Scaleway Provider

The Scaleway provider is used to manage Scaleway resources. The provider needs to be configured with the proper credentials before it can be used.

This is the documentation for the version `>= 1.11.0` of the provider. If you come from `< v1.11.0`, checkout to migration guide (/docs/providers/scaleway/guides/migration_guide_v2.html).

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

Example

Here is an example that will setup a web server with an additional volume, a public IP and a security group.

You can test this config by creating a `test.tf` and run terraform commands from this directory:

- Get your Scaleway credentials (https://console.scaleway.com/account/credentials)
- Initialize a Terraform working directory: `terraform init`
- Generate and show the execution plan: `terraform plan`
- Build the infrastructure: `terraform apply`

```
provider "scaleway" {
  access_key = "<SCALEWAY-ACCESS-KEY>"
  secret_key = "<SCALEWAY-SECRET-KEY>"
  organization_id = "<SCALEWAY-ORGANIZATION-ID>"
  zone       = "fr-par-1"
  region     = "fr-par"
}

resource "scaleway_instance_ip" "public_ip" {
  server_id = "${scaleway_instance_server.web.id}"
}

resource "scaleway_instance_volume" "data" {
  size_in_gb = 100
}

resource "scaleway_instance_security_group" "www" {
  inbound_default_policy = "drop"
  outbound_default_policy = "accept"

  inbound_rule {
    action = "accept"
    port = "22"
    ip = "212.47.225.64"
  }

  inbound_rule {
    action = "accept"
    port = "80"
  }

  inbound_rule {
    action = "accept"
    port = "443"
  }
}

resource "scaleway_instance_server" "web" {
  type = "DEV1-L"
  image = "f974feac-abae-4365-b988-8ec7d1cec10d"

  tags = [ "front", "web" ]

  additional_volume_ids = [ "${scaleway_instance_volume.data.id}" ]

  security_group_id= "${scaleway_instance_security_group.www.id}"
}
```

Authentication

The Scaleway authentication is based on an **access key** and a **secret key**. Since secret keys are only revealed one time (when it is first created) you might need to create a new one in the section "API Tokens" of the Scaleway console (https://console.scaleway.com/account/credentials). Click on the "Generate new token" button to create them. Giving it a friendly-name is recommended.

The Scaleway provider offers three ways of providing these credentials. The following methods are supported, in this priority order:

1. Static credentials
2. Environment variables
3. Shared configuration file

Static credentials

Warning: Hard-coding credentials into any Terraform configuration is not recommended, and risks secret leakage should this file ever be committed to a public version control system.

Static credentials can be provided by adding `access_key` and `secret_key` attributes in-line in the Scaleway provider block:

Example:

```
provider "scaleway" {
  access_key = "my-access-key"
  secret_key = "my-secret-key"
}
```

Environment variables

You can provide your credentials via the `SCW_ACCESS_KEY`, `SCW_SECRET_KEY` environment variables.

Example:

```
provider "scaleway" {}
```

Usage:

```
$ export SCW_ACCESS_KEY="my-access-key"
$ export SCW_SECRET_KEY="my-secret-key"
$ terraform plan
```

Shared configuration file

It is a YAML configuration file shared between the majority of the Scaleway developer tools (https://developers.scaleway.com/en/community-tools/#official-repos). Its default location is `$HOME/.config/scw/config.yaml` (`%USERPROFILE%/.config/scw/config.yaml` on Windows). If it fails to detect credentials inline, or in the environment, Terraform will check this file.

You can optionally specify a different location with `SCW_CONFIG_PATH` environment variable. You can find more information about this configuration in the documentation (https://github.com/scaleway/scaleway-sdk-go/blob/master/scw/README.md#scaleway-config).

Arguments Reference

In addition to generic provider arguments (https://www.terraform.io/docs/configuration/providers.html) (e.g. `alias` and `version`), the following arguments are supported in the Scaleway provider block:

- `access_key` - (Optional) The Scaleway access key. It must be provided, but it can also be sourced from the `SCW_ACCESS_KEY` environment variable, or via a shared configuration file, in this priority order.
- `secret_key` - (Optional) The Scaleway secert key. It must be provided, but it can also be sourced from the `SCW_SECRET_KEY` environment variable, or via a shared configuration file, in this priority order.
- `organization_id` - (Optional) The organization ID that will be used as default value for all resources. It can also be sourced from the `SCW_DEFAULT_ORGANIZATION_ID` environment variable (https://github.com/scaleway/scaleway-sdk-go/blob/master/scw/README.md#environment-variables), or via a shared configuration file (https://github.com/scaleway/scaleway-sdk-go/blob/master/scw/README.md#scaleway-config), in this priority order.
- `region` - (Optional) The region (/docs/providers/scaleway/guides/regions_and_zones.html#regions) that will be used as default value for all resources. It can also be sourced from the `SCW_DEFAULT_REGION` environment variable (https://github.com/scaleway/scaleway-sdk-go/blob/master/scw/README.md#environment-variables), or via a shared configuration file (https://github.com/scaleway/scaleway-sdk-go/blob/master/scw/README.md#scaleway-config), in this priority order.
- `zone` - (Optional) The zone (/docs/providers/scaleway/guides/regions_and_zones.html#zones) that will be used as default value for all resources. It can also be sourced from the `SCW_DEFAULT_ZONE` environment variable (https://github.com/scaleway/scaleway-sdk-go/blob/master/scw/README.md#environment-variables), or via a shared configuration file (https://github.com/scaleway/scaleway-sdk-go/blob/master/scw/README.md#scaleway-config), in this priority order.

