

Optimized Cloud Compute

High-performance cloud compute instances optimized for specific workloads with enhanced resources and specialized configurations.

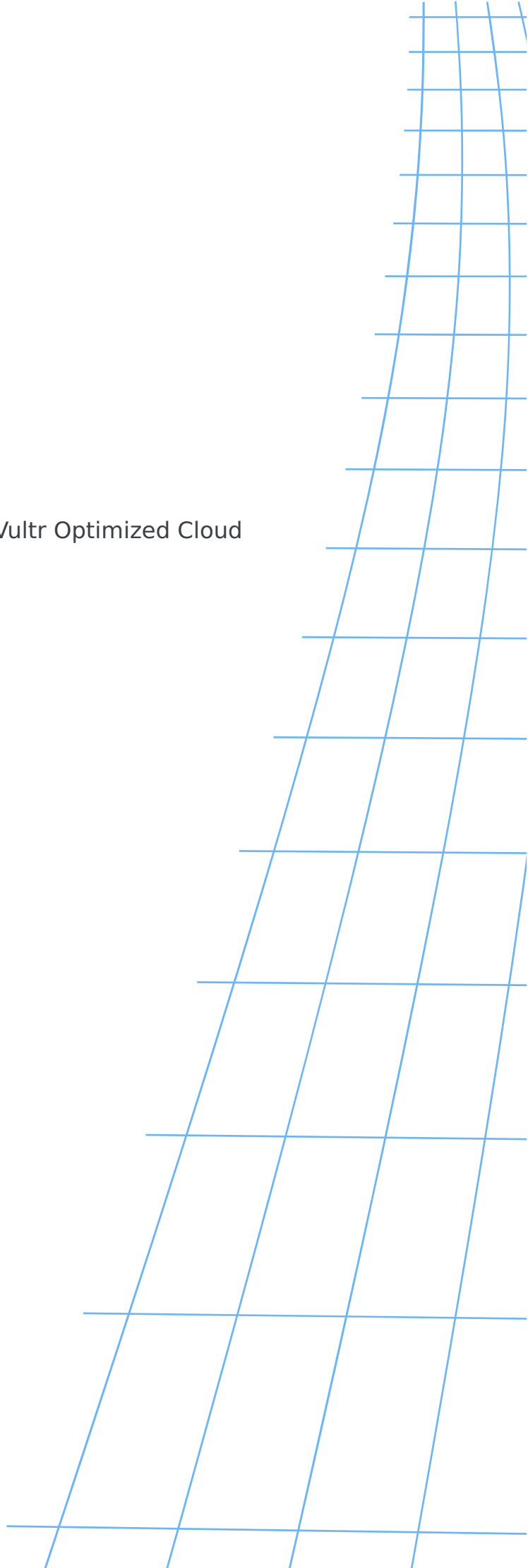
Contents

01	Provisioning	4
02	Connection	11
	OpenSSH	13
	PuTTY	18
	Vultr Console	23
03	Features	26
	Auto Backups	28
	Cloud-Init	33
	DDoS Protection	38
	Snapshots	42
04	Management	46
	Change Hostname	48
	Change OS	52
	Change Startup Script	57
	Custom ISO	61
	Delete	67
	Monitor	72
	Reinstall	76
	Reinstall SSH Keys	81
	Resize	85
	Restart	90
	Stop	94
	Tags	98
05	Networking	103
	Enable Firewall	105
	IPv4	111

IPv6	116
Reserved IPs	122
VPC 2.0	127
VPC	133
06 FAQ	139

Provisioning

A guide explaining how to deploy and set up Vultr Optimized Cloud Compute instances for your workloads.



Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Provision Vultr Optimized Cloud Compute Instances

Introduction

Vultr Optimized Cloud Compute instances are dedicated virtual machines designed for demanding business applications such as production websites, CI/CD, video transcoding, and large databases. Vultr Optimized Cloud Compute instances are capable of running resource-intensive applications that require specific CPU, memory or storage resources.

Follow this guide to provision Vultr Optimized Cloud Compute instances using the Vultr customer portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click **Deploy**.
3. Choose **Dedicated CPU** as the instance type.
4. Select your desired Vultr location to deploy the instance to.
5. Select a plan category from the sidebar:
 - **VX1**: Provide the best price-to-performance for efficient compute workloads.
 - **General Purpose**: Support resource-demanding business applications such as production websites, CI/CD, video transcoding, and large databases.
 - **CPU Optimized**: Support compute-bound applications that require proportionally more CPU than RAM (memory) and storage.

- **Memory Optimized:** Support memory-bound applications that require more RAM than CPU and storage.
 - **Storage Optimized:** Provide more NVMe SSD storage to balance the available CPU and RAM.
6. Select a plan from the available options based on your vCPU, memory, storage, and bandwidth requirements.
 7. Click **Configure Software**.
 8. Select a cloud image to install on the instance based on the following options:
 - **Operating System:** Installs a fresh operating system image on the instance.
 - **Marketplace Apps:** Installs a prebuilt software stack or application and the recommended operating system image on the instance.
 - **ISO/iPXE:** Boots a specific ISO available or iPXE-compatible image on the instance.
 - **ISO Library:** Installs a specific ISO image from the Vultr ISOs library.
 - **Backup:** Recovers a specific backup available in your Vultr account to the instance.
 - **Snapshot:** Installs a specific snapshot available in your Vultr account to the instance.
 9. Select optional **Server Settings** to apply on the instance.
 - **SSH Keys:** Installs a specific SSH key on the instance.
 - **Startup Script:** Enables a startup script to execute at deployment or a PXE script to automate the operating system installation.
 - **Firewall Group:** Activates a Vultr Firewall group to filter incoming network traffic on the instance.
 10. Enter a new hostname in the **Server Hostname** field and a descriptive label in the **Server Label** field to identify the instance.

11. Configure **Additional Features** for the instance.

- **Instance Connectivity:** Select how the instance connects to the internet.
 - **Instance(s) with Public IP:** Assigns public IP addresses directly to the instance. Under **Instance Address, Public IPv4** is enabled by default. Select **Public IPv6** to enable IPv6 addressing. After selecting IPv6, you can optionally deselect **Public IPv4** to create an IPv6-only instance.
 - **Private Instance(s) behind NAT Gateway:** Routes internet traffic through a NAT Gateway in a Virtual Private Cloud (VPC) Network. Select an existing VPC Network with a NAT Gateway or click **Add VPC Network** to create a new one.
- **VPC Network:** Connects the instance to a VPC Network in the current location.
- **Automatic Backups:** Automatically creates backups for data recovery in case of instance failures.
- **DDoS Protection:** Prevents potential Distributed Denial of Service (DDoS) attacks on the instance.
- **Limited User Login:** Creates a `linuxuser` non-root user with sudo privileges as the default user account instead of `root`.
- **Cloud-Init User Data:** Enables Cloud-Init user data to initialize and customize the instance at boot.

