» huaweicloudstack_images_image_v2

Use this data source to get the ID of an available HuaweiCloudStack image.

» Example Usage

```
data "huaweicloudstack_images_image_v2" "ubuntu" {
  name        = "Ubuntu 16.04"
  visibility  = "public"
  most_recent = true
}
```

» Argument Reference

- `region` - (Optional) The region in which to obtain the V2 Glance client. A Glance client is needed to create an Image that can be used with a compute instance. If omitted, the `region` argument of the provider is used.
- `most_recent` - (Optional) If more than one result is returned, use the most recent image.
- `name` - (Optional) The name of the image.
- `owner` - (Optional) The owner (UUID) of the image.
- `size_min` - (Optional) The minimum size (in bytes) of the image to return.
- `size_max` - (Optional) The maximum size (in bytes) of the image to return.
- `sort_direction` - (Optional) Order the results in either `asc` or `desc`.
- `sort_key` - (Optional) Sort images based on a certain key. Must be one of ”name”, ”container_format”, ”disk_format”, ”status”, ”id” or ”size”. Defaults to `name`.
- `tag` - (Optional) Search for images with a specific tag.
- `visibility` - (Optional) The visibility of the image. Must be one of ”public”, ”private”, ”community”, or ”shared”. Defaults to `private`.

» Attributes Reference

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

- `checksum` - The checksum of the data associated with the image.
- `created_at` - The date the image was created.

- `container_format`: The format of the image's container.
- `disk_format`: The format of the image's disk.
- `file` - the trailing path after the glance endpoint that represent the location of the image or the path to retrieve it.
- `metadata` - The metadata associated with the image. Image metadata allow for meaningfully define the image properties and tags.
- `min_disk_gb` - The minimum amount of disk space required to use the image.
- `min_ram_mb` - The minimum amount of ram required to use the image.
- `properties` - Freeform information about the image.
- `protected` - Whether or not the image is protected.
- `schema` - The path to the JSON-schema that represent the image or image
- `size_bytes` - The size of the image (in bytes).
- `tags` - See Argument Reference above.
- `update_at` - The date the image was last updated.

» huaweicloudstack_networking_network_v2

Use this data source to get the ID of an available HuaweiCloudStack network.

» Example Usage

```
data "huaweicloudstack_networking_network_v2" "network" {
  name = "tf_test_network"
}
```

» Argument Reference

- `region` - (Optional) The region in which to obtain the V2 Neutron client. A Neutron client is needed to retrieve networks ids. If omitted, the `region` argument of the provider is used.
- `network_id` - (Optional) The ID of the network.
- `name` - (Optional) The name of the network.
- `status` - (Optional) The status of the network.
- `matching_subnet_cidr` - (Optional) The CIDR of a subnet within the network.
- `tenant_id` - (Optional) The owner of the network.

» Attributes Reference

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

- `admin_state_up` - (Optional) The administrative state of the network.
- `name` - See Argument Reference above.
- `region` - See Argument Reference above.
- `shared` - (Optional) Specifies whether the network resource can be accessed by any tenant or not.

» huaweicloudstack_networking_port_v2

Use this data source to get the ID of an available HuaweiCloudStack port.

» Example Usage

```
data "huaweicloudstack_networking_port_v2" "port_1" {
  name = "port_1"
}
```

» Argument Reference

- `region` - (Optional) The region in which to obtain the V2 Neutron client. A Neutron client is needed to retrieve port ids. If omitted, the `region` argument of the provider is used.
- `project_id` - (Optional) The owner of the port.
- `port_id` - (Optional) The ID of the port.
- `name` - (Optional) The name of the port.
- `admin_state_up` - (Optional) The administrative state of the port.
- `network_id` - (Optional) The ID of the network the port belongs to.
- `device_owner` - (Optional) The device owner of the port.
- `mac_address` - (Optional) The MAC address of the port.
- `device_id` - (Optional) The ID of the device the port belongs to.
- `fixed_ip` - (Optional) The port IP address filter.
- `status` - (Optional) The status of the port.

- `security_group_ids` - (Optional) The list of port security group IDs to filter.

» Attributes Reference

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

- `region` - See Argument Reference above.
- `project_id` - See Argument Reference above.
- `port_id` - See Argument Reference above.
- `name` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `network_id` - See Argument Reference above.
- `device_owner` - See Argument Reference above.
- `mac_address` - See Argument Reference above.
- `device_id` - See Argument Reference above.
- `all_fixed_ips` - The collection of Fixed IP addresses on the port in the order returned by the Network v2 API.
- `all_security_group_ids` - The set of security group IDs applied on the port.

» huaweicloudstack_networking_secgroup_v2

Use this data source to get the ID of an available HuaweiCloudStack security group.

» Example Usage

```
data "huaweicloudstack_networking_secgroup_v2" "secgroup" {
  name = "tf_test_secgroup"
}
```

» Argument Reference

- `region` - (Optional) The region in which to obtain the V2 Neutron client. A Neutron client is needed to retrieve security groups ids. If omitted, the `region` argument of the provider is used.
- `secgroup_id` - (Optional) The ID of the security group.
- `name` - (Optional) The name of the security group.
- `tenant_id` - (Optional) The owner of the security group.

» Attributes Reference

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

- `name` - See Argument Reference above.
- `description`- The description of the security group.
- `region` - See Argument Reference above.

» huaweicloudstack_networking_subnet_v2

Use this data source to get the ID of an available HuaweiCloudStack subnet.

» Example Usage

```
data "huaweicloudstack_networking_subnet_v2" "subnet_1" {
  name = "subnet_1"
}
```

» Argument Reference

- `region` - (Optional) The region in which to obtain the V2 Neutron client. A Neutron client is needed to retrieve subnet ids. If omitted, the `region` argument of the provider is used.
- `name` - (Optional) The name of the subnet.
- `dhcp_enabled` - (Optional) If the subnet has DHCP enabled.
- `dhcp_disabled` - (Optional) If the subnet has DHCP disabled.
- `ip_version` - (Optional) The IP version of the subnet (either 4 or 6).
- `gateway_ip` - (Optional) The IP of the subnet's gateway.

- `cidr` - (Optional) The CIDR of the subnet.
- `subnet_id` - (Optional) The ID of the subnet.
- `network_id` - (Optional) The ID of the network the subnet belongs to.
- `tenant_id` - (Optional) The owner of the subnet.

» Attributes Reference

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

- `allocation_pools` - Allocation pools of the subnet.
- `enable_dhcp` - Whether the subnet has DHCP enabled or not.
- `dns_nameservers` - DNS Nameservers of the subnet.
- `host_routes` - Host Routes of the subnet.
- `region` - See Argument Reference above.

» huaweicloudstack_as_configuration_v1

Manages a V1 AS Configuration resource within HuaweiCloudStack.

» Example Usage

» Basic AS Configuration

```
resource "huaweicloudstack_as_configuration_v1" "my_as_config" {
  scaling_configuration_name = "my_as_config"
  instance_config {
    flavor = "${var.flavor}"
    image  = "${var.image_id}"
    disk {
      size        = 40
      volume_type = "SATA"
      disk_type   = "SYS"
    }
    key_name  = "${var.keyname}"
    user_data = "${file("userdata.txt")}"
  }
}
```

» AS Configuration With User Data and Metadata

```
resource "huaweicloudstack_as_configuration_v1" "my_as_config" {
  scaling_configuration_name = "my_as_config"
  instance_config {
    flavor = "${var.flavor}"
    image  = "${var.image_id}"
    disk {
      size        = 40
      volume_type = "SATA"
      disk_type   = "SYS"
    }
    key_name  = "${var.keyname}"
    user_data = "${file("userdata.txt")}"
    metadata = {
      some_key = "some_value"
    }
  }
}
```

`user_data` can come from a variety of sources: inline, read in from the `file` function, or the `template_cloudinit_config` resource.

» AS Configuration uses the existing instance specifications as the template

```
resource "huaweicloudstack_as_configuration_v1" "my_as_config" {
  scaling_configuration_name = "my_as_config"
  instance_config {
    instance_id = "4579f2f5-cbe8-425a-8f32-53dcb9d9053a"
    key_name    = "${var.keyname}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the AS configuration. If omitted, the `region` argument of the provider is used. Changing this creates a new AS configuration.
- `scaling_configuration_name` - (Required) The name of the AS configuration. The name can contain letters, digits, underscores(_), and hyphens(-), and cannot exceed 64 characters.

- `instance_config` - (Required) The information about instance configurations. The instance_config dictionary data structure is documented below.

The `instance_config` block supports:

- `instance_id` - (Optional) When using the existing instance specifications as the template to create AS configurations, specify this argument. In this case, flavor, image, and disk arguments do not take effect. If the instance_id argument is not specified, flavor, image, and disk arguments are mandatory.
- `flavor` - (Optional) The flavor ID.
- `image` - (Optional) The image ID.
- `disk` - (Optional) The disk group information. System disks are mandatory and data disks are optional. The dick structure is described below.
- `key_name` - (Required) The name of the SSH key pair used to log in to the instance.
- `user_data` - (Optional) The user data to provide when launching the instance. The file content must be encoded with Base64.
- `personality` - (Optional) Customize the personality of an instance by defining one or more files and their contents. The personality structure is described below.
- `public_ip` - (Optional) The elastic IP address of the instance. The public_ip structure is described below.
- `metadata` - (Optional) Metadata key/value pairs to make available from within the instance.

The `disk` block supports:

- `size` - (Required) The disk size. The unit is GB. The system disk size ranges from 40 to 32768, and the data disk size ranges from 10 to 32768.
- `volume_type` - (Required) The disk type, which must be the same as the disk type available in the system. The available types are `SSD`, `SAS`, `SATA` or other types defined in CCS.
- `disk_type` - (Required) Whether the disk is a system disk or a data disk. Option `DATA` indicates a data disk. option `SYS` indicates a system disk.

The `personality` block supports:

- `path` - (Required) The absolute path of the destination file.
- `contents` - (Required) The content of the injected file, which must be encoded with base64.

The `public_ip` block supports:

- `eip` - (Required) The configuration parameter for creating an elastic IP address that will be automatically assigned to the instance. The eip structure is described below.

The `eip` block supports:

- `ip_type` - (Required) The IP address type. The system only supports `5_bgp` (indicates dynamic BGP).
- `bandwidth` - (Required) The bandwidth information. The structure is described below.

The `bandwidth` block supports:

- `size` - (Required) The bandwidth (Mbit/s). The value range is 1 to 300.
- `share_type` - (Required) The bandwidth sharing type. The system only supports `PER` (indicates exclusive bandwidth).
- `charging_mode` - (Required) The bandwidth charging mode. The system only supports `traffic`.

» huaweicloudstack_as_group_v1

Manages a V1 Autoscaling Group resource within HuaweiCloudStack.

» Example Usage

» Basic Autoscaling Group

```
resource "huaweicloudstack_as_group_v1" "my_as_group" {
  scaling_group_name       = "my_as_group"
  scaling_configuration_id = "37e310f5-db9d-446e-9135-c625f9c2bbfc"
  desire_instance_number   = 2
  min_instance_number      = 0
  max_instance_number      = 10
  vpc_id                   = "1d8f7e7c-fe04-4cf5-85ac-08b478c290e9"
  delete_publicip          = true
  delete_instances         = "yes"

  networks {
    id = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  }
  security_groups {
    id = "45e4c6de-6bf0-4843-8953-2babde3d4810"
  }
}
```

» Autoscaling Group Only Remove Members When Scaling Down

