» 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 `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_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 `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 SSH public key.
- `public_key` - The SSH public key string

» scaleway_instance_ip

Creates and manages Scaleway Compute Instance IPs. For more information, see the documentation.

» Example Usage

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

» Arguments Reference

The following arguments are supported:

- `zone` - (Defaults to provider `zone`) The zone in which the IP should be reserved.
- `organization_id` - (Defaults to provider `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.
- `reverse` - The reverse dns attached to this IP

» Import

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

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

» scaleway_instance_placement_group

Creates and manages Compute Instance Placement Groups. For more information, see the documentation.

» 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 `max_availability`) The policy type of the placement group. Possible values are: `low_latency` or `max_availability`.
- `policy_mode` - (Defaults to `optional`) The policy mode of the placement group. Possible values are: `optional` or `enforced`.
- `zone` - (Defaults to provider `zone`) The zone in which the placement group should be created.
- `organization_id` - (Defaults to provider `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-1/11111111-11
```

» scaleway_instance_security_group

Creates and manages Scaleway Compute Instance security groups. For more information, see the documentation.

» 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 an

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

  inbound_rule {
    action = "accept"
    port = 80
  }

  inbound_rule {
    action = "accept"
    protocol = "UDP"
    port_range = "22-23"
  }
}
```

» 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 an
  outbound_default_policy = "drop" # By default we drop outgoing trafic that do not match an

  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` 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.)
- `external_rules` - (Defaults to `false`) A boolean to specify whether to use instance_security_group_rules. If `external_rules` is set to `true`, `inbound_rule` and `outbound_rule` can not be set directly in the security group.
- `zone` - (Defaults to provider `zone`) The zone in which the security group should be created.
- `organization_id` - (Defaults to provider `organization_id`) The ID of the project the security group 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 applies to. If no `port` nor `port_range` are specified, the rule will apply to all port. Only one of `port` and `port_range` should be specified.

- `port_range`- Need terraform >= 0.13.0 (Optional) The port range (e.g `22-23`) this rule applies to. Port range MUST comply the Scaleway-notation: interval between ports must be a power of 2 `2^X-1` number (e.g $2^{13\text{-}1=8191}$ in port_range = "10000-18191").If no `port` nor `port_range` are specified, rule will apply to all port. Only one of `port` and `port_range` should be specified.
- `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 applies 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 security group.

» Import

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

```
$ terraform import scaleway_instance_security_group.web fr-par-1/11111111-1111-1111-1111-111
```

» scaleway_instance_security_group_rules

Creates and manages Scaleway Compute Instance security group rules. For more information, see the documentation.

» Examples

» Basic

```
resource scaleway_instance_security_group sg01 {
    external_rules = true
}

resource scaleway_instance_security_group_rules sgrs01 {
    security_group_id = scaleway_instance_security_group.sg01.id
    inbound_rule {
```

```
        action = "accept"
        port = 80
        ip_range = "0.0.0.0/0"
    }
}
```

» Arguments Reference

The following arguments are supported:

- `security_group_id` - (Required) The ID of the security group.
- `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.)

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 applies 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 security group.

» Import

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

```
$ terraform import scaleway_instance_security_group_rules.web fr-par-1/11111111-1111-1111-11
```

» scaleway_instance_server

Creates and manages Scaleway Compute Instance servers. For more information, see the documentation.

» Examples

» Basic

```
resource "scaleway_instance_ip" "public_ip" {}

resource "scaleway_instance_server" "web" {
  type = "DEV1-S"
  image = "ubuntu-bionic"
  ip_id = scaleway_instance_ip.public_ip.id
}
```

» With additional volumes and tags

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

resource "scaleway_instance_server" "web" {
  type = "DEV1-L"
  image = "ubuntu-bionic"

  tags = [ "hello", "public" ]

  root_volume {
    delete_on_termination = false
  }

  additional_volume_ids = [ scaleway_instance_volume.data.id ]
}
```

» With a reserved IP

