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

Connecting from Google Kubernetes Engine

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

This page describes how to set up a connection from an application running in Google Kubernetes Engine to a Cloud SQL instance, using private IP or the Cloud SQL Proxy Docker image.

Introduction

To access a Cloud SQL instance from an application running in Google Kubernetes Engine, you can use either a private IP address or the Cloud SQL Proxy Docker image. Private IP is the easiest way to connect from Google Kubernetes Engine if your cluster meets the requirements.

To connect securely to Cloud SQL from Google Kubernetes Engine using a public IP address, you must use the Cloud SQL Proxy.

Database connections consume resources on the server and the connecting application. Always use good connection management practices to minimize your application's footprint and reduce the likelihood of exceeding Cloud SQL connection limits (https://cloud.google.com/sql/docs/mysql/quotas#fixed-limits). For more information, see Managing database connections (https://cloud.google.com/sql/docs/mysql/manage-connections).

Connecting using a private IP address

When you connect using a private IP address, the IP traffic is never exposed to the public internet. For more information about private IP, see Private IP (https://cloud.google.com/sql/docs/mysql/private-ip).

Before you begin

To connect to Cloud SQL using private IP, you must have:

- GKE version 1.8 or higher. GKE must be running on a VPC-native
  (https://cloud.google.com/kubernetes-engine/docs/how-to/alias-ips) cluster. You need to install
  the `kubectl` command-line tool and configure it to communicate with the cluster.

  For help getting started with GKE, see the Quickstart
  (https://cloud.google.com/kubernetes-engine/docs/quickstart).

- An application container in a pod on the GKE cluster.

- A Second Generation instance created.

  For help creating a Cloud SQL instance, see Creating Instances
  (https://cloud.google.com/sql/docs/mysql/create-instance).

- A MySQL user account configured on the instance.

  Your application will use this account to connect to the database.

- A Private IP configured and assigned
  (https://cloud.google.com/sql/docs/mysql/configure-private-ip) to the Cloud SQL instance.

Connection overview

The steps below outline the general process for connecting to Cloud SQL from GKE using a
private IP address.

1. Create a Secret to provide the MySQL username and password to the database.

   For more information about secrets, see Secret
   (https://cloud.google.com/kubernetes-engine/docs/concepts/secret).

2. Update your pod configuration file with the following items:

   - Provide the Cloud SQL instance's private IP address as the host address your
     application will use to access your database.

   - Provide the Secret you previously created to enable the application to log into the
     database.

3. Bring up your Deployment using the Kubernetes manifest file.

Connecting using the Cloud SQL Proxy Docker image

When you connect using the Cloud SQL Proxy Docker image, the Cloud SQL Proxy is added to your pod using the "sidecar" container pattern—the proxy container is in the same pod as your application, which enables the application to connect to the proxy using `localhost`, increasing security and performance. Learn more (https://kubernetes.io/docs/concepts/workloads/pods/pod-overview).

For more information about the Cloud SQL Proxy, see About the Cloud SQL Proxy (https://cloud.google.com/sql/docs/mysql/sql-proxy). For more information about working with pods, see Pod Overview (https://kubernetes.io/docs/concepts/workloads/pods/pod-overview) in the Kubernetes documentation.

> **Note:** The Cloud SQL Proxy is available only for Cloud SQL Second Generation instances.

Before you begin

Before you start this procedure, you must have:

- A GKE cluster running version 1.2 or higher, with the `kubectl` command-line tool installed and configured to communicate with the cluster.

  For help getting started with GKE, see the Quickstart (https://cloud.google.com/kubernetes-engine/docs/quickstart).

- An application container in a pod on the GKE cluster.

- A Second Generation instance created.

  For help creating a Cloud SQL instance, see Creating Instances (https://cloud.google.com/sql/docs/mysql/create-instance).

- A MySQL user account configured on the instance.

  Your application will use this account to connect to the database. For help with creating a user account, see Creating a user (https://cloud.google.com/sql/docs/mysql/create-manage-users#creating).

- The Cloud SQL Admin API is enabled.

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

- The location of the key file associated with a service account with the proper privileges for your Cloud SQL instance.

See Creating a service account
 (https://cloud.google.com/sql/docs/mysql/sql-proxy#create-service-account) for more
information.

- Your Cloud SQL instance connection name.

    The instance connection name is available in the **Cloud SQL Instance details** page of the
    Cloud Console or from the `gcloud sql instances describe` command.

Connection overview

The steps below outline the general process for connecting to Cloud SQL from GKE using the
Cloud SQL Proxy Docker container.

1. Create two Secrets: one to provide the MySQL credentials and one to provide the Google
   credentials (the service account).

   For more information about secrets, see Secret
    (https://cloud.google.com/kubernetes-engine/docs/concepts/secret).

2. Update your pod configuration file with the following items:

   - Provide `127.0.0.1:3306` as the host address your application will use to access your
     database.

   - Provide the Secret you previously created to enable the application to log into the
     database.

   - Start the proxy in its own container (in the same pod).

     Here is a sample configuration file for the proxy container:

     mysql_deployment.yaml
      (https://github.com/GoogleCloudPlatform/kubernetes-engine-
     samples/blob/master/cloudsql/mysql_wordpress_deployment.yaml)

     'KUBERNETES-ENGINE-SAMPLES/BLOB/MASTER/CLOUDSQL/MYSQL_WORDPRESS_DEPLOYMENT.YAML)

```
- name: cloudsql-proxy
  image: gcr.io/cloudsql-docker/gce-proxy:1.16
  command: ["/cloud_sql_proxy",
           "-instances=<INSTANCE_CONNECTION_NAME>=tcp:3306",
           # If running on a VPC, the Cloud SQL proxy can connect via Pri
           # https://cloud.google.com/sql/docs/mysql/private-ip for more
           # "-ip_address_types=PRIVATE",
           "-credential_file=/secrets/cloudsql/credentials.json"]
```

```
securityContext:
  runAsUser: 2  # non-root user
  allowPrivilegeEscalation: false
volumeMounts:
  - name: cloudsql-instance-credentials
    mountPath: /secrets/cloudsql
    readOnly: true
```

3. Bring up your Deployment using the Kubernetes manifest file.

For a complete sample deployment with the Cloud SQL Proxy in the sidecar pattern, including sample code, see sample Kubernetes Deployment manifest (https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/master/cloudsql/mysql_wordpress_deployment.yaml) file on GitHub.

Need help? For help troubleshooting the proxy, see Troubleshooting Cloud SQL Proxy connections (https://cloud.google.com/sql/docs/mysql/sql-proxy#troubleshooting). Or, see our Cloud SQL Support page (https://cloud.google.com/sql/docs/mysql/support).

What's next

- Learn more about private IP (https://cloud.google.com/sql/docs/mysql/private-ip).

- Learn more about the proxy (https://cloud.google.com/sql/docs/mysql/sql-proxy) and the proxy docker image (https://cloud.google.com/sql/docs/mysql/connect-docker).

- View the sample code connecting using the proxy Docker image on GitHub (https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/master/cloudsql/mysql_wordpress_deployment.yaml) .

- Get help troubleshooting proxy connections (https://cloud.google.com/sql/docs/mysql/sql-proxy#troubleshooting).

- Learn more about Google Kubernetes Engine (https://cloud.google.com/kubernetes-engine/docs/).

- Learn about options for support (https://cloud.google.com/sql/docs/mysql/support).