» vcd_lb_service_monitor

Provides a vCloud Director Edge Gateway Load Balancer Service Monitor data source. A service monitor defines health check parameters for a particular type of network traffic. It can be associated with a pool. Pool members are monitored according to the service monitor parameters. See example usage of this data source in server pool resource page.

Note: See additional support notes in service monitor resource page.

Supported in provider *v2.4+*

» Example Usage

```
data "vcd_lb_service_monitor" "my-monitor" {
  org                 = "my-org"
  vdc                 = "my-org-vdc"
  edge_gateway        = "my-edge-gw"

  name = "not-managed"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the service monitor is defined
- `name` - (Required) Service Monitor name for identifying the exact service monitor

» Attribute Reference

All the attributes defined in `vcd_lb_service_monitor` resource are available.

» vcd_lb_server_pool

Provides a vCloud Director Edge Gateway Load Balancer Server Pool data source. A Server Pool defines a group of backend servers (defined as pool members), manages load balancer distribution methods, and has a service monitor attached to it for health check parameters.

Note: See additional support notes in server pool resource page.

Supported in provider *v2.4+*

» Example Usage

```
data "vcd_lb_server_pool" "sp-ds" {
  org                 = "my-org"
  vdc                 = "my-org-vdc"
  edge_gateway        = "my-edge-gw"

  name = "not-managed"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the server pool is defined
- `name` - (Required) Server Pool name for identifying the exact server pool

» Attribute Reference

All the attributes defined in `vcd_lb_server_pool` resource are available.

» vcd_lb_app_profile

Provides a vCloud Director Edge Gateway Load Balancer Application Profile data source. An application profile defines the behavior of the load balancer for a particular type of network traffic. After configuring a profile, you associate it

with a virtual server. The virtual server then processes traffic according to the values specified in the profile.

Note: See additional support notes in application profile resource page.

Supported in provider *v2.4+*

» Example Usage

```
data "vcd_lb_app_profile" "my-profile" {
  org                 = "my-org"
  vdc                 = "my-org-vdc"
  edge_gateway        = "my-edge-gw"

  name = "not-managed"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the service monitor is defined
- `name` - (Required) Application profile name for identifying the exact application profile

» Attribute Reference

All the attributes defined in `vcd_lb_app_profile` resource are available.

» vcd__lb__app__rule

Provides a vCloud Director Edge Gateway Load Balancer Application Rule data source. An application rule allows to directly manipulate and manage IP application traffic with load balancer.

Note: See additional support notes in application rule resource page.

Supported in provider *v2.4+*

» Example Usage

```
data "vcd_lb_app_rule" "my-rule" {
  org                 = "my-org"
  vdc                 = "my-org-vdc"
  edge_gateway        = "my-edge-gw"

  name = "not-managed"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the service monitor is defined
- `name` - (Required) Application rule name for identifying the exact application rule

» Attribute Reference

All the attributes defined in `vcd_lb_app_rule` resource are available.

» vcd_lb_virtual_server

Provides a vCloud Director edge gateway load balancer virtual server data source. Adds an edge gateway internal or uplink interface as a virtual server. A virtual server has a public IP address and services all incoming client requests.

Note: To make load balancing work one must ensure that load balancing is enabled on edge gateway (edge gateway must be advanced). This depends on NSX version to work properly. Please refer to VMware Product Interoperability Matrices to check supported vCloud director and NSX for vSphere configurations.

Note: The vCloud Director API for NSX supports a subset of the operations and objects defined in the NSX vSphere API Guide. The API supports NSX 6.2, 6.3, and 6.4.

Supported in provider *v2.4+*

» Example Usage

```
data "vcd_lb_virtual_server" "my-vs" {
  org                 = "my-org"
  vdc                 = "my-org-vdc"
  edge_gateway        = "my-edge-gw"

  name = "not-managed"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the virtual server is defined
- `name` - (Required) Name for identifying the exact virtual server

» Attribute Reference

All the attributes defined in `vcd_lb_virtual_server` resource are available.

» vcd_org

Provides a vCloud Director Org resource. This can be used to create and delete an organization. Requires system administrator privileges.

Supported in provider *v2.0+*

» Example Usage

```
provider "vcd" {
  user     = "${var.admin_user}"
  password = "${var.admin_password}"
  org      = "System"
  url      = "https://AcmeVcd/api"
}
```

```
resource "vcd_org" "my-org" {
  name             = "my-org"
  full_name        = "My organization"
  description      = "The pride of my work"
  is_enabled       = "true"
  delete_recursive = "true"
  delete_force     = "true"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) Org name
- `full_name` - (Required) Org full name
- `delete_recursive` - (Required) - pass `delete_recursive`=true as query parameter to remove an organization or VDC and any objects it contains that are in a state that normally allows removal.
- `delete_force` - (Required) - pass `delete_force=true` and `delete_recursive=true` to remove an organization or VDC and any objects it contains, regardless of their state.
- `is_enabled` - (Optional) - True if this organization is enabled (allows login and all other operations). Default is `true`.
- `description` - (Optional) - Org description. Default is empty.
- `deployed_vm_quota` - (Optional) - Maximum number of virtual machines that can be deployed simultaneously by a member of this organization. Default is unlimited (-1)
- `stored_vm_quota` - (Optional) - Maximum number of virtual machines in vApps or vApp templates that can be stored in an undeployed state by a member of this organization. Default is unlimited (-1)
- `can_publish_catalogs` - (Optional) - True if this organization is allowed to share catalogs. Default is `true`.
- `delay_after_power_on_seconds` - (Optional) - Specifies this organization's default for virtual machine boot delay after power on. Default is `0`.

» Sources

- OrgType
- ReferenceType
- Org deletion

» vcd_org_user

Provides a vCloud Director Org User. This can be used to create, update, and delete organization users, including org administrators.

Supported in provider *v2.4+*

Note: Only `System Administrator` or `Org Administrator` users can create users.

» Example Usage

```
# A simple user created with the minimum of properties
# uses the "password" field
resource "vcd_org_user" "my-org-admin" {
  org = "my-org"

  name          = "my-org-admin"
  description   = "a new org admin"
  role          = "Organization Administrator"
  password      = "change-me"
}

# Another user, created by filling all the fields
# Uses the "password_file" field.
resource "vcd_org_user" "test_user_vapp_author" {
  org = "datacloud"

  name              = "test_user_vapp_author"
  password_file     = "pwd201907101300.txt"
  full_name         = "test user vapp author"
  description       = "Org user test_user_vapp_author"
  role              = "vApp Author"
  enabled           = true
  take_ownership    = true
  provider_type     = "INTEGRATED"
  stored_vm_quota   = 20
  deployed_vm_quota = 20
  instant_messaging = "@test_user_vapp_author"
  email_address     = "test_user_vapp_author@test.company.org"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to which the VDC belongs. Optional if defined at provider level.
- `name` - (Required) A unique name for the user.
- `password` - (Optional, but required if `password_file` was not given) The user password. This value is never returned on read. It is inspected on create and modify. To modify, fill with a different value. Note that if you remove the password *on update*, Terraform will indicate that a change was occurring, but the empty password will be ignored by vCD.
- `password_file` (Optional, but required if `password` was not given). A text file containing the password. Recommended usage: after changing the password, run an apply again with the password blank. Using this property instead of `password` has the advantage that the sensitive data is not saved into Terraform state file. The disadvantage is that a password change requires also changing the file name.
- `provider_type` - (Optional) Identity provider type for this this user. One of: `INTEGRATED`, `SAML`, `OAUTH`. The default is `INTEGRATED`.
- `role` - (Required) The role of the user. Role names can be retrieved from the organization. Both built-in roles and custom built can be used. The roles normally available are:
  - `Organization Administrator`
  - `Catalog Author`
  - `vApp Author`
  - `vApp User`
  - `Console Access Only`
  - `Defer to Identity Provider`
- `full_name` - (Optional) The full name of the user.
- `description` - (Optional) An optional description of the user.
- `telephone` - (Optional) The Org User telephone number.
- `email_address` - (Optional) The Org User email address. Needs to be a properly formatted email address.
- `instant_messaging` - (Optional) The Org User instant messaging.
- `enabled` - (Optional) True if the user is enabled and can log in. The default is `true`.
- `is_group_role` - (Optional) True if this user has a group role.. The default is `false`.
- `is_locked` - (Optional)aIf the user account has been locked due to too many invalid login attempts, the value will change to true (only the system can lock the user). To unlock the user re-set this flag to false.
- `take_ownership` - (Optional) Take ownership of user's objects on deletion.
- `deployed_vm_quota` - (Optional) Quota of vApps that this user can deploy. A value of 0 specifies an unlimited quota. The default is 10.
- `stored_vm_quota` - (Optional) Quota of vApps that this user can store. A value of 0 specifies an unlimited quota. The default is 10.

» Attribute Reference

The following attributes are exported on this resource:

- `id` - The ID of the Organization user

» Importing

Note: The current implementation of Terraform import can only import resources into the state. It does not generate configuration. More information.

An existing user can be imported into this resource via supplying the full dot separated path for an org user. For example, using this structure, representing an existing user that was **not** created using Terraform:

```
resource "vcd_org_user" "my-org-admin" {
  org  = "my-org"
  name = "my-org-admin"
  role = "Organization Administrator"
}
```

You can import such user into terraform state using this command

```
terraform import vcd_org_user.my-org-admin my-org.my-org-admin
```

The state (in `terraform.tfstate`) would look like this:

```
{
  "version": 4,
  "terraform_version": "0.12.0",
  "serial": 1,
  "lineage": "f3fb8d07-8fe5-4fe3-3afe-c9050ffe68f6",
  "outputs": {},
  "resources": [
    {
      "mode": "managed",
      "type": "vcd_org_user",
      "name": "my-org-user",
      "provider": "provider.vcd",
      "instances": [
        {
          "schema_version": 0,
          "attributes": {
            "deployed_vm_quota": 50,
            "description": "This is my-org main user",
            "email_address": "my-org-admin@mycompany.com",
            "full_name": "My Org Admin",
            "id": "urn:vcloud:user:5fd69dfa-6bbe-40a6-9ee3-70448b6601ef",
```

```
            "instant_messaging": "@my_org_admin",
            "enabled": true,
            "is_group_role": false,
            "is_locked": false,
            "name": "my-org-user",
            "org": "my-org",
            "password": null,
            "password_file": null,
            "provider_type": "INTEGRATED",
            "role": "Organization Administrator",
            "stored_vm_quota": 50,
            "take_ownership": null,
            "telephone": "123-456-7890"
          }
        }
      ]
    }
  ]
}
```

After that, you can expand the configuration file and either update or delete the user as needed. Running `terraform plan` at this stage will show the difference between the minimal configuration file and the user's stored properties.

» vcd_catalog

Provides a vCloud Director catalog resource. This can be used to create and delete a catalog.

Supported in provider *v2.0+*

» Example Usage

```
resource "vcd_catalog" "myNewCatalog" {
  org = "my-org"

  name             = "my-catalog"
  description      = "catalog for files"
  delete_recursive = "true"
  delete_force     = "true"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `name` - (Required) Catalog name
- `description` - (Optional) - Description of catalog
- `delete_recursive` - (Required) - When destroying use delete_recursive=True to remove the catalog and any objects it contains that are in a state that normally allows removal
- `delete_force` -(Required) - When destroying use delete_force=True with delete_recursive=True to remove a catalog and any objects it contains, regardless of their state

» vcd_catalog_item

Provides a vCloud Director catalog item resource. This can be used to upload OVA to catalog and delete it.

Supported in provider *v2.0+*

» Example Usage

```
resource "vcd_catalog_item" "myNewCatalogItem" {
  org     = "my-org"
  catalog = "my-catalog"

  name                = "my ova"
  description         = "new vapp template"
  ova_path            = "/home/user/file.ova"
  upload_piece_size   = 10
  show_upload_progress = true
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `catalog` - (Required) The name of the catalog where to upload OVA file

- `name` - (Required) Item name in catalog
- `description` - (Optional) - Description of item
- `ova_path` - (Required) - Absolute or relative path to file to upload
- `upload_piece_size` - (Optional) - Size in MB for splitting upload size. It can possibly impact upload performance. Default 1MB.
- `show_upload_progress` - (Optional) - Default false. Allows to see upload progress

» vcd_catalog_media

Provides a vCloud Director media resource. This can be used to upload media to catalog and delete it.

Supported in provider *v2.0+*

» Example Usage

```
resource "vcd_catalog_media" "myNewMedia" {
  org     = "my-org"
  catalog = "my-catalog"

  name                = "my iso"
  description         = "new os versions"
  media_path          = "/home/user/file.iso"
  upload_piece_size   = 10
  show_upload_progress = true
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `catalog` - (Required) The name of the catalog where to upload media file
- `name` - (Required) Media file name in catalog
- `description` - (Optional) - Description of media file
- `media_path` - (Required) - Absolute or relative path to file to upload
- `upload_piece_size` - (Optional) - size in MB for splitting upload size. It can possibly impact upload performance. Default 1MB.
- `show_upload_progress` - (Optional) - Default false. Allows to see upload progress

» vcd_dnat

Provides a vCloud Director DNAT resource. This can be used to create, modify, and delete destination NATs to map an external IP/port to an internal IP/port.

Note: From v2.4+ `protocol` requires lower case values. This may result in invalid configuration if upper case was used previously.

Warning: When advanced edge gateway is used and the rule is updated using UI, then ID mapping will be lost and Terraform won't find the rule anymore and remove it from state.

» Example Usage

```
resource "vcd_dnat" "web" {
  org = "my-org" # Optional
  vdc = "my-vdc" # Optional

  edge_gateway    = "Edge Gateway Name"
  external_ip     = "78.101.10.20"
  port            = 80
  internal_ip     = "10.10.0.5"
  translated_port = 8080
}

resource "vcd_dnat" "forIcmp" {
  org = "my-org" # Optional
  vdc = "my-vdc" # Optional

  network_name = "my-external-network"
  network_type = "ext"

  edge_gateway  = "Edge Gateway Name"
  external_ip   = "78.101.10.20"
  port          = -1                    # "-1" == "any"
  internal_ip   = "10.10.0.5"
  protocol      = "icmp"
  icmp_sub_type = "router-solicitation"
}
```

» Argument Reference

The following arguments are supported:

- `edge_gateway` - (Required) The name of the edge gateway on which to apply the DNAT
- `external_ip` - (Required) One of the external IPs available on your Edge Gateway
- `port` - (Required) The port number to map. -1 translates to "any"
- `translated_port` - (Optional) The port number to map
- `internal_ip` - (Required) The IP of the VM to map to
- `protocol` - (Optional; *v2.0+*) The protocol type. Possible values are `tcp`, `udp`, `tcpupd`, `icmp`, `any`. `tcp` is default to be backward compatible with previous version
- `icmp_sub_type` - (Optional; *v2.0+*) The name of ICMP type. Possible values are address-mask-request, destination-unreachable, echo-request, echo-reply, parameter-problem, redirect, router-advertisement, router-solicitation, source-quench, time-exceeded, timestamp-request, timestamp-reply, any
- `network_type` - (Optional; *v2.4+*) Type of the network on which to apply the NAT rule. Possible values `org` or `ext`. `ext` requires system administrator privileges. ***`network_type`*** *will be a required field in the next major version.*
- `network_name` - (Optional; *v2.4+*) The name of the network on which to apply the SNAT. ***`network_name`*** *will be a required field in the next major version.*
- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level
- `description` - (Optional; *v2.4+*) - Description of item

» vcd_external_network

Provides a vCloud Director external network resource. This can be used to create and delete external networks. Requires system administrator privileges.

Supported in provider *v2.2+*

» Example Usage

```
provider "vcd" {
  user     = "${var.admin_user}"
  password = "${var.admin_password}"
  org      = "System"
  url      = "https://Vcd/api"
}
```

```
resource "vcd_external_network" "net" {
  name        = "my-ext-net"
  description = "Reference for vCD external network"

  ip_scope {
    gateway    = "192.168.30.49"
    netmask    = "255.255.255.240"
    dns1       = "192.168.0.164"
    dns2       = "192.168.0.196"
    dns_suffix = "mybiz.biz"

    static_ip_pool {
      start_address = "192.168.30.51"
      end_address   = "192.168.30.62"
    }
  }

  # It's possible to define more than one IP scope
  ip_scope {
    gateway    = "192.168.31.49"
    netmask    = "255.255.255.240"
    dns1       = "192.168.1.164"
    dns2       = "192.168.1.196"
    dns_suffix = "my.biz"

    static_ip_pool {
      start_address = "192.168.31.51"
      end_address   = "192.168.31.60"
    }

    static_ip_pool {
      start_address = "192.168.31.31"
      end_address   = "192.168.31.40"
    }
  }

  vsphere_network {
    name    = "myNetwork"
    type    = "DV_PORTGROUP"
    vcenter = "vcenter-name"
  }

  # It's possible to define more than one vSphere network
  vsphere_network {
    name    = "myNetwork2"
```

```
    type    = "DV_PORTGROUP"
    vcenter = "vcenter-name2"
  }

  retain_net_info_across_deployments = "false"
}

resource "vcd_network_direct" "net" {
  org              = "my-org"
  vdc              = "my-vdc"
  name             = "my-net"
  external_network = "${vcd_external_network.net.name}"
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the network
- `description` - (Optional) Network friendly description
- `ip_scope` - (Required) A list of IP scopes for the network. See IP Scope below for details.
- `vsphere_network` - (Required) A list of DV_PORTGROUP or NETWORK objects names that back this network. Each referenced DV_PORTGROUP or NETWORK must exist on a vCenter server registered with the system. See vSphere Network below for details.
- `retain_net_info_across_deployments` - (Optional) Specifies whether the network resources such as IP/MAC of router will be retained across deployments. Default is false.

» IP Scope

- `gateway` - (Required) Gateway of the network
- `netmask` - (Required) Network mask
- `dns1` - (Optional) Primary DNS server
- `dns2` - (Optional) Secondary DNS server
- `dns_suffix` (Optional) A FQDN for the virtual machines on this network.
- `static_ip_pool` - (Required) IP ranges used for static pool allocation in the network. See IP Pool below for details.

» IP Pool

- `start_address` - (Required) Start address of the IP range

- `end_address` - (Required) End address of the IP range

» vSphere Network

- `name` - (Required) Port group name
- `type` - (Required) The vSphere type of the object. One of: DV_PORTGROUP (distributed virtual port group), NETWORK (standard switch port group)
- `vcenter` - (Required) The vCenter server name

» vcd_firewall_rules

Provides a vCloud Director Firewall resource. This can be used to create, modify, and delete firewall settings and rules.

» Example Usage

