

## » flexibleengine\_\_networking\_\_network\_\_v2

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

### » Example Usage

```
data "flexibleengine_networking_network_v2" "network" {  
  name = "tf_test_network"  
}
```

### » Argument Reference

- **region** - (Optional) The region in which to obtain the V2 Neutron client. A Neutron client is needed to retrieve networks ids. If omitted, the **region** argument of the provider is used.
- **network\_id** - (Optional) The ID of the network.
- **name** - (Optional) The name of the network.
- **matching\_subnet\_cidr** - (Optional) The CIDR of a subnet within the network.
- **tenant\_id** - (Optional) The owner of the network.

### » Attributes Reference

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

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

## » flexibleengine\_\_networking\_\_secgroup\_\_v2

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

## » Example Usage

```
data "flexibleengine_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.

## » flexibleengine\_rds\_flavors\_v1

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

## » Example Usage

```
data "flexibleengine_rds_flavors_v1" "flavor" {
  region = "eu-de"
  datastore_name = "PostgreSQL"
  datastore_version = "9.5.5"
  speccode = "rds.pg.s1.medium"
}
```

## » Argument Reference

- **region** - (Required) The region in which to obtain the V1 rds client.
- **datastore\_name** - (Required) The datastore name of the rds.
- **datastore\_version** - (Required) The datastore version of the rds.
- **speccode** - (Optional) The spec code of a rds flavor.

## » Available value for attributes

datastore_name	datastore_version	speccode
PostgreSQL	9.5.5	rds.pg.s1.xlarge rds.pg.m1.2xlarge rds.pg.c2.xlarge rds.pg.s1.medium rds.pg.s1.xsmall
	9.6.3	
	9.6.5	
MySQL	5.6.33	rds.mysql.s1.medium rds.mysql.s1.large rds.mysql.s1.xlarge rds.mysql.s1.2xlarge
	5.6.30	
	5.6.34	
	5.6.35	
	5.7.17	
SQLServer	2014 SP2 SE	rds.mssql.s1.xlarge rds.mssql.m1.2xlarge rds.mssql.c2.xlarge rds.mssql.s1.2xlarge

## » Attributes Reference

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

- **region** - See Argument Reference above.
- **datastore\_name** - See Argument Reference above.
- **datastore\_version** - See Argument Reference above.
- **speccode** - See Argument Reference above.
- **name** - The name of the rds flavor.
- **ram** - The name of the rds flavor.

## » flexibleengine\_\_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 "flexibleengine_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
- \* ``range`` - (Optional) Obtains the specified range bytes of an object. The value is a range
- \* ``version_id`` - (Optional) Specific version ID of the object returned (defaults to latest v

### ## Attributes Reference

The following attributes are exported:

- \* ``body`` - Object data (see **limitations above** to understand cases in which this field is
- \* ``cache_control`` - Specifies caching behavior along the request/reply chain.
- \* ``content_disposition`` - Specifies presentational information for the object.
- \* ``content_encoding`` - Specifies what content encodings have been applied to the object and
- \* ``content_language`` - The language the content is in.
- \* ``content_length`` - Size of the body in bytes.
- \* ``content_type`` - A standard MIME type describing the format of the object data.
- \* ``etag`` - [ETag] ([https://en.wikipedia.org/wiki/HTTP\\_ETag](https://en.wikipedia.org/wiki/HTTP_ETag)) generated for the object (an MD5
- \* ``expiration`` - If the object expiration is configured, the field includes this header. It
- \* ``expires`` - The date and time at which the object is no longer cacheable.
- \* ``last_modified`` - Last modified date of the object in RFC1123 format (e.g. ``Mon, 02 Jan 20`
- \* ``metadata`` - A map of metadata stored with the object in S3
- \* ``server_side_encryption`` - If the object is stored using server-side encryption (KMS or AR
- \* ``sse_kms_key_id`` - If present, specifies the ID of the Key Management Service (KMS) master
- \* ``website_redirect_location`` - If the bucket is configured as a website, redirects requests

## » flexibleengine\_blockstorage\_volume\_v2

Manages a V2 volume resource within FlexibleEngine.

## » Example Usage

```
resource "flexibleengine_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 flexibleengine_blockstorage_volume_v2.volume_1 ea257959-eeb1-4c10-8d33-20
```

## » flexibleengine\_\_compute\_\_floatingip\_\_v2

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

## » Example Usage

```
resource "flexibleengine_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 flexibleengine_compute_floatingip_v2.floatip_1 89c60255-9bd6-460c-822a-e2
```

## » flexibleengine\_\_compute\_\_floatingip\_\_associate\_\_v2

Associate a floating IP to an instance. This can be used instead of the `floating_ip` options in `flexibleengine_compute_instance_v2`.

## » Example Usage

### » Automatically detect the correct network

