

Entornos Cloud en DevOps

José Ángel de Bustos Pérez

Guía Caso Práctico 2

Introducción y Objetivos

- ▶ Los objetivos de este documento es facilitar una guía para realizar el Caso Práctico 2.
- ▶ No es de obligado cumplimiento el seguimiento de esta guía.
- ▶ Se dispone de material y ejemplos tutorizados en el repositorio de GitHub: <https://github.com/jadebustos/devopslabs/>
- ▶ Se irá actualizando el repositorio con ejemplos, con lo cual sincronízalo periódicamente.

¿Qué he de entregar al terminar la práctica?

- Se debe entregar un archivo comprimido incluyendo:
- Un plan de terraform que creará la infraestructura necesaria en Azure.
- Unos playbooks de ansible que desplegarán kubernetes en máquinas virtuales en Azure.
- Un informe en PDF.

Sugerencia

- Hay un saldo limitado en la cuenta de Azure.
- Para evitar que se consuma el saldo es recomendable realizar la automatización en un entorno local.
- Crear un par de máquinas virtuales, realizar la automatización del despliegue de kubernetes con ansible.
- Cuando este terminada la automatización crear un plan con terraform para crear la infraestructura en Azure.
- Despues de crear la infraestructura en Azure, crear un inventario que incluya los nodos creados en Azure y lanzar el despliegue con ansible en los nodos de Azure.

Sugerencia

- Si se quiere realizar el caso práctico directamente en azure, lo primero que se debe hacer es el plan de Terraform para crear la infraestructura.
- Para la automatización con ansible del despliegue de kubernetes proceder como se describe para el laboratorio.
- Para el controller de ansible en se puede utilizar una máquina en local o bien uno de los nodos desplegados en Azure.

Crear laboratorio en local

- Instala un par de máquinas Linux (CentOS 8 o CentOS 8 Stream).
- Si utilizas Linux puedes hacerlo con KVM. Si no con VMware Workstation (requiere pago de licencia) o VirtualBox.
- Si sabes utilizar Vagrant también puedes utilizarlo.
- En lo sucesivo supondremos que hemos creado las máquinas virtuales **master.local** y **worker.local**.
- En ambas máquinas instala el paquete chrony y activarlo para que se arranque en el inicio del sistema, actualizamos el sistema y lo reiniciamos para tener las últimas actualizaciones:

```
# dnf install chrony -y
# systemctl enable chronyd
# dnf update -y
# reboot
```

Crear laboratorio en local

- En ambas máquinas crear un usuario, por ejemplo llamado ansible que utilizaremos con ansible, le ponemos una contraseña:

```
# adduser -md /home/ansible ansible  
# passwd ansible
```

- Será necesario generar unaa claves públicas/privadas para SSH en **master.local** para el usuario ansible
(<https://github.com/jadebustos/devopslabs/blob/master/labs-ansible/00-primeros-pasos.md>)
- A continuación con ssh-copy-id deberás copiar la clave pública del usuario ansible a **master.local** y **worker.local** para el usuario ansible. De esta forma desde el usuario ansible en **master.local** podremos ejecutar tareas con el usuario ansible en ambas máquinas (<https://github.com/jadebustos/devopslabs/blob/master/labs-ansible/00-primeros-pasos.md>).

Crear laboratorio en local

- Ahora es necesario permitir el escalado de privilegios en ambos nodos para el usuario ansible (<https://github.com/jadebustos/devopslabs/blob/master/labs-ansible/00-primeros-pasos.md>).
- Prueba a ejecutar `sudo su` como usuario ansible, si puedes pasar a root sin facilitar el password el escalado de privilegios lo tendrás bien.
- Instala en ambos nodos el paquete python3 para permitir la ejecución de tareas con ansible:

```
# dnf install python3 -y
```

- En el nodo **master.local** instala los paquetes git y ansible:

```
# dnf install git ansible -y
```


Crear laboratorio en local

- Crea un repositorio en github, gitbucket, ... y clonalo en la máquina **master.local** como usuario ansible. Crea un directorio dentro del repositorio para los playbooks de ansible y dentro de ese directorio crea un inventario:

```
[all:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ansible
```

```
[master]
master.local
```

```
[workers]
worker.local
```

```
[nfs]
master.local
```

Crear laboratorio en local

- En el laboratorio vamos a desplegar solo un worker y el nfs lo configuraremos en el nodo máster.
- Ahora para comprobar si tenemos bien configurados los accesos para ansible desde el usuario ansible en el nodo **master.local** y desde el directorio donde se encuentre el fichero de inventario que hemos creado, suponiendo que este se llama **hosts**, ejecutaremos

```
$ ansible -i hosts -m ping all
master.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
worker.local | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
$
```

- Si obtenemos SUCCESS en ambos nodos estamos listos para empezar a automatizar el despliegue. Si no revisar la configuración de las claves públicas.

