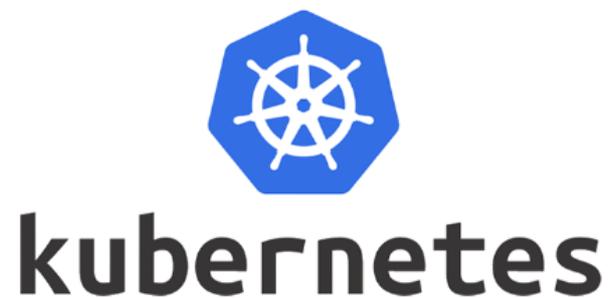


***CREAR Y ADMINISTRAR CLUSTER  
DE KUBERNETES CON RANCHER***



*Jesús Arias*  
*Administración de Sistemas Informáticos en Red*

# Sumario

|                                       |    |
|---------------------------------------|----|
| 1.Introducción.....                   | 3  |
| 1.1.Tecnologías.....                  | 3  |
| 2.Instalación de Rancher.....         | 4  |
| 3.Cluster con maquinas virtuales..... | 5  |
| 3.1.Crear cluster.....                | 5  |
| 3.2.Despliegue app.....               | 9  |
| 4.Cluster en OpenStack.....           | 12 |
| 5.Conclusión.....                     | 16 |
| 6.Trabajo futuro.....                 | 16 |
| 7.Referencias.....                    | 16 |

## 1. Introducción

En este proyecto la principal función es desplegar y administrar cluster de kubernetes con la herramienta rancher, realizare despliegues sobre maquinas virtuales locales, OpenStack y Amazon EC2.

Rancher es una plataforma para la administración de cluster de kubernetes, nos permite desplegar y administrar clusters de kubernetes en varios proveedores, Amazon, Azure, Google, OpenStack, etc.

Tenemos que tener en cuenta que rancher y rancheros no es lo mismo pero pueden ir juntos, rancheros es el sistema operativo orientado a trabajar con Docker, y rancher es la plataforma con la que trabajaremos para lo dicho anteriormente.

### 1.1. Tecnologías

**OpenStack:** Es un proyecto de computación en la nube para proporcionar una infraestructura como servicio (IaaS).

**Amazon EC2:** Amazon Elastic Compute Cloud (Amazon EC2) proporciona capacidad de computación escalable en la nube de Amazon Web Services (AWS). El uso de Amazon EC2 elimina la necesidad de invertir inicialmente en hardware, de manera que puede desarrollar e implementar aplicaciones en menos tiempo. Puede usar Amazon EC2 para lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes y administrar el almacenamiento.

**Kubernetes:** Es un sistema de código libre para la automatización del despliegue, ajuste de escala y manejo de aplicaciones en contenedores que fue originalmente diseñado por Google y donado a la Cloud Native Computing Foundation (parte de la Linux Foundation).

