

Sử dụng VKS với Terraform

Terraform là gì?

Terraform là một cơ sở hạ tầng nguồn mở dưới dạng công cụ mã cho phép người dùng quản lý cơ sở hạ tầng của họ một cách dễ dàng và hiệu quả trên các nền tảng đám mây khác nhau, chẳng hạn như VNG Cloud, AWS, Google Cloud và Azure. Máy chủ Terraform đề cập đến phiên bản của công cụ Terraform đang chạy trên một máy chủ hoặc máy cụ thể. Đây là nơi mã cơ sở hạ tầng được viết và thực thi, cho phép người dùng tạo, sửa đổi và hủy tài nguyên trên nền tảng đám mây.

Bản thân Terraform không có giao diện người dùng đồ họa, thay vào đó người dùng tương tác với nó bằng giao diện dòng lệnh. Terraform yêu cầu tài khoản và khóa của nhà cung cấp đám mây được định cấu hình cùng với tệp cấu hình Terraform để thực thi cơ sở hạ tầng dưới dạng mã. Ngoài ra, Terraform có thể hoạt động trong môi trường nhóm nơi nhiều người dùng có thể cộng tác trên cùng một cơ sở mã cơ sở hạ tầng, khiến nó trở thành một công cụ mạnh mẽ và linh hoạt để quản lý cơ sở hạ tầng đám mây.

Các bước thực hiện

Để khởi tạo một Cluster Kubernetes bằng Terraform, bạn cần thực hiện các bước sau:

1. **Truy cập IAM Portal** tại [đây](#), thực hiện tạo Service Account với quyền hạn **VKS Full Access**. Cụ thể, tại trang IAM, bạn có thể:
2. Chọn "**Create a Service Account**", điền tên cho Service Account và nhấn **Next Step** để gán quyền cho Service Account.
3. Tìm và chọn **Policy: VKSFullAccess** sau đó nhấn "**Create a Service Account**" để tạo Service Account, **Policy: VKSFullAccess** do VNG Cloud tạo ra, bạn không thể xóa các policy này.
4. Sau khi tạo thành công bạn cần phải lưu lại **Client_ID** và **Secret_Key** của Service Account để thực hiện bước tiếp theo.
5. **Truy cập VKS Portal** tại [đây](#), thực hiện **Activate** dịch vụ VKS ở tab **Overview**. Hãy chờ đợi tới khi chúng tôi khởi tạo thành công tài khoản VKS của bạn.
6. **Cài đặt Terraform:**
7. Tải xuống và cài đặt Terraform cho hệ điều hành của bạn từ <https://developer.hashicorp.com/terraform/install>.
8. **Khởi tạo cấu hình Terraform:**
9. Tạo tệp `variable.tf` và khai báo thông tin Service Account trong file này.
10. Tạo tệp `main.tf` và định nghĩa các tài nguyên Kubernetes Cluster mà bạn muốn tạo.

Ví dụ:

- Tệp `variable.tf`: bạn cần thay thế Client ID và Client Secret đã khởi tạo ở bước 1 ở file này.

```
variable "client_id" {
  type = string
  default = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
}
variable "client_secret" {
  type = string
  default = "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxx"
}
```

- Trên file **main.tf**, bạn cần có thể thêm resource để tạo Cluster/ Node Group:
- Tạo Cluster my-vks-cluster và Node Group my-nodegroup độc lập:

```
resource "vngcloud_vks_cluster" "primary" {
  name      = "my-cluster"
  cidr      = "172.16.0.0/16"
  vpc_id    = "net-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  subnet_id = "sub-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}

resource "vngcloud_vks_cluster_node_group" "primary" {
  name      = "my-nodegroup"
  ssh_key_id = "ssh-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  cluster_id = vngcloud_vks_cluster.primary.id
}
```

- Tạo Cluster với Default Node Group

```
resource "vngcloud_vks_cluster" "primary" {
  name      = "my-cluster"
  cidr      = "172.16.0.0/16"
  vpc_id    = "net-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  subnet_id = "sub-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  node_group {
    name      = "my-nodegroup"
    ssh_key_id = "ssh-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
  }
}
```

{% hint style="info" %} **Chú ý:**

- Chúng tôi khuyên bạn nên tạo và quản lý các Cluster, Node Group dưới dạng resource riêng biệt, như trong ví dụ bên dưới. Điều này cho phép bạn thêm hoặc xóa các Node Group mà không cần tạo lại toàn bộ Cluster. Nếu bạn khai báo trực tiếp Node Group Default trong tài nguyên vngcloud_vks_cluster, bạn không thể xóa chúng mà không tạo lại chính Cluster đó.
- Trong file main.tf, để khởi tạo một cluster với một node group thành công, bạn bắt buộc cần nhập thông tin của 4 field sau:

```
vpc_id = "net-xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" subnet_id = "sub-
xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx" ssh_key_id= "ssh-xxxxxxxx-xxxx-xxxx-
xxxx-xxxxxxxxxxxx"
```

