

José Manuel Martínez Sánchez

Administración de Sistemas Informáticos en red

15-06-2020

KUBERNETES

**ÍNDICE**

[1 Descripción del proyecto 3](#_Toc43128636)

[2 Introducción teórica. 3](#_Toc43128637)

[3 Mapa de red 4](#_Toc43128638)

[4 Configuración de las máquinas 5](#_Toc43128639)

[4.1 Virtualbox 5](#_Toc43128640)

[4.1.1 Configuración de Master. 5](#_Toc43128641)

[4.1.2 Configuracion de Node 1 6](#_Toc43128642)

[4.1.3 Configuracion de Node 2 7](#_Toc43128643)

[4.2 Preparación del entorno 7](#_Toc43128644)

[4.2.1 Configuración de master. 8](#_Toc43128645)

[4.2.2 Configuración de Node 1 9](#_Toc43128646)

[4.2.3 Configuración de Node 2 10](#_Toc43128647)

[5 Instalación de Kubernetes y Docker 10](#_Toc43128648)

[5.1 Configuración de Master 11](#_Toc43128649)

[5.2 Configuración Nodo 1 14](#_Toc43128650)

[5.3 Configuración Nodo 2 16](#_Toc43128651)

[6 Instalación de Dashboard. 18](#_Toc43128652)

[7 Volúmenes 19](#_Toc43128653)

[7.1 Configuración Master. 20](#_Toc43128654)

[7.2 Configuración de nodos. 20](#_Toc43128655)

[8 Persistent Volumes 21](#_Toc43128656)

[9 Secrets 22](#_Toc43128657)

[10 Deployment drupal 23](#_Toc43128658)

[11 Deployment Mysql 25](#_Toc43128659)

[12 Volumes Nfs y hostpath 28](#_Toc43128660)

[13 Escalar deployment drupal 31](#_Toc43128661)

[14 Asignar pods a determinados nodos. 32](#_Toc43128662)

[15 Cambiar versión deployment Drupal 33](#_Toc43128663)

[16 Instalación de drupal 34](#_Toc43128664)

[17 Conclusiones 36](#_Toc43128665)

[18 Bibliografía 37](#_Toc43128666)

# 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.

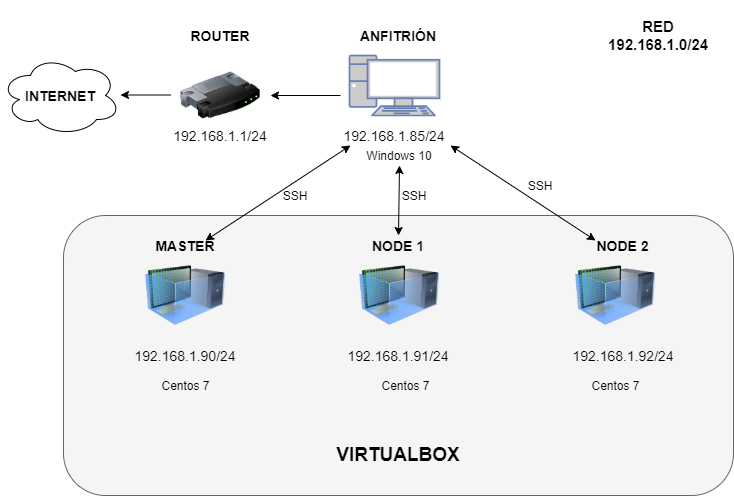
También crearemos el dashboard de Kubernetes para una vista más gráfica de nuestro clúster.

# 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.

# Mapa de red



8.8.8.8/8.8.4.4

8.8.8.8/8.8.4.4

8.8.8.8/8.8.4.4

# Configuración de las máquinas

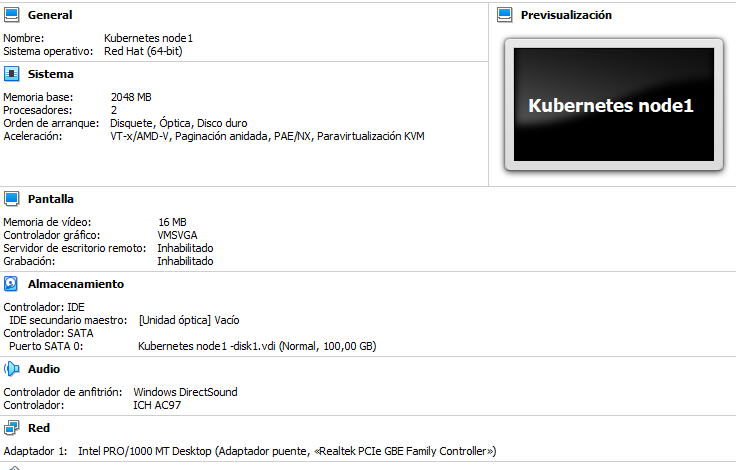
## Virtualbox

Las 3 máquinas tendrán la misma configuración:

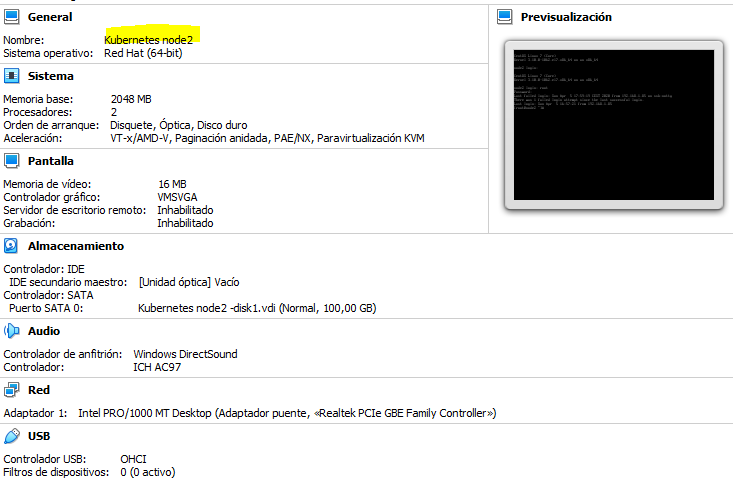
|  |  |  |  |
| --- | --- | --- | --- |
|  | MASTER | NODE 1 | NODE 2 |
| Red | Adaptador Puente | Adaptador Puente | Adaptador Puente |
| RAM | 2048 MB | 2048 MB | 2048 MB |
| Disco duro | 100 GB | 100 GB | 100 GB |
| Sistema operativo | Centos 7 | Centos 7 | Centos 7 |
| N.º procesadores | 2 | 2 |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

