

# Entornos Cloud en DevOps

José Ángel de Bustos Pérez

## Caso Práctico II

# Información importante

- ▶ Habrá dos clases de apoyo para explicar de forma práctica Azure y Terraform enfocado a la resolución del caso práctico.
- ▶ Habrá una clase de apoyo para resolver dudas del caso práctico.
- ▶ En el repositorio que hemos utilizado en clase se encontrará código y ejemplos explicativos que se pueden utilizar como base: <https://github.com/jadebustos/devopslabs>
- ▶ A los ejemplos que ya se encuentran y que hemos visto en clase se irán añadiendo más ejemplos.

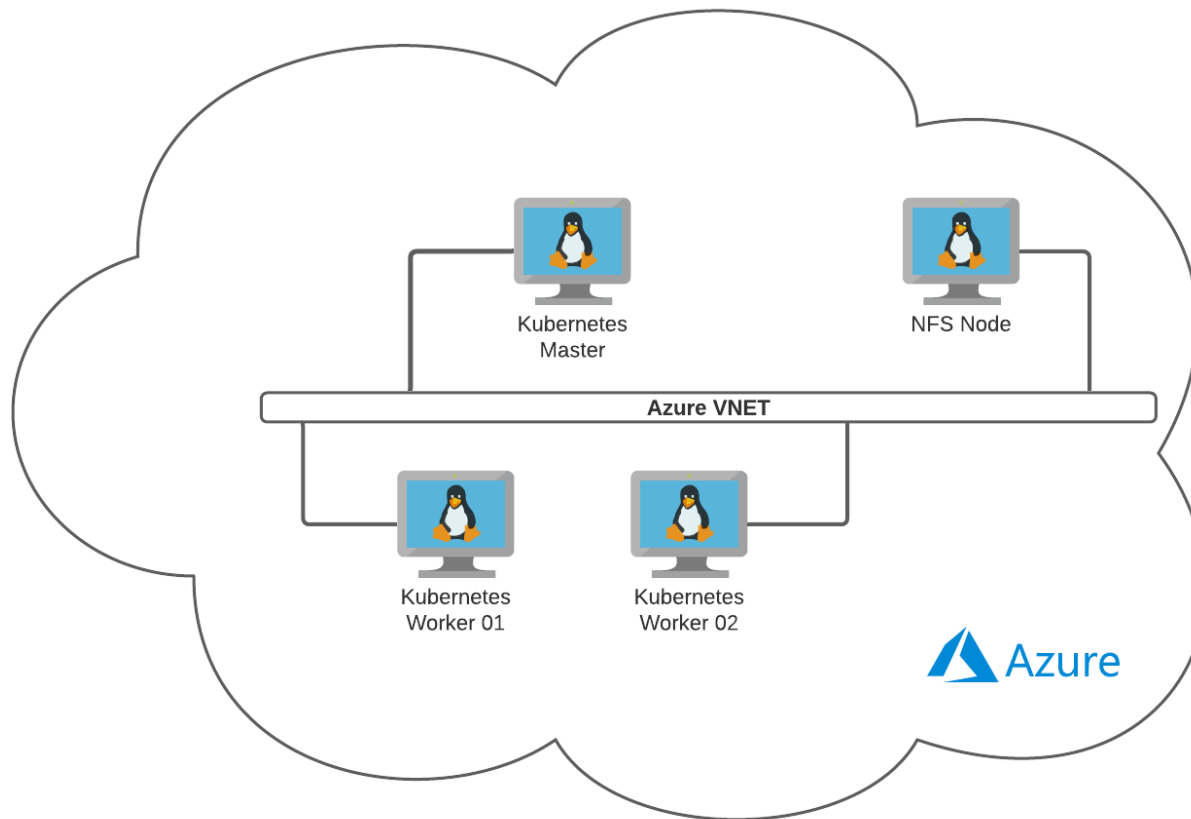
# Objetivos del Caso Práctico II

- Automatizar utilizando Ansible y Terraform.
- Desplegar Kubernetes sobre máquinas Linux.
- Se desplegará un Kubernetes Vanilla, no AKS.
- Desplegar una aplicación en Kubernetes con un volumen mapeado desde un servidor NFS desplegado.
- Generar un informe describiendo el despliegue que se ha realizado, con diagramas explicativos donde sea necesario.
- Detallar en el informe los problemas que se hallan detectado, como se han solucionado. Si no se han podido solucionar analizarlo, describir acciones realizadas, conclusiones y posibles soluciones.

# Objetivos del Caso Práctico II

- Se entregará **UNICAMENTE** un **PDF**, el informe generado, sin comprimir.
- El informe contendrá la URL del repositorio Git donde se encuentre la práctica, con el formato indicado.
- El repositorio deberá tener el formato indicado en <https://github.com/jadebustos/devopslabs/tree/master/estructura-practica> para facilitar su corrección.
- Una vez pasada la fecha de entrega el repositorio Git deberá ser accesible publicamente para acceder al código y corregirlo.
- El trabajo debe ser original del alumno, si se detecta copia de algún compañero, repositorio de código, ... automáticamente se calificará con un cero toda la práctica.