```
resource "flexibleengine_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 "flexibleengine_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "flexibleengine_compute_floatingip_associate_v2" "fip_1" {
```

```

floating_ip = "${flexibleengine_networking_floatingip_v2.fip_1.address}"
instance_id = "${flexibleengine_compute_instance_v2.instance_1.id}"
}

```

## » Explicitly set the network to attach to

```

resource "flexibleengine_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 "flexibleengine_networking_floatingip_v2" "fip_1" {
  pool = "my_pool"
}

resource "flexibleengine_compute_floatingip_associate_v2" "fip_1" {
  floating_ip = "${flexibleengine_networking_floatingip_v2.fip_1.address}"
  instance_id = "${flexibleengine_compute_instance_v2.instance_1.id}"
  fixed_ip    = "${flexibleengine_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 associate 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 flexibleengine_compute_floatingip_associate_v2.fip_1 <floating_ip>/<instance_id>
```

## » flexibleengine\_compute\_instance\_v2

Manages a V2 VM instance resource within FlexibleEngine.

## » Example Usage

### » Basic Instance

```
resource "flexibleengine_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 "flexibleengine_blockstorage_volume_v2" "myvol" {
```

```

    name = "myvol"
    size = 1
}

resource "flexibleengine_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 "flexibleengine_compute_volume_attach_v2" "attached" {
  compute_id = "${flexibleengine_compute_instance_v2.myinstance.id}"
  volume_id = "${flexibleengine_blockstorage_volume_v2.myvol.id}"
}

```

## » Boot From Volume

```

resource "flexibleengine_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 "flexibleengine_blockstorage_volume_v1" "myvol" {
  name      = "myvol"
  size      = 5
  image_id  = "<image-id>"
}

resource "flexibleengine_compute_instance_v2" "boot-from-volume" {
  name          = "bootfromvolume"
  flavor_id     = "3"
  key_pair      = "my_key_pair_name"
  security_groups = ["default"]

  block_device {
    uuid              = "${flexibleengine_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 "flexibleengine_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 "flexibleengine_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}

resource "flexibleengine_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          = "${flexibleengine_blockstorage_volume_v2.volume_1.id}"
    source_type   = "volume"
    destination_type = "volume"
    boot_index    = 1
    delete_on_termination = true
  }
}

```

## » Instance With Multiple Networks

```

resource "flexibleengine_networking_floatingip_v2" "myip" {
  pool = "my_pool"
}

```

```

}

resource "flexibleengine_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 "flexibleengine_compute_floatingip_associate_v2" "myip" {
  floating_ip = "${flexibleengine_networking_floatingip_v2.myip.address}"
  instance_id = "${flexibleengine_compute_instance_v2.multi-net.id}"
  fixed_ip    = "${flexibleengine_compute_instance_v2.multi-net.network.1.fixed_ip_v4}"
}

```

## » Instance With Personality

```

resource "flexibleengine_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 "flexibleengine_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 "flexibleengine_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 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.
- **force\_delete** - (Optional) Whether to force the FlexibleEngine instance to be forcefully deleted. This is useful for environments that have reclaim / soft deletion enabled.

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/fix_ip_v4` - The Fixed IPv4 address of the Instance on that network.
- `network/fix_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 "flexibleengine_compute_instance_v2" "foo" {
  name           = "terraform-test"
  security_groups = ["default"]

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

  block_device {
    boot_index          = -1
    delete_on_termination = true
  }
}
```

```

        destination_type    = "local"
        source_type         = "blank"
        volume_size         = 1
    }

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

```

## » Instances and Ports

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

- When attaching an Instance to one or more networks using Ports, place the security groups on the Port and not the Instance. If you place the security groups on the Instance, the security groups will not be applied upon creation, but they will be applied upon a refresh. This is a known FlexibleEngine 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 "flexibleengine_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 "flexibleengine_compute_instance_v2" "instance_1" {
  name = "instance_1"

  network {
    port = "${flexibleengine_networking_port_v2.port_1.id}"
  }

  connection {
    user      = "root"
    host      = "${flexibleengine_networking_port_v2.port_1.fixed_ip.0.ip_address}"
    private_key = "~/path/to/key"
  }

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

```

## » flexibleengine\_compute\_keypair\_v2

Manages a V2 keypair resource within FlexibleEngine.

