

UNIVERSIDAD INTERNACIONAL DE LA RIOJA  
EXPERTO UNIVERSITARIO EN DEVOPS & CLOUD

## **TRABAJO PRACTICO 2**

Automatización de despliegues en entornos  
Cloud

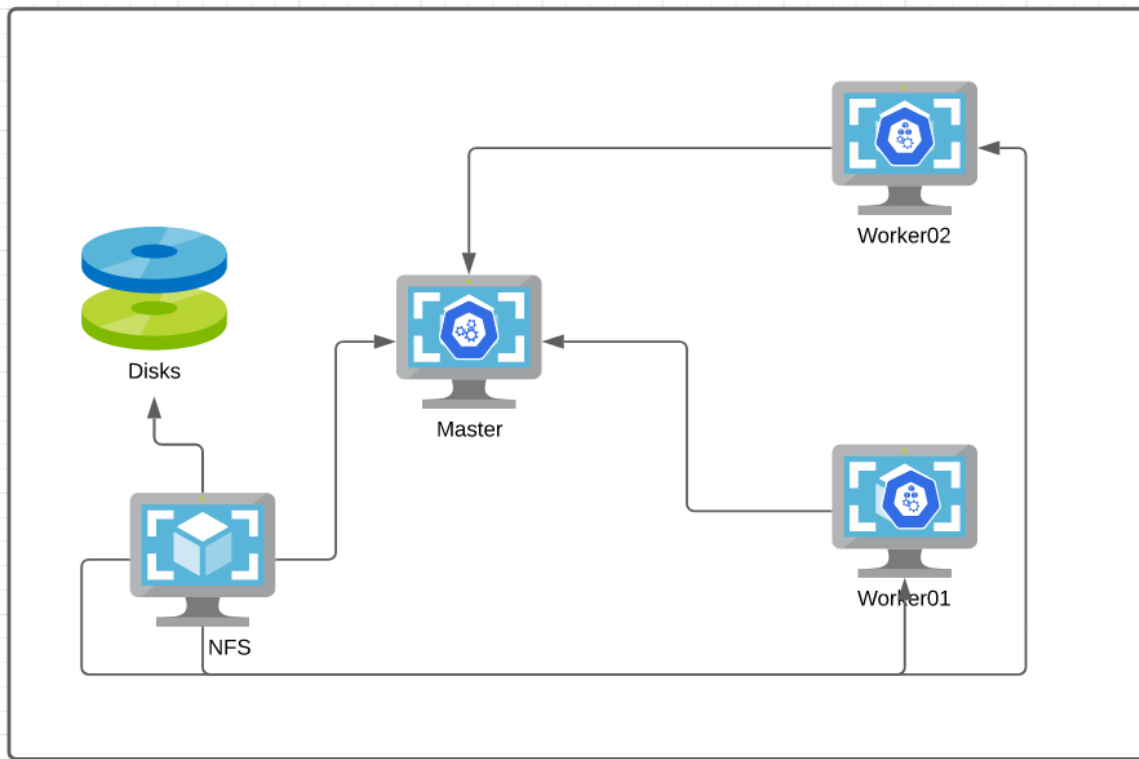
**ALUMNO:** ABEL ALEJANDRO NIEVA  
**FECHA:** 05 DE MARZO DE 2021

# Introducción

Se desarrolló una plataforma local y en la nube que permite desplegar aplicativos en Kubernetes. Para la misma de , se utilizó Ansible para la gestión de la configuración de equipos y despliegue de los aplicativos. Por otro lado el despliegue se realizo con Vagrant y Terraform , tanto local como en la nube de azure.

A continuación detallaremos los componentes mas relevantes.

Role	Sistema Operativo	vCPUs	Memoria (GiB)	Disco Duro	DNS	IP
NFS	CentOS 8.3	2	4	1 x 20 GiB (boot), 1 x 10 GiB (data)	<a href="#">nfs.acme.es</a>	192.168.2.115/24
Master	CentOS 8.3	2	8	1 x 20 GiB (boot)	<a href="#">master.acme.es</a>	192.168.2.110/24
Worker 01	CentOS 8.3	2	4	1 x 20 GiB (boot)	<a href="#">worker01.acme.es</a>	192.168.2.111/24
Worker02	CentOS 8.3	2	4	1 x 20 GiB (boot)	<a href="#">worker02.acme.es</a>	192.168.2.112/24
Bastion	CentOS 8.3	2	4	1 x 20 GiB (boot)	n/a	n/a



Además existe un bastión host en la nube, que permite la comunicación desde nuestra maquina con la infraestructura interna

