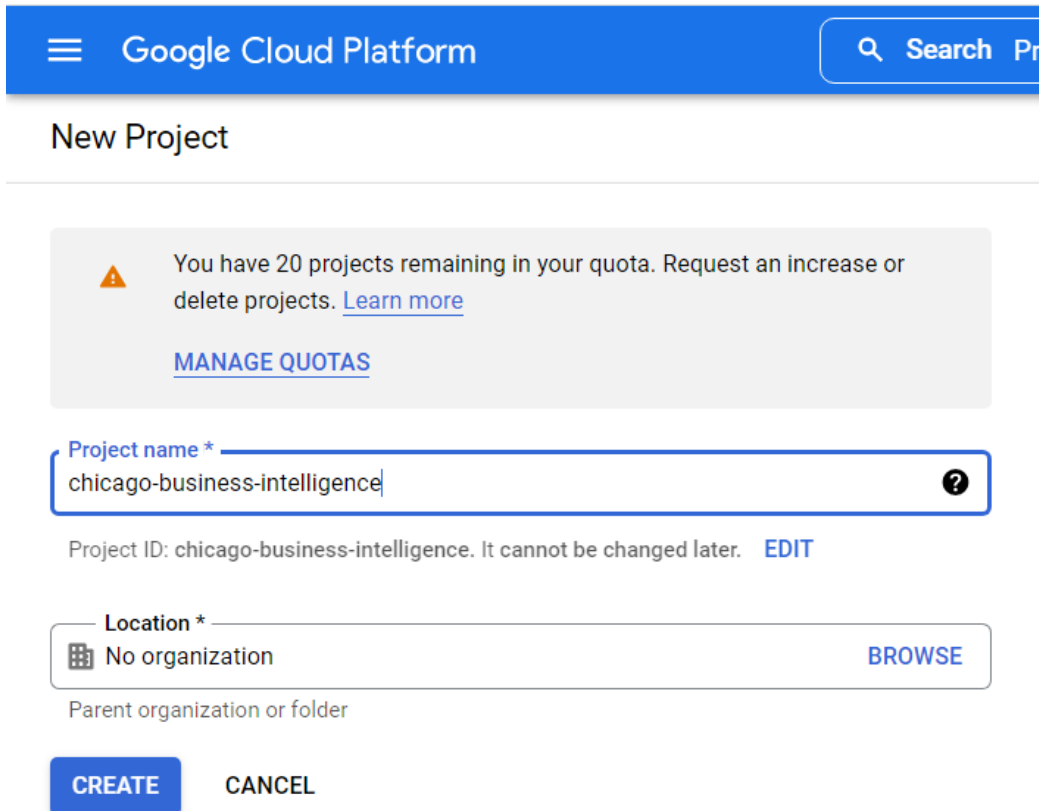


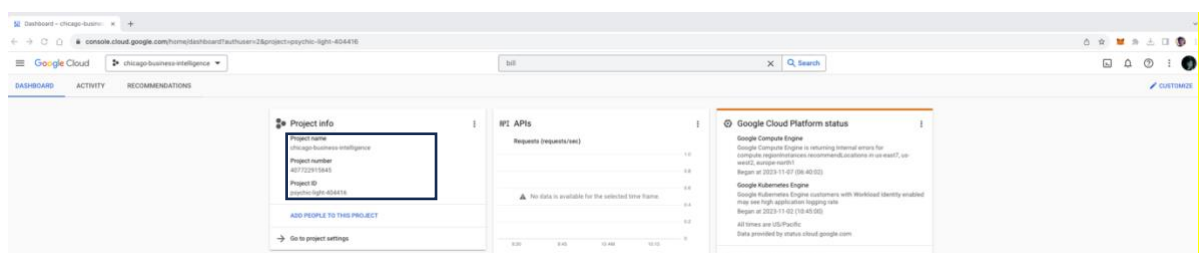
Steps to Deploy Go Microservice for Chicago Business Intelligence on GCP

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.



The screenshot shows the 'New Project' form in the Google Cloud Platform console. At the top, there's a blue header with the Google Cloud Platform logo and a search bar. Below the header, the title 'New Project' is displayed. A warning message states: 'You have 20 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)'. Below this, there's a link 'MANAGE QUOTAS'. The 'Project name' field is filled with 'chicago-business-intelligence'. Below it, the 'Project ID' is shown as 'chicago-business-intelligence' with a note that it cannot be changed later and an 'EDIT' link. The 'Location' field is set to 'No organization' with a 'BROWSE' button. At the bottom, there are 'CREATE' and 'CANCEL' buttons.