### Configuración de Master.

### Configuracion de Node 1



### Configuracion de Node 2

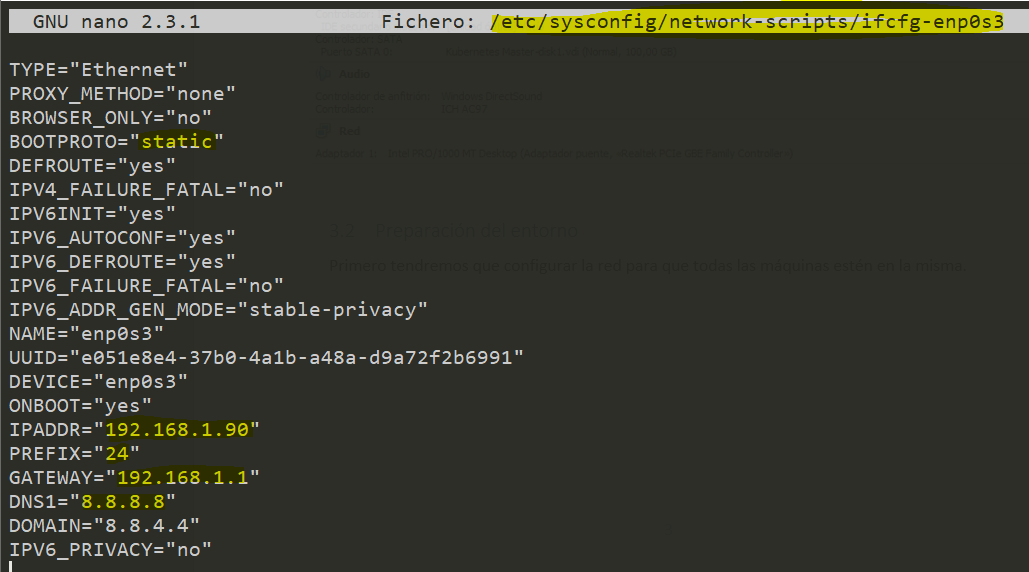


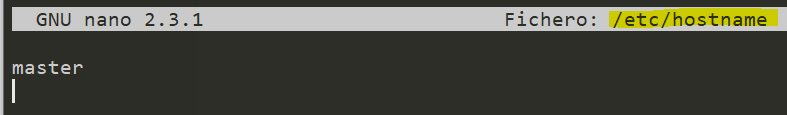
## Preparación del entorno

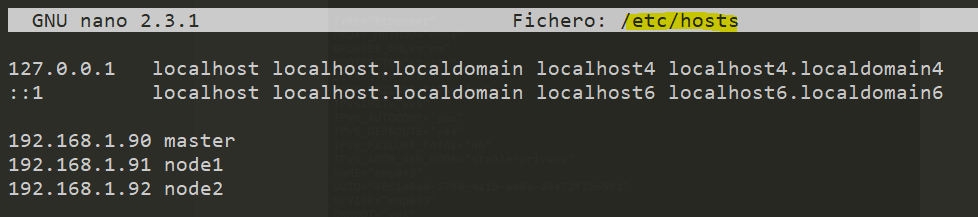
Primero tendremos que configurar la red para que todas las máquinas estén en la misma.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 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 |

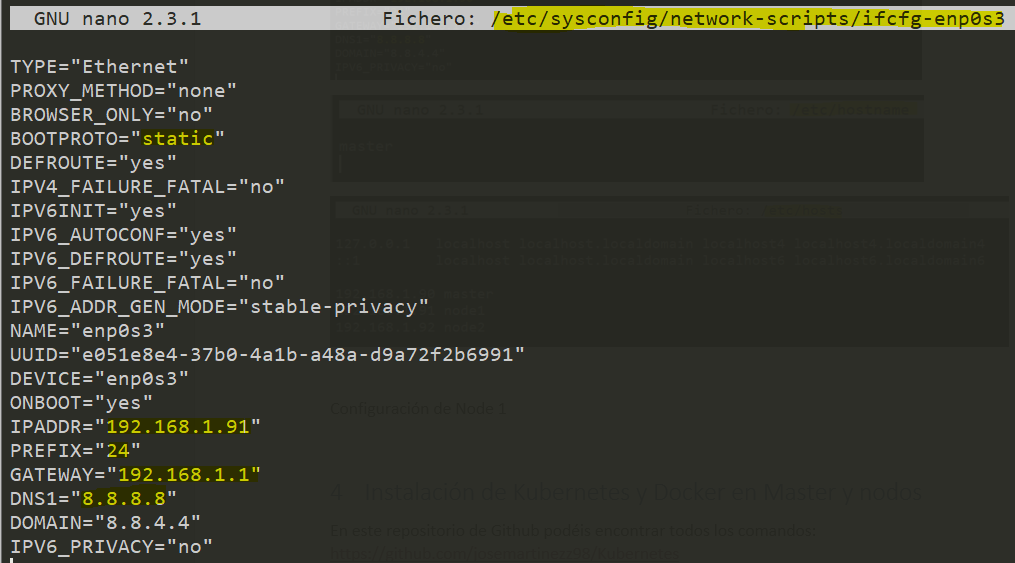
### Configuración de master.

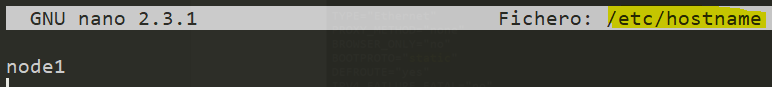


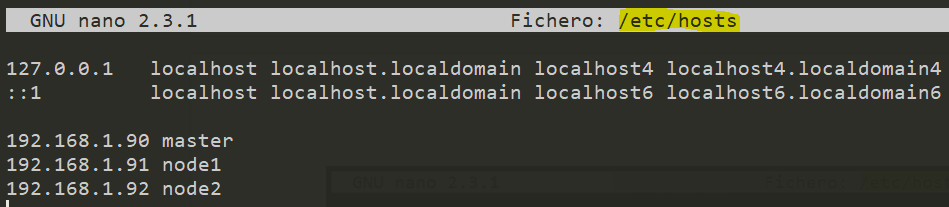




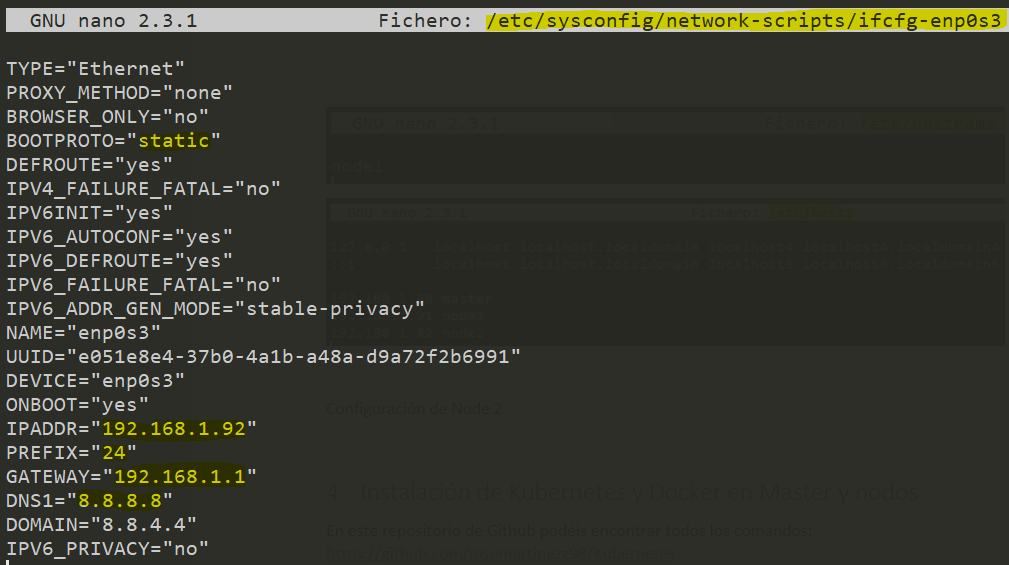
### Configuración de Node 1

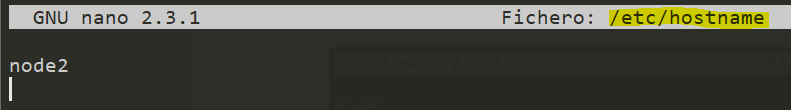


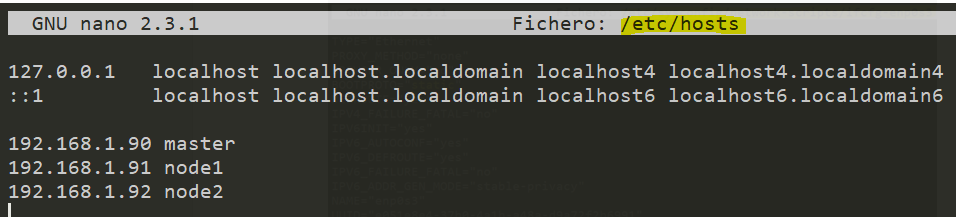




### Configuración de Node 2







# Instalación de Kubernetes y Docker

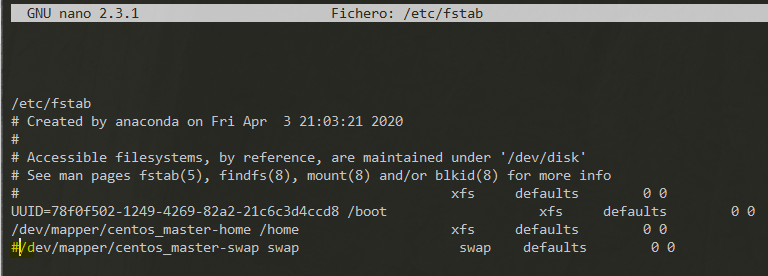
En este repositorio de GitHub podéis encontrar todos los comandos: <https://github.com/josemartinezz98/Kubernetes/blob/master/Comandos_Instalacion_Kubernetes.txt>