```
resource "scaleway_instance_ip" "ip" {}

resource "scaleway_instance_server" "web" {
  type = "DEV1-L"
```

```
  image = "f974feac-abae-4365-b988-8ec7d1cec10d"

  tags = [ "hello", "public" ]

  ip_id = scaleway_instance_ip.ip.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 = "ubuntu-bionic"

  security_group_id= scaleway_instance_security_group.www.id
}
```

» With user data and cloud-init

```
resource "scaleway_instance_server" "web" {
  type = "DEV1-L"
  image = "ubuntu-bionic"

  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. 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 to find either the right `label` or the right local image `ID` for a given `type`.
- `name` - (Optional) The name of the server.
- `tags` - (Optional) The tags associated with the server.
- `security_group_id` - (Optional) The security group the server is attached to.
- `placement_group_id` - (Optional) The placement group the server is attached to.

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

- `root_volume` - (Optional) Root volume 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 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 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.
- `ip_id` = (Optional) The ID of the reserved IP that is attached to the server.
- `enable_dynamic_ip` - (Defaults to `false`) If true a dynamic IP will be attached to 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 or filebase64 for example.
- `zone` - (Defaults to provider `zone`) The zone in which the server should be created.
- `organization_id` - (Defaults to provider `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)
- `boot_type` - The boot Type of the server. Possible values are: `local`, `bootscript` or `rescue`.

» Import

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

```
$ terraform import scaleway_instance_server.web fr-par-1/11111111-1111-1111-1111-1111111111
```

» scaleway_instance_volume

Creates and manages Scaleway Compute Instance Volumes. For more information, see the documentation.

» 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. Only one of `size_in_gb`, `from_volume_id` and `from_volume_id` should be specified.
- `from_volume_id` - (Optional) If set, the new volume will be copied from this volume. Only one of `size_in_gb`, `from_volume_id` and `from_volume_id` should be specified.

- `from_snapshot_id` - (Optional) If set, the new volume will be created from this snapshot. Only one of `size_in_gb`, `from_volume_id` and `from_volume_id` should be specified.
- `name` - (Optional) The name of the volume. If not provided it will be randomly generated.
- `zone` - (Defaults to provider `zone`) The zone in which the volume should be created.
- `organization_id` - (Defaults to provider `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-1/11111111-1111-1111-1111-1
```

» scaleway_instance_server

Gets information about an instance server.

» Example Usage

```
# Get info by server name
data "scaleway_instance_server" "my_key" {
  name  = "my-server-name"
}

# Get info by server id
data "scaleway_instance_server" "my_key" {
  server_id = "11111111-1111-1111-1111-111111111111"
}
```

» Argument Reference

- `name` - (Optional) The server name. Only one of `name` and `server_id` should be specified.
- `server_id` - (Optional) The server id. Only one of `name` and `server_id` should be specified.
- `zone` - (Defaults to provider `zone`) The zone in which the server exists.

» Attributes Reference

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

- `id` - The ID of the server.
- `type` - The commercial type of the server. You find all the available types on the pricing page.
- `image` - The UUID and the label of the base image used by the server.
- `organization_id` - The ID of the organization the server is associated with.
- `tags` - The tags associated with the server.
- `security_group_id` - The security group the server is attached to.
- `placement_group_id` - The placement group the server is attached to.
- `root_volume` - Root volume attached to the server on creation.
  - `size_in_gb` - Size of the root volume in gigabytes.
  - `delete_on_termination` - Forces deletion of the root volume on instance termination.
- `additional_volume_ids` - The additional volumes attached to the server.
- `enable_ipv6` - Determines if IPv6 is enabled for the server.
- `enable_dynamic_ip` - True is dynamic IP in enable on the server.
- `state` - The state of the server. Possible values are: `started`, `stopped` or `standby`.
- `cloud_init` - The cloud init script associated with this server.
- `user_data` - The user data associated with the server.
  - `key` - The user data key. The `cloud-init` key is reserved, please use `cloud_init` attribute instead.
  - `value` - The user data content.

- `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)

» scaleway_instance_image

Gets information about an instance image.

» Example Usage

