» ovh_cloud_region

Use this data source to retrieve information about a region associated with a public cloud project. The region must be associated with the project.

» Example Usage

```
data "ovh_cloud_region" "GRA1" {
   project_id = "XXXXXX"
   name = "GRA1"
}
```

» Argument Reference

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.
- `name` - (Required) The name of the region associated with the public cloud project.

» Attributes Reference

`id` is set to the ID of the project concatenated with the name of the region. In addition, the following attributes are exported:

- `continent_code` - the code of the geographic continent the region is running. E.g.: EU for Europe, US for America...
- `datacenter_location` - The location code of the datacenter. E.g.: "GRA", meaning Gravelines, for region "GRA1"
- `continentCode` - (Deprecated) Use `continent_code` instead.
- `datacenterLocation` - (Deprecated) Use `datacenter_location` instead.
- `services` - The list of public cloud services running within the region
  - `name` - the name of the public cloud service
  - `status` - the status of the service

» ovh_cloud_regions

Use this data source to get the regions of a public cloud project.

» Example Usage

```
data "ovh_cloud_regions" "regions" {
  project_id = "XXXXXX"

  has_services_up = ["network"]
}
```

» Argument Reference

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.
- `has_services_up` - (Optional) List of services which has to be UP in regions. Example: "image", "instance", "network", "storage", "volume", "workflow", ... If left blank, returns all regions associated with the project_id.

» Attributes Reference

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

- `names` - The list of regions associated with the project, filtered by services UP.

» ovh_dedicated_installation_templates

Use this data source to get the list of installation templates available for dedicated servers.

» Example Usage

```
data "ovh_dedicated_installation_templates" "templates" {}
```

» Argument Reference

This datasource takes no argument.

» Attributes Reference

The following attributes are exported:

- `result` - The list of installation templates IDs available for dedicated servers.

» ovh_dedicated_server

Use this data source to retrieve information about a dedicated server associated with your OVH Account.

» Example Usage

```
data "ovh_dedicated_server" "server" {
   service_name = "XXXXXX"
}
```

» Argument Reference

- `service_name` - (Required) The service_name of your dedicated server.

» Attributes Reference

`id` is set with the service_name of the dedicated server. In addition, the following attributes are exported:

- `boot_id` - boot id of the server
- `commercial_range` - dedicater server commercial range
- `datacenter` - dedicated datacenter localisation (bhs1,bhs2,...)
- `ip` - dedicated server ip (IPv4)
- `link_speed` - link speed of the server
- `monitoring` - Icmp monitoring state
- `name` - dedicated server name
- `os` - Operating system
- `professional_use` - Does this server have professional use option
- `rack` - rack id of the server
- `rescue_mail` - rescue mail of the server
- `reverse` - dedicated server reverse
- `root_device` - root device of the server
- `server_id` - your server id
- `state` - error, hacked, hackedBlocked, ok

- `support_level` - Dedicated server support level (critical, fastpath, gs, pro)
- `vnis` - the list of Virtualnetworkinterface assiociated with this server
  - `enabled` - VirtualNetworkInterface activation state
  - `mode` - VirtualNetworkInterface mode (public,vrack,vrack_aggregation)
  - `name` - User defined VirtualNetworkInterface name
  - `server_name` - Server bound to this VirtualNetworkInterface
  - `uuid` - VirtualNetworkInterface unique id
  - `vrack` - vRack name
  - `ncis` - NetworkInterfaceControllers bound to this VirtualNetworkInterface
- `enabled_vrack_vnis` - List of enabled vrack VNI uuids
- `enabled_vrack_aggregation_vnis` - List of enabled vrack_aggregation VNI uuids
- `enabled_public_vnis` - List of enabled public VNI uuids

» ovh_me_dedicated_server_boots

Use this data source to get the list of compatible netboots for a dedicated server associated with your OVH Account.

» Example Usage

```
data "ovh_dedicated_server_boots" "netboots" {
  service_name = "myserver"
  boot_type    = "harddisk"
}
```

» Argument Reference

- `service_name` - (Required) The internal name of your dedicated server.
- `boot_type` - (Optional) Filter the value of bootType property (harddisk, rescue, ipxeCustomerScript, internal, network)

» Attributes Reference

The following attributes are exported:

- `result` - The list of dedicated server netboots.

» ovh_dedicated_servers

Use this data source to get the list of dedicated servers associated with your OVH Account.

» Example Usage

```
data "ovh_dedicated_servers" "servers" {}
```

» Argument Reference

This datasource takes no argument.

» Attributes Reference

The following attributes are exported:

- `result` - The list of dedicated servers IDs associated with your OVH Account.

» ovh_domain_zone

Use this data source to retrieve information about a domain zone.

» Example Usage

```
data "ovh_domain_zone" "rootzone" {
    name = "mysite.ovh"
}
```

» Argument Reference

- `name` - (Required) The name of the domain zone.

» Attributes Reference

`id` is set to the domain zone name. In addition, the following attributes are exported:

- `last_update` - Last update date of the DNS zone
- `has_dns_anycast` - hasDnsAnycast flag of the DNS zone
- `name_servers` - Name servers that host the DNS zone
- `dnssec_supported` - Is DNSSEC supported by this zone

» ovh_iploadbalancing

Use this data source to retrieve information about an IP Load Balancing product

» Example Usage

```
data "ovh_iploadbalancing" "lb" {
   service_name = "xxx"
   state        = "ok"
}
```

» Argument Reference

- `ipv6` - The IPV6 associated to your IP load balancing
- `ipv4` - The IPV4 associated to your IP load balancing
- `zone` - Location where your service is. This takes an array of values.
- `offer` - The offer of your IP load balancing
- `service_name` - The internal name of your IP load balancing
- `ip_loadbalancing` - Your IP load balancing
- `state` - Current state of your IP. Can take any of the following value: ”blacklisted”, ”deleted”, ”free”, ”ok”, ”quarantined”, ”suspended”
- `vrack_eligibility` - Vrack eligibility. Takes a boolean value.
- `vrack_name` - Name of the vRack on which the current Load Balancer is attached to, as it is named on vRack product
- `ssl_configuration` - Modern oldest compatible clients : Firefox 27, Chrome 30, IE 11 on Windows 7, Edge, Opera 17, Safari 9, Android 5.0, and Java 8. Intermediate oldest compatible clients : Firefox 1, Chrome 1, IE 7, Opera 5, Safari 1, Windows XP IE8, Android 2.3, Java 7. Can take any of the following value: ”intermediate”, ”modern”
- `display_name` - the name displayed in ManagerV6 for your iplb (max 50 chars)

» Attributes Reference

`id` is set to the service_name of your IP load balancing In addition, the following attributes are exported:

- `metrics_token` - The metrics token associated with your IP load balancing This attribute is sensitive.
- `orderable_zone` - Available additional zone for your Load Balancer
  - `name` - The zone three letter code
  - `plan_code` - The billing planCode for this zone

» ovh_iploadbalancing_vrack_network

Use this data source to get the details of Vrack network available for your IPLoadbalancer associated with your OVH account.

» Example Usage

```
data ovh_iploadbalancing_vrack_network "lb_network" {
  service_name = "xxx"
  vrack_network_id = "yyy"
}
```

» Argument Reference

- `service_name` - (Required) The internal name of your IP load balancing
- `vrack_network_id` - (Required) Internal Load Balancer identifier of the vRack private network

» Attributes Reference

The following attributes are exported:

- `display_name` - Human readable name for your vrack network
- `farm_id` - Farm id your vRack network is attached to and their type
  - `type` - farm type
  - `id` - farm id
- `nat_ip` - An IP block used as a pool of IPs by this Load Balancer to connect to the servers in this private network. The blck must be in the private network and reserved for the Load Balancer
- `subnet` - IP block of the private network in the vRack

- `vlan` - VLAN of the private network in the vRack. 0 if the private network is not in a VLAN

» ovh_iploadbalancing_vrack_networks

Use this data source to get the list of Vrack network ids available for your IPLoadbalancer associated with your OVH account.