## Configuración de Master

Primero de todo vamos a deshabilitar la swap.

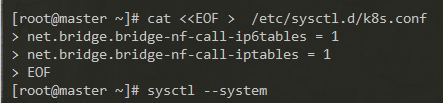


Para desactivarla totalmente en /etc/fstab comentamos la línea de swap.



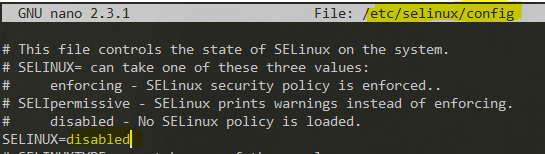
Para la instalación de Kubernetes necesitamos desactivar algunas reglas como desactivar Ipv6.

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



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

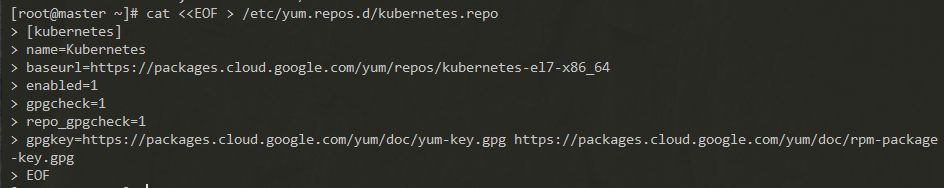




Instalamos Docker y lo habilitamos.





Creamos el fichero que apunta al repositorio para instalar Kubernetes.

Y a continuación instalamos y lo iniciamos.





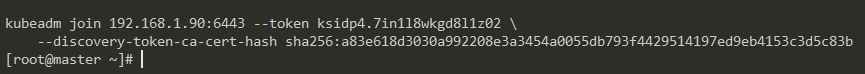
Desactivamos el cortafuegos porque después da fallos.



Inicializamos el clúster.



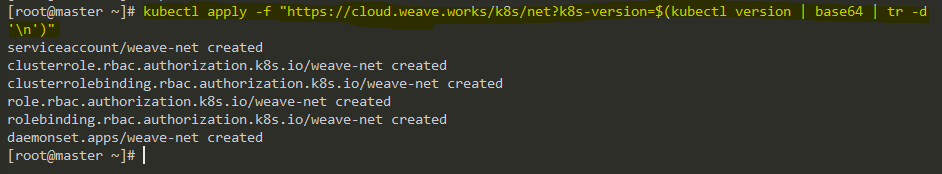
Nos vamos a guardar este comando que nos genera para poder unir el master con los otros nodos, este comando se tendrá que ejecutar en todas las máquinas que sean nodos.



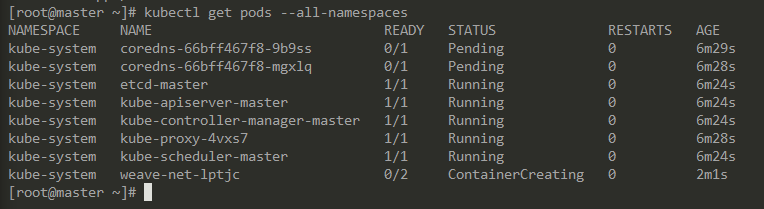
Para exportar la variable de entorno de kubeconfig.

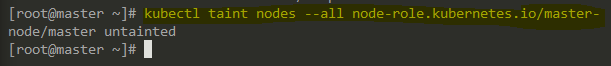


Vamos a ejecutar el siguiente comando para descargar Weave Net, una herramienta que creauna red virtual dentro de nuestro clúster para la configuración automática de red de nuestros pods.



Con este comando podemos ver los pods.



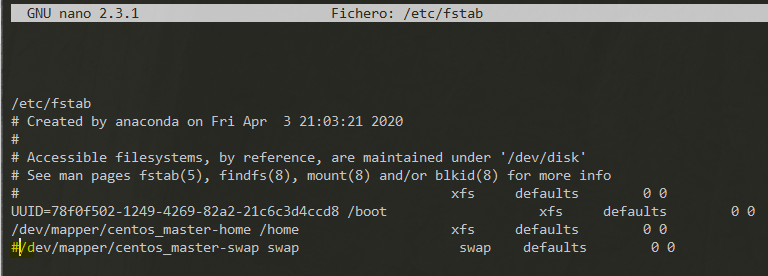
Normalmente Master no suele ejecutar pods, si lo desea hacer debemos ejecutar este comando.

## Configuración Nodo 1

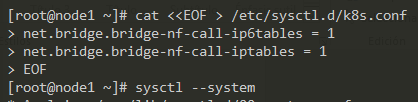
Deshabilitamos la swap



La deshabilitamos permanentemente



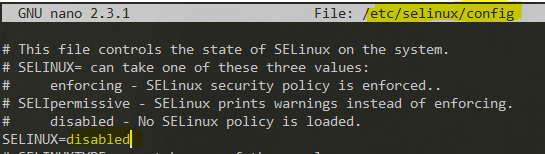
Desactivamos las reglas de Ipv6



Desactivamos Selinux



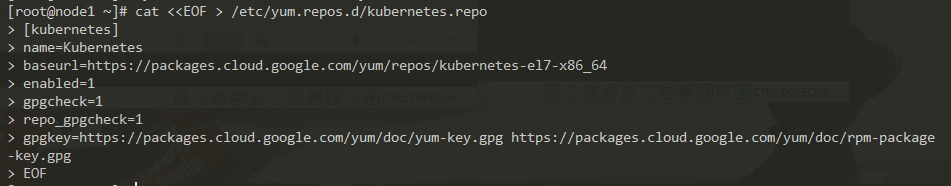
Desactivamos selinux permanentemente en el archivo /etc/selinux/config



Instalamos e iniciamos Docker





Introducimos repositorio para kubernetes.

Instalamos kubelet y lo iniciamos



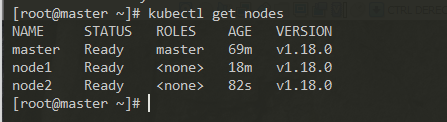


Deshabilitamos y paramos el firewalld para los posibles errores.



