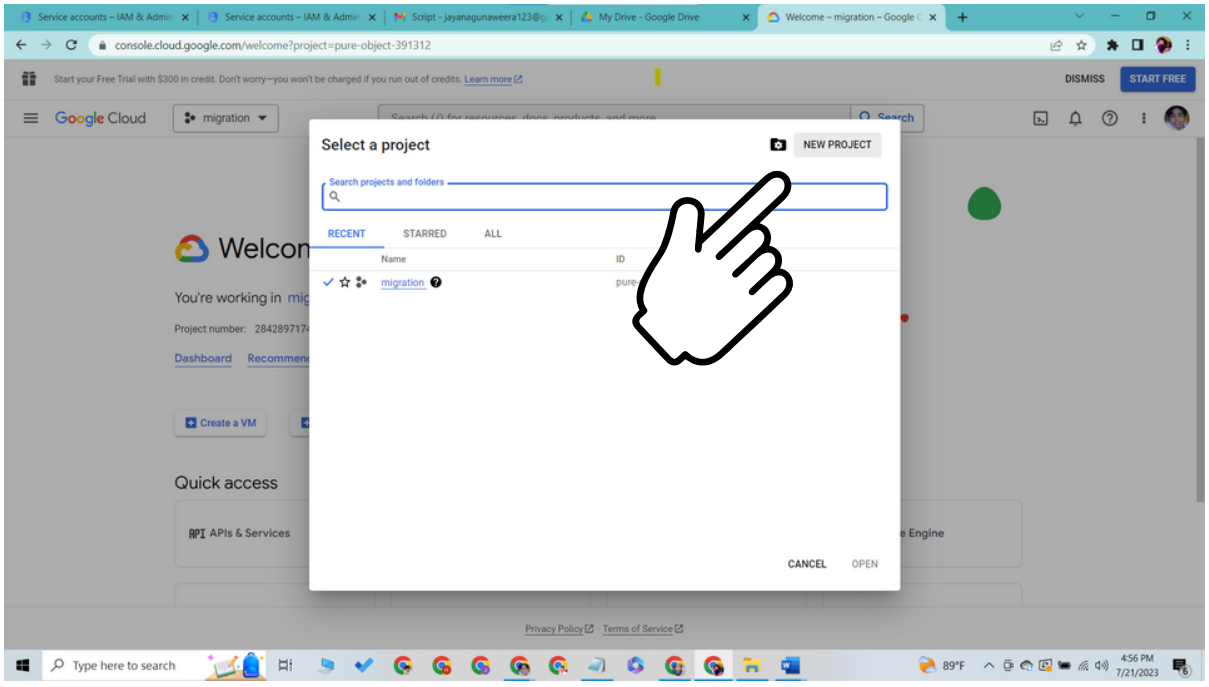
Please follow these steps to achieve the file download:

STEP 01

Set up a Service Account and Enable the Google Drive API

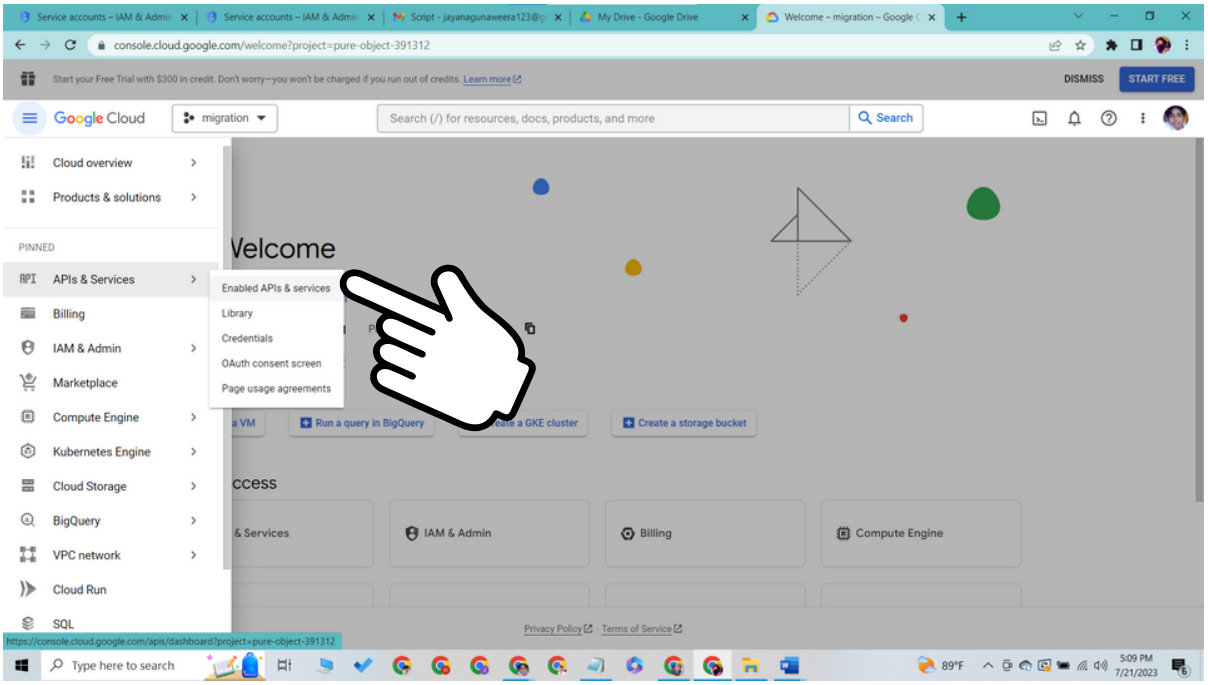
1.Go to the Google Cloud Console.

