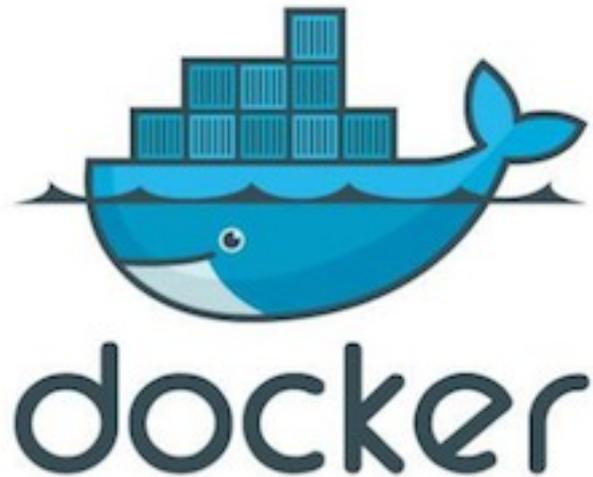


# Orquestación Docker Con Nomad Hashicorp



+



**Centro Docente:** IES Gonzalo Nazareno

**Autor/a:** Kevin Ariza García

**Curso:** 2º ASIR

**Módulo:** Proyecto y Formación en Centros de Trabajo (FCT)



# Índice De La Presentación

- ¿Qué Es Nomad Hashicorp?
- ¿Cómo Funciona Nomad Hashicorp?
- Características De Nomad Hashicorp.
- Ventajas E Inconvenientes De Uso De Nomad Hashicorp.
- Nomad Hashicorp VS Kubernetes y Docker Swarm (Comparativa y Conclusión).
- Descripción y Creación Del Escenario A Montar De Nomad Hashicorp.
- Iniciar Agente De Nomad Hashicorp.
- Crear Job Con Nomad Hashicorp.
- Modificar Job Creado Con Nomad Hashicorp.
- Crear Un Clúster Con Nomad Hashicorp.
- Consola Web-UI De Nomad Hashicorp.
- Conclusión De Nomad Hashicorp.



# ¿Qué Es Nomad Hashicorp?

- **Orquestador de contenedores** de aplicaciones, donde podremos desplegar una o varias aplicaciones en uno o varios contenedores, de una forma automatizada y configurada.
- **Nomad Hashicorp** es fácil de utilizar y flexible en la orquestación de contenedores.
- Capaz de ser implementada, en una combinación de **microservicios, lotes y aplicaciones** que estén o no, dentro de contenedores de aplicaciones.



# ¿Cómo Funciona Nomad Hashicorp?

- Administra un grupo de máquinas, para poder ejecutar aplicaciones en cada uno de los equipos.
- Se encarga de controlar la ubicación de ejecución de la aplicación, y la forma en la que se va ejecutar dicha aplicación.
- Gestiona y supervisa los despliegues de contenedores de una orquestación (**Equipos Servidores**).
- Gestionar el alojamiento de los despliegues realizados en una orquestación de contenedores (**Equipos Clientes**).



# Características De Nomad Hashicorp

- Soporte de contenedores **Docker**, como tipo de carga de trabajo de primera clase.
- Realizar **operaciones simples**, sin necesidad de requerir servicios externos, para la coordinación o almacenamiento de contenedores.
- Soporte **Multi-Datacenter y Multi-Region**, para poder modelar la infraestructura de un clúster de contenedores, para migrar las cargas de trabajo de forma incremental, y utilizar la nube para el despliegue.
- **Cargas de trabajo flexibles**, para poder ejecutar aplicaciones en contenedores, VM y de forma independiente.
- Diseñado para poder **escalar** miles de nodos en un sólo clúster, en una infraestructura de escala global.



# Ventajas De Uso De Nomad Hashicorp

- Capacidad de poder conectar, asegurar y configurar servicios, en cualquier plataforma de tiempo de ejecución, en nube pública o privada.
- Capacidad de asegurar, almacenar y controlar el acceso de tokens, contraseñas, certificados, claves cifradas y otros datos confidenciales, mediante una interfaz de usuario, **CLI** o **API HTTP**.



