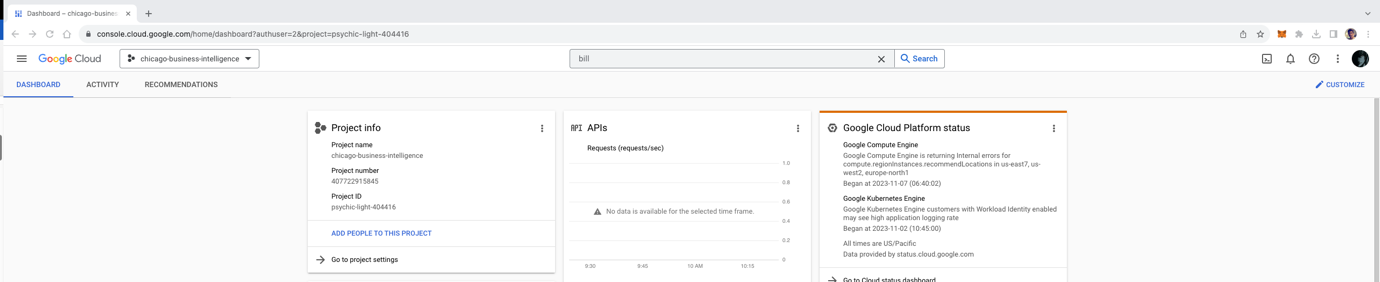
Steps to Deploy Go Microservice for Chicago Business Intelligence on GCP

**Step1: Initial Setup for Google Cloud Platform**

* Install the [google cloud CLI](https://cloud.google.com/sdk/docs/install) on your local machine.
* Create a new project on your [google cloud console](https://console.cloud.google.com/projectcreate). Make a note of the project id and project Name.

Graphical user interface, text, application

Description automatically generated



* After creating a project on Google Cloud Console execute “**gcloud init**” command on your local machine and select the project created above when prompted.

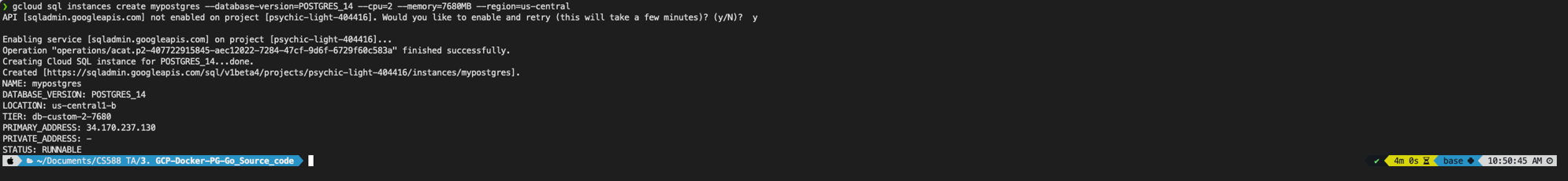
Text

Description automatically generated with low confidence

**Step 2: Postgres database Setup**

* Create database instance of postgres using the following command.

“**gcloud sql instances create mypostgres --database-version=POSTGRES\_14 --cpu=2 --memory=7680MB --region=us-central**”



* Create sql users on the database instance using the following command.

“gcloud sql users set-password postgres --instance=mypostgres --password=root”

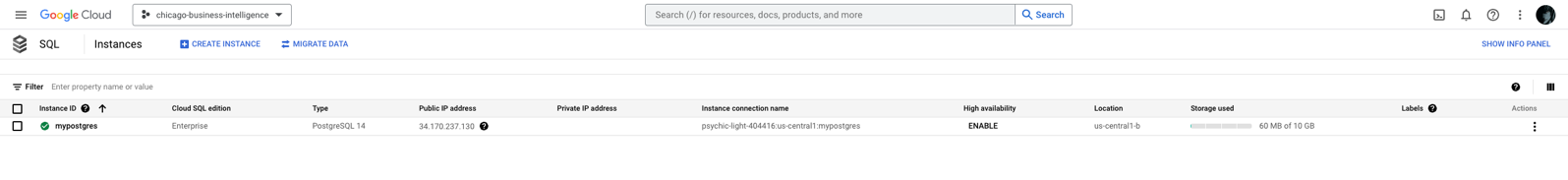


* Create a database for our microservice using the following command.