### » Example Usage

```

resource "flexibleengine_compute_keypair_v2" "test-keypair" {
  name      = "my-keypair"
  public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDAjpc1hwi0CCmKEWxJ4qzTTsJbKzndLotBCz5L"
}

```

### » 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 flexibleengine_compute_keypair_v2.my-keypair test-keypair
```

## » flexibleengine\_\_compute\_\_servergroup\_\_v2

Manages a V2 Server Group resource within FlexibleEngine.

## » Example Usage

```
resource "flexibleengine_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 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 flexibleengine_compute_servergroup_v2.test-sg 1bc30ee9-9d5b-4c30-bdd5-7f1
```

## » flexibleengine\_\_compute\_\_volume\_\_attach\_\_v2

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

## » Example Usage

### » Basic attachment of a single volume to a single instance

```
resource "flexibleengine_blockstorage_volume_v2" "volume_1" {
  name = "volume_1"
  size = 1
}
```

```

}

resource "flexibleengine_compute_instance_v2" "instance_1" {
  name          = "instance_1"
  security_groups = ["default"]
}

resource "flexibleengine_compute_volume_attach_v2" "va_1" {
  instance_id = "${flexibleengine_compute_instance_v2.instance_1.id}"
  volume_id   = "${flexibleengine_blockstorage_volume_v2.volume_1.id}"
}

```

### » Attaching multiple volumes to a single instance

```

resource "flexibleengine_blockstorage_volume_v2" "volumes" {
  count = 2
  name   = "${format("vol-%02d", count.index + 1)}"
  size   = 1
}

resource "flexibleengine_compute_instance_v2" "instance_1" {
  name          = "instance_1"
  security_groups = ["default"]
}

resource "flexibleengine_compute_volume_attach_v2" "attachments" {
  count          = 2
  instance_id    = "${flexibleengine_compute_instance_v2.instance_1.id}"
  volume_id      = "${element(flexibleengine_blockstorage_volume_v2.volumes.*.id, count.index)}"
}

output "volume devices" {
  value = "${flexibleengine_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 flexibleengine_compute_volume_attach_v2.va_1 89c60255-9bd6-460c-822a-e2b5
```

## » flexibleengine\_dns\_recordset\_v2

Manages a DNS record set in the FlexibleEngine DNS Service.

## » Example Usage

### » Automatically detect the correct network

```
resource "flexibleengine_dns_zone_v2" "example_zone" {
  name = "example.com."
  email = "email2@example.com"
  description = "a zone"
  ttl = 6000
```



```

    type = "PRIMARY"
  }

  resource "flexibleengine_dns_recordset_v2" "rs_example_com" {
    zone_id = "${flexibleengine_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 flexibleengine_dns_recordset_v2.recordset_1 <zone_id>/<recordset_id>
```

## » `flexibleengine_dns_zone_v2`

Manages a DNS zone in the FlexibleEngine DNS Service.

## » Example Usage

### » Automatically detect the correct network

```
resource "flexibleengine_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.
- **t1** - (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.
- **t1** - 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 flexibleengine_dns_zone_v2.zone_1 <zone_id>
```

## » flexibleengine\_networking\_floatingip\_v2

Manages a V2 floating IP resource within FlexibleEngine 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 "flexibleengine_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 flexibleengine_networking_floatingip_v2.floatip_1 2c7f39f3-702b-48d1-940c
```

## » flexibleengine\_networking\_network\_v2

Manages a V2 Neutron network resource within FlexibleEngine.

## » Example Usage

```
resource "flexibleengine_networking_network_v2" "network_1" {
  name          = "network_1"
  admin_state_up = "true"
}

resource "flexibleengine_networking_subnet_v2" "subnet_1" {
  name          = "subnet_1"
  network_id    = "${flexibleengine_networking_network_v2.network_1.id}"
  cidr          = "192.168.199.0/24"
  ip_version    = 4
}

resource "flexibleengine_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 "flexibleengine_networking_port_v2" "port_1" {
  name          = "port_1"
  network_id    = "${flexibleengine_networking_network_v2.network_1.id}"
  admin_state_up = "true"
  security_group_ids = ["${flexibleengine_compute_secgroup_v2.secgroup_1.id}"]

  fixed_ip {
    "subnet_id" = "${flexibleengine_networking_subnet_v2.subnet_1.id}"
    "ip_address" = "192.168.199.10"
  }
}

resource "flexibleengine_compute_instance_v2" "instance_1" {
  name          = "instance_1"
  security_groups = ["${flexibleengine_compute_secgroup_v2.secgroup_1.name}"]

  network {
    port = "${flexibleengine_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 capabilities 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 physical 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 flexibleengine_networking_network_v2.network_1 d90ce693-5ccf-4136-a0ed-15
```

## » flexibleengine\_networking\_port\_v2

Manages a V2 port resource within FlexibleEngine.