Comando para la unión del nodo 1 al clúster

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

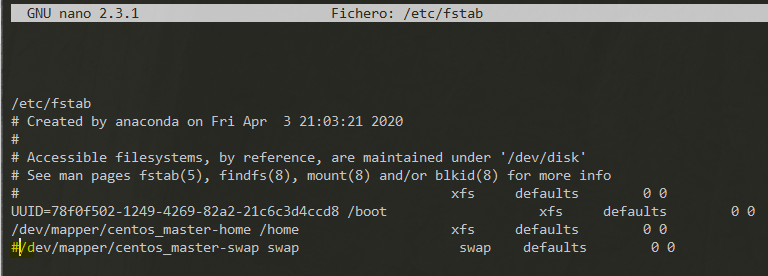


## Configuración Nodo 2

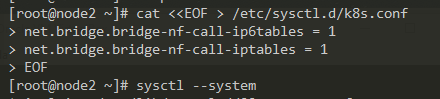
Deshabilitamos la swap



La deshabilitamos permanentemente

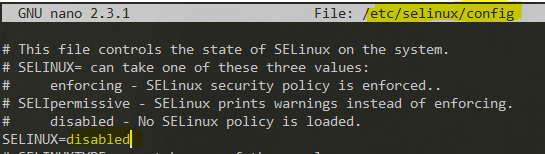


Desactivamos las reglas de Ipv6



Desactivamos Selinux con setenforce y en el fichero para consolidar

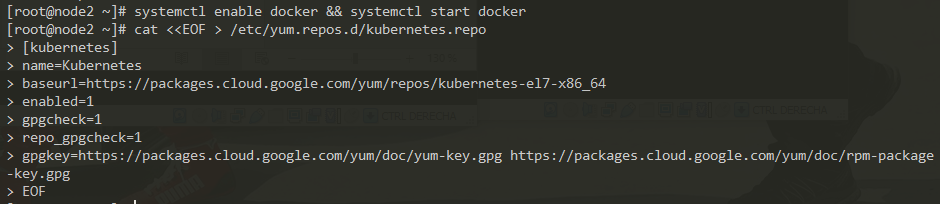




Instalamos e iniciamos docker





Introducimos repositorio para kubernetes

Instalamos kubelet y lo activamos



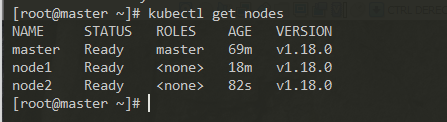


Desactivamos el firewall para posibles problemas



Unimos el nodo 2 al clúster

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



# Instalación de Dashboard.

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

Desplegamos el yaml que crea el dashboard.

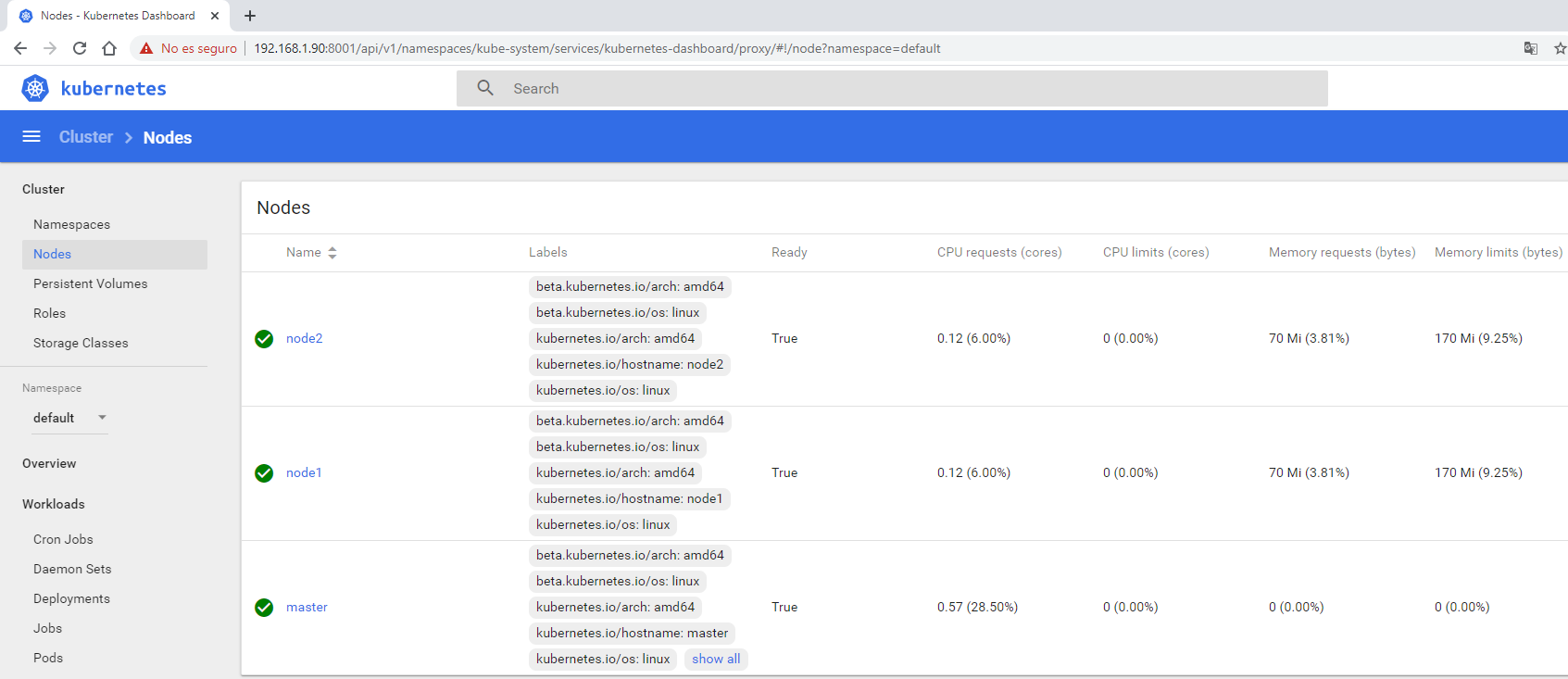
Damos los permisos necesarios para poder acceder a él.

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



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

# Volúmenes

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

## Configuración Master.

Instalamos Nfs



Creamos el directorio que vamos a compartir



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



Comprobamos que está compartido

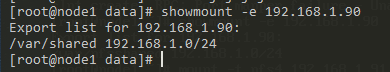


## Configuración de nodos.

Instalamos Nfs



Comprobamos que podemos ver la carpeta compartida



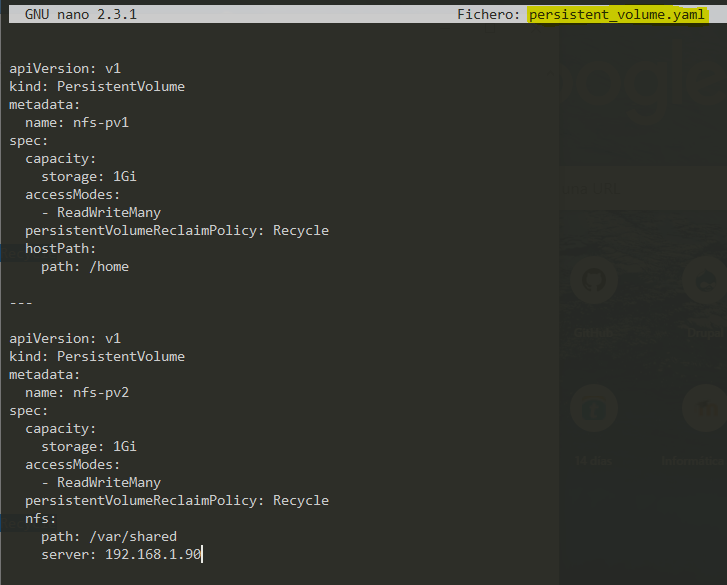
Montamos la carpeta compartida en una carpeta de nuestra máquina en este caso /var/data



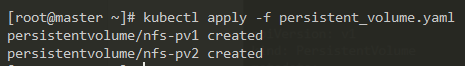
# Persistent Volumes

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

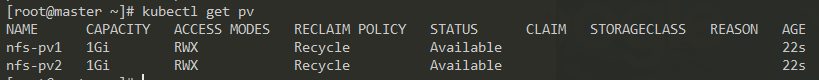
Creamos dos Persistent Volumes para mantener información de forma persistente de los pods.



