

[MySQL](#) (/sql/docs/mysql/configure-cmek) | **PostgreSQL** | [SQL Server](#)
(/sql/docs/sqlserver/configure-cmek)

This page describes how to set up a service account and keys for customer-managed encryption keys, and how to create an instance that uses a customer-managed encryption key. To learn more about using customer-managed encryption keys with Cloud SQL, see [Overview of customer-managed encryption keys](#) (/sql/docs/postgres/cmek).

1. [Sign in](https://accounts.google.com/Login) (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) (https://accounts.google.com/SignUp).

2. In the Cloud Console, on the project selector page, select or create a 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/projectselector2/home/dashboard) (https://console.cloud.google.com/projectselector2/home/dashboard)

3. Make sure that billing is enabled for your Google Cloud project. [Learn how to confirm billing is enabled for your project](#) (/billing/docs/how-to/modify-project).

4. [Install and initialize the Cloud SDK](#) (/sdk/docs/).

5. Make sure you have the Cloud SQL Admin role on your user account.

[GO TO THE Cloud IAM PAGE](https://console.cloud.google.com/iam-admin/iam) (https://console.cloud.google.com/iam-admin/iam)

6. Enable the Cloud Key Management Service API.

[Enable the API](https://console.cloud.google.com/flows/enableapi?apiid=cloudkms.googleapis.com) (https://console.cloud.google.com/flows/enableapi?apiid=cloudkms.googleapis.com)

1. Create a service account (#service-account) for each project that requires customer-managed encryption keys.
2. Create a keyring and key (#key), and set the location for each key.
3. Grant the key access (#grantkey) to the service account.
4. Note the key ID (KMS_RESOURCE_ID) and location. You need this information to create instances enabled with CMEK.
5. Go to a project and create a Cloud SQL instance (#createcmekinstance) with the following options:
 - a. The same location as the customer-managed encryption key
 - b. The customer-managed key configuration
 - c. The customer-managed encryption key ID

Your Cloud SQL instance is now enabled with CMEK.

To create a service account with the required permissions, you must have **resourcemanager.projects.setIamPolicy** permission. This permission is included in the Project Owner, Project IAM Admin, and Organization Administrator roles.

You must also enable the Cloud SQL Admin API.

You need to create a service account for each project that requires customer-managed encryption keys. Currently, you can only use `gcloud` command-line tool commands to create the type of service account you need for customer-managed encryption keys.

To create a service account with `gcloud` command-line tool, run the following command:

