» opentelekomcloud_images_image_v2

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

» Example Usage

```
data "opentelekomcloud_images_image_v2" "ubuntu" {
  name = "Ubuntu 16.04"
  most_recent = true
}
```

» Argument Reference

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

» Attributes Reference

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

- `checksum` - The checksum of the data associated with the image.
- `created_at` - The date the image was created.
- `container_format`: The format of the image's container.
- `disk_format`: The format of the image's disk.

- `file` - the trailing path after the glance endpoint that represent the location of the image or the path to retrieve it.
- `metadata` - The metadata associated with the image. Image metadata allow for meaningfully define the image properties and tags. See http://docs.openstack.org/developer/glance/metadefs-concepts.html.
- `min_disk_gb`: The minimum amount of disk space required to use the image.
- `min_ram_mb`: The minimum amount of ram required to use the image.
- `protected` - Whether or not the image is protected.
- `schema` - The path to the JSON-schema that represent the image or image
- `size_bytes` - The size of the image (in bytes).
- `tags` - See Argument Reference above.
- `update_at` - The date the image was last updated.

» opentelekomcloud_networking_network_v2

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

» Example Usage

```
data "opentelekomcloud_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.
- `matching_subnet_cidr` - (Optional) The CIDR of a subnet within the network.
- `tenant_id` - (Optional) The owner of the network.

» Attributes Reference

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

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

» opentelekomcloud_networking_secgroup_v2

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

» Example Usage

```
data "opentelekomcloud_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.

» opentelekomcloud_kms_key_v1

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

» Example Usage

```
data "opentelekomcloud_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"
  origin           = "kms"
}
```

» 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 OpenTelekomCloud 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.
- `origin` - Origin of a key. such as: kms. 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.
- `origin` - 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.

» opentelekomcloud_kms_data_key_v1

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

» Example Usage

```
resource "opentelekomcloud_kms_key_v1" "key1" {
  key_alias       = "key_1"
  pending_days    = "7"
  key_description = "first test key"
}

data "opentelekomcloud_kms_data_key_v1" "kms_datakey1" {
  key_id         = "${opentelekomcloud_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.

» opentelekomcloud_rds_flavors_v1

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

» Example Usage

```
data "opentelekomcloud_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 9.6.3 9.6.5	rds.pg.s1.xlarge rds.pg.m1.2xlarge rds.pg.c2.xlarge rds.pg.s1.medium r

datastore_name	datastore_version	speccode
MySQL	5.6.33	rds.mysql.s1.medium rds.mysql.s1.large rds.mysql.s1.xlarge rds.mysql.s
	5.6.30	
	5.6.34	
	5.6.35	
	5.6.36	
	5.7.17	
	5.7.20	
SQLServer	2014 SP2 SE	rds.mssql.s1.xlarge rds.mssql.m1.2xlarge rds.mssql.c2.xlarge rds.mssql.

» 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.

» opentelekomcloud_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 "opentelekomcloud_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.
```

» opentelekomcloud_vpc_v1

opentelekomcloud_vpc_v1 provides details about a specific VPC.

This resource can prove useful when a module accepts a vpc id as an input variable and needs to, for example, determine the CIDR block of that VPC.

» 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 "vpc_name" {}

data "opentelekomcloud_vpc_v1" "vpc" {
  name = "${var.vpc_name}"
```

```
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available VPCs in the current region. The given filters must match exactly one VPC whose data will be exported as attributes.

- `region` - (Optional) The region in which to obtain the V1 VPC client. A VPC client is needed to retrieve VPCs. If omitted, the region argument of the provider is used.
- `id` - (Optional) The id of the specific VPC to retrieve.
- `status` - (Optional) The current status of the desired VPC. Can be either CREATING, OK, DOWN, PENDING_UPDATE, PENDING_DELETE, or ERROR.
- `name` - (Optional) A unique name for the VPC. The name must be unique for a tenant. The value is a string of no more than 64 characters and can contain digits, letters, underscores (_), and hyphens (-).
- `cidr` - (Optional) The cidr block of the desired VPC.

» Attributes Reference

The following attributes are exported:

- `id` - ID of the VPC.
- `name` - See Argument Reference above.
- `status` - See Argument Reference above.
- `cidr` - See Argument Reference above.
- `routes` - The list of route information with destination and nexthop fields.
- `shared` - Specifies whether the cross-tenant sharing is supported.
- `region` - See Argument Reference above.

» Data Source: opentelekomcloud_vpc_subnet_v1

`opentelekomcloud_vpc_subnet_v1` provides details about a specific VPC subnet.

This resource can prove useful when a module accepts a subnet id as an input variable and needs to, for example, determine the id of the VPC that the subnet belongs to.

» Example Usage

```
data "opentelekomcloud_vpc_subnet_v1" "subnet_v1" {
  id   = "${var.subnet_id}"
 }

output "subnet_vpc_id" {
  value = "${data.opentelekomcloud_vpc_subnet_v1.subnet_v1.vpc_id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available subnets in the current tenant. The given filters must match exactly one subnet whose data will be exported as attributes.

- `id` - (Optional) - Specifies a resource ID in UUID format.
- `name` (Optional) - The name of the specific subnet to retrieve.
- `cidr` (Optional) - The network segment of specific subnet to retrieve. The value must be in CIDR format.
- `status` (Optional) - The value can be ACTIVE, DOWN, UNKNOWN, or ERROR.
- `vpc_id` (Optional) - The id of the VPC that the desired subnet belongs to.
- `gateway_ip` (Optional) - The subnet gateway address of specific subnet.
- `primary_dns` (Optional) - The IP address of DNS server 1 on the specific subnet.
- `secondary_dns` (Optional) - The IP address of DNS server 2 on the specific subnet.
- `availability_zone` (Optional) - The availability zone (AZ) to which the subnet should belong.

» Attributes Reference

All of the argument attributes are also exported as result attributes. This data source will complete the data by populating any fields that are not included in

the configuration with the data for the selected subnet.

- `dns_list` - The IP address list of DNS servers on the subnet.
- `dhcp_enable` - DHCP function for the subnet.
- `subnet_id` - Specifies the subnet (Native OpenStack API) ID.

» Data Source: opentelekomcloud_vpc_subnet_ids_v1

`opentelekomcloud_vpc_subnet_ids_v1` provides a list of subnet ids for a vpc_id

This resource can be useful for getting back a list of subnet ids for a vpc.

» Example Usage

The following example shows outputing all cidr blocks for every subnet id in a vpc.

```
data "opentelekomcloud_vpc_subnet_ids_v1" "subnet_ids" {
  vpc_id = "${var.vpc_id}"
}

data "opentelekomcloud_vpc_subnet_v1" "subnet" {
  count = "${length(data.opentelekomcloud_vpc_subnet_ids_v1.subnet_ids.ids)}"
  id    = "${data.opentelekomcloud_vpc_subnet_ids_v1.subnet_ids.ids[count.index]}"
 }

output "subnet_cidr_blocks" {
  value = "${data.opentelekomcloud_vpc_subnet_v1.subnet.*.cidr}"
}
```

» Argument Reference

The following arguments are supported:

- `vpc_id` (Required) - Specifies the VPC ID used as the query filter.

» Attributes Reference

The following attributes are exported:

- `ids` - A list of all the subnet ids found. This data source will fail if none are found.

» Data Source: opentelekomcloud_vpc_peering_connection_v2

The VPC Peering Connection data source provides details about a specific VPC peering connection.

» Example Usage

```
data "opentelekomcloud_vpc_peering_connection_v2" "peering" {
   vpc_id          = "${opentelekomcloud_vpc_v1.vpc.id}"
   peer_vpc_id     = "${opentelekomcloud_vpc_v1.peer_vpc.id}"
 }


resource "opentelekomcloud_vpc_route_v2" "vpc_route" {
  type       = "peering"
  nexthop    = "${data.opentelekomcloud_vpc_peering_connection_v2.peering.id}"
  destination = "192.168.0.0/16"
  vpc_id = "${opentelekomcloud_vpc_v1.vpc.id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available VPC peering connection. The given filters must match exactly one VPC peering connection whose data will be exported as attributes.

- `id` (Optional) - The ID of the specific VPC Peering Connection to retrieve.
- `status` (Optional) - The status of the specific VPC Peering Connection to retrieve.
- `vpc_id` (Optional) - The ID of the requester VPC of the specific VPC Peering Connection to retrieve.
- `peer_vpc_id` (Optional) - The ID of the accepter/peer VPC of the specific VPC Peering Connection to retrieve.
- `peer_tenant_id` (Optional) - The Tenant ID of the accepter/peer VPC of the specific VPC Peering Connection to retrieve.
- `name` (Optional) - The name of the specific VPC Peering Connection to retrieve.

» Attributes Reference

All of the argument attributes are exported as result attributes.

» Data Source: opentelekomcloud_vpc_route_v2

`opentelekomcloud_vpc_route_v2` provides details about a specific VPC route.

» Example Usage

```
 variable "route_id" { }

data "opentelekomcloud_vpc_route_v2" "vpc_route" {
  id = "${var.route_id}"
}

resource "opentelekomcloud_vpc_subnet_v1" "subnet_v1" {
  name = "test-subnet"
  cidr = "192.168.0.0/24"
  gateway_ip = "192.168.0.1"
  vpc_id = "${data.opentelekomcloud_vpc_route_v2.vpc_route.vpc_id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available routes in the current tenant. The given filters must match exactly one route whose data will be exported as attributes.

- `id` (Optional) - The id of the specific route to retrieve.
- `vpc_id` (Optional) - The id of the VPC that the desired route belongs to.
- `destination` (Optional) - The route destination address (CIDR).
- `tenant_id` (Optional) - Only the administrator can specify the tenant ID of other tenants.
- `type` (Optional) - Route type for filtering.

» Attribute Reference

All of the argument attributes are also exported as result attributes.

- `nexthop` - The next hop of the route. If the route type is peering, it will provide VPC peering connection ID.

» Data Source: opentelekomcloud_vpc_route_ids_v2

`opentelekomcloud_vpc_route_ids_v2` provides a list of route ids for a vpc_id.

This resource can be useful for getting back a list of route ids for a vpc.

» Example Usage

```
 variable "vpc_id" { }

data "opentelekomcloud_vpc_route_ids_v2" "example" {
  vpc_id = "${var.vpc_id}"
}

data "opentelekomcloud_vpc_route_v2" "vpc_route" {
  count = "${length(data.opentelekomcloud_vpc_route_ids_v2.example.ids)}"
  id = "${data.opentelekomcloud_vpc_route_ids_v2.example.ids[count.index]}"
}

output "route_nexthop" {
  value = ["${data.opentelekomcloud_vpc_route_v2.vpc_route.*.nexthop}"]
}
```

» Argument Reference

- `vpc_id` (Required) - The VPC ID that you want to filter from.

» Attributes Reference

- `ids` - A list of all the route ids found. This data source will fail if none are found.

» Data Source: opentelekomcloud_rts_stack_v1

The OpenTelekomCloud RTS Stack data source allows access to stack outputs and other useful data including the template body.

» Example Usage

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

```
data "opentelekomcloud_rts_stack_v1" "mystack" {
  name = "${var.stack_name}"
}
```

» Argument Reference

The following arguments are supported:

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

» Attributes Reference

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

- `id` - A unique identifier of the stack.
- `capabilities` - List of stack capabilities for stack.
- `notification_topics` - List of notification topics for stack.
- `status` - Specifies the stack status.
- `disable_rollback` - Whether the rollback of the stack is disabled when stack creation fails.
- `outputs` - A list of stack outputs.
- `parameters` - A map of parameters that specify input parameters for the stack.
- `template_body` - Structure containing the template body.
- `timeout_mins` - Specifies the timeout duration.

» Data Source: opentelekomcloud_rts_stack_resource_v1

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

» Example Usage

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

data "opentelekomcloud_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.

» Data Source: opentelekomcloud_sfs_file_system_v2

Provides information about an Shared File System (SFS).

» Example Usage

```
variable "share_name" { }

variable "share_id" { }

data "opentelekomcloud_sfs_file_system_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.

» Data Source: opentelekomcloud_rts_software_deployment_v1

The RTS Software Deployment data source provides details about a specific RTS Software Deployment.

» Example Usage

```
variable "deployment_id" {}

data "opentelekomcloud_rts_software_deployment_v1" "mydeployment" {
  id = "${var.deployment_id}"
}
```

» Argument Reference

The following arguments are supported:

- `id` - (Optional) The id of the software deployment.
- `config_id` - (Optional) The id of the software configuration resource running on an instance.
- `server_id` - (Optional) The id of the instance.
- `status` - (Optional) The current status of deployment resources.
- `action` - (Optional) The stack action that triggers this deployment resource.

» Attributes Reference

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

- `input_values` - The input data stored in the form of a key-value pair.
- `output_values` - The output data stored in the form of a key-value pair.
- `status_reason` - The cause of the current deployment resource status.

» Data Source: opentelekomcloud_rts_software_config_v1

The RTS Software Config data source provides details about a specific RTS Software Config.

» Example Usage

```
variable "config_name" {}

variable "server_id" {}

data "opentelekomcloud_rts_software_config_v1" "myconfig" {
  id = "${var.config_name}"
}

resource "opentelekomcloud_rts_software_deployment_v1" "mydeployment" {
  config_id = "${data.opentelekomcloud_rts_software_config_v1.myconfig.id}"
  server_id = "${var.server_id}"
}
```

» Argument Reference

The following arguments are supported:

- `id` - (Optional) The id of the software configuration.
- `name` - (Optional) The name of the software configuration.

» Attributes Reference

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

- `group` - The namespace that groups this software configuration by when it is delivered to a server.
- `inputs` - A list of software configuration inputs.
- `outputs` - A list of software configuration outputs.
- `config` - The software configuration code.
- `options` - The software configuration options.

» Data Source: opentelekomcloud_compute_bms_flavors_v2

`opentelekomcloud_compute_bms_flavors_v2` used to query flavors of BMSs.

» Example Usage

```
variable "flavor_id" { }
variable "disk_size" { }

data "opentelekomcloud_compute_bms_flavors_v2" "Query_BMS_flavors"
{
    id = "${var.bms_id}",
    min_disk = "${var.disk_size}",
    sort_key = "id",
    sort_dir = "desc",
}
```

» Argument Reference

The arguments of this data source act as filters for querying the BMSs details.

- `name` - (Optional) - The name of the BMS flavor.
- `id` (Optional) - The BMS flavor id.
- `min_ram` (Optional) - The minimum memory size in MB. Only the BMSs with the memory size greater than or equal to the minimum size can be queried.
- `min_disk` (Optional) - The minimum disk size in GB. Only the BMSs with a disk size greater than or equal to the minimum size can be queried.
- `sort_key` (Optional) - The sorting field. The default value is **flavorid**. The other values are **name**, **memory_mb**, **vcpus**, **root_gb**, or **flavorid**.
- `sort_dir` (Optional) - The sorting order, which can be **ascending** (**asc**) or **descending** (**desc**). The default value is **asc**.

» Attributes Reference

All of the argument attributes are also exported as result attributes.

- `ram` - It is the memory size (in MB) of the flavor.
- `vcpus` - It is the number of CPU cores in the BMS flavor.
- `disk` - Specifies the disk size (GB) in the BMS flavor.
- `swap` - This is a reserved attribute.
- `rx_tx_factor` - This is a reserved attribute.

» Data Source: opentelekomcloud_compute_bms_keypairs_v2

`opentelekomcloud_compute_bms_keypairs_v2` used to query SSH key pairs.

» Example Usage

```
variable "keypair_name" {}

data "opentelekomcloud_compute_bms_keypairs_v2" "Query_BMS_keypair"
{
    name = "${var.keypair_name}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the BMSs details.

- `name` - (Required) - It is the key pair name.

» Attributes Reference

All of the argument attributes are also exported as result attributes.

- `public_key` - It gives the information about the public key in the key pair.
- `fingerprint` - It is the fingerprint information about the key pair.

» Data Source: opentelekomcloud_compute_bms_nic_v2

`opentelekomcloud_compute_bms_nic_v2` used to query information about a BMS NIC based on the NIC ID.

» Example Usage

```
variable "bms_id" {}
variable "nic_id" {}

data "opentelekomcloud_compute_bms_nic_v2" "Query_BMS_Nic"
{
    server_id = "${var.bms_id}",
    id = "${var.nic_id}",
```

```
}
```

» Argument Reference

The arguments of this data source act as filters for querying the BMSs details.