Ejecutamos el yaml creado anteriormente



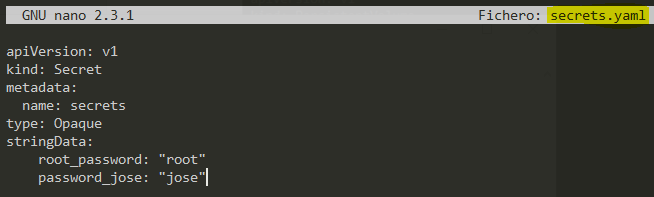
Comprobamos que están disponibles para ser utilizados



# Secrets

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

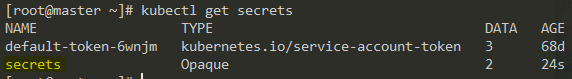
Vamos a crear el objeto secret para crear variables de entorno que no estén directamente en texto plano.



Ejecutamos el yaml y vemos que se crea.



Comprobamos que se ha creado correctamente.



# Deployment drupal

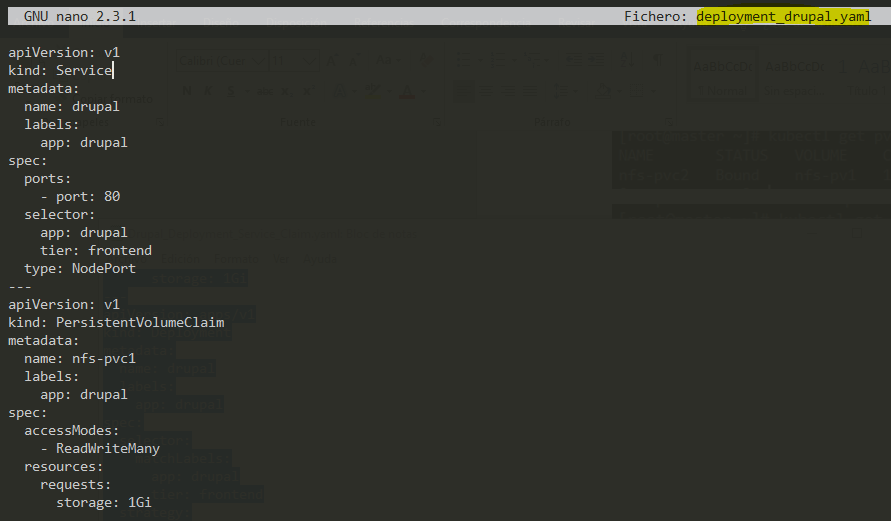
Archivo yaml: <https://github.com/josemartinezz98/Kubernetes/blob/master/Drupal_Deployment_Service_Claim.yaml>

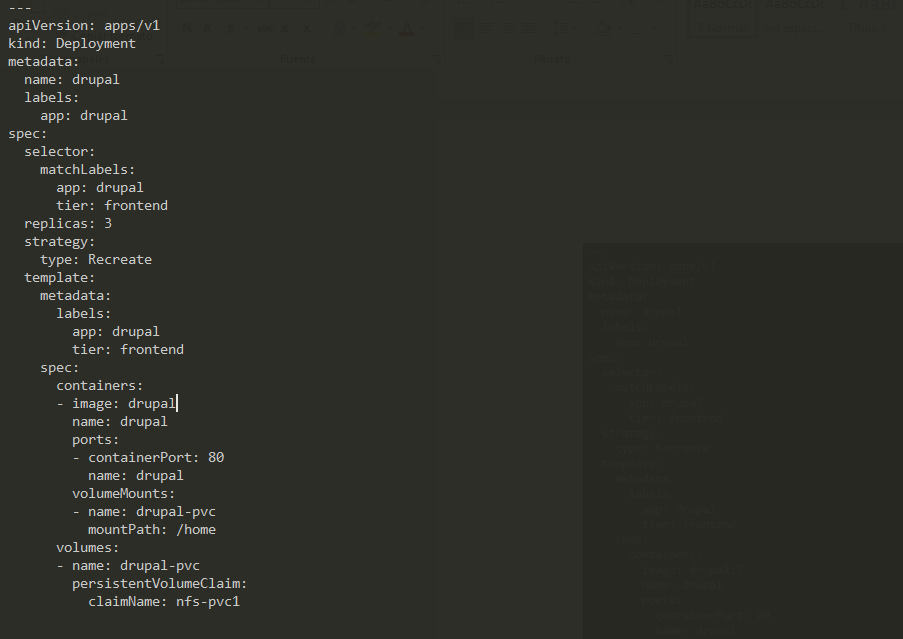
Este yaml permite crear el cms drupal lo implementaremos con los objetos Service, Persistent Volume Claim y Deployment:

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

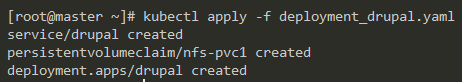
**Persistent Volume Claim** 🡪 Es una reclamación de almacenamiento para los datos de estos pods, esto reclamará el espacio a un Persistent Volume creado anteriormente.

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



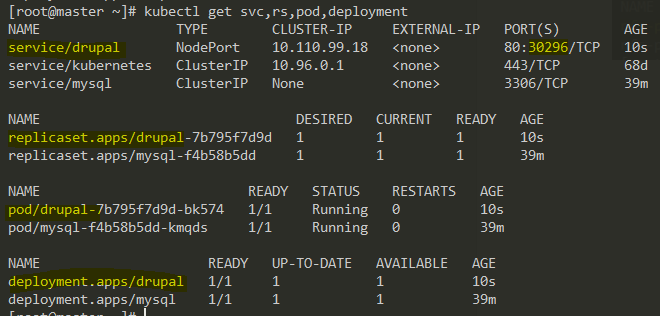


Ejecutamos el archivo yaml para su creación.



Comprobamos que la reclamación de volumen persistente ha sido correcta.



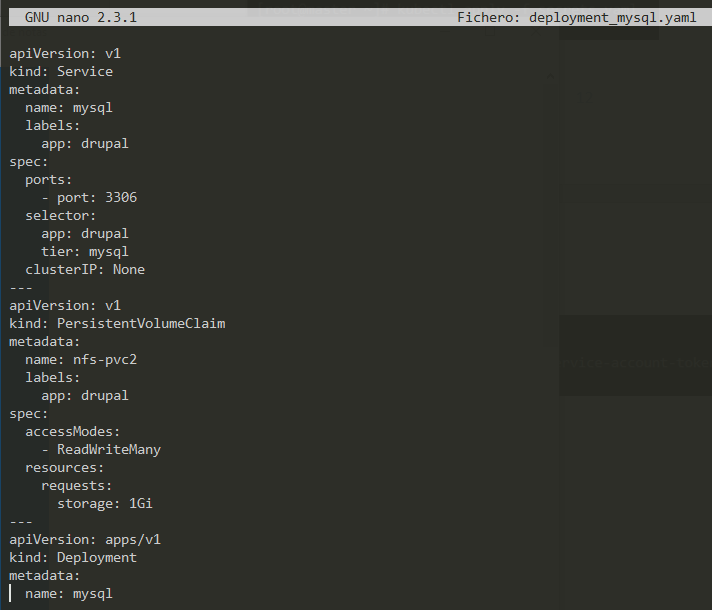
Vemos los diferentes objetos creados por el archivo yaml.

