

Vultr Provider

The Vultr provider is used to interact with the resources supported by Vultr (<https://www.vultr.com>). The provider needs to be configured with the proper credentials before it can be used.

Example Usage

```
# Configure the Vultr Provider
provider "vultr" {
  api_key = "VULTR_API_KEY"
  rate_limit = 700
}

# Create a web server
resource "vultr_server" "web" {
  # ...
}
```

Argument Reference

The following arguments are supported:

- `api_key` - (Required) This is the Vultr API API (<https://my.vultr.com/settings/#settingsapi>). This can also be specified with the `VULTR_API_KEY` shell environment variable.
- `rate_limit` - (Optional) Vultr limits API calls to 3 calls per second. This field lets you configure how the rate limit using milliseconds. The default value if this field is omitted is `650 milliseconds` per call.

vultr_account

Get information about your Vultr account. This data source provides the balance, pending charges, last payment date, and last payment amount for your Vultr account.

Example Usage

Get the information for an account:

```
data "vultr_account" "my_account" {}
```

Argument Reference

This data source does not take any arguments. It will return the account information associated with the Vultr API key you have set.

Attributes Reference

The following attributes are exported:

- `balance` - The current balance on your Vultr account.
- `pending_charges` - The pending charges on your Vultr account.
- `last_payment_date` - The date of the last payment made on your Vultr account.
- `last_payment_amount` - The amount of the last payment made on your Vultr account.

vultr_api_key

Get information about your Vultr API key. This data source provides the name, email, and access control list for your Vultr API key.

Example Usage

Get the information for your API key:

```
data "vultr_api_key" "my_api_key" {}
```

Argument Reference

This data source does not take any arguments. It will return the API key information associated with the Vultr API key you have set.

Attributes Reference

The following attributes are exported:

- `name` - The name associated with your Vultr API key.
- `email` - The email associated with your Vultr API key.
- `acl` - The access control list for your Vultr API key.

vultr_application

Get information about applications that can be launched when creating a Vultr VPS.

Example Usage

Get the information for an application by `deploy_name` :

```
data "vultr_application" "docker" {
  filter {
    name   = "deploy_name"
    values = ["Docker on CentOS 7 x64"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding applications.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The name of the application.
- `deploy_name` - The deploy name of the application.
- `short_name` - The short name of the application.

vultr_backup

Get information about a Vultr backup. This data source provides a list of backups which contain the description, size, status, and the creation date for your Vultr backup.

Example Usage

Get the information for a backup by description :

```
data "vultr_backup" "my_backup" {
  filter {
    name = "description"
    values = ["my-backup-description"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding backups.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `BACKUPID` - The ID of the backup
- `description` - The description of the backup.
- `size` - The size of the backup in bytes.
- `status` - The status of the backup.
- `date_created` - The date the backup was added to your Vultr account.

vultr_bare_metal_plan

Get information about a Vultr bare metal server plan.

Example Usage

Get the information for a plan by name :

```
data "vultr_bare_metal_plan" "my_plan" {
  filter {
    name = "name"
    values = ["32768 MB RAM,4x 240 GB SSD,1.00 TB BW"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding plans.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The name of the plan.
- `cpu_count` - The number of CPUs available on the plan.
- `cpu_model` - The CPU model of the plan.
- `ram` - The amount of memory available on the plan in MB.
- `disk` - The description of the disk(s) on the plan.
- `bandwidth_tb` - The bandwidth available on the plan in TB.
- `price_per_month` - The price per month of the plan in USD.
- `plan_type` - The type of plan it is.
- `available_locations` - A list of DCIDs (used as `region_id` in Terraform) where the plan can be deployed.
- `deprecated` - Indicates that the plan will be going away in the future. New deployments of it will still be accepted, but

you should begin to transition away from its usage.

vultr_bare_metal_server

Get information about a Vultr bare metal server.

Example Usage

Get the information for a server by `label` :

```
data "vultr_bare_metal_server" "my_server" {
  filter {
    name   = "label"
    values = ["my-server-label"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding servers.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `os` - The operating system of the server.
- `ram` - The amount of memory available on the server in MB.
- `disk` - The description of the disk(s) on the server.
- `main_ip` - The server's main IP address.
- `cpu_count` - The number of CPUs available on the server.
- `location` - The location of the server.
- `region_id` - The region ID (`DCID` in the Vultr API) of the server.
- `default_password` - The server's default password.
- `date_created` - The date the server was added to your Vultr account.
- `status` - The status of the server's subscription.

- `netmask_v4` - The server's IPv4 netmask.
- `gateway_v4` - The server's IPv4 gateway.
- `plan_id` - The server's plan ID.
- `v6_networks` - A list of the server's IPv6 networks.
- `label` - The server's label.
- `tag` - The server's tag.
- `os_id` - The server's operating system ID.
- `app_id` - The server's application ID.

vultr_block_storage

Get information about a Vultr block storage subscription.