- `server_id` - (Required) - This is the unique BMS id.
- `id` - (Optional) - The ID of the NIC.
- `status` - (Optional) - The NIC port status.

» Attributes Reference

All of the argument attributes are also exported as result attributes.

- `mac_address` - It is NIC's mac address.
- `fixed_ips` - The NIC IP address.
- `network_id` - The ID of the network to which the NIC port belongs.

» Data Source: opentelekomcloud_compute_bms_server_v2

`opentelekomcloud_compute_bms_server_v2` used to query a BMS or BMSs details.

» Example Usage

```
variable "bms_id" {}
variable "bms_name" {}

data "opentelekomcloud_compute_bms_server_v2" "Query_BMS"
{
    id = "${var.bms_id}",
    name = "${var.bms_name}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the BMSs details.

- `id` - (Optional) - The unique ID of the BMS.
- `user_id` (Optional) - The ID of the user to which the BMS belongs.

- `name` (Optional) - The name of BMS.
- `status` (Optional) - The BMS status.
- `host_status` (Optional) - The nova-compute status: **UP, UNKNOWN, DOWN, MAINTENANCE** and **Null**.
- `key_name` (Optional) - It is the SSH key name.
- `flavor_id` (Optional) - It gives the BMS flavor information.
- `image_id` (Optional) - The BMS image.

» Attributes Reference

All of the argument attributes are also exported as result attributes.

- `host_id` - It is the host ID of the BMS.
- `progress` - This is a reserved attribute.
- `metadata` - The BMS metadata is specified.
- `access_ip_v4` - This is a reserved attribute.
- `access_ip_v6` - This is a reserved attribute.
- `addresses` - It gives the BMS network address.
- `security_groups` - The list of security groups to which the BMS belongs.
- `tags` - Specifies the BMS tag.
- `locked` - It specifies whether a BMS is locked, true: The BMS is locked, false: The BMS is not locked.
- `config_drive` - This is a reserved attribute.
- `availability_zone` - Specifies the AZ ID.
- `description` - Provides supplementary information about the pool.
- `kernel_id` - The UUID of the kernel image when the AMI image is used.
- `hypervisor_hostname` - It is the name of a host on the hypervisor.
- `instance_name` - Instance name is specified.

» Data Source: opentelekomcloud_csbs_backup_v1

The OpenTelekomCloud CSBS Backup data source allows access of backup resources.

» Example Usage

```
variable "backup_name" { }

data "opentelekomcloud_csbs_backup_v1" "csbs" {
  backup_name = "${var.backup_name}"
}
```

» Argument Reference

The following arguments are supported:

- `id` - (Optional) Specifies the ID of backup.
- `backup_name` - (Optional) Specifies the backup name.
- `status` - (Optional) Specifies the backup status.
- `resource_name` - (Optional) Specifies the backup object name.
- `backup_record_id` - (Optional) Specifies the backup record ID.
- `resource_type` - (Optional) Specifies the type of backup objects.
- `resource_id` - (Optional) Specifies the backup object ID.
- `policy_id` - (Optional) Specifies the Policy Id.
- `vm_ip` - (Optional) Specifies the ip of VM.

» Attributes Reference

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

- `description` - Provides the backup description.
- `auto_trigger` - Specifies whether automatic trigger is enabled.
- `average_speed` - Specifies average speed.
- `size` - Specifies the backup capacity.
- volume_backups
  - `space_saving_ratio` - Specifies the space saving rate.
  - `volume_backups` block supports the following arguments:
  - `status` - Status of backup Volume.
  - `space_saving_ratio` - Specifies space saving rate.
  - `name` - It gives EVS disk backup name.
  - `bootable` - Specifies whether the disk is bootable.
  - `average_speed` - Specifies the average speed.

  - `source_volume_size` - Shows source volume size in GB.
  - `source_volume_id` - It specifies source volume ID.
  - `incremental` - Shows whether incremental backup is used.
  - `snapshot_id` - ID of snapshot.
  - `source_volume_name` - Specifies source volume name.
  - `image_type` - It specifies backup. The default value is backup.
  - `id` - Specifies Cinder backup ID.
  - `size` - Specifies accumulated size (MB) of backups.

- `vm_metadata` block supports the following arguments:
  - `name` - Name of backup data.
  - `eip` - Specifies elastic IP address of the ECS.
  - `cloud_service_type` - Specifies ECS type.
  - `ram` - Specifies memory size of the ECS, in MB.
  - `vcpus` - Specifies CPU cores corresponding to the ECS.
  - `private_ip` - It specifies internal IP address of the ECS.
  - `disk` - Shows system disk size corresponding to the ECS specifications.
  - `image_type` - Specifies image type.

- `tags` block supports the following arguments:
  - `key` - Specifies tag key.
  - `value` - Specifies tag value.

» Data Source: opentelekomcloud_csbs_backup_policy_v1

The OpenTelekomCloud CSBS Backup Policy data source allows access of backup Policy resources.

» Example Usage

```
variable "policy_id" { }

data "opentelekomcloud_csbs_backup_policy_v1" "csbs_policy" {
  id = "${var.policy_id}"
}
```

» Argument Reference

The following arguments are supported:

- `id` - (Optional) Specifies the ID of backup policy.

- `name` - (Optional) Specifies the backup policy name.
- `status` - (Optional) Specifies the backup policy status.

» Attributes Reference

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

- `description` - Specifies the backup policy description.
- `provider_id` - Provides the Backup provider ID.
- `parameters` - Specifies the parameters of a backup policy.
- `scheduled_operation` block supports the following arguments:
  - `name` - Specifies Scheduling period name.
  - `description` - Specifies Scheduling period description.
  - `enabled` - Specifies whether the scheduling period is enabled.
  - `max_backups` - Specifies maximum number of backups that can be automatically created for a backup object.
  - `retention_duration_days` - Specifies duration of retaining a backup, in days.
  - `permanent` - Specifies whether backups are permanently retained.
  - `trigger_pattern` - Specifies Scheduling policy of the scheduler.
  - `operation_type` - Specifies Operation type, which can be backup.
  - `id` - Specifies Scheduling period ID.
  - `trigger_id` - Specifies Scheduler ID.
  - `trigger_name` - Specifies Scheduler name.
  - `trigger_type` - Specifies Scheduler type.
- `resource` block supports the following arguments:
  - `id` - Specifies the ID of the object to be backed up.
  - `type` - Entity object type of the backup object.
  - `name` - Specifies backup object name.
- `tags` block supports the following arguments:
  - `key` - Tag key. It cannot be an empty string.
  - `value` - Tag value. It can be an empty string.

» Data Source: opentelekomcloud_deh_server_v1

`opentelekomcloud_deh_server_v1` used to query server on a specified Dedicated Host.

» Example Usage

```
variable "deh_id" { }

variable "server_id" { }

data "opentelekomcloud_deh_server_v1" "deh_server"
{
    id = "${var.deh_id}",
    server_id = "${var.server id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the server on specified dedicated host.

- `dedicated_host_id` - (Optional) -The Dedicated Host ID.
- `server_id` (Optional) - The Server ID.

» Attributes Reference

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

- `user_id` - The ID of the user to which the server belongs.
- `name` - The server name.
- `flavor` - The ID of server specifications.
- `metadata` - The metadata of the server.
- `status` - The status of the server.
- `tenant_id` - The ID of the tenant to which the server belongs.
- `addresses` - The network addresses of the server.

» Data Source: opentelekomcloud_deh_host_v1

`opentelekomcloud_deh_host_v1` used to query allocated dedicated hosts.

» Example Usage

```
variable "deh_id" { }

data "opentelekomcloud_deh_host_v1" "deh_host"
{
    id = "${var.deh_id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the allocated dedicated host.

- `id` - (Optional) - The Dedicated Host ID.
- `name` (Optional) - The Dedicated Host name.

» Attributes Reference

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

- `host_type` - The Dedicated Host type.
- `host_type_name` - The Dedicated Host name of type.
- `status` - The Dedicated Host status.
- `availability_zone` - The Availability Zone to which the Dedicated Host belongs.
- `tenant_id` - The UUID of the tenant in a multi-tenancy cloud.
- `auto_placement` - Allows a instance to be automatically placed onto the available Dedicated Hosts.
- `available_vcpus` - Thenumber of available vCPUs for the Dedicated Host.
- `available_memory` - The size of available memory for the Dedicated Host.
- `sockets` - The number of host physical sockets.
- `instance_total` - The number of the placed VMs.
- `memory` - The size of host physical memory (MB).
- `vcpus` - The number of host vCPUs.
- `available_instance_capacities` - The VM flavors placed on the Dedicated Host.

- `cores` - The number of hosts physical cores.
- `instance_uuids` -The VMs started on the Dedicated Host.

» Data Source: opentelekomcloud_vbs_backup_policy_v2

The VBS Backup Policy data source provides details about a specific VBS backup policy.

» Example Usage

```
 variable "policy_name" { }

 variable "policy_id" { }

data "opentelekomcloud_vbs_backup_policy_v2" "policies" {
  name = "${var.policy_name}"
  id = "${var.policy_id}"
}
```

» Argument Reference

The arguments of this data source act as filters for querying the available VBS backup policy. The given filters must match exactly one VBS backup policy whose data will be exported as attributes.

- `id` (Optional) - The ID of the specific VBS backup policy to retrieve.
- `name` (Optional) - The name of the specific VBS backup policy to retrieve.
- `status` (Optional) - The status of the specific VBS backup policy to retrieve. The values can be ON or OFF

filter_tags - (Optional) Represents the list of tags. Backup policy with these tags will be filtered.

- `key` - (Required) Specifies the tag key. Tag keys must be unique.
- `values` - (Required) Specifies the List of tag values. This list can have a maximum of 10 values and all be unique.

» Attributes Reference

The following attributes are exported:

- `id` - See Argument Reference above.
- `name` - See Argument Reference above.
- `status` - See Argument Reference above.
- `start_time` - Specifies the start time of the backup job.The value is in the HH:mm format.
- `retain_first_backup` - Specifies whether to retain the first backup in the current month.
- `rentention_num` - Specifies number of retained backups.
- `frequency` - Specifies the backup interval. The value is in the range of 1 to 14 days.
- `policy_resource_count` - Specifies the number of volumes associated with the backup policy.

tags - Represents the list of tag details associated with the backup policy.

- `key` - Specifies the tag key.
- `value` - Specifies the tag value.

» Data Source: opentelekomcloud_vbs_backup_v2

The VBS Backup data source provides details about a specific VBS Backup.

» Example Usage

```
variable "backup_id" {}

data "opentelekomcloud_vbs_backup_v2" "mybackup" {
  id = "${var.backup_id}"
}
```

» Argument Reference

The following arguments are supported:

- `id` - (Optional) The id of the vbs backup.
- `name` - (Optional) The name of the vbs backup.
- `volume_id` - (Optional) The source volume ID of the backup.
- `snapshot_id` - (Optional) ID of the snapshot associated with the backup.

- `status` - (Optional) The status of the VBS backup.

» Attributes Reference

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

- `description` - The description of the vbs backup.
- `availability_zone` - The AZ where the backup resides.
- `size` - The size of the vbs backup.
- `container` - The container of the backup.
- `service_metadata` - The metadata of the vbs backup.
- `to_project_ids` - IDs of projects with which the backup is shared.
- `share_ids` - The backup share IDs.

» opentelekomcloud_ces_alarmrule

Manages a V2 topic resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_ces_alarmrule" "alarm_rule" {
  "alarm_name" = "alarm_rule"
  "metric" {
    "namespace" = "SYS.ECS"
    "metric_name" = "network_outgoing_bytes_rate_inband"
    "dimensions" {
        "name" = "instance_id"
        "value" = "${opentelekomcloud_compute_instance_v2.webserver.id}"
    }
  }
  "condition"  {
    "period" = 300
    "filter" = "average"
    "comparison_operator" = ">"
    "value" = 6
    "unit" = "B/s"
    "count" = 1
  }
  "alarm_actions" {
    "type" = "notification"
```

```
    "notification_list" = [
      "${opentelekomcloud_smn_topic_v2.topic.id}"
    ]
  }
}
```

» Argument Reference

The following arguments are supported:

- `alarm_name` - (Required) Specifies the name of an alarm rule. The value can be a string of 1 to 128 characters that can consist of numbers, lowercase letters, uppercase letters, underscores (_), or hyphens (-).
- `alarm_description` - (Optional) The value can be a string of 0 to 256 characters.
- `metric` - (Required) Specifies the alarm metrics. The structure is described below.
- `condition` - (Required) Specifies the alarm triggering condition. The structure is described below.
- `alarm_actions` - (Optional) Specifies the action triggered by an alarm. The structure is described below.
- `insufficientdata_actions` - (Optional) Specifies the action triggered by data insufficiency. The structure is described below.
- `ok_actions` - (Optional) Specifies the action triggered by the clearing of an alarm. The structure is described below.
- `alarm_enabled` - (Optional) Specifies whether to enable the alarm. The default value is true.
- `alarm_action_enabled` - (Optional) Specifies whether to enable the action to be triggered by an alarm. The default value is true. Note: If alarm_action_enabled is set to true, at least one of the following parameters alarm_actions, insufficientdata_actions, and ok_actions cannot be empty. If alarm_actions, insufficientdata_actions, and ok_actions coexist, their corresponding notification_list must be of the same value.

The `metric` block supports:

- `namespace` - (Required) Specifies the namespace in service.item format. service.item can be a string of 3 to 32 characters that must start with a letter and can consists of uppercase letters, lowercase letters, numbers, or underscores (_).

- `metric_name` - (Required) Specifies the metric name. The value can be a string of 1 to 64 characters that must start with a letter and can consists of uppercase letters, lowercase letters, numbers, or underscores (_).
- `dimensions` - (Required) Specifies the list of metric dimensions. Currently, the maximum length of the dimesion list that are supported is 3. The structure is described below.

The `dimensions` block supports:

- `name` - (Required) Specifies the dimension name. The value can be a string of 1 to 32 characters that must start with a letter and can consists of uppercase letters, lowercase letters, numbers, underscores (_), or hyphens (-).
- `value` - (Required) Specifies the dimension value. The value can be a string of 1 to 64 characters that must start with a letter or a number and can consists of uppercase letters, lowercase letters, numbers, underscores (_), or hyphens (-).

The `condition` block supports:

- `period` - (Required) Specifies the alarm checking period in seconds. The value can be 1, 300, 1200, 3600, 14400, and 86400. Note: If period is set to 1, the raw metric data is used to determine whether to generate an alarm.
- `filter` - (Required) Specifies the data rollup methods. The value can be max, min, average, sum, and vaiance.
- `comparison_operator` - (Required) Specifies the comparison condition of alarm thresholds. The value can be >, =, <, >=, or <=.
- `value` - (Required) Specifies the alarm threshold. The value ranges from 0 to Number of 1.7976931348623157e+308.
- `unit` - (Optional) Specifies the data unit.
- `count` - (Required) Specifies the number of consecutive occurrence times. The value ranges from 1 to 5.

the `alarm_actions` block supports:

- `type` - (Optional) specifies the type of action triggered by an alarm. the value can be notification or autoscaling. notification: indicates that a notification will be sent to the user. autoscaling: indicates that a scaling action will be triggered.
- `notification_list` - (Optional) specifies the topic urn list of the target notification objects. the maximum length is 5. the topic urn list can be obtained from simple message notification (smn) and in the following format: urn: smn:([a-z]|[a-z]|[0-9]|-){1,32}:([a-z]|[a-z]|[0-9]){32}:([a-z]|[a-z]|[0-9]|-|_){1,256}. if type is set to notification, the value of notification_list cannot be empty. if type is set to autoscaling, the value of notification_list

must be [] and the value of namespace must be sys.as. Note: to enable the as alarm rules take effect, you must bind scaling policies. for details, see the auto scaling api reference.

the `insufficientdata_actions` block supports:

- `type` - (Optional) specifies the type of action triggered by an alarm. the value is notification. notification: indicates that a notification will be sent to the user.
- `notification_list` - (Optional) indicates the list of objects to be notified if the alarm status changes. the maximum length is 5.

the `ok_actions` block supports:

- `type` - (Optional) specifies the type of action triggered by an alarm. the value is notification. notification: indicates that a notification will be sent to the user.
- `notification_list` - (Optional) indicates the list of objects to be notified if the alarm status changes. the maximum length is 5.