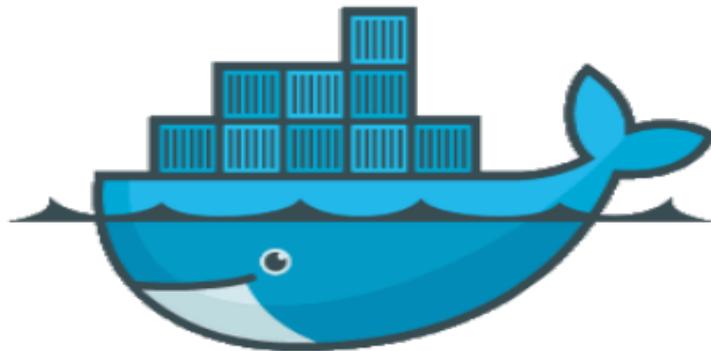
# Inconvenientes De Uso De Nomad Hashicorp

- No está capacitado para poder funcionar como **balanceador de carga**, para las aplicaciones desplegadas en un clúster de contenedores.
- No está capacitado para poder funcionar como **proxy**, para poder realizar la redirección de peticiones, en una aplicación desplegada en un clúster de contenedores.



# Nomad Hashicorp VS Kubernetes

- **Kubernetes** es un sistema de orquestación de contenedores, diseñado originalmente por la empresa “**Google**”, donde proporciona la ejecución de aplicaciones con **Docker** o **Rkt** entre otros.



# Nomad Hashicorp VS Kubernetes

- **Kubernetes** solo se centra en la orquestación de contenedores **Docker**. **Nomad Hashicorp**, tiene un propósito más general.
- **Kubernetes** está diseñado para poder proporcionar una funcionalidad compleja, para las aplicaciones desplegadas de más de media docena de servicios.
- **Kubernetes** admite la ejecución en una configuración compleja y de alta disponibilidad. **Nomad Hashicorp** es más simple, siendo de binario único tanto para Servidores como para Clientes.
- **Kubernetes** admite clústeres de contenedores de más de **5000 nodos**, mientras que **Nomad Hashicorp**, admite clústeres de contenedores de más de **10000 nodos**.



# Nomad Hashicorp VS Kubernetes (Conclusión)

- **Kubernetes** es una opción más ideal que **Nomad Hashicorp**, debido a que presenta tener diferentes herramientas de despliegue de aplicaciones y orquestación de contenedores, para poder llevar una mejor gestión de dichas aplicaciones desplegadas en contenedores.



Nomad (HashiCorp)

