Scaleway Provider

The Scaleway provider is used to manage Scaleway resources.

Use the navigation to the left to read about the available resources.

Example Usage

Here is an example that will setup the following: + A Server. + An IP Address. + A security group.

(create this as sl.tf and run terraform commands from this directory):

```
provider "scaleway" {
  organization = "<YOUR-ORGANIZATION-ID>"
  token        = "<YOUR-SECRET-TOKEN>"
  region       = "par1"
}

resource "scaleway_ip" "ip" {
  server = "${scaleway_server.test.id}"
}

resource "scaleway_server" "test" {
  name  = "test"
  image = "aecaed73-51a5-4439-a127-6d8229847145"
  type  = "C2S"
}

resource "scaleway_volume" "test" {
  name       = "test"
  size_in_gb = 50
  type       = "l_ssd"
}

resource "scaleway_volume_attachment" "test" {
  server = "${scaleway_server.test.id}"
  volume = "${scaleway_volume.test.id}"
}

resource "scaleway_security_group" "http" {
  name        = "http"
  description = "allow HTTP and HTTPS traffic"
}

resource "scaleway_security_group_rule" "http_accept" {
  security_group = "${scaleway_security_group.http.id}"

  action    = "accept"
  direction = "inbound"
  ip_range  = "0.0.0.0/0"
  protocol  = "TCP"
  port      = 80
}

resource "scaleway_security_group_rule" "https_accept" {
  security_group = "${scaleway_security_group.http.id}"

  action    = "accept"
  direction = "inbound"
  ip_range  = "0.0.0.0/0"
  protocol  = "TCP"
  port      = 443
}
```

You'll need to provide your Scaleway **organization ID** and a **secret token**. Both are UUIDs.

Your **organization ID** can be found in the *Account* tab of the Scaleway control panel. It is labeled "Organization ID". Alternatively, if you already have a **secret token** you can issue a request to the Scaleway API directly and query for your **organization ID**: `shell $ curl https://account.scaleway.com/organizations -H "X-Auth-Token: <YOUR-SECRET-TOKEN>"`

A **secret token** can be generated by visiting the *Credentials* tab of the Scaleway control panel and looking in the *Tokens* section at the bottom of the page. Each listed "Secret Key" (if any tokens have already been created) can be used as your **secret token**. Since secret keys are only revealed one time (when the token is first created) you might need to create a new token to get a new "Secret Key". Giving each token a friendly-name is recommended.

If you do not want to put credentials in your configuration file, you can leave them out:

```
provider "scaleway" {
  region       = "par1"
}
```

...and instead set these environment variables:

- **SCALEWAY_ORGANIZATION**: Your Scaleway `organization` ID
- **SCALEWAY_TOKEN**: Your API access `token`, generated by you
- **SCALEWAY_REGION**: The Scaleway region

Volume usage

You can add volumes to baremetal instances. The minimal size of increment is 50GB and you can add at most 15 different volumes on an instance.

Additional volumes cannot be added to virtual cloud servers.

Check out the list of different instances on the pricing page (https://www.scaleway.com/pricing).

scaleway_bootscript

Use this data source to get the ID of a registered Bootscript for use with the `scaleway_server` resource.

Example Usage

```
data "scaleway_bootscript" "debug" {
  architecture = "arm"
  name_filter  = "Rescue"
}
```

Argument Reference

- `architecture` - (Optional) any supported Scaleway architecture, e.g. `x86_64`, `arm`
- `name_filter` - (Optional) Regexp to match Bootscript name by
- `name` - (Optional) Exact name of desired Bootscript

Attributes Reference

`id` is set to the ID of the found Bootscript. In addition, the following attributes are exported:

- `architecture` - architecture of the Bootscript, e.g. `arm` or `x86_64`
- `organization` - uuid of the organization owning this Bootscript
- `public` - is this a public bootscript
- `boot_cmd_args` - command line arguments used for booting
- `dtb` - path to Device Tree Blob detailing hardware information
- `initrd` - URL to initial ramdisk content
- `kernel` - URL to used kernel

scaleway_image

Use this data source to get the ID of a registered Image for use with the `scaleway_server` resource.

