

Kubectrl / minikube

Antonio Espín Herranz

minikube

minikube

- Crea un clúster de un solo nodo, pero es un cluster real.
- Nos sirve para hacer desarrollo, pruebas.
- Trabajar con minikube
- Acceso al Dashboard
- Crear nuevos clusters (locales, en la propia máquina)

minikube

- El comando sin opciones responde con la ayuda del comando.
- **minikube status**
 - Comprobar si minikube funciona correctamente.
 - Muestra información del host, kubelet, apiserver y kubeconfig
 - Si lo muestra parados, podemos reiniciar con:
 - minikube stop
 - minikube start
- **host:** Indica si la máquina virtual (VM) o el contenedor que ejecuta el clúster está en funcionamiento.
- **kubelet:** Muestra el estado del proceso de Kubelet, encargado de la comunicación entre los nodos y el clúster.
- **apiserver:** Indica si el servidor de la API de Kubernetes está disponible y en funcionamiento.
- **kubeconfig:** Verifica si tu configuración está correctamente vinculada al clúster.

```
C:\Users\Anton>minikube status
minikube
type: Control Plane
host: Running
kubelet: Running
apiserver: Running
kubeconfig: Configured
```

minikube

- **minikube logs**

- Genera la información de lo que va haciendo minikube.

- **minikube ip**

- Nos muestra la IP por si nos tenemos que conectar.

- Arranque y parada del cluster

- **minikube start**
 - **minibuke stop**

minikube

- Borrar el cluster (ojo, se pierde todo)
 - **minikube delete**
- El dashboard (herramienta de monitorización)
 - **minikube dashboard**
 - Lanza una aplicación Web en un navegador para monitorizar el clúster.
- Pausar / Continuar el cluster
 - **minikube pause**
 - **minikube unpause**

minikube

- Login en el entorno de minikube (entramos dentro del contenedor de minikube)
 - **minikube ssh**
- **Directorios** que crea **minikube**, en la carpeta del usuario de **Windows**:
 - **.kube**: dentro tiene un fichero **config** que almacena información sobre el cluster.
 - **.minikube**
 - Ficheros de minikube, no hace falta modificarlo.
 - Hay un directorio con las máquinas

minikube config

- Podemos verificar la configuración del clúster de minikube:
- Comando
 - **minikube config view**

```
C:\Users\Anton>minikube config view

C:\Users\Anton>minikube config set memory 4096
! These changes will take effect upon a minikube delete and then a minikube start

C:\Users\Anton>minikube config view
- memory: 4096

C:\Users\Anton>minikube config unset memory

C:\Users\Anton>minikube config view

C:\Users\Anton>
```


Cambiar configuración del clúster

- Se puede configurar un montón de opciones en la configuración de un clúster:
 - **Las opciones se pueden consultar, en:**
 - <https://minikube.sigs.k8s.io/docs/commands/config/>
 - Podemos recuperar el valor de un parámetro con:
 - **minikube config get memory**
 - Solo muestra las opciones que hemos cambiado.
 - Las opciones de configuración se almacenan en la carpeta de usuario:
 - **.minikube/config**
 - Y dentro del fichero: **config.json**

minikube

- Para **recrear** el cluster de kubernetes:
 - **minikube delete**
 - **minikube start**
- Si es otro clúster distinto para borrarlo, indicar el nombre:
 - **minikube delete --profile <nombre_del_cluster>**
 - **El clúster se conoce por profile**

Ejemplo

- Crear otro clúster dentro de minikube.
 - El clúster queda representado por un nombre de perfil (**profile**)
- Ver los cluster que tengo dentro de minikube:
 - **minikube profile list**

Profile	VM Driver	Runtime	IP	Port	Version	Status	Nodes	Active Profile	Active Kubecontext
cluster2	docker	docker	192.168.58.2	8443	v1.32.0		2		
minikube	docker	docker	192.168.49.2	8443	v1.32.0	OK	1	*	*

- Se pueden crear mas clúster y se pueden añadir varios nodos
 - **minikube start --driver=docker -p cluster2 --nodes=2**
 - -p El nombre del clúster, driver → Docker y el número de nodos 2
 - Al crearlo puede tardar un poco.
 - Después consultarlo con el comando: **minikube profile list**
 - **Ojo, los nodos consumen memoria y CPU**

minikube

- minikube **profile**
 - Nos informa con que clúster estamos trabajando
- Otra forma de saber con que clúster estamos trabajando:
 - Dentro del directorio **usuario/.kube** es donde se indica la configuración de los clusters de kubernetes
 - Este directorio está presente en todos los ordenadores que se conectan con kubernetes (no es de la herramienta minikube)
 - Dentro del directorio tenemos un fichero **config** que es donde se almacenan los clusters que tenemos.
 - Cada vez que se añade un cluster viene a este fichero y lo añade.

