


# KUBERNETES



José Manuel Martínez Sánchez


Administración de Sistemas Informáticos en red

15-06-2020


	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## ÍNDICE

1	Descripción del proyecto.....	3
2	Introducción teórica.....	3
3	Mapa de red.....	4
4	Configuración de las máquinas .....	5
4.1	Virtualbox .....	5
4.1.1	Configuración de Master. ....	5
4.1.2	Configuración de Node 1 .....	6
4.1.3	Configuración de Node 2.....	7
4.2	Preparación del entorno .....	7
4.2.1	Configuración de master.....	8
4.2.2	Configuración de Node 1 .....	9
4.2.3	Configuración de Node 2.....	10
5	Instalación de Kubernetes y Docker.....	10
5.1	Configuración de Master .....	11
5.2	Configuración Nodo 1 .....	14
5.3	Configuración Nodo 2 .....	16
6	Instalación de Dashboard. ....	18
7	Volúmenes.....	19
7.1	Configuración Master. ....	20
7.2	Configuración de nodos. ....	20
8	Persistent Volumes.....	21
9	Secrets .....	22

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

10	Deployment drupal .....	23
11	Deployment Mysql .....	25
12	Volumes Nfs y hostpath .....	28
13	Escalar deployment drupal .....	31
14	Asignar pods a determinados nodos. ....	32
15	Cambiar versión deployment Drupal.....	33
16	Instalación de drupal.....	34
17	Conclusiones.....	36
18	Bibliografía.....	37

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 1 Descripción del proyecto

El proyecto consiste en desplegar el CMS Drupal junto al SGBD MySQL en Kubernetes, esto va a permitir una mejor administración de estos despliegues como escalabilidad, tolerancia a fallos, autorreparación, actualizaciones, etc.

Para esta implementación utilizaremos 3 máquinas virtuales en Virtualbox con el sistema operativo Centos 7, esto formará un clúster que estarán compuestos por una máquina Master (encargada de mantener el estado deseado para el clúster) y dos máquinas Nodos (donde se ejecutan las aplicaciones).


Vamos a implementar un deployment por cada contenedor para permitir la escalabilidad, tolerancia a fallos, actualizaciones sin que el CMS presente ninguna caída.

También crearemos el dashboard de Kubernetes para una vista más gráfica de nuestro clúster.

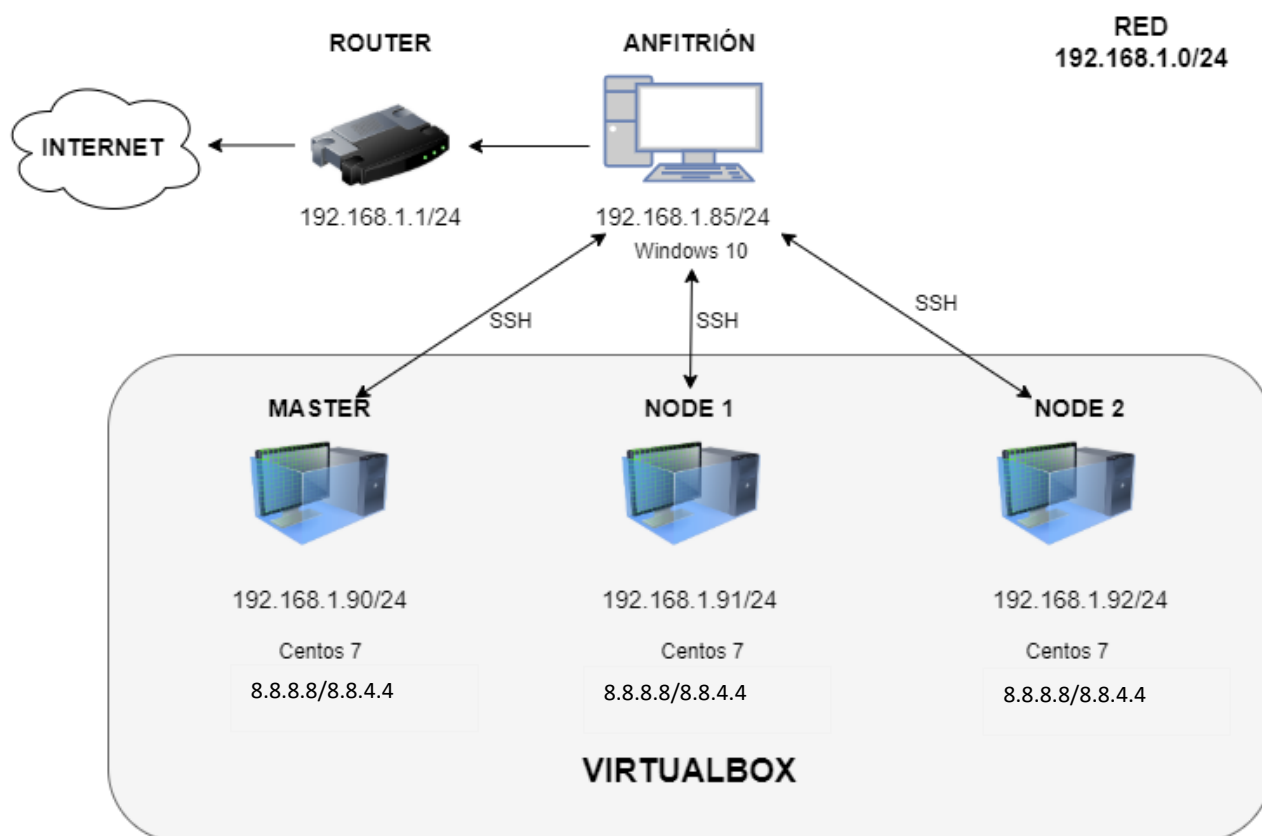
## 2 Introducción teórica.


Kubernetes es un sistema de orquestación de contenedores, lo que significa que el software no se encarga de crearlos, sino de administrarlos. Para ello, Kubernetes aplica la automatización de procesos, lo que vuelve más fácil para los desarrolladores comprobar, mantener o publicar aplicaciones. La arquitectura de Kubernetes consta de una clara jerarquía, compuesta por los siguientes elementos:

- Contenedor: incluye las aplicaciones.
- Pod: se encarga de agrupar aquellos contenedores que necesitan trabajar juntos para el funcionamiento de una aplicación.
- Nodo: uno o varios pods se ejecutan en un nodo, que puede ser tanto una máquina virtual como física.
- Clúster: los nodos se agrupan en clústeres.

