

This tutorial walks you through the process of running a MySQL database on a private network in Google Cloud to allow secure, remote access to the database by using [Compute Engine \(/compute/\)](#).

Use this tutorial if you want to install your own MySQL database on Compute Engine, but want to restrict access to only authorized MySQL clients also running on Compute Engine. You might want to manage your own MySQL instance instead of using the [managed service \(/sql/\)](#), due to cross-region instances, advanced usage of parameters, and specific performance needs.

This tutorial describes how to configure your MySQL server app to accept remote traffic from a MySQL client that is installed on a Compute Engine instance on the same private network.

For information about how to choose the right MySQL deployment option, see [How to install MySQL on Compute Engine \(/solutions/setup-mysql\)](#).

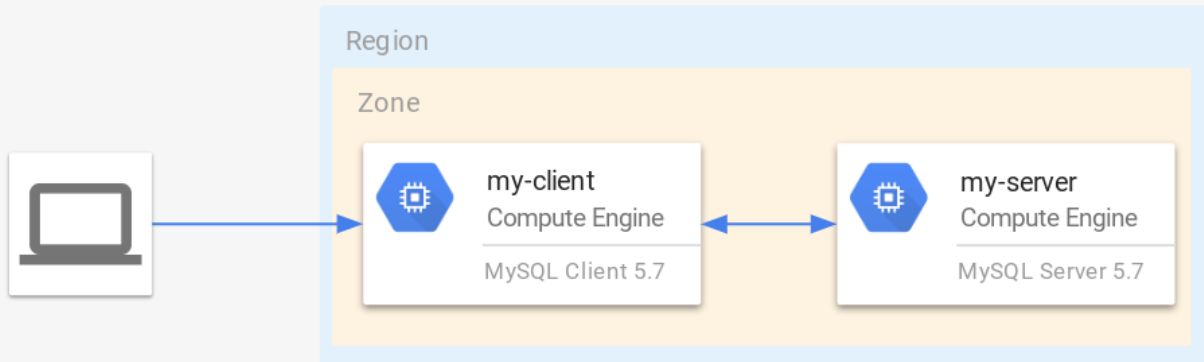
This tutorial assumes that you are familiar with the following:

- Basic Linux commands
- Ubuntu-server 18.04
- MySQL 5.7
- Compute Engine

In this tutorial, you deploy two Compute Engine instances. One instance is the server and the other instance is the client as depicted in the following diagram:

MySQL Compute Engine instance architecture

Google Cloud Platform



- Create a Compute Engine instance and install MySQL server.
- Create a Compute Engine instance and install MySQL client.
- Configure MySQL server for remote access.
- Remove public access to the MySQL server.
- Connect remotely to MySQL.
- Create a VPC Service Controls firewall rule.

This tutorial uses the following billable components of Google Cloud:

- Compute Engine
- Cloud Storage

To generate a cost estimate based on your projected usage, use the [pricing calculator](#) (/products/calculator). New Google Cloud users might be eligible for a [free trial](#) (/free-trial).

1. [Sign in](https://accounts.google.com/Login) (https://accounts.google.com/Login) to your Google Account.

If you don't already have one, [sign up for a new account](https://accounts.google.com/SignUp) (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a Cloud project.

★ **Note:** If you don't plan to keep the resources that you create in this procedure, create a project instead of selecting an existing project. After you finish these steps, you can delete the project, removing all resources associated with the project.

[Go to the project selector page](https://console.cloud.google.com/projectselector2/home/dashboard) (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](/billing/docs/how-to/modify-project) (/billing/docs/how-to/modify-project).

4. Enable the Compute Engine API.

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

When you finish this tutorial, you can avoid continued billing by deleting the resources you created. See [Cleaning up](#) (#cleaning-up) for more detail.

Create two instances for MySQL—a client and a server instance.

- Create a Compute Engine instance. Configure the instance as follows:
 - Name the instance `my-client`.
 - Set the `--zone` flag to the [zone](/compute/docs/zones#available) (/compute/docs/zones#available) in which you want to create your instance.
 - Set the `--image-project` flag to `ubuntu-os-cloud`.
 - Set the `--image-family` flag to `ubuntu-1804-lts`.

