Documento que acompaña a la práctica 2.

Experto Universitario

DevOps&Cloud

Antonio Sánchez Antón

Información General

|  |  |  |  |
| --- | --- | --- | --- |
| Propietario: | Antonio Sánchez | Versión: | 1.0 |
| **Creado:** | 20/02/2021 | **Actualizado:** | 21/02/2021 |
| **Revisado para versión:** |  | **Dirigido a:** | Profesores UNIR |
|  |  |  |  |

Control de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Acción | Descripción | Fecha |
| 1.0 | Nuevo | Versión inicial | 20/02/2021 |

TABLA DE CONTENIDOS

[1 Introducción 4](#_Toc64829378)

[2 Infraestructura desplegada 5](#_Toc64829379)

[3 Despliegue Realizado 7](#_Toc64829380)

[4 Problemas encontrados 9](#_Toc64829381)

[5 ANEXO: PReparacion maquina local 10](#_Toc64829382)

# Introducción

El objetivo de la práctica es desplegar, en una serie de maquinas virtuales Linux, un clúster de Kubernetes y sobre él, una aplicación.

Como aplicación he elegido

* <https://hub.docker.com/r/alexwhen/docker-2048>

El objetivo de este documento es describir los elementos utilizados para desplegar un clúster de Kubernetes en máquinas virtuales Azure.

Para ello se hace uso de las herramientas Terraform:

<https://www.terraform.io/>

con la que se hace la creación de la infraestructura en Azure

y Ansible:

<https://www.ansible.com/>

con la que se automatiza toda la configuración de dicha infraestructura, así como, el despliegue de la aplicación.

Todo el código está disponible en:

<https://github.com/asanton/Practica2Unir>

Se utilizará una maquina local Linux, que será desde donde se lancen todas las automatizaciones, tanto terraform como ansible.

# Infraestructura desplegada

Como se ha indicado anteriormente se hace uso de Terraform para desplegar la infraestructura necesaria en Azure.

Dicha infraestructura está formada por:

Diagrama

Descripción generada automáticamente

2 máquinas virtuales de tipo “Standard\_A2\_v2” con 2 CPU y 4Gb de RAM

Sobre ellas se despliega un sistema operativo Linux CentOS 8

Se crea una red “kubernetesnet” 10.0.0.0 /16 y sobre ella, la subnet “terraformsubnet” 10.0.1.0 /24 que será a la que pertenezcan las máquinas virtuales a desplegar.

Se crea un grupo de seguridad para ambas maquina donde, inicialmente, llevará una regla para permitir el acceso al puerto 22 (SSH).

Posteriormente, una vez desplegada la aplicación, se deberá abrir el puerto donde se ofrezca dicha aplicación

Una de la maquinas virtuales desempeñará los roles de máster de kubernetes y de servidor NFS y la otra desempeñará el rol de worker de kubernetes

# Despliegue Realizado

Si se requiere la preparación de una máquina desde la que hacer el despliegue revisar el “ANEXO: PREPARACIÓN MÁQUINA LOCAL”.

En caso contrario basta con clonar el repositorio:

<https://github.com/asanton/Practica2Unir.git>

**Creación Infraestructura Azure**

Nos posicionamos en la carpeta terraform

Hay que editar el archivo de plantilla main.tf y configurar

provider "azurerm" {

features {}

subscription\_id = "<SUBSCRIPCION ID>"

client\_id = "<APP\_ID>"

client\_secret = "<PASSWORD>"

tenant\_id = "<TENANT>"

}

Una vez realizado, ejecutar los comandos:

*-terraform init*

-*terraform apply*

La infraestructura se creará

Una vez creada hay que:

1. Identificar las direcciones ip de las maquinas creadas
2. Conectarnos via ssh con el usuario “adminUsername” para verificar que el despliegue es correcto

**Configuración infraestructura y despliegue de la aplicación**

He utilizado diferentes módulos de ansible para realizar las tareas de configuración (command, shell, user, file, dnf, lineinfile…….).

He utilizado “ansible-galaxy” para crear los roles a utilizar:

Nos posicionamos en la carpeta ansible donde tenemos:

* inventario\_local.yml
* inventario\_azure.yml
* playbookDespliegueApp.yml
* playbookDespliegueK8s.yml
* playbookPrueba.yml

utilizado internamente para probar el funcionamiento de los módulos de Ansible a utilizar

# Problemas encontrados

Fundamentalmente he encontrado 3 problemas que me han retrasado

1. Problema con la cuenta Azure for Students que me impedía crear un StorageAccount

Al crear el plan de terraform da un error:

**"Error: Error reading queue properties for AzureRM Storage Account..............Code= AutenticationFailed"**

Dicho error esta notificado a los profesores de la asignatura y, en el momento actual, no hay solución.

