

Network

Manage your Vultr network infrastructure including routing, security, and domain services.

Contents

| | | |
|-----------|--|------------|
| 01 | BGP | 4 |
| | ASN Info | 6 |
| | AS20473 | 8 |
| | AS64515 | 13 |
| | AS20473 BGP Communities Customer Guide | 17 |
| | Access Credentials | 24 |
| | Add Prefixes | 28 |
| | Request Access | 31 |
| | RPKI | 36 |
| 02 | DNS | 41 |
| | Provisioning | 43 |
| | Management | 48 |
| | Manage Records | 50 |
| | Manage Zone Settings | 56 |
| | Delete Domains | 62 |
| | Configure rDNS | 67 |
| | Point to Vultr Name Servers | 72 |
| | FAQ | 77 |
| 03 | Firewall | 81 |
| | Provisioning | 83 |
| | Management | 88 |
| | Delete | 90 |
| | Groups | 96 |
| | Link | 101 |
| | Rules | 106 |
| | FAQ | 112 |
| 04 | Reserved IPs | 116 |
| | Provisioning | 118 |

| | |
|-----------------------|------------|
| Management | 123 |
| Attachment | 125 |
| Convert Existing IP | 130 |
| Delete | 134 |
| Detachment | 139 |
| FAQ | 144 |
| 05 VPC 2.0 | 147 |
| Provisioning | 149 |
| Management | 154 |
| Attachment | 156 |
| Delete | 161 |
| Monitor | 166 |
| FAQ | 171 |
| 06 VPC | 175 |
| Provisioning | 177 |
| Management | 182 |
| Delete | 184 |
| Monitor | 189 |
| FAQ | 193 |
| NAT Gateway | 197 |
| Provisioning | 199 |
| Management | 204 |
| Delete | 206 |
| Get Info | 210 |
| Configuration | 215 |
| Firewall Rules | 217 |
| Create | 219 |
| Delete | 225 |
| List | 230 |
| Update | 235 |
| Read | 240 |
| Port Forwarding Rules | 245 |
| Create | 247 |
| Delete | 252 |
| List | 257 |
| Read | 261 |
| Update | 266 |

BGP

Configure and manage Border Gateway Protocol (BGP) for advanced network routing with autonomous system numbers and IP prefixes on Vultr.

Contents

| | | |
|----|--|----|
| 01 | ASN Info | 6 |
| | AS20473 | 8 |
| | AS64515 | 13 |
| | AS20473 BGP Communities Customer Guide | 17 |
| 02 | Access Credentials | 24 |
| 03 | Add Prefixes | 28 |
| 04 | Request Access | 31 |
| 05 | RPKI | 36 |

ASN Information

Provides information about Autonomous System Numbers (ASNs) used to identify networks on the internet.

Contents

| | | |
|----|--|----|
| 01 | AS20473 | 8 |
| 02 | AS64515 | 13 |
| 03 | AS20473 BGP Communities Customer Guide | 17 |

AS20473

Vultr's public Autonomous System Number (AS20473) used for routing traffic across the global internet.

Contents

| | | |
|----|--|----|
| 01 | Introduction | 10 |
| 02 | Understanding Vultr Public ASN - AS20473 | 10 |
| 03 | Peering with Vultr Using AS20473 | 10 |
| 04 | RPKI and Prefix Validation | 11 |

Vultr's Public ASN - AS20473

Introduction

Vultr offers **Public ASN (AS20473)** for managing network routing and peering with external networks. This ASN is essential for advertising IP prefixes and enabling effective routing policies for your network.

Follow this guide to configure and use **AS20473** for your peering setup, focusing on its use within Vultr's network infrastructure.

Understanding Vultr Public ASN - AS20473

Public ASN: Vultr's **Public ASN (AS20473)** is used for external BGP peering and managing routing policies with networks outside of Vultr's infrastructure.

- **AS20473** is assigned to customers who need to establish public peering, typically for Bare Metal instances. It allows for advertising IP address space and establishing routing relationships with external peers or transit providers.

Note

Public ASN (AS20473) can only be used for peering with **Vultr's Bare Metal instances**. **Private ASN (AS64515)** is not available for Vultr Bare Metal instances.

Peering with Vultr Using AS20473

When you peer with **Vultr's Bare Metal instances** using **AS20473** (public ASN) or your own ASN, you can advertise your IP prefixes. Your ASN will be used to advertise these prefixes and manage the routing to external peers or transit providers.

Public ASN Peering:

Peering with **AS20473** allows you to control your routing policies and advertise your IP space to the broader internet. This includes the ability to prepend your ASN in the **AS_PATH** when advertising your IP prefixes.

- **Example of ASN Prepending:** If your primary ASN is **AS64500** and you also use **AS64501** and **AS64502** as secondary ASNs, your advertised prefixes to Vultr's network could look like this:

- 64500
- 64500_64501
- 64500_64502
- 64500_64501_64502

When Vultr advertises these prefixes to its transit providers, it will prepend **AS20473** (Vultr's public ASN), resulting in the following **AS_PATHs** on the public internet:

- 20473_64500
- 20473_64500_64501
- 20473_64500_64502
- 20473_64500_64501_64502

In this case, **AS64500** will always appear before **AS64501** and **AS64502**, indicating the customer's primary ASN.

Note

For more information about **AS20473** BGP policies, including **AS20473 Action Communities and Informational Communities**, such as **Prefix Type** and **Locations**, visit the [AS20473 BGP Customer Guide](#).

RPKI and Prefix Validation

Vultr uses **Resource Public Key Infrastructure (RPKI)** to validate BGP announcements and ensure the integrity of network prefixes. When configuring **AS20473**, it's important to ensure that only **valid RPKI prefixes** are

advertised to external networks. This helps mitigate the risks of **hijacking** or **misrouting**.

AS64515

is a private Autonomous System Number (AS64515) used for internal network routing and connectivity between Vultr infrastructure.

Contents

| | | |
|----|---|----|
| 01 | Introduction | 10 |
| 02 | Understanding Vultr Private ASN - AS64515 | 15 |
| 03 | Peering with Vultr Using AS64515 | 16 |
| 04 | RPKI and Prefix Validation | 11 |

Vultr's Private ASN - AS64515

Introduction

Vultr offers **Private ASN (AS64515)** for internal routing and network management within its infrastructure. This ASN is specifically designed for use with Vultr's network services, and it ensures private, isolated routing of your IP addresses within Vultr's infrastructure.

Follow this guide to configure and use **AS64515** as your private ASN with Vultr's network, focusing on internal routing and BGP peering.

Understanding Vultr Private ASN - AS64515

Private ASN: An ASN used within a private network for internal routing and network management.

- **AS64515** is an ASN designated for private use within Vultr's internal network infrastructure.
- It is assigned for customers who wish to use private peering for managing their reserved IPs and internal routing within Vultr's infrastructure.
- If you do not have your own ASN, **AS64515** is automatically assigned when you choose to broadcast your Reserved IPs.

Note

Private ASN (AS64515) can only be used for peering with Vultr Cloud Compute instances for internal routing. It is not available for Vultr Bare Metal instances.

Peering with Vultr Using AS64515

When peering using **AS64515**, you will establish BGP sessions for private, internal routing. The **AS64515** ASN will only be used within Vultr's network for managing IPs, ensuring a secure and private routing setup.

Private ASN Peering:

- Customers using **AS64515** will only perform **internal BGP routing** within Vultr's infrastructure.
- **AS64515** does not allow public ASN peering and is not used for advertising to external networks or the public internet.
- All routing with **AS64515** will be confined to Vultr's internal network, offering private and secure connectivity for your resources.

Example: When using **AS64515**, you will be able to configure your private BGP sessions, and the **AS_PATH** will only include **AS64515**, with no modifications for external network peering.

RPKI and Prefix Validation

Vultr uses Resource Public Key Infrastructure (RPKI) to validate BGP announcements and ensure the integrity of network prefixes. When using **AS64515**, it's crucial to ensure that only **valid RPKI prefixes** are announced within Vultr's internal network to minimize the risk of misrouting or hijacking.

AS20473 BGP Communities Customer Guide

Comprehensive guide to AS20473 BGP and large community tags.

Contents

| | | |
|----|---------------------------|----|
| 01 | Informational Communities | 19 |
| 02 | Action Communities | 22 |
| 03 | Other BGP Communities | 23 |

AS20473 BGP Communities

Customer Guide

Informational Communities

Prefix Type

AS20473 tags prefixes that are learned or originated as follows:

| Prefix Type | Community | Large Community |
|---|------------------|--|
| Prefix learned from Transit | 20473:100 | 20473:100:transit-as |
| Prefix learned from Public Peer via route servers | 20473:200 | 20473:200:ixp-as |
| Prefix learned from Public Peer via bilateral peering | 20473:200 | 20473:200:ixp-as, 20473:200:peer-as |
| Prefix learned from Private Peer | 20473:300 | 20473:300:peer-as |
| Prefix originated by Customer | 20473:4000 | |
| Prefix originated by 20473 | 20473:500 | |
| Prefix learned from AS number (if ≤ 65535) | 20473:peer-as | 20473:peer-type:peer-as |

Location

Routes announced from AS20473 are also tagged with a 2-digit community to provide information about the POP it was originated from. For example, prefixes tagged with 20473:11 are generated in Piscataway. These locations are defined in the following table.

United States

| POP Name | Code |
|-----------------|-------------|
| Atlanta, GA | 14 |
| Chicago, IL | 13 |
| Dallas, TX | 15 |
| Honolulu, HI | 34 |
| Los Angeles, CA | 17 |
| Miami, FL | 12 |
| Piscataway, NJ | 11 |
| San Jose, CA | 18 |
| Saint Louis, MO | 47 |
| Seattle, WA | 16 |

Other North & South America

| POP Name | Code |
|-----------------|-------------|
| Mexico, MX | 28 |
| Santiago, CL | 42 |
| Sao Paulo, BR | 30 |
| Toronto, CA | 33 |

Europe

| POP Name | Code |
|-----------------|-------------|
| Amsterdam, NL | 20 |
| Frankfurt, DE | 22 |
| London, GB | 19 |
| Madrid, ES | 31 |
| Manchester, GB | 46 |
| Milan, IT | 38 |
| Paris, FR | 21 |
| Stockholm, SE | 27 |

| POP Name | Code |
|-----------------|-------------|
| Warsaw, PL | 29 |

Asia

| POP Name | Code |
|-----------------|-------------|
| Bangalore, IN | 43 |
| Korea, KR | 26 |
| Mumbai, IN | 35 |
| New Delhi, IN | 44 |
| Osaka, JP | 45 |
| Singapore, SG | 25 |
| Tel Aviv, IL | 36 |
| Tokyo, JP | 23 |

Oceania

| POP Name | Code |
|-----------------|-------------|
| Melbourne, AU | 32 |
| Sydney, AU | 24 |

Africa

| POP Name | Code |
|------------------|-------------|
| Johannesburg, ZA | 37 |

Large communities are also used for location with the following format

`20473:0:3RRRCCC1PP` where:

- `RRR` is the [M49 region code](#)
- `CCC` is the [M49 country code](#)
- `PP` is the location code as explained above

Action Communities

Customers may choose to influence traffic for prefixes advertised outside of AS20473 using the communities below. Customers may also add third party communities which are passed on to our providers and peers. We have incorporated large communities to support actions on 32-bit autonomous system numbers.

IXP route servers use large communities only!

| Action | Community | Large Community |
|--|---------------|--------------------|
| Do not advertise to specific AS | 64600:peer-as | 20473:6000:peer-as |
| Prepend 1x to specific AS | 64601:peer-as | 20473:6001:peer-as |
| Prepend 2x to specific AS | 64602:peer-as | 20473:6002:peer-as |
| Prepend 3x to specific AS | 64603:peer-as | 20473:6003:peer-as |
| Set Metric to 0 to specific AS | 64609:peer-as | 20473:6009:peer-as |
| Override 20473:6000 to specific AS | 64699:peer-as | 20473:6099:peer-as |
| Do not advertise out of AS20473 | 20473:6000 | |
| Prepend 1x to all peers | 20473:6001 | |
| Prepend 2x to all peers | 20473:6002 | |
| Prepend 3x to all peers | 20473:6003 | |
| Set Metric to 0 to all peers | 20473:64609 | |
| Do not announce to IXP peers | 20473:6601 | |
| Announce to IXP route servers only | 20473:6602 | |
| Override "do not advertise" for IX route servers & peers | 20473:6603 | |

| Action | Community | Large Community |
|-------------------------------|-----------|--------------------|
| Export blackhole to all peers | 20473:666 | |

Other BGP Communities

AS20473 transparently advertises communities set by its customers. Customers can use these communities to affect their inbound traffic from our different transit providers and IXPs.

Access Credentials

A guide explaining how to find and access your BGP routing credentials for Vultr network configurations

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Retrieve BGP Credentials

Introduction

After BGP access is approved for your Vultr account, you can retrieve your credentials to configure your instances with BGP routing daemons. BGP credentials include your ASN, password, and configuration examples.

Note

You must have at least one active **Compute instance** to view your BGP credentials in the Customer Portal.

Follow this guide to retrieve your BGP credentials using the Vultr Customer Portal and API.

Vultr Customer Portal

1. Navigate to **Products** and click **Compute**.
2. Click your target instance (VPS or Bare Metal) to open its management page.
3. Navigate to the **BGP** section on the instance management page to view your **BGP** credentials.

Vultr API

- Send a `GET` request to the [Get Account BGP Info endpoint](#) to retrieve your account's BGP credentials.

CONSOLE

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

A successful response will include `enabled`, `asn`, and `password` fields.

Add Prefixes

A guide explaining how to add network prefixes to your Vultr infrastructure

Contents

| | |
|-----------------|----|
| 01 Introduction | 10 |
|-----------------|----|

How to Add Prefixes

Introduction

Adding prefixes allows you to control which IP address blocks are advertised to your BGP peers. This is essential for managing routing policies and ensuring that only authorized prefixes are announced.

Follow the steps below to add the prefixes using the Vultr Customer Portal.

1. Navigate to **Products**, expand the **Network** menu, and click **BGP**.
2. On the BGP page, you will see the previously added Prefix List, including the Date Added and their current RPKI status.
3. Enter your new prefix in the **Prefix Insert** column and click **Add New Prefix** to add the prefix.

