

Deploying Go Microservice for Chicago Business Intelligence on GCP using SQL, Cloud Run, CI/CD Triggers for GitHub Repo

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' page in the Google Cloud Platform. At the top, there's a blue header bar with the 'Google Cloud Platform' logo and a search bar. Below the header, the text 'New Project' is displayed. A warning message in a grey box says: '⚠ You have 20 projects remaining in your quota. Request an increase or delete projects. [Learn more](#)' with a 'MANAGE QUOTAS' link. The 'Project name *' field contains 'chicago-business-intelligence'. To the right of the field is a help icon (a question mark inside a circle) and an 'EDIT' link. Below the project name, it says 'Project ID: chicago-business-intelligence. It cannot be changed later.' followed by another 'EDIT' link. Under the 'Location *' section, it says 'No organization' with a 'BROWSE' button and a 'Parent organization or folder' input field. At the bottom, there are two buttons: a blue 'CREATE' button and a white 'CANCEL' button. A black banner at the very bottom displays the message 'Your current project has been set to: [chicago-business-intelligence].'

- 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"
```

The screenshot shows a Command Prompt window with the following text:

```
[2] amsds432@gmail.com
[3] amsds432@gmail.com
[4] amsds432@gmail.com
[5] Log in with a new account
Please enter your numeric choice: 4

You are logged in as: [amsds432@gmail.com].

Pick cloud project to use:
[1] neural-truth-358715
[2] solid-league-358715
[3] styling-pot-358715
[4] tensile-medium-358715
[5] Create a new project
Please enter numeric choice or text value (must exactly match list item): 4

Your current project has been set to: [tensile-medium-358715].
```

Not setting default zone/region (this feature makes it easier to use [gcloud compute] by setting an appropriate default value for the -zone and -region flag). See <https://cloud.google.com/compute/docs/gcloud-compute> section on how to set default zone/region and zone manually. If you would like [gcloud init] to be able to do this for you the next time you run it, make sure the Compute Engine API is enabled for your project on the <https://console.developers.google.com/apis> page.

Your Google Cloud SDK is configured and ready to use!

- * Commands that require authentication will use amsds432@gmail.com by default
- * Commands will reference project tensile-medium-358715 by default
- Run "gcloud help config" to learn how to change individual settings

This gcloud configuration is called [default]. You can create additional configurations if you work with multiple accounts and/or projects. Run "gcloud topic configurations" to learn more.

Some things to try next:

- * Run "gcloud --help" to see the Cloud Platform services you can interact with. And run "gcloud help COMMAND" to get help on any gcloud command.
- * Run gcloud topic --help to learn about advanced features of the SDK like arg files and output formatting

```
C:\Users\user1>gcloud sql instances create mypostgres --database-version=POSTGRES_14 --cpu=2 --memory=7680MB --region=us-central
API [sqladmin.googleapis.com] not enabled on project [753970993858]. Would you like to enable and retry (this will take a few minutes)? (y/N)? y

Enabling service [sqladmin.googleapis.com] on project [753970993858]...
Operation 'operations/acat.p2-753970993858-87e383f9-b886-4541-af05-d0243b572649' finished successfully.
Creating Cloud SQL instance...done.
Created [https://sqladmin.googleapis.com/sql/v1beta4/projects/tensile-medium-358715/instances/mypostgres].
NAME  DATABASE_VERSION  LOCATION  TIER  PRIMARY_ADDRESS  PRIVATE_ADDRESS  STATUS
mypostgres  POSTGRES_14  us-central1-f  db-custom-2-7680  35.224.24.146  -  RUNNABLE
```

C:\Users\user1>

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

```
"gcloud sql users set-password postgres --instance=mypostgres --password=root"
```

The screenshot shows a Command Prompt window with the following text:

```
C:\Users\user1>gcloud sql instances create mypostgres --database-version=POSTGRES_14 --cpu=2 --memory=7680MB --region=us-central
API [sqladmin.googleapis.com] not enabled on project [753970993858]. Would you like to enable and retry (this will take a few minutes)? (y/N)? y

Enabling service [sqladmin.googleapis.com] on project [753970993858]...
Operation 'operations/acat.p2-753970993858-87e383f9-b886-4541-af05-d0243b572649' finished successfully.
Creating Cloud SQL instance...done.
Created [https://sqladmin.googleapis.com/sql/v1beta4/projects/tensile-medium-358715/instances/mypostgres].
NAME  DATABASE_VERSION  LOCATION  TIER  PRIMARY_ADDRESS  PRIVATE_ADDRESS  STATUS
mypostgres  POSTGRES_14  us-central1-f  db-custom-2-7680  35.224.24.146  -  RUNNABLE
```

C:\Users\user1>

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

```
"gcloud sql databases create chicago_business_intelligence --instance=mypostgres"
```

The screenshot shows a Command Prompt window with the following text:

```
C:\Users\user1>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: tensile-medium-358715

