MySQL (/sql/docs/mysql/cmek) | **PostgreSQL** | SQL Server (/sql/docs/sqlserver/cmek)

This page describes how customer-managed encryption keys work with Cloud SQL. To use this feature right away, see Using customer-managed encryption keys (CMEK) (/sql/docs/postgres/configure-cmek).

Customer-managed encryption keys are intended for organizations that have sensitive or regulated data that requires them to manage their own encryption key.

The customer-managed encryption keys feature lets you use your own cryptographic keys for data at rest in Cloud SQL. After adding customer-managed encryption keys, whenever an API call is made, Cloud SQL uses your key to access data.

Cloud SQL uses Google-managed data encryption keys (DEK) and key encryption keys (KEK) to encrypt Cloud SQL. There are two levels of encryption:

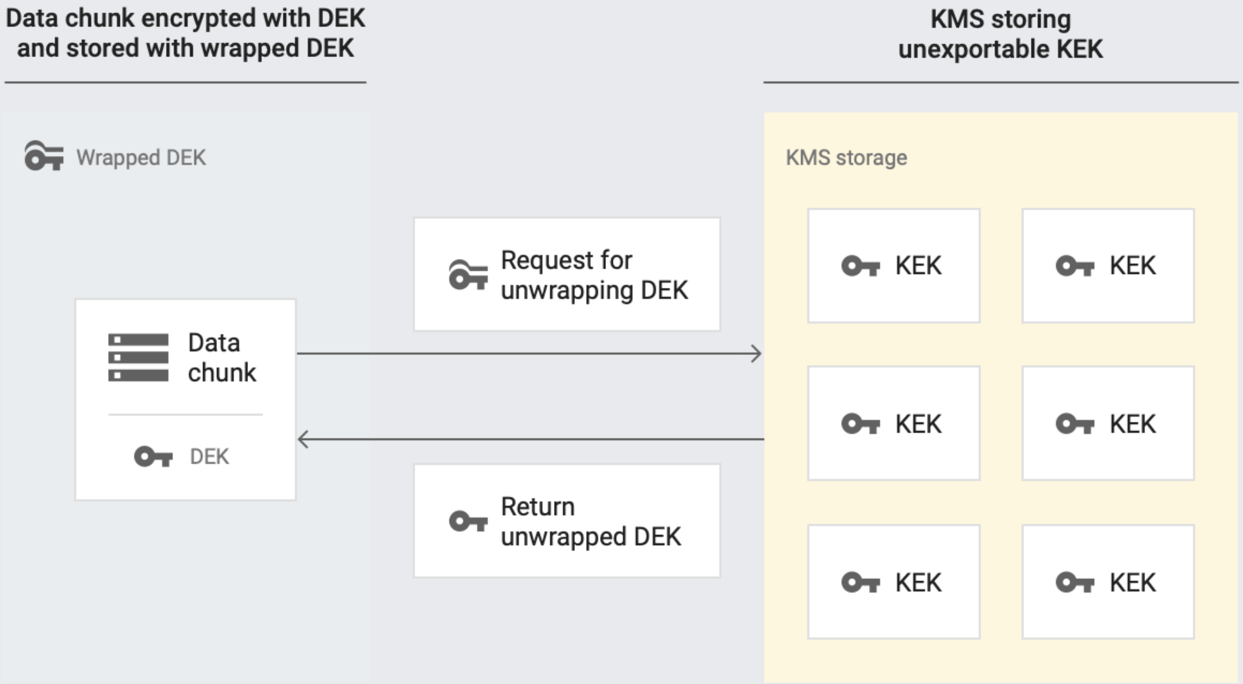
1. The DEK encrypts data.
2. The KEK encrypts the DEK.

The Cloud SQL instance stores the encrypted DEK alongside the encrypted data on the PD and Google manages the Google KEK. With customer-managed encryption keys, you create a key that wraps the Google KEK. Customer-managed encryption keys let you create, revoke, and delete the KEK.

Customer-managed encryption keys are stored within a managed service called Cloud Key Management Service (Cloud KMS) (/kms/docs).

