TelefonicaOpenCloud Provider

The TelefonicaOpenCloud provider is used to interact with the many resources supported by TelefonicaOpenCloud. The provider needs to be configured with the proper credentials before it can be used.

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

Example Usage

```
# Configure the TelefonicaOpenCloud Provider
provider "telefonicaopencloud" {
  user_name   = "admin"
  tenant_name = "admin"
  password    = "pwd"
  auth_url    = "http://myauthurl:5000/v2.0"
  region      = "RegionOne"
}

# Create a web server
resource "telefonicaopencloud_compute_instance_v2" "test-server" {
  # ...
}
```

Configuration Reference

The following arguments are supported:

- `access_key` - (Optional) The access key of the TelefonicaOpenCloud to use. If omitted, the `OS_ACCESS_KEY` environment variable is used.

- `secret_key` - (Optional) The secret key of the TelefonicaOpenCloud to use. If omitted, the `OS_SECRET_KEY` environment variable is used.

- `auth_url` - (Required) The Identity authentication URL. If omitted, the `OS_AUTH_URL` environment variable is used.

- `region` - (Optional) The region of the TelefonicaOpenCloud cloud to use. If omitted, the `OS_REGION_NAME` environment variable is used. If `OS_REGION_NAME` is not set, then no region will be used. It should be possible to omit the region in single-region TelefonicaOpenCloud environments, but this behavior may vary depending on the TelefonicaOpenCloud environment being used.

- `user_name` - (Optional) The Username to login with. If omitted, the `OS_USERNAME` environment variable is used.

- `user_id` - (Optional) The User ID to login with. If omitted, the `OS_USER_ID` environment variable is used.

- `tenant_id` - (Optional) The ID of the Tenant (Identity v2) or Project (Identity v3) to login with. If omitted, the `OS_TENANT_ID` or `OS_PROJECT_ID` environment variables are used.

- `tenant_name` - (Optional) The Name of the Tenant (Identity v2) or Project (Identity v3) to login with. If omitted, the `OS_TENANT_NAME` or `OS_PROJECT_NAME` environment variable are used.

- `password` - (Optional) The Password to login with. If omitted, the `OS_PASSWORD` environment variable is used.

- `token` - (Optional; Required if not using `user_name` and `password`) A token is an expiring, temporary means of access issued via the Keystone service. By specifying a token, you do not have to specify a username/password combination, since the token was already created by a username/password out of band of Terraform. If omitted, the `OS_AUTH_TOKEN` environment variable is used.

- `domain_id` - (Optional) The ID of the Domain to scope to (Identity v3). If If omitted, the following environment variables are checked (in this order): `OS_USER_DOMAIN_ID`, `OS_PROJECT_DOMAIN_ID`, `OS_DOMAIN_ID`.

- `domain_name` - (Optional) The Name of the Domain to scope to (Identity v3). If omitted, the following environment variables are checked (in this order): `OS_USER_DOMAIN_NAME`, `OS_PROJECT_DOMAIN_NAME`, `OS_DOMAIN_NAME`, `DEFAULT_DOMAIN`.

- `insecure` - (Optional) Trust self-signed SSL certificates. If omitted, the `OS_INSECURE` environment variable is used.

- `cacert_file` - (Optional) Specify a custom CA certificate when communicating over SSL. You can specify either a path to the file or the contents of the certificate. If omitted, the `OS_CACERT` environment variable is used.

- `cert` - (Optional) Specify client certificate file for SSL client authentication. You can specify either a path to the file or the contents of the certificate. If omitted the `OS_CERT` environment variable is used.

- `key` - (Optional) Specify client private key file for SSL client authentication. You can specify either a path to the file or the contents of the key. If omitted the `OS_KEY` environment variable is used.

- `endpoint_type` - (Optional) Specify which type of endpoint to use from the service catalog. It can be set using the `OS_ENDPOINT_TYPE` environment variable. If not set, public endpoints is used.

- `swauth` - (Optional) Set to `true` to authenticate against Swauth, a Swift-native authentication system. If omitted, the `OS_SWAUTH` environment variable is used. You must also set `username` to the Swauth/Swift username such as `username:project`. Set the `password` to the Swauth/Swift key. Finally, set `auth_url` as the location of the Swift service. Note that this will only work when used with the TelefonicaOpenCloud Object Storage resources.

- `use_octavia` - (Optional) If set to `true`, API requests will go the Load Balancer service (Octavia) instead of the Networking service (Neutron).

Additional Logging

This provider has the ability to log all HTTP requests and responses between Terraform and the TelefonicaOpenCloud cloud which is useful for troubleshooting and debugging.

To enable these logs, set the `OS_DEBUG` environment variable to 1 along with the usual `TF_LOG=DEBUG` environment variable:

```
$ OS_DEBUG=1 TF_LOG=DEBUG terraform apply
```

If you submit these logs with a bug report, please ensure any sensitive information has been scrubbed first!

Testing and Development

In order to run the Acceptance Tests for development, the following environment variables must also be set:

- `OS_REGION_NAME` - The region in which to create the server instance.

- `OS_IMAGE_ID` or `OS_IMAGE_NAME` - a UUID or name of an existing image in Glance.

- `OS_FLAVOR_ID` or `OS_FLAVOR_NAME` - an ID or name of an existing flavor.

- `OS_POOL_NAME` - The name of a Floating IP pool.

- `OS_NETWORK_ID` - The UUID of a network in your test environment.

- `OS_EXTGW_ID` - The UUID of the external gateway.

You should be able to use any TelefonicaOpenCloud environment to develop on as long as the above environment variables are set.

telefonicaopencloud_dns_zone_v2

Use this data source to get the ID of an available TelefonicaOpenCloud DNS zone.

Example Usage

```
data "telefonicaopencloud_dns_zone_v2" "zone_1" {
  name = "example.com"
}
```

Argument Reference

- `region` - (Optional) The region in which to obtain the V2 DNS client. A DNS client is needed to retrieve zone ids. If omitted, the `region` argument of the provider is used.

- `name` - (Optional) The name of the zone.

- `description` - (Optional) A description of the zone.

- `email` - (Optional) The email contact for the zone record.

- `status` - (Optional) The zone's status.

- `ttl` - (Optional) The time to live (TTL) of the zone.

- `type` - (Optional) The type of the zone. Can either be `PRIMARY` or `SECONDARY`.

Attributes Reference

`id` is set to the ID of the found zone. In addition, 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.

- `status` - See Argument Reference above.

- `attributes` - Attributes of the DNS Service scheduler.

- `masters` - An array of master DNS servers. When `type` is `SECONDARY`.

- `created_at` - The time the zone was created.

- `updated_at` - The time the zone was last updated.

- `transferred_at` - The time the zone was transferred.

- `version` - The version of the zone.

- `serial` - The serial number of the zone.

- `pool_id` - The ID of the pool hosting the zone.

- `project_id` - The project ID that owns the zone.

telefonicaopencloud_networking_network_v2

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

Example Usage

```
data "telefonicaopencloud_networking_network_v2" "network" {
  name = "tf_test_network"
}
```

Argument Reference

- `region` - (Optional) The region in which to obtain the V2 Neutron client. A Neutron client is needed to retrieve networks ids. If omitted, the `region` argument of the provider is used.

- `network_id` - (Optional) The ID of the network.

- `name` - (Optional) The name of the network.

- `status` - (Optional) The status of the network.

- `matching_subnet_cidr` - (Optional) The CIDR of a subnet within the network.

- `tenant_id` - (Optional) The owner of the network.

Attributes Reference

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

- `admin_state_up` - (Optional) The administrative state of the network.

- `name` - See Argument Reference above.

- `region` - See Argument Reference above.

