Software Modelling Dev with UML

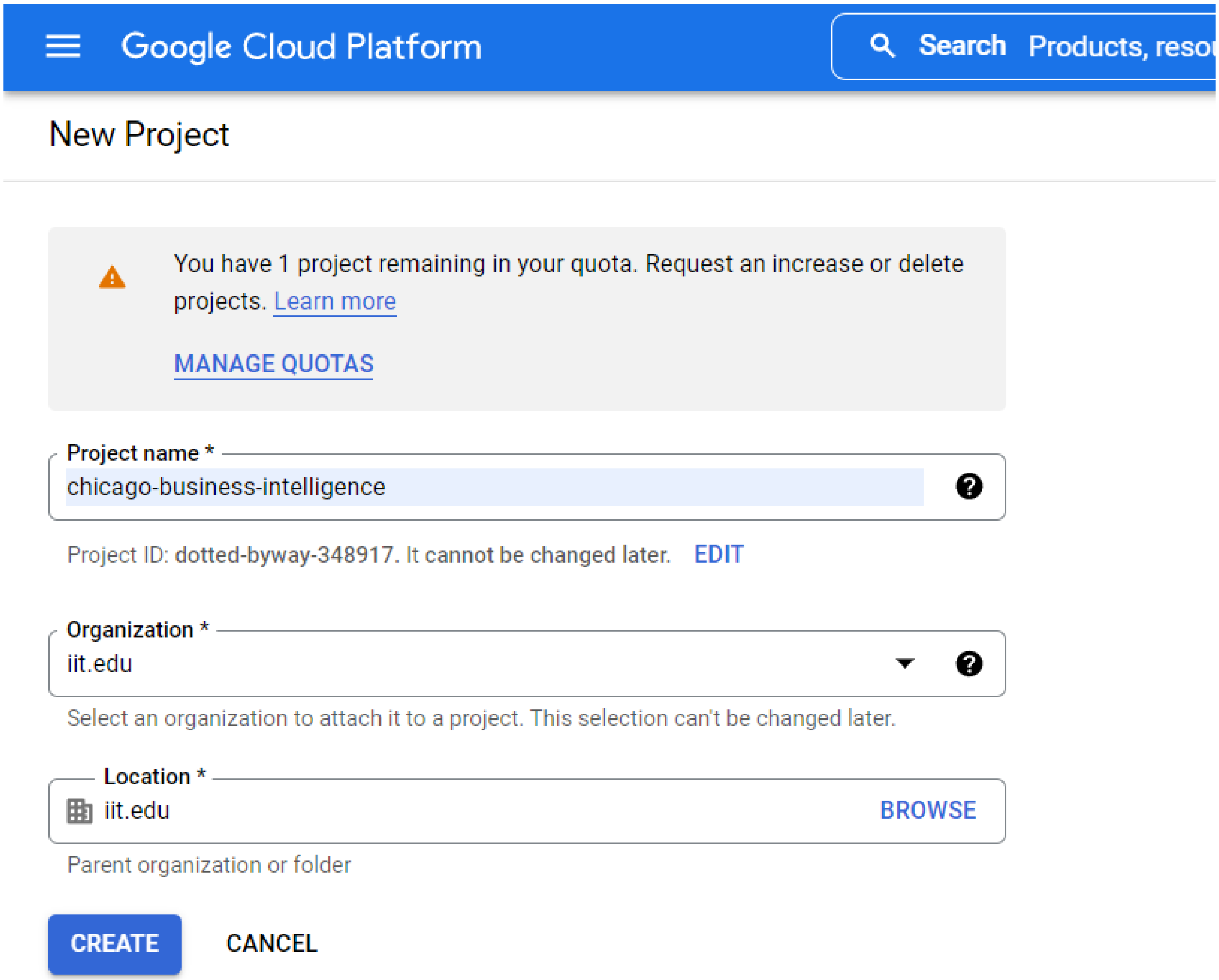
Assignment-V Part III

Divyeshkumar Patel (A20495602)

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.



You have 1 project remaining in your quota. Request an increase or delete projects. Learn more

Project name * chicago-business-intelligence

Project ID: dotted-byway-348917. It cannot be changed later. EDIT

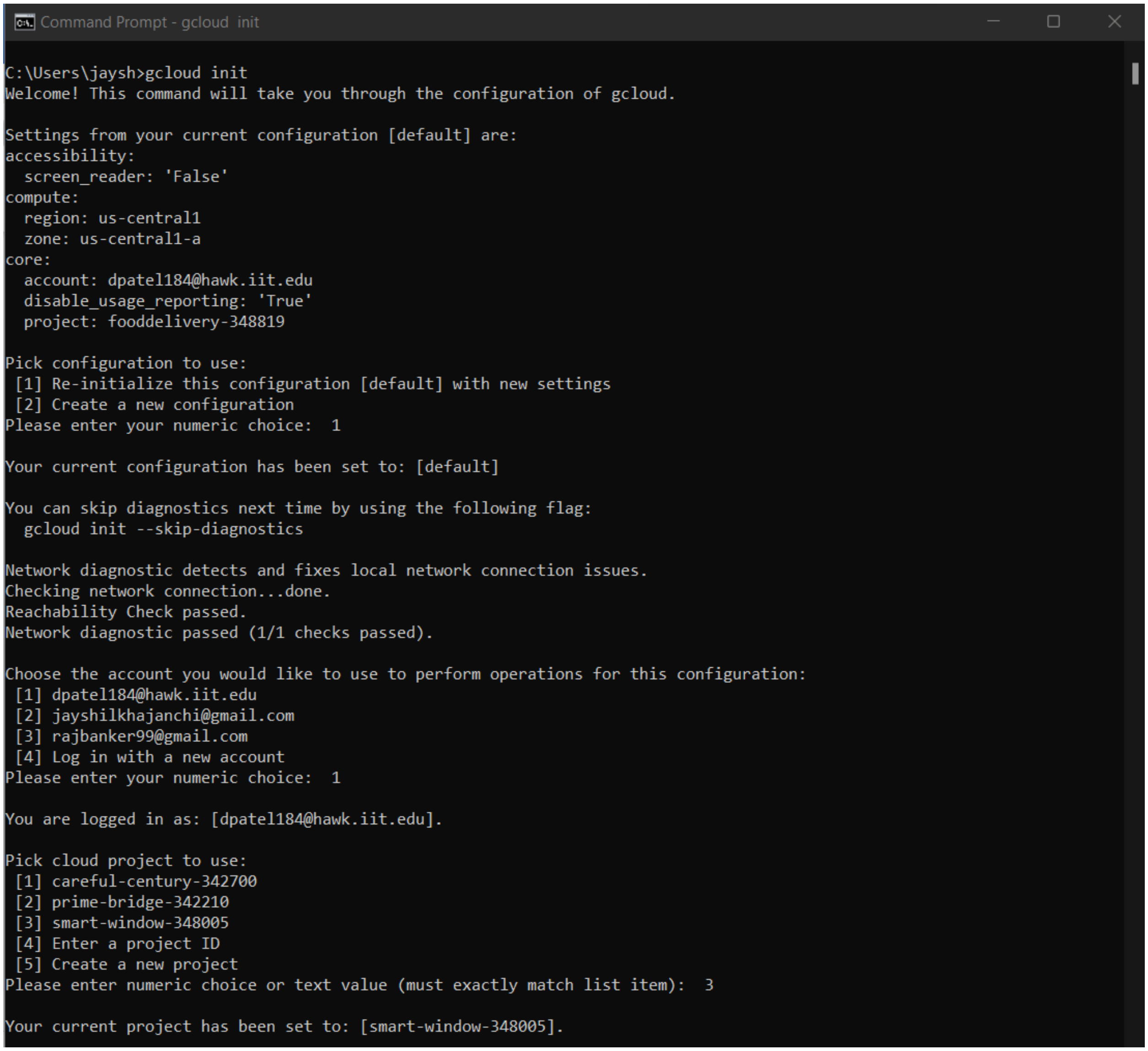
Organization * iit.edu

Select an organization to attach it to a project. This selection can't be changed later.

