» huaweicloud_networking_network_v2

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

» Example Usage

```
data "huaweicloud_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.
- `availability_zone_hints` - (Optional) The availability zone candidates for 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.
- `availability_zone_hints` - (Optional) The availability zone candidates for the network.

» huaweicloud_networking_secgroup_v2

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

» Example Usage

```
data "huaweicloud_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.

» huaweicloud_networking_subnet_v2

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

» Example Usage

```
data "huaweicloud_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.

» huaweicloud_kms_key_v1

Use this data source to get the ID of an available HuaweiCloud KMS key.

» Example Usage

```
data "huaweicloud_kms_key_v1" "key_1" {
  key_alias        = "test_key"
  key_description  = "test key description"
  key_state        = "2"
  key_id           = "af650527-a0ff-4527-aef3-c493df1f3012"
  realm            = "cn-north-1"
```

```
  default_key_flag = "0"
  domain_id        = "b168fe00ff56492495a7d22974df2d0b"
}
```

» Argument Reference

- `key_alias` - (Optional) The alias in which to create the key. It is required when we create a new key. Changing this gets the new key.
- `key_description` - (Optional) The description of the key as viewed in Huawei console. Changing this gets a new key.
- `realm` - (Optional) Region where a key resides. Changing this gets a new key.
- `key_id` - (Optional) The globally unique identifier for the key. Changing this gets the new key.
- `default_key_flag` - (Optional) Identification of a Master Key. The value "1" indicates a Default Master Key, and the value "0" indicates a key. Changing this gets a new key.
- `key_state` - (Optional) The state of a key. "1" indicates that the key is waiting to be activated. "2" indicates that the key is enabled. "3" indicates that the key is disabled. "4" indicates that the key is scheduled for deletion. Changing this gets a new key.
- `domain_id` - (Optional) - ID of a user domain for the key. Changing this gets a new key.

» Attributes Reference

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

- `key_alias` - See Argument Reference above.
- `key_description` - See Argument Reference above.
- `realm` - See Argument Reference above.
- `key_id` - See Argument Reference above.
- `default_key_flag` - See Argument Reference above.
- `scheduled_deletion_date` - Scheduled deletion time (time stamp) of a key.
- `domain_id` - See Argument Reference above.
- `expiration_time` - Expiration time.
- `creation_date` - Creation time (time stamp) of a key.
- `key_state` - See Argument Reference above.

» huaweicloud_kms_data_key_v1

Use this data source to get the plaintext and the ciphertext of an available HuaweiCloud KMS DEK (data encryption key).

» Example Usage

```
resource "huaweicloud_kms_key_v1" "key1" {
  key_alias       = "key_1"
  pending_days    = "7"
  key_description = "first test key"
}

data "huaweicloud_kms_data_key_v1" "kms_datakey1" {
  key_id         = "${huaweicloud_kms_key_v1.key1.id}"
  datakey_length = "512"
}
```

» Argument Reference

- `key_id` - (Required) The globally unique identifier for the key. Changing this gets the new data encryption key.
- `encryption_context` - (Optional) The value of this parameter must be a series of ”key:value” pairs used to record resource context information. The value of this parameter must not contain sensitive information and must be within 8192 characters in length. Example: {”Key1”:”Value1”,”Key2”:”Value2”}
- `datakey_length` - (Required) Number of bits in the length of a DEK (data encryption keys). The maximum number is 512. Changing this gets the new data encryption key.

» Attributes Reference

`id` is set to the date of the found data key. In addition, the following attributes are exported:

- `plain_text` - The plaintext of a DEK is expressed in hexadecimal format, and two characters indicate one byte.
- `cipher_text` - The ciphertext of a DEK is expressed in hexadecimal format, and two characters indicate one byte.

» huaweicloud_rds_flavors_v1

Use this data source to get the ID of an available HuaweiCloud rds flavor.

» Example Usage

```
data "huaweicloud_rds_flavors_v1" "flavor" {
    region = "eu-de"
    datastore_name = "PostgreSQL"
    datastore_version = "9.5.5"
    speccode = "rds.pg.s1.medium"
}
```

» Argument Reference

- `region` - (Required) The region in which to obtain the V1 rds client.
- `datastore_name` - (Required) The datastore name of the rds.
- `datastore_version` - (Required) The datastore version of the rds.
- `speccode` - (Optional) The spec code of a rds flavor.

» Available value for attributes

datastore_name	datastore_version	speccode
PostgreSQL	9.5.5	ha = True:
	9.6.3	rds.pg.m1.2xlarge.ha rds.pg.c2.large.ha rds.pg.s1.2xlarge.ha rds.pg.c2.
		ha = False:
		rds.pg.s1.xlarge rds.pg.m1.2xlarge rds.pg.c2.xlarge rds.pg.s1.medium r
MySQL	5.6.33	ha = True:
	5.6.30	rds.mysql.s1.medium.ha rds.mysql.s1.large.ha rds.mysql.s1.xlarge.ha r
	5.6.34	ha = False:
	5.6.35	rds.mysql.s1.medium rds.mysql.s1.large rds.mysql.s1.xlarge rds.mysql.
	5.6.36	
	5.7.17	
SQLServer	2014 SP2 SE	
		ha = True:
		rds.mssql.m1.2xlarge.ha rds.mssql.m1.xlarge.ha rds.mssql.m1.4xlarge.h
		ha = False:
		rds.mssql.m1.2xlarge rds.mssql.m1.xlarge rds.mssql.m1.4xlarge rds.ms

» Attributes Reference

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

- `region` - See Argument Reference above.
- `datastore_name` - See Argument Reference above.
- `datastore_version` - See Argument Reference above.
- `speccode` - See Argument Reference above.
- `name` - The name of the rds flavor.
- `ram` - The name of the rds flavor.

» huaweicloud_s3_bucket_object

The S3 object data source allows access to the metadata and *optionally* (see below) content of an object stored inside S3 bucket.

Note: The content of an object (`body` field) is available only for objects which have a human-readable `Content-Type` (`text/*` and `application/json`). This is to prevent printing unsafe characters and potentially downloading large amount of data which would be thrown away in favour of metadata.

» Example Usage

```
data "huaweicloud_s3_bucket_object" "b" {
  bucket = "my-test-bucket"
  key    = "hello-world.zip"
}

## Argument Reference

The following arguments are supported:

* `bucket` - (Required) The name of the bucket to read the object from
* `key` - (Required) The full path to the object inside the bucket
* `version_id` - (Optional) Specific version ID of the object returned (defaults to latest v

## Attributes Reference

The following attributes are exported:

* `body` - Object data (see **limitations above** to understand cases in which this field is
* `cache_control` - Specifies caching behavior along the request/reply chain.
* `content_disposition` - Specifies presentational information for the object.
```

```
* `content_encoding` - Specifies what content encodings have been applied to the object and
* `content_language` - The language the content is in.
* `content_length` - Size of the body in bytes.
* `content_type` - A standard MIME type describing the format of the object data.
* `etag` - [ETag](https://en.wikipedia.org/wiki/HTTP_ETag) generated for the object (an MD5
* `expiration` - If the object expiration is configured (see [object lifecycle management](h
* `expires` - The date and time at which the object is no longer cacheable.
* `last_modified` - Last modified date of the object in RFC1123 format (e.g. `Mon, 02 Jan 20
* `metadata` - A map of metadata stored with the object in S3
* `server_side_encryption` - If the object is stored using server-side encryption (KMS or Am
* `sse_kms_key_id` - If present, specifies the ID of the Key Management Service (KMS) master
* `storage_class` - [Storage class](http://docs.aws.amazon.com/AmazonS3/latest/dev/storage-c
* `version_id` - The latest version ID of the object returned.
* `website_redirect_location` - If the bucket is configured as a website, redirects requests
* `tags`  - A mapping of tags assigned to the object.
```

» Data Source: huaweicloud_sfs_file_sharing_v2

Provides information about an Shared File System (SFS).

» Example Usage

```
variable "share_name" { }

variable "share_id" { }

data "huaweicloud_sfs_file_sharing_v2" "shared_file"
{
       name = "${var.share_name}"
       id = "${var.share_id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Optional) The name of the shared file system.
- `id` - (Optional) The UUID of the shared file system.
- `status` - (Optional) The status of the shared file system.

» Attributes Reference

The following attributes are exported:

- `availability_zone` - The availability zone name.
- `size` - The size (GB) of the shared file system.
- `share_type` - The storage service type for the shared file system, such as high-performance storage (composed of SSDs) or large-capacity storage (composed of SATA disks).
- `status` - The status of the shared file system.
- `host` - The host name of the shared file system.
- `is_public` - The level of visibility for the shared file system.
- `share_proto` - The protocol for sharing file systems.
- `volume_type` - The volume type.
- `metadata` - Metadata key and value pairs as a dictionary of strings.
- `export_location` - The path for accessing the shared file system.
- `access_level` - The level of the access rule.
- `access_rules_status` - The status of the share access rule.
- `access_type` - The type of the share access rule.
- `access_to` - The access that the back end grants or denies.
- `share_access_id` - The UUID of the share access rule.
- `mount_id` - The UUID of the mount location of the shared file system.
- `share_instance_id` - The access that the back end grants or denies.
- `preferred` - Identifies which mount locations are most efficient and are used preferentially when multiple mount locations exist.

» huaweicloud_rts_stack_v1

The Huaweicloud `Resource Template Service` Stack data source allows access to stack outputs and other useful data including the template body.

» Example Usage

The following example shows how one might accept a VPC id as a variable and use this data source to obtain the data necessary to create a subnet within it.

```
variable "stack_name" { }

data "huaweicloud_rts_stack_v1" "stacks" {
  name = "${var.stack_name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the stack.

» Attributes Reference

The following attributes are exported:

- `capabilities` - List of stack capabilities for stack.
- `description` - Describes the stack.
- `disable_rollback` - Specifies whether to perform a rollback if the update fails.
- `outputs` - A list of stack outputs.
- `parameters` - Specifies the stack parameters.
- `template_body` - Structure containing the template body.
- `timeout_mins` - Specifies the timeout duration.
- `status` - Specifies the stack status.
- `name` - Specifies the stack name.
- `status_reason` - Specifies the description of the stack operation.
- `notification_topics` - List of notification topics for stack.

» Data Source: huaweicloud_rts_stack_resource_v1

The HuaweiCloud RTS Stack Resource data source allows access to stack resource metadata.

» Example Usage

```
variable "stack_name" { }
variable "resource_name" { }

data "huaweicloud_rts_stack_resource_v1" "stackresource" {
  stack_name = "${var.stack_name}"
  resource_name = "${var.resource_name}"

}
```

» Argument Reference

The following arguments are supported:

- `stack_name` - (Required) The unique stack name.
- `resource_name` - (Optional) The name of a resource in the stack.
- `physical_resource_id` - (Optional) The physical resource ID.
- `resource_type` - (Optional) The resource type.

» Attributes Reference

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

- `logical_resource_id` - The logical resource ID.
- `resource_status` - The status of the resource.
- `resource_status_reason` - The resource operation reason.
- `required_by` - Specifies the resource dependency.

» huaweicloud_blockstorage_volume_v2

Manages a V2 volume resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_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.

» 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 huaweicloud_blockstorage_volume_v2.volume_1 ea257959-eeb1-4c10-8d33-26f04
```

» huaweicloud_compute_floatingip_v2

Manages a V2 floating IP resource within HuaweiCloud Nova (compute) that can be used for compute instances.