» Attributes Reference

The following attributes are exported:

- `alarm_name` - See Argument Reference above.
- `alarm_description` - See Argument Reference above.
- `metric` - See Argument Reference above.
- `condition` - See Argument Reference above.
- `alarm_actions` - See Argument Reference above.
- `insufficientdata_actions` - See Argument Reference above.
- `ok_actions` - See Argument Reference above.
- `alarm_enabled` - See Argument Reference above.
- `alarm_action_enabled` - See Argument Reference above.
- `id` - Specifies the alarm rule ID.
- `update_time` - Specifies the time when the alarm status changed. The value is a UNIX timestamp and the unit is ms.
- `alarm_state` - Specifies the alarm status. The value can be: ok: The alarm status is normal, alarm: An alarm is generated, insufficient_data: The required data is insufficient.

» opentelekomcloud_blockstorage_volume_v2

Manages a V2 volume resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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.
- `tags` - (Optional) Tags key/value pairs to associate with the volume. Changing this updates the existing volume tags.
- `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. Changing this creates a new volume.

- `cascade` - (Optional, Default:false) Specifies to delete all snapshots associated with the EVS disk.

» Attributes Reference

The following attributes are exported:

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

» Import

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

```
$ terraform import opentelekomcloud_blockstorage_volume_v2.volume_1 ea257959-eeb1-4c10-8d33-
```

» opentelekomcloud_compute_floatingip_v2

Manages a V2 floating IP resource within OpenTelekomCloud Nova (compute) that can be used for compute instances. These are similar to Neutron (networking) floating IP resources, but only networking floating IPs can be used with load balancers.

» Example Usage

```
resource "opentelekomcloud_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 opentelekomcloud_compute_floatingip_v2.floatip_1 89c60255-9bd6-460c-822a-
```

» opentelekomcloud_compute_floatingip_associate_v2

Associate a floating IP to an instance. This can be used instead of the `floating_ip` options in `opentelekomcloud_compute_instance_v2`.

» Example Usage

» Automatically detect the correct network

```
resource "opentelekomcloud_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 "opentelekomcloud_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "opentelekomcloud_compute_floatingip_associate_v2" "fip_1" {
  floating_ip = "${opentelekomcloud_networking_floatingip_v2.fip_1.address}"
  instance_id = "${opentelekomcloud_compute_instance_v2.instance_1.id}"
}
```

» Explicitly set the network to attach to

```
resource "opentelekomcloud_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 "opentelekomcloud_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "opentelekomcloud_compute_floatingip_associate_v2" "fip_1" {
  floating_ip = "${opentelekomcloud_networking_floatingip_v2.fip_1.address}"
  instance_id = "${opentelekomcloud_compute_instance_v2.instance_1.id}"
  fixed_ip    = "${opentelekomcloud_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 opentelekomcloud_compute_floatingip_associate_v2.fip_1 <floating_ip>/<ins
```

» opentelekomcloud_compute_instance_v2

Manages a V2 VM instance resource within OpenTelekomCloud.

» Example Usage

» Basic Instance

```
resource "opentelekomcloud_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"]
  tags = ["foo.bar", "tag.value"]

  metadata {
```

```
    this = "that"
  }

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

» Instance With Attached Volume

```
resource "opentelekomcloud_blockstorage_volume_v2" "myvol" {
  name = "myvol"
  size = 1
}

resource "opentelekomcloud_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"]

  network {
    name = "my_network"
  }
}

resource "opentelekomcloud_compute_volume_attach_v2" "attached" {
  instance_id = "${opentelekomcloud_compute_instance_v2.myinstance.id}"
  volume_id = "${opentelekomcloud_blockstorage_volume_v2.myvol.id}"
}
```

» Boot From Volume

```
resource "opentelekomcloud_compute_instance_v2" "boot-from-volume" {
  name            = "boot-from-volume"
  flavor_id       = "3"
  key_pair        = "my_key_pair_name"
  security_groups = ["default"]

  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 "opentelekomcloud_blockstorage_volume_v1" "myvol" {
  name     = "myvol"
  size     = 5
  image_id = "<image-id>"
}

resource "opentelekomcloud_compute_instance_v2" "boot-from-volume" {
  name            = "bootfromvolume"
  flavor_id       = "3"
  key_pair        = "my_key_pair_name"
  security_groups = ["default"]

  block_device {
    uuid                  = "${opentelekomcloud_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 "opentelekomcloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  image_id        = "<image-id>"
  flavor_id       = "3"
```

```
  key_pair        = "my_key_pair_name"
  security_groups = ["default"]

  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 "opentelekomcloud_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}

resource "opentelekomcloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  image_id        = "<image-id>"
  flavor_id       = "3"
  key_pair        = "my_key_pair_name"
  security_groups = ["default"]

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

  block_device {
    uuid                  = "${opentelekomcloud_blockstorage_volume_v2.volume_1.id}"
    source_type           = "volume"
```

```
    destination_type      = "volume"
    boot_index            = 1
    delete_on_termination = true
  }
}
```

» Instance With Multiple Networks

```
resource "opentelekomcloud_networking_floatingip_v2" "myip" {
  pool = "my_pool"
}

resource "opentelekomcloud_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"]

  network {
    name = "my_first_network"
  }

  network {
    name = "my_second_network"
  }
}

resource "opentelekomcloud_compute_floatingip_associate_v2" "myip" {
  floating_ip = "${opentelekomcloud_networking_floatingip_v2.myip.address}"
  instance_id = "${opentelekomcloud_compute_instance_v2.multi-net.id}"
  fixed_ip = "${opentelekomcloud_compute_instance_v2.multi-net.network.1.fixed_ip_v4}"
}
```

» Instance With Personality

```
resource "opentelekomcloud_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"]

  personality {
```

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

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

» Instance with Multiple Ephemeral Disks

```
resource "opentelekomcloud_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"]

  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 "opentelekomcloud_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"]
  user_data       = "#cloud-config\nhostname: instance_1.example.com\nfqdn: instance_1.examp

  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` - (Optional) The availability zone in which to create the server. Changing this creates a new server.
- `network` - (Optional) An array of one or more networks to attach to the instance. Required when there are multiple networks defined for the tenant. 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.
- `tags` - (Optional) The tags of the image. It must be a list of strings.
- `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.
- `force_delete` - (Optional) Whether to force the OpenTelekomCloud instance to be forcefully deleted. This is useful for environments that have reclaim / soft deletion enabled.
- `auto_recovery` - (Optional) Configures or deletes automatic recovery of an instance

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.
  - `tenancy` - (Optional) The tenancy specifies whether the ECS is to be created on a Dedicated Host (DeH) or in a shared pool.
  - `deh_id` - (Optional) The ID of DeH. This parameter takes effect only when the value of tenancy is dedicated.

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.
- `auto_recovery` - See Argument Reference above.

» 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 "opentelekomcloud_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 OpenTelekomCloud 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 "opentelekomcloud_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 "opentelekomcloud_compute_instance_v2" "instance_1" {
  name = "instance_1"

  network {
    port = "${opentelekomcloud_networking_port_v2.port_1.id}"
  }

  connection {
    user        = "root"
    host        = "${opentelekomcloud_networking_port_v2.port_1.fixed_ip.0.ip_address}"
    private_key = "~/path/to/key"
  }

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

» opentelekomcloud_compute_keypair_v2

Manages a V2 keypair resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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 opentelekomcloud_compute_keypair_v2.my-keypair test-keypair
```

» opentelekomcloud_compute_secgroup_v2

Manages a V2 security group resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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 "opentelekomcloud_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 = ["${opentelekomcloud_compute_secgroup_v2.secgroup_1.name}"]
}
```

» Import

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

```
$ terraform import opentelekomcloud_compute_secgroup_v2.my_secgroup 1bc30ee9-9d5b-4c30-bdd5-
```

» opentelekomcloud_compute_servergroup_v2

Manages a V2 Server Group resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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 opentelekomcloud_compute_servergroup_v2.test-sg 1bc30ee9-9d5b-4c30-bdd5-7
```

» opentelekomcloud_compute_volume_attach_v2

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

» Example Usage

» Basic attachment of a single volume to a single instance

```
resource "opentelekomcloud_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
```

```
}

resource "opentelekomcloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "opentelekomcloud_compute_volume_attach_v2" "va_1" {
  instance_id = "${opentelekomcloud_compute_instance_v2.instance_1.id}"
  volume_id   = "${opentelekomcloud_blockstorage_volume_v2.volume_1.id}"
}
```

» Attaching multiple volumes to a single instance

```
resource "opentelekomcloud_blockstorage_volume_v2" "volumes" {
  count = 2
  name  = "${format("vol-%02d", count.index + 1)}"
  size  = 1
}

resource "opentelekomcloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "opentelekomcloud_compute_volume_attach_v2" "attachments" {
  count       = 2
  instance_id = "${opentelekomcloud_compute_instance_v2.instance_1.id}"
  volume_id   = "${element(opentelekomcloud_blockstorage_volume_v2.volumes.*.id, count.index
}

output "volume devices" {
  value = "${opentelekomcloud_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 opentelekomcloud_compute_volume_attach_v2.va_1 89c60255-9bd6-460c-822a-e2
```

» opentelekomcloud_dns_recordset_v2

Manages a DNS record set in the OpenTelekomCloud DNS Service.

» Example Usage

» Automatically detect the correct network

```
resource "opentelekomcloud_dns_zone_v2" "example_zone" {
  name = "example.com."
  email = "email2@example.com"
  description = "a zone"
  ttl = 6000
```

```
  type = "PRIMARY"
}

resource "opentelekomcloud_dns_recordset_v2" "rs_example_com" {
  zone_id = "${opentelekomcloud_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. Examples: "A", "MX". Changing this creates a new DNS record set.
- `ttl` - (Optional) The time to live (TTL) of the record set.
- `description` - (Optional) A description of the record set.
- `records` - (Optional) 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 opentelekomcloud_dns_recordset_v2.recordset_1 <zone_id>/<recordset_id>
```

» opentelekomcloud_dns_zone_v2

Manages a DNS zone in the OpenTelekomCloud DNS Service.

» Example Usage

» Public Zone Configuration

```
resource "opentelekomcloud_dns_zone_v2" "public_example_com" {
  name = "public.example.com."
  email = "public@example.com"
  description = "An example for public zone"
  ttl = 3000
  type = "public"
}
```

» Private Zone Configuration

```
resource "opentelekomcloud_dns_zone_v2" "private_example_com" {
  name = "private.example.com."
  email = "private@example.com"
  description = "An example for private zone"
  ttl = 3000
  type = "private"
  router = {
     router_id = "${var.vpc_id}"
     router_region = "${var.region}"
  }
}
```

» 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.
- `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.
- `type` - (Optional) The type of zone. Can either be `public` or `private`. Changing this creates a new zone.
- `ttl` - (Optional) The time to live (TTL) of the zone.
- `description` - (Optional) A description of the zone.
- `router` (Optional) The Router(VPC) configuration for the private zone. it is required when type is `private`. Changing this creates a new zone.
- `masters` - (Optional) An array of master DNS servers.
- `value_specs` - (Optional) Map of additional options. Changing this creates a new zone.

The `router` block supports:

- `router_id` (Required) The Router(VPC) ID. which VPC network will assicate with.
- `router_region` (Required) The Region name for this private zone.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `email` - See Argument Reference above.
- `type` - See Argument Reference above.
- `ttl` - See Argument Reference above.
- `description` - See Argument Reference above.
- `masters` - See Argument Reference above.
- `value_specs` - See Argument Reference above.

» Import

This resource can be imported by specifying the zone ID:

```
$ terraform import opentelekomcloud_dns_zone_v2.zone_1 <zone_id>
```

» opentelekomcloud_elb_loadbalancer

Manages an elastic loadbalancer resource within OpentelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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:

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

- `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. Changing this creates a new elb loadbalancer.
- `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. Changing this creates a new elb loadbalancer.
- `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. Changing this creates a new elb loadbalancer.
- `vip_address` - (Optional) Specifies the IP address provided by ELB. When type is 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, parameter bandwidth is invalid. Changing this creates a new elb loadbalancer.
- `tenantid` - (Optional) Specifies the tenant ID. This parameter is mandatory only when type is set to Internal.

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `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.
- `security_group_id` - See Argument Reference above.
- `vip_address` - See Argument Reference above.
- `tenantid` - See Argument Reference above.

- `id` - Specifies the load balancer ID.

» opentelekomcloud_elb_listener

Manages an elastic loadbalancer listener resource within OpentelekomCloud.

» Example Usage

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

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

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the elb listener. If omitted, the `region` argument of the provider is used. Changing this creates a new elb listener.
- `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. Changing this creates a new elb listener.
- `protocol` - (Required) Specifies the listening protocol used for layer 4 or 7. The value can be HTTP, TCP, HTTPS, or UDP. Changing this creates a new elb listener.
- `protocol_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. Changing this creates a new elb listener.
- `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. Changing this creates a new elb listener.
- `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. Changing this creates a new elb listener.
- `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. Changing this creates a new elb listener.
- `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. Changing this creates a new elb listener.
- `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. Changing this creates a new elb listener.
- `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:

- `region` - See Argument Reference above.
- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `loadbalancer_id` - See Argument Reference above.
- `protocol` - See Argument Reference above.
- `protocol_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.
- `id` - Specifies the listener ID.
- `admin_state_up` - Specifies the status of the load balancer. Value range: false: The load balancer is disabled. true: The load balancer runs properly.

» opentelekomcloud_elb_health

Manages an elastic loadbalancer health resource within OpentelekomCloud.

» Example Usage

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

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

resource "opentelekomcloud_elb_health" "healthcheck" {
  listener_id = "${opentelekomcloud_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:

- `region` - (Optional) The region in which to create the elb health. If omitted, the `region` argument of the provider is used. Changing this creates a new elb health.
- `listener_id` - (Required) Specifies the ID of the listener to which the health check task belongs. Changing this creates a new elb health.
- `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:

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

» opentelekomcloud_elb_backend

Manages an elastic loadbalancer backend resource within OpentelekomCloud.

» Example Usage

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

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

```
}

resource "opentelekomcloud_elb_backend" "backend" {
  address = "192.168.0.211"
  listener_id = "${opentelekomcloud_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. Changing this creates a new elb backend.
- `server_id` - (Required) Specifies the backend member ID. Changing this creates a new elb backend.
- `address` - (Required) Specifies the private IP address of the backend member. Changing this creates a new elb backend.

» Attributes Reference

The following attributes are exported:

- `listener_id` - See Argument Reference above.
- `server_id` - See Argument Reference above.
- `address` - See Argument Reference above.
- `server_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.

» opentelekomcloud_images_image_v2

Manages a V2 Image resource within OpenTelekomCloud Glance.

» Example Usage

```
resource "opentelekomcloud_images_image_v2" "rancheros" {
  name   = "RancherOS"
  image_source_url = "https://releases.rancher.com/os/latest/rancheros-openstack.img"
  container_format = "bare"
  disk_format = "qcow2"
  tags = ["foo.bar", "tag.value"]
}
```

» Argument Reference

The following arguments are supported:

- `container_format` - (Required) The container format. Must be one of ”ami”, ”ari”, ”aki”, ”bare”, ”ovf”.
- `disk_format` - (Required) The disk format. Must be one of ”ami”, ”ari”, ”aki”, ”vhd”, ”vmdk”, ”raw”, ”qcow2”, ”vdi”, ”iso”.
- `local_file_path` - (Optional) This is the filepath of the raw image file that will be uploaded to Glance. Conflicts with `image_source_url`.
- `image_cache_path` - (Optional) This is the directory where the images will be downloaded. Images will be stored with a filename corresponding to the url’s md5 hash. Defaults to ”$HOME/.terraform/image_cache”
- `image_source_url` - (Optional) This is the url of the raw image that will be downloaded in the `image_cache_path` before being uploaded to Glance. Glance is able to download image from internet but the `gophercloud` library does not yet provide a way to do so. Conflicts with `local_file_path`.
- `min_disk_gb` - (Optional) Amount of disk space (in GB) required to boot image. Defaults to 0.
- `min_ram_mb` - (Optional) Amount of ram (in MB) required to boot image. Defauts to 0.
- `name` - (Required) The name of the image.
- `protected` - (Optional) If true, image will not be deletable. Defaults to false.
- `region` - (Optional) The region in which to obtain the V2 Glance client. A Glance client is needed to create an Image that can be used with a compute instance. If omitted, the `region` argument of the provider is used. Changing this creates a new Image.

