

Instalación de Kubernetes



KEEPCODING
Tech School

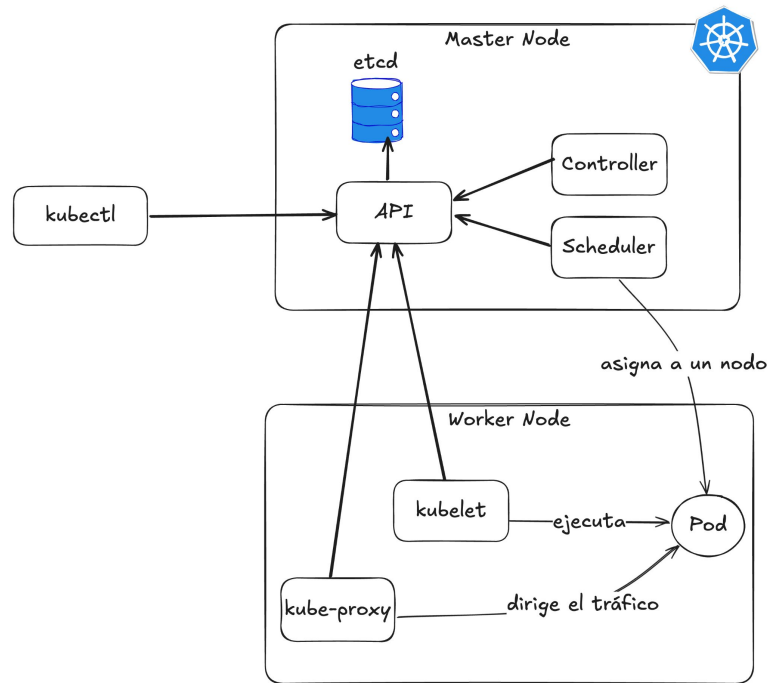
■ Anteriormente...

- Introducción a Kubernetes
 - Kubernetes es una plataforma de orquestación de contenedores que permite automatizar el despliegue, escalabilidad y operación de aplicaciones.
 - Funciona como un sistema distribuido y extensible que gestiona cargas de trabajo en contenedores.
 - Principales ventajas de Kubernetes:
 - Escalabilidad automática
 - Alta disponibilidad
 - Resiliencia



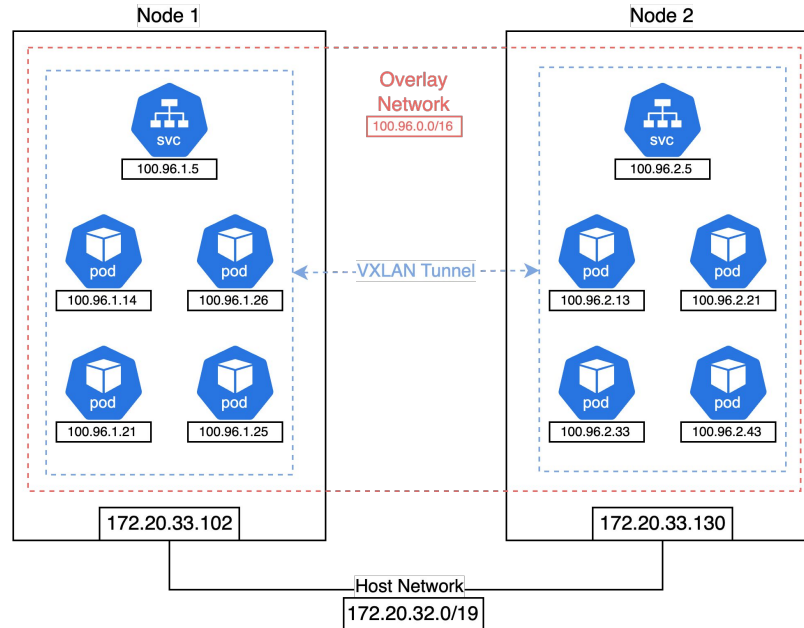
Anteriormente...

