|  |  |
| --- | --- |
|  | |
| Experto Universitario en DevOps & Cloud | |
| Caso práctico 2: Automatización de despliegues en entornos Cloud | |
| Alumno | Gabriel Guarino |
| Fecha | 23/07/2021 |

Tabla de contenido

[Objetivo 3](#_Toc77926019)

[GITHUB Repository 3](#_Toc77926020)

[GNU General Public License v2.0 3](#_Toc77926021)

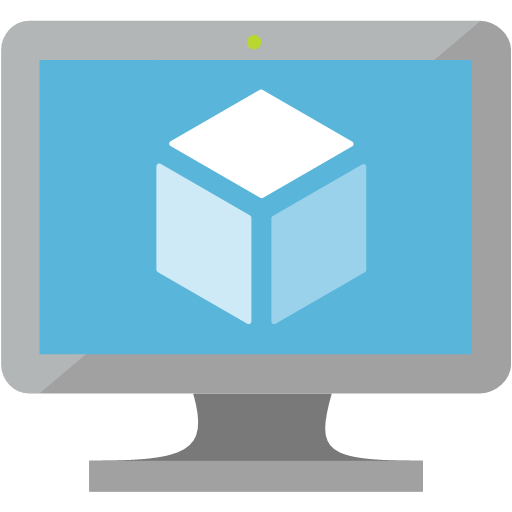
[Diagramas 4](#_Toc77926022)

[Diagrama Logico 4](#_Toc77926023)

[Diagrama CLOUD 5](#_Toc77926024)

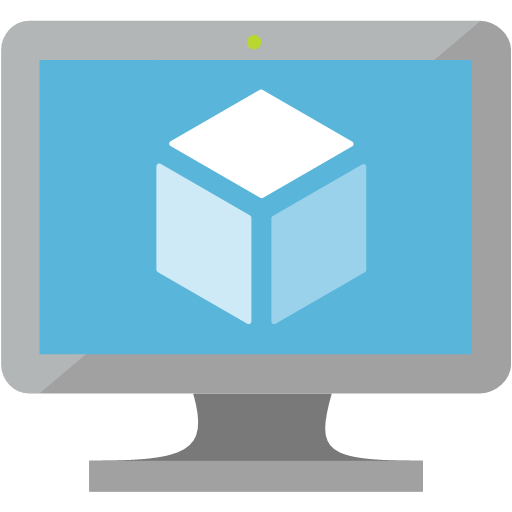
[Detalles Nodos 6](#_Toc77926025)

[Master 6](#_Toc77926026)

[ Settings 6](#_Toc77926027)

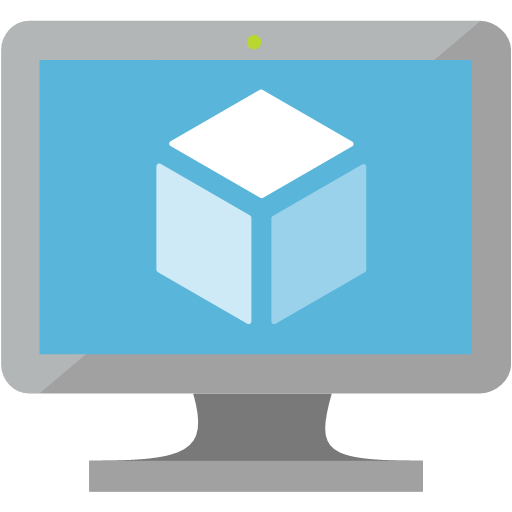
[ Virtual Disks 6](#_Toc77926028)

[NFS 7](#_Toc77926029)

[ Settings 7](#_Toc77926030)