- `tags` - (Optional) The tags of the image. It must be a list of strings. At this time, it is not possible to delete all tags of an image.
- `visibility` - (Optional) The visibility of the image. Must be one of "public", "private", "community", or "shared". The ability to set the visibility depends upon the configuration of the OpenTelekomCloud cloud.

Note: The `properties` attribute handling in the gophercloud library is currently buggy and needs to be fixed before being implemented in this resource.

» Attributes Reference

The following attributes are exported:

- `checksum` - The checksum of the data associated with the image.
- `container_format` - See Argument Reference above.
- `created_at` - The date the image was created.
- `disk_format` - See Argument Reference above.
- `file` - the trailing path after the glance endpoint that represent the location of the image or the path to retrieve it.
- `id` - A unique ID assigned by Glance.
- `metadata` - The metadata associated with the image. Image metadata allow for meaningfully define the image properties and tags. See http://docs.openstack.org/developer/glance/metadefs-concepts.html.
- `min_disk_gb` - See Argument Reference above.
- `min_ram_mb` - See Argument Reference above.
- `name` - See Argument Reference above.
- `owner` - The id of the opentelekomcloud user who owns the image.
- `protected` - See Argument Reference above.
- `region` - See Argument Reference above.
- `schema` - The path to the JSON-schema that represent the image or image
- `size_bytes` - The size in bytes of the data associated with the image.
- `status` - The status of the image. It can be "queued", "active" or "saving".
- `tags` - See Argument Reference above.
- `update_at` - The date the image was last updated.
- `visibility` - See Argument Reference above.

» Import

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

```
$ terraform import opentelekomcloud_images_image_v2.rancheros 89c60255-9bd6-460c-822a-e2b959
```

» opentelekomcloud_mrs_cluster_v1

Manages resource cluster within OpenTelekomCloud MRS.

» Example Usage: Creating a MRS cluster

```
resource "opentelekomcloud_mrs_cluster_v1" "cluster1" {
  cluster_name = "mrs-cluster"
  region = "eu-de"
  billing_type = 12
  master_node_num = 2
  core_node_num = 3
  master_node_size = "c2.4xlarge.linux"
  core_node_size = "s1.xlarge.linux"
  available_zone_id = "sa-chile-1a"
  vpc_id = "51edfb75-f9f0-4bbc-b4dc-21466b93f60d"
  subnet_id = "1d7a8646-43ee-455a-a3ab-40da87a1304c"
  cluster_version = "MRS 1.5.0"
  volume_type = "SATA"
  volume_size = 100
  safe_mode = 0
  cluster_type = 0
  node_public_cert_name = "KeyPair-ci"
  cluster_admin_secret = ""
  component_list {
      component_name = "Hadoop"
  }
  component_list {
      component_name = "Spark"
  }
  component_list {
      component_name = "Hive"
  }
}
```

» Argument Reference

The following arguments are supported:

- `billing_type` - (Required) The value is 12, indicating on-demand payment.
- `region` - (Required) Cluster region information. Obtain the value from Regions and Endpoints.

- `master_node_num` - (Required) Number of Master nodes The value is 2.
- `master_node_size` - (Required) Best match based on several years of commissioning experience. MRS supports specifications of hosts, and host specifications are determined by CPUs, memory, and disks space. Master nodes support h1.2xlarge.linux.mrs h1.4xlarge.linux.mrs, h1.8xlarge.linux.mrs, s1.4xlarge.linux.mrs, and s1.8xlarge.linux.mrs. Core nodes of a streaming cluster support all specifications c2.2xlarge.linux.mrs, c2.4xlarge.linux.mrs, s1.xlarge.linux.mrs, s1.4xlarge.linux.mrs, s1.8xlarge.linux.mrs, d1.xlarge.linux.mrs, d1.2xlarge.linux.mrs, d1.4xlarge.linux.mrs, d1.8xlarge.linux.mrs, h1.2xlarge.linux.mrs, h1.4xlarge.linux.mrs and h1.8xlarge.linux.mrs. Task nodes support c2.2xlarge.linux.mrs, c2.4xlarge.linux.mrs, s1.xlarge.linux.mrs, s1.4xlarge.linux.mrs, s1.8xlarge.linux.mrs, h1.2xlarge.linux.mrs, h1.4xlarge.linux.mrs, and h1.8xlarge.linux.mrs.
- `core_node_num` - (Required) Number of Core nodes Value range: 1 to 500 A maximum of 500 Core nodes are supported by default. If more than 500 Core nodes are required, contact technical support engineers or invoke background APIs to modify the database.
- `core_node_size` - (Required) Instance specification of a Core node Configuration method of this parameter is identical to that of master_node_size.
- `available_zone_id` - (Required) ID of an available zone. Obtain the value from Regions and Endpoints.
- `cluster_name` - (Required) Cluster name, which is globally unique and contains only 1 to 64 letters, digits, hyphens (-), and underscores (_).
- `vpc_id` - (Required) ID of the VPC where the subnet locates Obtain the VPC ID from the management console as follows: Register an account and log in to the management console. Click Virtual Private Cloud and select Virtual Private Cloud from the left list. On the Virtual Private Cloud page, obtain the VPC ID from the list.
- `subnet_id` - (Required) Subnet ID Obtain the subnet ID from the management console as follows: Register an account and log in to the management console. Click Virtual Private Cloud and select Virtual Private Cloud from the left list. On the Virtual Private Cloud page, obtain the subnet ID from the list.
- `cluster_version` - (Optional) Version of the clusters Currently, MRS 1.2, MRS 1.3.0, MRS 1.5.0, MRS 1.6.0 and MRS 1.6.3 are supported. The latest version of MRS is used by default. Currently, the latest version is MRS 1.6.3.
- `cluster_type` - (Optional) Type of clusters 0: analysis cluster 1: streaming cluster The default value is 0.

- `volume_type` - (Required) Type of disks SATA and SSD are supported. SATA: common I/O SSD: super high-speed I/O
- `volume_size` - (Required) Data disk storage space of a Core node Users can add disks to expand storage capacity when creating a cluster. There are the following scenarios: Separation of data storage and computing: Data is stored in the OBS system. Costs of clusters are relatively low but computing performance is poor. The clusters can be deleted at any time. It is recommended when data computing is not frequently performed. Integration of data storage and computing: Data is stored in the HDFS system. Costs of clusters are relatively high but computing performance is good. The clusters cannot be deleted in a short term. It is recommended when data computing is frequently performed. Value range: 100 GB to 32000 GB
- `node_public_cert_name` - (Required) Name of a key pair You can use a key to log in to the Master node in the cluster.
- `safe_mode` - (Required) MRS cluster running mode 0: common mode The value indicates that the Kerberos authentication is disabled. Users can use all functions provided by the cluster. 1: safe mode The value indicates that the Kerberos authentication is enabled. Common users cannot use the file management or job management functions of an MRS cluster and cannot view cluster resource usage or the job records of Hadoop and Spark. To use these functions, the users must obtain the relevant permissions from the MRS Manager administrator. The request has the cluster_admin_secret parameter only when safe_mode is set to 1.
- `cluster_admin_secret` - (Optional) Indicates the password of the MRS Manager administrator. The password for MRS 1.5.0: Must contain 6 to 32 characters. Must contain at least two types of the following: Lowercase letters Uppercase letters Digits Special characters of `~!@#$%^&*()-_=+\|[{}];:'",<.>/? Spaces Must be different from the username. Must be different from the username written in reverse order. The password for MRS 1.3.0: Must contain 8 to 64 characters. Must contain at least four types of the following: Lowercase letters Uppercase letters Digits Special characters of` ~!@#$%^&*()-_=+|[{}];:'",<.>/? Spaces Must be different from the username. Must be different from the username written in reverse order. This parameter needs to be configured only when safe_mode is set to 1.
- `log_collection` - (Optional) Indicates whether logs are collected when cluster installation fails. 0: not collected 1: collected The default value is 0. If log_collection is set to 1, OBS buckets will be created to collect the MRS logs. These buckets will be charged.
- `component_list` - (Required) Service component list.

- `add_jobs` - (Optional) You can submit a job when you create a cluster to save time and use MRS easily. Only one job can be added.

The `component_list` block supports:

- `component_name` - (Required) Component name Currently, Hadoop, Spark, HBase, Hive, Hue, Loader, Flume, Kafka and Storm are supported. Loader and Flume are not supported by MRS 1.3.0. Kafka and Storm are not supported by MRS 1.2.
- `componentId` - Component ID Component IDs supported by MRS 1.5.0 include: MRS 1.5.0_001: Hadoop MRS 1.5.0_002: Spark MRS 1.5.0_003: HBase MRS 1.5.0_004: Hive MRS 1.5.0_005: Hue MRS 1.5.0_006: Kafka MRS 1.5.0_007: Storm MRS 1.5.0_008: Loader MRS 1.5.0_009: Flume Component IDs supported by MRS 1.3.0 include: MRS 1.3.0_001: Hadoop MRS 1.3.0_002: Spark MRS 1.3.0_003: HBase MRS 1.3.0_004: Hive MRS 1.3.0_005: Hue MRS 1.3.0_006: Kafka MRS 1.3.0_007: Storm For example, the component ID of Hadoop is MRS 1.5.0_001, or MRS 1.3.0_001.
- `componentName` - Component name Currently, Hadoop, Spark, HBase, Hive, Hue, Loader, Flume, Kafka and Storm are supported. Loader and Flume are not supported by MRS 1.3.0.
- `componentVersion` - Component version MRS 1.5.0 supports the following component version: Component version of an analysis cluster: Hadoop: 2.7.2 Spark: 2.1.0 HBase: 1.0.2 Hive: 1.2.1 Hue: 3.11.0 Loader: 2.0.0 Component version of a streaming cluster: Kafka: 0.10.0.0 Storm: 1.0.2 Flume: 1.6.0 MRS 1.3.0 supports the following component version: Component version of an analysis cluster: Hadoop: 2.7.2 Spark: 1.5.1 HBase: 1.0.2 Hive: 1.2.1 Hue: 3.11.0 Component version of a streaming cluster: Kafka: 0.10.0.0 Storm: 1.0.2
- `componentDesc` - Component description

The `add_jobs` block supports: * `job_type` - (Required) Job type 1: MapReduce 2: Spark 3: Hive Script 4: HiveQL (not supported currently) 5: DistCp, importing and exporting data (not supported in this API currently). 6: Spark Script 7: Spark SQL, submitting Spark SQL statements (not supported in this API currently). NOTE: Spark and Hive jobs can be added to only clusters including Spark and Hive components.

- `job_name` - (Required) Job name It contains only 1 to 64 letters, digits, hyphens (-), and underscores (_). NOTE: Identical job names are allowed but not recommended.

- `jar_path` - (Required) Path of the .jar file or .sql file for program execution The parameter must meet the following requirements: Contains a maximum of 1023 characters, excluding special characters such as ;|&><'$. The address cannot be empty or full of spaces. Starts with / or s3a://. Spark Script must end with .sql; while MapReduce and Spark Jar must end with .jar. sql and jar are case-insensitive.

- `arguments` - (Optional) Key parameter for program execution The parameter is specified by the function of the user's program. MRS is only responsible for loading the parameter. The parameter contains a maximum of 2047 characters, excluding special characters such as ;|&>'<$, and can be empty.
- `input` - (Optional) Path for inputting data, which must start with / or s3a://. A correct OBS path is required. The parameter contains a maximum of 1023 characters, excluding special characters such as ;|&>'<$, and can be empty.
- `output` - (Optional) Path for outputting data, which must start with / or s3a://. A correct OBS path is required. If the path does not exist, the system automatically creates it. The parameter contains a maximum of 1023 characters, excluding special characters such as ;|&>'<$, and can be empty.
- `job_log` - (Optional) Path for storing job logs that record job running status. This path must start with / or s3a://. A correct OBS path is required. The parameter contains a maximum of 1023 characters, excluding special characters such as ;|&>'<$, and can be empty.
- `shutdown_cluster` - (Optional) Whether to delete the cluster after the jobs are complete true: Yes false: No
- `file_action` - (Optional) Data import and export import export
- `submit_job_once_cluster_run` - (Required) true: A job is submitted when a cluster is created. false: A job is submitted separately. The parameter is set to true in this example.
- `hql` - (Optional) HiveQL statement
- `hive_script_path` - (Optional) SQL program path This parameter is needed by Spark Script and Hive Script jobs only and must meet the following requirements: Contains a maximum of 1023 characters, excluding special characters such as ;|&><'$. The address cannot be empty or full of spaces. Starts with / or s3a://. Ends with .sql. sql is case-insensitive.

» Attributes Reference

The following attributes are exported:

- `billing_type` - See Argument Reference above.
- `data_center` - See Argument Reference above.
- `master_node_num` - See Argument Reference above.
- `master_node_size` - See Argument Reference above.
- `core_node_num` - See Argument Reference above.
- `core_node_size` - See Argument Reference above.

- `available_zone_id` - See Argument Reference above.
- `cluster_name` - See Argument Reference above.
- `vpc_id` - See Argument Reference above.
- `subnet_id` - See Argument Reference above.
- `cluster_version` - See Argument Reference above.
- `cluster_type` - See Argument Reference above.
- `volume_type` - See Argument Reference above.
- `volume_size` - See Argument Reference above.
- `node_public_cert_name` - See Argument Reference above.
- `safe_mode` - See Argument Reference above.
- `cluster_admin_secret` - See Argument Reference above.
- `log_collection` - See Argument Reference above.
- `component_list` - See Argument Reference below.
- `add_jobs` - See Argument Reference above.
- `order_id` - Order ID for creating clusters.
- `cluster_id` - Cluster ID.
- `available_zone_name` - Name of an availability zone.
- `instance_id` - Instance ID.
- `hadoop_version` - Hadoop version.
- `master_node_ip` - IP address of a Master node.
- `externalIp` - Internal IP address.
- `private_ip_first` - Primary private IP address.
- `external_ip` - External IP address.
- `slave_security_groups_id` - Standby security group ID.
- `security_groups_id` - Security group ID.
- `external_alternate_ip` - Backup external IP address.
- `master_node_spec_id` - Specification ID of a Master node.
- `core_node_spec_id` - Specification ID of a Core node.
- `master_node_product_id` - Product ID of a Master node.
- `core_node_product_id` - Product ID of a Core node.
- `duration` - Cluster subscription duration.
- `vnc` - URI address for remote login of the elastic cloud server.
- `fee` - Cluster creation fee, which is automatically calculated.
- `deployment_id` - Deployment ID of a cluster.
- `cluster_state` - Cluster status Valid values include: existing history starting running terminated failed abnormal terminating rebooting shutdown frozen scaling-out scaling-in scaling-error.
- `tenant_id` - Project ID.
- `create_at` - Cluster creation time.
- `update_at` - Cluster update time.
- `error_info` - Error information.
- `charging_start_time` - Time when charging starts.
- `remark` - Remarks of a cluster.

The component_list attributes: * `component_name` - (Required) Component name Currently, Hadoop, Spark, HBase, Hive, Hue, Loader, Flume, Kafka and

Storm are supported. Loader and Flume are not supported by MRS 1.3.0. * `component_id` - Component ID Component IDs supported by MRS 1.5.0 include: MRS 1.5.0_001: Hadoop MRS 1.5.0_002: Spark MRS 1.5.0_003: HBase MRS 1.5.0_004: Hive MRS 1.5.0_005: Hue MRS 1.5.0_006: Kafka MRS 1.5.0_007: Storm MRS 1.5.0_008: Loader MRS 1.5.0_009: Flume Component IDs supported by MRS 1.3.0 include: MRS 1.3.0_001: Hadoop MRS 1.3.0_002: Spark MRS 1.3.0_003: HBase MRS 1.3.0_004: Hive MRS 1.3.0_005: Hue MRS 1.3.0_006: Kafka MRS 1.3.0_007: Storm For example, the component ID of Hadoop is MRS 1.5.0_001, or MRS 1.3.0_001. * `component_version` - Component version MRS 1.5.0 supports the following component version: Component version of an analysis cluster: Hadoop: 2.7.2 Spark: 2.1.0 HBase: 1.0.2 Hive: 1.2.1 Hue: 3.11.0 Loader: 2.0.0 Component version of a streaming cluster: Kafka: 0.10.0.0 Storm: 1.0.2 Flume: 1.6.0 MRS 1.3.0 supports the following component version: Component version of an analysis cluster: Hadoop: 2.7.2 Spark: 1.5.1 HBase: 1.0.2 Hive: 1.2.1 Hue: 3.11.0 Component version of a streaming cluster: Kafka: 0.10.0.0 Storm: 1.0.2 * `component_desc` - Component description

» opentelekomcloud_nat_gateway_v2

Manages a V2 nat gateway resource within OpenTelekomCloud Nat

» Example Usage

```
resource "opentelekomcloud_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.

» opentelekomcloud_nat_snat_rule_v2

Manages a V2 snat rule resource within OpenTelekomCloud Nat

» Example Usage

```
resource "opentelekomcloud_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.

