CREAR YADMINISTRAR 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 (laaS).

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-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

se Member Roles Control who has access to the cluster and what permession they have to change it. Edit as YAML Customer Roles Customer Roles Customer Role in the cluster Mindows Support (Experimental) Project Network Isolation Network Isolation Available) Project Network Isolation Project Network Isolation Project Network Isolation Network Isolation Available Project Network Isolation Project Network Isolation Network Isolation Available Project Network Isolation Project Network Isol	Cluster Name *			Add a Description
Member Roles Control who has access to the cluster and what permission they have to change it. Edit as YAML C Cluster Options Customite Robernetes options for the cluster Kubernetes Version V135-rancher1-2 Network Isolation Available) Enabled Enabl	se			
Edit as YAML	Member Roles Control who has access to the cluster and what permission t			
Cluster Options Customize Rubernetes options for the cluster Kubernetes Version V135-rancher1-2 Network Provider Canal (Network Isolation Available) Enabled Enabled Disabled Cloud Provider f your cloud provider is not listed, please use the Custom option. None Amazon Azure Custom Read more about the Kubernetes cloud providers	Edit as YAML 🛍			
Kubernetes Version v113.5-rancher1-2 Network Provider Vindows Support (Experimental) Project Network Isolation Image: Image	Cluster Options Customize Kubernetes options for the cluster			
v1135-rancher1-2 Network Provider Canal (Network Isolation Available) Enabled Disabled Cloud Provider Cloud Provider If your cloud provider is not listed, please use the Custom option. None Annazon Azure Custom Read more about the Kubernetes cloud providers	Kubernetes Version			
Network Provider Windows Support (Experimental) Project Network Isolation Canal (Network Isolation Available) Enabled Enabled Image: Cloud Provider Disabled Disabled	v1.13.5-rancher1-2	\checkmark		
Canal (Network Isolation Available)	Network Provider	Windows Support (Experimental)	Project Network Isolation	
Oisabled Oisabled Cloud Provider If your cloud provider is not listed, please use the Custom option. None Anazon Azure Custom Read more about the Kubernetes cloud providers	Canal (Network Isolation Available) $\qquad \checkmark$	Enabled	Enabled	
Cloud Provider If your cloud provider is not listed, please use the Custom option. In Annacon In Annacon In Azure In Custom Read more about the Kubernetes cloud providers		Disabled	Disabled	
If your cloud provider is not listed, please use the Custom option. None Amazon Azure Custom Read more about the Kubernetes cloud providers	Cloud Provider			
 None Amazon Azure Custom Read more about the Kubernetes cloud providers	() If your cloud provider is not listed, please use the C	Custom option.		
Amazon Azure Custom Read more about the Kubernetes cloud providers	None			
Azure Custom Read more about the Kubernetes cloud providers	Amazon			
Custom Read more about the Kubernetes cloud providers	O Azure			
Read more about the Kubernetes cloud providers	Custom			
	Read more about the Kubernetes cloud providers			

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.

Glob	al 🗸 Clusters A	Apps Users Settings S	iecurity 🗸 Tools 🗸			~
dd Cluster	: se					
Edit as YAML	4					
Customize I Editing node	Node Run Comman options will update the	nd e command you will run on your				
1	Node Options Choose what roles t					
	Node Noie	etcd		Control Plane	🗹 Worke	r
					Sh	ow advanced options
2	Run this comman sudo docker ru /var/run:/var/ jys66p682df4xt 4005736d9912fc	nd on one or more existin un -dprivilegedres /run rancher/rancher-age S9VrCt6pxrwdztklf8pyr dcbc55a53f15fd2a1097c80e	ng machines already ru tart=unless-stopped nt:v2.2.3server h 9lvn28qtcvnvqr88v 71ffcc70690a8f1ad09b	nning a supported version net=host -v /etc/kubern ttps://192.168.2.99tok ca-checksum 5221787worker	of Docker. etes:/etc/kubernetes -v en	ũ
3 new noo	des have registered			Done		

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
jvs66p682df4xt59vfct6pxrww2dtxdlf8p7q9lvn28qtcvnvqr88v --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
jvs66p682df4xt59vfct6pxrww2dtxdlf8p7q9lvn28qtcvnvqr88v --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 Default Workloads Apps Resources Namespaces Members Tools V		•
Workloads Load Balancing Service Discovery Volumes Pipelines	o 🗏 💩 📜	Import YAML Deploy
•		
+ Deploy		

Se elige la imagen de nginx.

Se Vorkloads Apps Resources V Namespaces Members	Tools 🗸	™ ×
Deploy Workload		
Name * Add a Description nginx	Workload Type	More options
Docker Image *	Namespace •	Add to a new namespace
nginx	default	\checkmark
Environment Veriables		Expand All
Set the environment that will be visible to the container, including injecting values from other re		
Node Scheduling Configure what nodes the pods can be deployed to.		
Health Check Periodically make a request to the container to see if it is alive and responding correctly		
• Volumes Persist and share data separate from the lifecycle of an individual container.		
Scaling/Upgrade Policy Configure how pods are replaced when performing an upgrade		
		Show advanced options

Se desplega, y ya estaría funcionando.