Please note that managing floating IPs through the HuaweiCloud Compute API has been deprecated. Unless you are using an older HuaweiCloud environment, it is recommended to use the `huaweicloud_networking_floatingip_v2` resource instead, which uses the HuaweiCloud Networking API.

» Example Usage

```
resource "huaweicloud_compute_floatingip_v2" "floatip_1" {
  pool = "public"
}
```

» 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 floating IP that can be used with a compute instance. 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` - (Required) The name of the pool from which to obtain the floating IP. Changing this creates a new floating IP.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `pool` - See Argument Reference above.
- `address` - The actual floating IP address itself.
- `fixed_ip` - The fixed IP address corresponding to the floating IP.
- `instance_id` - UUID of the compute instance associated with the floating IP.

» Import

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

```
$ terraform import huaweicloud_compute_floatingip_v2.floatip_1 89c60255-9bd6-460c-822a-e2b95
```

» huaweicloud_compute_floatingip_associate_v2

Associate a floating IP to an instance. This can be used instead of the `floating_ip` options in `huaweicloud_compute_instance_v2`.

» Example Usage

» Automatically detect the correct network

```
resource "huaweicloud_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 "huaweicloud_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "huaweicloud_compute_floatingip_associate_v2" "fip_1" {
```

```
  floating_ip = "${huaweicloud_networking_floatingip_v2.fip_1.address}"
  instance_id = "${huaweicloud_compute_instance_v2.instance_1.id}"
}
```

» Explicitly set the network to attach to

```
resource "huaweicloud_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 "huaweicloud_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "huaweicloud_compute_floatingip_associate_v2" "fip_1" {
  floating_ip = "${huaweicloud_networking_floatingip_v2.fip_1.address}"
  instance_id = "${huaweicloud_compute_instance_v2.instance_1.id}"
  fixed_ip    = "${huaweicloud_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 huaweicloud_compute_floatingip_associate_v2.fip_1 <floating_ip>/<instance
```

» huaweicloud_compute_instance_v2

Manages a V2 VM instance resource within HuaweiCloud.

» Example Usage

» Basic Instance

```
resource "huaweicloud_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 {
    name = "my_network"
  }
}
```

» Instance With Attached Volume

```
resource "huaweicloud_blockstorage_volume_v2" "myvol" {
  name = "myvol"
  size = 1
}

resource "huaweicloud_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 {
    name = "my_network"
  }
}

resource "huaweicloud_compute_volume_attach_v2" "attached" {
  compute_id = "${huaweicloud_compute_instance_v2.myinstance.id}"
  volume_id = "${huaweicloud_blockstorage_volume_v2.myvol.id}"
}
```

» Boot From Volume

```
resource "huaweicloud_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 {
    name = "my_network"
```

```
  }
}
```

» Boot From an Existing Volume

```
resource "huaweicloud_blockstorage_volume_v1" "myvol" {
  name     = "myvol"
  size     = 5
  image_id = "<image-id>"
}

resource "huaweicloud_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                  = "${huaweicloud_blockstorage_volume_v1.myvol.id}"
    source_type           = "volume"
    boot_index            = 0
    destination_type      = "volume"
    delete_on_termination = true
  }

  network {
    name = "my_network"
  }
}
```

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

```
resource "huaweicloud_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 "huaweicloud_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}

resource "huaweicloud_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                  = "${huaweicloud_blockstorage_volume_v2.volume_1.id}"
    source_type           = "volume"
    destination_type      = "volume"
    boot_index            = 1
    delete_on_termination = true
  }
```

```
}
```

» Instance With Multiple Networks

```
resource "huaweicloud_networking_floatingip_v2" "myip" {
  pool = "my_pool"
}

resource "huaweicloud_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 {
    name = "my_first_network"
  }

  network {
    name = "my_second_network"
  }
}

resource "huaweicloud_compute_floatingip_associate_v2" "myip" {
  floating_ip = "${huaweicloud_networking_floatingip_v2.myip.address}"
  instance_id = "${huaweicloud_compute_instance_v2.multi-net.id}"
  fixed_ip = "${huaweicloud_compute_instance_v2.multi-net.network.1.fixed_ip_v4}"
}
```

» Instance With Personality

```
resource "huaweicloud_compute_instance_v2" "personality" {
  name              = "personality"
  image_id          = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id         = "3"
  key_pair          = "my_key_pair_name"
  security_groups   = ["default"]
  availability_zone = "az"

  personality {
    file    = "/path/to/file/on/instance.txt"
    content = "contents of file"
```

```
  }

  network {
    name = "my_network"
  }
}
```

» Instance with Multiple Ephemeral Disks

```
resource "huaweicloud_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 "huaweicloud_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 {
    name = "my_network"
  }
}
```

`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` - (Required) The availability zone in which to create the server. Please refer to https://developer.huaweicloud.com/endpoint for the values. 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.
- `personality` - (Optional) Customize the personality of an instance by defining one or more files and their contents. The personality structure is 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.

The `personality` block supports:

- `file` - (Required) The absolute path of the destination file.
- `contents` - (Required) The contents of the file. Limited to 255 bytes.

» 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 "huaweicloud_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 "huaweicloud_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 "huaweicloud_compute_instance_v2" "instance_1" {
  name = "instance_1"

  network {
    port = "${huaweicloud_networking_port_v2.port_1.id}"
  }

  connection {
    user        = "root"
    host        = "${huaweicloud_networking_port_v2.port_1.fixed_ip.0.ip_address}"
    private_key = "~/path/to/key"
  }

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

» huaweicloud_compute_keypair_v2

Manages a V2 keypair resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_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.
- `value_specs` - (Optional) Map of additional options.

» 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 huaweicloud_compute_keypair_v2.my-keypair test-keypair
```

» huaweicloud_compute_secgroup_v2

Manages a V2 security group resource within HuaweiCloud.

Please note that managing security groups through the HuaweiCloud Compute API has been deprecated. Unless you are using an older HuaweiCloud environment, it is recommended to use the `huaweicloud_networking_secgroup_v2` and `huaweicloud_networking_secgroup_rule_v2` resources instead, which uses the HuaweiCloud Networking API.

» Example Usage

```
resource "huaweicloud_compute_secgroup_v2" "secgroup_1" {
  name        = "my_secgroup"
```

```
  description = "my security group"

  rule {
    from_port   = 22
    to_port     = 22
    ip_protocol = "tcp"
    cidr        = "0.0.0.0/0"
  }

  rule {
    from_port   = 80
    to_port     = 80
    ip_protocol = "tcp"
    cidr        = "0.0.0.0/0"
  }
}
```

» 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 security group. 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. Changing this updates the `name` of an existing security group.
- `description` - (Required) A description for the security group. Changing this updates the `description` of an existing security group.
- `rule` - (Optional) A rule describing how the security group operates. The rule object structure is documented below. Changing this updates the security group rules. As shown in the example above, multiple rule blocks may be used.

The `rule` block supports:

- `from_port` - (Required) An integer representing the lower bound of the port range to open. Changing this creates a new security group rule.
- `to_port` - (Required) An integer representing the upper bound of the port range to open. Changing this creates a new security group rule.
- `ip_protocol` - (Required) The protocol type that will be allowed. Changing this creates a new security group rule.

- `cidr` - (Optional) Required if `from_group_id` or `self` is empty. The IP range that will be the source of network traffic to the security group. Use 0.0.0.0/0 to allow all IP addresses. Changing this creates a new security group rule. Cannot be combined with `from_group_id` or `self`.
- `from_group_id` - (Optional) Required if `cidr` or `self` is empty. The ID of a group from which to forward traffic to the parent group. Changing this creates a new security group rule. Cannot be combined with `cidr` or `self`.
- `self` - (Optional) Required if `cidr` and `from_group_id` is empty. If true, the security group itself will be added as a source to this ingress rule. Cannot be combined with `cidr` or `from_group_id`.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `rule` - See Argument Reference above.

» Notes

» ICMP Rules

When using ICMP as the `ip_protocol`, the `from_port` sets the ICMP *type* and the `to_port` sets the ICMP *code*. To allow all ICMP types, set each value to `-1`, like so:

```
rule {
  from_port = -1
  to_port = -1
  ip_protocol = "icmp"
  cidr = "0.0.0.0/0"
}
```

A list of ICMP types and codes can be found here.

» Referencing Security Groups

When referencing a security group in a configuration (for example, a configuration creates a new security group and then needs to apply it to an instance being created in the same configuration), it is currently recommended to reference the security group by name and not by ID, like this:

```
resource "huaweicloud_compute_instance_v2" "test-server" {
  name            = "tf-test"
  image_id        = "ad091b52-742f-469e-8f3c-fd81cadf0743"
  flavor_id       = "3"
  key_pair        = "my_key_pair_name"
  security_groups = ["${huaweicloud_compute_secgroup_v2.secgroup_1.name}"]
}
```

» Import

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

```
$ terraform import huaweicloud_compute_secgroup_v2.my_secgroup 1bc30ee9-9d5b-4c30-bdd5-7f1e6
```

» huaweicloud_compute_servergroup_v2

Manages a V2 Server Group resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_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.
- `value_specs` - (Optional) Map of additional options.

» 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 huaweicloud_compute_servergroup_v2.test-sg 1bc30ee9-9d5b-4c30-bdd5-7f1e66
```

» huaweicloud_compute_volume_attach_v2

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

» Example Usage

» Basic attachment of a single volume to a single instance

```
resource "huaweicloud_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}

resource "huaweicloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloud_compute_volume_attach_v2" "va_1" {
```

```
  instance_id = "${huaweicloud_compute_instance_v2.instance_1.id}"
  volume_id   = "${huaweicloud_blockstorage_volume_v2.volume_1.id}"
}
```

» Attaching multiple volumes to a single instance

```
resource "huaweicloud_blockstorage_volume_v2" "volumes" {
  count = 2
  name  = "${format("vol-%02d", count.index + 1)}"
  size  = 1
}

resource "huaweicloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "huaweicloud_compute_volume_attach_v2" "attachments" {
  count       = 2
  instance_id = "${huaweicloud_compute_instance_v2.instance_1.id}"
  volume_id   = "${element(huaweicloud_blockstorage_volume_v2.volumes.*.id, count.index)}"
}

output "volume devices" {
  value = "${huaweicloud_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 huaweicloud_compute_volume_attach_v2.va_1 89c60255-9bd6-460c-822a-e2b959e
```

» huaweicloud_dns_recordset_v2

Manages a DNS record set in the HuaweiCloud DNS Service.

