

Steps to Deploy Go Microservice for Chicago Business Intelligence on GCP

Step1: Initial Setup for Google Cloud Platform

- Install the [google cloud CLI](#) on your local machine.
- Create a new project on your [google cloud console](#). Make a note of the project id and project Name.

Google Cloud Platform

New Project

You have 20 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)

[MANAGE QUOTAS](#)

Project name *

Project ID: chicago-business-intelligence. It cannot be changed later. [EDIT](#)

Location * [BROWSE](#)

Parent organization or folder

[CREATE](#) [CANCEL](#)

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

```
Your current project has been set to: [chicago-business-intelligence].
```

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**”

```
C:\Users\Nikhil>gcloud sql instances create mypostgres --database-version=POSTGRES_14 --cpu=2 --memory=7680MB --region=us-central
API [sqladmin.googleapis.com] not enabled on project [506143517878]. Would you like to enable and retry (this will take a few minutes)? (y/N)? y

Enabling service [sqladmin.googleapis.com] on project [506143517878]...
Operation "operations/acat.p2-506143517878-5e8e21f1-0f64-4a77-a17f-991576d8491a" finished successfully.
Creating Cloud SQL instance...done.
Created [https://sqladmin.googleapis.com/sql/v1beta4/projects/chicago-business-intelligence/instances/mypostgres].
NAME: mypostgres
DATABASE_VERSION: POSTGRES_14
LOCATION: us-central1-a
TIER: db-custom-2-7680
PRIMARY_ADDRESS: 35.188.176.43
PRIVATE_ADDRESS: -
STATUS: RUNNABLE
```

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

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

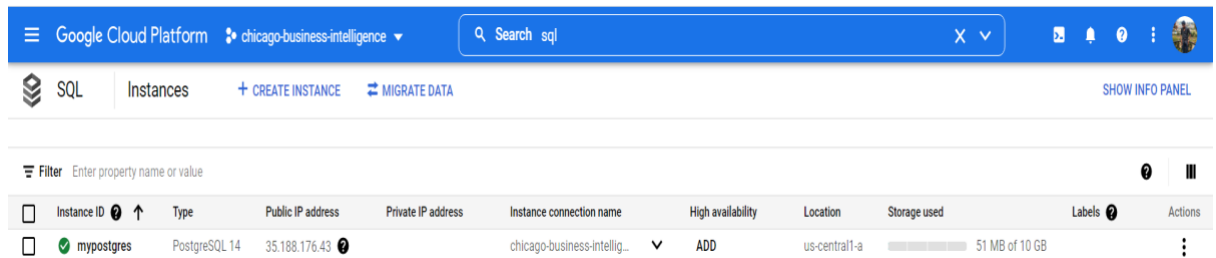
```
C:\Users\Nikhil>gcloud sql users set-password postgres --instance=mypostgres --password=root
Updating Cloud SQL user...done.
```

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

“gcloud sql databases create chicago_business_intelligence --instance=mypostgres”

```
C:\Users\Nikhil>gcloud sql databases create chicago_business_intelligence --instance=mypostgres
Creating Cloud SQL database...done.
Created database [chicago_business_intelligence].
instance: mypostgres
name: chicago_business_intelligence
project: chicago-business-intelligence
```