Note

In order to remove a prefix, [please create a support ticket](#).

Request Access

A guide explaining how to request and set up Border Gateway Protocol (BGP) access for your Vultr account.

Contents

| | | |
|----|--------------------|----|
| 01 | Introduction | 10 |
| 02 | Request BGP Access | 33 |
| 03 | LOA Template | 35 |

How to Request BGP Access

Introduction

Border Gateway Protocol (BGP) enables you to announce your own IP space or use a Vultr-assigned private ASN to broadcast Reserved IPs. Requesting BGP access enables advanced network routing and provides greater flexibility for network management. You can request BGP access by submitting your ASN, IP block details, or use a private ASN provided by Vultr for broadcasting Reserved IPs.

Note

You must have at least one active Compute instance in your Vultr Customer Portal before you can request BGP access.

Follow this guide to request BGP access through the Vultr Customer Portal.

Request BGP Access

1. Navigate to **Products**, expand the **Network** menu, and click **BGP**.
2. On the **Getting Started with BGP page**, click **Get Started** to begin the BGP setup process.
3. Provide your BGP setup details:
 - **Option 1:** Using Your Own IP Space.

If you have your own ASN, IP Space and want to use it with Vultr instances. Follow these steps:

1. Toggle **I have my own IP space** option to use your IP blocks registered with a regional internet registry (e.g., ARIN, RIPE, APNIC).

2. Enter your ASN, BGP password, and the IP block(s) you wish to announce.
 3. Upload your Letter of Authorization (LOA) for IP announcements.
- **Option 2:** Using Vultr's IP Space.

If you do not have your own ASN, you must use the Vultr's ASN and IP Space. Follow these steps:

1. Do not toggle any options under **I have my own IP space**.
2. In this case, Vultr will assign you a private ASN for broadcasting Reserved IPs.

Note

Vultr Reserved IPs are region-bound and can only be assigned to instances in the same region using BGP where the Reserved IP is located. This ensures that your network traffic is routed efficiently within the specific region.

4. Choose your preferred route announcement option under **Routes we should send you**:
 - **No Routes:** Do not send any routes.
 - **Default Only:** Receive a default BGP routing table.
 - **Full Table:** Receive the full BGP routing table.

Note

The **Full Table** routing option is not supported on **Bare Metal instances**.

5. In the **Description of use case for BGP**, enter a short explanation describing your intended BGP use.
6. Click **Request BGP Setup** to submit the form. A Vultr team member will review your request and enable BGP for your account.

LOA Template

If you are using your own IP space, you must upload a Letter of Authorization (LOA). Below is a template for the LOA you can use:

AUTHORIZATION LETTER

[DATE]

To whom it may concern,

This letter serves as authorization for Vultr with AS20473 to announce the following IP address blocks:

[IP SPACE / ASN / SUBNET]

[IP SPACE / ASN / SUBNET]

[...]

As a representative of the company [COMPANY] that is the owner of the subnet and/or ASN, I hereby declare that I'm authorized to represent and sign for this LOA.

Should you have questions about this request, email me at [E-MAIL ADDRESS], or call: [TELEPHONE NUMBER]

From,

[NAME]

[COMPANY]

[TITLE]

[TELEPHONE NUMBER]

This LOA template ensures that Vultr has the required permission to announce the provided IP blocks to external peers on your behalf.

RPKI

A security framework that validates the ownership of IP address blocks to prevent BGP route hijacking and improve internet routing security.

Contents

| | | |
|----|--|----|
| 01 | Introduction | 10 |
| 02 | About Route Origin Authorization (ROA) | 38 |
| 03 | RPKI Status | 40 |
| 04 | Configuring RPKI | 40 |

About Resource Public Key Infrastructure (RPKI) at Vultr

Introduction

Resource Public Key Infrastructure (RPKI) is a system designed to enhance the security of Border Gateway Protocol (BGP) by preventing BGP hijacking. It uses cryptographic signatures to validate that an Autonomous System Number (ASN) is permitted to announce a particular IP subnet. This system secures the internet's routing infrastructure and prevents malicious actors from redirecting or intercepting traffic.

About Route Origin Authorization (ROA)

Route Origin Authorization (ROA) is the core component of RPKI. A ROA specifies the allowed ASNs, IP prefixes, and the maximum prefix length that can be advertised. These ROAs are cryptographically signed and publicly published, allowing routers to verify that a given ASN is permitted to announce a specific IP prefix.

Example ROA With Maximum Length of /29

In the following example of a ROA, AS20473 is permitted to announce the 192.0.2.0/24 network and all smaller subnets within the /29 range:

```
{
  "asn" : "AS20473",
  "prefix" : "192.0.2.0/24",
  "maxLength" : 29,
  "ta" : "ARIN"
}
```

In the above example:

- **asn**: This must be a public ASN. If you're using a private ASN, your ROA should list Vultr's public ASN **AS20473**, as RPKI requires a globally routable ASN to authorize prefix announcements.
- **prefix**: This is the IP block being authorized. It defines the base prefix that the ASN is allowed to announce.
- **maxLength**: This specifies the most specific subnet (smallest prefix) that can be announced from the base prefix. In this case, any subnet between `/24` and `/29` is allowed, such as `/25`, `/26`, `/27`, `/28`, or `/29`.
- **ta**: This is the Trust Anchor or Regional Internet Registry that issued the prefix, such as ARIN, RIPE, or APNIC.

Example ROA With Exact Prefix Length

In the following example of a ROA, `AS20473` is permitted to announce only the exact `192.0.2.0/24` prefix, with no smaller subnets allowed:

```
{
  "asn" : "AS20473",
  "prefix" : "192.0.2.0/24",
  "maxLength" : 24,
  "ta" : "ARIN"
}
```

You can check individual ROAs using the [RIPE Validator](#), a public service provided by RIPE.

RPKI Status

Vultr performs nightly checks on the RPKI status of every customer subnet. You can view the RPKI status in the [BGP section](#) of your Vultr customer portal. You may encounter the following RPKI status states:

- **Valid:** This means that an ROA exists for the ASN/prefix pair, and everything is in order. This is the state you want to see for your subnets.
- **Unknown:** No ROA exists for the given prefix. This is common for many IP addresses and is typically not an issue, as most ISPs are not yet requiring RPKI validation. You will probably not encounter problems with this state.

Invalid Signatures

Several types of invalid signatures can prevent your IP space from being advertised across the internet. These need to be fixed:

- **Invalid ASN:** An ROA exists for this prefix, but none of the ASNs match what your account is configured for. If you're using a private ASN, your ROAs should list Vultr's public ASN `AS20473`.
- **Invalid Prefix Length:** An ROA exists for this prefix/ASN, but the maximum allowed prefix length is not correct. Typically, you would need to issue a new ROA with the correct maximum prefix length, such as `24` for IPv4 or `48` for IPv6. You can also issue a new ROA for a smaller prefix if needed.

Configuring RPKI

RPKI setup is managed through your Regional Internet Registry (RIR) (such as RIPE, ARIN, APNIC, etc.). Only the owner of IP space can manage RPKI ROAs.

If you are leasing IP space, contact your provider for help with configuring RPKI.

DNS

Manage your domain name system settings, configure DNS records, and utilize Vultr's name servers for reliable domain resolution.

Contents

| | | |
|----|-----------------------------|----|
| 01 | Provisioning | 43 |
| 02 | Management | 48 |
| | Manage Records | 50 |
| | Manage Zone Settings | 56 |
| | Delete Domains | 62 |
| | Configure rDNS | 67 |
| 03 | Point to Vultr Name Servers | 72 |
| 04 | FAQ | 77 |

Provisioning

The process of setting up and configuring a new server or service to make it ready for use.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Add a Domain to Vultr DNS

Introduction

Vultr DNS is a domain record management service that enables you to manage DNS records and zones using your domain. Vultr DNS uses Vultr's nameservers that run on the Anycast network to enable fast DNS resolution and application of domain changes.

Follow this guide to add a domain to Vultr DNS in your Vultr account using the Vultr Customer Portal, API, CLI, or Terraform.

Note

Point your domain's nameservers to `ns1.vultr.com`, `ns2.vultr.com` to manage it using Vultr DNS.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **DNS** from the list of options.
2. Click **Add Domain** to set up a new domain.
3. Enter your domain in the **Domain** field and click the **Vultr instance** drop-down to point the domain **A** record to your instance's IP address.
4. Click **Add** to apply the new domain and its DNS records using Vultr DNS.

Vultr API

1. Send a `GET` request to the [List DNS Domains endpoint](#) and verify all active domains in your Vultr account.

CONSOLE

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

2. Send a `POST` request to the [Create DNS Domain endpoint](#) to create a new domain to manage using Vultr DNS.

CONSOLE

```
$ curl "https://api.vultr.com/v2/domains" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "domain" : "<domain>",  
  "ip" : "<default-IP>",  
  "dns_sec" : "enabled"  
'
```

Visit the [Create DNS Domain API page](#) to view additional attributes to apply on the domain.

Vultr CLI

1. List all domains in your Vultr account.

CONSOLE

```
$ vultr-cli dns domain list
```

2. Create a new domain to manage using Vultr DNS.

CONSOLE

```
$ vultr-cli dns domain create --domain <domain> --ip  
<default-IP>
```

Run `vultr-cli dns domain create --help` to view additional options to apply on the domain.

Terraform

1. Ensure the [Vultr Terraform provider](#) is configured in your Terraform project.
2. Create a DNS domain and an A record, then apply.

TERRAFORM

```
resource "vultr_dns_domain" "example" {
  domain = "example.com"
}

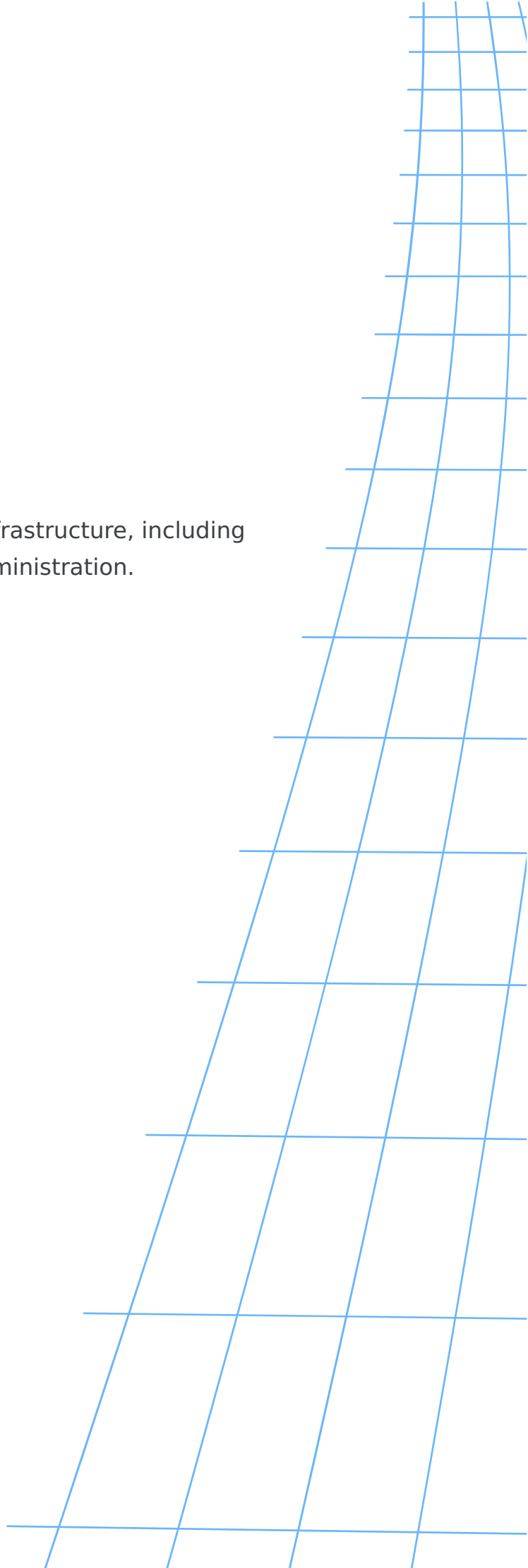
resource "vultr_dns_record" "www" {
  domain = vultr_dns_domain.example.domain
  name   = "www"
  type   = "A"
  data   = "192.0.2.1"
  ttl    = 300
}
```

3. Apply the configuration and observe the following output:

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

Management

Tools and features for managing your Vultr infrastructure, including access controls, monitoring, and resource administration.



Contents

| | | |
|----|----------------------|----|
| 01 | Manage Records | 50 |
| 02 | Manage Zone Settings | 56 |
| 03 | Delete Domains | 62 |
| 04 | Configure rDNS | 67 |

Manage Records

Configure and manage DNS records for your domains within the Vultr DNS management interface.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Manage Domain DNS Records using Vultr DNS

Introduction

Managing DNS records enables the activation of specific values that define a domain services such as IP addresses, TXT and Mail Exchange (MX) details. Vultr DNS supports all types of DNS records you can create for your domain.

Follow this guide to manage domain DNS records using Vultr DNS, the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** group and click **DNS** to view all domains in your account.
2. Click your target domain to manage its DNS records.
3. Click the **Type** drop-down and select your new DNS record type.
4. Enter your desired DNS path in the **Name** field.
5. Enter the DNS record value in the **Data** field.
6. Keep the default Time to Live (TTL) value in seconds or change it depending on your needs.
7. Set the DNS record priority in the **Priority** field if applicable.
8. Click **Add Record** within the **Actions** section to validate and apply the new DNS record on your domain.
9. Select a DNS record and click **Delete Records** to remove the specific DNS record from your domain.

