

## Guía Número 2 de DevOps – Kubernetes

- Prerrequisito: Tener instalado Docker, sino es así acceder a este enlace para descargar el instalador y posteriormente realizar la instalación.

[https://drive.google.com/file/d/1mbykZVf\\_4ZGKwwVAoBq4ugzmJUEyOWdS/view?usp=sharing](https://drive.google.com/file/d/1mbykZVf_4ZGKwwVAoBq4ugzmJUEyOWdS/view?usp=sharing)

### Deploying your first Kubernetes Cluster

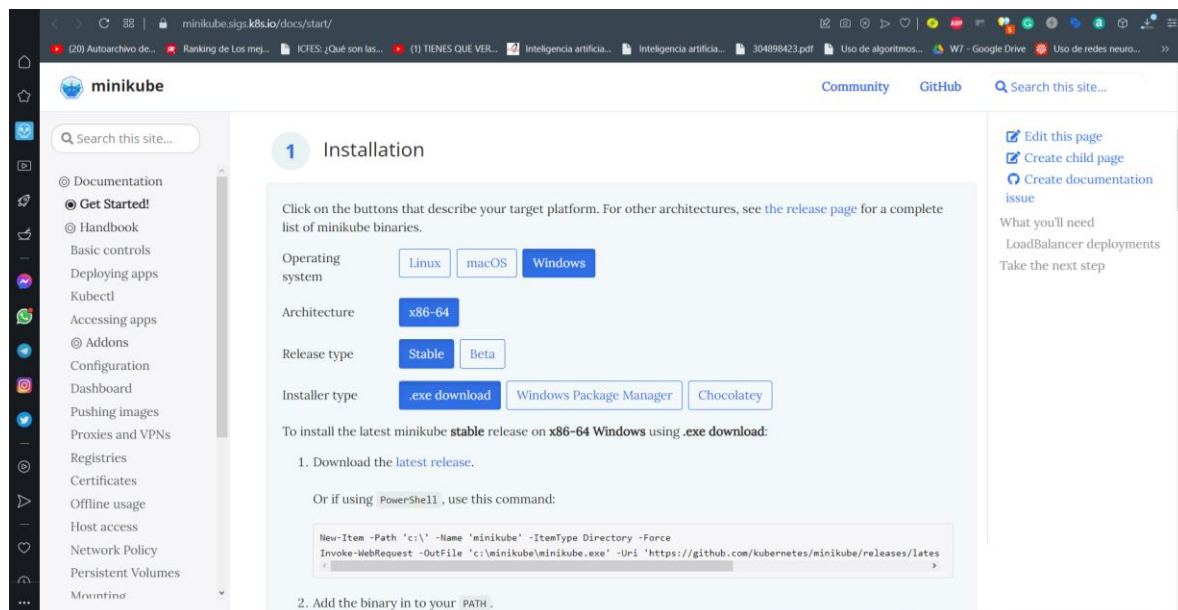
En este caso, vamos a poner en marcha un clúster de Kubernetes en nuestra máquina local usando minikube.

- What is Minikube?

“minikube configura rápidamente un clúster local de Kubernetes en macOS, Linux y Windows. Estamos orgullosos de centrarnos en ayudar a los desarrolladores de aplicaciones y a los nuevos usuarios de Kubernetes”.

Para empezar, independientemente del sistema operativo de su estación de trabajo, puede ejecutar minikube. Primero, dirígete a la página del proyecto <https://minikube.sigs.k8s.io/docs/start/>, o en su defecto acceder a la semana 2 del campus donde encuentra subido el instalador.

Se debe tener un "Administrador de contenedores o máquinas virtuales, como: Docker, Hyperkit, Hyper-V, KVM, Parallels, Podman, VirtualBox o VMware". Aquí es donde se ejecutará MiniKube.



- Getting a Kubernetes cluster up and running

Minikube se usa en la línea de comandos y, simplemente, una vez que tenga todo instalado, puede ejecutar minikube start para implementar su primer clúster de Kubernetes. Verá a continuación que

Docker Driver es el predeterminado en cuanto a dónde ejecutaremos nuestro nodo de virtualización anidado.

Un solo clúster de Minikube consistirá en un solo contenedor acoplable en esta instancia que tendrá el nodo del plano de control y el nodo de trabajo en una instancia. Donde normalmente separaría esos nodos.

```
C:\Users\Nicolas>minikube start
* minikube v1.25.2 en Microsoft Windows 10 Home Single Language 10.0.19044 Build 19044
* Controlador docker seleccionado automáticamente. Otras opciones: virtualbox, ssh
* Starting control plane minikube in cluster minikube
* Pulling base image ...
  > gcr.io/k8s-minikube/kicbase: 320.68 MiB / 379.06 MiB  84.60% 5.60 MiB p/s
  > gcr.io/k8s-minikube/kicbase: 379.06 MiB / 379.06 MiB 100.00% 5.06 MiB p/
* Creando docker container (CPUs=2, Memory=3800MB) ...| E0424 16:50:35.480849 17584 kic.go:267] icacfs failed applying
permissions - err - [%!s(<nil>)], output - [archivo procesado: C:\Users\Nicolas\.minikube\machines\minikube\id_rsa
Se procesaron correctamente 1 archivos; error al procesar 0 archivos]
* Preparando Kubernetes v1.23.3 en Docker 20.10.12...
  - kubelet.housekeeping-interval=5m
  - Generando certificados y llaves
  - Iniciando plano de control
  - Configurando reglas RBAC...
* Verifying Kubernetes components...
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
* Complementos habilitados: storage-provisioner, default-storageclass
* Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
C:\Users\Nicolas>
```

Minikube es una excelente herramienta debido a los complementos disponibles, la capacidad de implementar un clúster con un comando simple que incluye todos los complementos necesarios desde el principio.