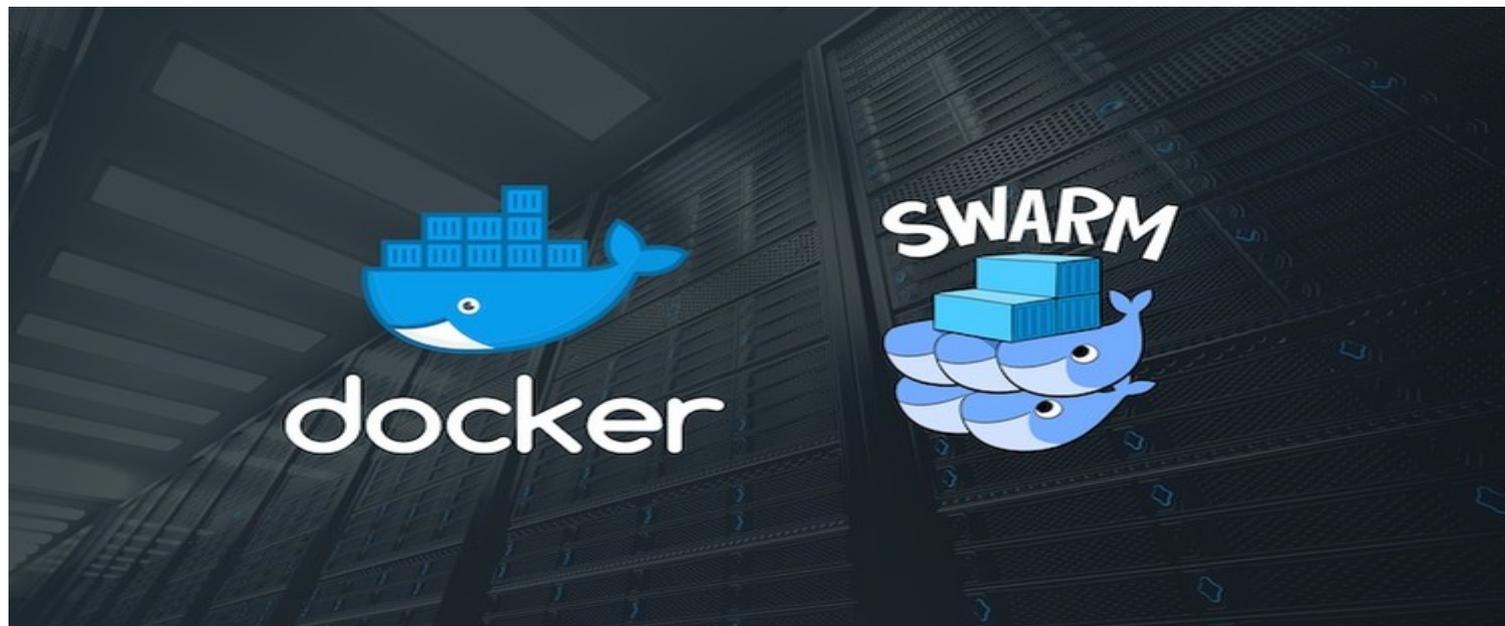


Kubernetes



# Nomad Hashicorp VS Docker Swarm

- **Docker Swarm** es la solución nativa de clúster para contenedores **Docker**, proporcionando una **API** compatible con la **API** remota de **Docker**, permitiendo que los contenedores sean programados en varias máquinas.