12. Click **Deploy** to provision the instance.

Vultr API

1. Send a `POST` request to the [Create Instance endpoint](#) to create a new Vultr Optimized Cloud Compute instance. Replace `VULTR_LOCATION`, `INSTANCE_PLAN`, `OS_ID`, `INSTANCE_LABEL`, and `HOSTNAME` with your target values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-d '{"location": "VULTR_LOCATION", "plan": "INSTANCE_PLAN", "os_id": "OS_ID", "label": "INSTANCE_LABEL", "hostname": "HOSTNAME"}'
```

```
-H "Content-Type: application/json" \  
--data '{  
  "region" : "VULTR_LOCATION",  
  "plan" : "INSTANCE_PLAN",  
  "os_id" : OS_ID,  
  "label" : "INSTANCE_LABEL",  
  "hostname": "HOSTNAME"  
}'
```

Visit the [Create Instance API page](#) to view additional attributes you can apply on the Vultr Optimized Cloud Compute instance.

2. Send a `GET` request to the [List Instances endpoint](#) to list all available instances.

CONSOLE

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

Vultr CLI

1. Create a new Vultr Optimized Cloud Compute instance. Replace `VULTR_LOCATION`, `INSTANCE_PLAN`, `OS_ID`, `INSTANCE_LABEL`, and `HOSTNAME` with your target values.

CONSOLE

```
$ vultr-cli instance create --region VULTR_LOCATION --plan  
INSTANCE_PLAN --os OS_ID --label INSTANCE_LABEL --host  
HOSTNAME
```

Run `vultr-cli instance create --help` to view additional options you can apply on the Vultr Optimized Cloud Compute instance.

2. List all available instances.

CONSOLE

```
$ vultr-cli instance list
```

Terraform

1. Ensure the [Vultr Terraform provider](#) is configured in your Terraform project.
2. Define the Optimized Cloud Compute instance in your Terraform configuration file.

TERRAFORM

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

provider "vultr" {}

resource "vultr_instance" "occ" {
  label          = "occ-instance-1"
  hostname       = "occ-instance-1"
  region         = "del"              # change to your target
  region (such as ewr, ams, sgp)
  plan           = "vhp-2c-4gb"      # OCC plan code
  os_id          = 2284               # Ubuntu 24.04 LTS x64
  enable_ipv6    = true
}

output "public_ip" {
  value = vultr_instance.occ.main_ip
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

Connection

Establish connectivity to your Vultr resources through various network protocols and access methods.

Contents

01	OpenSSH	13
02	PuTTY	18
03	Vultr Console	23

OpenSSH

A secure protocol for remote server access that comes pre-installed on Vultr instances, allowing encrypted connections from your local machine.

Contents

01	Introduction	6
02	Connect to an Instance Using the Default User Credentials	15
03	Connect to an Instance Using SSH Keys	16

How To Connect to a Vultr Optimized Cloud Compute Instance Using SSH

Introduction

OpenSSH is a connection protocol that enables SSH access to an instance. It is pre-installed and active on Vultr Optimized Cloud Compute instances by default to enable secure connections.

Follow this guide to connect to a Vultr Optimized Cloud Compute instance using SSH on your workstation.

Connect to an Instance Using the Default User Credentials

1. Open your instance's management page.
2. Note the default credentials within the **Overview** tab and copy the user password to your clipboard.
3. Open a new terminal or command prompt application on your workstation.
4. Connect to your Vultr Optimized Cloud Compute instance using SSH.

CONSOLE

```
$ ssh username@SERVER-IP
```

5. Enter `yes` and press `ENTER` when prompted to add the instance's public key to your known hosts.

```
The authenticity of host '192.0.2.123 (192.0.2.123)' can't be established.  
ED25519 key fingerprint is SHA256:gTA0uCiCa3Us4tpVaVHVk9d3q0jKrsqXP0sAFQbB8xw.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

6. Enter your instance user's password when prompted and press **ENTER** to log in.

```
username@SERVER-IP's password:
```

7. View the active user in your SSH session.

```
CONSOLE
```

```
$ whoami
```

Connect to an Instance Using SSH Keys

Note

Generate an SSH key on your workstation and add it to your instance during deployment. Adding an SSH key using the Vultr Customer Portal after deployment will result in data loss and wipe your instance to install the new key.

1. Open a new terminal or command prompt application on your workstation.
2. Connect to your Vultr Optimized Cloud Compute instance using a specific SSH key on your workstation.

```
CONSOLE
```

```
$ ssh -i /path/to/private/key username@SERVER-IP
```

3. Enter **yes** and press **ENTER** when prompted to add the instance's public key to your known hosts.

```
The authenticity of host '192.0.2.123 (192.0.2.123)' can't be established.  
ED25519 key fingerprint is SHA256:gTA0uCiCa3Us4tpVaVHVk9d3q0jKrsqXP0sAFQbB8xw.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
```

4. View the active user in your SSH session.

CONSOLE

```
$ whoami
```

PuTTY

A free SSH client for Windows that allows secure remote connections to Vultr instances using terminal access.

Contents

01	Introduction	6
02	Connect to an Instance Using the Default User Credentials	15
03	Connect to an Instance Using SSH Keys	16

How to Connect to a Vultr Optimized Cloud Compute Instance Using PuTTY

Introduction

PuTTY is an open-source terminal emulator and SSH client for Windows workstations. It provides a user-friendly interface and terminal to connect to instances using SSH. PuTTY supports both password-based authentication and SSH keys for secure connections to an instance.

Follow this guide to connect to a Vultr Optimized Cloud Compute instance using PuTTY on Windows.

Connect to an Instance Using the Default User Credentials

1. Open PuTTY from your applications menu.
2. Enter your instance's public IP address in the **Host Name (or IP address)** field.
3. Keep `22` as the **Port** value and **SSH** as the connection type.
4. Click **Open** to connect to your instance using SSH.
5. Click **Accept** when prompted to add the instance's public key to your workstation's known hosts.
6. Enter your username when prompted for the `login as` value.
7. Enter the user's password when prompted and press Enter to log in.

8. View the active user information in your SSH session.

```
CONSOLE
$ whoami
```

Connect to an Instance Using SSH Keys

1. Enter your instance's public IP address in the **Host Name (or IP address)** field.
2. Keep `22` as the **Port** and **SSH** as the connection type.
3. Expand the **SSH** group on the main navigation menu to access additional connection options.
4. Expand the **Auth** group and select **Credentials** from the list of options.
5. Click **Browse** within the **Private key file for authentication** field to load your private key.
6. Click **Data** within the **Connection** group and enter the default instance username in the **Auto-login username** field to use with your SSH key.
7. Navigate to **Session** on the main navigation menu and enter a new session name in the **Saved Sessions** field.
8. Click **Save** to store your SSH key, user, and the instance IP configurations.
9. Click **Open** to connect to the instance using the SSH key session information.
10. Click **Accept** when prompted to add the instance's public key to your workstation's known hosts.
11. View the active user information in your SSH session.

```
CONSOLE
```

```
$ whoami
```

Vultr Console

Access your Vultr instance directly through the web browser without SSH clients using the built-in console interface.