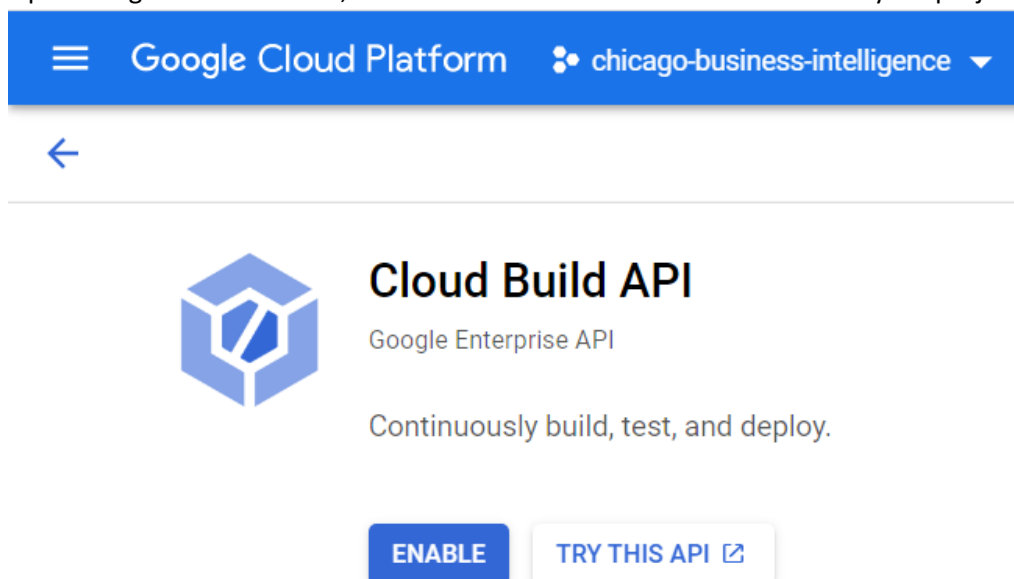
- Open Google Cloud console, search for SQL and confirm that database instance is up and running



Instance ID	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Storage used	Labels	Actions
✓ mypostgres	PostgreSQL 14	35.188.176.43		chicago-business-intellig...	ADD	us-central1-a	51 MB of 10 GB		

Step 3: Setting up continuous deployment using cloud build.


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



Google Cloud Platform

chicago-business-intelligence

←



Cloud Build API

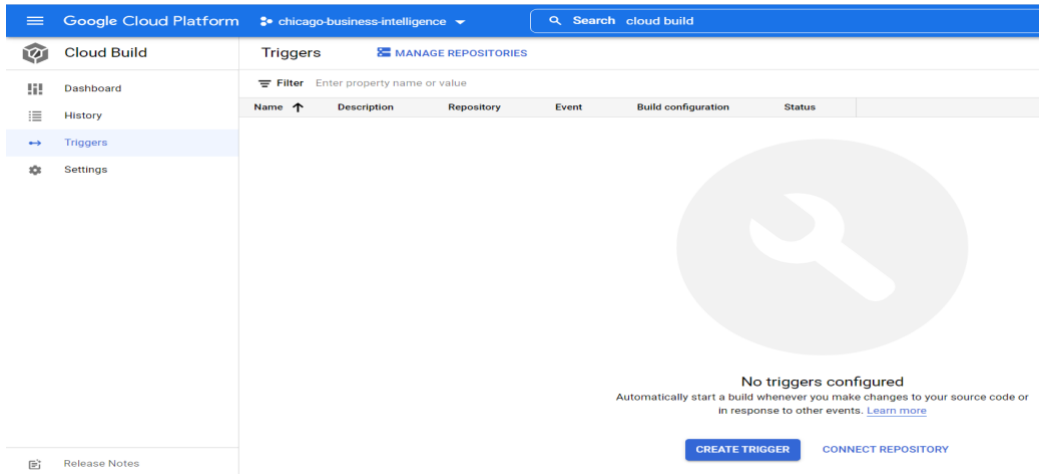
Google Enterprise API

Continuously build, test, and deploy.

