

# Desplegamos Kubernetes

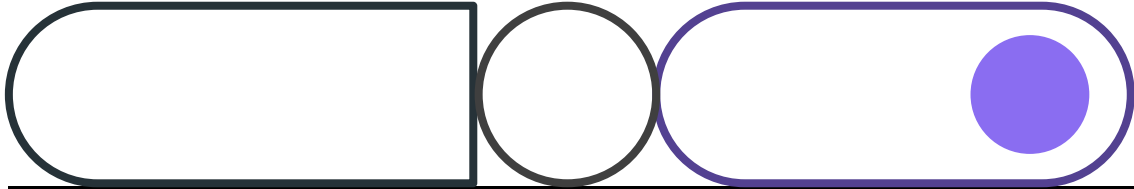
# Índice

- 01 [Kubectl - Nuestra navaja suiza](#)
- 02 [Comandos básicos](#)
- 03 [Instalación](#)
- 04 [Verificar la configuración](#)

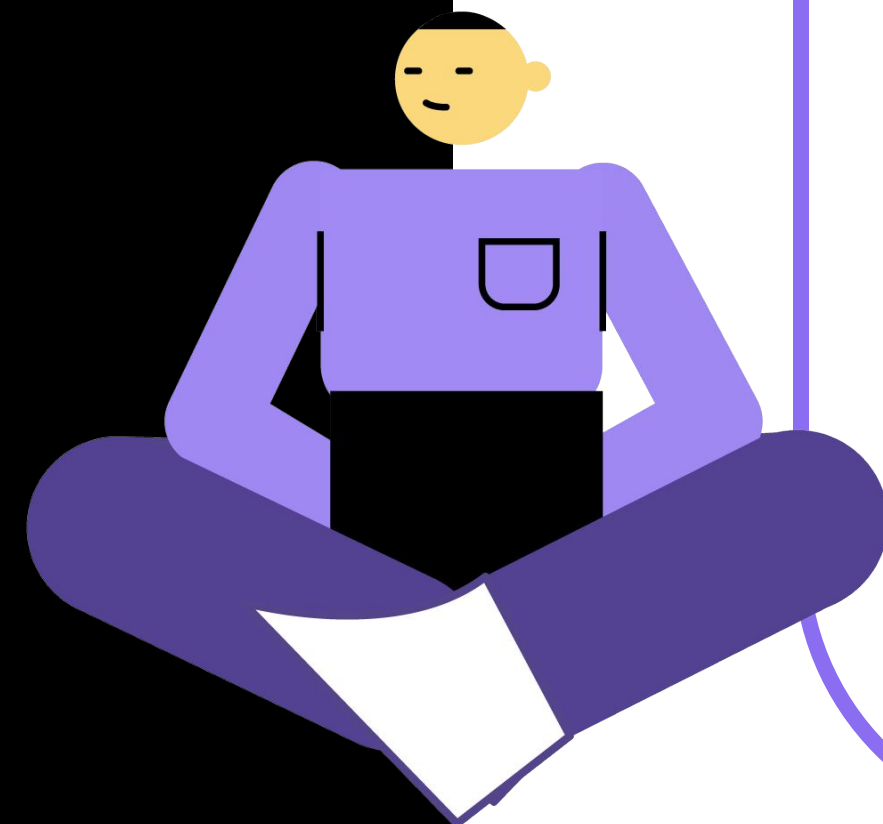


01

# Kubectl - Nuestra navaja suiza

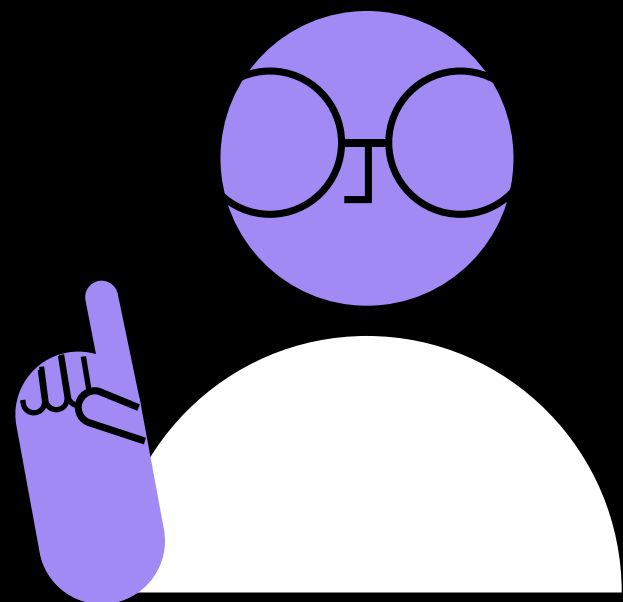


Kubectl es la herramienta de terminal para interactuar con el cluster mediante la API.



# ¿Qué es Kubectl?

Es una interfaz de línea de comandos multiplataforma que se instala en un equipo cliente. La proporciona el proyecto Kubernetes a través de una Interfaz de Programación de Aplicaciones Api.