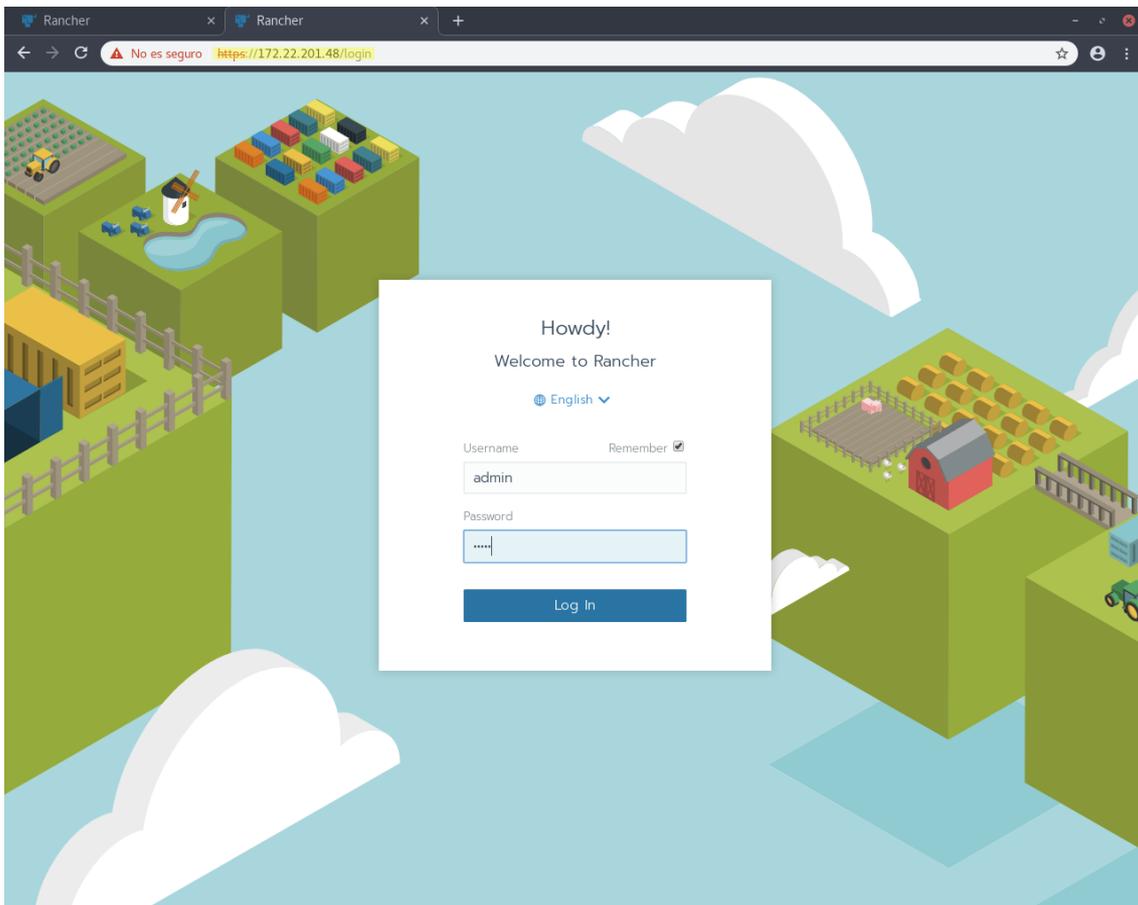
## 2. Instalación de Rancher

La instalación de Rancher es sencilla, lo primero que nos dice que tengamos una maquina Linux con al menos 4GB de memoria y con Docker instalado.

Para instalarlo ejecutamos en la maquina el siguiente comando:

```
$ sudo docker run -d --restart=unless-stopped \
  -p 80:80 -p 443:443 rancher/rancher
```

Una vez que tenemos el contenedor corriendo, ya podemos acceder a rancher vía web.



## 3. Cluster con maquinas virtuales

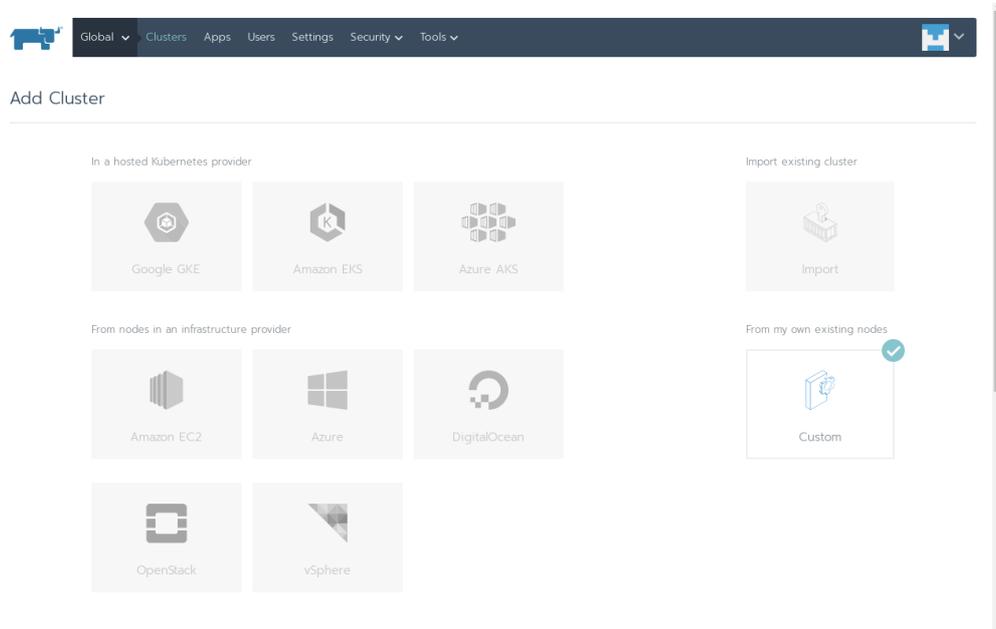
### 3.1. Crear cluster

En este escenario he creado 3 maquinas virtuales con el sistema operativo Ubuntu 16.04, donde tendremos un master y dos nodos worker.