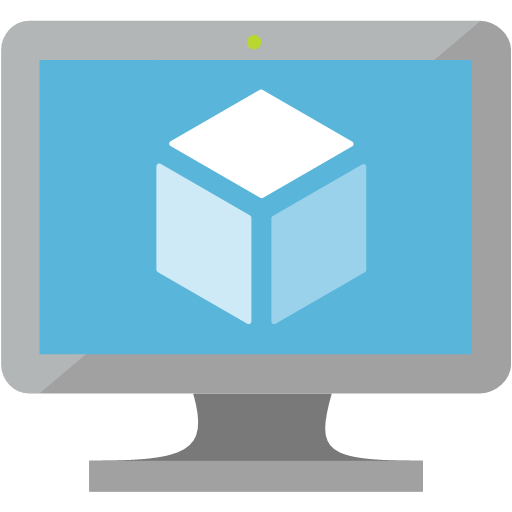
[ Virtual Disks 7](#_Toc77926031)

[Worker01 7](#_Toc77926032)

[ Settings 7](#_Toc77926033)

[ Virtual Disks 8](#_Toc77926034)

[Worker02 8](#_Toc77926035)

[ Settings 8](#_Toc77926036)

[ Virtual Disks 9](#_Toc77926037)

[Proceso De Despliegue 9](#_Toc77926038)

[Aplicación Propuesta 10](#_Toc77926039)

# Objetivo

El siguiente documento tiene como objetivo la definición y explicación de todo lo realizado en el Caso Practico 2 del curso experto en DevOps de la UNIR:

Para ello se requiere montar una infraestructura en el cloud de Azure mediante el uso de Terraform, y sucesivamente desplegar un entorno de K8s a través de Ansible.

Por lo cual tanto el despliegue de la infraestructura, como el despliegue de Kubernetes se han automatizado todo lo posible.

Gracias a la automatización de todo el proceso la infraestructura es fácilmente escalable de manera horizontal.

# GITHUB Repository

El código empleado para el caso practico se puede consultar en el siguiente repositorio:

<https://github.com/GabGua/CP2.git>

## GNU General Public License v2.0

La licencia empleada para nuestro repositorio de GitHub es la GNU General Public License V2.0

La Licencia Pública General GNU (comúnmente conocida como GNU GPL o simplemente GPL) es una licencia fuertemente copyleft para software libre, originalmente emitida en 1989 por Richard Stallman para patrocinar programas creados para el sistema operativo GNU.

De hecho, a diferencia de otras licencias gratuitas sin copyleft, una obra protegida por la GNU GPL debe permanecer libre, es decir, con los siguientes cambios debe seguir garantizando a sus usuarios las llamadas cuatro libertades.

**Versión 2**

GNU GPLv2 se publica en 1991. Según Richard Stallman, la modificación más importante de esta versión es la cláusula que llamó "Libertad o muerte".

Esta cláusula requiere que, si existen restricciones de cualquier tipo sobre la distribución gratuita del software bajo los términos indicados en la licencia, el software no podrá distribuirse en absoluto.

Por ejemplo, si una ley le impide distribuir código fuente, no se podrá distribuir todo el software protegido por GNU GPLv2.

La esperanza es hacer menos atractivo para las empresas recurrir a amenazas de patentes o exigir pagos a los desarrolladores de software libre.