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

### 3 Mapa de red



	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 4 Configuración de las máquinas

### 4.1 Virtualbox

Las 3 máquinas tendrán la misma configuración:

	MASTER	NODE 1	NODE 2
Red	Adaptador Puente	Adaptador Puente	Adaptador Puente
RAM	2048 MB	2048 MB	2048 MB
Disco duro	100 GB	100 GB	100 GB
Sistema operativo	Centos 7	Centos 7	Centos 7
N.º procesadores	2	2	2
Entorno gráfico	X	X	X

#### 4.1.1 Configuración de Master.

General

Nombre: Kubernetes Master  
Sistema operativo: Red Hat (64-bit)

Sistema

Memoria base: 2048 MB  
Procesadores: 2  
Orden de arranque: Disquete, Óptica, Disco duro  
Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB  
Controlador gráfico: VMSVGA  
Servidor de escritorio remoto: Inhabilitado  
Grabación: Inhabilitado

Almacenamiento

Controlador: IDE  
IDE secundario maestro: [Unidad óptica] Vacío  
Controlador: SATA  
Puerto SATA 0: Kubernetes Master-disk1.vdi (Normal, 100,00 GB)

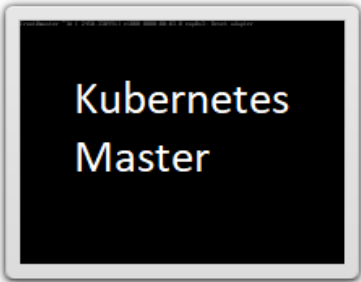
Audio


Controlador de anfitrión: Windows DirectSound  
Controlador: ICH AC97

Red








Adaptador 1: Intel PRO/1000 MT Desktop (Adaptador puente, «Realtek PCIe GBE Family Controller»)


Previsualización



	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

#### 4.1.2 Configuración de Node 1

<div> <b>General</b></div> <p>Nombre: Kubernetes node1 Sistema operativo: Red Hat (64-bit)</p> <div> <b>Sistema</b></div> <p>Memoria base: 2048 MB Procesadores: 2 Orden de arranque: Disquete, Óptica, Disco duro Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM</p>	<div> <b>Previsualización</b></div> <div style="border: 1px solid gray; padding: 10px; text-align: center; background-color: black; color: white; width: 200px; margin: 0 auto;"> <b>Kubernetes node1</b> </div>
<div> <b>Pantalla</b></div> <p>Memoria de vídeo: 16 MB Controlador gráfico: VMSVGA Servidor de escritorio remoto: Inhabilitado Grabación: Inhabilitado</p>	
<div> <b>Almacenamiento</b></div> <p>Controlador: IDE IDE secundario maestro: [Unidad óptica] Vacío Controlador: SATA Puerto SATA 0: Kubernetes node1 -disk1.vdi (Normal, 100,00 GB)</p>	
<div> <b>Audio</b></div> <p>Controlador de anfitrión: Windows DirectSound Controlador: ICH AC97</p>	
<div> <b>Red</b></div> <p>Adaptador 1: Intel PRO/1000 MT Desktop (Adaptador puente, «Realtek PCIe GBE Family Controller»)</p>	

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

#### 4.1.3 Configuración de Node 2

General

Nombre: **Kubernetes node2**

Sistema operativo: Red Hat (64-bit)

Sistema

Memoria base: 2048 MB

Procesadores: 2

Orden de arranque: Disquete, Óptica, Disco duro

Aceleración: VT-x/AMD-V, Paginación anidada, PAE/NX, Paravirtualización KVM

Pantalla

Memoria de vídeo: 16 MB

Controlador gráfico: VMSVGA

Servidor de escritorio remoto: Inhabilitado

Grabación: Inhabilitado

Almacenamiento

Controlador: IDE

IDE secundario maestro: [Unidad óptica] Vacío

Controlador: SATA

Puerto SATA 0: Kubernetes node2 -disk1.vdi (Normal, 100,00 GB)

Audio

Controlador de anfitrión: Windows DirectSound

Controlador: ICH AC97

Red

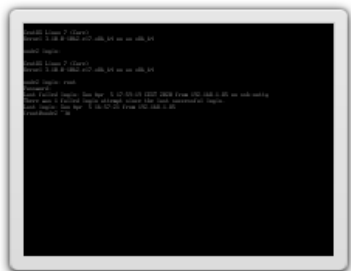
Adaptador 1: Intel PRO/1000 MT Desktop (Adaptador puente, «Realtek PCIe GBE Family Controller»)

USB

Controlador USB: OHCI

Filtros de dispositivos: 0 (0 activo)

Previsualización




#### 4.2 Preparación del entorno

Primero tendremos que configurar la red para que todas las máquinas estén en la misma.

	MASTER	NODE 1	NODE 2
Ip	192.168.1.90	192.168.1.91	192.168.1.92
Máscara de red	/24	/24	/24
Gateway	192.168.1.1	192.168.1.1	192.168.1.1
Ip estática	Si	SI	Si
DNS	8.8.8.8 / 8.8.4.4	8.8.8.8 / 8.8.4.4	8.8.8.8 / 8.8.4.4



	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	


#### 4.2.1 Configuración de master.

```
GNU nano 2.3.1 Fichero: /etc/sysconfig/network-scripts/ifcfg-enp0s3
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="enp0s3"
UUID="e051e8e4-37b0-4a1b-a48a-d9a72f2b6991"
DEVICE="enp0s3"
ONBOOT="yes"
IPADDR="192.168.1.90"
PREFIX="24"
GATEWAY="192.168.1.1"
DNS1="8.8.8.8"
DOMAIN="8.8.4.4"
IPV6_PRIVACY="no"
```

```
GNU nano 2.3.1 Fichero: /etc/hostname
master
```

```
GNU nano 2.3.1 Fichero: /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4 localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.1.90 master
192.168.1.91 node1
192.168.1.92 node2
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

#### 4.2.2 Configuración de Node 1

```

GNU nano 2.3.1          Fichero: /etc/sysconfig/network-scripts/ifcfg-enp0s3
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="enp0s3"
UUID="e051e8e4-37b0-4a1b-a48a-d9a72f2b6991"
DEVICE="enp0s3"
ONBOOT="yes"
IPADDR="192.168.1.91"
PREFIX="24"
GATEWAY="192.168.1.1"
DNS1="8.8.8.8"
DOMAIN="8.8.4.4"
IPV6_PRIVACY="no"

```

```

GNU nano 2.3.1          Fichero: /etc/hostname
node1


```

```

GNU nano 2.3.1          Fichero: /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.1.90 master
192.168.1.91 node1
192.168.1.92 node2

```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

### 4.2.3 Configuración de Node 2