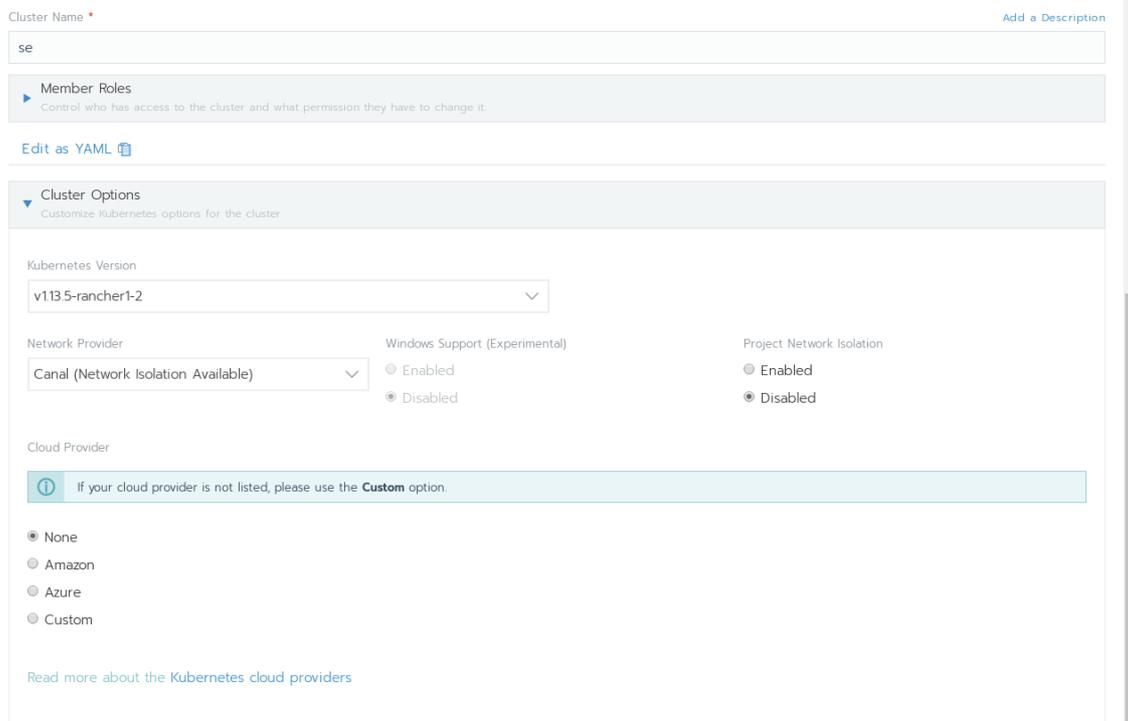
Necesitaremos tener docker instalado en las 3 maquinas, procedimiento de instalación:

```
$ sudo apt-get update
$ sudo apt-key adv --keyserver \
hkp://p80.pool.sks-keyservers.net:80 \
--recv-keys 58118E89F3A912897C070ADBF76221572C52609D
$ sudo apt-add-repository \
'deb https://apt.dockerproject.org/repo ubuntu-xenial main'
$ sudo apt-get update
$ sudo apt-cache policy docker-engine
$ sudo apt-get install -y docker-engine
```

Lo siguiente que hacemos iniciamos rancher y accedemos al panel web, una vez ahí le damos a añadir cluster, y por ultimo elegimos la opción custom.



Los siguientes parámetros son las opciones del cluster



Kubernetes tiene varios proveedores de red CNI (Container Network Interfaces), los principales son:

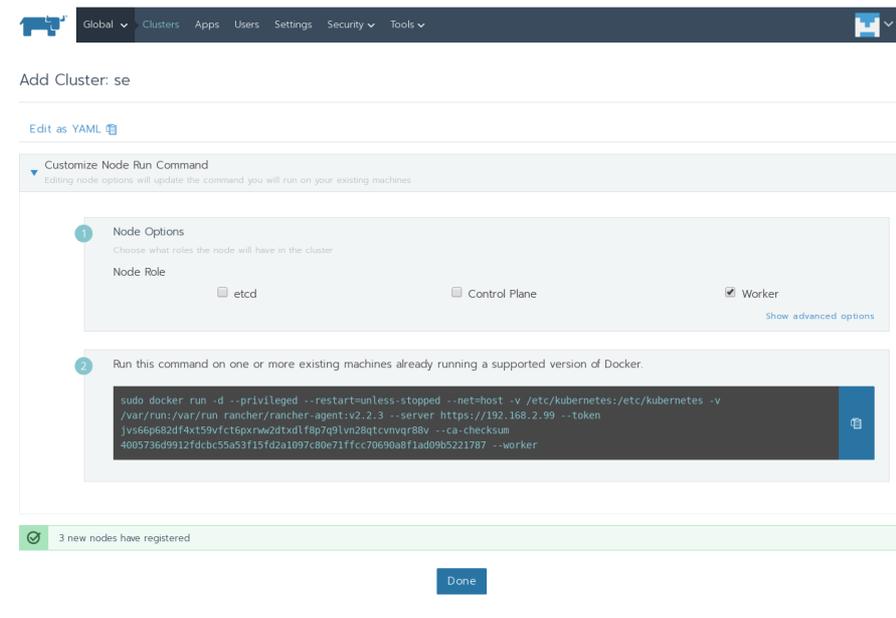
**Flannel**, un proyecto desarrollado por CoreOS, es quizás el complemento CNI más sencillo y popular disponible. Es uno de los ejemplos más maduros de tejido de redes para sistemas de orquestación de contenedores, destinado a permitir una mejor interconexión entre contenedores e inter-host.

**Calico**, es conocido por su rendimiento, flexibilidad y potencia. Calico tiene una visión más holística de las redes, preocupándose no solo por proporcionar conectividad de red entre hosts y pods, sino también por la administración y seguridad de la red.

**Canal**, es una combinación de Flannel y Calico

**Weave** crea una red de superposición de malla entre cada uno de los nodos del clúster, lo que permite un enrutamiento flexible entre los participantes.

El siguiente paso es añadir los nodos, elegimos el rol que va a tener el nodo y copiamos y pegamos el comando en las maquinas.



