

[MySQL](#) (/sql/docs/mysql/connect-kubernetes-engine) | **PostgreSQL** | [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.

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](#) (/sql/docs/postgres/quotas#fixed-limits). For more information, see [Managing database connections](#) (/sql/docs/postgres/manage-connections).

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](#) (/sql/docs/postgres/private-ip).

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](#) (/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](#) (/kubernetes-engine/docs/quickstart).

- An application container in a pod on the GKE cluster.
- A PostgreSQL instance created.

For help creating a Cloud SQL instance, see [Creating Instances](#) (/sql/docs/postgres/create-instance).

- A PostgreSQL user account configured on the instance.

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

- A Private IP [configured and assigned](#) (/sql/docs/postgres/configure-private-ip) to the Cloud SQL instance.

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 PostgreSQL username and password to the database.

For more information about secrets, see [Secret \(/kubernetes-engine/docs/concepts/secret\)](/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.

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 \(/sql/docs/postgres/sql-proxy\)](/sql/docs/postgres/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.

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 \(/kubernetes-engine/docs/quickstart\)](/kubernetes-engine/docs/quickstart).

- An application container in a pod on the GKE cluster.
- A PostgreSQL instance created.

For help creating a Cloud SQL instance, see [Creating Instances \(/sql/docs/postgres/create-instance\)](/sql/docs/postgres/create-instance).

- A PostgreSQL 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 \(/sql/docs/postgres/create-manage-users#creating\)](/sql/docs/postgres/create-manage-users#creating).

- The Cloud SQL Admin API is enabled.

[Enable the API \(https://console.cloud.google.com/flows/enableapi?apiid=sqladmin&redirect=https://console.cloud.google.com\)](https://console.cloud.google.com/flows/enableapi?apiid=sqladmin&redirect=https://console.cloud.google.com)

- 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](#) (/sql/docs/postgres/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.

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 PostgreSQL credentials and one to provide the Google credentials (the service account).

For more information about secrets, see [Secret](#) (/kubernetes-engine/docs/concepts/secret).

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

- Provide `127.0.0.1:5432` 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:

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

[on GitHub](#) (https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/master/cloudsql/postgres_deployment.yaml)

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/postgres_deployment.yaml) file on GitHub.

For step-by-step instructions for connecting to Cloud SQL from the Kubernetes environment, see the [PostgreSQL and Kubernetes Codelab](#) (<https://codelabs.developers.google.com/codelabs/cloud-potgresql-gke-memegen>).

Need help? For help troubleshooting the proxy, see [Troubleshooting Cloud SQL Proxy connections](#) (/sql/docs/postgres/sql-proxy#troubleshooting). Or, see our [Cloud SQL Support page](#) (/sql/docs/postgres/support).

- Learn more about [private IP](#) (/sql/docs/postgres/private-ip).
- Learn more about the [proxy](#) (/sql/docs/postgres/sql-proxy) and the [proxy docker image](#) (/sql/docs/postgres/connect-docker).
- View the [sample code connecting using the proxy Docker image on GitHub](#) (https://github.com/GoogleCloudPlatform/kubernetes-engine-samples/blob/master/cloudsql/postgres_deployment.yaml).
- Get help [troubleshooting proxy connections](#) (/sql/docs/postgres/sql-proxy#troubleshooting).
- Learn more about [Google Kubernetes Engine](#) (/kubernetes-engine/docs/).
- Learn about [options for support](#) (/sql/docs/postgres/support).