## URL del repositorio de código

<https://github.com/abelnieva/playing-kubernetes-ansible>

## Estructura del Repo

- |— azcreds.json
- |— check\_req.sh
- |— files\_extra
- |— imgs

```
|   └─ infra.png
└─ loginazure.py
└─ Makefile
└─ playbooks
|   └─ ansible.sh
|   └─ apps.sh
|   └─ apps.yaml
|   └─ group_vars
|       └─ apps.yaml
|       └─ common.yaml
|       └─ general.yaml
|       └─ master.yaml
|       └─ nfs.yaml
|       └─ nodes.yaml
|   └─ hosts
|   └─ init.sh
|   └─ master.yaml
|   └─ nfs.yaml
|   └─ nodes.yaml
|   └─ roles
|       └─ common
|           └─ tasks
|               └─ 01-repositories.yaml
|               └─ 02-rpms.yaml
|               └─ 03-services.yaml
|               └─ 04-misc.yaml
|               └─ 05-docker.yaml
|               └─ 06-kubernetes.yaml
|               └─ main.yaml
|           └─ templates
```

```
| | | | — k8s.conf
| | | | — kube.repo
| | | | — resolv.j2
| | | — helm
| | | | — tasks
| | | | | — 01-install.yaml
| | | | | — main.yaml
| | | | — templates
| | | — master
| | | | — tasks
| | | | | — 01-firewall.yaml
| | | | | — 02-kubernetes.yaml
| | | | | — 03-user.yaml
| | | | | — main.yaml
| | | | — templates
| | | | — custom-resources.j2
| | | | — sudoers.j2
| | | — nfs
| | | | — tasks
| | | | | — 01-rpms.yaml
| | | | | — 02-service.yaml
| | | | | — 03-volumesetup.yaml
| | | | | — 04-nfs.yaml
| | | | | — 05-firewall.yaml
| | | | | — main.yaml
| | | | — templates
| | | | — export.j2
| | | — nodes
| | | — tasks
| | | | — 01-firewall.yaml
```

```
| | | |— 02-kubernetes.yaml
| | | |— main.yaml
| | |— templates
| | |— kubeadm_node_join.yml.j2
| |— worker.yaml
|— README.md
|— terraform
| |— apps.tf
| |— bastion.tf
| |— disks.tf
| |— files
| | |— key
| | |— key.pub
| |— local.tfe
| |— master.tf
| |— network.tf
| |— nfs.tf
| |— providers.tf
| |— variables.tf
| |— worker01.tf
| |— worker02.tf
|— Vagrantfile
```

Nota en el encabezado de los archivos encontraras comentarios que te permitirán entender como funcionan .

# Gestión de DNS

El manejo de los nombres de dominio para el cluster de kubernetes es manejada de manera ad-hoc describiendo los hosts en cada uno de los /etc/hosts de las máquinas.

## Requerimientos de software

### Para la versión local

- Virtual Box
- Vagrant
- Ansible
- python

### Para la versión en azure

- Ansible
- terraform
- azure cli
- python
- Cuenta de Azure

Para mayor información dirigirse al archivo README del proyecto

# Ejecución de automatizaciones

Existe un Makefile que describe muchos comandos que ayudan a ejecutar las tareas mas comunes desde ahí.

En el README del repositorio podrás encontrar más detalles como utilizarlo.

## Desplegando localmente

Es necesario satisfacer todos los requerimientos de software y hardware previos

No se ha desarrollado automatizaciones para instalar localmente los requerimientos de software, por lo que deben ser instalado por parte del usuario.

Todas las automatizaciones para entorno local comienzan con local , por ejemplo “make local up”, esta automatización desplegara las maquinas y el software .

En el README del proyecto podrán encontrar más detalles al respecto.



# Desplegando en la nube.

Al día de hoy solo es posible desplegar en Azure, el desarrollo de código para otros proveedores de Cloud queda en el TODO. El despliegue en la nube se realizará gracias a los archivos de terraform incluidos en el repo.

## Consideraciones para el despliegue en la nube

- Debido a que el despliegue se hace en una red remota a nuestro ordenador, debemos desplegar un Bastion.
- Debemos crear una cuenta de azure y una cuenta de tipo service principal para acceder desde nuestro código. En el README del proyecto encontrará más detalles al respecto
- Si usamos la automatización “make azuresetupCreds“, exportara nuestras credenciales como variables de entorno

# Desplegando Apps en Kuberntes.

El repositorio de código supone que disponemos de dos entornos para desplegar nuestras apps, cada entorno es un namespace en kubernetes para su completo aislamiento lógico. Toda declaración de aplicaciones a instalarse en kubernetes debemos hacerla en el ansible rol llamado helm.

Para hacer efectivo el cambio se debe hacer uso de la automatización **make appsReDeploy**, para mas información al respecto revisar README del proyecto

# Limitaciones y trabajo a futuro

## Local

- Las maquinas virtuales master requieren al menos 2 VCPU , Por lo que se debe tener cuidado al modificar las características de las maquinas.

## Azure

- La suscripción student de azure solo permite ejecutar hasta 6 VCPU, por lo que no se pudo cumplir con los requerimientos de HW que disponemos en local. La configuración en azure fue el master con 2 vCPU y el resto de nodos con un 1vCPU
- Los discos root de las máquinas deben ser de al menos 30 GB, valores inferiores no estan permitidos
- Los archivos de estado de terraform se almacenan localmente. En un futuro podrian ponerse en algun lugar con acceso desde internet para poder ser trabajado colaborativamente.

## Apps

- El actual código no permite la eliminación de aplicaciones desplegadas. Mejorar esta mecánica es parte del TODO. Quizas podriamos utilizar herramientas como ARGO cd y desacoplar la parte de apps a otro repo .
- No existen pruebas al desplegar las apps

## NFS

- El espacio en el server de NFS no se agrega a kubernetes como PV

## Misc

- El despliegue local y en azure no disponen de la misma fuente de configuración de infraestructura. En un futuro se debería combinarla