Automatizando el despliegue de kubernetes

- En <https://github.com/jadebustos/devopslabs/blob/master/labs-ansible/05-desplegando-una-aplicacion-balanceada.md> se describe como desplegar una aplicación balanceada con varios playbooks que se ejecutarán en diferentes grupos del inventario.
- En <https://github.com/jadebustos/devopslabs/blob/master/labs-k8s/00-00-instalando-kubernetes.md> se detalla el despliegue de kubernetes.
- Revisa el procedimiento y fijate en las tareas a realizar, ejecutar comandos, crear ficheros, instalar paquetes, crear reglas de firewall, ...
- Busca módulos de ansible para realizar estas tareas. En el documento que se facilitó en la clase presentando el caso práctico 2 se comentan módulos de ansible y tareas que hacen. Búscalos en la documentación de ansible (googlea ansible modulo copy y tendrás la url, asegúrate de que cargas la versión 2.9).

Automatizando el despliegue de kubernetes

- Siguiendo el procedimiento vemos que hay varios grupos de tareas a realizar. Se puede crear un playbook de ansible para cada grupo de tareas.
- Vete creando los playbooks en orden secuencial, cuando ejecutes el playbook de forma satisfactoria pasa al siguiente playbook.
- En el material publicado en el campus del tema 9 se encuentra publicado un vídeo "Uso de Ansible para la automatización de tareas".
- **Tareas previas de configuración**, estas tareas se han de realizar en todos los nodos. Crea un playbook para realizar estas tarea.

Automatizando el despliegue de kubernetes

- **Instalación del servidor NFS**, que se realizará en el servidor de NFS. En nuestro laboratorio será el nodo **master.local**. Crea un playbook para realizar estas tareas.
- En la instalación del servidor NFS no será necesario añadir un disco y empezaremos a partir de

```
# dnf install nfs-utils net-tools -y
```

- **Tareas comunes a realizar en el nodo master y los workers**, que se realizarán en los nodos **master.local** y **worker.local**. Crea un playbook para realizar estas tareas.
- **Configurando kubernetes en el nodo master**, que se realizará solo en el nodo **master.local**. Crea un playbook para realizar estas tareas.

Automatizando el despliegue de kubernetes

- **Instalación del servidor NFS**, que se realizará en el servidor de NFS. En nuestro laboratorio será el nodo **master.local**. Crea un playbook para realizar estas tareas.
- En la instalación del servidor NFS no será necesario añadir un disco y empezaremos a partir de

```
# dnf install nfs-utils net-tools -y
```

- **Tareas comunes a realizar en el nodo master y los workers**, que se realizarán en los nodos **master.local** y **worker.local**. Crea un playbook para realizar estas tareas.
- **Configurando kubernetes en el nodo master**, que se realizará solo en el nodo **master.local**. Crea un playbook para realizar estas tareas. Añade también a este playbook las tareas que aparecen en **Creamos un usuario administrador** que se encuentra al final del procedimiento.

Automatizando el despliegue de kubernetes

- **Instalando la SDN**, se realizará en **master.local**. Crea un playbook para realizar estas tareas.
- **Configurando los workers**, se realizará en **worker.local**. Crea un playbook para realizar estas tareas.
- **Desplegando un ingress controller**, se realizará en **master.local**. Crea un playbook para realizar estas tareas.

Creación de infraestructura en Azure

- Una vez automatizado el despliegue crea la infraestructura necesaria en Azure (consultar documentación de clase de refuerzo).