- Kubectl permite gestionar el clúster de Kubernetes, es decir, el conjunto de máquinas de nodos que se encargan de la ejecución de las aplicaciones en contenedores.
- Con esta interfaz, el usuario también puede realizar implementaciones o deployments, donde se le indica a los kubernetes cómo crear y modificar instancias de los pods que almacenan una app en un contenedor. Estas implementaciones permiten escalar la cantidad de pods de réplica, al tiempo que facilita la habilitación de un código actualizado controladamente, o bien, retroceder hacia una versión anterior de implementación en caso de que sea necesario.

02

# Comandos básicos

# Comandos básicos

Los comandos básicos en general inician con:

**kubectl**

+

Seguido de:

**Get:** para listar objetos.

**Describe:** para describir objetos.

**Delete:** para borrar objetos.

**Create:** para crear objetos.

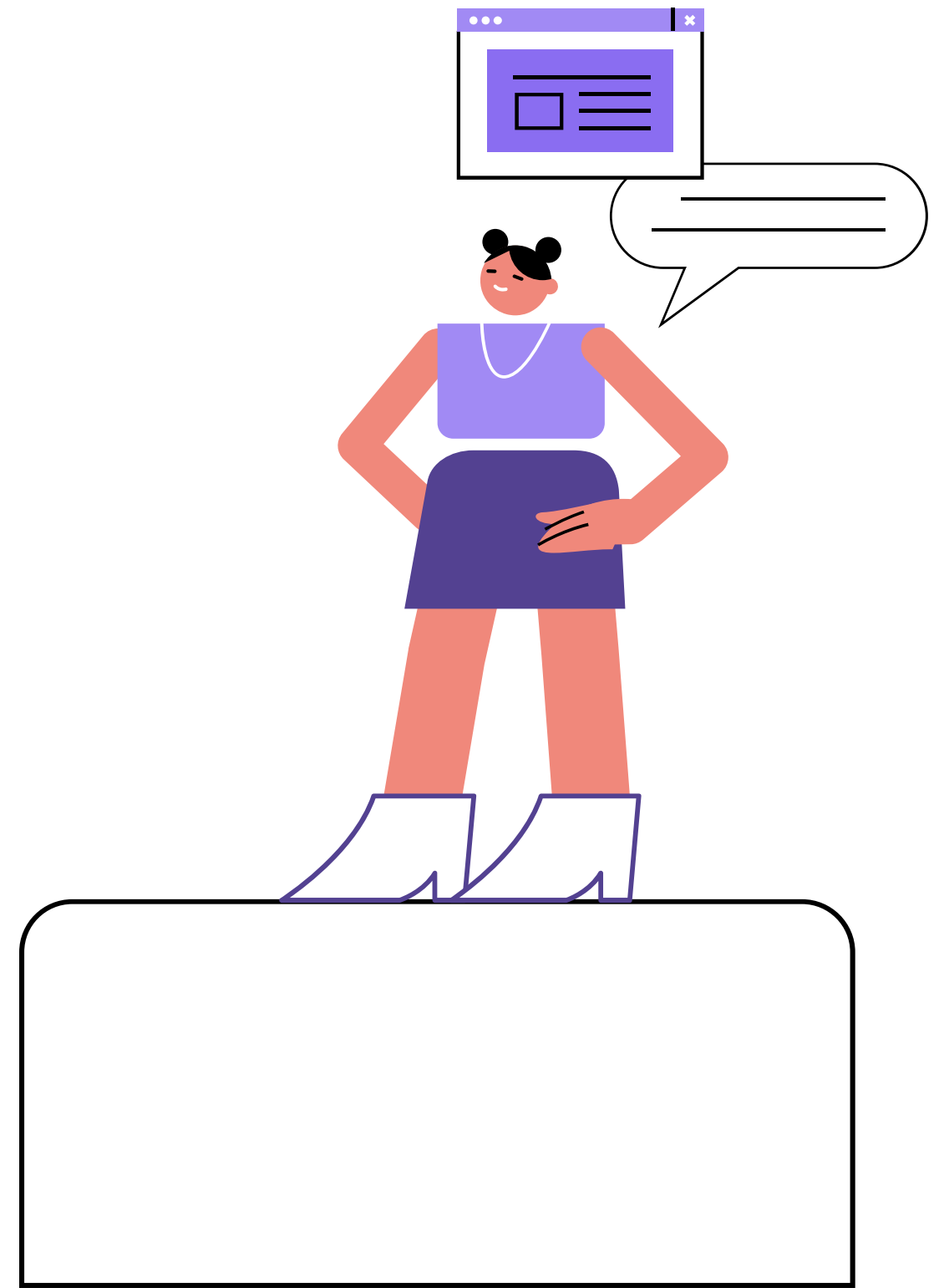
**Apply:** para aplicar un manifiesto.

+

Luego se agrega el objeto que se desea modificar.