Các ví dụ tham khảo

Example Usage 1 - Create a Cluster with Network type CALICO OVERLAY and a Node Group with AutoScale Mode

```
terraform {
  required_providers {
    vngcloud = {
      source = "vngcloud/vngcloud"
      version = "1.2.7"
    }
  }
}
```

```

    }
  }
}
resource "vngcloud_vks_cluster" "primary" {
  name          = "cluster-demo"
  description   = "Cluster create via terraform"
  version       = "v1.29.1"
  cidr          = "172.16.0.0/16"
  enable_private_cluster = false
  network_type  = "CALICO"
  vpc_id        = "net-70ef12d4-d619-43fc-88f0-1c1511683123"
  subnet_id     = "sub-0725ef54-a32e-404c-96f2-34745239c123"
  enabled_load_balancer_plugin = true
  enabled_block_store_csi_plugin = true
}

resource "vngcloud_vks_cluster_node_group" "primary" {
  cluster_id = vngcloud_vks_cluster.primary.id
  name       = "nodegroup1"
  num_nodes  = 3
  auto_scale_config {
    min_size = 0
    max_size = 5
  }
  upgrade_config {
    strategy = "SURGE"
    max_surge = 1
    max_unavailable = 0
  }
  image_id = "img-108b3a77-ab58-4000-9b3e-190d0b4b07fc"
  flavor_id = "flav-9e88cfb4-ec31-4ad4-8ba5-243459f6d123"
  disk_size = 50
  disk_type = "vtype-61c3fc5b-f4e9-45b4-8957-8aa7b6029018"
  enable_private_nodes = false
  ssh_key_id = "ssh-f923c53c-cba7-4131-9f86-175d04ae2123"
  security_groups = ["secg-faf05344-fbd6-4f10-80a2-cda08d15ba5e"]
  labels = {
    "test" = "terraform"
  }
  taint {
    key    = "key1"
    value  = "value1"
    effect = "PreferNoSchedule"
  }
}
}

```

Example Usage 2 - Create a Cluster with Network type CILIUM OVERLAY and a Node Group with AutoScale Mode

```

terraform {
  required_providers {
    vngcloud = {
      source = "vngcloud/vngcloud"
    }
  }
}

```

```

        version = "1.2.7"
    }
}
}
resource "vngcloud_vks_cluster" "primary" {
    name          = "cluster-demo"
    description   = "Cluster create via terraform"
    version       = "v1.29.1"
    cidr          = "172.16.0.0/16"
    enable_private_cluster = false
    network_type  = "CILIUM_OVERLAY"
    vpc_id        = "net-70ef12d4-d619-43fc-88f0-1c1511683123"
    subnet_id     = "sub-0725ef54-a32e-404c-96f2-34745239c123"
    enabled_load_balancer_plugin = true
    enabled_block_store_csi_plugin = true
}

resource "vngcloud_vks_cluster_node_group" "primary" {
    cluster_id = vngcloud_vks_cluster.primary.id
    name       = "nodegroup1"
    num_nodes  = 3
    auto_scale_config {
        min_size = 0
        max_size = 5
    }
    upgrade_config {
        strategy = "SURGE"
        max_surge = 1
        max_unavailable = 0
    }
    image_id = "img-108b3a77-ab58-4000-9b3e-190d0b4b07fc"
    flavor_id = "flav-9e88cfb4-ec31-4ad4-8ba5-243459f6d123"
    disk_size = 50
    disk_type = "vtype-61c3fc5b-f4e9-45b4-8957-8aa7b6029018"
    enable_private_nodes = false
    ssh_key_id = "ssh-f923c53c-cba7-4131-9f86-175d04ae2123"
    security_groups = ["secg-faf05344-fbd6-4f10-80a2-cda08d15ba5e"]
    labels = {
        "test" = "terraform"
    }
    taint {
        key    = "key1"
        value  = "value1"
        effect = "PreferNoSchedule"
    }
}
}

```

Example Usage 3 - Create a Cluster with Network type CILIUM VPC Native Routing and a Node Group with AutoScale Mode

```

terraform {
    required_providers {
        vngcloud = {

