Replicating from an external server (external master)

This page describes how to create 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 that meets all of the server requirements (#server-requirements), including Cloud SQL instances. Instances hosted by other cloud providers can also serve as the data source, provided they meet the requirements, including supporting GTID.

For more detailed information about this configuration, see About Replicating from an External Server (https://cloud.google.com/sql/docs/mysql/replication/external-server).

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).

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).

Before you begin

Before you set up replication from an external server, you must complete the following steps:

- Sign in (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).

- In the Cloud Console, on the project selector page, select or create a Google 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/PROJECTSELECT

- Make sure that billing is enabled for your Google Cloud project. Learn how to confirm billing is enabled for your project (https://cloud.google.com/billing/docs/how-to/modify-project).
- Enable the Cloud SQL Admin API.

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

- Install and authenticate the `gcloud` command-line tool (https://cloud.google.com/sdk/downloads).
- Ensure that your server meets the requirements for the source database server (#server-requirements).
- Determine what level of security (https://cloud.google.com/sql/docs/mysql/replication/external-server#ssl-options) you require for the connections between the master and the replicas, and obtain the required certificate files.
- Decide which Google Cloud region (https://cloud.google.com/sql/docs/mysql/instance-locations) you want to use to contain your Cloud SQL replicas.
- Assemble the required information about your source database server:
  - External IPv4 address and port number

    By default, MySQL uses port 3306.
  - MySQL replication user account and password
  - MySQL version number
  - Location of all required SSL/TLS certificates and keys, depending on the level of security (https://cloud.google.com/sql/docs/mysql/replication/external-server#ssl-options) you selected
- Be prepared to update the network firewall for your source database server to accept connections from the Cloud SQL replica.

This step must be completed within 30 minutes of creating the replica.

- If you are not a project owner, you must have the `Storage Admin` role.

Requirements for the source database server

Before you can replicate from an external server to a Cloud SQL replica, you must ensure that the source database server meets these configuration requirements:

- Running MySQL Community Edition, version 5.6 or 5.7.
- Binary logs enabled. Learn more (https://dev.mysql.com/doc/refman/5.7/en/binary-log.html).
- Binary logs retained long enough for replica to complete the import.

  Generally, a week should be sufficient. Learn more about setting the binary log retention policy (https://dev.mysql.com/doc/refman/5.7/en/replication-options-binary-log.html#sysvar_expire_logs_days).
- Using row-based binary logging. Learn more (https://dev.mysql.com/doc/refman/5.7/en/replication-rbr-usage.html).
- GTID enabled, and GTID consistency enforced.

  Learn more about GTID (https://dev.mysql.com/doc/refman/5.7/en/replication-gtids.html). See the option for enforcing GTID consistency (https://dev.mysql.com/doc/refman/5.7/en/replication-options-gtids.html#sysvar_enforce_gtid_consistency).
- All tables (except tables in system databases) use the InnoDB storage engine.

  Learn more about InnoDB (https://dev.mysql.com/doc/refman/5.7/en/innodb-storage-engine.html). Learn more about converting to InnoDB (https://dev.mysql.com/doc/refman/5.7/en/converting-tables-to-innodb.html).
- MySQL user account with the `REPLICATION_SLAVE` privilege.

  Configure this account to accept connections from anywhere (host = `%`). You can restrict access to this user in a later step (#restrict-replication-user). You should not use this user account for any purpose other than replication.

Learn more about privileges (https://dev.mysql.com/doc/refman/5.7/en/privileges-provided.html#priv_replication-slave). Learn more about user accounts (https://cloud.google.com/sql/docs/mysql/users#mysql_user_account_format).

- An externally accessible IPv4 address and TCP port.

Configuration process

To set up replication from an external database server, you perform the following steps:

1. Create a Cloud Storage bucket for your data (#bucket).
2. Export your data to Cloud Storage (#export).
3. Set up the replication configuration (#setup).
4. Configure the source database server to accept connections from the replica (#authorize-access).

★ **Note:** this step must be done within 30 minutes of replica creation.

5. Restrict access to the MySQL replication user (#restrict-replication-user).
6. Finalize the replica configuration (#finalize-replica).
7. Confirm replication status (#confirm-status).
8. Clean up your storage (#delete-file).

1. Create a Cloud Storage bucket for your data

Create a bucket to temporarily hold your data during the export process, or use an existing bucket.

For more information about creating a bucket, see Creating Storage Buckets (https://cloud.google.com/storage/docs/creating-buckets). For more information about the Storage Admin role, see Cloud Storage IAM Roles (https://cloud.google.com/storage/docs/access-control/iam-roles).

2. Export your data to Cloud Storage

The replica looks for its data in Cloud Storage, so you put a copy of the server's data there. You can continue to accept write operations to your server after the export. After the replica finishes importing the dump file, it starts processing all changes since the export was taken and eventually catches up to the master.