Scaleway S3-compatible

Scaleway object storage (https://www.scaleway.com/object-storage/) can be used to store your Terraform state. Configure your backend as:

```
terraform {
  backend "s3" {
    bucket      = "terraform_state"
    key         = "my_state.tfstate"
    region      = "fr-par"
    endpoint    = "https://s3.fr-par.scw.cloud"
    access_key = "my-access-key"
    secret_key = "my-secret-key"
    skip_credentials_validation = true
    skip_region_validation = true
  }
}
```

Beware as no locking mechanism are yet supported. Using scaleway object storage as terraform backend is not suitable if you work in a team with a risk of simultaneous access to the same plan.

scaleway_account_ssh_key

Use this data source to get SSH key information based on its ID or name.

Example Usage

```
// Get info by SSH key name
data "scaleway_account_ssh_key" "my_key" {
  name  = "my-key-name"
}

// Get info by SSH key id
data "scaleway_account_ssh_key" "my_key" {
  ssh_key_id = "11111111-1111-1111-1111-111111111111"
}
```

Argument Reference

- `name` - The SSH key name. Only one of `name` and `ssh_key_id` should be specified.
- `ssh_key_id` - The SSH key id. Only one of `name` and `ssh_key_id` should be specified.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the organization the server is associated with.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the server.
- `public_key` - The SSH public key string

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.

Migrating from v1 to v2

Note: The version 2 is not released yet but versions `v1.11+` allow you to do a smooth migration to the `v2`. In other words, there will be no breaking change between `v1.11+` and `v2`. The `v2` roadmap is available here (https://github.com/terraform-providers/terraform-provider-scaleway/issues/125).

This page guides you through the process of migrating your version 1 resources to their version 2 equivalent. To prepare the launch of all new Scaleway products, we completely changed the naming of all resources (as well as their attributes) in version 2 of the Terraform provider.

Provider

Version configuration

Note: Before upgrading to `v2+`, it is recommended to upgrade to the most recent `1.X` version of the provider (`v1.11.0`) and ensure that your environment successfully runs `terraform plan` (https://www.terraform.io/docs/commands/plan.html) without unexpected change or deprecation notice.

It is recommended to use version constraints when configuring Terraform providers (https://www.terraform.io/docs/configuration/providers.html#version-provider-versions). If you are following these recommendation, update the version constraints in your Terraform configuration and run `terraform init` (https://www.terraform.io/docs/commands/init.html) to download the new version.

Update to latest `1.X` version:

```
provider "scaleway" {
  # ... other configuration ...

  version = "~> 1.11"
}
```

Update to latest 2.X version:

```
provider "scaleway" {
  # ... other configuration ...

  version = "~> 2.0"
}
```

Provider configuration

In order to unify configuration management across all scaleway developer tools, we changed the configuration management in version 2.

Below you find an overview of changes in the provider config:

Old provider attribute	New provider attribute
`access_key`	`access_key`
`token`	`secret_key`
`organization`	`organization_id`

Important: `access_key` should now only be used for your access key (e.g. `SCWZFD9BPQ4TZ14SM1YS`). Your secret key (previously known as *token*) must be set in `secret_key` (`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`).

Below you find an overview of the changes in environment variables:

Old env variable	New env variable
`SCALEWAY_ACCESS_KEY`	`SCW_ACCESS_KEY`
`SCALEWAY_TOKEN`	`SCW_SECRET_KEY`
`SCALEWAY_ORGANIZATION`	`SCW_DEFAULT_ORGANIZATION_ID`
`SCALEWAY_REGION`	`SCW_DEFAULT_REGION` and `SCW_DEFAULT_ZONE`
`SCW_TLSVERIFY`	`SCW_INSECURE`
`SCW_ORGANIZATION`	`SCW_DEFAULT_ORGANIZATION_ID`
`SCW_REGION`	`SCW_DEFAULT_REGION`
`SCW_TOKEN`	`SCW_SECRET_KEY`

Important: `SCALEWAY_ACCESS_KEY` was changed to `SCW_ACCESS_KEY`. This should be your access key (e.g. `SCWZFD9BPQ4TZ14SM1YS`). Your secret key (previously known as *token*) must be set in `SCW_SECRET_KEY` (`xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx`).

Resources

All resources are from now on prefixed by `scaleway`, their product category and their product name (`scaleway_{product-category-name}_{product-name}_{resource-name}`). For instances an S3 bucket belongs to the `Storage` product category and is a resource of the `Object` product. Hence it is named: `scaleway_object_bucket`.

Instance

All the old instance resources have been regrouped under a new name: `Instance`. This means that all old instance resources are now prefixed with `scaleway_instance_`.

Renamed: `scaleway_server -> scaleway_instance_server`

`scaleway_server` was renamed to `scaleway_instance_server`.

In version 1, attachments of volumes where done on the volume resource. But from now on, this is done on the `scaleway_instance_server` resource.