A continuación, puede ver una lista de esos complementos:

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Nicolas>minikube addons list
```

ADDON NAME	PROFILE	STATUS	MAINTAINER
ambassador	minikube	disabled	third-party (ambassador)
auto-pause	minikube	disabled	google
csi-hostpath-driver	minikube	disabled	kubernetes
dashboard	minikube	disabled	kubernetes
default-storageclass	minikube	enabled	kubernetes
efk	minikube	disabled	third-party (elastic)
freshpod	minikube	disabled	google
gcp-auth	minikube	disabled	google
gvisor	minikube	disabled	google
helm-tiller	minikube	disabled	third-party (helm)
ingress	minikube	disabled	unknown (third-party)
ingress-dns	minikube	disabled	google
istio	minikube	disabled	third-party (istio)
istio-provisioner	minikube	disabled	third-party (istio)
kong	minikube	disabled	third-party (Kong HQ)
kubevirt	minikube	disabled	third-party (kubevirt)
logviewer	minikube	disabled	unknown (third-party)
metallb	minikube	disabled	third-party (metallb)
metrics-server	minikube	disabled	kubernetes
nvidia-driver-installer	minikube	disabled	google
nvidia-gpu-device-plugin	minikube	disabled	third-party (nvidia)
olm	minikube	disabled	third-party (operator framework)
pod-security-policy	minikube	disabled	unknown (third-party)
portainer	minikube	disabled	portainer.io
registry	minikube	disabled	google
registry-aliases	minikube	disabled	unknown (third-party)
registry-creds	minikube	disabled	third-party (upmc enterprises)
storage-provisioner	minikube	enabled	google
storage-provisioner-gluster	minikube	disabled	unknown (third-party)
volumesnapshots	minikube	disabled	kubernetes

Se está definiendo en el proyecto alguna configuración adicional, <sup>1</sup>api server está configurado en el puerto 6433 en lugar de un puerto API aleatorio, defino el tiempo de ejecución del contenedor, sin embargo, la ventana acoplable es predeterminada y <sup>2</sup>CRI-O también está disponible. También se configura una versión específica de Kubernetes.

<sup>1</sup> El servidor de la API de Kubernetes valida y configura los datos para los objetos de la API, que incluyen pods, servicios, controladores de replicación y otros. El servidor API da servicio a las operaciones REST y proporciona la interfaz para el estado compartido del clúster a través del cual interactúan todos los demás componentes.

<sup>2</sup> CRI-O es una implementación de la **Container Runtime Interface (CRI)** para Kubernetes, que utiliza instancias y entornos en tiempo de ejecución (runtimes) de Open Container Initiative (OCI). La empresa Red Hat inició el proyecto en 2016 y se lo cedió a la Cloud Native Computing Foundation (CNCF) en la primavera de 2019. (<https://www.ionos.es/digitalguide/servidores/know-how/que-es-cri-o/>)

```
C:\WINDOWS\system32\cmd.exe - minikube start --addons volumesnapshost,csi-hostpath-driver --apiserver-port=6443 --container-...
C:\Users\Nicolas>minikube start --addons volumesnapshost,csi-hostpath-driver --apiserver-port=6443 --container-runtime=containerd -p mc-demo --kubernetes-version=1.21.2
* [mc-demo] minikube v1.25.2 en Microsoft Windows 10 Home Single Language 10.0.19044 Build 19044
* Controlador docker seleccionado automáticamente. Otras opciones: virtualbox, ssh
* Starting control plane node mc-demo in cluster mc-demo
* Pulling base image ...
* Descargando Kubernetes v1.21.2 ...
  > preloaded-images-k8s-v17-v1...: 566.06 MiB / 566.06 MiB 100.00% 5.72 MiB
* Creando docker container (CPUs=2, Memory=3800MB) ...- E0424 16:59:21.364273 6464 kic.go:267] icacfs failed applying permissions - err - [%!s(<nil>)], output - [archivo procesado: C:\Users\Nicolas\.minikube\machines\mc-demo\id_rsa
Se procesaron correctamente 1 archivos; error al procesar 0 archivos]
* Preparando Kubernetes v1.21.2 en containerd 1.4.12...
  - kubelet.housekeeping-interval=5m
  - kubelet.cni-conf-dir=/etc/cni/net.mk
  - Generando certificados y llaves
  - Iniciando plano de control
  - Configurando reglas RBAC...
* Configurando CNI CNI ...
* Verifying Kubernetes components...
! [WARNING] For full functionality, the 'csi-hostpath-driver' addon requires the 'volumesnapshots' addon to be enabled.
You can enable 'volumesnapshots' addon by running: 'minikube addons enable volumesnapshots'
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
  - Using image k8s.gcr.io/sig-storage/livenessprobe:v2.2.0
  - Using image k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.0.1
  - Using image k8s.gcr.io/sig-storage/csi-external-health-monitor-controller:v0.2.0
  - Using image k8s.gcr.io/sig-storage/csi-resizer:v1.1.0
  - Using image k8s.gcr.io/sig-storage/hostpathplugin:v1.6.0
  - Using image k8s.gcr.io/sig-storage/csi-snapshotter:v4.0.0
  - Using image k8s.gcr.io/sig-storage/csi-attacher:v3.1.0
  - Using image k8s.gcr.io/sig-storage/csi-external-health-monitor-agent:v0.2.0
  - Using image k8s.gcr.io/sig-storage/csi-provisioner:v2.1.0
! Executing "docker container inspect mc-demo --format={{.State.Status}}" took an unusually long time: 3.0442799s
* Restarting the docker service may improve performance.
* Verifying csi-hostpath-driver addon...
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Nicolas>minikube start --addons volumesnapshots,csi-hostpath-driver --apiserver-port=6443 --container-runtime=containerd -p mc-demo --kubernetes-version=1.21.2
* [mc-demo] minikube v1.25.2 en Microsoft Windows 10 Home Single Language 10.0.19044 Build 19044
* Controlador docker seleccionado automáticamente. Otras opciones: virtualbox, ssh
* Starting control plane node mc-demo in cluster mc-demo
* Pulling base image ...
* Descargando Kubernetes v1.21.2 ...
  > preloaded-images-k8s-v17-v1...: 566.06 MiB / 566.06 MiB 100.00% 5.72 MiB
* Creando docker container (CPUs=2, Memory=3800MB) ...- E0424 16:59:21.364273 6464 kic.go:267] icacIs failed applying permissions - err - [%!s(<nil>)], output - [archivo procesado: C:\Users\Nicolas\.minikube\machines\mc-demo\id_rsa
Se procesaron correctamente 1 archivos; error al procesar 0 archivos]
* Preparando Kubernetes v1.21.2 en containerd 1.4.12...
  - kubelet.housekeeping-interval=5m
  - kubelet.cni-conf-dir=/etc/cni/net.mk
  - Generando certificados y llaves
  - Iniciando plano de control
  - Configurando reglas RBAC...
* Configurando CNI CNI ...
* Verifying Kubernetes components...
! [WARNING] For full functionality, the 'csi-hostpath-driver' addon requires the 'volumesnapshots' addon to be enabled.
You can enable 'volumesnapshots' addon by running: 'minikube addons enable volumesnapshots'
  - Using image gcr.io/k8s-minikube/storage-provisioner:v5
  - Using image k8s.gcr.io/sig-storage/livenessprobe:v2.2.0
  - Using image k8s.gcr.io/sig-storage/csi-node-driver-registrar:v2.0.1
  - Using image k8s.gcr.io/sig-storage/csi-external-health-monitor-controller:v0.2.0
  - Using image k8s.gcr.io/sig-storage/csi-resizer:v1.1.0
  - Using image k8s.gcr.io/sig-storage/hostpathplugin:v1.6.0
  - Using image k8s.gcr.io/sig-storage/csi-snapshotter:v4.0.0
  - Using image k8s.gcr.io/sig-storage/csi-attacher:v3.1.0
  - Using image k8s.gcr.io/sig-storage/csi-external-health-monitor-agent:v0.2.0
  - Using image k8s.gcr.io/sig-storage/csi-provisioner:v2.1.0
! Executing "docker container inspect mc-demo --format={{.State.Status}}" took an unusually long time: 3.0442799s
* Restarting the docker service may improve performance.
* Verifying csi-hostpath-driver addon...
* Complementos habilitados: storage-provisioner, default-storageclass, csi-hostpath-driver
* Done! kubectI is now configured to use "mc-demo" cluster and "default" namespace by default