Contents

01 Introduction	6
-----------------	---

How to Connect to a Vultr Optimized Cloud Compute Instance Using the Vultr Console

Introduction

Vultr Console is a noVNC terminal that provides direct access to an instance's console. You can run commands, install applications and manage processes through the Vultr Console. Additionally, it offers multiple features like clipboard sharing, a virtual keyboard, and special commands such as `CTRL` `ALT` `DEL` to manage an instance.

Follow this guide to connect to a Vultr Optimized Cloud Compute instance using the Vultr Console.

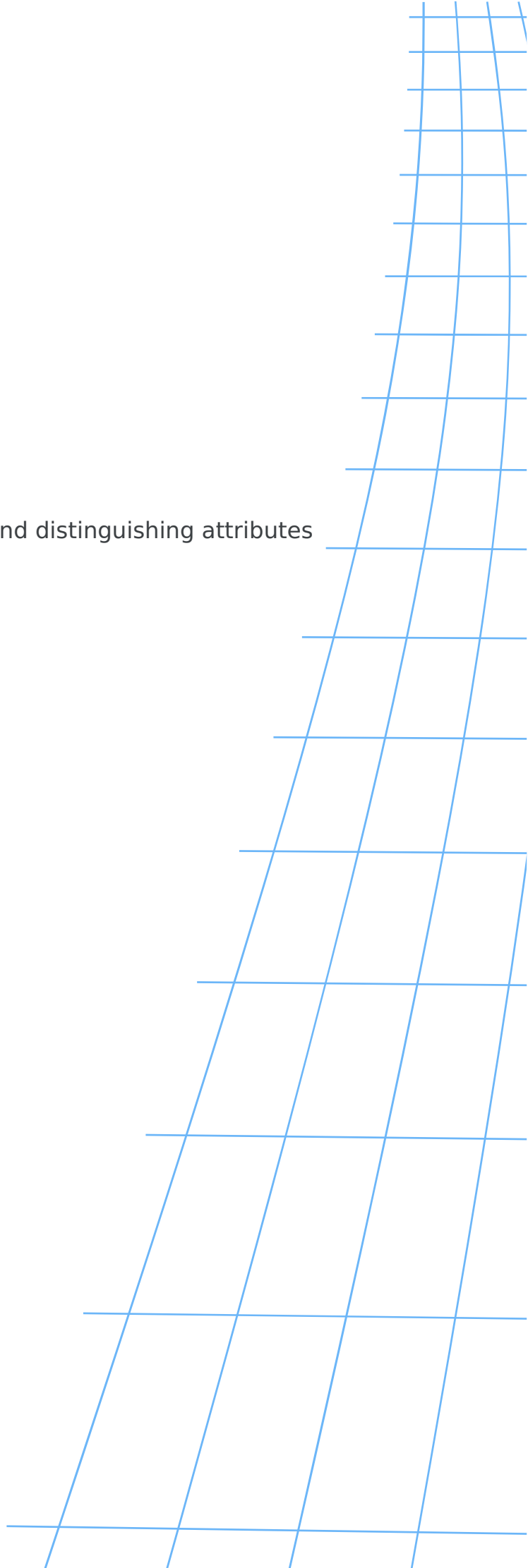
Note

Enable pop-ups in your web browser settings to access the Vultr Console.

1. Open your instance's management page.
2. Find the default credentials in the **Overview** tab and copy the user password to your clipboard.
3. Find and click **View Console** on the top-right navigation menu to open the Vultr Console.
4. Enter your default username and press `ENTER` when prompted.
5. Find and click **Send Clipboard** on the list of control bar options to paste the user password in your Vultr Console session.
6. Press `ENTER` to log in to the instance.

Features

Provides an overview of the key capabilities and distinguishing attributes of Vultr's cloud infrastructure services.



Contents

01	Auto Backups	28
02	Cloud-Init	33
03	DDoS Protection	38
04	Snapshots	42

Auto Backups

A service that automatically creates and stores regular backups of your Vultr instance for data protection and recovery.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Terraform	10

How to Enable Automatic Backups on a Vultr Optimized Cloud Compute Instance

Introduction

Automatic Backups allow you to create full backups of your instance's data and file system on a scheduled basis, ensuring recovery in case of unexpected failures. These backups follow specific schedules and retention policies to ensure your instance is securely backed up in your Vultr account.

Follow this guide to enable automatic backups on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Backups** tab.
4. Click **Enable Backups** to turn on automatic backups.
5. Click **Enable Backups** in the confirmation prompt to enable automatic backups.
6. Click the **Schedule Backups** drop-down to choose a backup schedule.
7. Click **Update** to create automatic backups based on your selection.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `PATCH` request to the [Update Instance endpoint](#) to update the instance and enable automatic backups.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}" \  
-X PATCH \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "backups" : "enabled"  
}'
```

3. Send a `POST` request to the [Set Instance Backup Schedule endpoint](#) to create a new automatic backups schedule.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/  
backup-schedule" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "type": "daily",  
  "hour": 10,  
  "dow": 1,  
  "dom": 1  
}'
```

Visit the [Set Instance Backup Schedule API page](#) to view additional backup schedule attributes.

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Enable backups and define a schedule in the configuration.

TERRAFORM

```
resource "vultr_instance" "occ" {
  label      = "occ-instance-1"
  hostname   = "occ-instance-1"
  region     = "del"
  plan       = "vhp-2c-4gb"
  os_id      = 2284

  backups = "enabled"

  backups_schedule {
    type = "daily" # daily | weekly | monthly
    hour = 10      # UTC hour (0-23)
    dow  = 1       # used for weekly
    dom  = 1       # used for monthly
  }
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Cloud-Init

A guide explaining how to modify the Cloud-Init user data on your Vultr Optimized Cloud Compute instance after deployment.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Update Cloud-Init User Data on a Vultr Optimized Cloud Compute Instance

Introduction

Cloud-Init enables the automatic initialization and configuration of instances during the initial boot phase. It runs user data scripts to customize an instance, install applications, and configure specific packages or services.

Follow this guide to update Cloud-Init user data on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **User-Data** tab.
4. Enter your script or cloud config in the **Cloud-Init User-Data** field.
5. Click **Update** to apply the changes.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `PATCH` request to the [Update Instance endpoint](#) to update the instance's Cloud-Init user data.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}" \  
-X PATCH \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "user_data" : "<cloud-init-data>,"  
}'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. Upload new Cloud-Init user data to the instance from a file on your workstation.

CONSOLE

```
$ vultr-cli instance user-data set <instance-id> --userdata  
"<script-path>"
```

Terraform

Cloud-Init user data can only be set during instance creation in Terraform and cannot be updated on an existing instance without recreating it.

1. Open your Terraform configuration for the new Optimized Cloud Compute instance.
2. Add the `user_data` argument to the instance resource to run a script at first boot.

TERRAFORM

```
resource "vultr_instance" "occ" {
  # ...existing fields (region, plan, os_id, label, etc.)

  user_data = <<-EOT
  #!/bin/bash
  apt-get update -y
  apt-get install -y nginx
  systemctl enable --now nginx
  EOT
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

DDoS Protection

Learn how to enable DDoS protection on your Vultr Optimized Cloud Compute instance to safeguard against distributed denial-of-service attacks.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Terraform	10

How to Enable DDoS Protection on a Vultr Optimized Cloud Compute Instance

Introduction

Distributed Denial of Service (DDoS) protection enables traffic monitoring and prevents potential DDoS attacks to an instance. It activates a set of tools that detect and block network flooding attempts, ensuring the instance remains active and operational.