## » Example Usage

```
resource "flexibleengine_networking_network_v2" "network_1" {
  name          = "network_1"
  admin_state_up = "true"
}

resource "flexibleengine_networking_port_v2" "port_1" {
  name          = "port_1"
  network_id    = "${flexibleengine_networking_network_v2.network_1.id}"
  admin_state_up = "true"
}
```

## » Argument Reference

The following arguments are supported:

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

- **tenant\_id** - (Optional) The owner of the Port. Required if admin wants to create a port for another tenant. Changing this creates a new port.
- **device\_owner** - (Optional) The device owner of the Port. Changing this creates a new port.
- **security\_group\_ids** - (Optional) A list of security group IDs to apply to the port. The security groups must be specified by ID and not name (as opposed to how they are configured with the Compute Instance).
- **device\_id** - (Optional) The ID of the device attached to the port. Changing this creates a new port.
- **fixed\_ip** - (Optional) An array of desired IPs for this port. The structure is described below.
- **allowed\_address\_pairs** - (Optional) An IP/MAC Address pair of additional IP addresses that can be active on this port. The structure is described below.
- **value\_specs** - (Optional) Map of additional options.

The **fixed\_ip** block supports:

- **subnet\_id** - (Required) Subnet in which to allocate IP address for this port.
- **ip\_address** - (Optional) IP address desired in the subnet for this port. If you don't specify **ip\_address**, an available IP address from the specified subnet will be allocated to this port.

The **allowed\_address\_pairs** block supports:

- **ip\_address** - (Required) The additional IP address.
- **mac\_address** - (Optional) The additional MAC address.

## » Attributes Reference

The following attributes are exported:

- **region** - See Argument Reference above.
- **admin\_state\_up** - See Argument Reference above.
- **mac\_address** - See Argument Reference above.
- **tenant\_id** - See Argument Reference above.
- **device\_owner** - See Argument Reference above.
- **security\_group\_ids** - See Argument Reference above.
- **device\_id** - See Argument Reference above.
- **fixed\_ip** - See Argument Reference above.
- **all\_fixed\_ips** - The collection of Fixed IP addresses on the port in the order returned by the Network v2 API.



## » Import

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

```
$ terraform import flexibleengine_networking_port_v2.port_1 eae26a3e-1c33-4cc1-9c31-0cd729c4
```

## » Notes

### » Ports and Instances

There are some notes to consider when connecting Instances to networks using Ports. Please see the `flexibleengine_compute_instance_v2` documentation for further documentation.

## » flexibleengine\_networking\_router\_interface\_v2

Manages a V2 router interface resource within FlexibleEngine.

### » Example Usage

```
resource "flexibleengine_networking_network_v2" "network_1" {
  name          = "tf_test_network"
  admin_state_up = "true"
}

resource "flexibleengine_networking_subnet_v2" "subnet_1" {
  network_id = "${flexibleengine_networking_network_v2.network_1.id}"
  cidr       = "192.168.199.0/24"
  ip_version = 4
}

resource "flexibleengine_networking_router_v2" "router_1" {
  name          = "my_router"
  external_gateway = "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"
}

resource "flexibleengine_networking_router_interface_v2" "router_interface_1" {
  router_id = "${flexibleengine_networking_router_v2.router_1.id}"
  subnet_id = "${flexibleengine_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.

## » flexibleengine\_networking\_router\_route\_v2

Creates a routing entry on a FlexibleEngine V2 router.

## » Example Usage

```
resource "flexibleengine_networking_router_v2" "router_1" {
  name           = "router_1"
  admin_state_up = "true"
}

resource "flexibleengine_networking_network_v2" "network_1" {
  name           = "network_1"
  admin_state_up = "true"
}

resource "flexibleengine_networking_subnet_v2" "subnet_1" {
  network_id = "${flexibleengine_networking_network_v2.network_1.id}"
}
```

```

    cidr          = "192.168.199.0/24"
    ip_version = 4
}

resource "flexibleengine_networking_router_interface_v2" "int_1" {
    router_id = "${flexibleengine_networking_router_v2.router_1.id}"
    subnet_id = "${flexibleengine_networking_subnet_v2.subnet_1.id}"
}

resource "flexibleengine_networking_router_route_v2" "router_route_1" {
    depends_on      = ["flexibleengine_networking_router_interface_v2.int_1"]
    router_id       = "${flexibleengine_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 `flexibleengine_networking_router_route_v2` resource creation time. You can ensure that by explicitly specifying a dependency on the `flexibleengine_networking_router_interface_v2` resource that connects the next hop to the router, as in the example above.

## » flexibleengine\_networking\_router\_v2

Manages a V2 router resource within FlexibleEngine.