» Example Usage

» Automatically detect the correct network

```
resource "huaweicloud_dns_zone_v2" "example_zone" {
  name = "example.com."
  email = "email2@example.com"
  description = "a zone"
  ttl = 6000
  zone_type = "public"
}

resource "huaweicloud_dns_recordset_v2" "rs_example_com" {
  zone_id = "${huaweicloud_dns_zone_v2.example_zone.id}"
  name = "rs.example.com."
  description = "An example record set"
  ttl = 3000
```

```
  type = "A"
  records = ["10.0.0.1"]
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 DNS client. If omitted, the `region` argument of the provider is used. Changing this creates a new DNS record set.
- `zone_id` - (Required) The ID of the zone in which to create the record set. Changing this creates a new DNS record set.
- `name` - (Required) The name of the record set. Note the `.` at the end of the name. Changing this creates a new DNS record set.
- `type` - (Optional) The type of record set. The options include `A`, `AAAA`, `MX`, `CNAME`, `TXT`, `NS`, `SRV`, and `PTR`. Changing this creates a new DNS record set.
- `ttl` - (Optional) The time to live (TTL) of the record set (in seconds). The value range is 300–2147483647. The default value is 300.
- `description` - (Optional) A description of the record set.
- `records` - (Required) An array of DNS records.
- `value_specs` - (Optional) Map of additional options. Changing this creates a new record set.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `type` - See Argument Reference above.
- `ttl` - See Argument Reference above.
- `description` - See Argument Reference above.
- `records` - See Argument Reference above.
- `zone_id` - See Argument Reference above.
- `value_specs` - See Argument Reference above.

» Import

This resource can be imported by specifying the zone ID and recordset ID, separated by a forward slash.

```
$ terraform import huaweicloud_dns_recordset_v2.recordset_1 <zone_id>/<recordset_id>
```

» huaweicloud_dns_zone_v2

Manages a DNS zone in the HuaweiCloud DNS Service.

» Example Usage

» Create a public DNS zone

```
resource "huaweicloud_dns_zone_v2" "my_public_zone" {
  name = "example.com."
  email = "jdoe@example.com"
  description = "An example zone"
  ttl = 3000
  zone_type = "public"
}
```

» Create a private DNS zone

```
resource "huaweicloud_dns_zone_v2" "my_private_zone" {
  name = "1.example.com."
  email = "jdoe@example.com"
  description = "An example zone"
  ttl = 3000
  zone_type = "private"
  router = [
    {router_region = "cn-north-1"
    router_id = "2c1fe4bd-ebad-44ca-ae9d-e94e63847b75"}]
}
```

» 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 DNS zone. Changing this creates a new DNS zone.

- `name` - (Required) The name of the zone. Note the `.` at the end of the name. Changing this creates a new DNS zone.
- `email` - (Optional) The email contact for the zone record. Changing this creates a new DNS zone.
- `zone_type` - (Optional) The type of zone. Can either be `public` or `private`. Changing this creates a new DNS zone.
- `router` - (Optional) Router configuration block which is required if zone_type is private. The router structure is documented below. Changing this creates a new DNS zone.
- `ttl` - (Optional) The time to live (TTL) of the zone. Changing this creates a new DNS zone.
- `description` - (Optional) A description of the zone. Changing this creates a new DNS zone.
- `value_specs` - (Optional) Map of additional options. Changing this creates a new DNS zone.

The `router` block supports:

- `router_id` - (Required) The router UUID. Changing this creates a new DNS zone.
- `router_region` - (Required) The region of the router. Changing this creates a new DNS zone.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `email` - See Argument Reference above.
- `zone_type` - See Argument Reference above.
- `ttl` - See Argument Reference above.
- `description` - See Argument Reference above.
- `masters` - An array of master DNS servers.
- `value_specs` - See Argument Reference above.

» Import

This resource can be imported by specifying the zone ID:

```
$ terraform import huaweicloud_dns_zone_v2.zone_1 <zone_id>
```

» huaweicloud_fw_firewall_group_v2

Manages a v2 firewall group resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_fw_rule_v2" "rule_1" {
  name             = "my-rule-1"
  description      = "drop TELNET traffic"
  action           = "deny"
  protocol         = "tcp"
  destination_port = "23"
  enabled          = "true"
}

resource "huaweicloud_fw_rule_v2" "rule_2" {
  name             = "my-rule-2"
  description      = "drop NTP traffic"
  action           = "deny"
  protocol         = "udp"
  destination_port = "123"
  enabled          = "false"
}

resource "huaweicloud_fw_policy_v2" "policy_1" {
  name = "my-policy"

  rules = ["${huaweicloud_fw_rule_v2.rule_1.id}",
    "${huaweicloud_fw_rule_v2.rule_2.id}",
  ]
}

resource "huaweicloud_fw_firewall_group_v2" "firewall_group_1" {
  name      = "my-firewall-group"
  ingress_policy_id = "${huaweicloud_fw_policy_v2.policy_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 firewall group. If omitted, the `region` argument of the provider is used. Changing this creates a new firewall group.
- `ingress_policy_id` - The ingress policy resource id for the firewall group. Changing this updates the `ingress_policy_id` of an existing firewall group.
- `egress_policy_id` - The egress policy resource id for the firewall group. Changing this updates the `egress_policy_id` of an existing firewall group.
- `name` - (Optional) A name for the firewall group. Changing this updates the `name` of an existing firewall group.
- `description` - (Required) A description for the firewall group. Changing this updates the `description` of an existing firewall group.
- `admin_state_up` - (Optional) Administrative up/down status for the firewall group (must be "true" or "false" if provided - defaults to "true"). Changing this updates the `admin_state_up` of an existing firewall group.
- `tenant_id` - (Optional) The owner of the floating IP. Required if admin wants to create a firewall group for another tenant. Changing this creates a new firewall group.
- `ports` - (Optional) Port(s) to associate this firewall group instance with. Must be a list of strings. Changing this updates the associated routers of an existing firewall group.
- `value_specs` - (Optional) Map of additional options.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `policy_id` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `ports` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloud_fw_firewall_group_v2.firewall_group_1 c9e39fb2-ce20-46c8-a964
```

» huaweicloud_fw_policy_v2

Manages a v2 firewall policy resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_fw_rule_v2" "rule_1" {
  name             = "my-rule-1"
  description      = "drop TELNET traffic"
  action           = "deny"
  protocol         = "tcp"
  destination_port = "23"
  enabled          = "true"
}

resource "huaweicloud_fw_rule_v2" "rule_2" {
  name             = "my-rule-2"
  description      = "drop NTP traffic"
  action           = "deny"
  protocol         = "udp"
  destination_port = "123"
  enabled          = "false"
}

resource "huaweicloud_fw_policy_v2" "policy_1" {
  name = "my-policy"

  rules = ["${huaweicloud_fw_rule_v2.rule_1.id}",
    "${huaweicloud_fw_rule_v2.rule_2.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 firewall policy. If omitted, the `region` argument of the provider is used. Changing this creates a new firewall policy.
- `name` - (Optional) A name for the firewall policy. Changing this updates the `name` of an existing firewall policy.
- `description` - (Optional) A description for the firewall policy. Changing this updates the `description` of an existing firewall policy.
- `rules` - (Optional) An array of one or more firewall rules that comprise the policy. Changing this results in adding/removing rules from the existing firewall policy.
- `audited` - (Optional) Audit status of the firewall policy (must be "true" or "false" if provided - defaults to "false"). This status is set to "false" whenever the firewall policy or any of its rules are changed. Changing this updates the `audited` status of an existing firewall policy.
- `shared` - (Optional) Sharing status of the firewall policy (must be "true" or "false" if provided). If this is "true" the policy is visible to, and can be used in, firewalls in other tenants. Changing this updates the `shared` status of an existing firewall policy. Only administrative users can specify if the policy should be shared.
- `tenant_id` - (Optional) The owner of the firewall policy. Required if admin wants to create a firewall policy for another tenant. Changing this creates a new firewall policy.
- `value_specs` - (Optional) Map of additional options.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `audited` - See Argument Reference above.
- `shared` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.

» Import

Firewall Policies can be imported using the `id`, e.g.

```
$ terraform import huaweicloud_fw_policy_v2.policy_1 07f422e6-c596-474b-8b94-fe2c12506ce0
```

» huaweicloud_fw_rule_v2

Manages a v2 firewall rule resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_fw_rule_v2" "rule_1" {
  name             = "my_rule"
  description      = "drop TELNET traffic"
  action           = "deny"
  protocol         = "tcp"
  destination_port = "23"
  enabled          = "true"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the v2 Networking client. A Compute client is needed to create a firewall rule. If omitted, the `region` argument of the provider is used. Changing this creates a new firewall rule.
- `name` - (Optional) A unique name for the firewall rule. Changing this updates the `name` of an existing firewall rule.
- `description` - (Optional) A description for the firewall rule. Changing this updates the `description` of an existing firewall rule.
- `protocol` - (Required) The protocol type on which the firewall rule operates. Valid values are: `tcp`, `udp`, `icmp`, and `any`. Changing this updates the `protocol` of an existing firewall rule.
- `action` - (Required) Action to be taken (must be "allow" or "deny") when the firewall rule matches. Changing this updates the `action` of an existing firewall rule.
- `ip_version` - (Optional) IP version, either 4 (default) or 6. Changing this updates the `ip_version` of an existing firewall rule.
- `source_ip_address` - (Optional) The source IP address on which the firewall rule operates. Changing this updates the `source_ip_address` of an existing firewall rule.
- `destination_ip_address` - (Optional) The destination IP address on which the firewall rule operates. Changing this updates the `destination_ip_address` of an existing firewall rule.

- `source_port` - (Optional) The source port on which the firewall rule operates. Changing this updates the `source_port` of an existing firewall rule.
- `destination_port` - (Optional) The destination port on which the firewall rule operates. Changing this updates the `destination_port` of an existing firewall rule.
- `enabled` - (Optional) Enabled status for the firewall rule (must be "true" or "false" if provided - defaults to "true"). Changing this updates the `enabled` status of an existing firewall rule.
- `tenant_id` - (Optional) The owner of the firewall rule. Required if admin wants to create a firewall rule for another tenant. Changing this creates a new firewall rule.
- `value_specs` - (Optional) Map of additional options.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `protocol` - See Argument Reference above.
- `action` - See Argument Reference above.
- `ip_version` - See Argument Reference above.
- `source_ip_address` - See Argument Reference above.
- `destination_ip_address` - See Argument Reference above.
- `source_port` - See Argument Reference above.
- `destination_port` - See Argument Reference above.
- `enabled` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloud_fw_rule_v2.rule_1 8dbc0c28-e49c-463f-b712-5c5d1bbac327
```

» huaweicloud_networking_floatingip_v2

Manages a V2 floating IP resource within HuaweiCloud Neutron (networking) that can be used for load balancers. These are similar to Nova (compute) floating

IP resources, but only compute floating IPs can be used with compute instances.

» Example Usage

```
resource "huaweicloud_networking_floatingip_v2" "floatip_1" {
  pool = "admin_external_net"
}
```

» 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` - (Required) The name of the pool from which to obtain the floating IP. 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.
- `value_specs` - (Optional) Map of additional options.

» 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 huaweicloud_networking_floatingip_v2.floatip_1 2c7f39f3-702b-48d1-940c-b5
```

» huaweicloud_networking_network_v2

Manages a V2 Neutron network resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloud_networking_subnet_v2" "subnet_1" {
  name       = "subnet_1"
  network_id = "${huaweicloud_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "huaweicloud_compute_secgroup_v2" "secgroup_1" {
  name        = "secgroup_1"
  description = "a security group"

  rule {
    from_port   = 22
    to_port     = 22
    ip_protocol = "tcp"
    cidr        = "0.0.0.0/0"
  }
}

resource "huaweicloud_networking_port_v2" "port_1" {
  name               = "port_1"
  network_id         = "${huaweicloud_networking_network_v2.network_1.id}"
  admin_state_up     = "true"
  security_group_ids = ["${huaweicloud_compute_secgroup_v2.secgroup_1.id}"]

  fixed_ip {
    "subnet_id"  = "${huaweicloud_networking_subnet_v2.subnet_1.id}"
```

```
    "ip_address" = "192.168.199.10"
  }
}

resource "huaweicloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["${huaweicloud_compute_secgroup_v2.secgroup_1.name}"]

  network {
    port = "${huaweicloud_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. Acceptable values are "true" and "false". Changing this value updates the state of the existing network.
- `segments` - (Optional) An array of one or more provider segment objects.
- `value_specs` - (Optional) Map of additional options.
- `availability_zone_hints` - (Optional) An availability zone is used to make network resources highly available. Used for resources with high availability so that they are scheduled on different availability zones. Changing this creates a new network.