- 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**”

```

$ gcloud sql instances create mypostgres --database-version=POSTGRES_14 --cpu=2 --memory=7680MB --region=us-central1
API [sqladmin.googleapis.com] not enabled on project [psychic-light-484416]. Would you like to enable and retry (this will take a few minutes)? (y/N)? y
Enabling service [sqladmin.googleapis.com] on project [psychic-light-484416]...
Operation "operations/acet.p2-48772915845-acc12822-7284-47cf-9d6f-6729f68c583a" finished successfully.
Creating Cloud SQL instance for POSTGRES_14...done.
Created [https://sqladmin.googleapis.com/sql/v1beta4/projects/psychic-light-484416/instances/mypostgres].
NAME: mypostgres
DATABASE_VERSION: POSTGRES_14
LOCATION: us-central1-b
TIER: db-custom-2-7680
PRIMARY_ADDRESS: 34.178.237.130
PRIVATE_ADDRESS: -
STATUS: RUNNABLE
~/Documents/CS588 TA/3. GCP-Docker-PG-Go_Source_code

```

- Create sql users on the database instance using the following command.
“gcloud sql users set-password postgres --instance=mypostgres --password=root”

```

$ gcloud sql users set-password postgres --instance=mypostgres --password=root
Updating Cloud SQL user...done.
~/Documents/CS588 TA/3. GCP-Docker-PG-Go_Source_code

```

- Create a database for our microservice using the following command.
“gcloud sql databases create chicago_business_intelligence --instance=mypostgres”

```

$ 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: psychic-light-484416
~/Documents/CS588 TA/3. GCP-Docker-PG-Go_Source_code

```

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

Google Cloud

chicago-business-intelligence

Search (/) for resources, docs, products, and more

Search

SQL

Instances

CREATE INSTANCE

MIGRATE DATA

SHOW INFO PANEL

Filter

Enter property name or value

| Instance ID | Cloud SQL edition | Type | Public IP address | Private IP address | Instance connection name | High availability | Location | Storage used | Labels | Actions |
|--|-------------------|---------------|-------------------|--------------------|--|-------------------|---------------|----------------|--------|---------|
| <div><div></div><div>mysqlpostgres</div></div> | Enterprise | PostgreSQL 14 | 34.178.237.130 | | psychic-light-484416-us-central1-mysqlpostgres | ENABLE | us-central1-b | 40 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 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

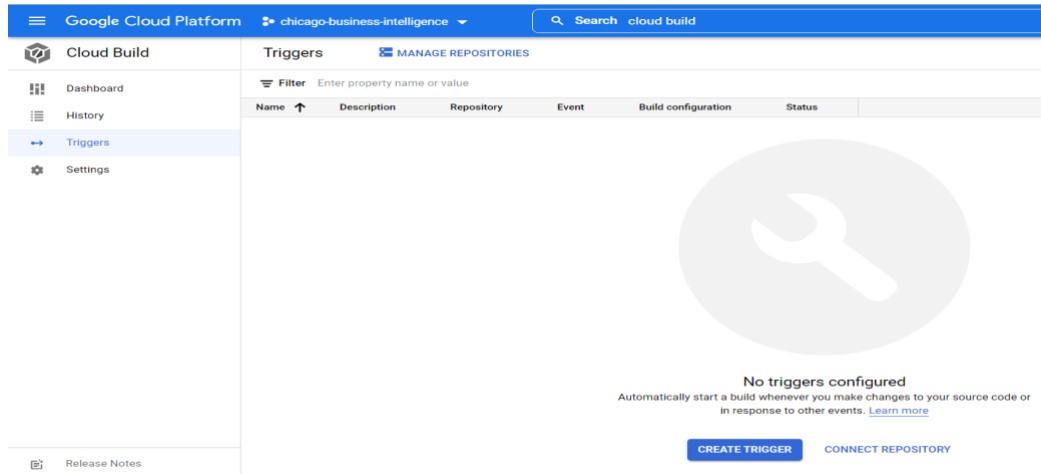
Google Enterprise API

Continuously build, test, and deploy.