» Example Usage

```
data ovh_iploadbalancing_vrack_networks "lb_networks" {
  service_name = "xxx"
  subnet       = "10.0.0.0/24"
}
```

» Argument Reference

- `service_name` - (Required) The internal name of your IP load balancing
- `subnet` - Filters networks on the subnet.
- `vlan_id` - Filters networks on the vlan id.

» Attributes Reference

The following attributes are exported:

- `result` - The list of vrack network ids.

» ovh_me_installation_template

Use this data source to get a custom installation template available for dedicated servers.

» Example Usage

```
data "ovh_me_installation_template" "mytemplate" {
  template_name = "mytemplate"
}
```

» Argument Reference

- `template_name`: This template name

» Attributes Reference

The following attributes are exported:

- `available_languages`: List of all language available for this template.
- `beta`: This distribution is new and, although tested and functional, may still display odd behaviour.
- `bit_format`: This template bit format (32 or 64).
- `category`: Category of this template (informative only). (basic, customer, hosting, other, readyToUse, virtualisation).
- `customization`:
  - `change_log`: Template change log details.
  - `custom_hostname`: Set up the server using the provided hostname instead of the default hostname.
  - `post_installation_script_link`: Indicate the URL where your postinstall customisation script is located.
  - `post_installation_script_return`: indicate the string returned by your postinstall customisation script on successful execution. Advice: your script should return a unique validation string in case of succes. A good example is 'loh1Xee7eo OK OK OK UGh8Ang1Gu'.
  - `rating`: Rating.
  - `ssh_key_name`: Name of the ssh key that should be installed. Password login will be disabled.
  - `use_distribution_kernel`: Use the distribution's native kernel instead of the recommended OVH Kernel.
- `default_language`: The default language of this template.
- `deprecated`: is this distribution deprecated.
- `description`: information about this template.
- `distribution`: the distribution this template is based on.
- `family`: this template family type (bsd,linux,solaris,windows).
- `hard_raid_configuration`: This distribution supports hardware raid configuration through the OVH API.
- `filesystems`: Filesystems available (btrfs,ext3,ext4,ntfs,reiserfs,swap,ufs,xfs,zfs).
- `last_modification`: Date of last modification of the base image.
- `partition_scheme`:
  - `name`: name of this partitioning scheme.
  - `priority`: on a reinstall, if a partitioning scheme is not specified, the one with the higher priority will be used by default, among all the compatible partitioning schemes (given the underlying hardware specifications).
  - `hardware_raid`:

        * `name`: Hardware RAID name.
        * `disks`: Disk List. Syntax is cX:dY for disks and [cX:dY,cX:dY] for groups. With X and Y resp. the controller id and the disk id.
        * `mode`: RAID mode (raid0, raid1, raid10, raid5, raid50, raid6, raid60).
        * `step`: Specifies the creation order of the hardware RAID.
    - `partition`:
        * `filesystem`: Partition filesystem.
        * `mountpoint`: partition mount point.
        * `raid`: raid partition type.
        * `size`: size of partition in MB, 0 => rest of the space.
        * `order`: step or order. specifies the creation order of the partition on the disk
        * `type`: partition type.
        * `volume_name`: The volume name needed for proxmox distribution
- `supports_distribution_kernel`: This distribution supports installation using the distribution's native kernel instead of the recommended OVH kernel.
- `supports_gpt_label`: This distribution supports the GUID Partition Table (GPT), providing up to 128 partitions that can have more than 2TB.
- `supports_rtm`: This distribution supports RTM software.
- `supports_sql_server`: This distribution supports the microsoft SQL server.
- `supports_uefi`: This distribution supports UEFI setup (no,only,yes).

» ovh_me_installation_templates

Use this data source to get the list of custom installation templates available for dedicated servers.

» Example Usage

```
data "ovh_me_installation_templates" "templates" {}
```

» Argument Reference

This datasource takes no argument.

» Attributes Reference

The following attributes are exported:

- `result` - The list of custom installation templates IDs available for dedicated servers.

» ovh_me_paymentmean_bankaccount

Use this data source to retrieve information about a bank account payment mean associated with an OVH account.

» Example Usage

```
data "ovh_me_paymentmean_bankaccount" "ba" {
   use_default = true
}
```

» Argument Reference

- `description_regexp` - (Optional) a regexp used to filter bank accounts on their `description` attributes.
- `use_default` - (Optional) Retrieve bank account marked as default payment mean.
- `use_oldest` - (Optional) Retrieve oldest bank account. project.
- `state` - (Optional) Filter bank accounts on their `state` attribute. Can be "blockedForIncidents", "valid", "pendingValidation"

» Attributes Reference

`id` is set to the ID of the bank account payment mean

- `description` - the description attribute of the bank account
- `default` - a boolean which tells if the retrieved bank account is marked as the default payment mean

» ovh_me_paymentmean_creditcard

Use this data source to retrieve information about a credit card payment mean associated with an OVH account.

» Example Usage

```
data "ovh_me_paymentmean_creditcard" "cc" {
   use_default = true
}
```

» Argument Reference

- `description_regexp` - (Optional) a regexp used to filter credit cards on their `description` attributes.
- `use_default` - (Optional) Retrieve credit card marked as default payment mean.
- `use_last_to_expire` - (Optional) Retrieve the credit card that will be the last to expire according to its expiration date.
- `states` - (Optional) Filter credit cards on their `state` attribute. Can be "expired", "valid", "tooManyFailures"

» Attributes Reference

`id` is set to the ID of the credit card payment mean

- `description` - the description attribute of the credit card
- `state` - the state attribute of the credit card
- `default` - a boolean which tells if the retrieved credit card is marked as the default payment mean

» ovh__me__ssh__key

Use this data source to retrieve information about an SSH key.

» Example Usage

```
data "ovh_me_ssh_key" "mykey" {
   key_name = "mykey"
}
```

» Argument Reference

- `key_name` - (Required) The name of the SSH key.

» Attributes Reference

- `key_name` - See Argument Reference above.
- `key` - The content of the public key. E.g.: "ssh-ed25519 AAAAC3..."
- `default` - True when this public SSH key is used for rescue mode and reinstallations.

» ovh_me_ssh_keys

Use this data source to retrieve list of names of the account's SSH keys.

» Example Usage

```
data "ovh_me_ssh_keys" "mykeys" {}
```

» Argument Reference

This datasource takes no arguments.

» Attributes Reference

- `names` - The list of the names of all the SSH keys.

» ovh_publiccloud_region

DEPRECATED: Use `ovh_cloud_region` instead.

Use this data source to retrieve information about a region associated with a public cloud project. The region must be associated with the project.

» Example Usage

```
data "ovh_publiccloud_region" "GRA1" {
   project_id = "XXXXXX"
   region = "GRA1"
}
```

» Argument Reference

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.
- `region` - (Required) The name of the region associated with the public cloud project.