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.
- `availability_zone_hints` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloud_networking_network_v2.network_1 d90ce693-5ccf-4136-a0ed-152ce
```

» huaweicloud_networking_port_v2

Manages a V2 port resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloud_networking_port_v2" "port_1" {
  name           = "port_1"
  network_id     = "${huaweicloud_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.
- `value_specs` - (Optional) Map of additional options.

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 huaweicloud_networking_port_v2.port_1 eae26a3e-1c33-4cc1-9c31-0cd729c438a
```

» Notes

» Ports and Instances

There are some notes to consider when connecting Instances to networks using Ports. Please see the `huaweicloud_compute_instance_v2` documentation for further documentation.

» huaweicloud_networking_router_interface_v2

Manages a V2 router interface resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_networking_network_v2" "network_1" {
  name           = "tf_test_network"
  admin_state_up = "true"
}

resource "huaweicloud_networking_subnet_v2" "subnet_1" {
  network_id = "${huaweicloud_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "huaweicloud_networking_router_v2" "router_1" {
  name                = "my_router"
  external_network_id = "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"
}

resource "huaweicloud_networking_router_interface_v2" "router_interface_1" {
  router_id = "${huaweicloud_networking_router_v2.router_1.id}"
  subnet_id = "${huaweicloud_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 huaweicloud_networking_router_interface_v2.int_1 <port id from above outp
```

» huaweicloud_networking_router_route_v2

Creates a routing entry on a HuaweiCloud V2 router.

» Example Usage

```
resource "huaweicloud_networking_router_v2" "router_1" {
  name           = "router_1"
  admin_state_up = "true"
}

resource "huaweicloud_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "huaweicloud_networking_subnet_v2" "subnet_1" {
  network_id = "${huaweicloud_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "huaweicloud_networking_router_interface_v2" "int_1" {
  router_id = "${huaweicloud_networking_router_v2.router_1.id}"
  subnet_id = "${huaweicloud_networking_subnet_v2.subnet_1.id}"
}
```

```
resource "huaweicloud_networking_router_route_v2" "router_route_1" {
  depends_on       = ["huaweicloud_networking_router_interface_v2.int_1"]
  router_id        = "${huaweicloud_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 `huaweicloud_networking_router_route_v2` resource creation time. You can ensure that by explicitly specifying a dependency on the `huaweicloud_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 huaweicloud_networking_router_route_v2.router_route_1 686fe248-386c-4f70-
```

» huaweicloud_networking_router_v2

Manages a V2 router resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_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.
- `external_fixed_ip` - (Optional) An external fixed IP for the router. This can be repeated. The structure is described below. An `external_network_id` has to be set in order to set this property. Changing this updates the external fixed IPs 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.
- `value_specs` - (Optional) Map of additional driver-specific options.
- `availability_zone_hints` - (Optional) An availability zone is used to make network resources highly available. Used for resources with high availability so that they are scheduled on different availability zones. Changing this creates a new router.

The `external_fixed_ip` block supports:

- `subnet_id` - (Optional) Subnet in which the fixed IP belongs to.
- `ip_address` - (Optional) The IP address to set on the 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.
- `external_fixed_ip` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `value_specs` - See Argument Reference above.
- `availability_zone_hints` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloud_networking_router_v2.router_1 014395cd-89fc-4c9b-96b7-13d1ee7
```

» huaweicloud_networking_subnet_v2

Manages a V2 Neutron subnet resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_networking_network_v2" "network_1" {
  name           = "tf_test_network"
  admin_state_up = "true"
}

resource "huaweicloud_networking_subnet_v2" "subnet_1" {
  network_id = "${huaweicloud_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.
- `value_specs` - (Optional) Map of additional options.

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 huaweicloud_networking_subnet_v2.subnet_1 da4faf16-5546-41e4-8330-4d0002b
```

» huaweicloud_networking_secgroup_v2

Manages a V2 neutron security group resource within HuaweiCloud. 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 "huaweicloud_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, HuaweiCloud 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 "huaweicloud_networking_secgroup_rule_v2" "secgroup_rule_v4" {
  direction = "egress"
  ethertype = "IPv4"
  security_group_id = "${huaweicloud_networking_secgroup_v2.secgroup.id}"
}

resource "huaweicloud_networking_secgroup_rule_v2" "secgroup_rule_v6" {
  direction = "egress"
  ethertype = "IPv6"
  security_group_id = "${huaweicloud_networking_secgroup_v2.secgroup.id}"
}
```

Please note that this behavior may differ depending on the configuration of the HuaweiCloud cloud. The above illustrates the current default Neutron behavior. Some HuaweiCloud clouds 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 huaweicloud_networking_secgroup_v2.secgroup_1 38809219-5e8a-4852-9139-6f4
```

» huaweicloud_networking_secgroup_rule_v2

Manages a V2 neutron security group rule resource within HuaweiCloud. 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 "huaweicloud_networking_secgroup_v2" "secgroup_1" {
  name        = "secgroup_1"
  description = "My neutron security group"
}
```

```
resource "huaweicloud_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 = "${huaweicloud_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 huaweicloud_networking_secgroup_rule_v2.secgroup_rule_1 aeb68ee3-6e9d-425
```

» huaweicloud_lb_loadbalancer_v2

Manages a V2 loadbalancer resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_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. A Networking client is needed to create an LB member. If omitted, the `region` argument of the provider is used. Changing this creates a new LB member.
- `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).
- `flavor` - (Optional) The UUID of a flavor. Changing this creates a new loadbalancer.
- `loadbalancer_provider` - (Optional) The name of the provider. Changing this creates a new loadbalancer.

- `security_group_ids` - (Optional) A list of security group IDs to apply to the loadbalancer. The security groups must be specified by ID and not name (as opposed to how they are configured with the Compute Instance).

» Attributes Reference

The following attributes are exported:

- `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.
- `flavor` - See Argument Reference above.
- `loadbalancer_provider` - See Argument Reference above.
- `security_group_ids` - See Argument Reference above.
- `vip_port_id` - The Port ID of the Load Balancer IP.

» huaweicloud_lb_listener_v2

Manages a V2 listener resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_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. A Networking client is needed to create an . If omitted, the `region` argument of the provider is used. Changing this creates a new Listener.
- `protocol` - (Required) The protocol - can either be TCP, HTTP, HTTPS or TERMINATED_HTTPS. Changing this creates a new Listener.

- `protocol_port` - (Required) The port on which to listen for client traffic. 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.
- `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`. See here for more information.
- `sni_container_refs` - (Optional) A list of references to Barbican Secrets containers which store SNI information. See here for more information.
- `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.
- `sni_container_refs` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.

» huaweicloud_lb_pool_v2

Manages a V2 pool resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_lb_pool_v2" "pool_1" {
  protocol    = "HTTP"
  lb_method   = "ROUND_ROBIN"
  listener_id = "d9415786-5f1a-428b-b35f-2f1523e146d2"

  persistence {
    type        = "HTTP_COOKIE"
    cookie_name = "testCookie"
  }
}
```

» 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 . If omitted, the `region` argument of the provider is used. Changing this creates a new pool.
- `tenant_id` - (Optional) Required for admins. The UUID of the tenant who owns the pool. Only administrative users can specify a tenant UUID other than their own. Changing this creates a new pool.
- `name` - (Optional) Human-readable name for the pool.
- `description` - (Optional) Human-readable description for the pool.
- `protocol` = (Required) The protocol - can either be TCP, HTTP or HTTPS. 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. Required if `type = APP_COOKIE`.

» Attributes Reference

The following attributes are exported:

- `id` - The unique ID for the pool.
- `tenant_id` - See Argument Reference above.
- `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.

» huaweicloud_lb_member_v2

Manages a V2 member resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_lb_member_v2" "member_1" {
  address       = "192.168.199.23"
  protocol_port = 8080
}
```

» 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 . If omitted, the `region` argument of the provider is used. Changing this creates a new member.
- `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.
- `tenant_id` - (Optional) Required for admins. The UUID of the tenant who owns the member. Only administrative users can specify a tenant UUID other than their own. Changing this creates a new 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.
- `tenant_id` - 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.

» huaweicloud_lb_monitor_v2

Manages a V2 monitor resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_lb_monitor_v2" "monitor_1" {
  pool_id     = "${huaweicloud_lb_pool_v2.pool_1.id}"
  type        = "PING"
  delay       = 20
  timeout     = 10
  max_retries = 5
}
```

» 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 . If omitted, the `region` argument of the provider is used. Changing this creates a new monitor.
- `pool_id` - (Required) The id of the pool that this monitor will be assigned to.
- `name` - (Optional) The Name of the Monitor.
- `tenant_id` - (Optional) Required for admins. The UUID of the tenant who owns the monitor. Only administrative users can specify a tenant UUID other than their own. Changing this creates a new monitor.
- `type` - (Required) The type of probe, which is PING, TCP, HTTP, or HTTPS, that is sent by the load balancer to verify the member state. Changing this creates a new monitor.
- `delay` - (Required) The time, in seconds, between sending probes to members.
- `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(S) types. URI path that will be accessed if monitor type is HTTP or HTTPS.
- `http_method` - (Optional) Required for HTTP(S) 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(S) 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.
- `tenant_id` - See Argument Reference above.
- `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.

» huaweicloud_kms_key_v1

Manages a V1 key resource within KMS.

» Example Usage

```
resource "huaweicloud_kms_key_v1" "key_1" {
  key_alias       = "key_1"
  pending_days    = "7"
  key_description = "first test key"
  realm           = "cn-north-1"
  is_enabled      = true
}
```

» Argument Reference

The following arguments are supported:

- `key_alias` - (Required) The alias in which to create the key. It is required when we create a new key. Changing this updates the alias of key.

- `key_description` - (Optional) The description of the key as viewed in Huawei console. Changing this updates the description of key.
- `realm` - (Optional) Region where a key resides. Changing this creates a new key.
- `pending_days` - (Optional) Duration in days after which the key is deleted after destruction of the resource, must be between 7 and 1096 days. It doesn't have default value. It only be used when delete a key.
- `is_enabled` - (Optional) Specifies whether the key is enabled. Defaults to true. Changing this updates the state of existing key.

» Attributes Reference

The following attributes are exported:

- `key_alias` - See Argument Reference above.
- `key_description` - See Argument Reference above.
- `realm` - See Argument Reference above.
- `key_id` - The globally unique identifier for the key.
- `default_key_flag` - Identification of a Master Key. The value 1 indicates a Default Master Key, and the value 0 indicates a key.
- `scheduled_deletion_date` - Scheduled deletion time (time stamp) of a key.
- `domain_id` - ID of a user domain for the key.
- `expiration_time` - Expiration time.
- `creation_date` - Creation time (time stamp) of a key.
- `is_enabled` - See Argument Reference above.

» Import

KMS Keys can be imported using the `id`, e.g.