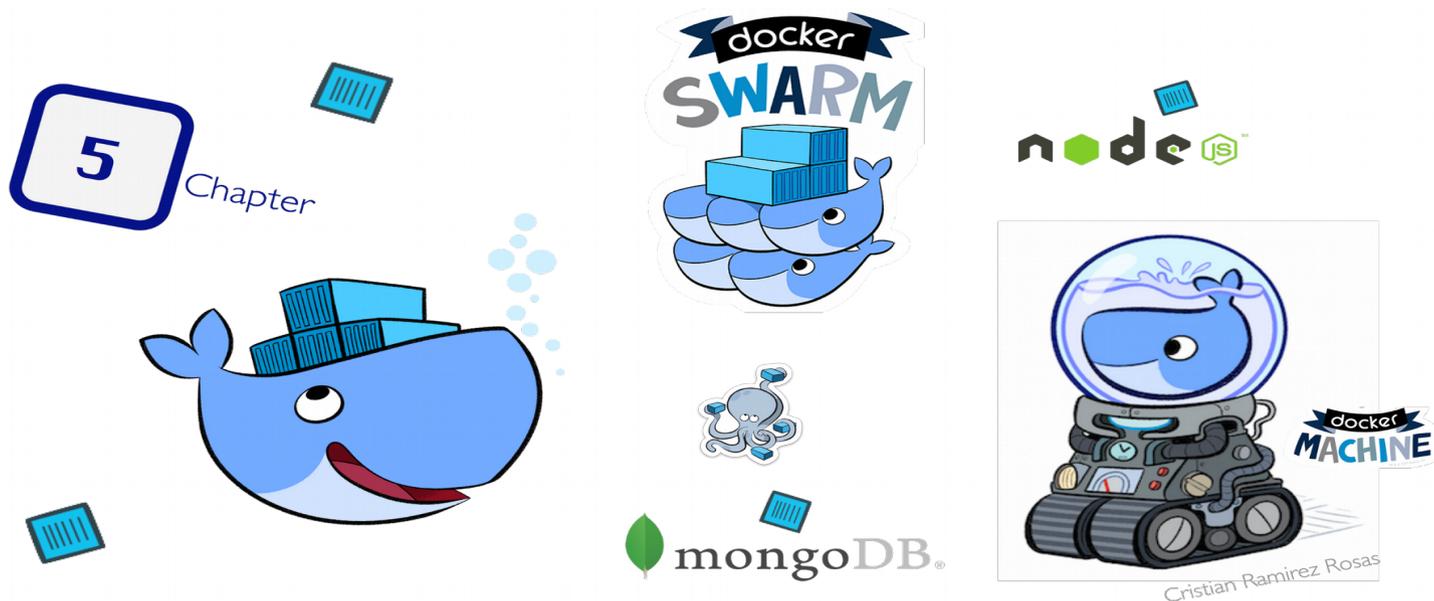
# Nomad Hashicorp VS Docker Swarm

- **Docker Swarm** solo se utiliza para la ejecución de contenedores **Docker**, mientras que **Nomad Hashicorp** realiza un uso más general.
- **Docker Swarm** proporciona compatibilidad de **API** con su **API** remota, centrándose en la abstracción de los contenedores **Docker**.
- **Nomad Hashicorp**, utiliza una abstracción de trabajos de alto nivel, permitiendo que las aplicaciones más complejas, se expresen y administren fácilmente.
- Las arquitecturas de **Nomad Hashicorp** y **Docker Swarm** difieren entre sí.

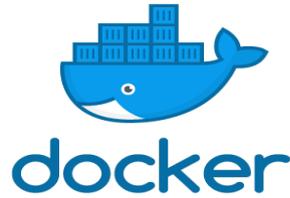


# Nomad Hashicorp VS Docker Swarm (Conclusión)

- **Nomad Hashicorp** es mejor opción para realizar orquestación, con un propósito más general. Pero si quieres orquestar sólo contenedores **Docker**, veo mejor opción **Docker Swarm**.



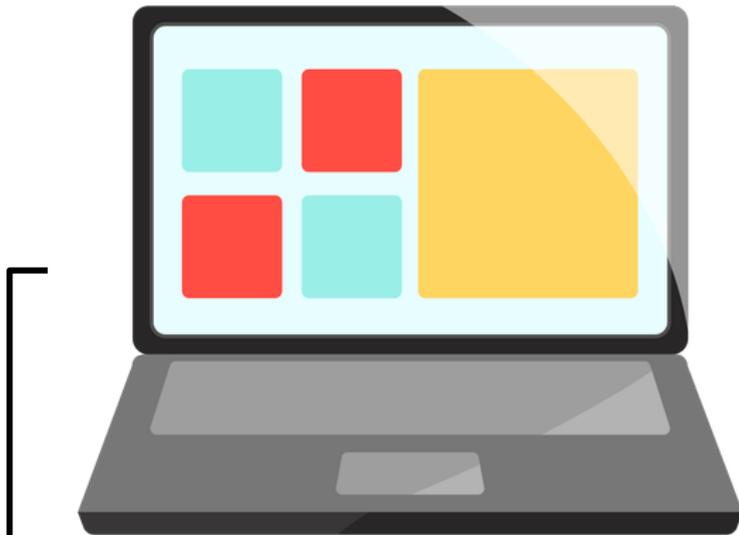
# Descripción Escenario A Montar



Eth0 (Red Nat) → 10.0.2.15



Ubuntu Bionic 18.04



Eth1 (Sólo Anfitrión) → 10.0.200.50



1 GB De RAM



# Creación Del Escenario Nomad Hashicorp

- Como usuario normal, creamos una carpeta (**en mi caso Nomad-Project**), y accedemos a dicha carpeta creada:

```
mkdir Nomad-Project && cd Nomad-Project
```

- Situados dentro del directorio creado, generamos un fichero **Vagrantfile**:

```
vagrant init
```