```
resource "huaweicloudstack_as_group_v1" "my_as_group_only_remove_members" {
  scaling_group_name       = "my_as_group_only_remove_members"
  scaling_configuration_id = "37e310f5-db9d-446e-9135-c625f9c2bbfc"
  desire_instance_number   = 2
  min_instance_number      = 0
  max_instance_number      = 10
  vpc_id                   = "1d8f7e7c-fe04-4cf5-85ac-08b478c290e9"
  delete_publicip          = true
  delete_instances         = "no"

  networks {
    id = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  }
  security_groups {
    id = "45e4c6de-6bf0-4843-8953-2babde3d4810"
  }
}
```

» Autoscaling Group With ELB Listener

```
resource "huaweicloudstack_lb_loadbalancer_v2" "loadbalancer_1" {
  name = "loadbalancer_1"
  vip_subnet_id = "d9415786-5f1a-428b-b35f-2f1523e146d2"
}

resource "huaweicloudstack_lb_listener_v2" "listener_1" {
  name = "listener_1"
  protocol = "HTTP"
  protocol_port = 8080
  loadbalancer_id = "${huaweicloudstack_lb_loadbalancer_v2.loadbalancer_1.id}"
}

resource "huaweicloudstack_lb_pool_v2" "pool_1" {
  name = "pool_1"
  protocol    = "HTTP"
  lb_method   = "ROUND_ROBIN"
  listener_id = "${huaweicloudstack_lb_listener_v2.listener_1.id}"
}

resource "huaweicloudstack_as_group_v1" "my_as_group_with_enhanced_lb"{
  scaling_group_name = "my_as_group_with_enhanced_lb"
  scaling_configuration_id = "37e310f5-db9d-446e-9135-c625f9c2bbfc"
  desire_instance_number   = 2
```

```
  min_instance_number      = 0
  max_instance_number      = 10
  vpc_id                   = "1d8f7e7c-fe04-4cf5-85ac-08b478c290e9"

  networks {
    id = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  }
  security_groups {
    id = "45e4c6de-6bf0-4843-8953-2babde3d4810"
  }
  lbaas_listeners {
    listener_id   = "${huaweicloudstack_lb_listener_v2.listener_1.id}"
    protocol_port = "${huaweicloudstack_lb_listener_v2.listener_1.protocol_port}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the AS group. If omitted, the `region` argument of the provider is used. Changing this creates a new AS group.
- `scaling_group_name` - (Required) The name of the scaling group. The name can contain letters, digits, underscores(_), and hyphens(-),and cannot exceed 64 characters.
- `scaling_configuration_id` - (Optional) The configuration ID which defines configurations of instances in the AS group.
- `desire_instance_number` - (Optional) The expected number of instances. The default value is the minimum number of instances. The value ranges from the minimum number of instances to the maximum number of instances.
- `min_instance_number` - (Optional) The minimum number of instances. The default value is 0.
- `max_instance_number` - (Optional) The maximum number of instances. The default value is 0.
- `cool_down_time` - (Optional) The cooling duration (in seconds). The value ranges from 0 to 86400, and is 900 by default.
- `lb_listener_id` - (Optional) The ELB (classic) listener IDs. The system supports up to three ELB listeners, the IDs of which are separated using a comma (,). This argument is deprecated, using `lbaas_listeners` instead.

- `lbaas_listeners` - (Optional) An array of one or more ELB (enhanced). The system supports the binding of up to three load balancers. The field is alternative to lb_listener_id. The lbaas_listeners object structure is documented below.
- `available_zones` - (Optional) The availability zones in which to create the instances in the autoscaling group.
- `networks` - (Required) An array of one or more network IDs. The system supports up to five networks. The networks object structure is documented below.
- `security_groups` - (Required) An array of one or more security group IDs to associate with the group. The security_groups object structure is documented below.
- `vpc_id` - (Required) The VPC ID. Changing this creates a new group.
- `health_periodic_audit_method` - (Optional) The health check method for instances in the AS group. The health check methods include `ELB_AUDIT` and `NOVA_AUDIT`. If load balancing is configured, the default value of this parameter is `ELB_AUDIT`. Otherwise, the default value is `NOVA_AUDIT`.
- `health_periodic_audit_time` - (Optional) The health check period for instances. The period has four options: 5 minutes (default), 15 minutes, 60 minutes, and 180 minutes.
- `instance_terminate_policy` - (Optional) The instance removal policy. The policy has four options: `OLD_CONFIG_OLD_INSTANCE` (default), `OLD_CONFIG_NEW_INSTANCE`, `OLD_INSTANCE`, and `NEW_INSTANCE`.
- `notifications` - (Optional) The notification mode. The system only supports `EMAIL` mode which refers to notification by email.
- `delete_publicip` - (Optional) Whether to delete the elastic IP address bound to the instances of AS group when deleting the instances. The options are `true` and `false`.
- `delete_instances` - (Optional) Whether to delete the instances in the AS group when deleting the AS group. The options are `yes` and `no`.

The `networks` block supports:

- `id` - (Required) The network UUID.

The `security_groups` block supports:

- `id` - (Required) The UUID of the security group.

The `lbaas_listeners` block supports:

- `listener_id` - (Required) Specifies the ELB listener ID.

- `protocol_port` - (Required) Specifies the backend protocol, which is the port on which a backend ECS listens for traffic. The number of the port ranges from 1 to 65535.
- `weight` - (Optional) Specifies the weight, which determines the portion of requests a backend ECS processes compared to other backend ECSs added to the same listener. The value of this parameter ranges from 0 to 100. The default value is 1.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `scaling_group_name` - See Argument Reference above.
- `scaling_group_status` - The status of the AS group.
- `current_instance_number` - The number of current instances in the AS group.
- `desire_instance_number` - See Argument Reference above.
- `min_instance_number` - See Argument Reference above.
- `max_instance_number` - See Argument Reference above.
- `cool_down_time` - See Argument Reference above.
- `lb_listener_id` - See Argument Reference above.
- `lbaas_listeners` - See Argument Reference above.
- `health_periodic_audit_method` - See Argument Reference above.
- `health_periodic_audit_time` - See Argument Reference above.
- `instance_terminate_policy` - See Argument Reference above.
- `scaling_configuration_id` - See Argument Reference above.
- `delete_publicip` - See Argument Reference above.
- `notifications` - See Argument Reference above.
- `instances` - The instances IDs of the AS group.

» huaweicloudstack_as_policy_v1

Manages a V1 AS Policy resource within HuaweiCloudStack.

» Example Usage

» AS Recurrence Policy

```
resource "huaweicloudstack_as_policy_v1" "hth_aspolicy" {
  scaling_policy_name = "hth_aspolicy"
  scaling_group_id    = "4579f2f5-cbe8-425a-8f32-53dcb9d9053a"
  cool_down_time      = 900
```

```
  scaling_policy_type = "RECURRENCE"
  scaling_policy_action {
    operation       = "ADD"
    instance_number = 1
  }
  scheduled_policy {
    launch_time     = "07:00"
    recurrence_type = "Daily"
    start_time      = "2017-11-30T12:00Z"
    end_time        = "2017-12-30T12:00Z"
  }
}
```

» AS Scheduled Policy

```
resource "huaweicloudstack_as_policy_v1" "hth_aspolicy_1" {
  scaling_policy_name = "hth_aspolicy_1"
  scaling_group_id    = "4579f2f5-cbe8-425a-8f32-53dcb9d9053a"
  cool_down_time      = 900
  scaling_policy_type = "SCHEDULED"
  scaling_policy_action {
    operation       = "REMOVE"
    instance_number = 1
  }
  scheduled_policy {
    launch_time = "2017-12-22T12:00Z"
  }
}
```

Please note that the `launch_time` of the `SCHEDULED` policy cannot be earlier than the current time.

» AS Alarm Policy

```
resource "huaweicloudstack_as_policy_v1" "hth_aspolicy_2" {
  scaling_policy_name = "hth_aspolicy_2"
  scaling_group_id    = "4579f2f5-cbe8-425a-8f32-53dcb9d9053a"
  cool_down_time      = 900
  scaling_policy_type = "ALARM"
  alarm_id            = "37e310f5-db9d-446e-9135-c625f9c2bbfc"
  scaling_policy_action {
    operation       = "ADD"
    instance_number = 1
  }
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the AS policy. If omitted, the `region` argument of the provider is used. Changing this creates a new AS policy.
- `scaling_policy_name` - (Required) The name of the AS policy. The name can contain letters, digits, underscores(_), and hyphens(-),and cannot exceed 64 characters.
- `scaling_group_id` - (Required) The AS group ID. Changing this creates a new AS policy.
- `scaling_policy_type` - (Required) The AS policy type. The values can be `ALARM`, `SCHEDULED`, and `RECURRENCE`.
- `alarm_id` - (Optional) The alarm rule ID. This argument is mandatory when `scaling_policy_type` is set to `ALARM`.
- `scheduled_policy` - (Optional) The periodic or scheduled AS policy. This argument is mandatory when `scaling_policy_type` is set to `SCHEDULED` or `RECURRENCE`. The scheduled_policy structure is documented below.
- `scaling_policy_action` - (Optional) The action of the AS policy. The scaling_policy_action structure is documented below.
- `cool_down_time` - (Optional) The cooling duration (in seconds), and is 900 by default.

The `scheduled_policy` block supports:

- `launch_time` - (Required) The time when the scaling action is triggered. If `scaling_policy_type` is set to `SCHEDULED`, the time format is YYYY-MM-DDThh:mmZ. If `scaling_policy_type` is set to `RECURRENCE`, the time format is hh:mm.
- `recurrence_type` - (Optional) The periodic triggering type. This argument is mandatory when `scaling_policy_type` is set to `RECURRENCE`. The options include `Daily`, `Weekly`, and `Monthly`.
- `recurrence_value` - (Optional) The frequency at which scaling actions are triggered.
- `start_time` - (Optional) The start time of the scaling action triggered periodically. The time format complies with UTC. The current time is used by default. The time format is YYYY-MM-DDThh:mmZ.
- `end_time` - (Optional) The end time of the scaling action triggered periodically. The time format complies with UTC. This argument is mandatory

when `scaling_policy_type` is set to `RECURRENCE`. The time format is YYYY-MM-DDThh:mmZ.

The `scaling_policy_action` block supports:

- `operation` - (Optional) The operation to be performed. The options include `ADD` (default), `REMOVE`, and `SET`.
- `instance_number` - (Optional) The number of instances to be operated. The default number is 1.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `scaling_policy_name` - See Argument Reference above.
- `scaling_policy_type` - See Argument Reference above.
- `alarm_id` - See Argument Reference above.
- `cool_down_time` - See Argument Reference above.
- `scaling_policy_action/operation` - See Argument Reference above.
- `scaling_policy_action/instance_number` - See Argument Reference above.
- `scheduled_policy/launch_time` - See Argument Reference above.
- `scheduled_policy/recurrence_type` - See Argument Reference above.
- `scheduled_policy/recurrence_value` - See Argument Reference above.
- `scheduled_policy/start_time` - See Argument Reference above.
- `scheduled_policy/end_time` - See Argument Reference above.

» huaweicloudstack_blockstorage_volume_v2

Manages a V2 volume resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_blockstorage_volume_v2" "volume_1" {
  region      = "RegionOne"
  name        = "volume_1"
  description = "first test volume"
  size        = 3
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the volume. If omitted, the `region` argument of the provider is used. Changing this creates a new volume.
- `size` - (Required) The size of the volume to create (in gigabytes). Changing this creates a new volume.
- `availability_zone` - (Optional) The availability zone for the volume. Changing this creates a new volume.
- `consistency_group_id` - (Optional) The consistency group to place the volume in.
- `description` - (Optional) A description of the volume. Changing this updates the volume's description.
- `image_id` - (Optional) The image ID from which to create the volume. Changing this creates a new volume.
- `metadata` - (Optional) Metadata key/value pairs to associate with the volume. Changing this updates the existing volume metadata.
- `name` - (Optional) A unique name for the volume. Changing this updates the volume's name.
- `snapshot_id` - (Optional) The snapshot ID from which to create the volume. Changing this creates a new volume.
- `source_replica` - (Optional) The volume ID to replicate with.
- `source_vol_id` - (Optional) The volume ID from which to create the volume. Changing this creates a new volume.
- `volume_type` - (Optional) The type of volume to create. Available types are `SSD`, `SAS` and `SATA`. Changing this creates a new volume.
- `cascade` - (Optional, Default:false) Specifies to delete all snapshots associated with the EVS disk.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `size` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `availability_zone` - See Argument Reference above.

- `image_id` - See Argument Reference above.
- `source_vol_id` - See Argument Reference above.
- `snapshot_id` - See Argument Reference above.
- `metadata` - See Argument Reference above.
- `volume_type` - See Argument Reference above.
- `attachment` - If a volume is attached to an instance, this attribute will display the Attachment ID, Instance ID, and the Device as the Instance sees it.

» Import

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