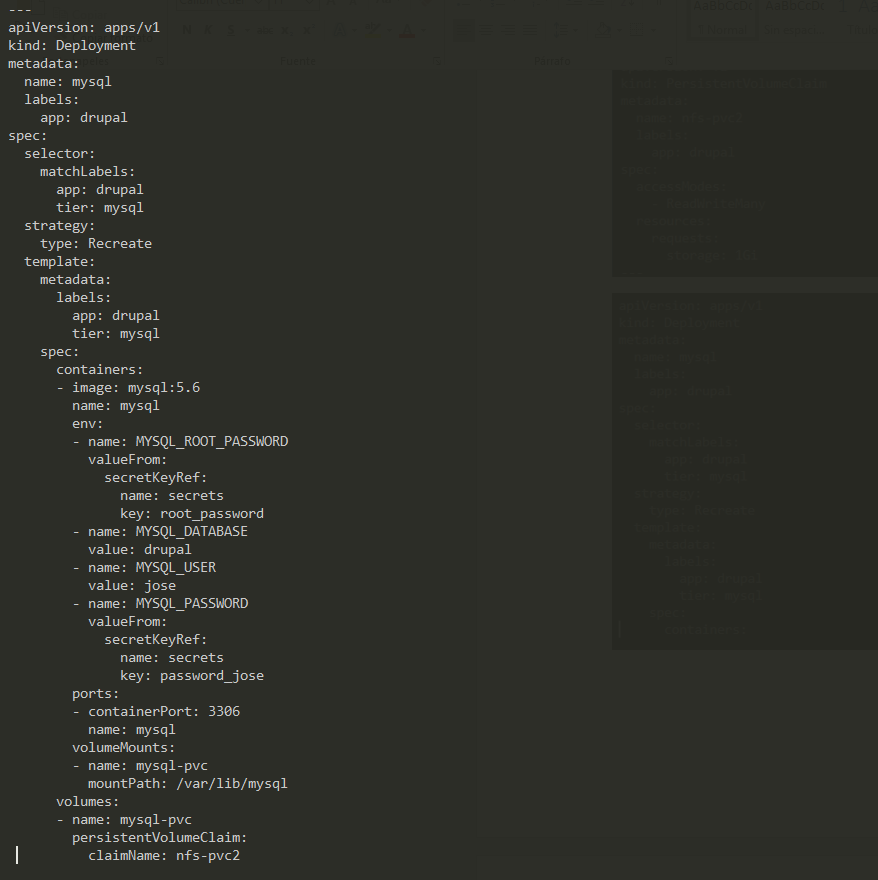
# Deployment Mysql

Archivo yaml:

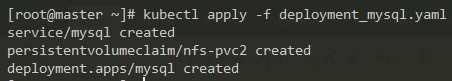
<https://github.com/josemartinezz98/Kubernetes/blob/master/Drupal_Deployment_Service_Claim.yaml>

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.

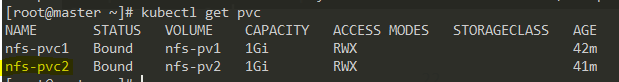




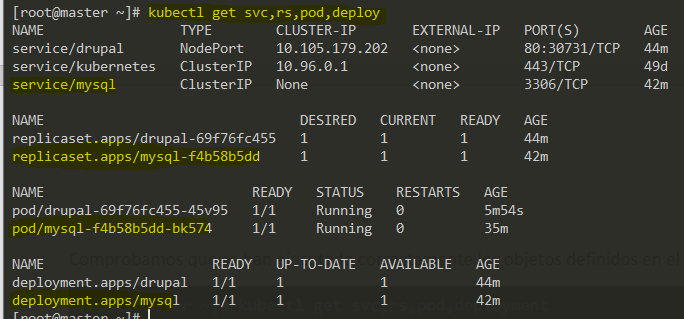
Ejecutamos el archivo yaml de MySQL



Vemos que se ha unido a un Persistent Volume.



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

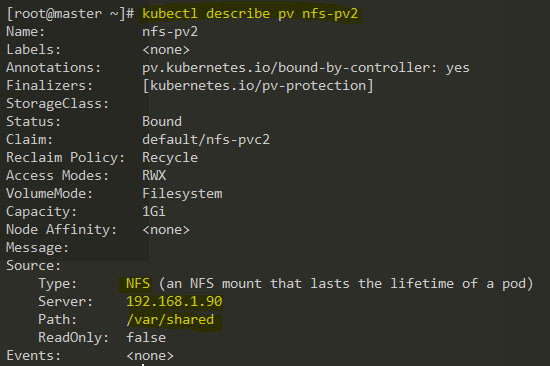


# Volumes Nfs y hostpath

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.



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

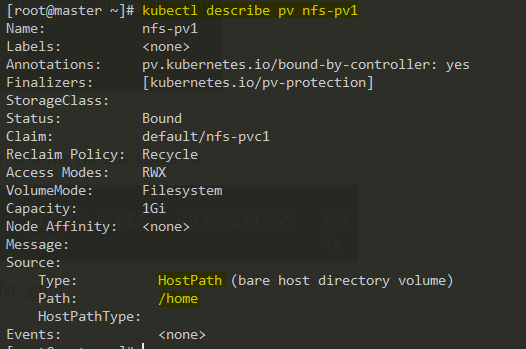
Maquina Master



* Pod Mysql



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



Creamos una carpeta en el directorio compartido



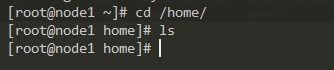
Vemos que se está ejecutando en el nodo 2



Solo comparte con el nodo 2



Vemos que en el nodo 1 no se ha compartido



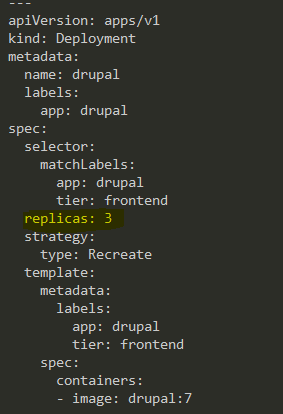
# Escalar deployment drupal

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

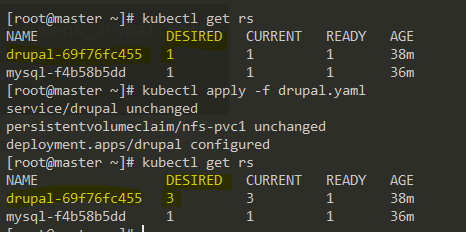
1. Indicando en el comando el nombre del deployment y cuantas replicas queremos



1. Modificamos el archivo yaml e introducimos la línea “replicas” y escribimos cuantas replicas querríamos.

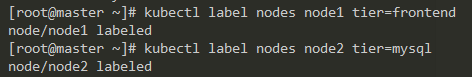


Ejecutamos el archivo yaml modificado y vemos que antes era 1 deseado y ahora son 3.

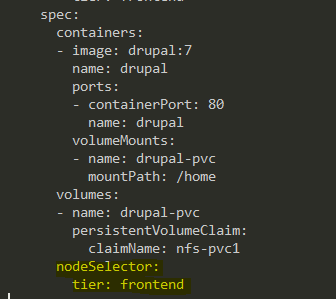


# Asignar pods a determinados nodos.

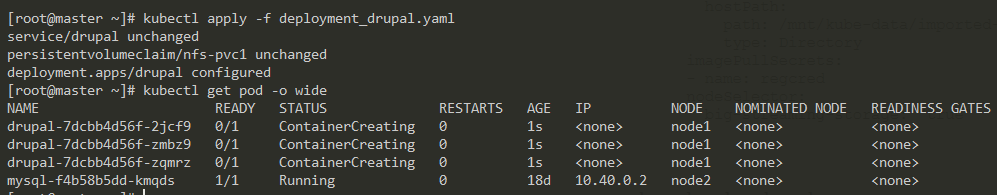
Introduciremos unos labels para identificar a los nodos