Vultr API

1. Send a `GET` request to the [List DNS domains endpoint](#) and note the target domain in your output.

CONSOLE

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

2. Send a `POST` request to the [Create a DNS Record endpoint](#) to create a new DNS record on the domain.

CONSOLE

```
$ curl "https://api.vultr.com/v2/domains/{dns-domain}/  
records" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

3. Send a `GET` request to the [List Records endpoint](#) to view all active DNS records and note the target record ID.

CONSOLE

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

4. Send a `DELETE` request to the [Delete DNS Domain Record endpoint](#) to the DNS record from your domain.

CONSOLE

```
$ curl "https://api.vultr.com/v2/domains/{dns-domain}" \  
-X DELETE \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

Vultr CLI

1. List all domains available in your Vultr account and note the target domain name.

CONSOLE

```
$ vultr-cli dns domain list
```

2. Create a new DNS record on the domain.

CONSOLE

```
$ vultr-cli dns record create --domain <dns-domain> --type  
<record-type> --data <record-data> --name <record-  
specification> --ttl <ttl-value>
```

3. List all active DNS records on the domain and note the target record ID.

CONSOLE

```
$ vultr-cli dns record list <dns-domain>
```

4. Delete the DNS record.

CONSOLE

```
$ vultr-cli dns record delete <dns-domain> <dns-record-id>
```

Terraform

1. Open your Terraform configuration where the domain is defined.

2. Add a record, update fields, or remove the block to delete.

TERRAFORM

```
resource "vultr_dns_record" "txt_spf" {  
  domain = var.domain  
  name   = "@"  
  type   = "TXT"  
  data   = "v=spf1 include:mail.example.com ~all"  
  ttl    = 3600  
}
```

3. Apply the configuration and observe the following output:

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

Manage Zone Settings

Configure and customize DNS zone behavior including TTL, DNSSEC, and other domain-specific settings.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Manage DNS Zones on Vultr DNS

Introduction

DNS Zones contain domain mapping information and essential DNS data associated with specific resources. Managing DNS Zone settings enables the activation of SOA (State of Authority) information and DNSSEC (Domain Name System Security Extensions) on a domain.

Follow this guide to manage domain DNS Zones using Vultr DNS, the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** group and click **DNS** to view all domains in your account.
2. Click your target domain to manage its DNS records.
3. Navigate to the **Zone Settings** tab.
4. Toggle the **DNSSEC Settings** status option to **Enabled** to generate a new DNS key and DS Records with cryptographic keys to submit to your domain registrar.
5. Replace `ns1.vultr.com` with your primary name server if available and enter your domain administrator email address in the **E-mail Address** field.
6. Click **Update SOA Record** to apply the domain SOA Information.

Vultr API

1. Send a `GET` request to the [List DNS domains endpoint](#) and note the target domain in your output.

CONSOLE

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

2. Send a `GET` request to the [Get DNSSec Info endpoint](#) to view the domain's DNSSec status.

CONSOLE

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

3. Send a `PUT` request to the [Update a DNS Domain endpoint](#) to enable DNSSec on the domain.

CONSOLE

```
$ curl "https://api.vultr.com/v2/domains/{dns-domain}" \  
-X PUT \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "dns_sec" : "enabled"  
}'
```

4. Send a `GET` request to the [Get SOA Information endpoint](#) to view the domain's SOA configuration.

CONSOLE

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

5. Send a `PATCH` request to the [Update SOA Information endpoint](#) to update the domain's SOA configuration.

CONSOLE

```
$ curl "https://api.vultr.com/v2/domains/{dns-domain}/soa" \
-X PATCH \
-H "Authorization: Bearer ${VULTR_API_KEY}" \
-H "Content-Type: application/json" \
--data '{
  "nsprimary" : "primary-nameserver",
  "email" : "adminEmail"
}'
```

Vultr CLI

1. List all domains in your Vultr account and note the target domain.

CONSOLE

```
$ vultr-cli dns domain list
```

2. View the domain's DNSSEC status.

CONSOLE

```
$ vultr-cli dns domain dnssec-info <domainName>
```

3. Enable DNSSEC on the domain.

CONSOLE

```
$ vultr-cli dns domain dnssec <domainName> --enabled enabled
```

4. View the domain's SOA information.

CONSOLE

```
$ vultr-cli dns domain soa-info <domainName>
```

5. Update the domain's SOA information.

CONSOLE

```
$ vultr-cli dns domain soa-update <domainName> --email  
<adminEmail>--ns-primary <primary-nameserver>
```

Terraform

1. Open your Terraform configuration where the domain is defined.
2. DNSSEC is managed via `vultr_dns_domain` attributes; SOA fields are not exposed via Terraform at this time.

TERRAFORM

```
resource "vultr_dns_domain" "example" {  
  domain = "example.com"  
  dns_sec = "enabled" # enabled | disabled  
}
```

3. Apply the configuration and observe the following output:

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

Delete Domains

Learn how to permanently remove a domain from your Vultr DNS management system.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Delete a Domain from Vultr DNS

Introduction

Deleting a domain removes it and all existing DNS records such as zones from Vultr DNS. DNS records cannot be recovered when a domain is deleted from Vultr DNS.

Follow this guide to delete a domain from Vultr DNS using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** group and click **DNS** to view all domains in your account.
2. Click **Delete Domain** next to your target domain.
3. Click **Delete Domain** in the confirmation prompt to delete the target domain.

Vultr API

1. Send a `GET` request to the [List DNS domains endpoint](#) and note the target domain in your output.

CONSOLE

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

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

CONSOLE

```
$ curl "https://api.vultr.com/v2/domains/{dns-domain}" \  
-X DELETE \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

Vultr CLI

1. List all domains available in your Vultr account and note the target domain.

CONSOLE

```
$ vultr-cli dns domain list
```

2. Delete the domain.

CONSOLE

```
$ vultr-cli dns domain delete <domain>
```

Terraform

1. Open your Terraform configuration where the domain is defined.
2. Remove the `vultr_dns_domain` resource block, or destroy it by target.

TERRAFORM

```
resource "vultr_dns_domain" "example" {  
  domain = "example.com"  
}  
  
# To delete, either remove this block from configuration  
# or run: terraform destroy -target vultr_dns_domain.example
```

3. Apply the configuration and observe the following output:

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

Configure rDNS

Learn how to configure reverse DNS (rDNS) on Vultr instances using the Customer Portal or Vultr CLI.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr CLI | 46 |

Introduction

Reverse DNS (rDNS) maps an IP address to a hostname using a PTR record, performing the opposite of a standard DNS lookup. Configuring rDNS allows services and monitoring tools to resolve your server's IP back to a fully qualified domain name (FQDN). Proper rDNS improves email deliverability, aids in network verification, and helps troubleshoot connectivity issues. Many services, including mail servers, rely on rDNS to verify legitimate sources, and monitoring tools use it to display human-readable hostnames.

Follow this guide to configure rDNS for Vultr instances using the Vultr Customer Portal or Vultr CLI.

Note

rDNS changes may take 6–12 hours to propagate and become active across the DNS system.

Vultr Customer Portal

1. Navigate to the **Compute** section under **Products**.
2. Select the instance for which you want to configure rDNS.
3. Click **Settings**, then choose **IPv4** or **IPv6** for which you want to set the rDNS.
4. Under **Public Network**, all your associated public IPs are listed. To update one, click the rDNS name under the **Reverse DNS** column.
5. Enter the **FQDN** you want to associate with your IP address.
6. Click the **tick icon** to save the rDNS record.
7. Run the below command to verify that the rDNS record is updated.

```
CONSOLE
```

```
$ dig -x <instance-ip>
```

The command above displays the PTR record associated with your IP.

Vultr CLI

1. Check the version of the Vultr CLI to ensure it is installed.

CONSOLE

```
$ vultr-cli version
```

Output:

```
Vultr-CLI v3.7.0
```

2. List all available instances.

CONSOLE

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

Note the `id` of the instance for which you want to update rDNS.

3. List all the IPv4 or IPv6 addresses associated with your instance.

- **For IPv4 rDNS:**

1. List all associated IPv4 addresses for the instance.

CONSOLE

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

Note the IP address you want to configure rDNS for.

2. Update the rDNS for the selected IPv4 address.

CONSOLE

```
$ vultr-cli instance reverse-dns set-ipv4 <instance-id> <ipv4-address> --entry "<FQDN-rDNS>"
```

Output:

```
Reverse DNS IPv4 has been set
```

- **For IPv6 rDNS:**

1. List all associated IPv6 addresses for the instance.

CONSOLE

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

2. Update the rDNS for the selected IPv6 address.

CONSOLE

```
$ vultr-cli instance reverse-dns set-ipv6 <instance-id> <ipv6-address> --entry "<FQDN-rDNS>"
```

Output:

```
Reverse DNS IPv6 has been set
```

4. Verify that the rDNS records have been updated.

CONSOLE

```
$ dig -x <instance-ip>
```

Point to Vultr Name Servers

Instructions for pointing your domain to Vultr's name servers from popular domain registrars

Contents

| | | |
|----|---------------------------------------|----|
| 01 | Introduction | 10 |
| 02 | Why Use Vultr DNS? | 74 |
| 03 | DNS Management Options | 74 |
| 04 | Vultr Name Servers | 75 |
| 05 | Update Name Servers at Your Registrar | 75 |
| 06 | Begin Managing DNS with Vultr | 76 |

Point to Vultr Name Servers From Common Domain Registrars

Introduction

Adding a domain you own to your Vultr account allows you to manage its DNS records using Vultr's DNS control panel or Vultr API. This lets you map your domains to your cloud infrastructure and manage records like **A**, **CNAME**, **MX**, and **TXT** directly within the [Vultr DNS dashboard](#).

Why Use Vultr DNS?

To manage your domain's DNS using Vultr, you must delegate your domain to use Vultr's name servers. This involves updating your domain's name server settings at your domain registrar (the seller from where you purchased your domain). Vultr is not a domain registrar and does not offer domain registration services.

Domain registrars such as [Namecheap](#), [GoDaddy](#), and others allow you to either manage DNS records directly on their platforms or delegate DNS to another provider such as Vultr. DNS records control how your domain behaves on the internet, including routing web traffic and email.

DNS Management Options

If you set up a server on Vultr and want to map your domain to that instance, you have two main options:

- Manage your domain's DNS records directly through your domain registrar control panel.

- Delegate your domain to Vultr and manage its DNS records from the [Vultr DNS dashboard](#).

We recommend using Vultr DNS for record management when hosting services with Vultr, as it centralizes control and simplifies configuration within the same environment.

Vultr Name Servers

To use Vultr's DNS service, you must update your domain's name servers at your registrar. Name servers act as the authoritative source for your domain's DNS records. When a user accesses your domain in a browser, DNS queries are directed to the configured name servers to retrieve the appropriate records.

Update your domain to use the following Vultr name servers:

```
ns1.vultr.com  
ns2.vultr.com
```

Update Name Servers at Your Registrar

The process for updating your domain's name servers varies depending on your domain registrar. Most registrars offer an option to configure custom name servers. Below are links to documentation from popular registrars that explain how to update your domain's name servers:

- [Bluehost](#)
- [Domain.com](#)
- [DreamHost](#)
- [Dynadot](#)
- [Enom](#)
- [Gandi](#)
- [GoDaddy](#)
- [HostGator](#)
- [Hover](#)
- [IONOS](#)

- [Name.com](#)
- [Namecheap](#)
- [Network Solutions](#)
- [Porkbun](#)
- [Squarespace](#)

After updating your domain's name servers, it can take up to 48 hours for the changes to fully propagate across the internet. During this period, DNS servers around the world update their records to reflect the new name server configuration.

You can verify the update using any public [DNS lookup tool](#) that checks NS records.

Begin Managing DNS with Vultr

After you have updated your domain's name servers to point to Vultr, you can begin managing DNS records from the [Vultr DNS dashboard](#). You can create and manage **A**, **CNAME**, **MX**, **TXT**, and other record types to control how your domain functions.

FAQ

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

Contents

| | | |
|----|---|----|
| 01 | Introduction | 10 |
| 02 | How can I point my domain to Vultr DNS? | 79 |
| 03 | Can I create subdomains using Vultr DNS? | 79 |
| 04 | How long does the DNS propagation to Vultr DNS take? | 79 |
| 05 | Can I manage a domain using Vultr DNS without changing nameservers? | 80 |

Frequently Asked Questions (FAQs) About Vultr DNS

Introduction

These are the frequently asked questions for Vultr DNS.

How can I point my domain to Vultr DNS?

Login to your registrar and change your domain's nameservers to `ns1.vultr.com`, `ns2.vultr.com`.

Can I create subdomains using Vultr DNS?

Yes, you can create subdomains using Vultr DNS. Create a new **A** record with your desired subdomain name and enter your instance IP to apply the record.

How long does the DNS propagation to Vultr DNS take?

Propagation of new DNS records may take between 6 to 12 hours depending on your domain record changes.

Can I manage a domain using Vultr DNS without changing nameservers?

No, you cannot manage a domain using Vultr DNS without changing the nameservers.

Firewall

Firewall provides network security by controlling inbound and outbound traffic to your cloud instances through customizable rule sets.

Contents

| | | |
|----|--------------|-----|
| 01 | Provisioning | 83 |
| 02 | Management | 88 |
| | Delete | 90 |
| | Groups | 96 |
| | Link | 101 |
| | Rules | 106 |
| 03 | FAQ | 112 |

Provisioning

A process that prepares and configures a server or service for use after initial deployment.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Create a Vultr Firewall Group

Introduction

Vultr Firewall is a web-based service that filters network traffic to instances in your Vultr account using groups. A Vultr Firewall group consists of multiple IPv4 and IPv6 network rules that enable you to define specific ports and traffic sources to your instances.

Follow this guide to create a new Vultr Firewall group to manage network traffic filtering rules using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Firewall** from the list of options.
2. Click **Add Firewall** to set up a new firewall group.
3. Enter your firewall group label in the **Description** field
4. Click **Add Firewall Group** to apply the group and manage the network filtering rules.

Vultr API

1. Send a `GET` request to the [List Firewall Groups endpoint](#) and verify all active firewall groups in your Vultr account.

CONSOLE

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

2. Send a `POST` request to the **Create Firewall Group** endpoint to create a new Vultr Firewall group.

CONSOLE

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

Visit the [List Firewall Groups API page](#) to view additional attributes to apply on the firewall group.

Vultr CLI

1. List all Vultr Firewall groups in your account.

CONSOLE

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

2. Create a new Vultr Firewall group.

CONSOLE

