

Logical Databases

A guide for creating and managing logical databases within your Vultr Managed MySQL database instance.



Contents

01	Introduction	3
02	Vultr Customer Portal	3
03	Vultr API	3
04	Vultr CLI	5
05	Terraform	5

How to Manage Logical Databases for Vultr Managed Databases for MySQL

Introduction

Logical databases provide a structured collection of data creating a workspace for tables that in turn consist of columns and rows. A single database can store dozens or even hundreds of tables. For instance, in an e-commerce website, a database can store `products`, `categories`, `sales`, and `payments` tables.

Follow this guide to manage logical databases for Vultr Managed Databases for MySQL using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and select **Databases**.
2. Click the target database instance.
3. Navigate to **User and Databases** and click **Add New Database**.
4. Enter the **Database Name** (For example, `sample_company_db`) and click **Add Database**.
5. Click **Destroy Database** to delete the database.

Vultr API

1. List all the database instances by sending a `GET` request to the [List Managed Databases endpoint](#) and note the database ID. For example, `43b4c774-5dff-4ac0-a01f-78a23c2205b5`.

CONSOLE

```
$ curl "https://api.vultr.com/v2/databases" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

2. Send a `POST` request to the [Create Logical Database endpoint](#) specifying the `name` of the logical database and the database ID.

CONSOLE

```
$ curl "https://api.vultr.com/v2/databases/database_id/dbs" \  
  -X POST \  
  -H "Authorization: Bearer ${VULTR_API_KEY}" \  
  -H "Content-Type: application/json" \  
  --data '{  
    "name" : "sample_company_db"  
  }'
```

3. Send a `GET` request to the [List Logical Databases endpoint](#) specifying the database ID to list the logical databases.

CONSOLE

```
$ curl "https://api.vultr.com/v2/databases/database_id/dbs" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

4. Send a `DELETE` request to the [Delete Logical Database endpoint](#) specifying the ID and the logical database `name` to delete the logical database.

CONSOLE

```
$ curl "https://api.vultr.com/v2/databases/database_id/dbs/  
logical_database_name" \  
  -X DELETE \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

Visit the [Create Logical Database endpoint](#) to view additional attributes to add to your request.

Vultr CLI

1. List all database instances and note the database ID. For instance,

```
d6ac2a3c-92ea-43ef-8185-71a23e58ad8c .
```

CONSOLE

```
$ vultr-cli database list --summarize
```

2. Create a logical database by specifying the name (For example, `sample_company_db`) and the database ID.

CONSOLE

```
$ vultr-cli database db create database_id --name  
sample_company_db
```

3. List all logical databases by specifying a database ID.

CONSOLE

```
$ vultr-cli database db list database_id
```

Run `vultr-cli database db --help` to view all options.

Terraform

1. Ensure the [Vultr Terraform provider](#) is configured.
2. Create a logical database with Terraform.

TERRAFORM

```
terraform {
  required_providers {
    vultr = {
      source = "vultr/vultr"
      version = "~> 2.26"
    }
  }
}

provider "vultr" {}

# Existing database assumed
variable "database_id" { type = string }

resource "vultr_database_db" "app" {
  database_id = var.database_id
  name        = "sample_company_db"
}
```

3. To delete the logical database, remove the resource block or run:

CONSOLE

```
$ terraform destroy -target vultr_database_db.app
```



VULTR