Location * iit.edu

Parent organization or folder

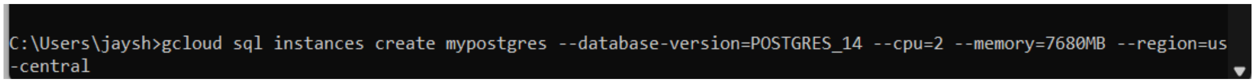
After creating a project on Google Cloud Console execute “gcloud init” command on your local machine and select the project created above when prompted



```
C:\Users\jaysh>gcloud init
Welcome! This command will take you through the configuration of gcloud.

Settings from your current configuration [default] are:
accessibility:
  screen_reader: 'False'
compute:
  region: us-central1
  zone: us-central1-a
core:
  account: dpatel184@hawk.iit.edu
  disable_usage_reporting: 'True'
  project: fooddelivery-348819

Pick configuration to use:
 [1] Re-initialize this configuration [default] with new settings
 [2] Create a new configuration
Please enter your numeric choice:  1

Your current configuration has been set to: [default]

You can skip diagnostics next time by using the following flag:
  gcloud init --skip-diagnostics

Network diagnostic detects and fixes local network connection issues.
Checking network connection...done.
Reachability Check passed.
Network diagnostic passed (1/1 checks passed).

Choose the account you would like to use to perform operations for this configuration:
 [1] dpatel184@hawk.iit.edu
 [2] jayshilkhajanchi@gmail.com
 [3] rajbanker99@gmail.com
 [4] Log in with a new account
Please enter your numeric choice:  1

You are logged in as: [dpatel184@hawk.iit.edu].

Pick cloud project to use:
 [1] careful-century-342700
 [2] prime-bridge-342210
 [3] smart-window-348005
 [4] Enter a project ID
 [5] Create a new project
Please enter numeric choice or text value (must exactly match list item):  3

Your current project has been set to: [smart-window-348005].
```

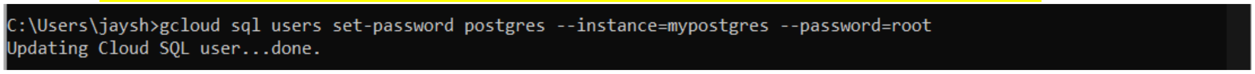
Step 2: Postgres database Setup

- 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



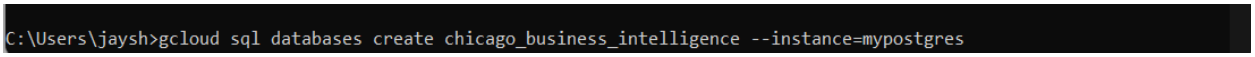
```
C:\Users\jaysh>gcloud sql instances create mypostgres --database-version=POSTGRES_14 --cpu=2 --memory=7680MB --region=us
-central
```

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



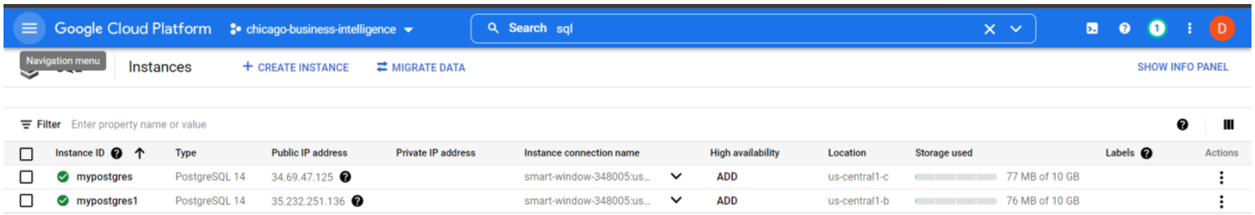
```
C:\Users\jaysh>gcloud sql users set-password postgres --instance=mypostgres --password=root
Updating Cloud SQL user...done.
```

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



```
C:\Users\jaysh>gcloud sql databases create chicago_business_intelligence --instance=mypostgres
```

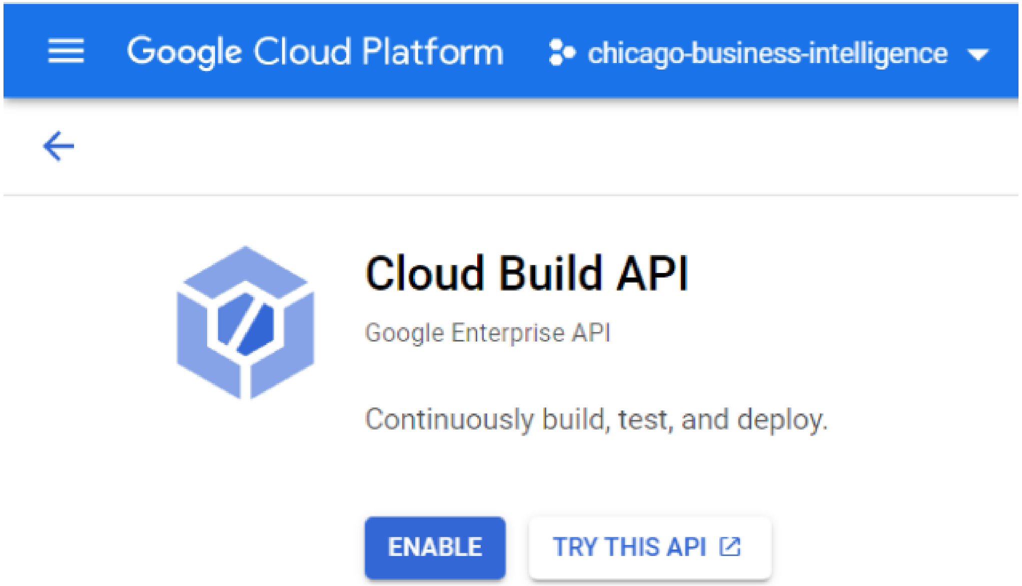
- Open Google Cloud console, search for SQL and confirm that database instance is up and running



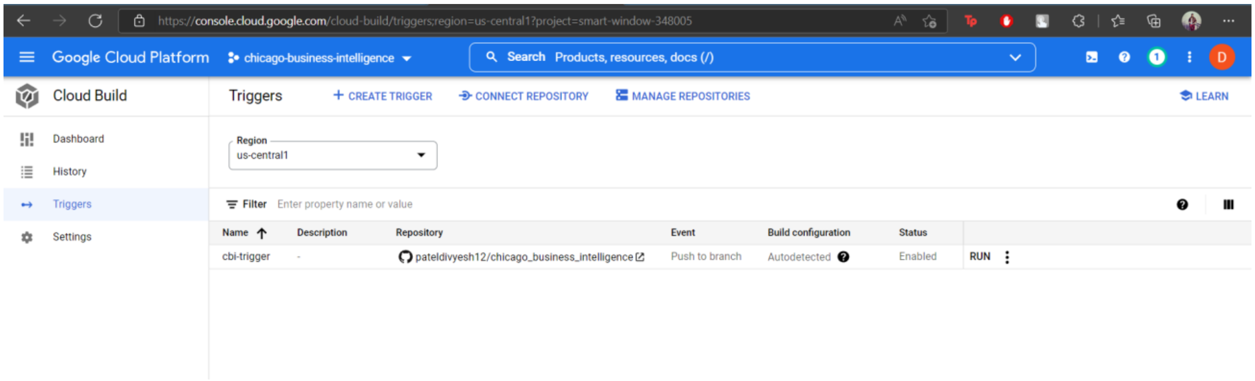
☐	Instance ID	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Storage used	Labels	Actions
☐	✓ mypostgres	PostgreSQL 14	34.69.47.125		smart-window-348005:us...	ADD	us-central1-c	77 MB of 10 GB		⋮
☐	✓ mypostgres1	PostgreSQL 14	35.232.251.136		smart-window-348005:us...	ADD	us-central1-b	76 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



