

# Google Cloud Platform

# **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.

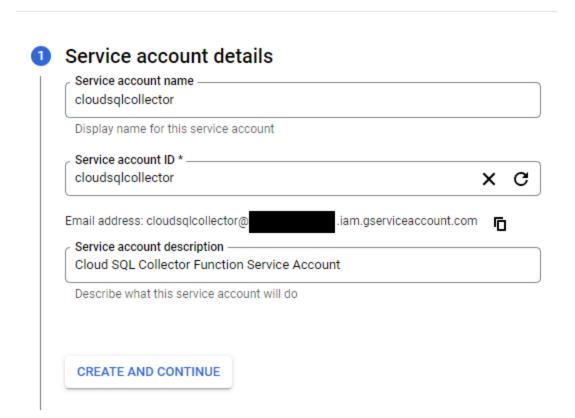
# **Get the Cloud SQL Collector code**

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

# **Create the Cloud SQL Collector Service Account**

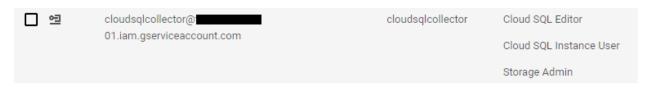
1. To create the service account, follow the following steps:

#### Create service account



Leave the other options default values, Service Account created.

2. Grant IAM necessary roles to operate:



3. MySQL granting all databases, read access to the Service Account User executing the next statement.

GRANT SELECT ON \*.\* TO 'cloudsqlcollector'@'%'

4. MSSQL granting all databases, read access to the Service Account User executing the next statement.

sp\_msforeachdb 'use [?]; CREATE USER [cloudsqlcollector] FOR LOGIN [cloudsqlcollector];EXEC sp\_addrolemember 'db\_owner', ''cloudsqlcollector'''

5. PostgreSQL granting all databases, read access to the Service Account User executing the next statement.

Add the IAM Service Account from the console

# **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.

#### Users

All instances > cloudsqlcollectors

## cloudsqlcollectors

MySQL 8.0

User accounts enable users and applications to connect to your instance. Learn more



•	User name 🏠	Host name	Authentication	
<u>.</u>	root	% (any host)	Built-in	:

#### Add a user account to instance cloudsqlcollectors

#### Choose how to authenticate

You can manage access to this instance using Cloud IAM or MySQL built-in authentication. Learn more

O Built-in authentication

Creates a new username and password specific to this instance. User account will have cloudsqlsuperuser root access, but you can customize that later as needed. Learn more

Cloud IAM

Associates an existing IAM principal with this user account. Must have a role providing instance-level access assigned to connect.

, Principal \* \_\_\_\_\_\_.iam.gserviceaccount.com

After you create a user account with Cloud IAM authentication, it will have no database privileges, so make sure permissions are granted as needed. Learn more



CANCEL

#### Users

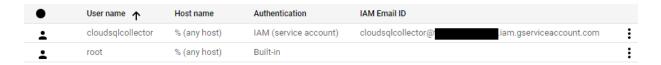
All instances > cloudsqlcollectors

#### cloudsqlcollectors

MySQL 8.0

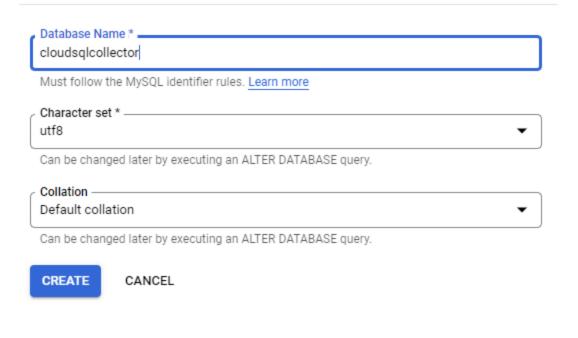
User accounts enable users and applications to connect to your instance. Learn more

+ ADD USER ACCOUNT



2. Creating the database "cloudsqlcollector"

#### Create a database



#### Databases

All instances > cloudsqlcollectors

# cloudsqlcollectors

MySQL 8.0

#### + CREATE DATABASE

Name ↑	Collation	Character set	Туре
cloudsqlcollector	utf8_general_ci	utf8	User
information_schema	utf8_general_ci	utf8	System
mysql	utf8_general_ci	utf8	System
performance_schema	utf8mb4_0900_ai_ci	utf8mb4	System
sys	utf8mb4_0900_ai_ci	utf8mb4	System

3. 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'@'%'

4. 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



# Name your bucket

Pick a globally unique, permanent name. Naming guidelines

cloudsqlcollector

Tip: Don't include any sensitive information



CONTINUE

#### Choose where to store your data

This permanent choice defines the geographic placement of your data and affects cost, performance, and availability. Learn more

Location type

Multi-region
Highest availability across largest area

Dual-region
High availability and low latency across 2 regions

Region
Lowest latency within a single region

us-west1 (Oregon)