- `shared` - (Optional) Specifies whether the network resource can be accessed by any tenant or not.

telefonicaopencloud_networking_secgroup_v2

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

Example Usage

```
data "telefonicaopencloud_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.

telefonicaopencloud_networking_subnet_v2

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

Example Usage

```
data "telefonicaopencloud_networking_subnet_v2" "subnet_1" {
  name = "subnet_1"
}
```

Argument Reference

- `region` - (Optional) The region in which to obtain the V2 Neutron client. A Neutron client is needed to retrieve subnet ids. If omitted, the `region` argument of the provider is used.

- `name` - (Optional) The name of the subnet.

- `dhcp_enabled` - (Optional) If the subnet has DHCP enabled.

- `dhcp_disabled` - (Optional) If the subnet has DHCP disabled.

- `ip_version` - (Optional) The IP version of the subnet (either 4 or 6).

- `gateway_ip` - (Optional) The IP of the subnet's gateway.

- `cidr` - (Optional) The CIDR of the subnet.

- `subnet_id` - (Optional) The ID of the subnet.

- `network_id` - (Optional) The ID of the network the subnet belongs to.

- `tenant_id` - (Optional) The owner of the subnet.

Attributes Reference

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

- `allocation_pools` - Allocation pools of the subnet.

- `enable_dhcp` - Whether the subnet has DHCP enabled or not.

- `dns_nameservers` - DNS Nameservers of the subnet.

- `host_routes` - Host Routes of the subnet.

- `region` - See Argument Reference above.

telefonicaopencloud_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 "telefonicaopencloud_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 version)

Attributes Reference

The following attributes are exported:

* `body` - Object data (see **limitations above** to understand cases in which this field is actually available)
* `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 thus what decoding mechanisms must be applied to obtain the media-type referenced by the Content-Type header field.
* `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 sum of the object content in case it's not encrypted)
* `expiration` - If the object expiration is configured (see [object lifecycle management](http://docs.telefonicaopencloud.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html)), the field includes this header. It includes the expiry-date and rule-id key value pairs providing object expiration information. The value of the rule-id is URL encoded.
* `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 2006 15:04:05 MST`)
* `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 Amazon S3-managed encryption key), this field includes the chosen encryption and algorithm used.
* `sse_kms_key_id` - If present, specifies the ID of the Key Management Service (KMS) master encryption key that was used for the object.
* `storage_class` - [Storage class](http://docs.aws.amazon.com/AmazonS3/latest/dev/storage-class-intro.html) information of the object. Available for all objects except for `Standard` storage class objects.
* `version_id` - The latest version ID of the object returned.
* `website_redirect_location` - If the bucket is configured as a website, redirects requests for this object to another object in the same bucket or to an external URL. Amazon S3 stores the value of this header in the object metadata.
* `tags` - A mapping of tags assigned to the object.

telefonicaopencloud_as_configuration_v1

Manages a V1 AS Configuration resource within TelefonicaOpenCloud.

Example Usage

Basic AS Configuration

```
resource "telefonicaopencloud_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 "telefonicaopencloud_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 "telefonicaopencloud_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`.

telefonicaopencloud_as_group_v1

Manages a V1 Autoscaling Group resource within TelefonicaOpenCloud.

Example Usage

Basic Autoscaling Group

```
resource "telefonicaopencloud_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 "telefonicaopencloud_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 "telefonicaopencloud_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 = "${telefonicaopencloud_elb_listener.my_listener.id}"
  delete_publicip = true
  delete_instances = "yes"
}

resource "telefonicaopencloud_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.

telefonicaopencloud_as_policy_v1

Manages a V1 AS Policy resource within TelefonicaOpenCloud.

Example Usage

AS Recurrence Policy

```
resource "telefonicaopencloud_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 "telefonicaopencloud_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 "telefonicaopencloud_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.

telefonicaopencloud_blockstorage_volume_v2

Manages a V2 volume resource within TelefonicaOpenCloud.

Example Usage

```
resource "telefonicaopencloud_blockstorage_volume_v2" "volume_1" {
  region      = "RegionOne"
  name        = "volume_1"
  description = "first test volume"
  size        = 3
}
```

Argument Reference

The following arguments are supported:

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

- `size` - (Required) The size of the volume to create (in gigabytes). Changing this creates a new volume.

- `availability_zone` - (Optional) The availability zone for the volume. Changing this creates a new volume.

- `consistency_group_id` - (Optional) The consistency group to place the volume in.

- `description` - (Optional) A description of the volume. Changing this updates the volume's description.

- `image_id` - (Optional) The image ID from which to create the volume. Changing this creates a new volume.

- `metadata` - (Optional) Metadata key/value pairs to associate with the volume. Changing this updates the existing volume metadata.

- `name` - (Optional) A unique name for the volume. Changing this updates the volume's name.

- `snapshot_id` - (Optional) The snapshot ID from which to create the volume. Changing this creates a new volume.

- `source_replica` - (Optional) The volume ID to replicate with.

- `source_vol_id` - (Optional) The volume ID from which to create the volume. Changing this creates a new volume.

- `volume_type` - (Optional) The type of volume to create. Changing this creates a new volume.

Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.

- `size` - See Argument Reference above.

- `name` - See Argument Reference above.

- `description` - See Argument Reference above.

- `availability_zone` - See Argument Reference above.

- `image_id` - See Argument Reference above.

- `source_vol_id` - See Argument Reference above.

- `snapshot_id` - See Argument Reference above.

- `metadata` - See Argument Reference above.

- `volume_type` - See Argument Reference above.

- `attachment` - If a volume is attached to an instance, this attribute will display the Attachment ID, Instance ID, and the Device as the Instance sees it.

Import

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

```
$ terraform import telefonicaopencloud_blockstorage_volume_v2.volume_1 ea257959-eeb1-4c10-8d33-26f0409a755d
```

telefonicaopencloud_ces_alarmrule

Manages an alarm rule resource within telefonica open cloud.

Example Usage

```
resource "telefonicaopencloud_ces_alarmrule" "alarm_rule" {
  "alarm_name" = "alarm_rule"
  "metric" {
    "namespace" = "SYS.ECS"
    "metric_name" = "network_outgoing_bytes_rate_inband"
    "dimensions" {
        "name" = "instance_id"
        "value" = "12232415-6cc9-4c80-83ff-55a6613d14c3"
    }
  }
  "condition"  {
    "period" = 300
    "filter" = "average"
    "comparison_operator" = ">"
    "value" = 6
    "unit" = "B/s"
    "count" = 1
  }
  "alarm_actions" {
    "type" = "notification"
    "notification_list" = [
      "urn:smn:southchina:68438a86d98e427e907e0097b7e35d47:sd"
    ]
  }
}
```

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.

telefonicaopencloud_compute_floatingip_associate_v2

Associate a floating IP to an instance. This can be used instead of the `floating_ip` options in `telefonicaopencloud_compute_instance_v2`.

Example Usage

Automatically detect the correct network

```
resource "telefonicaopencloud_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 "telefonicaopencloud_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "telefonicaopencloud_compute_floatingip_associate_v2" "fip_1" {
  floating_ip = "${telefonicaopencloud_networking_floatingip_v2.fip_1.address}"
  instance_id = "${telefonicaopencloud_compute_instance_v2.instance_1.id}"
}
```

Explicitly set the network to attach to

```
resource "telefonicaopencloud_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 "telefonicaopencloud_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "telefonicaopencloud_compute_floatingip_associate_v2" "fip_1" {
  floating_ip = "${telefonicaopencloud_networking_floatingip_v2.fip_1.address}"
  instance_id = "${telefonicaopencloud_compute_instance_v2.instance_1.id}"
  fixed_ip    = "${telefonicaopencloud_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 telefonicaopencloud_compute_floatingip_associate_v2.fip_1 <floating_ip>/<instance_id>/<fixed_ip>
```

telefonicaopencloud_compute_floatingip_v2

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

Please note that managing floating IPs through the TelefonicaOpenCloud Compute API has been deprecated. Unless you are using an older TelefonicaOpenCloud environment, it is recommended to use the `telefonicaopencloud_networking_floatingip_v2` (/docs/providers/telefonicaopencloud/r/networking_floatingip_v2.html) resource instead, which uses the TelefonicaOpenCloud Networking API.