Follow this guide to enable DDoS protection on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **DDoS** tab.
4. Click **Enable DDoS Protection**.
5. Click **Enable DDoS Protection** in the confirmation prompt to enable DDoS protection on your instance.

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Enable DDoS protection in the instance resource.

TERRAFORM

```
resource "vultr_instance" "occ" {  
  # ...existing fields (region, plan, os_id, label, etc.)  
  
  ddos_protection = true  
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Snapshots

A feature that allows you to create point-in-time copies of your Vultr instances for backup, cloning, or recovery purposes.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9

How to Create Snapshots on a Vultr Optimized Cloud Compute Instance

Introduction

A snapshot is a point-in-time copy of an instance's state, including its entire file system and disk contents. Snapshots offer a quick backup solution for instances, making it easier to restore data in case of unexpected failures or data loss.

Follow this guide to create snapshots on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click the target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Snapshots** tab.
4. Enter a new descriptive label in the **Label** field and click **Create Snapshot** to take a new snapshot of your instance. Snapshot creation can take 20 to 30 minutes depending on the instance size.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `POST` request to the [Create Snapshot endpoint](#) to create a new snapshot of the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/snapshots" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "instance_id" : "<instance-id>",  
  "description" : "<label>"  
}'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. Create a new snapshot of the instance.

CONSOLE

```
$ vultr-cli snapshot create --id <instance-id>
```

Management

Tools and features for managing your Vultr infrastructure, including billing, access controls, and account settings.

Contents

01	Change Hostname	48
02	Change OS	52
03	Change Startup Script	57
04	Custom ISO	61
05	Delete	67
06	Monitor	72
07	Reinstall	76
08	Reinstall SSH Keys	81
09	Resize	85
010	Restart	90
011	Stop	94
012	Tags	98

Change Hostname

Learn how to modify the hostname on your Vultr Optimized Cloud Compute instance.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Terraform	10

How to Change the Hostname on a Vultr Optimized Cloud Compute Instance

Introduction

Changing the hostname on an instance modifies the default server configuration and reinstalls the operating system. This operation may result into data loss when the instance is reinstalled to apply the new hostname.

Follow this guide to change the hostname on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, or Terraform.

Warning

Changing the hostname reinstalls the operating system and wipes all the data on your server.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Find and click **Change Hostname** on the left navigation menu.
5. Replace the existing value with your new hostname.
6. Click **Reinstall** to change your instance hostname.
7. Check the confirmation prompt and click **Change Hostname** to apply the new hostname.

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Update the `hostname` value in the instance resource.

TERRAFORM

```
resource "vultr_instance" "occ" {  
  # ...existing fields (region, plan, os_id, label, etc.)  
  
  hostname = "new-hostname"  
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
```

Change OS

A guide explaining how to replace the operating system on your Vultr Optimized Cloud Compute instance.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Change the Operating System on a Vultr Optimized Cloud Compute Instance

Introduction

Changing the instance operating system wipes all data on your server and installs a new file system. This is important when changing the default operating system while preserving the instance's IP and networking information.

Follow this guide to change the operating system on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Warning

Changing to a different operating system wipes all the data on your server.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Find and click **Change OS** on the left navigation menu.
5. Click the **Choose new OS** drop-down and select a new operating system to install on your instance.
6. Click **Change OS** to change the instance operating system.
7. Check the **Change OS** confirmation prompt and click **Change OS** to apply the new instance changes.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `GET` request to the [List OS endpoint](#) to view all available operating systems and note the target OS ID.

CONSOLE

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

3. Send a `PATCH` request to the [Update Instance endpoint](#) with a new `os_id` value to change the instance's operating system.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}" \  
-X PATCH \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "os_id" : "new-instance-os_id"  
}'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. List all available operating systems and note the target OS ID.

CONSOLE

```
$ vultr-cli instance os list <instance-id>
```

3. Change the target instance's operating system.

CONSOLE

```
$ vultr-cli instance os change <instance-id> --os <os_id>
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Update the `os_id` value in the instance resource to the new operating system ID.

TERRAFORM

```
resource "vultr_instance" "occ" {  
  # ...existing fields (region, plan, label, etc.)  
  
  os_id = 1743 # Example: Ubuntu 22.04 LTS x64  
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Change Startup Script

Learn how to modify the startup script that runs when your Vultr instance boots up

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Terraform	10

How to Change the Startup Script on a Vultr Optimized Cloud Compute Instance

Introduction

Startup scripts allow you to automate specific configurations during the operating system installation. Changing the startup script on an instance wipes the file system and reinstalls the default operating system.

Follow this guide to change the startup script on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, or Terraform.

Warning

Changing the startup script on an instance will wipe all data and reinstall the operating system.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Find and click **Change StartUp Script** on the left navigation menu.
5. Click **Add Startup Script**.
6. Enter a new script name and click the **Type** drop down to select the script type.
7. Add your startup script data and click **Add Script** to apply the script to your instance.
8. Navigate to **Change StartUp Script** in your instance settings to verify that the new script is added.

9. Select the startup script and click **Change Startup Script**.
10. Check the confirmation prompt and click **Change Startup Script** to apply the new script.

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Create or reference a `vultr_startup_script` resource and attach it to the instance.

```
TERRAFORM

# Create a new startup script
resource "vultr_startup_script" "init" {
  name = "init-nginx"
  type = "boot" # boot | pxeboot
  script = <<-EOT
  #!/bin/bash
  apt-get update -y
  apt-get install -y nginx
  systemctl enable --now nginx
  EOT
}

# Attach the startup script to the instance
resource "vultr_instance" "occ" {
  # ...existing fields (region, plan, os_id, label, etc.)

  script_id = vultr_startup_script.init.id
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Custom ISO

A feature that allows you to upload and attach your own ISO image to Vultr instances for custom operating system installations.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Attach a Custom ISO to a Vultr Optimized Cloud Compute Instance

Introduction

ISO images allow you to install a specific operating system on an Vultr Optimized Cloud Compute instance. Custom ISOs enable you to deploy operating systems not available in the default Vultr installer list. They are useful for creating tailored environments or booting into specialized modes, such as rescue and recovery environments.

Follow this guide to attach a custom ISO to a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Warning

Installing a custom ISO on a Vultr Optimized Cloud Compute instance disables the default user credentials listed in your instance's management page.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Find and click **Custom ISO** on the left navigation menu.
5. Select a custom ISO available in your Vultr account or click the **ISO Library** drop-down to select from a list of public ISOs.
6. Click **Attach ISO and Reboot** to attach the ISO to your instance.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `GET` request to the [List ISOs endpoint](#) and note your target ISO ID.