# Diagramas

## Diagrama Logico

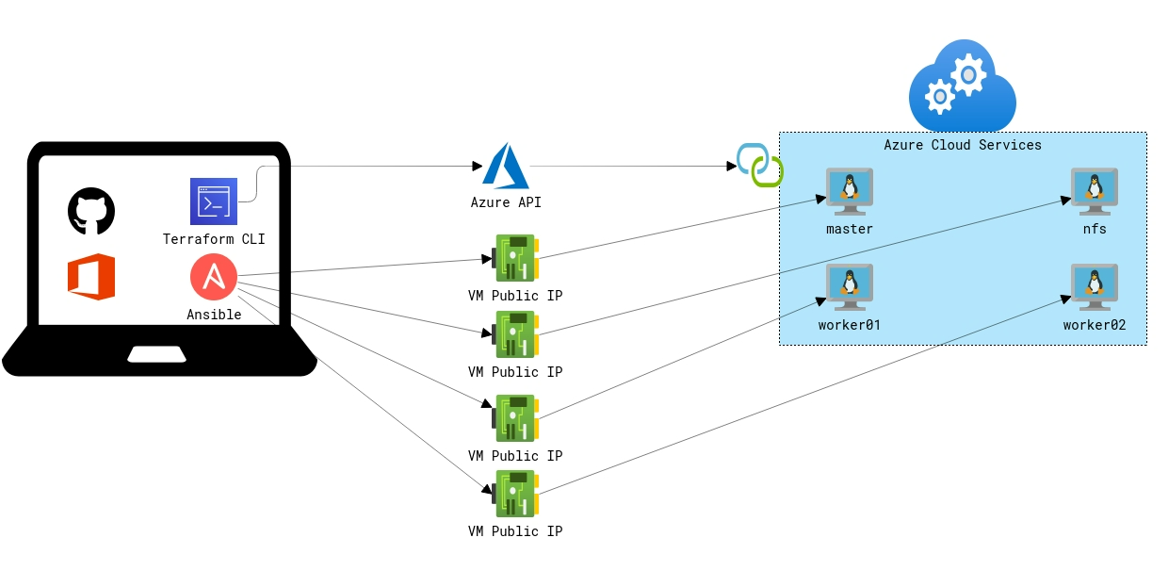
El proyecto se compone de:

* 1 VM con rol master de Kubernetes
* 2 VMs con rol worker de Kubernetes
* 1 VM con rol de server NFS

Las instancias cloud se desplegarán de forma automática utilizando Terraform, a través de la API de Azure.

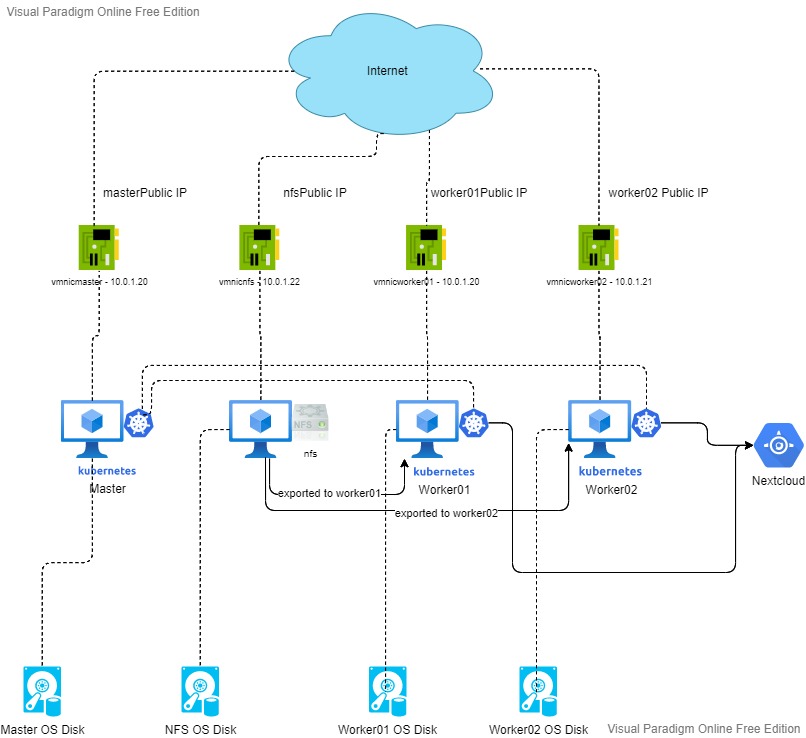
Una vez desplegada las instancias se lanzarán varios playbook de Ansible desde nuestra equipo local, que instalaran y configuraran un clúster Kubernetes en dichas instancias.

La conexión entre Ansible y las VMs se realiza tramite IPs públicas y una clave SSH privada, guardada localmente.