Example Usage

```
resource "telefonicaopencloud_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 telefonicaopencloud_compute_floatingip_v2.floatip_1 89c60255-9bd6-460c-822a-e2b959ede9
d2
```

telefonicaopencloud_compute_instance_v2

Manages a V2 VM instance resource within TelefonicaOpenCloud.

Example Usage

Basic Instance

```
resource "telefonicaopencloud_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"]

  metadata {
    this = "that"
  }

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

Instance With Attached Volume

```
resource "telefonicaopencloud_blockstorage_volume_v2" "myvol" {
  name = "myvol"
  size = 1
}

resource "telefonicaopencloud_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 "telefonicaopencloud_compute_volume_attach_v2" "attached" {
  compute_id = "${telefonicaopencloud_compute_instance_v2.myinstance.id}"
  volume_id = "${telefonicaopencloud_blockstorage_volume_v2.myvol.id}"
}
```

Boot From Volume

```
resource "telefonicaopencloud_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 "telefonicaopencloud_blockstorage_volume_v1" "myvol" {
  name    = "myvol"
  size    = 5
  image_id = "<image-id>"
}

resource "telefonicaopencloud_compute_instance_v2" "boot-from-volume" {
  name            = "bootfromvolume"
  flavor_id       = "3"
  key_pair        = "my_key_pair_name"
  security_groups = ["default"]

  block_device {
    uuid                 = "${telefonicaopencloud_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 "telefonicaopencloud_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 "telefonicaopencloud_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}

resource "telefonicaopencloud_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                  = "${telefonicaopencloud_blockstorage_volume_v2.volume_1.id}"
    source_type           = "volume"
    destination_type      = "volume"
    boot_index            = 1
    delete_on_termination = true
  }
}
```

Instance With Multiple Networks

```
resource "telefonicaopencloud_networking_floatingip_v2" "myip" {
  pool = "my_pool"
}

resource "telefonicaopencloud_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 "telefonicaopencloud_compute_floatingip_associate_v2" "myip" {
  floating_ip = "${telefonicaopencloud_networking_floatingip_v2.myip.address}"
  instance_id = "${telefonicaopencloud_compute_instance_v2.multi-net.id}"
  fixed_ip = "${telefonicaopencloud_compute_instance_v2.multi-net.network.1.fixed_ip_v4}"
}
```

Instance With Personality

```
resource "telefonicaopencloud_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 "telefonicaopencloud_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 "telefonicaopencloud_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.example.com"

  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. 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 (http://docs.telefonicaopencloud.org/developer/nova/block_device_mapping.html) for more information.

- `scheduler_hints` - (Optional) Provide the Nova scheduler with hints on how the instance should be launched. The available hints are described below.

- `personality` - (Optional) Customize the personality of an instance by defining one or more files and their contents. The personality structure is described below.

- `stop_before_destroy` - (Optional) Whether to try stop instance gracefully before destroying it, thus giving chance for guest OS daemons to stop correctly. If instance doesn't stop within timeout, it will be destroyed anyway.

The `network` block supports:

- `uuid` - (Required unless `port` or `name` is provided) The network UUID to attach to the server. Changing this creates a

new server.

- `name` - (Required unless `uuid` or `port` is provided) The human-readable name of the network. Changing this creates a new server.

- `port` - (Required unless `uuid` or `name` is provided) The port UUID of a network to attach to the server. Changing this creates a new server.

- `fixed_ip_v4` - (Optional) Specifies a fixed IPv4 address to be used on this network. Changing this creates a new server.

- `fixed_ip_v6` - (Optional) Specifies a fixed IPv6 address to be used on this network. Changing this creates a new server.

- `access_network` - (Optional) Specifies if this network should be used for provisioning access. Accepts true or false. Defaults to false.

The `block_device` block supports:

- `uuid` - (Required unless `source_type` is set to `"blank"`) The UUID of the image, volume, or snapshot. Changing this creates a new server.

- `source_type` - (Required) The source type of the device. Must be one of "blank", "image", "volume", or "snapshot". Changing this creates a new server.

- `volume_size` - The size of the volume to create (in gigabytes). Required in the following combinations: source=image and destination=volume, source=blank and destination=local, and source=blank and destination=volume. Changing this creates a new server.

- `boot_index` - (Optional) The boot index of the volume. It defaults to 0. Changing this creates a new server.

- `destination_type` - (Optional) The type that gets created. Possible values are "volume" and "local". Changing this creates a new server.

- `delete_on_termination` - (Optional) Delete the volume / block device upon termination of the instance. Defaults to false. Changing this creates a new server.

The `scheduler_hints` block supports:

- `group` - (Optional) A UUID of a Server Group. The instance will be placed into that group.

- `different_host` - (Optional) A list of instance UUIDs. The instance will be scheduled on a different host than all other instances.

- `same_host` - (Optional) A list of instance UUIDs. The instance will be scheduled on the same host of those specified.

- `query` - (Optional) A conditional query that a compute node must pass in order to host an instance.

- `target_cell` - (Optional) The name of a cell to host the instance.

- `build_near_host_ip` - (Optional) An IP Address in CIDR form. The instance will be placed on a compute node that is in the same subnet.

The `personality` block supports:

- `file` - (Required) The absolute path of the destination file.

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

Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.

- `name` - See Argument Reference above.

- `access_ip_v4` - The first detected Fixed IPv4 address *or* the Floating IP.

- `access_ip_v6` - The first detected Fixed IPv6 address.

- `metadata` - See Argument Reference above.

- `security_groups` - See Argument Reference above.

- `flavor_id` - See Argument Reference above.

- `flavor_name` - See Argument Reference above.

- `network/uuid` - See Argument Reference above.

- `network/name` - See Argument Reference above.

- `network/port` - See Argument Reference above.

- `network/fixed_ip_v4` - The Fixed IPv4 address of the Instance on that network.

- `network/fixed_ip_v6` - The Fixed IPv6 address of the Instance on that network.

- `network/mac` - The MAC address of the NIC on that network.

- `all_metadata` - Contains all instance metadata, even metadata not set by Terraform.

Notes

Multiple Ephemeral Disks

It's possible to specify multiple `block_device` entries to create an instance with multiple ephemeral (local) disks. In order to create multiple ephemeral disks, the sum of the total amount of ephemeral space must be less than or equal to what the chosen flavor supports.

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

```
resource "telefonicaopencloud_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:

- In TelefonicaOpenCloud environments prior to the Kilo release, deleting or recreating an Instance will cause the Instance's Port(s) to be deleted. One way of working around this is to taint any Port(s) used in Instances which are to be recreated. See here (https://review.telefonicaopencloud.org/#/c/126309/) for further information.

- 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 TelefonicaOpenCloud 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 "telefonicaopencloud_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 "telefonicaopencloud_compute_instance_v2" "instance_1" {
  name = "instance_1"

  network {
    port = "${telefonicaopencloud_networking_port_v2.port_1.id}"
  }

  connection {
    user        = "root"
    host        = "${telefonicaopencloud_networking_port_v2.port_1.fixed_ip.0.ip_address}"
    private_key = "~/path/to/key"
  }

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

telefonicaopencloud_compute_keypair_v2

Manages a V2 keypair resource within TelefonicaOpenCloud.

Example Usage

```
resource "telefonicaopencloud_compute_keypair_v2" "test-keypair" {
  name       = "my-keypair"
  public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDAjpC1hwiOCCmKEWxJ4qzTTsJbKzndLotBCz5PcwtUnflmU+gHJ
tWMZKpuEGVi29h0A/+ydKek1O18k10Ff+4tyFjiHDQAnOfgWf7+b1yK+qDip3X1C0UPMbwHlTfSGWLGZqd9LvEFx9k3h/M+VtMvwR1lJ9
LUyTAImnNjWG7TaIPmui30HvM2UiFEmqkr4ijq45MyX2+fLIePLRIF61p4whjHAQYufqyno3BS48icQb4p6iVEZPo4AE2o9oIyQvj2mx4
dk5Y8CgSETOZTYDOR3rU2fZTRDRgPJDH9FWvQjF5tA0p3d9CoWWd2s6GKKbfoUIi8R/Db1BSPJwkqB"
}
```

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 telefonicaopencloud_compute_keypair_v2.my-keypair test-keypair
```

telefonicaopencloud_compute_secgroup_v2

Manages a V2 security group resource within TelefonicaOpenCloud.