C:\Users\Nicolas>
```

Ahora estamos listos para implementar nuestro primer clúster de Kubernetes usando minikube. Sin embargo, también necesitará kubectI para interactuar con su clúster.

- Instalación de Chocolatey

Chocolatey es un administrador de paquetes para Windows diseñado para permitir a los usuarios descargar e instalar aplicaciones directamente de Internet con un simple comando para PowerShell o CMD

community.chocolatey.org/courses/installation/installing

## Basic Chocolatey Install

Chocolatey installs in seconds. You are just a few steps from running choco right now!

1. First, ensure that you are using an **administrative shell** - you can also install as a non-admin, check out Non-Administrative Installation.

2. Copy the text specific to your command shell below.

**NOTE:** Please inspect <https://community.chocolatey.org/install.ps1> prior to running any of these scripts to ensure safety. We already know it's safe, but you should verify the security and contents of **any** script from the internet you are not familiar with. All of these scripts download a remote PowerShell script and execute it on your machine. We take security very seriously. [Learn more about our security protocols.](#)

Install with cmd.exe    Install with powershell.exe

Install with cmd.exe

Run the following command:

```
> @"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -E:
```

3. Paste the copied text into your shell and press Enter.

4. Wait a few seconds for the command to complete.

5. If you don't see any errors, you are ready to use Chocolatey! Type **choco** or **choco -?** now, or see [Getting Started](#) for usage instructions.

chocolatey.org uses cookies to enhance the user experience of the site. [I accept](#)

```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19044.1645]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Nicolas>@"%SystemRoot%\System32\WindowsPowerShell\v1.0\powershell.exe" -NoProfile -InputFormat None -ExecutionPolicy Bypass -Command "iex ((New-Object System.Net.WebClient).DownloadString('https://community.chocolatey.org/install.ps1'))" && SET "PATH=%PATH%;%ALLUSERSPROFILE%\chocolatey\bin"
ADVERTENCIA: 'choco' was found at 'C:\ProgramData\chocolatey\bin\choco.exe'.
ADVERTENCIA: An existing Chocolatey installation was detected. Installation will not continue.
For security reasons, this script will not overwrite existing installations.

Please use choco upgrade chocolatey to handle upgrades of Chocolatey itself.
```

- Instalación de Kubectl con Chocolatey

Usa la herramienta de línea de comandos de Kubernetes, **kubectl**, para desplegar y gestionar aplicaciones en Kubernetes. Usando **kubectl**, puedes inspeccionar recursos del clúster; crear, eliminar, y actualizar componentes; explorar tu nuevo clúster y arrancar aplicaciones.

kubernetes.io/docs/tasks/tools/install-kubectl-windows/

kubernetes Documentation Kubernetes Blog Training Partners Community Case Studies Versions English

Search

Home  
Getting started  
Concepts  
Tasks  
Install Tools  
Install and Set Up kubectl on Linux  
Install and Set Up kubectl on macOS  
**Install and Set Up kubectl on Windows**  
Tutorials  
Reference  
Contribute

## Install on Windows using Chocolatey or Scoop

1. To install kubectl on Windows you can use either [Chocolatey](#) package manager or [Scoop](#) command-line installer.

choco scoop

```
choco install kubernetes-cli
```

2. Test to ensure the version you installed is up-to-date:

```
kubectl version --client
```

3. Navigate to your home directory:

```
# If you're using cmd.exe, run: cd %USERPROFILE%  
cd ~
```

4. Create the `.kube` directory:

```
mkdir .kube
```

[Edit this page](#)  
[Create child page](#)  
[Create an issue](#)  
[Print entire section](#)

Before you begin  
Install kubectl on Windows  
Install kubectl binary with curl on Windows  
Install on Windows using Chocolatey or Scoop  
Verify kubectl configuration  
Optional kubectl configurations and plugins  
Enable shell autocompletion  
Install kubectl convert plugin  
What's next

```
C:\WINDOWS\system32\cmd.exe  
location. See  
https://chocolatey.org/install#non-administrative-install for details.  
  
