

ORQUESTADORES DE CONTENEDORES

Presentado por:
Emilio García Riba
María Lucía Rivas Gil

ÍNDICE

- HISTORIA
- ORQUESTACIÓN
- CONTENEDORES / VM'S
- DOCKER
- KUBERNETES
- CONCLUSIÓN
- BIBLIOGRAFÍA

O1

HISTORIA

2004-2007



Google usa contenedores
(cgroups) a gran escala

2013



- Aparece Docker
- Lanzamiento de CoreOS linux

2015



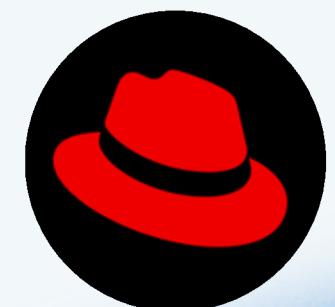
- Open Container Project
- Kubernetes 1.0 liberado

2018



- Actualizaciones para Kubernetes
- Google Kubernetes Engine
- VMware compra Heptio

2021

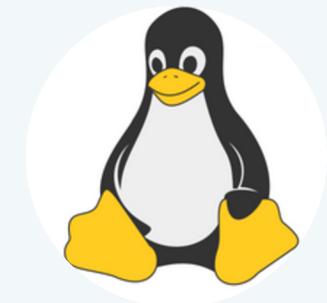


- Red Hat adquiere StackRox
- Google envía Knative a CNCF



2008

- cgroups se integran al kernel de Linux
- Se lanza LXC



2014

- Proyecto Kubernetes.
- Docker 1.0
- Canonical lanza LXD
- CoreOS introduce rkt



2017

- Kubernetes añade soporte total
- Microsoft Azure Container Instances ofrece gestión de contenedores para Linux
- Rancher adopta Kubernetes
- Cloud Foundry Container Runtime



2019

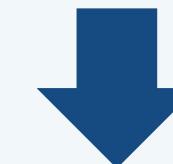
- Amazon EKS y Microsoft AKS
- IBM adquiere Red Hat
- Mirantis adquiere Docker Enterprise



ORQUESTACIÓN DE CONTENEDORES

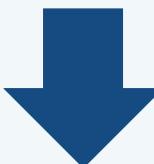
02

APROVISIONAMIENTO



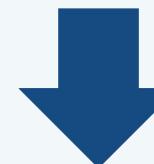
Los desarrolladores definen el estado deseado de su aplicación en un archivo de configuración declarativo (YAML o JSON). Este archivo es la "receta" que le indica a la herramienta qué imágenes de contenedores usar, los recursos necesarios, cómo deben conectarse y las políticas de control de versiones.