```
$ terraform import huaweicloud_kms_key_v1.key_1 7056d636-ac60-4663-8a6c-82d3c32c1c64
```

» huaweicloud_nat_gateway_v2

Manages a V2 nat gateway resource within HuaweiCloud Nat

» Example Usage

```
resource "huaweicloud_nat_gateway_v2" "nat_1" {
  name   = "Terraform"
```

```
  description = "test for terraform2"
  spec = "3"
  router_id = "2c1fe4bd-ebad-44ca-ae9d-e94e63847b75"
  internal_network_id = "dc8632e2-d9ff-41b1-aa0c-d455557314a0"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 nat client. If omitted, the `region` argument of the provider is used. Changing this creates a new nat gateway.
- `name` - (Required) The name of the nat gateway.
- `description` - (Optional) The description of the nat gateway.
- `spec` - (Required) The specification of the nat gateway, valid values are ”1”, ”2”, ”3”, ”4”.
- `tenant_id` - (Optional) The target tenant ID in which to allocate the nat gateway. Changing this creates a new nat gateway.
- `router_id` - (Required) ID of the router this nat gateway belongs to. Changing this creates a new nat gateway.
- `internal_network_id` - (Optional) ID of the network this nat gateway connects to. Changing this creates a new nat gateway.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `spec` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `router_id` - See Argument Reference above.
- `internal_network_id` - See Argument Reference above.

» huaweicloud_nat_snat_rule_v2

Manages a V2 snat rule resource within HuaweiCloud Nat

» Example Usage

```
resource "huaweicloud_nat_snat_rule_v2" "snat_1" {
  nat_gateway_id = "3c0dffda-7c76-452b-9dcc-5bce7ae56b17"
  network_id = "dc8632e2-d9ff-41b1-aa0c-d455557314a0"
  floating_ip_id = "0a166fc5-a904-42fb-b1ef-cf18afeeddca"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 nat client. If omitted, the `region` argument of the provider is used. Changing this creates a new snat rule.
- `nat_gateway_id` - (Required) ID of the nat gateway this snat rule belongs to. Changing this creates a new snat rule.
- `network_id` - (Required) ID of the network this snat rule connects to. Changing this creates a new snat rule.
- `floating_ip_id` - (Required) ID of the floating ip this snat rule connets to. Changing this creates a new snat rule.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `nat_gateway_id` - See Argument Reference above.
- `network_id` - See Argument Reference above.
- `floating_ip_id` - See Argument Reference above.

» huaweicloud_elb_loadbalancer

Manages an elastic loadbalancer resource within huawei cloud.

» Example Usage

```
resource "huaweicloud_elb_loadbalancer" "elb" {
  name = "elb"
  type = "External"
  description = "test elb"
```

```
  vpc_id = "e346dc4a-d9a6-46f4-90df-10153626076e"
  admin_state_up = 1
  bandwidth = 5
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the load balancer name. The name is a string of 1 to 64 characters that consist of letters, digits, underscores (_), and hyphens (-).
- `description` - (Optional) Provides supplementary information about the listener. The value is a string of 0 to 128 characters and cannot be <>.
- `vpc_id` - (Required) Specifies the VPC ID.
- `bandwidth` - (Optional) Specifies the bandwidth (Mbit/s). This parameter is mandatory when type is set to External, and it is invalid when type is set to Internal. The value ranges from 1 to 300.
- `type` - (Required) Specifies the load balancer type. The value can be Internal or External.
- `admin_state_up` - (Required) Specifies the status of the load balancer. Value range: 0 or false: indicates that the load balancer is stopped. Only tenants are allowed to enter these two values. 1 or true: indicates that the load balancer is running properly. 2 or false: indicates that the load balancer is frozen. Only tenants are allowed to enter these two values.
- `vip_subnet_id` - (Optional) Specifies the ID of the private network to be added. This parameter is mandatory when type is set to Internal, and it is invalid when type is set to External.
- `az` - (Optional) Specifies the ID of the availability zone (AZ). This parameter is mandatory when type is set to Internal, and it is invalid when type is set to External.
- `charge_mode` - (Optional) This is a reserved field. If the system supports charging by traffic and this field is specified, then you are charged by traffic for elastic IP addresses. The value is traffic.
- `eip_type` - (Optional) This parameter is reserved.
- `security_group_id` - (Optional) Specifies the security group ID. The value is a string of 1 to 200 characters that consists of uppercase and lowercase letters, digits, and hyphens (-). This parameter is mandatory only when type is set to Internal.

- `vip_address` - (Optional) Specifies the IP address provided by ELB. When typeis set to External, the value of this parameter is the elastic IP address. When type is set to Internal, the value of this parameter is the private network IP address. You can select an existing elastic IP address and create a public network load balancer. When this parameter is configured, parameters bandwidth, charge_mode, and eip_type are invalid.
- `tenantid` - (Optional) Specifies the tenant ID. This parameter is mandatory only when type is set to Internal.

» Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `vpc_id` - See Argument Reference above.
- `bandwidth` - See Argument Reference above.
- `type` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `vip_subnet_id` - See Argument Reference above.
- `az` - See Argument Reference above.
- `charge_mode` - See Argument Reference above.
- `eip_type` - See Argument Reference above.
- `security_group_id` - See Argument Reference above.
- `vip_address` - See Argument Reference above.
- `tenantid` - See Argument Reference above.
- `update_time` - Specifies the time when information about the load balancer was updated.
- `create_time` - Specifies the time when the load balancer was created.
- `id` - Specifies the load balancer ID.
- `status` - Specifies the status of the load balancer. The value can be ACTIVE, PENDING_CREATE, or ERROR.

» huaweicloud_elb_listener

Manages an elastic loadbalancer listener resource within huawei cloud.

» Example Usage

```
resource "huaweicloud_elb_loadbalancer" "elb" {
  name = "elb"
```

```
  type = "External"
  description = "test elb"
  vpc_id = "e346dc4a-d9a6-46f4-90df-10153626076e"
  admin_state_up = 1
  bandwidth = 5
}

resource "huaweicloud_elb_listener" "listener" {
  name = "test-elb-listener"
  description = "great listener"
  protocol = "TCP"
  backend_protocol = "TCP"
  port = 12345
  backend_port = 8080
  lb_algorithm = "roundrobin"
  loadbalancer_id = "${huaweicloud_elb_loadbalancer.elb.id}"
  timeouts {
    create = "5m"
    update = "5m"
    delete = "5m"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the load balancer name. The name is a string of 1 to 64 characters that consist of letters, digits, underscores (_), and hyphens (-).
- `description` - (Optional) Provides supplementary information about the listener. The value is a string of 0 to 128 characters and cannot be <>.
- `loadbalancer_id` - (Required) Specifies the ID of the load balancer to which the listener belongs.
- `protocol` - (Required) Specifies the listening protocol used for layer 4 or 7. The value can be HTTP, TCP, HTTPS, or UDP.
- `port` - (Required) Specifies the listening port. The value ranges from 1 to 65535.
- `backend_protocol` - (Required) Specifies the backend protocol. If the value of protocol is UDP, the value of this parameter can only be UDP. The value can be HTTP, TCP, or UDP.

- `backend_port` - (Required) Specifies the backend port. The value ranges from 1 to 65535.
- `lb_algorithm` - (Required) Specifies the load balancing algorithm for the listener. The value can be roundrobin, leastconn, or source.
- `session_sticky` - (Optional) Specifies whether to enable sticky session. The value can be true or false. The Sticky session is enabled when the value is true, and is disabled when the value is false. If the value of protocol is HTTP, HTTPS, or TCP, and the value of lb_algorithm is not roundrobin, the value of this parameter can only be false.
- `sticky_session_type` - (Optional) Specifies the cookie processing method. The value is insert. insert indicates that the cookie is inserted by the load balancer. This parameter is valid when protocol is set to HTTP, and session_sticky to true. The default value is insert. This parameter is invalid when protocol is set to TCP or UDP, which means the parameter is empty.
- `cookie_timeout` - (Optional) Specifies the cookie timeout period (minutes). This parameter is valid when protocol is set to HTTP, session_sticky to true, and sticky_session_type to insert. This parameter is invalid when protocol is set to TCP or UDP. The value ranges from 1 to 1440.
- `tcp_timeout` - (Optional) Specifies the TCP timeout period (minutes). This parameter is valid when protocol is set to TCP. The value ranges from 1 to 5.
- `tcp_draining` - (Optional) Specifies whether to maintain the TCP connection to the backend ECS after the ECS is deleted. This parameter is valid when protocol is set to TCP. The value can be true or false.
- `tcp_draining_timeout` - (Optional) Specifies the timeout duration (minutes) for the TCP connection to the backend ECS after the ECS is deleted. This parameter is valid when protocol is set to TCP, and tcp_draining to true. The value ranges from 0 to 60.
- `certificate_id` - (Optional) Specifies the ID of the SSL certificate used for security authentication when HTTPS is used to make API calls. This parameter is mandatory if the value of protocol is HTTPS. The value can be obtained by viewing the details of the SSL certificate.
- `udp_timeout` - (Optional) Specifies the UDP timeout duration (minutes). This parameter is valid when protocol is set to UDP. The value ranges from 1 to 1440.
- `ssl_protocols` - (Optional) Specifies the SSL protocol standard supported by a tracker, which is used for enabling specified encryption protocols. This parameter is valid only when the value of protocol is set to

HTTPS. The value is TLSv1.2 or TLSv1.2 TLSv1.1 TLSv1. The default value is TLSv1.2.

- `ssl_ciphers` - (Optional) Specifies the cipher suite of an encryption protocol. This parameter is valid only when the value of protocol is set to HTTPS. The value is Default, Extended, or Strict. The default value is Default. The value can only be set to Extended if the value of ssl_protocols is set to TLSv1.2 TLSv1.1 TLSv1.

» Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `loadbalancer_id` - See Argument Reference above.
- `protocol` - See Argument Reference above.
- `port` - See Argument Reference above.
- `backend_protocol` - See Argument Reference above.
- `backend_port` - See Argument Reference above.
- `lb_algorithm` - See Argument Reference above.
- `session_sticky` - See Argument Reference above.
- `sticky_session_type` - See Argument Reference above.
- `cookie_timeout` - See Argument Reference above.
- `tcp_timeout` - See Argument Reference above.
- `tcp_draining` - See Argument Reference above.
- `tcp_draining_timeout` - See Argument Reference above.
- `certificate_id` - See Argument Reference above.
- `udp_timeout` - See Argument Reference above.
- `ssl_protocols` - See Argument Reference above.
- `ssl_ciphers` - See Argument Reference above.
- `update_time` - Specifies the time when information about the listener was updated.
- `id` - Specifies the listener ID.
- `create_time` - Specifies the time when the listener was created.
- `status` - Specifies the listener status. The value can be ACTIVE, PENDING_CREATE, or ERROR.
- `admin_state_up` - Specifies the status of the load balancer. Value range: false: The load balancer is disabled. true: The load balancer runs properly.
- `member_number` - Specifies the number of backend members.
- `healthcheck_id` - Specifies the health check task ID.

» huaweicloud_elb_healthcheck

Manages an elastic loadbalancer healthcheck resource within huawei cloud.

» Example Usage