C:\Users\user1>
```

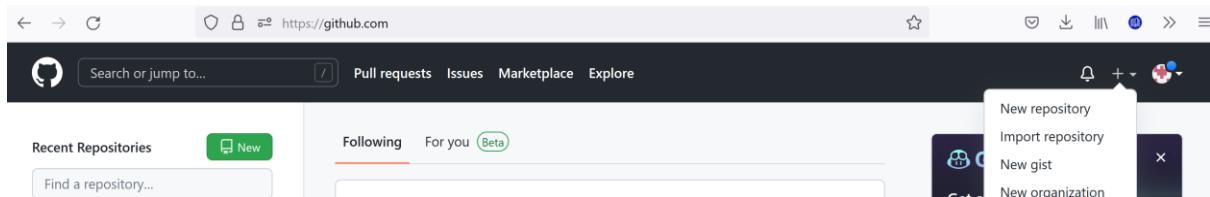
- Open Google Cloud console, search for SQL and confirm that database instance is up and running

The screenshot shows the Google Cloud SQL Instances page. The table displays the following information for the 'mypostgres' instance:

Instance ID	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Storage used
mypostgres	PostgreSQL 14	35.224.24.146		tensile-medium-358...	ADD	us-central1-f	

Step 3: Setting up continuous deployment using cloud build.

- Create a **Repository** on **GitHub** to store the source code for your CBI project.



- Open Google Cloud Console, Search for Cloud build API and Enable it for your project

A screenshot of the Google Cloud API Library. The search bar at the top contains 'cloud build api'. Below the search bar, the 'Cloud Build API' is listed under the 'Google Enterprise API' category. A tooltip for the 'Cloud Build API' says: 'Cloud Build, Google Cloud's continuous integration (CI) and continuous delivery (CD) platform, lets you build software quickly across all languages. Get complete control over defining custom workflows for building, testing, and deploying across multiple environments such as serverless, Kubernetes, or Firebase.' There are other APIs listed on the left side, such as Big data, Analytics, Databases, Developer tools, Compute, Advertising, Social, Google Enterprise APIs, Developer stacks, Google Workspace, and Other.

A screenshot of the Google Cloud Cloud Build API page. The page title is 'Cloud Build API' and it is described as 'Google Enterprise API'. Below the title, there is a brief description: 'Continuously build, test, and deploy.' Two buttons are present: 'ENABLE' and 'TRY THIS API'. At the bottom of the page, there are tabs for 'OVERVIEW', 'PRICING', and 'DOCUMENTATION'. The 'OVERVIEW' tab is selected. The 'Additional details' section indicates that the type is 'SaaS & APIs'.

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

The screenshot shows the Google Cloud Platform interface for the Cloud Build API. The URL is https://console.cloud.google.com/apis/api/cloudbuild.googleapis.com/metrics?project=tensile-medium-358715. A banner at the top indicates a free trial status with \$300.00 credit and 91 days remaining. The main content area displays 'Service Details' for Cloud Build, including tabs for Dashboard, History, Triggers, and Settings. The 'Triggers' tab is selected. It shows a dropdown menu for 'Event type' with 'Source code commit' selected. Below this are filters for 'Time range' (1 hour to 30 days), 'Credentials' (Unspecified, Anonymous, ...), and 'Methods' (52 options selected). A chart titled 'Traffic by response code' is shown, stating 'No data is available for the selected time frame.'

The screenshot shows the Google Cloud Platform interface for managing Cloud Build triggers. The URL is https://console.cloud.google.com/cloud-build/triggers?authuser=6&project=tensile-medium-358715. A banner at the top indicates a free trial status with \$300.00 credit and 91 days remaining. The main content area shows the 'Cloud Build' dashboard with a sidebar containing 'Dashboard', 'History', 'Triggers' (which is selected and highlighted in blue), and 'Settings'. The 'Triggers' section features a large circular icon with a wrench, indicating 'No triggers found in global'. Below this, text says 'Automatically start a build whenever you make changes to your source code or in response to other events.' with a link to 'Learn more'. At the bottom are two buttons: 'CREATE TRIGGER' and 'CONNECT REPOSITORY'.

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

The screenshot shows the 'Create trigger' page in the Google Cloud Platform Cloud Build interface. The left sidebar shows 'Triggers' is selected. The main form has the following fields filled:

- Name ***: cbi-trigger
- Region ***: global (non-regional)
- Description**: (empty)
- Tags**: (empty)
- Event** section:
 - Repository event that invokes trigger
 - Push to a branch
 - Push new tag
 - Pull request
Not available for Cloud Source Repositories

- Click on connect repository, select github and authenticate.

