



Cloud SQL Collector

26.05.2022

**Get whatever you can imagine**

Juan Carlos Cruz

Telus International

San Salvador

El Salvador, CA

# Overview

Cloud SQL Collector is a Cloud Function (1er Generation) built upon Python 3.9 which operates over Cloud SQL Admin API calls and Cloud SQL direct connection, enabling your team to gather the following Cloud SQL Instances information out-of-the-box:

Cloud SQL Admin API Calls method:

* Cloud SQL Instances details (project name, instance name, region, status)
* Databases Details (name)
* Login Details (name)

Cloud SQL direct connection method:

* Databases Details (Last update date, Size)
* Databases Tables Size

# Creating and Deploying Cloud SQL Collector

This manual shows you how to create and deploy it using the Google Cloud Console.

# Before you begin

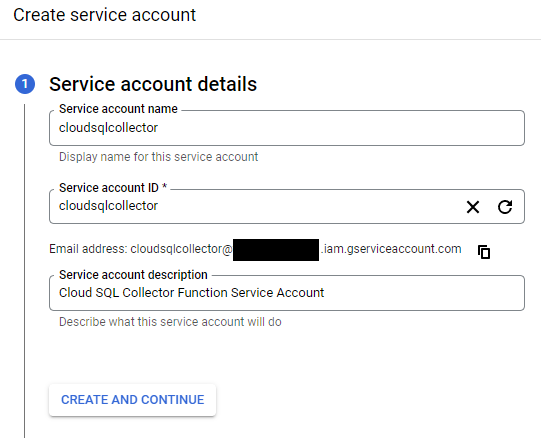
1. Sign in to the Google Cloud console using your Google Cloud account.
2. Enable the Cloud Functions, Cloud Build and Cloud SQL Admin APIs.
3. Prepare your development environment.

# Get the Cloud SQL Collector code

1. Clone the GCPCloudSQLCollector repository to your local machine:
   1. git clone <https://github.com/jotaccruz/GCPCloudSQLCollector.git>
2. Change to the directory that contains the Cloud SQL Collector code:
   1. cd GCPCloudSQLCollector/
3. Take a look at the code.

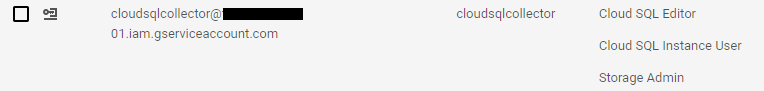
# Create the Cloud SQL Collector Service Account

1. To create the service account, follow the following steps:



Leave the other options default values, Service Account created.

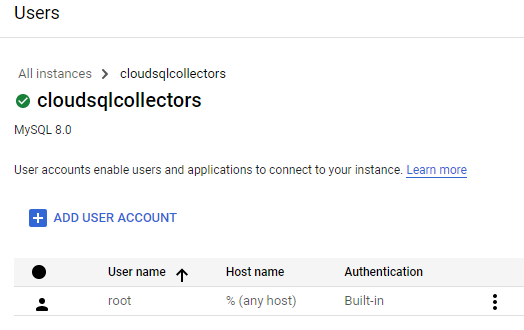
1. Grant IAM necessary roles to operate:

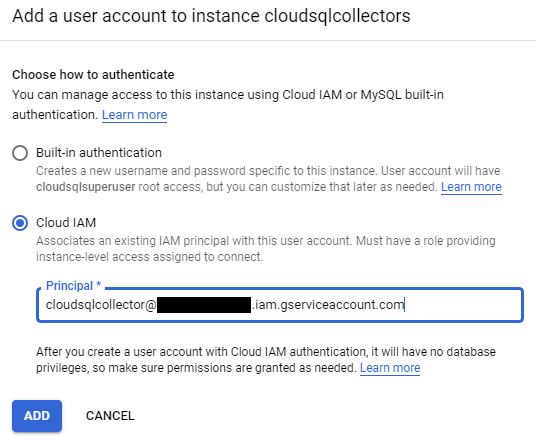


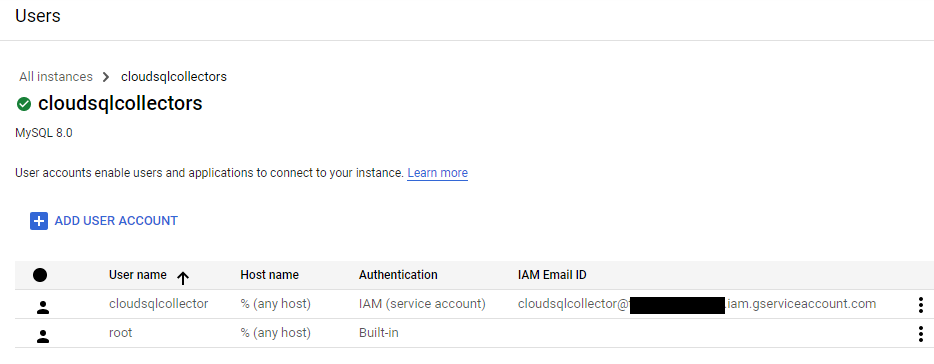
# Create the Cloud SQL Collector Config Database

Cloud SQL Collector Function stores all its configurations in a Cloud SQL MySQL database, we need to have a Cloud SQL Instance ready to use, where this database will reside.

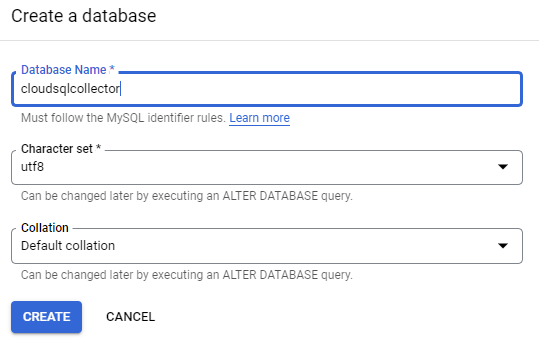
1. Adding to the Cloud SQL Instance Logins the Service Account.

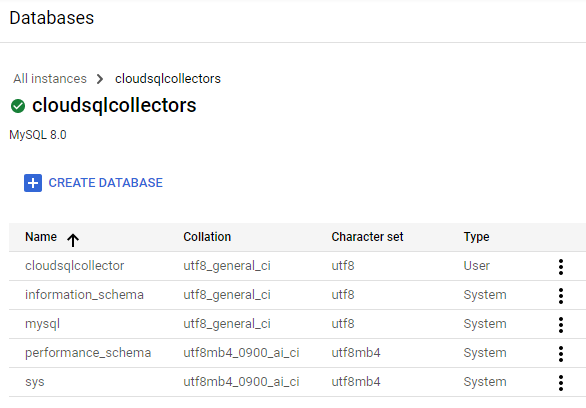






1. Creating the database “cloudsqlcollector”





1. Granting cloudsqlcollector database, read/write access to the Service Account User executing the next statement.

GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, REFERENCES, ALTER, EXECUTE, CREATE VIEW, SHOW VIEW ON `cloudsqlcollector`.\* TO 'cloudsqlcollector'@'%'

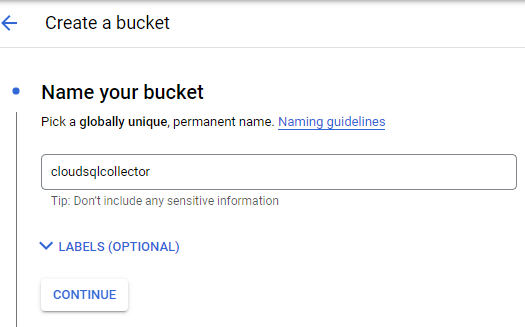
1. Setting up the initial configuration.

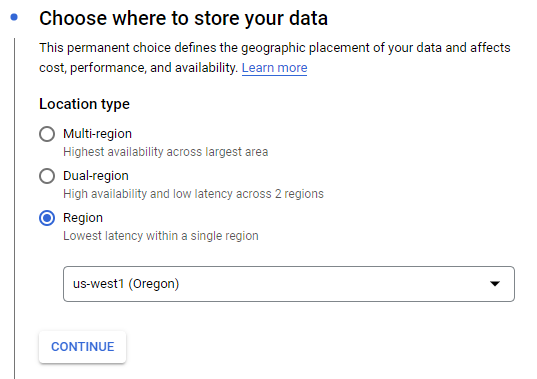
Run all the statements in the file init\_database.sql against the database cloudsqlcollector.