```
$ vultr-cli firewall group create --description <label>
```

Run `vultr-cli firewall group create --help` to view additional options to apply on the firewall group.

Terraform

1. Ensure the [Vultr Terraform provider](#) is configured in your Terraform project.

2. Create a firewall group (and optionally a rule), then apply.

TERRAFORM

```
resource "vultr_firewall_group" "web" {
  description = "web-fw"
}

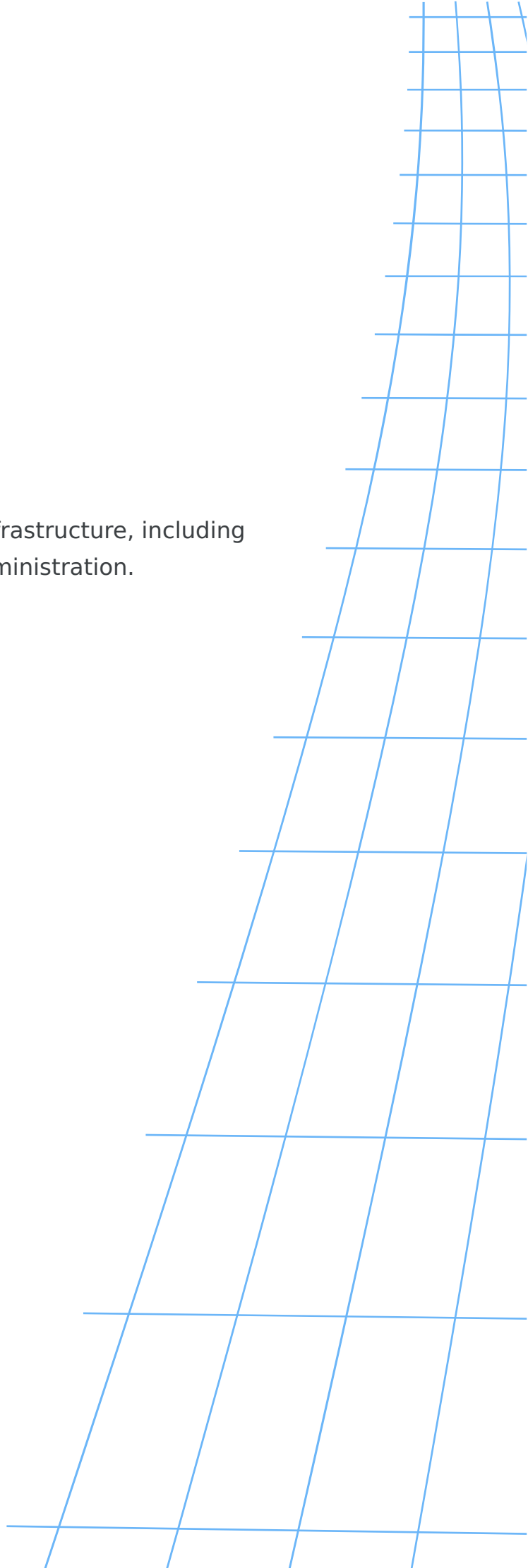
resource "vultr_firewall_rule" "allow_http" {
  firewall_group_id = vultr_firewall_group.web.id
  protocol          = "tcp"
  port              = "80"
  ip_type           = "v4"
  subnet            = "0.0.0.0"
  subnet_size       = 0
  notes             = "Allow HTTP"
}
```

3. Apply the configuration and observe the following output:

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

Management

Tools and features for managing your Vultr infrastructure, including access controls, monitoring, and resource administration.



Contents

| | | |
|----|--------|-----|
| 01 | Delete | 90 |
| 02 | Groups | 96 |
| 03 | Link | 101 |
| 04 | Rules | 106 |

Delete

Permanently removes the selected resource from your Vultr account.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Delete Vultr Firewall Groups and Rules

Introduction

Deleting a firewall group removes all existing rules and detaches all active instances from the group while deleting firewall rules removes traffic filtering on a specific port. Deleted firewall groups or rules cannot be recovered unless recreated using the Vultr Firewall.

Follow this guide to delete Vultr Firewall groups and rules using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Firewall** from the list of options.
2. Select your target firewall group to manage it.
3. Click **Delete Firewall Rule** within the action section of your target firewall rule to delete it.
4. Click **Delete Group** in the top right corner to delete the firewall group and all existing rules.
5. Click **Delete Firewall Group** in the confirmation prompt to remove the firewall group and unlink active instances.

Vultr API

1. Send a `GET` request to the [List Firewall Groups endpoint](#) and note the target firewall group's ID in your output.

CONSOLE

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

2. Send a `GET` request to the [List Firewall Rules endpoint](#) and note the target firewall rule ID in your output.

CONSOLE

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

3. Send a `DELETE` request to the [Delete Firewall Rule endpoint](#) to delete the firewall rule.

CONSOLE

```
$ curl "https://api.vultr.com/v2/firewalls/{firewall-group-id}/rules/{firewall-rule-id}" \  
-X DELETE \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

4. Send a `DELETE` request to the [Delete Firewall Group endpoint](#) to delete the firewall group.

CONSOLE

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

Vultr CLI

1. List all firewall groups in your Vultr account and note the target group ID.

CONSOLE

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

2. List all available firewall rules in the firewall group and note the target rule number.

CONSOLE

```
$ vultr-cli firewall rule list <firewall-group-id>
```

3. Delete the firewall rule.

CONSOLE

```
$ vultr-cli firewall rule delete <firewall-group-id>  
<firewall-rule-number>
```

4. Delete the firewall group.

CONSOLE

```
$ vultr-cli firewall group delete <firewall-group-id>
```

Terraform

1. Open your Terraform configuration where the firewall group and rules are defined.
2. Remove the `vultr_firewall_rule` blocks you want to delete, or destroy by target; remove the `vultr_firewall_group` block to delete the group.

TERRAFORM

```
resource "vultr_firewall_group" "web" {
  description = "web-fw"
}

resource "vultr_firewall_rule" "allow_http" {
  firewall_group_id = vultr_firewall_group.web.id
  protocol          = "tcp"
  port              = "80"
  ip_type           = "v4"
  subnet            = "0.0.0.0"
  subnet_size       = 0
  notes             = "Allow HTTP"
}

# To delete a specific rule, remove its block or run:
# terraform destroy -target vultr_firewall_rule.allow_http

# To delete the group (and its rules), remove the group block
or run:
# terraform destroy -target vultr_firewall_group.web
```

3. Apply the configuration and observe the following output:

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

Groups

A feature that allows you to organize and manage multiple resources collectively for easier administration and access control.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Manage Vultr Firewall Groups

Introduction

Vultr Firewall groups enable the creation and application of network filtering rules that apply to attached instances. A firewall group can consist of multiple rules that filter IPv4 and IPv6 network traffic when attached to an instance.

Follow this guide to manage Vultr Firewall groups using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Firewall** from the list of options.
2. Select your target firewall group to manage it.

Vultr API

1. Send a `GET` request to the [List Firewall Groups endpoint](#) to view all firewall groups in your Vultr account.

CONSOLE

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

Vultr CLI

1. List all firewall groups in your Vultr account.

CONSOLE

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

Terraform

1. Open your Terraform configuration for the existing Firewall groups.
2. Add or update a `vultr_firewall_group` with example rules, then apply.

TERRAFORM

```
resource "vultr_firewall_group" "default" {
  description = "default-fw"
}

resource "vultr_firewall_rule" "allow_http" {
  firewall_group_id = vultr_firewall_group.default.id
  protocol          = "tcp"
  port              = "80"
  ip_type           = "v4"
  subnet            = "0.0.0.0"
  subnet_size       = 0
  notes             = "Allow HTTP"
}

resource "vultr_firewall_rule" "allow_https" {
  firewall_group_id = vultr_firewall_group.default.id
  protocol          = "tcp"
  port              = "443"
  ip_type           = "v4"
  subnet            = "0.0.0.0"
  subnet_size       = 0
  notes             = "Allow HTTPS"
}
```

3. Apply the configuration and observe the following output:

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

Link

A feature that allows you to connect your Vultr account with third-party services for enhanced functionality and integrations.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |

How to Link a Vultr Firewall Group to an Instance

Introduction

Linking a Vultr Firewall group to an instance enables the group's filtering rules to the main network interface. A Vultr Firewall group consists of multiple rules that define the flow of traffic and enable filtering of specific requests when linked to an instance.

Follow this guide to link a Vultr Firewall group to an instance using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Firewall** from the list of options.
2. Select your target firewall group to manage it.
3. Click **Linked Instances** on the left navigation menu.
4. Click the **Server** drop-down and select your target instance from the list and click **Add** to link the firewall group to the instance.

Vultr API

1. Send a `GET` request to the [List Firewall Groups endpoint](#) and note the target firewall group's ID in your output.

```
CONSOLE
```

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

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

CONSOLE

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

3. Send a `PATCH` request to the [Update Instance endpoint](#) with a new `firewall_group_id` value to link the instance to the firewall group.

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" : "<target-firewall-group>"  
}'
```

Vultr CLI

1. List all firewall groups in your Vultr account and note the target firewall group ID..

CONSOLE

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

2. List all instances in your Vultr account and note the target instance ID.

CONSOLE

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

3. Link the firewall group to the instance.

CONSOLE

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

Rules

Define network security policies that control inbound and outbound traffic to your Vultr resources.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Create Vultr Firewall Rules

Introduction

Vultr Firewall rules enable traffic filtering using port numbers and source IP addresses for incoming network requests. A Vultr Firewall group contains multiple rules that define specific the flow of network traffic to attached instances.

Follow this guide to create Vultr Firewall rules using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Firewall** from the list of options.
2. Select your target firewall group to manage it.
3. Click your target network type on the left navigation menu to modify the incoming traffic rules.
4. Click the **Protocol** drop-down to select a common network application profile or choose **Custom** from the list and enter your target network port in the **Port (or range)** field.
5. Click the **Source** drop-down, select your traffic source and enter the target source IP address.
6. Click **Add note** and enter a descriptive label to identify the new firewall rule.
7. Click **Add Firewall Rule** within the **Action** section to apply the new rule to your firewall group.

Vultr API

1. Send a `GET` request to the [List Firewall Groups endpoint](#) and note the target firewall group ID in your output.

CONSOLE

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

2. Send a `GET` request to the [List Firewall Rules endpoint](#) to view all active rules in the firewall group.

CONSOLE

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

3. Send a `POST` request to the [Create Firewall Rules endpoint](#) to create a new rule in the firewall group.

CONSOLE

```
$ curl "https://api.vultr.com/v2/firewalls/{firewall-group-id}/rules" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "ip_type" : "<network-type>",  
  "protocol" : "<protocol>",  
  "port" : "<target-instance-port>",  
  "source" : "<source-address>",  
  "notes" : "<label>"  
'
```

Visit the [List Firewall Rules API page](#) to view additional attributes to apply on the firewall rule.

Vultr CLI

1. List all firewall groups in your Vultr account and note the target firewall group ID.

CONSOLE

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

2. List all rules in the firewall group.

CONSOLE

```
$ vultr-cli firewall rule list <firewall-group-id>
```

3. Create a new firewall rule.

CONSOLE

```
$ vultr-cli firewall rule create --id=<firewall-group-id> --  
ip-type=<network-type> --protocol=<protocol> --  
source=<source-address> --port=<target-instance-port>
```

Run `vultr-cli firewall rule create --help` to view additional options to apply on the firewall rule.

Terraform

1. Open your Terraform configuration for the existing Firewall group.
2. Add a `vultr_firewall_rule` for that group, then apply.

TERRAFORM

```
resource "vultr_firewall_rule" "allow_ssh" {
  firewall_group_id = var.firewall_group_id
  protocol          = "tcp"
  port              = "22"
  ip_type           = "v4"
  subnet            = "0.0.0.0"
  subnet_size       = 0
  notes             = "Allow SSH"
}
```

3. Apply the configuration and observe the following output:

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

FAQ

Frequently asked questions and answers about Vultr services, features, and common issues.

Contents

| | | |
|----|--|-----|
| 01 | Introduction | 10 |
| 02 | Do Vultr firewall rules override my software-based firewall rules? | 114 |
| 03 | Can I create service based rules using Vultr firewall? | 114 |
| 04 | How many instances can I attach to a Vultr firewall group? | 114 |
| 05 | Does Vultr firewall work filter outgoing traffic from my instance? | 115 |

Frequently Asked Questions (FAQs) About Vultr Firewall

Introduction

These are the frequently asked questions for Vultr Firewall.

Do Vultr firewall rules override my software-based firewall rules?

If the Vultr Firewall allows HTTPS traffic but the instance's internal firewall blocks HTTPS, the traffic will still be blocked, because the instance-level firewall does not permit it. Conversely, if the Vultr Firewall blocks HTTPS traffic but the instance's internal firewall allows HTTPS, the traffic will still be blocked, because the Vultr Firewall prevents it from ever reaching the instance.

Can I create service based rules using Vultr firewall?

Yes, you can create Vultr Firewall Rules using specific service profiles. Click the **Protocol** drop-down in your firewall group to select a specific service.

How many instances can I attach to a Vultr firewall group?

You can attach multiple instances to a Vultr Firewall group.

Does Vultr firewall work filter outgoing traffic from my instance?

No, Vultr Firewall does not filter outgoing network traffic from your instance.

Reserved IPs

Dedicated static IP addresses that remain assigned to your account even when not attached to a server, providing flexibility for high-availability setups and seamless server migrations.

Contents

| | | |
|----|---------------------|-----|
| 01 | Provisioning | 118 |
| 02 | Management | 123 |
| | Attachment | 125 |
| | Convert Existing IP | 130 |
| | Delete | 134 |
| | Detachment | 139 |
| 03 | FAQ | 144 |

Provisioning

The process of setting up and configuring a new server or service to make it ready for use.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Create Vultr Reserved IPs

Introduction

Reserved IPs enable dedicated IP Addresses that are isolated from the public pool of Vultr's public IP addresses for attachment to your instances. Reserved IPs enable you to reserve a specific public IP address in a single Vultr location you can attach and use with your instances.

Follow this guide to create reserved IPs in your Vultr account using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Reserved IPs** from the list of options.
2. Click **Add Reserved IP** to create a new reserved IP address.
3. Click the **Location** drop-down and select your target Vultr location.
4. Click the **Type** drop-down and select the reserved IP address type.
5. Enter a new descriptive label in the **Label** field and click **Add** to create the reserved IP.

Vultr API

1. Send a `GET` request to the [List Reserved IPs endpoint](#) to view all reserved IPs active in your Vultr account.

CONSOLE

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

2. Send a `POST` request to the [Create Reserved IP endpoint](#) to create a new reserved IP address in a specific Vultr location.

CONSOLE