Example Usage

```
data "scaleway_image" "ubuntu" {
  architecture = "arm"
  name         = "Ubuntu Precise"
}

resource "scaleway_server" "base" {
  name  = "test"
  image = "${data.scaleway_image.ubuntu.id}"
  type  = "C1"
}
```

Argument Reference

- `architecture` - (Required) any supported Scaleway architecture, e.g. `x86_64`, `arm`
- `name_filter` - (Optional) Regexp to match Image name by
- `name` - (Optional) Exact name of desired Image
- `most_recent` - (Optional) Return most recent image if multiple exist. Can not be used together with name_filter.

Attributes Reference

`id` is set to the ID of the found Image. In addition, the following attributes are exported:

- `architecture` - architecture of the Image, e.g. `arm` or `x86_64`
- `organization` - uuid of the organization owning this Image
- `public` - is this a public image
- `creation_date` - date when image was created

scaleway_security_group

Gets information about a Security Group.

Example Usage

```
data "scaleway_security_group" "test" {
  name = "my-security-group"
}
```

Argument Reference

- `name` - (Required) Exact name of desired Security Group

Attributes Reference

`id` is set to the ID of the found Image. In addition, the following attributes are exported:

- `description` - description of the security group
- `enable_default_security` - have default security group rules been added to this security group?

scaleway_volume

Gets information about a Volume.

Example Usage

```
data "scaleway_volume" "data" {
  name = "data"
}

resource "scaleway_server" "test" {
  # ...
}

resource "scaleway_volume_attachment" "data" {
  server = "${scaleway_server.test.id}"
  volume = "${scaleway_volume.data.id}"
}
```

Argument Reference

- `name` - (Required) Exact name of the Volume.

Attributes Reference

`id` is set to the ID of the found Volume. In addition, the following attributes are exported:

- `size_in_gb` - (Required) size of the volume in GB
- `type` - The type of volume this is, such as `l_ssd`.
- `server` - The ID of the Server which this Volume is currently attached to.

scaleway_bucket

Creates Scaleway object storage buckets.

Example Usage

```
resource "scaleway_bucket" "test" {
  name = "sample-bucket"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) Name of the Scaleway objectstorage bucket

Attributes Reference

The following attributes are exported:

- `name` - Name of the resource

Import

Instances can be imported using the `name`, e.g.

```
$ terraform import scaleway_bucket.releases releases
```

scaleway_ip

Provides IPs for servers. This allows IPs to be created, updated and deleted. For additional details please refer to API documentation (https://developer.scaleway.com/#ips).

Example Usage

```
resource "scaleway_ip" "test_ip" {}
```

Argument Reference

The following arguments are supported:

- `server` - (Optional) ID of server to associate IP with
- `reverse` - (Deprecated) Please us the scaleway_ip_reverse_dns resource instead.

Attributes Reference

The following attributes are exported:

- `id` - ID of the new resource
- `ip` - IP of the new resource
- `server` - ID of the associated server resource
- `reverse` - reverse DNS setting of the IP resource

Import

Instances can be imported using the `id`, e.g.

```
$ terraform import scaleway_ip.jump_host 5faef9cd-ea9b-4a63-9171-9e26bec03dbc
```

scaleway_ip_reverse_dns

Provides reverse DNS settings for IPs. For additional details please refer to API documentation (https://developer.scaleway.com/#ips).

Example Usage

```
resource "scaleway_ip" "test_service" {}

resource "scaleway_ip_reverse_dns" "google" {
  ip = "${scaleway_ip.test_service.id}"
  reverse = "test_service.awesome-corp.com"
}
```

Argument Reference

The following arguments are supported:

- `ip` - (Required) ID or Address of IP
- `reverse` - (Required) Reverse DNS of the IP

Attributes Reference

The following attributes are exported:

- `id` - ID of the new resource
- `reverse` - reverse DNS setting of the IP resource

scaleway_security_group

Provides security groups. This allows security groups to be created, updated and deleted. For additional details please refer to API documentation (https://developer.scaleway.com/#security-groups).

Example Usage

```
resource "scaleway_security_group" "test" {
  name                    = "test"
  description             = "test"
  enable_default_security = true
  stateful                = true
  inbound_default_policy  = "accept"
  outbound_default_policy = "drop"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) name of security group
- `description` - (Required) description of security group
- `enable_default_security` - (Optional) default: true. Add default security group rules
- `stateful` - (Optional) default: false. Mark the security group as stateful. Note that stateful security groups can not be associated with bare metal servers
- `inbound_default_policy` - (Optional) default policy for inbound traffic. Can be one of accept or drop
- `outbound_default_policy` - (Optional) default policy for outbound traffic. Can be one of accept or drop

Field `name`, `description` are editable.