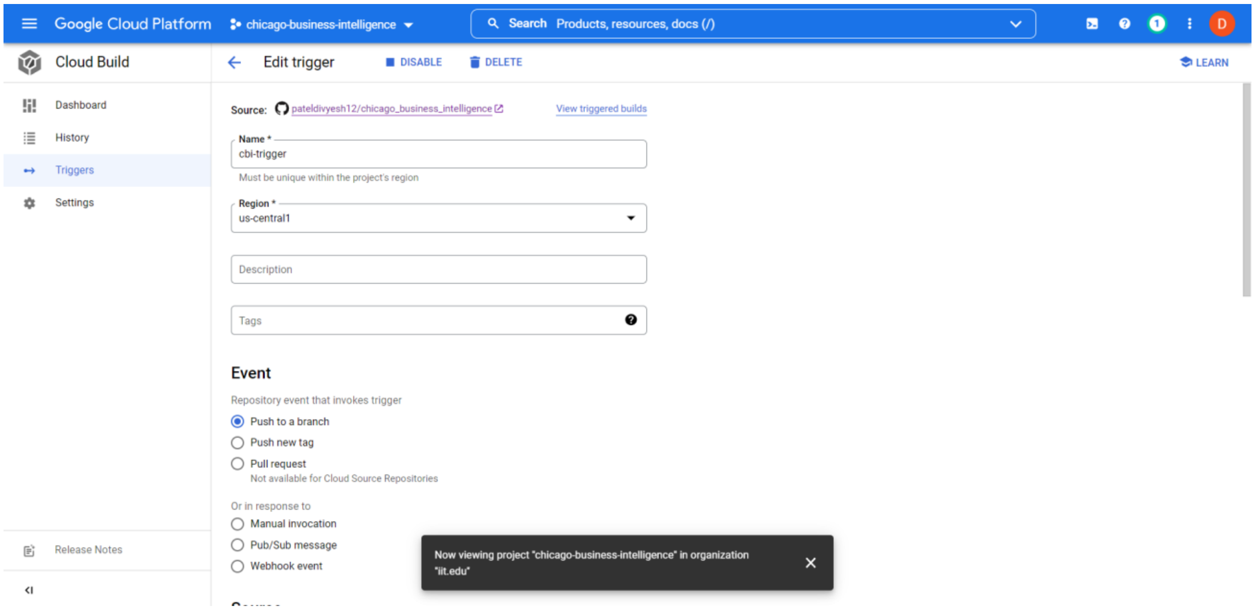
- After the API is enabled, click on the create trigger button.


https://console.cloud.google.com/cloud-build/triggers;region=us-central1?project=smart-window-348005
Google Cloud Platform
chicago-business-intelligence
Search Products, resources, docs (/)
Cloud Build
Triggers
+ CREATE TRIGGER
CONNECT REPOSITORY
MANAGE REPOSITORIES
LEARN
Dashboard
History
Triggers
Settings
Filter Enter property name or value

Region: us-central1

Name	Description	Repository	Event	Build configuration	Status	
cbi-trigger	-	pateldivyesh12/chicago_business_intelligence	Push to branch	Autodetected	Enabled	RUN ⋮

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



Source: pateldivyesh12/chicago_business_intelligence

Name *
cbi-trigger
Must be unique within the project's region

Region *
us-central1

Description

Tags

Event

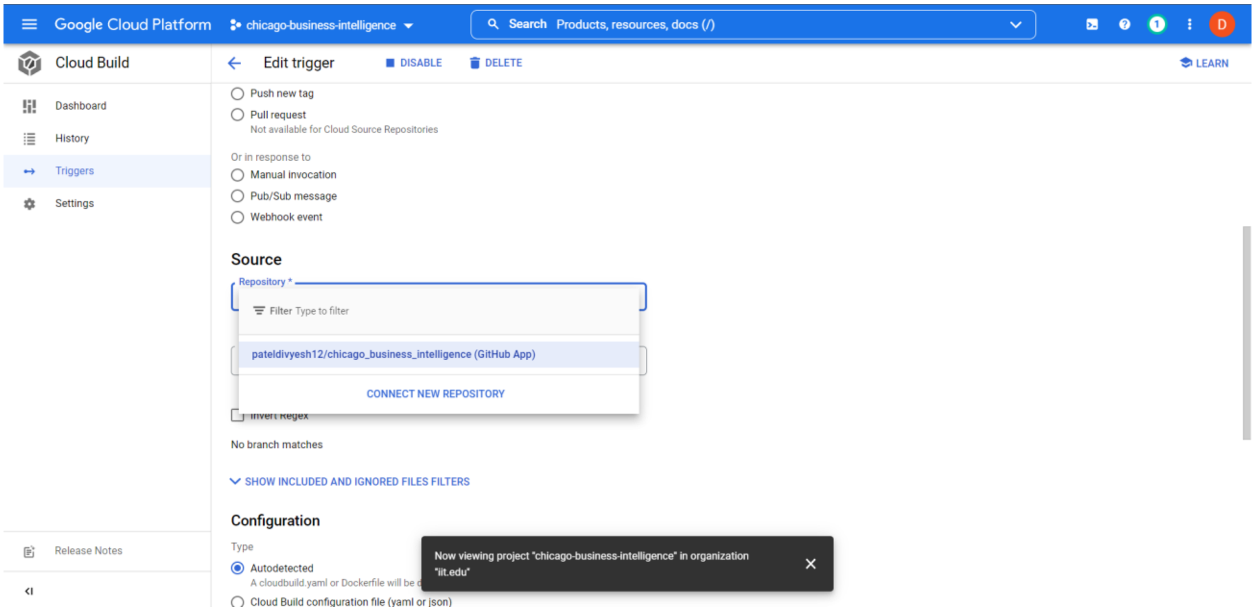
Repository event that invokes trigger

● Push to a branch
○ Push new tag
○ Pull request
Not available for Cloud Source Repositories

Or in response to
○ Manual invocation
○ Pub/Sub message
○ Webhook event

Now viewing project "chicago-business-intelligence" in organization "iit.edu"

- Click on connect repository, select github and authenticate.



○ Push new tag
○ Pull request
Not available for Cloud Source Repositories

Or in response to
○ Manual invocation
○ Pub/Sub message
○ Webhook event

Source

Repository *
Filter Type to filter
pateldivyesh12/chicago_business_intelligence (GitHub App)