```
$ terraform import huaweicloudstack_blockstorage_volume_v2.volume_1 ea257959-eeb1-4c10-8d33-
```

» huaweicloudstack_compute_floatingip_associate_v2

Associate a floating IP to an instance. This can be used instead of the `floating_ip` options in `huaweicloudstack_compute_instance_v2`.

» Example Usage

» Automatically detect the correct network

```
resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  image_id        = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id       = 3
  key_pair        = "my_key_pair_name"
  security_groups = ["default"]
}

resource "huaweicloudstack_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "huaweicloudstack_compute_floatingip_associate_v2" "fip_1" {
  floating_ip = "${huaweicloudstack_networking_floatingip_v2.fip_1.address}"
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
}
```

» Explicitly set the network to attach to

```
resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  image_id        = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id       = 3
  key_pair        = "my_key_pair_name"
  security_groups = ["default"]

  network {
    name = "my_network"
  }

  network {
    name = "default"
  }
}

resource "huaweicloudstack_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "huaweicloudstack_compute_floatingip_associate_v2" "fip_1" {
  floating_ip = "${huaweicloudstack_networking_floatingip_v2.fip_1.address}"
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  fixed_ip    = "${huaweicloudstack_compute_instance_v2.instance_1.network.1.fixed_ip_v4}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Compute client. Keypairs are associated with accounts, but a Compute client is needed to create one. If omitted, the `region` argument of the provider is used. Changing this creates a new floatingip_associate.
- `floating_ip` - (Required) The floating IP to associate.
- `instance_id` - (Required) The instance to associte the floating IP with.
- `fixed_ip` - (Optional) The specific IP address to direct traffic to.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `floating_ip` - See Argument Reference above.
- `instance_id` - See Argument Reference above.
- `fixed_ip` - See Argument Reference above.

» Import

This resource can be imported by specifying all three arguments, separated by a forward slash:

```
$ terraform import huaweicloudstack_compute_floatingip_associate_v2.fip_1 <floating_ip>/<ins
```

» huaweicloudstack_compute_instance_v2

Manages a V2 VM instance resource within HuaweiCloudStack.

» Example Usage

» Basic Instance

```
resource "huaweicloudstack_compute_instance_v2" "basic" {
  name              = "basic"
  image_id          = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  metadata = {
    this = "that"
  }

  network {
    uuid = "55534eaa-533a-419d-9b40-ec427ea7195a"
  }
}
```

» Instance With Attached Volume

```
resource "huaweicloudstack_blockstorage_volume_v2" "myvol" {
  name = "myvol"
  size = 1
```

```
}

resource "huaweicloudstack_compute_instance_v2" "myinstance" {
  name              = "myinstance"
  image_id          = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  network {
    uuid = "55534eaa-533a-419d-9b40-ec427ea7195a"
  }
}

resource "huaweicloudstack_compute_volume_attach_v2" "attached" {
  compute_id = "${huaweicloudstack_compute_instance_v2.myinstance.id}"
  volume_id  = "${huaweicloudstack_blockstorage_volume_v2.myvol.id}"
}
```

» Boot From Volume

```
resource "huaweicloudstack_compute_instance_v2" "boot-from-volume" {
  name              = "boot-from-volume"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  block_device {
    uuid                  = "<image-id>"
    source_type           = "image"
    volume_size           = 5
    boot_index            = 0
    destination_type      = "volume"
    delete_on_termination = true
  }

  network {
    uuid = "55534eaa-533a-419d-9b40-ec427ea7195a"
  }
}
```

» Boot From an Existing Volume

```
resource "huaweicloudstack_blockstorage_volume_v1" "myvol" {
  name     = "myvol"
  size     = 5
  image_id = "<image-id>"
}

resource "huaweicloudstack_compute_instance_v2" "boot-from-volume" {
  name              = "bootfromvolume"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  block_device {
    uuid                  = "${huaweicloudstack_blockstorage_volume_v1.myvol.id}"
    source_type           = "volume"
    boot_index            = 0
    destination_type      = "volume"
    delete_on_termination = true
  }

  network {
    uuid = "55534eaa-533a-419d-9b40-ec427ea7195a"
  }
}
```

» Boot Instance, Create Volume, and Attach Volume as a Block Device

```
resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name              = "instance_1"
  image_id          = "<image-id>"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  block_device {
    uuid                  = "<image-id>"
    source_type           = "image"
    destination_type      = "local"
    boot_index            = 0
    delete_on_termination = true
```

```
  }

  block_device {
    source_type           = "blank"
    destination_type      = "volume"
    volume_size           = 1
    boot_index            = 1
    delete_on_termination = true
  }
}
```

» Boot Instance and Attach Existing Volume as a Block Device

```
resource "huaweicloudstack_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name              = "instance_1"
  image_id          = "<image-id>"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  block_device {
    uuid                  = "<image-id>"
    source_type           = "image"
    destination_type      = "local"
    boot_index            = 0
    delete_on_termination = true
  }

  block_device {
    uuid                  = "${huaweicloudstack_blockstorage_volume_v2.volume_1.id}"
    source_type           = "volume"
    destination_type      = "volume"
    boot_index            = 1
    delete_on_termination = true
  }
}
```

» Instance With Multiple Networks

```
resource "huaweicloudstack_networking_floatingip_v2" "myip" {
  pool = "admin_external_net"
}

resource "huaweicloudstack_compute_instance_v2" "multi-net" {
  name              = "multi-net"
  image_id          = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  network {
    uuid = "55534eaa-533a-419d-9b40-ec427ea7195a"
  }

  network {
    name = "my_second_network"
  }
}

resource "huaweicloudstack_compute_floatingip_associate_v2" "myip" {
  floating_ip = "${huaweicloudstack_networking_floatingip_v2.myip.address}"
  instance_id = "${huaweicloudstack_compute_instance_v2.multi-net.id}"
  fixed_ip    = "${huaweicloudstack_compute_instance_v2.multi-net.network.1.fixed_ip_v4}"
}
```

» Instance with Multiple Ephemeral Disks

```
resource "huaweicloudstack_compute_instance_v2" "multi-eph" {
  name              = "multi_eph"
  image_id          = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  block_device {
    boot_index            = 0
    delete_on_termination = true
    destination_type      = "local"
    source_type           = "image"
    uuid                  = "<image-id>"
```

```
  }

  block_device {
    boot_index            = -1
    delete_on_termination = true
    destination_type      = "local"
    source_type           = "blank"
    volume_size           = 1
  }

  block_device {
    boot_index            = -1
    delete_on_termination = true
    destination_type      = "local"
    source_type           = "blank"
    volume_size           = 1
  }
}
```

» Instance with User Data (cloud-init)

```
resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name              = "basic"
  image_id          = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"
  user_data         = "#cloud-config\nhostname: instance_1.example.com\nfqdn: instance_1.exa

  network {
    uuid = "55534eaa-533a-419d-9b40-ec427ea7195a"
  }
}
```

`user_data` can come from a variety of sources: inline, read in from the `file` function, or the `template_cloudinit_config` resource.

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the server instance. If omitted, the `region` argument of the provider is used. Changing this creates a new server.

- `name` - (Required) A unique name for the resource.
- `image_id` - (Optional; Required if `image_name` is empty and not booting from a volume. Do not specify if booting from a volume.) The image ID of the desired image for the server. Changing this creates a new server.
- `image_name` - (Optional; Required if `image_id` is empty and not booting from a volume. Do not specify if booting from a volume.) The name of the desired image for the server. Changing this creates a new server.
- `flavor_id` - (Optional; Required if `flavor_name` is empty) The flavor ID of the desired flavor for the server. Changing this resizes the existing server.
- `flavor_name` - (Optional; Required if `flavor_id` is empty) The name of the desired flavor for the server. Changing this resizes the existing server.
- `user_data` - (Optional) The user data to provide when launching the instance. Changing this creates a new server.
- `security_groups` - (Optional) An array of one or more security group names to associate with the server. Changing this results in adding/removing security groups from the existing server. *Note*: When attaching the instance to networks using Ports, place the security groups on the Port and not the instance.
- `availability_zone` - (Optional) The availability zone in which to create the server. Changing this creates a new server.
- `network` - (Optional) An array of one or more networks to attach to the instance. The network object structure is documented below. Changing this creates a new server.
- `metadata` - (Optional) Metadata key/value pairs to make available from within the instance. Changing this updates the existing server metadata.
- `config_drive` - (Optional) Whether to use the config_drive feature to configure the instance. Changing this creates a new server.
- `admin_pass` - (Optional) The administrative password to assign to the server. Changing this changes the root password on the existing server.
- `key_pair` - (Optional) The name of a key pair to put on the server. The key pair must already be created and associated with the tenant's account. Changing this creates a new server.
- `block_device` - (Optional) Configuration of block devices. The block_device structure is documented below. Changing this creates a new server. You can specify multiple block devices which will create an instance with multiple disks. This configuration is very flexible, so please see the following reference for more information.

- `scheduler_hints` - (Optional) Provide the Nova scheduler with hints on how the instance should be launched. The available hints are described below.
- `stop_before_destroy` - (Optional) Whether to try stop instance gracefully before destroying it, thus giving chance for guest OS daemons to stop correctly. If instance doesn't stop within timeout, it will be destroyed anyway.

The `network` block supports:

- `uuid` - (Required unless `port` or `name` is provided) The network UUID to attach to the server. Changing this creates a new server.
- `name` - (Required unless `uuid` or `port` is provided) The human-readable name of the network. Changing this creates a new server.
- `port` - (Required unless `uuid` or `name` is provided) The port UUID of a network to attach to the server. Changing this creates a new server.
- `fixed_ip_v4` - (Optional) Specifies a fixed IPv4 address to be used on this network. Changing this creates a new server.
- `fixed_ip_v6` - (Optional) Specifies a fixed IPv6 address to be used on this network. Changing this creates a new server.
- `access_network` - (Optional) Specifies if this network should be used for provisioning access. Accepts true or false. Defaults to false.

The `block_device` block supports:

- `uuid` - (Required unless `source_type` is set to `"blank"`) The UUID of the image, volume, or snapshot. Changing this creates a new server.
- `source_type` - (Required) The source type of the device. Must be one of "blank", "image", "volume", or "snapshot". Changing this creates a new server.
- `volume_size` - The size of the volume to create (in gigabytes). Required in the following combinations: source=image and destination=volume, source=blank and destination=local, and source=blank and destination=volume. Changing this creates a new server.
- `boot_index` - (Optional) The boot index of the volume. It defaults to 0. Changing this creates a new server.
- `destination_type` - (Optional) The type that gets created. Possible values are "volume" and "local". Changing this creates a new server.
- `delete_on_termination` - (Optional) Delete the volume / block device upon termination of the instance. Defaults to false. Changing this creates a new server.