# Ejemplos

- **kubectl cluster-info:**  
información del cluster.
- **kubectl get nodes:**  
lista los nodos del cluster.
- **kubectl get service:**  
lista los servicios.
- **kubectl get pods:**  
lista los pods.
- **kubectl get deployments:**  
lista los deployments.
- **kubectl get namespaces:**  
lista los namespaces.
- **kubectl describe pod apache1:**  
describe el pod apache1.
- **kubectl delete service hello-world:**  
borra el servicio hello-world.
- **kubectl apply -f deployment.yaml:**  
aplica la configuración detallada en el archivo deployment.yaml.





03

# Instalación



# Instalación en Linux

**Antes de comenzar:** Se debe utilizar una versión de kubectl que esté dentro de una versión menor de su cluster. Por ejemplo, un cliente v1.24 puede comunicarse con los planos de control v1.23, v1.24 y v1.25. El uso de la última versión compatible de kubectl ayuda a evitar problemas imprevistos.

01

Desde la terminal tendrán que descargar el último release con el comando debajo:

```
curl -LO "https://dl.k8s.io/release/$(curl -L -s https://dl.k8s.io/release/stable.txt)/bin/linux/amd64/kubectl"
```

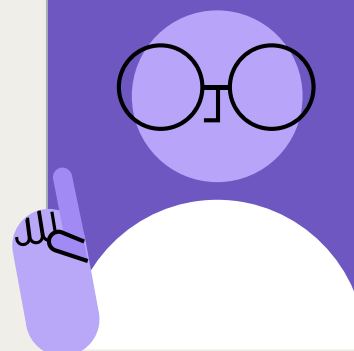
02

Instalar kubectl con el binario descargado. Elevar los privilegios con sudo:

```
sudo install -o root -g root -m 0755 kubectl /usr/local/bin/kubectl
```

03

Para verificar que haya sido exitosa la instalación podemos chequear la versión: `kubectl version --client`



Para otras alternativas de instalación consultar la documentación oficial:  
<https://kubernetes.io/docs/tasks/tools/install-kubectl-linux/#install-kubectl-binary-with-curl-on-linux>



# Instalación en Windows

**Antes de comenzar:** Se debe utilizar una versión de kubectl que esté dentro de una versión menor de su cluster. Por ejemplo, un cliente v1.24 puede comunicarse con los planos de control v1.23, v1.24 y v1.25. El uso de la última versión compatible de kubectl ayuda a evitar problemas imprevistos.

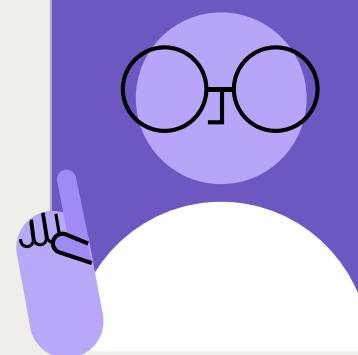
01

Para instalar kubectl en Windows, pueden usar el administrador de paquetes Chocolatey desde el Command Prompt: `choco install kubernetes-cli`

02

Para verificar que haya sido exitosa la instalación podemos chequear la versión:

`kubectl version --client`



Para otras alternativas de instalación consultar la documentación oficial:  
[https://kubernetes.io/docs/tasks/tools/install-k  
ubectl-windows/](https://kubernetes.io/docs/tasks/tools/install-kubectl-windows/)



# Instalación en MacOS

**Antes de comenzar:** Se debe utilizar una versión de kubectl que esté dentro de una versión menor de su clúster. Por ejemplo, un cliente v1.24 puede comunicarse con los planos de control v1.23, v1.24 y v1.25. El uso de la última versión compatible de kubectl ayuda a evitar problemas imprevistos.

01

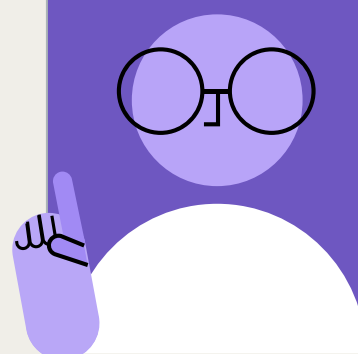
En MacOS se puede usar el manejador de paquetes Homebrew:

```
brew install kubectl
```

02

Para verificar que haya sido exitosa la instalación podemos chequear la versión:

```
kubectl version --client
```



Para otras alternativas de instalación, consultar la documentación oficial:  
<https://kubernetes.io/docs/tasks/tools/install-kubectl-macos/>

04

# Verificar la configuración

# Verificar la configuración

Para que kubectl encuentre y acceda a un cluster de Kubernetes, necesita un archivo kubeconfig, que se realiza automáticamente cuando creamos un clúster con kube-up.sh o implementamos correctamente un cluster de Minikube. De forma predeterminada, la configuración de kubectl se encuentra en ~/.kube/config.

Verifiquen que kubectl esté configurado correctamente obteniendo el estado del cluster con este comando:

**kubectl cluster-info**

Si se muestra una respuesta de URL, kubectl está configurado correctamente para acceder al cluster.

Si se muestra un mensaje similar al siguiente, significa que kubectl no está configurado correctamente o no puede conectarse al cluster de Kubernetes.

**The connection to the server  
<server-name:port> was refused -  
did you specify the right host or  
port?**

¡Muchas gracias!