- Arquitectura de Kubernetes
 - Nodos master y worker:
 - Master: Gestiona el cluster y expone la API.
 - Workers: Donde se ejecutan nuestras aplicaciones.
 - Componentes clave:
 - etcd (base de datos key/value).
 - kube-apiserver, scheduler, controller, y kube-proxy.
 - Red y conectividad mediante CNI (Plugins de red).



Anteriormente...

- Arquitectura de Kubernetes
 - Red y conectividad mediante CNI (Plugins de red).



■ Instalación de Kubernetes

- En esta sección:
 - Tipos de instalación y herramientas de instalación
 - kubectl - instalación, configuración, plugins y extras.
 - Instalación Minikube
 - Instalación GKE
 - Ejercicio



Tipos de Instalación / Herramientas

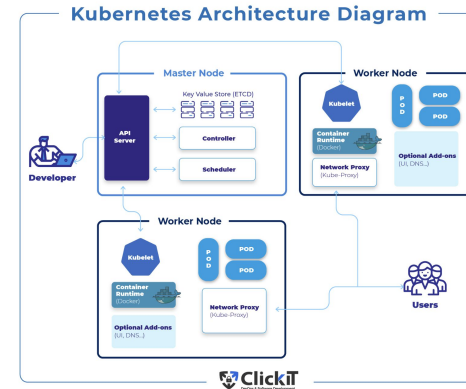
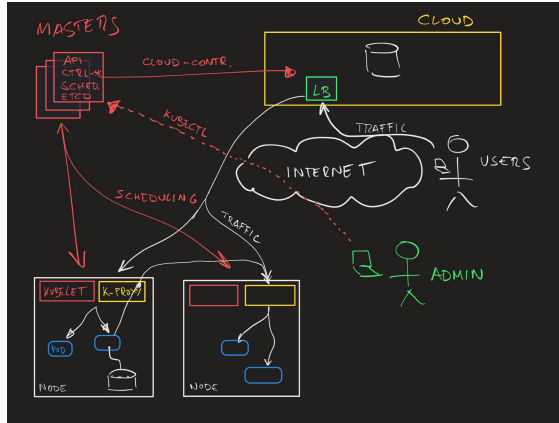


KEEPCODING

Tech School

■ Instalación de Kubernetes

- ¿En qué consiste la instalación de Kubernetes a nivel teórico?
 - Desplegar máquinas (masters y workers)
 - Instalar y configurar software (CRI, Kubernetes, CNI, ...)
 - Integración con Cloud (cloud-controller-manager)
 - Configurar el cliente (kubectl) para conectarnos al cluster.



■ Instalación de Kubernetes

- Existen muchísimas formas de instalar kubernetes
 - <https://kubernetes.io/docs/setup/>
 - Producción:
 - Como servicio (as a service): GKE (google cloud), EKS (aws), AKS (azure).
 - Herramientas de Instalación:
 - [kubeadm](#) (herramienta oficial) / [kubespray](#)
 - [kops](#) (compatible con AWS y GCP) → Muy fácil y configurable
 - [Cluster API](#) → Operar múltiples clusters de forma declarativa
 - Soluciones [Turnkey](#) ("llavero") / hosted o instaladores.
 - [Full landscape](#) :)
 - Learning Environments (en local)
 - Minikube
 - Kind
 - kubeadm



■ Instalación de Kubernetes

- Para los más valientes :)

“Kubernetes the hard way”

<https://github.com/kelseyhightower/kubernetes-the-hard-way>

"The target audience for this tutorial is someone planning to support a production Kubernetes cluster and wants to understand how everything fits together."



kubeadm

Kubeadm es una herramienta para crear clústers de Kubernetes con facilidad, en un par de comandos.

Kubeadm se centra en el mantenimiento y la gestión de los clusters de kubernetes, no se preocupa de la instalación y configuración / bastionado del sistema operativo.

No provisiona infraestructura (VMs), es una herramienta ligera.