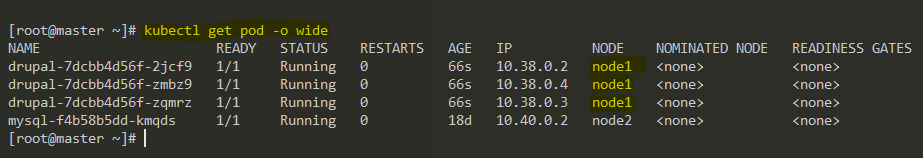
Añadiremos el label creado al deployment con la línea “nodeSelector”



Ejecutamos el archivo yaml

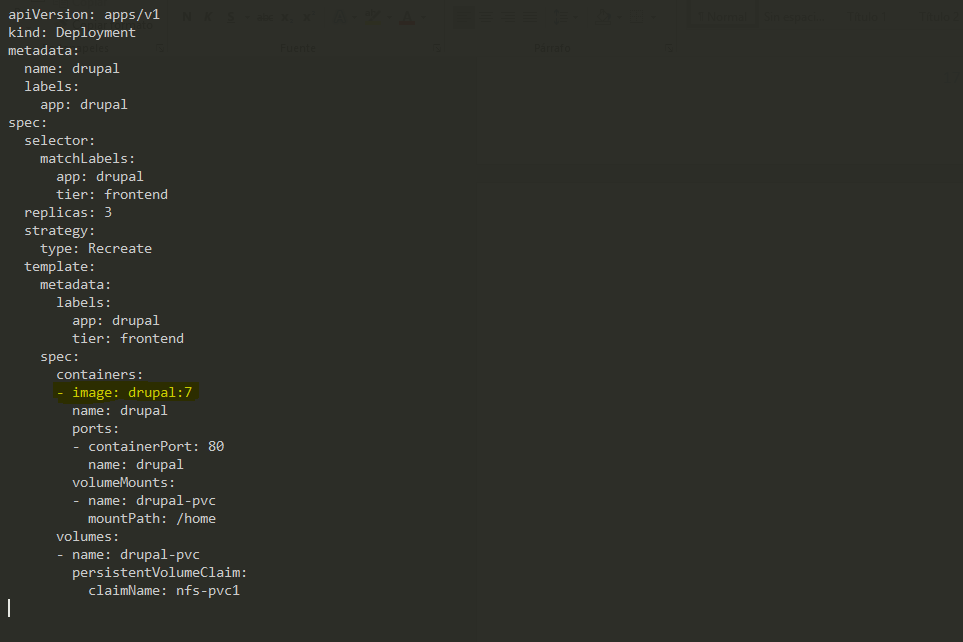


Vemos que todos los pods de ese deployment se han cambiado al nodo 1

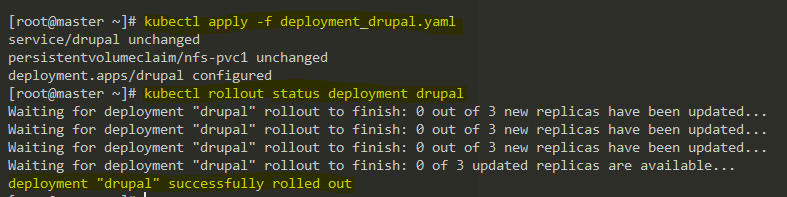


# Cambiar versión deployment Drupal

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

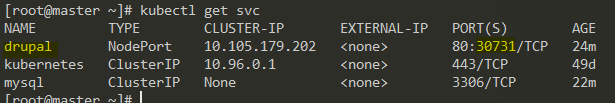


Ejecutamos el archivo modificado anteriormente y vemos el estado del deployment que como podemos ver está actualizando la imagen de los pods. Este actualiza pod a pod para que nunca haya indisponibilidad.