2.Create a new project or select an existing one.



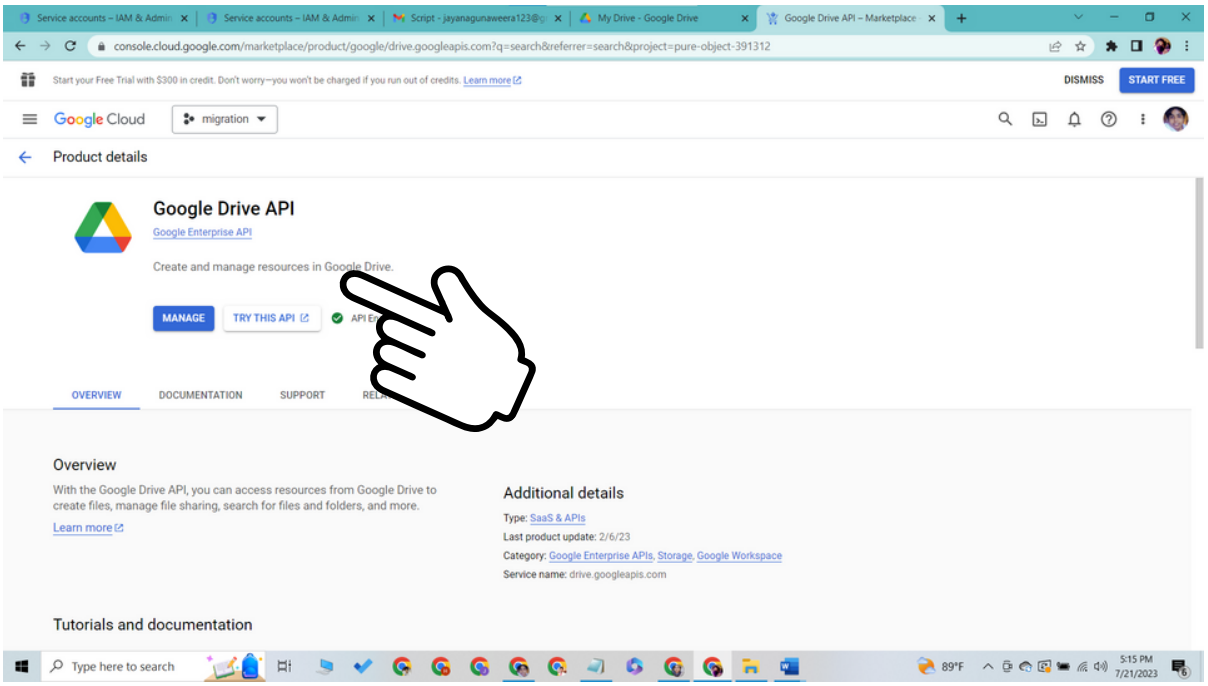
3.In the left sidebar, click on "APIs & Services" > "Dashboard."

4.Click on the "+ ENABLE APIS AND SERVICES" button.



5. Search for "Google Drive API" and select it.

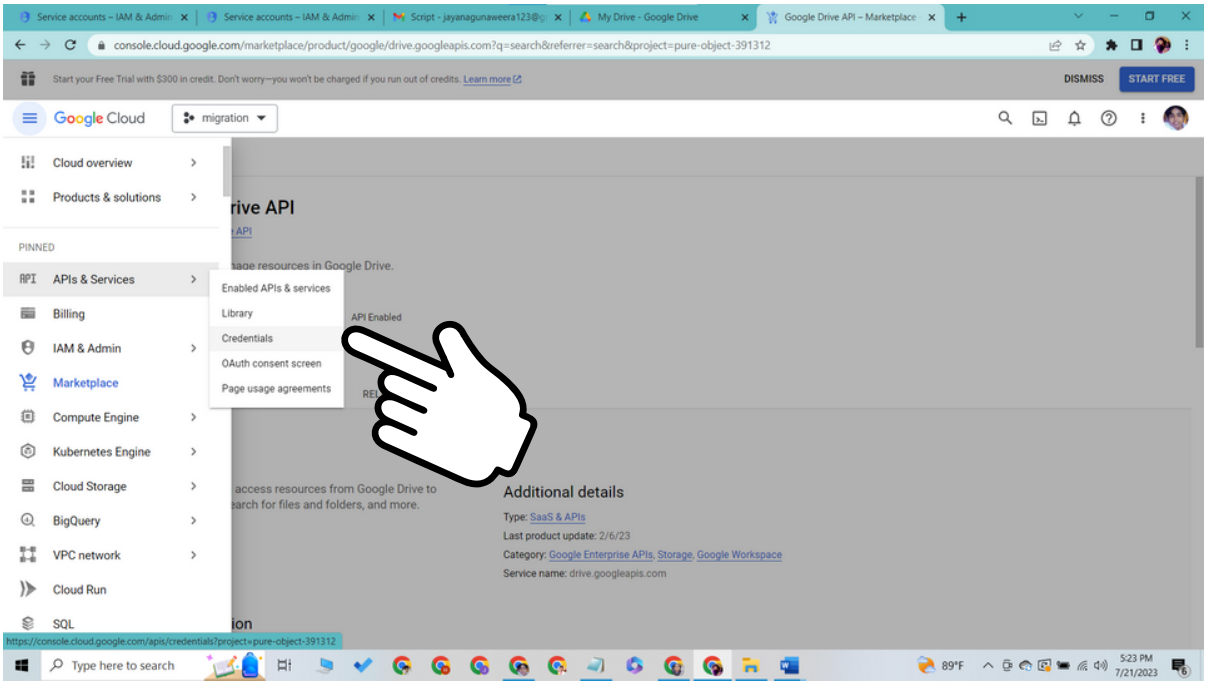
6.Click on the "ENABLE" button to enable the API for your project.



