

Firewall

A security feature that allows you to control network traffic to your Vultr resources by configuring access rules.

Contents

01	Introduction	3
02	Vultr Customer Portal	3
03	Vultr API	3
04	Vultr CLI	4
05	Terraform	5

How to Configure Firewall Rules for Vultr Load Balancer

Introduction

Configuring Firewall Rules for Vultr Load Balancer allows you to control the traffic that can reach your load-balanced applications. By setting up firewall rules, you can specify which IP addresses and ports are allowed or blocked, enhancing the security of your services. This feature helps protect your backend servers from unauthorized access and potential threats by filtering incoming traffic based on your defined criteria.

Follow this guide to configure firewall rules for your Vultr Load Balancer using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products** and click **Load Balancers**.
2. Click your target Load Balancer to open its management page.
3. Scroll down to **Load Balancer Configuration**, then click the pencil icon in the **Firewall Rules** section.
4. Add rules by specifying **Port**, **IP Type (v4/v6)**, and **Source** (CIDR).
5. Click **Save changes**.

Vultr API

1. Send a `GET` request to the [List Load Balancers endpoint](#) and note the target Load Balancer's ID.

CONSOLE

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

2. Send a `PATCH` request to the [Update Load Balancer endpoint](#) to add a firewall rule to the target Load Balancer's algorithm.

CONSOLE

```
$ curl "https://api.vultr.com/v2/load-balancers/{load-balancer-id}" \  
-X PATCH \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "firewall_rules": [  
    {  
      "port" : {allowed_port_number},  
      "ip_type" : "{v4_or_v6}",  
      "source" : "{source_ip_cidr}"  
    }  
  ]  
}'
```

3. Send a `GET` request to the [List Firewall Rules endpoint](#) to view all firewall rules set for the target Load Balancer.

CONSOLE

```
$ curl "https://api.vultr.com/v2/load-balancers/{load-balancer-id}/firewall-rules" \  
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

Vultr CLI

1. List all available instances and note the target Load Balancer's ID.

CONSOLE

```
$ vultr-cli load-balancer list
```

2. Add firewall rule to the target Load Balancer.

CONSOLE

```
$ vultr-cli load-balancer update <loadbalancer-id> --  
firewall-rules  
"port:<allowed_port_number>,ip_type:<v4_or_v6>,source:<source  
_ip_cidr>"
```

3. View all firewall rules set for the target Load Balancer.

CONSOLE

```
$ vultr-cli load-balancer firewall list <load-balancer-id>
```

Terraform

1. Open your Terraform configuration file for the existing Load Balancer.
2. Add `firewall_rules` blocks to allow specific ports and sources, then apply.

TERRAFORM

```
resource "vultr_load_balancer" "lb" {  
  # ...existing fields (region, label, forwarding_rules,  
  health_check, etc.)  
  
  firewall_rules {  
    frontend_port = 443  
    ip_type       = "v4"  
    source        = "0.0.0.0/0"  
  }  
  
  firewall_rules {  
    frontend_port = 8080  
    ip_type       = "v6"  
    source        = ":::/0"  
  }  
}
```

```
}  
}
```

3. Apply the configuration and observe the following output:

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



VULTR