```
resource "vcd_firewall_rules" "fw" {
  edge_gateway   = "Edge Gateway Name"
  default_action = "drop"

  rule {
    description      = "drop-ftp-out"
    policy           = "drop"
    protocol         = "tcp"
    destination_port = "21"
    destination_ip   = "any"
    source_port      = "any"
    source_ip        = "10.10.0.0/24"
  }

  rule {
    description      = "allow-outbound"
    policy           = "allow"
    protocol         = "any"
    destination_port = "any"
    destination_ip   = "any"
    source_port      = "any"
    source_ip        = "10.10.0.0/24"
  }
}
```

```
resource "vcd_vapp" "web" {
  # ...
}

resource "vcd_firewall_rules" "fw-web" {
  edge_gateway   = "Edge Gateway Name"
  default_action = "drop"

  rule {
    description      = "allow-web"
    policy           = "allow"
    protocol         = "tcp"
    destination_port = "80"
    destination_ip   = "${vcd_vapp.web.ip}"
    source_port      = "any"
    source_ip        = "any"
  }
}
```

» Argument Reference

The following arguments are supported:

- `edge_gateway` - (Required) The name of the edge gateway on which to apply the Firewall Rules
- `default_action` - (Required) Either ”allow” or ”drop”. Specifies what to do should none of the rules match
- `rule` - (Optional) Configures a firewall rule; see Rules below for details.
- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level

» Rules

Each firewall rule supports the following attributes:

- `description` - (Required) Description of the fireall rule
- `policy` - (Required) Specifies what to do when this rule is matched. Either ”allow” or ”drop”
- `protocol` - (Required) The protocol to match. One of ”tcp”, ”udp”, ”icmp” or ”any”
- `destination_port` - (Required) The destination port to match. Either a port number or ”any”

- `destination_ip` - (Required) The destination IP to match. Either an IP address, IP range or "any"
- `source_port` - (Required) The source port to match. Either a port number or "any"
- `source_ip` - (Required) The source IP to match. Either an IP address, IP range or "any"

» vcd_independent_disk

Provides a vCloud Director independent disk resource. This can be used to create and delete independent disks.

Supported in provider *v2.1+*

» Example Usage

```
resource "vcd_independent_disk" "myNewIndependentDisk" {
  org             = "my-org"
  vdc             = "my-vcd"

  name            = "logDisk"
  size            = "33000"
  bus_type        = "SCSI"
  bus_sub_type    = "VirtualSCSI"
  storage_profile = "external"
}

resource "vcd_vapp_vm" "web2" {
  vapp_name     = "${vcd_vapp.web.name}"

...

  disk {
    name = "${vcd_independent_disk.myNewIndependentDisk.name}"
    bus_number = 1
    unit_number = 0
  }

  depends_on = ["vcd_independent_disk.myNewIndependentDisk"]
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `name` - (Required) Disk name
- `size` - (Required) - Size of disk in MB
- `bus_type` - (Optional) - Disk bus type. Values can be: IDE, SCSI, SATA
- `bus_sub_type` - (Optional) - Disk bus subtype. Values can be: "IDE" for IDE. buslogic, lsilogic, lsilogicsas, VirtualSCSI for SCSI and ahci for SATA
- `storage_profile` - (Optional) - The name of storage profile where disk will be created

» vcd_inserted_media

Provides a vCloud Director resource for inserting or ejecting media (ISO) file for the VM. Create this resource for inserting the media, and destroy it for ejecting.

Supported in provider *v2.0+*

» Example Usage

```
resource "vcd_inserted_media" "myInsertedMedia" {
  org     = "my-org"
  vdc     = "my-vcd"
  catalog = "my-catalog"
  name    = "my-iso"

  vapp_name = "my-vApp"
  vm_name   = "my-VM"

  eject_force = true
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level
- `catalog` - (Required) The name of the catalog where to find media file
- `name` - (Required) Media file name in catalog which will be inserted to VM
- `vapp_name` - (Required) - The name of vApp to find
- `vm_name` - (Required) - The name of VM to be used to insert media file
- `eject_force` - (Optional; *v2.1+*) Allows to pass answer to question in vCD "The guest operating system has locked the CD-ROM door and is probably using the CD-ROM. Disconnect anyway (and override the lock)?" when ejecting from a VM which is powered on. True means "Yes" as answer to question. Default is `true`

» vcd_network (Deprecated)

Provides a vCloud Director Org VDC Network. This can be used to create, modify, and delete internal networks for vApps to connect.

Deprecated in v2.0+ : this resource is deprecated and replaced by vcd-network-routed. It is also complemented by vcd-network-isolated and vcd-network-direct.

» Example Usage

```
resource "vcd_network" "net" {
  name         = "my-net"
  edge_gateway = "Edge Gateway Name"
  gateway      = "10.10.0.1"

  dhcp_pool {
    start_address = "10.10.0.2"
    end_address   = "10.10.0.100"
  }

  static_ip_pool {
    start_address = "10.10.0.152"
    end_address   = "10.10.0.254"
  }
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the network
- `edge_gateway` - (Required) The name of the edge gateway
- `netmask` - (Optional) The netmask for the new network. Defaults to `255.255.255.0`
- `gateway` (Required) The gateway for this network
- `dns1` - (Optional) First DNS server to use. Defaults to `8.8.8.8`
- `dns2` - (Optional) Second DNS server to use. Defaults to `8.8.4.4`
- `dns_suffix` - (Optional) A FQDN for the virtual machines on this network
- `shared` - (Optional) Defines if this network is shared between multiple vDCs in the vOrg. Defaults to `false`.
- `dhcp_pool` - (Optional) A range of IPs to issue to virtual machines that don't have a static IP; see IP Pools below for details.
- `static_ip_pool` - (Optional) A range of IPs permitted to be used as static IPs for virtual machines; see IP Pools below for details.

» IP Pools

Static IP Pools and DHCP Pools support the following attributes:

- `start_address` - (Required) The first address in the IP Range
- `end_address` - (Required) The final address in the IP Range

DHCP Pools additionally support the following attributes:

- `default_lease_time` - (Optional) The default DHCP lease time to use. Defaults to `3600`.
- `max_lease_time` - (Optional) The maximum DHCP lease time to use. Defaults to `7200`.

» vcd_network_routed

Provides a vCloud Director Org VDC routed Network. This can be used to create, modify, and delete internal networks for vApps to connect.

Supported in provider *v2.0+*

» Example Usage

```
resource "vcd_network_routed" "net" {
  org = "my-org" # Optional
  vdc = "my-vdc" # Optional
```

```
  name         = "my-net"
  edge_gateway = "Edge Gateway Name"
  gateway      = "10.10.0.1"

  dhcp_pool {
    start_address = "10.10.0.2"
    end_address   = "10.10.0.100"
  }

  static_ip_pool {
    start_address = "10.10.0.152"
    end_address   = "10.10.0.254"
  }
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level
- `name` - (Required) A unique name for the network
- `edge_gateway` - (Required) The name of the edge gateway
- `netmask` - (Optional) The netmask for the new network. Defaults to `255.255.255.0`
- `gateway` (Required) The gateway for this network
- `dns1` - (Optional) First DNS server to use. Defaults to `8.8.8.8`
- `dns2` - (Optional) Second DNS server to use. Defaults to `8.8.4.4`
- `dns_suffix` - (Optional) A FQDN for the virtual machines on this network
- `shared` - (Optional) Defines if this network is shared between multiple vDCs in the vOrg. Defaults to `false`.
- `dhcp_pool` - (Optional) A range of IPs to issue to virtual machines that don't have a static IP; see IP Pools below for details.
- `static_ip_pool` - (Optional) A range of IPs permitted to be used as static IPs for virtual machines; see IP Pools below for details.

» IP Pools

Static IP Pools and DHCP Pools support the following attributes:

- `start_address` - (Required) The first address in the IP Range

- `end_address` - (Required) The final address in the IP Range

DHCP Pools additionally support the following attributes:

- `default_lease_time` - (Optional) The default DHCP lease time to use. Defaults to `3600`.
- `max_lease_time` - (Optional) The maximum DHCP lease time to use. Defaults to `7200`.

» vcd_network_direct

Provides a vCloud Director Org VDC Network directly connected to an external network. This can be used to create, modify, and delete internal networks for vApps to connect.

Supported in provider *v2.0+*

Note: Only `System Administrator` can create an organization virtual datacenter network that connects directly to an external network. You must use `System Adminstrator` account in `provider` configuration and then provide `org` and `vdc` arguments for direct networks to work.

» Example Usage

```
resource "vcd_network_direct" "net" {
  org = "my-org" # Optional
  vdc = "my-vdc" # Optional

  name             = "my-net"
  external_network = "my-ext-net"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level
- `name` - (Required) A unique name for the network
- `external_network` - (Required) The name of the external network.
- `shared` - (Optional) Defines if this network is shared between multiple vDCs in the vOrg. Defaults to `false`.

» vcd_network_isolated

Provides a vCloud Director Org VDC isolated Network. This can be used to create, modify, and delete internal networks for vApps to connect. This network is not attached to external networks or routers.