## » Example Usage

```
resource "flexibleengine_networking_router_v2" "router_1" {
  name          = "my_router"
  external_gateway = "f67f0d72-0ddf-11e4-9d95-e1f29f417e2f"
}
```

## » Argument Reference

The following arguments are supported:

- **region** - (Optional) The region in which to obtain the V2 networking client. A networking client is needed to create a router. If omitted, the `region` argument of the provider is used. Changing this creates a new router.
- **name** - (Optional) A unique name for the router. Changing this updates the `name` of an existing router.
- **admin\_state\_up** - (Optional) Administrative up/down status for the router (must be "true" or "false" if provided). Changing this updates the `admin_state_up` of an existing router.
- **distributed** - (Optional) Indicates whether or not to create a distributed router. The default policy setting in Neutron restricts usage of this property to administrative users only.
- **external\_gateway** - (Optional) The network UUID of an external gateway for the router. A router with an external gateway is required if any compute instances or load balancers will be using floating IPs. Changing this updates the `external_gateway` of an existing router.

- **tenant\_id** - (Optional) The owner of the floating IP. Required if admin wants to create a router for another tenant. Changing this creates a new router.
- **value\_specs** - (Optional) Map of additional driver-specific options.

## » Attributes Reference

The following attributes are exported:

- **id** - ID of the router.
- **region** - See Argument Reference above.
- **name** - See Argument Reference above.
- **admin\_state\_up** - See Argument Reference above.
- **external\_gateway** - See Argument Reference above.
- **tenant\_id** - See Argument Reference above.
- **value\_specs** - See Argument Reference above.

## » flexibleengine\_networking\_subnet\_v2

Manages a V2 Neutron subnet resource within FlexibleEngine.

## » Example Usage

```
resource "flexibleengine_networking_network_v2" "network_1" {
  name          = "tf_test_network"
  admin_state_up = "true"
}

resource "flexibleengine_networking_subnet_v2" "subnet_1" {
  network_id = "${flexibleengine_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 flexibleengine_networking_subnet_v2.subnet_1 da4faf16-5546-41e4-8330-4d00
```

## » flexibleengine\_networking\_secgroup\_v2

Manages a V2 neutron security group resource within FlexibleEngine. 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 "flexibleengine_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, FlexibleEngine 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 "flexibleengine_networking_secgroup_rule_v2" "secgroup_rule_v4" {
  direction = "egress"
  ethertype = "IPv4"
  security_group_id = "${flexibleengine_networking_secgroup_v2.secgroup.id}"
}
```

```
resource "flexibleengine_networking_secgroup_rule_v2" "secgroup_rule_v6" {
  direction = "egress"
  ethertype = "IPv6"
  security_group_id = "${flexibleengine_networking_secgroup_v2.secgroup.id}"
}
```

Please note that this behavior may differ depending on the configuration of the FlexibleEngine cloud. The above illustrates the current default Neutron behavior. Some FlexibleEngine 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 flexibleengine_networking_secgroup_v2.secgroup_1 38809219-5e8a-4852-9139-
```

## » flexibleengine\_networking\_secgroup\_rule\_v2

Manages a V2 neutron security group rule resource within FlexibleEngine. 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 "flexibleengine_networking_secgroup_v2" "secgroup_1" {
  name          = "secgroup_1"
  description   = "My neutron security group"
}

