

[Cloud SQL](https://cloud.google.com/sql/) (<https://cloud.google.com/sql/>)

[Documentation](https://cloud.google.com/sql/docs/) (<https://cloud.google.com/sql/docs/>)

[MySQL](https://cloud.google.com/sql/docs/mysql/) (<https://cloud.google.com/sql/docs/mysql/>) [Guides](#)

# About replicating from an external server

---

[MySQL](#) | [PostgreSQL](#) | [SQL Server](#)

This page describes a configuration that replicates data from a source database server to Cloud SQL Second Generation replicas. This configuration is sometimes referred to as an *external master* configuration.

The source database server can be any MySQL server, including servers running on other Google Cloud services (such as Compute Engine) or on other cloud providers (such as AWS RDS), provided they meet the [requirements](https://cloud.google.com/sql/docs/mysql/replication/replication-from-external#server-requirements) (<https://cloud.google.com/sql/docs/mysql/replication/replication-from-external#server-requirements>).

For step-by-step instructions for setting up this configuration, see [Replicating from an External Server](https://cloud.google.com/sql/docs/mysql/replication/replication-from-external) (<https://cloud.google.com/sql/docs/mysql/replication/replication-from-external>).

**Note:** This configuration supports only Second Generation replicas. For information about creating a First Generation replica from an external server, see [External Masters for First Generation](https://cloud.google.com/sql/docs/mysql/replication/configure-external-master) (<https://cloud.google.com/sql/docs/mysql/replication/configure-external-master>).

**Note:** Second Generation is replacing First Generation; support for First Generation instances ends January 30, 2020. To upgrade a First Generation instance to Second Generation, see [Upgrading a First Generation Instance to Second Generation](https://cloud.google.com/sql/docs/mysql/upgrade-2nd-gen) (<https://cloud.google.com/sql/docs/mysql/upgrade-2nd-gen>).

## Use cases

This configuration provides a way to achieve the following goals:

1. Migrate your data from your self-managed MySQL server to Google Cloud with a minimum of downtime.

For the migration use case, the replication configuration is temporary. For more information, see [Migrating data to Cloud SQL](#)

(<https://cloud.google.com/sql/docs/mysql/migrate-data#migrating-to-sql>).

2. Retain colocation and control of your server while off-loading the administration of the replicas to Cloud SQL.

This use case is sometimes called a *hybrid cloud*. Replication between your self-managed server and the Cloud SQL replica continues indefinitely.

## Configuration description

This configuration includes the following instances:

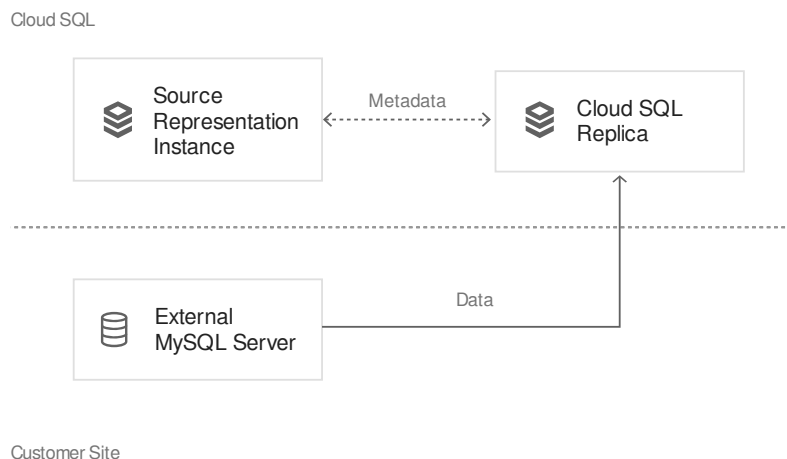
- The MySQL server that you manage, also called the *source database server*.
- The Cloud SQL replica.

There can be multiple replicas for a single source database server.

- The *source representation instance*.

The source representation instance is a Cloud SQL instance that represents the source database server to the Cloud SQL replica. It is visible in the Cloud Console and appears the same as a regular Cloud SQL instance, but it contains no data, requires no configuration or maintenance, and does not affect billing. You cannot update the source representation instance.

The following diagram shows these instances:



## SSL/TLS configuration

Replicating from an external server requires that all changes to the data are sent between the source database server and the Cloud SQL replicas using public networks. For this reason, you should always use SSL/TLS for the connections between the source database server and the replicas.

### SSL/TLS options

You have two options for the SSL/TLS configuration:

- Server-only authentication

When the replica connects to the master, the replica authenticates the master, ensuring that the replica is connecting to the correct host and preventing man-in-the-middle attacks. The master does not authenticate the replica.

- Server-client authentication

When the replica connects to the master, the replica authenticates the master and the master authenticates the replica.

Generally, you should use server-client authentication, which provides the strongest security. However, if you do not want to provide the client certificate and private key when you create the replica, you can still use server-only authentication.

### Preparing for server-only authentication

To use server-only authentication, you must provide (at replica creation time) the x509 PEM-encoded certificate of the certificate authority (CA) that signed the external server's certificate. The CA must contain only a single certificate, and it must be self signed. (In other words, the Certificate Authority that signed the server's certificate must be a root CA.)

For more information about creating certificates and keys for your external server, see [Creating SSL and RSA Certificates and Keys using MySQL](#)

(<https://dev.mysql.com/doc/refman/5.7/en/creating-ssl-rsa-files-using-mysql.html>).

### Preparing for server-client authentication

To use server-client authentication, you must provide the following items when you create the replica:

- The x509 PEM-encoded certificate of the CA that signed the source database server's certificate (`caCertificate`).
- The x509 PEM-encoded certificate that will be used by the replica to authenticate against the source database server (`clientCertificate`).
- The unencrypted [PKCS#1](https://tools.ietf.org/html/rfc3447) (<https://tools.ietf.org/html/rfc3447>) or [PKCS#8](https://tools.ietf.org/html/rfc5208) (<https://tools.ietf.org/html/rfc5208>) PEM-encoded private key associated with the `clientCertificate` (`clientKey`).

For more information about creating certificates and keys for your source database server, see [Creating SSL and RSA Certificates and Keys using MySQL](https://dev.mysql.com/doc/refman/5.7/en/creating-ssl-rsa-files-using-mysql.html) (<https://dev.mysql.com/doc/refman/5.7/en/creating-ssl-rsa-files-using-mysql.html>).

## About creating multiple replicas from the same source database server

You can create multiple replicas from the same source database server. You might want to provide more bandwidth, or create replicas in different regions.

If you are creating multiple replicas in the same region, they can all use the same source representation instance, or different ones. If you use the Cloud Console to create multiple replicas, they will have different source representation instances.

If you are creating multiple replicas in different regions, they must have different source representation instances.

You cannot create more than one replica in the same operation. As soon as you finish creating the replica configuration for the first replica, you can start [creating the replica configuration](https://cloud.google.com/sql/docs/mysql/replication/replication-from-external#setup) (<https://cloud.google.com/sql/docs/mysql/replication/replication-from-external#setup>) for the other replicas. You do not need to wait until the first replica is completely functional before starting to create other replicas. However, make sure you can complete the [network access authorization step](https://cloud.google.com/sql/docs/mysql/replication/replication-from-external#authorize-access) (<https://cloud.google.com/sql/docs/mysql/replication/replication-from-external#authorize-access>) within 15 minutes for each replica you create.

*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/) (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) (https://www.apache.org/licenses/LICENSE-2.0). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (https://developers.google.com/terms/site-policies). Java is a registered trademark of Oracle and/or its affiliates.*

*Last updated December 5, 2019.*