```
# Get info by image name
data "scaleway_instance_image" "my_image" {
  name  = "my-image-name"
}

# Get info by image id
data "scaleway_instance_image" "my_image" {
  image_id = "11111111-1111-1111-1111-111111111111"
}
```

» Argument Reference

- `name` - (Optional) The image name. Only one of `name` and `image_id` should be specified.
- `image_id` - (Optional) The image id. Only one of `name` and `image_id` should be specified.
- `architecture` - (Optional, default `x86_64`) The architecture the image is compatible with. Possible values are: `x86_64` or `arm`.

- `latest` - (Optional, default `true`) Use the latest image ID.
- `zone` - (Defaults to provider `zone`) The zone in which the image exists.

» Attributes Reference

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

- `id` - The ID of the image.
- `organization_id` - The ID of the organization the image is associated with.
- `creation_date` - Date of the image creation.
- `modification_date` - Date of image latest update.
- `public` - Set to `true` if the image is public.
- `from_server_id` - ID of the server the image if based from.
- `state` - State of the image. Possible values are: `available`, `creating` or `error`.
- `default_bootscript_id` - ID of the default bootscript for this image.
- `root_volume_id` - ID of the root volume in this image.
- `additional_volume_ids` - IDs of the additional volumes in this image.

» scaleway_instance_volume

Gets information about an instance volume.

» Example Usage

```
# Get info by volume name
data "scaleway_instance_volume" "my_volume" {
  name = "my-volume-name"
}

# Get info by volume ID
data "scaleway_instance_volume" "my_volume" {
  volume_id = "11111111-1111-1111-1111-111111111111"
}
```

» Argument Reference

- `name` - (Optional) The volume name. Only one of `name` and `volume_id` should be specified.
- `volume_id` - (Optional) The volume id. Only one of `name` and `volume_id` should be specified.
- `zone` - (Defaults to provider `zone`) The zone in which the volume exists.
- `organization_id` - (Defaults to provider `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:

- `volume_type` - The type of the volume. `l_ssd` for local SSD, `b_ssd` for block storage SSD.
- `creation_date` - Volume creation date.
- `modification_date` - Volume last modification date.
- `state` - State of the volume. Possible values are `available`, `snapshotting` and `error`. The default value is available.
- `size` - The volumes disk size (in bytes).
- `server` - Information about the server attached to the volume.

» scaleway_security_group

Gets information about a Security Group.

» Example Usage

```
# Get info by security group name
data "scaleway_instance_security_group" "my_key" {
  name  = "my-security-group-name"
}

# Get info by security group id
data "scaleway_instance_security_group" "my_key" {
  security_group_id = "11111111-1111-1111-1111-111111111111"
}
```

» Argument Reference

- `name` - (Optional) The security group name. Only one of `name` and `security_group_id` should be specified.
- `security_group_id` - (Optional) The security group id. Only one of `name` and `security_group_id` should be specified.
- `zone` - (Defaults to provider `zone`) The zone in which the security group exists.

» Attributes Reference

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

- `id` - The ID of the security group.
- `organization_id` - The ID of the organization the security group is associated with.
- `inbound_default_policy` - The default policy on incoming traffic. Possible values are: `accept` or `drop`.
- `outbound_default_policy` - The default policy on outgoing traffic. Possible values are: `accept` or `drop`.
- `inbound_rule` - A list of inbound rule to add to the security group. (Structure is documented below.)
- `outbound_rule` - A list of outbound rule to add to the security group. (Structure is documented below.)

The `inbound_rule` and `outbound_rule` block supports:

- `action` - The action to take when rule match. Possible values are: `accept` or `drop`.
- `protocol`- The protocol this rule apply to. Possible values are: `TCP`, `UDP`, `ICMP` or `ANY`.
- `port`- The port this rule apply to. If no port is specified, rule will apply to all port.
- `ip`- The ip this rule apply to.
- `ip_range`- The ip range (e.g `192.168.1.0/24`) this rule apply to.

» scaleway_object_bucket

Creates and manages Scaleway object storage buckets. For more information, see the documentation.

» 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 you want to apply to the bucket.
- `region` - (Optional) The region in which the bucket should be created.

» Attributes Reference

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

- `id` - The unique name of the bucket.

» Import

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