The screenshot shows the Rancher UI interface for adding a cluster. The top navigation bar includes 'Global', 'Clusters', 'Apps', 'Users', 'Settings', 'Security', and 'Tools'. Below the navigation, the page title is 'Add Cluster: se'. There is a link to 'Edit as YAML'. The main section is titled 'Customize Node Run Command' and contains two steps:

- Node Options**: Choose what roles the node will have in the cluster. Under 'Node Role', there are three checkboxes: 'etcd' (unchecked), 'Control Plane' (unchecked), and 'Worker' (checked). A link 'Show advanced options' is visible.
- Run this command on one or more existing machines already running a supported version of Docker.**: A terminal window displays the following command:

```
sudo docker run -d --privileged --restart=unless-stopped --net=host -v /etc/kubernetes:/etc/kubernetes -v /var/run:/var/run rancher/rancher-agent:v2.2.3 --server https://192.168.2.99 --token jvs6p883df4kt59vfct6pxrw2dtxdlf8p7q9lvm28atcvnvr88x --ca-checksum 4005736d9912fdcbc55a53f15f42a1097c80e71ffc70690a8f1ad09b5221787 --worker
```

At the bottom, a green status bar indicates '3 new nodes have registered' with a checkmark icon. A 'Done' button is located below the status bar.

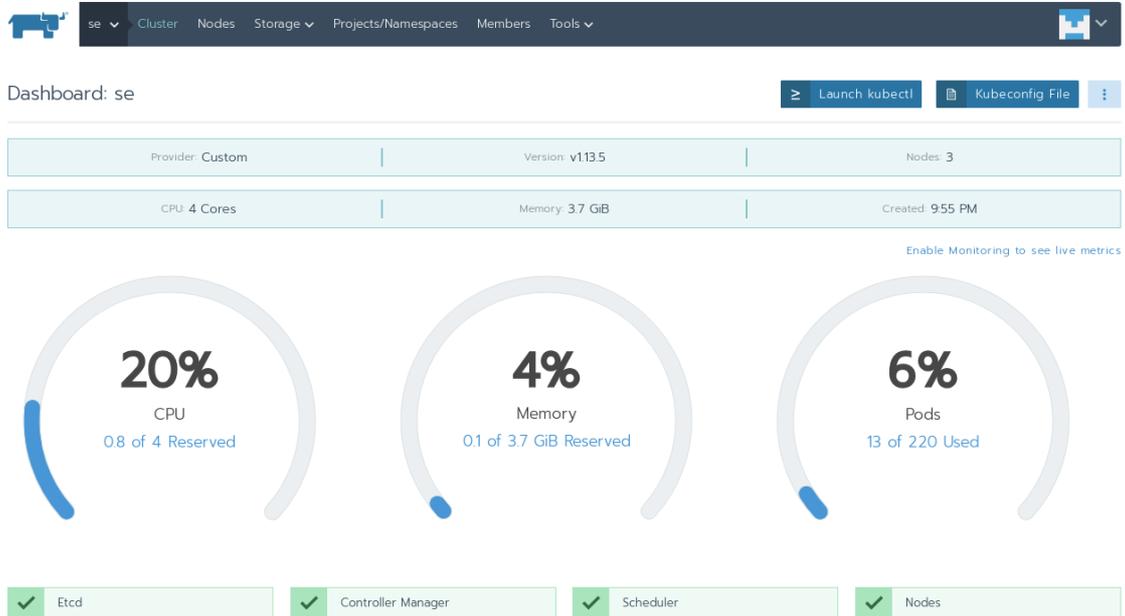
Comando para nodo master:

```
$ sudo docker run -d --privileged --restart=unless-stopped
--net=host -v /etc/kubernetes:/etc/kubernetes -v
/var/run:/var/run rancher/rancher-agent:v2.2.3 --server
https://192.168.2.99 --token
jvs66p682df4xt59vfct6pxrww2dtxdlf8p7q9lvn28qtcvqvqr88v --ca-
checksum
4005736d9912fdcbc55a53f15fd2a1097c80e71ffcc70690a8f1ad09b5221787
--etcd --controlplane
```

Comando para nodos worker:

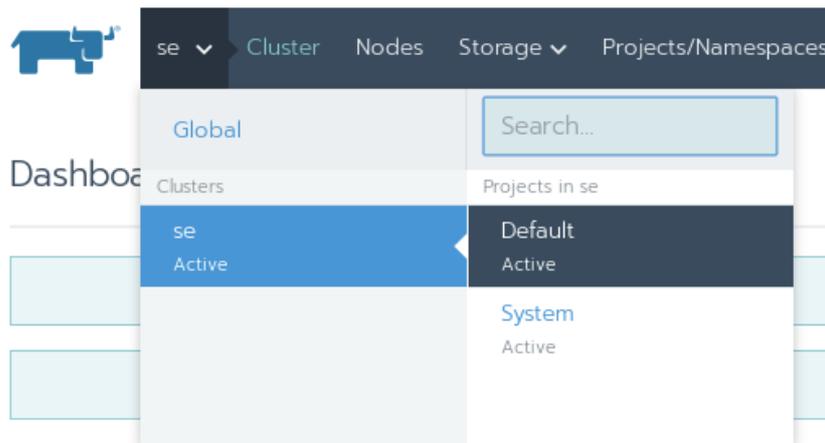
```
$ sudo docker run -d --privileged --restart=unless-stopped
--net=host -v /etc/kubernetes:/etc/kubernetes -v
/var/run:/var/run rancher/rancher-agent:v2.2.3 --server
https://192.168.2.99 --token
jvs66p682df4xt59vfct6pxrww2dtxdlf8p7q9lvn28qtcvqvqr88v --ca-
checksum
4005736d9912fdcbc55a53f15fd2a1097c80e71ffcc70690a8f1ad09b5221787
--worker
```

Por ultimo, empezara a crear el cluster y al cabo de unos minutos estará disponible.