```
resource "huaweicloud_elb_loadbalancer" "elb" {
  name = "elb"
  type = "External"
  description = "test elb"
  vpc_id = "e346dc4a-d9a6-46f4-90df-10153626076e"
  admin_state_up = 1
  bandwidth = 5
}

resource "huaweicloud_elb_listener" "listener" {
  name = "test-elb-listener"
  description = "great listener"
  protocol = "TCP"
  backend_protocol = "TCP"
  port = 12345
  backend_port = 8080
  lb_algorithm = "roundrobin"
  loadbalancer_id = "${huaweicloud_elb_loadbalancer.elb.id}"
  timeouts {
    create = "5m"
    update = "5m"
    delete = "5m"
  }
}

resource "huaweicloud_elb_healthcheck" "healthcheck" {
  listener_id = "${huaweicloud_elb_listener.listener.id}"
  healthcheck_protocol = "TCP"
  healthcheck_connect_porta = 22
  healthy_threshold = 5
  healthcheck_timeout = 25
  healthcheck_interval = 3
  timeouts {
    create = "5m"
    update = "5m"
    delete = "5m"
  }
}
```

» Argument Reference

The following arguments are supported:

- `listener_id` - (Required) Specifies the ID of the listener to which the health check task belongs.
- `healthcheck_protocol` - (Optional) Specifies the protocol used for the health check. The value can be HTTP or TCP (case-insensitive).
- `healthcheck_uri` - (Optional) Specifies the URI for health check. This parameter is valid when healthcheck_ protocol is HTTP. The value is a string of 1 to 80 characters that must start with a slash (/) and can only contain letters, digits, and special characters, such as -/.%?#&.
- `healthcheck_connect_port` - (Optional) Specifies the port used for the health check. The value ranges from 1 to 65535.
- `healthy_threshold` - (Optional) Specifies the threshold at which the health check result is success, that is, the number of consecutive successful health checks when the health check result of the backend server changes from fail to success. The value ranges from 1 to 10.
- `unhealthy_threshold` - (Optional) Specifies the threshold at which the health check result is fail, that is, the number of consecutive failed health checks when the health check result of the backend server changes from success to fail. The value ranges from 1 to 10.
- `healthcheck_timeout` - (Optional) Specifies the maximum timeout duration (s) for the health check. The value ranges from 1 to 50.
- `healthcheck_interval` - (Optional) Specifies the maximum interval (s) for health check. The value ranges from 1 to 5.

» Attributes Reference

The following attributes are exported:

- `listener_id` - See Argument Reference above.
- `healthcheck_protocol` - See Argument Reference above.
- `healthcheck_uri` - See Argument Reference above.
- `healthcheck_connect_port` - See Argument Reference above.
- `healthy_threshold` - See Argument Reference above.
- `unhealthy_threshold` - See Argument Reference above.
- `healthcheck_timeout` - See Argument Reference above.
- `healthcheck_interval` - See Argument Reference above.
- `id` - Specifies the health check task ID.
- `update_time` - Specifies the time when information about the health check task was updated.

- `create_time` - Specifies the time when the health check task was created.

» huaweicloud_elb_backendecs

Manages an elastic loadbalancer backendecs resource within huawei cloud.

» Example Usage

```
resource "huaweicloud_elb_loadbalancer" "elb" {
  name = "elb"
  type = "External"
  description = "test elb"
  vpc_id = "e346dc4a-d9a6-46f4-90df-10153626076e"
  admin_state_up = 1
  bandwidth = 5
}

resource "huaweicloud_elb_listener" "listener" {
  name = "test-elb-listener"
  description = "great listener"
  protocol = "TCP"
  backend_protocol = "TCP"
  port = 12345
  backend_port = 8080
  lb_algorithm = "roundrobin"
  loadbalancer_id = "${huaweicloud_elb_loadbalancer.elb.id}"
  timeouts {
    create = "5m"
    update = "5m"
    delete = "5m"
  }
}

resource "huaweicloud_elb_backendecs" "backend" {
  private_address = "192.168.0.211"
  listener_id = "${huaweicloud_elb_listener.listener.id}"
  server_id = "8f7a32f1-f66c-4d13-9b17-3a13f9f0bb8d"
}
```

» Argument Reference

The following arguments are supported:

- `listener_id` - (Required) Specifies the listener ID.
- `server_id` - (Required) Specifies the backend member ID.
- `private_address` - (Required) Specifies the private IP address of the backend member.

» Attributes Reference

The following attributes are exported:

- `listener_id` - See Argument Reference above.
- `server_id` - See Argument Reference above.
- `private_address` - See Argument Reference above.
- `public_address` - Specifies the floating IP address assigned to the backend member.
- `id` - Specifies the backend member ID.
- `status` - Specifies the backend ECS status. The value is ACTIVE, PENDING, or ERROR.
- `health_status` - Specifies the health check status. The value is NORMAL, ABNORMAL, or UNAVAILABLE.
- `update_time` - Specifies the time when information about the backend member was updated.
- `create_time` - Specifies the time when the backend member was created.
- `server_name` - Specifies the backend member name.
- `listeners` - Specifies the listener to which the backend member belongs.

» huaweicloud_s3_bucket

Provides a S3 bucket resource.

» Example Usage

» Private Bucket w/ Tags

```
resource "huaweicloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"
}
```

» Static Website Hosting

```
resource "huaweicloud_s3_bucket" "b" {
```

```
  bucket = "s3-website-test.hashicorp.com"
  acl    = "public-read"
  policy = "${file("policy.json")}"

  website {
    index_document = "index.html"
    error_document = "error.html"

    routing_rules = <<EOF
[{
    "Condition": {
        "KeyPrefixEquals": "docs/"
    },
    "Redirect": {
        "ReplaceKeyPrefixWith": "documents/"
    }
}]
EOF
  }
}
```

» Using CORS

```
resource "huaweicloud_s3_bucket" "b" {
  bucket = "s3-website-test.hashicorp.com"
  acl    = "public-read"

  cors_rule {
    allowed_headers = ["*"]
    allowed_methods = ["PUT", "POST"]
    allowed_origins = ["https://s3-website-test.hashicorp.com"]
    expose_headers  = ["ETag"]
    max_age_seconds = 3000
  }
}
```

» Using versioning

```
resource "huaweicloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  versioning {
    enabled = true
```

```
  }
}
```

» Enable Logging

```
resource "huaweicloud_s3_bucket" "log_bucket" {
  bucket = "my-tf-log-bucket"
  acl    = "log-delivery-write"
}

resource "huaweicloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  logging {
    target_bucket = "${huaweicloud_s3_bucket.log_bucket.id}"
    target_prefix = "log/"
  }
}
```

» Using object lifecycle

```
resource "huaweicloud_s3_bucket" "bucket" {
  bucket = "my-bucket"
  acl    = "private"

  lifecycle_rule {
    id      = "log"
    enabled = true

    prefix  = "log/"

    expiration {
      days = 90
    }
  }

  lifecycle_rule {
    id      = "tmp"
    prefix  = "tmp/"
    enabled = true

    expiration {
      date = "2016-01-12"
```

```
    }
  }
}

resource "huaweicloud_s3_bucket" "versioning_bucket" {
  bucket = "my-versioning-bucket"
  acl    = "private"

  versioning {
    enabled = true
  }

  lifecycle_rule {
    prefix  = "config/"
    enabled = true
  }
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Optional, Forces new resource) The name of the bucket. If omitted, Terraform will assign a random, unique name.
- `bucket_prefix` - (Optional, Forces new resource) Creates a unique bucket name beginning with the specified prefix. Conflicts with `bucket`.
- `acl` - (Optional) The canned ACL to apply. Defaults to "private".
- `policy` - (Optional) A valid bucket policy JSON document. Note that if the policy document is not specific enough (but still valid), Terraform may view the policy as constantly changing in a `terraform plan`. In this case, please make sure you use the verbose/specific version of the policy.
- `force_destroy` - (Optional, Default:false) A boolean that indicates all objects should be deleted from the bucket so that the bucket can be destroyed without error. These objects are *not* recoverable.
- `website` - (Optional) A website object (documented below).
- `cors_rule` - (Optional) A rule of Cross-Origin Resource Sharing (documented below).
- `versioning` - (Optional) A state of versioning (documented below)
- `logging` - (Optional) A settings of bucket logging (documented below).
- `lifecycle_rule` - (Optional) A configuration of object lifecycle management (documented below).
- `region` - (Optional) If specified, the region this bucket should reside in. Otherwise, the region used by the callee.

The `website` object supports the following:

- `index_document` - (Required, unless using `redirect_all_requests_to`) Amazon S3 returns this index document when requests are made to the root domain or any of the subfolders.
- `error_document` - (Optional) An absolute path to the document to return in case of a 4XX error.
- `redirect_all_requests_to` - (Optional) A hostname to redirect all website requests for this bucket to. Hostname can optionally be prefixed with a protocol (`http://` or `https://`) to use when redirecting requests. The default is the protocol that is used in the original request.
- `routing_rules` - (Optional) A json array containing routing rules describing redirect behavior and when redirects are applied.

The `CORS` object supports the following:

- `allowed_headers` (Optional) Specifies which headers are allowed.
- `allowed_methods` (Required) Specifies which methods are allowed. Can be `GET`, `PUT`, `POST`, `DELETE` or `HEAD`.
- `allowed_origins` (Required) Specifies which origins are allowed.
- `expose_headers` (Optional) Specifies expose header in the response.
- `max_age_seconds` (Optional) Specifies time in seconds that browser can cache the response for a preflight request.

The `versioning` object supports the following:

- `enabled` - (Optional) Enable versioning. Once you version-enable a bucket, it can never return to an unversioned state. You can, however, suspend versioning on that bucket.
- `mfa_delete` - (Optional) Enable MFA delete for either `Change the versioning state of your bucket` or `Permanently delete an object version`. Default is `false`.

The `logging` object supports the following:

- `target_bucket` - (Required) The name of the bucket that will receive the log objects.
- `target_prefix` - (Optional) To specify a key prefix for log objects.

The `lifecycle_rule` object supports the following:

- `id` - (Optional) Unique identifier for the rule.
- `prefix` - (Optional) Object key prefix identifying one or more objects to which the rule applies.
- `enabled` - (Required) Specifies lifecycle rule status.
- `abort_incomplete_multipart_upload_days` (Optional) Specifies the number of days after initiating a multipart upload when the multipart upload must be completed.
- `expiration` - (Optional) Specifies a period in the object's expire (documented below).
- `noncurrent_version_expiration` - (Optional) Specifies when noncurrent object versions expire (documented below).

At least one of `expiration`, `noncurrent_version_expiration` must be specified.