DESPLIEGUE

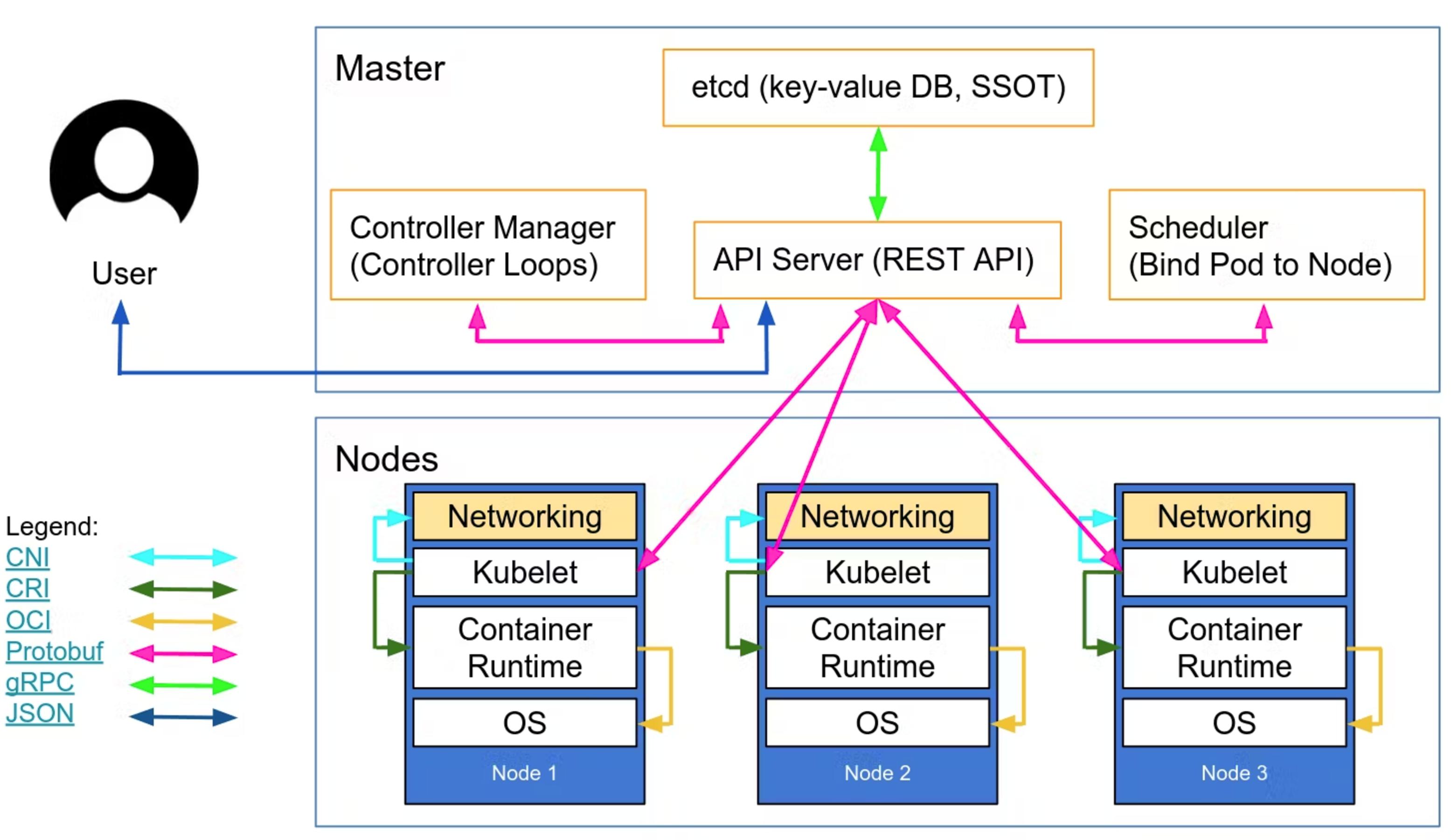


La herramienta de orquestación interpreta el archivo de configuración y programa inteligentemente los contenedores y sus réplicas en los hosts más adecuados. Para esta tarea, evalúa los requisitos de la aplicación, como la capacidad de CPU y memoria, para garantizar una distribución eficiente de la carga de trabajo.

GESTIÓN DEL CICLO
DE VIDA



Esta es la fase crucial donde mantiene el estado de la aplicación de forma continua, realizando tareas como escalar los contenedores para manejar el tráfico, balancear la carga de trabajo y reubicarlos automáticamente si un host falla. La herramienta también recopila datos para monitorizar la salud y el rendimiento del sistema.

