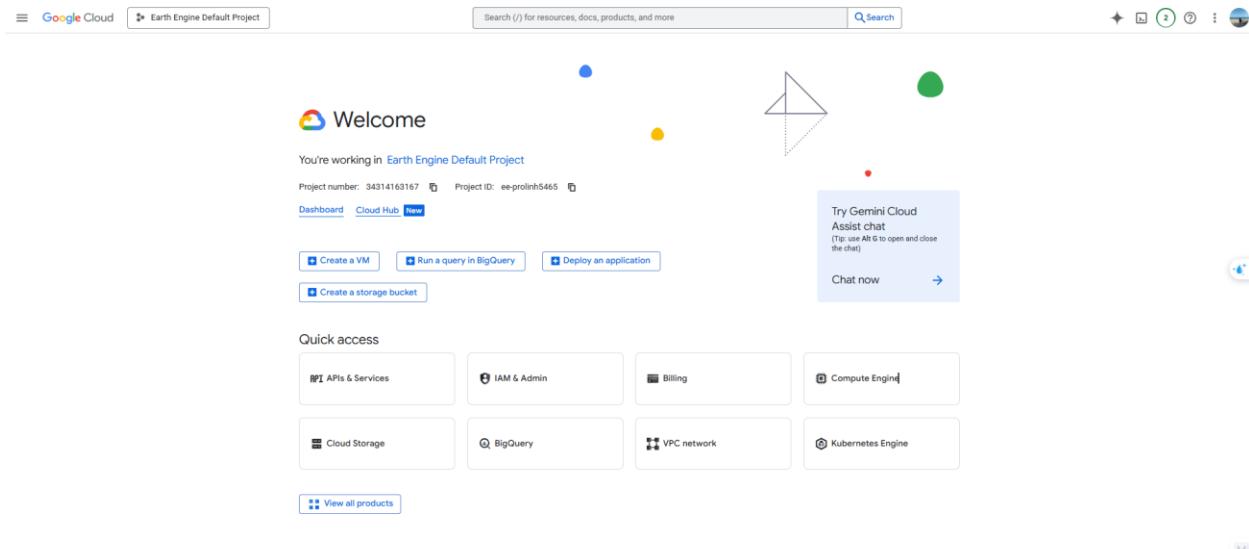


1. Introduction

In order to automate data reading and writing processes to Google Sheets through Python, it is necessary to establish a service account on the Google Cloud Platform (GCP). This account enables secure communication between Python scripts and Google services using OAuth 2.0 credentials. The following sections describe the step-by-step procedure to create a service account, enable the required APIs, and configure the authentication environment for Python integration.

2. Creating a Google Cloud Project

1. Access the Google Cloud Console at <https://console.cloud.google.com/>.
2. Log in using a valid Google account.
3. From the project selection menu, choose “**New Project**”.
4. Assign a project name (e.g., *Sheet Automation System*) and click **Create**.
5. Once created, ensure the project is selected as the active workspace.



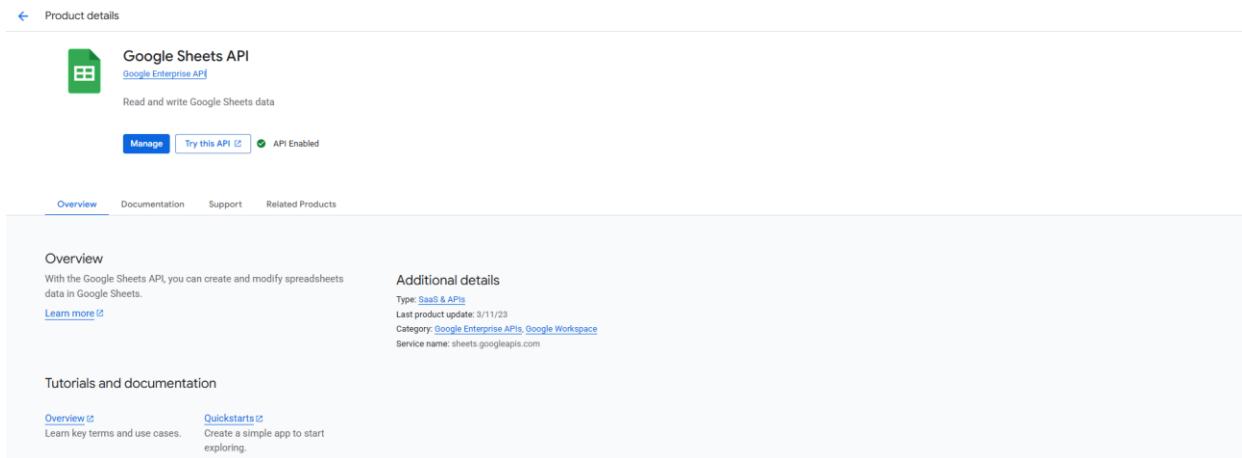
3. Enabling Google APIs

To allow programmatic interaction with Google Sheets, the following APIs must be activated within the project:

- **Google Sheets API**
- **Google Drive API**

These can be enabled by navigating to:

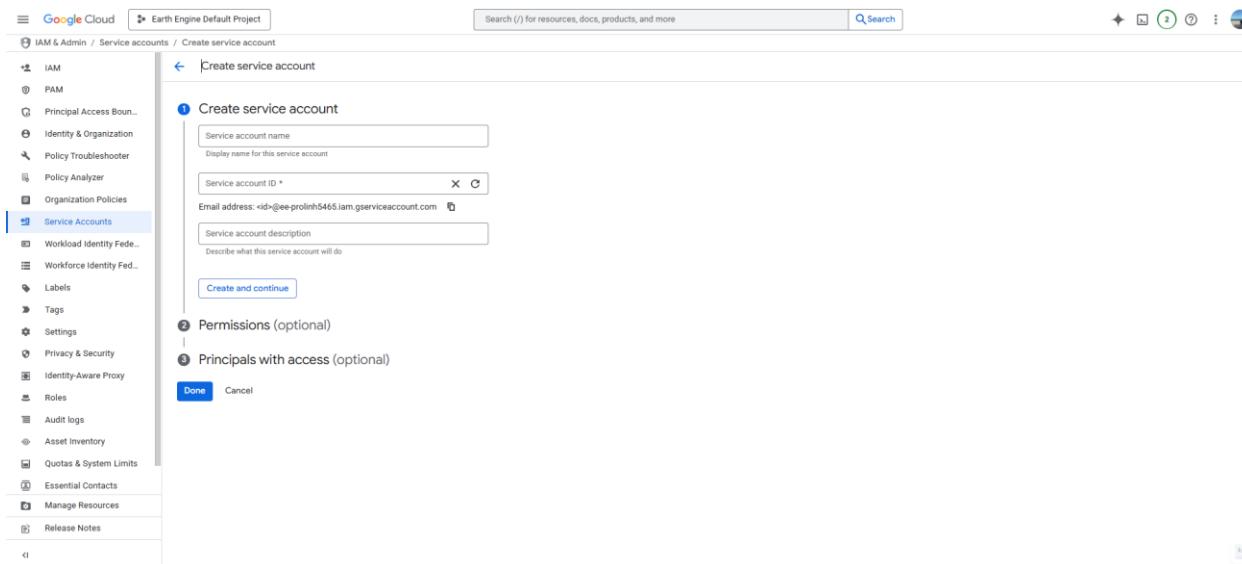
APIs & Services → Library, searching for each API by name, and selecting **Enable**.



The screenshot shows the 'Product details' page for the Google Sheets API. At the top, there's a green icon of a sheet of paper with a grid, followed by the text 'Google Sheets API' and 'Google Enterprise API'. Below this, a description reads 'Read and write Google Sheets data'. There are two buttons: 'Manage' and 'Try this API'. A status indicator shows 'API Enabled' with a green checkmark. Below the main title, there are tabs for 'Overview', 'Documentation', 'Support', and 'Related Products'. The 'Overview' tab is selected. Under 'Overview', there's a section titled 'Overview' with a brief description and a 'Learn more' link. To the right, under 'Additional details', it lists the type as 'SaaS & APIs', last update as '3/11/23', category as 'Google Enterprise APIs, Google Workspace', and service name as 'sheets.googleapis.com'. Below the overview, there's a section for 'Tutorials and documentation' with links to 'Overview' and 'Quickstarts'.

4. Creating a Service Account

1. Go to **APIs & Services → Credentials**.
2. Select **Create Credentials → Service Account**.
3. Provide an account name and description (e.g., *Python Sheets Access*).
4. Assign a role such as **Editor**, which grants permissions for reading and writing data.
5. Complete the setup and return to the service account management page.



The screenshot shows the 'Create service account' dialog in the Google Cloud IAM & Admin interface. On the left, a sidebar lists various IAM components like IAM, PAM, Principal Access Boundaries, Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, and Service Accounts. The 'Service Accounts' option is selected. The main panel has a heading 'Create service account' with a back arrow. It contains fields for 'Service account name' (with placeholder 'Display name for this service account'), 'Service account ID' (with placeholder 'Email address: <id>@ee-prolinh5465.iam.gserviceaccount.com'), and 'Service account description' (with placeholder 'Describe what this service account will do'). Below these are 'Create and continue' and 'Permissions (optional)' buttons. Under 'Permissions (optional)', there's a section for 'Principals with access (optional)'. At the bottom are 'Done' and 'Cancel' buttons.

5. Generating a Service Account Key

1. Under the created service account, open the **Keys** tab.
2. Select **Add Key → Create new key**.

3. Choose the **JSON** format and click **Create**.
4. A key file (e.g., *service-account-key.json*) will be downloaded.
5. Rename the file to `credentials.json` and store it securely in the working directory of the Python project.

The screenshot shows the Google Cloud IAM & Admin / Service accounts interface. The left sidebar has a tree view with 'Service Accounts' selected. The main panel shows a service account named 'sheet-writer'. The 'Details' tab is active. The 'Service account details' section contains fields for 'Name' (set to 'sheet-writer') and 'Description'. Below this, the 'Service account status' section shows the account is 'Enabled'. There is also a 'Disable service account' button. At the bottom, there is an 'Advanced settings' dropdown.

6. Granting Access to the Target Google Sheet

1. Open the desired Google Sheet.
2. Click **Share** and enter the service account's email address, which has the form:
3. `service-account-name@project-id.iam.gserviceaccount.com`
4. Assign the **Editor** role to allow modification privileges.