Thus, to create a server with a volume attached:

```
resource "scaleway_instance_volume" "data" {
  size_in_gb = 100
}

resource "scaleway_instance_server" "web" {
  type = "DEV1-L"
  image = "f974feac-abae-4365-b988-8ec7d1cec10d"

  tags = [ "hello", "public" ]

  root_volume {
    delete_on_termination = false
  }

  additional_volume_ids = [ "${scaleway_instance_volume.data.id}" ]
}
```

Renamed: `scaleway_ip -> scaleway_instance_ip`

`scaleway_ip` was renamed to `scaleway_instance_ip` and the attribute `server` was renamed to `server_id`.

```
resource "scaleway_instance_ip" "test_ip" {
  server_id = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

Renamed: `scaleway_volume -> scaleway_instance_volume`

`scaleway_volume` was renamed to `scaleway_instance_volume`. The former attributes can still be used on the new volume resource.

Additionally, from now on, you can also create new volumes based on other volumes or snapshots. For more information check the new volume `scaleway_instance_volume` resource (/docs/providers/scaleway/r/instance_volume.html).

Renamed: `scaleway_ssh_key -> scaleway_account_ssk_key`

`scaleway_ssh_key` was renamed to `scaleway_account_ssk_key` The `key` attribute has been renamed to `public_key`. A `name` required attribute and an `organization_id` optional attribute have been added.

Removed: `scaleway_user_data`

`scaleway_user_data` is now part of the `scaleway_instance_server` resource.

Removed: `scaleway_token`

The `scaleway_token` was removed in version 2.

Tokens should be created in the console.

Removed: `scaleway_ip_reverse_dns`

The `scaleway_ip_reverse_dns` was removed in version 2.

Reverse DNS must be set on the IP resource itself:

```
resource "scaleway_instance_ip" "test_ip" {
  reverse = "scaleway.com"
}
```

Removed: `scaleway_volume_attachment`

The `scaleway_volume_attachment` was removed in version 2.

Volumes can in version 2 only be attached on the server resource. The above example shows how this works.

Storage

Renamed: `scaleway_bucket -> scaleway_object_bucket`

The `scaleway_bucket` was moved to the `object` product in the `storage` product category.

It's behaviour remained the same, but we also added an `acl` attribute (/docs/providers/scaleway/r/object_bucket.html#acl). This attribute takes canned ACLs.

Scaleway Zones and Regions

Scaleway's products are deployed across multiple datacenter in the world.

For technical and legal reasons, some products are splitted by Region or by Availability Zones. When using such product, you can choose the location that better fits your need (country, latency, ...).

Regions

A Region is represented as a Geographical area such as France (Paris: `fr-par`) or the Netherlands (Amsterdam: `nl-ams`). It can contain multiple Availability Zones.

Zones

In order to deploy highly available application, a region can be splitted in many Availability Zones (AZ). Latency between multiple AZ of the same region are low as they have a common network layer.

List of availability zones by regions:

- France - Paris (`fr-par`)
  - `fr-par-1`
  - `fr-par-2`
- The Netherlands - Amsterdam (`nl-ams`)
  - `nl-ams-1`

Resource IDs

To save this notion of regions and zones in the state, all the Terraform IDs of Scaleway contain the region or zone. This is saved in the following format: `{zone|region}/{resource_id}`. Where `zone` or `region` is the place where the resource is created and where `resource_id` is the ID that is used on Scaleway's console/API.

More information regarding zones and regions can be found here (https://developers.scaleway.com/en/quickstart/#region-and-zone).

scaleway_account_ssh_key

Manages user SSH keys to access servers provisioned on Scaleway.

Example Usage

```
resource "scaleway_account_ssh_key" "main" {
    name       = "main"
    public_key = "<YOUR-PUBLIC-SSH-KEY>"
}
```

Arguments Reference

The following arguments are supported:

- `name` - (Required) The name of the SSH key.
- `public_key` - (Required) The public SSH key to be added.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the organization the IP is associated with.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the SSH key.

Import

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

```
$ terraform import scaleway_account_ssh_key.main 11111111-1111-1111-1111-111111111111
```

scaleway_baremetal_server_beta

Creates and manages Scaleway Compute Baremetal servers. For more information, see the documentation (https://developers.scaleway.com/en/products/baremetal/api).

Examples

Basic

```
resource "scaleway_baremetal_server_beta" "base" {
  zone        = "fr-par-2"
  offer_id    = "9eebce52-f7d5-484f-9437-b234164c4c4b"
  os_id       = "d17d6872-0412-45d9-a198-af82c34d3c5c"
  ssh_key_ids = ["f974feac-abae-4365-b988-8ec7d1cec10d"] // get ssh key ids from the console
}
```

Arguments Reference

The following arguments are supported:

- `offer_id` - (Required) The type of the baremetal server. Use this endpoint (https://developers.scaleway.com/en/products/baremetal/api/#get-334154) to find the right offer ID.

Important: Updates to `offer_id` will recreate the server.

- `os_id` - (Required) The UUID of the base image used by the server. Use this endpoint (https://developers.scaleway.com/en/products/baremetal/api/#get-87598a) to find the right OS ID.

Important: Updates to `os_id` will reinstall the server.

- `name` - (Optional) The name of the server.
- `description` - (Optional) A description for the server.
- `ssh_key_ids` - (Defaults to all user SSH keys) List of SSH keys allowed to connect to the server.

Important: Updates to `ssh_key_ids` will reinstall the server.

- `tags` - (Optional) The tags associated with the server.
- `zone` - (Defaults to provider (/docs/providers/scaleway/index.html#zone) `zone`) The zone (/docs/providers/scaleway/guides/regions_and_zones.html#zones) in which the server should be created.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the organization the server is associated with.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the server.

Import

Baremetal servers can be imported using the `{zone}/{id}`, e.g.