» Attributes Reference

`id` is set to the ID of the project concatenated with the name of the region. In addition, the following attributes are exported:

- `continent_code` - the code of the geographic continent the region is running. E.g.: EU for Europe, US for America...
- `datacenter_location` - The location code of the datacenter. E.g.: "GRA", meaning Gravelines, for region "GRA1"
- `continentCode` - (Deprecated) Use `continent_code` instead.
- `datacenterLocation` - (Deprecated) Use `datacenter_location` instead.
- `services` - The list of public cloud services running within the region
  - `name` - the name of the public cloud service
  - `status` - the status of the service

» ovh_publiccloud_regions

DEPRECATED: Use `ovh_cloud_regions` instead.

Use this data source to get the regions of a public cloud project.

» Example Usage

```
data "ovh_publiccloud_regions" "regions" {
  project_id = "XXXXXX"

  has_services_up = ["network"]
}
```

» Argument Reference

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.

- `has_services_up` - (Optional) List of services which has to be UP in regions. Example: ”image”, ”instance”, ”network”, ”storage”, ”volume”, ”workflow”, ... If left blank, returns all regions associated with the project_id.

» Attributes Reference

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

- `names` - The list of regions associated with the project, filtered by services UP.

» ovh_vracks

Use this data source to get the list of Vrack ids available for your OVH account.

» Example Usage

```
data ovh_vracks vracks {}
```

» Argument Reference

This datasource takes no argument.

» Attributes Reference

The following attributes are exported:

- - The list of vrack IDs available for your OVH account.

» ovh_cloud_network_private

Creates a private network in a public cloud project.

» Example Usage

```
resource "ovh_cloud_network_private" "net" {
   project_id = "67890"
   name       = "admin_network"
   regions    = ["GRA1", "BHS1"]
}
```

» Argument Reference

The following arguments are supported:

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.
- `name` - (Required) The name of the network.
- `vlan_id` - a vlan id to associate with the network. Changing this value recreates the resource. Defaults to 0.
- `regions` - an array of valid OVH public cloud region ID in which the network will be available. Ex.: "GRA1". Defaults to all public cloud regions.

» Attributes Reference

The following attributes are exported:

- `project_id` - See Argument Reference above.
- `name` - See Argument Reference above.
- `vlan_id` - See Argument Reference above.
- `regions` - See Argument Reference above.
- `regions_status` - A map representing the status of the network per region.
- `regions_status/region` - The id of the region.
- `regions_status/status` - The status of the network in the region.
- `status` - the status of the network. should be normally set to 'ACTIVE'.
- `type` - the type of the network. Either 'private' or 'public'.

» ovh_cloud_network_private_subnet

Creates a subnet in a private network of a public cloud project.

» Example Usage

```
resource "ovh_cloud_network_private_subnet" "subnet" {
   project_id = "67890"
   network_id = "0234543"
   region     = "GRA1"
   start      = "192.168.168.100"
   end        = "192.168.168.200"
   network    = "192.168.168.0/24"
   dhcp       = true
   no_gateway = false
}
```

» Argument Reference

The following arguments are supported:

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used. Changing this forces a new resource to be created.
- `network_id` - (Required) The id of the network. Changing this forces a new resource to be created.
- `dhcp` - (Optional) Enable DHCP. Changing this forces a new resource to be created. Defaults to false. _
- `start` - (Required) First ip for this region. Changing this value recreates the subnet.
- `end` - (Required) Last ip for this region. Changing this value recreates the subnet.
- `network` - (Required) Global network in CIDR format. Changing this value recreates the subnet
- `region` - The region in which the network subnet will be created. Ex.: "GRA1". Changing this value recreates the resource.
- `no_gateway` - Set to true if you don't want to set a default gateway IP. Changing this value recreates the resource. Defaults to false.

» Attributes Reference

The following attributes are exported:

- `project_id` - See Argument Reference above.
- `network_id` - See Argument Reference above.

- `dhcp_id` - See Argument Reference above.
- `start` - See Argument Reference above.
- `end` - See Argument Reference above.
- `network` - See Argument Reference above.
- `region` - See Argument Reference above.
- `gateway_ip` - The IP of the gateway
- `no_gateway` - See Argument Reference above.
- `cidr` - Ip Block representing the subnet cidr.
- `ip_pools` - List of ip pools allocated in the subnet.
- `ip_pools/network` - Global network with cidr.
- `ip_pools/region` - Region where this subnet is created.
- `ip_pools/dhcp` - DHCP enabled.
- `ip_pools/end` - Last ip for this region.
- `ip_pools/start` - First ip for this region.

» ovh_cloud_user

Creates a user in a public cloud project.

» Example Usage

```
resource "ovh_cloud_user" "user1" {
   project_id = "67890"
}
```

» Argument Reference

The following arguments are supported:

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.
- `description` - A description associated with the user.

» Attributes Reference

The following attributes are exported:

- `project_id` - See Argument Reference above.
- `description` - See Argument Reference above.
- `username` - the username generated for the user. This username can be used with the Openstack API.

- `password` - (Sensitive) the password generated for the user. The password can be used with the Openstack API. This attribute is sensitive and will only be retrieve once during creation.
- `status` - the status of the user. should be normally set to 'ok'.
- `creation_date` - the date the user was created.
- `openstack_rc` - a convenient map representing an openstack_rc file. Note: no password nor sensitive token is set in this map.

» ovh_publiccloud_private_network

DEPRECATED: Use `ovh_cloud_network_private` instead.

Creates a private network in a public cloud project.

» Example Usage

```
resource "ovh_publiccloud_private_network" "net" {
   project_id = "67890"
   name       = "admin_network"
   regions    = ["GRA1", "BHS1"]
}
```

» Argument Reference

The following arguments are supported:

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.
- `name` - (Required) The name of the network.
- `vlan_id` - a vlan id to associate with the network. Changing this value recreates the resource. Defaults to 0.
- `regions` - an array of valid OVH public cloud region ID in which the network will be available. Ex.: "GRA1". Defaults to all public cloud regions.

» Attributes Reference

The following attributes are exported:

- `project_id` - See Argument Reference above.
- `name` - See Argument Reference above.

- `vlan_id` - See Argument Reference above.
- `regions` - See Argument Reference above.
- `regions_status` - A map representing the status of the network per region.
- `regions_status/region` - The id of the region.
- `regions_status/status` - The status of the network in the region.
- `status` - the status of the network. should be normally set to 'ACTIVE'.
- `type` - the type of the network. Either 'private' or 'public'.

» ovh_publiccloud_private_network_subnet

DEPRECATED: Use `ovh_cloud_network_private_subnet` instead.

Creates a subnet in a private network of a public cloud project.

» Example Usage

```
resource "ovh_publiccloud_private_network_subnet" "subnet" {
   project_id = "67890"
   network_id = "0234543"
   region     = "GRA1"
   start      = "192.168.168.100"
   end        = "192.168.168.200"
   network    = "192.168.168.0/24"
   dhcp       = true
   no_gateway = false
}
```

» Argument Reference

The following arguments are supported:

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used. Changing this forces a new resource to be created.
- `network_id` - (Required) The id of the network. Changing this forces a new resource to be created.
- `dhcp` - (Optional) Enable DHCP. Changing this forces a new resource to be created. Defaults to false. _
- `start` - (Required) First ip for this region. Changing this value recreates the subnet.

- `end` - (Required) Last ip for this region. Changing this value recreates the subnet.
- `network` - (Required) Global network in CIDR format. Changing this value recreates the subnet
- `region` - The region in which the network subnet will be created. Ex.: "GRA1". Changing this value recreates the resource.
- `no_gateway` - Set to true if you don't want to set a default gateway IP. Changing this value recreates the resource. Defaults to false.