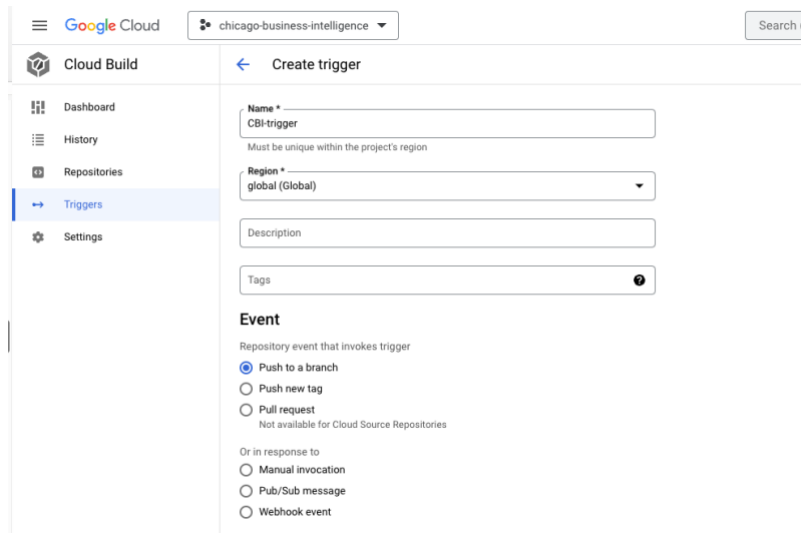
ENABLE

TRY THIS API

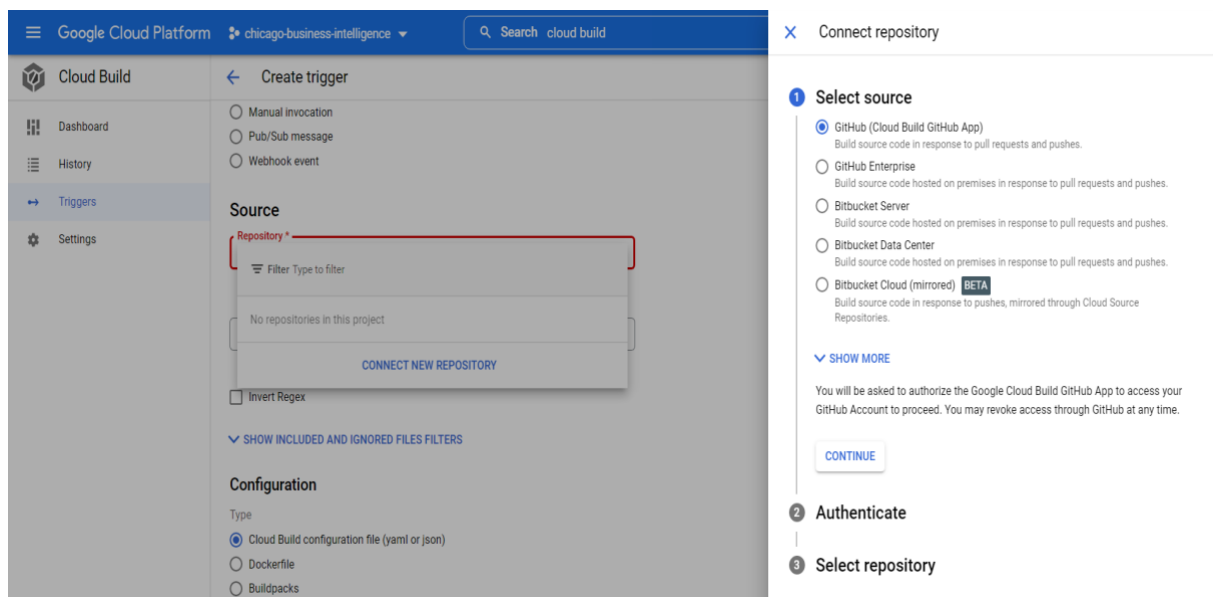
- After the API is enabled, click on the create trigger button.



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



- Click on connect repository, select github and authenticate.



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

✕

Connect repository

Region: global ⓘ

✓

Select source code management provider

✓

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 *

maxslimb

Repository *

maxslimb/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 generation

☒

1st gen

☐

2nd gen

Repository *

maxslimb/Chicago_Business_Intelligence (GitHub App)

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

Branch *

*main\$

Trigger only for a branch that matches the given regular expression [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

maxslimb/Chicago_Business_Intelligence (GitHub App)

☐

Inline

Write inline YAML

Advanced

Approval

☐

Require approval before build executes

Build logs

☐

Send build logs to GitHub

Service account

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

Service account email

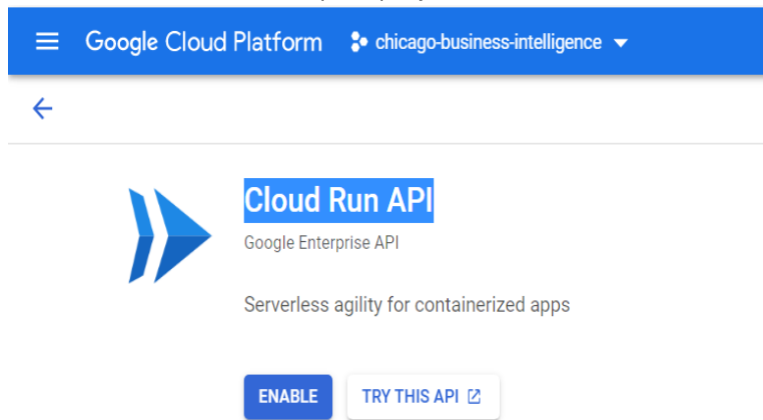
?