Fichero config

```
clusters:
- cluster:
  certificate-authority: C:\Users\Anton\.minikube\ca.crt
  extensions:
  - extension:
    last-update: Fri, 28 Mar 2025 12:19:33 CET
    provider: minikube.sigs.k8s.io
    version: v1.35.0
    name: cluster_info
    server: https://127.0.0.1:63221
    name: cluster2
- cluster:
  certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB
YVdOQm9ZcW5rQnhmL25hckpVVTh0enBQQ25KQWw5dG12akFTUkt3QXZXZTY1TC
  server: https://kubernetes.docker.internal:6443
  name: docker-desktop
- cluster:
  certificate-authority: C:\Users\Anton\.minikube\ca.crt
  extensions:
  - extension:
    last-update: Fri, 28 Mar 2025 11:42:47 CET
    provider: minikube.sigs.k8s.io
    version: v1.35.0
    name: cluster_info
    server: https://127.0.0.1:61253
    name: minikube
```

Vemos también las URLs
De conexión

Fichero config

- A cada clúster le corresponde un contexto:
 - El nombre del contexto coincide con el nombre del clúster

```
contexts:
- context:
  cluster: cluster2
  extensions:
  - extension:
    last-update: Fri, 28 Mar 2025 12:19:33 CET
    provider: minikube.sigs.k8s.io
    version: v1.35.0
    name: context_info
  namespace: default
  user: cluster2
name: cluster2
```

Buscar dentro del fichero
current-context nos indica
Cual es el contexto actual.

Es el valor que muestra el comando
minikube profile

Cambiar de profile

- **minikube profile otro_cluster**
 - Cambiar el cluster por defecto al indicado
 - En el listado de clúster veremos activado el otro clúster.
 - minikube profile list
 - Si editamos el fichero **config** la propiedad **current-context** habrá cambiando → **comprobarlo**
 - Si cambiamos la línea del fichero config tendríamos el mismo efecto y a partir del cambio, cualquier comando que lancemos será sobre el nuevo clúster

Prueba

- **kubectl get nodes**

- Este comando se lanzará siempre sobre el clúster actual

```
C:\Users\Anton>minikube profile
* cluster2

C:\Users\Anton>kubectl get nodes
NAME                STATUS    ROLES          AGE      VERSION
cluster2            Ready     control-plane   4h35m    v1.32.0
cluster2-m02        Ready     <none>          4h32m    v1.32.0

C:\Users\Anton>minikube profile minikube
* minikube profile was successfully set to minikube

C:\Users\Anton>kubectl get nodes
NAME        STATUS    ROLES          AGE    VERSION
minikube    Ready     control-plane   27h    v1.32.0
```


minikube dashboard

- Herramienta gráfica (aplicación Web)
- Podemos gestionar los clústers y asignarles recursos.
- Tampoco es la panacea, pero podemos crear objetos.
- Es otra opción a parte del modo comando.



minikube dashboard

- En la parte superior tenemos el namespace (es como una carpeta donde vamos a poder tener objetos).
- En la barra superior derecha el +, nos permite añadir objetos de 3 maneras distintas:
 - En línea,
 - con un fichero YAML o
 - con un formulario.



The screenshot shows the 'Create' form in the Minikube dashboard. At the top, there are three tabs: 'Crear desde entrada', 'Crear desde un fichero', and 'Crear desde un formulario'. The 'Crear desde un formulario' tab is selected and underlined. Below the tabs, there are five input fields with labels and asterisks indicating they are required:

- 'Nombre de la aplicación *' with an empty text input field.
- 'Imagen del contenedor *' with an empty text input field.
- 'Número de pods *' with a text input field containing the value '1'.
- 'Service *' with a dropdown menu showing 'None'.
- 'Espacio de nombre *' with a text input field containing the value 'default'.