Important: While mysqldump (https://dev.mysql.com/doc/refman/5.7/en/mysqldump.html) is running, do not perform any DDL operations (https://dev.mysql.com/doc/refman/5.7/en/glossary.html#glos_ddl) on the master. Doing so could cause inconsistencies in the export file.

From a machine with the **gcloud** tool installed and network connectivity to your MySQL server, run the following command:

```
mysqldump \
    -h [MASTER_IP] -P [MASTER_PORT] -u [USERNAME] -p \
    --databases [DBS]  \
    --hex-blob  --skip-triggers  --master-data=1  \
    --order-by-primary --no-autocommit \
    --default-character-set=utf8mb4 --ignore-table [VIEW] \
    --single-transaction --set-gtid-purged=on | gzip | \
    gsutil cp - gs://[BUCKET]/[PATH_TO_DUMP]

If the source of the migration is a Relational Database Service (RDS) for MySQL,
the master-data flag is not supported and should not be specified.
This command might look like the following example:

mysqldump \
    -h [MASTER_IP] -P [MASTER_PORT] -u [USERNAME] -p \
    --databases [DBS]  \
    --hex-blob  --skip-triggers \
    --order-by-primary --no-autocommit \
    --default-character-set=utf8mb4 --ignore-table [VIEW] \
    --single-transaction --set-gtid-purged=on | gzip | \
    gsutil cp - gs://[BUCKET]/[PATH_TO_DUMP]

Additionally, you should configure RDS instances to retain binlogs for a
a longer period of time.
This command might look like the following example:

// Sets the retention period to one day.
call mysql.rds_set_configuration('binlog retention hours', 24);
```

Note: You must use a version of mysqldump that supports the `--set-gtid-purged` flag. This is required for the export to be usable by the replica. The `mysqldump` utility bundled with MySQL versions 5.6 and 5.7 supports this flag.

Replace `[PROPERTIES_IN_BRACKETS]` with the following values:

Property	Value
`[MASTER_IP]`	The IPv4 address for the source database server.
`[MASTER_PORT]`	The port for the source database server.
`[USERNAME]`	The MySQL replication user account.
`[PASSWORD]`	The password for the MySQL replication user account.
`[DBS]`	Space-separated list of all databases on the source database server, except for the system databases (**`sys`**, **`mysql`**, **`performance_schema`**, and **`information_schema`**). Use the **`SHOW DATABASES`** MySQL command to list your databases.
`[VIEW]`	All views present in the databases you are dumping must be ignored. For multiple views, use this option multiple times.
`[BUCKET]`	The name of the bucket you created to hold the export file.
`[PATH_TO_DUMP]`	The path to the export file.

Note: The `gzip` command causes the dump file to be compressed. If your database contains data that does not compress well, such as binary incompressible data or JPG images, remove `gzip |` from the above command.

If you see a warning about partial dumps and GTIDs, that is expected.

This command might look like the following example:

```
mysqldump \
    -h 192.0.2.1 -P 3306 -u replicationUser \
    --databases guestbook  \
    --hex-blob  --skip-triggers  --master-data=1  \
    --order-by-primary --no-autocommit \
    --default-character-set=utf8mb4 --ignore-table view1 --ignore-table view2 \
```

```
  --single-transaction --set-gtid-purged=on | gzip | \
  gsutil cp - gs://export-bucket/hq-master1.sql.gz
```

3. Create the replication configuration

CONSOLE GCLOUD CURL

This step creates the Cloud SQL replica and the source representation instance (https://cloud.google.com/sql/docs/mysql/replication/external-server#config-description).

1. Open the Cloud SQL instance list in the Google Cloud Console.

   OPEN THE INSTANCE LIST (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/SQL/INSTANCES)

2. Click **Migrate data** in the button bar to open the Cloud SQL Migration Assistant.
3. Click **Begin migration**.
4. In the **Source database name** field, provide a name for the replication configuration in Cloud SQL.

   This name is a reference for the configuration, use any valid Cloud SQL instance name.
5. Enter the IP address and port number for your source database server.
6. Provide the user name and password for the MySQL user that will be used for the replication connection.
7. Select the MySQL version of your source database server.
8. If you are using SSL/TLS for the connection between the replica and the source database server (recommended), select **Enable SSL/TLS security** and provide the SSL/TLS certificate information for your source server.

   For more information about SSL/TLS options, see SSL/TLS options (https://cloud.google.com/sql/docs/mysql/replication/external-server#ssl-options).
9. Click **Next**, and fill in the details for your replica.

   To ensure that the import happens as efficiently as possible, configure the replica with roughly enough storage to contain the database. Size the machine type to be similar to the source server.
10. Provide the path to the dump file you uploaded to Cloud Storage earlier.
11. Click **Create** to create the replica.
12. Click **Next**.

When the read replica can access the dump file, it begins the import process. The duration of the import process depends on the dump file size, the replica machine type, and the database

schema. As a general guide for estimation, expect the replica to import approximately 25-50 GB per hour.