```

GNU nano 2.3.1          Fichero: /etc/sysconfig/network-scripts/ifcfg-enp0s3
TYPE="Ethernet"
PROXY_METHOD="none"
BROWSER_ONLY="no"
BOOTPROTO="static"
DEFROUTE="yes"
IPV4_FAILURE_FATAL="no"
IPV6INIT="yes"
IPV6_AUTOCONF="yes"
IPV6_DEFROUTE="yes"
IPV6_FAILURE_FATAL="no"
IPV6_ADDR_GEN_MODE="stable-privacy"
NAME="enp0s3"
UUID="e051e8e4-37b0-4a1b-a48a-d9a72f2b6991"
DEVICE="enp0s3"
ONBOOT="yes"
IPADDR="192.168.1.92"
PREFIX="24"
GATEWAY="192.168.1.1"
DNS1="8.8.8.8"
DOMAIN="8.8.4.4"
IPV6_PRIVACY="no"

```

```

GNU nano 2.3.1          Fichero: /etc/hostname
node2
|

```

```

GNU nano 2.3.1          Fichero: /etc/hosts
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6


192.168.1.90 master
192.168.1.91 node1
192.168.1.92 node2
|

```

## 5 Instalación de Kubernetes y Docker

En este repositorio de GitHub podéis encontrar todos los comandos:

[https://github.com/josemartinezz98/Kubernetes/blob/master/Comandos\\_Instalacion\\_Kubernetes.txt](https://github.com/josemartinezz98/Kubernetes/blob/master/Comandos_Instalacion_Kubernetes.txt)

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 5.1 Configuración de Master

Primero de todo vamos a deshabilitar la swap.

```
[root@master ~]# swapoff -a
[root@master ~]# |
```

Para desactivarla totalmente en /etc/fstab comentamos la línea de swap.

```
GNU nano 2.3.1          Fichero: /etc/fstab

/etc/fstab
# Created by anaconda on Fri Apr  3 21:03:21 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=78f0f502-1249-4269-82a2-21c6c3d4ccd8 /boot          xfs     defaults        0 0
/dev/mapper/centos_master-home /home          xfs     defaults        0 0
# /dev/mapper/centos_master-swap swap            swap    defaults        0 0
```


Para la instalación de Kubernetes necesitamos desactivar algunas reglas como desactivar Ipv6.

Con el comando “sysctl –system” aplicamos los cambios.

```
[root@master ~]# cat <<EOF > /etc/sysctl.d/k8s.conf
> net.bridge.bridge-nf-call-ip6tables = 1
> net.bridge.bridge-nf-call-iptables = 1
> EOF
[root@master ~]# sysctl --system
```

Tenemos que desactivar Selinux con el comando setenforce 0 y en el fichero /etc/selinux/config para consolidar el cambio

```
[root@master ~]# setenforce 0
[root@master ~]# |
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

```
GNU nano 2.3.1 File: /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced..
#     SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
```

Instalamos Docker y lo habilitamos.

```
[root@master ~]# yum install -y docker

[reboot]
[root@master ~]# systemctl enable docker && systemctl start docker
```

Creamos el fichero que apunta al repositorio para instalar Kubernetes.

```
[root@master ~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
> [kubernetes]
> name=Kubernetes
> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
> enabled=1
> gpgcheck=1
> repo_gpgcheck=1
> gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> EOF
```


Y a continuación instalamos y lo iniciamos.

```
[root@master ~]# yum install -y kubelet kubeadm kubectl

[root@master ~]# systemctl enable kubelet && systemctl start kubelet
```

Desactivamos el cortafuegos porque después da fallos.

```
[root@master ~]# systemctl stop firewalld
[root@master ~]#
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Inicializamos el clúster.

```
[root@master ~]# kubeadm init
```

Nos vamos a guardar este comando que nos genera para poder unir el master con los otros nodos, este comando se tendrá que ejecutar en todas las máquinas que sean nodos.

```
kubeadm join 192.168.1.90:6443 --token ksidp4.7in118wkgd811z02 \
--discovery-token-ca-cert-hash sha256:a83e618d3030a992208e3a3454a0055db793f4429514197ed9eb4153c3d5c83b
[root@master ~]#
```

Para exportar la variable de entorno de kubeconfig.

```
[root@master ~]# export KUBECONFIG=/etc/kubernetes/admin.conf
```

Vamos a ejecutar el siguiente comando para descargar Weave Net, una herramienta que crea una red virtual dentro de nuestro clúster para la configuración automática de red de nuestros pods.

```
[root@master ~]# kubectl apply -f "https://cloud.weave.works/k8s/net?k8s-version=$(kubectl version | base64 | tr -d '\n')"
```


```
serviceaccount/weave-net created
clusterrole.rbac.authorization.k8s.io/weave-net created
clusterrolebinding.rbac.authorization.k8s.io/weave-net created
role.rbac.authorization.k8s.io/weave-net created
rolebinding.rbac.authorization.k8s.io/weave-net created
daemonset.apps/weave-net created
[root@master ~]#
```

Con este comando podemos ver los pods.

```
[root@master ~]# kubectl get pods --all-namespaces
```

NAMESPACE	NAME	READY	STATUS	RESTARTS	AGE
kube-system	coredns-66bff467f8-9b9ss	0/1	Pending	0	6m29s
kube-system	coredns-66bff467f8-mgxlq	0/1	Pending	0	6m28s
kube-system	etcd-master	1/1	Running	0	6m24s
kube-system	kube-apiserver-master	1/1	Running	0	6m24s
kube-system	kube-controller-manager-master	1/1	Running	0	6m24s
kube-system	kube-proxy-4vxs7	1/1	Running	0	6m28s
kube-system	kube-scheduler-master	1/1	Running	0	6m24s
kube-system	weave-net-lptjc	0/2	ContainerCreating	0	2m1s

```
[root@master ~]#
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Normalmente Master no suele ejecutar pods, si lo desea hacer debemos ejecutar este

```
[root@master ~]# kubectl taint nodes --all node-role.kubernetes.io/master-  
node/master untainted  
[root@master ~]#
```

comando.

