Creating instances

cloud.google.com/sql/docs/mysql/create-instance

MySQL | PostgreSQL | SQL Server
This page describes how to create a Cloud SQL for a MySQL instance.

For detailed information about all instance settings, see Instance Settings.

A newly-created instance has four system databases: `information_schema`, `mysql`, `performance_schema`, and `sys`.

You can add additional databases by creating or importing them.

Before you begin

1. In the Google Cloud Console, on the project selector page, select or create a Google 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 project selector

2. Make sure that billing is enabled for your Cloud project. Learn how to confirm that billing is enabled for your project.

3. Install and initialize the Cloud SDK.
4. Make sure you have the Cloud SQL Admin and Compute Viewer roles on your user account.
   Go to the IAM page

   Learn more about roles and permissions.

Note: By default, you can have up to 100 instances per project. If you need more, file a support case to request the increase.

Creating a MySQL instance

To create a MySQL instance:

1. Go to the Cloud SQL Instances page in the Google Cloud Console.
   Go to the Cloud SQL Instances page

2. Click **Create instance**.
3. Select **MySQL** and click **Next**.

4. Enter a name.
   Do not include sensitive or personally identifiable information in your instance name; it is externally visible.

   You do not need to include the project ID in the instance name. This is done automatically where appropriate (for example, in the log files).

   Note: You cannot reuse an instance name for up to a week after you have deleted the instance.
5. Enter the password for the `'root'@'%'` user.
6. Select the database version for your instance: MySQL 8.0, MySQL 5.7 (default), or MySQL 5.6.
   The database version cannot be edited after the instance has been created.

7. Under **Choose region and zonal availability**, select the region and zone for your instance.
   Place your instance in the same region as the resources that access it. The region you select can't be modified in the future. In most cases, you don't need to specify a zone.

   Note: If there is a resource location constraint on your organization policy, you must select one of the regions that the organization policy allows. You see a message about Resource Location Restriction in the **Choose region and zonal availability** section if a constraint exists. Learn more.
   If you are configuring your instance for high availability, you can select both a primary and secondary zone.

   The following conditions apply when the secondary zone is used during instance creation:

   - The zones default to `Any` for the primary zone and `Any (different from primary)` for the secondary zone.
   - If both the primary and secondary zones are specified, they must be distinct zones.

8. Under **Customize your instance**, update settings for your instance. Begin by clicking **SHOW CONFIGURATION OPTIONS** to display the groups of settings. Then, expand desired groups to review and customize settings. A **Summary** of all the options you select is shown on the right.
The following table is a quick reference to instance settings. For more details about each setting, see the instance settings page.

Setting	Notes
	Machine type
Machine type	Select from Shared core, Lightweight, Standard (Most common), or High memory.
Custom	Select this button to create an instance with a flexible configuration. When you select this option, you need to select the number of cores and amount of memory for your instance. Learn more.
Cores	The number of vCPUs for your instance. Learn more.
Memory	The amount of memory for your instance, in GBs. Learn more.
	Storage
Storage type	Determines whether your instance uses SSD or HDD storage. Learn more.
Storage capacity	The amount of storage provisioned for the instance. Learn more.
Enable automatic storage increases	Determines whether Cloud SQL automatically provides more storage for your instance when free space runs low. Learn more.
	Encryption
Google-managed encryption	The default option.
Customer key-managed encryption key (CMEK)	Select to use your key with Google Cloud Key Management Service. Learn more.
	Connectivity
Private IP	Adds a private IP address for your instance. To enable connecting to the instance, additional configuration is required. Learn more.
Public IP	Adds a public IP address for your instance. You can then add authorized networks to connect to the instance. Learn more.

Setting	Notes
Authorized networks	Add the name for the new network and the Network address. Learn more.
Backups	
Automate backups	The window of time when you would like backups to start. Learn more.
Choose where to store your backups	Select Multi-region for most use cases. If you need to store backups in a specific region, for example, if there are regulatory reasons to do so, select Region and select your region from the Location drop-down menu.
Choose how many automated backups to store	The number of automated backups you would like to retain (from 1 to 365 days). Learn more.
Enable point-in-time recovery	Enables point-in-time recovery and write-ahead logging. Learn more.
Choose how many days of logs to retain	Configure write-ahead log retention from 1 to 7 days. The default setting is 7 days. Learn more.
Maintenance	
Preferred window	Determines a one-hour window when Cloud SQL can perform disruptive maintenance on your instance. If you do not set the window, then disruptive maintenance can be done at any time. Learn more.
Order of updates	Your preferred timing for instance updates, relative to other instances in the same project. Learn more.
Flags	
ADD FLAG	You can use database flags to control settings and parameters for your instance. Learn more.
Labels	
ADD LABEL	Add a key and value for each label that you add. You use labels to help organize your instances.

9. Click **Create**.

To see how the underlying REST API request is constructed for this task, see the APIs Explorer on the instances:insert page.

Flexible instance configurations

Flexible instance configurations let you select the amount of memory and CPUs that your instance needs. This flexibility lets you choose the appropriate VM shape for your workload. Machine type names use the format db-custom-*CPU*-*RAM*, where *CPU* is the number of CPUs in the machine, and *RAM* is the amount of memory in the machine.

When selecting the number of CPUs and amount of memory, there are some restrictions on the configuration you choose:

- vCPUs must be either 1 or an even number between 2 and 96.
- Memory must be:
  - 0.9 to 6.5 GB per vCPU
  - A multiple of 256 MB
  - At least 3.75 GB (3840 MB)

In the table below, the legacy machine type name (previously used in the Cloud Console) is mapped to its equivalent string in the db-custom-*CPU*-*RAM* format. You can create the equivalent machine type by specifying the equivalent CPU and RAM in the Google Cloud Console or using gcloud, or use the db-custom-*CPU*-*RAM* format string in the API.