```
$ terraform import scaleway_object_bucket.some_bucket fr-par/some-bucket
```

» scaleway_baremetal_server_beta

Creates and manages Scaleway Compute Baremetal servers. For more information, see the documentation.

» Examples

» Basic

```
data "scaleway_account_ssh_key" "main" {
  name = "main"
}

resource "scaleway_baremetal_server_beta" "base" {
  zone          = "fr-par-2"
  offer       = "GP-BM1-M"
  os_id       = "d17d6872-0412-45d9-a198-af82c34d3c5c"
  ssh_key_ids = [data.scaleway_account_ssh_key.main]
}
```

» Arguments Reference

The following arguments are supported:

- `offer` - (Required) The offer name or UUID of the baremetal server. Use this endpoint to find the right offer.

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 to find the right OS ID.

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

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

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

- `tags` - (Optional) The tags associated with the server.
- `zone` - (Defaults to provider `zone`) The zone in which the server should be created.
- `organization_id` - (Defaults to provider `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.
- `ips` - (List of) The IPs of the server.
  - `id` - The ID of the IP.
  - `address` - The address of the IP.
  - `reverse` - The reverse of the IP.
- `domain` - The domain of the server.

» Import

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

```
$ terraform import scaleway_baremetal_server_beta.web fr-par-2/11111111-1111-1111-1111-11111
```

» scaleway_baremetal_offer_beta

Gets information about a baremetal offer. For more information, see the documentation.

» Example Usage

```
# Get info by offer name
data "scaleway_baremetal_offer_beta" "my_offer" {
  zone = "fr-par-2"
  name = "HC-BM1-L"
}

# Get info by offer id
data "scaleway_baremetal_offer_beta" "my_offer" {
  zone     = "fr-par-2"
  offer_id = "3ab0dc29-2fd4-486e-88bf-d08fbf49214b"
}
```

» Argument Reference

- `name` - (Optional) The offer name. Only one of `name` and `offer_id` should be specified.
- `offer_id` - (Optional) The offer id. Only one of `name` and `offer_id` should be specified.
- `allow_disabled` - (Optional, default `false`) Include disabled offers.
- `zone` - (Defaults to provider `zone`) The zone in which the offer should be created.

» Attributes Reference

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

- `id` - The ID of the offer.
- `bandwidth` - Available Bandwidth with the offer.
- `commercial_range` - Commercial range of the offer.
- `cpu` - A list of cpu specifications. (Structure is documented below.)
- `disk` - A list of disk specifications. (Structure is documented below.)
- `memory` - A list of memory specifications. (Structure is documented below.)
- `stock` - Stock status for this offer. Possible values are: `empty`, `low` or `available`.

The `cpu` block supports:

- `name` - Name of the CPU.
- `core_count`- Number of core on this CPU.
- `frequency`- Frequency of the CPU in MHz.
- `thread_count`- Number of thread on this CPU.

The `disk` block supports:

- `type` - Type of disk.
- `capacity`- Capacity of the disk in GB.

The `memory` block supports:

- `type` - Type of memory.
- `capacity`- Capacity of the memory in GB.
- `frequency` - Frequency of the memory in MHz.
- `ecc`- True if error-correcting code is available on this memory.

» scaleway_registry_namespace_beta

Creates and manages Scaleway Container Registry. For more information see the documentation.

» Examples

» Basic

```
resource "scaleway_registry_namespace_beta" "main" {
    name = "main_cr"
    description = "Main container registry"
    is_public = false
}
```

» Arguments Reference

The following arguments are supported:

- `name` - (Required) The unique name of the container registry namespace.

Important Updates to `name` will recreate the namespace.

- `description` (Optional) The description of the container registry namespace.
- `is_public` (Defaults to `false`) Whether or not the registry images stored in the namespace should be downloadable publicly (docker pull).
- `region` - (Defaults to provider `region`). The region in which the container registry namespace should be created.
- `organization_id` - (Defaults to provider `organization_id`) The ID of the organization the registry is associated with.

» Attibutes Reference

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

- `id` - The ID of the namespace
- `endpoint` - Endpoint reachable by docker.

» Import

Container Registry Namespace can be imported using the `{region}/{id}`, eg.

