MySQL (/sql/docs/mysql/backup-recovery/backing-up) | **PostgreSQL** | SQL Server (/sql/docs/sqlserver/backup-recovery/backing-up)

ng: All data on an instance, including backups, is permanently lost when that instance is deleted. To preserve your da it to Cloud Storage (/sql/docs/postgres/import-export/exporting) before you delete it.

This page describes how to schedule and disable automatic backups, create and manage on-demand backups, and view backups for your Cloud SQL instance.

For more information about how backups work, see Overview of backups (/sql/docs/postgres/backup-recovery/backups).

To create an on-demand backup:

To schedule automated backups for an instance:

To disable automated backups for an instance:

Currently, you can only use the API to set custom locations for backups.

Backups fail if they are in a region the organization policy does not allow. Backups taken in the same region as the ce always succeed, regardless of the organization policy or other constraints.

To set the backup location for *automated* backups for an existing instance, use the instances `settings.backupConfiguration.location` (/sql/docs/mysql/admin-api/v1beta4/instances) parameter:

To set the backup location for *on-demand* backups, use the backupRuns `location` (/sql/docs/postgres/admin-api/v1beta4/backupRuns) parameter:

To see the locations of the backups taken for an instance, use backupRuns:

In the previous examples:

- `REGION` is the region for the original instance.
- `BACKUP_REGION` is the region where you want Cloud SQL to store the backup. For a complete list of valid region values, see the Instance locations (/sql/docs/postgres/instance-locations).

The Cloud Console does not show skipped or failed backups in the backup history. To see skipped backups, use the gcloud command line tool or the API.

You can delete automated and on-demand backups.

Deleting a backup might not free up as much space as the size of the backup. This is because backups are incremental, so deleting an older backup might transfer some of its contents to a more recent backup to preserve the integrity of the newer backup.

- Learn more about backing up your data (/sql/docs/postgres/backup-recovery/backups).
- Learn more about restoring data (/sql/docs/postgres/backup-recovery/restore).
- See how to restore your data from a backup (/sql/docs/postgres/backup-recovery/restoring#restorebackups).