# Instalar Kubernetes en formato microk8s en Ubuntu 22

## Necesitamos:

* Un entorno Ubuntu 22.04 LTS, 20.04 LTS, 18.04 LTS o 16.04 LTS para ejecutar los comandos (u otro sistema operativo que admita snapd; consulte la documentación de snapd)
* MicroK8s se ejecuta en tan solo 540 MB de memoria, pero para adaptarse a las cargas de trabajo, recomendamos un sistema con al menos 20 G de espacio en disco y 4 G de memoria.
* Una conexión a internet

## Instalar MicroK8

MicroK8s instalará un Kubernetes mínimo y liviano que puede ejecutar y usar en prácticamente cualquier máquina. Se puede instalar con un complemento:

$ sudo snap install microk8s --classic

## Únete al grupo

MicroK8s crea un grupo para permitir el uso continuo de comandos que requieren privilegios de administrador. Para agregar su usuario actual al grupo y obtener acceso al directorio de almacenamiento en caché .kube, ejecute los siguientes dos comandos:

$ sudo usermod -a -G microk8s $USER

$ sudo chown -f -R $USER ~/.kube

También deberá volver a ingresar a la sesión para que se realice la actualización del grupo:

$ su - $USER

## Comprobar el estado

MicroK8s tiene un comando incorporado para mostrar su estado. Durante la instalación, puede usar el indicador --wait-ready para esperar a que se inicialicen los servicios de Kubernetes:

$ microk8s status --wait-ready

# Acceder a Kubernetes

MicroK8s incluye su propia versión de kubectl para acceder a Kubernetes. Úselo para ejecutar comandos para monitorear y controlar sus Kubernetes. Por ejemplo, para ver su nodo:

$ microk8s kubectl get nodes

…o para ver los servicios en ejecución:

$ microk8s kubectl get services

MicroK8s utiliza un comando kubectl con espacio de nombres para evitar conflictos con cualquier instalación existente de kubectl. Si no tiene una instalación existente, es más fácil agregar un alias (añadir a ~/.bash\_aliases) como este:

$ alias kubectl='microk8s kubectl'

# Implementar una aplicación

Por supuesto, Kubernetes está diseñado para implementar aplicaciones y servicios. Puede usar el comando kubectl para hacerlo como con cualquier Kuberenetes. Intenta instalar una aplicación de demostración:

$ microk8s kubectl create deployment nginx --image=nginx

La instalación puede demorar uno o dos minutos, pero puede verificar el estado:

$ microk8s kubectl get pods

# Usar complementos

MicroK8s utiliza el mínimo de componentes para un Kubernetes puro y ligero. Sin embargo, hay muchas funciones adicionales disponibles con solo presionar unas pocas teclas usando "complementos": componentes preempaquetados que proporcionarán capacidades adicionales para su Kubernetes, desde la administración simple de DNS hasta el aprendizaje automático con Kubeflow.

Para empezar se recomienda añadir la gestión de DNS para facilitar la comunicación entre servicios. Para las aplicaciones que necesitan almacenamiento, el complemento de "almacenamiento" proporciona espacio de directorio en el host. Estos son fáciles de configurar:

$ microk8s enable dns storage

### Cómo usar los complementos de MicroK8s

Con el comando de estado de microk8s puede ver la lista de los complementos disponibles:

$micro8ks status

Para habilitar un complemento, debe llamar a microk8s enable seguido del nombre del complemento, por ejemplo, microk8s enable dns. Del mismo modo, la desactivación de microk8s desactivará el complemento nombrado y lo desinstalará.

En el resultado del ejemplo anterior, puede notar que la descripción de cada complemento comienza con "núcleo". Esto indica el repositorio desde el que está disponible el complemento. Los complementos principales son los que mantiene y respalda el equipo de MicroK8s. Además de los complementos principales, enviamos MicroK8 con un conjunto de complementos mantenidos por la comunidad. Para mostrarlos, solo necesita ejecutar el comando microk8s enable community.

La adición, eliminación y actualización de repositorios se realiza a través del comando microk8s addons repo. La comunidad de habilitación de microk8s es un atajo para hacer visibles las contribuciones de la comunidad sin conocer la URL exacta del repositorio respectivo.

### Complemento: dashboard

El Panel de Kubernetes estándar es una forma conveniente de realizar un seguimiento de los

actividad y uso de recursos de MicroK8s.

En todas las plataformas, puede instalar el tablero con un solo comando:

$ microk8s enable dashboard

Para acceder al panel instalado:

Para iniciar sesión en el dashboasd, necesitará el token de acceso (a menos que RBAC haya sido también habilitado). Esto se genera aleatoriamente en la implementación, por lo que algunos comandos son necesarios para recuperarlo:

Para MicroK8s 1.24 o posterior:

$ microk8s kubectl create token default

A continuación, debe conectarse al servicio del tablero. Mientras que el complemento MicroK8s debe tener una dirección IP en su red local (la IP del clúster del servicio kubernetes-dashboard), también puede llegar al tablero reenviando su puerto a uno libre en su host con:

$ microk8s kubectl port-forward -n kube-system service/kubernetes-dashboard 10443:443

Luego puede acceder al Tablero en https://127.0.0.1:10443.

### Iniciar y detener MicroK8

MicroK8s seguirá funcionando hasta que decidas detenerlo. Puede detener e iniciar MicroK8 con estos simples comandos:

$ microk8s stop

… detendrá MicroK8s y sus servicios. Puede comenzar de nuevo en cualquier momento ejecutando:

$ microk8s start

Tenga en cuenta que si deja MicroK8s en ejecución, se reiniciará automáticamente después de un reinicio. Si no desea que esto suceda, simplemente recuerde ejecutar microk8s stop antes de apagarlo.