» Attributes Reference

The following attributes are exported:

- `project_id` - See Argument Reference above.
- `network_id` - See Argument Reference above.
- `dhcp_id` - See Argument Reference above.
- `start` - See Argument Reference above.
- `end` - See Argument Reference above.
- `network` - See Argument Reference above.
- `region` - See Argument Reference above.
- `gateway_ip` - The IP of the gateway
- `no_gateway` - See Argument Reference above.
- `cidr` - Ip Block representing the subnet cidr.
- `ip_pools` - List of ip pools allocated in the subnet.
- `ip_pools/network` - Global network with cidr.
- `ip_pools/region` - Region where this subnet is created.
- `ip_pools/dhcp` - DHCP enabled.
- `ip_pools/end` - Last ip for this region.
- `ip_pools/start` - First ip for this region.

» ovh_publiccloud_user

DEPRECATED: Use `ovh_cloud_user` instead.

Creates a user in a public cloud project.

» Example Usage

```
resource "ovh_publiccloud_user" "user1" {
   project_id = "67890"
}
```

» Argument Reference

The following arguments are supported:

- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.
- `description` - A description associated with the user.

» Attributes Reference

The following attributes are exported:

- `project_id` - See Argument Reference above.
- `description` - See Argument Reference above.
- `username` - the username generated for the user. This username can be used with the Openstack API.
- `password` - (Sensitive) the password generated for the user. The password can be used with the Openstack API. This attribute is sensitive and will only be retrieve once during creation.
- `status` - the status of the user. should be normally set to 'ok'.
- `creation_date` - the date the user was created.
- `openstack_rc` - a convenient map representing an openstack_rc file. Note: no password nor sensitive token is set in this map.

» ovh_dedicated_server_reboot_task

Reboot your Dedicated Server.

WARNING: After some delay, if the task is marked as `done`, the Provider may purge it. To avoid raising errors when terraform refreshes its plan, 404 errors are ignored on Resource Read, thus some information may be lost after a while.

» Example Usage

```
data ovh_dedicated_server_boots "rescue" {
  service_name = "ns00000.ip-1-2-3.eu"
  boot_type    = "rescue"
  kernel       = "rescue64-pro"
}

resource ovh_dedicated_server_update "server_on_rescue" {
  service_name = "ns00000.ip-1-2-3.eu"
  boot_id      = data.ovh_dedicated_server_boots.rescue.result[0]
```

```
  monitoring   = true
  state        = "ok"
}

resource ovh_dedicated_server_reboot_task "server_reboot" {
  service_name = data.ovh_dedicated_server_boots.rescue.service_name

  keepers = [
     ovh_dedicated_server_update.server_on_rescue.boot_id,
  ]
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The service_name of your dedicated server.
- `keepers` - List of values traccked to trigger reboot, used also to form implicit dependencies

» Attributes Reference

The following attributes are exported:

- `id` - The task id
- `comment` - Details of this task. (should be `Reboot asked`)
- `done_date` - Completion date in RFC3339 format.
- `function` - Function name (should be `hardReboot`).
- `last_update` - Last update in RFC3339 format.
- `start_date` - Task creation date in RFC3339 format.
- `status` - Task status (should be `done`)

» ovh_dedicated_server_update

Update various properties of your Dedicated Server.

WARNING: `rescue_mail` and `root_device` properties aren't updated consistently. This is an issue on the OVH API which has been reported. Meanwhile, these properties aren't not mapped on this terraform resource.

» Example Usage

```
data ovh_dedicated_server_boots "rescue" {
```

```
  service_name = "ns00000.ip-1-2-3.eu"
  boot_type    = "rescue"
  kernel       = "rescue64-pro"
}

resource ovh_dedicated_server_update "server" {
  service_name = "ns00000.ip-1-2-3.eu"
  boot_id      = data.ovh_dedicated_server_boots.rescue.result[0]
  monitoring   = true
  state        = "ok"
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The service_name of your dedicated server.
- `boot_id` - boot id of the server
- `monitoring` - Icmp monitoring state
- `state` - error, hacked, hackedBlocked, ok

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `boot_id` - See Argument Reference above.
- `monitoring` - See Argument Reference above.
- `state` - See Argument Reference above.

» ovh_domain_zone_record

Provides a OVH domain zone record.

» Example Usage

```
# Add a record to a sub-domain
resource "ovh_domain_zone_record" "test" {
    zone = "testdemo.ovh"
    subdomain = "test"
    fieldtype = "A"
    ttl = "3600"
```

```
    target = "0.0.0.0"
}
```

» Argument Reference

The following arguments are supported:

- `zone` - (Required) The domain to add the record to
- `subdomain` - (Required) The name of the record
- `target` - (Required) The value of the record
- `fieldtype` - (Required) The type of the record
- `ttl` - (Optional) The TTL of the record

» Attributes Reference

The following attributes are exported:

- `id` - The record ID
- `zone` - The domain to add the record to
- `subDomain` - The name of the record
- `target` - The value of the record
- `fieldType` - The type of the record
- `ttl` - The TTL of the record

» Import

OVH record can be imported using the `id` and the `zone`, eg:

```
$ terraform import ovh_domain_zone_record.test 1234OVH_ID.zone.tld
```

» ovh_domain_zone_redirection

Provides a OVH domain zone redirection.

» Example Usage

```
# Add a redirection to a sub-domain
resource "ovh_domain_zone_redirection" "test" {
    zone = "testdemo.ovh"
    subdomain = "test"
    type = "visiblePermanent"
    target = "http://www.ovh"
```

```
}
```

» Argument Reference

The following arguments are supported:

- `zone` - (Required) The domain to add the redirection to
- `subdomain` - (Optional) The name of the redirection
- `target` - (Required) The value of the redirection
- `type` - (Required) The type of the redirection, with values:
  - `visible` -> Redirection by http code 302
  - `visiblePermanent` -> Redirection by http code 301
  - `invisible` -> Redirection by html frame
- `description` - (Optional) A description of this redirection
- `keywords` - (Optional) Keywords to describe this redirection
- `title` - (Optional) Title of this redirection

» Attributes Reference

The following attributes are exported:

- `id` - The redirection ID
- `zone` - The domain to add the redirection to
- `subDomain` - The name of the redirection
- `target` - The value of the redirection
- `type` - The type of the redirection
- `description` - The description of the redirection
- `keywords` - Keywords of the redirection
- `title` - The title of the redirection

» ovh_ip_reverse

Provides a OVH IP reverse.

» Example Usage

```
# Set the reverse of an IP
resource "ovh_ip_reverse" "test" {
    ip = "192.0.2.0/24"
    ipreverse = "192.0.2.1"
    reverse = "example.com"
}
```

» Argument Reference

The following arguments are supported:

- `ip` - (Required) The IP block to which the IP belongs
- `reverse` - (Required) The value of the reverse
- `ipreverse` - (Optional) The IP to set the reverse of, default to `ip` if `ip` is a /32 (IPv4) or a /128 (IPv6)

» Attributes Reference

The following attributes are exported:

- `ipreverse` - The IP to set the reverse of
- `reverse` - The value of the reverse

» ovh_iploadbalancing_http_farm

Creates a http backend server group (farm) to be used by loadbalancing frontend(s)

