Connecting to Cloud SQL from Cloud Run (fully managed)

This page contains information and examples for connecting to a Cloud SQL instance from a service running in Cloud Run (fully managed).

Cloud SQL is a fully-managed database service that makes it easy to set up, maintain, manage, and administer your relational PostgreSQL and MySQL databases in the cloud.

Before you begin

To connect a service running in a Cloud Run instance to a Cloud SQL, you need to have your service containerized and uploaded to the Container Registry.

If you don't already have one, see these instructions about building and deploying (https://cloud.google.com/run/docs/quickstarts/build-and-deploy) a container image.

Setting up a Cloud SQL instance

1. Create a Cloud SQL for MySQL instance (https://cloud.google.com/sql/docs/mysql/create-instance#create-2nd-gen).

★ **Note:** These instructions require your Cloud SQL instance to have a public IP address configured.

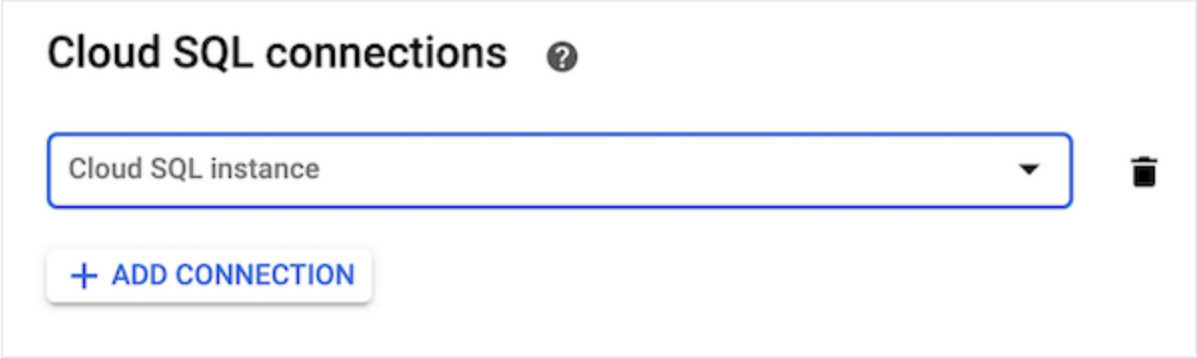
2. Find the ***`INSTANCE_CONNECTION_NAME`*** for the instance on the **Instance details** page. It uses the format ***`PROJECT_ID`*****`:`*****`REGION`*****`:`*****`INSTANCE_ID`***, and is used to identify the Cloud SQL instance you are connecting to.
3. Enable the Cloud SQL Admin API, if you haven't already done so:

ENABLE THE API (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/FLOWS/ENABLEAPI?APIID=SQLADMIN&

Configuring Cloud Run (fully managed)

CONSOLE COMMAND LINE

1. GO TO CLOUD RUN (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/RUN)
2. Configure the service:
   - If you are adding a Cloud SQL connection to a new service:
     - You need to have your service containerized and uploaded to the Container Registry. If you don't already have one, see these instructions about building and deploying (https://cloud.google.com/run/docs/quickstarts/build-and-deploy) a container image.
     - Click **CREATE SERVICE**.
   - If you are adding Cloud SQL connections to an existing service:
     - Click on the service name.
     - Click **DEPLOY NEW REVISION**.
3. Enable connecting to a Cloud SQL:
   - Click **SHOW OPTIONAL SETTINGS**:



   - If you are adding a connection to a Cloud SQL instance in your project, select the desired Cloud SQL instance from the dropdown menu.
   - If you are using a Cloud SQL instance from another project, select **custom connection string** in the dropdown and then enter the full instance connection name in the format ***PROJECT-ID:REGION:INSTANCE-ID***.
   - If you are deleting a connection, hover your cursor to the right of the connection to display the Trash icon, and click it.

4. Click **Create** or **Deploy**.

Cloud Run (fully managed) uses a service account to authorize your connections to Cloud SQL. This service account must have the correct IAM permissions to successfully connect. Unless otherwise configured, the default service account is in the format `PROJECT_NUMBER-compute@developer.gserviceaccount.com`.

When connecting resources in two different projects, make sure that both projects have enabled the correct IAM roles and have given the service account the correct permissions.