resource "flexibleengine_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 = "${flexibleengine_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 FlexibleEngine 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 FlexibleEngine 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 flexibleengine_networking_secgroup_rule_v2.secgroup_rule_1 aeb68ee3-6e9d-
```

## » flexibleengine\_elb\_loadbalancer

Manages an elastic loadbalancer resource within FlexibleEngine.

## » Example Usage

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

## » Argument Reference

The following arguments are supported:

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

- **name** - (Required) Specifies the load balancer name. The name is a string of 1 to 64 characters that consist of letters, digits, underscores (`_`), and hyphens (`-`).
- **description** - (Optional) Provides supplementary information about the listener. The value is a string of 0 to 128 characters and cannot be `<>`.
- **vpc\_id** - (Required) Specifies the VPC ID.
- **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.
- **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 type is set to External, the value of this parameter is the elastic IP address. When type is set to Internal, the value of this parameter is the private network IP address. You can select an existing elastic IP address and create a public network load balancer. When this parameter is configured, parameter bandwidth is invalid.
- **tenantid** - (Optional) Specifies the tenant ID. This parameter is mandatory only when type is set to Internal.

## » Attributes Reference

The following attributes are exported:

- **region** - See Argument Reference above.

- `name` - See Argument Reference above.
- `description` - See Argument Reference above.
- `vpc_id` - See Argument Reference above.
- `bandwidth` - See Argument Reference above.
- `type` - See Argument Reference above.
- `admin_state_up` - See Argument Reference above.
- `vip_subnet_id` - See Argument Reference above.
- `az` - See Argument Reference above.
- `security_group_id` - See Argument Reference above.
- `vip_address` - See Argument Reference above.
- `tenantid` - See Argument Reference above.
- `id` - Specifies the load balancer ID.

## » `flexibleengine_elb_listener`

Manages an elastic loadbalancer listener resource within FlexibleEngine.

### » Example Usage

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

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

## » Argument Reference

The following arguments are supported:

- **region** - (Optional) The region in which to create the elb listener. If omitted, the **region** argument of the provider is used. Changing this creates a new elb listener.
- **name** - (Required) Specifies the load balancer name. The name is a string of 1 to 64 characters that consist of letters, digits, underscores (`_`), and hyphens (`-`).
- **description** - (Optional) Provides supplementary information about the listener. The value is a string of 0 to 128 characters and cannot be `<>`.
- **loadbalancer\_id** - (Required) Specifies the ID of the load balancer to which the listener belongs.
- **protocol** - (Required) Specifies the listening protocol used for layer 4 or 7. The value can be HTTP, TCP, HTTPS, or UDP.
- **protocol\_port** - (Required) Specifies the listening port. The value ranges from 1 to 65535.
- **backend\_protocol** - (Required) Specifies the backend protocol. If the value of protocol is UDP, the value of this parameter can only be UDP. The value can be HTTP, TCP, or UDP.
- **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:

- **region** - See Argument Reference above.
- **name** - See Argument Reference above.
- **description** - See Argument Reference above.
- **loadbalancer\_id** - See Argument Reference above.
- **protocol** - See Argument Reference above.
- **protocol\_port** - See Argument Reference above.
- **backend\_protocol** - See Argument Reference above.
- **backend\_port** - See Argument Reference above.

- `lb_algorithm` - See Argument Reference above.
- `session_sticky` - See Argument Reference above.
- `sticky_session_type` - See Argument Reference above.
- `cookie_timeout` - See Argument Reference above.
- `tcp_timeout` - See Argument Reference above.
- `tcp_draining` - See Argument Reference above.
- `tcp_draining_timeout` - See Argument Reference above.
- `certificate_id` - See Argument Reference above.
- `udp_timeout` - See Argument Reference above.
- `ssl_protocols` - See Argument Reference above.
- `ssl_ciphers` - See Argument Reference above.
- `id` - Specifies the listener ID.
- `admin_state_up` - Specifies the status of the load balancer. Value range:  
false: The load balancer is disabled. true: The load balancer runs properly.

## » `flexibleengine_elb__health`

Manages an elastic loadbalancer health resource within FlexibleEngine.

### » Example Usage

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

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