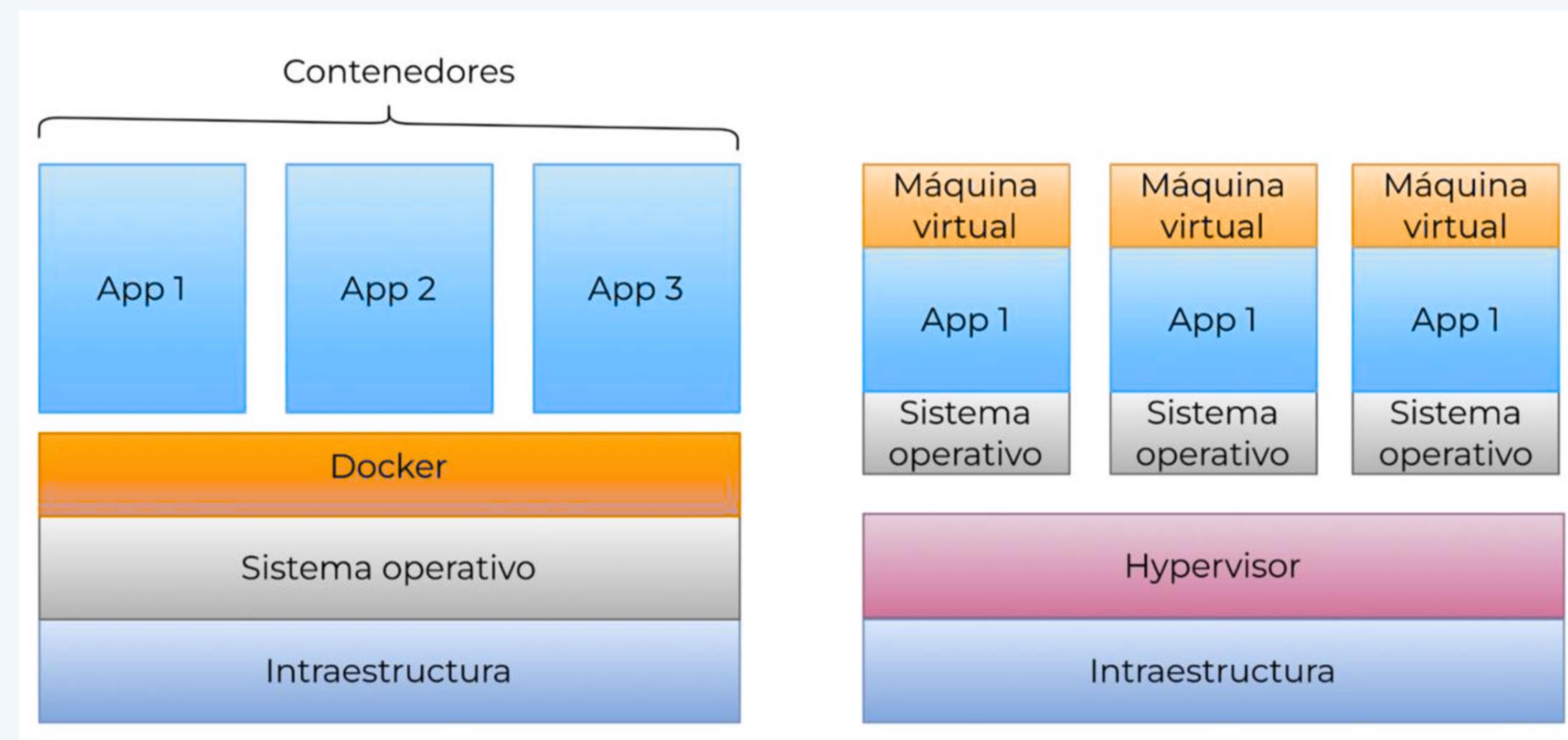


CONTENEDORES

VS

VM'S

Mientras que una máquina virtual (VM) virtualiza el hardware y ejecuta su propio sistema operativo completo (lo que la hace pesada y lenta), un contenedor virtualiza solo la capa del sistema operativo.



Característica	Contenedores	Máquinas Virtuales (VMs)
Nivel de Virtualización	Sistema Operativo	Hardware
Uso del Kernel	Comparten el kernel del host.	Cada VM tiene su propio kernel (SO invitado).
Uso de Recursos	Ligeros; solo incluyen la app y sus dependencias.	Pesados; incluyen un SO completo y sus bibliotecas.
Peso y Agilidad	Muy ligeros y rápidos.	Pesadas y más lentas para iniciar.
Aislamiento	Aislamiento de procesos (ligero).	Aislamiento completo (robusto).
Portabilidad	Alta; se mueven fácilmente entre entornos.	Requieren un hipervisor compatible para moverse.