# Creación Del Escenario Nomad Hashicorp

- Editamos el fichero **Vagrantfile**, donde en mi caso, el escenario que he montado, ha sido el siguiente (**con la diferencia que sólo crearé un equipo servidor virtual**):  
<https://raw.githubusercontent.com/hashicorp/nomad/master/demo/vagrant/Vagrantfile>
- Cuando hayamos editado el fichero de **Vagrantfile**, procedemos a la creación del escenario definido en el fichero de configuración:

```
vagrant up
```



# Creación Del Escenario Nomad Hashicorp

- Accedemos al equipo servidor virtual, que hemos creado al levantar el escenario donde vamos a trabajar:

```
vagrant ssh
```

- Comprobamos que **Docker y Nomad Hashicorp** se han instalado correctamente en el equipo servidor virtual:

## Nomad Hashicorp:

```
vagrant@nomad-server:~$ nomad --version  
Nomad v0.9.1 (4b2bdbd9ab68a27b10c2ee781cceaaf62e114399)  
vagrant@nomad-server:~$
```

## Docker:

```
vagrant@nomad-server:~$ docker --version  
Docker version 18.09.6, build 481bc77  
vagrant@nomad-server:~$
```



# Iniciar Agente De Nomad Hashicorp

- Para iniciar el agente de **Nomad Hashicorp**, ejecutamos el siguiente comando:

```
sudo nomad agent -dev
```

- Si queremos ver el estado del agente del nodo de **Nomad Hashicorp**, lo haremos con la siguiente instrucción:

```
nomad node status
```

- Para saber los miembros que hay en el nodo de **Nomad Hashicorp** en ejecución, vamos a hacerlo con el siguiente comando:

```
nomad server members
```



# Crear Job Con Nomad Hashicorp

- Creamos fichero de configuración del **Job** a ejecutar, en el nodo de **Nomad Hashicorp**:

```
nomad job init
```

- Iniciamos el **Job** que hemos creado, con el siguiente comandos:

```
nomad job run example.nomad
```

- Para comprobar el estado del **Job** ejecutado, lo tenemos que hacer con la siguiente instrucción:

```
nomad status example
```



# Modificar Job Creado Con Nomad Hashicorp

- Editamos el fichero de configuración del **Job** creado:

```
sudo nano example.nomad
```

- La opción “**Count**”, sirve para definir el número de contenedores a crear para el despliegue de la aplicación a utilizar:

```
count = 3
```

- La opción “**Task**”, sirve para poder definir el nombre de la tarea a ejecutar, en el nodo de **Nomad Hashicorp**:

```
task "nginx" {  
  driver = "docker"
```

- La opción “**Config**”, sirve para poder configurar los parámetros correspondientes, de la aplicación a desplegar con **Nomad Hashicorp**:

```
config {  
  image = "nginx:latest"  
  port_map {  
    app = 80  
  }  
}
```



# Modificar Job Creado Con Nomad Hashicorp

- Comprobamos los cambios realizados, en los parámetros del fichero de configuración del **Job** creado y ejecutado:

```
nomad job plan example.nomad
```

- Actualizamos el **Job** en ejecución en el nodo de **Nomad Hashicorp**, con los nuevos parámetros de configuración:

```
nomad job run -check-index 94 example.nomad
```

- Comprobamos el estado del **Job** creado, con las modificaciones realizadas en dicho **Job** en ejecución:

```
nomad job status example.nomad
```



# Crear Un Clúster Con Nomad Hashicorp

- Creamos un fichero para poder definir el equipo servidor, para realizar la orquestación de contenedores, en el nodo activo de **Nomad Hashicorp**:

```
sudo nano server.hcl
```

- El contenido que tendrá el fichero, será el siguiente a reflejarse a continuación:

```
# Increase log verbosity
log_level = "DEBUG"
# Setup data dir
data_dir = "/tmp/server1"
# Give the agent a unique name. Defaults to hostname
name = "server1"
# Enable the server
server {
  enabled = true
  # Self-elect, should be 3 or 5 for production
  bootstrap_expect = 1
}
```



# Crear Un Clúster Con Nomad Hashicorp

- Iniciamos el agente del equipo servidor, mediante el fichero de configuración creado:

```
nomad agent -config server.hcl
```

- Podemos definir los equipos clientes, para el clúster de contenedores a crear. Pero en mi caso, como sólo tendré un equipo funcionando como equipo Servidor, pues no será necesario.