```

    }
  }

resource "flexibleengine_elb_health" "healthcheck" {
  listener_id = "${flexibleengine_elb_listener.listener.id}"
  healthcheck_protocol = "TCP"
  healthcheck_connect_port = 22
  healthy_threshold = 5
  healthcheck_timeout = 25
  healthcheck_interval = 3
  timeouts {
    create = "5m"
    update = "5m"
    delete = "5m"
  }
}

```

## » Argument Reference

The following arguments are supported:

- **region** - (Optional) The region in which to create the elb health. If omitted, the **region** argument of the provider is used. Changing this creates a new elb health.
- **listener\_id** - (Required) Specifies the ID of the listener to which the health check task belongs.
- **healthcheck\_protocol** - (Optional) Specifies the protocol used for the health check. The value can be HTTP or TCP (case-insensitive).
- **healthcheck\_uri** - (Optional) Specifies the URI for health check. This parameter is valid when **healthcheck\_protocol** is HTTP. The value is a string of 1 to 80 characters that must start with a slash (/) and can only contain letters, digits, and special characters, such as `-.%?#&`.
- **healthcheck\_connect\_port** - (Optional) Specifies the port used for the health check. The value ranges from 1 to 65535.
- **healthy\_threshold** - (Optional) Specifies the threshold at which the health check result is success, that is, the number of consecutive successful health checks when the health check result of the backend server changes from fail to success. The value ranges from 1 to 10.
- **unhealthy\_threshold** - (Optional) Specifies the threshold at which the health check result is fail, that is, the number of consecutive failed health checks when the health check result of the backend server changes from success to fail. The value ranges from 1 to 10.

- `healthcheck_timeout` - (Optional) Specifies the maximum timeout duration (s) for the health check. The value ranges from 1 to 50.
- `healthcheck_interval` - (Optional) Specifies the maximum interval (s) for health check. The value ranges from 1 to 5.

## » Attributes Reference

The following attributes are exported:

- `region` - See Argument Reference above.
- `listener_id` - See Argument Reference above.
- `healthcheck_protocol` - See Argument Reference above.
- `healthcheck_uri` - See Argument Reference above.
- `healthcheck_connect_port` - See Argument Reference above.
- `healthy_threshold` - See Argument Reference above.
- `unhealthy_threshold` - See Argument Reference above.
- `healthcheck_timeout` - See Argument Reference above.
- `healthcheck_interval` - See Argument Reference above.
- `id` - Specifies the health check task ID.

## » `flexibleengine_elb_backend`

Manages an elastic loadbalancer backend resource within FlexibleEngine.

## » Example Usage

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

resource "flexibleengine_elb_listener" "listener" {
  name = "test-elb-listener"
  description = "great listener"
  protocol = "TCP"
  backend_protocol = "TCP"
  protocol_port = 12345
  backend_port = 8080
}
```

```

lb_algorithm = "roundrobin"
loadbalancer_id = "${flexibleengine_elb_loadbalancer.elb.id}"
timeouts {
    create = "5m"
    update = "5m"
    delete = "5m"
}
}

resource "flexibleengine_elb_backend" "backend" {
    address = "192.168.0.211"
    listener_id = "${flexibleengine_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.
- **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.
- **address** - See Argument Reference above.
- **server\_address** - Specifies the floating IP address assigned to the backend member.
- **id** - Specifies the backend member ID.
- **status** - Specifies the backend ECS status. The value is ACTIVE, PENDING, or ERROR.
- **health\_status** - Specifies the health check status. The value is NORMAL, ABNORMAL, or UNAVAILABLE.
- **update\_time** - Specifies the time when information about the backend member was updated.
- **create\_time** - Specifies the time when the backend member was created.
- **server\_name** - Specifies the backend member name.
- **listeners** - Specifies the listener to which the backend member belongs.

## » flexibleengine\_s3\_bucket\_policy

Attaches a policy to an S3 bucket resource.

### » Example Usage

#### » Basic Usage

```
resource "flexibleengine_s3_bucket" "b" {
  bucket = "my_tf_test_bucket"
}

resource "flexibleengine_s3_bucket_policy" "b" {
  bucket = "${flexibleengine_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.

## » **flexibleengine\_\_smn\_\_topic\_\_v2**

Manages a V2 topic resource within FlexibleEngine.

### » **Example Usage**

```
resource "flexibleengine_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.

## » **flexibleengine\_\_smn\_\_subscription\_\_v2**

Manages a V2 subscription resource within FlexibleEngine.

## » Example Usage

```
resource "flexibleengine_smn_topic_v2" "topic_1" {
  name          = "topic_1"
  display_name   = "The display name of topic_1"
}

resource "flexibleengine_smn_subscription_v2" "subscription_1" {
  topic_urn      = "${flexibleengine_smn_topic_v2.topic_1.id}"
  endpoint       = "mailto:mailtest@gmail.com"
  protocol       = "email"
  remark         = "O&M"
}

resource "flexibleengine_smn_subscription_v2" "subscription_2" {
  topic_urn      = "${flexibleengine_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.

## » `flexibleengine_rds_instance_v1`

Manages rds instance resource within FlexibleEngine

### » Example Usage: Creating a SQLServer RDS instance

```
data "flexibleengine_rds_flavors_v1" "flavor" {
  region = "eu-west-0"
  datastore_name = "SQLServer"
  datastore_version = "2014 SP2 SE"
  speccode = "rds.mssql.s1.2xlarge"
}

resource "flexibleengine_compute_secgroup_v2" "secgrp_rds" {
  name          = "secgrp-rds-instance"
  description = "Rds Security Group"
}

resource "flexibleengine_rds_instance_v1" "instance" {
  name = "rds-instance"
  datastore {
    type = "SQLServer"
    version = "2014 SP2 SE"
  }
  flavorref = "${data.flexibleengine_rds_flavors_v1.flavor.id}"
  volume {
    type = "COMMON"
    size = 200
  }
  region = "eu-west-0"
  availabilityzone = "eu-west-0a"
  vpc = "c1095fe7-03df-4205-ad2d-6f4c181d436e"
```

```

    nics {
      subnetid = "b65f8d25-c533-47e2-8601-cfaa265a3e3e"
    }
    securitygroup {
      id = "${flexibleengine_compute_secgroup_v2.secgrp_rds.id}"
    }
    dbport = "8635"
    backupstrategy = {
      starttime = "04:00:00"
      keepdays = 4
    }
    dbbrtpd = "Huangwei!120521"
    depends_on = ["flexibleengine_compute_secgroup_v2.secgrp_rds"]
  }
}