“gcloud sql databases create chicago\_business\_intelligence --instance=mypostgres”



* Open Google Cloud console, search for SQL and confirm that database instance is up and running



**Step 3: Setting up continuous deployment using cloud build.**

* Create a repository on GitHub to store the source code for our project.
* Open Google Cloud Console, Search for Cloud build API, and Enable it for your project

Graphical user interface, text, application

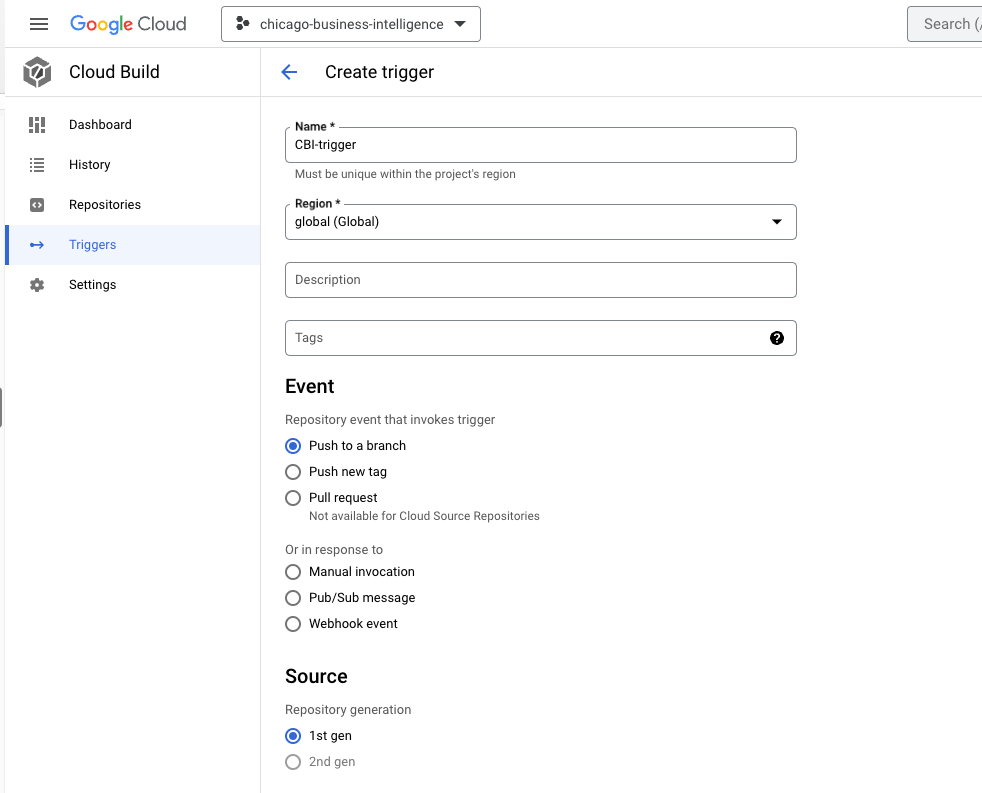
Description automatically generated

* After the API is enabled, click on the create trigger button.

Graphical user interface, text, application

Description automatically generated

* Fill the details for the trigger as shown in the below images.