» opentelekomcloud_networking_floatingip_v2

Manages a V2 floating IP resource within OpenTelekomCloud 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 "opentelekomcloud_networking_floatingip_v2" "floatip_1" {
  pool = "public"
}
```

» 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 opentelekomcloud_networking_floatingip_v2.floatip_1 2c7f39f3-702b-48d1-94
```

» opentelekomcloud_networking_network_v2

Manages a V2 Neutron network resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "opentelekomcloud_networking_subnet_v2" "subnet_1" {
  name       = "subnet_1"
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"
```

```
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "opentelekomcloud_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 "opentelekomcloud_networking_port_v2" "port_1" {
  name               = "port_1"
  network_id         = "${opentelekomcloud_networking_network_v2.network_1.id}"
  admin_state_up     = "true"
  security_group_ids = ["${opentelekomcloud_compute_secgroup_v2.secgroup_1.id}"]

  fixed_ip {
    "subnet_id"  = "${opentelekomcloud_networking_subnet_v2.subnet_1.id}"
    "ip_address" = "192.168.199.10"
  }
}

resource "opentelekomcloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["${opentelekomcloud_compute_secgroup_v2.secgroup_1.name}"]

  network {
    port = "${opentelekomcloud_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.

The `segments` block supports:

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

» Attributes Reference

The following attributes are exported:

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

» Import

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

```
$ terraform import opentelekomcloud_networking_network_v2.network_1 d90ce693-5ccf-4136-a0ed-
```

» opentelekomcloud_networking_port_v2

Manages a V2 port resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "opentelekomcloud_networking_port_v2" "port_1" {
  name           = "port_1"
  network_id     = "${opentelekomcloud_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) 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).
- `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.

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.

» Import

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

```
$ terraform import opentelekomcloud_networking_port_v2.port_1 eae26a3e-1c33-4cc1-9c31-0cd729
```

» Notes

» Ports and Instances

There are some notes to consider when connecting Instances to networks using Ports. Please see the `opentelekomcloud_compute_instance_v2` documenta-

tion for further documentation.

» opentelekomcloud_networking_router_interface_v2

Manages a V2 router interface resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_networking_network_v2" "network_1" {
  name           = "tf_test_network"
  admin_state_up = "true"
}

resource "opentelekomcloud_networking_subnet_v2" "subnet_1" {
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "opentelekomcloud_networking_router_v2" "router_1" {
  name             = "my_router"
  external_gateway = "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"
}

resource "opentelekomcloud_networking_router_interface_v2" "router_interface_1" {
  router_id = "${opentelekomcloud_networking_router_v2.router_1.id}"
  subnet_id = "${opentelekomcloud_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.

» opentelekomcloud_networking_router_route_v2

Creates a routing entry on a OpenTelekomCloud V2 router.

» Example Usage

```
resource "opentelekomcloud_networking_router_v2" "router_1" {
  name           = "router_1"
  admin_state_up = "true"
}

resource "opentelekomcloud_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "opentelekomcloud_networking_subnet_v2" "subnet_1" {
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "opentelekomcloud_networking_router_interface_v2" "int_1" {
  router_id = "${opentelekomcloud_networking_router_v2.router_1.id}"
  subnet_id = "${opentelekomcloud_networking_subnet_v2.subnet_1.id}"
}

resource "opentelekomcloud_networking_router_route_v2" "router_route_1" {
  depends_on       = ["opentelekomcloud_networking_router_interface_v2.int_1"]
  router_id        = "${opentelekomcloud_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 `opentelekomcloud_networking_router_route_v2` resource creation time. You can ensure that by explicitly specifying a dependency on the `opentelekomcloud_networking_router_interface_v2` resource that connects the next hop to the router, as in the example above.

» opentelekomcloud_networking_router_v2

Manages a V2 router resource within OpenTelekomCloud. The router is the top-level resource for the VPC within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_networking_router_v2" "router_1" {
  name             = "my_router"
  external_gateway = "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_gateway` - (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 an existing router.
- `enable_snat` - (Optional) Enable Source NAT for the router. Valid values are "true" or "false". An `external_gateway` has to be set in order to set this property. Changing this updates the `enable_snat` of the router.
- `tenant_id` - (Optional) The owner of the floating IP. Required if admin wants to create a router for another tenant. Changing this creates a new router.
- `value_specs` - (Optional) Map of additional driver-specific options.

» 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_gateway` - See Argument Reference above.
- `enable_snat` - See Argument Reference above.
- `tenant_id` - See Argument Reference above.
- `value_specs` - See Argument Reference above.

» opentelekomcloud_networking_subnet_v2

Manages a V2 Neutron subnet resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_networking_network_v2" "network_1" {
  name           = "tf_test_network"
  admin_state_up = "true"
}

resource "opentelekomcloud_networking_subnet_v2" "subnet_1" {
  network_id = "${opentelekomcloud_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. Acceptable values are "true" and "false". Changing this value enables or disables the DHCP capabilities of the existing subnet. Defaults to 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 opentelekomcloud_networking_subnet_v2.subnet_1 da4faf16-5546-41e4-8330-4c
```

» opentelekomcloud_networking_secgroup_v2

Manages a V2 neutron security group resource within OpenTelekomCloud. 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 "opentelekomcloud_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, OpenTelekomCloud 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 "opentelekomcloud_networking_secgroup_rule_v2" "secgroup_rule_v4" {
  direction = "egress"
  ethertype = "IPv4"
  security_group_id = "${opentelekomcloud_networking_secgroup_v2.secgroup.id}"
}

resource "opentelekomcloud_networking_secgroup_rule_v2" "secgroup_rule_v6" {
  direction = "egress"
  ethertype = "IPv6"
  security_group_id = "${opentelekomcloud_networking_secgroup_v2.secgroup.id}"
}
```

Please note that this behavior may differ depending on the configuration of the OpenTelekomCloud cloud. The above illustrates the current default Neutron behavior. Some OpenTelekomCloud 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 opentelekomcloud_networking_secgroup_v2.secgroup_1 38809219-5e8a-4852-913
```

» opentelekomcloud_networking_secgroup_rule_v2

Manages a V2 neutron security group rule resource within OpenTelekomCloud. 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 "opentelekomcloud_networking_secgroup_v2" "secgroup_1" {
  name        = "secgroup_1"
  description = "My neutron security group"
}

resource "opentelekomcloud_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 = "${opentelekomcloud_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 OpenTelekomCloud 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 OpenTelekomCloud 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 opentelekomcloud_networking_secgroup_rule_v2.secgroup_rule_1 aeb68ee3-6e9
```

» opentelekomcloud_vpc_v1

Manages a VPC resource within OpenTelekomCloud.

» Example Usage

```
variable "vpc_name" {
  default = "opentelekomcloud_vpc"
}

variable "vpc_cidr" {
  default = "192.168.0.0/16"
}

resource "opentelekomcloud_vpc_v1" "vpc_v1" {
  name = "${var.vpc_name}"
  cidr = "${var.vpc_cidr}"
}
```

» Argument Reference

The following arguments are supported:

- `cidr` - (Required) The range of available subnets in the VPC. The value ranges from 10.0.0.0/8 to 10.255.255.0/24, 172.16.0.0/12 to 172.31.255.0/24, or 192.168.0.0/16 to 192.168.255.0/24.

- `region` - (Optional) The region in which to obtain the V1 VPC client. A VPC client is needed to create a VPC. If omitted, the region argument of the provider is used. Changing this creates a new VPC.
- `name` - (Required) The name of the VPC. The name must be unique for a tenant. The value is a string of no more than 64 characters and can contain digits, letters, underscores (_), and hyphens (-). Changing this updates the name of the existing VPC.

» Attributes Reference

The following attributes are exported:

- `id` - ID of the VPC.
- `name` - See Argument Reference above.
- `cidr` - See Argument Reference above.
- `status` - The current status of the desired VPC. Can be either CREATING, OK, DOWN, PENDING_UPDATE, PENDING_DELETE, or ERROR.
- `shared` - Specifies whether the cross-tenant sharing is supported.
- `region` - See Argument Reference above.

» Import

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

```
$ terraform import opentelekomcloud_vpc_v1.vpc_v1 7117d38e-4c8f-4624-a505-bd96b97d024c
```

» opentelekomcloud_vpc_subnet_v1

Provides an VPC subnet resource.

» Example Usage

```
resource "opentelekomcloud_vpc_v1" "vpc_v1" {
  name = "${var.vpc_name}"
  cidr = "${var.vpc_cidr}"
}
```

```
resource "opentelekomcloud_vpc_subnet_v1" "subnet_v1" {
  name = "${var.subnet_name}"
  cidr = "${var.subnet_cidr}"
  gateway_ip = "${var.subnet_gateway_ip}"
  vpc_id = "${opentelekomcloud_vpc_v1.vpc_v1.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) - The subnet name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), and hyphens (-).
- `cidr` (Required) - Specifies the network segment on which the subnet resides. The value must be in CIDR format. The value must be within the CIDR block of the VPC. The subnet mask cannot be greater than 28. Changing this creates a new Subnet.
- `gateway_ip` (Required) - Specifies the gateway of the subnet. The value must be a valid IP address. The value must be an IP address in the subnet segment. Changing this creates a new Subnet.
- `vpc_id` (Required) - Specifies the ID of the VPC to which the subnet belongs. Changing this creates a new Subnet.
- `dhcp_enable` (Optional) - Specifies whether the DHCP function is enabled for the subnet. The value can be true or false. If this parameter is left blank, it is set to true by default.
- `primary_dns` (Optional) - Specifies the IP address of DNS server 1 on the subnet. The value must be a valid IP address.
- `secondary_dns` (Optional) - Specifies the IP address of DNS server 2 on the subnet. The value must be a valid IP address.
- `dns_list` (Optional) - Specifies the DNS server address list of a subnet. This field is required if you need to use more than two DNS servers. This parameter value is the superset of both DNS server address 1 and DNS server address 2.
- `availability_zone` (Optional) - Identifies the availability zone (AZ) to which the subnet belongs. The value must be an existing AZ in the system. Changing this creates a new Subnet.

» Attributes Reference

All of the argument attributes are also exported as result attributes:

- `id` - Specifies a resource ID in UUID format.
- `status` - Specifies the status of the subnet. The value can be ACTIVE, DOWN, UNKNOWN, or ERROR.
- `subnet_id` - Specifies the subnet (Native OpenStack API) ID.

» Import

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

```
$ terraform import opentelekomcloud_vpc_subnet_v1 4779ab1c-7c1a-44b1-a02e-93dfc361b32d
```

» opentelekomcloud_vpc_route_v2

Provides a resource to create a route.

» Example Usage

```
resource "opentelekomcloud_vpc_route_v2" "vpc_route" {
  type  = "peering"
  nexthop  = "${var.nexthop}"
  destination = "192.168.0.0/16"
  vpc_id = "${var.vpc_id}"
 }
```

» Argument Reference

The following arguments are supported:

- `destination` (Required) - Specifies the destination IP address or CIDR block. Changing this creates a new Route.
- `nexthop` (Required) - Specifies the next hop. If the route type is peering, enter the VPC peering connection ID. Changing this creates a new Route.
- `type` (Required) - Specifies the route type. Currently, the value can only be **peering**. Changing this creates a new Route.

- `vpc_id` (Required) - Specifies the VPC for which a route is to be added. Changing this creates a new Route.
- `tenant_id` (Optional) - Specifies the tenant ID. Only the administrator can specify the tenant ID of other tenant. Changing this creates a new Route.

» Attributes Reference

All of the argument attributes are also exported as result attributes:

- `id` - The route ID.

» opentelekomcloud_vpc_peering_connection_v2

Provides a resource to manage a VPC Peering Connection resource.

Note: For cross-tenant (requester's tenant differs from the accepter's tenant) VPC Peering Connections, use the `opentelekomcloud_vpc_peering_connection_v2` resource to manage the requester's side of the connection and use the `opentelekomcloud_vpc_peering_connection_accepter_v2` resource to manage the accepter's side of the connection.

» Example Usage

```
resource "opentelekomcloud_vpc_peering_connection_v2" "peering" {
  name = "${var.peer_conn_name}"
  vpc_id = "${var.vpc_id}"
  peer_vpc_id = "${var.accepter_vpc_id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) - Specifies the name of the VPC peering connection. The value can contain 1 to 64 characters.
- `vpc_id` (Required) - Specifies the ID of a VPC involved in a VPC peering connection. Changing this creates a new VPC peering connection.
- `peer_vpc_id` (Required) - Specifies the VPC ID of the accepter tenant. Changing this creates a new VPC peering connection.

- `peer_tenant_id` (Optional) - Specified the Tenant Id of the accepter tenant. Changing this creates a new VPC peering connection.

» Attributes Reference

All of the argument attributes are also exported as result attributes:

- `id` - The VPC peering connection ID.
- `status` - The VPC peering connection status. The value can be PENDING_ACCEPTANCE, REJECTED, EXPIRED, DELETED, or ACTIVE.

» Notes

If you create a VPC peering connection with another VPC of your own, the connection is created without the need for you to accept the connection.

» Import

VPC Peering resources can be imported using the `vpc peering id`, e.g.

```
$ terraform import opentelekomcloud_vpc_peering_connection_v2.test_connection
22b76469-08e3-4937-8c1d-7aad34892be1
```

» opentelekomcloud_vpc_peering_connection_accepter_v2

Provides a resource to manage the accepter's side of a VPC Peering Connection.

When a cross-tenant (requester's tenant differs from the accepter's tenant) VPC Peering Connection is created, a VPC Peering Connection resource is automatically created in the accepter's account. The requester can use the `opentelekomcloud_vpc_peering_connection_v2` resource to manage its side of the connection and the accepter can use the `opentelekomcloud_vpc_peering_connection_accepter_v2` resource to "adopt" its side of the connection into management.

» Example Usage

```
provider "opentelekomcloud"  {
   alias = "main"
   user_name   = "${var.username}"
   domain_name = "${var.domain_name}"
```

```
    password    = "${var.password}"
    auth_url    = "${var.auth_url}"
    region      = "${var.region}"
    tenant_id   = "${var.tenant_id}"
}

provider "opentelekomcloud"  {
    alias = "peer"
    user_name = "${var.peer_username}"
    domain_name = "${var.peer_domain_name}"
    password    = "${var.peer_password}"
    auth_url    = "${var.peer_auth_url}"
    region      = "${var.peer_region}"
    tenant_id   = "${var.peer_tenant_id}"
}

resource "opentelekomcloud_vpc_v1" "vpc_main" {
    provider = "opentelekomcloud.main"
    name = "${var.vpc_name}"
    cidr = "${var.vpc_cidr}"
}

resource "opentelekomcloud_vpc_v1" "vpc_peer" {
    provider = "opentelekomcloud.peer"
    name = "${var.peer_vpc_name}"
    cidr = "${var.peer_vpc_cidr}"
}

# Requester's side of the connection.
resource "opentelekomcloud_vpc_peering_connection_v2" "peering" {
    provider = "opentelekomcloud.main"
    name = "${var.peer_name}"
    vpc_id = "${opentelekomcloud_vpc_v1.vpc_main.id}"
    peer_vpc_id = "${opentelekomcloud_vpc_v1.vpc_peer.id}"
    peer_tenant_id =  "${var.tenant_id}"
}

# Accepter's side of the connection.
resource "opentelekomcloud_vpc_peering_connection_accepter_v2" "peer" {
    provider = "opentelekomcloud.peer"
    vpc_peering_connection_id = "${opentelekomcloud_vpc_peering_connection_v2.peering.id}"
    accept = true