The screenshot shows the 'Create trigger' page in the Google Cloud Platform Cloud Build interface. The left sidebar shows 'Triggers' is selected. The main form has the following sections:

- Source** section:
 - Repository ***: (input field with placeholder 'Filter Type to filter')
 - CONNECT NEW REPOSITORY**
 - Invert Regex
 - SHOW INCLUDED AND IGNORED FILES FILTERS**
- Configuration** section:
 - Type**:
 - Cloud Build configuration file (yaml or json)
 - Dockerfile
 - Buildpacks

To the right, a modal window titled 'Connect repository' is open, showing the 'Select source' step. It lists several options, with 'GitHub (Cloud Build GitHub App)' selected:

- Select source**
 - GitHub (Cloud Build GitHub App)
Build source code in response to pull requests and pushes.
 - GitHub Enterprise
Build source code hosted on premises in response to pull requests and pushes.
 - Bitbucket Server
Build source code hosted on premises in response to pull requests and pushes.
 - Bitbucket Data Center
Build source code hosted on premises in response to pull requests and pushes.
 - Bitbucket Cloud (mirrored) **BETA**
Build source code in response to pushes, mirrored through Cloud Source Repositories.
- SHOW MORE**
- CONTINUE**

Below the modal, steps 2 and 3 are listed:

- 2 Authenticate
- 3 Select repository

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

The screenshot shows the 'Create trigger' page in the Google Cloud Build interface. On the left, there's a sidebar with 'Cloud Build' selected, followed by 'Dashboard', 'History', 'Triggers' (which is highlighted in blue), and 'Settings'. The main area has a title 'Create trigger' with a back arrow. Under 'Event', it says 'Repository event that invokes trigger' and has three options: 'Push to a branch' (selected with a blue radio button), 'Push new tag', and 'Pull request'. Below that, under 'Or in response to', are four options: 'Manual invocation', 'Pub/Sub message', and 'Webhook event'. Under 'Source', there's a dropdown labeled 'Repository *' which has 'Connected repository "ajb254/cbi"' selected. A tooltip below the dropdown says 'Select the repository to watch for events and clone when the trigger is invoked'. At the bottom right of the main area, there's a small modal window with an 'X' icon and the text 'Connected repository "ajb254/cbi"'.

- Select the repository after connecting the project.

This screenshot continues from the previous one, showing the 'Create trigger' page. The 'Source' section is now visible, showing the 'Repository *' dropdown with 'ajb254/cbi (GitHub App)' selected. Below it is a tooltip: 'Select the repository to watch for events and clone when the trigger is invoked'. Underneath is a 'Branch *' input field with a placeholder 'Trigger only for a branch that matches the given regular expression' and a link 'Learn more'. There's also an unchecked checkbox for 'Invert Regex'. Below this is a section titled 'SHOW INCLUDED AND IGNORED FILES FILTERS' with a downward arrow. The 'Configuration' section is partially visible at the bottom. A small modal window at the bottom right shows the connected repository 'Connected repository "ajb254/cbi"' with an 'X' icon.

- Click on create to create the trigger.

The screenshot shows the 'Create trigger' page in the Google Cloud Build interface. The left sidebar has 'Triggers' selected. The main area shows a GitHub repository 'ajb254/cbi (GitHub App)' and an 'Advanced' section with options for Approval, Build logs, and Service account. A 'Service account email' input field is present, and at the bottom are 'CREATE' and 'CANCEL' buttons.

Free trial status: \$300.00 credit and 91 days remaining - with a full account, you'll get unlimited access to all of Google Cloud Platform.

DISMISST ACTIVATE

Google Cloud Chicago-Business-Intelligence

Cloud Build Create trigger

ajb254/cbi (GitHub App)

Inline Write inline YAML

Advanced

Approval

Require approval before build executes

Build logs

Send build logs to GitHub

Service account

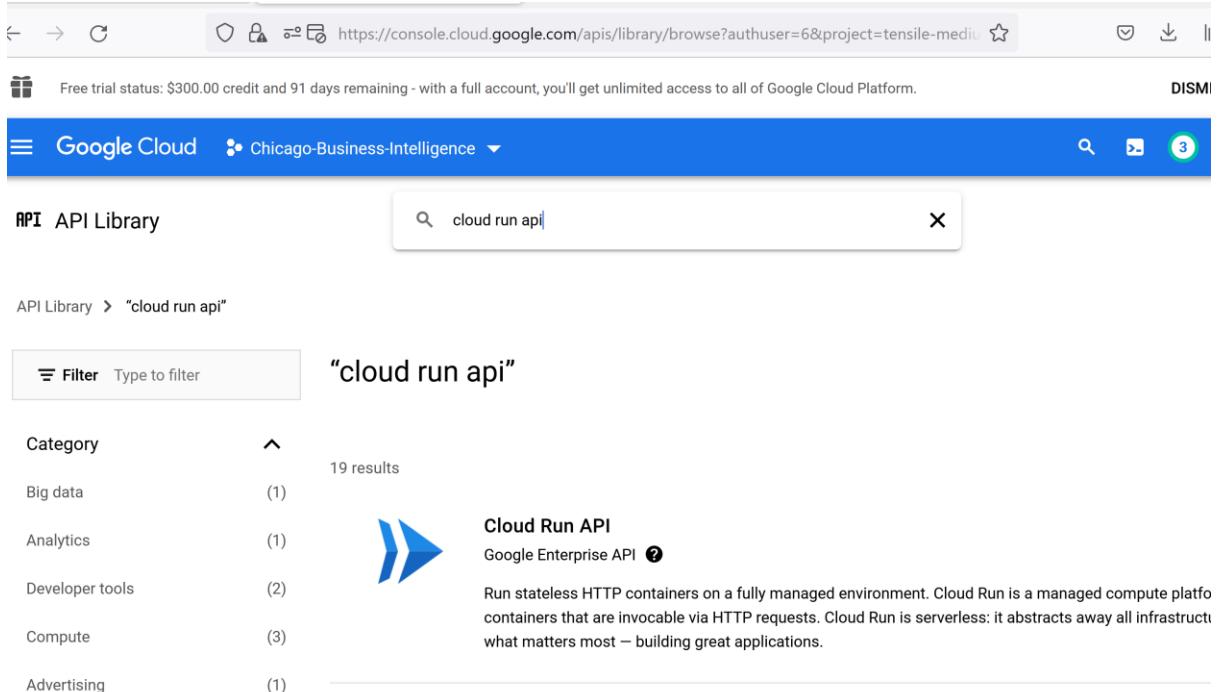
Trigger a build with the following service account [Learn more](#)

Service account email

CREATE CANCEL

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

