Creation d'une VM linux avec opentofu

Prérequis

- Disposer d'un cluster proxmox valide
- Avoir Opentofu d'installé
- Avoir un serveur linux (Bastion) au seins du cluster avec un e paire de clé SSH de disponible

Mise en place de la configuration (SSH)

- Transférer la clé SSH publique sur le serveur proxmox
  - Sur le Serveur (Bastion)

```
> cat .ssh/id_ed25519.pub
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAICqLxAZ0VeOyxKeDgxb9Gp7upKGs/w1+NrDRXcVYjLii
root@adm-front-01
```

  - Sur le Proxmox

```
root@pve-front-01:~# echo "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAICqLxAZ0VeOyxKeDgxb9Gp7upKGs/w1+NrDRXcVYjLii root@adm-
front-01" >> /root/.ssh/authorized_keys
```

  - Test de la connexion depuis le serveur sur le proxmox

```
ssh -q -o BatchMode=yes -o ConnectTimeout=5 root@$PVE_ADDRESS exit
echo $?
# Si le retour de la commande est différent de 0 la connexion est en echec
```

- Mise en place du fichier provider
  - créer le fichier provider.tf

```
terraform {
    required_providers {
    proxmox = {
      source = "bpg/proxmox" # -->
```

```
https://registry.terraform.io/providers/bpg/proxmox/latest/docs
      version = "0.74.1"
    }
  }
}

provider "proxmox" {
  endpoint = "https://{PVE_ADDRESS}:8006"
  username = "root@pam"
  password = "{PVE_ROOT_PASSWORD}"
  insecure = true

  ssh {
    agent = true
    username = "root"

    node {
      name = "pve-front-01"
      address = "{PVE_ADDRESS}"
      port = 2222 # -> PVE SSH PORT
    }
  }
}
```

> Lien vers le fichier provider.tf

- Initialisation du provider

```
> tofu init

Initializing the backend...

Initializing provider plugins...
- Reusing previous version of bpg/proxmox from the dependency lock file
- Using previously-installed bpg/proxmox v0.74.1

OpenTofu has been successfully initialized!

You may now begin working with OpenTofu. Try running "tofu plan" to see
any changes that are required for your infrastructure. All OpenTofu commands
```

```
should now work.

If you ever set or change modules or backend configuration for OpenTofu,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

Mise en place des fichiers de configuration VMs

- Téléchargement du fichier cloudinit via tofu
  - Créer le fichier file.tf

```
resource "proxmox_virtual_environment_file" "latest_debian_12" {
  content_type = "vztmpl"
  datastore_id = "local"
  node_name = "pve-front-01"
  source_file {
    path = "http://download.proxmox.com/images/system/debian-12-standard_12.7-
1_amd64.tar.zst"
  }
}
```

- Déclaration de la ressource Qemu (VM)
  - Créer le fichier test_qemu.tf

```
resource "proxmox_virtual_environment_vm" "test-qemu" {
  count = 1
  name      = "test-qemu-0${ count.index + 1 }"
  node_name = "pve-front-01"

  initialization {
    datastore_id = "local"
    dns {
      domain = "kvega.local"
      servers = ["1.1.1.1", "8.8.8.8"]
    }
    ip_config {
      ipv4 {
        address = "192.168.210.1${ count.index + 1 }/26"
        gateway = "192.168.210.1"
        # ou encore address = "192.168.1.100/24"
```

```
      }
    }
    user_account {
      # do not use this in production, configure your own ssh key instead!
      username = "admin"
      keys  = ["ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAICqLxAZ0VeOyxKeDgxb9Gp7upKGs/w1+NrDRXcVYjLii
root@adm-front-01",
        "ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIJHXgvmOWJLUrFgmlVa49RdqYqUsyjVAQgAuBq7K7nis
kvega@FR-LAP10398",]
      password = "Strongpassword"
    }

  }

  memory {
      dedicated = 4096
  }

  cpu {
      cores = 2
      type = "host"
  }

  network_device {
      bridge = "ovsbr2"
      model = "virtio"
  }

  lifecycle {
    ignore_changes = [
      network_device, # on conserve l'adresse MAC pour éviter de régénérer
la VM
    ]
  }

  operating_system {
    type = "l26"
```

```
  }

  disk {
    datastore_id = "local"
    file_id      = proxmox_virtual_environment_file.debian_cloud_image.id
    interface    = "virtio0"
    iothread     = true
    discard      = "on"
    size         = 30
  }
}
```

Application des configurations

Sur un terminal

- Test de configuration
  - `tofu plan` --> permet de voire ce qui sera fait en quelque sorte un dry-run
  - `tofu apply` --> va appliquer les modification si vous entrez le mot yes à la fin

---

Revision #2
Created 2025-04-04 06:05:38 UTC by kvega
Updated 2025-05-02 09:55:03 UTC by kvega