» Example Usage

```
data "ovh_iploadbalancing" "lb" {
  service_name = "ip-1.2.3.4"
   state       = "ok"
}

resource "ovh_iploadbalancing_http_farm" "farmname" {
  service_name = "${data.ovh_iploadbalancing.lb.id}"
  display_name = "ingress-8080-gra"
  zone = "GRA"
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `balance` - Load balancing algorithm. `roundrobin` if null (`first`, `leastconn`, `roundrobin`, `source`)
- `display_name` - Readable label for loadbalancer farm

- `port` - Port attached to your farm ([1..49151]). Inherited from frontend if null
- `stickiness` - Stickiness type. No stickiness if null (`sourceIp`)
- `vrack_network_id` - Internal Load Balancer identifier of the vRack private network to attach to your farm, mandatory when your Load Balancer is attached to a vRack
- `zone` - (Required) Zone where the farm will be defined (ie. `GRA`, `BHS` also supports `ALL`)
- `probe` - define a backend healthcheck probe
  - `type` - (Required) Valid values : `http`, `internal`, `mysql`, `oko`, `pgsql`, `smtp`, `tcp`
  - `interval` - probe interval, Value between 30 and 3600 seconds, default 30
  - `match` - What to mach `pattern` against (`contains`, `default`, `internal`, `matches`, `status`)
  - `port` - Port for backends to recieve traffic on.
  - `negate` - Negate probe result
  - `pattern` - Pattern to match against `match`
  - `force_ssl` - Force use of SSL (TLS)
  - `url` - URL for HTTP probe type.
  - `method` - HTTP probe method (`GET`, `HEAD`, `OPTIONS`, `internal`)

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `balance` - See Argument Reference above.
- `display_name` - See Argument Reference above.
- `port` - See Argument Reference above.
- `stickiness` - See Argument Reference above.
- `vrack_network_id` - See Argument Reference above.
- `zone` - See Argument Reference above.
- `probe` - See Argument Reference above.
  - `type` - See Argument Reference above.
  - `interval` - See Argument Reference above.
  - `match` - See Argument Reference above.
  - `port` - See Argument Reference above.
  - `negate` - See Argument Reference above.
  - `pattern` - See Argument Reference above.
  - `force_ssl` - See Argument Reference above.
  - `url` - See Argument Reference above.
  - `method` - See Argument Reference above.

» ovh_iploadbalancing_tcp_farm

Creates a backend server group (farm) to be used by loadbalancing frontend(s)

» Example Usage

```
data "ovh_iploadbalancing" "lb" {
  service_name = "ip-1.2.3.4"
   state       = "ok"
}

resource "ovh_iploadbalancing_tcp_farm" "farmname" {
  service_name = "${data.ovh_iploadbalancing.lb.id}"
  display_name = "ingress-8080-gra"
  zone = "GRA"
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `balance` - Load balancing algorithm. `roundrobin` if null (`first`, `leastconn`, `roundrobin`, `source`)
- `display_name` - Readable label for loadbalancer farm
- `port` - Port attached to your farm ([1..49151]). Inherited from frontend if null
- `stickiness` - Stickiness type. No stickiness if null (`sourceIp`)
- `vrack_network_id` - Internal Load Balancer identifier of the vRack private network to attach to your farm, mandatory when your Load Balancer is attached to a vRack
- `zone` - (Required) Zone where the farm will be defined (ie. `GRA`, `BHS` also supports `ALL`)
- `probe` - define a backend healthcheck probe
  - `type` - (Required) Valid values : `http`, `internal`, `mysql`, `oko`, `pgsql`, `smtp`, `tcp`
  - `interval` - probe interval, Value between 30 and 3600 seconds, default 30
  - `match` - What to mach `pattern` against (`contains`, `default`, `internal`, `matches`, `status`)
  - `port` - Port for backends to recieve traffic on.
  - `negate` - Negate probe result
  - `pattern` - Pattern to match against `match`
  - `force_ssl` - Force use of SSL (TLS)

  - `url` - URL for HTTP probe type.
  - `method` - HTTP probe method (`GET`, `HEAD`, `OPTIONS`, `internal`)

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `balance` - See Argument Reference above.
- `display_name` - See Argument Reference above.
- `port` - See Argument Reference above.
- `stickiness` - See Argument Reference above.
- `vrack_network_id` - See Argument Reference above.
- `zone` - See Argument Reference above.
- `probe` - See Argument Reference above.
  - `type` - See Argument Reference above.
  - `interval` - See Argument Reference above.
  - `match` - See Argument Reference above.
  - `port` - See Argument Reference above.
  - `negate` - See Argument Reference above.
  - `pattern` - See Argument Reference above.
  - `force_ssl` - See Argument Reference above.
  - `url` - See Argument Reference above.
  - `method` - See Argument Reference above.

» ovh_iploadbalancing_http_farm_server

Creates a backend server entry linked to http loadbalancing group (farm)

» Example Usage

```
data "ovh_iploadbalancing" "lb" {
  service_name = "ip-1.2.3.4"
   state       = "ok"
}

resource "ovh_iploadbalancing_http_farm" "farmname" {
  service_name = "${data.ovh_iploadbalancing.lb.id}"
  port = 8080
  zone = "all"
}

resource "ovh_iploadbalancing_http_farm_server" "backend" {
```

```
  service_name           = "${data.ovh_iploadbalancing.lb.id}"
  farm_id                = "${ovh_iploadbalancing_http_farm.farmname.id}"
  display_name           = "mybackend"
  address                = "4.5.6.7"
  status                 = "active"
  port                   = 80
  proxy_protocol_version = v2
  weight                 = 2
  probe                  = true
  ssl                    = false
  backup                 = true
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `farm_id` - ID of the farm this server is attached to
- `display_name` - Label for the server
- `address` - Address of the backend server (IP from either internal or OVH network)
- `status` - backend status - `active` or `inactive`
- `port` - Port that backend will respond on
- `proxy_protocol_version` - version of the PROXY protocol used to pass origin connection information from loadbalancer to recieving service (`v1`, `v2`, `v2-ssl`, `v2-ssl-cn`)
- `weight` - used in loadbalancing algorithm
- `probe` - defines if backend will be probed to determine health and keep as active in farm if healthy
- `ssl` - is the connection ciphered with SSL (TLS)
- `backup` - is it a backup server used in case of failure of all the non-backup backends

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `farm_id` - See Argument Reference above.
- `display_name` - See Argument Reference above.
- `address` - See Argument Reference above.
- `status` - See Argument Reference above.
- `port` - See Argument Reference above.
- `proxy_protocol_version` - See Argument Reference above.

- `weight` - See Argument Reference above.
- `probe` - See Argument Reference above.
- `ssl` - See Argument Reference above.
- `backup` - See Argument Reference above.
- `cookie` - Value of the stickiness cookie used for this backend.

» ovh_iploadbalancing_tcp_farm_server

Creates a backend server entry linked to loadbalancing group (farm)

» Example Usage

```
data "ovh_iploadbalancing" "lb" {
  service_name = "ip-1.2.3.4"
   state       = "ok"
}

resource "ovh_iploadbalancing_tcp_farm" "farmname" {
  service_name = "${data.ovh_iploadbalancing.lb.id}"
  port = 8080
  zone = "all"
}

resource "ovh_iploadbalancing_tcp_farm_server" "backend" {
  service_name           = "${data.ovh_iploadbalancing.lb.id}"
  farm_id                = "${ovh_iploadbalancing_tcp_farm.farmname.id}"
  display_name           = "mybackend"
  address                = "4.5.6.7"
  status                 = "active"
  port                   = 80
  proxy_protocol_version = v2
  weight                 = 2
  probe                  = true
  ssl                    = false
  backup                 = true
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `farm_id` - ID of the farm this server is attached to

- `display_name` - Label for the server
- `address` - Address of the backend server (IP from either internal or OVH network)
- `status` - backend status - `active` or `inactive`
- `port` - Port that backend will respond on
- `proxy_protocol_version` - version of the PROXY protocol used to pass origin connection information from loadbalancer to recieving service (`v1`, `v2`, `v2-ssl`, `v2-ssl-cn`)
- `weight` - used in loadbalancing algorithm
- `probe` - defines if backend will be probed to determine health and keep as active in farm if healthy
- `ssl` - is the connection ciphered with SSL (TLS)
- `backup` - is it a backup server used in case of failure of all the non-backup backends

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `farm_id` - See Argument Reference above.
- `display_name` - See Argument Reference above.
- `address` - See Argument Reference above.
- `status` - See Argument Reference above.
- `port` - See Argument Reference above.
- `proxy_protocol_version` - See Argument Reference above.
- `weight` - See Argument Reference above.
- `probe` - See Argument Reference above.
- `ssl` - See Argument Reference above.
- `backup` - See Argument Reference above.
- `cookie` - Value of the stickiness cookie used for this backend.