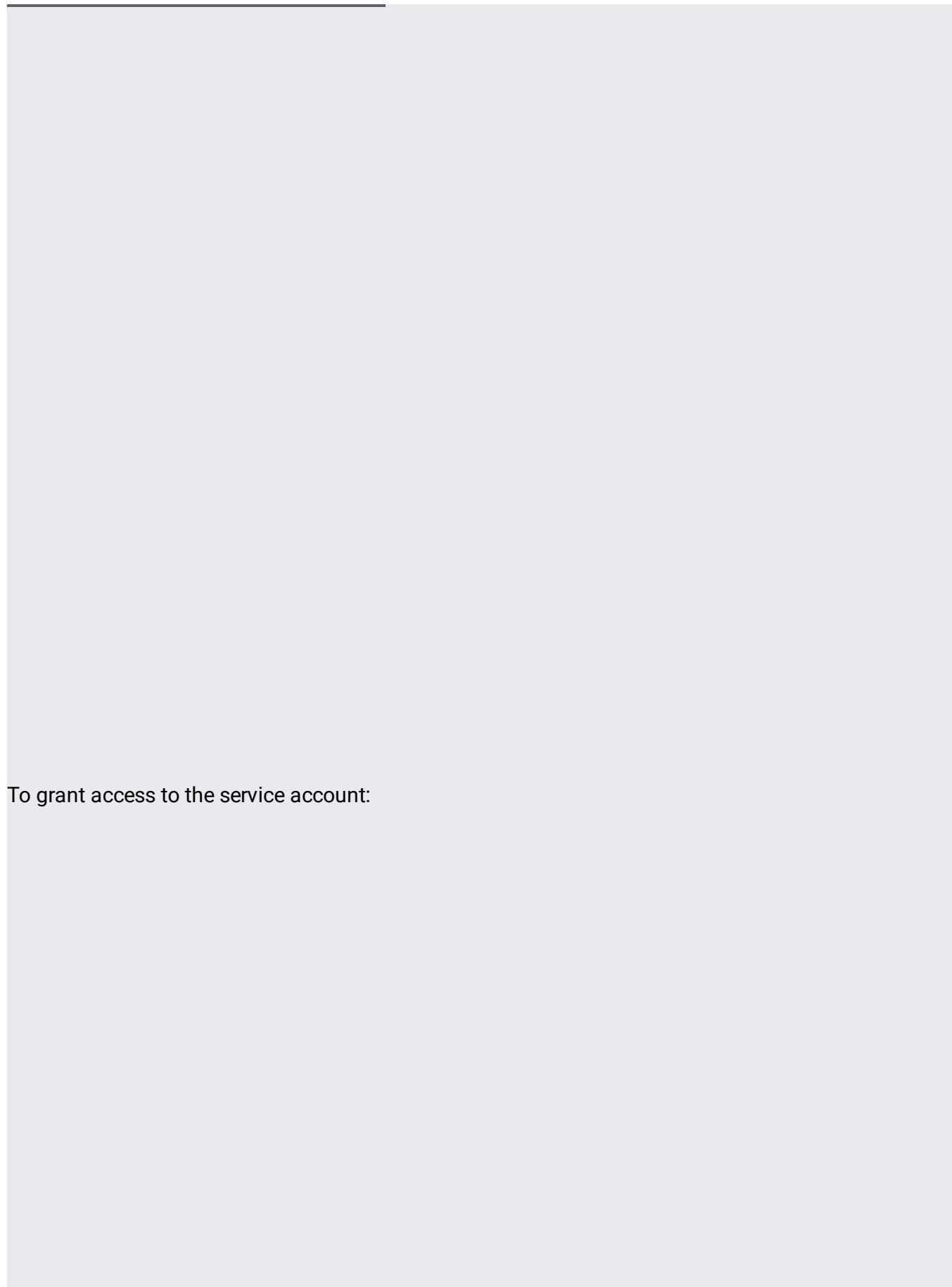
The previous command returns a service account name, using the following format:

For example:

You use this service account name during the procedure in [Granting the key access to the service account](#) (#grantkey).

You can create the key in the same GCP project as the Cloud SQL instance or in a separate user project. The Cloud KMS key ring location must match the region where you want to create Cloud SQL instance. A multi-region or global region key **will not** work. The Cloud SQL instance create request fails if the regions don't match.

To create a Cloud KMS key:



To grant access to the service account:

Currently, you can only use `gcloud` command-line tool commands and `cURL` to create Cloud SQL instances with CM

To create an instance with customer-managed encryption keys:

This cURL command uses [Instances:Insert](#) (/sql/docs/postgres/admin-api/v1beta4/instances/insert).

When you create a backup of a Cloud SQL instance, the backup is encrypted with the same customer-managed encryption key as the primary instance.

See [Creating and managing on-demand and automatic backups](#)

(/sql/docs/postgres/backup-recovery/backing-up). The only change is that you see a warning on the **Create a backup** page that says: "Your backup will be encrypted with this instance's customer-managed encryption key. If anyone destroys this key, all data encrypted with the key will be permanently lost."

When you create a read replica of a Cloud SQL instance, it inherits the same customer-managed encryption key as the parent instance.

See [Creating read replicas \(/sql/docs/postgres/replication/create-replica\)](/sql/docs/postgres/replication/create-replica). The only change is that you see a warning on the **Create read replica** page that says: "Your replica will be encrypted with the master instance's customer-managed key. If anyone destroys this key, all data encrypted with it will be permanently lost."

When you create a clone of Cloud SQL instance, it inherits the same customer-managed encryption key as the one used to encrypt the source instance.