ENABLE

TRY THIS API

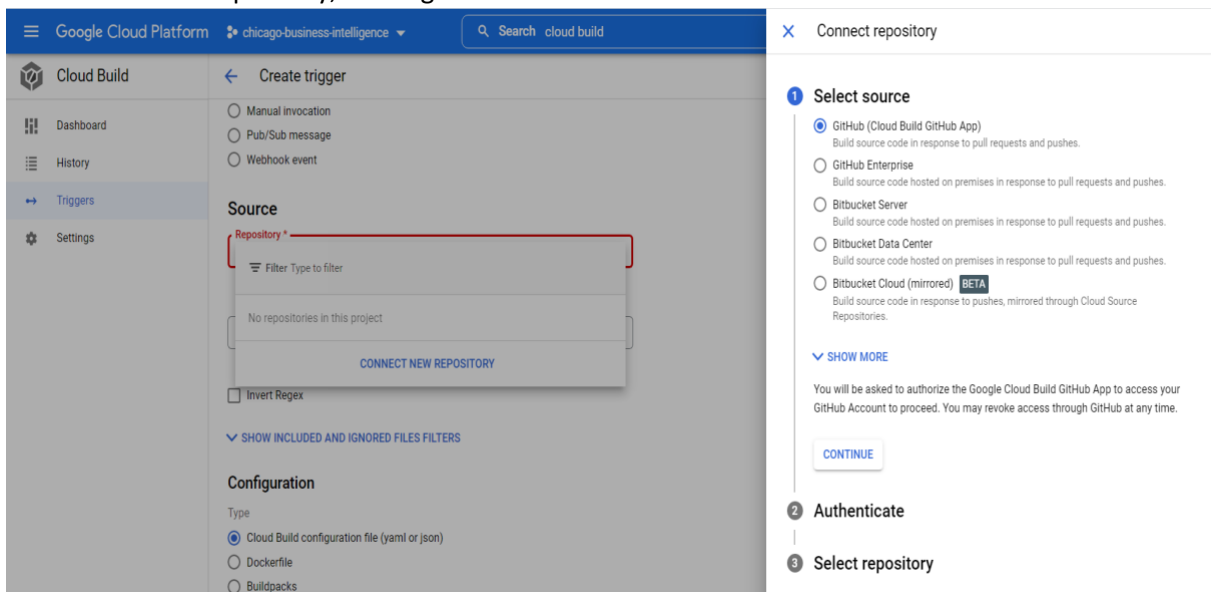
- After the API is enabled, click on the create trigger button.



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

The screenshot shows the 'Create trigger' form in Google Cloud Platform. The left sidebar is the same as the previous image. The main panel is titled 'Create trigger'. It has three input fields: 'Name *' with the value 'cbl-trigger', 'Description', and 'Tags'. Below these is the 'Event' section. It starts with the text 'Repository event that invokes trigger'. There are three radio buttons: 'Push to a branch' (selected), 'Push new tag', and 'Pull request'. Below 'Pull request' is the text 'Not available for Cloud Source Repositories'. Then, it says 'Or in response to' followed by three radio buttons: 'Manual invocation', 'Pub/Sub message', and 'Webhook event'.

- Click on connect repository, select github and authenticate.



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

×

Connect repository

✓

Select source

↓

✓

Authenticate

↓

3

Select repository

Select the GitHub repositories to connect to Cloud Build. Principals with access to this Google Cloud project will be able to create and run triggers on these repositories.

GitHub Account *

nikhilsarika

▼

Repository *

nikhilsarika/chicago_business_intelligence

▼

☒

I understand that GitHub content for the selected repositories will be transferred to this GCP project to provide the connected service. Principals with access to this GCP project with sufficient permissions will be able to create and run triggers on these repositories, based on transferred GitHub content. I also understand that content from all GitHub App triggers in this GCP project may be transferred to GitHub in order to provide functionality like showing trigger names in GitHub build results. This will apply to all existing and future GitHub App triggers in this project. [Learn more](#)

CONNECT

- Select the repository after connecting the project.

Source

Repository *

nikhilsarika/chicago_business_intelligence (GitHub App)

▼

Select the repository to watch for events and clone when the trigger is invoked

Branch *

^main\$

Use a regular expression to match to a specific branch [Learn more](#)

☐

Invert Regex

Matches the branch: main

- Click on create to create the trigger.