Supported in provider *v2.0+*

» Example Usage

```
resource "vcd_network_isolated" "net" {
  org = "my-org" # Optional
  vdc = "my-vdc" # Optional

  name    = "my-net"
  gateway = "192.168.2.1"
  dns1    = "192.168.2.1"

  dhcp_pool {
    start_address = "192.168.2.2"
    end_address   = "192.168.2.50"
  }

  static_ip_pool {
    start_address = "192.168.2.51"
    end_address   = "192.168.2.100"
  }
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level
- `name` - (Required) A unique name for the network
- `netmask` - (Optional) The netmask for the new network. Defaults to `255.255.255.0`
- `gateway` (Required) The gateway for this network
- `dns1` - (Optional) First DNS server to use. Defaults to `8.8.8.8`
- `dns2` - (Optional) Second DNS server to use. Defaults to `8.8.4.4`

- `dns_suffix` - (Optional) A FQDN for the virtual machines on this network
- `shared` - (Optional) Defines if this network is shared between multiple vDCs in the vOrg. Defaults to `false`.
- `dhcp_pool` - (Optional) A range of IPs to issue to virtual machines that don't have a static IP; see IP Pools below for details.
- `static_ip_pool` - (Optional) A range of IPs permitted to be used as static IPs for virtual machines; see IP Pools below for details.

» IP Pools

Static IP Pools and DHCP Pools support the following attributes:

- `start_address` - (Required) The first address in the IP Range
- `end_address` - (Required) The final address in the IP Range

DHCP Pools additionally support the following attributes:

- `default_lease_time` - (Optional) The default DHCP lease time to use. Defaults to `3600`.
- `max_lease_time` - (Optional) The maximum DHCP lease time to use. Defaults to `7200`.

» vcd_snat

Provides a vCloud Director SNAT resource. This can be used to create, modify, and delete source NATs to allow vApps to send external traffic.

Warning: When advanced edge gateway is used and the rule is updated using UI, then ID mapping will be lost and Terraform won't find the rule anymore and remove it from state.

» Example Usage

```
resource "vcd_snat" "outbound" {
  edge_gateway = "Edge Gateway Name"
  network_name = "my-org-vdc-network"
  network_type = "org"
  external_ip  = "78.101.10.20"
  internal_ip  = "10.10.0.0/24"
}
```

» Argument Reference

The following arguments are supported:

- `edge_gateway` - (Required) The name of the edge gateway on which to apply the SNAT
- `external_ip` - (Required) One of the external IPs available on your Edge Gateway
- `internal_ip` - (Required) The IP or IP Range of the VM(s) to map from
- `network_type` - (Optional; *v2.4+*) Type of the network on which to apply the NAT rule. Possible values `org` or `ext`. ***`network_type`*** *will be a required field in the next major version.*
- `network_name` - (Optional; *v2.4+*) The name of the network on which to apply the SNAT. ***`network_name`*** *will be a required field in the next major version.*
- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level
- `description` - (Optional; *v2.4+*) - Description of item

» vcd_edgegateway

Provides a vCloud Director edge gateway directly connected to one or more external networks. This can be used to create and delete edge gateways for Org VDC networks to connect.

Supported in provider *v2.4+*

Note: Only `System Administrator` can create an edge gateway. You must use `System Adminstrator` account in `provider` configuration and then provide `org` and `vdc` arguments for edge gateway to work.

Note: Load balancing capabilities will work only when edge gateway is `advanced`. Load balancing settings will be **ignored** when it is not. Refer to official vCloud Director documentation for more information.

» Example Usage

```
resource "vcd_edgegateway" "egw" {
  org = "my-org"
  vdc = "my-vdc"

  name                    = "my-egw"
  description             = "new edge gateway"
  configuration           = "compact"
  default_gateway_network = "my-ext-net1"
  external_networks       = [ "my-ext-net1", "my-ext-net2" ]
```

```
  advanced                = true
}

resource "vcd_network_routed" "rnet1" {
  name         = "rnet1"
  org          = "my-org"
  vdc          = "my-vdc"
  edge_gateway = "${vcd_edgegateway.egw.name}"
  gateway      = "192.168.2.1"

  static_ip_pool {
    start_address = "192.168.2.2"
    end_address   = "192.168.2.100"
  }
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to which the VDC belongs. Optional if defined at provider level.
- `vdc` - (Optional) The name of VDC that owns the edge gateway. Optional if defined at provider level.
- `name` - (Required) A unique name for the edge gateway.
- `external_networks` - (Required) An array of external network names.
- `configuration` - (Required) Configuration of the vShield edge VM for this gateway. One of: `compact`, `full` ("Large"), `x-large`, `full4` ("Quad Large").
- `default_gateway_network` - (Optional) Name of the external network to be used as default gateway. It must be included in the list of `external_networks`. Providing an empty string or omitting the argument will create the edge gateway without a default gateway.
- `advanced` - (Optional) True if the gateway uses advanced networking. Default is `true`.
- `ha_enabled` - (Optional) Enable high availability on this edge gateway. Default is `false`.
- `distributed_routing` - (Optional) If advanced networking enabled, also enable distributed routing. Default is `false`.
- `lb_enabled` - (Optional) Enable load balancing. Default is `false`.
- `lb_acceleration_enabled` - (Optional) Enable to configure the load balancer to use the faster L4 engine rather than L7 engine. The L4 TCP VIP is processed before the edge gateway firewall so no `allow` firewall rule is required. Default is `false`. **Note:** L7 VIPs for HTTP and HTTPS are processed after the firewall, so when Acceleration Enabled is not selected,

an edge gateway firewall rule must exist to allow access to the L7 VIP for those protocols. When Acceleration Enabled is selected and the server pool is in non-transparent mode, an SNAT rule is added, so you must ensure that the firewall is enabled on the edge gateway.

- `lb_logging_enabled` - (Optional) Enables the edge gateway load balancer to collect traffic logs. Default is `false`.
- `lb_loglevel` - (Optional) Choose the severity of events to be logged. One of `emergency`, `alert`, `critical`, `error`, `warning`, `notice`, `info`, `debug`

» vcd_edgegateway_vpn

Provides a vCloud Director IPsec VPN. This can be used to create, modify, and delete VPN settings and rules.

» Example Usage

```
resource "vcd_edgegateway_vpn" "vpn" {
  edge_gateway        = "Internet_01(nti0000bi2_123-456-2)"
  name                = "west-to-east"
  description         = "Description"
  encryption_protocol = "AES256"
  mtu                 = 1400
  peer_id             = "64.121.123.11"
  peer_ip_address     = "64.121.123.11"
  local_id            = "64.121.123.10"
  local_ip_address    = "64.121.123.10"
  shared_secret       = "**********************"

  peer_subnets {
    peer_subnet_name    = "DMZ_WEST"
    peer_subnet_gateway = "10.0.10.1"
    peer_subnet_mask    = "255.255.255.0"
  }

  peer_subnets {
    peer_subnet_name    = "WEB_WEST"
    peer_subnet_gateway = "10.0.20.1"
    peer_subnet_mask    = "255.255.255.0"
  }

  local_subnets {
    local_subnet_name    = "DMZ_EAST"
    local_subnet_gateway = "10.0.1.1"
```

```
    local_subnet_mask    = "255.255.255.0"
  }

  local_subnets {
    local_subnet_name    = "WEB_EAST"
    local_subnet_gateway = "10.0.22.1"
    local_subnet_mask    = "255.255.255.0"
  }
}
```

» Argument Reference

The following arguments are supported:

- `edge_gateway` - (Required) The name of the edge gateway on which to apply the Firewall Rules
- `name` - (Required) The name of the VPN
- `description` - (Required) A description for the VPN
- `encryption_protocol` - (Required) - E.g. `AES256`
- `local_ip_address` - (Required) - Local IP Address
- `local_id` - (Required) - Local ID
- `mtu` - (Required) - The MTU setting
- `peer_ip_address` - (Required) - Peer IP Address
- `peer_id` - (Required) - Peer ID
- `shared_secret` - (Required) - Shared Secret
- `local_subnets` - (Required) - List of Local Subnets see Local Subnets below for details.
- `peer_subnets` - (Required) - List of Peer Subnets see Peer Subnets below for details.
- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level

» Local Subnets

Each Local Subnet supports the following attributes:

- `local_subnet_name` - (Required) Name of the local subnet
- `local_subnet_gateway` - (Required) Gateway of the local subnet
- `local_subnet_mask` - (Required) Subnet mask of the local subnet

» Peer Subnets

Each Peer Subnet supports the following attributes:

- `peer_subnet_name` - (Required) Name of the peer subnet
- `peer_subnet_gateway` - (Required) Gateway of the peer subnet
- `peer_subnet_mask` - (Required) Subnet mask of the peer subnet

» vcd_vapp

Provides a vCloud Director vApp resource. This can be used to create, modify, and delete vApps.