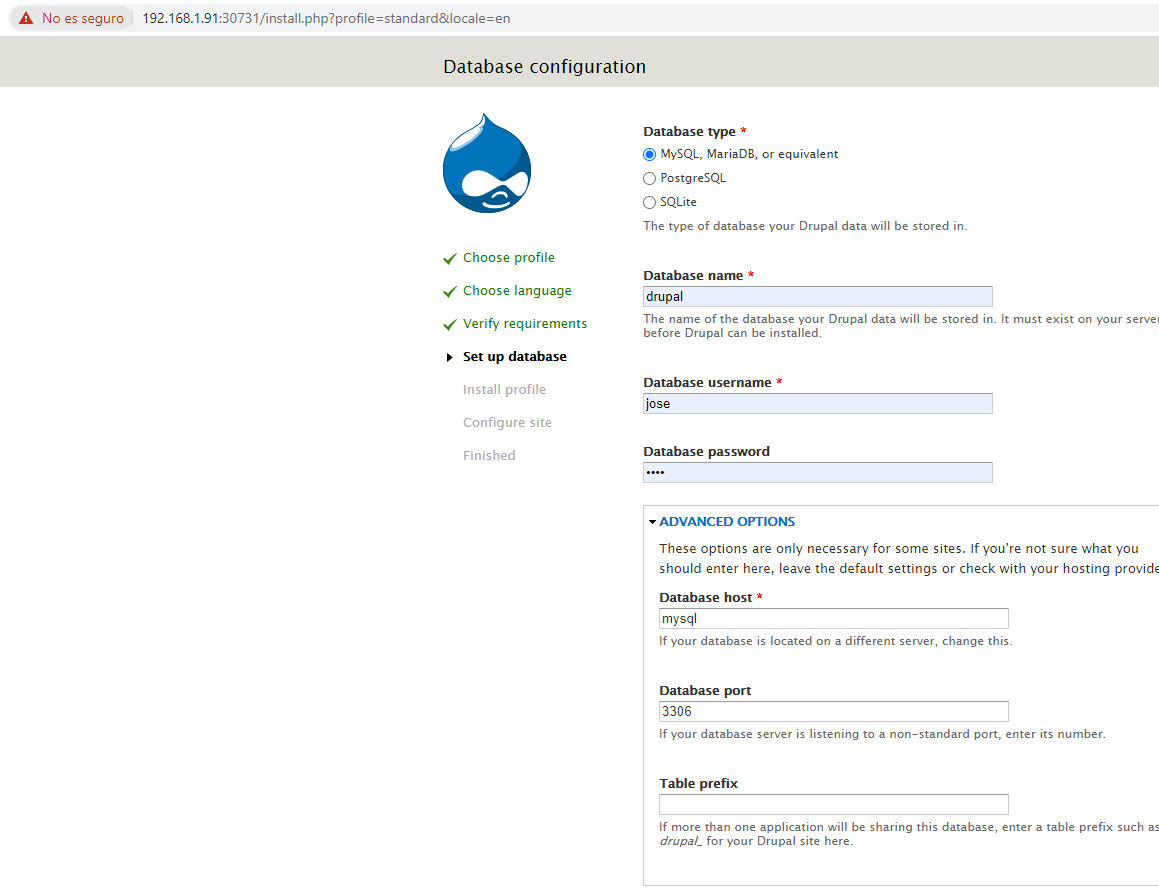
# Instalación de drupal

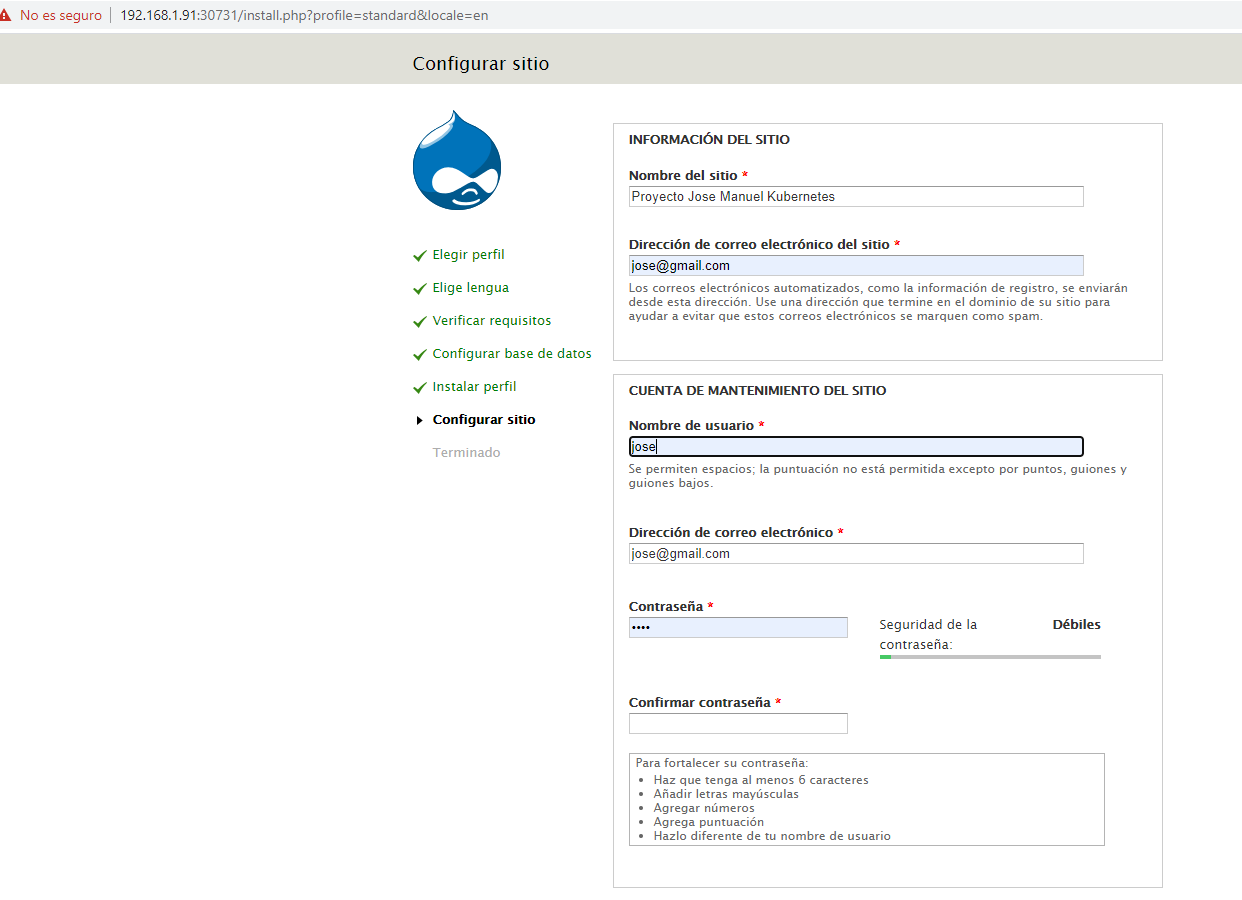
Para acceder al pod desde el exterior del clúster necesitamos ver el servicio NodePort creado anteriormente llamado drupal, y copiaremos el puerto que nos ofrece.



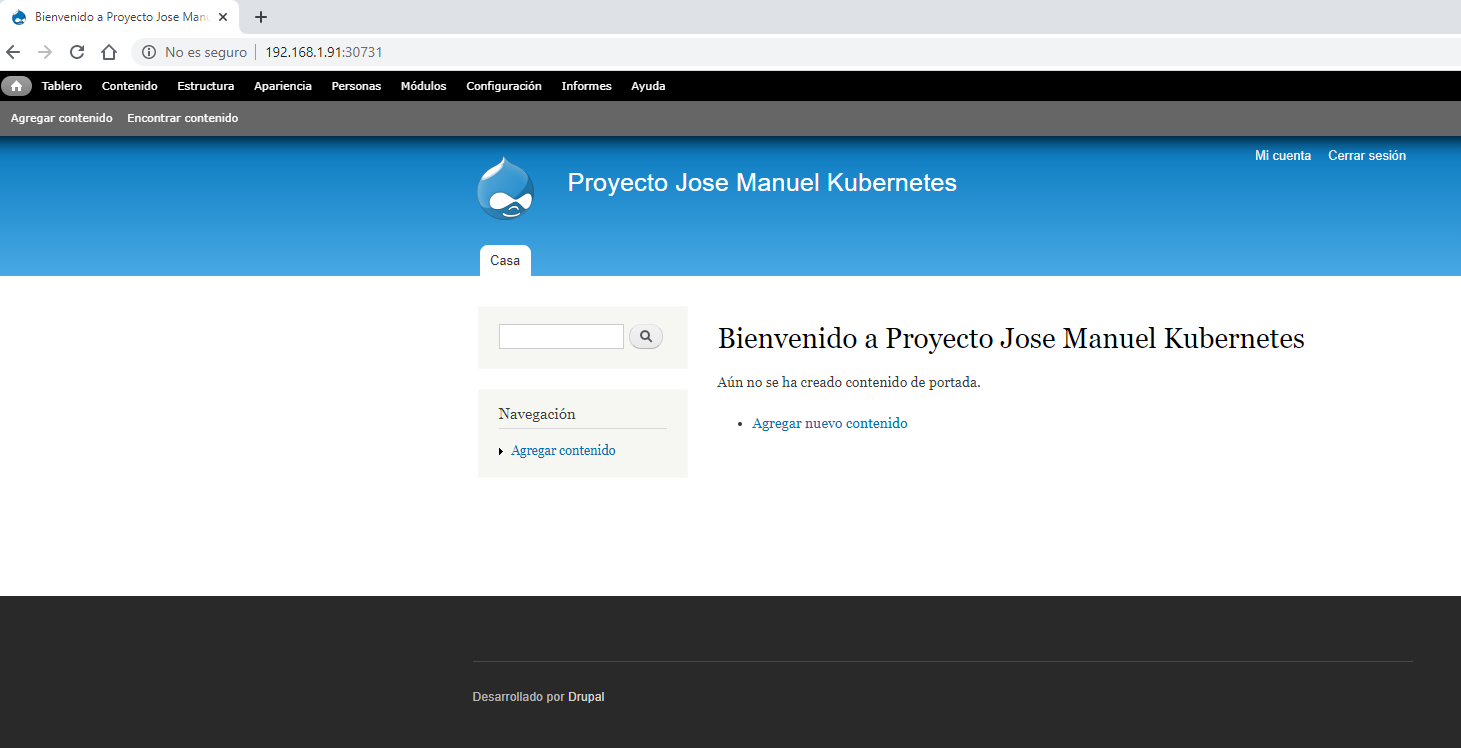
Escribimos la Ip de alguna maquina donde se esté ejecutando el pod junto al puerto y accederemos.

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





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



# 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.

# Bibliografía

[Deployment Drupal y Mysql](https://kubernetes.io/docs/tutorials/stateful-application/mysql-wordpress-persistent-volume/)

[Servicios en Kubernetes](https://www.josedomingo.org/pledin/2018/11/recursos-de-kubernetes-services/)

[Persistent Volumes nfs](https://www.josedomingo.org/pledin/2019/03/almacenamiento-kubernetes/)

[Dashboard Kubernetes](https://www.returngis.net/2019/03/instalar-kubernetes-dashboard-en-tu-cluster/)

[Cursos OpenWebinars sobre Kubernetes](https://openwebinars.net/)

[Cambiar fecha Centos 7](https://www.linux-party.com/35-linux/1732-cambiar-la-hora-y-la-fecha-al-sistema-linux)

[NodeSelector](https://medium.com/@jmarhee/using-nodeselector-to-schedule-deployments-with-large-volumes-of-stateful-data-on-kubernetes-46bd3ac6059d)