Example Usage

Get the information for a block storage subscription by `label` :

```
data "vultr_block_storage" "my_block_storage" {
  filter {
    name = "label"
    values = ["my-block-storage-label"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding block storage subscriptions.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `label` - The label of the block storage subscription.
- `cost_per_month` - The cost per month of the block storage subscription in USD.
- `status` - The status of the block storage subscription.
- `size_gb` - The size of the block storage subscription in GB.
- `region_id` - The region ID (`DCID` in the Vultr API) of the block storage subscription.
- `attached_to_vps` - The ID of the VPS the block storage subscription is attached to.
- `date_created` - The date the block storage subscription was added to your Vultr account.

vultr_dns_domain

Get information about a DNS domain associated with your Vultr account.

Example Usage

Get the information for a DNS domain:

```
data "vultr_dns_domain" "my_domain" {
  domain = "example.com"
}
```

Argument Reference

The following arguments are supported:

- `domain` - (Required) The name you're searching for.

Attributes Reference

The following attributes are exported:

- `domain` - Name of domain.
- `date_created` - The date the DNS domain was added to your Vultr account.

vultr_firewall_group

Get information about a firewall group on your Vultr account.

Example Usage

Get the information for a firewall group by `description`:

```
data "vultr_firewall_group" "my_fw" {
  filter {
    name   = "description"
    values = ["fw-description"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding firewall groups.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `description` - The description of the firewall group.
- `date_created` - The date the firewall group was added to your Vultr account.
- `date_modified` - The date the firewall group was last modified.
- `instance_count` - The number of instances this firewall group is applied to.
- `rule_count` - The number of rules added to this firewall group.
- `max_rule_count` - The maximum number of rules this firewall group can have.

vultr_iso_private

Get information about an ISO file uploaded to your Vultr account.

Example Usage

Get the information for a ISO file by `filename` :

```
data "vultr_iso_private" "my_iso" {
  filter {
    name   = "filename"
    values = ["my-iso-filename"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding ISO files.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `filename` - The ISO file's filename.
- `status` - The status of the ISO file.
- `size` - The size of the ISO file in bytes.
- `md5sum` - The md5 hash of the ISO file.
- `sha512sum` - The sha512 hash of the ISO file.
- `date_created` - The date the ISO file was added to your Vultr account.

vultr_iso_public

Get information about an ISO file offered in the Vultr ISO library.

Example Usage

Get the information for a ISO file by `description` :

```
data "vultr_iso_public" "my_iso" {
  filter {
    name   = "description"
    values = ["iso-description"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding ISO files.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The ISO file's name.
- `description` - The description of the ISO file.

vultr_network

Get information about a Vultr private network.

Example Usage

Get the information for a private network by `description` :

```
data "vultr_network" "my_network" {
  filter {
    name = "description"
    values = ["my-network-description"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding private networks.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `region_id` - The ID of the region that the private network is in.
- `cidr_block` - The CIDR block of the private network.
- `description` - The private network's description.
- `date_created` - The date the private network was added to your Vultr account.

vultr_os

Get information about operating systems that can be launched when creating a Vultr VPS.

Example Usage

Get the information for an operating system by `name` :

```
data "vultr_os" "centos" {
  filter {
    name = "name"
    values = ["CentOS 7 x64"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding operating systems.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The name of the operating system.
- `arch` - The architecture of the operating system.
- `family` - The family of the operating system.
- `windows` - If true, a Windows license will be included with the instance, which will increase the cost.

vultr_plan

Get information about a Vultr plan.

Example Usage

Get the information for a plan by name :

```
data "vultr_plan" "my_plan" {
  filter {
    name = "name"
    values = ["8192 MB RAM,160 GB SSD,4.00 TB BW"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding plans.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The name of the plan.
- `vcpu_count` - The number of virtual CPUs available on the plan.
- `ram` - The amount of memory available on the plan in MB.
- `disk` - The amount of disk space in GB available on the plan.
- `bandwidth` - The bandwidth available on the plan in TB.
- `bandwidth_gb` - The bandwidth available on the plan in GB.
- `price_per_month` - The price per month of the plan in USD.
- `plan_type` - The type of plan it is.
- `available_locations` - A list of DCIDs (used as `region_id` in Terraform) where the plan can be deployed.
- `deprecated` - Indicates that the plan will be going away in the future. New deployments of it will still be accepted, but

you should begin to transition away from its usage.

vultr_region

Get information about a Vultr region.

Example Usage

Get the information for a region by name :

```
data "vultr_region" "my_region" {
  filter {
    name   = "name"
    values = ["Miami"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding regions.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The name of the region.
- `continent` - The continent the region is in.
- `country` - The country the region is in.
- `state` - The state the region is in.
- `ddos_protection` - Whether the region has DDoS protection.
- `block_storage` - Whether the region has block storage.
- `regioncode` - The region code of the region.

vultr_reserved_ip

Get information about a Vultr reserved IP address.