}
```

» Argument Reference

The following arguments are supported:

- `vpc_peering_connection_id` (Required) - The VPC Peering Connection ID to manage. Changing this creates a new VPC peering connection accepter.
- `accept` (Optional)- Whether or not to accept the peering request. Defaults to `false`.

» Removing opentelekomcloud_vpc_peering_connection_accepter_v2 from your configuration

OpenTelekomCloud allows a cross-tenant VPC Peering Connection to be deleted from either the requester's or accepter's side. However, Terraform only allows the VPC Peering Connection to be deleted from the requester's side by removing the corresponding `opentelekomcloud_vpc_peering_connection_v2` resource from your configuration. Removing a `opentelekomcloud_vpc_peering_connection_accepter_v2` resource from your configuration will remove it from your state file and management, but will not destroy the VPC Peering Connection.

» Attributes Reference

All of the argument attributes except accept are also exported as result attributes.

- `name` - The VPC peering connection name.
- `id` - The VPC peering connection ID.
- `status` - The VPC peering connection status.
- `vpc_id` - The ID of requester VPC involved in a VPC peering connection.
- `peer_vpc_id` - The VPC ID of the accepter tenant.
- `peer_tenant_id` - The Tenant Id of the accepter tenant.

» opentelekomcloud_networking_vip_v2

Manages a V2 vip resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_networking_network_v2" "network_1" {
  name = "network_1"
  admin_state_up = "true"
}

resource "opentelekomcloud_networking_subnet_v2" "subnet_1" {
  name = "subnet_1"
  cidr = "192.168.199.0/24"
  ip_version = 4
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"
}

resource "opentelekomcloud_networking_router_interface_v2" "router_interface_1" {
  router_id = "${opentelekomcloud_networking_router_v2.router_1.id}"
  subnet_id = "${opentelekomcloud_networking_subnet_v2.subnet_1.id}"
}

resource "opentelekomcloud_networking_router_v2" "router_1" {
  name = "router_1"
  external_gateway = "0a2228f2-7f8a-45f1-8e09-9039e1d09975"
}

resource "opentelekomcloud_networking_vip_v2" "vip_1" {
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"
  subnet_id = "${opentelekomcloud_networking_subnet_v2.subnet_1.id}"
}
```

» Argument Reference

The following arguments are supported:

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

» Attributes Reference

The following attributes are exported:

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

» opentelekomcloud_networking_vip_associate_v2

Manages a V2 vip associate resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_networking_network_v2" "network_1" {
  name = "network_1"
  admin_state_up = "true"
}

resource "opentelekomcloud_networking_subnet_v2" "subnet_1" {
  name = "subnet_1"
  cidr = "192.168.199.0/24"
  ip_version = 4
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"
}

resource "opentelekomcloud_networking_router_interface_v2" "router_interface_1" {
  router_id = "${opentelekomcloud_networking_router_v2.router_1.id}"
  subnet_id = "${opentelekomcloud_networking_subnet_v2.subnet_1.id}"
}

resource "opentelekomcloud_networking_router_v2" "router_1" {
  name = "router_1"
  external_gateway = "0a2228f2-7f8a-45f1-8e09-9039e1d09975"
}

resource "opentelekomcloud_networking_port_v2" "port_1" {
  name = "port_1"
  admin_state_up = "true"
```

```
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"

  fixed_ip {
    subnet_id =  "${opentelekomcloud_networking_subnet_v2.subnet_1.id}"
  }
}

resource "opentelekomcloud_compute_instance_v2" "instance_1" {
  name = "instance_1"
  security_groups = ["default"]

  network {
    port = "${opentelekomcloud_networking_port_v2.port_1.id}"
  }
}

resource "opentelekomcloud_networking_port_v2" "port_2" {
  name = "port_2"
  admin_state_up = "true"
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"

  fixed_ip {
    subnet_id =  "${opentelekomcloud_networking_subnet_v2.subnet_1.id}"
  }
}

resource "opentelekomcloud_compute_instance_v2" "instance_2" {
  name = "instance_2"
  security_groups = ["default"]

  network {
    port = "${opentelekomcloud_networking_port_v2.port_1.id}"
  }
}

resource "opentelekomcloud_networking_vip_v2" "vip_1" {
  network_id = "${opentelekomcloud_networking_network_v2.network_1.id}"
  subnet_id = "${opentelekomcloud_networking_subnet_v2.subnet_1.id}"
}

resource "opentelekomcloud_networking_vip_associate_v2" "vip_associate_1" {
  vip_id = "${opentelekomcloud_networking_vip_v2.vip_1.id}"
  port_ids = ["${opentelekomcloud_networking_port_v2.port_1.id}", "${opentelekomcloud_networ
}
```

» Argument Reference

The following arguments are supported:

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

» Attributes Reference

The following attributes are exported:

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

» opentelekomcloud_lb_loadbalancer_v2

Manages a V2 loadbalancer resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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.

» opentelekomcloud_lb_listener_v2

Manages a V2 listener resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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.

» opentelekomcloud_lb_pool_v2

Manages a V2 pool resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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.

» opentelekomcloud_lb_member_v2

Manages a V2 member resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_lb_member_v2" "member_1" {
  address       = "192.168.199.23"
  protocol_port = 8080
  pool_id       = POOL_ID
  subnet_id     = SUBNET_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 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.

» opentelekomcloud_lb_monitor_v2

Manages a V2 monitor resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_lb_monitor_v2" "monitor_1" {
  pool_id     = "${opentelekomcloud_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.

» opentelekomcloud_fw_firewall_group_v2

Manages a v1 firewall group resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_fw_rule_v2" "rule_1" {
  name             = "my-rule-1"
  description      = "drop TELNET traffic"
  action           = "deny"
  protocol         = "tcp"
  destination_port = "23"
  enabled          = "true"
}

resource "opentelekomcloud_fw_rule_v2" "rule_2" {
  name             = "my-rule-2"
  description      = "drop NTP traffic"
  action           = "deny"
  protocol         = "udp"
  destination_port = "123"
  enabled          = "false"
}

resource "opentelekomcloud_fw_policy_v2" "policy_1" {
  name = "my-policy"

  rules = ["${opentelekomcloud_fw_rule_v2.rule_1.id}",
    "${opentelekomcloud_fw_rule_v2.rule_2.id}",
  ]
}

resource "opentelekomcloud_fw_firewall_group_v2" "firewall_group_1" {
  name      = "my-firewall-group"
  ingress_policy_id = "${opentelekomcloud_fw_policy_v2.policy_1.id}"
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the v1 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 opentelekomcloud_fw_firewall_group_v2.firewall_group_1 c9e39fb2-ce20-46c8
```

» opentelekomcloud_fw_policy_v2

Manages a v1 firewall policy resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_fw_rule_v2" "rule_1" {
  name             = "my-rule-1"
  description      = "drop TELNET traffic"
  action           = "deny"
  protocol         = "tcp"
  destination_port = "23"
  enabled          = "true"
}

resource "opentelekomcloud_fw_rule_v2" "rule_2" {
  name             = "my-rule-2"
  description      = "drop NTP traffic"
  action           = "deny"
  protocol         = "udp"
  destination_port = "123"
  enabled          = "false"
}

resource "opentelekomcloud_fw_policy_v2" "policy_1" {
  name = "my-policy"

  rules = ["${opentelekomcloud_fw_rule_v2.rule_1.id}",
    "${opentelekomcloud_fw_rule_v2.rule_2.id}",
  ]
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the v1 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.
- `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.

» Import

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

```
$ terraform import opentelekomcloud_fw_policy_v2.policy_1 07f422e6-c596-474b-8b94-fe2c12506c
```

» opentelekomcloud_fw_rule_v2

Manages a v2 firewall rule resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_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 v1 Compute 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 opentelekomcloud_fw_rule_v2.rule_1 8dbc0c28-e49c-463f-b712-5c5d1bbac327
```

» opentelekomcloud_rds_instance_v1

Manages rds instance resource within OpenTelekomCloud

» Example Usage: Creating a PostgreSQL RDS instance

```
data "opentelekomcloud_rds_flavors_v1" "flavor" {
    region = "eu-de"
    datastore_name = "PostgreSQL"
    datastore_version = "9.5.5"
    speccode = "rds.pg.s1.large.ha"
}

resource "opentelekomcloud_compute_secgroup_v2" "secgrp_rds" {
  name        = "secgrp-rds-instance"
  description = "Rds Security Group"
}

resource "opentelekomcloud_rds_instance_v1" "instance" {
  name = "rds-instance"
```

```
  datastore {
    type = "PostgreSQL"
    version = "9.5.5"
  }
  flavorref = "${data.opentelekomcloud_rds_flavors_v1.flavor.id}"
  volume {
    type = "COMMON"
    size = 200
  }
  region = "eu-de"
  availabilityzone = "eu-de-01"
  vpc = "c1095fe7-03df-4205-ad2d-6f4c181d436e"
  nics {
    subnetid = "b65f8d25-c533-47e2-8601-cfaa265a3e3e"
  }
  securitygroup {
    id = "${opentelekomcloud_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 = ["opentelekomcloud_compute_secgroup_v2.secgrp_rds"]
}
```

» Example Usage: Creating a SQLServer RDS instance

```
data "opentelekomcloud_rds_flavors_v1" "flavor" {
    region = "eu-de"
    datastore_name = "SQLServer"
    datastore_version = "2014 SP2 SE"
    speccode = "rds.mssql.s1.2xlarge"
}

resource "opentelekomcloud_compute_secgroup_v2" "secgrp_rds" {
  name        = "secgrp-rds-instance"
  description = "Rds Security Group"
}
```

```
resource "opentelekomcloud_rds_instance_v1" "instance" {
  name = "rds-instance"
  datastore {
    type = "SQLServer"
    version = "2014 SP2 SE"
  }
  flavorref = "${data.opentelekomcloud_rds_flavors_v1.flavor.id}"
  volume {
    type = "COMMON"
    size = 200
  }
  region = "eu-de"
  availabilityzone = "eu-de-01"
  vpc = "c1095fe7-03df-4205-ad2d-6f4c181d436e"
  nics {
    subnetid = "b65f8d25-c533-47e2-8601-cfaa265a3e3e"
  }
  securitygroup {
    id = "${opentelekomcloud_compute_secgroup_v2.secgrp_rds.id}"
  }
  dbport = "8635"
  backupstrategy = {
    starttime = "04:00:00"
    keepdays = 4
  }
  dbrtpd = "Huangwei!120521"
  depends_on = ["opentelekomcloud_compute_secgroup_v2.secgrp_rds"]
}
```

» Example Usage: Creating a MySQL RDS instance

```
data "opentelekomcloud_rds_flavors_v1" "flavor" {
    region = "eu-de"
    datastore_name = "MySQL"
    datastore_version = "5.6.33"
    speccode = "rds.mysql.s1.medium"
}

resource "opentelekomcloud_compute_secgroup_v2" "secgrp_rds" {
  name        = "secgrp-rds-instance"
  description = "Rds Security Group"
}

resource "opentelekomcloud_rds_instance_v1" "instance" {
  name = "rds-instance"
```

```
  datastore {
    type = "MySQL"
    version = "5.6.33"
  }
  flavorref = "${data.opentelekomcloud_rds_flavors_v1.flavor.id}"
  volume {
    type = "COMMON"
    size = 200
  }
  region = "eu-de"
  availabilityzone = "eu-de-01"
  vpc = "c1095fe7-03df-4205-ad2d-6f4c181d436e"
  nics {
    subnetid = "b65f8d25-c533-47e2-8601-cfaa265a3e3e"
  }
  securitygroup {
    id = "${opentelekomcloud_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 = ["opentelekomcloud_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). If you want to enable ha for the rds instance, a flavor with ha speccode is required.
- `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
	9.6.5
MySQL	5.6.33
	5.6.30
	5.6.34
	5.6.35
	5.6.36
	5.7.17
	5.7.20
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.

» opentelekomcloud_s3_bucket

Provides a S3 bucket resource.

» Example Usage

» Private Bucket w/ Tags

```
resource "opentelekomcloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  tags {
    Name        = "My bucket"
    Environment = "Dev"
  }
}
```

» Static Website Hosting

```
resource "opentelekomcloud_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 "opentelekomcloud_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 "opentelekomcloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  versioning {
    enabled = true
  }
}
```

» Enable Logging

```
resource "opentelekomcloud_s3_bucket" "log_bucket" {
  bucket = "my-tf-log-bucket"
  acl    = "log-delivery-write"
}

resource "opentelekomcloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  logging {
    target_bucket = "${opentelekomcloud_s3_bucket.log_bucket.id}"
    target_prefix = "log/"
  }
}
```

» Using object lifecycle

```
resource "opentelekomcloud_s3_bucket" "bucket" {
  bucket = "my-bucket"
  acl    = "private"

  lifecycle_rule {
    id      = "log"
    enabled = true

    prefix  = "log/"
    tags {
      "rule"      = "log"
      "autoclean" = "true"
    }

    expiration {
      days = 90
    }
  }

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

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

```
    }
  }
}

resource "opentelekomcloud_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.
- `tags` - (Optional) A mapping of tags to assign to the bucket.
- `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 AWS 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.
- `tags` - (Optional) Specifies object tags key and value.

- `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 AWS 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 opentelekomcloud_s3_bucket.bucket bucket-name
```

» opentelekomcloud_s3_bucket_object

Provides a S3 bucket object resource.

» Example Usage

» Uploading a file to a bucket

```
resource "opentelekomcloud_s3_bucket_object" "object" {
  bucket = "your_bucket_name"
  key    = "new_object_key"
  source = "path/to/file"
  etag   = "${md5(file("path/to/file"))}"
}
```

resource ”opentelekomcloud_s3_bucket” ”examplebucket” { bucket = ”examplebuckettftest” acl = ”private” }

resource ”opentelekomcloud_s3_bucket_object” ”examplebucket_object” { key = ”someobject” bucket = ”${opentelekomcloud_s3_bucket.examplebucket.bucket}” source = ”index.html” } “‘

» Server Side Encryption with S3 Default Master Key

```
resource "opentelekomcloud_s3_bucket" "examplebucket" {
  bucket = "examplebuckettftest"
```

```
  acl    = "private"
}

resource "opentelekomcloud_s3_bucket_object" "examplebucket_object" {
  key                    = "someobject"
  bucket                 = "${opentelekomcloud_s3_bucket.examplebucket.bucket}"
  source                 = "index.html"
  server_side_encryption = "aws:kms"
}
```

» Argument Reference

Note: If you specify `content_encoding` you are responsible for encoding the body appropriately (i.e. `source` and `content` both expect already encoded/compressed bytes)

The following arguments are supported:

- `bucket` - (Required) The name of the bucket to put the file in.
- `key` - (Required) The name of the object once it is in the bucket.
- `source` - (Required) The path to the source file being uploaded to the bucket.
- `content` - (Required unless `source` given) The literal content being uploaded to the bucket.
- `acl` - (Optional) The canned ACL to apply. Defaults to "private".
- `cache_control` - (Optional) Specifies caching behavior along the request/reply chain Read w3c cache_control for further details.
- `content_disposition` - (Optional) Specifies presentational information for the object. Read wc3 content_disposition for further information.
- `content_encoding` - (Optional) Specifies what content encodings have been applied to the object and thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field. Read w3c content encoding for further information.
- `content_language` - (Optional) The language the content is in e.g. en-US or en-GB.
- `content_type` - (Optional) A standard MIME type describing the format of the object data, e.g. application/octet-stream. All Valid MIME Types are valid for this input.
- `website_redirect` - (Optional) Specifies a target URL for website redirect.
- `etag` - (Optional) Used to trigger updates. The only meaningful value is `${md5(file("path/to/file"))}`. This attribute is not compatible with `kms_key_id`.
- `server_side_encryption` - (Optional) Specifies server-side encryption of the object in S3. Valid values are "`AES256`" and "`aws:kms`".
- `tags` - (Optional) A mapping of tags to assign to the object.

Either `source` or `content` must be provided to specify the bucket content. These two arguments are mutually-exclusive.

» Attributes Reference

The following attributes are exported

- `id` - the `key` of the resource supplied above
- `etag` - the ETag generated for the object (an MD5 sum of the object content).
- `version_id` - A unique version ID value for the object, if bucket versioning is enabled.

» opentelekomcloud_s3_bucket_policy

Attaches a policy to an S3 bucket resource.

» Example Usage

» Basic Usage

```
resource "opentelekomcloud_s3_bucket" "b" {
  bucket = "my_tf_test_bucket"
}

resource "opentelekomcloud_s3_bucket_policy" "b" {
  bucket = "${opentelekomcloud_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.

» opentelekomcloud_kms_key_v1

Manages a V1 key resource within KMS.