The `scheduler_hints` block supports:

- `group` - (Optional) A UUID of a Server Group. The instance will be placed into that group.
- `different_host` - (Optional) A list of instance UUIDs. The instance will be scheduled on a different host than all other instances.
- `same_host` - (Optional) A list of instance UUIDs. The instance will be scheduled on the same host of those specified.
- `query` - (Optional) A conditional query that a compute node must pass in order to host an instance.
- `target_cell` - (Optional) The name of a cell to host the instance.
- `build_near_host_ip` - (Optional) An IP Address in CIDR form. The instance will be placed on a compute node that is in the same subnet.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `access_ip_v4` - The first detected Fixed IPv4 address *or* the Floating IP.
- `access_ip_v6` - The first detected Fixed IPv6 address.
- `metadata` - See Argument Reference above.
- `security_groups` - See Argument Reference above.
- `flavor_id` - See Argument Reference above.
- `flavor_name` - See Argument Reference above.
- `network/uuid` - See Argument Reference above.
- `network/name` - See Argument Reference above.
- `network/port` - See Argument Reference above.
- `network/fixed_ip_v4` - The Fixed IPv4 address of the Instance on that network.
- `network/fixed_ip_v6` - The Fixed IPv6 address of the Instance on that network.
- `network/mac` - The MAC address of the NIC on that network.
- `all_metadata` - Contains all instance metadata, even metadata not set by Terraform.

» Notes

» Multiple Ephemeral Disks

It's possible to specify multiple `block_device` entries to create an instance with multiple ephemeral (local) disks. In order to create multiple ephemeral disks,

the sum of the total amount of ephemeral space must be less than or equal to what the chosen flavor supports.

The following example shows how to create an instance with multiple ephemeral disks:

```
resource "huaweicloudstack_compute_instance_v2" "foo" {
  name            = "terraform-test"
  security_groups = ["default"]

  block_device {
    boot_index            = 0
    delete_on_termination = true
    destination_type      = "local"
    source_type           = "image"
    uuid                  = "<image uuid>"
  }

  block_device {
    boot_index            = -1
    delete_on_termination = true
    destination_type      = "local"
    source_type           = "blank"
    volume_size           = 1
  }

  block_device {
    boot_index            = -1
    delete_on_termination = true
    destination_type      = "local"
    source_type           = "blank"
    volume_size           = 1
  }
}
```

» Instances and Ports

Neutron Ports are a great feature and provide a lot of functionality. However, there are some notes to be aware of when mixing Instances and Ports:

- When attaching an Instance to one or more networks using Ports, place the security groups on the Port and not the Instance. If you place the security groups on the Instance, the security groups will not be applied upon creation, but they will be applied upon a refresh. This is a known HuaweiCloud bug.
- Network IP information is not available within an instance for networks

that are attached with Ports. This is mostly due to the flexibility Neutron Ports provide when it comes to IP addresses. For example, a Neutron Port can have multiple Fixed IP addresses associated with it. It's not possible to know which single IP address the user would want returned to the Instance's state information. Therefore, in order for a Provisioner to connect to an Instance via it's network Port, customize the `connection` information:

```
resource "huaweicloudstack_networking_port_v2" "port_1" {
  name           = "port_1"
  admin_state_up = "true"

  network_id = "0a1d0a27-cffa-4de3-92c5-9d3fd3f2e74d"

  security_group_ids = [
    "2f02d20a-8dca-49b7-b26f-b6ce9fddaf4f",
    "ca1e5ed7-dae8-4605-987b-fadaeeb30461",
  ]
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name = "instance_1"

  network {
    port = "${huaweicloudstack_networking_port_v2.port_1.id}"
  }

  connection {
    user        = "root"
    host        = "${huaweicloudstack_networking_port_v2.port_1.fixed_ip.0.ip_address}"
    private_key = "~/path/to/key"
  }

  provisioner "remote-exec" {
    inline = [
      "echo terraform executed > /tmp/foo",
    ]
  }
}
```

» huaweicloudstack_compute_interface_attach_v2

Attaches a Network Interface (a Port) to an Instance using the HuaweiCloudStack Compute (Nova) v2 API.

» Example Usage

» Basic Attachment

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloudstack_compute_interface_attach_v2" "ai_1" {
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  network_id  = "${huaweicloudstack_networking_port_v2.network_1.id}"
}
```

» Attachment Specifying a Fixed IP

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloudstack_compute_interface_attach_v2" "ai_1" {
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  network_id  = "${huaweicloudstack_networking_port_v2.network_1.id}"
  fixed_ip    = "10.0.10.10"
}
```

» Attachment Using an Existing Port

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}
```

```
resource "huaweicloudstack_networking_port_v2" "port_1" {
  name           = "port_1"
  network_id     = "${huaweicloudstack_networking_network_v2.network_1.id}"
  admin_state_up = "true"
}


resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloudstack_compute_interface_attach_v2" "ai_1" {
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  port_id     = "${huaweicloudstack_networking_port_v2.port_1.id}"
}
```

» Attaching Multiple Interfaces

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_port_v2" "ports" {
  count          = 2
  name           = "${format("port-%02d", count.index + 1)}"
  network_id     = "${huaweicloudstack_networking_network_v2.network_1.id}"
  admin_state_up = "true"
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloudstack_compute_interface_attach_v2" "attachments" {
  count          = 2
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  port_id     = "${huaweicloudstack_networking_port_v2.ports.*.id[count.index]}"
}
```

Note that the above example will not guarantee that the ports are attached in a deterministic manner. The ports will be attached in a seemingly random order.

If you want to ensure that the ports are attached in a given order, create explicit

dependencies between the ports, such as:

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_port_v2" "ports" {
  count          = 2
  name           = "${format("port-%02d", count.index + 1)}"
  network_id     = "${huaweicloudstack_networking_network_v2.network_1.id}"
  admin_state_up = "true"
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloudstack_compute_interface_attach_v2" "ai_1" {
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  port_id     = "${huaweicloudstack_networking_port_v2.ports.*.id[0]}"
}

resource "huaweicloudstack_compute_interface_attach_v2" "ai_2" {
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  port_id     = "${huaweicloudstack_networking_port_v2.ports.*.id[1]}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the interface attachment. If omitted, the `region` argument of the provider is used. Changing this creates a new attachment.
- `instance_id` - (Required) The ID of the Instance to attach the Port or Network to.
- `port_id` - (Optional) The ID of the Port to attach to an Instance. *NOTE*: This option and `network_id` are mutually exclusive.
- `network_id` - (Optional) The ID of the Network to attach to an Instance. A port will be created automatically. *NOTE*: This option and `port_id` are mutually exclusive.

- `fixed_ip` - (Optional) An IP address to assosciate with the port. *NOTE*: This option cannot be used with port_id. You must specifiy a network_id. The IP address must lie in a range on the supplied network.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `instance_id` - See Argument Reference above.
- `port_id` - See Argument Reference above.
- `network_id` - See Argument Reference above.
- `fixed_ip` - See Argument Reference above.

» Import

Interface Attachments can be imported using the Instance ID and Port ID separated by a slash, e.g.

```
$ terraform import huaweicloudstack_compute_interface_attach_v2.ai_1 89c60255-9bd6-460c-822a
```

» huaweicloudstack_compute_keypair_v2

Manages a V2 keypair resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_compute_keypair_v2" "test-keypair" {
  name       = "my-keypair"
  public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDAjpC1hwiOCCmKEWxJ4qzTTsJbKzndLotBCz5
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Compute client. Keypairs are associated with accounts, but a Compute client is needed to create one. If omitted, the `region` argument of the provider is used. Changing this creates a new keypair.
- `name` - (Required) A unique name for the keypair. Changing this creates a new keypair.

- `public_key` - (Required) A pregenerated OpenSSH-formatted public key. Changing this creates a new keypair.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `public_key` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloudstack_compute_keypair_v2.my-keypair test-keypair
```

» huaweicloudstack_compute_servergroup_v2

Manages a V2 Server Group resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_compute_servergroup_v2" "test-sg" {
  name     = "my-sg"
  policies = ["anti-affinity"]
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Compute client. If omitted, the `region` argument of the provider is used. Changing this creates a new server group.
- `name` - (Required) A unique name for the server group. Changing this creates a new server group.
- `policies` - (Required) The set of policies for the server group. Only two two policies are available right now, and both are mutually exclusive. See the Policies section for more information. Changing this creates a new server group.

» Policies

- `affinity` - All instances/servers launched in this group will be hosted on the same compute node.
- `anti-affinity` - All instances/servers launched in this group will be hosted on different compute nodes.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `policies` - See Argument Reference above.
- `members` - The instances that are part of this server group.

» Import

Server Groups can be imported using the `id`, e.g.

```
$ terraform import huaweicloudstack_compute_servergroup_v2.test-sg 1bc30ee9-9d5b-4c30-bdd5-7
```

» huaweicloudstack_compute_volume_attach_v2

Attaches a Block Storage Volume to an Instance using the HuaweiCloudStack Compute (Nova) v2 API.

» Example Usage

» Basic attachment of a single volume to a single instance

```
resource "huaweicloudstack_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloudstack_compute_volume_attach_v2" "va_1" {
```

```
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  volume_id   = "${huaweicloudstack_blockstorage_volume_v2.volume_1.id}"
}
```

» Attaching multiple volumes to a single instance

```
resource "huaweicloudstack_blockstorage_volume_v2" "volumes" {
  count = 2
  name  = "${format("vol-%02d", count.index + 1)}"
  size  = 1
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloudstack_compute_volume_attach_v2" "attachments" {
  count       = 2
  instance_id = "${huaweicloudstack_compute_instance_v2.instance_1.id}"
  volume_id   = "${element(huaweicloudstack_blockstorage_volume_v2.volumes.*.id, count.index
}

output "volume devices" {
  value = "${huaweicloudstack_compute_volume_attach_v2.attachments.*.device}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Compute client. A Compute client is needed to create a volume attachment. If omitted, the `region` argument of the provider is used. Changing this creates a new volume attachment.
- `instance_id` - (Required) The ID of the Instance to attach the Volume to.
- `volume_id` - (Required) The ID of the Volume to attach to an Instance.
- `device` - (Optional) The device of the volume attachment (ex: `/dev/vdc`). *NOTE*: Being able to specify a device is dependent upon the hypervisor in use. There is a chance that the device specified in Terraform will not be the same device the hypervisor chose. If this happens, Terraform will wish to

update the device upon subsequent applying which will cause the volume to be detached and reattached indefinitely. Please use with caution.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `instance_id` - See Argument Reference above.
- `volume_id` - See Argument Reference above.
- `device` - See Argument Reference above. *NOTE*: The correctness of this information is dependent upon the hypervisor in use. In some cases, this should not be used as an authoritative piece of information.