```
$ terraform import scaleway_baremetal_server_beta.web fr-par-2/11111111-1111-1111-1111-111111111111
```

scaleway_bucket

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_object_bucket` instead.

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_instance_ip

Creates and manages Scaleway Compute Instance IPs. For more information, see the documentation (https://developers.scaleway.com/en/products/instance/api/#ips-268151).

Example Usage

```
resource "scaleway_instance_ip" "server_ip" {}
```

Arguments Reference

The following arguments are supported:

- `reverse` - (Optional) The reverse DNS for this IP.
- `server_id` - (Optional) The ID of the server you want to attach this resource to.
- `zone` - (Defaults to provider (/docs/providers/scaleway/index.html#zone) `zone`) The zone (/docs/providers/scaleway/guides/regions_and_zones.html#zones) in which the IP should be reserved.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the organization the IP is associated with.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the IP.
- `address` - The IP address.

Import

IPs can be imported using the `{zone}/{id}` , e.g.

```
$ terraform import scaleway_instance_ip.server_ip fr-par-1/11111111-1111-1111-1111-111111111111
```

scaleway_instance_placement_group

Creates and manages Compute Instance Placement Groups. For more information, see the documentation (https://developers.scaleway.com/en/products/instance/api/#placement-groups-d8f653).

Example Usage

```
resource "scaleway_instance_placement_group" "availability_group" {}
```

Arguments Reference

The following arguments are supported:

- `name` - (Optional) The name of the placement group.
- `policy_type` - (Defaults to `low_latency`) The policy type (https://developers.scaleway.com/en/products/instance/api/#placement-groups-d8f653) of the placement group. Possible values are: `low_latency` or `max_availability`.
- `policy_mode` - (Defaults to `optional`) The policy mode (https://developers.scaleway.com/en/products/instance/api/#placement-groups-d8f653) of the placement group. Possible values are: `optional` or `enforced`.
- `zone` - (Defaults to provider (/docs/providers/scaleway/index.html#zone) `zone`) The zone (/docs/providers/scaleway/guides/regions_and_zones.html#zones) in which the placement group should be created.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the project the placement group is associated with.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the placement group.
- `policy_respected` - Is true when the policy is respected.

Import

Placement groups can be imported using the `{zone}/{id}`, e.g.

```
$ terraform import scaleway_instance_placement_group.availability_group fr-par-1/11111111-1111-1111-1111-111111111111
```

scaleway_instance_server

Creates and manages Scaleway Compute Instance security groups. For more information, see the documentation (https://developers.scaleway.com/en/products/instance/api/#security-groups-8d7f89).

Examples

Basic

```
resource "scaleway_instance_security_group" "allow_all" {
}

resource "scaleway_instance_security_group" "web" {
  inbound_default_policy = "drop" # By default we drop incomming trafic that do not match any inbound_rul
e

  inbound_rule {
    action = "accept"
    port = 22
    ip = "212.47.225.64"
  }

  inbound_rule {
    action = "accept"
    port = 80
  }
}
```

Web server with banned IP and restricted internet access

```
resource "scaleway_instance_security_group" "web" {
  inbound_default_policy = "drop" # By default we drop incomming trafic that do not match any inbound_rul
e.
  outbound_default_policy = "drop" # By default we drop outgoing trafic that do not match any outbound_ru
le.

  inbound_rule {
    action = "drop"
    ip = "1.1.1.1" # Banned IP
  }

  inbound_rule {
    action = "accept"
    port = 22
    ip = "212.47.225.64"
  }

  inbound_rule {
    action = "accept"
    port = 443
  }

  outbound_rule {
    action = "accept"
    ip = "8.8.8.8" # Only allow outgoing conection to this IP.
  }
}
```

Trusted IP for SSH access (using for_each)

If you use terraform >= 0.12.6, you can leverage the `for_each` (https://www.terraform.io/docs/configuration/resources.html#for_each-multiple-resource-instances-defined-by-a-map-or-set-of-strings) feature with this resource.

```
locals {
  trusted = ["192.168.0.1", "192.168.0.2", "192.168.0.3"]
}

resource "scaleway_instance_security_group" "dummy" {
  inbound_default_policy  = "drop"
  outbound_default_policy = "accept"

  dynamic "inbound_rule" {
    for_each = local.trusted

    content {
      action = "accept"
      port   = 22
      ip     = inbound_rule.value
    }
  }
}
```

Arguments Reference

The following arguments are supported:

- `name` - (Optional) The name of the security group.
- `description` - (Optional) The description of the security group.
- `inbound_default_policy` - (Defaults to `accept`) The default policy on incoming traffic. Possible values are: `accept` or `drop`.
- `outbound_default_policy` - (Defaults to `accept`) The default policy on outgoing traffic. Possible values are: `accept` or `drop`.
- `inbound_rule` - (Optional) A list of inbound rule to add to the security group. (Structure is documented below.)
- `outbound_rule` - (Optional) A list of outbound rule to add to the security group. (Structure is documented below.)
- `zone` - (Defaults to provider (/docs/providers/scaleway/index.html#zone) `zone`) The zone (/docs/providers/scaleway/guides/regions_and_zones.html#zones) in which the server should be created.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the project the server is associated with.

The `inbound_rule` and `outbound_rule` block supports:

- `action` - (Required) The action to take when rule match. Possible values are: `accept` or `drop`.
- `protocol` - (Defaults to `TCP`) The protocol this rule apply to. Possible values are: `TCP`, `UDP`, `ICMP` or `ANY`.
- `port` - (Optional) The port this rule apply to. If no port is specified, rule will apply to all port.
- `ip` - (Optional) The ip this rule apply to. If no `ip` nor `ip_range` are specified, rule will apply to all ip. Only one of `ip` and `ip_range` should be specified.
- `ip_range` - (Optional) The ip range (e.g `192.168.1.0/24`) this rule apply to. If no `ip` nor `ip_range` are specified, rule will apply to all ip. Only one of `ip` and `ip_range` should be specified.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the server.

Import

Instance security group can be imported using the `{zone}/{id}`, e.g.