```
$ curl "https://api.vultr.com/v2/reserved-ips" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "region" : "<location>",  
  "ip_type" : "<address_type>",  
  "label" : "<label>"  
}'
```

Visit the [Create Reserved IP API page](#) to view additional attributes to apply on the reserved IP.

Vultr CLI

1. List all active reserved IPs in your Vultr account.

CONSOLE

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

2. Create a new reserved IP in a specific Vultr location.

CONSOLE

```
$ vultr-cli reserved-ip create --label <label> --region  
<location> --type <address_type>
```

Run `vultr-cli reserved-ip create --help` to view additional options to apply on the reserved IP.

Terraform

1. Ensure the [Vultr Terraform provider](#) is configured in your Terraform project.
2. Create a reserved IP (optionally attach to an instance), then apply.

TERRAFORM

```
resource "vultr_reserved_ip" "rip" {
  region = "ewr"
  ip_type = "v4" # v4 | v6
  label = "web-rip"
  # instance_id = vultr_instance.server.id
}
```

3. Apply the configuration and observe the following output:

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

Management

Tools and features for managing your Vultr infrastructure, including access controls, monitoring, and account administration.

Contents

| | |
|------------------------|-----|
| 01 Attachment | 125 |
| 02 Convert Existing IP | 130 |
| 03 Delete | 134 |
| 04 Detachment | 139 |

Attachment

A storage volume that can be mounted to a Vultr instance to provide additional disk space.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Attach Reserved IPs to a Vultr Instance

Introduction

Attaching a reserved IP to an instance enables a new public network address on the instance's main network interface. You can attach multiple reserved IPs on an instance to enable fast resolution and system management.

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

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Reserved IPs** from the list of options.
2. Click your target reserved IP to open its management page.
3. Click the **Attach to Server** drop-down, select your target instance and click **Attach**.
4. Click **Attach Reserved IP** in the confirmation prompt to apply the reserved IP to your instance.

Vultr API

1. 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}"
```

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

CONSOLE

```
$ curl "https://api.vultr.com/v2/instances" \  
-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" : "<target-instance-id>"  
}'
```

Vultr CLI

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

CONSOLE

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

2. List all instances in your Vultr account and note the target instance 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="<target-instance-id>"
```

Terraform

1. Open your Terraform configuration for the existing instance and reserved IP.
2. Set `instance_id` on the `vultr_reserved_ip` to attach, then apply.

TERRAFORM

```
resource "vultr_instance" "server" {  
  # ...existing fields  
}  
  
resource "vultr_reserved_ip" "rip" {  
  region      = "ewr"  
  ip_type     = "v4"  
  label       = "web-rip"  
  instance_id = vultr_instance.server.id  
}
```

3. Apply the configuration and observe:

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

Convert Existing IP

Learn how to convert an existing IP address on your Vultr instance to a different type of IP address.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |

How to Convert an Existing Vultr Instance IP Address to a Reserved IP

Introduction

Converting an existing instance IP address to a reserved IP enables it as a dedicated address you can attach or detach to other instances in your Vultr account. A reserved IP is compatible with any instance you can attach or detach it to.

Follow this guide to convert an existing Vultr instance IP Address to a reserved IP using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Reserved IPs** from the list of options.
2. Click **Add Reserved IP** to set up a new reserved IP address.
3. Click the **IPv4 Address** or **IPv6 subnet** drop-down to select your existing IP and click **Convert**.
4. Click **Convert IP Address** in the confirmation prompt to create a new reserved IP using the existing IP.

Vultr API

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

CONSOLE

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

2. Send a `POST` request to the [Convert Existing IP endpoint](#) to create a new reserved IP using the instance's IP address.

CONSOLE

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

Vultr CLI

1. List all instances in your Vultr account and note the target instance's IP.

CONSOLE

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

2. Convert the target instance IP to a reserved IP address.

CONSOLE

```
$ vultr-cli reserved-ip convert --ip="<instance-ip>" --  
label="<label>"
```

Delete

Permanently removes a resource from your Vultr account.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Delete a Reserved IP

Introduction

Deleting a reserved IP detaches it from any linked instances and removes it from your Vultr account. You can re-create a reserved IP after deletion if it's available in the specific Vultr location.

Follow this guide to delete a reserved IP using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **Reserved IPs** from the list of options.
2. Click your target reserved IP to open its management page.
3. Click **Remove Reserved IP** in the top-right corner.
4. Click **Remove Reserved IP** in the confirmation prompt to delete the reserved IP.

Vultr API

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

CONSOLE

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

2. Send a `DELETE` request to the [Delete Reserved IP endpoint](#) to delete the reserved IP.

CONSOLE

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

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. Delete the reserved IP.

CONSOLE

```
$ vultr-cli reserved-ip delete <reserved-ip-id>
```

Terraform

1. Open your Terraform configuration where the Reserved IP is defined.
2. Remove the `vultr_reserved_ip` resource block, or destroy it by target.

TERRAFORM

```
resource "vultr_reserved_ip" "rip" {
  region = "ewr"
  ip_type = "v4"
  label = "web-rip"
```

```
}  
  
# To delete, either remove this block from configuration  
# or run: terraform destroy -target vultr_reserved_ip.rip
```

3. Apply the configuration and observe the following output:

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

Detachment

A guide explaining how to remove a Reserved IP address from a Vultr instance while keeping the IP in your account for future use.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Detach Reserved IPs from a Vultr Instance

Introduction

Detaching a Reserved IP from an instance removes the associated public network address from the instance's main network interface. You can detach a Reserved IP to reassign it to another instance, release unused resources, or adjust your network configuration.

Follow this guide to detach Reserved IPs from a Vultr instance using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down, and select **Reserved IPs** from the list of options.
2. Click your target Reserved IP to open its management page.
3. In the **Attached to Server** section, click **Detach** to begin removing the Reserved IP from the instance.
4. Click **Detach Reserved IP** in the confirmation prompt to finalize the detachment.

Vultr API

1. 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}"
```

2. Send a `POST` request to the [Detach Reserved IP endpoint](#) to detach the Reserved IP from the instance.

CONSOLE

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

Vultr CLI

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

CONSOLE

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

2. Detach the Reserved IP from the instance.

CONSOLE

```
$ vultr-cli reserved-ip detach <reserved-ip-id>
```

Terraform

1. Open your Terraform configuration for the existing Reserved IP.
2. Remove `instance_id` from the `vultr_reserved_ip` to detach, then apply.

TERRAFORM

```
resource "vultr_reserved_ip" "rip" {  
  region = "ewr"  
  ip_type = "v4"  
  label   = "web-rip"  
  # instance_id removed to detach  
}
```

3. Apply the configuration and observe:

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

FAQ

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

Contents

| | | |
|----|--|-----|
| 01 | Introduction | 10 |
| 02 | Can I attach a reserved IP to multiple instances at the same time? | 146 |
| 03 | Can I select a specific IP address to reserve without converting an existing IP address? | 146 |
| 04 | Do reserved IPS expire? | 146 |
| 05 | Are reserved IPS free? | 146 |

Frequently Asked Questions (FAQs) About Vultr Reserved IPs

Introduction

These are the frequently asked questions for Reserved IPs.

Can I attach a reserved IP to multiple instances at the same time?

No, you can only attach a reserved IP to one instance at a time.

Can I select a specific IP address to reserve without converting an existing IP address?

No, you cannot select a specific public IP address to reserve without converting your existing instance's IP address to a reserved IP.

Do reserved IPS expire?

No, reserved IPs do not expire and remain available for use unless deleted from your Vultr account.

Are reserved IPS free?

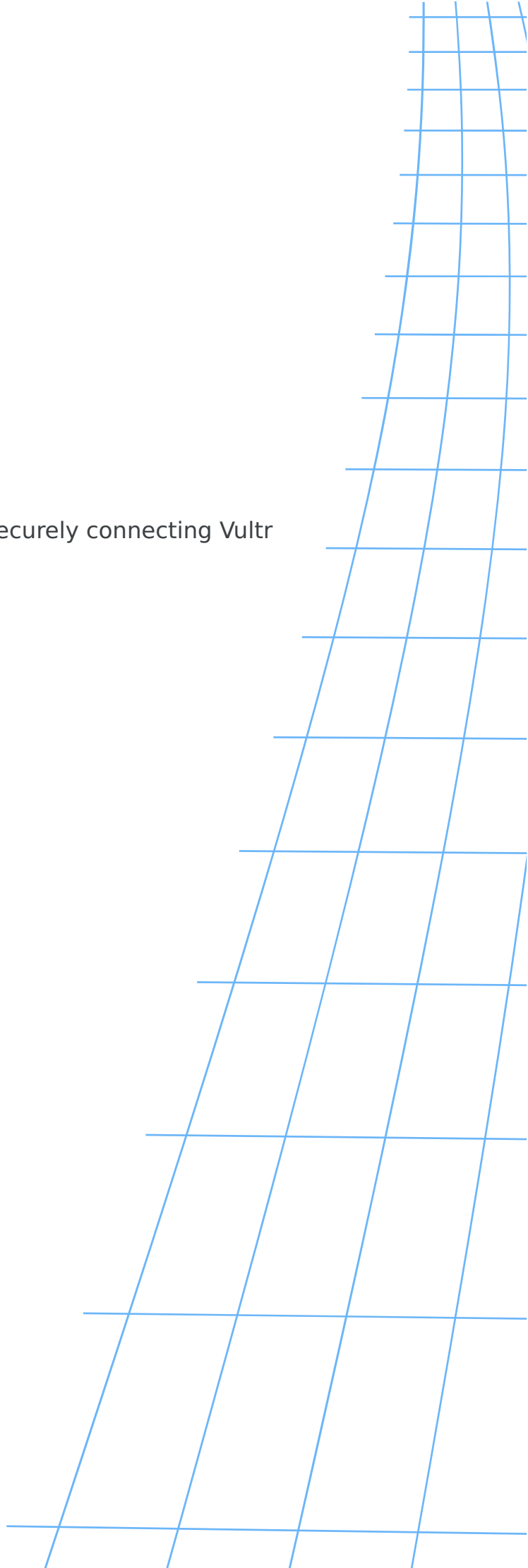
No, reserved IPs are not free and cost \$3 per month.



[Vultr Docs](#) > [Product Documentation](#)

VPC 2.0

A private, isolated network environment for securely connecting Vultr resources within the same region.



Contents

| | | |
|----|--------------|-----|
| 01 | Provisioning | 149 |
| 02 | Management | 154 |
| | Attachment | 156 |
| | Delete | 161 |
| | Monitor | 166 |
| 03 | FAQ | 171 |

Provisioning

A process that prepares and configures a server or service for use after initial deployment.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Create a VPC 2.0 Network

Introduction

A Virtual Private Cloud (VPC) 2.0 network creates a secure, enhanced, and isolated private networking environment for instances to connect and share resources. VPC 2.0 networks support IPv4 network addresses to enable communication between instances.

Follow this guide to create a Virtual Private Network (VPC) 2.0 network using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC 2.0** from the list of options.
2. Click **Add VPC 2.0 Network** to set up a new VPC.
3. Select your target Vultr location to create the VPC network.
4. Click **Configure IPv4 Range** to enter a custom private IPv4 subnet to assign the network, or keep **Auto-Assign IP Range** selected.
5. Enter a new label in the **VPC 2.0 Network Description** field to identify the network.
6. Click **Add Network** to create the new VPC network and verify that it's available in your Vultr account.

Vultr API

1. Send `POST` request to the [Create a VPC 2.0 Network endpoint](#) to create a new VPC 2.0 network in a specific Vultr location with an automatic IP address range.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpc2" \  
  -X POST \  
  -H "Authorization: Bearer ${VULTR_API_KEY}" \  
  -H "Content-Type: application/json" \  
  --data '{  
    "region" : "<vultr-location-id>",  
    "description" : "<label>"  
  }'
```

Visit the [Create a VPC 2.0 Network API page](#) to view additional attributes to apply on the VPC 2.0 network.

Vultr CLI

1. Create a new VPC 2.0 network in a specific Vultr location with an automatic IP address range.

CONSOLE

```
$ vultr-cli vpc2 create --region="ewr" --  
description="example-vpc" --ip-type="v4"
```

Run the `vultr-cli vpc2 create --help` command to view additional options to enable on the VPC 2.0 network.

Terraform

1. Ensure the [Vultr Terraform provider](#) is configured in your Terraform project.
2. Create a VPC 2.0 network, then apply.

TERRAFORM

```
resource "vultr_vpc2" "net" {  
  region      = "ewr"
```

```
description = "example-vpc2"  
# optional: auto IP range; or specify:  
# v4_subnet      = "10.0.0.0"  
# v4_subnet_mask = 24  
}
```

3. Apply the configuration and observe the following output:

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

Management

Tools and resources for managing your Vultr infrastructure, including account settings, billing, and server administration.

Contents

| | |
|---------------|-----|
| 01 Attachment | 156 |
| 02 Delete | 161 |
| 03 Monitor | 166 |

Attachment

A feature that allows you to connect additional resources like block storage volumes to your Vultr instances for expanded capabilities.

Contents

| | | |
|----|--------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr API | 26 |
| 03 | Vultr CLI | 46 |
| 04 | Terraform | 47 |

How to Attach a VPC 2.0 Network to a Vultr Instance

Introduction

Attaching a VPC 2.0 network to an instance enables a new private networking interface that supports communication with other nodes on the same network. You can attach multiple VPC 2.0 networks to an instance to enable connections with other hosts on each subnet.

Follow this guide to attach a VPC 2.0 network to a Vultr instance using the Vultr API, CLI, or Terraform.

Vultr API

1. Send a `GET` request to the [List VPC 2.0 networks endpoint](#) and note the target VPC 2.0 network's ID in your output.

CONSOLE

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

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

CONSOLE

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

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

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

Vultr CLI

1. View the list of VPC 2.0 networks in your Vultr account and note the target network's ID.

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

2. View the all instances and note your target instance's ID.