- Set the `--scopes` flag to `https://www.googleapis.com/auth/cloud-platform`.
- Create a Compute Engine instance. Configure the instance as follows:
 - Name the instance `my-server`.
 - Set the `--zone` flag to the zone (`/compute/docs/zones#available`) in which you want to create your instance.
 - Set the `--image-project` flag to `ubuntu-os-cloud`.
 - Set the `--image-family` flag to `ubuntu-1804-lts`.
 - Set the `--scopes` flag to `https://www.googleapis.com/auth/cloud-platform`.

The following steps describe how to install MySQL on a Compute Engine instance.

1. In the Cloud Console, go to the **VM instances** page.

[Go to the VM instances page](https://console.cloud.google.com/compute/instances) (`https://console.cloud.google.com/compute/instances`)

2. In the list of virtual machine instances, click **SSH** in the row of the instance that you want to connect to.

<input type="checkbox"/>	Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/>	<input checked="" type="checkbox"/> instance-1	us-east1-b		10.142.0.2 (nic0)	35.231.114.114 ↗	SSH ▾ ⋮

3. Update the `apt-get` package manager.

★ **Note:** If you receive an error **Could not get lock /var/lib/dpkg/lock**, wait another minute for the boot processes to complete and try again.

4. Install the MySQL client package.

The following steps describe how to install MySQL on a Compute Engine instance.

1. In the Cloud Console, go to the **VM instances** page.

[Go to the VM instances page \(https://console.cloud.google.com/compute/instances\)](https://console.cloud.google.com/compute/instances)

2. In the list of virtual machine instances, click **SSH** in the row of the instance that you want to connect to.

<input type="checkbox"/>	Name ^	Zone	Recommendation	Internal IP	External IP	Connect
<input type="checkbox"/>	instance-1	us-east1-b		10.142.0.2 (nic0)	35.231.114.114 ↗	SSH ▾ ⋮

3. Update the `apt-get` package manager.

★ **Note:** If you receive an error, "Could not get lock /var/lib/dpkg/lock", wait another minute for the boot processes to complete and try again.

4. Install the MySQL server package.

You must establish a root password for MySQL and perform basic security maintenance on your MySQL server configuration. For more information, see the MySQL documentation for [mysql_secure_installation](https://dev.mysql.com/doc/refman/5.0/en/mysql-secure-installation.html) (https://dev.mysql.com/doc/refman/5.0/en/mysql-secure-installation.html).

1. In Cloud Shell, improve the security of your MySQL installation.
2. Press `enter` to skip setting up the `VALIDATE PASSWORD` plugin.
3. Enter a new root password twice.
4. To remove anonymous users, enter `Y` and press `enter`.
5. To prevent remote root login, enter `Y` and press `enter`.
6. To remove the test database, enter `Y` and press `enter`.
7. To reload the privilege tables, enter `Y` and press `enter`.

Before you can remotely connect to the MySQL server, you need to configure it to listen on its internal IP address. Then, you create a non-root user account for the MySQL client to connect to the server.

All MySQL client commands must include certain command-line flags (for example, to authenticate). The MySQL commands in this section include the following flags: `--user` for the username, `-p` for the password, and `-e` to execute the given statement and immediately quit. For more information, see the [MySQL 5.7 command options reference](https://dev.mysql.com/doc/refman/5.7/en/mysql-command-options.html) (https://dev.mysql.com/doc/refman/5.7/en/mysql-command-options.html).

1. In Cloud Shell, use SSH to connect to the `my-server` instance.

2. Update the `/etc/mysql/mysql.conf.d/mysqld.cnf` configuration file with the following information:

3. Restart the MySQL service to apply the changes to the running server.

4. Verify that the server is running locally. Replace `[ROOT_PASSWORD]` with the MySQL server root password you established in a previous step.

The output appears similar to the following:

Remotely connecting as the root user was disabled with the preceding `mysql_secure_installation` command. You need to create a new user with the necessary permissions to allow remote connections.

1. In Cloud Shell, create an environment variable for the `my-client` internal IP address.

2. Create a new MySQL user with a password. Replace [MY_PASSWORD] with your password, [ROOT_PASSWORD] with your MySQL root user password.

3. Grant the new MySQL user permission to log on to the server from the internal IP address of my-client.

The my-server instance doesn't need an external IP address because the client can access my-server through an internal IP address.