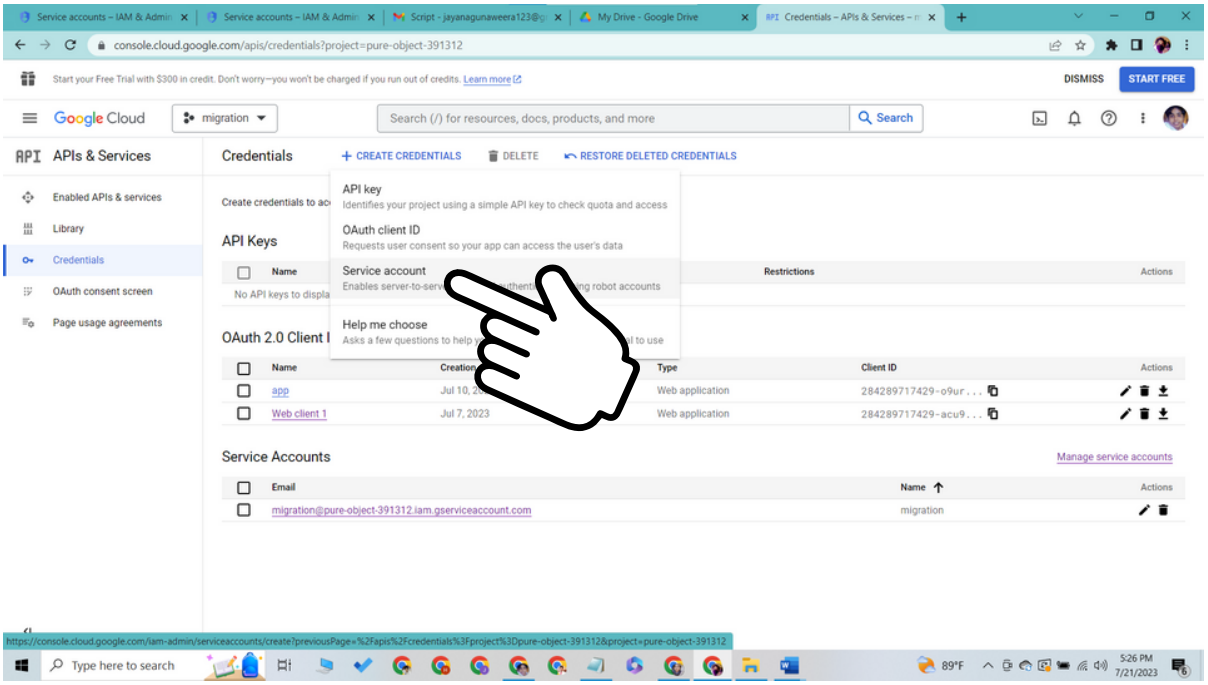
STEP 02

Create a Service Account and Obtain the Key

1.In the left sidebar, click on "APIs & Services" > "Credentials."

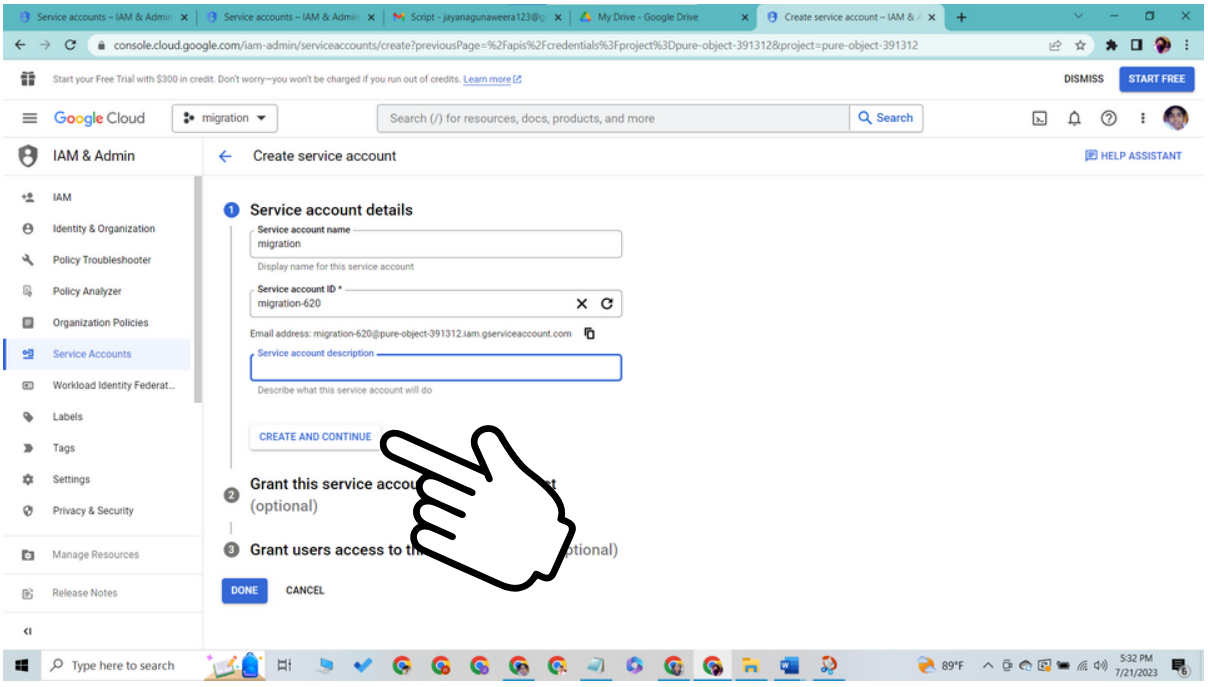


2.Click on the "+ CREATE CREDENTIALS" >"Service account."