```
$ terraform import scaleway_instance_security_group.web fr-par-1/11111111-1111-1111-1111-111111111111
```

scaleway_instance_server

Creates and manages Scaleway Compute Instance servers. For more information, see the documentation (https://developers.scaleway.com/en/products/instance/api/#servers-8bf7d7).

Examples

Basic

```
resource "scaleway_instance_ip" "public_ip" {
  server_id = "${scaleway_instance_server.web.id}"
}

resource "scaleway_instance_server" "web" {
  type = "DEV1-S"
  image = "f974feac-abae-4365-b988-8ec7d1cec10d"
}
```

With additional volumes, public IP and tags

```
resource "scaleway_instance_volume" "data" {
  size_in_gb = 100
}

resource "scaleway_instance_server" "web" {
  type = "DEV1-L"
  image = "f974feac-abae-4365-b988-8ec7d1cec10d"

  tags = [ "hello", "public" ]

  root_volume {
    delete_on_termination = false
  }

  additional_volume_ids = [ "${scaleway_instance_volume.data.id}" ]
}
```

With security group

```
resource "scaleway_instance_security_group" "www" {
  inbound_default_policy = "drop"
  outbound_default_policy = "accept"

  inbound_rule {
    action = "accept"
    port = "22"
    ip = "212.47.225.64"
  }

  inbound_rule {
    action = "accept"
    port = "80"
  }

  inbound_rule {
    action = "accept"
    port = "443"
  }

  outbound_rule {
    action = "drop"
    ip_range = "10.20.0.0/24"
  }
}

resource "scaleway_instance_server" "web" {
  type = "DEV1-S"
  image = "f974feac-abae-4365-b988-8ec7d1cec10d"

  security_group_id= "${scaleway_instance_security_group.www.id}"
}
```

With user data and could-init

```
resource "scaleway_instance_server" "web" {
  type = "DEV1-L"
  image = "f974feac-abae-4365-b988-8ec7d1cec10d"

  tags = [ "web", "public" ]

  user_data {
    key = "plop"
    value = "world"
  }

  user_data {
    key = "xavier"
    value = "niel"
  }

  cloud_init = file("${path.module}/cloud-init.yml")
}
```

Arguments Reference

The following arguments are supported:

- `type` - (Required) The commercial type of the server. You find all the available types on the pricing page (https://www.scaleway.com/en/pricing/). Updates to this field will recreate a new resource.
- `image` - (Required) The UUID or the label of the base image used by the server. You can use this endpoint (https://api-marketplace.scaleway.com/images?page=1&per_page=100) to find either the right `label` or the right local image `ID` for a given `commercial_type`.
- `name` - (Optional) The name of the server.
- `tags` - (Optional) The tags associated with the server.
- `security_group_id` - (Optional) The security group (https://developers.scaleway.com/en/products/instance/api/#security-groups-8d7f89) the server is attached to.
- `placement_group_id` - (Optional) The placement group (https://developers.scaleway.com/en/products/instance/api/#placement-groups-d8f653) the server is attached to.

Important: Updates to `placement_group_id` may trigger a stop/start of the server.

- `root_volume` - (Optional) Root volume (https://developers.scaleway.com/en/products/instance/api/#volumes-7e8a39) attached to the server on creation.
  - `size_in_gb` - (Required) Size of the root volume in gigabytes. To find the right size use this endpoint (https://api.scaleway.com/instance/v1/zones/fr-par-1/products/servers) and check the `volumes_constraint.{min|max}_size` (in bytes) for your `commercial_type`. Updates to this field will recreate a new resource.
  - `delete_on_termination` - (Defaults to `true`) Forces deletion of the root volume on instance termination.

Important: Updates to `root_volume.size_in_gb` will trigger a stop/start of the server.

- `additional_volume_ids` - (Optional) The additional volumes (https://developers.scaleway.com/en/products/instance/api/#volumes-7e8a39) attached to the server. Updates to this field will trigger a stop/start of the server.

Important: If this field contains local volumes, updates will trigger a stop/start of the server.

- `enable_ipv6` - (Defaults to `false`) Determines if IPv6 is enabled for the server.
- `disable_dynamic_ip` - (Defaults to `false`) Disable dynamic IP on the server.
- `state` - (Defaults to `started`) The state of the server. Possible values are: `started`, `stopped` or `standby`.
- `cloud_init` - (Optional) The cloud init script associated with this server. Updates to this field will trigger a stop/start of the server.
- `user_data` - (Optional) The user data associated with the server.
  - `key` - (Required) The user data key. The `cloud-init` key is reserved, please use `cloud_init` attribute instead.
  - `value` - (Required) The user data content. It could be a string or a file content using file

(https://www.terraform.io/docs/configuration/functions/file.html) or filebase64 (https://www.terraform.io/docs/configuration/functions/filebase64.html) for example.

- `zone` - (Defaults to provider (/docs/providers/scaleway/index.html#zone) `zone`) The zone (/docs/providers/scaleway/guides/regions_and_zones.html#zones) in which the server should be created.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the organization the server is associated with.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the server.
- `placement_group_policy_respected` - True when the placement group policy is respected.
- `root_volume`
  - `volume_id` - The volume ID of the root volume of the server.
- `private_ip` - The Scaleway internal IP address of the server.
- `public_ip` - The public IPv4 address of the server.
- `ipv6_address` - The default ipv6 address routed to the server. (Only set when enable_ipv6 is set to true)
- `ipv6_gateway` - The ipv6 gateway address. (Only set when enable_ipv6 is set to true)
- `ipv6_prefix_length` - The prefix length of the ipv6 subnet routed to the server. (Only set when enable_ipv6 is set to true)

Import

Instance servers can be imported using the `{zone}/{id}`, e.g.