Do you want to continue?([Y]es/[N]o): Y  
  
Installing the following packages:  
kubernetes-cli  
By installing you accept licenses for the packages.  
Progress: Downloading kubernetes-cli 1.23.6... 100%  
kubernetes-cli not installed. An error occurred during installation:  
Acceso denegado a la ruta de acceso 'C:\ProgramData\chocolatey\lib\kubernetes-cli\legal'.  
kubernetes-cli package files install completed. Performing other installation steps.  
The install of kubernetes-cli was NOT successful.  
kubernetes-cli not installed. An error occurred during installation:  
Acceso denegado a la ruta de acceso 'C:\ProgramData\chocolatey\lib\kubernetes-cli\legal'.  
  
Chocolatey installed 0/1 packages. 1 packages failed.  
See the log for details (C:\ProgramData\chocolatey\logs\chocolatey.log).  
  
Failures  
- kubernetes-cli (exited 1) - kubernetes-cli not installed. An error occurred during installation:  
Acceso denegado a la ruta de acceso 'C:\ProgramData\chocolatey\lib\kubernetes-cli\legal'.  
  
C:\Users\Nicolas>kubectl version --client  
Client Version: version.Info{Major:"1", Minor:"22", GitVersion:"v1.22.5", GitCommit:"5c99e2ac2ff9a3c549d9ca665e7bc05a3e18f07e", GitTreeState:"clean", BuildDate:"2021-12-16T08:38:33Z", GoVersion:"go1.16.12", Compiler:"gc", Platform:"windows/amd64"}  
C:\Users\Nicolas>
```

Una vez que haya instalado kubectl, podemos interactuar con nuestro clúster con un comando simple como `kubectl get nodes`.

```
C:\Users\Nicolas>kubectl get nodes  
NAME        STATUS    ROLES                  AGE     VERSION  
mc-demo     Ready     control-plane,master   7m44s   v1.21.2  
  
C:\Users\Nicolas>
```

### What is kubectl?

We now have our minikube | Kubernetes cluster up and running and I have asked you to install both Minikube where I have explained at least what it does but I have not really explained what kubectl is and what it does.

kubectl is a cli that is used or allows you to interact with Kubernetes clusters, we are using it here for interacting with our minikube cluster but we would also use kubectl for interacting with our enterprise clusters across the public cloud.

We use kubectl to deploy applications, inspect and manage cluster resources. A much better [Overview of kubectl](#) can be found here on the Kubernetes official documentation.

kubectl interacts with the API server found on the Control Plane node which we briefly covered in an earlier post.

### kubectl cheat sheet

Along with the official documentation, I have also found myself with this page open all the time when looking for kubectl commands. [Unofficial Kubernetes](#)

### kubectl cheat sheet

Along with the official documentation, I have also found myself with this page open all the time when looking for kubectl commands. [Unofficial Kubernetes](#)

Listing Resources	
kubectl get nodes	List all nodes in cluster
kubectl get namespaces	List all namespaces in cluster
kubectl get pods	List all pods in default namespace cluster
kubectl get pods -n name	List all pods in "name" namespace
kubectl get pods -n name	List all pods in "name" namespace

Creating Resources	
kubectl create namespace name	Create a namespace called "name"
kubectl create -f [filename]	Create a resource from a JSON or YAML file:

Editing Resources	
kubectl edit svc/servicename	To edit a service

More detail on Resources	
kubectl describe nodes	display the state of any number of resources in detail,



Delete Resources	
kubectl delete pod	Remove resources, this can be from stdin or file

You will find yourself wanting to know the short names for some of the kubectl commands, for example `-n` is the short name for `namespace` which makes it easier to type a command but also if you are scripting anything you can have much tidier code.

Short name	Full name
csr	certificatesigningrequests
cs	componentstatuses
cm	configmaps
ds	daemonsets
deploy	deployments
ep	endpoints
ev	events
hpa	horizontalpodautoscalers
ing	ingresses
limits	limitranges
ns	namespaces
no	nodes

pvc	persistentvolumeclaims
pv	persistentvolumes
po	pods
pdb	poddisruptionbudgets
psp	podsecuritypolicies
rs	replicasets
rc	replicationcontrollers
quota	resourcequotas
sa	serviceaccounts
svc	services

Algunos ejemplos de las salidas al aplicar los comandos son:

```

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Versión 10.0.19044.1645]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\Nicolas>kubectl get nodes
NAME          STATUS    ROLES          AGE   VERSION
mc-demo       Ready     control-plane,master  58m   v1.21.2

C:\Users\Nicolas>kubectl get pods
No resources found in default namespace.