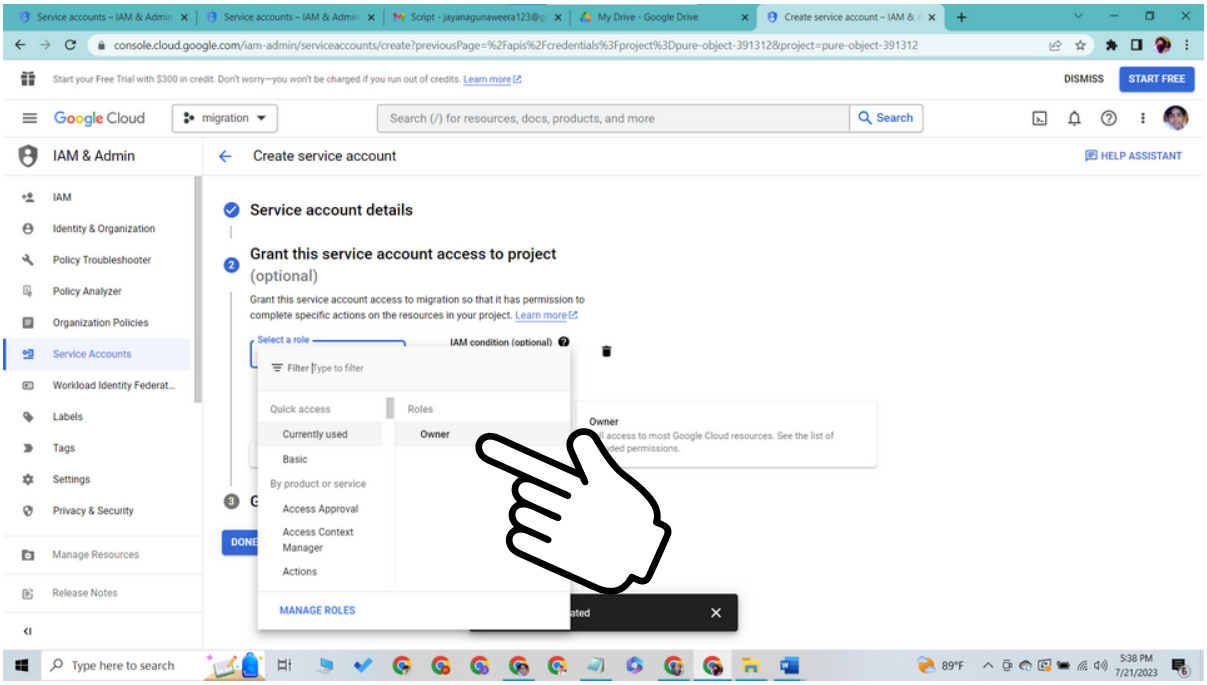


3.Enter a name for the service account,thus a service account ID will be automatically created.

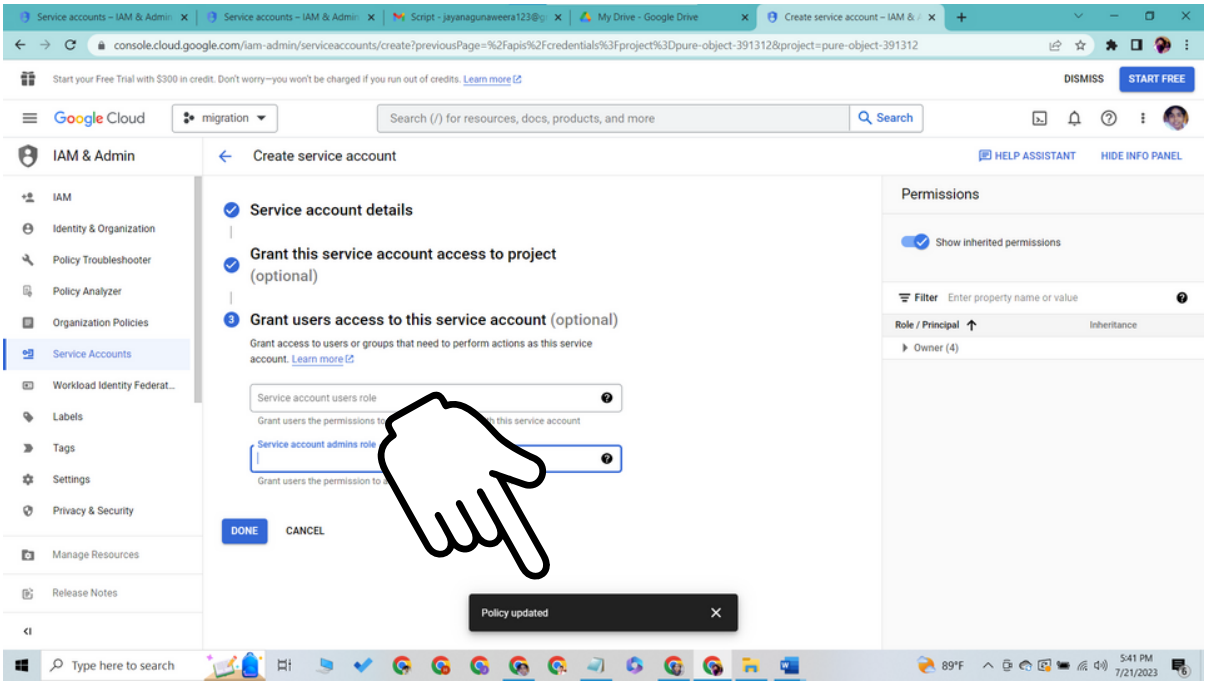
4.Click create and continue.



