GESTION DE LXC/LXD CON TERRAFORM

Alumno: Ángel Almazán García
Curso: 2º CFGS ASIR

Indice

1. Terraform

Terraform es una herramienta para construir, cambiar y crear versiones de infraestructura de manera segura y eficiente. Terraform puede gestionar proveedores de servicios existentes y populares, así como soluciones internas personalizadas.

Los archivos de configuración describen a Terraform los componentes necesarios para ejecutar una sola aplicación o todo su centro de datos. Terraform genera un plan de ejecución que describe lo que hará para alcanzar el estado deseado y luego lo ejecuta para construir la infraestructura descrita. A medida que cambia la configuración, Terraform puede determinar qué cambió y crear planes de ejecución incrementales que se pueden aplicar.

Desde el punto de vista de un administrador esta herramienta proporciona la posibilidad de crear entornos fáciles de gestionar y con la posibilidad de usar los proveedores que se requieran o prefieran.

A diferencia de ansible, terraform es inmutable, es decir, la ejecución de un código terraform resultará siempre en el estado final de la infraestructura, mientras que ansible no será capaz de registrar cambios anteriores al código ejecutado.

Por ejemplo, en terraform, si tenemos 5 contenedores y queremos desplegar 10, solo tendremos que añadir 5 mas al código o contador del script y el resultado serán 10 contenedores. En el caso de ansible si tenemos esos mismos 5 contenedores y queremos desplegar 10, no podremos simplemeten añadirlos al código, ya que acabaríamos con un total de 15 contenedores.

Esto quiere decir, que para Terraform, lo que se borre o añada en código será lo que tengamos. Si declaramos 3 contenedores, tendremos 3. Si borramos uno del código y aplicamos, este contenedor será eliminado.

2. LXC

LXC es un administrador de contenedores del sistema, que ofrece una experiencia de usuario similar a la de las máquinas virtuales, pero en su lugar utiliza contenedores de Linux.

Los contenedores aprovechan el sistema operativo y el kernel del host en el que se encuentran, por eso los contenedores (en Linux) no pueden ejecutar Windows. Un hipervisor de máquina virtual es pesado, ya que tienen recursos reservados para su propio kernel de Linux y sistema operativo host. Los contenedores son más rápidos de implementar y pueden ser empaquetarlos de manera más ajustadaen el hardware.

2.1. Diferencia entre Docker y LXC

LXC está diseñado para ejecutarse como un sistema operativo de máquina completa (copia limpia de una distribución de Linux o un dispositivo completo) y proporcionar la experiencia de un sistema operativo completo. A LXC no le importa lo que se esté ejecutando en el contenedor, Docker sí. Docker, por otro lado, está diseñado para ejecutar una sola aplicación; están enfocados en ser contenedores basados en procesos.

3. Terraform + LXC/LXD

Para el desarrollo del proyecto vamos a utilizar LXD, una Rest API que facilitará la instalación de los contenedores usando terraform. Este sería nuestro proveedor de máquinas.

En Debian 10 este paquete debe ser instalado con snap. LXC recomienda el uso de ZFS para crear storage pools, pero en nuestro caso usuaremos BTRFS como sustituto.

Tendremos que asegurarnos de que el usuario que va a realizar la gestión de los contenedores este en el grupo lxd.

Ejecutaremos un lxd init y configuraremos la aplicación, dejando todo en predeterminado excepto el nombre del storage backend que será BTRFS y permitiremos que LXD sea accesible por la red:

Would you like LXD to be available over the network (yes/no) [default=no]: yes

También tendremos en cuenta que todos los contenedores que creemos serán automaticamente conectados a un adaptador puente que se creará entonces, e l cual se llamará lxdbr0.

4. LXD.TF

Este archivo será el que gestione la instalación de la infraestructura de contenedores. Para manterner simple el escenario, se compondrá de dos contenedores, uno de aplicación y otro de base de datos.