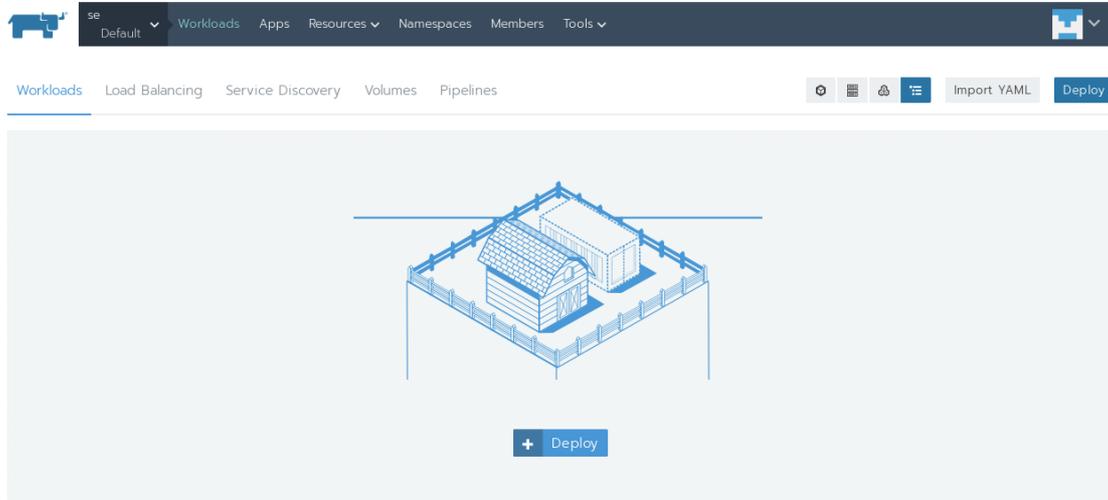


### 3.2. Despliegue app

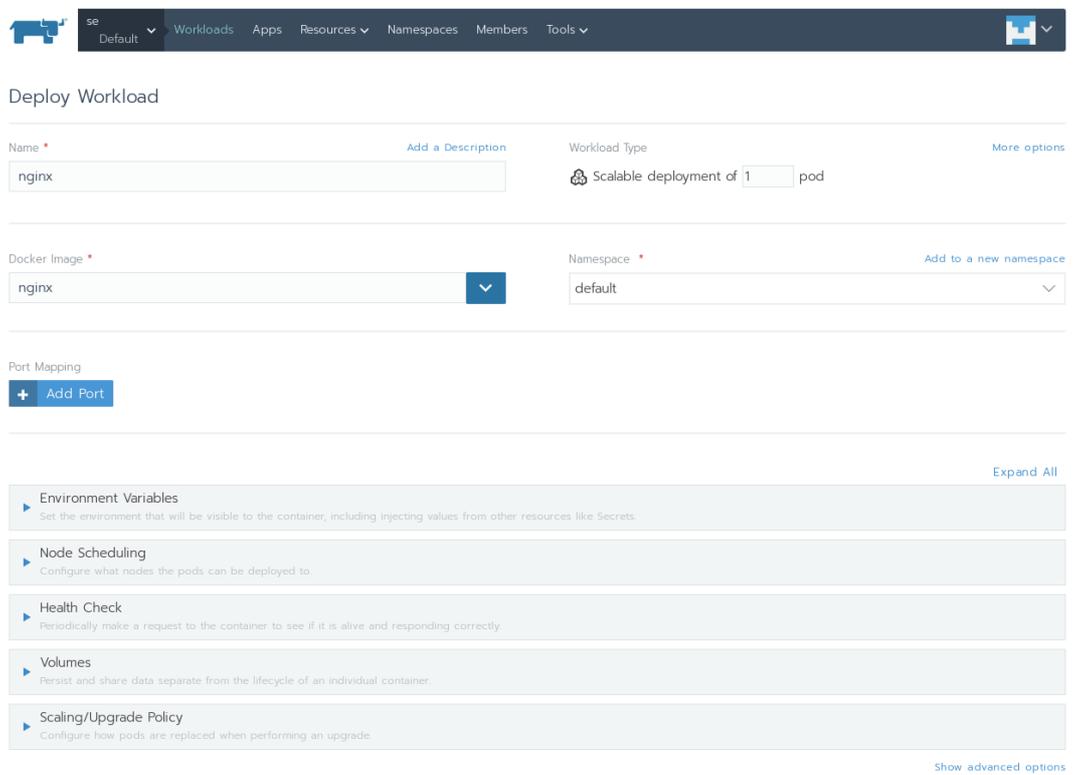
Vamos a desplegar un nginx en nuestro cluster, nos dirigimos nuestro proyecto por defecto que es default