- Enable cloud run api for your project.



The screenshot shows the Google Cloud API Library interface. A search bar at the top contains the query "cloud run api". Below the search bar, a list of categories is shown under the heading "cloud run api". The categories include Big data, Analytics, Developer tools, Compute, and Advertising, with a total of 19 results. The "Compute" category is expanded, showing the "Cloud Run API" entry. The "Cloud Run API" entry includes a description: "Run stateless HTTP containers on a fully managed environment. Cloud Run is a managed compute platform that abstracts away all infrastructure so you can focus on what matters most – building great applications." Below this, another screenshot of the Google Cloud API Library is shown, specifically for the "Cloud Run API" page. This page features a large blue "ENABLE" button and a "TRY THIS API" button. At the bottom, there are tabs for "OVERVIEW" and "DOCUMENTATION", with the "OVERVIEW" tab currently selected. The "Overview" section contains a brief description of Cloud Run and its benefits.

API Library > "cloud run api"

Filter Type to filter

Category

Big data (1)

Analytics (1)

Developer tools (2)

Compute (3)

Advertising (1)

19 results

Cloud Run API

Google Enterprise API

Run stateless HTTP containers on a fully managed environment. Cloud Run is a managed compute platform that abstracts away all infrastructure so you can focus on what matters most – building great applications.

Cloud Run API

Google Enterprise API

Serverless agility for containerized apps

ENABLE TRY THIS API

OVERVIEW DOCUMENTATION

Overview

Run stateless HTTP containers on a fully managed environment.

Additional details

Type: SaaS & APIs

Step 5: Enable IAM permissions/roles

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

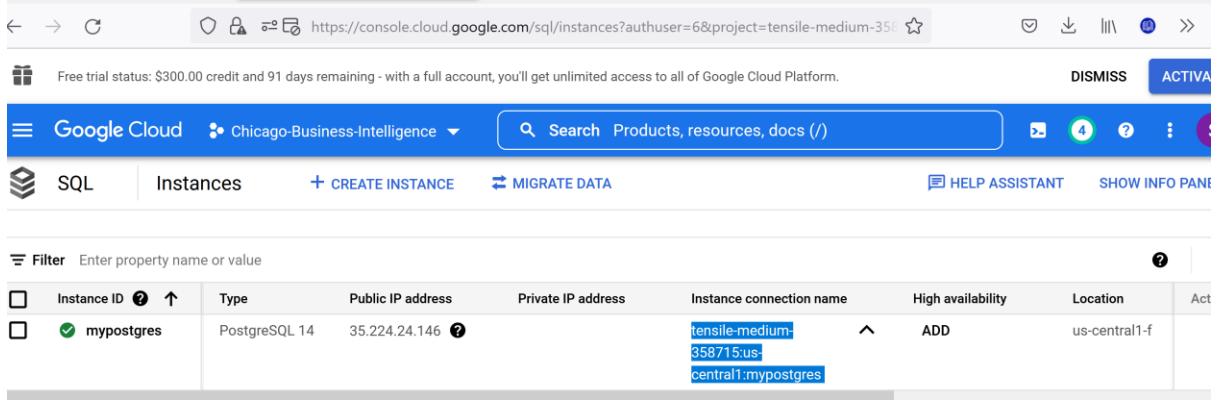
The screenshot shows the Google Cloud IAM & Admin interface. The left sidebar has 'IAM & Admin' selected. A modal window titled 'IAM' is open, listing various IAM components: Identity & Organization, Policy Troubleshooter, Policy Analyzer, Organization Policies, Service Accounts, Workload Identity Federation, Labels, Tags, Settings, Privacy & Security, Identity-Aware Proxy, and Roles. Below the modal, there are buttons for 'Create a GKE cluster' and 'Create a storage bucket'. At the bottom of the screen, there are links for 'Privacy Policy' and 'Terms of Service'.

The screenshot shows the Google Cloud IAM & Admin interface. The left sidebar has 'IAM' selected. The main area displays the 'PERMISSIONS' tab, which lists the permissions affecting the project. It includes a 'View By' dropdown set to 'PRINCIPALS', a 'Filter' input field, and a table of permissions. The table columns are 'Type', 'Principal ↑', 'Name', 'Role', and 'Security insights'. The table data includes:

Type	Principal ↑	Name	Role	Security insights
user	753970993858-compute@developer.gserviceaccount.com	Default compute service account	Editor	Edit
user	753970993858@cloudbuild.gserviceaccount.com	Cloud Build Service Account	Cloud Run Service Agent	Edit
user	amsds432e@gmail.com	Steve Bader	Owner	Edit

Step 6: Get the Postgres DB instance connection name

- 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.



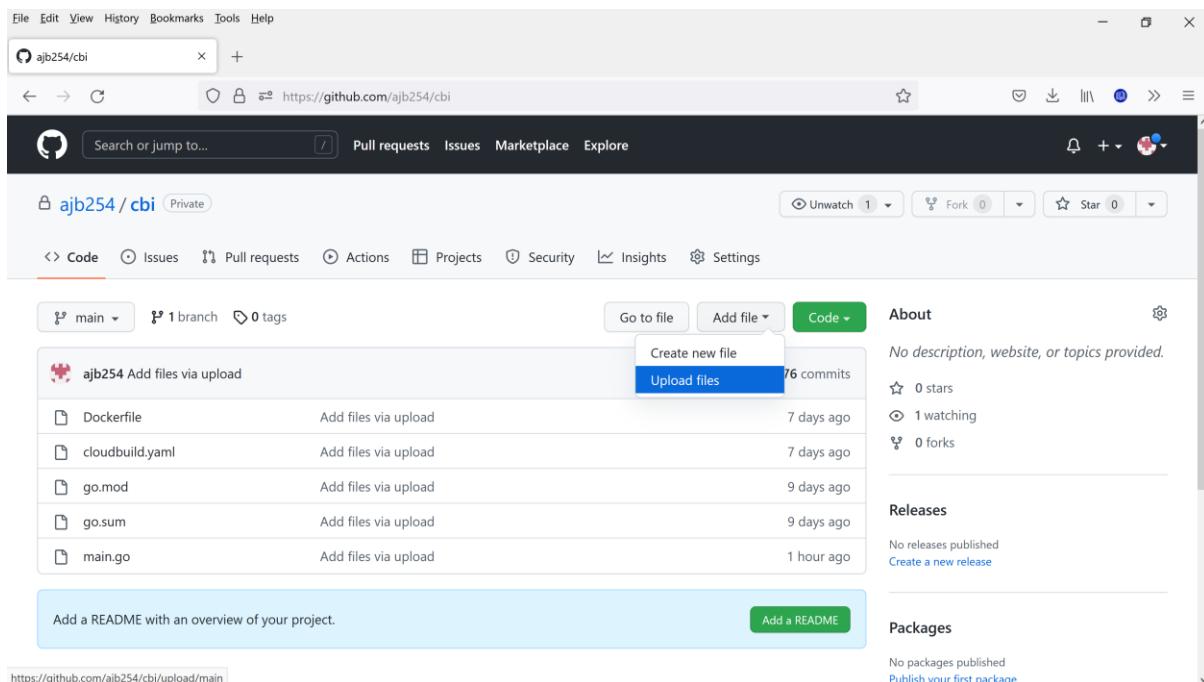
The screenshot shows the Google Cloud SQL Instances page. At the top, there's a banner about a free trial with \$300.00 credit. Below the banner, the navigation bar includes 'Google Cloud' and 'Chicago-Business-Intelligence'. The main area shows a table with one row for the instance 'mypostgres'. The table columns are: Instance ID, Type, Public IP address, Private IP address, Instance connection name, High availability, Location, and Action. The 'Instance connection name' column displays the value 'tensile-medium-358715-us-central1:mypostgres'.

Instance ID	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Action
mypostgres	PostgreSQL 14	35.224.24.146		tensile-medium-358715-us-central1:mypostgres	ADD	us-central1-f	

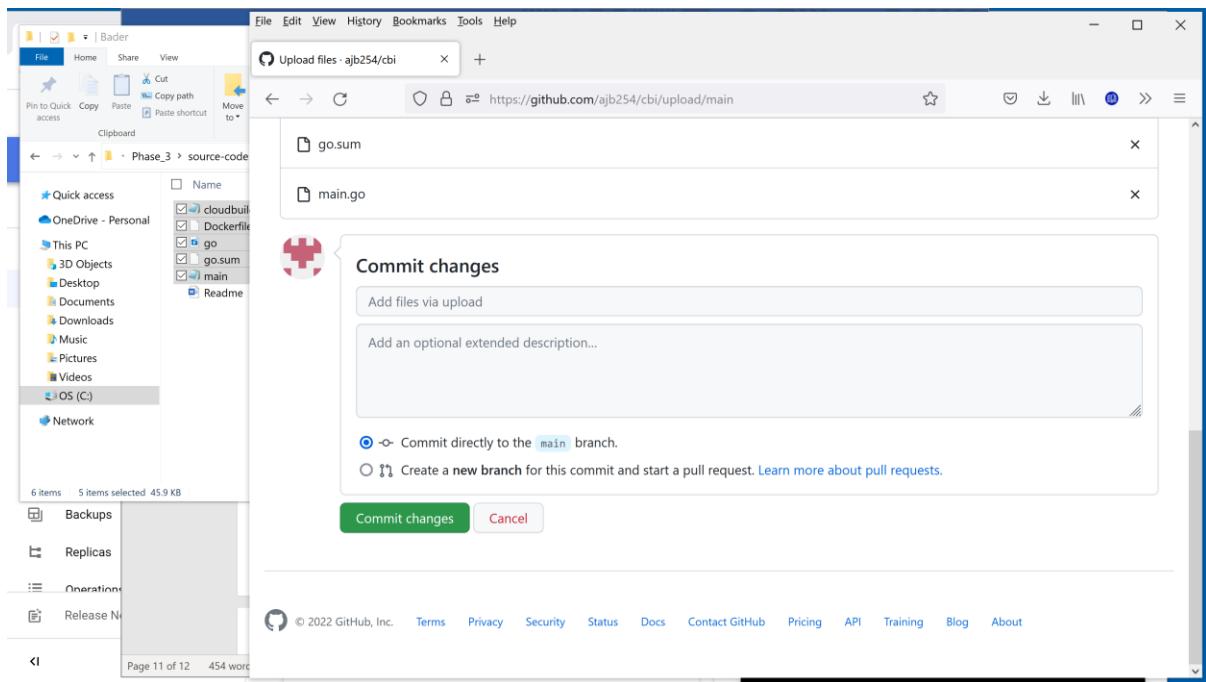
- Go to line 225 in the main.go source code file and update the connection string with your Instance connection name as shown below.