Please note that managing security groups through the TelefonicaOpenCloud Compute API has been deprecated. Unless you are using an older TelefonicaOpenCloud environment, it is recommended to use the `telefonicaopencloud_networking_secgroup_v2` (/docs/providers/telefonicaopencloud/r/networking_secgroup_v2.html) and `telefonicaopencloud_networking_secgroup_rule_v2` (/docs/providers/telefonicaopencloud/r/networking_secgroup_rule_v2.html) resources instead, which uses the TelefonicaOpenCloud Networking API.

Example Usage

```
resource "telefonicaopencloud_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

(https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol#Control_messages).

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 "telefonicaopencloud_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 = ["${telefonicaopencloud_compute_secgroup_v2.secgroup_1.name}"]
}
```

Import

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

```
$ terraform import telefonicaopencloud_compute_secgroup_v2.my_secgroup 1bc30ee9-9d5b-4c30-bdd5-7f1e663f5e
df
```

telefonicaopencloud_compute_servergroup_v2

Manages a V2 Server Group resource within TelefonicaOpenCloud.

Example Usage

```
resource "telefonicaopencloud_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 telefonicaopencloud_compute_servergroup_v2.test-sg 1bc30ee9-9d5b-4c30-bdd5-7f1e663f5ed
f
```

telefonicaopencloud_compute_volume_attach_v2

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

Example Usage

Basic attachment of a single volume to a single instance

```
resource "telefonicaopencloud_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}

resource "telefonicaopencloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "telefonicaopencloud_compute_volume_attach_v2" "va_1" {
  instance_id = "${telefonicaopencloud_compute_instance_v2.instance_1.id}"
  volume_id   = "${telefonicaopencloud_blockstorage_volume_v2.volume_1.id}"
}
```

Attaching multiple volumes to a single instance

```
resource "telefonicaopencloud_blockstorage_volume_v2" "volumes" {
  count = 2
  name  = "${format("vol-%02d", count.index + 1)}"
  size  = 1
}

resource "telefonicaopencloud_compute_instance_v2" "instance_1" {
  name            = "instance_1"
  security_groups = ["default"]
}

resource "telefonicaopencloud_compute_volume_attach_v2" "attachments" {
  count       = 2
  instance_id = "${telefonicaopencloud_compute_instance_v2.instance_1.id}"
  volume_id   = "${element(telefonicaopencloud_blockstorage_volume_v2.volumes.*.id, count.index)}"
}

output "volume devices" {
  value = "${telefonicaopencloud_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 telefonicaopencloud_compute_volume_attach_v2.va_1 89c60255-9bd6-460c-822a-e2b959ede9d2
/45670584-225f-46c3-b33e-6707b589b666
```

telefonicaopencloud_dns_recordset_v2

Manages a DNS record set in the TelefonicaOpenCloud DNS Service.

Example Usage

Automatically detect the correct network

```
resource "telefonicaopencloud_dns_zone_v2" "example_zone" {
  name = "example.com."
  email = "email2@example.com"
  description = "a zone"
  ttl = 6000
  type = "PRIMARY"
}

resource "telefonicaopencloud_dns_recordset_v2" "rs_example_com" {
  zone_id = "${telefonicaopencloud_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 telefonicaopencloud_dns_recordset_v2.recordset_1 <zone_id>/<recordset_id>
```

telefonicaopencloud_dns_zone_v2

Manages a DNS zone in the TelefonicaOpenCloud DNS Service.

Example Usage

Automatically detect the correct network

```
resource "telefonicaopencloud_dns_zone_v2" "example.com" {
  name = "example.com."
  email = "jdoe@example.com"
  description = "An example zone"
  ttl = 3000
  type = "PRIMARY"
}
```

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 `PRIMARY` or `SECONDARY`. Changing this creates a new zone.

- `attributes` - (Optional) Attributes for the DNS Service scheduler. Changing this creates a new zone.

- `ttl` - (Optional) The time to live (TTL) of the zone.

- `description` - (Optional) A description of the zone.

- `masters` - (Optional) An array of master DNS servers. For when `type` is `SECONDARY`.

- `value_specs` - (Optional) Map of additional options. Changing this creates a new 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.

