

Cloud SQL instances can have multiple types of replication instances for a master instance. Failover replicas provide [high availability configuration](/sql/docs/high-availability) and [semisynchronous replication](https://dev.mysql.com/doc/refman/5.6/en/replication-semisync.html) of data in an instance that can be switched over to in the event of a zone outage. Read replica instances provide data replication in order to make a failover to create a new, semi-identical, master instance.

This page describes how to use PowerShell cmdlets to perform operations on replication instances that exist in your Cloud Console project—operations such as falling back to the failover replica, or stopping replication in a read replica. You should use these operations if a Cloud SQL instance or zone experiences problems or suffers an outage. However, managing replicas should be done with care so as to not disrupt the existing data or future data.

For a more in-depth discussion of replication, see [Replication Options](/sql/docs/replication/) and [Requirements and Tips for Configuring Replication](/sql/docs/replication/tips). See [Setting Up Cloud SQL Instances](/tools/powershell/docs/sql/setup) for details on how to set up both a failover replica instance and a read replica.

You can start or stop data replication in an instance's read replica instance(s). The following snippet starts replication for the read replica instance **myreplica**, in order to replicate all the data in a master instance and ensure that it is backed up:

In the event data should no longer be replicated, you can similarly stop it:

If a read replica instance has all the replicated data it needs and you want to turn it into a standalone instance (for example, to diverge on its data or perform analysis on it), use the cmdlet `ConvertTo-GcSqlInstance`:

Sometimes mistakes occur and an instance becomes unusable. Using the [high availability configuration](#) (/sql/docs/high-availability) options (through the use of failover replica instances), you can failover to the failover replica instance for a master instance. Note this process happens automatically if the master instance's zone suffers an outage.

The following code snippet activates the failover replica **myfailover**, causing it to become the master instance:

Note that this snippet includes a variable called `settingVersion`, which contains a version number. This number prevents race conditions from occurring when updating an instance or using a failover, since the version number in the call must match the instance's current version.