□ Invert Regex

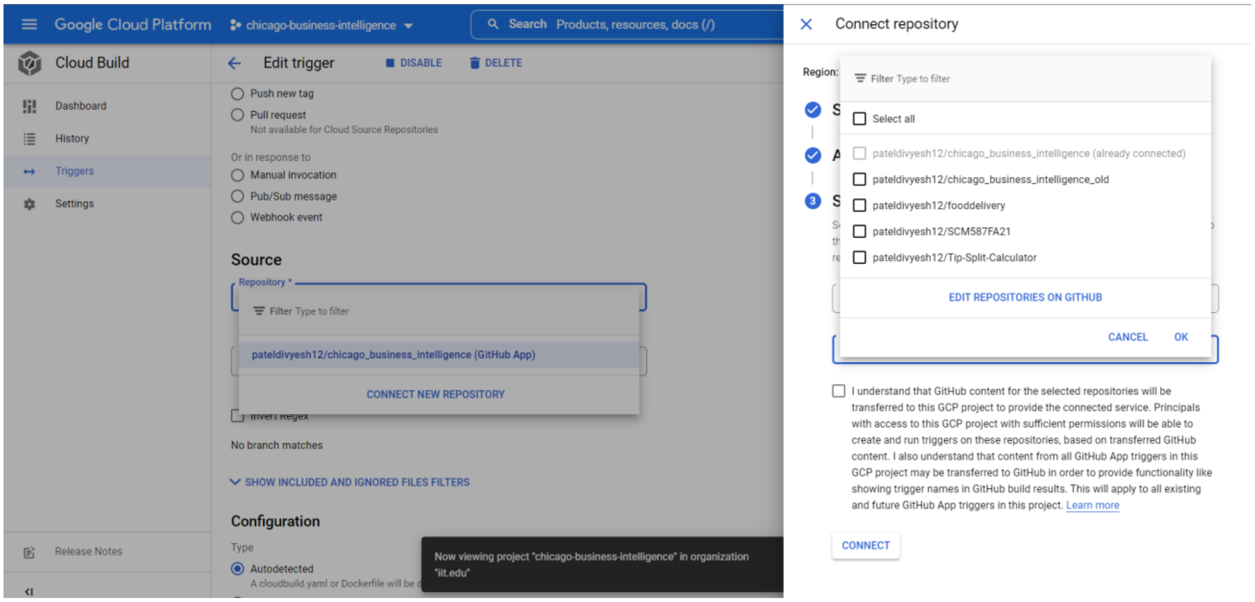
No branch matches

Configuration

Type
● Autodetected
A cloudbuild.yaml or Dockerfile will be d
○ Cloud Build configuration file (yaml or json)

Now viewing project "chicago-business-intelligence" in organization "iit.edu"

- After authentication select the repository created for Chicago business intelligence



○ Push new tag
○ Pull request
Not available for Cloud Source Repositories

Or in response to
○ Manual invocation
○ Pub/Sub message
○ Webhook event

Source

Repository *
Filter Type to filter
pateldivyesh12/chicago_business_intelligence (GitHub App)
CONNECT NEW REPOSITORY

□ Invert Regex

No branch matches

SHOW INCLUDED AND IGNORED FILES FILTERS

Configuration

Type
● Autodetected
A cloudbuild.yaml or Dockerfile will be d

Now viewing project "chicago-business-intelligence" in organization "iit.edu"

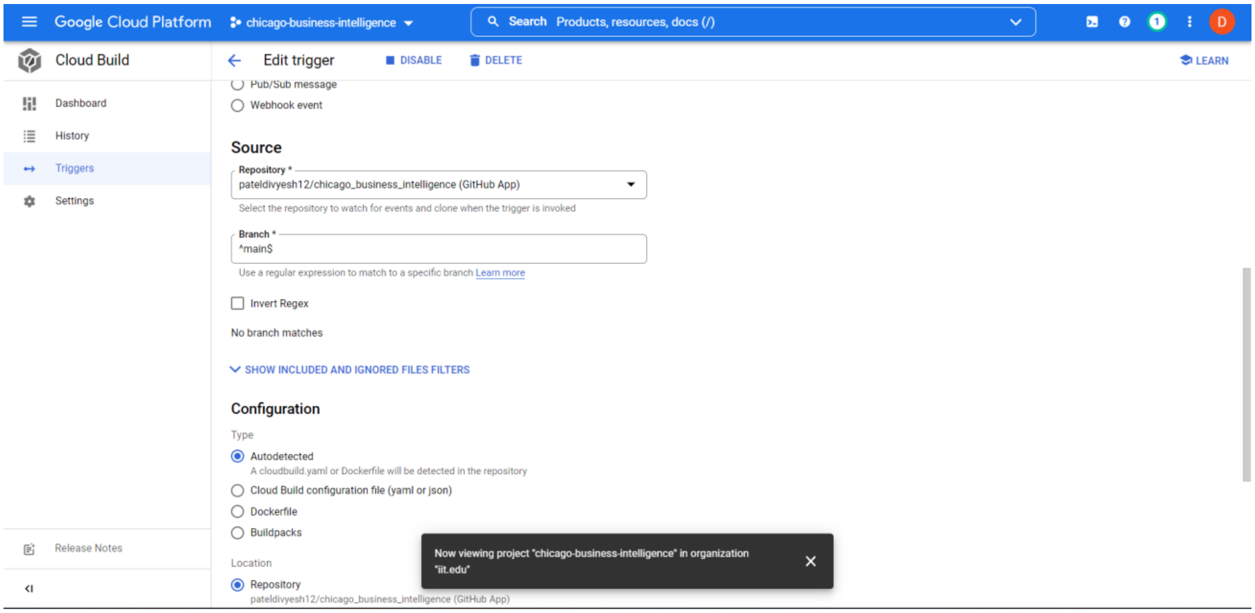
Connect repository

Region:
Filter Type to filter
□ Select all
□ pateldivyesh12/chicago_business_intelligence (already connected)
□ pateldivyesh12/chicago_business_intelligence_old
□ pateldivyesh12/fooddelivery
□ pateldivyesh12/SCM587FA21
□ pateldivyesh12/Tip-Split-Calculator
EDIT REPOSITORIES ON GITHUB
CANCEL OK

□ 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



○ Pub/Sub message
○ Webhook event

Source

Repository *
pateldivyesh12/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

No branch matches

SHOW INCLUDED AND IGNORED FILES FILTERS

Configuration

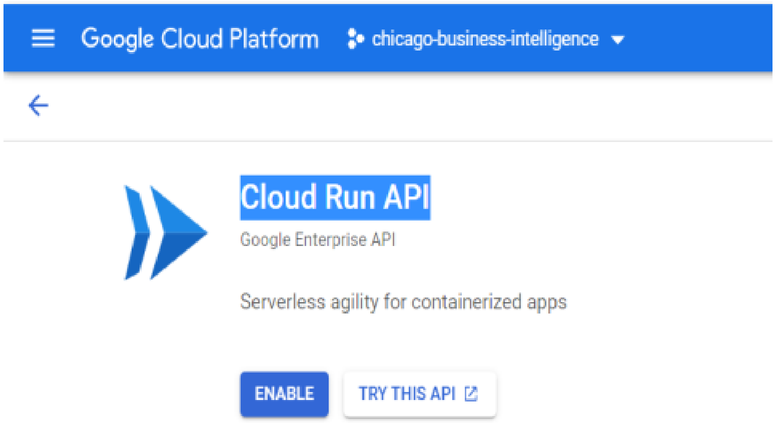
Type
● Autodetected
A cloudbuild.yaml or Dockerfile will be detected in the repository
○ Cloud Build configuration file (yaml or json)
○ Dockerfile
○ Buildpacks

Location
● Repository
pateldivyesh12/chicago_business_intelligence (GitHub App)

Now viewing project "chicago-business-intelligence" in organization "iit.edu"

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

- Enable cloud run api for your project.