```
$ terraform import scaleway_container_registy.cr01 fr-par/11111111-1111-1111-1111-111111111
```

» scaleway_registry_namespace_beta

Gets information about a registry namespace.

» Example Usage

```
// Get info by namespace name
data "scaleway_registry_namespace_beta" "my_namespace" {
  name = "my-namespace-name"
}

// Get info by namespace ID
data "scaleway_registry_namespace_beta" "my_namespace" {
  namespace_id = "11111111-1111-1111-1111-111111111111"
}
```

» Argument Reference

- `name` - (Optional) The namespace name. Only one of `name` and `namespace_id` should be specified.
- `namespace_id` - (Optional) The namespace id. Only one of `name` and `namespace_id` should be specified.
- `region` - (Defaults to provider `region`) The region in which the namespace exists.
- `organization_id` - (Defaults to provider `organization_id`) The ID of the organization the namespace is associated with.

» Attributes Reference

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

- `id` - The ID of the Registry Namespace.
- `is_public` - The Namespace Privacy Policy: whether or not the images are public.
- `endpoint` - The endpoint of the Registry Namespace.

» scaleway_registry_image_beta

Gets information about a registry image.

» Example Usage

```
# Get info by image name
data "scaleway_registry_image_beta" "my_image" {
```

```
  name = "my-image-name"
}

# Get info by image ID
data "scaleway_registry_image_beta" "my_image" {
  image_id = "11111111-1111-1111-1111-111111111111"
  namespace_id = "11111111-1111-1111-1111-111111111111" # Optional
}
```

» Argument Reference

- `name` - (Optional) The image name. Only one of `name` and `image_id` should be specified.
- `image_id` - (Optional) The image ID. Only one of `name` and `image_id` should be specified.
- `namespace_id` - (Optional) The namespace ID in which the image is.
- `region` - (Defaults to provider `region`) The region in which the image exists.
- `organization_id` - (Defaults to provider `organization_id`) The ID of the organization the image is associated with.

» Attributes Reference

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

- `id` - The ID of the registry image.
- `size` - The size of the registry image.
- `visibility` - The privacy policy of the registry image.
- `tags` - The tags associated with the registry image

» scaleway_k8s_cluster_beta

Creates and manages Scaleway Kubernetes clusters. For more information, see the documentation.

» 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.
- `description` - (Optional) A description for the Kubernetes cluster.
- `version` - (Required) The version of the Kubernetes cluster.
- `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 for the Kubernetes cluster. ~> **Important:** Updates to this field will recreate a new resource.
- `ingress` - (Defaults to `none`) The ingress controller 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.
  - `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.

- `auto_upgrade` - (Optional) The auto upgrade configuration.
  - `enable` - (Optional) Set to `true` to enable Kubernetes patch version auto upgrades.
  - `maintenance_window_start_hour` - (Optional) The start hour (UTC) of the 2-hour auto upgrade maintenance window (0 to 23).
  - `maintenance_window_day` - (Optional) The day of the auto upgrade maintenance window (`monday` to `sunday`, or `any`).

- `feature_gates` - (Optional) The list of feature gates to enable on the cluster.

- `admission_controllers` - (Optional) The list of admission plugins to enable on the cluster.

- `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 default pool in a rolling upgrade fashion. It will first create the new pool, wait until all the nodes are ready and then delete the old pool. Errors may occur if you don't have enough quotas.
  - `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.
  - `tags` - (Optional) The tags associated with the default pool.
  - `placement_group_id` - (Optional) The placement group the nodes of the pool will be attached to. ~> **Important:** Updates to this field will recreate a new default pool.
  - `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. ~> **Important:** Updates to this field will recreate

a new default pool.
  - `wait_for_pool_ready` - (Default to `false`) Whether to wait for the pool to be ready.
- `region` - (Defaults to provider `region`) The region in which the cluster should be created.
- `organization_id` - (Defaults to provider `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.
  - `status` - The status of the default pool.
  - `nodes` - (List of) The nodes in the default pool. Defined below.
  - `created_at` - The creation date of the default pool.
  - `updated_at` - The last update date of the default pool.
- `upgrade_available` - Set to `true` if a newer Kubernetes version is available.

» nodes

- `name` - The name of the node.
- `public_ip` - The public IPv4.
- `public_ip_v6` - The public IPv6.
- `status` - The status of the node.

» Import

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