#### Choose a default storage class for your data

CONTINUE

CONTINUE

A storage class sets costs for storage, retrieval, and operations. Pick a default storage class based on how long you plan to store your data and how often it will be accessed. Learn more

Standard Best for short-term storage and frequently accessed data
 Nearline
 Best for backups and data accessed less than once a month
 Coldline
 Best for disaster recovery and data accessed less than once a quarter
 Archive
 Best for long-term digital preservation of data accessed less than once a year

#### Choose how to control access to objects

# Prevent public access Restrict data from being publicly accessible via the internet. Will prevent this bucket from being used for web hosting. Learn more Enforce public access prevention on this bucket Access control Uniform Ensure uniform access to all objects in the bucket by using only bucket-level permissions (IAM). This option becomes permanent after 90 days. Learn more Fine-grained Specify access to individual objects by using object-level permissions (ACLs) in addition to your bucket-level permissions (IAM). Learn more

# Choose how to protect object data

Your data is always protected with Cloud Storage but you can also choose from these additional data protection options to prevent data loss. Note that object versioning and retention policies cannot be used together.

#### Protection tools

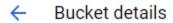
- None
- Object versioning (best for data recovery)

For restoring deleted or overwritten objects. To minimize the cost of storing versions, we recommend limiting the number of noncurrent versions per object and scheduling them to expire after a number of days. Learn more

Retention policy (best for compliance)

For preventing the deletion or modification of the bucket's objects for a specified minimum duration of time after being uploaded. Learn more

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



OBJECTS CONFIGURATION

**PERMISSIONS** 

#### Public access

#### Not public

This bucket is not publicly accessible. If you know objects never be exposed on the public internet, you should also p public access to this bucket. <u>Learn more</u>

PREVENT PUBLIC ACCESS

Permissions

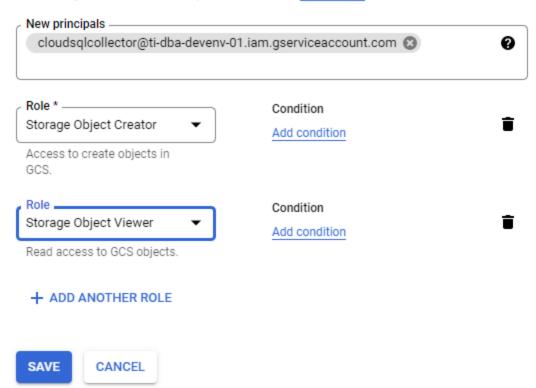


- REMOVE

## Add principals to "cloudsqlcollector"

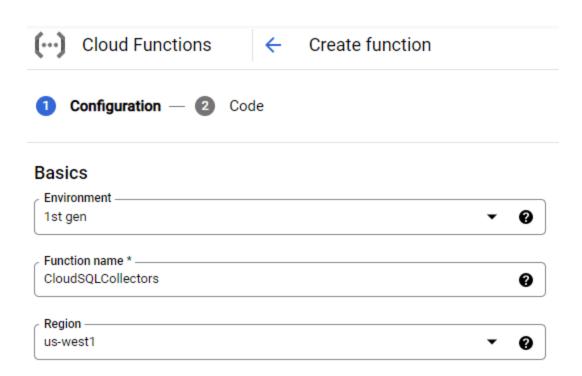
#### Add principals and roles for "cloudsqlcollector" resource

Enter one or more principals below. Then select a role for these principals to grant them access to your resources. Multiple roles allowed. Learn more



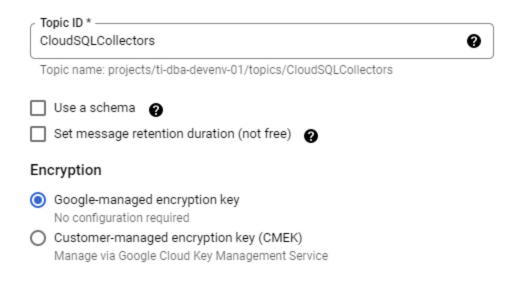
# **Deploy Cloud SQL Collector Function**

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:

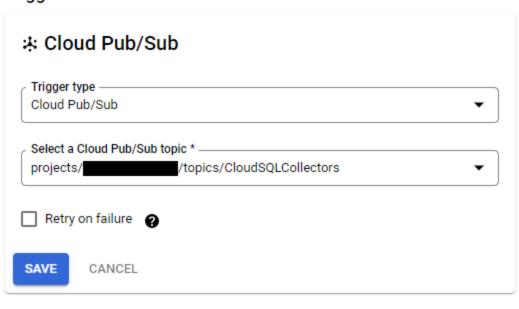
# Create a topic

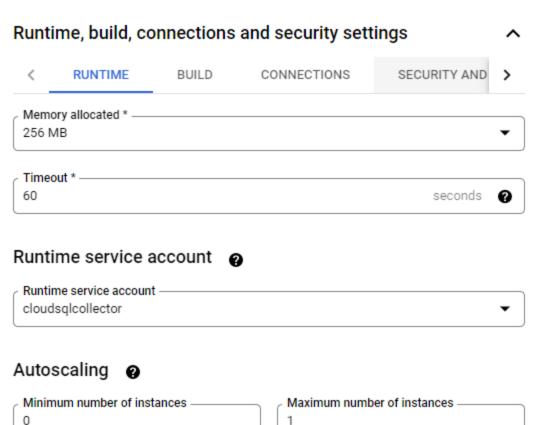


CANCEL CREATE TOPIC

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

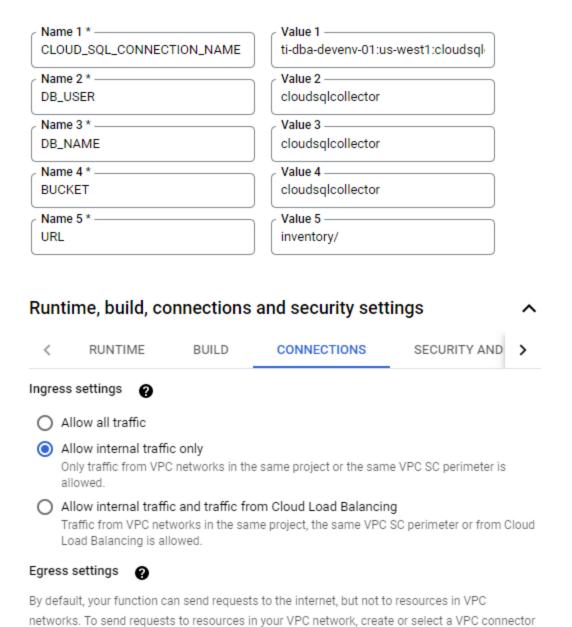
#### Trigger





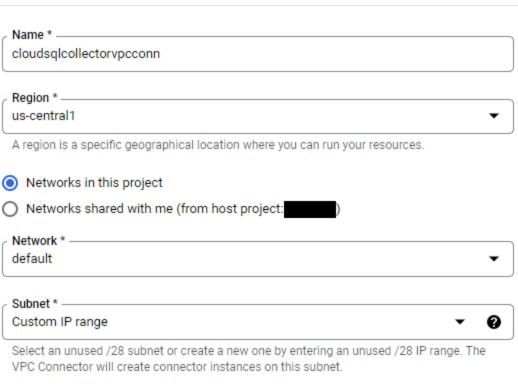
## Runtime environment variables @

already created in the same region as the function.



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.

#### Create connector



IP range must be an unused /28 CIDR range in your VPC network, such as 10.8.0.0/28. The VPC Connector will create connector instances on IP addresses in this range. Ensure the range does not overlap with an existing subnet. Learn more

#### Scaling Settings

Minimum instances \* \_\_\_\_\_\_\_2

The minimum number of instances provisioned at any time. The connector will autoscale upward if more capacity is needed. Minimum number of instances cannot be changed later. Larger values increase your cost.

The maximum number of instances provisioned at any time. The connector will not autoscale above this value. Maximum number of instances cannot be changed later. This setting limits your maximum cost.

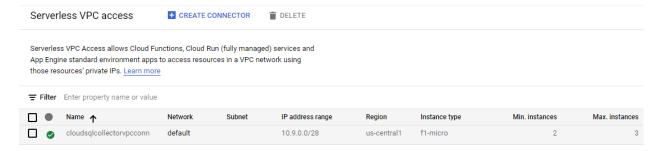


Connectors don't scale down automatically. Once the connector has reached maximum number of instances it will remain at this number.

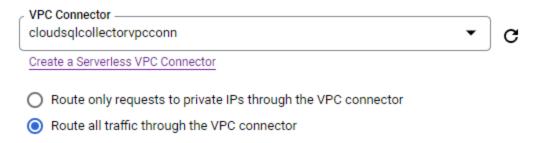
f1-micro 

Instance type \*

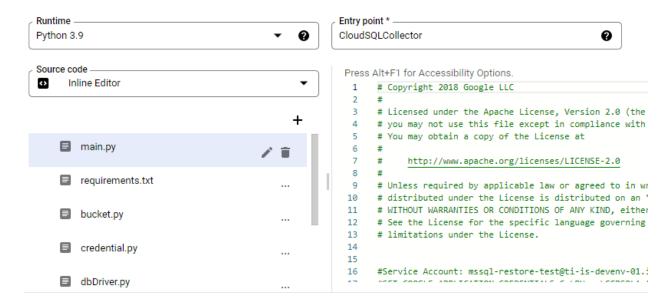
Larger instances support higher bandwidth but raise your costs.



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.

#### To test it:



#### Configure Triggering Event @

Press Alt+F1 for Accessibility Options.

1 {"message":{"data":"Hi-Bro"}}