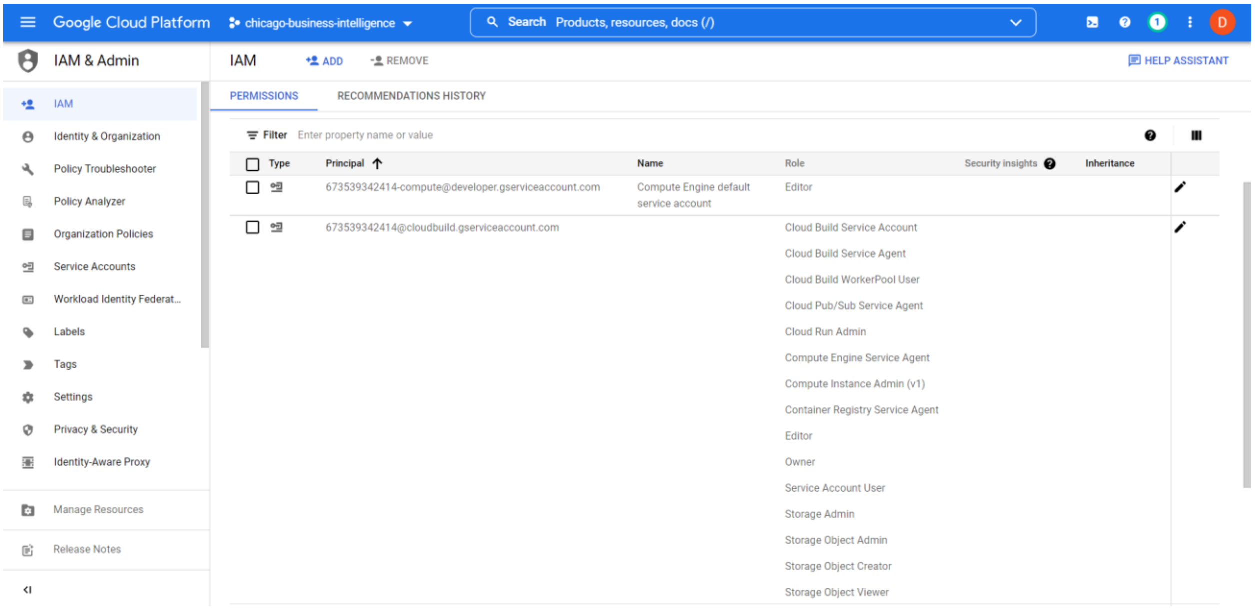


Cloud Run API

Google Enterprise API

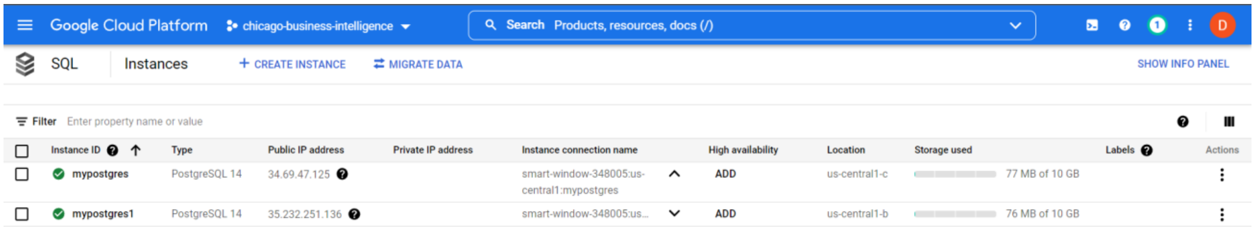
Serverless agility for containerized apps

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



☐	Type	Principal ↑	Name	Role	Security insights	Inheritance
☐		673539342414-compute@developer.gserviceaccount.com	Compute Engine default service account	Editor		
☐		673539342414@cloudbuild.gserviceaccount.com		Cloud Build Service Account Cloud Build Service Agent Cloud Build WorkerPool User Cloud Pub/Sub Service Agent Cloud Run Admin Compute Engine Service Agent Compute Instance Admin (v1) Container Registry Service Agent Editor Owner Service Account User Storage Admin Storage Object Admin Storage Object Creator Storage Object Viewer		

- 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.



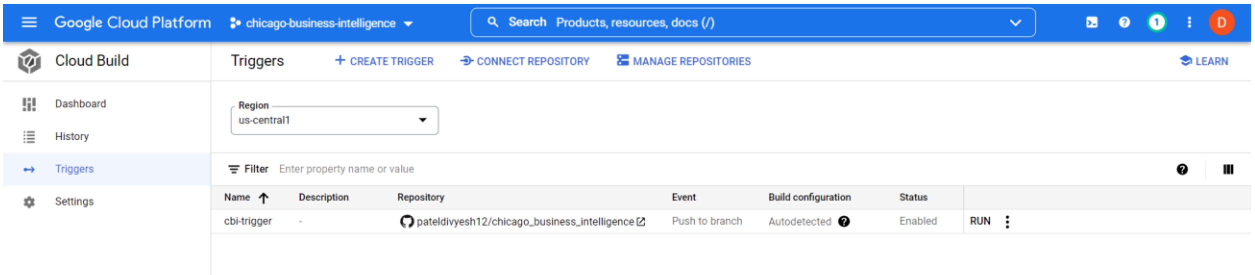
Google Cloud Platform
chicago-business-intelligence
Search Products, resources, docs (/)
SQL
Instances
CREATE INSTANCE
MIGRATE DATA
SHOW INFO PANEL
Filter Enter property name or value

☐	Instance ID ↑	Type	Public IP address	Private IP address	Instance connection name	High availability	Location	Storage used	Labels	Actions
☐	mypostgres	PostgreSQL 14	34.69.47.125		smart-window-348005:us-central1:mypostgres	ADD	us-central1-c	77 MB of 10 GB		⋮
☐	mypostgres1	PostgreSQL 14	35.232.251.136		smart-window-348005:us...	ADD	us-central1-b	76 MB of 10 GB		⋮

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

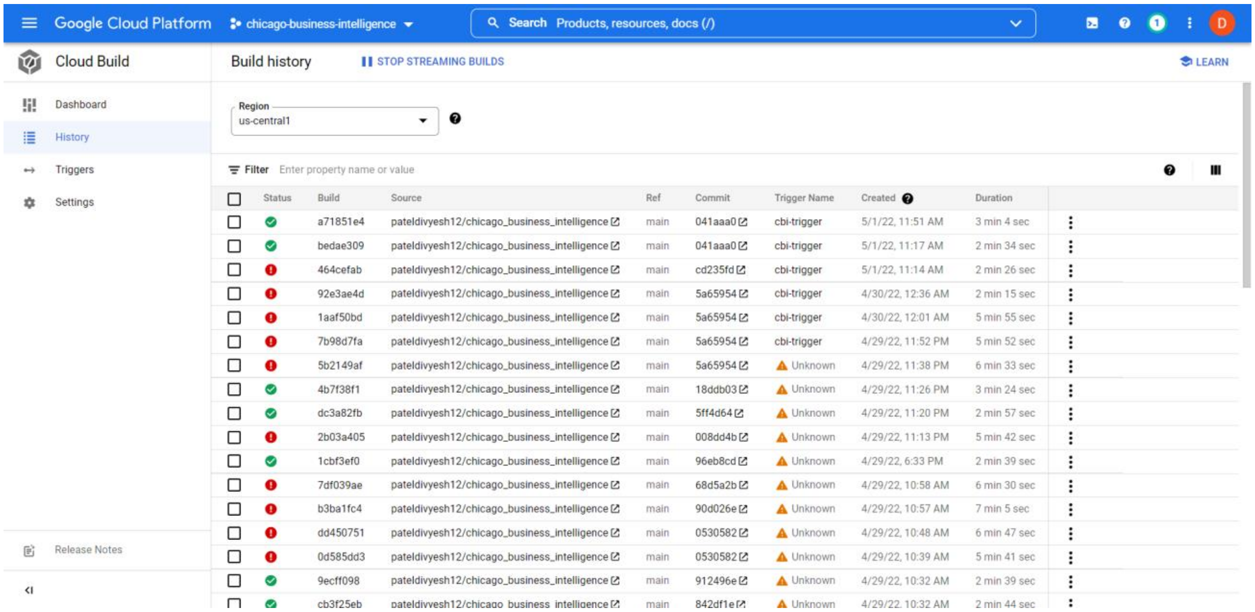
user=postgres dbname=chicago_business_intelligence password=root host=/cloudsql/smart-window-348005:us-central1:mypostgres:us-central1:mypostgres sslmode=disable port = 5432

- 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.