Note: The db-custom* strings in the following table are not a complete list of the possible VM shapes you can use.

Legacy machine type	vCPUs	Memory (MBs)	db-custom-*CPU*-*RAM* string (API tier string)
db-n1-standard-1	1	3840	db-custom-1-3840
db-n1-standard-2	2	7680	db-custom-2-7680
db-n1-standard-4	4	15360	db-custom-4-15360
db-n1-standard-8	8	30720	db-custom-8-30720
db-n1-standard-16	16	61440	db-custom-16-61440
db-n1-standard-32	32	122880	db-custom-32-122880
db-n1-standard-64	64	245760	db-custom-64-245760
db-n1-standard-96	96	368640	db-custom-96-368640
db-n1-highmem-2	2	13312	db-custom-2-13312
db-n1-highmem-4	4	26624	db-custom-4-26624
db-n1-highmem-8	8	53248	db-custom-8-53248
db-n1-highmem-16	16	106496	db-custom-16-106496
db-n1-highmem-32	32	212992	db-custom-32-212992
db-n1-highmem-64	64	425984	db-custom-64-425984

db-n1-highmem-96	96	638976	db-custom-96-638976

Troubleshooting

Click the links in the table for details:

For this problem...	The issue might be...	Try this...
`Internal error` .	Missing service networking service account.	Disable and re-enable the `Service Networking API`.
Terraform instance creation fails.	Terraform configuration error.	Inspect and repair the Terraform configuration file.
`HTTP Error 409` in Terraform script.	Another operation is already in progress.	Fix the Terraform script to wait for each operation to finish.
`Unknown error`	The Service Networking API might not be enabled. You may be trying to create an instance with the same name as one recently deleted. You may be trying to create multiple instances simultaneously. Your subnet creation may have failed if there were no more available addresses in the IP range.	Enable the Service Networking API. Use a different name for the instance, or wait until it's been a week since the instance was deleted. Create instances consecutively. See other Unknown error messages if this doesn't match your case. Allocate new ranges.
`Failed to create subnetwork` .	No more available addresses in the IP range.	Allocate new ranges.

Internal error

You see the error message `{"ResourceType":"sqladmin.v1beta4.instance", "ResourceErrorCode":"INTERNAL_ERROR","ResourceErrorMessage":null}` .

The issue might be

The service project is likely missing the service networking service account required for this feature.

Things to try

To repair service permissions, disable the `Service Networking API` , wait five minutes and then re-enable it.

Terraform instance creation fails

Terraform instance creation fails.

The issue might be

This is usually an issue within the Terraform script itself.

Things to try

Inspect and repair the Terraform configuration file.

409 error in Terraform script

You see the error message `HTTP Error 409` in Terraform scripts.

The issue might be

`Operation failed because another operation was already in progress`

Things to try

Revise the script to halt execution until each instance operation is completed. Have the script poll and wait until a `200` is returned for the previous operation ID before continuing to the next step.

Unknown error

When trying to create an instance, you see an error message like `Cloud SQL creation failed, error UNKNOWN` .

The issue might be

- The Service Networking API enabled might not be enabled.
- You may be trying to re-use the name of an instance you recently deleted. Instance names cannot be re-used for one week after deletion.
- You may be trying to create multiple instances simultaneously. In this case, only the first instance is created and the ret fail with `Unknown error` . You can only run one create operation at a time.
- Your subnet creation may have failed if there were no more available addresses in the IP range.

Things to try

- Enable the Service Networking API.

- Use a different name for the instance, or wait one week to create a new one using that name.
- Create multiple instances consecutively instead of simultaneously.
- See the Failed to create subnetwork section below.

Failed to create subnetwork

You get the error message: `Failed to create subnetwork. Couldn't find free blocks in allocated IP ranges. Please allocate new ranges for this service provider.`

The issue might be

There are no more available addresses in the allocated IP range.

If you encounter this error when attempting to create a Cloud SQL instance with private IP against a Shared VPC network using private service connections. there are five possible scenarios:

- The size of the allocated IP range for the private service connection is smaller than /24.
- The size of the allocated IP range for the private service connection is too small for the number of Cloud SQL instances.
- You're attempting to create both MySQL or SQL Server and PostgreSQL instances on the same private service connection in the VPC host project. MySQL and SQL Server can share the same service connection. PostgreSQL requires its own service connection.
- You're attempting to create instances on the same private service connection in different regions, which is not supported.

Things to try

For each of the above scenarios you can elect to either expand the existing or allocate an additional IP range to the private service connection.

If you're allocating a new range, take care to not create an allocation that overlaps with any existing allocations.

After creating a new IP range, update the vpc peering with the following command:

```
gcloud services vpc-peerings update --service=servicenetworking.googleapis.com
--ranges=[OLD_RESERVED_RANGE_NAME],[NEW_RESERVED_RANGE_NAME] --network=
[VPC_NETWORK]
--project=[PROJECT_ID] --force
```

If you're expanding an existing allocation, take care to only increase the allocation range and not decrease it. For example, if the original allocation was 10.0.10.0/24 then make the new allocation at least 10.0.10.0/23.

In general, if starting from a /24 allocation, decrementing the /mask by 1 for each condition (additional instance type group, additional region) is a good rule of thumb. For example, if trying to create both instance type groups on the same allocation, going from /24 to /23 is enough.

After expanding an existing IP range, update the vpc peering with following command:

```
gcloud services vpc-peerings update --service=servicenetworking.googleapis.com
--ranges=[RESERVED_RANGE_NAME] --network=[VPC_NETWORK] --project=[PROJECT_ID]
```

What's next