Attributes Reference

The following attributes are exported:

- `id` - id of the new resource

Import

Instances can be imported using the `id`, e.g.

```
$ terraform import scaleway_security_group.test 5faef9cd-ea9b-4a63-9171-9e26bec03dbc
```

scaleway_security_group_rule

Provides security group rules. This allows security group rules to be created, updated and deleted. For additional details please refer to API documentation (https://developer.scaleway.com/#security-groups-manage-rules).

Example Usage

```
resource "scaleway_security_group" "test" {
  name        = "test"
  description = "test"
}

resource "scaleway_security_group_rule" "smtp_drop_1" {
  security_group = "${scaleway_security_group.test.id}"

  action    = "accept"
  direction = "inbound"
  ip_range  = "0.0.0.0/0"
  protocol  = "TCP"
  port      = 25
}
```

Argument Reference

The following arguments are supported:

- `security_group` - (Required) the security group which should be associated with this rule
- `action` - (Required) action of rule (`accept`, `drop`)
- `direction` - (Required) direction of rule (`inbound`, `outbound`)
- `ip_range` - (Required) ip_range of rule
- `protocol` - (Required) protocol of rule (`ICMP`, `TCP`, `UDP`)
- `port` - (Optional) port of the rule

Fields `action`, `direction`, `ip_range`, `protocol`, `port` are editable.

Attributes Reference

The following attributes are exported:

- `id` - id of the new resource

scaleway_server

Provides servers. This allows servers to be created, updated and deleted. For additional details please refer to API documentation (https://developer.scaleway.com/#servers).

Example Usage

```
resource "scaleway_server" "test" {
  name  = "test"
  image = "5faef9cd-ea9b-4a63-9171-9e26bec03dbc"
  type  = "VC1M"

  volume {
    size_in_gb = 20
    type       = "l_ssd"
  }
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) name of server
- `image` - (Required) base image of server
- `type` - (Required) type of server
- `bootscript` - (Optional) server bootscript
- `boot_type` - (Optional) the boot mechanism for this server. Possible values include `local` and `bootscript`
- `tags` - (Optional) list of tags for server
- `enable_ipv6` - (Optional) enable ipv6
- `dynamic_ip_required` - (Optional) make server publicly available
- `public_ip` - (Optional) set a public ip previously created (a real ip is expected here, not its resource id)
- `security_group` - (Optional) assign security group to server
- `volume` - (Optional) attach additional volumes to your instance (see below)
- `public_ipv6` - (Read Only) if `enable_ipv6` is set this contains the ipv6 address of your instance
- `state` - (Optional) allows you to define the desired state of your server. Valid values include (`stopped`, `running`)
- `cloudinit` - (Optional) allows you to define cloudinit script for this server
- `state_detail` - (Read Only) contains details from the scaleway API the state of your instance

Field `name`, `type`, `tags`, `dynamic_ip_required`, `security_group` are editable.

Volume

You can attach additional volumes to your instance, which will share the lifetime of your `scaleway_server` resource.

Warning: Using the `volume` attribute does not modify the System Volume provided default with every `scaleway_server` instance. Instead it adds additional volumes to the server instance.

Warning: Some instance types require an additional volume to work. This includes for example *START-1M* and *VC1M*. If you run into this issue add an additional volume of the specified size.

The `volume` mapping supports the following:

- `type` - (Required) The type of volume. Can be `"l_ssd"`
- `size_in_gb` - (Required) The size of the volume in gigabytes.

Attributes Reference

The following attributes are exported:

- `id` - id of the new resource
- `private_ip` - private ip of the new resource
- `public_ip` - public ip of the new resource

Import

Instances can be imported using the `id`, e.g.