» Import

Volume Attachments can be imported using the Instance ID and Volume ID separated by a slash, e.g.

```
$ terraform import huaweicloudstack_compute_volume_attach_v2.va_1 89c60255-9bd6-460c-822a-e2
```

» huaweicloudstack_lb_certificate_v2

Manages a V2 certificate resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_certificate_v2" "certificate_1" {
  name = "certificate_1"
  description = "terraform test certificate"
  domain = "www.elb.com"
  private_key = <<EOT
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEAwZ5UJULAjWr7p6FVwGRQRjFN2s8tZ/6LC3X82fajpVsYqF1x
qEuUDndDXVD09E4u83MS6HO6a3bIVQDp6/klnYldiE6Vp8HH5BSKaCWKVg8lGWg1
UM9wZFnlryi14KgmpIFmcu9nA8yV/6MZAe6RSDmb3iyNBmiZ8aZhGw2pI1YwR+15
MVqFFGB+7ExkziROi7L8CFCyCezK2/oOOvQsH1dzQ8z1JXWdg8/9Zx7Ktvgwu5PQ
M3cJtSHX6iBPOkMU8Z8TugLlTqQXKZOEgwajwvQ5mf2DPkVgMO8XAgaLJcLigwD5
13koAdtJd5v+9irw+5LAuO3JclqwTvwy7u/YwwIDAQABAoIBACU9S5fjD9/jTMXA
DRsO8A+gGgZUxLn0xk+NAPX3LyB1tfdkCaFB8BccLzO6h3KZuwQOBPv6jkdvEDbx
Nwyw3eA/9GJsIvKiHc0rejdvyPymaw9I8MA7NbXHaJrY7KpqDQyk6sx+aUTcy5jg
iMXLWdwXYHhJ/1HVOo603oZyiS6HZeYU089NDUcX+1SJi3e5Ke0gPVXEqCq1O11/
```

```
rh24bMxnwZo4PKBWdcMBN5Zf/4ij9vrZE+fFzW7vGBO48A5lvZxWU2U5t/OZQRtN
1uLOHmMFaOFIF2aWbTVfwdUWAFsvAOkHj9VV8BXOUwKOUuEktdkfAlvrxmsFrO/H
yDeYYPkCgYEA/S55CBbROsMXpSZ56uRn8JHApZJhgkgvYr+FqDlJq/e92nAzf01P
RoEBUajwrnf1ycevN/SDfbtWzq2XJGqhWdJmtpO16b7KBsC6BdRcH6dnOYh31jgA
vABMIP3wzI4zSVTyxRE8LDuboytF1mSCeV5tHYPQTZNwrplDnLQhywcCgYEAw8Yc
Uk/eiFr3hfH/ZohMfV5p82Qp7DNIGRzw8YtVG/3+vNXrAXW1VhugNhQY6L+zLtJC
aKn84ooup0m3YCg0hvINqJuvzfsuzQgtjTXyaE0cEwsjUusOmiuj09vVx/3U7siK
Hdjd2ICPCvQ6Q8tdi8jV320gMs05AtaBkZdsiWUCgYEAtLw4Kk4f+xTKDFsrLUNf
75wcqhWVBiwBp7yQ7UX4EYsJPKZcHMRTk0EEcAbpyaJZE3I44vjp5ReXIHNLMfPs
uvI34J4Rfot0LN3n7cFrAi2+wpNo+MOBwrNzpRmijGP2uKKrq4JiMjFbKV/6utGF
Up7VxfwS904JYpqGaZctiIECgYA1A6nZtF0riY6ry/uAdXpZHL8ONNqRZtWoT0kD
79otSVu5ISiRbaGcXsDExC52oKrSDAgFtbqQUiEOFg09UcXfoR6HwRkba2CiDwve
yHQLQI5Qrdxz8Mk0gIrNrSM4FAmcW9vi9z4kCbQyoC5C+4gqeUlJRpDIkQBWP2Y4
2ct/bQKBgHv8qCsQTZphOxc31BJPa2xVhuv18cEU3XLUrVfUZ/1f43JhLp7gynS2
ep++LKUi9D0VGXY8bqvfJjbECoCeu85vl8NpCXwe/LoVoIn+7KaVIZMwqoGMfgNl
nEqm7HWkNxHhf8A6En/IjleuddS1sf9e/x+TJN1Xhnt9W6pe7Fk1
-----END RSA PRIVATE KEY-----
EOT

  certificate = <<EOT
-----BEGIN CERTIFICATE-----
MIIDpTCCAo2gAwIBAgIJAKdmmOBYnFvoMA0GCSqGSIb3DQEBCwUAMGkxCzAJBgNV
BAYTAnh4MQswCQYDVQQIDAJ4eDELMAkGA1UEBwwCeHgxCzAJBgNVBAoMAnh4MQsw
CQYDVQQLDAJ4eDELMAkGA1UEAwwCeHgxGTAXBgkqhkiG9w0BCQEWCnh4QDE2My5j
b20wHhcNMTcxMjA0MDM0MjQ5WhcNMjAxMjAzMDM0MjQ5WjBpMQswCQYDVQQGEwJ4
eDELMAkGA1UECAwCeHgxCzAJBgNVBAcMAnh4MQswCQYDVQQKDAJ4eDELMAkGA1UE
CwwCeHgxCzAJBgNVBAMMAnh4MRkwFwYJKoZIhvcNAQkBFgp4eEAxNjMuY29tMIIB
IjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAwZ5UJULAjWr7p6FVwGRQRjFN
2s8tZ/6LC3X82fajpVsYqF1xqEuUDndDXVD09E4u83MS6HO6a3bIVQDp6/klnYld
iE6Vp8HH5BSKaCWKVg8lGWg1UM9wZFnlryi14KgmpIFmcu9nA8yV/6MZAe6RSDmb
3iyNBmiZ8aZhGw2pI1YwR+15MVqFFGB+7ExkziROi7L8CFCyCezK2/oOOvQsH1dz
Q8z1JXWdg8/9Zx7Ktvgwu5PQM3cJtSHX6iBPOkMU8Z8TugLlTqQXKZOEgwajwvQ5
mf2DPkVgMO8XAgaLJcLigwD513koAdtJd5v+9irw+5LAuO3JclqwTvwy7u/YwwID
AQABo1AwTjAdBgNVHQ4EFgQUo5A2tIu+bcUfvGTD7wmEkhXKFjcwHwYDVR0jBBgw
FoAUo5A2tIu+bcUfvGTD7wmEkhXKFjcwDAYDVR0TBAUwAwEB/zANBgkqhkiG9w0B
AQsFAAOCAQEAWJ2rS6Mvlqk3GfEpboezx2J3X7l1z8Sxoqg6ntwB+rezvK3mc9H0
83qcVeUcoH+0A0lSHyFN4FvRQL6X1hEheHarYwJK4agb231vb5erasuGO463eYEG
r4SfTuOm7SyiV2xxbaBKrXJtpBp4WLL/s+LF+nklKjaOxkmxUXOsM4CTA7uFJypY
c8Tdr8lDDNqoUtMD8BrUCJi+7lmMXRcC3Qi3oZJW76ja+kZA5mKVFPd1ATih8TbA
i34R7EQDtFeiSvBdeKRsPp8c0KT8H1B4lXNkkCQs2WX5p4lm99+ZtLD4glw8x6Ic
i1YhgnQbn5E0hz55OLu5jvOkKQjPCW+8Kg==
-----END CERTIFICATE-----
EOT

  timeouts {
    create = "5m"
```

```
    update = "5m"
    delete = "5m"
  }
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. A Networking client is needed to create an LB certificate. If omitted, the `region` argument of the provider is used. Changing this creates a new LB certificate.
- `name` - (Optional) Human-readable name for the Certificate. Does not have to be unique.
- `description` - (Optional) Human-readable description for the Certificate.
- `domain` - (Optional) The domain of the Certificate.
- `private_key` - (Required) The private encrypted key of the Certificate, PEM format.
- `certificate` - (Required) The public encrypted key of the Certificate, PEM format.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `domain` - See Argument Reference above.
- `private_key` - See Argument Reference above.
- `certificate` - See Argument Reference above.
- `update_time` - Indicates the update time.
- `create_time` - Indicates the creation time.

» huaweicloudstack_lb_loadbalancer_v2

Manages a V2 loadbalancer resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_loadbalancer_v2" "lb_1" {
  vip_subnet_id = "d9415786-5f1a-428b-b35f-2f1523e146d2"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. If omitted, the `region` argument of the provider is used. Changing this creates a new loadbalancer.
- `vip_subnet_id` - (Required) The network on which to allocate the Loadbalancer's address. A tenant can only create Loadbalancers on networks authorized by policy (e.g. networks that belong to them or networks that are shared). Changing this creates a new loadbalancer.
- `name` - (Optional) Human-readable name for the Loadbalancer. Does not have to be unique.
- `description` - (Optional) Human-readable description for the Loadbalancer.
- `tenant_id` - (Optional) Required for admins. The UUID of the tenant who owns the Loadbalancer. Only administrative users can specify a tenant UUID other than their own. Changing this creates a new loadbalancer.
- `vip_address` - (Optional) The ip address of the load balancer. Changing this creates a new loadbalancer.
- `admin_state_up` - (Optional) The administrative state of the Loadbalancer. A valid value is true (UP) or false (DOWN).
- `loadbalancer_provider` - (Optional) The name of the provider. For Region Type I, only VLB is supported. For Region Type II, VLB, native load balancers, and load balancers of third-party providers are supported. Changing this creates a new loadbalancer.

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the load balancer.
- `region` - See Argument Reference above.
- `vip_subnet_id` - See Argument Reference above.
- `name` - See Argument Reference above.

- `description` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `vip_address` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `loadbalancer_provider` - See Argument Reference above.
- `vip_port_id` - The Port ID of the Load Balancer IP.

» huaweicloudstack_lb_listener_v2

Manages a V2 listener resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_listener_v2" "listener_1" {
  protocol        = "HTTP"
  protocol_port   = 8080
  loadbalancer_id = "d9415786-5f1a-428b-b35f-2f1523e146d2"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. If omitted, the `region` argument of the provider is used. Changing this creates a new Listener.
- `protocol` - (Required) The listening protocol. Converged ELB in Region Type I and Region Type II supports TCP, UDP, HTTP, and TERMINATED_HTTPS. Non-converged ELB in Region Type II supports TCP and HTTP. Changing this creates a new Listener.
- `protocol_port` - (Required) The port on which to listen for client traffic. Must be an integer in the range of 1-65535. Changing this creates a new Listener.
- `tenant_id` - (Optional) Required for admins. The UUID of the tenant who owns the Listener. Only administrative users can specify a tenant UUID other than their own. Changing this creates a new Listener.
- `loadbalancer_id` - (Required) The load balancer on which to provision this Listener. Changing this creates a new Listener.
- `name` - (Optional) Human-readable name for the Listener. Does not have to be unique.

- `default_pool_id` - (Optional) The ID of the default pool with which the Listener is associated. Changing this creates a new Listener.
- `description` - (Optional) Human-readable description for the Listener.
- `connection_limit` - (Optional) The maximum number of connections allowed for the Listener. A valid value is from -1 to 2147483647. The default value for this attribute will be -1, indicating an infinite limit.
- `default_tls_container_ref` - (Optional) A reference to a Barbican Secrets container which stores TLS information. This is required if the protocol is `TERMINATED_HTTPS`. For converged ELB in Region Type I and Region Type II, enter a certificate ID.
- `admin_state_up` - (Optional) The administrative state of the Listener. A valid value is true (UP) or false (DOWN).

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the Listener.
- `protocol` - See Argument Reference above.
- `protocol_port` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `name` - See Argument Reference above.
- `default_port_id` - See Argument Reference above.
- `description` - See Argument Reference above.
- `connection_limit` - See Argument Reference above.
- `default_tls_container_ref` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.