Configuration

Type

☒

Autodetected

A cloudbuild.yaml or Dockerfile will be detected in the repository

☐

Cloud Build configuration file (yaml or json)

☐

Dockerfile

☐

Buildpacks

Location

☒

Repository

nikhilsarika/chicago_business_intelligence (GitHub App)

☐

Inline

Write inline YAML

Advanced

Approval

☐

Require approval before build executes

Service account

Trigger a build with the following service account [Learn more](#)

Service account email

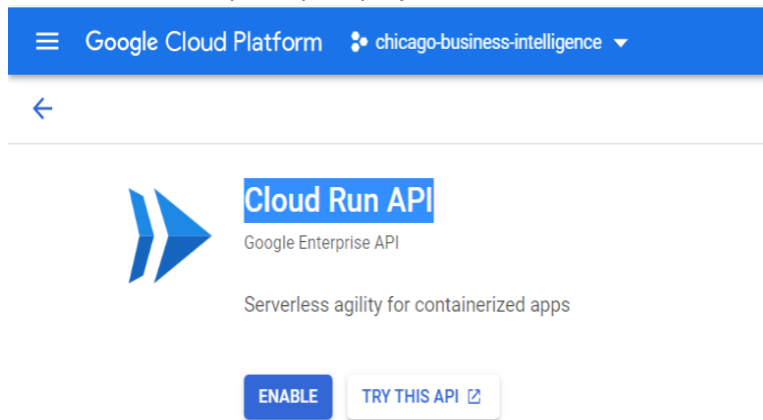
?