» ovh_iploadbalancing_http_frontend

Creates a backend http server group (frontend) to be used by loadbalancing frontend(s)

» Example Usage

```
data "ovh_iploadbalancing" "lb" {
  service_name = "ip-1.2.3.4"
  state       = "ok"
}
```

```
resource "ovh_iploadbalancing_http_farm" "farm80" {
   service_name = "${data.ovh_iploadbalancing.lb.service_name}"
   display_name = "ingress-8080-gra"
   zone = "all"
   port = 80
}

resource "ovh_iploadbalancing_http_frontend" "testfrontend" {
   service_name = "${data.ovh_iploadbalancing.lb.service_name}"
   display_name = "ingress-8080-gra"
   zone = "all"
   port = "80,443"
   default_farm_id = "${ovh_iploadbalancing_http_farm.farm80.id}"
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `display_name` - Human readable name for your frontend, this field is for you
- `port` - Port(s) attached to your frontend. Supports single port (numerical value), range (2 dash-delimited increasing ports) and comma-separated list of 'single port' and/or 'range'. Each port must be in the [1;49151] range
- `zone` - (Required) Zone where the frontend will be defined (ie. `gra`, `bhs` also supports `all`)
- `allowed_source` - Restrict IP Load Balancing access to these ip block. No restriction if null. List of IP blocks.
- `dedicated_ipfo` - Only attach frontend on these ip. No restriction if null. List of Ip blocks.
- `default_farm_id` - Default TCP Farm of your frontend
- `default_ssl_id` - Default ssl served to your customer
- `disabled` - Disable your frontend. Default: 'false'
- `ssl` - SSL deciphering. Default: 'false'

» Attributes Reference

The following attributes are exported:

- `id` - Id of your frontend
- `display_name` - See Argument Reference above.
- `allowed_source` - See Argument Reference above.

- `dedicated_ipfo` - See Argument Reference above.
- `default_farm_id` - See Argument Reference above.
- `default_ssl_id` - See Argument Reference above.
- `disabled` - See Argument Reference above.
- `ssl` - See Argument Reference above.

» ovh_iploadbalancing_tcp_frontend

Creates a backend server group (frontend) to be used by loadbalancing frontend(s)

» Example Usage

```
data "ovh_iploadbalancing" "lb" {
  service_name = "ip-1.2.3.4"
  state       = "ok"
}

resource "ovh_iploadbalancing_tcp_farm" "farm80" {
   service_name = "${data.ovh_iploadbalancing.lb.service_name}"
   display_name = "ingress-8080-gra"
   zone = "all"
   port = 80
}

resource "ovh_iploadbalancing_tcp_frontend" "testfrontend" {
   service_name = "${data.ovh_iploadbalancing.lb.service_name}"
   display_name = "ingress-8080-gra"
   zone = "all"
   port = "80,443"
   default_farm_id = "${ovh_iploadbalancing_tcp_farm.farm80.id}"
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `display_name` - Human readable name for your frontend, this field is for you
- `port` - Port(s) attached to your frontend. Supports single port (numerical value), range (2 dash-delimited increasing ports) and comma-separated

list of 'single port' and/or 'range'. Each port must be in the [1;49151] range
- `zone` - (Required) Zone where the frontend will be defined (ie. `gra`, `bhs` also supports `all`)
- `allowed_source` - Restrict IP Load Balancing access to these ip block. No restriction if null. List of IP blocks.
- `dedicated_ipfo` - Only attach frontend on these ip. No restriction if null. List of Ip blocks.
- `default_farm_id` - Default TCP Farm of your frontend
- `default_ssl_id` - Default ssl served to your customer
- `disabled` - Disable your frontend. Default: 'false'
- `ssl` - SSL deciphering. Default: 'false'

» Attributes Reference

The following attributes are exported:

- `id` - Id of your frontend
- `display_name` - See Argument Reference above.
- `allowed_source` - See Argument Reference above.
- `dedicated_ipfo` - See Argument Reference above.
- `default_farm_id` - See Argument Reference above.
- `default_ssl_id` - See Argument Reference above.
- `disabled` - See Argument Reference above.
- `ssl` - See Argument Reference above.

» ovh_iploadbalancing_http_route

Manage http route for a loadbalancer service

» Example Usage

Route which redirect all url to https.

```
resource "ovh_iploadbalancing_http_route" "httpsredirect" {
  service_name = "loadbalancer-xxxxxxxxxxxxxxxxxx"
  display_name = "Redirect to HTTPS"
  weight = 1

  action {
    status = 302
    target = "https://$${host}$${path}$${arguments}"
    type = "redirect"
```

```
  }
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `display_name` - Human readable name for your route, this field is for you
- `weight` - Route priority ([0..255]). 0 if null. Highest priority routes are evaluated first. Only the first matching route will trigger an action
- `action.status` - HTTP status code for ”redirect” and ”reject” actions
- `action.target` - Farm ID for ”farm” action type or URL template for ”redirect” action. You may use ${uri}, ${protocol}, ${host}, ${port} and ${path} variables in redirect target
- `action.type` - (Required) Action to trigger if all the rules of this route matches
- `frontend_id` - Route traffic for this frontend

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `display_name` - See Argument Reference above.
- `weight` - See Argument Reference above.
- `action.status` - See Argument Reference above.
- `action.target` - See Argument Reference above.
- `action.type` - See Argument Reference above.
- `frontend_id` - See Argument Reference above.

» ovh_iploadbalancing_http_route_rule

Manage rules for HTTP route.

» Example Usage

Route which redirect all url to https for example.com (Vhost).

```
resource "ovh_iploadbalancing_http_route" "httpsredirect" {
  service_name = "loadbalancer-xxxxxxxxxxxxxxxxx"
  display_name = "Redirect to HTTPS"
  weight       = 1
```

```
  frontend_id  = 11111

  action {
    status = 302
    target = "https://$${host}$${path}$${arguments}"
    type = "redirect"
  }
}

resource "ovh_iploadbalancing_http_route_rule" "examplerule" {
  service_name = "loadbalancer-xxxxxxxxxxxxxxxxx"
  route_id     = "${ovh_iploadbalancing_http_route.httpsredirect.id}"
  display_name = "Match example.com host"
  field        = "host"
  match        = "is"
  negate       = false
  pattern      = "example.com"
}
```

Rule which match a specific header (same effect as the host match above).