CREATE

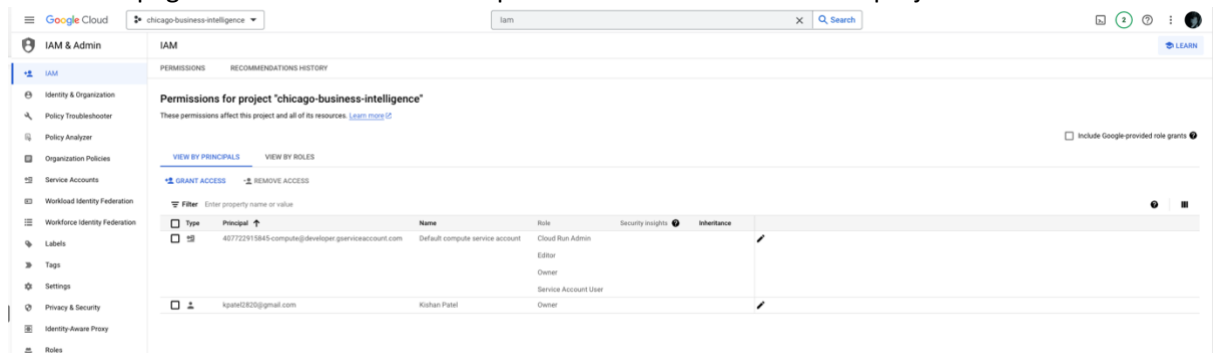
CANCEL

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, you have to **update your Project ID** in various places in the cloudbuild.yaml file

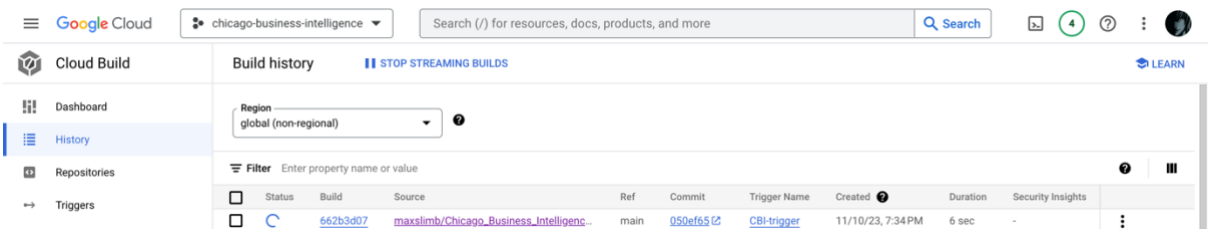
```
! cloudbuild.yaml
You, 2 minutes ago | 1 author (You)
1 steps:
2 # Steps to pull a docker image for pgadmin, push it to container registry and deploy it to cloud run.
3 - name: "gcr.io/cloud-builders/docker"
4   args: ['pull', 'dpape/pgadmin4']
5 - name: "gcr.io/cloud-builders/docker"
6   args: ['tag', 'dpape/pgadmin4', 'gcr.io/PROJECT-ID/pgadmin']
7 - name: "gcr.io/cloud-builders/docker"
8   args: ['push', 'gcr.io/PROJECT-ID/pgadmin']
9 #deploy pg-admin
10 - name: "gcr.io/google.com/cloudsdktool/cloud-sdk"
11   entrypoint: gcloud
12   args: ['run', 'deploy', 'pg-admin', '--image', 'gcr.io/PROJECT-ID/pgadmin', '--region', 'us-central1', '--add-cloudsq
13
14 # Steps to build a docker image for go-microservice, push it to container registry and deploy it to cloud run.
15 - name: "gcr.io/cloud-builders/docker"
16   args: ['build', '-t', 'gcr.io/PROJECT-ID/go-microservice', '.']
17 - name: "gcr.io/cloud-builders/docker"
18   args: ['push', 'gcr.io/PROJECT-ID/go-microservice']
19
20 - name: "gcr.io/google.com/cloudsdktool/cloud-sdk"
21   entrypoint: gcloud
22   args: ['run', 'deploy', 'go-microservice', '--image', 'gcr.io/PROJECT-ID/go-microservice', '--region', 'us-central1',
23
24   images:
25     - gcr.io/PROJECT-ID/go-microservice
26     - gcr.io/PROJECT-ID/pgadmin
```