```
$ terraform import scaleway_k8s_cluster_beta.mycluster fr-par/11111111-1111-1111-1111-111111
```

» scaleway_k8s_pool_beta

Creates and manages Scaleway Kubernetes cluster pools. For more information, see the documentation.

» 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.
- `tags` - (Optional) The tags associated with the pool.
- `placement_group_id` - (Optional) The placement group 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 `region`) The region in which the pool should be created.
- `wait_for_pool_ready` - (Default to `false`) Whether to wait for the pool to be ready.

» Attributes Reference

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

- `id` - The ID of the pool.
- `status` - The status of the pool.
- `nodes` - (List of) The nodes in the default pool.
  - `name` - The name of the node.
  - `public_ip` - The public IPv4.
  - `public_ip_v6` - The public IPv6.
  - `status` - The status of the node.
- `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_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.

» 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 `region`) The region in which the load-balancer should be created.
- `organization_id` - (Defaults to provider `organization_id`) The ID of the organization the load-balancer 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_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.

» Examples

» Basic

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

» With HTTP Health Check

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

    health_check_http {
      uri: "www.test.com/health"
```

```
    }
}
```

» Arguments Reference

The following arguments are supported:

» Basic arguments

- `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`)

» Health Check arguments

Backends use Health Check to test if a backend server is ready to receive requests. You may use one of the following health check types: `TCP`, `HTTP` or `HTTPS`. (Default: `TCP`)

- `health_check_timeout` - (Default: `30s`) Timeout before we consider a HC request failed.
- `health_check_delay` - (Default: `60s`) Interval between two HC requests.
- `health_check_port` - (Default: `forward_port`) Port the HC requests will be send to.
- `health_check_max_retries` - (Default: `2`) Number of allowed failed HC requests before the backend server is marked down.

- `health_check_tcp` - (Optional) This block enable TCP health check. Only one of `health_check_tcp`, `health_check_http` and `health_check_https` should be specified.
- `health_check_http` - (Optional) This block enable HTTP health check. Only one of `health_check_tcp`, `health_check_http` and `health_check_https` should be specified.
  - `uri` - (Required) The HTTP endpoint URL to call for HC requests.
  - `method` - (Default: `GET`) The HTTP method to use for HC requests.
  - `code` - (Default: `200`) The expected HTTP status code.
- `health_check_https` - (Optional) This block enable HTTPS health check. Only one of `health_check_tcp`, `health_check_http` and `health_check_https` should be specified.
  - `uri` - (Required) The HTTPS endpoint URL to call for HC requests.
  - `method` - (Default: `GET`) The HTTP method to use for HC requests.
  - `code` - (Default: `200`) The expected HTTP status code.
- `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/11111111-1111-1111-1111-111111
```

» scaleway_lb_certificate_beta

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

Creates and manages Scaleway Load-Balancer Certificates. For more information, see the documentation.

» Examples

» Let's Encrypt

```
resource scaleway_lb_certificate_beta cert01 {
    lb_id = scaleway_lb_beta.lb01.id
    name = "cert1"
    letsencrypt {
        common_name = "example.org"
        subject_alternative_name = [
          "sub1.example.com",
          "sub2.example.com"
        ]
    }
}
```

» Custom Certificate