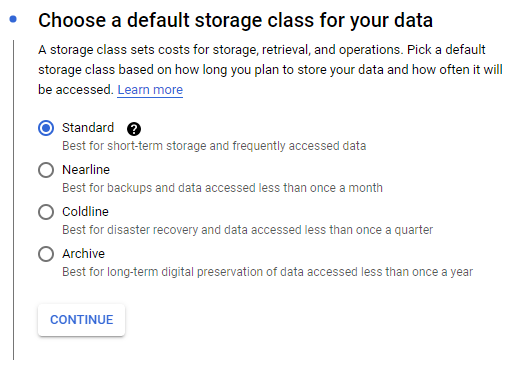
# Create a bucket

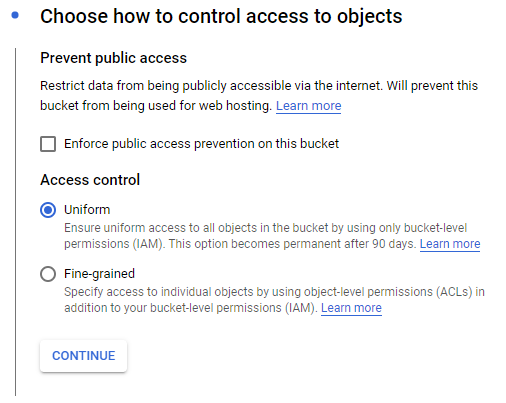
Cloud SQL Collector will generate csv formatted files and it will be saving them in a bucket, this is because all the gathered data is already stored in GCP metadata so you don’t need to store it on your side.

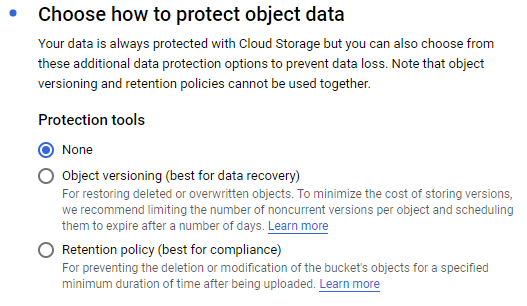
1. Creating the bucket



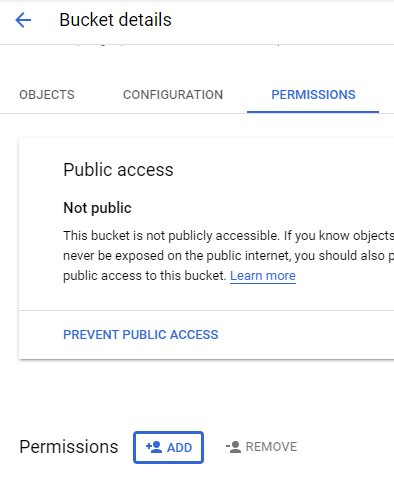


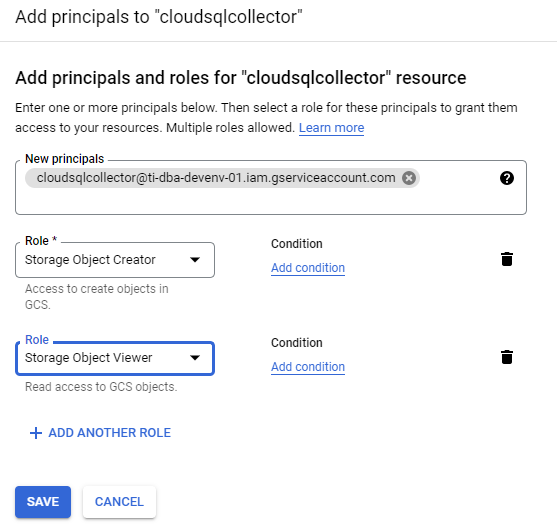






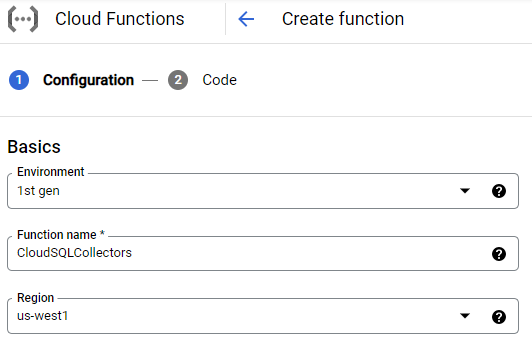
1. Adding Read/Write access to the Cloud SQL Collector Service Account