```
CONSOLE
$ vultr-cli instance 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 instance.
2. Add the VPC 2.0 network ID to the instance `vpc2_ids` list, then apply.

TERRAFORM

```
resource "vultr_instance" "server" {  
  # ...existing fields (region, plan, os_id, label)  
  vpc2_ids = [var.vpc2_id]  
}
```

3. Apply the configuration and observe the following output:

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

Delete

Permanently removes the selected resource from your Vultr account.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Delete a VPC 2.0 Network

Introduction

Deleting a Virtual Private Cloud (VPC) 2.0 network removes all existing routes and deactivates the associated interfaces on any attached instances.

Follow this guide to delete a VPC 2.0 network using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC 2.0** from the list of options.
2. Select your target VPC 2.0 network to open its management page.
3. Click **Delete VPC 2.0 Network**.
4. Click **Delete VPC 2.0 Network** in the confirmation prompt to destroy the VPC network.

Vultr API

1. Send a `GET` request to the [List VPC 2.0 networks endpoint](#) and note the target VPC 2.0 network's ID in your output.

CONSOLE

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

2. Send a `DELETE` request to the [Delete a VPC 2.0 network endpoint](#) to destroy the VPC 2.0 network.

CONSOLE

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

Vultr CLI

1. List all VPC 2.0 networks in your Vultr account and note the target VPC 2.0 network's ID.

CONSOLE

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

2. Delete the target VPC 2.0 network.

CONSOLE

```
$ vultr-cli vpc2 delete <VPC2 ID>
```

Terraform

1. Open your Terraform configuration where the VPC 2.0 network is defined.
2. Remove the `vultr_vpc2` resource block, or destroy it by target.

TERRAFORM

```
resource "vultr_vpc2" "net" {  
  region      = "ewr"  
  description = "example-vpc2"  
}  
  
# To delete, either remove this block from configuration  
# or run: terraform destroy -target vultr_vpc2.net
```

3. Apply the configuration and observe the following output:

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

Monitor

A system for tracking server performance metrics, resource usage, and uptime to ensure optimal operation of your Vultr infrastructure.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |

How to Monitor a VPC 2.0 Network

Introduction

Monitoring a Virtual Private Cloud (VPC) 2.0 network enables the management and verification of the network information. You can view attached instances and active route information by monitoring a VPC 2.0 network.

Follow this guide to monitor VPC 2.0 networks using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC 2.0** from the list of options.
2. Select your target VPC 2.0 network to open its management page.
3. Monitor the attached instances to your VPC 2.0 network within the **Attached Nodes**.

Vultr API

1. Send a `GET` request to the [List VPC 2.0 networks endpoint](#) to view 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}"
```

2. Send a `GET` request to the [Get a VPC 2.0 Network endpoint](#) to view information about the target VPC 2.0 network.

CONSOLE

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

3. Send a `GET` request to the [Get a list of nodes attached to a VPC 2.0 network endpoint](#) to view all instances attached to the VPC 2.0 network.

CONSOLE

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

Vultr CLI

1. List all VPC 2.0 networks in your Vultr account and note the target VPC 2.0 network's ID.

CONSOLE

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

2. Get information about the VPC 2.0 network.

CONSOLE

```
$ vultr-cli vpc2 get <VPC2 ID>
```

3. View all nodes attached to the VPC 2.0 network.

CONSOLE

```
$ vultr-cli vpc2 nodes list <VPC2 ID>
```

FAQ

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

Contents

| | | |
|----|---|-----|
| 01 | Introduction | 10 |
| 02 | Can I specify a private network address range for my VPC 2.0 network? | 173 |
| 03 | Does VPC 2.0 support public network gateway services? | 173 |
| 04 | Can I connect my VPC 2.0 network to my On-Premises network? | 173 |
| 05 | How many VPC 2.0 networks am I limited to? | 174 |

Frequently Asked Questions (FAQs) About Vultr VPC 2.0

Introduction

These are the frequently asked questions for VPC 2.0 networks.

Can I specify a private network address range for my VPC 2.0 network?

Yes, you can specify a custom network address range to use in a VPC 2.0 network. Use the **RFC1918** private range for IPv4 networks `10.0.0.0/8`, `172.16.0.0/12`, or `192.168.0.0/16` to assign to your network.

Does VPC 2.0 support public network gateway services?

No, VPC 2.0 does not support direct gateway services to public networks. Create a dedicated network gateway instance in your VPC 2.0 network to perform routing and gateway services.

Can I connect my VPC 2.0 network to my On-Premises network?

Yes, you can connect your VPC 2.0 network to your on-premises network by creating a VPN connection to any instance attached to the network and apply the correct routing information to enable data transfer.

How many VPC 2.0 networks am I limited to?

You are limited to **5** VPC 2.0 networks per Vultr location.

VPC

A private, isolated network that allows secure communication between your Vultr resources without traversing the public internet. This network also supports deployment of VPC-only instances without public IPv4 addresses. See the [VPC-only Instances documentation](#) for details.

Contents

| | | |
|-----------|-----------------------|------------|
| 01 | Provisioning | 177 |
| 02 | Management | 182 |
| | Delete | 184 |
| | Monitor | 189 |
| 03 | FAQ | 193 |
| 04 | NAT Gateway | 197 |
| | Provisioning | 199 |
| | Management | 204 |
| | Delete | 206 |
| | Get Info | 210 |
| | Configuration | 215 |
| | Firewall Rules | 217 |
| | Create | 219 |
| | Delete | 225 |
| | List | 230 |
| | Update | 235 |
| | Read | 240 |
| | Port Forwarding Rules | 245 |
| | Create | 247 |
| | Delete | 252 |
| | List | 257 |
| | Read | 261 |
| | Update | 266 |

Provisioning

A process that prepares and configures a server or service for use after initial deployment.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Create a VPC Network

Introduction

A Vultr Virtual Private Cloud (VPC) network creates a secure and isolated private networking environment for instances to connect and share resources. A VPC network supports IPv4 network addresses to enable communication and data sharing between instances.

Follow this guide to create a Virtual Private Network (VPC) network using the Vultr Customer Portal, API, CLI, or Terraform

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks** from the list of options.
2. Click **Add VPC Network** to set up a new VPC.
3. Select your target Vultr location to create the VPC network.
4. Click **Configure IPv4 Range** to enter a custom private IPv4 subnet to assign the network, or keep **Auto-Assign IP Range** selected.
5. Click **Custom Routes** within the **Manage Routes** section to set up custom static routes to update on all instances attached to the VPC network, or keep **No Routes** selected.
6. Enter a new label in the **VPC Network Name** field to identify the network.
7. Click **Add Network** to create the new VPC network and verify that it's available in your Vultr account

Vultr API

1. Send `POST` request to the [Create VPC endpoint](#) to create a new VPC network in a specific Vultr location with an automatic IP address range.

CONSOLE

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

Visit the [Create VPC API page](#) to view additional attributes to apply on the VPC network.

Vultr CLI

1. Create a new VPC network in a specific Vultr location with an automatic IP address range.

CONSOLE

```
$ vultr-cli vpc create --region="ewr" --description="test  
VPC"
```

Run `vultr-cli vpc create --help` to view additional options to apply on the VPC network.

Terraform

1. Ensure the [Vultr Terraform provider](#) is configured in your Terraform project.
2. Create a VPC network, then apply.

TERRAFORM

```
resource "vultr_vpc" "net" {  
  region      = "ewr"
```

```
description = "example-vpc"  
# optional manual range  
# v4_subnet      = "10.10.0.0"  
# v4_subnet_mask = 24  
}
```

3. Apply the configuration and observe the following output:

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

Management

Tools and features for managing your Vultr infrastructure, including account settings, billing, and resource administration.

Contents

| | |
|------------|-----|
| 01 Delete | 184 |
| 02 Monitor | 189 |

Delete

Permanently removes the selected resource from your Vultr account.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |
| 05 | Terraform | 47 |

How to Delete a VPC Network from a Vultr Account

Introduction

Deleting a Virtual Private Cloud (VPC) network removes routing information and deactivates the associated interfaces on any attached instances.

Follow this guide to delete a VPC network using the Vultr Customer Portal, API, CLI, or Terraform.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks** from the list of options.
2. Select your target VPC network to open its management page.
3. Click **Delete VPC Network**.
4. Click **Delete VPC Network** in the confirmation prompt to destroy the VPC network.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) and note the target VPC network's ID in your output.

CONSOLE

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

2. Send a `DELETE` request to the [Delete a VPC endpoint](#) to destroy the target VPC network.

CONSOLE

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

Vultr CLI

1. List all VPC networks in your Vultr account and note the target VPC network's ID.

CONSOLE

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

2. Delete the VPC network.

CONSOLE

```
$ vultr-cli vpc delete <vpc-id>
```

Terraform

1. Open your Terraform configuration where the VPC is defined.
2. Remove the `vultr_vpc` resource block, or destroy it by target.

TERRAFORM

```
resource "vultr_vpc" "net" {  
  region      = "ewr"  
  description = "example-vpc"  
}
```

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

3. Apply the configuration and observe the following output:

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

Monitor

A system that tracks server performance metrics and sends alerts when issues are detected.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |
| 04 | Vultr CLI | 46 |

How to Monitor a VPC Network

Introduction

Monitoring a Virtual Private Cloud (VPC) network enables the management and verification of the network information. You can view attached instances and subnet information by monitoring a VPC network.

Follow this guide to monitor VPC networks using the Vultr Customer Portal, API, or CLI.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks** from the list of options.
2. Select your target VPC network to open its management page.
3. Click **Add Routes** to create new static routes in your network, or click **Edit Routes** to modify the existing route information.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to view all VPC networks in your Vultr account and note the target VPC network ID.

CONSOLE

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

2. Send a `GET` request to the [Get a VPC endpoint](#) to view information about the target VPC network.

CONSOLE

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

Vultr CLI

1. List all VPC networks in your Vultr account and note the target VPC network's ID.

CONSOLE

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

2. Get information about the VPC network.

CONSOLE

```
$ vultr-cli vpc get <vpc-id>
```

FAQ

A comprehensive resource addressing common questions about Vultr's Virtual Private Cloud networking solution.

Contents

| | | |
|----|--|-----|
| 01 | Introduction | 10 |
| 02 | Can I specify a custom network address range for my VPC network? | 195 |
| 03 | Does VPC support public network gateway services? | 195 |
| 04 | Can I connect my VPC network to my On-Premises network? | 195 |
| 05 | How many VPC networks am I limited to? | 196 |
| 06 | Can I upload static routing information to a VPC network? | 196 |
| 07 | Does VPC support Multicast? | 196 |

Frequently Asked Questions (FAQ) About Vultr VPC

Introduction

These are the frequently asked questions for VPC networks.

Can I specify a custom network address range for my VPC network?

Yes, you can specify a custom network address range to use in a VPC network. Use the **RFC1918** private range for IPv4 networks `10.0.0.0/8`, `172.16.0.0/12`, or `192.168.0.0/16` to assign to your network.

Does VPC support public network gateway services?

No, VPC does not support direct gateway services to public networks. Create a dedicated network gateway instance in your VPC network to perform routing and gateway services.

Can I connect my VPC network to my On-Premises network?

Yes, you can connect your VPC network to your on-premises network by creating a VPN connection to any instance attached to the network and apply the correct routing information to enable data transfer.

How many VPC networks am I limited to?

You are limited to **5** VPC networks per Vultr location.

Can I upload static routing information to a VPC network?

No. Static routes cannot be uploaded or configured for a VPC network.

Does VPC support Multicast?

Yes, VPC supports Multicast. This enables efficient one-to-many communication within your virtual network infrastructure, allowing a single data stream to be sent to multiple recipients simultaneously.

NAT Gateway

Vultr NAT Gateway provides secure outbound internet access for private VPC instances.

Contents

| | | |
|-----------|-----------------------|------------|
| 01 | Provisioning | 199 |
| 02 | Management | 204 |
| | Delete | 206 |
| | Get Info | 210 |
| 03 | Configuration | 215 |
| | Firewall Rules | 217 |
| | Create | 219 |
| | Delete | 225 |
| | List | 230 |
| | Update | 235 |
| | Read | 240 |
| | Port Forwarding Rules | 245 |
| | Create | 247 |
| | Delete | 252 |
| | List | 257 |
| | Read | 261 |
| | Update | 266 |

Provisioning

Learn how to provision a Vultr NAT Gateway for secure, private internet access.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Provision a NAT Gateway Subscription

Introduction

Vultr NAT Gateway provides centralized, secure internet egress for compute instances on a VPC that do not have public IP addresses. The service supports controlled inbound access via port forwarding, with all traffic governed by NAT Gateway firewall rules. Typical use cases include isolating workloads from the public internet while allowing package updates, conserving IPv4 addresses, and simplifying perimeter security by managing rules at the gateway.

Follow this guide to provision a NAT Gateway subscription using the Vultr Customer Portal or API.

Vultr Customer Portal

NAT Gateway provisions automatically when you create a VPC Network with **Private Instances behind NAT Gateway** connectivity. This approach creates both the VPC and NAT Gateway subscription simultaneously.

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Click **Add VPC Network**.
3. Enter a new label in the **VPC Network Name** field to identify the network.
4. In the **VPC Connectivity** section, select **Private Instances behind NAT Gateway**.

 Note

This option automatically provisions a NAT Gateway subscription for the VPC.

5. Select your target Vultr location to deploy the VPC Network and NAT Gateway.
6. In the **Network Settings** section, enable **Custom IP Range** to enter a custom private IPv4 subnet, or keep it disabled to automatically assign an IP range.
7. Review the **Summary** panel to verify the configuration and monthly cost.
8. Click **Create VPC Network**.

Vultr API

1. Create a VPC in your target region and note the VPC ID. Replace `REGION_ID` with your Vultr region identifier and `LABEL` with a descriptive name. See the [VPC Provisioning](#) guide for details and additional options.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs" \  
  -X POST \  
  -H "Authorization: Bearer ${VULTR_API_KEY}" \  
  -H "Content-Type: application/json" \  
  --data '{  
    "region": "REGION_ID",  
    "description": "LABEL"  
  }'
```

The output displays the VPC details. Note the `id` field value.