01

Contenedores

Son unidades de software ligeras y portables que empaquetan una aplicación con todas sus dependencias. A diferencia de las máquinas virtuales, comparten el kernel del sistema operativo anfitrión, lo que las hace más eficientes y rápidas.

03

Docker Daemon

Es el servicio en segundo plano que gestiona el ciclo de vida de los contenedores, construyéndolos, ejecutándolos y distribuyéndolos. Se comunica con el cliente Docker y gestiona las imágenes y contenedores en el sistema.

02

Imágenes

Son plantillas de solo lectura que definen el entorno y las dependencias necesarias para ejecutar una aplicación. Se construyen a partir de un Dockerfile y sirven como base para crear contenedores.

04

Docker Hub

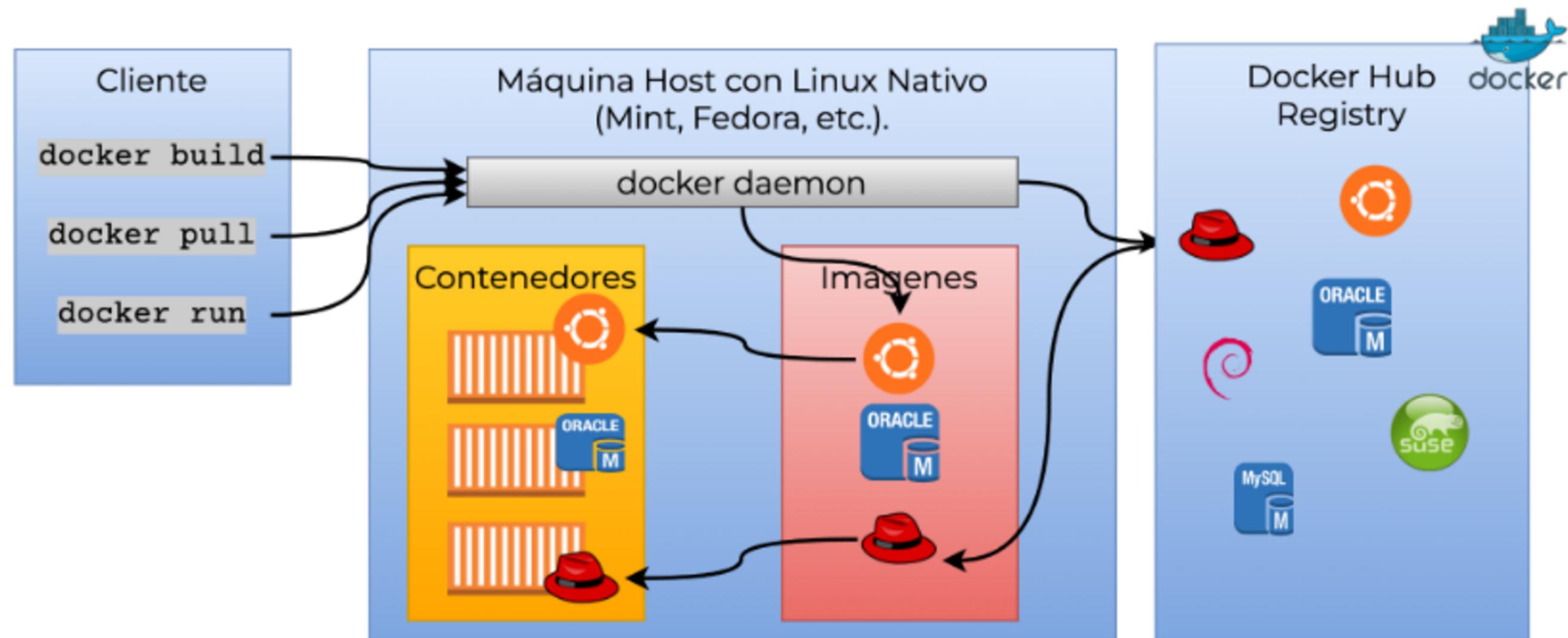
Es un registro público de imágenes donde los desarrolladores pueden almacenar, compartir y descargar imágenes preconstruidas. Sirve como repositorio central para distribuir aplicaciones empaquetadas en contenedores.