C:\Users\Nicolas>kubectl get namespaces
NAME          STATUS    AGE
default       Active    58m
kube-node-lease  Active    58m
kube-public    Active    58m
kube-system    Active    58m

```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Nicolas>kubectl describe nodes
Name: mc-demo
Roles: control-plane,master
Labels: beta.kubernetes.io/arch=amd64
        beta.kubernetes.io/os=linux
        kubernetes.io/arch=amd64
        kubernetes.io/hostname=mc-demo
        kubernetes.io/os=linux
        minikube.k8s.io/commit=362d5fdc0a3dbec389b3d3f1034e8023e72bd3a7
        minikube.k8s.io/name=mc-demo
        minikube.k8s.io/primary=true
        minikube.k8s.io/updated_at=2022_04_24T17_00_10_0700
        minikube.k8s.io/version=v1.25.2
        node-role.kubernetes.io/control-plane=
        node-role.kubernetes.io/master=
        node.kubernetes.io/exclude-from-external-load-balancers=
        topology.hostpath.csi/node=mc-demo
Annotations: csi.volume.kubernetes.io/nodeid: {"hostpath.csi.k8s.io":"mc-demo"}
              kubeadm.alpha.kubernetes.io/cri-socket: /run/containerd/containerd.sock
              node.alpha.kubernetes.io/ttl: 0
              volumes.kubernetes.io/controller-managed-attach-detach: true
CreationTimestamp: Sun, 24 Apr 2022 17:00:03 -0500
Taints: <none>
Unschedulable: false
Lease:
  HolderIdentity: mc-demo
  AcquireTime: <unset>
  RenewTime: Sun, 24 Apr 2022 17:58:51 -0500
Conditions:
  Type      Status  LastHeartbeatTime      LastTransitionTime      Reason
```

```
C:\WINDOWS\system32\cmd.exe
Type      Status  LastHeartbeatTime      LastTransitionTime      Reason
Message
-----
MemoryPressure False   Sun, 24 Apr 2022 17:57:48 -0500   Sun, 24 Apr 2022 16:59:56 -0500   KubeletHasSufficientMemory
  kubelet has sufficient memory available
DiskPressure   False   Sun, 24 Apr 2022 17:57:48 -0500   Sun, 24 Apr 2022 16:59:56 -0500   KubeletHasNoDiskPressure
  kubelet has no disk pressure
PIDPressure    False   Sun, 24 Apr 2022 17:57:48 -0500   Sun, 24 Apr 2022 16:59:56 -0500   KubeletHasSufficientPID
  kubelet has sufficient PID available
Ready          True    Sun, 24 Apr 2022 17:57:48 -0500   Sun, 24 Apr 2022 17:00:54 -0500   KubeletReady
  kubelet is posting ready status
Addresses:
  InternalIP: 192.168.58.2
  Hostname: mc-demo
Capacity:
  cpu: 8
  ephemeral-storage: 263174212Ki
  hugepages-1Gi: 0
  hugepages-2Mi: 0
  memory: 12192248Ki
  pods: 110
Allocatable:
  cpu: 8
  ephemeral-storage: 263174212Ki
  hugepages-1Gi: 0
  hugepages-2Mi: 0
  memory: 12192248Ki
  pods: 110
System Info:
```

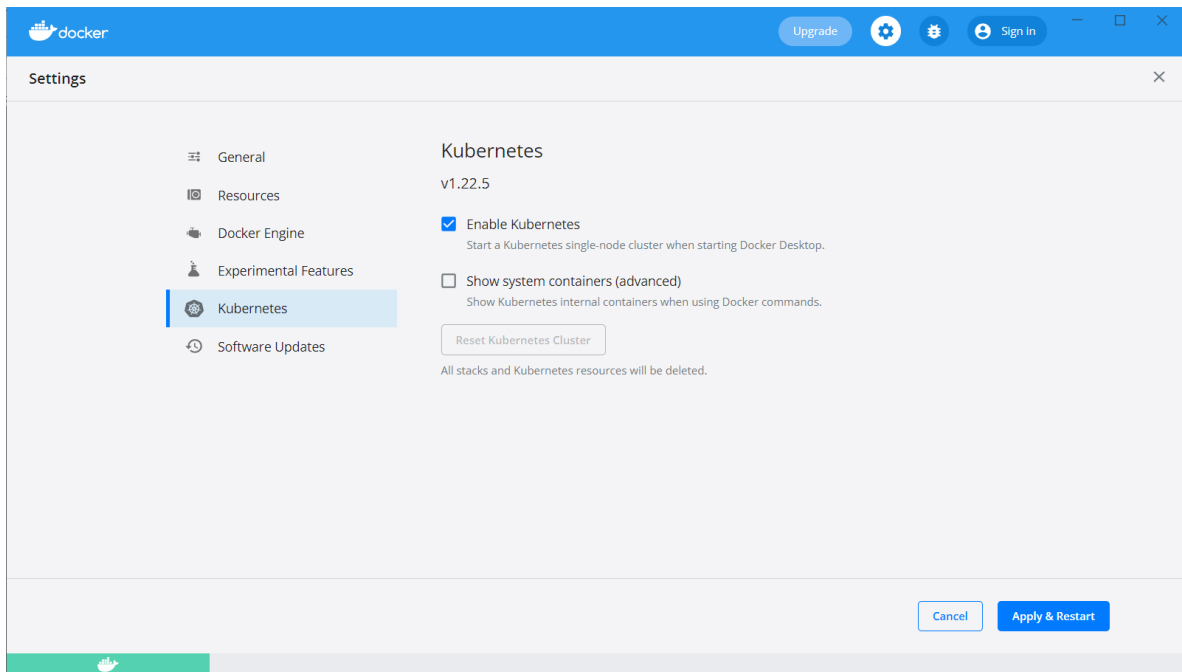
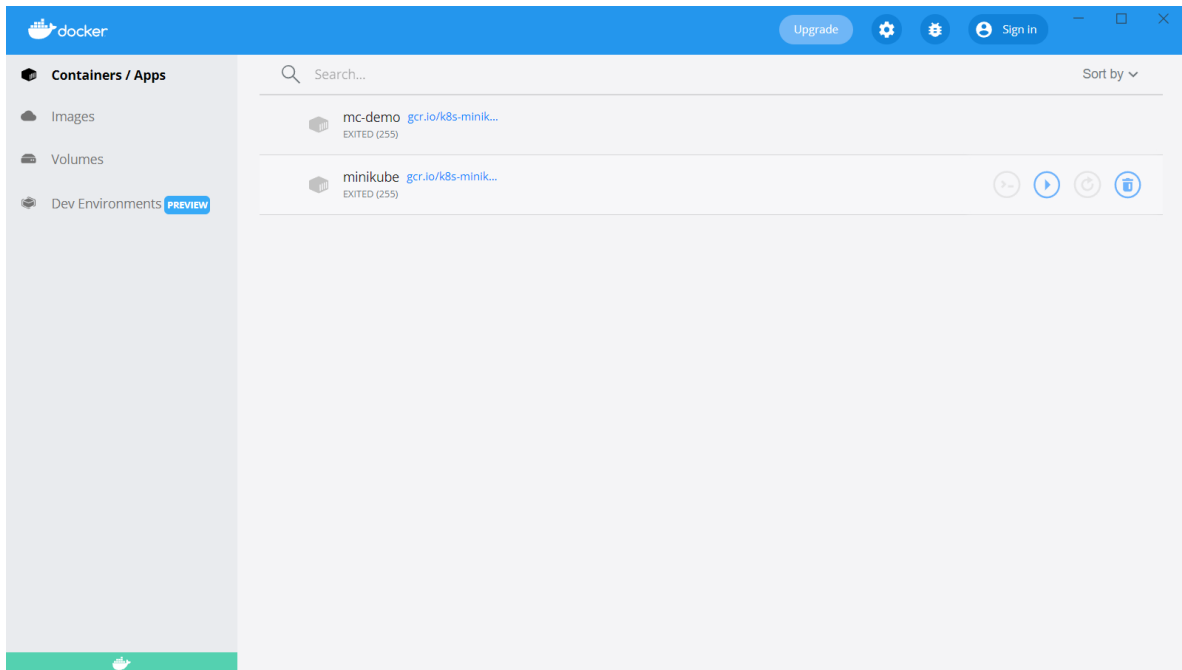
```
C:\WINDOWS\system32\cmd.exe
System Info:
Machine ID: b6a262faae404a5db719705fd34b5c8b
System UUID: b6a262faae404a5db719705fd34b5c8b
Boot ID: eca43ad4-4241-41e8-bcb3-8d5d19d5c083
Kernel Version: 5.10.16.3-microsoft-standard-WSL2
OS Image: Ubuntu 20.04.2 LTS
Operating System: linux
Architecture: amd64
Container Runtime Version: containerd://1.4.12
Kubelet Version: v1.21.2
Kube-Proxy Version: v1.21.2
PodCIDR: 10.244.0.0/24
PodCIDRs: 10.244.0.0/24
Non-terminated Pods: (13 in total)
Namespace      Name      CPU Requests  CPU Limits  Memory Requests  Memory Limit
s Age
-----
kube-system    coredns-558bd4d5db-t8bx9    100m (1%)    0 (0%)    70Mi (0%)    170Mi (1%)
58m
kube-system    csi-hostpath-attacher-0      0 (0%)       0 (0%)    0 (0%)       0 (0%)
58m
kube-system    csi-hostpath-provisioner-0    0 (0%)       0 (0%)    0 (0%)       0 (0%)
58m
kube-system    csi-hostpath-resizer-0       0 (0%)       0 (0%)    0 (0%)       0 (0%)
58m
kube-system    csi-hostpath-snapshotter-0    0 (0%)       0 (0%)    0 (0%)       0 (0%)
58m
kube-system    csi-hostpathplugin-0         0 (0%)       0 (0%)    0 (0%)       0 (0%)
58m
```