```
$ terraform import scaleway_server.web 5faef9cd-ea9b-4a63-9171-9e26bec03dbc
```

scaleway_ssh_key

Manages user SSH Keys to access servers provisioned on scaleway. For additional details please refer to API documentation (https://developer.scaleway.com/#users-user-get).

Example Usage

```
resource "scaleway_ssh_key" "test" {
    key = "ssh-rsa <some-key>"
}
```

Argument Reference

The following arguments are supported:

- `key` - (Required) public key of the SSH key to be added

Attributes Reference

The following attributes are exported:

- `id` - fingerprint of the SSH key

Import

Instances can be imported using the `id`, e.g.

```
$ terraform import scaleway_ssh_key.awesome "d1:4c:45:59:a8:ee:e6:41:10:fb:3c:3e:54:98:5b:6f"
```

scaleway_token

Provides Tokens for scaleway API access. For additional details please refer to API documentation (https://developer.scaleway.com/#tokens-tokens-post).

Example Usage

```
resource "scaleway_token" "karls_token" {
    expires = false
    description = "karls scaleway access: karl@company.com"
}
```

Argument Reference

The following arguments are supported:

- `expires` - (Optional) Define if the token should automatically expire or not
- `email` - (Optional) Scaleway account email. Defaults to registered account
- `password` - (Optional) Scaleway account password. Required for cross-account token management
- `description` - (Optional) Token description

Attributes Reference

The following attributes are exported:

- `id` - Token ID - can be used to access scaleway API
- `access_key` - Token Access Key
- `secret_key` - Token Secret Key
- `creation_ip` - IP used to create the token
- `expiration_date` - Expiration date of token, if expiration is requested

Import

Instances can be imported using the `id`, e.g.

```
$ terraform import scaleway_token.karls_token 5faef9cd-ea9b-4a63-9171-9e26bec03dbc
```

scaleway_user_data

Provides user data for servers. For additional details please refer to API documentation (https://developer.scaleway.com/#user-data).

Example Usage

```
resource "scaleway_server" "base" {
  name = "test"
  # ubuntu 14.04
  image = "5faef9cd-ea9b-4a63-9171-9e26bec03dbc"
  type = "C1"
  state = "stopped"
}

resource "scaleway_user_data" "gcp" {
    server = "${scaleway_server.base.id}"
    key = "gcp_username"
    value = "supersecret"
}
```

Argument Reference

The following arguments are supported:

- `server` - (Required) ID of server to associate the user data with
- `key` - (Required) The key of the user data object
- `value` - (Required) The value of the user data object

Import

Instances can be imported using the `id`, e.g.

```
$ terraform import scaleway_user_data.gcp userdata-<server-id>-<key>
```

scaleway_volume

Provides volumes. This allows volumes to be created, updated and deleted. For additional details please refer to API documentation (https://developer.scaleway.com/#volumes).

Example Usage

```
resource "scaleway_server" "test" {
  name    = "test"
  image   = "aecaed73-51a5-4439-a127-6d8229847145"
  type    = "C2S"
  volumes = ["${scaleway_volume.test.id}"]
}

resource "scaleway_volume" "test" {
  name       = "test"
  size_in_gb = 20
  type       = "l_ssd"
}
```

Argument Reference

The following arguments are supported:

- `name` - (Required) name of volume
- `size_in_gb` - (Required) size of the volume in GB
- `type` - (Required) type of volume

Attributes Reference

The following attributes are exported:

- `id` - id of the new resource
- `server` - (Read Only) the `scaleway_server` instance which has this volume mounted right now

Import

Instances can be imported using the `id`, e.g.

```
$ terraform import scaleway_volume.test 5faef9cd-ea9b-4a63-9171-9e26bec03dbc
```

scaleway_volume_attachment

This allows volumes to be attached to servers.

> **Warning:** Attaching volumes requires the servers to be powered off. This will lead to downtime if the server is already in use.

Example Usage

```
resource "scaleway_server" "test" {
  name  = "test"
  image = "aecaed73-51a5-4439-a127-6d8229847145"
  type  = "C2S"
}

resource "scaleway_volume" "test" {
  name       = "test"
  size_in_gb = 20
  type       = "l_ssd"
}

resource "scaleway_volume_attachment" "test" {
  server = "${scaleway_server.test.id}"
  volume = "${scaleway_volume.test.id}"
}
```

Argument Reference

The following arguments are supported:

- `server` - (Required) id of the server
- `volume` - (Required) id of the volume to be attached

Attributes Reference

The following attributes are exported:

- `id` - id of the new resource