The `expiration` object supports the following

- `date` (Optional) Specifies the date after which you want the corresponding action to take effect.
- `days` (Optional) Specifies the number of days after object creation when the specific rule action takes effect.
- `expired_object_delete_marker` (Optional) On a versioned bucket (versioning-enabled or versioning-suspended bucket), you can add this element in the lifecycle configuration to direct Amazon S3 to delete expired object delete markers.

The `noncurrent_version_expiration` object supports the following

- `days` (Required) Specifies the number of days an object is noncurrent object versions expire.

The `rules` object supports the following:

- `id` - (Optional) Unique identifier for the rule.
- `destination` - (Required) Specifies the destination for the rule (documented below).
- `prefix` - (Required) Object keyname prefix identifying one or more objects to which the rule applies. Set as an empty string to replicate the whole bucket.
- `status` - (Required) The status of the rule. Either `Enabled` or `Disabled`. The rule is ignored if status is not Enabled.

The `destination` object supports the following:

- `bucket` - (Required) The ARN of the S3 bucket where you want Amazon S3 to store replicas of the object identified by the rule.
- `storage_class` - (Optional) The class of storage used to store the object.

» Attributes Reference

The following attributes are exported:

- `id` - The name of the bucket.
- `arn` - The ARN of the bucket. Will be of format `arn:aws:s3:::bucketname`.
- `bucket_domain_name` - The bucket domain name. Will be of format `bucketname.s3.amazonaws.com`.
- `hosted_zone_id` - The Route 53 Hosted Zone ID for this bucket's region.
- `region` - The region this bucket resides in.
- `website_endpoint` - The website endpoint, if the bucket is configured with a website. If not, this will be an empty string.

- `website_domain` - The domain of the website endpoint, if the bucket is configured with a website. If not, this will be an empty string. This is used to create Route 53 alias records.

» Import

S3 bucket can be imported using the `bucket`, e.g.

```
$ terraform import huaweicloud_s3_bucket.bucket bucket-name
```

» huaweicloud_s3_bucket_policy

Attaches a policy to an S3 bucket resource.

» Example Usage

» Basic Usage

```
resource "huaweicloud_s3_bucket" "b" {
  bucket = "my_tf_test_bucket"
}

resource "huaweicloud_s3_bucket_policy" "b" {
  bucket = "${huaweicloud_s3_bucket.b.id}"
  policy =<<POLICY
{
  "Version": "2012-10-17",
  "Id": "MYBUCKETPOLICY",
  "Statement": [
    {
      "Sid": "IPAllow",
      "Effect": "Deny",
      "Principal": "*",
      "Action": "s3:*",
      "Resource": "arn:aws:s3:::my_tf_test_bucket/*",
      "Condition": {
         "IpAddress": {"aws:SourceIp": "8.8.8.8/32"}
      }
    }
  ]
}
POLICY
}
```

» Argument Reference

The following arguments are supported:

- `bucket` - (Required) The name of the bucket to which to apply the policy.
- `policy` - (Required) The text of the policy.

» huaweicloud_smn_subscription_v2

Manages a V2 subscription resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_smn_topic_v2" "topic_1" {
  name        = "topic_1"
  display_name    = "The display name of topic_1"
}

resource "huaweicloud_smn_subscription_v2" "subscription_1" {
  topic_urn       = "${huaweicloud_smn_topic_v2.topic_1.id}"
  endpoint        = "mailtest@gmail.com"
  protocol        = "email"
  remark          = "O&M"
}

resource "huaweicloud_smn_subscription_v2" "subscription_2" {
  topic_urn       = "${huaweicloud_smn_topic_v2.topic_1.id}"
  endpoint        = "13600000000"
  protocol        = "sms"
  remark          = "O&M"
}
```

» Argument Reference

The following arguments are supported:

- `topic_urn` - (Required) Resource identifier of a topic, which is unique.
- `endpoint` - (Required) Message endpoint. For an HTTP subscription, the endpoint starts with http://. For an HTTPS subscription, the endpoint starts with https://. For an email subscription, the endpoint is a mail address. For an SMS message subscription, the endpoint is a phone number.

- `protocol` - (Required) Protocol of the message endpoint. Currently, email, sms, http, and https are supported.
- `remark` - (Optional) Remark information. The remarks must be a UTF-8-coded character string containing 128 bytes.
- `subscription_urn` - (Optional) Resource identifier of a subscription, which is unique.
- `owner` - (Optional) Project ID of the topic creator.
- `status` - (Optional) Subscription status. 0 indicates that the subscription is not confirmed. 1 indicates that the subscription is confirmed. 3 indicates that the subscription is canceled.

» Attributes Reference

The following attributes are exported:

- `topic_urn` - See Argument Reference above.
- `endpoint` - See Argument Reference above.
- `protocol` - See Argument Reference above.
- `remark` - See Argument Reference above.
- `subscription_urn` - See Argument Reference above.
- `owner` - See Argument Reference above.
- `status` - See Argument Reference above.

» huaweicloud_smn_topic_v2

Manages a V2 topic resource within HuaweiCloud.

» Example Usage

```
resource "huaweicloud_smn_topic_v2" "topic_1" {
  name            = "topic_1"
  display_name    = "The display name of topic_1"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the topic to be created.

- `display_name` - (Optional) Topic display name, which is presented as the name of the email sender in an email message.
- `topic_urn` - (Optional) Resource identifier of a topic, which is unique.
- `push_policy` - (Optional) Message pushing policy. 0 indicates that the message sending fails and the message is cached in the queue. 1 indicates that the failed message is discarded.
- `create_time` - (Optional) Time when the topic was created.
- `update_time` - (Optional) Time when the topic was updated.

» Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.
- `display_name` - See Argument Reference above.
- `topic_urn` - See Argument Reference above.
- `push_policy` - See Argument Reference above.
- `create_time` - See Argument Reference above.
- `update_time` - See Argument Reference above.

» huaweicloud_rds_instance_v1

Manages rds instance resource within HuaweiCloud

» Example Usage: Creating a PostgreSQL RDS instance

```
data "huaweicloud_rds_flavors_v1" "flavor" {
    region = "eu-de"
    datastore_name = "PostgreSQL"
    datastore_version = "9.5.5"
    speccode = "rds.pg.s1.large"
}

resource "huaweicloud_compute_secgroup_v2" "secgrp_rds" {
  name        = "secgrp-rds-instance"
  description = "Rds Security Group"
}

resource "huaweicloud_rds_instance_v1" "instance" {
  name = "rds-instance"
  datastore {
```

```
    type = "PostgreSQL"
    version = "9.5.5"
  }
  flavorref = "${data.huaweicloud_rds_flavors_v1.flavor.id}"
  volume {
    type = "COMMON"
    size = 200
  }
  region = "eu-de"
  availabilityzone = "eu-de-01"
  vpc = "c1095fe7-03df-4205-ad2d-6f4c181d436e"
  nics {
    subnetid = "b65f8d25-c533-47e2-8601-cfaa265a3e3e"
  }
  securitygroup {
    id = "${huaweicloud_compute_secgroup_v2.secgrp_rds.id}"
  }
  dbport = "8635"
  backupstrategy = {
    starttime = "04:00:00"
    keepdays = 4
  }
  dbrtpd = "Huangwei!120521"
  ha = {
    enable = true
    replicationmode = "async"
  }
  depends_on = ["huaweicloud_compute_secgroup_v2.secgrp_rds"]
}
```

» Example Usage: Creating a SQLServer RDS instance

```
data "huaweicloud_rds_flavors_v1" "flavor" {
    region = "eu-de"
    datastore_name = "SQLServer"
    datastore_version = "2014 SP2 SE"
    speccode = "rds.mssql.s1.2xlarge"
}

resource "huaweicloud_compute_secgroup_v2" "secgrp_rds" {
  name        = "secgrp-rds-instance"
  description = "Rds Security Group"
}

resource "huaweicloud_rds_instance_v1" "instance" {
```

```
  name = "rds-instance"
  datastore {
    type = "SQLServer"
    version = "2014 SP2 SE"
  }
  flavorref = "${data.huaweicloud_rds_flavors_v1.flavor.id}"
  volume {
    type = "COMMON"
    size = 200
  }
  region = "eu-de"
  availabilityzone = "eu-de-01"
  vpc = "c1095fe7-03df-4205-ad2d-6f4c181d436e"
  nics {
    subnetid = "b65f8d25-c533-47e2-8601-cfaa265a3e3e"
  }
  securitygroup {
    id = "${huaweicloud_compute_secgroup_v2.secgrp_rds.id}"
  }
  dbport = "8635"
  backupstrategy = {
    starttime = "04:00:00"
    keepdays = 4
  }
  dbrtpd = "Huangwei!120521"
  depends_on = ["huaweicloud_compute_secgroup_v2.secgrp_rds"]
}
```