The replica stays in the `PENDING_CREATE` state until the import process is complete; at that time, the replica changes to the `RUNNABLE` state.

Important: After the import process completes, the replica uses the binary logs on the source server to catch up to the current state of the source server. The time this step takes depends on the size of your database and your database load. Make sure your binary log retention policy retains binary logs long enough to let the replica copy all the binary logs. A retention policy of a week should provide ample time.

4. Configure the source database server to accept connections from the replica

The replica needs to connect to the source database server for replication to succeed. If your source database server is behind a firewall or some other network restriction, you must enable network access for the replica, using the replica's `OUTGOING` IP address. Note that this is *not* the IP address displayed in the main listing for the replica in the Cloud Console. You can retrieve the `OUTGOING` IP address by hovering over the **More info** tool tip for the IP address, or by using the `gcloud` command below.

Important: You must complete this step within 30 minutes of creating the replica. If you do not, the replica will not be successfully created, and you will need to recreate the replica.

1. Retrieve the read replica's IP addresses:

```
gcloud sql instances describe [REPLICA_NAME] --format="default(ipAddresses)"
```

2. Configure the network firewall for your source database server to accept connections from the `OUTGOING` IP address.
3. Confirm that the replica has successfully connected to the source database server.
    a. The replica's icon in the instance listing page should no longer be spinning, and it should be green.
    b. Go to the Logs Viewer in the Google Cloud Console.

    GO TO THE LOGS VIEWER (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/LOGS?RESOURCE=CLOU

    c. Select the replica from the **Instance** dropdown.

d. Select the `replication-setup.log` log file.

If the replica is not able to connect to the source database server, confirm the following items:

- Any firewall on the source database server is configured to allow connections from the replica's `OUTGOING` IP address.
- Your SSL/TLS configuration is correct.
- Your replication user, host, and password are correct.

5. Restrict access to the MySQL replication user

This step is optional but recommended for security.

The MySQL replication user on the source database server is configured to accept connections from any host (`%`). Update that user account to accept connections only from the replica's `OUTGOING` IP address:

```
UPDATE mysql.user SET Host='[OUTGOING_IP]' WHERE Host='%' AND User='[USERNAME]';
FLUSH PRIVILEGES;
```

6. Finalize the replica configuration

1. Configure a user account on the replica.

   You can do this by using the Google Cloud Console, the `gcloud` command-line tool, or the Cloud SQL API. However, you cannot use the `mysql` client. For more information about MySQL users, see MySQL users (https://cloud.google.com/sql/docs/mysql/users).

   To create a user using the Cloud Console:

   a. Go to the Cloud SQL Instances page in the Google Cloud Console.

   GO TO THE CLOUD SQL INSTANCES PAGE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/SQL/INS

   b. Click the instance name to open its **Instance details** page.

   c. Select the **Users** tab.

   d. Click **Create user account**.

   e. In the **Create user account** dialog, specify the details for your user.

   f. Click **Create**.

2. Authorize access to the replica from any client you plan to use to connect to the replica using IP addresses.

   For instructions, see Adding an authorized address or address range (https://cloud.google.com/sql/docs/mysql/configure-ip#add). To connect to the replica, you use the replica's `PRIMARY` IP address. This IP address is displayed in the Cloud Console.

   You can also use any other connection method (https://cloud.google.com/sql/docs/mysql/how-to#connecting) to connect to the replica.

7. Confirm replication status

When the replica finishes importing the dump file, it connects to the on- premises server and applies all the updates that were made after the export was taken. You can confirm its status in the Instance listing page of the Cloud Console.

1. Go to the Cloud SQL Instances page in the Google Cloud Console.

   GO TO THE CLOUD SQL INSTANCES PAGE (HTTPS://CONSOLE.CLOUD.GOOGLE.COM/SQL/INSTANCE

2. Click **Show Info Panel** and select **Replication Delay** from the metrics dropdown list.

   Replication delay should be at or trending towards 0. If it is not, you should take steps to address it (https://cloud.google.com/sql/docs/mysql/high-availability#replication-lag-address).

8. Clean up your storage

1. Delete the export file:

   ```
   gsutil rm gs://[BUCKET]/[PATH_TO_DUMP]
   ```

2. If you no longer need the bucket, delete the bucket:

   ```
   gsutil rm -r gs://[BUCKET]
   ```

For more information, see the Cloud Storage documentation for Deleting Objects (https://cloud.google.com/storage/docs/deleting-objects) and Deleting Buckets (https://cloud.google.com/storage/docs/deleting-buckets).

What's next

- Learn about managing replicas (https://cloud.google.com/sql/docs/mysql/replication/manage-replicas).
- Learn about promoting your Cloud SQL replica to migrate your data to Google Cloud (https://cloud.google.com/sql/docs/mysql/migrate-data#migrating-to-sql).

Last updated November 19, 2019.