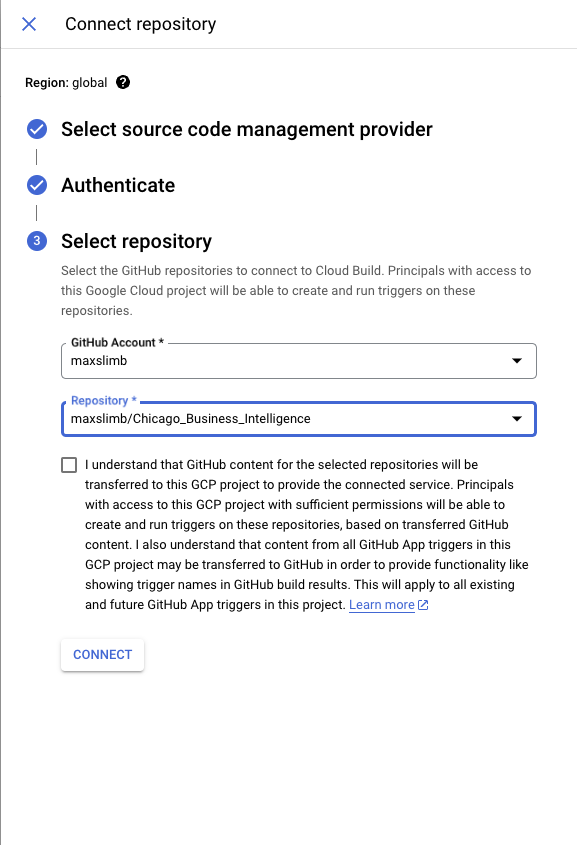


* Click on connect repository, select github and authenticate.

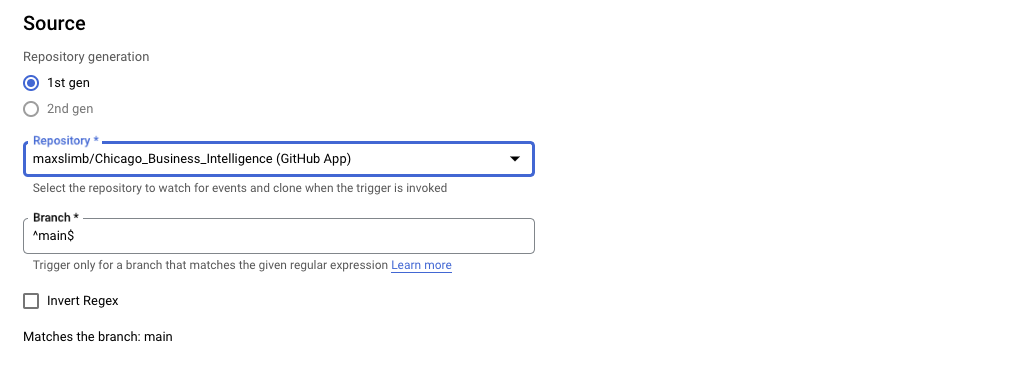
Graphical user interface, application

Description automatically generated

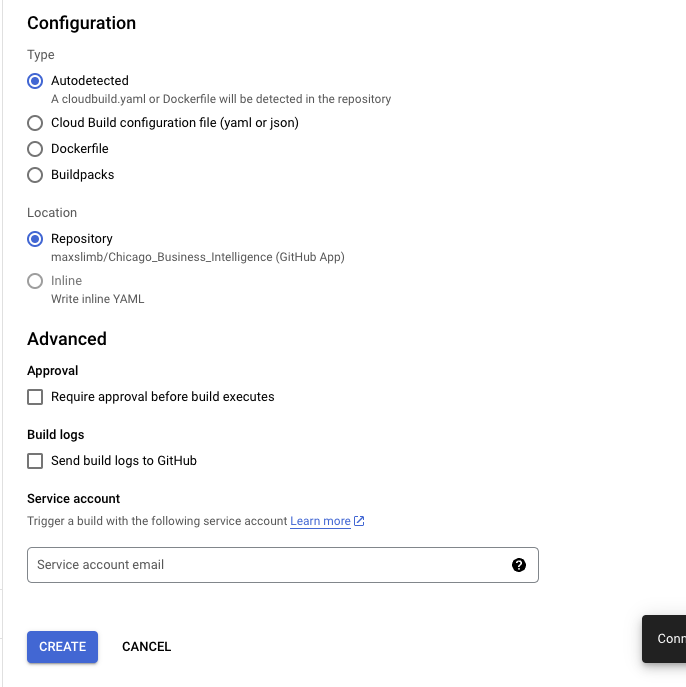
* After authentication select the repository created for Chicago business intelligence.



