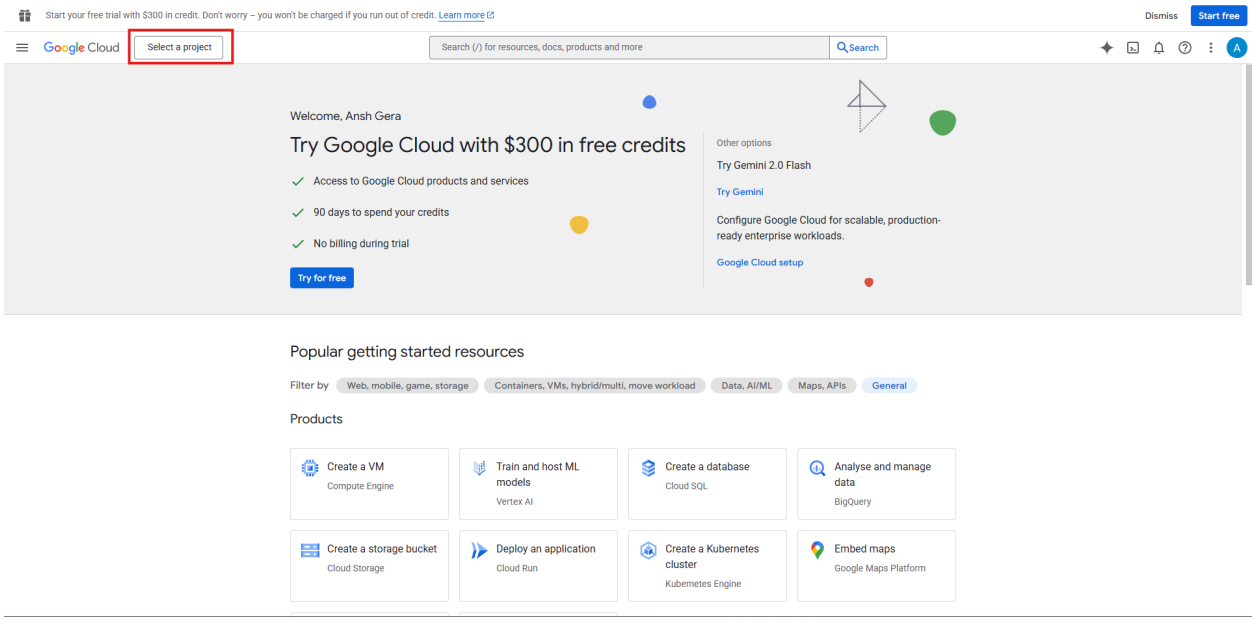
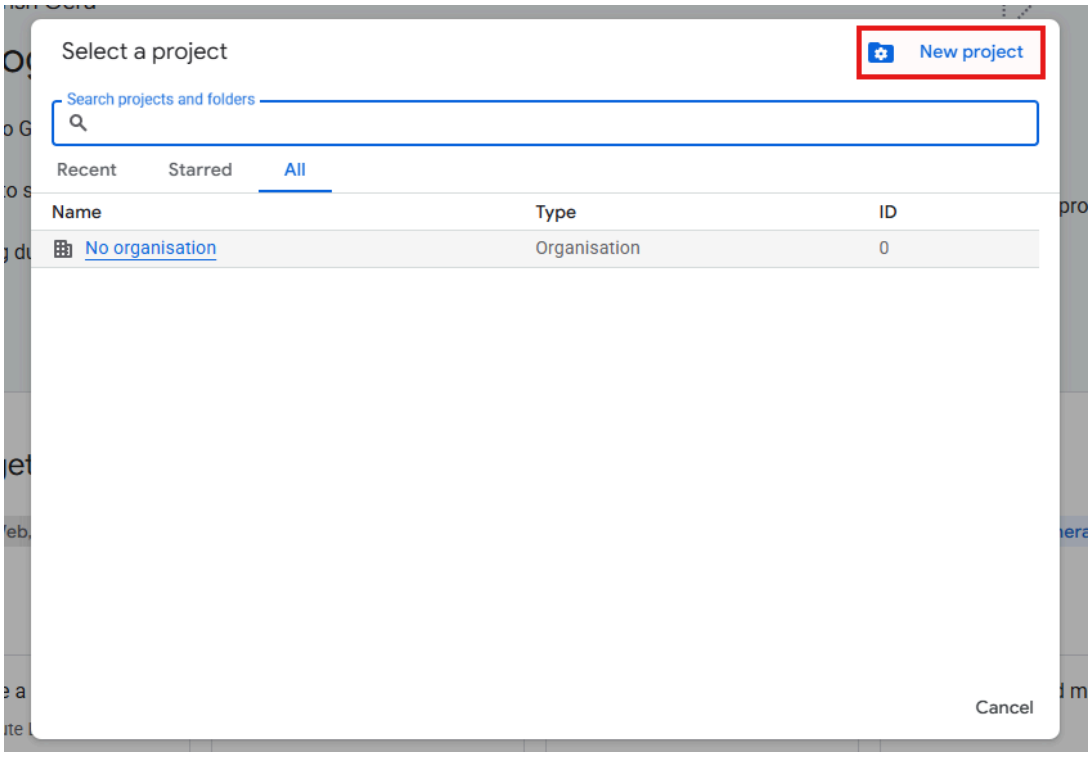
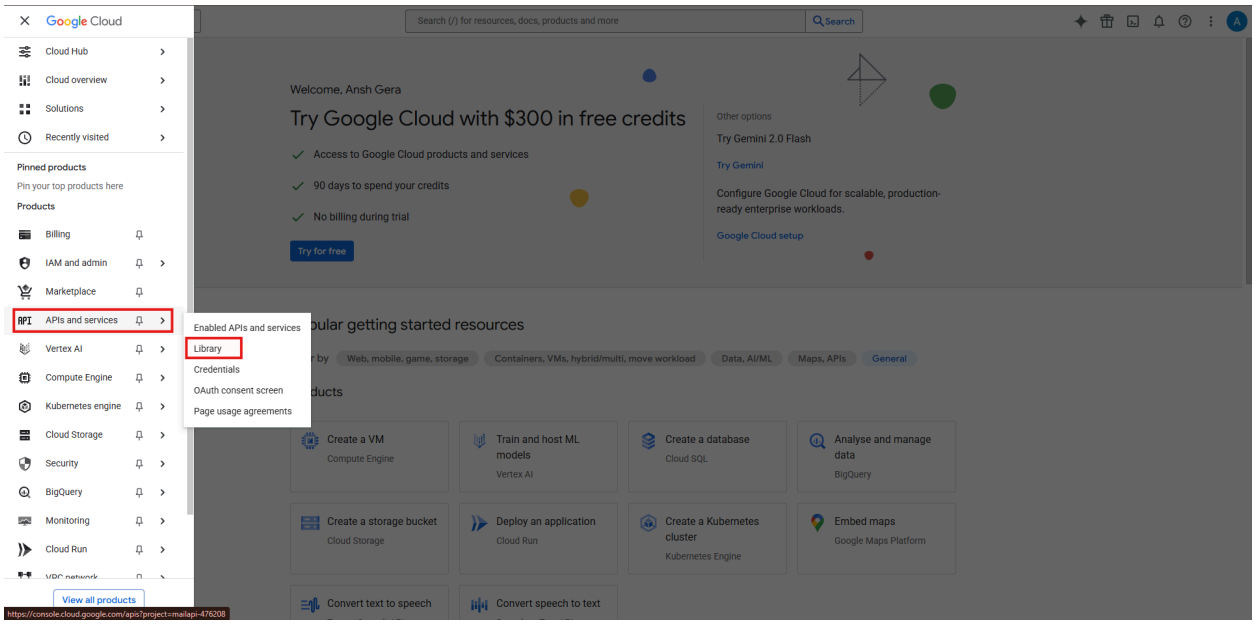
Steps to generate a Client ID & Client Secret