» Example Usage

Example with more than one VM under a vApp.

```
resource "vcd_network_direct" "net" {
  name             = "net"
  external_network = "corp-network"
}

resource "vcd_vapp" "web" {
  name = "web"

  metadata = {
    CostAccount = "Marketing Department"
  }

  depends_on = ["vcd_network_direct.net"]
}

resource "vcd_vapp_vm" "web1" {
  vapp_name     = "${vcd_vapp.web.name}"
  name          = "web1"
  catalog_name  = "Boxes"
  template_name = "lampstack-1.10.1-ubuntu-10.04"
  memory        = 2048
  cpus          = 1

  network_name = "net"
  ip           = "10.10.104.161"

  depends_on = ["vcd_vapp.web"]
```

```
}

resource "vcd_vapp_vm" "web2" {
  vapp_name     = "${vcd_vapp.web.name}"
  name          = "web2"
  catalog_name  = "Boxes"
  template_name = "lampstack-1.10.1-ubuntu-10.04"
  memory        = 2048
  cpus          = 1

  network_name = "net"
  ip           = "10.10.104.162"

  depends_on = ["vcd_vapp.web"]
}
```

» Example of vApp with single VM

Not recommended in v2.0+ : in the earlier version of the provider it was possible to define a vApp with a single VM in one resource, but it is not recommended as of *v2.0+* provider. Please define vApp and VM in separate resources instead.

```
resource "vcd_network_routed" "net" {
  # ...
}

resource "vcd_vapp" "web" {
  name          = "web"
  catalog_name  = "Boxes"
  template_name = "lampstack-1.10.1-ubuntu-10.04"
  memory        = 2048
  cpus          = 1

  network_name = "${vcd_network.net.name}"
  ip           = "10.10.104.160"

  metadata = {
    role    = "web"
    env     = "staging"
    version = "v1"
  }

  ovf {
    hostname = "web"
```

```
  }

  depends_on = ["vcd_network_routed.net"]
}
```

» Example of Empty vApp with no VMs

```
resource "vcd_network_routed" "net" {
  # ...
}

resource "vcd_vapp" "web" {
  name = "web"

  metadata = {
    boss = "Why is this vApp empty?"
    john = "I don't really know. Maybe somebody did forget to clean it up."
  }

  depends_on = ["vcd_network_routed.net"]
}
```

» Argument Reference

The following arguments are supported:

- `name` - (Required) A unique name for the vApp
- `catalog_name` - (Optional) The catalog name in which to find the given vApp Template
- `template_name` - (Optional) The name of the vApp Template to use
- `memory` - (Optional) The amount of RAM (in MB) to allocate to the vApp
- `cpus` - (Optional) The number of virtual CPUs to allocate to the vApp
- `initscript` (Optional) A script to be run only on initial boot
- `network_name` - (Optional) Name of the network this vApp should join
- `network_href` - (Optional; **Deprecated**) The vCloud Director generated href of the network this vApp should join. If empty it will use the network name and query vCloud Director to discover this
- `ip` - (Optional) The IP to assign to this vApp. Must be an IP address or one of dhcp, allocated or none. If given the address must be within the `static_ip_pool` set for the network. If left blank, and the network has `dhcp_pool` set with at least one available IP then this will be set with DHCP.
- `metadata` - (Optional) Key value map of metadata to assign to this vApp. Key and value can be any string. (Since *v2.2+* metadata is added directly

to vApp instead of first VM in vApp)
- `ovf` - (Optional) Key value map of ovf parameters to assign to VM product section
- `power_on` - (Optional) A boolean value stating if this vApp should be powered on. Default is `true`
- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level
- `accept_all_eulas` - (Optional; *v2.0+*) Automatically accept EULA if OVA has it. Default is `true`

» vcd_vapp_network

Provides a vCloud Director vApp isolated Network. This can be used to create and delete internal networks for vApps to connect. This network is not attached to external networks or routers.

Supported in provider *v2.1+*

» Example Usage

```
resource "vcd_vapp_network" "vappNet" {
  org = "my-org" #Optional
  vdc = "my-vdc" #Optional

  name               = "my-net"
  vapp_name          = "my-vapp"
  gateway            = "192.168.2.1"
  netmask            = "255.255.255.0"
  dns1               = "192.168.2.1"
  dns2               = "192.168.2.2"
  dns_suffix         = "mybiz.biz"
  guest_vlan_allowed = true

  static_ip_pool {
    start_address = "192.168.2.51"
    end_address   = "192.168.2.100"
  }

  dhcp_pool {
    start_address = "192.168.2.2"
    end_address   = "192.168.2.50"
```

```
  }
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations.
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level.
- `name` - (Required) A unique name for the network.
- `vapp_name` - (Required) The vApp this VM should belong to.
- `netmask` - (Optional) The netmask for the new network. Default is `255.255.255.0`.
- `gateway` (Optional) The gateway for this network.
- `dns1` - (Optional) First DNS server to use. Default is `8.8.8.8`.
- `dns2` - (Optional) Second DNS server to use. Default is `8.8.4.4`.
- `dns_suffix` - (Optional) A FQDN for the virtual machines on this network.
- `guest_vlan_allowed` (Optional) True if Network allows guest VLAN tagging. This value supported from vCD version 9.0
- `static_ip_pool` - (Optional) A range of IPs permitted to be used as static IPs for virtual machines; see IP Pools below for details.
- `dhcp_pool` - (Optional) A range of IPs to issue to virtual machines that don't have a static IP; see IP Pools below for details.

» IP Pools

Static IP Pools and DHCP Pools support the following attributes:

- `start_address` - (Required) The first address in the IP Range.
- `end_address` - (Required) The final address in the IP Range.

DHCP Pools additionally support the following attributes:

- `default_lease_time` - (Optional) The default DHCP lease time to use. Defaults to `3600`.
- `max_lease_time` - (Optional) The maximum DHCP lease time to use. Defaults to `7200`.
- `enabled` - (Optional) Allows to enable or disable service. Default is true.

» vcd_vapp_vm

Provides a vCloud Director VM resource. This can be used to create, modify, and delete VMs within a vApp.

Note: To make sure resources are created in the right order and both plan apply and destroy succeeds, use the `depends_on` clause (see example below)

» Example Usage

```
resource "vcd_network_direct" "net" {
  name             = "net"
  external_network = "corp-network"
}

resource "vcd_vapp" "web" {
  name = "web"

  depends_on = ["vcd_network_direct.net"]
}

resource "vcd_vapp_vm" "web1" {
  vapp_name     = "${vcd_vapp.web.name}"
  name          = "web1"
  catalog_name  = "Boxes"
  template_name = "lampstack-1.10.1-ubuntu-10.04"
  memory        = 2048
  cpus          = 2
  cpu_cores     = 1

  metadata = {
    role    = "web"
    env     = "staging"
    version = "v1"
    my_key  = "my value"
  }

  network {
    type               = "org"
    name               = "net"
    ip                 = "10.10.104.161"
    ip_allocation_mode = "MANUAL"
    is_primary         = true
  }
```

```
  depends_on = ["vcd_vapp.web"]
}

resource "vcd_vapp_vm" "web2" {
  vapp_name     = "${vcd_vapp.web.name}"
  name          = "web2"
  catalog_name  = "Boxes"
  template_name = "lampstack-1.10.1-ubuntu-10.04"
  memory        = 2048
  cpus          = 1

  metadata = {
    role    = "web"
    env     = "staging"
    version = "v1"
    my_key  = "my value"
  }

  network {
    type               = "org"
    name               = "net"
    ip                 = "10.10.104.162"
    ip_allocation_mode = "MANUAL"
    is_primary         = true
  }

  network {
    type               = "vapp"
    name               = "vapp-network"
    ip_allocation_mode = "POOL"
  }

  network {
    type               = "none"
    ip_allocation_mode = "NONE"
  }

  disk {
    name        = "logDisk1"
    bus_number  = 1
    unit_number = 0
  }

  disk {
    name        = "logDisk2"
    bus_number  = 1
```

```
    unit_number = 1
  }

  depends_on = ["vcd_vapp.web"]
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional; *v2.0+*) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional; *v2.0+*) The name of VDC to use, optional if defined at provider level
- `vapp_name` - (Required) The vApp this VM should belong to.
- `name` - (Required) A unique name for the VM
- `catalog_name` - (Required) The catalog name in which to find the given vApp Template
- `template_name` - (Required) The name of the vApp Template to use
- `memory` - (Optional) The amount of RAM (in MB) to allocate to the VM
- `cpus` - (Optional) The number of virtual CPUs to allocate to the VM. Socket count is a result of: virtual logical processors/cores per socket
- `cpu_cores` - (Optional; *v2.1+*) The number of cores per socket
- `metadata` - (Optional; *v2.2+*) Key value map of metadata to assign to this VM
- `initscript` (Optional) A script to be run only on initial boot
- `network_name` - (Optional; **Deprecated** by `network`) Name of the network this VM should connect to.
- `vapp_network_name` - (Optional; v2.1+; **Deprecated** by `network`) Name of the vApp network this VM should connect to.
- `ip` - (Optional; **Deprecated** by `network`) The IP to assign to this vApp. Must be an IP address or one of `dhcp`, `allocated`, or `none`. If given the address must be within the `static_ip_pool` set for the network. If left blank, and the network has `dhcp_pool` set with at least one available IP then this will be set with DHCP.
- `power_on` - (Optional) A boolean value stating if this vApp should be powered on. Default is `true`
- `accept_all_eulas` - (Optional; *v2.0+*) Automatically accept EULA if OVA has it. Default is `true`
- `disk` - (Optional; *v2.1+*) Independent disk attachment configuration. See Disk below for details.
- `expose_hardware_virtualization` - (Optional; *v2.2+*) Boolean for exposing full CPU virtualization to the guest operating system so that applications that require hardware virtualization can run on virtual machines

without binary translation or paravirtualization. Useful for hypervisor nesting provided underlying hardware supports it. Default is `false`.

- `network` - (Optional; *v2.2+*) A block to define network interface. Multiple can be used. See Network and example for usage details. **Deprecates**: `network_name`, `ip`, `vapp_network_name`. **Note**: this property and all its parameters do force recreation of VMs!