* Select the repository after connecting the project.



* Click on Create to create the trigger.



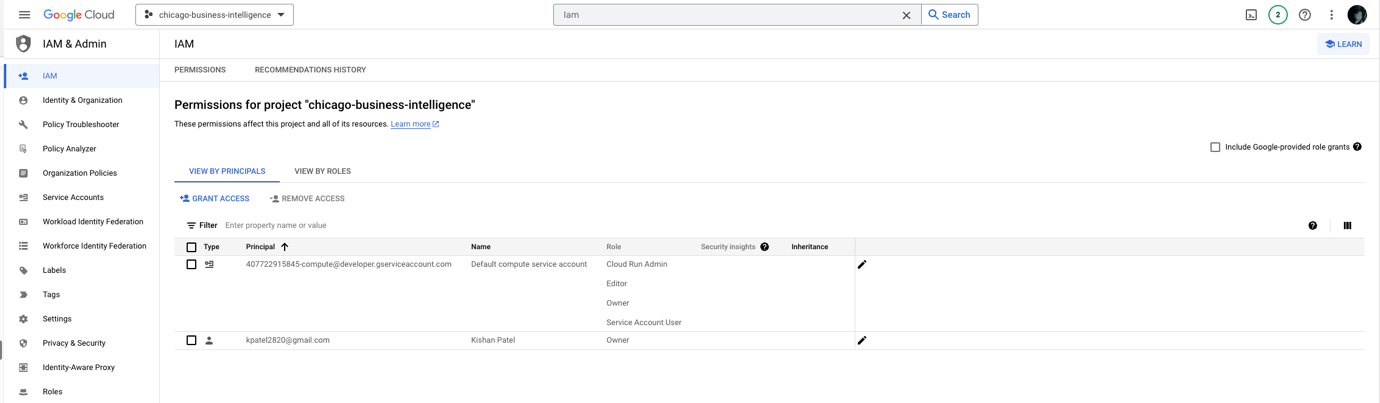
**Step 4: Setting up the containers for Go-microservice and Pgadmin**

* Enable Cloud Run API for your project.

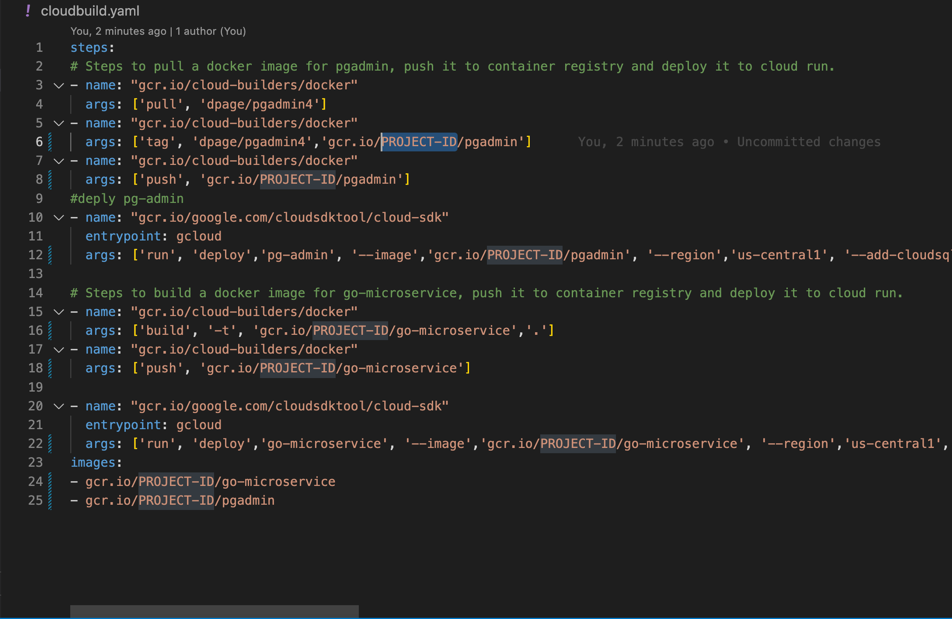
Graphical user interface, text, application