## 5.2 Configuración Nodo 1

Deshabilitamos la swap

```
[root@node1 ~]# swapoff -a
```


La deshabilitamos permanentemente

```
GNU nano 2.3.1          Fichero: /etc/fstab

/etc/fstab
# Created by anaconda on Fri Apr  3 21:03:21 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=78f0f502-1249-4269-82a2-21c6c3d4ccd8 /boot          xfs     defaults        0 0
/dev/mapper/centos_master-home /home          xfs     defaults        0 0
# /dev/mapper/centos_master-swap swap           swap    defaults        0 0
```

Desactivamos las reglas de Ipv6

```
[root@node1 ~]# cat <<EOF > /etc/sysctl.d/k8s.conf
> net.bridge.bridge-nf-call-ip6tables = 1
> net.bridge.bridge-nf-call-iptables = 1
> EOF
[root@node1 ~]# sysctl --system
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Desactivamos Selinux

```
[root@node1 ~]# setenforce 0
```

Desactivamos selinux permanentemente en el archivo /etc/selinux/config

```
GNU nano 2.3.1 File: /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced..
#   SELinux permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
```

Instalamos e iniciamos Docker

```
[root@node1 ~]# yum install -y docker
```

```
[root@node1 ~]# systemctl enable docker && systemctl start docker
```

Introducimos repositorio para kubernetes.


```
[root@node1 ~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
[kubernetes]
name=Kubernetes
baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
enabled=1
gpgcheck=1
repo_gpgcheck=1
gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
EOF
```

Instalamos kubelet y lo iniciamos

```
[root@node1 ~]# yum install -y kubelet kubeadm kubectl
```

```
[root@node1 ~]# systemctl enable kubelet && systemctl start kubelet
```



	Curso escolar: 2019/2020	Ciclo: CFGs de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Deshabilitamos y paramos el firewalld para los posibles errores.

```
[root@node1 ~]# systemctl disable firewalld && systemctl stop firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

Comando para la unión del nodo 1 al clúster

```
[root@node1 ~]# kubeadm join 192.168.1.90:6443 --token ksidp4.7in1l8wkgd811z02 --discovery-token-ca-cert-hash sha256:a83e618d3030a992208e3a3454a0055db793f4429514197ed9eb4153c3d5c83b
```

Con el siguiente comando comprobamos que se ha unido correctamente el nodo 1 al cluster. (Status Ready)

```
[root@master ~]# kubectl get nodes
NAME      STATUS    ROLES    AGE   VERSION
master    Ready     master   69m   v1.18.0
node1     Ready     <none>   18m   v1.18.0
```

### 5.3 Configuración Nodo 2


Deshabilitamos la swap

```
[root@node2 ~]# swapoff -a
```

La deshabilitamos permanentemente

```
GNU nano 2.3.1          Fichero: /etc/fstab

/etc/fstab
# Created by anaconda on Fri Apr  3 21:03:21 2020
#
# Accessible filesystems, by reference, are maintained under '/dev/disk'
# See man pages fstab(5), findfs(8), mount(8) and/or blkid(8) for more info
#
UUID=78f0f502-1249-4269-82a2-21c6c3d4ccd8 /boot xfs defaults 0 0
/dev/mapper/centos_master-home /home xfs defaults 0 0
# /dev/mapper/centos_master-swap swap xfs defaults 0 0
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Desactivamos las reglas de Ipv6

```
[root@node2 ~]# cat <<EOF > /etc/sysctl.d/k8s.conf
> net.bridge.bridge-nf-call-ip6tables = 1
> net.bridge.bridge-nf-call-iptables = 1
> EOF
[root@node2 ~]# sysctl --system
```

Desactivamos Selinux con setenforce y en el fichero para consolidar

```
[root@node2 ~]# setenforce 0
```

```
GNU nano 2.3.1 File: /etc/selinux/config
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#   enforcing - SELinux security policy is enforced..
#   SELinux permissive - SELinux prints warnings instead of enforcing.
#   disabled - No SELinux policy is loaded.
SELINUX=disabled
```


Instalamos e iniciamos docker

```
[root@node2 ~]# yum install -y docker
```

```
[root@node2 ~]# systemctl enable docker && systemctl start docker
```

Introducimos repositorio para kubernetes

```
[root@node2 ~]# systemctl enable docker && systemctl start docker
[root@node2 ~]# cat <<EOF > /etc/yum.repos.d/kubernetes.repo
> [kubernetes]
> name=Kubernetes
> baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86_64
> enabled=1
> gpgcheck=1
> repo_gpgcheck=1
> gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg
> EOF
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Instalamos kubelet y lo activamos

```
[root@node2 ~]# yum install -y kubelet kubeadm kubectl
```

```
[root@node2 ~]# systemctl enable kubelet && systemctl start kubelet
```

Desactivamos el firewall para posibles problemas

```
[root@node2 ~]# systemctl disable firewalld && systemctl stop firewalld
Removed symlink /etc/systemd/system/multi-user.target.wants/firewalld.service.
Removed symlink /etc/systemd/system/dbus-org.fedoraproject.FirewallD1.service.
```

Unimos el nodo 2 al clúster

```
[root@node2 ~]# kubeadm join 192.168.1.90:6443 --token ksidp4.7in1l8wkgd8l1z02 --discovery-token-ca-cert-hash sha256:a83e618d3030a992208e3a3454a0055db793f4429514197ed9eb4153c3d5c83b
```

Comprobamos que se han unido correctamente todos los nodos al cluster.


```
[root@master ~]# kubectl get nodes
NAME      STATUS   ROLES    AGE   VERSION
master    Ready    master   69m   v1.18.0
node1     Ready    <none>   18m   v1.18.0
node2     Ready    <none>   82s   v1.18.0
[root@master ~]#
```

## 6 Instalación de Dashboard.

Comandos: <https://www.returngis.net/2019/03/instalar-kubernetes-dashboard-en-tu-cluster/>

Desplegamos el yaml que crea el dashboard.

```
[root@master ~]# kubectl create -f https://raw.githubusercontent.com/kubernetes/dashboard/v1.10.1/src/deploy/alternative/kubernetes-dashboard.yaml
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Damos los permisos necesarios para poder acceder a él.