# Objetivos del Caso Práctico II



# Instalación de Kubernetes

- Se podrá utilizar la distribución de Linux que se considere oportuno.
- El proceso de instalación se encuentra descrito en <https://github.com/jadebustos/devopslabs/blob/master/labs-k8s/00-00-instalando-kubernetes.md>
- El procedimiento anterior es para CentOS 8. Para Debian/Ubuntu o otras distribuciones el procedimiento es similar adecuandolo a la distribución. El gestor de paquetes será diferente, nombres de paquetes pueden cambiar, ...
- Crear roles de ansible para las tareas, crear playbooks que utilicen los roles y ejecutarlos por grupos en el inventario.
- Se pueden crear grupos de inventario para NFS, Master y Worker.

# Instalación de Kubernetes

- Existen tareas que se realizan en todos los nodos, otras solo en el master y otras en los workers.
- El proceso de instalación se realizará ejecutando los playbooks de forma secuencial.
- Además de kubernetes será necesario instalar una SDN y el ingress controller (haproxy). El proceso de instalación se encuentra descrito también en <https://github.com/jadebustos/devopslabs/blob/master/labs-k8s/00-00-instalando-kubernetes.md>
- Si el alumno prefiere desplegar otra SDN o ingress controller en lugar de los anteriores podrá hacerlo.
- Se deberá crear un usuario no administrador en el nodo máster con acceso al clúster de kubernetes.

# Terraform y Azure

- Dedicaremos una clase práctica en exclusiva a este tema.
- Utilizaremos Azure como IaaS.
- Crearemos las máquinas virtuales y todos los recursos necesarios.
- Es recomendable trabajar sobre máquinas virtuales en local para automatizar el despliegue de Kubernetes para no consumir el saldo de la cuenta de Azure.
- Una vez terminada la automatización podemos proceder a desplegar la infraestructura en Azure y lanzar el despliegue.



# Ansible

- Los playbooks de ansible se deben ejecutar desde un nodo, nodo controller.
- Se pueden ejecutar desde un nodo externo a Azure, con lo cual será necesario abrir comunicación al puerto 22 a todos los nodos.
- Se puede reutilizar el nodo de NFS como controller de Ansible y ejecutar desde el los playbooks a todos los nodos. De esta forma sólo será necesario abrir el puerto 22 al nodo de NFS.
- Desde el controller de Ansible se deberá configurar el acceso con clave pública a un usuario no root y permitir a este usuario el escalado de privilegios sin password.
- En <https://github.com/jadebustos/devopslabs/tree/master/labs-ansible/> se encontrarán ejemplos de ansible.

# Ansible (módulos útiles)

- Los módulos **command** y **shell** se pueden utilizar para ejecutar comandos en el sistema operativo.
- Para instalar paquetes en linux se puede utilizar el módulo **dnf** en CentOS 8/RHEL 8, el módulo **yum** en CentOS 7/RHEL 7/SuSe y el módulo **apt** en Debian/Ubuntu.
- Para configurar los servicios del sistema (demonios) se puede utilizar el módulo **systemd**.
- Los módulos **lvvg** y **lvvol** se pueden utilizar para crear volume groups y logical volumes respectivamente.
- El módulo **filesystem** se puede utilizar para crear un filesystem.
- El módulo **mount** se puede utilizar para configurar sistemas de ficheros en /etc/fstab.

# Ansible (módulos útiles)

- El módulo **firewalld** se puede utilizar para configurar reglas de firewall.
- El módulo **lineinfile** se puede utilizar para añadir/eliminar líneas a ficheros.
- El módulo **file** se puede utilizar para gestionar ficheros y crear directorios.
- El módulo **copy** se puede utilizar para copiar ficheros desde el controller de ansible a los nodos remotos.
- El módulo **get\_url** se puede utilizar para descargar ficheros por http/https a los nodos remotos.
- El módulo **sysctl** se puede utilizar para la configuración de los parámetros del kernel.

# Ansible (módulos útiles)

- El módulo **useradd** se puede utilizar para crear usuarios.

# Recomendaciones

- La cuenta de Azure está limitada a unos 120 €.
- Por este motivo lo mejor es realizar la automatización de la instalación de kubernetes sobre máquinas virtuales en un entorno local.
- Mi sugerencia es crear 3 VMs en la misma red con 2 vCPU y 4 GBs de memoria.
- Una VM hará de servidor NFS y controller de Ansible.
- Otra VM hará de nodo máster de Kubernetes.
- Otra VM hará de nodo worker de Kubernetes.
- Si logramos desplegar un worker, el código podrá desplegar tantos como necesitemos.

# Recomendaciones

- Cuando tengamos las máquinas listas para empezar a desplegar es conveniente hacer un snapshot por si es necesario dar marcha atrás, tener de forma rápida una instalación lista para el despliegue.
- Si utilizas Linux con KVM puedes crear máquinas virtuales.
- Si utilizas otros sistemas operativos puedes utilizar [VirtualBox](#) (gratuito) o [VMware Workstation](#) (requiere pago de licencia).
- Utiliza un repositorio de código para matener el código fuente.
- Comenta el código, utiliza nombres de variables descriptivos.

# Informe

- Incluye diagramas HLD con los componentes, incluidos los elementos de Azure que se creen.
- Explica el role de cada componente.
- Incluye el dimensionamiento (CPU y memoria).
- Si te encuentras problemas técnicos descríbelos, argumenta como los has solucionado, si habría una solución que consideres mejor.
- Si no has podido solucionarlos, que te dice tu intuición, como crees que se podría solucionar.
- Describir la aplicación que se ha desplegado sobre Kubernetes, como se utilizaría (como conectarse, ...).



[www.unir.net](http://www.unir.net)