Cloud SQL (https://cloud.google.com/sql/)
Documentation (https://cloud.google.com/sql/docs/)
MySQL (https://cloud.google.com/sql/docs/mysql/) Guides

Connecting to Cloud SQL from App Engine

MySQL | PostgreSQL (https://cloud.google.com/sql/docs/postgres/connect-app-engine) | SQL Server

This page contains information and examples for connecting to a Cloud SQL instance from a service running in App Engine.

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.

App Engine is a fully managed, serverless platform for developing and hosting web applications at scale. You can choose from several popular languages, libraries, and frameworks to develop your apps, then let App Engine take care of provisioning servers and scaling your app instances based on demand.

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. If you want to use a private IP address, see Configuring Serverless VPC Access (https://cloud.google.com/vpc/docs/configure-serverless-vpc-access) and connect directly using the private IP.

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 App Engine

STANDARD FLEXIBLE

App Engine does not require any special configuration beyond making sure the service account that you use has the correct permissions.

Note: App Engine standard enviroments **do 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`) unless you have configured Serverless VPC Access (https://cloud.google.com/vpc/docs/configure-serverless-vpc-access).

App Engine 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 `service-PROJECT_NUMBER@gae-api-prod.google.com.iam.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 App Engine.

App Engine flexible environment also supports connecting through TCP. If you have configured an instance with a TCP port, configure your application to connect to `172.17.0.1:PORT` instead.

PYTHON JAVA NODE.JS MORE

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.

Connection Pools

Connections to underlying databases may be dropped, either by the database server itself, or by the underlying infrastructure. To mitigate this, we recommend that you use a client library that supports connection pools and automatic reconnection.

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

Connection Limits

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

App Engine has the ability to automatically create more instances as load increases, which may cause you to exceed these limits. To avoid this issue, limit the maximum number of App Engine instances. For more information, see Scaling elements (https://cloud.google.com/appengine/docs/standard/python/config/appref#scaling_elements).

Each App Engine instance running in a standard environment cannot have more than *100 concurrent connections* to an instance. For PHP 5.5 apps, the limit is *60 concurrent connections*.

App Engine applications are subject to request time limits depending on usage and environment. For more information, see how instances are managed in App Engine standard (https://cloud.google.com/appengine/docs/standard/python/how-instances-are-managed#timeout) and flexible (https://cloud.google.com/appengine/docs/flexible/python/how-instances-are-managed) environments.

App Engine applications are also subject to additional App Engine quotas and limits as discussed on the App Engine Quotas (https://cloud.google.com/appengine/docs/quotas) page.

Last updated January 21, 2020.