```
resource "ovh_iploadbalancing_http_route_rule" "examplerule" {
  service_name = "loadbalancer-xxxxxxxxxxxxxxxxx"
  route_id     = "${ovh_iploadbalancing_http_route.httpsredirect.id}"
  display_name = "Match example.com Host header"
  field        = "headers"
  match        = "is"
  negate       = false
  pattern      = "example.com"
  sub_field    = "Host"
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `route_id` - (Required) The route to apply this rule
- `display_name` - Human readable name for your rule, this field is for you
- `field` - (Required) Name of the field to match like ”protocol” or ”host”. See ”/ipLoadbalancing/{serviceName}/availableRouteRules” for a list of available rules
- `match` - (Required) Matching operator. Not all operators are available for all fields. See ”/ipLoadbalancing/{serviceName}/availableRouteRules”
- `negate` - Invert the matching operator effect

- `pattern` - Value to match against this match. Interpretation if this field depends on the match and field
- `sub_field` - Name of sub-field, if applicable. This may be a Cookie or Header name for instance

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `route_id` - See Argument Reference above.
- `display_name` - See Argument Reference above.
- `field` - See Argument Reference above.
- `match` - See Argument Reference above.
- `negate` - See Argument Reference above.
- `pattern` - See Argument Reference above.
- `sub_field` - See Argument Reference above.

» ovh_iploadbalancing_refresh

Applies changes from other ovh_iploadbalancing_* resourcesto the production configuration of loadbalancers.

» Example Usage

```
data "ovh_iploadbalancing" "lb" {
  service_name = "ip-1.2.3.4"
   state       = "ok"
}

resource "ovh_iploadbalancing_tcp_farm" "farmname" {
  service_name = "${data.ovh_iploadbalancing.lb.id}"
  port = 8080
  zone = "all"
}

resource "ovh_iploadbalancing_tcp_farm_server" "backend" {
  service_name           = "${data.ovh_iploadbalancing.lb.id}"
  farm_id                = "${ovh_iploadbalancing_tcp_farm.farmname.id}"
  display_name           = "mybackend"
  address                = "4.5.6.7"
  status                 = "active"
  port                   = 80
```

```
  proxy_protocol_version = v2
  weight                 = 2
  probe                  = true
  ssl                    = false
  backup                 = true
}

resource "ovh_iploadbalancing_refresh" "mylb" {
  service_name = "${data.ovh_iploadbalancing.lb.id}"
  keepers = [
    "${ovh_iploadbalancing_tcp_farm_server.backend.*.address}",
    ]
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `keepers` - List of values traccked to trigger refresh, used also to form implicit dependencies

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `keepers` - See Argument Reference above.

» ovh_iploadbalancing_vrack_network

Manage a vrack network for your IP Loadbalancing service.

» Example Usage

```
data ovh_iploadbalancing "iplb" {
  service_name = "loadbalancer-xxxxxxxxxxxxxxxxx"
}

resource "ovh_vrack_iploadbalancing" "viplb" {
  service_name     = "xxx"
  ip_loadbalancing = data.ovh_iploadbalancing.iplb.service_name
```

```
}

resource ovh_iploadbalancing_vrack_network "network" {
  service_name = ovh_vrack_iploadbalancing.viplb.ip_loadbalancing
  subnet       = "10.0.0.0/16"
  vlan         = 1
  nat_ip       = "10.0.0.0/27"
  display_name = "mynetwork"
}

resource "ovh_iploadbalancing_tcp_farm" "testfarm" {
  service_name     = ovh_iploadbalancing_vrack_network.network.service_name
  display_name     = "mytcpbackends"
  port             = 80
  vrack_network_id = ovh_iploadbalancing_vrack_network.network.vrack_network_id
  zone             = tolist(data.ovh_iploadbalancing.iplb.zone)[0]
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The internal name of your IP load balancing
- `display_name` - Human readable name for your vrack network
- `farm_id` - List of existing farm ids your vRack network is attached to
- `nat_ip` - (Required) An IP block used as a pool of IPs by this Load Balancer to connect to the servers in this private network. The blck must be in the private network and reserved for the Load Balancer
- `subnet` - (Required) IP block of the private network in the vRack
- `vlan` - VLAN of the private network in the vRack. 0 if the private network is not in a VLAN

» Attributes Reference

The following attributes are exported:

- `vrack_network_id` - (Required) Internal Load Balancer identifier of the vRack private network

» ovh_me_installation_template

Use this resource to create a custom installation template available for dedicated servers.

» Example Usage

```
resource "ovh_me_installation_template" "mytemplate" {
  base_template_name = "centos7_64"
  template_name      = "mytemplate"
  default_language   = "fr"
}
```

» Argument Reference

- `available_languages`: List of all language available for this template.
- `base_template_name`: (Required) OVH template name yours will be based on, choose one among the list given by compatibleTemplates function.
- `beta`: This distribution is new and, although tested and functional, may still display odd behaviour.
- `bit_format`: This template bit format (32 or 64).
- `category`: Category of this template (informative only). (basic, customer, hosting, other, readyToUse, virtualisation).
- `customization`:
  - `change_log`: Template change log details.
  - `custom_hostname`: Set up the server using the provided hostname instead of the default hostname.
  - `post_installation_script_link`: Indicate the URL where your postinstall customisation script is located.
  - `post_installation_script_return`: indicate the string returned by your postinstall customisation script on successful execution. Advice: your script should return a unique validation string in case of succes. A good example is 'loh1Xee7eo OK OK OK UGh8Ang1Gu'.
  - `rating`: Rating.
  - `ssh_key_name`: Name of the ssh key that should be installed. Password login will be disabled.
  - `use_distribution_kernel`: Use the distribution's native kernel instead of the recommended OV
- `default_language`: (Required) The default language of this template.
- `deprecated`: is this distribution deprecated.
- `description`: information about this template.
- `distribution`: the distribution this template is based on.
- `family`: this template family type (bsd,linux,solaris,windows).
- `filesystems`: Filesystems available (btrfs,ext3,ext4,ntfs,reiserfs,swap,ufs,xfs,zfs).
- `hard_raid_configuration`: This distribution supports hardware raid configuration through the OVH API.
- `last_modification`: Date of last modification of the base image.
- `remove_default_partition_schemes`: (Required) Remove default partition schemes at creation.

- `supports_distribution_kernel`: This distribution supports installation using the distribution's native kernel instead of the recommended OVH kernel.
- `supports_gpt_label`: This distribution supports the GUID Partition Table (GPT), providing up to 128 partitions that can have more than 2TB.
- `supports_rtm`: This distribution supports RTM software.
- `supports_sql_server`: This distribution supports the microsoft SQL server.
- `supports_uefi`: This distribution supports UEFI setup (no,only,yes).
- `template_name`: (Required) This template name.

» Attributes Reference

The following attributes are exported in addition to the arguments above:

- `id`: This template name.

» Import

Use the following id format to import the resource : `base_template_name/template_name`

» ovh_me_installation_template_partition_scheme

Use this resource to create partition scheme for a custom installation template available for dedicated servers.

» Example Usage

```
resource "ovh_me_installation_template" "mytemplate" {
  base_template_name = "centos7_64"
  template_name      = "mytemplate"
  default_language   = "fr"
}

resource "ovh_me_installation_template_partition_scheme" "scheme" {
  template_name      = ovh_me_installation_template.mytemplate.template_name
  name               = "myscheme"
  priority           = 1
}
```

» Argument Reference

- `template_name`: (Required) The template name of the partition scheme.
- `name`: (Required) (Required) This partition scheme name.
- `priority`: on a reinstall, if a partitioning scheme is not specified, the one with the higher priority will be used by default, among all the compatible partitioning schemes (given the underlying hardware specifications).