Example Usage

Get the information for a reserved IP by `label` :

```
data "vultr_reserved_ip" "my_reserved_ip" {
  filter {
    name = "label"
    values = ["my-reserved-ip-label"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding reserved IP addresses.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `region_id` - The ID of the region that the reserved IP is in.
- `ip_type` - The IP type of the reserved IP.
- `subnet` - The subnet of the reserved IP.
- `subnet_size` - The subnet size of the reserved IP.
- `label` - The label of the reserved IP.
- `attached_to_vps` - The ID of the VPS the reserved IP is attached to.

vultr_server

Get information about a Vultr server.

Example Usage

Get the information for a server by `label` :

```
data "vultr_server" "my_server" {
  filter {
    name   = "label"
    values = ["my-server-label"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding servers.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `os` - The operating system of the server.
- `ram` - The amount of memory available on the server in MB.
- `disk` - The description of the disk(s) on the server.
- `main_ip` - The server's main IP address.
- `vps_cpu_count` - The number of virtual CPUs available on the server.
- `location` - The physical location of the server.
- `region_id` - The region ID (`DCID` in the Vultr API) of the server.
- `default_password` - The server's default password.
- `date_created` - The date the server was added to your Vultr account.
- `pending_charges` - Charges pending for this server's subscription in USD.

- `cost_per_month` - The server's cost per month in USD.
- `current_bandwidth` - The server's current bandwidth usage in GB.
- `allowed_bandwidth` - The server's allowed bandwidth usage in GB.
- `netmask_v4` - The server's IPv4 netmask.
- `gateway_v4` - The server's IPv4 gateway.
- `status` - The status of the server's subscription.
- `power_status` - Whether the server is powered on or not.
- `server_state` - A more detailed server status (none, locked, installingbooting, isomounting, ok).
- `plan_id` - The server's plan ID.
- `v6_networks` - A list of the server's IPv6 networks.
- `label` - The server's label.
- `internal_ip` - The server's internal IP address.
- `kvm_url` - The server's current KVM URL. This URL will change periodically. It is not advised to cache this value.
- `auto_backups` - Whether auto backups are enabled on this server.
- `tag` - The server's tag.
- `os_id` - The server's operating system ID.
- `app_id` - The server's application ID.
- `firewall_group_id` - The ID of the firewall group applied to this server.

vultr_snapshot

Get information about a Vultr snapshot.

Example Usage

Get the information for a snapshot by `description` :

```
data "vultr_snapshot" "my_snapshot" {
  filter {
    name = "description"
    values = ["my-snapshot-description"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding snapshots.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `description` - The description of the snapshot.
- `size` - The size of the snapshot in bytes.
- `status` - The status of the snapshot.
- `date_created` - The date the snapshot was added to your Vultr account.
- `os_id` - The operating system ID of the snapshot.
- `app_id` - The application ID of the snapshot.

vultr_ssh_key

Get information about a Vultr SSH key. This data source provides the name, public SSH key, and the creation date for your Vultr SSH key.

Example Usage

Get the information for an SSH key by `name` :

```
data "vultr_ssh_key" "my_ssh_key" {
  filter {
    name = "name"
    values = ["my-ssh-key-name"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding SSH keys.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The name of the SSH key.
- `ssh_key` - The public SSH key.
- `date_created` - The date the SSH key was added to your Vultr account.

vultr_startup_script

Get information about a Vultr startup script. This data source provides the name, script, type, creation date, and the last modification date for your Vultr startup script.

Example Usage

Get the information for an startup script by `name` :

```
data "vultr_startup_script" "my_startup_script" {
  filter {
    name = "name"
    values = ["my-startup-script-name"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding startup scripts.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The name of the startup script.
- `script` - The contents of the startup script.
- `type` - The type of the startup script.
- `date_created` - The date the startup script was added to your Vultr account.
- `date_modified` - The date the startup script was last modified.

vultr_user

Get information about a Vultr user associated with your account. This data source provides the name, email, access control list, and API status for a Vultr user associated with your account.

Example Usage

Get the information for a user by email :

```
data "vultr_user" "my_user" {
  filter {
    name = "email"
    values = ["jdoe@example.com"]
  }
}
```

Get the information for a user by name :

```
data "vultr_user" "my_user" {
  filter {
    name = "name"
    values = ["John Doe"]
  }
}
```

Argument Reference

The following arguments are supported:

- `filter` - (Required) Query parameters for finding users.

The `filter` block supports the following:

- `name` - Attribute name to filter with.
- `values` - One or more values filter with.

Attributes Reference

The following attributes are exported:

- `name` - The name of the user.
- `email` - The email of the user.
- `api_enabled` - Whether API is enabled for the user.
- `acl` - The access control list for the user.

vultr_bare_metal_server

Provides a Vultr bare metal server resource. This can be used to create, read, modify, and delete bare metal servers on your Vultr account.

Example Usage

Create a new bare metal server:

```
resource "vultr_bare_metal_server" "my_server" {
  plan_id = "100"
  region_id = "40"
  os_id = "270"
}
```

Create a new bare metal server with options:

```
resource "vultr_bare_metal_server" "my_server" {
  plan_id = "100"
  region_id = "40"
  os_id = "270"
  label = "my-server-label"
  tag = "my-server-tag"
  hostname = "my-server-hostname"
  user_data = '{"foo': true}"
  enable_ipv6 = true
  notify_activate = false
}
```

Argument Reference

The following arguments are supported:

- `region_id` - (Required) The ID of the region that the server is to be created in.
- `plan_id` - (Required) The ID of the plan that you want the server to subscribe to.
- `os_id` - (Optional) The ID of the operating system to be installed on the server.
- `app_id` - (Optional) The ID of the Vultr application to be installed on the server.
- `snapshot_id` - (Optional) The ID of the Vultr snapshot that the server will restore for the initial installation.
- `script_id` - (Optional) The ID of the startup script you want added to the server.
- `ssh_key_ids` - (Optional) A list of SSH key IDs to apply to the server on install (only valid for Linux/FreeBSD).
- `user_data` - (Optional) Generic data store, which some provisioning tools and cloud operating systems use as a configuration file. It is generally consumed only once after an instance has been launched, but individual needs may vary.

- `enable_ipv6` - (Optional) Whether the server has IPv6 networking activated.
- `notify_activate` - (Optional) Whether an activation email will be sent when the server is ready.
- `hostname` - (Optional) The hostname to assign to the server.
- `tag` - (Optional) The tag to assign to the server.
- `label` - (Optional) A label for the server.

Attributes Reference

The following attributes are exported:

- `id` - ID of the server.
- `region_id` - The ID of the region that the server is in.
- `os` - The string description of the operating system installed on the server.
- `ram` - The amount of memory available on the server in MB.
- `disk` - The description of the disk(s) on the server.
- `main_ip` - The server's main IP address.
- `cpu_count` - The number of CPUs available on the server.
- `location` - The physical location of the server.
- `default_password` - The server's default password.
- `date_created` - The date the server was added to your Vultr account.
- `netmask_v4` - The server's IPv4 netmask.
- `gateway_v4` - The server's IPv4 gateway.
- `status` - The status of the server's subscription.
- `v6_networks` - A list of the server's IPv6 networks.
- `plan_id` - The ID of the plan that server is subscribed to.
- `os_id` - The ID of the operating system installed on the server.
- `app_id` - The ID of the Vultr application installed on the server.
- `snapshot_id` - The ID of the Vultr snapshot that the server was restored from.
- `script_id` - The ID of the startup script that was added to the server.
- `ssh_key_ids` - A list of SSH key IDs applied to the server on install.
- `user_data` - Base64 encoded generic data store, which some provisioning tools and cloud operating systems use as a configuration file. It is generally consumed only once after an instance has been launched, but individual needs may vary.

- `enable_ipv6` - Whether the server has IPv6 networking activated.
- `notify_activate` - Whether an activation email was sent when the server was ready.
- `hostname` - The hostname assigned to the server.
- `tag` - The tag assigned to the server.
- `label` - A label for the server.

Import

Bare Metal Servers can be imported using the server `SUBID`, e.g.

```
terraform import vultr_bare_metal_server.my_server 1312965
```

vultr_block_storage

Provides a Vultr Block Storage resource. This can be used to create, read, modify, and delete Block Storage.

Example Usage

Create a new Block Storage

```
resource "vultr_block_storage" "my_blockstorage" {  
  size_gb = 10  
  region_id = 1  
}
```

Argument Reference

The following arguments are supported:

- `size_gb` - (Required) The size of the given block storage.
- `region_id` - (Required) Region in which this block storage will reside in. (Currently only NJ/NY supported region_id 1)
- `attached_id` - (Optional) VPS ID that you want to have this block storage attached to.
- `label` - (Optional) Label that is given to your block storage.

Attributes Reference

The following attributes are exported:

- `size_gb` - The size of the given block storage.
- `region_id` - Region in which this block storage will reside in. (Currently only NJ/NY supported region_id 1)
- `attached_id` - VPS ID that is attached to this block storage.
- `label` - Label that is given to your block storage.
- `cost_per_month` - The monthly cost of this block storage.
- `date_created` - The date this block storage was created.
- `status` - Current status of your block storage.
- `id` - The ID for this block storage.

Import

Block Storage can be imported using the Block Storage SUBID , e.g.