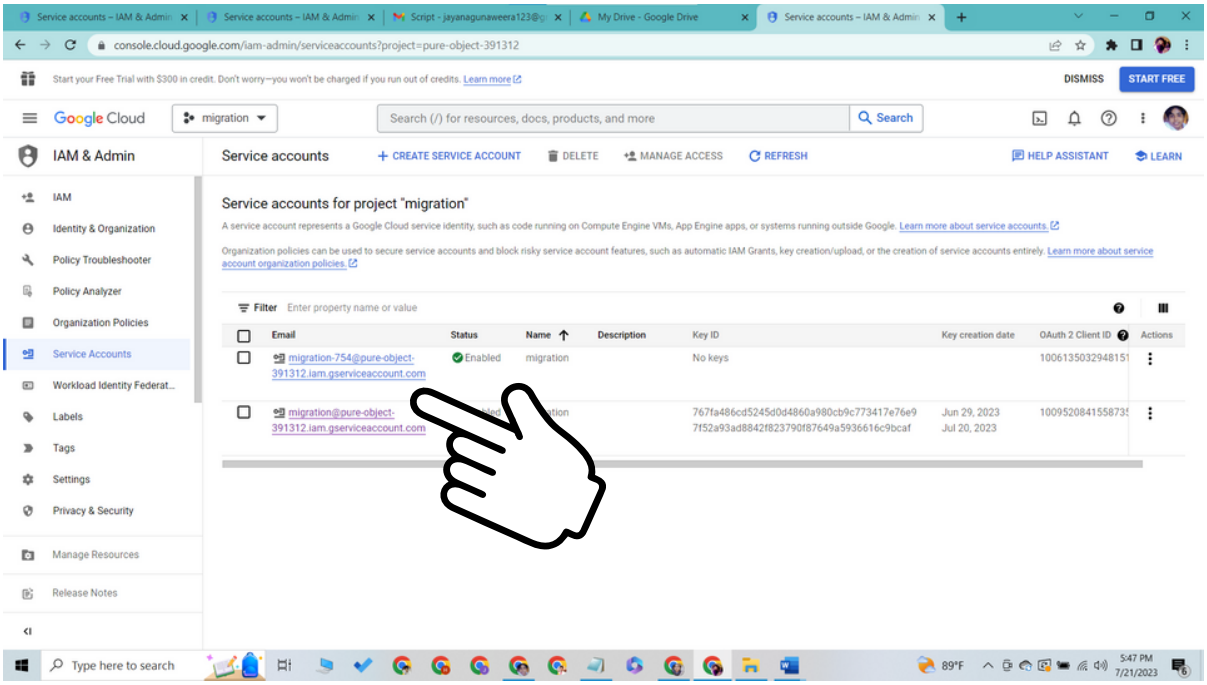
5.Grant this service account access to project as owner



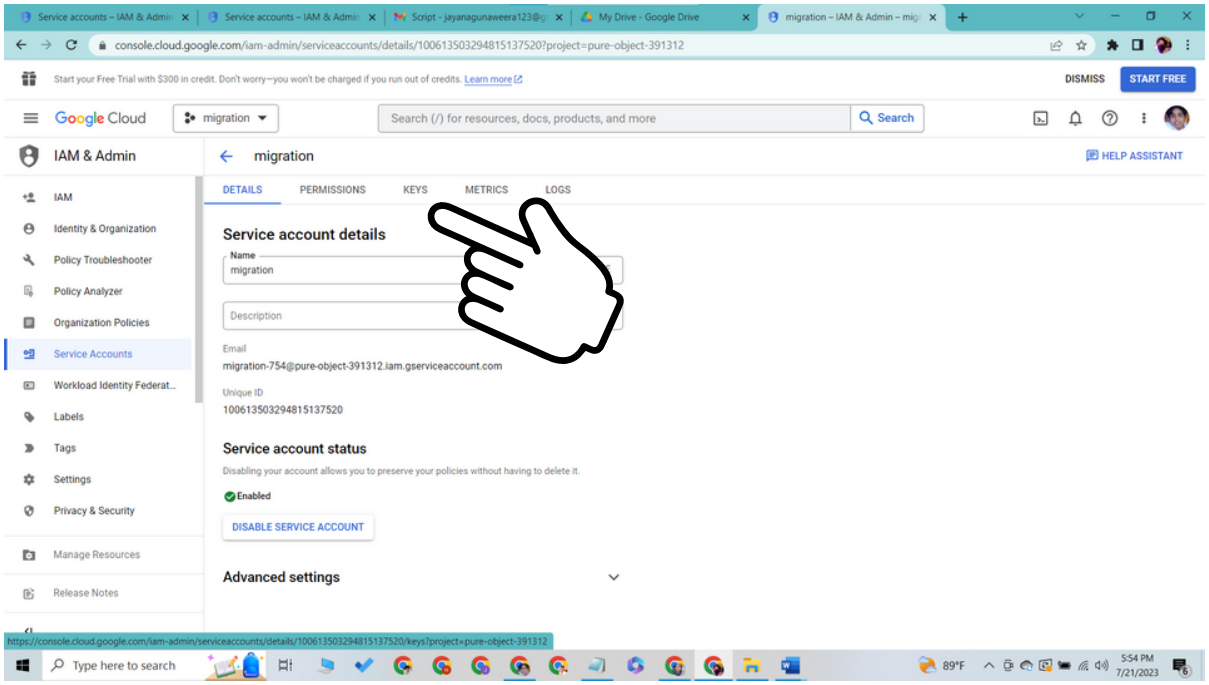
6.Click continue.You will see a message showing policy has been updated.Then click done.