- `attributes` - 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 telefonicaopencloud_dns_zone_v2.zone_1 <zone_id>
```

telefonicaopencloud_elb_backendecs

Manages an elastic loadbalancer backendecs resource within telefonica open cloud.

Example Usage

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

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

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

Argument Reference

The following arguments are supported:

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

- `server_id` - (Required) Specifies the backend member ID.

- `private_address` - (Required) Specifies the private IP address of the backend member.

Attributes Reference

The following attributes are exported:

- `listener_id` - See Argument Reference above.

- `server_id` - See Argument Reference above.

- `private_address` - See Argument Reference above.

- `public_address` - Specifies the floating IP address assigned to the backend member.

- `id` - Specifies the backend member ID.

- `status` - Specifies the backend ECS status. The value is ACTIVE, PENDING, or ERROR.

- `health_status` - Specifies the health check status. The value is NORMAL, ABNORMAL, or UNAVAILABLE.

- `update_time` - Specifies the time when information about the backend member was updated.

- `create_time` - Specifies the time when the backend member was created.

- `server_name` - Specifies the backend member name.

- `listeners` - Specifies the listener to which the backend member belongs.

telefonicaopencloud_elb_healthcheck

Manages an elastic loadbalancer healthcheck resource within telefonica open cloud.

Example Usage

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

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

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

Argument Reference

The following arguments are supported:

- `listener_id` - (Required) Specifies the ID of the listener to which the health check task belongs.

- `healthcheck_protocol` - (Optional) Specifies the protocol used for the health check. The value can be HTTP or TCP (case-insensitive).

- `healthcheck_uri` - (Optional) Specifies the URI for health check. This parameter is valid when healthcheck_ protocol is HTTP. The value is a string of 1 to 80 characters that must start with a slash (/) and can only contain letters, digits, and special characters, such as -/.%?#&.

- `healthcheck_connect_port` - (Optional) Specifies the port used for the health check. The value ranges from 1 to 65535.

- `healthy_threshold` - (Optional) Specifies the threshold at which the health check result is success, that is, the number of consecutive successful health checks when the health check result of the backend server changes from fail to success. The value ranges from 1 to 10.

- `unhealthy_threshold` - (Optional) Specifies the threshold at which the health check result is fail, that is, the number of consecutive failed health checks when the health check result of the backend server changes from success to fail. The value ranges from 1 to 10.

- `healthcheck_timeout` - (Optional) Specifies the maximum timeout duration (s) for the health check. The value ranges from 1 to 50.

- `healthcheck_interval` - (Optional) Specifies the maximum interval (s) for health check. The value ranges from 1 to 5.

Attributes Reference

The following attributes are exported:

- `listener_id` - See Argument Reference above.

- `healthcheck_protocol` - See Argument Reference above.

- `healthcheck_uri` - See Argument Reference above.

- `healthcheck_connect_port` - See Argument Reference above.

- `healthy_threshold` - See Argument Reference above.

- `unhealthy_threshold` - See Argument Reference above.

- `healthcheck_timeout` - See Argument Reference above.

- `healthcheck_interval` - See Argument Reference above.

- `id` - Specifies the health check task ID.

- `update_time` - Specifies the time when information about the health check task was updated.

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

telefonicaopencloud_elb_listener

Manages an elastic loadbalancer listener resource within telefonica open cloud.

Example Usage

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

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

Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the load balancer name. The name is a string of 1 to 64 characters that consist of letters, digits, underscores (_), and hyphens (-).

- `description` - (Optional) Provides supplementary information about the listener. The value is a string of 0 to 128 characters and cannot be <>.

- `loadbalancer_id` - (Required) Specifies the ID of the load balancer to which the listener belongs.

- `protocol` - (Required) Specifies the listening protocol used for layer 4 or 7. The value can be HTTP, TCP, HTTPS, or UDP.

- `port` - (Required) Specifies the listening port. The value ranges from 1 to 65535.

- `backend_protocol` - (Required) Specifies the backend protocol. If the value of protocol is UDP, the value of this parameter can only be UDP. The value can be HTTP, TCP, or UDP.

- `backend_port` - (Required) Specifies the backend port. The value ranges from 1 to 65535.

- `lb_algorithm` - (Required) Specifies the load balancing algorithm for the listener. The value can be roundrobin, leastconn, or source.

- `session_sticky` - (Optional) Specifies whether to enable sticky session. The value can be true or false. The Sticky session is enabled when the value is true, and is disabled when the value is false. If the value of protocol is HTTP, HTTPS, or TCP, and the value of lb_algorithm is not roundrobin, the value of this parameter can only be false.

- `sticky_session_type` - (Optional) Specifies the cookie processing method. The value is insert. insert indicates that the cookie is inserted by the load balancer. This parameter is valid when protocol is set to HTTP, and session_sticky to true. The default value is insert. This parameter is invalid when protocol is set to TCP or UDP, which means the parameter is empty.

- `cookie_timeout` - (Optional) Specifies the cookie timeout period (minutes). This parameter is valid when protocol is set to HTTP, session_sticky to true, and sticky_session_type to insert. This parameter is invalid when protocol is set to TCP or UDP. The value ranges from 1 to 1440.

- `tcp_timeout` - (Optional) Specifies the TCP timeout period (minutes). This parameter is valid when protocol is set to TCP. The value ranges from 1 to 5.

- `tcp_draining` - (Optional) Specifies whether to maintain the TCP connection to the backend ECS after the ECS is deleted. This parameter is valid when protocol is set to TCP. The value can be true or false.

- `tcp_draining_timeout` - (Optional) Specifies the timeout duration (minutes) for the TCP connection to the backend ECS after the ECS is deleted. This parameter is valid when protocol is set to TCP, and tcp_draining to true. The value ranges from 0 to 60.

- `certificate_id` - (Optional) Specifies the ID of the SSL certificate used for security authentication when HTTPS is used to make API calls. This parameter is mandatory if the value of protocol is HTTPS. The value can be obtained by viewing the details of the SSL certificate.

- `udp_timeout` - (Optional) Specifies the UDP timeout duration (minutes). This parameter is valid when protocol is set to UDP. The value ranges from 1 to 1440.

- `ssl_protocols` - (Optional) Specifies the SSL protocol standard supported by a tracker, which is used for enabling specified encryption protocols. This parameter is valid only when the value of protocol is set to HTTPS. The value is TLSv1.2 or TLSv1.2 TLSv1.1 TLSv1. The default value is TLSv1.2.

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

Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.

- `description` - See Argument Reference above.

- `loadbalancer_id` - See Argument Reference above.

- `protocol` - See Argument Reference above.

- `port` - See Argument Reference above.

- `backend_protocol` - See Argument Reference above.

- `backend_port` - See Argument Reference above.

- `lb_algorithm` - See Argument Reference above.

- `session_sticky` - See Argument Reference above.

- `sticky_session_type` - See Argument Reference above.

- `cookie_timeout` - See Argument Reference above.

- `tcp_timeout` - See Argument Reference above.

- `tcp_draining` - See Argument Reference above.

- `tcp_draining_timeout` - See Argument Reference above.

- `certificate_id` - See Argument Reference above.

- `udp_timeout` - See Argument Reference above.

- `ssl_protocols` - See Argument Reference above.

- `ssl_ciphers` - See Argument Reference above.

- `update_time` - Specifies the time when information about the listener was updated.

- `id` - Specifies the listener ID.

- `create_time` - Specifies the time when the listener was created.

- `status` - Specifies the listener status. The value can be ACTIVE, PENDING_CREATE, or ERROR.

- `admin_state_up` - Specifies the status of the load balancer. Value range: false: The load balancer is disabled. true: The load balancer runs properly.

- `member_number` - Specifies the number of backend members.

- `healthcheck_id` - Specifies the health check task ID.

telefonicaopencloud_elb_loadbalancer

Manages an elastic loadbalancer resource within telefonica open cloud.

Example Usage

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

Argument Reference

The following arguments are supported:

- `name` - (Required) Specifies the load balancer name. The name is a string of 1 to 64 characters that consist of letters, digits, underscores (_), and hyphens (-).

- `description` - (Optional) Provides supplementary information about the listener. The value is a string of 0 to 128 characters and cannot be <>.

- `vpc_id` - (Required) Specifies the VPC ID.

- `bandwidth` - (Optional) Specifies the bandwidth (Mbit/s). This parameter is mandatory when type is set to External, and it is invalid when type is set to Internal. The value ranges from 1 to 300.

- `type` - (Required) Specifies the load balancer type. The value can be Internal or External.

- `admin_state_up` - (Required) Specifies the status of the load balancer. Value range: 0 or false: indicates that the load balancer is stopped. Only tenants are allowed to enter these two values. 1 or true: indicates that the load balancer is running properly. 2 or false: indicates that the load balancer is frozen. Only tenants are allowed to enter these two values.

- `vip_subnet_id` - (Optional) Specifies the ID of the private network to be added. This parameter is mandatory when type is set to Internal, and it is invalid when type is set to External.

- `az` - (Optional) Specifies the ID of the availability zone (AZ). This parameter is mandatory when type is set to Internal, and it is invalid when type is set to External.

- `charge_mode` - (Optional) This is a reserved field. If the system supports charging by traffic and this field is specified, then you are charged by traffic for elastic IP addresses. The value is traffic.

- `eip_type` - (Optional) This parameter is reserved.

- `security_group_id` - (Optional) Specifies the security group ID. The value is a string of 1 to 200 characters that consists of uppercase and lowercase letters, digits, and hyphens (-). This parameter is mandatory only when type is set to Internal.

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

- `tenantid` - (Optional) Specifies the tenant ID. This parameter is mandatory only when type is set to Internal.

Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.

- `description` - See Argument Reference above.

- `vpc_id` - See Argument Reference above.

- `bandwidth` - See Argument Reference above.

- `type` - See Argument Reference above.

- `admin_state_up` - See Argument Reference above.

- `vip_subnet_id` - See Argument Reference above.

- `az` - See Argument Reference above.

- `charge_mode` - See Argument Reference above.

- `eip_type` - See Argument Reference above.

- `security_group_id` - See Argument Reference above.

- `vip_address` - See Argument Reference above.

- `tenantid` - See Argument Reference above.

- `update_time` - Specifies the time when information about the load balancer was updated.

- `create_time` - Specifies the time when the load balancer was created.

- `id` - Specifies the load balancer ID.

- `status` - Specifies the status of the load balancer. The value can be ACTIVE, PENDING_CREATE, or ERROR.

telefonicaopencloud_networking_floatingip_v2

Manages a V2 floating IP resource within TelefonicaOpenCloud 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 "telefonicaopencloud_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 telefonicaopencloud_networking_floatingip_v2.floatip_1 2c7f39f3-702b-48d1-940c-b503841
77ee1
```

telefonicaopencloud_networking_network_v2

Manages a V2 Neutron network resource within TelefonicaOpenCloud.