» Example Usage

```
resource "opentelekomcloud_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 OpenTelekomCloud 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.
- `origin` - Origin of a key. The default value is kms.
- `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 opentelekomcloud_kms_key_v1.key_1 7056d636-ac60-4663-8a6c-82d3c32c1c64
```

» opentelekomcloud_smn_subscription_v2

Manages a V2 subscription resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_smn_topic_v2" "topic_1" {
  name         = "topic_1"
  display_name    = "The display name of topic_1"
}

resource "opentelekomcloud_smn_subscription_v2" "subscription_1" {
  topic_urn       = "${opentelekomcloud_smn_topic_v2.topic_1.id}"
  endpoint        = "mailtest@gmail.com"
  protocol        = "email"
```

```
  remark          = "O&M"
}

resource "opentelekomcloud_smn_subscription_v2" "subscription_2" {
  topic_urn       = "${opentelekomcloud_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.

» opentelekomcloud_smn_subscription_v2

Manages a V2 subscription resource within OpenTelekomCloud.

» Example Usage

```
resource "opentelekomcloud_smn_topic_v2" "topic_1" {
  name        = "topic_1"
  display_name    = "The display name of topic_1"
}

resource "opentelekomcloud_smn_subscription_v2" "subscription_1" {
  topic_urn       = "${opentelekomcloud_smn_topic_v2.topic_1.id}"
  endpoint        = "mailtest@gmail.com"
  protocol        = "email"
  remark          = "O&M"
}

resource "opentelekomcloud_smn_subscription_v2" "subscription_2" {
  topic_urn       = "${opentelekomcloud_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.

» opentelekomcloud_vpc_eip_v1

Manages a V1 EIP resource within OpenTelekomCloud VPC.

» Example Usage

```
resource "opentelekomcloud_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 opentelekomcloud_vpc_eip_v1.eip_1 2c7f39f3-702b-48d1-940c-b50384177ee1
```

» Resource: opentelekomcloud_compute_bms_tags_v2

Used to add tags to a BMS.

» Example Usage

```
variable "bms_id" { }

resource "opentelekomcloud_compute_bms_tags_v2" "add_tags"
{
     server_id = "${var.bms_id}",
     tags = [
               "tags__type_baremetal"
             ]
}
```

» Argument Reference

The following arguments are supported:

- `server_id` (Required) -The unique id of bare metal server.
- `tags` (Required) - The tags of a BMS. Changing this parameter creates a new resource.

» Attributes Reference

All above argument parameters can be exported as attribute parameters.

» Import

BMS tags can be imported using the server_id, e.g. `$ terraform import opentelekomcloud_compute_bms_tags_v2.add_tags 4779ab1c-7c1a-44b1-a02e-93dfc361b32d`

» opentelekomcloud_rts_stack_v1

Provides an OpenTelekomCloud Stack.

» Example Usage

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

resource "opentelekomcloud_rts_stack_v1" "mystack" {
  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:

- `name` - (Required) A unique name for the stack. The value must meet the regular expression rule (`^[a-zA-Z][a-zA-Z0-9_.-]{0,254}$`). Changing this creates a new stack.
- `template_body` - (Optional; Required if `template_url` is empty) Structure containing the template body. The template content must use the yaml syntax.
- `template_url` - (Optional; Required if `template_body` is empty) Location of a file containing the template body.
- `environment` - (Optional) Tthe environment information about the stack.
- `files` - (Optional) Files used in the environment.
- `parameters` - (Optional) A list of Parameter structures that specify input parameters for the stack.
- `disable_rollback` - (Optional) Set to true to disable rollback of the stack if stack creation failed.
- `timeout_mins` - (Optional) Specifies the timeout duration.

» Attributes Reference

In addition to all arguments above, 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 opentelekomcloud_rts_stack_v1.mystack rts-stack
```

» Timeouts

`opentelekomcloud_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.

» opentelekomcloud_rts_software_config_v1

Provides an RTS software config resource.

» Example Usage

```
variable "config_name" {}

resource "opentelekomcloud_rts_software_config_v1" "myconfig" {
  name = "${var.config_name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the software configuration.
- `group` - (Optional) The namespace that groups this software configuration by when it is delivered to a server.
- `inputs` - (Optional) A list of software configuration inputs.
- `outputs` - (Optional) A list of software configuration outputs.
- `config` - (Optional) The software configuration code.
- `options` - (Optional) The software configuration options.

» Attributes Reference

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

- `id` - The id of the software config.

» Import

Software Config can be imported using the `config id`, e.g. `$ terraform import opentelekomcloud_rts_software_config_v1 4779ab1c-7c1a-44b1-a02e-93dfc361b32d`

» opentelekomcloud_rts_software_deployment_v1

Provides an RTS software deployment resource.

» Example Usage

```
variable "config_id" {}

variable "server_id" {}

resource "opentelekomcloud_rts_software_deployment_v1" "mydeployment" {
  config_id = "${var.config_id}"
  server_id = "${var.server_id}"
}
```

» Argument Reference

The following arguments are supported:

- `config_id` - (Required) The id of the software configuration resource running on an instance.
- `server_id` - (Required) The id of the instance.
- `status` - (Optional) The current status of deployment resources.
- `action` - (Optional) The stack action that triggers this deployment resource.
- `input_values` - (Optional) The input data stored in the form of a key-value pair.
- `output_values` - (Optional) The output data stored in the form of a key-value pair.
- `status_reason` - (Optional) The cause of the current deployment resource status.
- `tenant_id` - (Optional) The id of the authenticated tenant who can perform operations on the deployment resources.

» Attributes Reference

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

- `id` - The id of the software deployment.

» Import

Software deployment can be imported using the `deployment id`, e.g.
`$ terraform import opentelekomcloud_rts_software_deployment_v1 4779ab1c-7c1a-44b1-a02e-93dfc361b32d`

» opentelekomcloud_sfs_file_system_v2

Provides an Shared File System (SFS) resource.

» Example Usage

```
variable "share_name" { }

variable "share_description" { }

variable "vpc_id" { }

resource "opentelekomcloud_sfs_file_system_v2" "share-file"
{
        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 opentelekomcloud_sfs_file_system_v2 4779ab1c-7c1a-44b1-a02e-93dfc361b32d
```

» Resource: opentelekomcloud_deh_host_v1

Allocates a Dedicated Host to a tenant and set minimum required parameters for this Dedicated Host.

» Example Usage

```
resource "opentelekomcloud_deh_host_v1" "deh_host"
{
    name = "high_performance_deh"
    auto_placement = "on"
    availability_zone = "eu-de-02"
    host_type = "h1"
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) - The name of the Dedicated Host.
- `auto_placement` (Optional) - Allows a instance to be automatically placed onto the available Dedicated Hosts. The default value is **on**.
- `availability_zone` (Required) - The Availability Zone to which the Dedicated Host belongs. Changing this parameter creates a new resource.
- `host_type` (Required) - The Dedicated Host type. Expected values are **h1, general** and **d1**. Changing this parameter creates a new resource.

» Attributes Reference

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

- `status` - Specifies the Dedicated Host status.
- `available_vcpus` - The number of available vCPUs for the Dedicated Host.
- `available_memory` - The size of available memory for the Dedicated Host.
- `instance_total` - The number of the placed VMs.
- `instance_uuids` - The VMs started on the Dedicated Host.
- `host_type_name` - The name of the Dedicated Host type.
- `vcpus` - The number of host vCPUs.
- `cores` - The number of host physical cores.
- `sockets` - The number of host physical sockets.
- `memory` - The size of host physical memory (MB).
- `available_instance_capacities` - The VM flavors placed on the Dedicated Host.

» Import

DeH can be imported using the `dedicated_host_id`, e.g. `$ terraform import opentelekomcloud_deh_host_v1.deh_host 4779ab1c-7c1a-44b1-a02e-93dfc361b32d`

» opentelekomcloud_as_configuration_v1

Manages a V1 AS Configuration resource within OpenTelekomCloud.

» Example Usage

» Basic AS Configuration

```
resource "opentelekomcloud_as_configuration_v1" "my_as_config" {
  scaling_configuration_name = "my_as_config"
  instance_config = {
    flavor = "${var.flavor}"
    image = "${var.image_id}"
    disk = [
      {size = 40
      volume_type = "SATA"
      disk_type = "SYS"}
    ]
    key_name = "${var.keyname}"
    user_data = "${file("userdata.txt")}"
  }
}
```

» AS Configuration With User Data and Metadata

```
resource "opentelekomcloud_as_configuration_v1" "my_as_config" {
  scaling_configuration_name = "my_as_config"
  instance_config = {
    flavor = "${var.flavor}"
    image = "${var.image_id}"
    disk = [
      {size = 40
      volume_type = "SATA"
      disk_type = "SYS"}
    ]
    key_name = "${var.keyname}"
    user_data = "${file("userdata.txt")}"
    metadata = {
      some_key = "some_value"
    }
  }
}
```

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

» AS Configuration uses the existing instance specifications as the template

```
resource "opentelekomcloud_as_configuration_v1" "my_as_config" {
  scaling_configuration_name = "my_as_config"
  instance_config = {
    instance_id = "4579f2f5-cbe8-425a-8f32-53dcb9d9053a"
    key_name = "${var.keyname}"
  }
}
```

» Argument Reference

The following arguments are supported:

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

The `instance_config` block supports:

- `instance_id` - (Optional) When using the existing instance specifications as the template to create AS configurations, specify this argument. In this case, flavor, image, and disk arguments do not take effect. If the instance_id argument is not specified, flavor, image, and disk arguments are mandatory.
- `flavor` - (Optional) The flavor ID.
- `image` - (Optional) The image ID.
- `disk` - (Optional) The disk group information. System disks are mandatory and data disks are optional. The dick structure is described below.
- `key_name` - (Required) The name of the SSH key pair used to log in to the instance.

- `user_data` - (Optional) The user data to provide when launching the instance. The file content must be encoded with Base64.
- `personality` - (Optional) Customize the personality of an instance by defining one or more files and their contents. The personality structure is described below.
- `public_ip` - (Optional) The elastic IP address of the instance. The public_ip structure is described below.
- `metadata` - (Optional) Metadata key/value pairs to make available from within the instance.

The `disk` block supports:

- `size` - (Required) The disk size. The unit is GB. The system disk size ranges from 40 to 32768, and the data disk size ranges from 10 to 32768.
- `volume_type` - (Required) The disk type, which must be the same as the disk type available in the system. The options include `SATA` (common I/O disk type) and `SSD` (ultra-high I/O disk type).
- `disk_type` - (Required) Whether the disk is a system disk or a data disk. Option `DATA` indicates a data disk. option `SYS` indicates a system disk.

The `personality` block supports:

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

The `public_ip` block supports:

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

The `eip` block supports:

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

The `bandwidth` block supports:

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

» opentelekomcloud_as_group_v1

Manages a V1 Autoscaling Group resource within OpenTelekomCloud.

» Example Usage

» Basic Autoscaling Group

```
resource "opentelekomcloud_as_group_v1" "my_as_group" {
  scaling_group_name = "my_as_group"
  scaling_configuration_id = "37e310f5-db9d-446e-9135-c625f9c2bbfc"
  desire_instance_number = 2
  min_instance_number = 0
  max_instance_number = 10
  networks = [{id = "ad091b52-742f-469e-8f3c-fd81cadf0743"}]
  security_groups = [{id = "45e4c6de-6bf0-4843-8953-2babde3d4810"}]
  vpc_id = "1d8f7e7c-fe04-4cf5-85ac-08b478c290e9"
  delete_publicip = true
  delete_instances = "yes"
}
```

» Autoscaling Group Only Remove Members When Scaling Down

```
resource "opentelekomcloud_as_group_v1" "my_as_group_only_remove_members" {
  scaling_group_name = "my_as_group_only_remove_members"
  scaling_configuration_id = "37e310f5-db9d-446e-9135-c625f9c2bbfc"
  desire_instance_number = 2
  min_instance_number = 0
  max_instance_number = 10
  networks = [{id = "ad091b52-742f-469e-8f3c-fd81cadf0743"}]
  security_groups = [{id = "45e4c6de-6bf0-4843-8953-2babde3d4810"}]
  vpc_id = "1d8f7e7c-fe04-4cf5-85ac-08b478c290e9"
  delete_publicip = true
  delete_instances = "no"
}
```

» Autoscaling Group With ELB Listener

```
resource "opentelekomcloud_as_group_v1" "my_as_group_with_elb" {
  scaling_group_name = "my_as_group_with_elb"
  scaling_configuration_id = "37e310f5-db9d-446e-9135-c625f9c2bbfc"
  desire_instance_number = 2
  min_instance_number = 0
```

```
  max_instance_number = 10
  networks = [{id = "ad091b52-742f-469e-8f3c-fd81cadf0743"}]
  security_groups = [{id = "45e4c6de-6bf0-4843-8953-2babde3d4810"}]
  vpc_id = "1d8f7e7c-fe04-4cf5-85ac-08b478c290e9"
  lb_listener_id = "${opentelekomcloud_elb_listener.my_listener.id}"
  delete_publicip = true
  delete_instances = "yes"
}

resource "opentelekomcloud_elb_listener" "my_listener" {
  name = "my_listener"
  description = "my test listener"
  protocol = "TCP"
  backend_protocol = "TCP"
  port = 12345
  backend_port = 21345
  lb_algorithm = "roundrobin"
  loadbalancer_id = "cba48790-baf5-4446-adb3-02069a916e97"
  timeouts {
        create = "5m"
        update = "5m"
        delete = "5m"
  }
}
```

» Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to create the AS group. If omitted, the `region` argument of the provider is used. Changing this creates a new AS group.
- `scaling_group_name` - (Required) The name of the scaling group. The name can contain letters, digits, underscores(_), and hyphens(-),and cannot exceed 64 characters.
- `scaling_configuration_id` - (Optional) The configuration ID which defines configurations of instances in the AS group.
- `desire_instance_number` - (Optional) The expected number of instances. The default value is the minimum number of instances. The value ranges from the minimum number of instances to the maximum number of instances.
- `min_instance_number` - (Optional) The minimum number of instances. The default value is 0.

- `max_instance_number` - (Optional) The maximum number of instances. The default value is 0.
- `cool_down_time` - (Optional) The cooling duration (in seconds). The value ranges from 0 to 86400, and is 900 by default.
- `lb_listener_id` - (Optional) The ELB listener IDs. The system supports up to three ELB listeners, the IDs of which are separated using a comma (,).
- `available_zones` - (Optional) The availability zones in which to create the instances in the autoscaling group.
- `networks` - (Required) An array of one or more network IDs. The system supports up to five networks. The networks object structure is documented below.
- `security_groups` - (Required) An array of one or more security group IDs to associate with the group. The security_groups object structure is documented below.
- `vpc_id` - (Required) The VPC ID. Changing this creates a new group.
- `health_periodic_audit_method` - (Optional) The health check method for instances in the AS group. The health check methods include `ELB_AUDIT` and `NOVA_AUDIT`. If load balancing is configured, the default value of this parameter is `ELB_AUDIT`. Otherwise, the default value is `NOVA_AUDIT`.
- `health_periodic_audit_time` - (Optional) The health check period for instances. The period has four options: 5 minutes (default), 15 minutes, 60 minutes, and 180 minutes.
- `instance_terminate_policy` - (Optional) The instance removal policy. The policy has four options: `OLD_CONFIG_OLD_INSTANCE` (default), `OLD_CONFIG_NEW_INSTANCE`, `OLD_INSTANCE`, and `NEW_INSTANCE`.
- `notifications` - (Optional) The notification mode. The system only supports `EMAIL` mode which refers to notification by email.
- `delete_publicip` - (Optional) Whether to delete the elastic IP address bound to the instances of AS group when deleting the instances. The options are `true` and `false`.
- `delete_instances` - (Optional) Whether to delete the instances in the AS group when deleting the AS group. The options are `yes` and `no`.

The `networks` block supports:

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

The `security_groups` block supports:

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

» Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `scaling_group_name` - See Argument Reference above.
- `desire_instance_number` - See Argument Reference above.
- `min_instance_number` - See Argument Reference above.
- `max_instance_number` - See Argument Reference above.
- `cool_down_time` - See Argument Reference above.
- `lb_listener_id` - See Argument Reference above.
- `health_periodic_audit_method` - See Argument Reference above.
- `health_periodic_audit_time` - See Argument Reference above.
- `instance_terminate_policy` - See Argument Reference above.
- `scaling_configuration_id` - See Argument Reference above.
- `delete_publicip` - See Argument Reference above.
- `notifications` - See Argument Reference above.
- `instances` - The instances IDs of the AS group.

» opentelekomcloud_as_policy_v1

Manages a V1 AS Policy resource within OpenTelekomCloud.