```
$ terraform import scaleway_instance_server.web fr-par-1/11111111-1111-1111-1111-111111111111
```

scaleway_instance_volume

Creates and manages Scaleway Compute Instance Volumes. For more information, see the documentation (https://developers.scaleway.com/en/products/instance/api/#volumes-7e8a39).

Example

```
resource "scaleway_instance_volume" "server_volume" {
    type       = "l_ssd"
    name       = "some-volume-name"
    size_in_gb = 20
}
```

Arguments Reference

The following arguments are supported:

- `type` - (Required) The type of the volume. The possible values are: `b_ssd` (Block SSD), `l_ssd` (Local SSD).
- `size_in_gb` - (Optional) The size of the volume (leave this empty when using `from_volume_id` or `from_snapshot_id`).
- `from_volume_id` - (Optional) If set, the new volume will be copied from this volume. (leave this empty when using `size_in_gb` or `from_snapshot_id`).
- `from_snapshot_id` - (Optional) If set, the new volume will be created from this snapshot. (leave this empty when using `size_in_gb` or `from_volume_id`).
- `name` - (Optional) The name of the volume. If not provided it will be randomly generated.
- `zone` - (Defaults to provider (/docs/providers/scaleway/index.html#zone) `zone`) The zone (/docs/providers/scaleway/guides/regions_and_zones.html#zones) in which the volume should be created.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the organization the volume is associated with.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the volume.
- `server_id` - The id of the associated server.

Import

volumes can be imported using the `{zone}/{id}`, e.g.

```
$ terraform import scaleway_instance_volume.server_volume fr-par-1/11111111-1111-1111-1111-111111111111
```

scaleway_ip

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_ip` instead.

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

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_ip` instead.

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_k8s_cluster_beta

Creates and manages Scaleway Kubernetes clusters. For more information, see the documentation (https://developers.scaleway.com/en/products/k8s/api/).

Examples

Basic

```
resource "scaleway_k8s_cluster_beta" "jack" {
  name = "jack"
  version = "1.16.1"
  cni = "calico"
  default_pool {
    node_type = "GP1-XS"
    size = 3
  }
}
```

With additional configuration

```
resource "scaleway_k8s_cluster_beta" "john" {
  name = "john"
  description = "my awesome cluster"
  version = "1.16.1"
  cni = "weave"
  enable_dashboard = true
  ingress = "traefik"
  tags = ["i'm an awsome tag", "yay"]

  default_pool {
    node_type = "GP1-XS"
    size = 3
    autoscaling = true
    autohealing = true
    min_size = 1
    max_size = 5
  }

  autoscaler_config {
    disable_scale_down = false
    scale_down_delay_after_add = 5m
    estimator = "binpacking"
    expander = "random"
    ignore_daemonsets_utilization = true
    balance_similar_node_groups = true
    expendable_pods_priority_cutoff = -5
  }
}
```

With the kubernetes provider

```
resource "scaleway_k8s_cluster_beta" "joy" {
  name = "joy"
  version = "1.16.1"
  cni = "flannel"
  default_pool {
    node_type = "GP1-XS"
    size = 3
  }
}