Example Usage

```
resource "telefonicaopencloud_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "telefonicaopencloud_networking_subnet_v2" "subnet_1" {
  name       = "subnet_1"
  network_id = "${telefonicaopencloud_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "telefonicaopencloud_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 "telefonicaopencloud_networking_port_v2" "port_1" {
  name              = "port_1"
  network_id        = "${telefonicaopencloud_networking_network_v2.network_1.id}"
  admin_state_up    = "true"
  security_group_ids = ["${telefonicaopencloud_compute_secgroup_v2.secgroup_1.id}"]

  fixed_ip {
    "subnet_id"  = "${telefonicaopencloud_networking_subnet_v2.subnet_1.id}"
    "ip_address" = "192.168.199.10"
  }
}

resource "telefonicaopencloud_compute_instance_v2" "instance_1" {
  name             = "instance_1"
  security_groups = ["${telefonicaopencloud_compute_secgroup_v2.secgroup_1.name}"]

  network {
    port = "${telefonicaopencloud_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 telefonicaopencloud_networking_network_v2.network_1 d90ce693-5ccf-4136-a0ed-152ce412b6
b9
```

telefonicaopencloud_networking_port_v2

Manages a V2 port resource within TelefonicaOpenCloud.

Example Usage

```
resource "telefonicaopencloud_networking_network_v2" "network_1" {
  name          = "network_1"
  admin_state_up = "true"
}

resource "telefonicaopencloud_networking_port_v2" "port_1" {
  name          = "port_1"
  network_id    = "${telefonicaopencloud_networking_network_v2.network_1.id}"
  admin_state_up = "true"
}
```

Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a port. If omitted, the `region` argument of the provider is used. Changing this creates a new port.

- `name` - (Optional) A unique name for the port. Changing this updates the `name` of an existing port.

- `network_id` - (Required) The ID of the network to attach the port to. Changing this creates a new port.

- `admin_state_up` - (Optional) Administrative up/down status for the port (must be "true" or "false" if provided). Changing this updates the `admin_state_up` of an existing port.

- `mac_address` - (Optional) Specify a specific MAC address for the port. Changing this creates a new port.

- `tenant_id` - (Optional) The owner of the Port. Required if admin wants to create a port for another tenant. Changing this creates a new port.

- `device_owner` - (Optional) The device owner of the Port. Changing this creates a new port.

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

- `no_security_groups` - (Optional - Conflicts with `security_group_ids`) If set to `true`, then no security groups are applied to the port. If set to `false` and no `security_group_ids` are specified, then the Port will yield to the default behavior of the Networking service, which is to usually apply the "default" security group.

- `device_id` - (Optional) The ID of the device attached to the port. Changing this creates a new port.

- `fixed_ip` - (Optional) An array of desired IPs for this port. The structure is described below.

- `allowed_address_pairs` - (Optional) An IP/MAC Address pair of additional IP addresses that can be active on this port. The structure is described below.

- `value_specs` - (Optional) Map of additional options.

The `fixed_ip` block supports:

- `subnet_id` - (Required) Subnet in which to allocate IP address for this port.

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

The `allowed_address_pairs` block supports:

- `ip_address` - (Required) The additional IP address.

- `mac_address` - (Optional) The additional MAC address.

Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.

- `admin_state_up` - See Argument Reference above.

- `mac_address` - See Argument Reference above.

- `tenant_id` - See Argument Reference above.

- `device_owner` - See Argument Reference above.

- `security_group_ids` - See Argument Reference above.

- `device_id` - See Argument Reference above.

- `fixed_ip` - See Argument Reference above.

- `all_fixed_ips` - The collection of Fixed IP addresses on the port in the order returned by the Network v2 API.

- `all_security_group_ids` - The collection of Security Group IDs on the port which have been explicitly and implicitly added.

Import

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

```
$ terraform import telefonicaopencloud_networking_port_v2.port_1 eae26a3e-1c33-4cc1-9c31-0cd729c438a1
```

Notes

Ports and Instances

There are some notes to consider when connecting Instances to networks using Ports. Please see the `telefonicaopencloud_compute_instance_v2` documentation for further documentation.

telefonicaopencloud_networking_router_interface_v2

Manages a V2 router interface resource within TelefonicaOpenCloud.

Example Usage

```
resource "telefonicaopencloud_networking_network_v2" "network_1" {
  name          = "tf_test_network"
  admin_state_up = "true"
}

resource "telefonicaopencloud_networking_subnet_v2" "subnet_1" {
  network_id = "${telefonicaopencloud_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "telefonicaopencloud_networking_router_v2" "router_1" {
  name             = "my_router"
  external_gateway = "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"
}

resource "telefonicaopencloud_networking_router_interface_v2" "router_interface_1" {
  router_id = "${telefonicaopencloud_networking_router_v2.router_1.id}"
  subnet_id = "${telefonicaopencloud_networking_subnet_v2.subnet_1.id}"
}
```

Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a router. If omitted, the `region` argument of the provider is used. Changing this creates a new router interface.

- `router_id` - (Required) ID of the router this interface belongs to. Changing this creates a new router interface.

- `subnet_id` - ID of the subnet this interface connects to. Changing this creates a new router interface.

- `port_id` - ID of the port this interface connects to. Changing this creates a new router interface.

Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.

- `router_id` - See Argument Reference above.

- `subnet_id` - See Argument Reference above.

- `port_id` - See Argument Reference above.

Import

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

```
$ telefonicaopencloud port list --router <router name or id>
$ terraform import telefonicaopencloud_networking_router_interface_v2.int_1 <port id from above output>
```

telefonicaopencloud_networking_router_route_v2

Creates a routing entry on a TelefonicaOpenCloud V2 router.

Example Usage

```
resource "telefonicaopencloud_networking_router_v2" "router_1" {
  name          = "router_1"
  admin_state_up = "true"
}

resource "telefonicaopencloud_networking_network_v2" "network_1" {
  name          = "network_1"
  admin_state_up = "true"
}

resource "telefonicaopencloud_networking_subnet_v2" "subnet_1" {
  network_id = "${telefonicaopencloud_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "telefonicaopencloud_networking_router_interface_v2" "int_1" {
  router_id = "${telefonicaopencloud_networking_router_v2.router_1.id}"
  subnet_id = "${telefonicaopencloud_networking_subnet_v2.subnet_1.id}"
}

resource "telefonicaopencloud_networking_router_route_v2" "router_route_1" {
  depends_on       = ["telefonicaopencloud_networking_router_interface_v2.int_1"]
  router_id        = "${telefonicaopencloud_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 `telefonicaopencloud_networking_router_route_v2` resource creation time. You can ensure that by explicitly specifying a dependency on the `telefonicaopencloud_networking_router_interface_v2` resource that connects the next hop to the router, as in the example above.

Import

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

```
$ terraform import telefonicaopencloud_networking_router_route_v2.router_route_1 686fe248-386c-4f70-9f6c-
281607dad079-route-10.0.1.0/24-192.168.199.25
```

telefonicaopencloud_networking_router_v2

Manages a V2 router resource within TelefonicaOpenCloud.

Example Usage