## Diagrama CLOUD

El siguiente diagrama muestra la infraestructura implementada sobre la nube publica de Azure utilizando Terraform y Ansible.



# Detalles Nodos

A continuación de detallan todas las características de todas las VMs.

## Master

### Settings

La máquina virtual tiene la siguiente configuración:

|  |  |
| --- | --- |
| Computer Name | master |
| Operating System | Linux (centos 8) |
| Publisher | cognosys |
| Plan | centos-8-stream-free |
| Location | westeurope |
| Size | Standard\_B2s   * Number of Cores : 2 * Memory (MB): 4096 * Max Data Disk Count: 4 |
| Network Interface | vmnicmaster |
| Private IP address | 10.0.1.10 |
| Public IP address | \*\*\*\*\*\*\*\* |
| State | VM running |
| Provisioning Date | 07/21/2021 06:49:09 |
| Resource Group | kubernetes\_rg |
| Tags | CP2 |
| Instance Type | Standard\_B2s |

### 

### Virtual Disks

La máquina virtual está usando los siguientes discos:

#### OS Hard Disks

##### Settings

|  |  |
| --- | --- |
| Managed Disk | master\_OsDisk\_1 |
| Disk state |  |
| Time created |  |
| Size (GB) | 30GB |
| Encryption | Not enabled |
| Host caching | ReadWrite |
| Operating system | Linux |
| Source image |  |
| Availability zone |  |

## NFS

### Settings

La máquina virtual tiene la siguiente configuración:

|  |  |
| --- | --- |
| Computer Name | nfs |
| Operating System | Linux (centos 8) |
| Publisher | cognosys |
| Plan | centos-8-stream-free |
| Location | westeurope |
| Size | Standard\_B2s   * Number of Cores : 1 * Memory (MB): 3584 * Max Data Disk Count: 4 |
| Network Interface | vmnicnfs |
| Private IP address | 10.0.1.12 |
| Public IP address | \*\*\*\*\*\*\*\* |
| State | VM running |
| Provisioning Date | 07/21/2021 06:49:09 |
| Resource Group | kubernetes\_rg |
| Tags | CP2 |
| Instance Type | Standard\_DS1\_v2 |

### Virtual Disks

La máquina virtual está usando los siguientes discos:

#### OS Hard Disks

##### Settings

|  |  |
| --- | --- |
| Managed Disk | nfs\_OsDisk\_1 |
| Disk state |  |
| Time created |  |
| Size (GB) | 30GB |
| Encryption | Not enabled |
| Host caching | ReadWrite |
| Operating system | Linux |
| Source image |  |
| Availability zone |  |

## Worker01

### Settings

La máquina virtual tiene la siguiente configuración:

|  |  |
| --- | --- |
| Computer Name | worker01 |
| Operating System | Linux (centos 8) |
| Publisher | cognosys |
| Plan | centos-8-stream-free |
| Location | westeurope |
| Size | Standard\_B2s   * Number of Cores : 1 * Memory (MB): 3584 * Max Data Disk Count: 4 |
| Network Interface | vmnicworker01 |
| Private IP address | 10.0.1.10 |
| Public IP address | \*\*\*\*\*\*\*\* |
| State | VM running |
| Provisioning Date | 07/21/2021 06:49:09 |
| Resource Group | kubernetes\_rg |
| Tags | CP2 |
| Instance Type | Standard\_DS1\_v2 |

### Virtual Disks

La máquina virtual está usando los siguientes discos:

#### OS Hard Disks

##### Settings

|  |  |
| --- | --- |
| Managed Disk | worker01\_OsDisk\_1 |
| Disk state |  |
| Time created |  |
| Size (GB) | 30GB |
| Encryption | Not enabled |
| Host caching | ReadWrite |
| Operating system | Linux |
| Source image |  |
| Availability zone |  |