Desplegando kubernetes en Azure

- Para desplegar kubernetes en Azure lo primero que deberás hacer es modificar el inventario para que coincida con el entorno que has desplegado en Azure:

```
# Inventario utilizando 4 servidores de 1
# cpu en Azure
[all:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ansible
```

```
[master]
master.azure
```

```
[workers]
worker01.azure
Worker02.azure
```

```
[nfs]
nfs.azure
```

```
# Inventario utilizando 2 servidores de 2
# cpu1 en Azure
[all:vars]
ansible_python_interpreter=/usr/bin/python3
ansible_user=ansible
```

```
[master]
master.azure
```

```
[workers]
worker.azure
```

```
[nfs]
master.azure
```

Desplegando kubernetes en Azure

- Ahora será necesario revisar las variables que hayamos definido para cambiar aquellos valores que cambien del entorno local al entorno de Azure, como por ejemplo las ips de los servidores que se utilizan en el fichero `/etc/exports` del NFS.
- Lanzamos la ejecución de los playbooks en el mismo orden, secuencial, que en el laboratorio.

Desplegando una aplicación

- Para desplegar una aplicación será necesario desplegar varios objetos de kubernetes.
- Crear un namespace.
- Crear un servicio.
- Crear un ingress con su configmap.
- Mapear un volumen a un directorio, aunque no se utilice.
- En <https://github.com/jadebustos/devopslabs/blob/master/labs-k8s/04-desplegando-aplicaci%C3%B3n-volumenes.md> se puede encontrar un ejemplo.

Desplegando una aplicación

- Se puede probar a desplegar la siguiente aplicación: <https://hub.docker.com/r/alexwhen/docker-2048>
- Se puede probar a modificar esa imagen y desplegarla tal y como se muestra en <https://github.com/jadebustos/devopslabs/blob/master/labs-k8s/07-instalando-una-aplicacion-mas-compleja.md>
- En el caso de modificar la imagen será necesario crearla y subirla a un repositorio de imágenes tal y como se muestra en el ejemplo.
- Si no se modifica la imagen se puede desplegar directamente desde dockerhub: **docker.io/alexwhen/docker-2048:latest**

Generación de informe

Se deberá entregar un informe incluyendo:

- Diagramas mostrando los elementos desplegados y su role. En el caso del PaaS además se deberán indicar todos sus componentes (workers, masters, ...)
- Descripción del entorno de PaaS desplegado. Número de instancias, role de cada una, características de cada una (memoria, ram, ...)
- Se deberá describir todo el proceso de despliegue, tanto de los elementos de infraestructura (instancia CentOS, PaaS y la aplicación). Se hará referencia al código en el repositorio para explicar el despliegue. Esta descripción deberá ser a alto nivel y los detalles técnicos deberán estar en los comentarios del código.
- Describir el funcionamiento de la aplicación. Como se utilizaría para que un cliente pueda evaluarla.

Generación de informe

- Describir el funcionamiento de la aplicación. Como se utilizaría para que un cliente pueda evaluarla.
- Se deberán describir los problemas que se han encontrado, si los hubiera, como se han solucionado y referencias que se hayan utilizado para resolverlas (en formato APA). Si algún problema no se ha podido solucionar, se deberá analizar y proponer soluciones o siguientes pasos, el motivo por el que se cree que ha fallado, ...
- URL del repositorio donde se encuentra el código (hacerlo público si se desea después de la fecha límite de entrega de la actividad).

Generación de informe

Como ejemplo se puede crear un informe con los siguientes capítulos:

- Título
- Índice
- HLD (High Level Design) incluyendo un diagrama mostrando la infraestructura, los componentes (nfs, workers, masters, ..)
- Requerimientos, describiendo la infraestructura necesaria en Azure, como se crea con terraform. Especificaciones de las máquinas virtuales, SO, memoria, cpu, grupos de seguridad, puertos abiertos y para que se utilizan, como se realiza la autenticación de terraform en azure.
- Plataforma PaaS, que plataforma se despliega, sus componentes y función. Incluir una breve descripción de kubernetes e incluir el procedimiento de despliegue con los playbooks de ansible.

Generación de informe

- Problemas, describir problemas que se han encontrado y como se han solucionado.
- Recomendaciones, si se te ocurren recomendaciones argumentadas. Por ejemplo, utilizar una CPU en las máquinas puede traer como consecuencia problemas de rendimiento. Si se despliega todo en dos máquinas con dos CPUs se tendrán que compartir roles, entonces que recomendaciones harías para no compartir roles, ...



www.unir.net