CONSOLE

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

3. Send a `POST` request to the [Attach ISO to Instance endpoint](#) to attach the ISO to your target instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/iso/  
attach" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "iso_id" : "<iso-id>"  
}'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. List all available ISO images and note your target ISO's ID.

CONSOLE

```
$ vultr-cli iso list
```

3. Attach the ISO to your target instance.

```
$ vultr-cli instance iso attach <instance-id> \  
  --iso-id="<iso-id>"
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Specify the ISO by its `os_id` when defining or rebuilding the instance.

TERRAFORM

```
# Example: attach a custom ISO by its ID (replace 159 with  
your ISO ID)  
resource "vultr_instance" "occ" {  
  label    = "occ-custom-iso"  
  region   = "del"  
  plan     = "vhp-2c-4gb"  
  os_id    = 159 # 'Custom' ISO type  
  iso_id   = "your-iso-id-here" # ID of uploaded or  
library ISO  
  hostname = "occ-custom-iso"  
}
```

3. Apply the configuration and observe the following output:

Apply complete! Resources: 0 added, 1 changed, 0 destroyed.

Note

Terraform does not currently support **attaching an ISO in-place** to a running instance.

The ISO must be set at **create time** or during a **rebuild**, which replaces the instance and wipes all existing data.

Delete

Learn how to permanently remove a Vultr Optimized Cloud Compute instance from your account.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Delete a Vultr Optimized Cloud Compute Instance

Introduction

Deleting an instance permanently erases the server's file system and removes all IP information. This action is irreversible and any data on the instance data is lost unless a backup or snapshot is available in your Vultr account.

Follow this guide to delete a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, or CLI.

Warning

Deleting an instance is permanent and irreversible. Take a snapshot of the instance if you want to recover the instance at a later time.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Click **Destroy Server** on the top-right navigation menu.
4. Check the confirmation prompt and click **Destroy Server** to apply changes.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `DELETE` request to the [Delete Instance endpoint](#) to delete the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}" \  
-X DELETE \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. Delete the target instance.

CONSOLE

```
$ vultr-cli instance delete <instance_id>
```

Terraform

1. Open your Terraform configuration and locate the Optimized Cloud Compute instance resource.
2. Remove the resource block or set its lifecycle to destroy.

TERRAFORM

```
resource "vultr_instance" "occ" {
  label      = "occ-instance-1"
  region    = "del"
  plan      = "vhp-2c-4gb"
  os_id     = 2284
  hostname  = "occ-instance-1"
}

# To delete, either remove this block from configuration
# or run: terraform destroy -target vultr_instance.occ
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 0 changed, 1 destroyed.
```

Monitor

Learn how to monitor your Vultr Optimized Cloud Compute instances performance metrics including CPU usage, disk operations, and bandwidth statistics.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9

How to Monitor a Vultr Optimized Cloud Compute Instance

Introduction

Monitoring an instance provides information about its performance and usage statistics. This enables you to track the instance's activity, health, and resource usage. You can monitor the vCPU usage, disk operations and bandwidth statistics on a Vultr Cloud Compute instance.

Follow this guide to monitor a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. View the instance usage summary within the **Overview** section.
4. Navigate to the **Usage Graphs** tab to monitor the instance's usage statistics.
5. Monitor the instance's **bandwidth usage statistics** within the **Monthly Bandwidth** section.
6. Monitor the instance's **performance statistics** within the **Server Monitors** section.
7. Click the **Range** drop-down to select a specific timeframe and view the monitoring information in the following categories:
 - **vCPU Usage**: Displays the vCPU usage statistics.
 - **Disk Operations**: Displays the read and write operations per second on the primary storage disk.
 - **Network**: Displays the instance's networking statistics in bytes.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `POST` request to the [Instance Bandwidth endpoint](#) to monitor the instance's bandwidth usage statistics.

CONSOLE

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

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. Monitor the instance's bandwidth usage statistics.

CONSOLE

```
$ vultr-cli instance bandwidth <instance-id>
```

Reinstall

Learn how to reinstall your Vultr Optimized Cloud Compute instance to reset it to a fresh state.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Reinstall a Vultr Optimized Cloud Compute Instance

Introduction

Reinstalling an instance wipes the file system, resets all configurations, and reinstalls the operating system. Any data on the instance's file system is permanently deleted and cannot be recovered unless a backup or snapshot is available in your Vultr account.

Follow this guide to reinstall a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Warning

Reinstalling an instance will wipe all data and reinstall the operating system.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Click **Server Reinstall** on the top-right navigation menu.
4. Check the confirmation prompt and click **Reinstall Server** to apply the changes.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `POST` request to the [Reinstall Instance endpoint](#) to reinstall the target instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/reinstall" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  

```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. Reinstall the target instance.

CONSOLE

```
$ vultr-cli instance reinstall <instance_id>
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Trigger a reinstall by replacing the instance during apply.

CONSOLE

```
$ terraform apply -replace=vultr_instance.occ
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.
```

Reinstall SSH Keys

A guide for reinstalling SSH keys on your Vultr Optimized Cloud Compute instance to restore secure access.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Terraform	10

How to Reinstall SSH Keys on a Vultr Optimized Cloud Compute Instance

Introduction

SSH keys enable secure, password-free authentication for users accessing your instance over SSH. Reinstalling SSH keys resets the instance, wiping all data and reinstalls the operating system to apply the new SSH key details.

Follow this guide to reinstall SSH keys on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, or Terraform.

Warning

Reinstalling SSH keys will wipe all data on the instance and reapply the selected SSH keys.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Find and click **Reinstall SSH Keys** on the left navigation menu.
5. Select the target SSH key and click **Reinstall**.
6. Check the confirmation prompt and click **Reinstall SSH Keys** to apply the changes, reinstall the instance and enable the SSH key.

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Update the `ssh_key_ids` in the instance resource to reference the new SSH key(s).

TERRAFORM

```
resource "vultr_ssh_key" "new_key" {
  name      = "mbp-ed25519"
  public_key = file("~/ssh/id_ed25519.pub")
}

resource "vultr_instance" "occ" {
  # ...existing fields (region, plan, os_id, label, etc.)

  ssh_key_ids = [vultr_ssh_key.new_key.id]
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Resize

Learn how to increase or decrease the resources of your Vultr Optimized Cloud Compute instance.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Resize a Vultr Optimized Cloud Compute Instance

Introduction

Resizing an instance activates a new plan with more vCPUs, RAM, and storage to match your needs. Downgrading is not supported while upgrading the instance enables a higher plan without changes to the instance's data or file system.

Follow this guide to resize a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Find and click **Change Plan** on the left navigation menu.
5. Click the **Change Plan** drop-down and select a new instance plan.
6. Click **Upgrade** to resize your instance.
7. Check the confirmation prompt and click **Change Plan** to apply changes.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `PATCH` request to the [Update Instance endpoint](#) to resize the instance with a new plan and note the Job ID.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}" \  
-X PATCH \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "plan" : "instance_plan_id"  
}'
```

3. Send a `GET` request to the [Get Instance Job endpoint](#) to monitor and get available information for the upgrade plan instance job.

CONSOLE

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

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. List all available plans the instance can resize to.

CONSOLE

```
$ vultr-cli instance plan list <instance-id>
```

3. Resize the instance to a new plan.

CONSOLE

```
$ vultr-cli instance plan upgrade <instance-id> --plan  
<instance_plan_id>
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Update the `plan` value in the instance resource to the new Optimized Cloud Compute plan code.