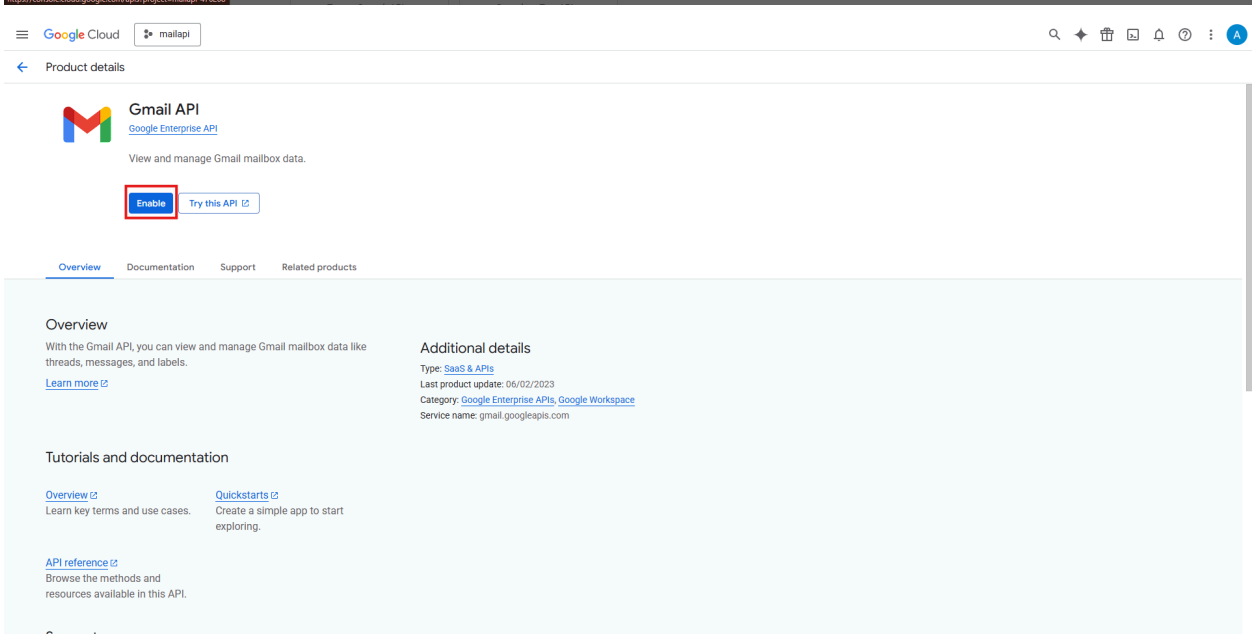
1. Open Google Cloud Console and pick (or create) a project
    - https://console.cloud.google.com/welcome/new?_gl=1*m5vlyu*_up*MQ..&gclid=CjwKCAjw6vHHBhBwEiwAq4zvA1hUdUi75rQP3pZ0Wcfzf64NcqC5w2c_pTVAk88K21IDJul2MDwr8RoCq68QAvD_BwE&gclsrc=aw.d
2. Visit the Google Cloud Console and either select an existing project or create a new one (top project selector → **New Project**).