provider "kubernetes" {
  host  = scaleway_k8s_cluster_beta.joy.kubeconfig[0].host
  token  = scaleway_k8s_cluster_beta.joy.kubeconfig[0].token
  cluster_ca_certificate = base64decode(
    scaleway_k8s_cluster_beta.joy.kubeconfig[0].cluster_ca_certificate
  )
}
```

Arguments Reference

The following arguments are supported:

- `name` - (Required) The name for the Kubernetes cluster. ~> **Important:** Updates to this field will recreate a new resource.
- `description` - (Optional) A description for the Kubernetes cluster.
- `version` - (Optional) The version of the Kubernetes cluster (will default to the latest).
- `cni` - (Required) The Container Network Interface (CNI) for the Kubernetes cluster. ~> **Important:** Updates to this field will recreate a new resource.
- `enable_dashboard` - (Defaults to `false`) Enables the Kubernetes dashboard (https://github.com/kubernetes/dashboard) for the Kubernetes cluster. ~> **Important:** Updates to this field will recreate a new resource.
- `ingress` - (Defaults to `no_ingress`) The ingress controller (https://kubernetes.io/docs/concepts/services-networking/ingress-controllers/) to be deployed on the Kubernetes cluster. ~> **Important:** Updates to this field will recreate a new resource.
- `tags` - (Optional) The tags associated with the Kubernetes cluster.
- `autoscaler_config` - (Optional) The configuration options for the Kubernetes cluster autoscaler (https://github.com/kubernetes/autoscaler/tree/master/cluster-autoscaler).
  - `disable_scale_down` - (Defaults to `false`) Disables the scale down feature of the autoscaler.
  - `scale_down_delay_after_add` - (Defaults to `10m`) How long after scale up that scale down evaluation resumes.
  - `estimator` - (Defaults to `binpacking`) Type of resource estimator to be used in scale up.
  - `expander` - (Default to `random`) Type of node group expander to be used in scale up.
  - `ignore_daemonsets_utilization` - (Defaults to `false`) Ignore DaemonSet pods when calculating resource utilization for scaling down.
  - `balance_similar_node_groups` - (Defaults to `false`) Detect similar node groups and balance the number of nodes between them.
  - `expendable_pods_priority_cutoff` - (Defaults to `-10`) Pods with priority below cutoff will be expendable. They can be killed without any consideration during scale down and they don't cause scale up. Pods with null priority (PodPriority disabled) are non expendable.
- `default_pool` - (Required) The cluster's default pool configuration.
  - `node_type` - (Required) The commercial type of the default pool instances. ~> **Important:** Updates to this field will recreate a new resource.
  - `size` - (Required) The size of the default pool.
  - `min_size` - (Defaults to `1`) The minimum size of the default pool, used by the autoscaling feature.
  - `max_size` - (Defaults to `size`) The maximum size of the default pool, used by the autoscaling feature.
  - `placement_group_id` - (Optional) The placement group (https://developers.scaleway.com/en/products/instance/api/#placement-groups-d8f653) the nodes of the pool

will be attached to.

    - `autoscaling` - (Defaults to `false`) Enables the autoscaling feature for the default pool. ~> **Important:** When enabled, an update of the `size` will not be taken into account.
    - `autohealing` - (Defaults to `false`) Enables the autohealing feature for the default pool.
    - `container_runtime` - (Defaults to `docker`) The container runtime of the default pool.
- `region` - (Defaults to provider (/docs/providers/scaleway/index.html#region) `region`) The region (/docs/providers/scaleway/guides/regions_and_zones.html#regions) in which the cluster should be created.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the organization the cluster is associated with.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the cluster.
- `created_at` - The creation date of the cluster.
- `updated_at` - The last update date of the cluster.
- `apiserver_url` - The URL of the Kubernetes API server.
- `wildcard_dns` - The DNS wildcard that points to all ready nodes.
- `kubeconfig`
    - `config_file` - The raw kubeconfig file.
    - `host` - The URL of the Kubernetes API server.
    - `cluster_ca_certificate` - The CA certificate of the Kubernetes API server.
    - `token` - The token to connect to the Kubernetes API server.
- `status` - The status of the Kubernetes cluster.
- `default_pool`
    - `pool_id` - The ID of the default pool.
    - `created_at` - The creation date of the default pool.
    - `updated_at` - The last update date of the default pool.

Import

Kubernetes clusters can be imported using the `{region}/{id}`, e.g.

```
$ terraform import scaleway_k8s_cluster_beta.mycluster fr-par/11111111-1111-1111-1111-111111111111
```

scaleway_k8s_pool_beta

Creates and manages Scaleway Kubernetes cluster pools. For more information, see the documentation (https://developers.scaleway.com/en/products/k8s/api/).

Examples

Basic

```
resource "scaleway_k8s_cluster_beta" "jack" {
  name = "jack"
  version = "1.16.1"
  cni = "calico"
  default_pool {
    node_type = "GP1-XS"
    size = 3
  }
}

resource "scaleway_k8s_pool_beta" "bill" {
  cluster_id = "${scaleway_k8s_cluster_beta.jack.id}"
  name = "bill"
  node_type = "GP1-S"
  size = 3
  min_size = 0
  max_size = 10
  autoscaling = true
  autohealing = true
  container_runtime = "docker"
  placement_group_id = "1267e3fd-a51c-49ed-ad12-857092ee3a3d"
}
```

Arguments Reference

The following arguments are supported:

- `cluster_id` - (Required) The ID of the Kubernetes cluster on which this pool will be created.
- `name` - (Required) The name for the pool. ~> **Important:** Updates to this field will recreate a new resource.
- `node_type` - (Required) The commercial type of the pool instances. ~> **Important:** Updates to this field will recreate a new resource.
- `size` - (Required) The size of the pool.
- `min_size` - (Defaults to `1`) The minimum size of the pool, used by the autoscaling feature.
- `max_size` - (Defaults to `size`) The maximum size of the pool, used by the autoscaling feature.

- `placement_group_id` - (Optional) The placement group (https://developers.scaleway.com/en/products/instance/api/#placement-groups-d8f653) the nodes of the pool will be attached to.
- `autoscaling` - (Defaults to `false`) Enables the autoscaling feature for this pool. ~> **Important:** When enabled, an update of the `size` will not be taken into account.
- `autohealing` - (Defaults to `false`) Enables the autohealing feature for this pool.
- `container_runtime` - (Defaults to `docker`) The container runtime of the pool.
- `region` - (Defaults to provider (/docs/providers/scaleway/index.html#region) `region`) The region (/docs/providers/scaleway/guides/regions_and_zones.html#regions) in which the pool should be created.

Attributes Reference

In addition to all above arguments, the following attributes are exported:

- `id` - The ID of the pool.
- `created_at` - The creation date of the pool.
- `updated_at` - The last update date of the pool.
- `version` - The version of the pool.

Import

Kubernetes pools can be imported using the `{region}/{id}`, e.g.