TERRAFORM

```
resource "vultr_instance" "occ" {  
  # ...existing fields (region, os_id, label, etc.)  
  
  plan = "vhp-4c-8gb" # Example: upgrade from vhp-2c-4gb  
to vhp-4c-8gb  
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Restart

Learn how to restart your Vultr Optimized Cloud Compute instance through the Customer Portal or API.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9

How to Restart a Vultr Optimized Cloud Compute Instance

Introduction

Restarting an instance performs a hard reboot, stopping all running processes before starting them again. It does not affect the instance's data or file system and allows application updates or configuration changes that require a reboot to take effect.

Follow this guide to restart a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Click **Server Restart** on the top-right navigation menu to restart your server.
4. Click **Restart Server** in the confirmation prompt to apply changes.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

```
CONSOLE
```

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

2. Send a `POST` request to the [Reboot Instances endpoint](#) to restart the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/reboot" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "instance_ids" : [  
    "instance_id"  
  ]  
'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

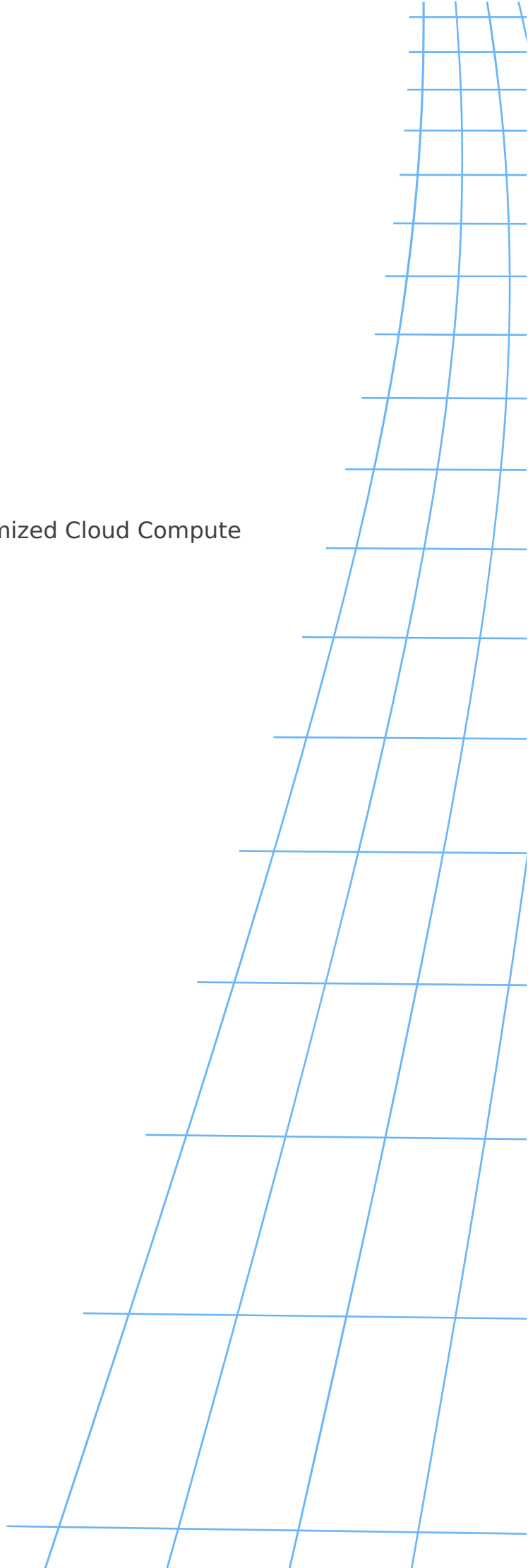
2. Restart the instance.

CONSOLE

```
$ vultr-cli instance restart <instance_id>
```

Stop

Learn how to safely stop a running Vultr Optimized Cloud Compute instance from the control panel.



Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9

How to Stop a Vultr Optimized Cloud Compute Instance

Introduction

Stopping an instance shuts it down and disables network connectivity until it is restarted. The operating system and all running processes are halted, but billing continues unless the instance is deleted.

Follow this guide to stop a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Click **Stop Server** on the top-right navigation menu to stop the instance.
4. Click **Stop Server** in the confirmation prompt to stop the instance.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `POST` request to the [Halt Instances endpoint](#) to stop the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/halt" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "instance_ids" : [  
    "your-instance-id"  
  ]  
'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. Stop the instance.

CONSOLE

```
$ vultr-cli instance stop <your-instance-id>
```

Tags

Learn how to organize and categorize your Vultr resources by adding tags to your Optimized Cloud Compute instances.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Add Tags on a Vultr Optimized Cloud Compute Instance

Introduction

Tagging allows you to assign specific labels, known as tags, to an instance for improved identification in your Vultr account. Tags consist of multiple characters that help identify, organize, and manage instances in your Vultr account.

Follow this guide to add tags on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Tags** tab.
4. Enter a new tag in the **Add Tag** field and click **Add** to apply the new tag.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `PATCH` request to the [Update Instance endpoint](#) to add tags to the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}" \
-X PATCH \
-H "Authorization: Bearer ${VULTR_API_KEY}" \
-H "Content-Type: application/json" \
--data '{
  "tags" : ["tag1", "tag2"]
}'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. Add new tags to the instance.

CONSOLE

```
$ vultr-cli instance tags <instance_id> --tags <tag1,tag2>
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Add or update the `tags` argument in the instance resource.

TERRAFORM

```
resource "vultr_instance" "occ" {  
  # ...existing fields (region, plan, os_id, label, etc.)  
  
  tags = ["production", "web-server"]  
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Networking

Manage your Vultr infrastructures connectivity with advanced networking features, configurations, and security controls.

Contents

01	Enable Firewall	105
02	IPv4	111
03	IPv6	116
04	Reserved IPs	122
05	VPC 2.0	127
06	VPC	133

Enable Firewall

Learn how to activate a Vultr Firewall Group to protect your Optimized Cloud Compute instance.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Enable a Vultr Firewall Group on a Vultr Optimized Cloud Compute Instance

Introduction

Vultr Firewall groups allow you to create rules that filter incoming network traffic to an instance. Firewall rules define the network ports and services to control, filter, and secure network connections to the instance.

Follow this guide to enable a Vultr Firewall group on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Click **Firewall** on the left navigation menu.
5. Click the **Firewall** drop-down to select a new firewall group.
6. Click **Update Firewall Group** to enable the firewall group on the instance.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `GET` request to the [List Firewall Groups endpoint](#) to list all available firewall groups.

CONSOLE

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

3. Send a `PATCH` request to the [Update Instance endpoint](#) to attach a firewall group to the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}" \  
-X PATCH \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "firewall_group_id" : "<firewall-id>",  
'
```