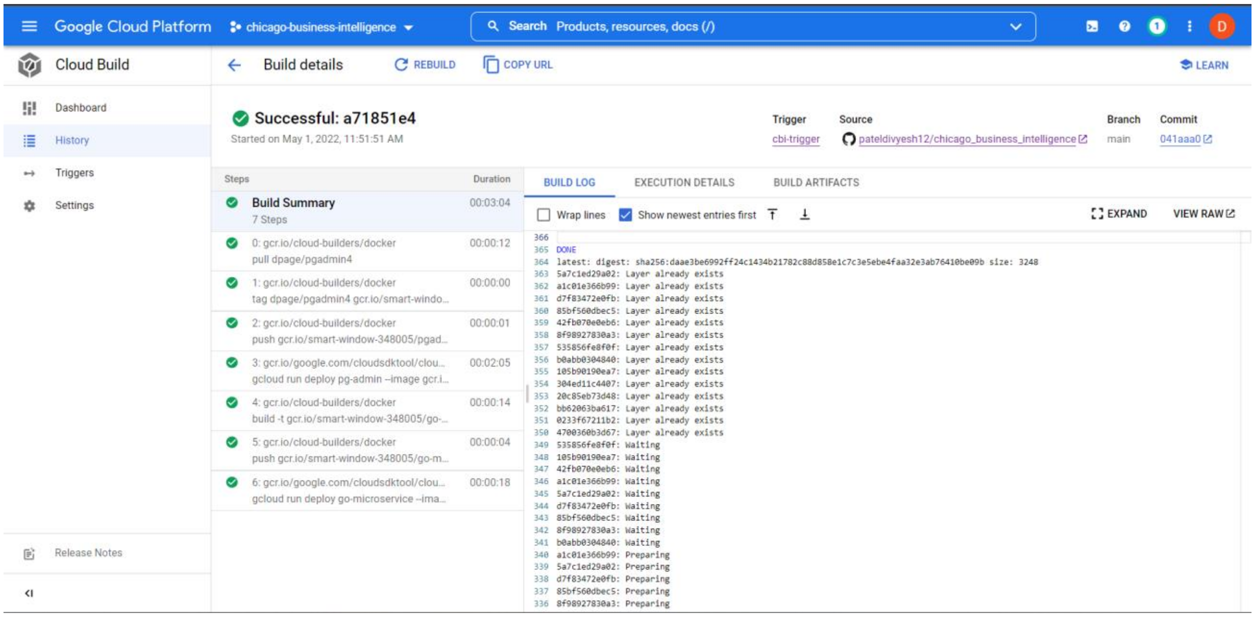
Region: us-central1

Name ↑	Description	Repository	Event	Build configuration	Status	
cbi-trigger	-	pateldivyesh12/chicago_business_intelligence	Push to branch	Autodetected	Enabled	RUN



Region: us-central1

☐	Status	Build	Source	Ref	Commit	Trigger Name	Created	Duration
☐	✓	a71851e4	pateldivyesh12/chicago_business_intelligence	main	041aaa0	cbi-trigger	5/1/22, 11:51 AM	3 min 4 sec
☐	✓	bedae309	pateldivyesh12/chicago_business_intelligence	main	041aaa0	cbi-trigger	5/1/22, 11:17 AM	2 min 34 sec
☐	!	464cefab	pateldivyesh12/chicago_business_intelligence	main	cd235fd	cbi-trigger	5/1/22, 11:14 AM	2 min 26 sec
☐	!	92e3ae4d	pateldivyesh12/chicago_business_intelligence	main	5a65954	cbi-trigger	4/30/22, 12:36 AM	2 min 15 sec
☐	!	1aaf50bd	pateldivyesh12/chicago_business_intelligence	main	5a65954	cbi-trigger	4/30/22, 12:01 AM	5 min 55 sec
☐	!	7b98d7fa	pateldivyesh12/chicago_business_intelligence	main	5a65954	cbi-trigger	4/29/22, 11:52 PM	5 min 52 sec
☐	!	5b2149af	pateldivyesh12/chicago_business_intelligence	main	5a65954	⚠ Unknown	4/29/22, 11:38 PM	6 min 33 sec
☐	✓	4b7f38f1	pateldivyesh12/chicago_business_intelligence	main	18ddb03	⚠ Unknown	4/29/22, 11:26 PM	3 min 24 sec
☐	✓	dc3a82fb	pateldivyesh12/chicago_business_intelligence	main	5ff4d64	⚠ Unknown	4/29/22, 11:20 PM	2 min 57 sec
☐	!	2b03a405	pateldivyesh12/chicago_business_intelligence	main	008dd4b	⚠ Unknown	4/29/22, 11:13 PM	5 min 42 sec
☐	✓	1cbf3ef0	pateldivyesh12/chicago_business_intelligence	main	96eb8cd	⚠ Unknown	4/29/22, 6:33 PM	2 min 39 sec
☐	!	7df039ae	pateldivyesh12/chicago_business_intelligence	main	68d5a2b	⚠ Unknown	4/29/22, 10:58 AM	6 min 30 sec
☐	!	b3ba1fc4	pateldivyesh12/chicago_business_intelligence	main	90d026e	⚠ Unknown	4/29/22, 10:57 AM	7 min 5 sec
☐	!	dd450751	pateldivyesh12/chicago_business_intelligence	main	0530582	⚠ Unknown	4/29/22, 10:48 AM	6 min 47 sec
☐	!	0d585dd3	pateldivyesh12/chicago_business_intelligence	main	0530582	⚠ Unknown	4/29/22, 10:39 AM	5 min 41 sec
☐	✓	9ecff098	pateldivyesh12/chicago_business_intelligence	main	912496e	⚠ Unknown	4/29/22, 10:32 AM	2 min 39 sec
☐	✓	cb3f25eb	pateldivyesh12/chicago_business_intelligence	main	842df1e	⚠ Unknown	4/29/22, 10:32 AM	2 min 44 sec



Successful: a71851e4

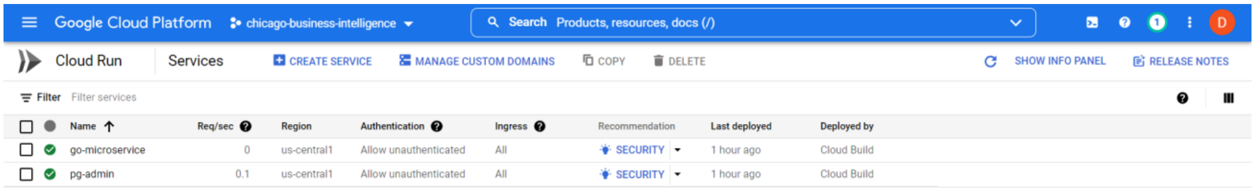
Started on May 1, 2022, 11:51:51 AM

Trigger	Source	Branch	Commit
cbi-trigger	pateldivyesh12/chicago_business_intelligence	main	041aaa0