Ensure that the service account for your service has one of the following IAM roles (https://cloud.google.com/iam/docs/understanding-roles#cloud-sql-roles):

- `Cloud SQL Client` (preferred)
- `Cloud SQL Editor`
- `Cloud SQL Admin`

Or, you can manually assign the following IAM permissions (https://cloud.google.com/storage/docs/access-control/using-iam-permissions):

- `cloudsql.instances.connect`
- `cloudsql.instances.get`

For detailed instructions on adding IAM roles to a service account, see Granting Roles to Service Accounts (https://cloud.google.com/iam/docs/granting-roles-to-service-accounts).

Connecting to Cloud SQL

Once correctly configured, you can connect your service to your Cloud SQL instance's unix domain socket using the format: `/cloudsql/INSTANCE_CONNECTION_NAME`.

These connections are automatically encrypted without any additional configuration.

Warning: Linux based operating systems have a maximum socket path length of 107 characters. If the total length of the path exceeds this length, you will not be able to connect with a socket from Cloud Run (fully managed).

Note: Cloud Run (fully managed) **does not** support connecting to the Cloud SQL instance using TCP. Your code should **not** try to access the instance using an IP address such as `127.0.0.1` or `172.17.0.1`.

PYTHON JAVA NODE.JS RUBY

cloud-sql/mysql/sqlalchemy/main.py (https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/cloud-sql/mysql/sqlalchemy/main.py)

)UDPLATFORM/PYTHON-DOCS-SAMPLES/BLOB/MASTER/CLOUD-SQL/MYSQL/SQLALCHEMY/MAIN.PY)

```
# The SQLAlchemy engine will help manage interactions, including automatically
# managing a pool of connections to your database
db = sqlalchemy.create_engine(
    # Equivalent URL:
    # mysql+pymysql://<db_user>:<db_pass>@/<db_name>?unix_socket=/cloudsql/<cloud_
    sqlalchemy.engine.url.URL(
        drivername="mysql+pymysql",
        username=db_user,
        password=db_pass,
        database=db_name,
        query={"unix_socket": "/cloudsql/{}".format(cloud_sql_connection_name)},
    ),
    # ... Specify additional properties here.
    # ...
)
```

To see this snippet in the context of a web application, view the source code on GitHub (https://github.com/GoogleCloudPlatform/python-docs-samples/blob/master/cloud-sql/mysql/sqlalchemy/)
.

Best Practices & Other Information

You can use the Cloud SQL proxy (https://cloud.google.com/sql/docs/mysql/sql-proxy) when testing your application locally. See the quickstart for using the proxy for local testing (https://cloud.google.com/sql/docs/mysql/quickstart-proxy-test) for detailed instructions.

You can also test using the Cloud SQL Proxy via a docker container (https://cloud.google.com/sql/docs/mysql/connect-docker).

Connection Pools

Connections to underlying databases may be dropped, either by the database server itself, or by the platform infrastructure. To mitigate this, we recommend that you use a client library that supports connection pools that automatically reconnect broken client connections.

For more detailed examples on how to use connection pools, see the Managing database connections (https://cloud.google.com/sql/docs/mysql/manage-connections) page.

Connection Limits

Both the MySQL and PostgreSQL editions of Cloud SQL impose a maximum limit on concurrent connections, and these limits may vary depending on the database engine chosen (see the Cloud SQL Quotas and Limits (https://cloud.google.com/sql/docs/quotas#fixed-limits)) page.

Cloud Run has the ability to automatically create more instances as load increases, which may cause you to exceed these limits. You can limit the maximum number of connections used per instance by using a connection pool. For more detailed examples on how to limit the number of connections, see the Managing database connections (https://cloud.google.com/sql/docs/mysql/manage-connections#count) page.

Next steps

- Learn more about Cloud Run (https://cloud.google.com/run/docs/).
- Learn more about building and deploying (https://cloud.google.com/run/docs/quickstarts/build-and-deploy) container images.
- See a complete example in Python for using Cloud Run with MySQL (https://github.com/GoogleCloudPlatform/python-docs-samples/tree/master/cloud-sql/mysql/sqlalchemy)
  .

Except as otherwise noted, the content of this page is licensed under the Creative Commons Attribution 4.0 License (https://creativecommons.org/licenses/by/4.0/), and code samples are licensed under the Apache 2.0 License (https://www.apache.org/licenses/LICENSE-2.0). For details, see our Site Policies (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.

Last updated January 6, 2020.