Puedes encontrar documentación al respecto en <https://kubernetes.io/docs/setup/production-environment/tools/kubeadm/install-kubeadm/>



kubespray

Kubespray se centra más en la configuración de las máquinas a nivel de sistema operativo además de lanzar el cluster.

*"Kubespray is a composition of **Ansible playbooks**, inventory, provisioning **tools**, and domain knowledge for generic OS/Kubernetes **clusters configuration** management tasks"*

Puedes encontrar documentación al respecto en kubespray.io





Kops es una herramienta del estilo de kubespray pero mucho mejor integrada con los servicios de AWS o Google Cloud.

Provisiona la infraestructura, configura, etc (lo hace todo).

Comparaciones entre kubespray / kops / kubeadm:

- https://github.com/kubernetes-sigs/kubespray/blob/master/docs/getting_started/comparisons.md



■ Kubernetes as a Service (KaaS)

- Proporcionado por las principales Clouds:
 - Google Cloud → GKE
 - Amazon Web Services → EKS
 - Microsoft Azure → AKS
- Gestión de los masters
- Escalado horizontal de workers
- Gestión de workers
- Actualización del sistema y parches de seguridad
- Integración con Cloud
 - Storage
 - Load Balancing
 - Logs



■ Comparativa entre KaaS

→ Comparación por Kubedex

- Control plane HA
- Tiempo creación del clúster
- Autoscaling
- Actualizaciones
- Coste
- Integraciones



■ Desarrollo y pruebas en local

- Minikube → Es el que usaremos durante el curso
- Kind
- Docker Desktop Kubernetes
- K3s



Minikube



■ Minikube

- Configura y levanta un cluster de Kubernetes en local (todos los componentes corriendo en la misma VM / docker).
- Soportado en Windows, Linux o Mac.
- Documentación: <https://minikube.sigs.k8s.io/docs/>
- [Requiere](#) tener instalado Docker o un sistema de virtualización como Virtualbox, KVM, HyperV, Hyperkit, VMWare, ...).
- Minikube crea el cluster a través de contenedores o máquinas virtuales con todo lo necesario.
- [Configuración](#) y [drivers](#).



■ Instalación de Minikube

<https://minikube.sigs.k8s.io/docs/start/>

```
# Descargamos e instalamos el binario siguiente las instrucciones
# Después...
$ minikube start

# instalamos kubectl
$ kubectl get nodes

# para poder ver los dockers como si estuviéramos en el worker
# apuntamos el comando docker al Docker engine de minikube
$ eval $(minikube docker-env)
```



GKE



■ GKE

- Servicio gestionado de Kubernetes de Google Cloud (Google Kubernetes Engine).
- Autoescalado de nodos.
- Vertical pod autoscaling
- Node auto-provisioning
- Auto-upgrade
- Auto-repair
- Monitorización integrada.
- Tipos de Clústers:
 - Zonal
 - Regional
 - Privado

```
gcloud container clusters create [CLUSTER_NAME] --zone
```



GKE - Instalación

Create cluster

Select the cluster mode that you want to use.

i Compare cluster modes to learn more about their differences. [COMPARE](#)

❖ GKE Standard

A pay-per-node Kubernetes cluster where you configure and manage your nodes.

[Learn more](#)

CONFIGURE

❖ GKE Autopilot

A pay-per-Pod Kubernetes cluster where GKE manages your nodes with minimal configuration required. [Learn more](#)

CONFIGURE

CANCEL

Google Cloud Platform

My First Project

Search products and resources

Create a Kubernetes cluster

[ADD NODE POOL](#)[REMOVE NODE POOL](#)[USE A SETUP GUIDE](#)

Cluster basics

NODE POOLS

default-pool

Nodes

Security

Metadata

CLUSTER

Automation

Networking

Security

Metadata

Features

Node pool details

The new cluster will be created with at least one node pool. A node pool is a template for groups of nodes created in this cluster. More node pools can be added and removed after cluster creation.