```
resource "telefonicaopencloud_networking_router_v2" "router_1" {
  name                = "my_router"
  admin_state_up      = true
  external_network_id = "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"
}
```

Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a router. If omitted, the `region` argument of the provider is used. Changing this creates a new router.

- `name` - (Optional) A unique name for the router. Changing this updates the `name` of an existing router.

- `admin_state_up` - (Optional) Administrative up/down status for the router (must be "true" or "false" if provided). Changing this updates the `admin_state_up` of an existing router.

- `distributed` - (Optional) Indicates whether or not to create a distributed router. The default policy setting in Neutron restricts usage of this property to administrative users only.

- `external_gateway` - (Deprecated - use `external_network_id` instead) 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.

- `external_network_id` - (Optional) The network UUID of an external gateway for the router. A router with an external gateway is required if any compute instances or load balancers will be using floating IPs. Changing this updates the external gateway of the router.

- `enable_snat` - (Optional) Enable Source NAT for the router. Valid values are "true" or "false". An `external_network_id` has to be set in order to set this property. Changing this updates the `enable_snat` of the router.

- `external_fixed_ip` - (Optional) An external fixed IP for the router. This can be repeated. The structure is described below. An `external_network_id` has to be set in order to set this property. Changing this updates the external fixed IPs of the router.

- `tenant_id` - (Optional) The owner of the floating IP. Required if admin wants to create a router for another tenant. Changing this creates a new router.

- `value_specs` - (Optional) Map of additional driver-specific options.

The `external_fixed_ip` block supports:

- `subnet_id` - (Optional) Subnet in which the fixed IP belongs to.

- `ip_address` - (Optional) The IP address to set on the router.

Attributes Reference

The following attributes are exported:

- `id` - ID of the router.

- `region` - See Argument Reference above.

- `name` - See Argument Reference above.

- `admin_state_up` - See Argument Reference above.

- `external_gateway` - See Argument Reference above.

- `external_network_id` - See Argument Reference above.

- `enable_snat` - See Argument Reference above.

- `external_fixed_ip` - See Argument Reference above.

- `tenant_id` - See Argument Reference above.

- `value_specs` - See Argument Reference above.

Import

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

```
$ terraform import telefonicaopencloud_networking_router_v2.router_1 014395cd-89fc-4c9b-96b7-13d1ee79dad2
```

telefonicaopencloud_networking_secgroup_rule_v2

Manages a V2 neutron security group rule resource within TelefonicaOpenCloud. 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 "telefonicaopencloud_networking_secgroup_v2" "secgroup_1" {
  name        = "secgroup_1"
  description = "My neutron security group"
}

resource "telefonicaopencloud_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 = "${telefonicaopencloud_networking_secgroup_v2.secgroup_1.id}"
}
```

Argument Reference

The following arguments are supported:

- `region` - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a port. If omitted, the `region` argument of the provider is used. Changing this creates a new security group rule.

- `direction` - (Required) The direction of the rule, valid values are **ingress** or **egress**. Changing this creates a new security group rule.

- `ethertype` - (Required) The layer 3 protocol type, valid values are **IPv4** or **IPv6**. Changing this creates a new security group rule.

- `protocol` - (Optional) The layer 4 protocol type, valid values are following. Changing this creates a new security group rule. This is required if you want to specify a port range.

  - **tcp**

  - **udp**

  - **icmp**

  - **ah**

  - **dccp**

  - **egp**

  - **esp**

  - **gre**

- **igmp**

- **ipv6-encap**

- **ipv6-frag**

- **ipv6-icmp**

- **ipv6-nonxt**

- **ipv6-opts**

- **ipv6-route**

- **ospf**

- **pgm**

- **rsvp**

- **sctp**

- **udplite**

- **vrrp**

- `port_range_min` - (Optional) The lower part of the allowed port range, valid integer value needs to be between 1 and 65535. Changing this creates a new security group rule.

- `port_range_max` - (Optional) The higher part of the allowed port range, valid integer value needs to be between 1 and 65535. Changing this creates a new security group rule.

- `remote_ip_prefix` - (Optional) The remote CIDR, the value needs to be a valid CIDR (i.e. 192.168.0.0/16). Changing this creates a new security group rule.

- `remote_group_id` - (Optional) The remote group id, the value needs to be an Openstack ID of a security group in the same tenant. Changing this creates a new security group rule.

- `security_group_id` - (Required) The security group id the rule should belong to, the value needs to be an Openstack ID of a security group in the same tenant. Changing this creates a new security group rule.

- `tenant_id` - (Optional) The owner of the security group. Required if admin wants to create a port for another tenant. Changing this creates a new security group rule.

Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.

- `direction` - See Argument Reference above.

- `ethertype` - See Argument Reference above.

- `protocol` - See Argument Reference above.

- `port_range_min` - See Argument Reference above.

- `port_range_max` - See Argument Reference above.

- `remote_ip_prefix` - See Argument Reference above.

- `remote_group_id` - See Argument Reference above.

- `security_group_id` - See Argument Reference above.

- `tenant_id` - See Argument Reference above.

Import

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

```
$ terraform import telefonicaopencloud_networking_secgroup_rule_v2.secgroup_rule_1 aeb68ee3-6e9d-4256-955
c-9584a6212745
```

telefonicaopencloud_networking_secgroup_v2

Manages a V2 neutron security group resource within TelefonicaOpenCloud. 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 "telefonicaopencloud_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, TelefonicaOpenCloud 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 "telefonicaopencloud_networking_secgroup_rule_v2" "secgroup_rule_v4" {
  direction = "egress"
  ethertype = "IPv4"
  security_group_id = "${telefonicaopencloud_networking_secgroup_v2.secgroup.id}"
}

resource "telefonicaopencloud_networking_secgroup_rule_v2" "secgroup_rule_v6" {
  direction = "egress"
  ethertype = "IPv6"
  security_group_id = "${telefonicaopencloud_networking_secgroup_v2.secgroup.id}"
}
```

Please note that this behavior may differ depending on the configuration of the TelefonicaOpenCloud cloud. The above illustrates the current default Neutron behavior. Some TelefonicaOpenCloud 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 telefonicaopencloud_networking_secgroup_v2.secgroup_1 38809219-5e8a-4852-9139-6f461c90
e8bc
```

telefonicaopencloud_networking_subnet_v2

Manages a V2 Neutron subnet resource within TelefonicaOpenCloud.

Example Usage

```
resource "telefonicaopencloud_networking_network_v2" "network_1" {
  name          = "tf_test_network"
  admin_state_up = "true"
}

resource "telefonicaopencloud_networking_subnet_v2" "subnet_1" {
  network_id = "${telefonicaopencloud_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 telefonicaopencloud_networking_subnet_v2.subnet_1 da4faf16-5546-41e4-8330-4d0002b74048
```

telefonicaopencloud_s3_bucket

Provides a S3 bucket resource.

Example Usage

Private Bucket w/ Tags

```
resource "telefonicaopencloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"
}
```

Static Website Hosting

```
resource "telefonicaopencloud_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 "telefonicaopencloud_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 "telefonicaopencloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  versioning {
    enabled = true
  }
}
```

Enable Logging

```
resource "telefonicaopencloud_s3_bucket" "log_bucket" {
  bucket = "my-tf-log-bucket"
  acl    = "log-delivery-write"
}

resource "telefonicaopencloud_s3_bucket" "b" {
  bucket = "my-tf-test-bucket"
  acl    = "private"

  logging {
    target_bucket = "${telefonicaopencloud_s3_bucket.log_bucket.id}"
    target_prefix = "log/"
  }
}
```

Using object lifecycle
```

```
resource "telefonicaopencloud_s3_bucket" "bucket" {
  bucket = "my-bucket"
  acl    = "private"

  lifecycle_rule {
    id      = "log"
    enabled = true

    prefix  = "log/"

    expiration {
      days = 90
    }
  }

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

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

resource "telefonicaopencloud_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 (https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html#canned-acl) to apply. Defaults to "private".

- `policy` - (Optional) A valid bucket policy (https://docs.aws.amazon.com/AmazonS3/latest/dev/example-bucket-policies.html) JSON document. Note that if the policy document is not specific enough (but still valid), Terraform may view the policy as constantly changing in a `terraform plan`. In this case, please make sure you use the verbose/specific version of the policy.

- `force_destroy` - (Optional, Default:false) A boolean that indicates all objects should be deleted from the bucket so that the bucket can be destroyed without error. These objects are *not* recoverable.

- `website` - (Optional) A website object (documented below).

- `cors_rule` - (Optional) A rule of Cross-Origin Resource Sharing (https://docs.aws.amazon.com/AmazonS3/latest/dev/cors.html) (documented below).