Se Default Workloads Apps Resources V Namespaces Members To	ools 🗸	~
Workloads Load Balancing Service Discovery Volumes Pipelines	o = 4	import YAML Deploy
Redeploy 🤊 Pause Orchestration 🔢 Download YAML 🛓 Delete 📋		Search
State 🗘 Name 🗘	Image 💠	Scale 💠
Namespace: default		I
□ ▶ Active nginx ♠	nginx 1 Pod / Created a few seconds ago	1

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

Port Mapping				
Publish the container port *	Protocol	As a	On listening port *	
80	TCP \checkmark	NodePort (On every node) $\qquad \checkmark$	32555	
+ Add Port				

Se accede desde el navegador correctamente.

\leftrightarrow \rightarrow	G	(i) No es seguro	www.nginxrancher.com:32555	¤ ☆ e :
			Welcome to nginx!	
			If you see this page, the nginx web server is successfully installed and working. Further configuration is required.	
			For online documentation and support please refer to <u>nginx.org</u> . Commercial support is available at <u>nginx.com</u> .	
			Thank you for using nginx.	

Se puede aumentar el escalado

Workload: nginx					Active	÷
Na	mespace: default	Image: nginx 🖷	I.	Workload Type: Deployment		
Er	dpoints: 32555/tcp	Config Scale: 2 Ready Scale: 2		Created: 10:34 PM		
Pods Pods					Expand	All
Download YAML	Delete 🍵 1 Item					
State 🗘	Name 💠	Image 💠		Node 🗘		
Running	nginx-56488fb547-x6hmm	nginx 10.42.2.9 / Created a few se	conds ago / Restarts O	worker2 192.168.2.12	÷	
Running	nginx-56488fb547-zcktf	nginx 10.42.1.18 / Created a minute		worker1	1	

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.

Globa	l ✔ Clusters Apps Users Settings Security ✔ Tools ✔		- ~
Cluster Drivers	Node Drivers		Add Node Driver
Activate 🕨 De	activate 🔢 Delete 💼 1 Driver	Search	
🔲 State 💠	Name 🗇		
Inactive	Aliyun ECS http://machine-driver.oss-cn-shanghai.aliyuncs.com/aliyun/10.2/linux/amd64/docker-machine-driver-aliyunecs.tgz		÷
Active	Amazon EC2 Built-In		÷
Active	Azure Built-In		÷
Inactive	Cloud ca https://github.com/cloud-ca/docker-machine-driver-cloudca/files/2446837/docker-machine-driver-cloudca_v200_linux-amd64.zip		÷
Active	DigitalOcean Built-In		÷
Inactive	Exoscale Built-In		÷
Inactive	Linode https://github.com/linode/docker-machine-driver-linode/releases/download/v0.16/docker-machine-driver-linode_linux-amd64.zip		÷
Active	OpenStack Built-In		÷

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

Grupos de seguridad	Pares de claves	IPs flotantes	Acceso a la API			
		Filtrar	Q	+ Crear par de claves	1 Importar par de claves	🛍 Eliminar Pares de Claves

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

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

Global 、	✔ Clusters Apps Users Settings Security ✔ Tools ✔		
Cluster Drivers	Node Drivers	2	Default Admin (admin) Local User
		API & Keys	
Activate 🕨 Deact	tivate 🚺 Delete 🍵 1 Driver	Cloud Credentials	
🔲 State 🗇	Name 🗇		
Inactive	Aliyun ECS	Preferences	
	http://machine-driver.oss-cn-shanghai.aliyuncs.com/aliyun/10.2/linux/amd64/docker-machine-driver-aliyunecs	Log Out	
Active	Amazon EC2		:

Le damos a añadir plantilla, y elegimos OpenStack.

Amazon EC2	Azure	DigitalOcean	OpenStack	vSphere	
activeTimeout 200 OpenStack active timeout		OPENSTACK OPTIONS	ack authentication UR	Ľ	

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.

OpenStack	N/A	N/A	÷
	OpenStack	OpenStack N/A	OpenStack N/A N/A

Creamos el cluster.

OpenStack	vSphere						
uster Name *						Add a	Desc
cluster-iesgn							
Member Roles Control who has access to the cluster and what							
Node Pools Configure the machines that will make up the clu							
Name Prefix 💠	Count ᅌ	Template 💠		etcd	Control Plane	Worker	
master	1 >	k iesgn-debian	~ +	•	¥		-
worker1	1	x iesgn-debian	~ +				-
worker2	1 >	x iesgn-debian	~ +				-
Number of nodes required:				Ø 1, 3, or 5	Ø 1 or more	Ø 1 or more	
+ Add Node Pool							
Edit as YAML 4							

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

Instancias

	Nom	Nombre de instancia = 🕶			Filtrar	🗅 Lanzar inst	ancia 💼 E	Eliminar instancias More Actions 🕶		
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
worker1	Debian Stretch 9.1	10.0.0.8	m1.medium	rancher- key	Construir	nova	Generando	Sin estado	0 minutos	Asociar IP flotante 💌
worker2	Debian Stretch 9.1	10.0.0.5	m1.medium	rancher- key	Construir	nova	Generando	Sin estado	0 minutos	Asociar IP flotante 💌
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/