# Crear Un Clúster Con Nomad Hashicorp

- Comprobamos el estado actual del nodo de **Nomad Hashicorp**, con la siguiente instrucción:

```
nomad node status
```

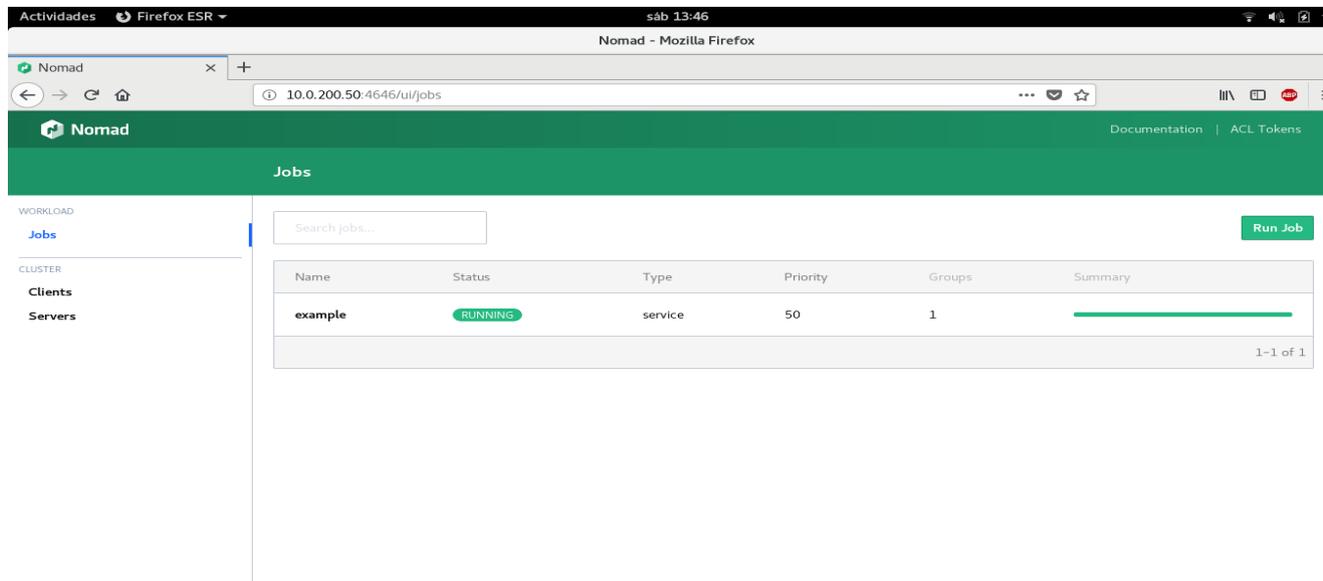
- Comprobamos el estado actual del **Job** en ejecución, en el nodo activo de **Nomad Hashicorp**:

```
nomad status example
```



# Acceso Consola Web-UI De Nomad Hashicorp

- Para el acceso al **Web-UI** de **Nomad Hashicorp**, lo haremos accediendo con la dirección IP del equipo y por el puerto **4646**. En mi caso, la **URL** a la que tengo que acceder es la siguiente: <http://10.0.200.50:4646>



# Conclusión De Nomad Hashicorp

Herramienta útil, que nos permite poder realizar la orquestación de contenedores, de una forma simple y sencilla, para entornos donde no vaya haber un despliegue masivo de grandes aplicaciones, con una consola que nos permite poder gestionar y monitorizar los contenedores activos en dicho nodo de Nomad Hashicorp. Pero sigo pensando, que tampoco es la mejor alternativa de Kubernetes, por su funcionalidad, arquitectura, complejidad y alta disponibilidad que presenta tener.



# Fin De La Presentación

- ¿Alguna/s Pregunta/s, Duda/s o Sugerencia/s que queráis hacer alguno/a/os/as ?