```
[root@master ~]# kubectl create clusterrolebinding kubernetes-dashboard --clusterrole=cluster-admin --serviceaccount=kube-system:kubernetes-dashboard
```

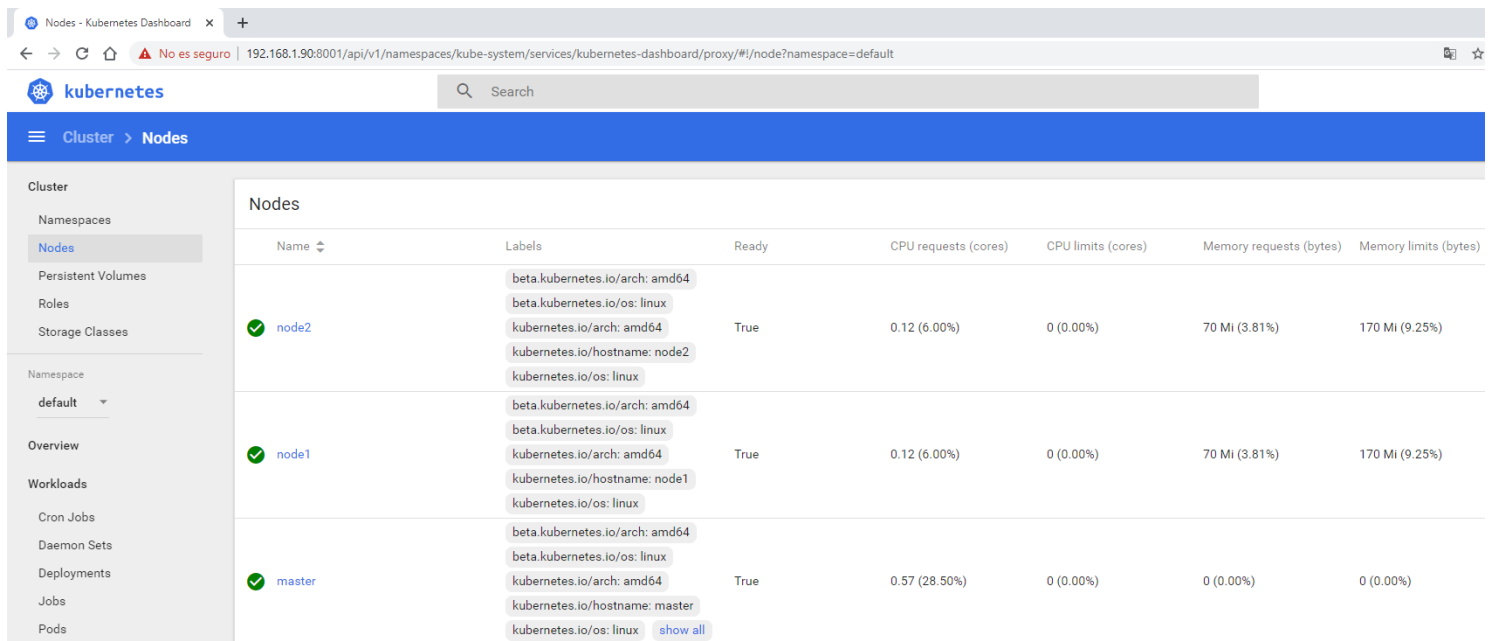
Ejecutamos el comando para poder acceder a él desde el navegador de nuestro ordenador anfitrión.

```
[root@master ~]# kubectl proxy --address 0.0.0.0 --accept-hosts '.*'
```

Con el siguiente comando accederíamos desde el navegador del pc anfitrión.

(cambiar 192.168.1.90 por ip de cada anfitrión):


<http://192.168.1.90:8001/api/v1/namespaces/kube-system/services/kubernetes-dashboard/proxy/#!/node?namespace=default>



Name	Labels	Ready	CPU requests (cores)	CPU limits (cores)	Memory requests (bytes)	Memory limits (bytes)
node2	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64 kubernetes.io/hostname: node2 kubernetes.io/os: linux	True	0.12 (6.00%)	0 (0.00%)	70 Mi (3.81%)	170 Mi (9.25%)
node1	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64 kubernetes.io/hostname: node1 kubernetes.io/os: linux	True	0.12 (6.00%)	0 (0.00%)	70 Mi (3.81%)	170 Mi (9.25%)
master	beta.kubernetes.io/arch: amd64 beta.kubernetes.io/os: linux kubernetes.io/arch: amd64 kubernetes.io/hostname: master kubernetes.io/os: linux	True	0.57 (28.50%)	0 (0.00%)	0 (0.00%)	0 (0.00%)

## 7 Volúmenes

Vamos a instalar nfs en nuestro clúster para poder compartir información entre todos los nodos.

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 7.1 Configuración Master.

Instalamos Nfs

```
[root@master ~]# yum install nfs-utils
```

Creamos el directorio que vamos a compartir

```
[root@master shared]# mkdir /var/shared
```

En el fichero /etc/exports introducimos lo siguiente para compartir el directorio

```
GNU nano 2.3.1 Fichero: /etc/exports
/var/shared 192.168.1.0/24(rw,sync,no_root_squash,no_all_squash)
```

Comprobamos que está compartido

```
[root@master shared]# showmount -e 127.0.0.1
Export list for 127.0.0.1:
/var/shared 192.168.1.0/24
```

## 7.2 Configuración de nodos.

Instalamos Nfs

```
[root@node1 data]# yum install nfs-utils
```


Comprobamos que podemos ver la carpeta compartida

```
[root@node1 data]# showmount -e 192.168.1.90
Export list for 192.168.1.90:
/var/shared 192.168.1.0/24
[root@node1 data]#
```

Montamos la carpeta compartida en una carpeta de nuestra máquina en este caso

/var/data

```
[root@node1 data]# mount -t nfs4 192.168.1.90:/var/shared /var/data
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 8 Persistent Volumes

Archivo yaml:

[https://github.com/josemartinezz98/Kubernetes/blob/master/Persistent\\_Volume.yaml](https://github.com/josemartinezz98/Kubernetes/blob/master/Persistent_Volume.yaml)

Creamos dos Persistent Volumes para mantener información de forma persistente de los pods.

```
GNU nano 2.3.1                                Fichero: persistent_volume.yaml


apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-pv1
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Recycle
  hostPath:
    path: /home

---

apiVersion: v1
kind: PersistentVolume
metadata:
  name: nfs-pv2
spec:
  capacity:
    storage: 1Gi
  accessModes:
    - ReadWriteMany
  persistentVolumeReclaimPolicy: Recycle
  nfs:
    path: /var/shared
    server: 192.168.1.90
```

Ejecutamos el yaml creado anteriormente

```
[root@master ~]# kubectl apply -f persistent_volume.yaml
persistentvolume/nfs-pv1 created
persistentvolume/nfs-pv2 created
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Comprobamos que están disponibles para ser utilizados

```
[root@master ~]# kubectl get pv
NAME          CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS   CLAIM  STORAGECLASS  REASON   AGE
nfs-pv1       1Gi       RWX           Recycle         Available             storageclass  22s
nfs-pv2       1Gi       RWX           Recycle         Available             storageclass  22s
```

## 9 Secrets

Archivo yaml:

<https://github.com/josemartinezz98/Kubernetes/blob/master/Secrets.yaml>

Vamos a crear el objeto secret para crear variables de entorno que no estén directamente en texto plano.

```
GNU nano 2.3.1                                Fichero: secrets.yaml


apiVersion: v1
kind: Secret
metadata:
  name: secrets
type: Opaque
stringData:
  root_password: "root"
  password_jose: "jose"
```

Ejecutamos el yaml y vemos que se crea.