```
"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 GitHub repository created in prior steps



The screenshot shows a GitHub repository page for 'ajb254/cbi'. The repository is private. The main branch is 'main'. The commit history shows several commits, with the most recent being 'ajb254 Add files via upload' made 1 hour ago. On the right side of the page, there are sections for 'About', 'Releases', and 'Packages'. The 'About' section notes 'No description, website, or topics provided.' The 'Releases' section says 'No releases published' and has a link to 'Create a new release'. The 'Packages' section indicates 'No packages published' and has a link to 'Publish your first package'.



- A build is triggered in cloud build immediately after pushing the code to the github.

The screenshot shows the Google Cloud Platform interface for Cloud Build. The left sidebar has 'Cloud Build' selected under 'History'. The main area is titled 'Build history' with a 'STOP STREAMING BUILD'S button. A dropdown menu for 'Region' is set to 'global (non-regional)'. A 'Filter' input field is present. Below is a table of build logs:

Status	Build	Source	Ref	Commit	Trigger Name	Created	Duration
Success	a0e5b425	ajb254/cbi	main	ed748a9	cbi-trigger	8/16/22, 10:03 AM	12 sec
Success	3f050826	ajb254/cbi	main	ba1e616	cbi-trigger	8/16/22, 8:33 AM	3 min
Error	9f5dd60b	ajb254/cbi	main	9e27dbc	cbi-trigger	8/16/22, 8:07 AM	2 min 41 sec
Success	74233dd2	ajb254/cbi	main	4461c5f	cbi-trigger	8/16/22, 7:28 AM	2 min 56 sec
Error	6eca790e	ajb254/cbi	main	cdf20d9	cbi-trigger	8/16/22, 7:15 AM	2 min 56 sec
Error	af479fb4	ajb254/cbi	main	cdf20d9	cbi-trigger	8/16/22, 7:10 AM	2 min 28 sec
Success	1834a2df	ajb254/cbi	main	9ce0dae	cbi-trigger	8/14/22, 10:23 PM	2 min 29 sec
Success	1717a407	ajb254/cbi	main	a877797	cbi-trigger	8/14/22, 9:48 PM	2 min 47 sec
Success	9a240f71	ajb254/cbi	main	2294b58	cbi-trigger	8/14/22, 8:43 PM	3 min 23 sec
Success	c34c5196	ajb254/cbi	main	a17eff4	cbi-trigger	8/14/22, 8:12 PM	2 min 56 sec
Success	de2960c6	ajb254/cbi	main	686307e	cbi-trigger	8/14/22, 2:34 PM	2 min 51 sec

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

This screenshot is identical to the one above, showing the Google Cloud Platform Cloud Build History page. The build log for the most recent successful build (id: de2960c6) has been expanded to show its detailed logs. The expanded log shows the following steps:

```

build step 0/1
  step name: 'npm install'
  status: SUCCESS
  duration: 0m 0s
  logs:
    npm WARN optional SKIPPING OPTIONAL DEPENDENCY: fsevents@2.3.2 (node_modules/fsevents):
    npm WARN not found
    ...
    npm info lifecycle tensile@0.0.1~install: tensile@0.0.1
    ...
  