```
terraform import vultr_block_storage.my_blockstorage 25058682
```

vultr_dns_domain

Provides a Vultr DNS Domain resource. This can be used to create, read, modify, and delete DNS Domains.

Example Usage

Create a new DNS Domain

```
resource "vultr_dns_domain" "my_domain" {
  domain = "domain.com"
  server_ip = "66.42.94.227"
}
```

Argument Reference

The following arguments are supported:

- `domain` - (Required) Name of domain.
- `server_ip` - (Required) Server IP you want associated to domain.

Attributes Reference

The following attributes are exported:

- `id` - The ID is the name of the domain.
- `domain` - Name of domain.
- `server_ip` - Server IP you want associated to domain.

Import

DNS Domains can be imported using the Dns Domain `domain`, e.g.

```
terraform import vultr_dns_domain.name domain.com
```

vultr_dns_record

Provides a Vultr DNS Record resource. This can be used to create, read, modify, and delete DNS Records.

Example Usage

Create a new DNS Record

```
resource "vultr_dns_domain" "my_domain" {
  domain = "domain.com"
  server_ip = "66.42.94.227"
}

resource "vultr_dns_record" "my_record" {
  domain = "${vultr_dns_domain.my_domain.id}"
  name = "www"
  data = "66.42.94.227"
  type = "A"
}
```

Argument Reference

The following arguments are supported:

- `data` - (Required) IP Address of the server the domain is associated with.
- `domain` - (Required) Name of the DNS Domain this record will belong to.
- `name` - (Required) Name (subdomain) for this record.
- `type` - (Required) Type of record.
- `priority` - (Optional) Priority of this record (only required for MX and SRV).
- `ttl` - (Optional) The time to live of this record.

Attributes Reference

The following attributes are exported:

- `id` - ID associated with the record.
- `data` - IP Address of the server the domain is associated with.
- `domain` - Name of the DNS Domain this record will belong to.
- `name` - Name for this record (Can be subdomain).
- `type` - Type of record.

- `priority` - Priority of this record (only required for MX and SRV).
- `ttl` - The time to live of this record.

vultr_firewall_group

Provides a Vultr Firewall Group resource. This can be used to create, read, modify, and delete Firewall Group.

Example Usage

Create a new Firewall group

```
resource "vultr_firewall_group" "my_firewallgroup" {  
  description = "base firewall"  
}
```

Argument Reference

The following arguments are supported:

- `description` - (Optional) Description of the firewall group.

Attributes Reference

The following attributes are exported:

- `id` - ID of the firewall group.
- `description` - Description of the firewall group.
- `date_created` - The date the firewall group was created.
- `date_modified` - The date the firewall group was modified.
- `instance_count` - The number of servers that are currently using this firewall group.
- `max_rule_count` - The number of max firewall rules this group can have.
- `rule_count` - The number of firewall rules this group currently has.

Import

Firewall Groups can be imported using the Firewall Group `FIREWALLGROUPID`, e.g.

```
terraform import vultr_firewall_group.my_firewallgroup c342f929
```

vultr_firewall_rule

Provides a Vultr Firewall Rule resource. This can be used to create, read, modify, and delete Firewall rules.

Example Usage

Create a Firewall Rule

```
resource "vultr_firewall_group" "my_firewallgroup" {
  description = "base firewall"
}

resource "vultr_firewall_rule" "my_firewallrule" {
  firewall_group_id = "${vultr_firewall_group.my_firewallgroup.id}"
  protocol = "tcp"
  network = "0.0.0.0/0"
  from_port = "8085"
  to_port = "8090"
}
```

Argument Reference

The following arguments are supported:

- `firewall_group_id` - (Required) The firewall group that the firewall rule will belong to.
- `protocol` - (Required) The type of protocol for this firewall rule. Possible values (icmp, tcp, udp, gre) **Note** they must be lowercase
- `network` - (Required) IP address that you want to define for this firewall rule.
- `from_port` - (Optional) Port that you want to define for this rule.
- `to_port` - (Optional) This can be used with the from port if you want to define multiple ports. Example from port 8085 to port 8090
- `notes` - (Optional) A simple note for a given firewall rule

Attributes Reference

The following attributes are exported:

- `id` - The given ID for a firewall rule.
- `firewall_group_id` - The firewall group that the firewall rule belongs to.
- `protocol` - The type of protocol for this firewall rule. Possible values (icmp, tcp, udp, gre)
- `network` - IP address that is defined for this rule.

- `from_port` - Port that is defined for this rule.
- `to_port` - This can be used with the from port if you want to define multiple ports. Example from port 8085 to port 8090
- `notes` - A simple note for a given firewall rule
- `ip_type` - The type of ip this rule is - may be either v4 or v6.

vultr_iso_private

Provides a Vultr ISO file resource. This can be used to create, read, and delete ISO files on your Vultr account.

Example Usage

Create a new ISO