Name

default-pool

Control plane version - 1.21.6-gke.1500

Size

Number of nodes *

3

Pod address range limits the maximum size of the cluster. [Learn more](#)

☐ Enable autoscaling

☐ Specify node locations

Default: us-west1-b

Automation

☒ Enable auto-upgrade

☒ Enable auto-repair

Surge upgrade

Max surge *

1

Max unavailable *

0



GKE - Instalación

Google Cloud Platform

My First Project

Search products and resources

Create a Kubernetes cluster

ADD NODE POOL

REMOVE NODE POOL

USE A SETUP GUIDE

Cluster basics

NODE POOLS

default-pool

Nodes

Security

Metadata

CLUSTER

Automation

Networking

Security

Metadata

Features

The default node image for newly created clusters and node pools with version 1.21.6-gke.1500 is now Container-optimized OS with containerd. We recommend you to start testing and planning your migration. For more information, see [Migrating from the Docker runtime to the containerd runtime](#). Learn more about the different [node images](#).

Machine Configuration

Machine family

GENERAL-PURPOSE

COMPUTE-OPTIMIZED

MEMORY-OPTIMIZED

GPU

Machine types for common workloads, optimized for cost and flexibility

Series

E2

CPU platform selection based on availability

Machine type

e2-medium (2 vCPU, 4 GB memory)

vCPU

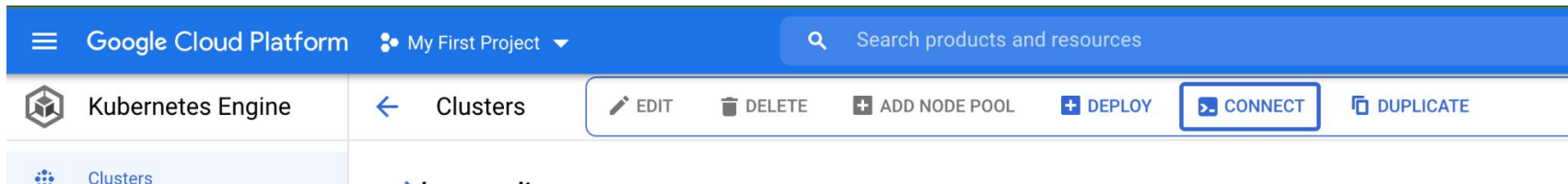
1 shared core

Memory

4 GB

© All rights reserved. www.keepcoding.io

GKE - Instalación



Connect to the cluster

You can connect to your cluster via command-line or using a dashboard.

Command-line access

Configure [kubectl](#) command line access by running the following command:

```
$ gcloud container clusters get-credentials keepcoding --zone us-west1-a --project chrome-plateau-338520
```

[RUN IN CLOUD SHELL](#)



■ GKE - Asignar rol cluster-admin a nuestro usuario

- Hace falta un plugin de gcloud para autenticación.

```
gcloud components install gke-gcloud-auth-plugin
```

- Cuando usemos clusters GKE por defecto nuestro usuario no tendrá el rol 'cluster-admin' del cluster. Para asignarlo:

```
kubectl create clusterrolebinding cluster-admin-binding \  
--clusterrole cluster-admin \  
--user $(gcloud config get-value account)
```



■ IaC con Terraform

- GKE (Google Cloud Platform):
 - <https://registry.terraform.io/modules/terraform-google-modules/kubernetes-engine/google/latest>
- EKS (AWS):
 - <https://registry.terraform.io/modules/terraform-aws-modules/eks/aws/latest>



■ Usar kubernetes sin instalar

<https://labs.play-with-k8s.com/>



Kubectl



■ kubectl

- Es el cliente que nos permitirá conectarnos al API de kubernetes.
- **Tenemos que saber manejarlo con soltura.**
- Cheatsheet:
 - <https://kubernetes.io/docs/reference/kubectl/cheatsheet/>

```
$ kubectl [command] [TYPE] [NAME] [flags]
```

- **command:** especifica la operación que queremos realizar sobre algún recurso, por ejemplo: “create”, “get”, “describe” o “delete” son bastante comunes.
- **TYPE:** indica el tipo de recurso, son case-insensitive y podemos usar su forma en singular, plural o abreviada. Por ejemplo: “pod”, “pods” o “po”.
- **NAME:** el nombre de un recurso del tipo elegido. Son case sensitive. Si se omite, el comando actúa sobre todos los recursos.