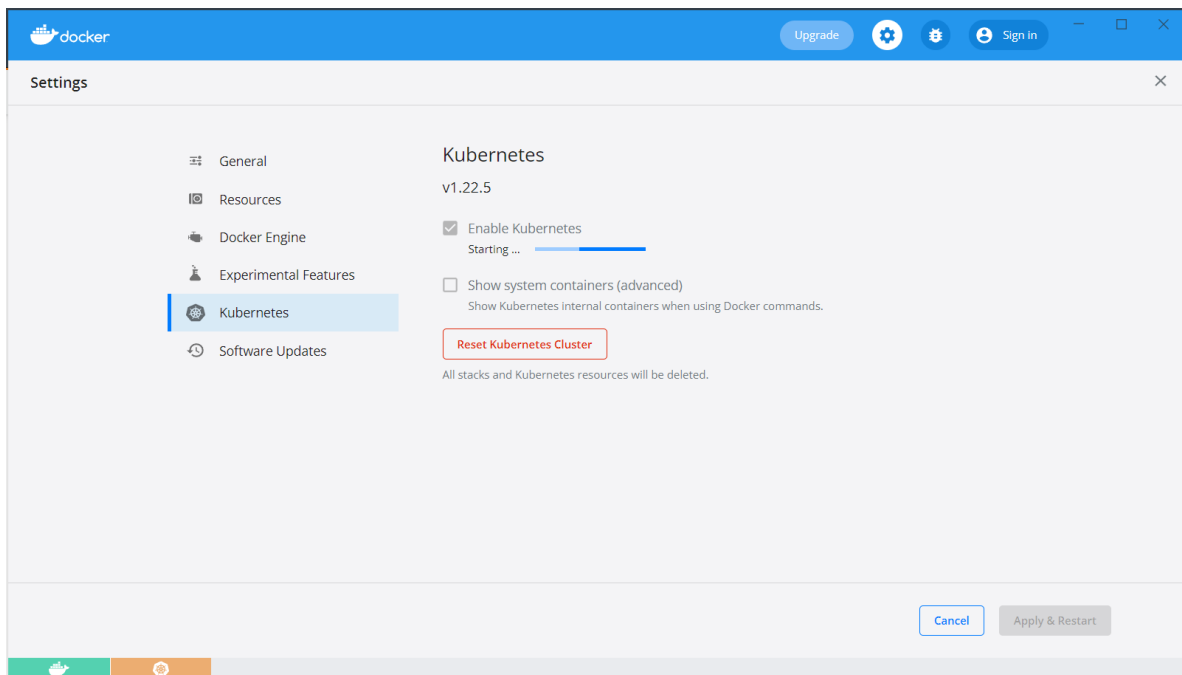
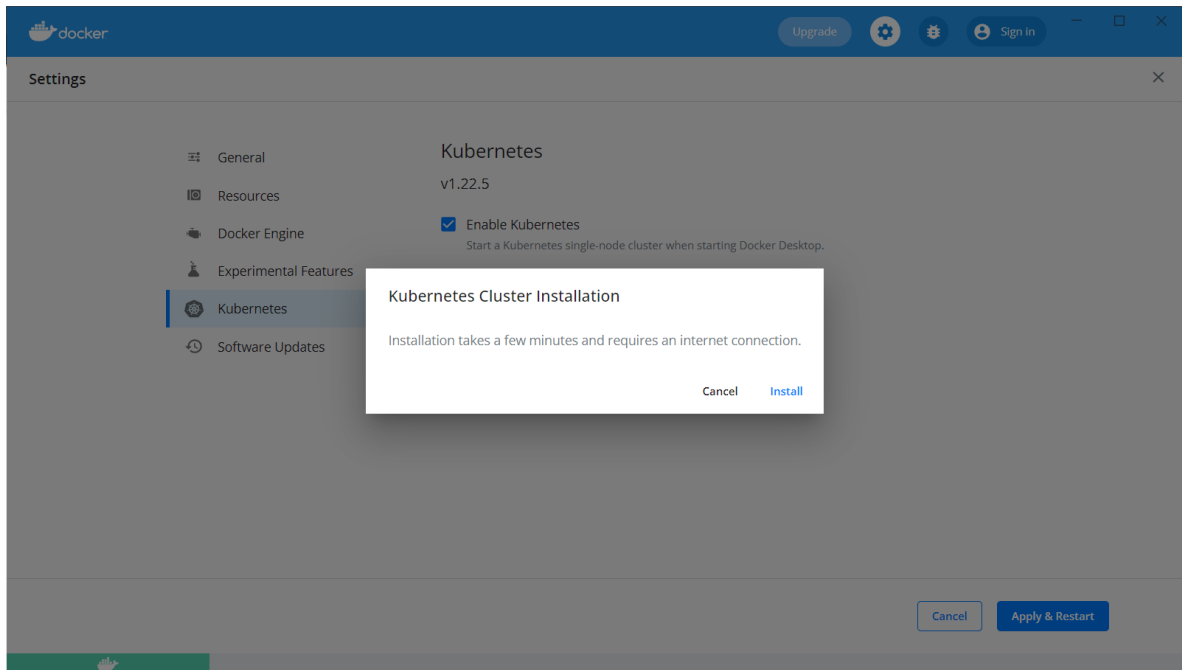
- Acceder al panel de Kubernetes