The diagrams below show how data-at-rest encryption (/security/encryption-at-rest/) works inside a Cloud SQL instance when using default Google encryption versus customer-managed encryption keys.

When decrypting data wrapped with customer-managed encryption keys, Cloud SQL uses the KEK to decrypt the DEK and the unencrypted DEK to decrypt data-at-rest.


Data chunk encrypted with DEK and stored with wrapped DEK
KMS storing unexportable KEK
Wrapped DEK
KMS storage
Request for unwrapping DEK
Data chunk
DEK
Return unwrapped DEK
KEK
KEK
KEK
KEK
KEK
KEK

Customer-managed encryption keys differs from Customer-Supplied Encryption Keys (CSEK), which let you specify t *ts* of the encryption key. Customer-managed encryption keys let you create and manage a key using Cloud KMS, and keys to specific resources across Google Cloud. Cloud SQL does not support CSEK.

Operation	Notes
Instance creation	During instance creation, you configure the instance to use customer- managed encryption keys.
Backup creation	During backups for a CMEK-enabled instance, customer-managed encryption keys encrypt user data, such as user queries and responses. Backups from a CMEK-enabled instance inherit its encryption with same Cloud KMS key as the source instance.
Instance restore	During restores for a CMEK-enabled instance, Cloud SQL uses the key to access data on the backup instance being restored. When restoring to a different instance, the target instance can use a different key for encryption.
Replica creation	Read replicas from a CMEK-enabled instance inherit CMEK encryption with the same Cloud KMS key as the master instance.
Clone creation	Clones from a CMEK-enabled instance inherit CMEK encryption with same Cloud KMS key as the source instance.
Instance update	During updates to a CMEK-enabled instance, Cloud SQL checks the CMEK key.

CMEK is available in all Cloud SQL instance locations (/sql/docs/mysql/locations).

When your Cloud SQL instances have CMEK enabled, you need to use a service account to request key access from kms_name_short.

To use a customer-managed encryption key on a project, you must create a service account and grant the customer-managed encryption key access to the service account. You create the service account inside of the project. The service account is visible in all regions.

In Cloud KMS, you need to create a keyring with a cryptographic key, set with a location. When you create a new Cloud SQL instance, you select this key to encrypt the instance.

You need to know the key ID and key region when you create new Cloud SQL instances that use customer-managed encryption keys. You must put new Cloud SQL instances in the same region as the customer-managed encryption key associated with the instance. You can create one project for both keys and Cloud SQL instances, or different projects for each.

Customer-managed encryption keys use the following format:

When you disable the key version, Cloud SQL suspends the instances encrypted with that key version. When you re-enable the key version, Cloud SQL resumes the instances encrypted with that key version.

When you rotate keys, instances encrypted with that key are not re-encrypted with the new primary key version.

You might have situations where you want to permanently destroy data encrypted with CMEK. To do this, you destroy the customer-managed encryption key version. You cannot destroy the keyring or key, but you can destroy key versions of the key.

ng: You have complete control over keys and data access. Once you destroy a key version that is associated with a C
stance, Google can't get the data back. However, if you disable the key version, you can re-enable it to get the data ba

There are no special requirements or restrictions to importing data to a new instance, when the data was previously stored on a CMEK-enabled instance.

The following restrictions apply when using customer-managed encryption keys:

- You cannot rotate key versions on existing instances.
- You cannot assign a different key version to a replica.
- You cannot assign a different key version to a clone.
- You cannot use customer-managed encryption keys to encrypt:
    - External servers (external master instances and external replicas)
    - Instance metadata, such as the instance ID, database version, machine type, flags, backup schedule, etc.
- You cannot use customer-managed encryption keys to encrypt user data in transit, such as user queries and responses.

- Learn about encryption at rest in Google Cloud Platform (/security/encryption-at-rest/default-encryption/).
- Learn how to create an instance with CMEK enabled (/sql/docs/postgres/configure-cmek).
- Learn about Cloud Key Management Service (Cloud KMS) (/kms/docs).
- Learn about IAM service accounts (/iam/docs/service-accounts).
- Find other products (/kms/docs/using-other-products) that use CMEK.