```
resource "vultr_iso_private" "my_iso" {  
  url = "http://dl-cdn.alpinelinux.org/alpine/v3.9/releases/x86_64/alpine-virt-3.9.3-x86_64.iso"  
}
```

Argument Reference

The following arguments are supported:

- `url` - (Required) URL pointing to the ISO file.

Attributes Reference

The following attributes are exported:

- `id` - ID of the ISO.
- `url` - URL pointing to the ISO file.
- `date_created` - The date the ISO was created.
- `filename` - The ISO filename.
- `size` - The ISO size in bytes.
- `md5sum` - The md5 hash of the ISO file.
- `sha512sum` - The sha512 hash of the ISO file.
- `status` - The status of the ISO file.

Import

ISOs can be imported using the ISO ID , e.g.

```
terraform import vultr_iso_private.my_iso 2349859
```

vultr_network

Provides a Vultr private network resource. This can be used to create, read, and delete private networks on your Vultr account.

Example Usage

Create a new private network with an automatically generated CIDR block:

```
resource "vultr_network" "my_network" {
  description = "my private network"
  region_id = 6
}
```

Create a new private network with a specified CIDR block:

```
resource "vultr_network" "my_network" {
  description = "my private network"
  region_id = 6
  cidr_block = "10.0.0.0/24"
}
```

Argument Reference

The following arguments are supported:

- `region_id` - (Required) The region ID that you want the network to be created in.
- `description` - (Optional) The description you want to give your network.
- `cidr_block` - (Optional) The IPv4 subnet and subnet mask to be used when attaching servers to this network.

Attributes Reference

The following attributes are exported:

- `id` - ID of the network.
- `region_id` - The region ID that the network operates in.
- `description` - The description of the network.
- `cidr_block` - The IPv4 subnet and subnet mask to be used when attaching servers to this network.
- `date_created` - The date that the network was added to your Vultr account.

Import

Networks can be imported using the network `NETWORKID`, e.g.

```
terraform import vultr_network.my_network net539626f0798d7
```

vultr_reserved_ip

Provides a Vultr reserved IP resource. This can be used to create, read, modify, and delete reserved IP addresses on your Vultr account.

Example Usage

Create a new reserved IP:

```
resource "vultr_reserved_ip" "my_reserved_ip" {
  label = "my-reserved-ip"
  region_id = 6
  ip_type = "v4"
}
```

Attach a reserved IP to a server:

```
resource "vultr_reserved_ip" "my_reserved_ip" {
  label = "my-reserved-ip"
  region_id = 6
  ip_type = "v4"
  attached_id = "923483"
}
```

Argument Reference

The following arguments are supported:

- `region_id` - (Required) The region ID that you want the reserved IP to be created in.
- `ip_type` - (Required) The type of reserved IP that you want. Either "v4" or "v6".
- `label` - (Optional) The label you want to give your reserved IP.
- `attached_id` - (Optional) The VPS ID you want this reserved IP to be attached to.

Attributes Reference

The following attributes are exported:

- `id` - ID of the reserved IP.
- `region_id` - The region ID (DCID in the Vultr API) that this reserved IP belongs to.
- `ip_type` - The reserved IP's type.
- `label` - The reserved IP's label.

- `attached_id` - The ID of the server the reserved IP is attached to.
- `subnet` - The reserved IP's subnet.
- `subnet_size` - The reserved IP's subnet size.

Import

Reserved IPs can be imported using the reserved IP SUBID , e.g.

```
terraform import vultr_reserved_ip.my_reserved_ip 1313044
```

vultr_server

Provides a Vultr server resource. This can be used to create, read, modify, and delete servers on your Vultr account.

Example Usage

Create a new server:

```
resource "vultr_server" "my_server" {
  plan_id = "201"
  region_id = "6"
  os_id = "167"
}
```

Create a new server with options:

```
resource "vultr_server" "my_server" {
  plan_id = "201"
  region_id = "6"
  os_id = "167"
  label = "my-server-label"
  tag = "my-server-tag"
  hostname = "my-server-hostname"
  user_data = '{"foo': true}"
  enable_ipv6 = true
  auto_backup = true
  ddos_protection = true
  notify_activate = false
}
```

Argument Reference

The following arguments are supported:

- `region_id` - (Required) The ID of the region that the server is to be created in.
- `plan_id` - (Required) The ID of the plan that you want the server to subscribe to.
- `os_id` - (Optional) The ID of the operating system to be installed on the server.
- `iso_id` - (Optional) The ID of the ISO file to be installed on the server.
- `app_id` - (Optional) The ID of the Vultr application to be installed on the server.
- `snapshot_id` - (Optional) The ID of the Vultr snapshot that the server will restore for the initial installation.
- `script_id` - (Optional) The ID of the startup script you want added to the server.
- `firewall_group_id` - (Optional) The ID of the firewall group to assign to the server.
- `network_ids` - (Optional) A list of private network IDs to be attached to the server.

- `ssh_key_ids` - (Optional) A list of SSH key IDs to apply to the server on install (only valid for Linux/FreeBSD).
- `user_data` - (Optional) Generic data store, which some provisioning tools and cloud operating systems use as a configuration file. It is generally consumed only once after an instance has been launched, but individual needs may vary.
- `auto_backup` - (Optional) Whether automatic backups will be enabled for this server (these have an extra charge associated with them).
- `enable_ipv6` - (Optional) Whether the server has IPv6 networking activated.
- `enable_private_network` - (Optional) Whether the server has private networking support enabled.
- `notify_activate` - (Optional) Whether an activation email will be sent when the server is ready.
- `ddos_protection` - (Optional) Whether DDOS protection will be enabled on the server (there is an additional charge for this).
- `hostname` - (Optional) The hostname to assign to the server.
- `tag` - (Optional) The tag to assign to the server.
- `label` - (Optional) A label for the server.

Attributes Reference

The following attributes are exported:

- `id` - ID of the server.
- `region_id` - The ID of the region that the server is in.
- `os` - The string description of the operating system installed on the server.
- `ram` - The amount of memory available on the server in MB.
- `disk` - The description of the disk(s) on the server.
- `main_ip` - The server's main IP address.
- `vps_cpu_count` - The number of virtual CPUs available on the server.
- `location` - The physical location of the server.
- `default_password` - The server's default password.
- `date_created` - The date the server was added to your Vultr account.
- `pending_charges` - Charges pending for this server's subscription in USD.
- `cost_per_month` - The server's cost per month in USD.
- `current_bandwidth` - The server's current bandwidth usage in GB.
- `allowed_bandwidth` - The server's allowed bandwidth usage in GB.
- `netmask_v4` - The server's IPv4 netmask.

- `gateway_v4` - The server's IPv4 gateway.
- `status` - The status of the server's subscription.
- `power_status` - Whether the server is powered on or not.
- `server_state` - A more detailed server status (none, locked, installingbooting, isomounting, ok).
- `v6_networks` - A list of the server's IPv6 networks.
- `internal_ip` - The server's internal IP address.
- `kvm_url` - The server's current KVM URL. This URL will change periodically. It is not advised to cache this value.
- `plan_id` - The ID of the plan that server is subscribed to.
- `os_id` - The ID of the operating system installed on the server.
- `iso_id` - The ID of the ISO file installed on the server.
- `app_id` - The ID of the Vultr application installed on the server.
- `snapshot_id` - The ID of the Vultr snapshot that the server was restored from.
- `script_id` - The ID of the startup script that was added to the server.
- `firewall_group_id` - The ID of the firewall group assigned to the server.
- `network_ids` - A list of private network IDs attached to the server.
- `ssh_key_ids` - A list of SSH key IDs applied to the server on install.
- `user_data` - Base64 encoded generic data store, which some provisioning tools and cloud operating systems use as a configuration file. It is generally consumed only once after an instance has been launched, but individual needs may vary.
- `auto_backup` - Whether automatic backups are enabled for this server.
- `enable_ipv6` - Whether the server has IPv6 networking activated.
- `enable_private_network` - Whether the server has private networking support enabled.
- `notify_activate` - Whether an activation email was sent when the server was ready.
- `ddos_protection` - Whether DDOS protection is enabled on the server.
- `hostname` - The hostname assigned to the server.
- `tag` - The tag assigned to the server.
- `label` - A label for the server.

Import

Servers can be imported using the server `SUBID`, e.g.

```
terraform import vultr_server.my_server 1312965
```


vultr_snapshot_from_url

Provides a Vultr Snapshots from URL resource. This can be used to create, read, modify, and delete Snapshots from URL.

Example Usage

Create a new Snapshots from URL

```
resource "vultr_snapshot_from_url" "my_snapshot" {  
  url = "http://dl-cdn.alpinelinux.org/alpine/v3.9/releases/x86_64/alpine-virt-3.9.1-x86_64.iso"  
}
```

Argument Reference

The following arguments are supported:

- `url` - (Required) URL of the given resource you want to create a snapshot from.

Attributes Reference

The following attributes are exported:

- `id` - The ID for the given snapshot.
- `description` - The description for the given snapshot.
- `url` - The url from where the raw image was used to create the snapshot.
- `date_created` - The date the snapshot was created.
- `size` - The size of the snapshot in Bytes.
- `status` - The status for the given snapshot.
- `os_id` - The os id which the snapshot is associated with.
- `app_id` - The app id which the snapshot is associated with.

Import

Snapshots from Url can be imported using the Snapshot `SNAPSHOTID`, e.g.