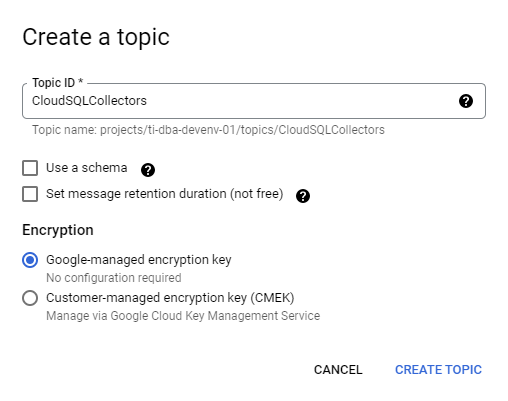


# Deploy Cloud SQL Collector Function

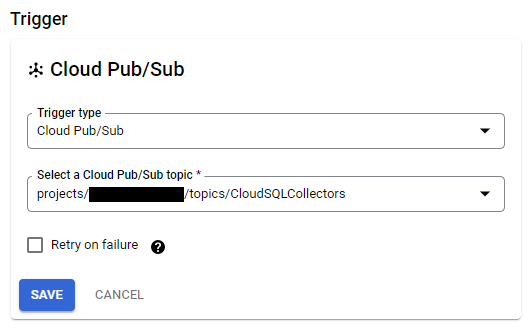
To deploy the function with the Pub/Sub method as the trigger, follow all the steps below in the Google Console:

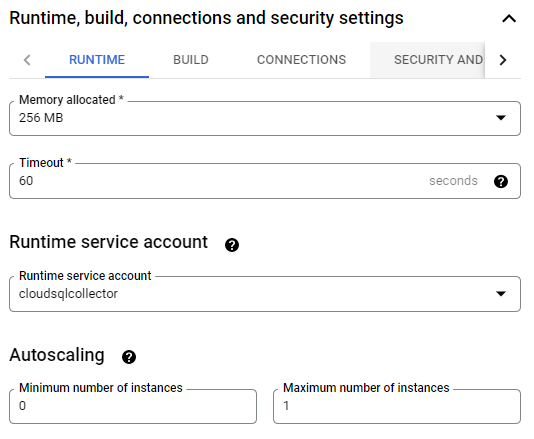


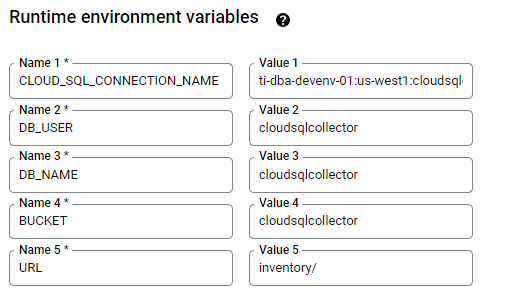
We will use Cloud Pub/Sub event to trigger the Function so we need to create a Pub/Sub Topic, select CREATE A TOPIC option:

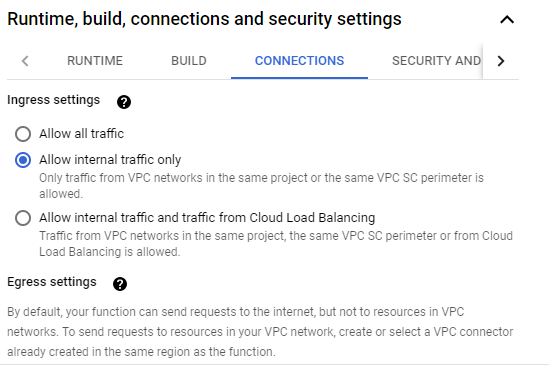


Now that the Topic is ready, select the one you have just created

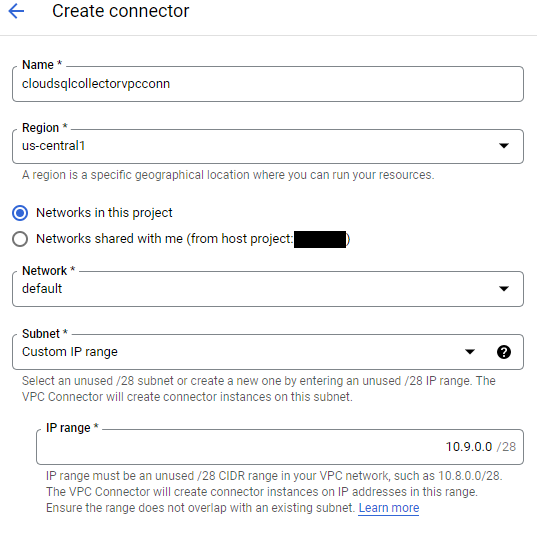


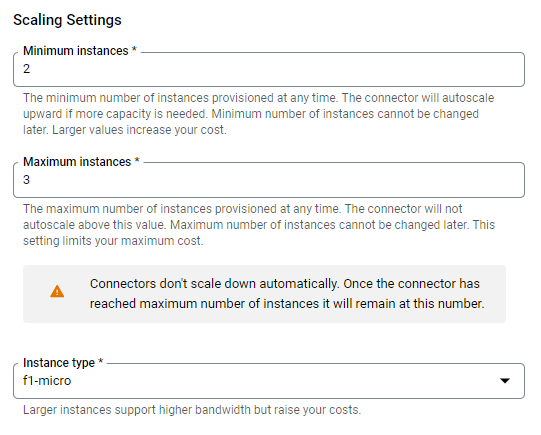


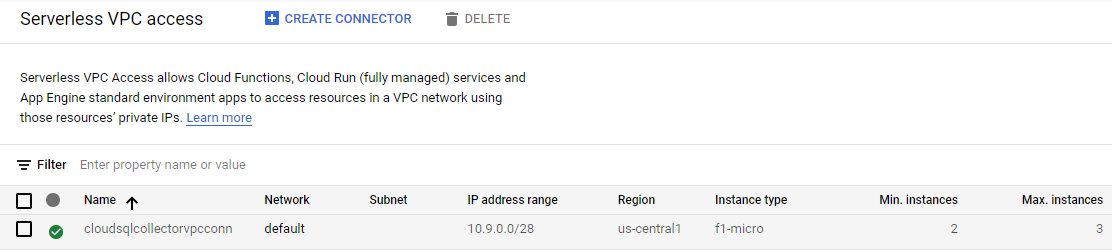




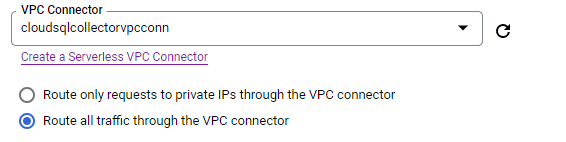
Cloud SQL Collector Function uses a Serverless VPC Connector to reach all the CloudSQL Instances in your project out, this is because Cloud Function is a Serverless Service needing to connect to a Private Services.



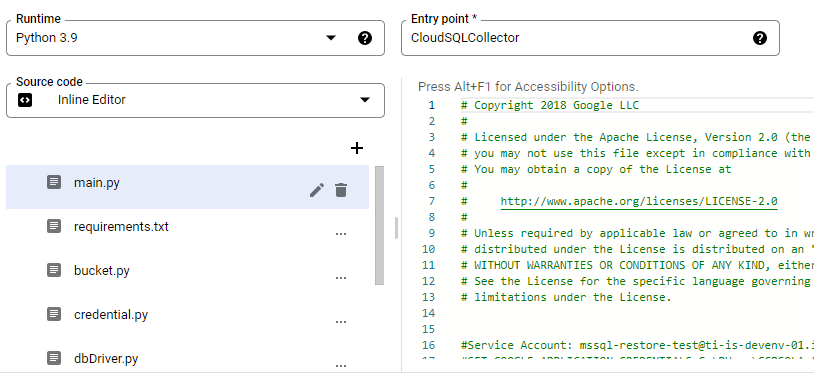




Now we are able to select the VPC Connector which will be assigned to the function.



Now we are ready to add the code:



Function Created.