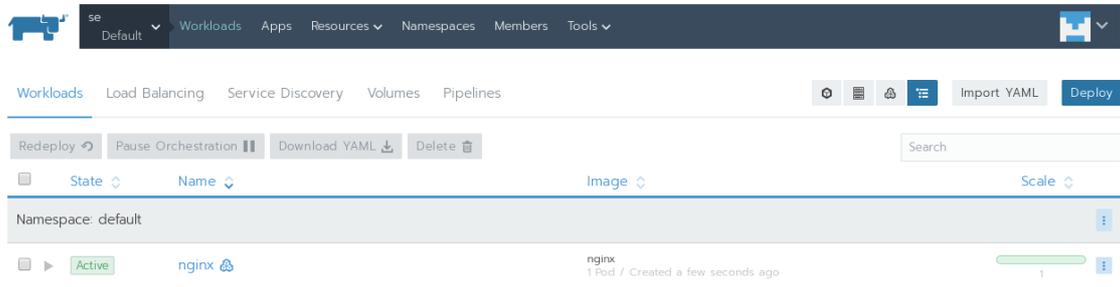
En la siguiente imagen se puede ver que en la parte derecha se puede importar un fichero yaml, en este caso se va a desplegar dándole a Deploy.



Se elige la imagen de nginx.



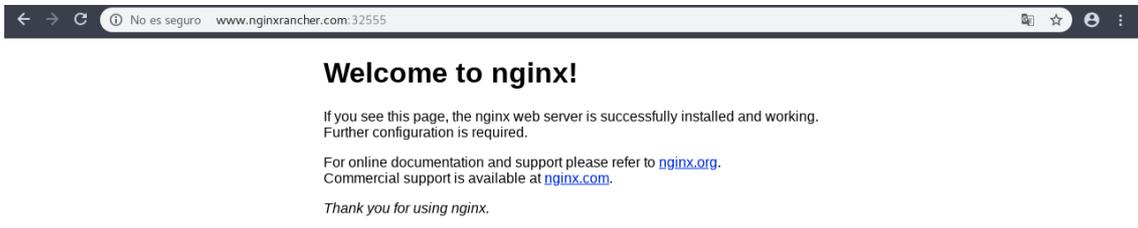
Se despliega, y ya estaría funcionando.



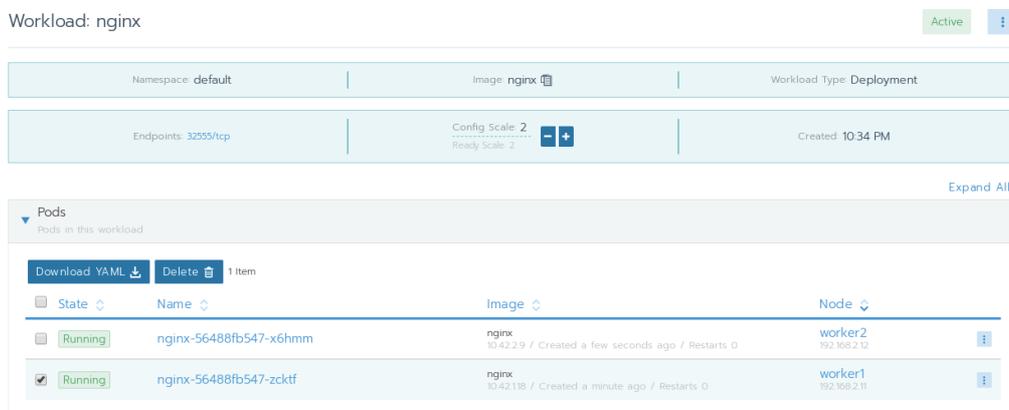
Para tener acceso se abre el puerto 80, o se redirecciona con otro.



Se accede desde el navegador correctamente.