» Example Usage: Creating a MySQL RDS instance

```
data "huaweicloud_rds_flavors_v1" "flavor" {
    region = "eu-de"
    datastore_name = "MySQL"
    datastore_version = "5.6.33"
    speccode = "rds.mysql.s1.medium"
}

resource "huaweicloud_compute_secgroup_v2" "secgrp_rds" {
  name        = "secgrp-rds-instance"
  description = "Rds Security Group"
}

resource "huaweicloud_rds_instance_v1" "instance" {
  name = "rds-instance"
  datastore {
```

```
  type = "MySQL"
  version = "5.6.33"
}
flavorref = "${data.huaweicloud_rds_flavors_v1.flavor.id}"
volume {
  type = "COMMON"
  size = 200
}
region = "eu-de"
availabilityzone = "eu-de-01"
vpc = "c1095fe7-03df-4205-ad2d-6f4c181d436e"
nics {
  subnetid = "b65f8d25-c533-47e2-8601-cfaa265a3e3e"
}
securitygroup {
  id = "${huaweicloud_compute_secgroup_v2.secgrp_rds.id}"
}
dbport = "8635"
backupstrategy = {
  starttime = "04:00:00"
  keepdays = 4
}
dbrtpd = "Huangwei!120521"
ha = {
  enable = true
  replicationmode = "async"
}
depends_on = ["huaweicloud_compute_secgroup_v2.secgrp_rds"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the DB instance name. The DB instance name of the same type is unique in the same tenant.
- `datastore` - (Required) Specifies database information. The structure is described below.
- `flavorref` - (Required) Specifies the specification ID (flavors.id in the response message in Obtaining All DB Instance Specifications).
- `volume` - (Required) Specifies the volume information. The structure is described below.
- `region` - (Required) Specifies the region ID.

- `availabilityzone` - (Required) Specifies the ID of the AZ.
- `vpc` - (Required) Specifies the VPC ID. For details about how to obtain this parameter value, see section "Virtual Private Cloud" in the Virtual Private Cloud API Reference.
- `nics` - (Required) Specifies the nics information. For details about how to obtain this parameter value, see section "Subnet" in the Virtual Private Cloud API Reference. The structure is described below.
- `securitygroup` - (Required) Specifies the security group which the RDS DB instance belongs to. The structure is described below.
- `dbport` - (Optional) Specifies the database port number.
- `backupstrategy` - (Optional) Specifies the advanced backup policy. The structure is described below.
- `dbrtpd` - (Required) Specifies the password for user root of the database.
- `ha` - (Optional) Specifies the parameters configured on HA and is used when creating HA DB instances. The structure is described below. NOTICE: RDS for Microsoft SQL Server does not support creating HA DB instances and this parameter is not involved.

The `datastore` block supports:

- `type` - (Required) Specifies the DB engine. Currently, MySQL, PostgreSQL, and Microsoft SQL Server are supported. The value is MySQL, PostgreSQL, or SQLServer.
- `version` - (Required) Specifies the DB instance version.
- Available value for attributes

type	version
PostgreSQL	9.5.5
	9.6.3
MySQL	5.6.33
	5.6.30
	5.6.34
	5.6.35
	5.6.36
	5.7.17
SQLServer	2014 SP2 SE

The `volume` block supports:

- `type` - (Required) Specifies the volume type. Valid value: It must be COMMON (SATA) or ULTRAHIGH (SSD) and is case-sensitive.

- `size` - (Required) Specifies the volume size. Its value must be a multiple of 10 and the value range is 100 GB to 2000 GB.

The `nics` block supports:

- `subnetId` - (Required) Specifies the subnet ID obtained from the VPC.

The `securitygroup` block supports:

- `id` - (Required) Specifies the ID obtained from the securitygroup.

The `backupstrategy` block supports:

- `starttime` - (Optional) Indicates the backup start time that has been set. The backup task will be triggered within one hour after the backup start time. Valid value: The value cannot be empty. It must use the hh:mm:ss format and must be valid. The current time is the UTC time.
- `keepdays` - (Optional) Specifies the number of days to retain the generated backup files. Its value range is 0 to 35. If this parameter is not specified or set to 0, the automated backup policy is disabled.

The `ha` block supports:

- `enable` - (Optional) Specifies the configured parameters on the HA. Valid value: The value is true or false. The value true indicates creating HA DB instances. The value false indicates creating a single DB instance.
- `replicationmode` - (Optional) Specifies the replication mode for the standby DB instance. The value cannot be empty. For MySQL, the value is async or semisync. For PostgreSQL, the value is async or sync.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `flavorref` - See Argument Reference above.
- `volume` - See Argument Reference above.
- `availabilityzone` - See Argument Reference above.
- `vpc` - See Argument Reference above.
- `nics` - See Argument Reference above.
- `securitygroup` - See Argument Reference above.
- `dbport` - See Argument Reference above.
- `backupstrategy` - See Argument Reference above.
- `dbrtpd` - See Argument Reference above.
- `ha` - See Argument Reference above.
- `status` - Indicates the DB instance status.
- `hostname` - Indicates the instance connection address. It is a blank string.

- `type` - Indicates the DB instance type, which can be master or readreplica.
- `created` - Indicates the creation time in the following format: yyyy-mm-dd Thh:mm:ssZ.
- `updated` - Indicates the update time in the following format: yyyy-mm-dd Thh:mm:ssZ.

» Attributes Reference

The following attributes can be updated:

- `volume.size` - See Argument Reference above.
- `flavorref` - See Argument Reference above.
- `backupstrategy` - See Argument Reference above.

» huaweicloud_vpc_eip_v1

Manages a V1 EIP resource within Huawei Cloud VPC.

» Example Usage

```
resource "huaweicloud_vpc_eip_v1" "eip_1" {
  publicip {
    type = "5_bgp"
  }
  bandwidth {
    name = "test"
    size = 8
    share_type = "PER"
    charge_mode = "traffic"
  }
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the eip. If omitted, the `region` argument of the provider is used. Changing this creates a new eip.
- `publicip` - (Required) The elastic IP address object.
- `bandwidth` - (Required) The bandwidth object.

The `publicip` block supports:

- `type` - (Required) The value must be a type supported by the system. Only `5_bgp` supported now. Changing this creates a new eip.
- `ip_address` - (Optional) The value must be a valid IP address in the available IP address segment. Changing this creates a new eip.
- `port_id` - (Optional) The port id which this eip will associate with. If the value is "" or this not specified, the eip will be in unbind state.

The `bandwidth` block supports:

- `name` - (Required) The bandwidth name, which is a string of 1 to 64 characters that contain letters, digits, underscores (_), and hyphens (-).
- `size` - (Required) The bandwidth size. The value ranges from 1 to 300 Mbit/s.
- `charge_type` - (Required) Whether the bandwidth is shared or exclusive. Changing this creates a new eip.
- `charge_mode` - (Optional) This is a reserved field. If the system supports charging by traffic and this field is specified, then you are charged by traffic for elastic IP addresses. Changing this creates a new eip.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `publicip/type` - See Argument Reference above.
- `publicip/ip_address` - See Argument Reference above.
- `publicip/port_id` - See Argument Reference above.
- `bandwidth/name` - See Argument Reference above.
- `bandwidth/size` - See Argument Reference above.
- `bandwidth/charge_type` - See Argument Reference above.
- `bandwidth/charge_mode` - See Argument Reference above.

» Import

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

```
$ terraform import huaweicloud_vpc_eip_v1.eip_1 2c7f39f3-702b-48d1-940c-b50384177ee1
```

» huaweicloud_sfs_file_system_v2

Provides an Shared File System (SFS) resource.

» Example Usage

```
variable "share_name" { }

variable "share_description" { }

variable "vpc_id" { }

resource "huaweicloud_sfs_file_system_v2" "sfs1" {
        size = 50
        name = "${var.share_name}"
        access_to = "${var.vpc_id}"
        access_level = "rw"
        description = "${var.share_description}"
        metadata = {
            "type"="nfs"
        }
}
```

» Argument Reference

The following arguments are supported:

- `size` - (Required) The size (GB) of the shared file system.
- `share_proto` - (Optional) The protocol for sharing file systems. The default value is NFS.
- `name` - (Optional) The name of the shared file system.
- `description` - (Optional) Describes the shared file system.
- `is_public` - (Optional) The level of visibility for the shared file system.
- `metadata` - (Optional) Metadata key and value pairs as a dictionary of strings.Changing this will create a new resource.
- `availability_zone` - (Optional) The availability zone name.Changing this parameter will create a new resource.
- `access_level` - (Required) The access level of the shared file system. Changing this will create a new access rule.

- `access_type` - (Optional) The type of the share access rule. Changing this will create a new access rule.
- `access_to` - (Required) The access that the back end grants or denies. Changing this will create a new access rule

» Attributes Reference

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

- `id` - The UUID of the shared file system.
- `status` - The status of the shared file system.
- `share_type` - The storage service type assigned for the shared file system, such as high-performance storage (composed of SSDs) and large-capacity storage (composed of SATA disks).
- `volume_type` - The volume type.
- `export_location` - The address for accessing the shared file system.
- `host` - The host name of the shared file system.
- `share_access_id` - The UUID of the share access rule.
- `access_rules_status` - The status of the share access rule.

» Import

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

```
> $ terraform import huaweicloud_sfs_file_system_v2 4779ab1c-7c1a-44b1-a02e-93dfc361b32d
```

» huaweicloud_rts_stack_v1_

Provides an Huawei Cloud Stack resource.

» Example Usage

```
 variable "name" { }
 variable "network_id" { }
 variable "instance_type" { }
variable "image_id" { }

resource "huaweicloud_rts_stack_v1" "stack" {
```

```
name = "${var.name}"
  disable_rollback = true
  timeout_mins=60
  parameters = {
      "network_id" = "${var.network_id}"
      "instance_type" = "${var.instance_type}"
      "image_id" = "${var.image_id}"
    }
  template_body = <<STACK
  {
    "heat_template_version": "2016-04-08",
    "description": "Simple template to deploy",
    "parameters": {
        "image_id": {
            "type": "string",
            "description": "Image to be used for compute instance",
            "label": "Image ID"
        },
        "network_id": {
            "type": "string",
            "description": "The Network to be used",
            "label": "Network UUID"
        },
        "instance_type": {
            "type": "string",
            "description": "Type of instance (Flavor) to be used",
            "label": "Instance Type"
        }
    },
    "resources": {
        "my_instance": {
            "type": "OS::Nova::Server",
            "properties": {
                "image": {
                    "get_param": "image_id"
                },
                "flavor": {
                    "get_param": "instance_type"
                },
                "networks": [{
                    "network": {
                        "get_param": "network_id"
                    }
                }]
            }
        }
```

```
      },
      "outputs":  {
        "InstanceIP":{
          "description": "Instance IP",
          "value": {  "get_attr": ["my_instance", "first_address"]  }
        }
      }
  }
  STACK
}
```

» Argument Reference

The following arguments are supported:

- `stack_name` - (Required) Specifies the stack name. The value must meet the regular expression rule (`[a-zA-Z][a-zA-Z0-9_.-]{0,254}$`). Changing this will create a new stack.
- `stack_id` - (Required) Specifies the stack UUID.
- `template` - (Optional) Specifies the template. The template content must use the json syntax.
- `environment` - (Optional) Specifies the environment information about the stack.
- `files` - (Optional) Specifies files used in the environment.
- `parameters` - (Optional) Specifies parameter information of the stack.
- `timeout_mins` - (Optional) Specifies the timeout duration.
- `template_url` - (Optional) Specifies the template URL.
- `disable_rollback` - (Optional) Specifies whether to perform a rollback if the creation fails.

» Attributes Reference

The following attributes are exported:

- `outputs` - A map of outputs from the stack.
- `capabilities` - List of stack capabilities for stack.
- `notification_topics` - List of notification topics for stack.
- `status` - Specifies the stack status.

» Import

RTS Stacks can be imported using the `name`, e.g.

```
$ terraform import huaweicloud_rts_stack_v1.stack rts-stack
```

» Timeouts

`huaweicloud_rts_stack_v1` provides the following Timeouts configuration options:

- `create` - (Default `30 minutes`) Used for Creating Stacks
- `update` - (Default `30 minutes`) Used for Stack modifications
- `delete` - (Default `30 minutes`) Used for destroying stacks.

» huaweicloud_iam_agency_v3

Manages an agency resource within huawei cloud.

» Example Usage

```
resource "huaweicloud_iam_agency_v3" "agency" {
  name = "test_agency"
  description = "test agency"
  delegated_domain_name = "***"
  project_role = [
    {
      project = "cn-north-1"
      roles = [
        "Tenant Administrator",
      ]
    }
  ]
  domain_roles = [
      "Anti-DDoS Administrator",
  ]
}
```

Note: It can not set `tenant_name` in `provider "huaweicloud"` when using this resource.

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of agency. The name is a string of 1 to 64 characters.
- `description` - (Optional) Provides supplementary information about the agency. The value is a string of 0 to 255 characters.
- `delegated_domain_name` - (Required) The name of delegated domain.
- `project_role` - (Optional) An array of roles and projects which are used to grant permissions to agency on project. The structure is documented below.
- `domain_roles` - (optional) An array of role names which stand for the permissionis to be granted to agency on domain.

The `project_role` block supports:

- `project` - (Required) The name of project
- `roles` - (Required) An array of role names

note: one or both of `project_role` and `domain_roles` must be input when creating an agency.

» Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `delegated_domain_name` - See Argument Reference above.
- `project_role` - See Argument Reference above.
- `domain_roles` - See Argument Reference above.
- `duration` - Validity period of an agency. The default value is null, indicating that the agency is permanently valid.
- `expire_time` - The expiration time of agency
- `create_time` - The time when the agency was created.
- `id` - The agency ID.