DOCKER

EL CONTENEDOR MAS USADO

COMO FUNCIONA

En el diagrama de abajo se describe el funcionamiento de docker:



KUBERNETES



Node (Nodo)

01

Un Node es una máquina física o virtual en un clúster de Kubernetes. Es la unidad básica de computación donde se ejecutan las cargas de trabajo (contenedores).



Master Node (Nodo Maestro)

02



El Master Node es el cerebro del clúster. Es el plano de control que toma todas las decisiones globales (como programar pods) y responde a eventos del clúster. Gestiona, pero no ejecuta, las cargas de trabajo de la aplicación.



Worker Node (Nodo Trabajador)

03

Un Worker Node es una máquina (física o virtual) donde se ejecutan realmente los contenedores. Su función es alojar los Pods y asegurar que estén en ejecución.



04

Pod

Un Pod es la unidad más pequeña y básica que se puede desplegar en Kubernetes. Es un "envoltorio" para uno o más contenedores que comparten almacenamiento, red y especificaciones sobre cómo ejecutarse. Es la unidad de escalamiento.



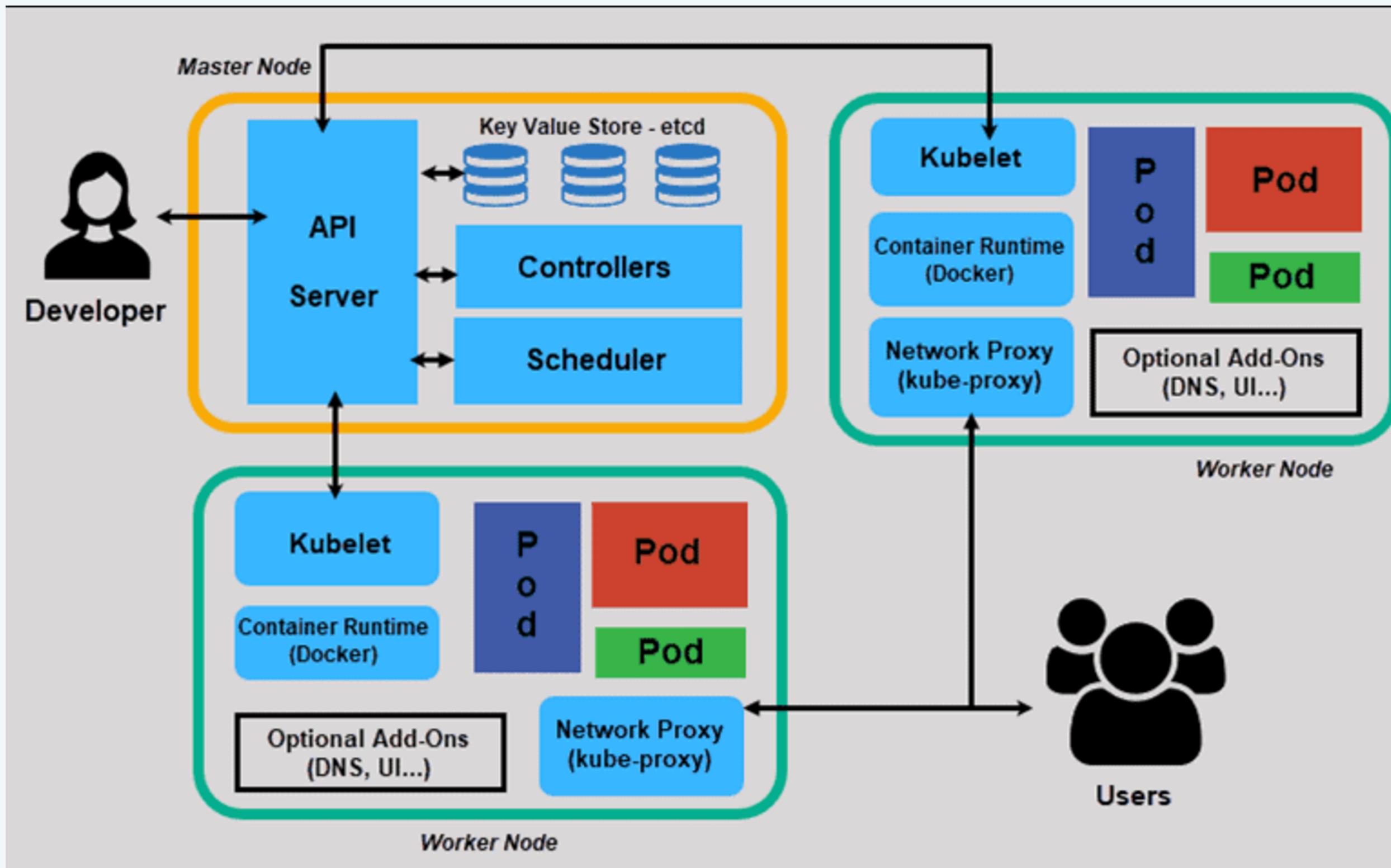
Kublet

05

El Kubelet es un agente que se ejecuta en cada Worker Node. Es el responsable de garantizar que los contenedores especificados en los Pods asignados a ese nodo se estén ejecutando y sanos. Se comunica con el Master Node.

05

COMO FUNCIONA



CONCLUSIÓN

06

File Actions Edit View Help

lap@lap-hplaptop14ck0xxx: ~

```
lap@lap-hplaptop14ck0xxx:~$ minikube start --driver=docker
🕒 minikube v1.37.0 on Ubuntu 24.04
  Using the docker driver based on user configuration
  Using Docker driver with root privileges