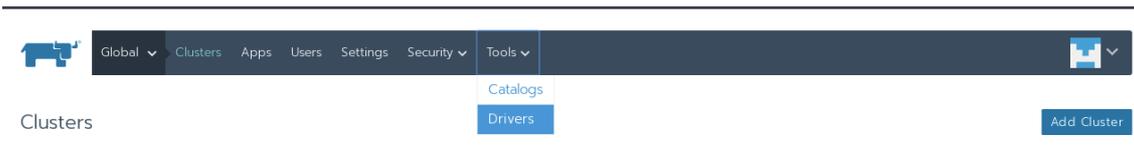
Se puede aumentar el escalado



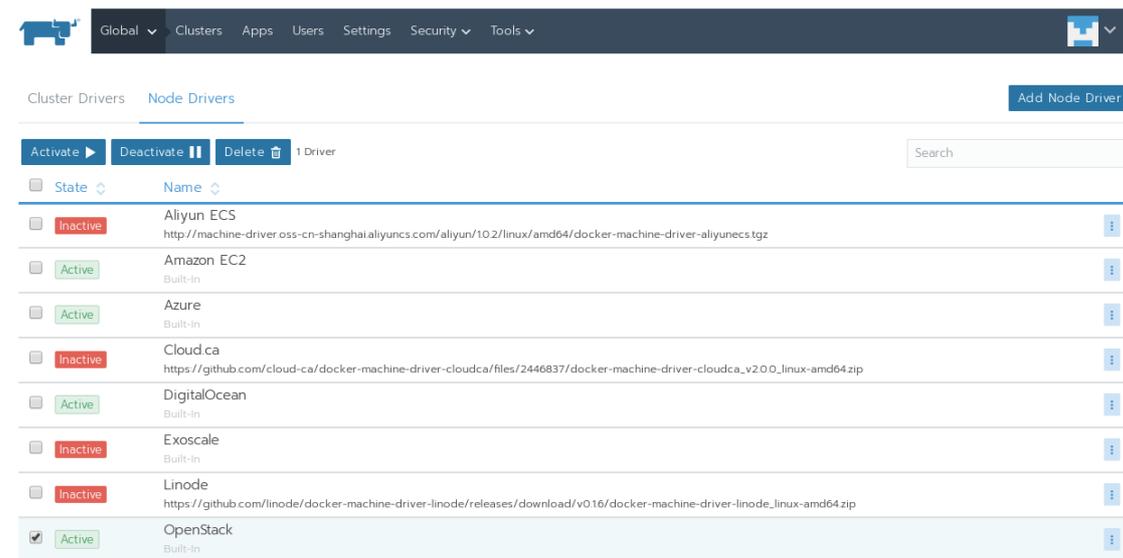
## 4. Cluster en OpenStack

Creación de cluster de kubernetes en OpenStack, se crearán las instancias automáticamente.

Primero se activa el driver de openstack en rancher, en la pestaña tools, sección drivers.



Buscamos el de OpenStack y lo activamos.

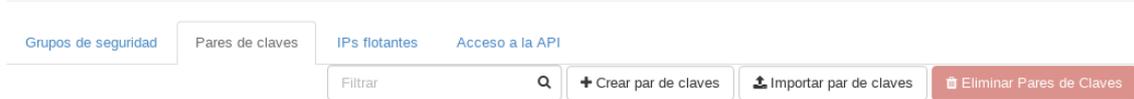


Ahora crearemos un par de claves dentro del contenedor donde tenemos desplegado rancher, ya que nos servirán para la creación de las instancias y configuración de las mismas.

```
# ssh-keygen -f rancher-key
```

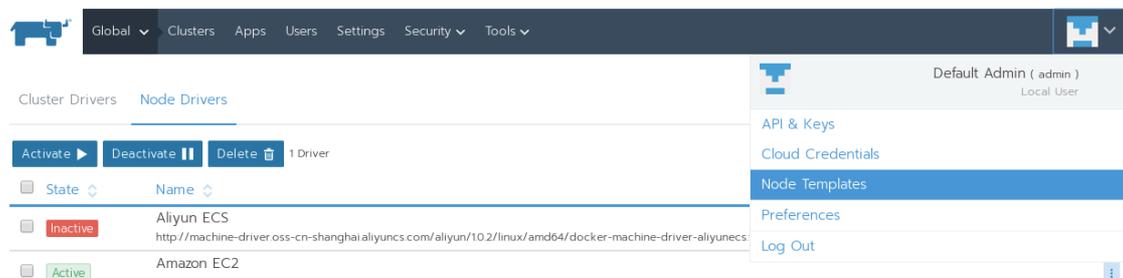
Importamos la clave publica generada al proyecto donde vamos a crear el cluster en OpenStack.

## Acceso y seguridad



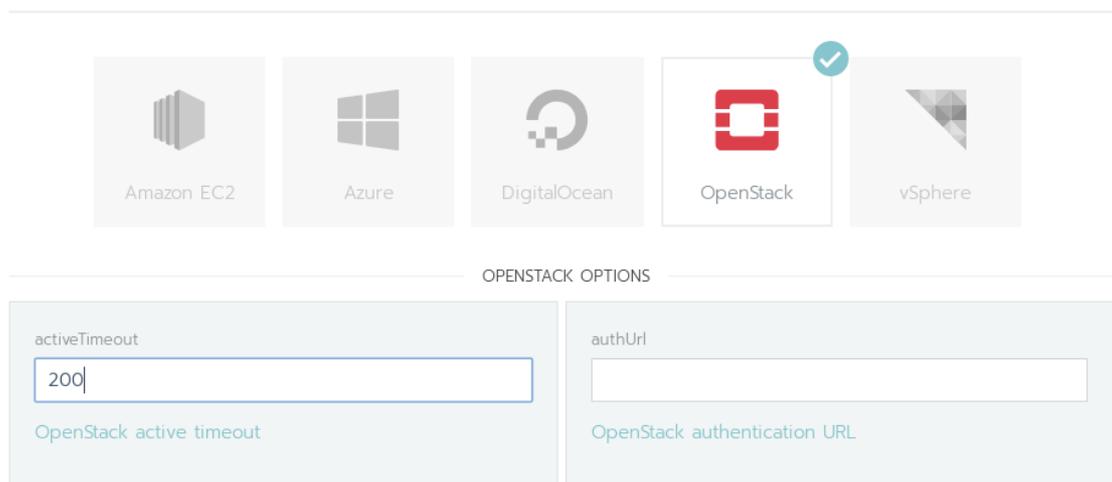
Ahora crearemos una plantilla para OpenStack, podemos crear todas las plantillas que necesitemos.

Nos dirigimos a la pestaña del usuario y a Node Templates.



Le damos a añadir plantilla, y elegimos OpenStack.