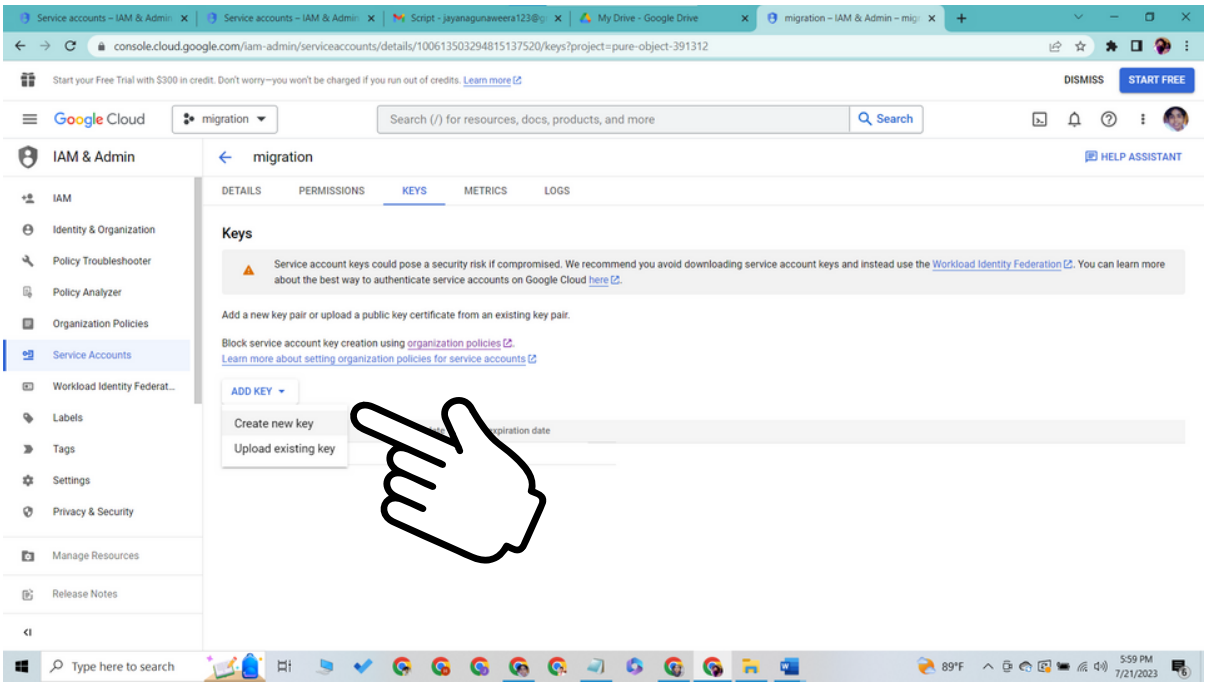
7.In the left sidebar, click on IAM and Admin and next select "Service account."You will be able to see the newly created email address for the service account.



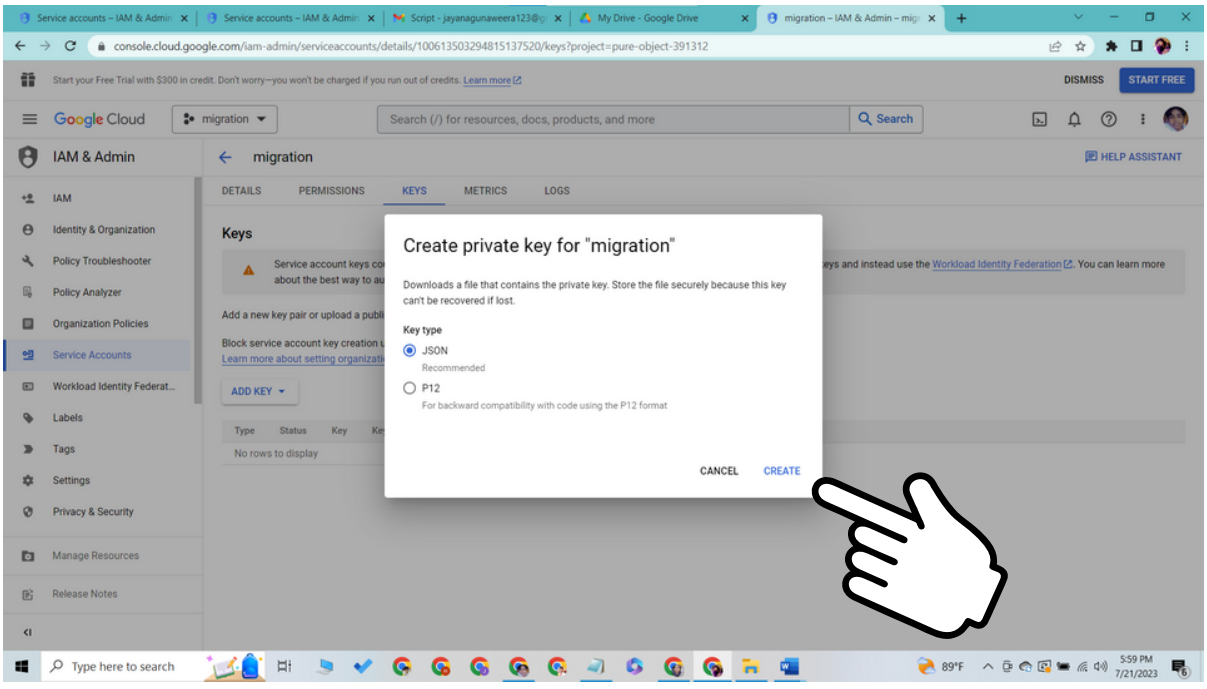
8.Click on the newly created email address and select "Keys" tab.