```
terraform import vultr_snapshot_from_url.my_snapshot 9735ced831ed2
```

vultr_snapshot

Provides a Vultr Snapshot resource. This can be used to create, read, modify, and delete Snapshot.

Example Usage

Create a new Snapshot

```
resource "vultr_server" "my_server" {
  label = "my_server"
  region_id = "1"
  plan_id = 201
  os_id = 147
}
resource "vultr_snapshot" "my_snapshot" {
  vps_id      = "${vultr_server.snap.id}"
  description = "my servers snapshot"
}
```

Argument Reference

The following arguments are supported:

- `vps_id` - (Required) ID of a given server that you want to create a snapshot from.
- `description` - (Optional) The description for the given snapshot.

Attributes Reference

The following attributes are exported:

- `id` - The ID for the given snapshot.
- `vps_id` - The ID of the server that the snapshot was created from.
- `description` - The description for the given snapshot.
- `date_created` - The date the snapshot was created.
- `size` - The size of the snapshot in Bytes.
- `status` - The status for the given snapshot.
- `os_id` - The os id which the snapshot is associated with.
- `app_id` - The app id which the snapshot is associated with.

Import

Snapshots can be imported using the Snapshot `SNAPSHOTID` , e.g.

```
terraform import vultr_snapshot_url.my_snapshot 9735ced831ed2
```

vultr_ssh_key

Provides a Vultr SSH key resource. This can be used to create, read, modify, and delete SSH keys.

Example Usage

Create an SSH key

```
resource "vultr_ssh_key" "my_ssh_key" {
  name = "my-ssh-key"
  ssh_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCyVGaw1PuEL98f4/7Kq309ZIVDw20FOSXAFVqiLSFNkHlefM1iMtPe
qsIBp2t9cbGUf55xNDULz/bD/4BCV43yZ5lh0cUYuXALg9NI29ui7PEGREXjSpNwUD6ceN/78YOK41KAcecq+SS0bJ4b4amKZIJG3JWmD
Kljtv1dmSBCrTmEAQa0orxqGGBYmZS7NqumRe4lav5r6w0s80ACMANE1ejkeZsGFzJFNqvr5DuHdDL5FAudW23me3BDmrM9ifUzzjl1Jw
ku3bnRaCcjaxH8oTumt1a00mWci/1qUlaVFft085yvVq7KZbF20PPbl+erDW91+EZ2FgEi+v1/CSJ5 your_username@hostname"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) The name/label of the SSH key.
- `ssh_key` - (Required) The public SSH key.

Attributes Reference

The following attributes are exported:

- `id` - The ID of the SSH key.
- `name` - The name/label of the SSH key.
- `ssh_key` - The public SSH key.
- `date_created` - The date the SSH key was added to your Vultr account.

Import

SSH keys can be imported using the SSH key `SSHKEYID`, e.g.

```
terraform import vultr_ssh_key.my_key 541b4960f23bd
```

vultr_startup_script

Provides a Vultr Startup Script resource. This can be used to create, read, modify, and delete Startup Scripts.

Example Usage

Create a new Startup Script

```
resource "vultr_startup_script" "my_script" {  
  name = "man_run_docs"  
  script = "echo $PATH"  
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the given script.
- `script` - (Required) Contents of the startup script.
- `type` - (Optional) Type of startup script. Default is boot.

Attributes Reference

The following attributes are exported:

- `id` - ID of the script.
- `name` - Name of the given script.
- `date_created` - Date the script was created.
- `date_modified` - Date the script was last modified.
- `type` - The type of startup script this is.
- `script` - The contents of the startup script.

Import

Startup Scripts can be imported using the Startup Scripts `SCRIPTID`, e.g.

```
terraform import vultr_startup_script.my_script 537932
```

vultr_user

Provides a Vultr User resource. This can be used to create, read, modify, and delete Users.

Example Usage

Create a new User without any ACLs

```
resource "vultr_user" "my_user" {
  name = "my user"
  email = "user@vultr.com"
  password = "myP@ssw0rd"
  api_enabled = true
}
```

Create a new User with all ACLs

```
resource "vultr_user" "my_user" {
  name = "my user"
  email = "user@vultr.com"
  password = "myP@ssw0rd"
  api_enabled = true
  acl = [
    "manage_users",
    "subscriptions",
    "provisioning",
    "billing",
    "support",
    "abuse",
    "dns",
    "upgrade",
  ]
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name for this user.
- `email` - (Required) Email for this user.
- `password` - (Required) Password for this user.
- `api_enabled` - (Optional) Whether API is enabled for the user. Default behavior is set to enabled.
- `acl` - (Optional) The access control list for the user.

Attributes Reference

The following attributes are exported: * `id` - ID associated with the user. * `name` - Name for this user. * `email` - Email for this user. * `api_enabled` - Whether API is enabled for the user. * `api_key` - API Key that is assigned to this user.

Import

Users can be imported using the User `USERID`, e.g.

```
terraform import vultr_user.myuser cbe5ced2ae716
```