```

The screenshot shows the Google Cloud Build details page for a build named 'a0e5b425'. The build was triggered by 'cbl-trigger' and is associated with the source 'ajb254/cbi' and branch 'main'. It was committed at 'ed748a9'. The build summary indicates 7 steps completed successfully in 0:01:44. The build log shows the command history, including pulling dependencies and pushing artifacts. The execution details and build artifacts sections are also visible.

- Go to Cloud Run and Verify you see your services are up and running (green)

The screenshot shows the Google Cloud Run Services page. It lists two services: 'go-microservice' and 'pg-admin'. Both services are currently running in the 'us-central1' region with 0 requests per second. They both have 'Allow unauthenticated' authentication and 'All' ingress. The security status for both is 'SECURITY'. On the right side of the screen, there is a message stating 'No services selected' and a note: 'Please select at least one resource.'

- From Cloud Run, click on gomicroservice app, copy the highlighted URL

The screenshot shows the Google Cloud Platform interface for a Cloud Run service named 'go-microservice'. The 'Logs' tab is active, displaying log entries from the service. A tooltip 'Copied' is shown above the URL field, indicating the URL has been copied. The URL listed is <https://go-microservice-3zznh4hkwq-uc.a.run.app>.

- Open the go-microservice URL in a new browser window and you should see the goroutines for the microservices started

The screenshot shows a browser window with the URL <https://go-microservice-3zznh4hkwq-uc.a.run.app>. The page content displays the message: "CBI data collection microservices' goroutines have started for CBI-Project!"

- From Cloud Run, click on pgadmin, copy the highlighted URL

The screenshot shows the Google Cloud Platform interface for a Cloud Run service named 'pg-admin'. The URL of the service is displayed as <https://pg-admin-3zznh4hkwq-uc.a.run.app>. A yellow box highlights this URL.

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

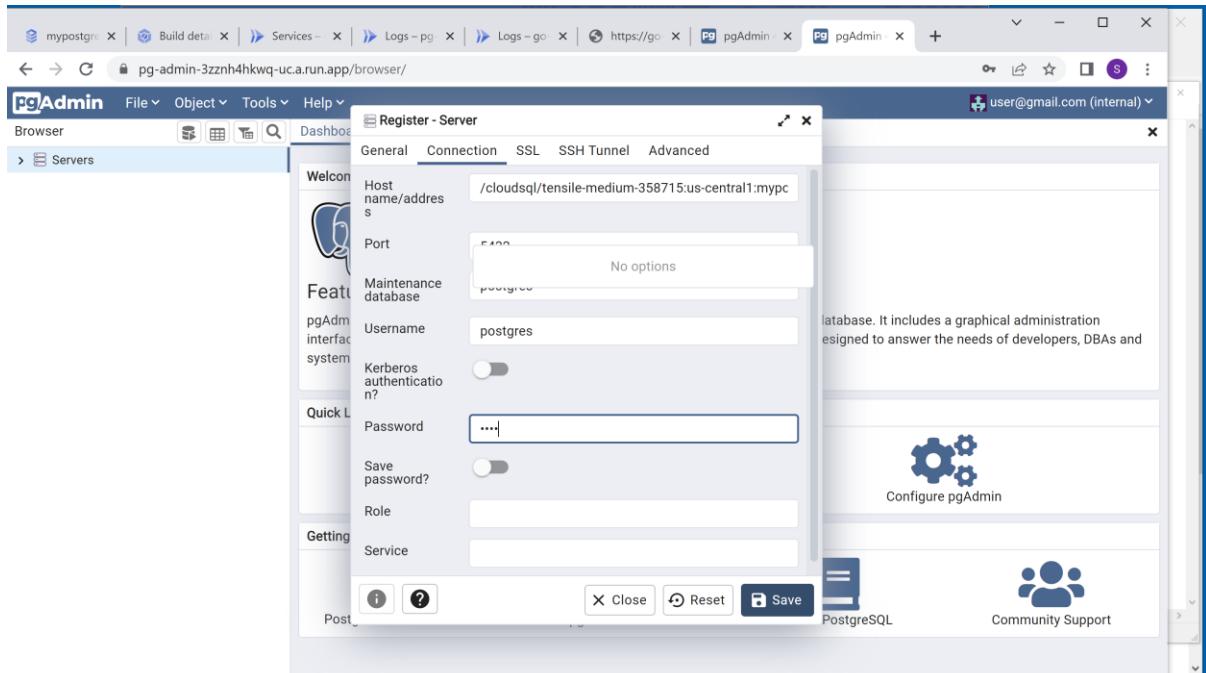
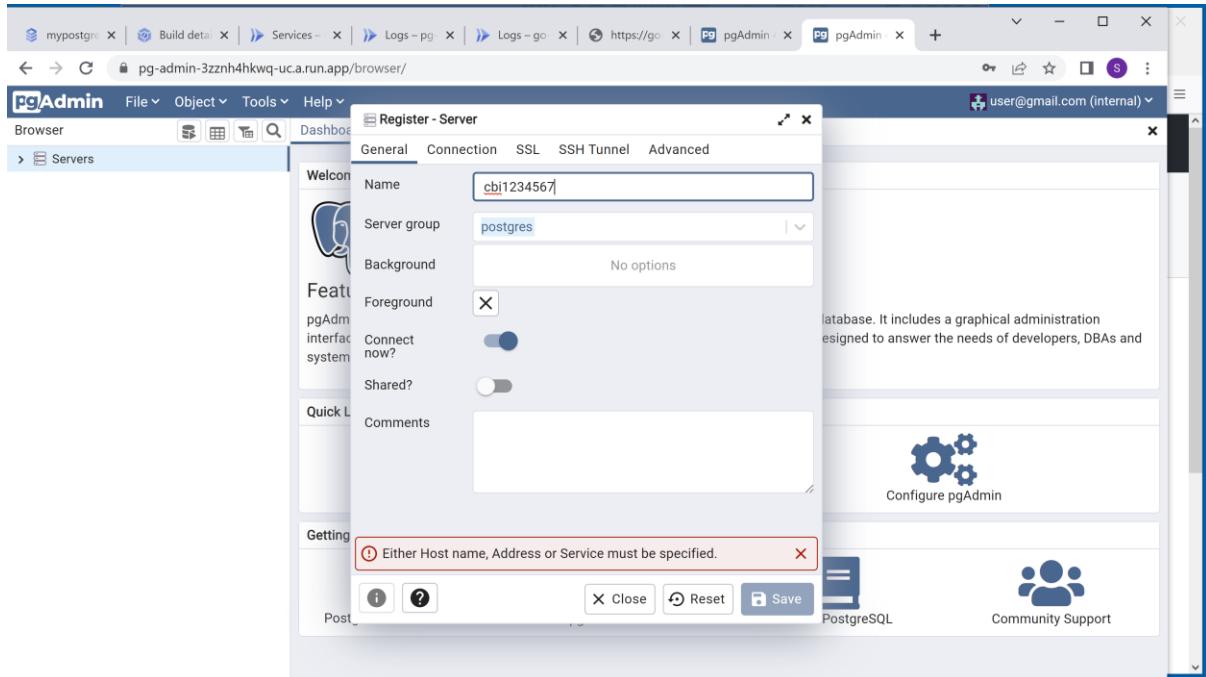
The screenshot shows a web browser displaying the pgAdmin 4 login interface. The URL in the address bar is <https://pg-admin-3zznh4hkwq-uc.a.run.app/login?next=%2F>. The pgAdmin logo and 'Login' button are visible on the right, while a world map graphic is on the left.