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



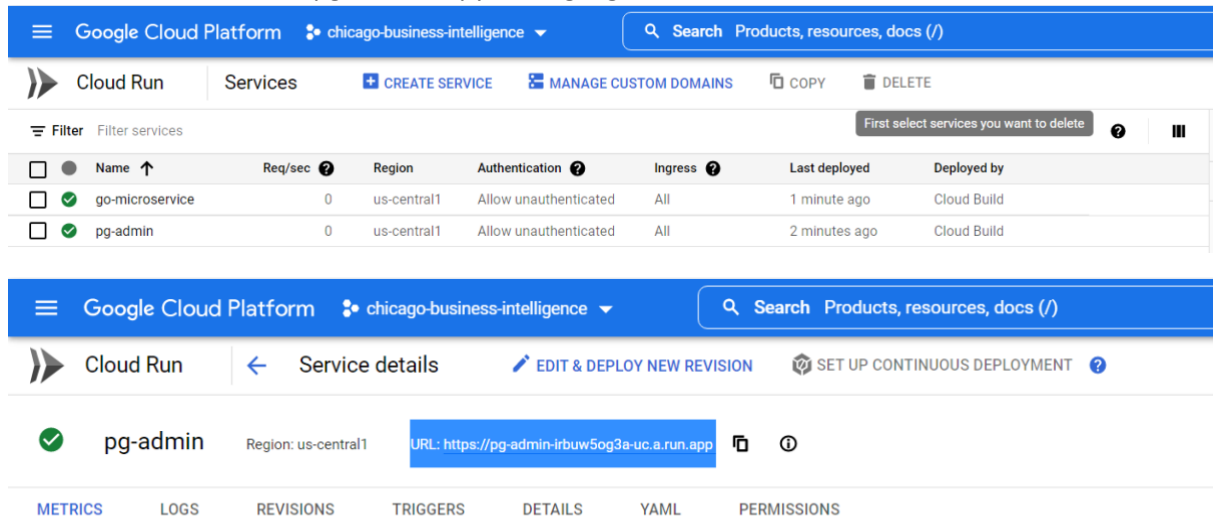
| Instance ID | Cloud SQL edition | Type | Public IP address | Private IP address | Instance connection name | High availability |
|-------------|-------------------|---------------|-------------------|--------------------|---|-------------------|
| mypostgres | Enterprise | PostgreSQL 14 | 34.170.237.130 | | psychic-light-404416-us-central1-mypostgres | ENABLE |

- Update line 198 of your main.go source code file and update the connection string with your Instance connection 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.

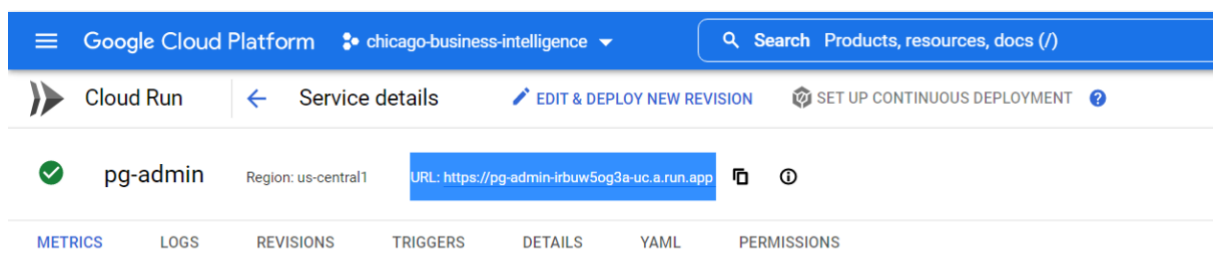


| Status | Build | Source | Ref | Commit | Trigger Name | Created | Duration | Security Insights |
|-----------|----------|---|------|---------|--------------|-------------------|----------|-------------------|
| Completed | 662b3d07 | maxsimb/Chicago_Business_Intelligenc... | main | 050ef65 | CBI-trigger | 11/10/23, 7:34 PM | 6 sec | - |

- 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 URL



| 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 |



pg-admin Region: us-central1 URL: <https://pg-admin-irbuw5og3a-uc.a.run.app>

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