```
resource scaleway_lb_certificate_beta cert01 {
    lb_id = scaleway_lb_beta.lb01.id
    name = "custom-cert"
    custom_certificate {
        certificate_chain = <<EOF
CERTIFICATE_CHAIN_CONTENTS
EOF
    }
}
```

» Arguments Reference

The following arguments are supported:

» Basic arguments

- `lb_id` - (Required) The load-balancer ID this certificate is attached to.

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

- `name` - (Optional) The name of the certificate backend.
- `letsencrypt` - (Optional) Configuration block for Let's Encrypt configuration. Only one of `letsencrypt` and `custom_certificate` should be specified.
  - `common_name` - (Required) Main domain of the certificate.
  - `subject_alternative_name` - (Optional) Array of alternative domain names.

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

- `custom_certificate` - (Optional) Configuration block for custom certificate chain. Only one of `letsencrypt` and `custom_certificate` should be specified.
  - `certificate_chain` - (Required) Full PEM-formatted certificate chain.

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

» Attributes Reference

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

- `id` - The ID of the loadbalancer certificate.
- `common_name` - Main domain of the certificate
- `subject_alternative_name` - The alternative domain names of the certificate
- `fingerprint` - The identifier (SHA-1) of the certificate
- `not_valid_before` - The not valid before validity bound timestamp
- `not_valid_after` - The not valid after validity bound timestamp
- `status` - Certificate status

» Additional notes

- Ensure that all domain names used in configuration are pointing to the load balancer IP. You can achieve this by creating a DNS record through terraform pointing to `ip_adress` property of `lb_beta` entity
- In case there are any issues with the certificate, you will receive a `400` error from the `apply` operation. Use `export TF_LOG=DEBUG` to view exact problem returned by the api.
- Wildcards are not supported with Let's Encrypt yet.

» 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.

» Examples

» Basic

```
resource "scaleway_lb_frontend_beta" "frontend01" {
    lb_id = scaleway_lb_beta.lb01.id
    backend_id = scaleway_lb_backend_beta.backend01.id
    name = "frontend01"
    inbound_port = "80"
}
```

» With ACLs

```
resource scaleway_lb_frontend_beta frontend01 {
    lb_id = scaleway_lb_beta.lb01.id
    backend_id = scaleway_lb_backend_beta.backend01.id
    name = "frontend01"
    inbound_port = "80"

    # Allow downstream requests from: 192.168.0.1, 192.168.0.2 or 192.168.10.0/24
    acl {
        name = "blacklist wellknwon IPs"
        action {
            type = "allow"
        }
        match {
            ip_subnet = ["192.168.0.1", "192.168.0.2", "192.168.10.0/24"]
        }
    }

    # Deny downstream requests from: 51.51.51.51 that match "^foo*bar$"
    acl {
        action {
            type = "deny"
        }
        match {
            ip_subnet = ["51.51.51.51"]
            http_filter = "regex"
            http_filter_value = ["^foo*bar$"]
        }
    }

    # Allow downstream http requests that begins with "/foo" or "/bar"
    acl {
        action {
```

```
            type = "allow"
        }
        match {
            http_filter = "path_begin"
            http_filter_value = ["foo", "bar"]
        }
    }

    # Allow upstream http requests that DO NOT begins with "/hi"
    acl {
        action {
            type = "allow"
        }
        match {
            http_filter = "path_begin"
            http_filter_value = ["hi"]
            invert = "true"
        }
    }
}
```

» 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` - (Optional) Certificate ID that should be used by the frontend.
- `acl` - (Optional) A list of ACL rules to apply to the load-balancer frontend. Defined below.

» acl

- `name` - (Optional) The ACL name. If not provided it will be randomly generated.
- `action` - (Required) Action to undertake when an ACL filter matches.
  - `type` - (Required) The action type. Possible values are: `allow` or `deny`.
- `match` - (Required) The ACL match rule. At least `ip_subnet` or `http_filter` and `http_filter_value` are required.
  - `ip_subnet` - (Optional) A list of IPs or CIDR v4/v6 addresses of the client of the session to match.
  - `http_filter` - (Optional) The HTTP filter to match. This filter is supported only if your backend protocol has an HTTP forward protocol. It extracts the request's URL path, which starts at the first slash and ends before the question mark (without the host part). Possible values are: `acl_http_filter_none`, `path_begin`, `path_end` or `regex`.
  - `http_filter_value` - (Optional) A list of possible values to match for the given HTTP filter.
  - `invert` - (Optional) If set to `true`, the condition will be of type "unless".

» Attributes Reference

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

- `id` - The ID of the load-balancer frontend.

» Import

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