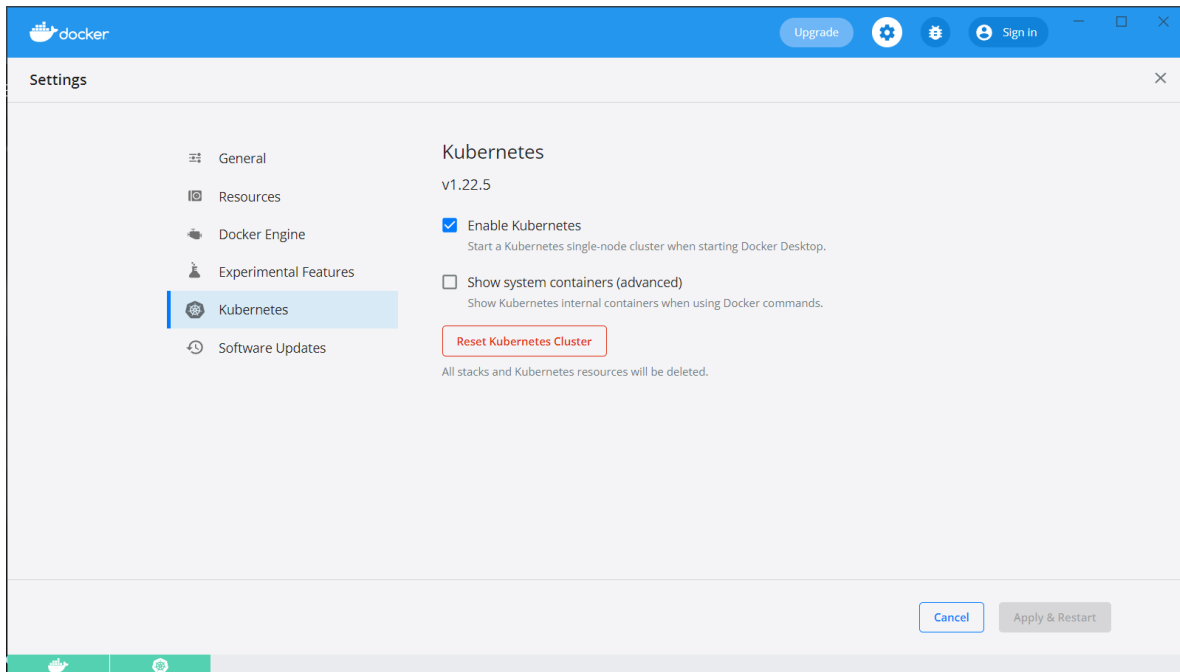
Dashboard es una interfaz de usuario de Kubernetes basada en la web. Puede usar Dashboard para implementar aplicaciones en contenedores en un clúster de Kubernetes, solucionar problemas de su aplicación en contenedores y administrar los recursos del clúster. Puede usar Dashboard para obtener una descripción general de las aplicaciones que se ejecutan en su clúster, así como para crear o modificar recursos individuales de Kubernetes (como implementaciones, trabajos, DaemonSets, etc.). Por ejemplo, puede escalar una implementación, iniciar una actualización progresiva, reiniciar un pod o implementar nuevas aplicaciones mediante un asistente de implementación.