» Attributes Reference

The following attributes are exported in addition to the arguments above:

- `id`: a fake id associated with this partition scheme formatted as follow: `template_name/name`

» Import

Use the fake id format to import the resource : `template_name/name`

» ovh__me__installation__template__partition__scheme__hardware__rai

Use this resource to create a hardware raid group in the partition scheme of a custom installation template available for dedicated servers.

» Example Usage

```
resource "ovh_me_installation_template" "mytemplate" {
  base_template_name = "centos7_64"
  template_name      = "mytemplate"
  default_language   = "fr"
}

resource "ovh_me_installation_template_partition_scheme" "scheme" {
  template_name      = ovh_me_installation_template.mytemplate.template_name
  name               = "myscheme"
  priority           = 1
}

resource "ovh_me_installation_template_partition_scheme_hardware_raid" "group1" {
  template_name = ovh_me_installation_template_partition_scheme.scheme.template_name
  scheme_name   = ovh_me_installation_template_partition_scheme.scheme.name
  name          = "group1"
```

```
  disks         = ["[c1:d1,c1:d2,c1:d3]", "[c1:d10,c1:d20,c1:d30]"]
  mode          = "raid50"
  step          = 1
}
```

» Argument Reference

- `disks`: Disk List. Syntax is cX:dY for disks and [cX:dY,cX:dY] for groups. With X and Y resp. the controller id and the disk id.
- `mode`: RAID mode (raid0, raid1, raid10, raid5, raid50, raid6, raid60).
- `name`: Hardware RAID name.
- `scheme_name`: (Required) The partition scheme name.
- `step`: Specifies the creation order of the hardware RAID.
- `template_name`: (Required) The template name of the partition scheme.

» Attributes Reference

The following attributes are exported in addition to the arguments above:

- `id`: a fake id associated with this partition scheme hardware raid group formatted as follow: `template_name/scheme_name/name`

» Import

Use the fake id format to import the resource : `template_name/scheme_name/name`.

» ovh_me_installation_template_partition_scheme_partition

Use this resource to create a partition in the partition scheme of a custom installation template available for dedicated servers.

» Example Usage

```
resource "ovh_me_installation_template" "mytemplate" {
  base_template_name = "centos7_64"
  template_name      = "mytemplate"
  default_language   = "fr"
}

resource "ovh_me_installation_template_partition_scheme" "scheme" {
  template_name      = ovh_me_installation_template.mytemplate.template_name
```

```
  name               = "myscheme"
  priority           = 1
}

resource "ovh_me_installation_template_partition_scheme_partition" "root" {
  template_name = ovh_me_installation_template_partition_scheme.scheme.template_name
  scheme_name   = ovh_me_installation_template_partition_scheme.scheme.name
  mountpoint    = "/"
  filesystem    = "ext4"
  size          = "400"
  order         = 1
  type          = "primary"
}
```

» Argument Reference

- `filesystem`: Partition filesystem.
- `mountpoint`: (Required) partition mount point.
- `order`: step or order. specifies the creation order of the partition on the disk
- `raid`: raid partition type.
- `scheme_name`: (Required) The partition scheme name.
- `size`: size of partition in MB, 0 => rest of the space.
- `template_name`: (Required) The template name of the partition scheme.
- `type`: partition type.
- `volume_name`: The volume name needed for proxmox distribution

» Attributes Reference

The following attributes are exported in addition to the arguments above:

- `id`: a fake id associated with this partition scheme partition formatted as follow: `template_name/scheme_name/mountpoint`

» Import

Use the fake id format to import the resource : `template_name/scheme_name/mountpoint` (example: "mytemplate/myscheme//").

» ovh_me_ssh_key

Creates an SSH Key.

» Example Usage

```
resource "ovh_me_ssh_key" "mykey" {
  key_name = "mykey"
  key      = "ssh-ed25519 AAAAC3..."
}
```

» Argument Reference

The following arguments are supported:

- `key_name` - (Required) The friendly name of this SSH key.
- `key` - (Required) The content of the public key in the form "ssh-algo content", e.g. "ssh-ed25519 AAAAC3...".
- `default` - True when this public SSH key is used for rescue mode and reinstallations.

» Attributes Reference

The following attributes are exported:

- `key_name` - See Argument Reference above.
- `key` - See Argument Reference above.
- `default` - See Argument Reference above.

» ovh_vrack_cloudproject

Attach a Public Cloud Project to a VRack.

» Example Usage

```
resource "ovh_vrack_cloudproject" "vcp" {
  vrack_id   = "12345"
  project_id = "67890"
}
```

» Argument Reference

The following arguments are supported:

- `vrack_id` - (Required) The id of the vrack. If omitted, the `OVH_VRACK_ID` environment variable is used. Note: The use of environment variable is deprecated.
- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used. Note: The use of environment variable is deprecated.

» Attributes Reference

The following attributes are exported:

- `vrack_id` - See Argument Reference above.
- `project_id` - See Argument Reference above.

» ovh_vrack_dedicated_server

Attach a dedicated server to a VRack.

» Example Usage

```
resource "ovh_vrack_dedicated_server" "vds" {
  vrack_id   = "12345"
  server_id = "67890"
}
```

» Argument Reference

The following arguments are supported:

- `vrack_id` - (Required) The id of the vrack.
- `server_id` - (Required) The id of the dedicated server.

» Attributes Reference

The following attributes are exported:

- `vrack_id` - See Argument Reference above.
- `server_id` - See Argument Reference above.

» ovh_vrack_dedicated_server_interface

Attach a Dedicated Server Network Interface to a VRack.

» Example Usage

```
resource "ovh_vrack_dedicated_server_interface" "vdsi" {
  vrack_id   = "12345"
  interface_id = "67890"
}
```

» Argument Reference

The following arguments are supported:

- `vrack_id` - (Required) The id of the vrack.
- `interface_id` - (Required) The id of dedicated server network interface.

» Attributes Reference

The following attributes are exported:

- `vrack_id` - See Argument Reference above.
- `interface_id` - See Argument Reference above.

» ovh_vrack_iploadbalancing

Attach a ip loadbalancing to a VRack.

» Example Usage

```
resource "ovh_vrack_iploadbalancing" "viplb" {
  service_name   = "xxx"
  ip_loadbalancing = "yyy"
}
```

» Argument Reference

The following arguments are supported:

- `service_name` - (Required) The id of the vrack.

- `ip_loadbalancing` - (Required) The id of the ip loadbalancing.

» Attributes Reference

The following attributes are exported:

- `service_name` - See Argument Reference above.
- `ip_loadbalancing` - See Argument Reference above.

» ovh_vrack_publiccloud_attachment

DEPRECATED: Use `ovh_vrack_cloudproject` instead.

Attach an existing PublicCloud project to an existing VRack.

» Example Usage

```
resource "ovh_vrack_publiccloud_attachment" "attach" {
  vrack_id   = "12345"
  project_id = "67890"
}
```

» Argument Reference

The following arguments are supported:

- `vrack_id` - (Required) The id of the vrack. If omitted, the `OVH_VRACK_ID` environment variable is used.
- `project_id` - (Required) The id of the public cloud project. If omitted, the `OVH_PROJECT_ID` environment variable is used.

» Attributes Reference

The following attributes are exported:

- `vrack_id` - See Argument Reference above.
- `project_id` - See Argument Reference above.

» Notes

The vrack attachment isn't a proper resource with an ID. As such, the resource id will be forged from the vrack and project ids and there's no correct way to import the resource in terraform. When the resource is created by terraform, it first checks if the attachment already exists within OVH infrastructure; if it exists it set the resource id without modifying anything. Otherwise, it will try to attach the vrack with the public cloud project.