Kubernetes en Raspberry PI 4

**Guia:**

<https://alexsniffin.medium.com/a-guide-to-building-a-kubernetes-cluster-with-raspberry-pis-23fa4938d420>:

Crea la imagen teniendo en cuenta:

* Usa una imagen headless (conectada por cable si se puede)
* Añade un usuario con password o no se puede acceder
* Añade una clave pública para acceder por ssh.

Configurar ip estatica con sudo nmtui (<https://forums.raspberrypi.com/viewtopic.php?t=357678>):

* sudo nmcli c show

To find out the name of the connection you want to modify. Then use nmtui tool:

* Get gateway:
  + route -n | grep 'UG[ \t]' | awk '{print $2}'
* Get DNS:
  + less /etc/resolv.conf
* sudo nmtui

Edit wired 1 connection

* Set to manual
* Add private IP
* Add gateway
* Add DNS

I.e.

nameserver 80.58.61.250

nameserver 80.58.61.254

Las siguientes instrucciones son SOLO para el MASTER, los nodos son diferentes, mira Kube adm

Configura el cgroup y reinicia

Apt-get update e instalar Docker siguiendo la guia (NO IMPORTES EL PAQUETE DE KUBERNETES SIGUIENDO LA GUIA, si lo haces tendras que borrarlo con apt del y limpiar el fichero con la referencia):

* sudo apt update && sudo apt upgrade -y (will take time)
* curl -sSL https://get.docker.com | sh
* sudo usermod -aG docker [user]

Instala cri-docker siguiendo la guia

Para el Paquete de kubernetes sigue la guia de Kubernetes para la versión 31 :

<https://v1-31.docs.kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

Para añadir un nuevo nodo se configura la maquina, pero no se instala Kubernetes igual, solo que hay que ejecutar el comando de unirse al cluster. Este comando se puede obtener desde el master:

kubeadm token create --print-join-command

COMMENTS JAVI -> Instalado netcat para los primeros pasos de la guía

Para crear el cluster cambia el fichero de configuración:

Para ver los valores por defecto:

kubeadm config print init-defaults

Reemplaza {token} con el token por defecto del anterior comando

Pon la IP de la maquina en advertiseAddress

Creo que

networking:

podSubnet: "10.244.0.0/16" # --pod-network-cidr

Esta bien y yo lo puse mal

apiVersion: kubeadm.k8s.io/v1beta3

bootstrapTokens:

- groups:

- system:bootstrappers:kubeadm:default-node-token

token: {token}

usages:

- signing

- authentication

kind: InitConfiguration

localAPIEndpoint:

advertiseAddress: 192.168.1.49

bindPort: 6443

nodeRegistration:

criSocket: unix:///var/run/cri-dockerd.sock

imagePullPolicy: IfNotPresent

name: node-0

---

apiVersion: kubeadm.k8s.io/v1beta3

kind: ClusterConfiguration

networking:

podSubnet: "10.244.0.0/16" # --pod-network-cidr

Problems?

[preflight] Running pre-flight checks

[WARNING SystemVerification]: missing optional cgroups: hugetlb

sudo kubeadm init --config medium-kubeadm-config.yaml

Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.

Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:

https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

ESTO SE CONSIGUE CON:

kubeadm token create --print-join-command