Vultr CLI

1. List all available firewall groups and note the target firewall group's ID.

CONSOLE

```
$ vultr-cli firewall group list
```

2. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

3. Attach the firewall group to the instance.

CONSOLE

```
$ vultr-cli instance update-firewall-group --instance-id  
<instance-id> --firewall-group-id <firewall-id>
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Create or reference a `vultr_firewall_group` resource and attach it to the instance.

TERRAFORM

```
# Create a firewall group  
resource "vultr_firewall_group" "web" {  
    description = "Web server firewall group"  
}  
  
# Add example rules to the group  
resource "vultr_firewall_rule" "allow_http" {  
    firewall_group_id = vultr_firewall_group.web.id  
    protocol           = "tcp"  
    ip_type           = "v4"  
    subnet            = "0.0.0.0"  
    subnet_size       = 0  
    port              = "80"  
}  
  
resource "vultr_firewall_rule" "allow_https" {  
    firewall_group_id = vultr_firewall_group.web.id  
    protocol           = "tcp"  
    ip_type           = "v4"  
    subnet            = "0.0.0.0"  
    subnet_size       = 0  
    port              = "443"  
}
```

```
# Attach the firewall group to the instance
resource "vultr_instance" "occ" {
  # ...existing fields (region, plan, os_id, label, etc.)

  firewall_group_id = vultr_firewall_group.web.id
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

IPv4

Learn how to add additional IPv4 addresses to your Vultr Optimized Cloud Compute instance.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9

How to Add IPv4 Addresses on a Vultr Optimized Cloud Compute Instance

Introduction

A public IPv4 address is automatically assigned to an instance upon deployment, unless disabled by default. You can attach multiple IPv4 addresses to the instance to enable external network connections. Additional addresses can also be used for tasks such as IP forwarding, static and dynamic routing.

Follow this guide to add the IPv4 information on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Click **IPv4** on the left navigation menu to view the instance's public IPv4 network information.
5. Click **Add Another IPv4 Address** to attach an additional public IP address to the instance.
6. Check the confirmation prompt and click **Add IPv4 Address** to attach the new public IP address and restart your instance.
7. Click the default IPv4 reverse DNS value and replace it a custom value to enable reverse DNS on the instance.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `GET` request to the [List Instance IPv4 Information endpoint](#) to view the instance's IPv4 information.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/  
ipv4" \  
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

3. Send a `POST` request to the [Create IPv4 endpoint](#) to attach a new IPv4 address to the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/  
ipv4" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "reboot" : true  
'
```

4. Send a `POST` request to the [Create Instance Reverse IPv4 endpoint](#) to enable reverse DNS on the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/
ipv4/reverse" \
-X POST \
-H "Authorization: Bearer ${VULTR_API_KEY}" \
-H "Content-Type: application/json" \
--data '{
  "ip" : "<ipv4-address>",
  "reverse" : "<domain>"
}'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. List the instance's IPv4 address information.

CONSOLE

```
$ vultr-cli instance ipv4 list <instance-id>
```

3. Create a new public IPv4 address and attach it to the instance.

CONSOLE

```
$ vultr-cli instance ipv4 create <instance-id> --reboot
```

4. Create a new IPv4 reverse DNS entry on the instance.

CONSOLE

```
$ vultr-cli instance reverse-dns set-ipv4 <instance-id> --
entry <domain> --ip <ipv4-address>
```

IPv6

A guide explaining how to configure IPv6 addressing on your Vultr Optimized Cloud Compute instance.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Add IPv6 Addresses on a Vultr Optimized Cloud Compute Instance

Introduction

IPv6 is available but a public address is not automatically assigned to a Vultr Optimized Cloud Compute instance unless enabled during instance configuration. Once enabled, you can manage the instance's IPv6 network settings and configure reverse DNS for specific networking tasks.

Follow this guide to add the IPv6 network information on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Click **IPv6** on the left navigation menu.
5. Click **Assign IPv6 Network** to assign a new subnet to the instance.
6. Click **Assign IPv6 Network Address** to attach an IPv6 address to the instance.
7. Find the **Reverse DNS** section, enter your **IPv6** address in the **IP** field, and a domain in the **Reverse DNS** field to enable reverse DNS.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `GET` request to the [Get Instance IPv6 Information endpoint](#) to view the instance's IPv6 information.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/  
ipv6" \  
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

3. Send a `POST` request to the [Create Instance Reverse IPv6 endpoint](#) to create a new reverse DNS entry on the IPv6 address.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/  
ipv6/reverse" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "ip" : "<ipv6-address>",  
  "reverse" : "<domain>"  
'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. List the instance's IPv6 network information.

CONSOLE

```
$ vultr-cli instance ipv6 list <instance-id>
```

3. Create a new IPv6 reverse DNS entry on the instance.

CONSOLE

```
$ vultr-cli instance reverse-dns set-ipv6 <instance-id> --  
entry <domain> --ip <ipv6-address>
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Enable IPv6 on the instance and (optionally) set reverse DNS.

TERRAFORM

```
# Enable IPv6 on the instance  
resource "vultr_instance" "occ" {  
  # ...existing fields (region, plan, os_id, label, etc.)  
  enable_ipv6 = true  
}  
  
# Optional: set reverse DNS for the instance's primary IPv6
```

```
# (v6 address is known after the instance exists)
resource "vultr_reverse_ipv6" "occ_ptr" {
  ip      = vultr_instance.occ.v6_main_ip
  reverse = "host.example.com."
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 0 added, 1 changed, 0 destroyed.
```

Reserved IPs

A guide explaining how to assign and use dedicated IP addresses with your Vultr Optimized Cloud Compute instance.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Attach Reserved IPs to a Vultr Optimized Cloud Compute Instance

Introduction

Reserved IPs allow you to reserve a specific public IP address you can assign to a Vultr Optimized Cloud Compute instance. You can attach multiple reserved IPs to a single instance to enable advanced networking capabilities like routing and IP forwarding with distinct public IP addresses.

Follow this guide to attach reserved IPs on a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** group and click **Reserved IPs**.
2. Click your target reserved IP to open its management page.
3. Click the **Attach to Server** drop-down and select your target Vultr Optimized Cloud Compute instance.
4. Click **Attach** to apply the reserved IP to the Vultr Optimized Cloud Compute instance.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `GET` request to the [List Reserved IPs endpoint](#) and note the target reserved IP's ID.

CONSOLE

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

3. Send a `POST` request to the [Attach Reserved IP endpoint](#) to attach the reserved IP to the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/reserved-ips/{reserved-ip}/  
attach" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "instance_id" : "<instance-id>"  
}'
```

Vultr CLI

1. List all reserved IPs in your Vultr account and note the target reserved IP's ID.

CONSOLE

```
$ vultr-cli reserved-ip list
```

2. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

3. Attach the reserved IP to the instance.

CONSOLE

```
$ vultr-cli reserved-ip attach <reserved-ip-id> --instance-id <instance-id>
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Create (or import) a Reserved IP and reference it from the instance using `reserved_ip_id`.