» Disk

- `name` - (Required) Independent disk name
- `bus_number` - (Required) Bus number on which to place the disk controller
- `unit_number` - (Required) Unit number (slot) on the bus specified by BusNumber.

» Network

- `type` (Required) Network type, one of: `none`, `vapp` or `org`. `none` creates a NIC with no network attached, `vapp` attaches a vApp network, while `org` attaches organization VDC network.
- `name` (Optional) Name of the network this VM should connect to. Always required except for `type NONE`.
- `is_primary` (Optional) Set to true if network interface should be primary. First network card in the list will be primary by default.
- `mac` - (Computed) Mac address of network interface.
- `ip_allocation_mode` (Required) IP address allocation mode. One of `POOL`, `DHCP`, `MANUAL`, `NONE`:
  - `POOL` - Static IP address is allocated automatically from defined static pool in network.
  - `DHCP` - IP address is obtained from a DHCP service. Field `ip` is not guaranteed to be populated. Because of this it may appear after multiple `terraform refresh` operations.
  - `MANUAL` - IP address is assigned manually in the `ip` field. Must be valid IP address from static pool.
  - `NONE` - No IP address will be set because VM will have a NIC without network.
- `ip` (Optional, Computed) Settings depend on `ip_allocation_mode`. Field requirements for each `ip_allocation_mode` are listed below:
  - `ip_allocation_mode=POOL` - `ip` value must be omitted or empty string "". Empty string may be useful when doing HCL variable

interpolation. Field `ip` will be populated with an assigned IP from static pool after run.
  - `ip_allocation_mode=DHCP` - `ip` value must be omitted or empty string "". Field `ip` is not guaranteed to be populated after run due to the VM lacking VMware tools or not working properly with DHCP. Because of this `ip` may also appear after multiple `terraform refresh` operations when is reported back to vCD.
  - `ip_allocation_mode=MANUAL` - `ip` value must be valid IP address from a subnet defined in `static pool` for network.
  - `ip_allocation_mode=NONE` - `ip` field can be omitted or set to an empty string "". Empty string may be useful when doing HCL variable interpolation.

» vcd_org_vdc

Provides a vCloud Director Organization VDC resource. This can be used to create and delete a Organization VDC. Requires system administrator privileges.

Supported in provider *v2.2+*

» Example Usage

```
provider "vcd" {
  user     = "${var.admin_user}"
  password = "${var.admin_password}"
  org      = "System"
  url      = "https://AcmeVcd/api"
}

resource "vcd_org_vdc" "my-vdc" {
  name        = "my-vdc"
  description = "The pride of my work"
  org         = "my-org"

  allocation_model = "ReservationPool"
  network_pool_name = "vDC1-VXLAN-NP"
  provider_vdc_name = "vDC1"

  compute_capacity {
    cpu {
      allocated = 2048
    }
```

```
    memory {
      allocated = 2048
    }
  }

  storage_profile {
    name     = "storage-name"
    limit    = 10240
    default  = true
  }

  metadata = {
    role    = "customerName"
    env     = "staging"
    version = "v1"
  }

  enabled                  = true
  enable_thin_provisioning = true
  enable_fast_provisioning = true
  delete_force             = true
  delete_recursive         = true
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) Organization to create the VDC in, optional if defined at provider level
- `name` - (Required) VDC name
- `description` - (Optional) VDC friendly description
- `provider_vdc_name` - (Required) A name of the Provider VDC from which this organization VDC is provisioned.
- `allocation_model` - (Required) The allocation model used by this VDC; must be one of {AllocationVApp ("Pay as you go"), AllocationPool ("Allocation pool"), ReservationPool ("Reservation pool")}
- `compute_capacity` - (Required) The compute capacity allocated to this VDC. See Compute Capacity below for details.
- `nic_quota` - (Optional) Maximum number of virtual NICs allowed in this VDC. Defaults to 0, which specifies an unlimited number.
- `network_quota` - (Optional) Maximum number of network objects that can be deployed in this VDC. Defaults to 0, which means no networks can be deployed.
- `vm_quota` - (Optional) The maximum number of VMs that can be created

in this VDC. Includes deployed and undeployed VMs in vApps and vApp templates. Defaults to 0, which specifies an unlimited number.

- `enabled` - (Optional) True if this VDC is enabled for use by the organization VDCs. Default is true.
- `storage_profile` - (Required) Storage profiles supported by this VDC. See Storage Profile below for details.
- `memory_guaranteed` - (Optional) Percentage of allocated memory resources guaranteed to vApps deployed in this VDC. For example, if this value is 0.75, then 75% of allocated resources are guaranteed. Required when AllocationModel is AllocationVApp or AllocationPool. When Allocation model is AllocationPool minimum value is 0.2. If left empty, vCD sets a value.
- `cpu_guaranteed` - (Optional) Percentage of allocated CPU resources guaranteed to vApps deployed in this VDC. For example, if this value is 0.75, then 75% of allocated resources are guaranteed. Required when AllocationModel is AllocationVApp or AllocationPool. If left empty, vCD sets a value.
- `cpu_speed` - (Optional) Specifies the clock frequency, in Megahertz, for any virtual CPU that is allocated to a VM. A VM with 2 vCPUs will consume twice as much of this value. Ignored for ReservationPool. Required when AllocationModel is AllocationVApp or AllocationPool, and may not be less than 256 MHz. Defaults to 1000 MHz if value isn't provided.
- `metadata` - (Optional; *v2.4+*) Key value map of metadata to assign to this VDC
- `enable_thin_provisioning` - (Optional) Boolean to request thin provisioning. Request will be honored only if the underlying data store supports it. Thin provisioning saves storage space by committing it on demand. This allows over-allocation of storage.
- `enable_fast_provisioning` - (Optional) Request fast provisioning. Request will be honored only if the underlying datastore supports it. Fast provisioning can reduce the time it takes to create virtual machines by using vSphere linked clones. If you disable fast provisioning, all provisioning operations will result in full clones.
- `network_pool_name` - (Optional) Reference to a network pool in the Provider VDC. Required if this VDC will contain routed or isolated networks.
- `allow_over_commit` - (Optional) Set to false to disallow creation of the VDC if the AllocationModel is AllocationPool or ReservationPool and the ComputeCapacity you specified is greater than what the backing Provider VDC can supply. Default is true.
- `enable_vm_discovery` - (Optional) If true, discovery of vCenter VMs is enabled for resource pools backing this VDC. If false, discovery is disabled. If left unspecified, the actual behaviour depends on enablement at the organization level and at the system level.
- `delete_force` - (Required) When destroying use `delete_force=True` to remove a VDC and any objects it contains, regardless of their state.

- `delete_recursive` - (Required) When destroying use `delete_recursive=True` to remove the VDC and any objects it contains that are in a state that normally allows removal.

» Storage Profile

- `name` - (Required) Name of Provider VDC storage profile.
- `enabled` - (Optional) True if this storage profile is enabled for use in the VDC. Default is true.
- `limit` - (Required) Maximum number of MB allocated for this storage profile. A value of 0 specifies unlimited MB.
- `default` - (Required) True if this is default storage profile for this VDC. The default storage profile is used when an object that can specify a storage profile is created with no storage profile specified.

» Compute Capacity

Capacity must be specified twice, once for `memory` and another for `cpu`. Each has the same structure:

- `allocated` - (Optional) Capacity that is committed to be available. Value in MB or MHz. Used with AllocationPool ("Allocation pool") and ReservationPool ("Reservation pool").
- `limit` - (Optional) Capacity limit relative to the value specified for Allocation. It must not be less than that value. If it is greater than that value, it implies over provisioning. A value of 0 specifies unlimited units. Value in MB or MHz. Used with AllocationVApp ("Pay as you go").

» vcd_lb_service_monitor

Provides a vCloud Director Edge Gateway Load Balancer Service Monitor resource. A service monitor defines health check parameters for a particular type of network traffic. It can be associated with a pool. Pool members are monitored according to the service monitor parameters.

Note: To make load balancing work one must ensure that load balancing is enabled on edge gateway (edge gateway must be advanced). This depends on NSX version to work properly. Please refer to VMware Product Interoperability Matrices to check supported vCloud director and NSX for vSphere configurations.

Note: The vCloud Director API for NSX supports a subset of the operations and objects defined in the NSX vSphere API Guide. The API supports NSX 6.2, 6.3, and 6.4.

Supported in provider *v2.4+*

» Example Usage

```
resource "vcd_lb_service_monitor" "monitor" {
  org          = "my-org"
  vdc          = "my-org-vdc"
  edge_gateway = "my-edge-gw"

  name        = "http-monitor"
  interval    = "5"
  timeout     = "20"
  max_retries = "3"
  type        = "http"
  method      = "GET"
  url         = "/health"
  send        = "{\"key\": \"value\"}"
  extension = {
    content-type = "application/json"
    linespan     = ""
  }
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the service monitor is to be created
- `name` - (Required) Service Monitor name
- `interval` - (Optional) Interval in seconds at which a server is to be monitored using the specified Method. Defaults to 10
- `timeout` - (Optional) Maximum time in seconds within which a response from the server must be received. Defaults to 15
- `max_retries` - (Optional) Number of times the specified monitoring Method must fail sequentially before the server is declared down. Defaults to 3
- `type` - (Required) Select the way in which you want to send the health check request to the server — `http`, `https`, `tcp`, `icmp`, or `udp`. Depending

on the type selected, the remaining attributes are allowed or not
- `method` - (Optional) For types `http` and `https`. Select http method to be used to detect server status. One of OPTIONS, GET, HEAD, POST, PUT, DELETE, TRACE, or CONNECT
- `url` - (Optional) For types `http` and `https`. URL to be used in the server status request
- `send` - (Optional) For types `http`, `https`, and `udp`. The data to be sent.
- `expected` - (Optional) For types `http` and `https`. String that the monitor expects to match in the status line of the HTTP or HTTPS response (for example, `HTTP/1.1`)
- `receive` - (Optional) For types `http`, `https`, and `udp`. The string to be matched in the response content. **Note**: When `expected` is not matched, the monitor does not try to match the Receive content
- `extension` - (Optional) A map of advanced monitor parameters as key=value pairs (i.e. `max-age=SECONDS`, `invert-regex`) **Note**: When you need a value of `key` only format just set value to empty string (i.e. `linespan = ""`)

» Attribute Reference

The following attributes are exported on the base level of this resource:

- `id` - The NSX ID of the load balancer service monitor

» Importing

Note: The current implementation of Terraform import can only import resources into the state. It does not generate configuration. More information.