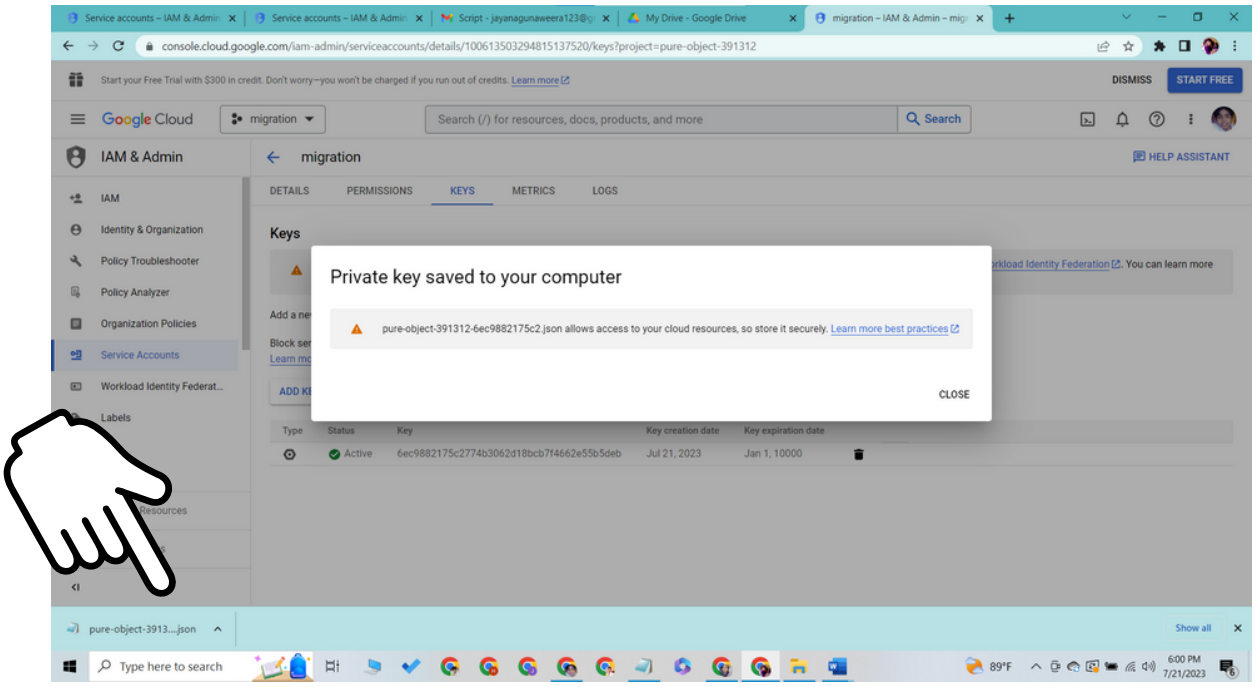
9.Click "add key" > "create a new key"