```

```

        source = "vngcloud/vngcloud"
        version = "1.2.7"
    }
}
}
resource "vngcloud_vks_cluster" "primary" {
    name          = "cluster-demo"
    description = "Cluster create via terraform"
    version = "v1.29.1"
    enable_private_cluster = false
    network_type = "CILIUM_NATIVE_ROUTING"
    vpc_id       = "net-70ef12d4-d619-43fc-88f0-1c1511683123"
    subnet_id    = "sub-0725ef54-a32e-404c-96f2-34745239c123"
    secondary_subnets = ["10.200.27.0/24", "10.200.28.0/24"]
    node_netmask_size = 25
    enable_service_endpoint = false
    enabled_load_balancer_plugin = true
    enabled_block_store_csi_plugin = true
}

resource "vngcloud_vks_cluster_node_group" "primary" {
    cluster_id = vngcloud_vks_cluster.primary.id
    name = "nodegroup1"
    num_nodes = 3
    auto_scale_config {
        min_size = 0
        max_size = 5
    }
    upgrade_config {
        strategy = "SURGE"
        max_surge = 1
        max_unavailable = 0
    }
    image_id = "img-108b3a77-ab58-4000-9b3e-190d0b4b07fc"
    flavor_id = "flav-9e88cfb4-ec31-4ad4-8ba5-243459f6d123"
    subnet_id = "sub-cddd7ffa-be05-4698-9b3d-794e1adfcfce"
    secondary_subnets = ["10.200.27.0/24", "10.200.28.0/24"]
    disk_size = 50
    disk_type = "vtype-61c3fc5b-f4e9-45b4-8957-8aa7b6029018"
    enable_private_nodes = false
    ssh_key_id = "ssh-f923c53c-cba7-4131-9f86-175d04ae2123"
    security_groups = ["secg-faf05344-fbd6-4f10-80a2-cda08d15ba5e"]
    labels = {
        "test" = "terraform"
    }
    taint {
        key    = "key1"
        value  = "value1"
        effect = "PreferNoSchedule"
    }
}
}

```

{% hint style="info" %} **Chú ý:**

- Để lấy image_id bạn mong muốn sử dụng, bạn có thể truy cập vào VKS Portal, chọn menu System Image và lấy ID mà bạn mong muốn hoặc lấy thông tin này tại [đây](#).
- Để lấy flavor_id bạn mong muốn sử dụng cho Node group của bạn, vui lòng lấy ID tại [đây](#).

Khởi chạy Terraform command

- Sau khi hoàn tất các thông tin trên, thực hiện chạy lệnh bên dưới:

```
terraform init
```

- Sau đó, bạn để xem những thay đổi sẽ được áp dụng trên những resource mà terraform đang quản lý bạn có thể chạy:

```
terraform plan
```

- Cuối cùng bạn chọn chạy dòng lệnh:

```
terraform apply
```

- Chọn **YES** để thực hiện việc khởi tạo Cluster, Node Group thông qua Terraform

Kiểm tra Cluster vừa tạo trên giao diện VNG Cloud Portal

Sau khi khởi tạo thành công Terraform, bạn có thể lên VKS Portal để xem thông tin Cluster vừa tạo.

Tham khảo thêm về cách sử dụng Terraform để làm việc với VKS tại [đây](#).

Một số lưu ý khi sử dụng VKS với Terraform:

Khi sử dụng **Terraform** để khởi tạo **Cluster** và **Node Group** trên hệ thống VKS, nếu bạn thay đổi một trong các field sau, hệ thống sẽ tự động xóa Node Group/ Cluster và thực hiện khởi tạo lại Node Group/ Cluster theo thông số mới tương ứng. Việc xóa sẽ được thực hiện trước khi tạo Node Group/ Cluster mới.

- Đối với resource vngcloud_vks_cluster, các field khi bạn thay đổi hệ thống sẽ xóa Cluster và tạo lại bao gồm:
 - name
 - description
 - enable_private_cluster
 - network_type
 - vpc_id
 - subnet_id
 - cidr
 - enabled_load_balancer_plugin
 - enabled_block_store_csi_plugin
 - node_group
 - secondary_subnets
 - node_netmask_size
- Đối với resource vngcloud_vks_cluster_node_group, các field khi bạn thay đổi hệ thống sẽ xóa Cluster và tạo lại bao gồm:
 - cluster_id
 - name
 - flavor_id

- disk_size
- disk_type
- enable_private_nodes
- ssh_key_id
- secondary_subnets
- enabled_encryption_volume
- subnet_id

Để chỉ định hệ thống tạo cluster/node group mới rồi mới thực hiện xóa cluster/ node group cũ, bạn có thể thêm tham số lifecycle { create_before_destroy = true } vào file main.tf của bạn. Cụ thể:

- Đối với resource vngcloud_vks_cluster

```
resource "vngcloud_vks_cluster" "example" {
  # ...

  lifecycle {
    create_before_destroy = true
  }
}
```

- Đối với resource vngcloud_vks_cluster_node_group

```
resource "vngcloud_vks_cluster_node_group" "example" {
  # ...

  lifecycle {
    create_before_destroy = true
  }
}
```