Description automatically generated

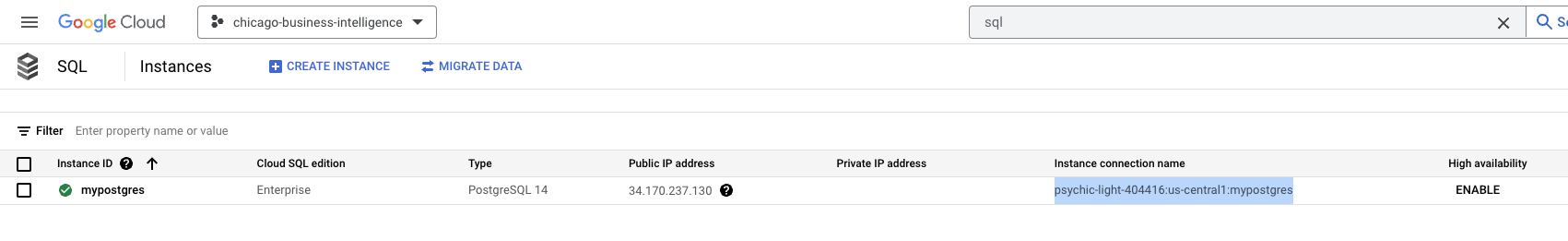
* Go to IAM page and make sure all the required roles are enabled for the project.



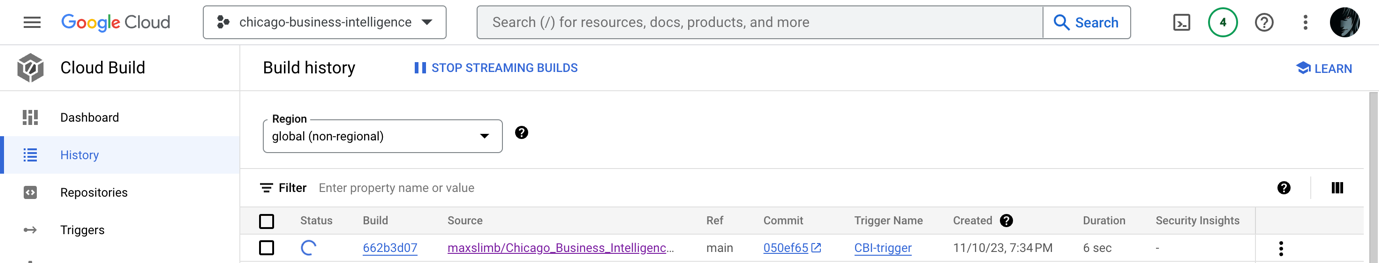
* The images for the go microservice and pgadmin are created with the help of cloudbuild.yaml file, you have to **update your Project ID** in various places in the cloudbuild.yaml file



* Go to the postgres instance created in the previous steps and copy the instance connection name.



* Update line 198 of your main.go source code file and update the connection string with your Instance connecton name as shown below.
* connectionName := “**psychic-light-404416:us-central1:mypostgres”**
* Update the line 285 with the Geocoder API-KEY
* Push the source code along with the cloudbuild.yaml file to the repository created in the above steps
* A build is triggered in the cloud build immediately after pushing the code to GitHub.



* Wait for the build to be complete. Build logs can be viewed by clicking on the build id.
* Go to Cloud Run, click on pgadmin, copy the highlighted URLGraphical user interface, application

  Description automatically generated

Graphical user interface, application, email

Description automatically generated

Open the URL in a Browser and Login to pgadmin to validate that tables are created.