This page describes how to set up a new Cloud SQL instance using Cloud Tools for PowerShell. Read the Cloud Tools for PowerShell cmdlet reference (http://googlecloudplatform.github.io/google-cloud-powershell/) to learn more about the cmdlets.

For most cmdlets you can specify a project ID, but if you omit it, PowerShell defaults to whatever is specified in the active Cloud SDK configuration (/sdk/docs/configurations).

A Cloud SQL instance is a MySQL database running in the cloud. You can use Cloud SQL instances to store, replicate, and protect your MySQL databases. You can configure the instance's behavior, such as when and where the data will be replicated, or when it is acceptable to perform database maintenance.

To create and start a Cloud SQL database, you first must specify its configuration and settings. To do that using Cloud Tools for PowerShell, use the `New-GcSqlSettingConfig` and `New-GcSqlInstanceConfig` cmdlets.

SQL settings, such as logging, maintenance windows, and so on, are parameters of the `New-GcSqlSettingConfig` cmdlet. Instance-level settings, such as the hardware characteristics of the database machine, are set via the `SqlInstanceConfig` object, or flags on the `New-GcSqlInstanceConfig` cmdlet.

Specifying no parameters creates the most basic Cloud SQL instance, using the "db-n1-standard-1" tier:

For information on tiers and pricing, see the Cloud SQL Pricing (/sql/pricing) topic.

For information on configuration options, use `Get-Help New-GcSqlSettingConfig` or see the DatabaseInstance (https://developers.google.com/resources/api-libraries/documentation/sqladmin/v1beta4/csharp/latest/classGoogle_1_1Apis_1_1SQLAdmin_1_1v1beta4_1_1Data_1_1DatabaseInstance.html) object documentation.

Once you have performed the configuration and setup described in the previous section, you can now create the Cloud SQL database using the `Add-GcSqlInstance` cmdlet:

Billing begins the moment an instance is created, regardless of if it is populated with MySQL databases or not.

Creating a new instance may take a minute or so. Once the process finishes, the cmdlet returns a `DatabaseInstance` object describing the instance that has just been created. You can go to the Cloud Console to see the instance.

From this point on, you can manage the database as described in the Cloud SQL documentation (/sql/docs/mysql-client).

Read replica (/sql/docs/replication/create-replica) instances provide replication functionality for the data in a master instance. After creation, they can be made into a failover or into their own standalone instance. Data can be read from read replica instances but it may be semi-out of date, in accordance with semisynchronous replication (https://dev.mysql.com/doc/refman/5.6/en/replication-semisync.html). By backing up data in multiple regions through read replica instances, data can still be read in the case of an outage.

The following code snippet creates a read replica **mynewreplica** for the already existing master instance **gootoso**. The snippet uses the cmdlet `New-GcSqlInstanceReplicaConfig`. This configures settings specific to read replica instances, such as how fast they replicate data or if they replicate from an external instance:

For more information on read replica requirements, see Requirements and Tips for Configuring Replication (/sql/docs/replication/tips).

For a demonstration of how to promote a read replica to be a failover or standalone instance, see Managing Instance Replications (/tools/powershell/docs/sql/replica).

Cloud SQL provides high availability configuration (/sql/docs/high-availability) for instances through the use of failover replicas. Failover replicas use semisynchronous replication (https://dev.mysql.com/doc/refman/5.6/en/replication-semisync.html) in order to replicate all changes in data in the master instance. If the zone in a master instance experiences an outage, the failover replica is activated and takes control. For this reason, the replica must be in a different zone than the master.

The following code snippet creates a new failover replica **myfailover** for the master instance **gootoso**. The replica instance is created the same way as above, using the config cmdlets. However, note the addition of the `-FailoverTarget` flag:

Note that replica instances impact billing the same way that normal instances do.