2. Send a `POST` request to the **Create NAT Gateway subscription** endpoint for the VPC. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

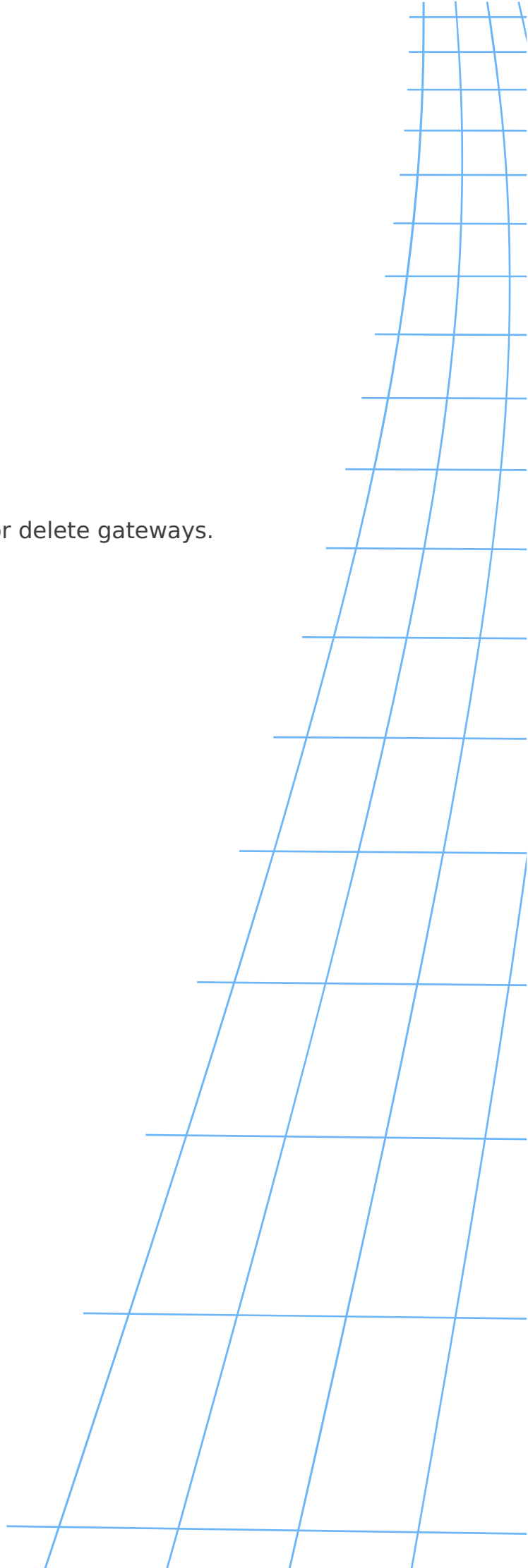
```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway" \  
  -X POST \  
  --data '{  
    "subscription": "nat-gateway"  
  }'
```

```
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{}'
```

The output displays the NAT Gateway subscription details. The gateway initially shows a `pending` status while provisioning completes. The system automatically assigns public IPv4, IPv6, and private IP addresses.

Management

Manage Vultr NAT Gateways to view details, or delete gateways.



Contents

| | |
|-------------|-----|
| 01 Delete | 206 |
| 02 Get Info | 210 |

Delete

Learn how to delete a NAT Gateway subscription from a Vultr VPC via API.

Contents

| | |
|-----------------|----|
| 01 Introduction | 10 |
|-----------------|----|

How to Delete a NAT Gateway Subscription

Introduction

Remove a NAT Gateway subscription from a VPC when it is no longer required. Deletion immediately decommissions the gateway and stops billing for that subscription. Ensure no workloads depend on its egress or port forwarding rules before proceeding.

Follow this guide to delete a NAT Gateway subscription using the Vultr API.

1. Send a `GET` request to the [List VPCs endpoint](#) and note the target VPC ID.

CONSOLE

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

2. Send a `GET` request to the **Get NAT Gateway subscription** endpoint specifying the VPC ID, and note the NAT Gateway ID from the output.

CONSOLE

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

3. Send a `DELETE` request to the **Delete NAT Gateway subscription** endpoint specifying the VPC ID and NAT Gateway ID.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/{vpc-id}/nat-gateway/{nat-gateway-id}" \  
-X DELETE \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

Get Info

Learn how to get NAT Gateway subscription information from the Vultr portal or API.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Get a NAT Gateway Subscription

Introduction

Retrieving a NAT Gateway subscription displays its current status, configuration, and assigned IP addresses. Review these details to verify the gateway's operational state, obtain identifiers for managing rules, or audit billing information.

Follow this guide to get a NAT Gateway subscription using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.

The VPC details page displays NAT Gateway information in the following sections:

Details Section

- **VPC Address:** Shows the NAT Gateway's private IP address within the VPC subnet.
- **Gateway IPv4 Address:** Displays the public IPv4 address used for outbound internet traffic.
- **Gateway IPv6 Address:** Shows the public IPv6 address assigned to the gateway.
- **UUID:** The VPC's unique identifier.

Attached Nodes Section

The NAT Gateway appears as a managed node with the following information:

- **Name / UUID:** Displays the NAT Gateway label and its unique identifier.
- **Status:** Displays the operational state (Active or Pending).
- **VPC Address:** The gateway's private IP address for internal VPC communication.
- **MAC Address:** The network interface MAC address.

Note

The NAT Gateway cannot be clicked or expanded. View and manage firewall rules and port forwarding configurations in the NAT Firewall and NAT Port Forwarding sections below the Attached Nodes.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `GET` request to the **Get NAT Gateway subscription** endpoint. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

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

The output displays the complete NAT Gateway subscription details:

- `id`: The NAT Gateway's unique identifier.
- `vpc_id`: The VPC's unique identifier.
- `date_created`: The gateway creation timestamp.
- `status`: The operational state (`active` or `pending`).
- `label`: The gateway's descriptive label.
- `tag`: Optional tag for organization.
- `public_ips`: Array containing one public IPv4 address for outbound internet traffic.
- `public_ips_v6`: Array containing one public IPv6 address.
- `private_ips`: Array containing one private IP address for VPC internal communication.
- `billing`: Object containing hourly charges and monthly cost estimates.

Configuration

Configure NAT Gateway firewall rules and port forwarding to control inbound traffic.

Contents

| | | |
|-----------|------------------------------|------------|
| 01 | Firewall Rules | 217 |
| | Create | 219 |
| | Delete | 225 |
| | List | 230 |
| | Update | 235 |
| | Read | 240 |
| 02 | Port Forwarding Rules | 245 |
| | Create | 247 |
| | Delete | 252 |
| | List | 257 |
| | Read | 261 |
| | Update | 266 |

Firewall Rules

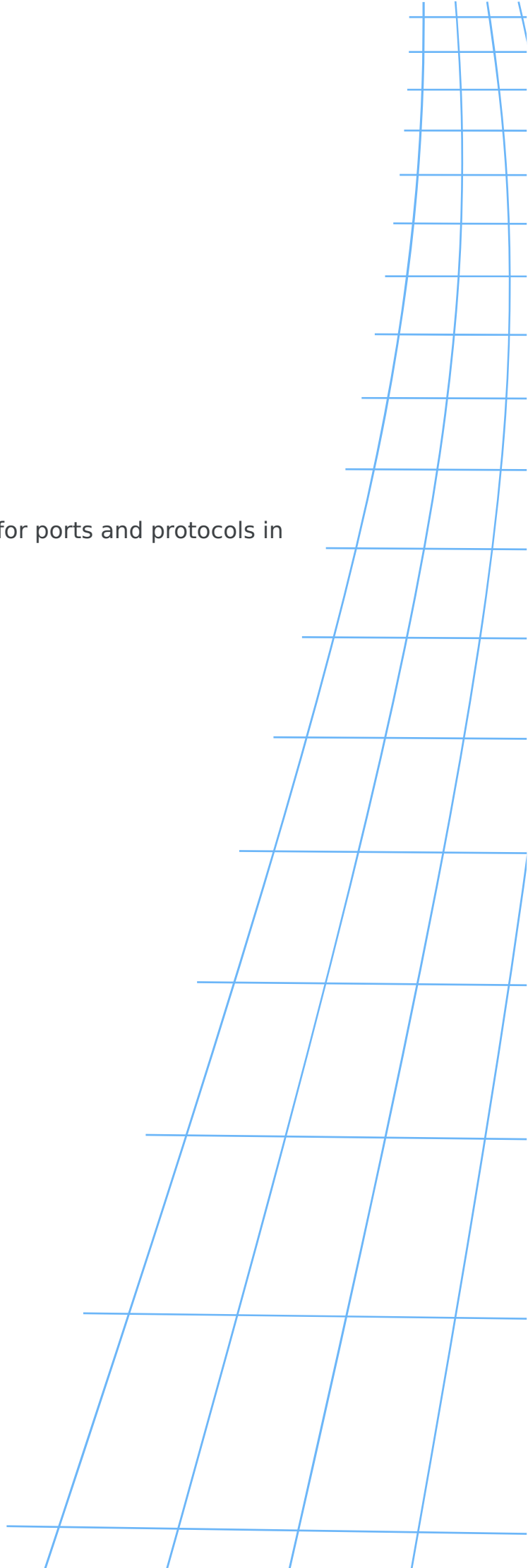
Manage NAT Gateway firewall rules to control allowed and denied traffic.

Contents

| | | |
|----|--------|-----|
| 01 | Create | 219 |
| 02 | Delete | 225 |
| 03 | List | 230 |
| 04 | Update | 235 |
| 05 | Read | 240 |

Create

Learn how to add NAT Gateway firewall rules for ports and protocols in Vultr.



Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Create a NAT Gateway Subscription Firewall Rule

Introduction

NAT Gateway firewall rules control which traffic is permitted or denied through the gateway. Rules can restrict traffic by IP protocol, port number, and source subnet to enforce least-privilege access policies. Each firewall rule requires a corresponding port forwarding rule for the same port to exist before creation.

Follow this guide to create a NAT Gateway subscription firewall rule using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Firewall** section.
4. Click **Add Firewall Rule**.

Note

If no firewall rules exist, the **Add Firewall Rule** button appears in the center of the section. Once rules are created, the button moves to the top right corner with a + icon.

A panel opens with the following configuration options:

- **Protocol:** Select the network protocol from the dropdown:
 - **TCP:** For TCP-based traffic.

- **UDP:** For UDP-based traffic.
- **Subnet:** Enter the source subnet in CIDR notation (e.g., `0.0.0.0` to allow all IPv4 addresses).
- **Subnet Size:** Enter the subnet mask size (use `0` with `0.0.0.0` to allow all addresses, or specify a value like `24` for restricted subnets).
- **Port / Range:** Enter the destination port or port range:
 - Single port: `443`, `22`
 - Port range: `8000:8002`
- **Note (optional):** Add a description for the rule.

5. Click **Save Changes**.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Verify a port forwarding rule exists for the target port. Send a `GET` request to the **List NAT Gateway Port Forwarding Rules** endpoint. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules" \  
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays existing port forwarding rules. Verify a rule exists for your target port. If no rule exists, create one before proceeding.

Note

Firewall rules cannot be created for ports without an associated port forwarding rule.

4. Send a `POST` request to the **Create NAT Gateway Firewall Rule** endpoint. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/firewall-rules" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "protocol": "tcp",  
  "port": "443",  
  "subnet": "0.0.0.0",  
  "subnet_size": 0,  
  "notes": "Allow HTTPS traffic"  
}'
```

Replace the field values as follows:

- `protocol`: Specify `tcp` or `udp`

- `port`: Specify the destination port or port range.
 - Single port: `"443"`, `"22"`
 - Port range: `"8000:8002"`
- `subnet`: Specify the source subnet in CIDR notation (use `0.0.0.0` to allow all IPv4 addresses)
- `subnet_size`: Enter the subnet mask size (use `0` with `0.0.0.0` to allow all addresses, or specify a value like `24` for restricted subnets)
- `notes`: Provide a description for the rule

The output displays the created firewall rule with an automatically assigned `id` and `action` field set to `accept`.

Delete

Learn how to permanently delete a NAT Gateway firewall rule in Vultr.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Delete a NAT Gateway Subscription Firewall Rule

Introduction

Deleting a firewall rule removes the traffic control policy immediately and stops the rule from affecting traffic through the NAT Gateway.

Warning

The deletion is permanent and cannot be undone. Verify no active flows depend on the rule before proceeding with removal.

Follow this guide to delete a NAT Gateway subscription firewall rule using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Firewall** section.
4. Locate your target firewall rule in the list.
5. Click the **Delete** icon (trash icon) for the rule you want to remove.

A confirmation dialog appears with the message "Delete Firewall Rule?" and warns that "This action cannot be undone. This will permanently delete this firewall rule."

6. Click **Delete Firewall Rule** to confirm deletion, or click **Cancel** to abort.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `GET` request to the **List NAT Gateway Firewall Rules** endpoint to retrieve firewall rule IDs. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/firewall-rules" \  
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays all firewall rules for the gateway. Each rule includes an `id`, `port`, `protocol`, `subnet`, and `notes` field. Note the `id` field for the rule you want to delete.

4. Send a `DELETE` request to the **Delete NAT Gateway Firewall Rule** endpoint. Replace `VPC_ID`, `NAT_GATEWAY_ID`, and `FIREWALL_RULE_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/firewall-rules/FIREWALL_RULE_ID" \  
-X DELETE \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

The API returns an HTTP 204 status code with no response body when the deletion succeeds. The rule is removed immediately and no longer affects traffic through the gateway.

List

Learn how to list NAT Gateway firewall rules in Vultr via portal or API.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to List NAT Gateway Subscription Firewall Rules

Introduction

NAT Gateway firewall rules control which traffic is permitted or denied through the gateway based on IP protocol, port, and source subnet. Listing rules allows you to review active policies, verify security configurations, troubleshoot connectivity issues, and audit traffic controls for workloads using the NAT Gateway.

Follow this guide to list NAT Gateway subscription firewall rules using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Firewall** section.

The section displays all configured firewall rules in a table with the following columns:

- **Action:** The rule action (Accept).
- **Protocol:** The network protocol (TCP or UDP).
- **Port / Range:** The destination port or port range.
- **IP:** The source subnet in CIDR notation with subnet size.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `GET` request to the **List NAT Gateway Firewall Rules** endpoint to retrieve all firewall rules. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/firewall-rules" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays all firewall rules configured for the gateway in the `firewall_rules` array. Each rule object contains:

- `id`: Unique identifier for the rule
- `action`: Rule action (`accept`)
- `protocol`: Network protocol (`tcp` or `udp`)
- `port`: Target port number as a string
- `subnet`: Source subnet in CIDR notation
- `subnet_size`: Subnet mask size
- `notes`: Rule description

The response includes a `meta` object with the total count and pagination links.

Update

Learn how to update NAT Gateway firewall rule notes via portal or API.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Update a NAT Gateway Firewall Rule

Introduction

Firewall rule updates allow you to modify the documentation notes attached to an existing rule. The traffic control parameters such as protocol, port, and subnet cannot be changed after the rule is created.

Note

Only the `notes` field can be modified after creating a firewall rule. To change the protocol, port, or subnet, delete the existing rule and create a new one with the desired configuration.

Follow this guide to update a NAT Gateway subscription firewall notes field using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Firewall** section.
4. Locate your target firewall rule in the list.
5. Click the **Edit** icon (pencil icon) for the rule you want to modify.

The Edit NAT Firewall Rule panel opens showing the current configuration. Only the **Note** field is editable. All other fields (Protocol, Subnet, Subnet Size, and Port/Range) are read-only.

6. Update the **Note** field with your new description.
7. Click **Save Changes**.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `GET` request to the **List NAT Gateway Firewall Rules** endpoint to retrieve firewall rule IDs. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/firewall-rules" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays all firewall rules for the gateway. Each rule includes an `id`, `port`, `protocol`, `subnet`, and `notes` field. Note the `id` field for the rule you want to update.

4. Send a `PUT` request to the **Update NAT Gateway Firewall Rule** endpoint. Replace `VPC_ID`, `NAT_GATEWAY_ID`, and `FIREWALL_RULE_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/firewall-rules/FIREWALL_RULE_ID" \  
-X PUT \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "notes": "Updated firewall rule description"  
}'
```