- To remove the external IP address, update the configuration settings in Cloud Shell. Replace [ZONE] with your Google Cloud zone.

The following steps describe how to connect to the MySQL server on `my-server` from your `my-client` instance.

1. In Cloud Shell, use SSH to connect to `my-client` instance.
2. Test your connection by listing the databases.

Output appears similar to the following:

These steps verify that your MySQL client can successfully connect to the MySQL server over the internal IP address.

The default network configuration in Google Cloud includes a firewall rule, `default-allow-internal`, that allows internal traffic between Compute Engine instances on a wide range of ports, including the MySQL port, port `3306`. In non-default environments with an established security footprint, you might need to create a firewall rule to allow your `my-client` instance to communicate with your `my-server` instance over the network.

You can base firewall rules on IP address ranges or tags. IP address ranges are useful if you want to grant access to a wide range of internal IP addresses. Alternatively, if you want to grant access to specific instances on your network, tags provide a more flexible solution. Tags make it easier to add new clients without granting access to a wide range of IP addresses. You need only assign

the appropriate tag to the new MySQL client instance. For example, you can create a new firewall rule that allows traffic from all client instances that are tagged with `mysql-client`.

To support firewall rules using tags, you can assign the appropriate tags to the `my-client` and `my-server` VMs in Cloud Shell.

The following steps describe how to create a new firewall rule to enable instances with the `my-client` tag to communicate with instances that have the `my-server` tag by using port 3306.

- In Cloud Shell, create a firewall rule to allow communications from `mysql-client` to `mysql-server`.

You can now connect to MySQL from `my-client`.

This tutorial covers access from MySQL clients to MySQL servers both running on Compute Engine. It is beyond the scope of this tutorial to allow access from a client not running on Compute Engine. If you need to allow non-Compute Engine access, modify the following:

- Add an external IP address to `my-server` to allow external connectivity.
- Add the source IP address of your external client to the [firewall rules](#) (`#add_a_new_firewall_rule`).
- Modify the `TESTUSER` account, or create a user account, that is bound to the source IP address of your external client.

To avoid incurring charges to your Google Cloud account for the resources used in this tutorial, you can either [delete the project](#) (#delete_the_project) or [delete the instances](#) (#delete_instances).

The easiest way to eliminate billing is to delete the project that you created for the tutorial.

To delete the project:


! **Caution:** Deleting a project has the following effects:

- **Everything in the project is deleted.** If you used an existing project for this tutorial, when you delete it, you also delete any other work you've done in the project.
- **Custom project IDs are lost.** When you created this project, you might have created a custom project ID that you want to use in the future. To preserve the URLs that use the project ID, such as an `appspot.com` URL, delete selected resources inside the project instead of deleting the whole project.

If you plan to explore multiple tutorials and quickstarts, reusing projects can help you avoid exceeding project quota limits.

1. In the Cloud Console, go to the **Manage resources** page.

[Go to the Manage resources page](https://console.cloud.google.com/iam-admin/projects) (<https://console.cloud.google.com/iam-admin/projects>)

2. In the project list, select the project you want to delete and click **Delete** .


3. In the dialog, type the project ID, and then click **Shut down** to delete the project.

To delete a Compute Engine instance:

1. In the Cloud Console, go to the **VM Instances** page.

[Go to the VM Instances page](https://console.cloud.google.com/compute/instances) (<https://console.cloud.google.com/compute/instances>)

2. Click the checkbox for your **my-server** instance.

3. Click **Delete**  to delete the instance.

- Read about [high availability for MySQL](https://dev.mysql.com/doc/mysql-ha-scalability/en/ha-overview.html) (<https://dev.mysql.com/doc/mysql-ha-scalability/en/ha-overview.html>).
- Read about [How to set up MySQL on Compute Engine](/solutions/setup-mysql/) (/solutions/setup-mysql/).
- [Configure Stackdriver Logging for MySQL logs](/monitoring/agent/plugins/mysql). (/monitoring/agent/plugins/mysql)
- Explore [Cloud SQL](/sql/) (/sql/).
- Try out other Google Cloud features for yourself. Have a look at our [tutorials](/docs/tutorials) (/docs/tutorials).