3. Enable the Gmail API (or other Google APIs you need)
    - In the left menu: **APIs & Services** → **Library** → search for **Gmail API** → **Enable**. This ensures your project can request Gmail scopes.





4. Configure the OAuth consent screen
    - In the left menu: **APIs & Services** → **OAuth consent screen** (or **Google Auth Platform** → **OAuth consent screen**).
    - Choose user type: **External** (for general consumer accounts) or **Internal** (only for users in your Workspace org).
    - Fill in app name, support email, authorized domains, and add required scopes

- If you keep the app in *Testing*, add test users (their Google emails) — they'll be able to authorize while the app is unverified. To avoid frequent reauthorization, you'll publish (move to production)

5. Create OAuth credentials (Client ID & Secret)
    - Go to **APIs & Services** → **Credentials**.
    - Click **Create credentials** → **OAuth client ID**.
    - Select an **Application type**:
    - **eb application** — for server + browser flows. You must provide **Authorized redirect URIs** (e.g., `https://yourdomain.com/oauth2callback`).
    - **Desktop app** (or **Other/Installed app**) — for local testing / CLI apps.
    - **JavaScript (Web) clients** (e.g., "Web application" with JS origins) — note: pure frontend JS apps typically *don't* use a secret.
    - Give the client a name, add authorized redirect URIs (and Authorized JavaScript origins if applicable), then **Create**. Google will display the **Client ID** and **Client secret** and let you download a JSON file. Save them securely

6. Where to find them later / regenerate secret
    - In the Cloud Console: **APIs & Services** → **Credentials** → **OAuth 2.0 Client IDs**. Click your client to view the **Client ID** and **Client secret**.
    - To generate a new secret, open the client details and use **Add Secret / Reset secret** (this invalidates the old secret). Update any apps that use the old secret.

7. If you need domain-wide access (G Suite / Workspace admin)
    - For server-to-server access across a Workspace domain, create a **Service Account** and then configure **Domain-wide delegation** in the Workspace Admin console (Admin → Security → API controls → Manage Domain Wide Delegation) and give the service account the necessary scopes. This is different from OAuth client ID/secret flows for individual users.