- Add a server

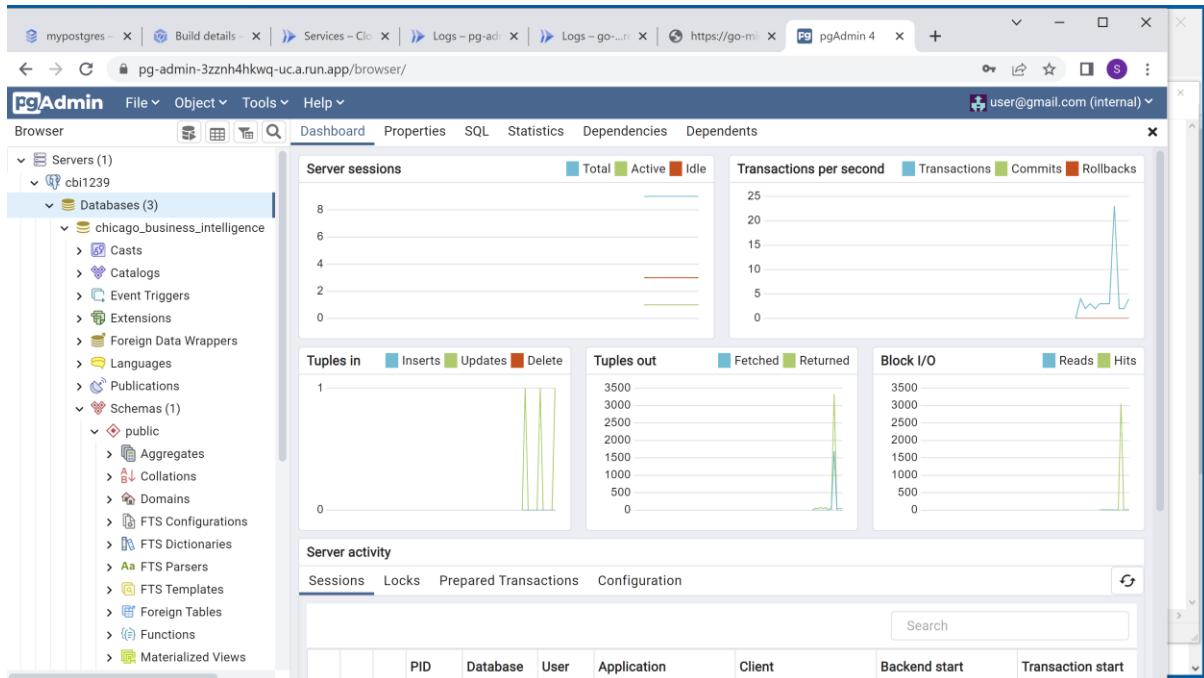
The screenshot shows the pgAdmin 4 welcome page. At the top, there's a navigation bar with tabs for Dashboard, Properties, SQL, Statistics, Dependencies, and Dependents. Below the navigation bar is a "Welcome" section featuring the pgAdmin logo and the text "Management Tools for PostgreSQL". It also includes the tagline "Feature rich | Maximises PostgreSQL | Open Source" and a brief description of what pgAdmin is. A "Quick Links" section contains icons for "Add New Server" (a server icon), "Configure pgAdmin" (two interlocking gears), "PostgreSQL Documentation" (a book icon), "pgAdmin Website" (a globe icon), "Planet PostgreSQL" (a document icon), and "Community Support" (three people icon). On the left side, there's a sidebar with a "Servers" section containing options like "Register", "Create", "Refresh...", "Remove Server Group", and "Properties...".