Código:

```
# proveedor para conectar a la infraestructura

terraform {
  required_providers {
    lxd = {
      source = "terraform-lxd/lxd"
    }
  }
}

provider "lxd" {
  generate_client_certificates = true
  accept_remote_certificate    = true
  lxd_remote {
    name     = "lxd-server-1"
    scheme   = "https"
    address  = "127.0.0.1"
```

```
    password = "pass"
    default  = true
  }
}



# Storage Pool
#resource "lxd_storage_pool" "default" {
#  config = {
#      "size"   = "5GB"
#      "source" = "/var/snap/lxd/common/lxd/disks/default.img"
#      "btrfs.pool_name" = "default"
#  }
#  driver = "btrfs"
#  name   = "default"
#}

# Contenedores
resource "lxd_container" "wp-database" {
        ephemeral = false
        limits = {
                "memory" = "256MB"
                "cpu" = 1
                }
        name = "wp-database"
        profiles = [
                "terraform_default",
                ]
        image = "2f63b2acd7d4"
        device {
                name = "eth0"
                type = "nic"

                properties = {
                        nictype = "bridged"
```

```
                parent = "lxdbr0"
                "ipv4.address" = "10.225.237.100"
                }
        }
    provisioner "remote-exec" {
        script = "/root/proyecto/database/dbscript.sh"
        connection {
            type = "ssh"
            host = "10.225.237.100"
            user = "root"
            password = "root"
            }
                }
}
resource "lxd_container" "wp-nginx" {
    ephemeral = false
    limits = {
        "memory" = "256MB"
        "cpu" = 1
        }
    name = "wp-nginx"
    profiles = [
        "terraform_default",
        ]
    image = "2f63b2acd7d4"
    device {
        name = "eth0"
        type = "nic"

        properties = {
            nictype = "bridged"
            parent = "lxdbr0"
            "ipv4.address" = "10.225.237.200"
            }
        }
```

```
    provisioner "remote-exec" {
        script = "/root/proyecto/wp-installer/script.sh"
        connection {
            type = "ssh"
            host = "10.225.237.200"
            user = "root"
            password = "root"
            }
                }
    provisioner "file" {
        source = "wp-installer/wordpress.conf"
        destination = "/etc/nginx/sites-enabled/wordpress"
        connection {
            type = "ssh"
            host = "10.225.237.200"
            user = "root"
            password = "root"
            }
        }
    provisioner "remote-exec" {
        inline = ["systemctl restart nginx",
        ]
        connection {
            type = "ssh"
            host = "10.225.237.200"
            user = "root"
            password = "root"
            }
        }
}



#Perfil
resource "lxd_profile" "terraform_default" {
  config     = {}
```

```
  description = "Default LXD profile created by terrraform"
  name        = "terraform_default"
  device {
    name       = "root"
    properties = {
      "path" = "/"
      "pool" = "default"
    }
    type       = "disk"
  }
}
```

Los scripts referenciados en el código de terraform se encuentran en /*root/*proyecto/, los cuales son ejecutados en sus respectivos contenedores. Se pueden observar en la OVA adjuntada.

5. Resultado de ejecución

Tras ejecutador el código, se ejecutan dos contenedores LXC que se configuración según el código indicado, resultando en el acceso a una app, en este caso Wordpress. Para comprobar el funcionamiento, ejecutamos en el host:

iptables -t nat -A PREROUTING -p tcp --dport 8080 -j DNAT --to 10.225.237.200:80

Esto nos permitirá visualizar la app desde el puerto 8080 del anfitrion del contenedor.

6. Conclusiones

Durante la realización del proyecto he comprendido ciertos aspectos de la aplicación que pondrián resultar bastante útil. En la empresa que trabaje de prácticas y en la que ahora estoy trabajando realiza instalaciones similares, sino iguales, en varias ocasiones, y de esta manera podriamos simplificar mucho las instalaciones que un principio tomarian horas, reducirlas a tal vez una hora o incluso, menos.

Hay otros muchos aspectos en los que me hubiese gustado expandir, como la idea original de instalar un proxy/balanceador de carga, pero al encontrarme con algunos problemas a la hora de programar la infraestructura, dedici que sería mas efectivo desarrollar una infraestructura mas simple para el proyecto, pero que funcionase sin problemas.