# Crear un cluster MicroK8s

Aunque MicroK8s está diseñado como una implementación ultraligera de Kubernetes, todavía es posible y útil poder crear un clúster de MicroK8s. Esta página explica cómo agregar y eliminar nodos y qué se requiere para que el clúster tenga alta disponibilidad.

Nota: Cada nodo en un clúster de MicroK8s requiere su propio entorno para trabajar, ya sea una máquina virtual o contenedor independiente en una sola máquina o una máquina diferente en la misma red. Tenga en cuenta que, como con casi todos los servicios en red, también es importante que estas instancias tengan la hora correcta (por ejemplo, actualizada desde un servidor ntp) para que funcione la comunicación entre nodos.

## Añadiendo un nodo

Para crear un clúster a partir de dos o más instancias de MicroK8s que ya se están ejecutando, use el comando microk8s add-node. La instancia de MicroK8s en la que se ejecuta este comando será el maestro del clúster y albergará el plano de control de Kubernetes:

$ microk8s add-node

Esto devolverá algunas instrucciones de unión que deben ejecutarse en la instancia de MicroK8s que desea unir al clúster (NO EL NODO DESDE EL QUE EJECUTÓ add-node)

Desde el nodo que desea unir a este clúster, ejecute lo siguiente:microk8s join 192.168.1.230:25000/92b2db237428470dc4fcfc4ebbd9dc81/2c0cb3284b05

Use el indicador '--worker' para unirse a un nodo como un trabajador que no ejecuta el plano de control, por ejemplo:

$ microk8s join 192.168.1.230:25000/92b2db237428470dc4fcfc4ebbd9dc81/2c0cb3284b05 --worker

Si no se puede acceder al nodo que está agregando a través de la interfaz predeterminada, puede usar uno de los siguientes:

$ microk8s join 192.168.1.230:25000/92b2db237428470dc4fcfc4ebbd9dc81/2c0cb3284b05

$ microk8s join 10.23.209.1:25000/92b2db237428470dc4fcfc4ebbd9dc81/2c0cb3284b05

$ microk8s join 172.17.0.1:25000/92b2db237428470dc4fcfc4ebbd9dc81/2c0cb3284b05

Unir un nodo al clúster solo debería llevar unos segundos. Después

debería poder ver que el nodo se ha unido:

$ microk8s kubectl get no

NAME STATUS ROLES AGE VERSION

10.22.254.79 Ready <none> 27s v1.15.3

ip-172-31-20-243 Ready <none> 53s v1.15.3

## Removing a node

Primero, en el nodo que desea eliminar, ejecute microk8s. MicroK8s en el nodo de salida

reiniciará su propio plano de control y reanudará las operaciones como un clúster completo de un solo nodo:

$ microk8s leave

Para completar la eliminación del nodo, llame a microk8s remove-node de los nodos restantes para indicar que el nodo saliente (no accesible ahora) debe eliminarse de forma permanente:

$ microk8s remove-node 10.22.254.79

### Almacenamiento

Si está utilizando el almacenamiento simple proporcionado por el complemento de almacenamiento de hostpath, tenga en cuenta que solo estará disponible para los nodos en los que se ha habilitado. Para el almacenamiento en clúster, debe configurar un almacenamiento alternativo. Por ejemplo, consulte la guía sobre el uso de NFS

## High Availability

Desde la versión 1.19 de MicroK8s, HA está habilitado de forma predeterminada. Si su clúster consta de tres o más nodos, el almacén de datos se replicará en los nodos y será resistente a un solo error (si un nodo desarrolla un problema, las cargas de trabajo continuarán ejecutándose sin interrupción).

El estado de microk8s ahora incluye información sobre el estado de HA. Por ejemplo:

microk8s is running

high-availability: yes

datastore master nodes: 10.128.63.86:19001 10.128.63.166:19001 10.128.63.43:19001

datastore standby nodes: none

## Worker nodes

A partir de la versión 1.23, un nodo puede unirse al clúster como nodo trabajador. Los nodos trabajadores pueden alojar cargas de trabajo, pero no ejecutan el plano de control de Kubernetes y, por lo tanto, no aumentan la disponibilidad (HA) del clúster. Los nodos trabajadores son ideales para dispositivos de gama baja, ya que consumen menos recursos. También tienen sentido en clústeres grandes con suficientes nodos de plano de control para garantizar HA. Para agregar un nodo trabajador, use el indicador --worker cuando ejecute el comando de unión microk8s:

$ microk8s join 192.168.1.230:25000/92b2db237428470dc4fcfc4ebbd9dc81/2c0cb3284b05 --worker

Un nodo trabajador ejecuta un servidor proxy de API local que se ocupa de la comunicación entre los servicios locales (kubelet, kube-proxy) y los servidores de API que se ejecutan en varios nodos del plano de control. Al agregar un nodo trabajador, MicroK8s intenta detectar todos los puntos finales del servidor API en el clúster y configurar el nuevo nodo en consecuencia. La lista de servidores API se almacena en/var/snap/microk8s/current/args/traefik/provider.yaml.

El proxy del servidor API buscará actualizaciones automáticamente cuando se cambien los nodos del plano de control del clúster (por ejemplo, se agrega un nuevo nodo del plano de control, se elimina uno anterior) y actualizará la lista de puntos finales del servidor API conocidos.

Si ya tiene un balanceador de carga frente al servidor API, puede configurar la dirección del balanceador de carga manualmente en /var/snap/microk8s/current/args/traefik/provider.yaml. En este caso, asegúrese de deshabilitar también la actualización automática de los extremos del plano de control configurando --refresh-interval 0 en /var/snap/microk8s/current/args/apiserver-proxy.