At the bottom of the form, there are four buttons: 'Desplegar' (disabled), 'Preview', 'Cancelar', and 'Show advanced options'.

minikube dashboard

- En la parte izquierda de la aplicación:
 - Los objetos más utilizados en kubernetes
 - Los servicios y como nos conectamos
 - Temas de configuración
 - Otro apartado para la gestión del clúster
- ***El Dashboard se cierra con un control + C en la consola donde lo lanzamos.***

Container Runtime

- Se puede cambiar **de container runtime**, aparte de **Docker** tenemos otras.
- Se puede cambiar:
 - <https://minikube.sigs.k8s.io/docs/handbook/config/>
 - Las opciones que tenemos:

Runtime configuration

The default container runtime in minikube varies. You can select one explicitly by using:

```
minikube start --container-runtime=docker
```

Options available are:

- `containerd`
- `cri-o`
- `docker`

See <https://kubernetes.io/docs/setup/production-environment/container-runtimes/>

Viene con varias por si en la empresa en vez de docker tenemos otras, y minikube ofrece la posibilidad de cambiar el runtime (es el motor que va a ejecutar los contenedores)

Runtime configuración

- Para cambiar el runtime:
 - **minikube start --container-runtime=cri-o -p otro_cluster**
 - Si ponemos un profile / clúster que no existe crea uno nuevo, si ya existe simplemente le cambia del runtime.

```
C:\Users\Anton>minikube profile list
```

Profile	VM Driver	Runtime	IP	Port	Version	Status	Nodes	Active Profile	Active Kubecontext
cluster2	docker	crio	192.168.58.2	8443	v1.32.0	OK	2		*
minikube	docker	docker	192.168.49.2	8443	v1.32.0	OK	1	*	

Borrar el clúster

- Minikube profile list → Ver los clústers
- **minikube delete -p nombre_profile**

```
C:\Users\Anton>minikube delete -p cluster2
* Eliminando "cluster2" en docker...
* Eliminando contenedor "cluster2" ...
* Eliminando contenedor "cluster2-m02" ...
* Eliminando C:\Users\Anton\.minikube\machines\cluster2...
* Eliminando C:\Users\Anton\.minikube\machines\cluster2-m02...
* Removed all traces of the "cluster2" cluster.
```

kubectl

Kubectl

- Es la herramienta que vamos a utilizar para interactuar con los clústers de kubernetes sin importar si son locales o están en la nube.
- Se utiliza en modo comando (igual que Docker)
- Ya viene instalada con Docker desktop
- Sin parámetros responde con las opciones del comando.
- Permite realizar tareas como desplegar aplicaciones, inspeccionar recursos, modificar configuraciones, y gestionar los distintos objetos de kubenertes que se almacenan en un clúster.

Resumen de comandos

- Listar recursos:
 - Kubectl get pods
 - Kubectl get nodes
 - Kubectl get services
 - Kubectl get deployments
 - Kubectl get all (ver todos)
 - Permite utilizar abreviaturas: kubectl get po, kubectl get pod
 - O varios tipos a la vez: kubectl get pod, service
- Describir recursos (como inspect de Docker):
 - Kubectl describe pod nombre_del_pod
 - Kubectl describe node nombre_del_nodo

Resumen de comandos

- Crear un recurso con un archivo YAML
- Si no existe, para crearlo (si existe → error)
 - `Kubectl create -f archivo.yaml`
- Si no existe, lo crea, si existe lo actualiza
 - `Kubectl apply -f archivo.yaml`
- Borrar recursos (hay que indicar el tipo de recurso):
 - `Kubectl delete pod nombre_pod`
 - `Kubectl delete service nombre_service`
 - Etc.

Resumen de comandos

- Ejecutar un comando en un POD:
 - `Kubectl exec -it nombre_pod`
- Los logs de un POD
 - `Kubectl logs nombre_pod`
 - `Kubectl logs -f nombre_pod`
- Escalar un deployment:
 - `kubectl scale deployment nombre-del-deployment --replicas=5`

Resumen de comandos

- Actualizar la imagen de un contenedor en el deployment
 - `kubectl set image deployment/nombre-del-deployment nombre-del-contenedor=nueva-imagen:etiqueta`
- Exponer un deployment como servicio:
 - `kubectl expose deployment nombre-del-deployment --type=LoadBalancer --port=80 --target-port=8080`

Resumen de comandos

- Información del clúster
 - `Kubectl get nodes`
- Ver información del clúster
 - `Kubectl cluster-info`
- Mostrar todos los objetos en un namespace específico.
 - `kubectl get all -n nombre-del-namespace`
- Borrar todos los recursos de un archivo YAML
 - `kubectl delete -f archivo.yaml`

Enlaces

- Documentación:
 - <https://kubernetes.io/docs/reference/generated/kubectl/kubectl-commands#-strong-getting-started-strong->