🕒 Starting "minikube" primary control-plane node in "minikube" cluster
  Pulling base image v0.0.48 ...
  Creating docker container (CPUs=2, Memory=3072MB) ...
  Preparing Kubernetes v1.34.0 on Docker 28.4.0 ...
  Configuring bridge CNI (Container Networking Interface) ...
  Verifying Kubernetes components...
    ■ Using image gcr.io/k8s-minikube/storage-provisioner:v5
  Enabled addons: storage-provisioner, default-storageclass
🕒 Done! kubectl is now configured to use "minikube" cluster and "default" namespace by default
lap@lap-hplaptop14ck0xxx:~$
```



BIBLIOGRAFIA

- Docker, Inc. (n.d.). Docker Docs. Retrieved September 13, 2025, from <https://docs.docker.com/>
- Docker, Inc. (n.d.). Manuals. Retrieved September 13, 2025, from <https://docs.docker.com/manuals/>
- Hostinger Tutoriales. (2025, July 2). ¿Qué es Docker? Funcionamiento y componentes básicos. Retrieved September 13, 2025, from <https://www.hostinger.com/es/tutoriales/que-es-docker>
- Rodríguez C., J. A. (n.d.). Manual-instalacion-docker.pdf [Unpublished manuscript]. Facultad de Ingeniería, Universidad Nacional Autónoma de México.
- Kubernetes Authors. (2024, 20 de octubre). Cluster Architecture. Kubernetes. Recuperado de <https://kubernetes.io/docs/concepts/architecture/>
- Kubernetes Authors. (2025, 19 de junio). Nodes. Kubernetes. Recuperado de <https://kubernetes.io/docs/concepts/architecture/nodes/>
- Amazon Web Services. (s. f.). What's the Difference Between Docker and a VM?. Recuperado de <https://aws.amazon.com/compare/the-difference-between-docker-vm/>