```
[root@master ~]# kubectl apply -f secrets.yaml
secret/secrets created
```

Comprobamos que se ha creado correctamente.

```
[root@master ~]# kubectl get secrets
NAME          TYPE          DATA  AGE
default-token-6wnjm  kubernetes.io/service-account-token  3      68d
secrets        Opaque        2      24s
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 10 Deployment drupal

Archivo yaml:

[https://github.com/josemartinezz98/Kubernetes/blob/master/Drupal\\_Deployment\\_Service\\_Claim.yaml](https://github.com/josemartinezz98/Kubernetes/blob/master/Drupal_Deployment_Service_Claim.yaml)

Este yaml permite crear el cms drupal lo implementaremos con los objetos Service, Persistent Volume Claim y Deployment:

**Service** → permite un acceso siempre disponible al pod, aunque este se reinicie, elimine y sea restaurado de nuevo, etc.

**Persistent Volume Claim** → Es una reclamación de almacenamiento para los datos de estos pods, esto reclamará el espacio a un Persistent Volume creado anteriormente.


**Deployment** → Esto permitirá crear pods, replica sets y el propio deployment.

```

GNU nano 2.3.1                                Fichero: deployment_drupal.yaml
apiVersion: v1
kind: Service
metadata:
  name: drupal
  labels:
    app: drupal
spec:
  ports:
    - port: 80
  selector:
    app: drupal
    tier: frontend
  type: NodePort
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-pvc1
  labels:
    app: drupal
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi

```



	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

```

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: drupal
  labels:
    app: drupal
spec:
  selector:
    matchLabels:
      app: drupal
      tier: frontend
  replicas: 3
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: drupal
        tier: frontend
    spec:
      containers:
      - image: drupal
        name: drupal
        ports:
        - containerPort: 80
          name: drupal
        volumeMounts:
        - name: drupal-pvc
          mountPath: /home
      volumes:
      - name: drupal-pvc
        persistentVolumeClaim:
          claimName: nfs-pvc1

```

Ejecutamos el archivo yaml para su creación.

```

[root@master ~]# kubectl apply -f deployment_drupal.yaml
service/drupal created
persistentvolumeclaim/nfs-pvc1 created
deployment.apps/drupal created


```

Comprobamos que la reclamación de volumen persistente ha sido correcta.

```

[root@master ~]# kubectl get pvc
NAME          STATUS    VOLUME   CAPACITY   ACCESS MODES   STORAGECLASS   AGE
nfs-pvc1      Bound    nfs-pv1   1Gi        RWX              nfs              4s

```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Vemos los diferentes objetos creados por el archivo yaml.

```
[root@master ~]# kubectl get svc,rs,pod,deployment
NAME                                     TYPE          CLUSTER-IP    EXTERNAL-IP    PORT(S)          AGE
service/drupal                         NodePort      10.110.99.18   <none>         80:30296/TCP     10s
service/kubernetes                     ClusterIP     10.96.0.1      <none>         443/TCP          68d
service/mysql                          ClusterIP     None           <none>         3306/TCP         39m

NAME                                     DESIRED       CURRENT        READY          AGE
replicaset.apps/drupal-7b795f7d9d       1             1              1             10s
replicaset.apps/mysql-f4b58b5dd         1             1              1             39m

NAME                                     READY        STATUS         RESTARTS       AGE
pod/drupal-7b795f7d9d-bk574             1/1         Running        0              10s
pod/mysql-f4b58b5dd-kmqds               1/1         Running        0              39m


NAME                                     READY        UP-TO-DATE     AVAILABLE      AGE
deployment.apps/drupal                   1/1         1              1             10s
deployment.apps/mysql                   1/1         1              1             39m
```

## 11 Deployment Mysql

Archivo yaml:

[https://github.com/josemartinezz98/Kubernetes/blob/master/Drupal\\_Deployment\\_Service\\_Claim.yaml](https://github.com/josemartinezz98/Kubernetes/blob/master/Drupal_Deployment_Service_Claim.yaml)


Vamos a crear un yaml para la implementación de Mysql con los objetos Service, Persistent Volume Claim y Deployment, este está configurado para que drupal lo utilice como base de datos.

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

```

GNU nano 2.3.1                               Fichero: deployment_mysql.yaml

apiVersion: v1
kind: Service
metadata:
  name: mysql
  labels:
    app: drupal
spec:
  ports:
    - port: 3306
  selector:
    app: drupal
    tier: mysql
  clusterIP: None
---
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: nfs-pvc2
  labels:
    app: drupal
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 1Gi
---
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

```

---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
  labels:
    app: drupal
spec:
  selector:
    matchLabels:
      app: drupal
      tier: mysql
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: drupal
        tier: mysql
    spec:
      containers:
      - image: mysql:5.6
        name: mysql
        env:
        - name: MYSQL_ROOT_PASSWORD
          valueFrom:
            secretKeyRef:
              name: secrets
              key: root_password
        - name: MYSQL_DATABASE
          value: drupal
        - name: MYSQL_USER
          value: jose
        - name: MYSQL_PASSWORD
          valueFrom:
            secretKeyRef:
              name: secrets
              key: password_jose
        ports:
        - containerPort: 3306
          name: mysql
        volumeMounts:
        - name: mysql-pvc
          mountPath: /var/lib/mysql
      volumes:
      - name: mysql-pvc
        persistentVolumeClaim:
          claimName: nfs-pvc2


```

Ejecutamos el archivo yaml de MySQL

```

[root@master ~]# kubectl apply -f deployment_mysql.yaml
service/mysql created
persistentvolumeclaim/nfs-pvc2 created
deployment.apps/mysql created

```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

Vemos que se ha unido a un Persistent Volume.

```
[root@master ~]# kubectl get pvc
```

NAME	STATUS	VOLUME	CAPACITY	ACCESS MODES	STORAGECLASS	AGE
nfs-pvc1	Bound	nfs-pv1	1Gi	RWX		42m
nfs-pvc2	Bound	nfs-pv2	1Gi	RWX		41m

Comprobamos que se han ejecutado correctamente los objetos definidos en el archivo.

```
[root@master ~]# kubectl get svc,rs,pod,deploy
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
service/drupal	NodePort	10.105.179.202	<none>	80:30731/TCP	44m
service/kubernetes	ClusterIP	10.96.0.1	<none>	443/TCP	49d
service/mysql	ClusterIP	None	<none>	3306/TCP	42m

NAME	DESIRED	CURRENT	READY	AGE
replicaset.apps/drupal-69f76fc455	1	1	1	44m
replicaset.apps/mysql-f4b58b5dd	1	1	1	42m

NAME	READY	STATUS	RESTARTS	AGE
pod/drupal-69f76fc455-45v95	1/1	Running	0	5m54s
pod/mysql-f4b58b5dd-bk574	1/1	Running	0	35m


NAME	READY	UP-TO-DATE	AVAILABLE	AGE
deployment.apps/drupal	1/1	1	1	44m
deployment.apps/mysql	1/1	1	1	42m

## 12 Volumes Nfs y hostpath

Crearemos dos tipos distintos de Volumes:

Nfs → Este volumen permite compartir la información de un directorio del pod con todos los nodos del clúster a través de nfs instalado en la máquina Master.