» huaweicloudstack_lb_pool_v2

Manages a V2 pool resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_pool_v2" "pool_1" {
  protocol    = "HTTP"
  lb_method   = "ROUND_ROBIN"
  listener_id = "d9415786-5f1a-428b-b35f-2f1523e146d2"

  persistence {
    type        = "APP_COOKIE"
```

```
    cookie_name = "testCookie"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) Human-readable name for the pool.
- `description` - (Optional) Human-readable description for the pool.
- `protocol` - (Required) The IP protocol, can either be TCP, HTTP or UDP. Changing this creates a new pool.
- `loadbalancer_id` - (Optional) The load balancer on which to provision this pool. Changing this creates a new pool. Note: One of LoadbalancerID or ListenerID must be provided.
- `listener_id` - (Optional) The Listener on which the members of the pool will be associated with. Changing this creates a new pool. Note: One of LoadbalancerID or ListenerID must be provided.
- `lb_method` - (Required) The load balancing algorithm to distribute traffic to the pool's members. Must be one of ROUND_ROBIN, LEAST_CONNECTIONS, or SOURCE_IP.
- `persistence` - Omit this field to prevent session persistence. Indicates whether connections in the same session will be processed by the same Pool member or not. Changing this creates a new pool.
- `admin_state_up` - (Optional) The administrative state of the pool. A valid value is true (UP) or false (DOWN).

The `persistence` argument supports:

- `type` - (Required) The type of persistence mode. The current specification supports SOURCE_IP, HTTP_COOKIE, and APP_COOKIE.
- `cookie_name` - (Optional) The name of the cookie if persistence mode is set appropriately. It's only supported in the `APP_COOKIE` type.

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the pool.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `protocol` - See Argument Reference above.

- `lb_method` - See Argument Reference above.
- `persistence` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.

» huaweicloudstack_lb_member_v2

Manages a V2 member resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_member_v2" "member_1" {
  address       = "192.168.199.23"
  protocol_port = 8080
  pool_id       = "${var.pool_id}"
  subnet_id     = "${var.subnet_id}"
}
```

» Argument Reference

The following arguments are supported:

- `pool_id` - (Required) The id of the pool that this member will be assigned to.
- `subnet_id` - (Required) The subnet in which to access the member
- `name` - (Optional) Human-readable name for the member.
- `address` - (Required) The IP address of the member to receive traffic from the load balancer. Changing this creates a new member.
- `protocol_port` - (Required) The port on which to listen for client traffic. Changing this creates a new member.
- `weight` - (Optional) A positive integer value that indicates the relative portion of traffic that this member should receive from the pool. For example, a member with a weight of 10 receives five times as much traffic as a member with a weight of 2.
- `admin_state_up` - (Optional) The administrative state of the member. A valid value is true (UP) or false (DOWN).

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the member.
- `name` - See Argument Reference above.
- `weight` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `subnet_id` - See Argument Reference above.
- `pool_id` - See Argument Reference above.
- `address` - See Argument Reference above.
- `protocol_port` - See Argument Reference above.

» huaweicloudstack_lb_monitor_v2

Manages a V2 monitor resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_monitor_v2" "monitor_1" {
  pool_id     = "${huaweicloudstack_lb_pool_v2.pool_1.id}"
  type        = "HTTP"
  delay       = 20
  timeout     = 10
  max_retries = 5
  url_path    = "/"
}
```

» Argument Reference

The following arguments are supported:

- `pool_id` - (Required) The id of the pool that this monitor will be assigned to.
- `name` - (Optional) The Name of the Monitor.
- `type` - (Required) The type of protocol. Converged ELB in Region Type I and Region Type II supports TCP, UDP_CONNECT, or HTTP. Non-converged ELB in Region Type II supports TCP, PING, or HTTP. For Region Type I, if protocol of the listener is set to UDP, type of the health check must be set to UDP_CONNECT. Changing this creates a new monitor.
- `delay` - (Required) The interval in seconds between health check. A valid value is from 1 to 50.

- `timeout` - (Required) Maximum number of seconds for a monitor to wait for a ping reply before it times out. The value must be less than the delay value.
- `max_retries` - (Required) Number of permissible ping failures before changing the member's status to INACTIVE. Must be a number between 1 and 10.
- `url_path` - (Optional) Required for HTTP types. URI path that will be accessed if monitor type is HTTP.
- `http_method` - (Optional) Required for HTTP types. The HTTP method used for requests by the monitor. If this attribute is not specified, it defaults to "GET".
- `expected_codes` - (Optional) Required for HTTP types. Expected HTTP codes for a passing HTTP(S) monitor. You can either specify a single status like "200", or a range like "200-202".
- `admin_state_up` - (Optional) The administrative state of the monitor. A valid value is true (UP) or false (DOWN).

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the monitor.
- `type` - See Argument Reference above.
- `delay` - See Argument Reference above.
- `timeout` - See Argument Reference above.
- `max_retries` - See Argument Reference above.
- `url_path` - See Argument Reference above.
- `http_method` - See Argument Reference above.
- `expected_codes` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.

» huaweicloudstack_lb_l7policy_v2

Manages a Load Balancer L7 Policy resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_loadbalancer_v2" "loadbalancer_1" {
  name          = "loadbalancer_1"
  vip_subnet_id = "SUBNET_ID"
```

```
}

resource "huaweicloudstack_lb_listener_v2" "listener_1" {
  name            = "listener_1"
  protocol        = "HTTP"
  protocol_port   = 8080
  loadbalancer_id = "${huaweicloudstack_lb_loadbalancer_v2.loadbalancer_1.id}"
}

resource "huaweicloudstack_lb_pool_v2" "pool_1" {
  name            = "pool_1"
  protocol        = "HTTP"
  lb_method       = "ROUND_ROBIN"
  loadbalancer_id = "${huaweicloudstack_lb_loadbalancer_v2.loadbalancer_1.id}"
}

resource "huaweicloudstack_lb_l7policy_v2" "l7policy_1" {
  name             = "test"
  description      = "test l7 policy"
  position         = 1
  listener_id      = "${huaweicloudstack_lb_listener_v2.listener_1.id}"
  redirect_pool_id = "${huaweicloudstack_lb_pool_v2.pool_1.id}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. If omitted, the `region` argument of the provider is used. Changing this creates a new L7 Policy.
- `tenant_id` - (Optional) Required for admins. The UUID of the tenant who owns the L7 Policy. Only administrative users can specify a tenant UUID other than their own. Changing this creates a new L7 Policy.
- `name` - (Optional) Human-readable name for the L7 Policy. Does not have to be unique.
- `description` - (Optional) Human-readable description for the L7 Policy.
- `action` - (Optional) The L7 Policy action. The value can only be REDIRECT_TO_POOL.
- `listener_id` - (Required) The Listener on which the L7 Policy will be associated with. Changing this creates a new L7 Policy.

- `position` - (Optional) The position of this policy on the listener. Positions start at 1. Changing this creates a new L7 Policy.
- `redirect_pool_id` - (Required) Requests matching this policy will be redirected to the pool with this ID.
- `admin_state_up` - (Optional) The administrative state of the L7 Policy. This value can only be true (UP).

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the L7 Policy.
- `region` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `action` - See Argument Reference above.
- `listener_id` - See Argument Reference above.
- `position` - See Argument Reference above.
- `redirect_pool_id` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.

» Import

Load Balancer L7 Policy can be imported using the L7 Policy ID, e.g.:

```
$ terraform import huaweicloudstack_lb_l7policy_v2.l7policy_1 8a7a79c2-cf17-4e65-b2ae-ddc8bf
```

» huaweicloudstack_lb_l7rule_v2

Manages a V2 L7 Rule resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_loadbalancer_v2" "loadbalancer_1" {
  name          = "loadbalancer_1"
  vip_subnet_id = "SUBNET_ID"
}

resource "huaweicloudstack_lb_listener_v2" "listener_1" {
  name            = "listener_1"
```

```
  protocol        = "HTTP"
  protocol_port   = 8080
  loadbalancer_id = "${huaweicloudstack_lb_loadbalancer_v2.loadbalancer_1.id}"
}

resource "huaweicloudstack_lb_pool_v2" "pool_1" {
  name            = "pool_1"
  protocol        = "HTTP"
  lb_method       = "ROUND_ROBIN"
  loadbalancer_id = "${huaweicloudstack_lb_loadbalancer_v2.loadbalancer_1.id}"
}

resource "huaweicloudstack_lb_l7policy_v2" "l7policy_1" {
  name         = "test"
  action       = "REDIRECT_TO_URL"
  description  = "test description"
  position     = 1
  listener_id  = "${huaweicloudstack_lb_listener_v2.listener_1.id}"
  redirect_url = "http://www.example.com"
}

resource "huaweicloudstack_lb_l7rule_v2" "l7rule_1" {
  l7policy_id  = "${huaweicloudstack_lb_l7policy_v2.l7policy_1.id}"
  type         = "PATH"
  compare_type = "EQUAL_TO"
  value        = "/api"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. If omitted, the `region` argument of the provider is used. Changing this creates a new L7 Rule.
- `tenant_id` - (Optional) Required for admins. The UUID of the tenant who owns the L7 Rule. Only administrative users can specify a tenant UUID other than their own. Changing this creates a new L7 Rule.
- `description` - (Optional) Human-readable description for the L7 Rule.
- `type` - (Required) The L7 Rule type - can either be HOST_NAME or PATH. Changing this creates a new L7 Rule.
- `compare_type` - (Required) The comparison type for the L7 rule - can either be STARTS_WITH, EQUAL_TO or REGEX

- `l7policy_id` - (Required) The ID of the L7 Policy to query. Changing this creates a new L7 Rule.
- `value` - (Required) The value to use for the comparison. For example, the file type to compare.
- `key` - (Optional) The key to use for the comparison. For example, the name of the cookie to evaluate. Valid when `type` is set to COOKIE or HEADER. Changing this creates a new L7 Rule.
- `admin_state_up` - (Optional) The administrative state of the L7 Rule. The value can only be true (UP).

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the L7 Rule.
- `region` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `type` - See Argument Reference above.
- `compare_type` - See Argument Reference above.
- `l7policy_id` - See Argument Reference above.
- `value` - See Argument Reference above.
- `key` - See Argument Reference above.
- `invert` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `listener_id` - The ID of the Listener owning this resource.

» Import

Load Balancer L7 Rule can be imported using the L7 Policy ID and L7 Rule ID separated by a slash, e.g.:

```
$ terraform import huaweicloudstack_lb_l7rule_v2.l7rule_1 e0bd694a-abbe-450e-b329-0931fd1cc5
```

» huaweicloudstack_lb_whitelist_v2

Manages an ELB whitelist resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_lb_whitelist_v2" "whitelist_1" {
  enable_whitelist = true
```

```
  whitelist        = "192.168.11.1,192.168.0.1/24,192.168.201.18/8"
  listener_id      = "d9415786-5f1a-428b-b35f-2f1523e146d2"
}
```

» Argument Reference

The following arguments are supported:

- `tenant_id` - (Optional) Required for admins. The UUID of the tenant who owns the whitelist. Only administrative users can specify a tenant UUID other than their own. Changing this creates a new whitelist.
- `listener_id` - (Required) The Listener ID that the whitelist will be associated with. Changing this creates a new whitelist.
- `enable_whitelist` - (Optional) Specify whether to enable access control.
- `whitelist` - (Optional) Specifies the IP addresses in the whitelist. Use commas(,) to separate the multiple IP addresses.

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the whitelist.
- `tenant_id` - See Argument Reference above.
- `listener_id` - See Argument Reference above.
- `enable_whitelist` - See Argument Reference above.
- `whitelist` - See Argument Reference above.

» huaweicloudstack_networking_floatingip_v2

Manages a V2 floating IP resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_networking_floatingip_v2" "floatip_1" {
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. A Networking client is needed to create a floating IP that can be used with another networking resource, such as a load balancer. If omitted, the `region` argument of the provider is used. Changing this creates a new floating IP (which may or may not have a different address).
- `pool` - (Optional) The name of the pool from which to obtain the floating IP. Only admin_external_net is valid. Changing this creates a new floating IP.
- `port_id` - (Optional) ID of an existing port with at least one IP address to associate with this floating IP.
- `tenant_id` - (Optional) The target tenant ID in which to allocate the floating IP, if you specify this together with a port_id, make sure the target port belongs to the same tenant. Changing this creates a new floating IP (which may or may not have a different address)
- `fixed_ip` - Fixed IP of the port to associate with this floating IP. Required if the port has multiple fixed IPs.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `pool` - See Argument Reference above.
- `address` - The actual floating IP address itself.
- `port_id` - ID of associated port.
- `tenant_id` - the ID of the tenant in which to create the floating IP.
- `fixed_ip` - The fixed IP which the floating IP maps to.

» Import

Floating IPs can be imported using the `id`, e.g.