## Worker02

### Settings

La máquina virtual tiene la siguiente configuración:

|  |  |
| --- | --- |
| Computer Name | worker02 |
| Operating System | Linux (centos 8) |
| Publisher | cognosys |
| Plan | centos-8-stream-free |
| Location | westeurope |
| Size | Standard\_B2s   * Number of Cores : 1 * Memory (MB): 3584 * Max Data Disk Count: 4 |
| Network Interface | vmnicworker02 |
| Private IP address | 10.0.1.12 |
| Public IP address | \*\*\*\*\*\*\*\* |
| State | VM running |
| Provisioning Date | 07/21/2021 06:49:09 |
| Resource Group | kubernetes\_rg |
| Tags | CP2 |
| Instance Type | Standard\_DS1\_v2 |

### Virtual Disks

La máquina virtual está usando los siguientes discos:

#### OS Hard Disks

##### Settings

|  |  |
| --- | --- |
| Managed Disk | worker02\_OsDisk\_1 |
| Disk state |  |
| Time created |  |
| Size (GB) | 30GB |
| Encryption | Not enabled |
| Host caching | ReadWrite |
| Operating system | Linux |
| Source image |  |
| Availability zone |  |

# Proceso De Despliegue

Lo primero que hay que realizar es descargar/clonar [el siguiente repositorio de GitHub](https://github.com/GabGua/CP2.git), donde queda almacenado todo el código.

Luego hay que acceder en el directorio de **Terraform**, definir el service principal “azurerm” y rellenar los datos correspondientes para la autenticación con Azure

Tras rellenar toda la información ejecutando un “Terraform apply” se realizará el despliegue.

La infraestructura será compuesta por 4 máquinas virtuales de 2 diferentes tipos:

* **Standard\_B2s**
* **Standard\_DS1\_v2**

Las instancias del Master Nodes será de tipo “Standard\_B2s”, mientas que los Worker Nodes y el servidor NFS serán de tipo Standard\_DS1\_v2.

Dicha elección es necesaria, puesto que debido a las cuotas de la cuenta “**Azure for students”,** no es posible aprovisionar ciertos tipos de instancias.

El plan de Terraform también aprovisionará tarjetas de red, IPs públicas y privadas, políticas de seguridad, etc.

Una vez se haya desplegado la infraestructura de VMs en Azure es necesario moverse a la carpeta “ansible y:

* actualizar el archivo de inventario “hosts” con las IPs públicas de cada VM.
* Ejecutar el script “deploy.sh”

Ansible lanzará un tipo de playbook dependiendo del nodo, es decir si es el nodo máster Ansible lanzará una jugada específica para ese nodo, mientras que para los Worker y NFS lanzará otro Playbook.

# Aplicación Propuesta



**Nextcloud** es una aplicación que se compone de una serie de programas cliente-servidor con el objetivo de crear servicio de alojamiento de archivos.

Su funcionalidad podríamos decir es similar al software Dropbox, con la gran diferencia que a diferencia de Dropbox Nextcloud es de tipo opensource, permitiendo a quien lo desee instalarlo en un servidor privado.

Su arquitectura abierta permite añadir funcionalidad al servidor en forma de aplicaciones.

Nextcloud dispone de una gran cantidad de aplicaciones complementarias, catalogadas en su propia Nextcloud App Store, que nos permiten añadir todo tipo de funciones extra, y usarlo como sustituto de muchos de los servicios online que usamos en nuestro día a día.

Así, su ecosistema incluye blogs, mapas, reproductores/organizadores de música, webmail, sincronizadores de SMS, suites ofimáticas a lo Google Docs, editores de texto Markdown, videochats, gestores de tareas, notas o contraseñas, lectores de RSS...

Gracias a un playbook de Ansible hemos automatizado el despliegue de esta aplicación sobre el clúster de Kubernetes en la nube de Azure.

Lamentablemente el despliegue de este aplicativo aún resulta incompleto, debido a falta de tiempo por poder completar el caso práctico.