The output displays the updated firewall rule configuration with the modified notes.

Read

Learn how to get a specific NAT Gateway firewall rule in Vultr.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Get a NAT Gateway Firewall Rule

Introduction

Retrieving a firewall rule displays its complete configuration, including the protocol, port range, subnet, and action type. Review rule details before modifying them or to verify security policies.

Follow this guide to get a NAT Gateway subscription firewall rule using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Firewall** section.

The firewall rules table displays the following information for each rule:

- **Action:** The rule action (Accept).
- **Protocol:** The network protocol (TCP or UDP).
- **Port / Range:** The destination port or port range.
- **IP:** The source subnet in CIDR notation with subnet size.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `GET` request to the **List NAT Gateway Firewall Rules** endpoint to retrieve firewall rule IDs. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/firewall-rules" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays all firewall rules for the gateway. Each rule includes an `id`, `port`, `protocol`, `subnet`, and `notes` field. Note the `id` field for the rule you want to retrieve.

4. Send a `GET` request to the **Get NAT Gateway Firewall Rule** endpoint. Replace `VPC_ID`, `NAT_GATEWAY_ID`, and `FIREWALL_RULE_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/firewall-rules/FIREWALL_RULE_ID" \  
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays the complete firewall rule configuration. The response includes the `id`, `action`, `protocol`, `port`, `subnet`, `subnet_size`, and `notes` fields for the specified rule.

Port Forwarding Rules

Manage NAT Gateway port forwarding rules to expose private services securely.

Contents

| | | |
|----|--------|-----|
| 01 | Create | 247 |
| 02 | Delete | 252 |
| 03 | List | 257 |
| 04 | Read | 261 |
| 05 | Update | 266 |

Create

Learn how to forward ports through a Vultr NAT Gateway using portal or API.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Create a NAT Gateway Port Forwarding Rule

Introduction

Port forwarding rules allow external traffic to reach specific services on private instances through the NAT Gateway. Configure the external port, internal destination IP, and protocol to enable controlled access for services like SSH, web servers, or custom applications.

Follow this guide to create a NAT Gateway subscription port forwarding rule using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Port Forwarding** section.
4. Click **Add Forwarding Rule**.

Note

If no port forwarding rules exist, the **Add Forwarding Rule** button appears in the center of the section. Once rules are created, the button moves to the top right corner with a + icon.

A panel opens with the following configuration options:

- **Rule Name:** Enter a descriptive name to identify the forwarding rule.

- **Protocol:** Select the network protocol:
 - **TCP:** For TCP-based services.
 - **UDP:** For UDP-based services.
 - **Both:** To forward both TCP and UDP traffic on the same ports.
- **External Port:** Enter the port number on the NAT Gateway's public IP that receives incoming traffic.
- **Internal IP:** Enter the private IP address of the target instance within the VPC.
- **Internal Port:** Enter the port number on the target instance to forward traffic to.
- **Note (optional):** Add a description or note for the rule.

5. Click **Save Changes**.

The port forwarding rule applies immediately. Traffic sent to the NAT Gateway's public IP on the external port now forwards to the specified internal instance.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `POST` request to the **Create NAT Gateway Port Forwarding Rule** endpoint. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values. Specify the rule name, protocol (tcp, udp, or both), external port, internal IP address, internal port, and whether the rule is enabled.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules" \  
-X POST \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "name": "ssh-to-webserver",  
  "protocol": "tcp",  
  "external_port": 2222,  
  "internal_ip": "10.0.0.10",  
  "internal_port": 22,  
  "enabled": true,  
  "description": "SSH access to web server"  
}'
```

Verify that the response contains the created port forwarding rule with its unique ID, configuration details, and timestamps. Traffic sent to the NAT Gateway's public IP on port 2222 now forwards to the internal instance at `10.0.0.10` on port 22.

Delete

Learn how to delete NAT Gateway port forwarding rules and disable external access.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Delete a NAT Gateway Port Forwarding Rule

Introduction

Deleting a port forwarding rule removes inbound access through the NAT Gateway on the specified port. The deletion takes effect immediately and stops traffic from reaching the internal destination.

Warning

Verify that no critical services depend on the port forwarding rule before deletion.

Follow this guide to delete a NAT Gateway subscription port forwarding rule using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Port Forwarding** section.
4. Locate your target port forwarding rule in the list.
5. Click the **Delete** icon (trash icon) for the rule you want to remove.

A confirmation dialog appears with the message "Delete Forwarding Rule?" and warns that "This action cannot be undone. This will permanently delete this port forwarding rule."

- Click **Delete Port Forwarding Rule!** to confirm deletion, or click **Cancel** to abort.

Vultr API

- Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

- Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

- Send a `GET` request to the **List NAT Gateway Port Forwarding Rules** endpoint to retrieve port forwarding rule IDs. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays all port forwarding rules for the gateway. Each rule includes an `id`, `name`, `external_port`, `internal_ip`, and `internal_port` field. Note the `id` field for the rule you want to delete.

4. Send a `DELETE` request to the **Delete NAT Gateway Port Forwarding Rule** endpoint. Replace `VPC_ID`, `NAT_GATEWAY_ID`, and `PORT_FORWARDING_RULE_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules/  
PORT_FORWARDING_RULE_ID" \  
-X DELETE \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

The API returns an HTTP 204 status code with no response body when the deletion succeeds. The port forwarding rule is removed immediately and traffic no longer routes to the internal destination.

List

Retrieve configured NAT Gateway port forwarding rules for auditing and troubleshooting.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to List NAT Gateway Port Forwarding Rules

Introduction

Port forwarding rules define which external ports on the NAT Gateway route traffic to specific internal instances.

Follow this guide to list NAT Gateway subscription port forwarding rules using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Port Forwarding** section.

The port forwarding rules table displays all configured rules with the following information:

- **Name / UUID:** The rule name and unique identifier.
- **Toggle Switch:** Enable or disable the rule.
- **Protocol:** The network protocol (TCP, UDP, or Both).
- **External Port:** The port on the NAT Gateway's public IP that receives incoming traffic.
- **Internal IP:** The private IP address of the target instance.
- **Internal Port:** The port on the target instance that receives forwarded traffic.
- **Actions:** Edit and delete icons for managing the rule.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `GET` request to the **List NAT Gateway Port Forwarding Rules** endpoint. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

Verify that the response contains all port forwarding rules configured for the NAT Gateway. Each rule includes the `id`, `name`, `protocol`, `external_port`, `internal_ip`, `internal_port`, `enabled` status, and `description` fields.

Read

Learn how to get a specific NAT Gateway port forwarding rule in Vultr.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Get a NAT Gateway Port Forwarding Rule

Introduction

Retrieving a specific port forwarding rule displays its complete configuration, including the external port, internal destination IP, protocol, and enabled status.

Follow this guide to get a NAT Gateway subscription port forwarding rule using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Port Forwarding** section.
4. Locate your target port forwarding rule in the table.

The table displays the complete rule configuration:

- **Name / UUID:** The rule name and unique identifier.
- **Protocol:** The network protocol (TCP, UDP, or Both).
- **External Port:** The port on the NAT Gateway's public IP that receives incoming traffic.
- **Internal IP:** The private IP address of the target instance.
- **Internal Port:** The port on the target instance that receives forwarded traffic.
- **Toggle Switch:** Shows whether the rule is enabled (blue) or disabled (gray).

 Note

To view additional details such as the description and timestamps, use the API.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `GET` request to the **List NAT Gateway Port Forwarding Rules** endpoint to retrieve port forwarding rule IDs. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

```
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays all port forwarding rules for the gateway. Each rule includes an `id`, `name`, `external_port`, `internal_ip`, and `internal_port` field. Note the `id` field for the rule you want to retrieve.

4. Send a `GET` request to the **Get NAT Gateway Port Forwarding Rule** endpoint. Replace `VPC_ID`, `NAT_GATEWAY_ID`, and `PORT_FORWARDING_RULE_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules/  
PORT_FORWARDING_RULE_ID" \  
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

Verify that the response contains the complete port forwarding rule configuration. The response includes the `id`, `name`, `protocol`, `external_port`, `internal_ip`, `internal_port`, `enabled` status, `description`, and `timestamp` fields for the specified rule.

Update

Learn how to update NAT Gateway port forwarding rules and enabled status.

Contents

| | | |
|----|-----------------------|----|
| 01 | Introduction | 10 |
| 02 | Vultr Customer Portal | 26 |
| 03 | Vultr API | 26 |

How to Update a NAT Gateway Port Forwarding Rule

Introduction

Updating a port forwarding rule modifies its configuration to reflect changes in service ports, internal destination addresses, protocols, or enabled status. Adjust rules as your infrastructure evolves to maintain accurate traffic routing and security posture.

Follow this guide to update a NAT Gateway subscription port forwarding rule using the Vultr Customer Portal or API.

Vultr Customer Portal

1. Navigate to **Products**, expand the **Network** drop-down and select **VPC Networks**.
2. Select your target VPC Network with NAT Gateway connectivity.
3. Scroll to the **NAT Port Forwarding** section.
4. Locate your target port forwarding rule in the list.
5. To enable or disable the rule, click the toggle switch next to the rule name.
6. Click the **Edit** icon (pencil icon) for the rule you want to modify.

The Edit Port Forwarding Rule panel opens with the current configuration:

- **Rule Name:** Modify the rule name if needed.
- **Protocol:** Change the protocol (TCP, UDP, or Both).
- **External Port:** Update the port number on the NAT Gateway's public IP.

- **Internal IP:** Change the private IP address of the target instance.
- **Internal Port:** Modify the port number on the target instance.
- **Note (optional):** Update the description or note.

7. Click **Save Changes**.

Vultr API

1. Send a `GET` request to the [List VPCs endpoint](#) to retrieve available VPCs.

CONSOLE

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

The output displays all VPCs in your account. Note the `id` field for the target VPC.

2. Send a `GET` request to the **List NAT Gateway subscriptions** endpoint to retrieve the gateway ID. Replace `VPC_ID` with the ID from the previous step.

CONSOLE

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

The output displays NAT Gateway subscriptions for the VPC. Note the `id` field for the target gateway.

3. Send a `GET` request to the **List NAT Gateway Port Forwarding Rules** endpoint to retrieve port forwarding rule IDs. Replace `VPC_ID` and `NAT_GATEWAY_ID` with your values.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules" \  
  -X GET \  
  -H "Authorization: Bearer ${VULTR_API_KEY}"
```

```
-X GET \  
-H "Authorization: Bearer ${VULTR_API_KEY}"
```

The output displays all port forwarding rules for the gateway. Each rule includes an `id`, `name`, `external_port`, `internal_ip`, and `internal_port` field. Note the `id` field for the rule you want to update.

4. Send a `PUT` request to the **Update NAT Gateway Port Forwarding Rule** endpoint. Replace `VPC_ID`, `NAT_GATEWAY_ID`, and `PORT_FORWARDING_RULE_ID` with your values. Specify the fields you want to update in the request body, including `name`, `protocol` (tcp, udp, or both), `external_port`, `internal_ip`, `internal_port`, `enabled` status, and `description`.

CONSOLE

```
$ curl "https://api.vultr.com/v2/vpcs/VPC_ID/nat-gateway/  
NAT_GATEWAY_ID/global/port-forwarding-rules/  
PORT_FORWARDING_RULE_ID" \  
-X PUT \  
-H "Authorization: Bearer ${VULTR_API_KEY}" \  
-H "Content-Type: application/json" \  
--data '{  
  "name": "ssh-to-webserver-updated",  
  "protocol": "tcp",  
  "external_port": 2223,  
  "internal_ip": "10.0.0.10",  
  "internal_port": 22,  
  "enabled": true,  
  "description": "SSH access - port updated to 2223"  
}'
```

Verify that the response contains the updated port forwarding rule configuration. The changes apply immediately and affect how traffic routes through the NAT Gateway to the internal destination.



VULTR