This screenshot is similar to the one above, but the "Server..." option under the "Register" menu in the sidebar is highlighted with a blue selection bar. The rest of the interface is identical to the first screenshot, showing the pgAdmin welcome page with its various links and tools.

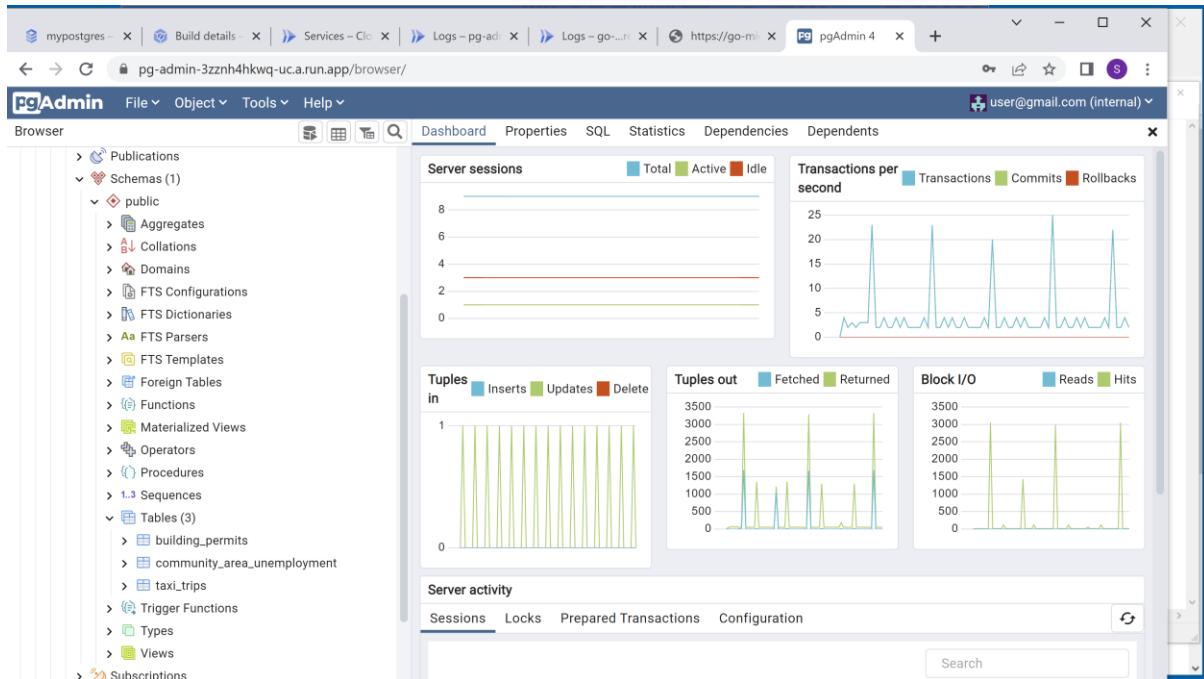
- Enter server name , host name, login/password in the dialog, click SAVE button



- After you login, click on Chicago_business intelligence



- Click on schemas/tables and verify you see the CBI tables



- Go to g-microservices and verify at least one of these tables got rows inserted into it

The screenshot shows the Google Cloud Platform interface for a Cloud Run service named "go-microservice". The service is located in the "Chicago-Business-Intelligence" project and is deployed to the "us-central1" region. The URL for the service is <https://go-microservice-3zznh4hkwq-uc.a.run.app>. The "LOGS" tab is selected, displaying 74 log entries. The logs show various database insertions into tables like "Community_Areas", "Building Permits", and "community_area_unemployment". A filter dropdown is set to "Default".

Log Entry	Message
2022-08-16T15:11:26.083287Z	Community Areas number of SODA records received = 77
2022-08-16T15:11:26.083296Z	
2022-08-16T15:11:30.988823Z	
2022-08-16T15:11:30.988841Z	
2022-08-16T15:11:30.988854Z	Building Permits: number of SODA records received = 580
2022-08-16T15:11:30.988862Z	
2022-08-16T15:11:39.779719Z	Completed Inserting Rows into the community_area_unemployment Table
2022-08-16T15:11:47.795527Z	Completed Inserting Rows into the Building Permits Table

- Go back to PG-Admin and select one of these tables and select view all rows:

The screenshot shows the pgAdmin 4 interface with the 'Dashboard' tab selected. On the left, the 'Browser' pane lists various database objects under 'Tables (3)'. The 'building_permit' table is currently selected. A context menu is open over the table, with 'View/Edit Data' expanded, showing 'All Rows' as the selected option.

The screenshot shows the pgAdmin 4 interface with the 'public.building_permits/chicago_business_intelligence/postgre...' connection selected. The 'Data output' tab is active, displaying the results of the query 'SELECT * FROM public.building_permits ORDER BY id ASC'. The results are shown in a table with the following data:

	id [PK] integer	permit_id character varying (255)	permit_code character varying (255)	permit_type character varying (255)	review_type character varying (255)	application_start_date character varying (255)	iss ch
1	1	3150374	100895747	PERMIT - ELECTRIC WI...	EASY PERMIT WEB	2020-10-15T00:00:00.0...	2C
2	2	3218940	100945227	PERMIT - NEW CONST...	SELF CERT	2021-12-22T00:00:00.0...	2C
3	3	3230645	100953233	PERMIT - EASY PERMI...	EASY PERMIT	2021-12-22T00:00:00.0...	2C
4	4	3177798	100915921	PERMIT - RENOVATION...	STANDARD PLAN REV...	2021-05-21T00:00:00.0...	2C

Total rows: 134 of 134 Query complete 00:00:00.359 Ln 1, Col 1