10.Select "JSON" as the key type and click on the "CREATE" button



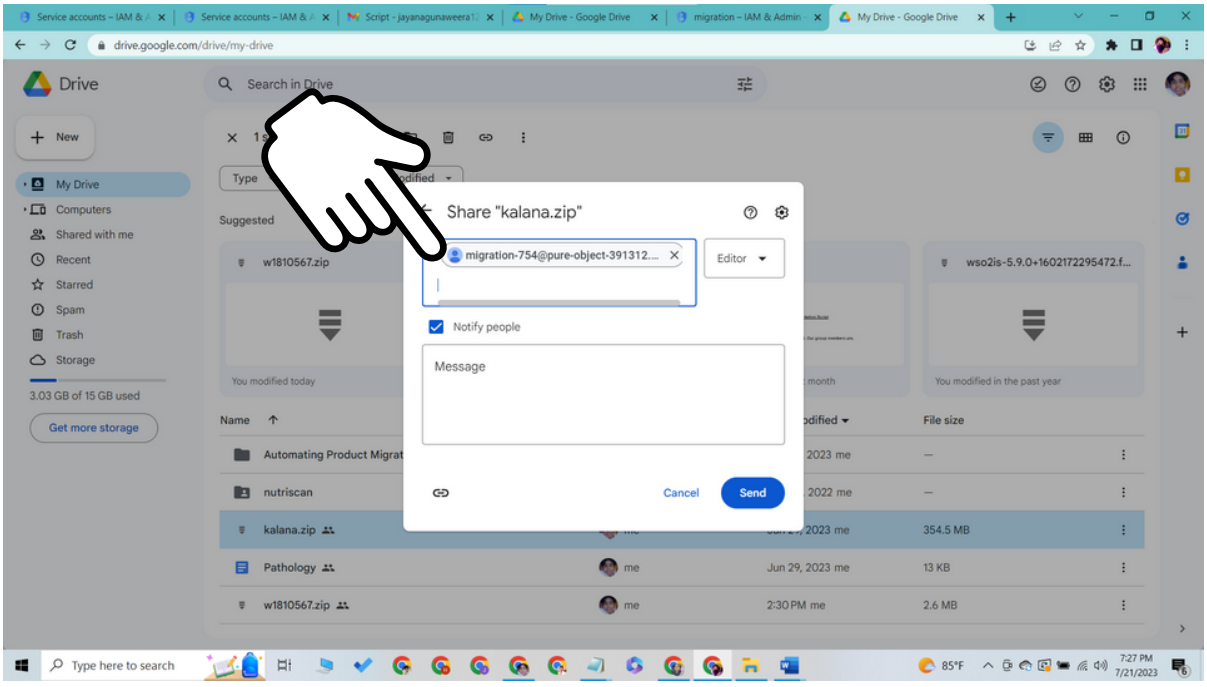
11.The JSON key file will be downloaded to your local machine. Keep this file secure, as it provides access to your Google Drive.



STEP 03

Grant Access to the Service Account in Google Drive

1. Open your Google Drive.
2. Locate the file you want to download and right-click on it.
3. Click on "Share."
4. In the sharing settings, enter the email address associated with your Service Account (can be found in the JSON key file).



STEP 04

Download the File using the Service Account Key

1. Now, let's download the file using the Service Account Key from your application code or any programming language of your choice (e.g., Python):

link:https://docs.google.com/document/d/1iR5u9Tjglvd_WErd6IzrmsPlMICeAjUVCOlp3nbZz6I/edit?usp=sharing

Replace path/to/your/service-account-key.json with the actual path to your Service Account key JSON file, and FILE_ID with the ID of the file you want to download (can be found in the file URL).

How to get the file ID of a file?

To get the file ID of a file in Google Drive, follow these steps:

1. Open Google Drive: Go to https://drive.google.com/ and sign in with your Google account.
2. Locate the File: Navigate to the file for which you want to get the file ID.
3. Get File Info: Right-click on the file and select "Get link" from the context menu.
4. Obtain File ID: In the link that appears, the file ID is located after the "/file/d/" part and before any other parameters or slashes. For example, if the link is "https://drive.google.com/file/d/abcdefg1234567/view", the file ID is "abcdefg1234567".

That's it! Now you have successfully obtained the file ID of the file in Google Drive. You can use this file ID in your scripts or applications to interact with the specific file using the Google Drive API.

LET'S BREAK DOWN THE STEPS OF THE SHELL SCRIPT

01

Generating a JSON Web Token (JWT)

The script proceeds to generate a JSON Web Token (JWT). The JWT will contain specific claims such as issuer (service account email), scope, audience, expiration time, and issued at time. These claims are combined into a request body and signed using the private key associated with the service account.

02

Requesting an Access Token

With the signed JWT ready, the script constructs a POST request to the Google OAuth 2.0 endpoint to request an access token. The request includes the signed JWT in the request body. This request is made using the **curl** command to communicate with the OAuth server.

03

Extracting the Access Token

The response from the OAuth server is captured and stored in the **token_response** variable. The script then extracts the access token from the **token_response** using the **jq** command and stores it in the **access_token** variable.

04

Downloading the File from Google Drive

The script specifies the Google Drive file's URL that needs to be downloaded. It includes the **access_token** in the request headers as an Authorization Bearer token. The **curl** command is used again, this time to download the file from Google Drive.

05

Handling the Response

The script checks if the response contains any error message by looking for the presence of **"error":** in the response. If there is an error, it prints a failure message with the error description. Otherwise, it prints a success message indicating that the file was downloaded successfully.

06

Unzipping the Downloaded File

If the file was successfully downloaded, the script proceeds to unzip the downloaded ZIP file using the **unzip** command.

DISCLAIMER!

Using the above method to access Google Drive files through a service account and obtaining file IDs comes with certain risks. Some of the potential risks include:

01 **Service Account Security**
Failure to adequately secure the service account's private key or credentials could potentially lead to unauthorized access by malicious actors.Note that private keys generate from service accounts never expire.

02 **API Changes and Deprecation**
Google may change or deprecate the Google Drive API, affecting the functionality of the script and requiring updates to keep it functional.

03 **Rate Limiting**
The Google Drive API imposes rate limits on the number of requests per day, which, if exceeded, may result in temporary service disruptions.

KEY LEAKAGE CONTINGENCY

In the event of a key leakage or unauthorized access to the service account key, immediate action must be taken to mitigate potential security risks.

Note that service account keys never expire.But you can manually delete them and can generate new keys.Maintaining a key rotation is recommended to prevent hazards.https://cloud.google.com/iam/docs/key-rotation

Additionally, ensuring proper authorization, monitoring, and access controls are crucial to safeguard sensitive data on Google Drive.
Always refer to Google's documentation and guidelines to stay up-to-date with best practices and any changes to the API.

END