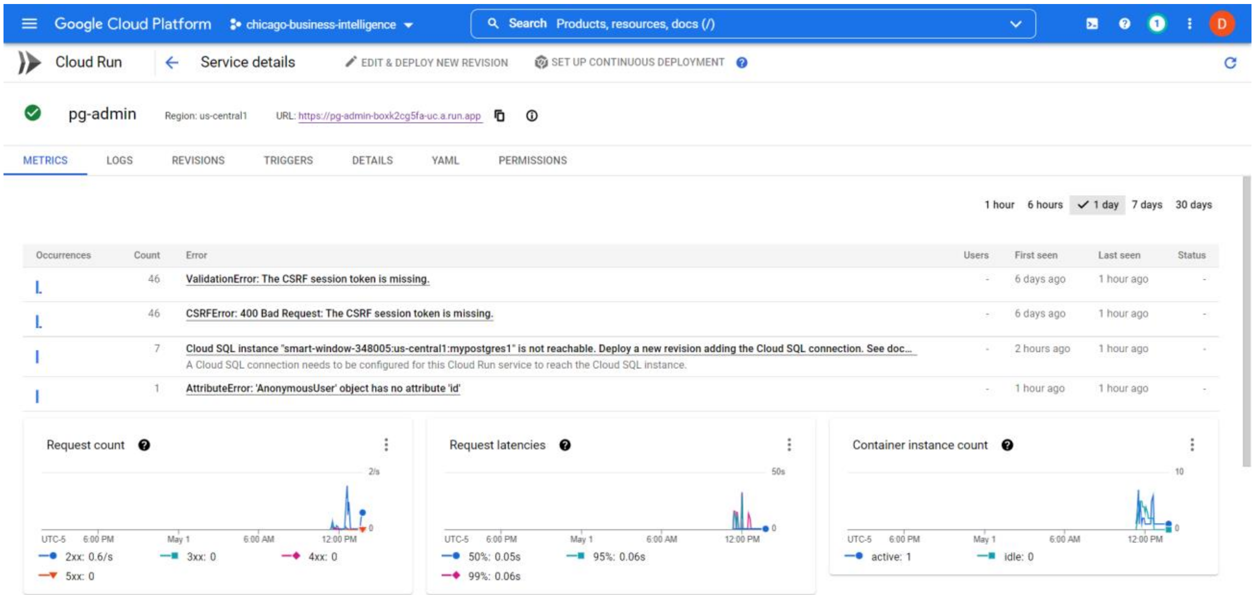
Steps	Duration
Build Summary 7 Steps	00:03:04
0: gcr.io/cloud-builders/docker pull dpage/pgadmin4	00:00:12
1: gcr.io/cloud-builders/docker tag dpage/pgadmin4 gcr.io/smart-windo...	00:00:00
2: gcr.io/cloud-builders/docker push gcr.io/smart-window-348005/pgad...	00:00:01
3: gcr.io/google.com/cloudsdktool/clou... gcloud run deploy pg-admin --image gcr.i...	00:02:05
4: gcr.io/cloud-builders/docker build -t gcr.io/smart-window-348005/go-...	00:00:14
5: gcr.io/cloud-builders/docker push gcr.io/smart-window-348005/go-m...	00:00:04
6: gcr.io/google.com/cloudsdktool/clou... gcloud run deploy go-microservice --ima...	00:00:18

□ Wrap lines ☑ Show newest entries first

```

DONE
latest: digest: sha256:daae3be6992ff24c1434b21782c88d858e1c7c3e5ebe4faa32e3ab76410be09b size: 3248
5a7c1ed29a02: Layer already exists
a1c01e366b99: Layer already exists
d7f83472e0fb: Layer already exists
85bf560dbec5: Layer already exists
42fb070e0eb6: Layer already exists
8f98927830a3: Layer already exists
535856fe8f0f: Layer already exists
b0abb0304840: Layer already exists
105b90190ea7: Layer already exists
304ed11c4407: Layer already exists
20c85eb73d48: Layer already exists
bb62063ba617: Layer already exists
0233f67211b2: Layer already exists
4700360b3d67: Layer already exists
535856fe8f0f: Waiting
105b90190ea7: Waiting
42fb070e0eb6: Waiting
a1c01e366b99: Waiting
5a7c1ed29a02: Waiting
d7f83472e0fb: Waiting
85bf560dbec5: Waiting
8f98927830a3: Waiting
b0abb0304840: Waiting
a1c01e366b99: Preparing
5a7c1ed29a02: Preparing
d7f83472e0fb: Preparing
85bf560dbec5: Preparing
8f98927830a3: Preparing
```



□	●	Name ↑	Req/sec	Region	Authentication	Ingress	Recommendation	Last deployed	Deployed by
□	✓	go-microservice	0	us-central1	Allow unauthenticated	All	SECURITY	1 hour ago	Cloud Build
□	✓	pg-admin	0.1	us-central1	Allow unauthenticated	All	SECURITY	1 hour ago	Cloud Build

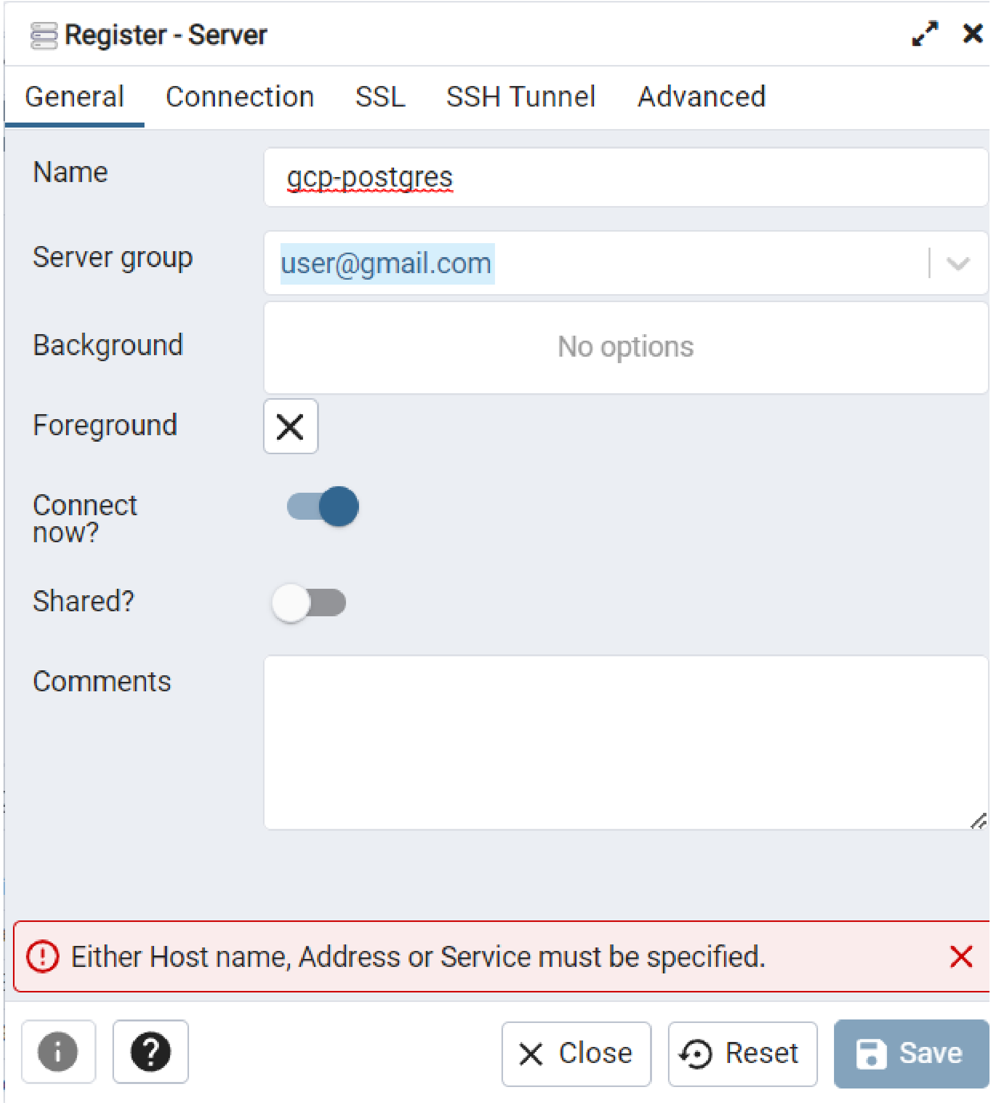


pg-admin Region: us-central1 URL: https://pg-admin-boxk2cg5fa-uc.a.run.app

Occurrences	Count	Error	Users	First seen	Last seen	Status
	46	ValidationError: The CSRF session token is missing.	-	6 days ago	1 hour ago	-
	46	CSRFError: 400 Bad Request: The CSRF session token is missing.	-	6 days ago	1 hour ago	-
	7	Cloud SQL instance "smart-window-348005:us-central1:mypostgres1" is not reachable. Deploy a new revision adding the Cloud SQL connection. See doc... A Cloud SQL connection needs to be configured for this Cloud Run service to reach the Cloud SQL instance.	-	2 hours ago	1 hour ago	-
	1	AttributeError: 'AnonymousUser' object has no attribute 'id'	-	1 hour ago	1 hour ago	-