Vemos que el persistent volumen creado anteriormente tiene esta configuración.

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

```
[root@master ~]# kubectl describe pv nfs-pv2
Name:          nfs-pv2
Labels:        <none>
Annotations:   pv.kubernetes.io/bound-by-controller: yes
Finalizers:    [kubernetes.io/pv-protection]
StorageClass:
Status:        Bound
Claim:         default/nfs-pvc2
Reclaim Policy: Recycle
Access Modes:  RWX
VolumeMode:    Filesystem
Capacity:      1Gi
Node Affinity: <none>
Message:
Source:
  Type:        NFS (an NFS mount that lasts the lifetime of a pod)
  Server:      192.168.1.90
  Path:        /var/shared
  ReadOnly:    false
Events:        <none>
```

Comprobamos que tenemos la información compartida del pod en la carpeta compartida del nodo master


Maquina Master

```
[root@master ~]# cd /var/shared/
[root@master shared]# ls
auto.cnf  ibdata1  ib_logfile0  ib_logfile1  mysql  performance_schema  test
```

- Pod Mysql

```
root@mysql-f4b58b5dd-bk574:/var/lib/mysql# ls
auto.cnf  ib_logfile0  ib_logfile1  ibdata1  mysql  performance_schema  test
```

Hostpath → Este volumen permite compartir la información de dentro del pod con el nodo que esté ejecutándolo.

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

```
[root@master ~]# kubectl describe pv nfs-pv1
Name:          nfs-pv1
Labels:        <none>
Annotations:   pv.kubernetes.io/bound-by-controller: yes
Finalizers:    [kubernetes.io/pv-protection]
StorageClass:
Status:        Bound
Claim:         default/nfs-pvc1
Reclaim Policy: Recycle
Access Modes:  RWX
VolumeMode:    Filesystem
Capacity:      1Gi
Node Affinity: <none>
Message:
Source:
  Type:        HostPath (bare host directory volume)
  Path:        /home
  HostPathType:
Events:        <none>
```

Creamos una carpeta en el directorio compartido

```
root@drupal-69f76fc455-kmqds:/home# ls
Share_Hostpath
```

Vemos que se está ejecutando en el nodo 2


```
[root@master ~]# kubectl get pod -o wide
NAME          READY   STATUS    RESTARTS   AGE   IP          NODE   NOMINATED NODE   READINESS GATES
drupal-69f76fc455-kmqds 1/1     Running   0          20m   10.40.0.2   node2   <none>           <none>
```

Solo comparte con el nodo 2

```
[root@node2 home]# ls
Share_Hostpath
```

Vemos que en el nodo 1 no se ha compartido

```
[root@node1 ~]# cd /home/
[root@node1 home]# ls
[root@node1 home]#
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

### 13 Escalar deployment drupal

Vamos a crear varias replicas de un pod, existen dos formas:

- 1) Indicando en el comando el nombre del deployment y cuantas replicas queremos

```
[root@master ~]# kubectl scale deployment drupal --replicas=3
deployment.apps/drupal scaled
```


- 2) Modificamos el archivo yaml e introducimos la línea “replicas” y escribimos cuantas replicas queríamos.

```
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: drupal
  labels:
    app: drupal
spec:
  selector:
    matchLabels:
      app: drupal
      tier: frontend
  replicas: 3
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: drupal
        tier: frontend
    spec:
      containers:
      - image: drupal:7
```

Ejecutamos el archivo yaml modificado y vemos que antes era 1 deseado y ahora son 3.

```
[root@master ~]# kubectl get rs
NAME                                DESIRED  CURRENT  READY  AGE
drupal-69f76fc455                   1        1        1      38m
mysql-f4b58b5dd                    1        1        1      36m
[root@master ~]# kubectl apply -f drupal.yaml
service/drupal unchanged
persistentvolumeclaim/nfs-pvc1 unchanged
deployment.apps/drupal configured
[root@master ~]# kubectl get rs
NAME                                DESIRED  CURRENT  READY  AGE
drupal-69f76fc455                   3        3        1      38m
mysql-f4b58b5dd                    1        1        1      36m
```



	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

#### 14 Asignar pods a determinados nodos.

Introduciremos unos labels para identificar a los nodos

```
[root@master ~]# kubectl label nodes node1 tier=frontend
node/node1 labeled
[root@master ~]# kubectl label nodes node2 tier=mysql
node/node2 labeled
```

Añadiremos el label creado al deployment con la línea “nodeSelector”


```
spec:
  containers:
  - image: drupal:7
    name: drupal
    ports:
    - containerPort: 80
      name: drupal
    volumeMounts:
    - name: drupal-pvc
      mountPath: /home
  volumes:
  - name: drupal-pvc
    persistentVolumeClaim:
      claimName: nfs-pvc1
  nodeSelector:
    tier: frontend
```

Ejecutamos el archivo yaml

```
[root@master ~]# kubectl apply -f deployment_drupal.yaml
service/drupal unchanged
persistentvolumeclaim/nfs-pvc1 unchanged
deployment.apps/drupal configured
```

Vemos que todos los pods de ese deployment se han cambiado al nodo 1

```
[root@master ~]# kubectl get pod -o wide
NAME READY STATUS RESTARTS AGE IP NODE NOMINATED NODE READINESS GATES
drupal-7dcbb4d56f-2jcf9 1/1 Running 0 66s 10.38.0.2 node1 <none> <none>
drupal-7dcbb4d56f-zmbz9 1/1 Running 0 66s 10.38.0.4 node1 <none> <none>
drupal-7dcbb4d56f-zqmrz 1/1 Running 0 66s 10.38.0.3 node1 <none> <none>
mysql-f4b58b5dd-kmqds 1/1 Running 0 18d 10.40.0.2 node2 <none> <none>
[root@master ~]#
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	


## 15 Cambiar versión deployment Drupal

Cambiamos en el archivo yaml de drupal la versión de drupal.