```
$ terraform import huaweicloudstack_networking_floatingip_v2.floatip_1 2c7f39f3-702b-48d1-94
```

» huaweicloudstack_networking_floatingip_associate_v2

Associates a floating IP to a port. This is useful for situations where you have a pre-allocated floating IP or are unable to use the `huaweicloudstack_networking_floatingip_v2` resource to create a floating IP.

» Example Usage

```
resource "huaweicloudstack_networking_port_v2" "port_1" {
  network_id = "a5bbd213-e1d3-49b6-aed1-9df60ea94b9a"
}

resource "huaweicloudstack_networking_floatingip_associate_v2" "fip_1" {
  floating_ip = "1.2.3.4"
  port_id     = "${huaweicloudstack_networking_port_v2.port_1.id}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. A Networking client is needed to create a floating IP that can be used with another networking resource, such as a load balancer. If omitted, the `region` argument of the provider is used. Changing this creates a new floating IP (which may or may not have a different address).
- `floating_ip` - (Required) IP Address of an existing floating IP.
- `port_id` - (Required) ID of an existing port with at least one IP address to associate with this floating IP.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `floating_ip` - See Argument Reference above.
- `port_id` - See Argument Reference above.

» Import

Floating IP associations can be imported using the `id` of the floating IP, e.g.

```
$ terraform import huaweicloudstack_networking_floatingip_associate_v2.fip 2c7f39f3-702b-48c
```

» huaweicloudstack_networking_network_v2

Manages a V2 Neutron network resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_subnet_v2" "subnet_1" {
  name       = "subnet_1"
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "huaweicloudstack_networking_secgroup_v2" "secgroup_1" {
  name = "secgroup_1"
  description = "a security group"
}

resource "huaweicloudstack_networking_secgroup_rule_v2" "secgroup_rule_1" {
  direction = "ingress"
  ethertype = "IPv4"
  port_range_max = 22
  port_range_min = 22
  protocol = "tcp"
  remote_ip_prefix = "0.0.0.0/0"
  security_group_id = "${huaweicloudstack_networking_secgroup_v2.secgroup_1.id}"
}

resource "huaweicloudstack_networking_port_v2" "port_1" {
  name               = "port_1"
  network_id         = "${huaweicloudstack_networking_network_v2.network_1.id}"
  admin_state_up     = "true"
  security_group_ids = ["${huaweicloudstack_networking_secgroup_v2.secgroup_1.id}"]

  fixed_ip {
    "subnet_id"  = "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
    "ip_address" = "192.168.199.10"
  }
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["${huaweicloudstack_networking_secgroup_v2.secgroup_1.name}"]
```

```
  network {
    port = "${huaweicloudstack_networking_port_v2.port_1.id}"
  }
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. A Networking client is needed to create a Neutron network. If omitted, the `region` argument of the provider is used. Changing this creates a new network.
- `name` - (Optional) The name of the network. Changing this updates the name of the existing network.
- `shared` - (Optional) Specifies whether the network resource can be accessed by any tenant or not. Changing this updates the sharing capabalities of the existing network.
- `tenant_id` - (Optional) The owner of the network. Required if admin wants to create a network for another tenant. Changing this creates a new network.
- `admin_state_up` - (Optional) The administrative state of the network. The value must be "true".
- `segments` - (Optional) An array of one or more provider segment objects.

The `segments` block supports:

- `physical_network` - The phisical network where this network is implemented.
- `segmentation_id` - An isolated segment on the physical network.
- `network_type` - The type of physical network.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `shared` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloudstack_networking_network_v2.network_1 d90ce693-5ccf-4136-a0ed-
```

» huaweicloudstack_networking_port_v2

Manages a V2 port resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_port_v2" "port_1" {
  name           = "port_1"
  network_id     = "${huaweicloudstack_networking_network_v2.network_1.id}"
  admin_state_up = "true"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a port. If omitted, the `region` argument of the provider is used. Changing this creates a new port.
- `name` - (Optional) A unique name for the port. Changing this updates the `name` of an existing port.
- `network_id` - (Required) The ID of the network to attach the port to. Changing this creates a new port.
- `admin_state_up` - (Optional) Administrative up/down status for the port (must be "true" or "false" if provided). Changing this updates the `admin_state_up` of an existing port.
- `mac_address` - (Optional) Specify a specific MAC address for the port. Changing this creates a new port.

- `tenant_id` - (Optional) The owner of the Port. Required if admin wants to create a port for another tenant. Changing this creates a new port.
- `device_owner` - (Optional) The device owner of the Port. Changing this creates a new port.
- `security_group_ids` - (Optional - Conflicts with `no_security_groups`) A list of security group IDs to apply to the port. The security groups must be specified by ID and not name (as opposed to how they are configured with the Compute Instance).
- `no_security_groups` - (Optional - Conflicts with `security_group_ids`) If set to `true`, then no security groups are applied to the port. If set to `false` and no `security_group_ids` are specified, then the Port will yield to the default behavior of the Networking service, which is to usually apply the "default" security group.
- `device_id` - (Optional) The ID of the device attached to the port. Changing this creates a new port.
- `fixed_ip` - (Optional) An array of desired IPs for this port. The structure is described below.
- `allowed_address_pairs` - (Optional) An IP/MAC Address pair of additional IP addresses that can be active on this port. The structure is described below.

The `fixed_ip` block supports:

- `subnet_id` - (Required) Subnet in which to allocate IP address for this port.
- `ip_address` - (Optional) IP address desired in the subnet for this port. If you don't specify `ip_address`, an available IP address from the specified subnet will be allocated to this port. This field will not be populated if it is left blank. To retrieve the assigned IP address, use the `all_fixed_ips` attribute.

The `allowed_address_pairs` block supports:

- `ip_address` - (Required) The additional IP address.
- `mac_address` - (Optional) The additional MAC address.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `mac_address` - See Argument Reference above.

- `tenant_id` - See Argument Reference above.
- `device_owner` - See Argument Reference above.
- `security_group_ids` - See Argument Reference above.
- `device_id` - See Argument Reference above.
- `fixed_ip` - See Argument Reference above.
- `all_fixed_ips` - The collection of Fixed IP addresses on the port in the order returned by the Network v2 API.
- `all_security_group_ids` - The collection of Security Group IDs on the port which have been explicitly and implicitly added.

» Import

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

```
$ terraform import huaweicloudstack_networking_port_v2.port_1 eae26a3e-1c33-4cc1-9c31-0cd729
```

» Notes

» Ports and Instances

There are some notes to consider when connecting Instances to networks using Ports. Please see the `huaweicloudstack_compute_instance_v2` documentation for further documentation.

» huaweicloudstack_networking_router_interface_v2

Manages a V2 router interface resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "tf_test_network"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_subnet_v2" "subnet_1" {
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "huaweicloudstack_networking_router_v2" "router_1" {
```

```
  name                = "my_router"
  external_network_id = "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"
}

resource "huaweicloudstack_networking_router_interface_v2" "router_interface_1" {
  router_id = "${huaweicloudstack_networking_router_v2.router_1.id}"
  subnet_id = "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a router. If omitted, the `region` argument of the provider is used. Changing this creates a new router interface.
- `router_id` - (Required) ID of the router this interface belongs to. Changing this creates a new router interface.
- `subnet_id` - ID of the subnet this interface connects to. Changing this creates a new router interface.
- `port_id` - ID of the port this interface connects to. Changing this creates a new router interface.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `router_id` - See Argument Reference above.
- `subnet_id` - See Argument Reference above.
- `port_id` - See Argument Reference above.

» Import

Router Interfaces can be imported using the port `id`, e.g.

```
$ openstack port list --router <router name or id>
$ terraform import huaweicloudstack_networking_router_interface_v2.int_1 <port id from above
```

» huaweicloudstack_networking_router_route_v2

Creates a routing entry on a HuaweiCloudStack V2 router.

» Example Usage

```
resource "huaweicloudstack_networking_router_v2" "router_1" {
  name           = "router_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_subnet_v2" "subnet_1" {
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "huaweicloudstack_networking_router_interface_v2" "int_1" {
  router_id = "${huaweicloudstack_networking_router_v2.router_1.id}"
  subnet_id = "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
}

resource "huaweicloudstack_networking_router_route_v2" "router_route_1" {
  depends_on       = ["huaweicloudstack_networking_router_interface_v2.int_1"]
  router_id        = "${huaweicloudstack_networking_router_v2.router_1.id}"
  destination_cidr = "10.0.1.0/24"
  next_hop         = "192.168.199.254"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to configure a routing entry on a router. If omitted, the `region` argument of the provider is used. Changing this creates a new routing entry.
- `router_id` - (Required) ID of the router this routing entry belongs to. Changing this creates a new routing entry.

- `destination_cidr` - (Required) CIDR block to match on the packet's destination IP. Changing this creates a new routing entry.
- `next_hop` - (Required) IP address of the next hop gateway. Changing this creates a new routing entry.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `router_id` - See Argument Reference above.
- `destination_cidr` - See Argument Reference above.
- `next_hop` - See Argument Reference above.

» Notes

The `next_hop` IP address must be directly reachable from the router at the `huaweicloudstack_networking_router_route_v2` resource creation time. You can ensure that by explicitly specifying a dependency on the `huaweicloudstack_networking_router_interface_v2` resource that connects the next hop to the router, as in the example above.

» Import

Routing entries can be imported using a combined ID using the following format:
`<router_id>-route-<destination_cidr>-<next_hop>`

```
$ terraform import huaweicloudstack_networking_router_route_v2.router_route_1 686fe248-386c-
```

» huaweicloudstack_networking_router_v2

Manages a V2 router resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_networking_router_v2" "router_1" {
  name                = "my_router"
  admin_state_up      = true
  external_network_id = "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a router. If omitted, the `region` argument of the provider is used. Changing this creates a new router.
- `name` - (Optional) A unique name for the router. Changing this updates the `name` of an existing router.
- `admin_state_up` - (Optional) Administrative up/down status for the router (must be "true" or "false" if provided). Changing this updates the `admin_state_up` of an existing router.
- `distributed` - (Optional) Indicates whether or not to create a distributed router. The default policy setting in Neutron restricts usage of this property to administrative users only.
- `external_network_id` - (Optional) The network UUID of an external gateway for the router. A router with an external gateway is required if any compute instances or load balancers will be using floating IPs. Changing this updates the external gateway of the router.
- `enable_snat` - (Optional) Enable Source NAT for the router. Valid values are "true" or "false". An `external_network_id` has to be set in order to set this property. Changing this updates the `enable_snat` of the router.
- `tenant_id` - (Optional) The owner of the floating IP. Required if admin wants to create a router for another tenant. Changing this creates a new router.