See [Cloning instances \(/sql/docs/postgres/clone-instance\)](/sql/docs/postgres/clone-instance). The only change is that you see a warning on the **Create clone** page: "Your clone will be encrypted with the source instance's customer-managed key. If anyone destroys this key, all data encrypted with it will be permanently lost."

See the following topics:

- [Disable an enabled key version \(/kms/docs/enable-disable#disable_an_enabled_key_version\)](/kms/docs/enable-disable#disable_an_enabled_key_version)
- [Enable a disabled key version \(/kms/docs/enable-disable#enable_a_disabled_key_version\)](/kms/docs/enable-disable#enable_a_disabled_key_version)

This section describes things to try when you get an error message while setting up or using CMEK-enabled instances.

Cloud SQL administrator operations, such as create, clone, or update, might fail due to Cloud KMS errors, and missing roles or permissions. Common reasons for failure include a missing Cloud KMS key version, a disabled or destroyed Cloud KMS key version, insufficient IAM permissions to access the Cloud KMS key version, or the Cloud KMS key version is in a different region than the Cloud SQL instance. Use the following troubleshooting table to diagnose and resolve common problems.

For this error...	The issue might be...	Try this...
-------------------	-----------------------	-------------

Per-product per-project service account not found	The service account name is incorrect.	Make sure you created a service account for the correct user project. GO TO THE SERVICE ACCOUNTS PAGE (https://console.cloud.google.com/iam-service-accounts)
Cannot grant access to the service account	The user account does not have permission to grant access to this key version.	Add the Organization Administrator role on your user or service account. GO TO THE IAM ACCOUNTS PAGE (https://console.cloud.google.com/iam-service-accounts)
Cloud KMS key version is destroyed	The key version is destroyed.	If the key version is destroyed, you cannot use it to encrypt or decrypt.
Cloud KMS key version is disabled	The key version is disabled.	Re-enable the Cloud KMS key version. GO TO THE CRYPTO KEYS PAGE (https://console.cloud.google.com/iam-service-accounts)
Insufficient permission to use the Cloud KMS key	The <code>cloudkms.cryptoKeyEncrypterDecrypter</code> role is missing on the user or service account you are using to run operations on Cloud SQL instances, or the Cloud KMS key version doesn't exist.	Add the <code>cloudkms.cryptoKeyEncrypterDecrypter</code> role on your user or service account. GO TO THE IAM ACCOUNTS PAGE (https://console.cloud.google.com/iam-service-accounts) If the role is already on your account, see Creating a key (/sql/docs/postgres/configure-cmek#key) . See note.
Cloud KMS key is not found	The key version does not exist.	Create a new key version. See Creating a key (/sql/docs/postgres/configure-cmek#key) .
Cloud SQL instance and Cloud KMS key in different regions	The Cloud KMS key version and Cloud SQL instance must be in the same region. It does not work if the Cloud KMS key version is in a global region or multi-region.	Create a key version in the same region where you want to create instances. See Creating a key (/sql/docs/postgres/configure-cmek#key) . See note.
If the instance is in a failed state during the <code>create</code> operation, you must delete it, add the role to the account you are using, and create a new instance with a active Cloud KMS key version.		

- [Configure access to the instance \(/sql/docs/postgres/instance-access-control\)](/sql/docs/postgres/instance-access-control).
- [Connect to the instance with a client \(/sql/docs/postgres/connect-admin-ip\)](/sql/docs/postgres/connect-admin-ip).
- [Create a database on the instance \(/sql/docs/postgres/create-manage-databases\)](/sql/docs/postgres/create-manage-databases).
- [Import data into the instance \(/sql/docs/postgres/import-export/importing\)](/sql/docs/postgres/import-export/importing).
- [Create users on the instance \(/sql/docs/postgres/create-manage-users\)](/sql/docs/postgres/create-manage-users).
- [Learn more about instance settings \(/sql/docs/postgres/instance-settings\)](/sql/docs/postgres/instance-settings).