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: drupal
  labels:
    app: drupal
spec:
  selector:
    matchLabels:
      app: drupal
      tier: frontend
  replicas: 3
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: drupal
        tier: frontend
    spec:
      containers:
        - image: drupal:7
          name: drupal
          ports:
            - containerPort: 80
              name: drupal
          volumeMounts:
            - name: drupal-pvc
              mountPath: /home
      volumes:
        - name: drupal-pvc
          persistentVolumeClaim:
            claimName: nfs-pvc1
```

Ejecutamos el archivo modificado anteriormente y vemos el estado del deployment que como podemos ver está actualizando la imagen de los pods. Este actualiza pod a pod para que nunca haya indisponibilidad.

```
[root@master ~]# kubectl apply -f deployment_drupal.yaml
service/drupal unchanged
persistentvolumeclaim/nfs-pvc1 unchanged
deployment.apps/drupal configured
[root@master ~]# kubectl rollout status deployment drupal
Waiting for deployment "drupal" rollout to finish: 0 out of 3 new replicas have been updated...
Waiting for deployment "drupal" rollout to finish: 0 out of 3 new replicas have been updated...
Waiting for deployment "drupal" rollout to finish: 0 out of 3 new replicas have been updated...
Waiting for deployment "drupal" rollout to finish: 0 of 3 updated replicas are available...
deployment "drupal" successfully rolled out
```

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 16 Instalación de drupal

Para acceder al pod desde el exterior del clúster necesitamos ver el servicio NodePort creado anteriormente llamado drupal, y copiaremos el puerto que nos ofrece.

```
[root@master ~]# kubectl get svc
NAME         TYPE        CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
drupal       NodePort    10.105.179.202 <none>        80:30731/TCP    24m
kubernetes   ClusterIP   10.96.0.1     <none>         443/TCP          49d
mysql        ClusterIP   None          <none>         3306/TCP         22m
[root@master ~]#
```


Escribimos la Ip de alguna maquina donde se esté ejecutando el pod junto al puerto y accederemos.

Introduciremos los siguientes datos, estos datos los hemos creado en el deployment como variables de entorno.

No es seguro

192.168.1.91:30731/install.php?profile=standard&locale=en

Database configuration



✓ Choose profile

✓ Choose language

✓ Verify requirements

► Set up database

Install profile

Configure site

Finished

Database type \*

☒ MySQL, MariaDB, or equivalent
 ☐ PostgreSQL
 ☐ SQLite

The type of database your Drupal data will be stored in.

Database name \*

drupal

The name of the database your Drupal data will be stored in. It must exist on your server before Drupal can be installed.

Database username \*

jose

Database password

\*\*\*\*

▼ ADVANCED OPTIONS

These options are only necessary for some sites. If you're not sure what you should enter here, leave the default settings or check with your hosting provide

Database host \*

mysql

If your database is located on a different server, change this.


Database port

3306

If your database server is listening to a non-standard port, enter its number.

Table prefix

If more than one application will be sharing this database, enter a table prefix such as *drupal\_* for your Drupal site here.

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

No es seguro | 192.168.1.91:30731/install.php?profile=standard&locale=en

## Configurar sitio



- ✓ Elegir perfil
- ✓ Elige lengua
- ✓ Verificar requisitos
- ✓ Configurar base de datos
- ✓ Instalar perfil
- **Configurar sitio**
- Terminado

### INFORMACIÓN DEL SITIO

#### Nombre del sitio \*

Proyecto Jose Manuel Kubernetes

#### Dirección de correo electrónico del sitio \*

jose@gmail.com

Los correos electrónicos automatizados, como la información de registro, se enviarán desde esta dirección. Use una dirección que termine en el dominio de su sitio para ayudar a evitar que estos correos electrónicos se marquen como spam.

### CUENTA DE MANTENIMIENTO DEL SITIO

#### Nombre de usuario \*

jose

Se permiten espacios; la puntuación no está permitida excepto por puntos, guiones y guiones bajos.

#### Dirección de correo electrónico \*

jose@gmail.com

#### Contraseña \*

\*\*\*\*

Seguridad de la contraseña: **Débiles**

#### Confirmar contraseña \*

Para fortalecer su contraseña:

- Haz que tenga al menos 6 caracteres
- Añadir letras mayúsculas
- Agregar números
- Agrega puntuación
- Hazlo diferente de tu nombre de usuario

Cuando finalicemos la instalación veremos nuestra página Drupal directamente para usar.

Bienvenido a Proyecto Jose Man... x

No es seguro | 192.168.1.91:30731

Tablero Contenido Estructura Apariencia Personas Módulos Configuración Informes Ayuda

Agregar contenido Encontrar contenido

Mi cuenta Cerrar sesión

Proyecto Jose Manuel Kubernetes

Casa

Bienvenido a Proyecto Jose Manuel Kubernetes


Aún no se ha creado contenido de portada.

• [Agregar nuevo contenido](#)

Navegación

► [Agregar contenido](#)

Desarrollado por Drupal

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 17 Conclusiones

El objetivo de este proyecto es tener un CMS Drupal junto a MySQL, implementado en Kubernetes, para la creación de un servidor de alta disponibilidad.


La idea es que en todo momento el CMS esté disponible, aunque haya fallos internos como caída de pods, de máquinas o incluso una actualización.

El proyecto me ha resultado bastante complejo ya que Kubernetes presenta conceptos totalmente nuevos para mí y tienes que buscar bastante información, una vez vas trabajando sobre el te das cuenta de la facilidad que tiene para arreglar diferentes problemas que hace unos cuantos años eran impensables.

Pienso que he llegado a conseguir la mayoría de los objetivos, aunque a pequeña escala ya que solo teníamos 3 máquinas virtuales.

Dificultades encontradas:

- Cuando intentaba acceder a los contenedores creados daba un fallo de backend, este fallo se resolvía deshabilitando el cortafuegos en todos los nodos.
- Me daba fallo constante en la instalación de Drupal para que encontrara MySQL, lo resolví poniendo otro label que los diferenciara.
- Fallo en descarga de las imágenes de Docker debido a que guardaba el estado de las máquinas virtuales y cuando las encendía estas tenían la fecha atrasada, resolví el problema poniendo la fecha actual en cada máquina.

	Curso escolar: 2019/2020	Ciclo: CFGS de Administración de Sistemas Informáticos en Red	<b>PROYECTO DE KUBERNETES</b>
	Dpto: Informática	Módulo: Proyecto de Administración de sistemas informáticos en red.	

## 18 Bibliografía

[Deployment Drupal y Mysql](#)

[Servicios en Kubernetes](#)

[Persistent Volumes nfs](#)

[Dashboard Kubernetes](#)

[Cursos OpenWebinars sobre Kubernetes](#)

[Cambiar fecha Centos 7](#)

[NodeSelector](#)