» Example Usage

» AS Recurrence Policy

```
resource "opentelekomcloud_as_policy_v1" "hth_aspolicy"{
  scaling_policy_name = "hth_aspolicy"
  scaling_group_id = "4579f2f5-cbe8-425a-8f32-53dcb9d9053a"
  cool_down_time = 900
  scaling_policy_type = "RECURRENCE"
  scaling_policy_action = {
    operation = "ADD"
    instance_number = 1
  }
  scheduled_policy = {
    launch_time = "07:00"
    recurrence_type = "Daily"
    start_time = "2017-11-30T12:00Z"
    end_time = "2017-12-30T12:00Z"
  }
}
```

» AS Scheduled Policy

```
resource "opentelekomcloud_as_policy_v1" "hth_aspolicy_1"{
  scaling_policy_name = "hth_aspolicy_1"
  scaling_group_id = "4579f2f5-cbe8-425a-8f32-53dcb9d9053a"
  cool_down_time = 900
  scaling_policy_type = "SCHEDULED"
  scaling_policy_action = {
    operation = "REMOVE"
    instance_number = 1
  }
  scheduled_policy = {
    launch_time = "2017-12-22T12:00Z"
  }
}
```

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

» AS Alarm Policy

```
resource "opentelekomcloud_as_policy_v1" "hth_aspolicy_2"{
  scaling_policy_name = "hth_aspolicy_2"
  scaling_group_id = "4579f2f5-cbe8-425a-8f32-53dcb9d9053a"
  cool_down_time = 900
  scaling_policy_type = "ALARM"
  alarm_id = "37e310f5-db9d-446e-9135-c625f9c2bbfc"
  scaling_policy_action = {
    operation = "ADD"
    instance_number = 1
  }
}
```

» Argument Reference

The following arguments are supported:

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

- `scaling_group_id` - (Required) The AS group ID. Changing this creates a new AS policy.
- `scaling_policy_type` - (Required) The AS policy type. The values can be `ALARM`, `SCHEDULED`, and `RECURRENCE`.
- `alarm_id` - (Optional) The alarm rule ID. This argument is mandatory when `scaling_policy_type` is set to `ALARM`.
- `scheduled_policy` - (Optional) The periodic or scheduled AS policy. This argument is mandatory when `scaling_policy_type` is set to `SCHEDULED` or `RECURRENCE`. The scheduled_policy structure is documented below.
- `scaling_policy_action` - (Optional) The action of the AS policy. The scaling_policy_action structure is documented below.
- `cool_down_time` - (Optional) The cooling duration (in seconds), and is 900 by default.

The `scheduled_policy` block supports:

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

The `scaling_policy_action` block supports:

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

» Attributes Reference

The following attributes are exported:

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

» opentelekomcloud_csbs_backup_v1

Provides an OpenTelekomCloud Backup of Resources.

» Example Usage

```
variable "backup_name" { }
variable "resource_id" { }

resource "opentelekomcloud_csbs_backup_v1" "backup_v1" {
  backup_name = "${var.backup_name}"
  resource_id = "${var.resource_id}"
  resource_type = "OS::Nova::Server"
}
```

» Argument Reference

The following arguments are supported:

- `backup_name` - (Optional) Name for the backup. The value consists of 1 to 255 characters and can contain only letters, digits, underscores (_), and hyphens (-). Changing backup_name creates a new backup.
- `description` - (Optional) Backup description. The value consists of 0 to 255 characters and must not contain a greater-than sign (>) or less-than sign (<). Changing description creates a new backup.

- `resource_id` - (Required) ID of the target to which the backup is restored. Changing this creates a new backup.
- `resource_type` - (Optional) Type of the target to which the backup is restored. The default value is **OS::Nova::Server** for an ECS. Changing this creates a new backup.
- `tags` - (Optional) block supports the following arguments:
  - `key` - (Required) Tag key. It cannot be an empty string.Changing key creates a new backup.
  - `value` - (Required) Tag value. It can be an empty string.Changing value creates a new backup.

» Attributes Reference

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

- `status` - It specifies the status of backup.
- `backup_record_id` - Specifies backup record ID.
- `volume_backups` block supports the following arguments:
  - `status` - Status of backup Volume.
  - `space_saving_ratio` - Specifies space saving rate.
  - `name` - It gives EVS disk backup name.
  - `bootable` - Specifies whether the disk is bootable.
  - `average_speed` - Specifies the average speed.
  - `source_volume_size` - Shows source volume size in GB.
  - `source_volume_id` - It specifies source volume ID.
  - `incremental` - Shows whether incremental backup is used.
  - `snapshot_id` - ID of snapshot.
  - `source_volume_name` - Specifies source volume name.
  - `image_type` - It specifies backup. The default value is backup.
  - `id` - Specifies Cinder backup ID.
  - `size` - Specifies accumulated size (MB) of backups.
- `vm_metadata` block supports the following arguments:
  - `name` - Name of backup data.
  - `eip` - Specifies elastic IP address of the ECS.
  - `cloud_service_type` - Specifies ECS type.
  - `ram` - Specifies memory size of the ECS, in MB.
  - `vcpus` - Specifies CPU cores corresponding to the ECS.
  - `private_ip` - It specifies internal IP address of the ECS.
  - `disk` - Shows system disk size corresponding to the ECS specifications.
  - `image_type` - Specifies image type.

» Import

Backup can be imported using `backup_record_id`, e.g.

```
$ terraform import opentelekomcloud_csbs_backup_v1.backup_v1.backup_v1 7056d636-ac60-4663-8a
```

» opentelekomcloud_csbs_backup_policy_v1

Provides an OpenTelekomCloud Backup Policy of Resources.

» Example Usage

```
variable "name" { }
variable "id" { }
variable "resource_name" { }

resource "opentelekomcloud_csbs_backup_policy_v1" "backup_policy_v1" {
  name            = "${var.name}"
  resource {
    id = "${var.id}"
    type = "OS::Nova::Server"
    name = "${var.resource_name}"
  }
  scheduled_operation {
    enabled = true
    operation_type ="backup"
    trigger_pattern = "BEGIN:VCALENDAR\r\nBEGIN:VEVENT\r\nRRULE:FREQ=WEEKLY;BYDAY=TH;BYHOUR
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the name of backup policy. The value consists of 1 to 255 characters and can contain only letters, digits, underscores (_), and hyphens (-).
- `description` - (Optional) Backup policy description. The value consists of 0 to 255 characters and must not contain a greater-than sign (>) or less-than sign (<).
- `provider_id` - (Required) Specifies backup provider ID. Default value is **fc4d5750-22e7-4798-8a46-f48f62c4c1da**

- `common` - (Optional) General backup policy parameters, which are blank by default.
- `scheduled_operation` block supports the following arguments:
  - `name` - (Optional) Specifies Scheduling period name.The value consists of 1 to 255 characters and can contain only letters, digits, underscores (_), and hyphens (-).
  - `description` - (Optional) Specifies Scheduling period description.The value consists of 0 to 255 characters and must not contain a greater-than sign (>) or less-than sign (<).
  - `enabled` - (Optional) Specifies whether the scheduling period is enabled. Default value is **true**
  - `max_backups` - (Optional) Specifies maximum number of backups that can be automatically created for a backup object.
  - `retention_duration_days` - (Optional) Specifies duration of retaining a backup, in days.
  - `permanent` - (Optional) Specifies whether backups are permanently retained.
  - `trigger_pattern` - (Required) Specifies Scheduling policy of the scheduler.
  - `operation_type` - (Required) Specifies Operation type, which can be backup.
- `resource` block supports the following arguments:
  - `id` - (Required) Specifies the ID of the object to be backed up.
  - `type` - (Required) Entity object type of the backup object. If the type is VMs, the value is **OS::Nova::Server**.
  - `name` - (Required) Specifies backup object name.
- `tags` block supports the following arguments:
  - `key` - (Required) Tag key. It cannot be an empty string.
  - `value` - (Required) Tag value. It can be an empty string.

» Attributes Reference

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

- `status` - Status of Backup Policy.
- `id` - Backup Policy ID.
- scheduled_operation - Backup plan information
  - `id` - Specifies Scheduling period ID.
  - `trigger_id` - Specifies Scheduler ID.
  - `trigger_name` - Specifies Scheduler name.
  - `trigger_type` - Specifies Scheduler type.

» Import

Backup Policy can be imported using `id`, e.g.

```
$ terraform import opentelekomcloud_csbs_backup_policy_v1.backup_policy_v1 7056d636-ac60-466
```

» opentelekomcloud_dms_queue_v2

Manages a DMS queue in the opentelekomcloud DMS Service.

» Example Usage

» Automatically detect the correct network

```
resource "opentelekomcloud_dms_queue_v1" "queue_1" {
  name  = "queue_1"
  description  = "test create dms queue"
  queue_mode  = "FIFO"
  redrive_policy  = "enable"
  max_consume_count = 80
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Indicates the unique name of a queue. A string of 1 to 64 characters that contain a-z, A-Z, 0-9, hyphens (-), and underscores (_). The name cannot be modified once specified.
- `queue_mode` - (Optional) Indicates the queue type. It only support 'NORMAL' and 'FIFO'. NORMAL: Standard queue. Best-effort ordering. Messages might be retrieved in an order different from which they were sent. Select standard queues when throughput is important. FIFO: First-In-First-out (FIFO) queue. FIFO delivery. Messages are retrieved in the order they were sent. Select FIFO queues when the order of messages is important. Default value: NORMAL.
- `description` - (Optional) Indicates the basic information about a queue. The queue description must be 0 to 160 characters in length, and does not contain angle brackets (<) and (>).
- `redrive_policy` - (Optional) Indicates whether to enable dead letter messages. Dead letter messages indicate messages that cannot be normally

consumed. The redrive_policy should be set to 'enable' or 'disable'. The default value is 'disable'.

- `max_consume_count` - (Optional) This parameter is mandatory only when redrive_policy is set to enable. This parameter indicates the maximum number of allowed message consumption failures. When a message fails to be consumed after the number of consumption attempts of this message reaches this value, DMS stores this message into the dead letter queue. The max_consume_count value range is 1–100.

» Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.
- `queue_mode` - See Argument Reference above.
- `description` - See Argument Reference above.
- `redrive_policy` - See Argument Reference above.
- `max_consume_count` - See Argument Reference above.
- `created` - Indicates the time when a queue is created.
- `reservation` - Indicates the retention period (unit: min) of a message in a queue.
- `max_msg_size_byte` - Indicates the maximum message size (unit: byte) that is allowed in queue.
- `produced_messages` - Indicates the total number of messages (not including the messages that have expired and been deleted) in a queue.
- `group_count` - Indicates the total number of consumer groups in a queue.

» opentelekomcloud_dms_group_v2

Manages a DMS group in the opentelekomcloud DMS Service.

» Example Usage

» Automatically detect the correct network

```
resource "opentelekomcloud_dms_group_v1" "queue_1" {
  name  = "queue_1"
  description  = "test create dms queue"
  queue_mode  = "FIFO"
  redrive_policy  = "enable"
  max_consume_count = 80
}
```

```
resource "opentelekomcloud_dms_group_v1" "group_1" {
  name = "group_1"
  queue_id = "${opentelekomcloud_dms_queue_v1.queue_1.id}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Indicates the unique name of a group. A string of 1 to 64 characters that contain a-z, A-Z, 0-9, hyphens (-), and underscores (_). The name cannot be modified once specified.
- `queue_id` - (Required) Indicates the ID of a specified queue.

» Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.
- `queue_id` - Indicates the ID of a queue.
- `redrive_policy` - Indicates whether to enable dead letter messages.
- `produced_messages` - Indicates the total number of messages (not including the messages that have expired and been deleted) in a queue.
- `consumed_messages` - Indicates the total number of messages that are successfully consumed.
- `available_messages` - Indicates the accumulated number of messages that can be consumed.
- `produced_deadletters` - Indicates the total number of dead letter messages generated by the consumer group.
- `available_deadletters` - Indicates the accumulated number of dead letter messages that have not been consumed.

» opentelekomcloud_vbs_backup_policy_v2

Provides an VBS Backup Policy resource.

» Example Usage

```
resource "opentelekomcloud_vbs_backup_policy_v2" "vbs" {
  name = "policy_002"
```

```
  start_time  = "12:00"
  status  = "ON"
  retain_first_backup = "N"
  rentention_num = 2
  frequency = 1
      tags =[
        {
          key = "k1"
          value = "v1"
          }]
}
```

» Argument Reference

The following arguments are supported:

- `name` (Required) - Specifies the policy name. The value is a string of 1 to 64 characters that can contain letters, digits, underscores (_), and hyphens (-). It cannot start with default.
- `start_time` (Required) - Specifies the start time of the backup job.The value is in the HH:mm format.
- `status` (Required) - Specifies the backup policy status. The value can ON or OFF.
- `retain_first_backup` (Required) - Specifies whether to retain the first backup in the current month. Possible values are Y or N.
- `rentention_num` (Required) - Specifies number of retained backups. Minimum value is 2.
- `frequency` (Required) - Specifies the backup interval. The value is in the range of 1 to 14 days.

tags - (Optional) Represents the list of tags to be configured for the backup policy.

- `key` - (Required) Specifies the tag key. A tag key consists of up to 36 characters, chosen from letters, digits, hyphens (-), and underscores (_).
- `value` - (Required) Specifies the tag value. A tag value consists of 0 to 43 characters, chosen from letters, digits, hyphens (-), and underscores (_).

» Attributes Reference

All of the argument attributes are also exported as result attributes:

- `id` - Specifies a backup policy ID.
- `policy_resource_count` - Specifies the number of volumes associated with the backup policy.

» Import

Backup Policy can be imported using the `id`, e.g.

```
$ terraform import opentelekomcloud_vbs_backup_policy_v2.vbs 4779ab1c-7c1a-44b1-a02e-93dfc36
```

» opentelekomcloud_vbs_backup_v2

Provides an VBS Backup resource.

» Example Usage

```
variable "backup_name" {}

variable "volume_id" {}

resource "opentelekomcloud_vbs_backup_v2" "mybackup" {
  volume_id = "${var.volume_id}"
  name = "${var.backup_name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) The name of the vbs backup. Changing the parameter will create new resource.
- `volume_id` - (Required) The id of the disk to be backed up. Changing the parameter will create new resource.
- `snapshot_id` - (Optional) The snapshot id of the disk to be backed up. Changing the parameter will create new resource.
- `description` - (Optional) The description of the vbs backup. Changing the parameter will create new resource.

tags - (Optional) List of tags to be configured for the backup resources. Changing the parameter will create new resource.

- `key` - (Required) Specifies the tag key.
- `value` - (Required) Specifies the tag value.

» Attributes Reference

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

- `id` - The id of the vbs backup.
- `container` - The container of the backup.
- `status` - The status of the VBS backup.
- `availability_zone` - The AZ where the backup resides.
- `size` - The size of the vbs backup.
- `service_metadata` - The metadata of the vbs backup.

» Import

VBS Backup can be imported using the `backup id`, e.g.

```
$ terraform import opentelekomcloud_vbs_backup_v2.mybackup 4779ab1c-7c1a-44b1-a02e-93dfc361
```

» opentelekomcloud_vbs_backup_share_v2

Provides an VBS Backup Share resource.

» Example Usage

```
variable "backup_id" {}

variable "to_project_ids" {}

resource "opentelekomcloud_vbs_backup_share_v2" "backupshare" {
  backup_id = "${var.backup_id}"
  to_project_ids = "${var.to_project_ids}"
}
```

» Argument Reference

The following arguments are supported:

- `backup_id` - (Required) The ID of the backup to be shared. Changing the parameter will create new resource.
- `to_project_ids` - (Required) The IDs of projects with which the backup is shared. Changing the parameter will create new resource.

» Attributes Reference

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

- `container` - The container of the backup.
- `backup_status` - The status of the VBS backup.
- `description` - The status of the VBS backup.
- `availability_zone` - The AZ where the backup resides.
- `size` - The size of the vbs backup.
- `backup_name` - The backup name.
- `snapshot_id` - The ID of the snapshot associated with the backup.
- `volume_id` - The ID of the tenant to which the backup belongs.
- `share_ids` - The backup share IDs.
- `service_metadata` - The metadata of the vbs backup.

» Import

VBS Backup Share can be imported using the `backup id`, e.g.

```
$ terraform import opentelekomcloud_vbs_backup_share_v2.backupshare 4779ab1c-7c1a-44b1-a02e
```