El panel también proporciona información sobre el estado de los recursos de Kubernetes en su clúster y sobre cualquier error que pueda haber ocurrido.

- a) Configurar Docker para kubernetes





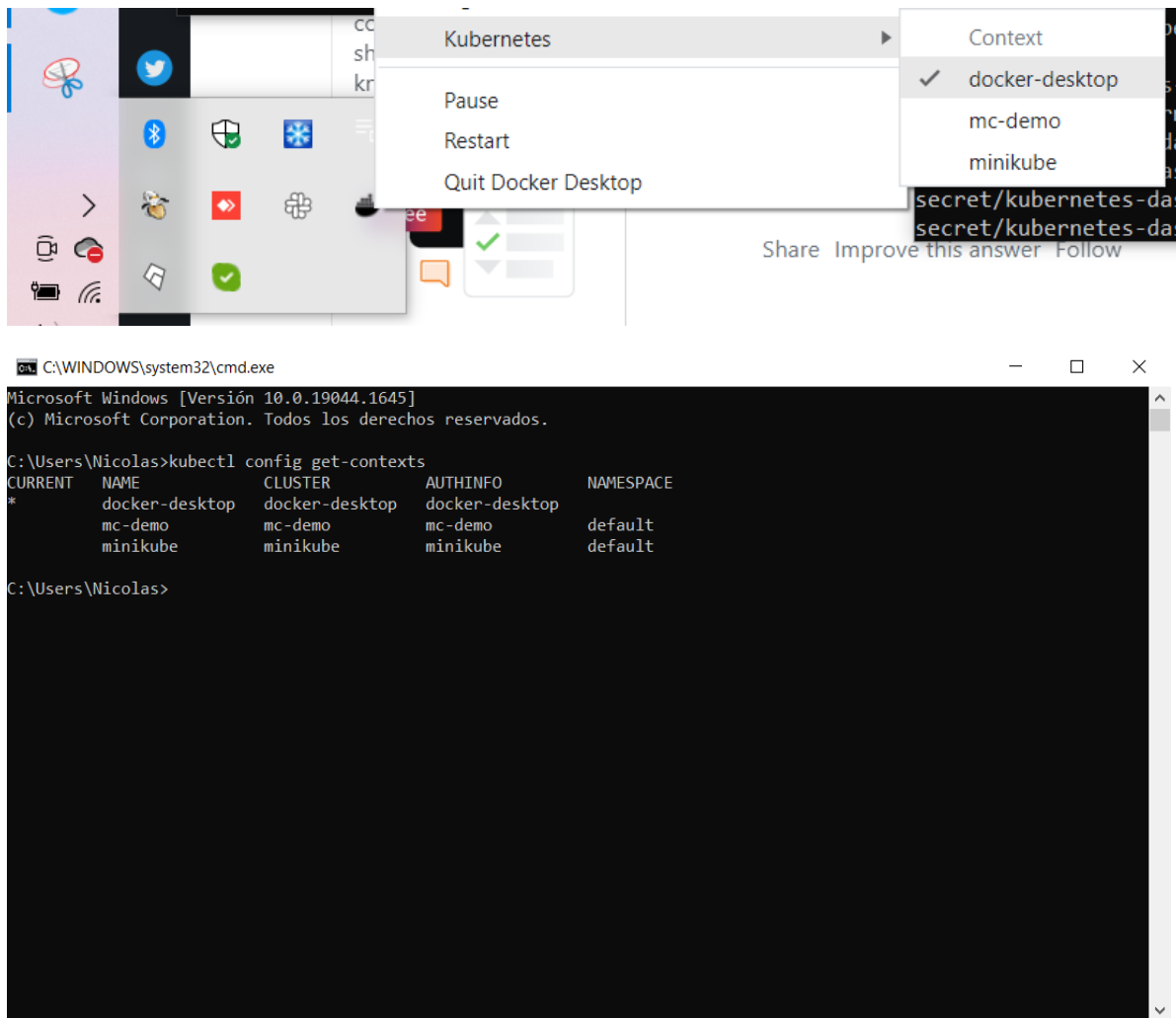


En Kubernetes, se usa un contexto para agregar parámetros de acceso en un archivo kubeconfig con nombres fáciles de recordar. El clúster, el espacio de nombres y el usuario son los tres parámetros que contiene cada contexto.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\Nicolas>kubect1 config get-contexts
CURRENT  NAME          CLUSTER          AUTHINFO          NAMESPACE
*        docker-desktop  docker-desktop  docker-desktop    default
minikube  minikube      minikube        minikube          default

C:\Users\Nicolas>
```



b) Iniciando el servidor y configurando los archivos relacionados con el dashboard.

```
C:\WINDOWS\system32\cmd.exe - kubect1 proxy
C:\Users\Nicolas>kubect1 proxy
Starting to serve on 127.0.0.1:8001
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\Nicolas>kubect1 get nodes
NAME                STATUS    ROLES    AGE     VERSION
docker-desktop      Ready    control-plane,master  5m38s   v1.22.5

C:\Users\Nicolas>kubect1 apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.5.0/aio/deploy/recommended.yaml
namespace/kubernetes-dashboard created
serviceaccount/kubernetes-dashboard created
service/kubernetes-dashboard created
secret/kubernetes-dashboard-certs created
secret/kubernetes-dashboard-csrf created
secret/kubernetes-dashboard-key-holder created
configmap/kubernetes-dashboard-settings created
role.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrole.rbac.authorization.k8s.io/kubernetes-dashboard created
rolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
clusterrolebinding.rbac.authorization.k8s.io/kubernetes-dashboard created
deployment.apps/kubernetes-dashboard created
service/dashboard-metrics-scraper created
deployment.apps/dashboard-metrics-scraper created
```

c) Acceder al dashboard desde el navegador.

Poner en el navegador

<http://localhost:8001/api/v1/namespaces/kubernetes-dashboard/services/https:kubernetes-dashboard:/proxy/>