```
$ terraform import scaleway_lb_frontend_beta.frontend01 fr-par/11111111-1111-1111-1111-11111
```

» scaleway_rdb_instance_beta

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

Creates and manages Scaleway Database Instances. For more information, see the documentation.

» Examples

» Basic

```
resource scaleway_rdb_instance_beta main {
    name = "test-rdb"
    node_type = "db-dev-s"
    engine = "PostgreSQL-11"
    is_ha_cluster = true
    disable_backup = true
    user_name = "my_initial_user"
    password = "thiZ_is_v&ry_s3cret"
}
```

» Arguments Reference

The following arguments are supported:

- `node_type` - (Required) The type of database instance you want to create (e.g. `db-dev-s`).

Important: Updates to `node_type` will upgrade the Database Instance to the desired `node_type` without any interruption. Keep in mind that you cannot downgrade a Database Instance.

- `engine` - (Required) Database Instance's engine version (e.g. `PostgreSQL-11`).

Important: Updates to `engine` will recreate the Database Instance.

- `user_name` - (Required) Identifier for the first user of the database instance.

Important: Updates to `user_name` will recreate the Database Instance.

- `password` - (Required) Password for the first user of the database instance.

Important: Updates to `password` will recreate the Database Instance.

- `is_ha_cluster` - (Optional) Enable or disable high availability for the database instance.

Important: Updates to `is_ha_cluster` will recreate the Database Instance.

- `name` - (Optional) The name of the Database Instance.
- `disable_backup` - (Optional) Disable automated backup for the database instance.
- `tags` - (Optional) The tags associated with the Database Instance.
- `region` - (Defaults to provider `region`) The region in which the Database Instance should be created.

- `organization_id` - (Defaults to provider `organization_id`) The ID of the organization the Database Instance is associated with.

» Attributes Reference

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

- `id` - The ID of the Database Instance.
- `endpoint_ip` - The IP of the Database Instance.
- `endpoint_port` - The port of the Database Instance.
- `read_replicas` - List of read replicas of the database instance.
  - `ip` - IP of the replica.
  - `port` - Port of the replica.
  - `name` - Name of the replica.
- `certificate` - Certificate of the database instance.

» Import

Database Instance can be imported using the `{region}/{id}`, e.g.

```
$ terraform import scaleway_rdb_instance_beta.rdb01 fr-par/11111111-1111-1111-1111-11111111
```

» scaleway_marketplace_image_beta

Gets local image ID of an image from its label name.

» Example Usage

```
data "scaleway_marketplace_image_beta" "my_image" {
  label  = "ubuntu_bionic"
}
```

» Argument Reference

- `label` - (Required) Exact label of the desired image. You can use this endpoint to find the right `label`.
- `instance_type` - (Optional, default `DEV1-S`) The instance type the image is compatible with. You find all the available types on the pricing page.
- `zone` - (Defaults to provider `zone`) The zone in which the image exists.

» Attributes Reference

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

- `id` - The ID of the image.

» 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_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.

» 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_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.

» 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.

» 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.

» 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.

» 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.

» 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.

» 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

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.

» 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

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_bootscript

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

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_security_group

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

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_image

DEPRECATED: This resource is deprecated and will be removed in `v2.0+`. Please use `scaleway_instance_image` instead or `scaleway_marketplace_image_beta` depending on your usage.

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_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.