Como “workaround” siguiendo las indicaciones de los profesores se crea el plan sin hacer uso del StorageAccount

1. Problema en el despliegue de Kubernetes

La infraestructura original para desplegar estaba formada por 4 máquinas Linux:

* + Kubernetes Máster
  + Kubernetes Worker1
  + Kubernetes Worker2
  + NFS Server

Cada maquina con 1 CPU y 3,5 GB de memoria RAM, dimensionamiento que era compatible con la limitación a 4 CPU que tiene la cuenta Azure for Students.

Al desplegar el Máster de Kubernetes tuve problemas, ya que el playbook no daba error alguno pero no podía unir ningún worker al clúster.

Tras múltiples pruebas e investigaciones, ejecutando a mano la inicialización de kubeadm con la opción -v=5 obtuve el error:

**[ERROR NumCPU]: the number of available CPUs 1 is less than the required 2**

El máster de kubernetes necesita como poco 2 CPU, lo que me obliga a replantear la infraestructura a desplegar, infraestructura que estará finalmente formada por 2 máquinas Linux:

* + Kubernetes Máster y NFS Server
  + Kubernetes Worker1

Cada máquina con 2 CPU y 4 Gb de memoria RAM, dimensionamiento compatible con la limitación a 4 CPU que tiene la cuenta Azure for Students.

# ANEXO: PReparacion maquina local

Como he comentado, todo se lanzará desde una maquina local Linux (CentOS 8) de la cual tengo acceso root.

La preparación de dicha maquina consiste en los siguientes elementos:

1. Actualización de dicha maquina (conectado como usuario “root”)

*dnf update -y*

1. Crear un usuario “ansible” en las maquinas a gestionar (Este paso se omite para despliegue Azure, ya que ya existe un usuario “adminUsername”) (conectado como usuario “root”)

*useradd -md /home/ansible ansible*

*passwd ansible*

*usermod -aG wheel ansible    --> privilegios sudo a ansible*

*usermod -s /bin/bash ansible*

*sudo visudo --> Descomentar lineas %wheel ALL=(ALL) PASSWD:ALL*

1. Generar clave ssh para este usuario y distribuirlas a las maquinas a gestionar (Este paso se omite para despliegue Azure, ya que ya existe) (conectado como usuario creado en punto anterior, es decir, “ansible”)

*ssh-keygen*

*ssh-copy-id ansible@<ip maquina kubernetesMaster>*

*ssh-copy-id ansible@<ip maquina kubernetesNodo1>*

1. Instalar Ansible, Terraform y GIT en Ansible Controller (conectado como usuario “ansible”)

Para ello:

d1) Actualizar cache dnf

*sudo dnf makecache*

d2) Instalar Git

*sudo dnf install git -y*

d3) Activar EPEL repositorio

*sudo dnf install epel-release -y*

d4) Actualizar cache dnf

*sudo dnf makecache -y*

d5) Instalar Ansible

*sudo dnf install ansible -y*

d6) Comprobar instalación

*ansible --version*

d7) Añadir al host del Ansible Controler mapeo ip dominio para las maquinas a gestionar (Este paso se omite para despliegue en Azure)

*sudo vi /etc/hosts*

d8) Instalar wget

*sudo dnf install wget -y*

d9) Instalar terraform

*sudo wget* [*https://releases.hashicorp.com/terraform/0.14.6/terraform\_0.14.6\_linux\_amd64.zip*](https://releases.hasicorp.com/terraform/0.14.6/terraform_0.14.6_linux_amd64.zip)

*sudo unzip ./terraform\_0.14.6\_linux\_amd64.zip -d /usr/local/bin*

*terraform -version*

d10) Clonamos el repositorioGIT

    sudo git clone <https://github.com/asanton/Practica2Unir.git> ./repositorioGIT/Practica2UNIR

Tras estos pasos dispondremos de 2 carpetas:

* terraform

Donde están las plantillas terraform para desplegar la infraestructura

* ansible

Donde están los playbooks para realizar las configuraciones necesarias para montar el clúster de kubernetes y desplegar la aplicacion