### Add Node Template



Vamos a explicar los campos mas importantes.

**authURL:** URL de autenticación.

En mi caso: *https://jupiter.gonzalonazareno.org:5000/v3*

**availabilityZone:** Zona disponible donde se crearán las instancias.

En mi caso: *nova*

**cacert:** Certificado del sitio web

En mi caso: *iesgn.crt*

**domainName:** Nombre del dominio al que pertenece el usuario.

En mi caso: *iesgn*

**flavorName:** Nombre del sabor con el que se crearán las instancias.

En mi caso: *m1.medium*

**floatingipPool:** Nombre de la red externa.

En mi caso: *ext-net*

**imageName:** Nombre de la imagen para crear las instancias.

En mi caso: *Debian Stretch 9.1*

**keypairName:** Nombre del par de claves importadas en el proyecto.

En mi caso: *rancher-key*

**netName:** Nombre de la red interna.

En mi caso: *red de jesus.arias*

**password:** Contraseña del usuario.

**privateKeyFile:** Ruta de la clave privada dentro del contenedor rancher.

En mi caso: */root/rancher-key*

**sshUser:** Nombre de usuario al que se va a conectar por ssh.

En mi caso: *debian*

**tenantName:** Nombre del proyecto donde se crearán las instancias.

En mi caso: *Proyecto de jesus.arias*

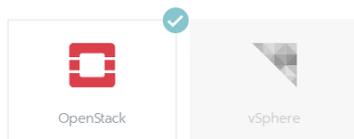
**userName:** Nombre de usuario.

En mi caso: *jesus.arias*

Por ultimo, le damos un nombre a la plantilla.

|  |              |           |     |     |                |
|--|--------------|-----------|-----|-----|----------------|
| <input checked="" type="checkbox"/> Active | iesgn-debian | OpenStack | N/A | N/A | <span>⋮</span> |
|--|--------------|-----------|-----|-----|----------------|

Creamos el cluster.



Cluster Name Add a Description

cluster-iesgn

**Member Roles**  
Control who has access to the cluster and what permission they have to change it.

**Node Pools**  
Configure the machines that will make up the cluster

| Name Prefix | Count                      | Template                              | etcd                                | Control Plane                       | Worker                              |
|-------------|----------------------------|---------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| master      | 1 <input type="checkbox"/> | iesgn-debian <input type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input type="checkbox"/>            |
| worker1     | 1 <input type="checkbox"/> | iesgn-debian <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |
| worker2     | 1 <input type="checkbox"/> | iesgn-debian <input type="checkbox"/> | <input type="checkbox"/>            | <input type="checkbox"/>            | <input checked="" type="checkbox"/> |

Number of nodes required: ✔ 1, 3, or 5 ✔ 1 or more ✔ 1 or more

[+ Add Node Pool](#)

[Edit as YAML](#)

Cuando se lanza, desde el OpenStack vemos como se crean las instancias.

## Instancias

| Nombre de instancia = <input type="text"/> |                        |                     |              |           |               |           |                        |           |                   |                          |                     |
|--|------------------------|---------------------|--------------|-----------|---------------|-----------|------------------------|-----------|-------------------|--------------------------|---------------------|
| Filtrar                                    |                        |                     |              |           |               |           |                        |           |                   |                          |                     |
| Lanzar instancia                           |                        |                     |              |           |               |           |                        |           |                   |                          |                     |
| Eliminar instancias                        |                        |                     |              |           |               |           |                        |           |                   |                          |                     |
| More Actions                               |                        |                     |              |           |               |           |                        |           |                   |                          |                     |
| <input type="checkbox"/>                   | Nombre de la instancia | Nombre de la imagen | Dirección IP | Tamaño    | Par de claves | Estado    | Zona de Disponibilidad | Tarea     | Estado de energía | Tiempo desde su creación | Actions             |
| <input type="checkbox"/>                   | worker1                | Debian Stretch 9.1  | 10.0.0.8     | m1.medium | rancher-key   | Construir | nova                   | Generando | Sin estado        | 0 minutos                | Asociar IP flotante |
| <input type="checkbox"/>                   | worker2                | Debian Stretch 9.1  | 10.0.0.5     | m1.medium | rancher-key   | Construir | nova                   | Generando | Sin estado        | 0 minutos                | Asociar IP flotante |
| <input type="checkbox"/>                   | master1                | Debian Stretch 9.1  | 10.0.0.9     | m1.medium | rancher-key   | Construir | nova                   | Generando | Sin estado        | 0 minutos                | Asociar IP flotante |

Al cabo de unos minutos, tendremos nuestro cluster funcionando.

## 5. Conclusión

Rancher es una herramienta que nos ayuda muchísimo al manejo de cluster de una manera muy sencilla, con respecto a la herramienta tiene muchísimas funcionalidades. La creación de cluster es muy rápida y sencilla.

## 6. Trabajo futuro

En este punto podríamos añadir muchas cosas, ya que tiene varios proveedores públicos como Google, Amazon, Azure, DigitalOcean... Podemos desplegar cluster y ver cada una de esas tecnologías, también podríamos añadir integración continua desplegando alguna aplicación.

## 7. Referencias

<https://rancher.com/docs/rancher/v2.x/en/quick-start-guide/>