» Attributes Reference

The following attributes are exported:

- `id` - ID of the router.
- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `external_network_id` - See Argument Reference above.
- `enable_snat` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloudstack_networking_router_v2.router_1 014395cd-89fc-4c9b-96b7-13
```

» huaweicloudstack_networking_subnet_v2

Manages a V2 Neutron subnet resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name           = "tf_test_network"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_subnet_v2" "subnet_1" {
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 Networking client. A Networking client is needed to create a Neutron subnet. If omitted, the `region` argument of the provider is used. Changing this creates a new subnet.
- `network_id` - (Required) The UUID of the parent network. Changing this creates a new subnet.
- `cidr` - (Required) CIDR representing IP range for this subnet, based on IP version. Changing this creates a new subnet.
- `ip_version` - (Optional) IP version, either 4 (default) or 6. Changing this creates a new subnet.
- `name` - (Optional) The name of the subnet. Changing this updates the name of the existing subnet.
- `tenant_id` - (Optional) The owner of the subnet. Required if admin wants to create a subnet for another tenant. Changing this creates a new subnet.
- `allocation_pools` - (Optional) An array of sub-ranges of CIDR available for dynamic allocation to ports. The allocation_pool object structure is documented below. Changing this creates a new subnet.

- `gateway_ip` - (Optional) Default gateway used by devices in this subnet. Leaving this blank and not setting `no_gateway` will cause a default gateway of `.1` to be used. Changing this updates the gateway IP of the existing subnet.
- `no_gateway` - (Optional) Do not set a gateway IP on this subnet. Changing this removes or adds a default gateway IP of the existing subnet.
- `enable_dhcp` - (Optional) The administrative state of the network. The value must be "true".
- `dns_nameservers` - (Optional) An array of DNS name server names used by hosts in this subnet. Changing this updates the DNS name servers for the existing subnet.
- `host_routes` - (Optional) An array of routes that should be used by devices with IPs from this subnet (not including local subnet route). The host_route object structure is documented below. Changing this updates the host routes for the existing subnet.

The `allocation_pools` block supports:

- `start` - (Required) The starting address.
- `end` - (Required) The ending address.

The `host_routes` block supports:

- `destination_cidr` - (Required) The destination CIDR.
- `next_hop` - (Required) The next hop in the route.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `network_id` - See Argument Reference above.
- `cidr` - See Argument Reference above.
- `ip_version` - See Argument Reference above.
- `name` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `allocation_pools` - See Argument Reference above.
- `gateway_ip` - See Argument Reference above.
- `enable_dhcp` - See Argument Reference above.
- `dns_nameservers` - See Argument Reference above.
- `host_routes` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloudstack_networking_subnet_v2.subnet_1 da4faf16-5546-41e4-8330-4c
```

» huaweicloudstack_networking_secgroup_v2

Manages a V2 security group resource within HuaweiCloudStack. Unlike Nova security groups, neutron separates the group from the rules and also allows an admin to target a specific tenant_id.

» Example Usage

```
resource "huaweicloudstack_networking_secgroup_v2" "secgroup_1" {
  name        = "secgroup_1"
  description = "My neutron security group"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a port. If omitted, the `region` argument of the provider is used. Changing this creates a new security group.
- `name` - (Required) A unique name for the security group.
- `description` - (Optional) A unique name for the security group.
- `tenant_id` - (Optional) The owner of the security group. Required if admin wants to create a port for another tenant. Changing this creates a new security group.
- `delete_default_rules` - (Optional) Whether or not to delete the default egress security rules. This is `false` by default. See the below note for more information.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.

- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.

» Default Security Group Rules

In most cases, HuaweiCloudStack will create some egress security group rules for each new security group. These security group rules will not be managed by Terraform, so if you prefer to have *all* aspects of your infrastructure managed by Terraform, set `delete_default_rules` to `true` and then create separate security group rules such as the following:

```
resource "huaweicloudstack_networking_secgroup_rule_v2" "secgroup_rule_v4" {
  direction         = "egress"
  ethertype         = "IPv4"
  security_group_id = "${huaweicloudstack_networking_secgroup_v2.secgroup.id}"
}

resource "huaweicloudstack_networking_secgroup_rule_v2" "secgroup_rule_v6" {
  direction         = "egress"
  ethertype         = "IPv6"
  security_group_id = "${huaweicloudstack_networking_secgroup_v2.secgroup.id}"
}
```

Please note that this behavior may differ depending on the configuration of the HuaweiCloudStack. The above illustrates the current default Neutron behavior. Some HuaweiCloudStack might provide additional rules and some might not provide any rules at all (in which case the `delete_default_rules` setting is moot).

» Import

Security Groups can be imported using the `id`, e.g.

```
$ terraform import huaweicloudstack_networking_secgroup_v2.secgroup_1 38809219-5e8a-4852-913
```

» huaweicloudstack_networking_secgroup_rule_v2

Manages a V2 neutron security group rule resource within HuaweiCloudStack. Unlike Nova security groups, neutron separates the group from the rules and also allows an admin to target a specific tenant_id.

» Example Usage

```
resource "huaweicloudstack_networking_secgroup_v2" "secgroup_1" {
  name        = "secgroup_1"
  description = "My neutron security group"
}

resource "huaweicloudstack_networking_secgroup_rule_v2" "secgroup_rule_1" {
  direction         = "ingress"
  ethertype         = "IPv4"
  protocol          = "tcp"
  port_range_min    = 22
  port_range_max    = 22
  remote_ip_prefix  = "0.0.0.0/0"
  security_group_id = "${huaweicloudstack_networking_secgroup_v2.secgroup_1.id}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a port. If omitted, the `region` argument of the provider is used. Changing this creates a new security group rule.
- `direction` - (Required) The direction of the rule, valid values are **ingress** or **egress**. Changing this creates a new security group rule.
- `ethertype` - (Required) The layer 3 protocol type, valid values are **IPv4** or **IPv6**. Changing this creates a new security group rule.
- `protocol` - (Optional) The layer 4 protocol type, valid values are following. Changing this creates a new security group rule. This is required if you want to specify a port range.
  - **tcp**
  - **udp**
  - **icmp**
  - **ah**
  - **dccp**
  - **egp**
  - **esp**
  - **gre**
  - **igmp**
  - **ipv6-encap**
  - **ipv6-frag**

  - **ipv6-icmp**
  - **ipv6-nonxt**
  - **ipv6-opts**
  - **ipv6-route**
  - **ospf**
  - **pgm**
  - **rsvp**
  - **sctp**
  - **udplite**
  - **vrrp**

- `port_range_min` - (Optional) The lower part of the allowed port range, valid integer value needs to be between 1 and 65535. Changing this creates a new security group rule.
- `port_range_max` - (Optional) The higher part of the allowed port range, valid integer value needs to be between 1 and 65535. Changing this creates a new security group rule.
- `remote_ip_prefix` - (Optional) The remote CIDR, the value needs to be a valid CIDR (i.e. 192.168.0.0/16). Changing this creates a new security group rule.
- `remote_group_id` - (Optional) The remote group id, the value needs to be an Openstack ID of a security group in the same tenant. Changing this creates a new security group rule.
- `security_group_id` - (Required) The security group id the rule should belong to, the value needs to be an Openstack ID of a security group in the same tenant. Changing this creates a new security group rule.
- `tenant_id` - (Optional) The owner of the security group. Required if admin wants to create a port for another tenant. Changing this creates a new security group rule.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `direction` - See Argument Reference above.
- `ethertype` - See Argument Reference above.
- `protocol` - See Argument Reference above.
- `port_range_min` - See Argument Reference above.
- `port_range_max` - See Argument Reference above.
- `remote_ip_prefix` - See Argument Reference above.
- `remote_group_id` - See Argument Reference above.
- `security_group_id` - See Argument Reference above.

- `tenant_id` - See Argument Reference above.

» Import

Security Group Rules can be imported using the `id`, e.g.

```
$ terraform import huaweicloudstack_networking_secgroup_rule_v2.secgroup_rule_1 aeb68ee3-6e9
```

» huaweicloudstack_networking_vip_v2

Manages a V2 vip resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_subnet_v2" "subnet_1" {
  name = "subnet_1"
  cidr = "192.168.199.0/24"
  ip_version = 4
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"
}

resource "huaweicloudstack_networking_router_interface_v2" "router_interface_1" {
  router_id = "${huaweicloudstack_networking_router_v2.router_1.id}"
  subnet_id = "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
}

resource "huaweicloudstack_networking_router_v2" "router_1" {
  name = "router_1"
  external_gateway = "0a2228f2-7f8a-45f1-8e09-9039e1d09975"
}

resource "huaweicloudstack_networking_vip_v2" "vip_1" {
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"
  subnet_id = "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
}
```

» Argument Reference

The following arguments are supported:

- `network_id` - (Required) The ID of the network to attach the vip to. Changing this creates a new vip.
- `subnet_id` - (Required) Subnet in which to allocate IP address for this vip. Changing this creates a new vip.
- `ip_address` - (Optional) IP address desired in the subnet for this vip. If you don't specify `ip_address`, an available IP address from the specified subnet will be allocated to this vip.
- `name` - (Optional) A unique name for the vip.

» Attributes Reference

The following attributes are exported:

- `network_id` - See Argument Reference above.
- `subnet_id` - See Argument Reference above.
- `ip_address` - See Argument Reference above.
- `name` - See Argument Reference above.
- `status` - The status of vip.
- `id` - The ID of the vip.
- `tenant_id` - The tenant ID of the vip.
- `device_owner` - The device owner of the vip.

» huaweicloudstack_networking_vip_associate_v2

Manages a V2 vip associate resource within HuaweiCloudStack.

» Example Usage

```
resource "huaweicloudstack_networking_network_v2" "network_1" {
  name = "network_1"
  admin_state_up = "true"
}

resource "huaweicloudstack_networking_subnet_v2" "subnet_1" {
  name = "subnet_1"
  cidr = "192.168.199.0/24"
  ip_version = 4
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"
```

```
}

resource "huaweicloudstack_networking_router_interface_v2" "router_interface_1" {
  router_id = "${huaweicloudstack_networking_router_v2.router_1.id}"
  subnet_id = "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
}

resource "huaweicloudstack_networking_router_v2" "router_1" {
  name = "router_1"
  external_gateway = "0a2228f2-7f8a-45f1-8e09-9039e1d09975"
}

resource "huaweicloudstack_networking_port_v2" "port_1" {
  name = "port_1"
  admin_state_up = "true"
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"

  fixed_ip {
    subnet_id =  "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
  }
}

resource "huaweicloudstack_compute_instance_v2" "instance_1" {
  name = "instance_1"
  security_groups = ["default"]

  network {
    port = "${huaweicloudstack_networking_port_v2.port_1.id}"
  }
}

resource "huaweicloudstack_networking_port_v2" "port_2" {
  name = "port_2"
  admin_state_up = "true"
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"

  fixed_ip {
    subnet_id =  "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
  }
}

resource "huaweicloudstack_compute_instance_v2" "instance_2" {
  name = "instance_2"
  security_groups = ["default"]

  network {
```

```
    port = "${huaweicloudstack_networking_port_v2.port_2.id}"
  }
}

resource "huaweicloudstack_networking_vip_v2" "vip_1" {
  network_id = "${huaweicloudstack_networking_network_v2.network_1.id}"
  subnet_id = "${huaweicloudstack_networking_subnet_v2.subnet_1.id}"
}

resource "huaweicloudstack_networking_vip_associate_v2" "vip_associate_1" {
  vip_id = "${huaweicloudstack_networking_vip_v2.vip_1.id}"
  port_ids = ["${huaweicloudstack_networking_port_v2.port_1.id}", "${huaweicloudstack_networ
}
```

» Argument Reference

The following arguments are supported:

- `vip_id` - (Required) The ID of vip to attach the port to. Changing this creates a new vip associate.
- `port_ids` - (Required) An array of one or more IDs of the ports to attach the vip to. Changing this creates a new vip associate.

» Attributes Reference

The following attributes are exported:

- `vip_id` - See Argument Reference above.
- `port_ids` - See Argument Reference above.
- `vip_subnet_id` - The ID of the subnet this vip connects to.
- `vip_ip_address` - The IP address in the subnet for this vip.