```
$ terraform import scaleway_k8s_pool_beta.mypool fr-par/11111111-1111-1111-1111-111111111111
```

scaleway_lb_backend_beta

Note: This terraform resource is flagged beta and might include breaking change in future releases.

Creates and manages Scaleway Load-Balancer Backends. For more information, see the documentation (https://developers.scaleway.com/en/products/lb/api).

Examples

Basic

```
resource "scaleway_lb_backend_beta" "backend01" {
    lb_id = scaleway_lb_beta.lb01.id
    name = "backend01"
    forward_protocol = "http"
    forward_port = "80"
}
```

Arguments Reference

The following arguments are supported:

- `lb_id` - (Required) The load-balancer ID this backend is attached to. ~> **Important:** Updates to `lb_id` will recreate the backend.
- `forward_protocol` - (Required) Backend protocol. Possible values are: `TCP` or `HTTP`.
- `name` - (Optional) The name of the load-balancer backend.
- `forward_port` - (Required) User sessions will be forwarded to this port of backend servers.
- `forward_port_algorithm` - (Default: `roundrobin`) Load balancing algorithm. Possible values are: `roundrobin` and `leastconn`.
- `sticky_sessions` - (Default: `none`) Load balancing algorithm. Possible values are: `none`, `cookie` and `table`.
- `sticky_sessions_cookie_name` - (Optional) Cookie name for for sticky sessions. Only applicable when sticky_sessions is set to `cookie`.
- `server_ips` - (Optional) List of backend server IP addresses. Addresses can be either IPv4 or IPv6.
- `send_proxy_v2` - (Default: `false`) Enables PROXY protocol version 2.
- `timeout_server` - (Optional) Maximum server connection inactivity time. (e.g.: `1s`)
- `timeout_connect` - (Optional) Maximum initial server connection establishment time. (e.g.: `1s`)
- `timeout_tunnel` - (Optional) Maximum tunnel inactivity time. (e.g.: `1s`)

- `on_marked_down_action` - (Default: `none`) Modify what occurs when a backend server is marked down. Possible values are: `none` and `shutdown_sessions`.

Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the loadbalancer backend.

Import

Load-Balancer backend can be imported using the `{region}/{id}`, e.g.

```
$ terraform import scaleway_lb_backend_beta.backend01 fr-par/11111111-1111-1111-1111-111111111111
```

scaleway_lb_beta

Note: This terraform resource is flagged beta and might include breaking change in future releases.

Creates and manages Scaleway Load-Balancers. For more information, see the documentation (https://developers.scaleway.com/en/products/lb/api).

Examples

Basic

```
resource "scaleway_lb_beta" "base" {
  region      = "fr-par"
  type        = "LB-S"
}
```

Arguments Reference

The following arguments are supported:

- `type` - (Required) The type of the load-balancer. For now only `LB-S` is available

Important: Updates to `type` will recreate the load-balancer.

- `name` - (Optional) The name of the load-balancer.
- `tags` - (Optional) The tags associated with the load-balancers.
- `region` - (Defaults to provider (/docs/providers/scaleway/index.html#region) `region`) The region (/docs/providers/scaleway/guides/regions_and_zones.html#regions) in which the load-balacer should be created.
- `organization_id` - (Defaults to provider (/docs/providers/scaleway/index.html#organization_id) `organization_id`) The ID of the organization the server is associated with.

Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the load-balancer.
- `ip_id` - The load-balance public IP ID
- `ip_address` - The load-balance public IP Address

Import

Load-Balancer can be imported using the `{region}/{id}`, e.g.

```
$ terraform import scaleway_lb_beta.lb01 fr-par/11111111-1111-1111-1111-111111111111
```

scaleway_lb_frontend_beta

Note: This terraform resource is flagged beta and might include breaking change in future releases.

Creates and manages Scaleway Load-Balancer Frontends. For more information, see the documentation (https://developers.scaleway.com/en/products/lb/api).

Examples

Basic

```
resource "scaleway_lb_frontend_beta" "frontend01" {
    lb_id = scaleway_lb_beta.lb01.id
    backend_id = scaleway_lb_backend_beta.bkd01.id
    name = "frontend01"
    inbound_port = "80"
}
```

Arguments Reference

The following arguments are supported:

- `lb_id` - (Required) The load-balancer ID this frontend is attached to.
- `backend_id` - (Required) The load-balancer backend ID this frontend is attached to. ~> **Important:** Updates to `lb_id` or `backend_id` will recreate the frontend.
- `inbound_port` - (Required) TCP port to listen on the front side.
- `name` - (Optional) The name of the load-balancer frontend.
- `timeout_client` - (Optional) Maximum inactivity time on the client side. (e.g.: `1s`)
- `certificate_id` - (Required) Certificate ID that should be used by the frontend.

Attributes Reference

In addition to all arguments above, the following attributes are exported:

- `id` - The ID of the loadbalancer frontend.

Import

Load-Balancer frontend can be imported using the `{region}/{id}`, e.g.

```
$ terraform import scaleway_lb_frontend_beta.frontend01 fr-par/11111111-1111-1111-1111-111111111111
```

scaleway_object_bucket

Creates and manages Scaleway object storage buckets. For more information, see the documentation (https://www.scaleway.com/en/docs/object-storage-feature/).

Example Usage

```
resource "scaleway_object_bucket" "some_bucket" {
    name = "some-unique-name"
    acl = "private"
}
```

Arguments Reference

The following arguments are supported:

- `name` - (Required) The name of the bucket.
- `acl` - (Optional) The canned ACL (https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html#canned-acl) you want to apply to the bucket.
- `region` - (Optional) The region (https://developers.scaleway.com/en/quickstart/#region-definition) in which the bucket should be created.

Attributes Reference

In addition to all above arguments, the following attribute is exported:

- `id` - The ID of the bucket.

Import

Buckets can be imported using the `{region}/{id}` identifier, e.g.

```
$ terraform import scaleway_object_bucket.some_bucket fr-par/11111111-1111-1111-1111-111111111111
```

scaleway_security_group

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_security_group` instead.

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

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_security_group_rule` instead.

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

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_server` instead.

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

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `account_ssh_key` instead.

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

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`.

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

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_server` instead.

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_attachment

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_server.additional_volumes` instead.

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

scaleway_volume

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_volume` instead.

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
```