An existing load balancer service monitor can be imported into this resource via supplying the full dot separated path for load balancer service monitor. An example is below:

```
terraform import vcd_lb_service_monitor.imported my-org.my-org-vdc.my-edge-gw.my-lb-service-
```

The above would import the service monitor named `my-lb-service-monitor` that is defined on edge gateway `my-edge-gw` which is configured in organization named `my-org` and vDC named `my-org-vdc`.

» vcd_lb_server_pool

Provides a vCloud Director Edge Gateway Load Balancer Server Pool resource. A Server Pool can have a group of backend servers set (defined as pool members),

manages load balancer distribution methods, and may have a service monitor attached to it for health check parameters.

Note: To make load balancing work one must ensure that load balancing is enabled on edge gateway. This depends on NSX version to work properly. Please refer to VMware Product Interoperability Matrices to check supported vCloud director and NSX for vSphere configurations.

Note: The vCloud Director API for NSX supports a subset of the operations and objects defined in the NSX vSphere API Guide. The API supports NSX 6.2, 6.3, and 6.4.

Supported in provider *v2.4+*

» Example Usage 1 (Simple Server Pool without Service Monitor)

```
resource "vcd_lb_server_pool" "web-servers" {
  org          = "my-org"
  vdc          = "my-org-vdc"
  edge_gateway = "my-edge-gw"

  name      = "web-servers"
  algorithm = "round-robin"

  member {
    condition       = "enabled"
    name            = "member1"
    ip_address      = "1.1.1.1"
    port            = 8443
    monitor_port    = 9000
    weight          = 1
    min_connections = 0
    max_connections = 100
  }
}
```

» Example Usage 2 (Server Pool with multiple members, algorithm parameters, and existing Service Monitor as data source)

```
data "vcd_lb_service_monitor" "web-monitor" {
  org          = "my-org"
  vdc          = "my-org-vdc"
  edge_gateway = "my-edge-gw"
```

```
  name = "existing-web-monitor-name"
}

resource "vcd_lb_server_pool" "web-servers" {
  org          = "my-org"
  vdc          = "my-org-vdc"
  edge_gateway = "my-edge-gw"

  name                 = "web-servers"
  description          = "description"
  algorithm            = "httpheader"
  algorithm_parameters = "headerName=host"
  enable_transparency  = "true"

  monitor_id = "${data.vcd_lb_service_monitor.web-monitor.id}"

  member {
    condition       = "enabled"
    name            = "member1"
    ip_address      = "1.1.1.1"
    port            = 8443
    monitor_port    = 9000
    weight          = 1
    min_connections = 0
    max_connections = 100
  }

  member {
    condition       = "drain"
    name            = "member2"
    ip_address      = "2.2.2.2"
    port            = 7000
    monitor_port    = 4000
    weight          = 2
    min_connections = 6
    max_connections = 8
  }
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at

provider level. Useful when connected as sysadmin working across different organisations

- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the server pool is to be created
- `name` - (Required) Server Pool name
- `description` - (Optional) Server Pool description
- `algorithm` - (Required) Server Pool load balancing method. Can be one of `ip-hash`, `round-robin`, `uri`, `leastconn`, `url`, or `httpheader`
- `algorithm_parameters` - (Optional) Valid only when `algorithm` is `httpheader` or `url`. The `httpheader` algorithm parameter has one option `headerName=<name>` while the `url` algorithm parameter has option `urlParam=<url>`.
- `enable_transparency` - (Optional) When transparency is `false` (default) backend servers see the IP address of the traffic source as the internal IP address of the load balancer. When it is `true` the source IP address is the actual IP address of the client and the edge gateway must be set as the default gateway to ensure that return packets go through the edge gateway.
- `monitor_id` - (Optional) `vcd_lb_service_monitor` resource `id` to attach to server pool for health check parameters
- `member` - (Optional) A block to define server pool members. Multiple can be used. See Member and example for usage details.

» Member

- `condition` - (Required) State of member in a pool. One of `enabled`, `disabled`, or `drain`. When member condition is set to `drain` it stops taking new connections and calls, while it allows its sessions on existing connections to continue until they naturally end. This allows to gracefully remove member node from load balancing rotation.
- `name` - (Required) Member name
- `ip_address` - (Required) Member IP address
- `port` - (Required) The port at which the member is to receive traffic from the load balancer.
- `monitor_port` - (Required) Monitor Port at which the member is to receive health monitor requests. **Note:** can be the same as `port`
- `weight` - (Required) The proportion of traffic this member is to handle. Must be an integer in the range 1-256.
- `min_connections` - (Optional) The maximum number of concurrent connections the member can handle. **Note:** when the number of incoming requests exceeds the maximum, requests are queued and the load balancer waits for a connection to be released.

- `max_connections` - (Optional) The minimum number of concurrent connections a member must always accept.

» Attribute Reference

The following attributes are exported on this resource:

- `id` - The NSX ID of the load balancer server pool

Additionally each of members defined in blocks expose their own `id` fields as well

» Importing

Note: The current implementation of Terraform import can only import resources into the state. It does not generate configuration. More information.

An existing load balancer server pool can be imported into this resource via supplying the full dot separated path for load balancer service monitor. An example is below:

```
terraform import vcd_lb_server_pool.imported my-org.my-org-vdc.my-edge-gw.my-lb-server-pool
```

The above would import the server pool named `my-lb-server-pool` that is defined on edge gateway `my-edge-gw` which is configured in organization named `my-org` and vDC named `my-org-vdc`.

» vcd_lb_app_profile

Provides a vCloud Director Edge Gateway Load Balancer Application Profile resource. An application profile defines the behavior of the load balancer for a particular type of network traffic. After configuring a profile, you associate it with a virtual server. The virtual server then processes traffic according to the values specified in the profile.

Note: This resource does not currently support attaching Pool and Virtual Server certificates. The `enable_pool_side_ssl` only toggles the option, but does not setup certificates.

Note: To make load balancing work one must ensure that load balancing is enabled on edge gateway (edge gateway must be advanced). This depends on NSX version to work properly. Please refer to VMware Product Interoperability Matrices to check supported vCloud director and NSX for vSphere configurations.

Note: The vCloud Director API for NSX supports a subset of the operations and objects defined in the NSX vSphere API Guide. The API supports NSX 6.2, 6.3, and 6.4.

Supported in provider *v2.4+*

» Example Usage 1 (TCP Application Profile)

```
resource "vcd_lb_app_profile" "tcp" {
  org          = "my-org"
  vdc          = "my-org-vdc"
  edge_gateway = "my-edge-gw"

  name = "tcp-app-profile"
  type = "tcp"
}
```

» Example Usage 2 (HTTP Cookie based Application Profile)

```
resource "vcd_lb_app_profile" "http" {
  org          = "my-org"
  vdc          = "my-org-vdc"
  edge_gateway = "my-edge-gw"

  name = "http-profile"
  type = "http"

  http_redirect_url              = "/service-one"
  persistence_mechanism          = "cookie"
  cookie_name                    = "JSESSIONID"
  cookie_mode                    = "insert"
  insert_x_forwarded_http_header = "true"
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level

- `edge_gateway` - (Required) The name of the edge gateway on which the application profile is to be created
- `name` - (Required) Application profile name
- `type` - (Required) Protocol type used to send requests to the server. One of `tcp`, `udp`, `http`, or `https`
- `enable_ssl_passthrough` - (Optional) Enable SSL authentication to be passed through to the virtual server. Otherwise SSL authentication takes place at the destination address
- `http_redirect_url` - (Optional) The URL to which traffic that arrives at the destination address should be redirected. Only applies for types `http` and `https`
- `persistence_mechanism` - (Optional) Persistence mechanism for the profile. One of 'cookie', 'ssl-sessionid', 'sourceip'
- `cookie_name` - (Optional) Used to uniquely identify the session the first time a client accesses the site. The load balancer refers to this cookie when connecting subsequent requests in the session, so that they all go to the same virtual server. Only applies for `persistence_mechanism` 'cookie'
- `cookie_mode` - (Optional) The mode by which the cookie should be inserted. One of 'insert', 'prefix', or 'appsession'
- `expiration` - (Optional) Length of time in seconds that persistence stays in effect
- `insert_x_forwarded_http_header` - (Optional) Enables 'X-Forwarded-For' header for identifying the originating IP address of a client connecting to a Web server through the load balancer. Only applies for types `http` and `https`
- `enable_pool_side_ssl` - (Optional) Enable to define the certificate, CAs, or CRLs used to authenticate the load balancer from the server side. **Note:** This resource does not currently support attaching Pool and Virtual Server certificates therefore this toggle only enables it. To make it fully work certificates must be currently attached manually.