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

Add new server and provide the name gcp-postgres



Name: gcp-postgres

Server group: user@gmail.com

Background: No options

Foreground

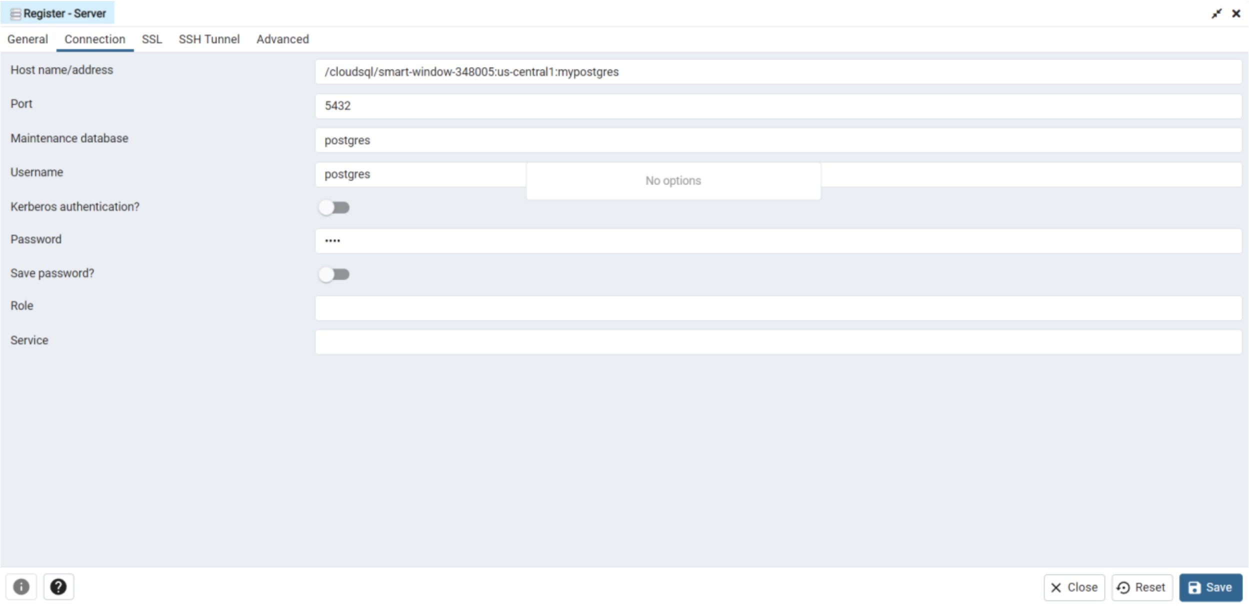
Connect now? ●

Shared? ○

Comments

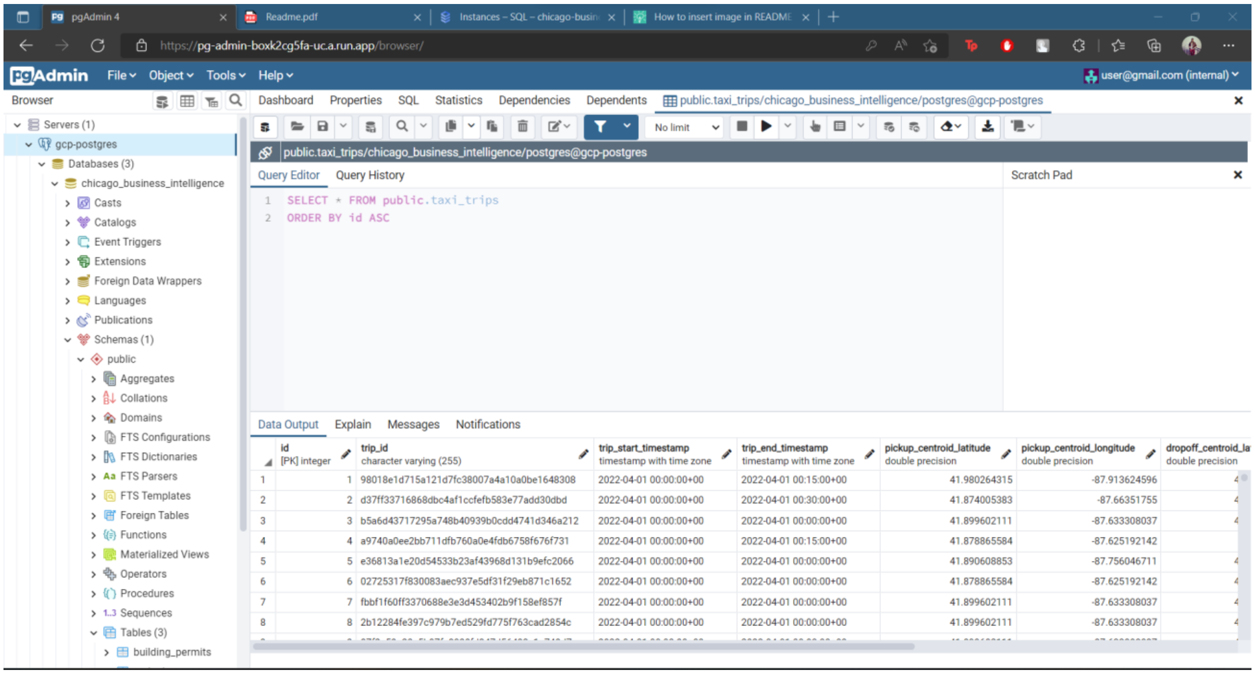
Either Host name, Address or Service must be specified.

Add host name as your sql instance name with /cloudsql/smart-window-348005:us-central1:mypostgres provide the post number as 5432 and user name postgres with password root as specified in code.



Host name/address	/cloudsql/smart-window-348005:us-central1:mypostgres
Port	5432
Maintenance database	postgres
Username	postgres
Kerberos authentication?	○
Password	••••
Save password?	○
Role	
Service	

After successful connection you will get your database and tables with data in postgres hosted on google cloud.



```
SELECT * FROM public.taxi_trips
ORDER BY id ASC
```

	id [PK] integer	trip_id character varying (255)	trip_start_timestamp timestamp with time zone	trip_end_timestamp timestamp with time zone	pickup_centroid_latitude double precision	pickup_centroid_longitude double precision	dropoff_centroid_la double precision
1	1	98018e1d715a121d7fc38007a4a10a0be1648308	2022-04-01 00:00:00+00	2022-04-01 00:15:00+00	41.980264315	-87.913624596	4
2	2	d37ff33716868dbc4af1ccfefb583e77add30dbd	2022-04-01 00:00:00+00	2022-04-01 00:30:00+00	41.874005383	-87.66351755	4
3	3	b5a6d43717295a748b40939b0cdd4741d346a212	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.899602111	-87.633308037	4
4	4	a9740a0ee2bb711dfb760a0e4fdb6758f676f731	2022-04-01 00:00:00+00	2022-04-01 00:15:00+00	41.878865584	-87.625192142	
5	5	e36813a1e20d54533b23af43968d131b9efc2066	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.890608853	-87.756046711	4
6	6	02725317f830083aec937e5df31f29eb871c1652	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.878865584	-87.625192142	4
7	7	fbbf1f60ff3370688e3e3d453402b9f158ef857f	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.899602111	-87.633308037	4
8	8	2b12284fe397c979b7ed529fd775f763cad2854c	2022-04-01 00:00:00+00	2022-04-01 00:00:00+00	41.899602111	-87.633308037	4

Link for Github Repo where the Code is hosted:

https://github.com/pateldivyesh12/chicago_business_intelligence