CREATE

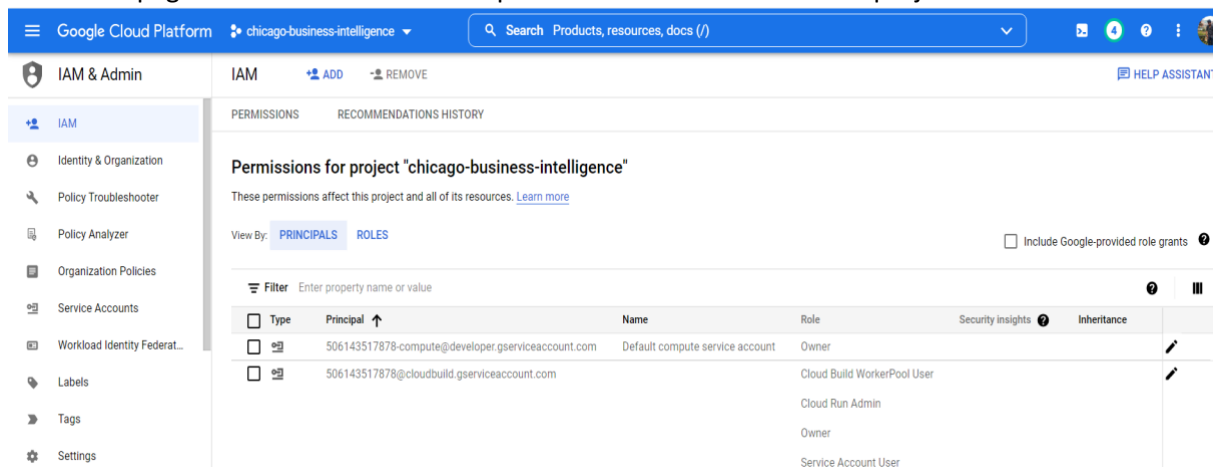
Cancel

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

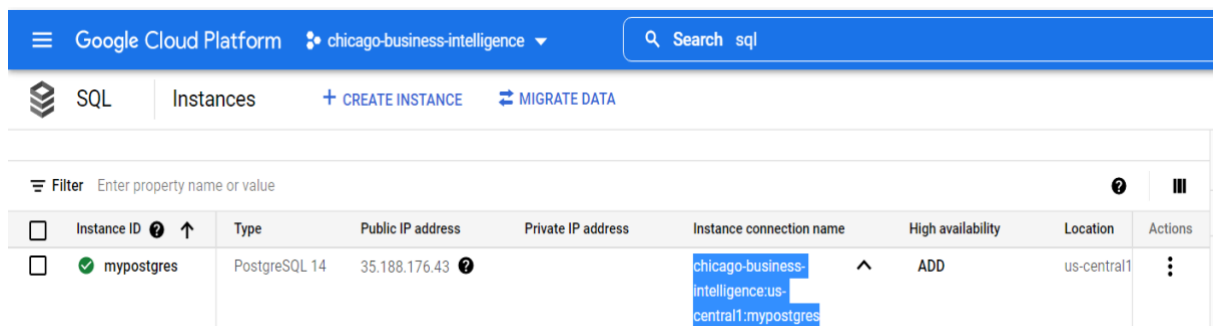
- Enable cloud run api for your project.



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



- Uncomment line 189 of your main.go source code file and update the connection string with your Instance connecton name as shown below.

"user=postgres dbname=chicago_business_intelligence password=root host=/cloudsql/chicago-business-intelligence:us-central1:mypostgres sslmode=disable port = 5432"

- Push the source code along with the cloudbuild.yaml file to the repository created in the above steps
- A build is triggered in cloud build immediately after pushing the code to the github.

The screenshot shows the Google Cloud Platform Cloud Build interface. The top navigation bar includes the Google Cloud Platform logo, the project name 'chicago-business-intelligence', and a search bar. The left sidebar contains links to Dashboard, History, Triggers, and Settings. The main content area is titled 'Build history' and shows a table of builds. The first build is highlighted with a blue row.

Status	Build	Source	Ref	Commit	Trigger Name	Created	Duration
<input checked="" type="checkbox"/>	06ba928e	nikhilsarika/chicago_business_intelligence	main	7134b02	cbi-trigger	4/13/22, 1:37 AM	12 sec

- Wait for the build to be complete. Build logs can be viewed by clicking on the build id.

The screenshot shows the Google Cloud Platform Cloud Build interface with the 'Build details' page for build '6e7f2f63'. The build is marked as 'Successful'. The left sidebar shows the 'History' link. The main content area displays the build steps and logs. The logs show the execution of various commands, including pulling Docker images, pushing to Cloud Storage, and deploying to Cloud Run.

Steps	Duration
Build Summary	00:02:43
0: gcr.io/cloud-builders/docker pull dpape/pgadmin4	00:00:10
1: gcr.io/cloud-builders/docker tag dpape/pgadmin4 gcr.io/chicago-busi...	00:00:00
2: gcr.io/cloud-builders/docker push gcr.io/chicago-business-intelligenc...	00:00:01
3: gcr.io/google.com/cloudsdktool/cloud-sdk run deploy chicago-business-intel...	00:01:45
4: gcr.io/cloud-builders/docker build -t gcr.io/chicago-business-intellige...	00:00:11
5: gcr.io/cloud-builders/docker push gcr.io/chicago-business-intelligenc...	00:00:04
6: gcr.io/google.com/cloudsdktool/cloud-sdk run deploy chicago-business-intel...	00:00:22

- Go to Cloud Run, click on pgadmin, copy the highlighted URL

The screenshot shows the Google Cloud Platform Cloud Run interface. The top navigation bar includes the Google Cloud Platform logo, the project name 'chicago-business-intelligence', and a search bar. The left sidebar shows the 'Services' link. The main content area displays a table of services. The 'pg-admin' service is highlighted, and its details are shown in a separate view. The details view shows the service name, region, and a URL to access the pgadmin interface.

Name	Req/sec	Region	Authentication	Ingress	Last deployed	Deployed by
go-microservice	0	us-central1	Allow unauthenticated	All	1 minute ago	Cloud Build
pg-admin	0	us-central1	Allow unauthenticated	All	2 minutes ago	Cloud Build

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