» Attribute Reference

The following attributes are exported on this resource:

- `id` - The NSX ID of the load balancer application profile

» Importing

Note: The current implementation of Terraform import can only import resources into the state. It does not generate configuration. More information.

An existing load balancer application profile can be imported into this resource via supplying the full dot separated path for load balancer application profile. An example is below:

```
terraform import vcd_lb_app_profile.imported my-org.my-org-vdc.my-edge-gw.my-lb-app-profile
```

The above would import the application profile named `my-lb-app-profile` that is defined on edge gateway `my-edge-gw` which is configured in organization named `my-org` and vDC named `my-org-vdc`.

» vcd__lb__app__rule

Provides a vCloud Director Edge Gateway Load Balancer Application Rule resource. An application rule allows to directly manipulate and manage IP application traffic with load balancer.

Note: To make load balancing work one must ensure that load balancing is enabled on edge gateway (edge gateway must be advanced). This depends on NSX version to work properly. Please refer to VMware Product Interoperability Matrices to check supported vCloud director and NSX for vSphere configurations.

Note: The vCloud Director API for NSX supports a subset of the operations and objects defined in the NSX vSphere API Guide. The API supports NSX 6.2, 6.3, and 6.4.

Supported in provider *v2.4+*

» Example Usage 1 (Application rule with single line script)

```
resource "vcd_lb_app_rule" "example-one" {
  edge_gateway = "my-edge-gw"
  org          = "my-org"
  vdc          = "my-org-vdc"

  name = "script1"
  script = "acl vmware_page url_beg / vmware redirect location https://www.vmware.com/ ifvmw
}
```

» Example Usage 2 (Application rule with multi line script)

```
resource "vcd_lb_app_rule" "example-two" {
  edge_gateway = "my-edge-gw"
  org          = "my-org"
  vdc          = "my-org-vdc"
  name         = "script1"
  script = <<-EOT
    acl vmware_page url_beg / vmware redirect location https://www.vmware.com/ ifvmware_page
```

```
    acl other_page2 url_beg / other2 redirect location https://www.other2.com/ ifother_page2
    acl hello payload(0,6) -m bin 48656c6c6f0a
  EOT
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the application rule is to be created
- `name` - (Required) Application rule name
- `script` - (Required) A multiline application rule script. Terraform's HEREDOC syntax may be useful for multiline scripts. **Note:** For information on the application rule syntax, see more in vCloud Director documentation

» Attribute Reference

The following attributes are exported on this resource:

- `id` - The NSX ID of the load balancer application rule

» Importing

Note: The current implementation of Terraform import can only import resources into the state. It does not generate configuration. More information.

An existing load balancer application rule can be imported into this resource via supplying the full dot separated path for load balancer application rule. An example is below:

```
terraform import vcd_lb_app_rule.imported my-org.my-org-vdc.my-edge-gw.my-lb-app-rule
```

The above would import the application rule named `my-lb-app-my-lb-app-rule` that is defined on edge gateway `my-edge-gw` which is configured in organization named `my-org` and vDC named `my-org-vdc`.

» vcd_lb_virtual_server

Provides a vCloud Director edge gateway load balancer virtual server resource. Adds an edge gateway internal or uplink interface as a virtual server. A virtual server has a public IP address and services all incoming client requests.

Note: To make load balancing work one must ensure that load balancing is enabled on edge gateway (edge gateway must be advanced). This depends on NSX version to work properly. Please refer to VMware Product Interoperability Matrices to check supported vCloud director and NSX for vSphere configurations.

Note: The vCloud Director API for NSX supports a subset of the operations and objects defined in the NSX vSphere API Guide. The API supports NSX 6.2, 6.3, and 6.4.

Supported in provider *v2.4+*

» Example Usage 1 (HTTP virtual server)

```
resource "vcd_lb_virtual_server" "http" {
  org          = "my-org"
  vdc          = "my-org-vdc"
  edge_gateway = "my-edge-gw"

  name       = "http-virtual-server"
  ip_address = "1.1.1.1" # Edge gateway uplink interface IP
  protocol   = "http"    # Must be the same as specified in application profile
  port       = 80

  app_profile_id = "${vcd_lb_app_profile.http.id}"
  server_pool_id = "${vcd_lb_server_pool.web-servers.id}"
  app_rule_ids   = ["${vcd_lb_app_rule.redirect.id}", "${vcd_lb_app_rule.language.id}"]
}
```

» Example Usage 2 (Complete load balancer setup)

```
variable "org" {
  default = "my-org"
}

variable "vdc" {
  default = "my-org-vdc"
}
```

```
variable "edge_gateway" {
  default = "my-edge-gw"
}

variable "protocol" {
  default = "http"
}

variable "edge_gateway_ip" {
  default = "192.168.1.110"  # IP address of edge gateway uplink interface
}

resource "vcd_lb_virtual_server" "http" {
  org          = "${var.org}"
  vdc          = "${var.vdc}"
  edge_gateway = "${var.edge_gateway}"

  name       = "my-virtual-server"
  ip_address = "${var.edge_gateway_ip}"
  protocol   = "${var.protocol}"
  port       = 8888

  app_profile_id = "${vcd_lb_app_profile.http.id}"
  server_pool_id = "${vcd_lb_server_pool.web-servers.id}"
  app_rule_ids   = ["${vcd_lb_app_rule.redirect.id}"]
}

resource "vcd_lb_service_monitor" "monitor" {
  org          = "${var.org}"
  vdc          = "${var.vdc}"
  edge_gateway = "${var.edge_gateway}"

  name        = "http-monitor"
  interval    = "5"
  timeout     = "20"
  max_retries = "3"
  type        = "${var.protocol}"
  method      = "GET"
  url         = "/health"
  send        = "{\"key\": \"value\"}"
  extension = {
    content-type = "application/json"
    linespan     = ""
  }
}
```

```
resource "vcd_lb_server_pool" "web-servers" {
  org          = "${var.org}"
  vdc          = "${var.vdc}"
  edge_gateway = "${var.edge_gateway}"

  name                 = "web-servers"
  description          = "description"
  algorithm            = "httpheader"
  algorithm_parameters = "headerName=host"
  enable_transparency  = "true"

  monitor_id = "${vcd_lb_service_monitor.monitor.id}"

  member {
    condition       = "enabled"
    name            = "member1"
    ip_address      = "1.1.1.1"
    port            = 8443
    monitor_port    = 9000
    weight          = 1
    min_connections = 0
    max_connections = 100
  }

  member {
    condition       = "drain"
    name            = "member2"
    ip_address      = "2.2.2.2"
    port            = 7000
    monitor_port    = 4000
    weight          = 2
    min_connections = 6
    max_connections = 8
  }
}

resource "vcd_lb_app_profile" "http" {
  org          = "${var.org}"
  vdc          = "${var.vdc}"
  edge_gateway = "${var.edge_gateway}"

  name = "http-app-profile"
  type = "${var.protocol}"
}

resource "vcd_lb_app_rule" "redirect" {
```

```
  org          = "${var.org}"
  vdc          = "${var.vdc}"
  edge_gateway = "${var.edge_gateway}"

  name   = "redirect"
  script = "acl vmware_page url_beg / vmware redirect location https://www.vmware.com/ ifvmw
}
```

» Argument Reference

The following arguments are supported:

- `org` - (Optional) The name of organization to use, optional if defined at provider level. Useful when connected as sysadmin working across different organisations
- `vdc` - (Optional) The name of VDC to use, optional if defined at provider level
- `edge_gateway` - (Required) The name of the edge gateway on which the virtual server is to be created
- `name` - (Required) Virtual server name
- `description` - (Optional) Virtual server description
- `enabled` - (Optional) Defines if the virtual server is enabled. Default `true`
- `enable_acceleration` - (Optional) Defines if the virtual server uses acceleration. Default `false`
- `ip_address` - (Required) Set the IP address that the load balancer listens on
- `protocol` - (Required) Select the protocol that the virtual server accepts. One of `tcp`, `udp`, `http`, or `https` **Note**: You must select the same protocol used by the selected **Application Profile**
- `port` - (Required) The port number that the load balancer listens on
- `connection_limit` - (Optional) Maximum concurrent connections that the virtual server can process
- `connection_rate_limit` - (Optional) Maximum incoming new connection requests per second
- `server_pool_id` - (Optional) The server pool that the load balancer will use
- `app_profile_id` - (Optional) Application profile ID to be associated with the virtual server
- `app_rule_ids` - (Optional) List of attached application rule IDs

» Attribute Reference

The following attributes are exported on the base level of this resource:

- `id` - The NSX ID of the load balancer virtual server

» Importing

Note: The current implementation of Terraform import can only import resources into the state. It does not generate configuration. More information.

An existing load balancer virtual server can be imported into this resource via supplying the full dot separated path for load balancer virtual server. An example is below:

```
terraform import vcd_lb_virtual_server.imported my-org.my-org-vdc.my-edge-gw.my-lb-virtual-s
```

The above would import the virtual server named `my-lb-virtual-server` that is defined on edge gateway `my-edge-gw` which is configured in organization named `my-org` and vDC named `my-org-vdc`.