- `versioning` - (Optional) A state of versioning (https://docs.aws.amazon.com/AmazonS3/latest/dev/Versioning.html) (documented below)

- `logging` - (Optional) A settings of bucket logging (https://docs.aws.amazon.com/AmazonS3/latest/UG/ManagingBucketLogging.html) (documented below).

- `lifecycle_rule` - (Optional) A configuration of object lifecycle management (http://docs.aws.amazon.com/AmazonS3/latest/dev/object-lifecycle-mgmt.html) (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 (https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/aws-properties-s3-websiteconfiguration-routingrules.html) describing redirect behavior and when redirects are applied.

The `CORS` object supports the following:

- `allowed_headers` (Optional) Specifies which headers are allowed.

- `allowed_methods` (Required) Specifies which methods are allowed. Can be `GET`, `PUT`, `POST`, `DELETE` or `HEAD`.

- `allowed_origins` (Required) Specifies which origins are allowed.

- `expose_headers` (Optional) Specifies expose header in the response.

- `max_age_seconds` (Optional) Specifies time in seconds that browser can cache the response for a preflight request.

The `versioning` object supports the following:

- `enabled` - (Optional) Enable versioning. Once you version-enable a bucket, it can never return to an unversioned state. You can, however, suspend versioning on that bucket.

- `mfa_delete` - (Optional) Enable MFA delete for either `Change the versioning state of your bucket` or `Permanently delete an object version`. Default is `false`.

The `logging` object supports the following:

- `target_bucket` - (Required) The name of the bucket that will receive the log objects.

- `target_prefix` - (Optional) To specify a key prefix for log objects.

The `lifecycle_rule` object supports the following:

- `id` - (Optional) Unique identifier for the rule.

- `prefix` - (Optional) Object key prefix identifying one or more objects to which the rule applies.

- `enabled` - (Required) Specifies lifecycle rule status.

- `abort_incomplete_multipart_upload_days` (Optional) Specifies the number of days after initiating a multipart upload when the multipart upload must be completed.

- `expiration` - (Optional) Specifies a period in the object's expire (documented below).

- `noncurrent_version_expiration` - (Optional) Specifies when noncurrent object versions expire (documented below).

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

The `expiration` object supports the following

- `date` (Optional) Specifies the date after which you want the corresponding action to take effect.

- `days` (Optional) Specifies the number of days after object creation when the specific rule action takes effect.

- `expired_object_delete_marker` (Optional) On a versioned bucket (versioning-enabled or versioning-suspended bucket), you can add this element in the lifecycle configuration to direct Amazon S3 to delete expired object delete markers.

The `noncurrent_version_expiration` object supports the following

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

The `rules` object supports the following:

- `id` - (Optional) Unique identifier for the rule.

- `destination` - (Required) Specifies the destination for the rule (documented below).

- `prefix` - (Required) Object keyname prefix identifying one or more objects to which the rule applies. Set as an empty string to replicate the whole bucket.

- `status` - (Required) The status of the rule. Either `Enabled` or `Disabled`. The rule is ignored if status is not Enabled.

The `destination` object supports the following:

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

- `storage_class` - (Optional) The class of storage used to store the object.

Attributes Reference

The following attributes are exported:

- `id` - The name of the bucket.

- `arn` - The ARN of the bucket. Will be of format `arn:aws:s3:::bucketname`.

- `bucket_domain_name` - The bucket domain name. Will be of format `bucketname.s3.amazonaws.com`.

- `hosted_zone_id` - The Route 53 Hosted Zone ID (https://docs.aws.amazon.com/general/latest/gr/rande.html#s3_website_region_endpoints) 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 telefonicaopencloud_s3_bucket.bucket bucket-name
```

telefonicaopencloud_s3_bucket_object

Provides a S3 bucket object resource.

Example Usage

Uploading a file to a bucket

```
resource "telefonicaopencloud_s3_bucket_object" "object" {
  bucket = "your_bucket_name"
  key    = "new_object_key"
  source = "path/to/file"
  etag   = "${md5(file("path/to/file"))}"
}
```

resource "telefonicaopencloud_s3_bucket" "examplebucket" { bucket = "examplebuckettftest" acl = "private" }

resource "telefonicaopencloud_s3_bucket_object" "examplebucket_object" { key = "someobject" bucket =

"${telefonicaopencloud_s3_bucket.examplebucket.bucket}" source = "index.html" } ```

Server Side Encryption with S3 Default Master Key

```
resource "telefonicaopencloud_s3_bucket" "examplebucket" {
  bucket = "examplebuckettftest"
  acl    = "private"
}

resource "telefonicaopencloud_s3_bucket_object" "examplebucket_object" {
  key                   = "someobject"
  bucket                = "${telefonicaopencloud_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 (https://docs.aws.amazon.com/AmazonS3/latest/dev/acl-overview.html#canned-acl) to apply. Defaults to "private".

- `cache_control` - (Optional) Specifies caching behavior along the request/reply chain Read w3c cache_control (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.9) for further details.

- `content_disposition` - (Optional) Specifies presentational information for the object. Read wc3 content_disposition (http://www.w3.org/Protocols/rfc2616/rfc2616-sec19.html#sec19.5.1) 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 (http://www.w3.org/Protocols/rfc2616/rfc2616-sec14.html#sec14.11) 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 (http://docs.aws.amazon.com/AmazonS3/latest/dev/how-to-page-redirect.html).

- `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".

- `sse_kms_key_id` - (Optional) The ID of the kms key.

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.

telefonicaopencloud_s3_bucket_policy

Attaches a policy to an S3 bucket resource.

Example Usage

Basic Usage

```
resource "telefonicaopencloud_s3_bucket" "b" {
  bucket = "my_tf_test_bucket"
}

resource "telefonicaopencloud_s3_bucket_policy" "b" {
  bucket = "${telefonicaopencloud_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.

telefonicaopencloud_smn_subscription_v2

Manages a V2 subscription resource within TelefonicaopenCloud.

Example Usage

```
resource "telefonicaopencloud_smn_topic_v2" "topic_1" {
  name          = "topic_1"
  display_name  = "The display name of topic_1"
}

resource "telefonicaopencloud_smn_subscription_v2" "subscription_1" {
  topic_urn       = "${telefonicaopencloud_smn_topic_v2.topic_1.id}"
  endpoint        = "mailtest@gmail.com"
  protocol        = "email"
  remark          = "O&M"
}

resource "telefonicaopencloud_smn_subscription_v2" "subscription_2" {
  topic_urn       = "${telefonicaopencloud_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.

telefonicaopencloud_smn_topic_v2

Manages a V2 topic resource within TelefonicaopenCloud.

Example Usage

```
resource "telefonicaopencloud_smn_topic_v2" "topic_1" {
  name            = "topic_1"
  display_name    = "The display name of topic_1"
}
```

Argument Reference

The following arguments are supported:

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

- `display_name` - (Optional) Topic display name, which is presented as the name of the email sender in an email message.

- `topic_urn` - (Optional) Resource identifier of a topic, which is unique.

- `push_policy` - (Optional) Message pushing policy. 0 indicates that the message sending fails and the message is cached in the queue. 1 indicates that the failed message is discarded.

- `create_time` - (Optional) Time when the topic was created.

- `update_time` - (Optional) Time when the topic was updated.

Attributes Reference

The following attributes are exported:

- `name` - See Argument Reference above.

- `display_name` - See Argument Reference above.

- `topic_urn` - See Argument Reference above.

- `push_policy` - See Argument Reference above.

- `create_time` - See Argument Reference above.

- `update_time` - See Argument Reference above.

telefonicaopencloud_vpc_eip_v1

Manages a V1 EIP resource within Telefonica Open Cloud VPC.

Example Usage

```
resource "telefonicaopencloud_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 telefonicaopencloud_vpc_eip_v1.eip_1 2c7f39f3-702b-48d1-940c-b50384177ee1
```