Kubeadm vs Minikube

Kubernetes con kubeadm

- **Kubeadm:** Para instalar k8s en producción
 - ¿Qué es? Una herramienta oficial de Kubernetes para configurar y administrar clusters. Se utiliza principalmente para entornos de producción y ofrece un enfoque estandarizado para iniciar y unir nodos al cluster.
 - Propósito principal: Facilitar la instalación y configuración de clusters Kubernetes en producción. Incluye procesos para inicializar el nodo maestro (kubeadm init) y para unir nodos (kubeadm join).
 - Uso típico: Administradores de sistemas lo emplean para implementar Kubernetes en servidores físicos o virtuales con configuraciones personalizadas.
- Para la **instalación:**
 - <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>

Kubernetes con minikube

- **Minikube:** Para probar y hacer desarrollos en **local**
 - ¿Qué es? Es una herramienta diseñada para ejecutar un entorno de Kubernetes local en tu máquina (generalmente para desarrollo o aprendizaje).
 - Propósito principal: Crear un cluster Kubernetes completo y funcional en un entorno local, ideal para probar y aprender.
 - Uso típico: Desarrolladores lo utilizan para ejecutar aplicaciones o experimentar con Kubernetes sin necesidad de infraestructura remota
- Para la **instalación**
 - <https://minikube.sigs.k8s.io/docs/start/?arch=%2Fwindows%2Fx86-64%2Fstable%2F.exe+download>

Diferencias

- **kubeadm** es para **configurar clusters en producción**, mientras que **minikube** es para crear clusters **locales para pruebas**.
- Por ejemplo, podrías usar kubeadm para configurar un cluster en AWS y minikube para simular un cluster en tu laptop.


Instalar kubeadm

- Pasos:
 - Instalar el sistema
 - Instalar Docker o Containerd(el runtime de los contenedores)
 - Agregar el repositorio de kubernetes
 - Instalar kubeadm, kubectl y kubelet
 - Inicializar el clúster
 - Configurar la red
 - Unir nodos de trabajo

Instalar el sistema

- Desactiva el **swap**:

Bash


 Copiar

```
sudo swapoff -a
```

Opcional: Comenta la línea del swap en `/etc/fstab` para desactivarlo permanentemente.

- Configura los parámetros del kernel requeridos por Kubernetes:

Bash

 Copiar

```
cat <<EOF | sudo tee /etc/sysctl.d/k8s.conf
net.bridge.bridge-nf-call-ip6tables = 1
net.bridge.bridge-nf-call-iptables = 1
EOF
sudo sysctl --system
```

Instalar docker

Kubernetes requiere un runtime de contenedores. Instala Docker (u otra opción como Containerd):

Bash


 Copiar

```
sudo apt-get install -y docker.io
```

Agregar repositorio de kubernetes

- Instala herramientas como **curl** si aún no están disponibles:


Bash

 Copiar

```
sudo apt-get update  
sudo apt-get install -y apt-transport-https curl
```

- Añade la clave GPG del repositorio oficial:


Bash

 Copiar

```
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt
```

- Agrega el repositorio a tu lista de fuentes:

Bash

 Copiar

```
echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" | sudo tee /
```


Agregar repositorio de kubernetes

- `curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg |
sudo apt-key add -`
- `echo "deb http://apt.kubernetes.io/ kubernetes-xenial main" |
sudo tee /etc/apt/sources.list.d/kubernetes.list`

Instalar kubeadm, kubectl y kubelet

- Actualiza los repositorios y paquetes:


Bash

 Copiar

```
sudo apt-get update
```

- Instala las herramientas:

Bash


 Copiar

```
sudo apt-get install -y kubelet kubeadm kubectl
```


Inicializar el clúster

En el nodo maestro, inicializa el cluster:


Bash

 Copiar

```
sudo kubeadm init
```

Luego configura el acceso con `kubectl`:

Bash

 Copiar

```
mkdir -p $HOME/.kube  
sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config  
sudo chown $(id -u):$(id -g) $HOME/.kube/config
```

Configurar la red

Instala un complemento de red (como Calico o Flannel) para la comunicación entre Pods.
Por ejemplo:

Bash

 Copiar

```
kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml
```

Unir nodos de trabajo

En los nodos de trabajo, usa el comando proporcionado por `kubeadm init` para unir los nodos al cluster maestro:

Bash

 Copiar

```
kubeadm join <IP-del-master>:<puerto> --token <token> --discovery-token-ca-c
```

- `kubeadm join <IP-del-master>:<puerto> --token <token> --discovery-token-ca-cert-hash <hash>`