■ kubectl - Instalación

- <https://kubernetes.io/docs/tasks/tools/>
 - Opciones de instalación:
 - Descarga del binario directamente (y dejarlo en /usr/local/bin)
 - Mediante gestor de paquetes (yum / apt)
 - Autocompletado opcional

```
$ kubectl --help
$ kubectl version -client
```



■ kubectl - Configuración (.kube/config)

- La configuración de kubectl se encuentra en el fichero "config".
- Podemos encontrarlo por defecto en `$HOME/.kube/config`
- Contiene la información para conectarnos a nuestros clusters.
- Podemos tener varias configuraciones (incluso para el mismo cluster) separadas por **contextos**.
- Podemos tener también varios ficheros de configuración
- No es especialmente amigable :)

```
~$ kubectl config view
~$ kubectl config current-context
~$ kubectl config get-contexts
~$ kubectl config use-context CONTEXT
```



■ kubectl - Configuración (.kube/config)

- Configurar acceso a múltiples clusters:
 - ~/.kube/config & variable **KUBECONFIG**
 - opción --kubeconfig
- Uso de múltiples ficheros 'kubeconfig' (config):
 - Merge
- **OJO:** Es fácil perder toda la configuración si una mala herramienta sobrescribe el fichero config!!
- Si queremos apuntar a otro(s) fichero(s) de configuración:

```
$ kubectl --kubeconfig="/dir1/dir2/config-pruebas" ...  
# Otra opción  
$ export KUBECONFIG="/dir1/dir2/config-pruebas:/dir1/dir2/otra-config:..."  
$ kubectl ...
```



■ kubectl - Plugins / Addons

- Los plugins extienden la funcionalidad de kubectl.
- krew (kubectl plugin manager)
 - <https://github.com/kubernetes-sigs/krew/>
 - [Install link](#)
- kubectx + kubens
 - Herramientas para cambiar de contexto y namespace de forma rápida. **Súper útiles!**
 - <https://github.com/ahmetb/kubectx>
 - Se puede instalar vía krew o directamente.
- kube-ps1: <https://github.com/ionmosco/kube-ps1> (kubemon / kubemonoff para habilitar / deshabilitar)



■ kubectl - Alias

- Por comodidad suelo configurar los siguientes alias en la shell:
 - `k='kubectl'`
 - `kns='kubectl ns'` → Requiere plugin kubectlx
 - `kctx='kubectl ctx'` → Requiere plugin kubectlx



Práctica: Minikube



■ Ejercicio

- Instalación de **kubectl** en Máquina Virtual Linux.
- Instalación y arranque de **Minikube** en Máquina Virtual Linux.
- Links:
 - Kubectl: <https://kubernetes.io/docs/tasks/tools/#kubectl>
 - Minikube: <https://minikube.sigs.k8s.io/docs/start/>

```
# Descargamos e instalamos el binario siguiendo las instrucciones
# Después...
$ minikube start
$ kubectl get nodes
$ eval $(minikube docker-env)
```



■ Verificar que Kubernetes está corriendo

En una terminal ejecutamos:

```
$ minikube start
...
...

$ kubectl get nodes
NAME          STATUS    ROLES          AGE    VERSION
minikube      Ready    control-plane,master  16d    v1.22.3
```



Práctica: GKE



■ Práctica: Kubernetes as a service

- Crear cluster GKE o EKS con las especificaciones mínimas.
- Conectar a ese cluster con nuestro kubectl
 - Darnos permisos de administradores (necesario en GKE)
- Controlar mediante contextos la conexión de kubectl tanto al **minikube** creado anteriormente como al nuevo Kubernetes as a service.
- Practicar el cambio de contexto entre clusters.





KEEPCODING

Tech School

Madrid | Barcelona | Bogotá