TERRAFORM

```
resource "vultr_reserved_ip" "public_ip" {
  region = "del"
  ip_type = "v4"           # or "v6"
  label   = "occ-reserved-ip"
}

resource "vultr_instance" "occ" {
  # ...existing fields (region, plan, os_id, label, etc.)

  reserved_ip_id = vultr_reserved_ip.public_ip.id
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 1 added, 0 changed, 0 destroyed.
```

VPC 2.0

A private network solution that allows secure communication between Vultr resources in isolated environments.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Attach a VPC 2.0 Network to a Vultr Optimized Cloud Compute Instance

Introduction

A Virtual Private Cloud (VPC) 2.0 network creates a secure and isolated private networking interface to enable connections to other instances attached to the same network. You can attach multiple VPC 2.0 networks to enable secure connections between a Vultr Optimized Cloud Compute instance and other instances attached to the same VPC 2.0 network.

Follow this guide to attach a VPC 2.0 network to a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Click **VPC 2.0** on the left navigation menu.
5. Click **Enable VPC 2.0** to activate a new VPC 2.0 network interface.
6. Click **Enable VPC 2.0** in the confirmation prompt to apply the changes.
7. Click the **VPC 2.0** drop-down to select a specific network and click **Attach** to apply the changes on your instance.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `GET` request to the [List Instance VPC 2.0 Networks endpoint](#) to list all VPC 2.0 networks in your Vultr account and note the target VPC 2.0 network's ID.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpc2" \  
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

3. Send a `POST` request to the [Attach VPC 2.0 to Instance endpoint](#) to attach a VPC 2.0 network to the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/vpc2/attach" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "vpc_id": "<vpc2-id>"  
}'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. List all available VPC 2.0 networks and note the target VPC 2.0 network ID.

CONSOLE

```
$ vultr-cli vpc2 list
```

3. Attach the VPC 2.0 network to the instance.

CONSOLE

```
$ vultr-cli vpc2 nodes attach <vpc2-id> \  
  --nodes="<instance-id>"
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Create (or reference) a VPC 2.0 network and attach it to the instance.

TERRAFORM

```
# Create a VPC 2.0 network  
resource "vultr_vpc2" "app_net" {  
  region      = "del"  
  description = "Private network for OCC workloads"  
}  
  
# Attach the VPC 2.0 network to the Optimized Cloud Compute
```

```
instance
resource "vultr_instance" "occ" {
  label      = "occ-01"
  region    = "del"
  plan      = "vhp-2c-4gb"
  os_id     = 2284
  hostname  = "occ-01"

  vpc2_ids = [vultr_vpc2.app_net.id]
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

VPC

A private network solution that allows secure communication between Vultr instances without using public internet connections.

Contents

01	Introduction	6
02	Vultr Customer Portal	6
03	Vultr API	8
04	Vultr CLI	9
05	Terraform	10

How to Attach a VPC Network to a Vultr Optimized Cloud Compute Instance

Introduction

A Virtual Private Cloud (VPC) network creates a secure and isolated private networking interface to enable connections to other instances attached to the same network. You can attach multiple VPC 2.0 networks to enable secure connections between a Vultr Optimized Cloud Compute instance and other instances attached to the same VPC network.

Follow this guide to attach a VPC network to a Vultr Optimized Cloud Compute instance using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target Vultr Optimized Cloud Compute instance to open its management page.
3. Navigate to the **Settings** tab.
4. Click **IPv4** on the left navigation menu.
5. Click **Enable VPC**.
6. Click **Enable VPC** in the confirmation prompt to apply the changes.
7. Click **Attach VPC** in the confirmation prompt to apply changes to your instance.

Vultr API

1. Send a `GET` request to the [List Instances endpoint](#) and note your target instance's ID.

CONSOLE

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

2. Send a `GET` request to the [List VPCs endpoint](#) to list all available VPCs and note the target VPC network ID.

CONSOLE

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

3. Send a `POST` request to the [Attach VPC to Instance endpoint](#) to attach the VPC network to the instance.

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances/{instance-id}/vpcs/attach" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "vpc_id": "<vpc-id>"  
}'
```

Vultr CLI

1. List all available instances and note your target instance's ID.

CONSOLE

```
$ vultr-cli instance list
```

2. List all available VPC networks and note the target VPC network ID.

CONSOLE

```
$ vultr-cli vpc list
```

3. Attach the VPC network to the instance.

CONSOLE

```
$ vultr-cli instance vpc attach <instance-id> <vpc-id>
```

Terraform

1. Open your Terraform configuration for the existing Optimized Cloud Compute instance.
2. Create (or reference) a VPC network and attach it to the instance.

TERRAFORM

```
# Create a VPC network
resource "vultr_vpc" "app_net" {
  region      = "del"
  description = "Private network for OCC workloads"
}

# Attach the VPC network to the Optimized Cloud Compute
instance
resource "vultr_instance" "occ" {
  label      = "occ-01"
  region    = "del"
  plan      = "vhp-2c-4gb"
  os_id     = 2284
  hostname  = "occ-01"
}
```

```
vpc_ids = [vultr_vpc.app_net.id]
}
```

3. Apply the configuration and observe the following output:

```
Apply complete! Resources: 2 added, 0 changed, 0 destroyed.
```

FAQ

A collection of frequently asked questions and answers about Vultr services and features.

Contents

01	Introduction	6
02	How can I access a Vultr Optimized Cloud Compute instance?	141
03	Can I run resource-intensive applications on a Vultr Optimized Cloud Compute instance?	141
04	Can I downgrade a Vultr Optimized Cloud Compute instance?	141
05	Does a Vultr Optimized Cloud Compute instance incur any charges when stopped?	142
06	Can I change my Vultr Optimized Cloud Compute instance type?	142

Frequently Asked Questions (FAQs) About Optimized Cloud Compute

Introduction

These are the frequently asked questions for Vultr Optimized Cloud Compute instances.

How can I access a Vultr Optimized Cloud Compute instance?

You can access a Vultr Optimized Cloud Compute instance depending on its operating system. Use SSH or the Vultr Console for Linux instances and the Remote Desktop Protocol for Windows instances.

Can I run resource-intensive applications on a Vultr Optimized Cloud Compute instance?

Yes, Vultr Optimized Cloud Compute supports resource-intensive applications depending on your server size. For example, if your applications require more RAM, deploy a memory-optimized instance.

Can I downgrade a Vultr Optimized Cloud Compute instance?

No, you cannot downgrade a Vultr Optimized Cloud Compute instance, but can upgrade the instance to higher plans as your needs grow.

Does a Vultr Optimized Cloud Compute instance incur any charges when stopped?

Yes, a stopped instance incurs normal charges as it would while running. However, the instance does not incur any charges when destroyed which may lead to data loss.

Can I change my Vultr Optimized Cloud Compute instance type?

Yes, you can change your Vultr Optimized Cloud Compute instance type. Navigate to **Settings -> Change Plan** to modify the instance type. For example, you can change a CPU-optimized instance to a memory-optimized instance.



VULTR