```

## » Example Usage: Creating a MySQL RDS instance

```

data "flexibleengine_rds_flavors_v1" "flavor" {
  region = "eu-west-0"
  datastore_name = "MySQL"
  datastore_version = "5.6.30"
  speccode = "rds.mysql.s1.medium"
}

resource "flexibleengine_compute_secgroup_v2" "secgrp_rds" {
  name          = "secgrp-rds-instance"
  description = "Rds Security Group"
}

resource "flexibleengine_rds_instance_v1" "instance" {
  name = "rds-instance"
  datastore {
    type = "MySQL"
    version = "5.6.30"
  }
  flavorref = "${data.flexibleengine_rds_flavors_v1.flavor.id}"
  volume {
    type = "COMMON"
    size = 200
  }
  region = "eu-west-0"
  availabilityzone = "eu-west-0a"
  vpc = "c1095fe7-03df-4205-ad2d-6f4c181d436e"
  nics {
    subnetid = "b65f8d25-c533-47e2-8601-cfaa265a3e3e"
  }
}

```



```

}
securitygroup {
    id = "${flexibleengine_compute_secgroup_v2.secgrp_rds.id}"
}
dbport = "8635"
backupstrategy = {
    starttime = "04:00:00"
    keepdays = 4
}
dbbrtpd = "Huangwei!120521"
ha = {
    enable = true
    replicationmode = "async"
}
depends_on = ["flexibleengine_compute_secgroup_v2.secgrp_rds"]
}

```

## » Argument Reference

The following arguments are supported:

- **name** - (Required) Specifies the DB instance name. The DB instance name of the same type is unique in the same tenant.
- **datastore** - (Required) Specifies database information. The structure is described below.
- **flavorref** - (Required) Specifies the specification ID (flavors.id in the response message in Obtaining All DB Instance Specifications).
- **volume** - (Required) Specifies the volume information. The structure is described below.
- **region** - (Required) Specifies the region ID.
- **availabilityzone** - (Required) Specifies the ID of the AZ.
- **vpc** - (Required) Specifies the VPC ID. For details about how to obtain this parameter value, see section "Virtual Private Cloud" in the Virtual Private Cloud API Reference.
- **nics** - (Required) Specifies the nics information. For details about how to obtain this parameter value, see section "Subnet" in the Virtual Private Cloud API Reference. The structure is described below.
- **securitygroup** - (Required) Specifies the security group which the RDS DB instance belongs to. The structure is described below.
- **dbport** - (Optional) Specifies the database port number.

- **backupstrategy** - (Optional) Specifies the advanced backup policy. The structure is described below.
- **dbrootpd** - (Required) Specifies the password for user root of the database.
- **ha** - (Optional) Specifies the parameters configured on HA and is used when creating HA DB instances. The structure is described below. NOTICE: RDS for Microsoft SQL Server does not support creating HA DB instances and this parameter is not involved.

The **datastore** block supports:

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

type	version
MySQL	5.6.33
	5.6.30
	5.6.34
	5.6.35
	5.7.17

The **volume** block supports:

- **type** - (Required) Specifies the volume type. Valid value: It must be COMMON (SATA) or ULTRAHIGH (SSD) and is case-sensitive.
- **size** - (Required) Specifies the volume size. Its value must be a multiple of 10 and the value range is 100 GB to 2000 GB.

The **nics** block supports:

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

The **securitygroup** block supports:

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

The **backupstrategy** block supports:

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

The **ha** block supports:

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

## » Attributes Reference

The following attributes are exported:

- **region** - See Argument Reference above.
- **name** - See Argument Reference above.
- **flavorref** - See Argument Reference above.
- **volume** - See Argument Reference above.
- **availabilityzone** - See Argument Reference above.
- **vpc** - See Argument Reference above.
- **nics** - See Argument Reference above.
- **securitygroup** - See Argument Reference above.
- **dbport** - See Argument Reference above.
- **backupstrategy** - See Argument Reference above.
- **dbrtpd** - See Argument Reference above.
- **ha** - See Argument Reference above.
- **status** - Indicates the DB instance status.
- **hostname** - Indicates the instance connection address. It is a blank string.
- **type** - Indicates the DB instance type, which can be master or readreplica.
- **created** - Indicates the creation time in the following format: yyyy-mm-dd Thh:mm:ssZ.
- **updated** - Indicates the update time in the following format: yyyy-mm-dd Thh:mm:ssZ.

## » Attributes Reference

The following attributes can be updated:

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