

# INTRODUCCIÓN AL DESARROLLO BACKEND CON NODEJS 2023



SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA  
UTN - FRC



\*UTN  
Facultad Regional Córdoba

Agencia  
CÓRDOBA  
JOVEN



CÓRDOBA  
entre todos

# Despliegue de una API en la nube

---

- Despliegue de una API en la Nube
- Plataformas de Nube
- Contenedores y Docker
- Creación de una Imagen Docker
- Registro de Imágenes
- Preparación del Proyecto
- Despliegue de la API
- Monitoreo y Registro



# Despliegue de la API Backend

- El despliegue de la API backend es una etapa crucial en el proceso de desarrollo de aplicaciones.
- Consiste en llevar la API desde el entorno de desarrollo hasta el entorno de producción.
- Proceso de Despliegue:
  - Pruebas y Validación.
  - Preparación de Entorno de Producción.
  - Empaquetado de la API.
  - Configuración de Entorno de Producción.
  - Implementación.
  - Verificación y Pruebas Post-Despliegue.
  - Monitorización y Mantenimiento: Monitorear constantemente el rendimiento y salud de la API en producción.
- Beneficios del Despliegue Continuo: Automatizar y agilizar el proceso de despliegue para lanzar cambios y actualizaciones de manera rápida y segura



# Introducción a las Plataformas de Nube

---



- En la actualidad, las plataformas de nube han transformado la manera en que las organizaciones diseñan, implementan y gestionan sus aplicaciones y servicios.
- Una plataforma de nube proporciona recursos de computación, almacenamiento y redes bajo demanda a través de Internet, permitiendo a las empresas escalar y administrar sus operaciones de manera flexible y eficiente.

# Ventajas y Beneficios de la Nube

---



- **Escalabilidad:** Las plataformas de nube permiten aumentar o disminuir los recursos según las necesidades cambiantes de la aplicación o el negocio, evitando costos innecesarios y garantizando un rendimiento óptimo.
- **Alta Disponibilidad:** Las aplicaciones desplegadas en la nube pueden aprovechar la distribución geográfica de los centros de datos del proveedor, lo que mejora la disponibilidad y la tolerancia a fallos.
- **Seguridad y Cumplimiento:** Los principales proveedores de nube implementan rigurosas medidas de seguridad y certificaciones para proteger los datos y garantizar el cumplimiento normativo.
- **Servicios Administrados:** Las plataformas de nube ofrecen una amplia variedad de servicios administrados, como bases de datos, colas de mensajes y servicios de identidad, que simplifican el desarrollo y la administración de aplicaciones.
- **Pago por Consumo:** El modelo de pago por consumo de la nube permite a las empresas pagar solo por los recursos que utilizan, lo que resulta en una mayor eficiencia y reducción de costos operativos.

# Principales Proveedores de Nube

---

- **Amazon Web Services (AWS):** Líder en el mercado de la nube, ofrece una amplia gama de servicios y una presencia global con múltiples centros de datos.
- **Microsoft Azure:** Una plataforma integral con una fuerte integración con productos y servicios de Microsoft, ideal para empresas que utilizan tecnologías Microsoft.
- **Google Cloud Platform (GCP):** Destaca por su enfoque en la inteligencia artificial y el aprendizaje automático, y ofrece soluciones innovadoras para diversas industrias.



# Tipos de Servicios en la Nube

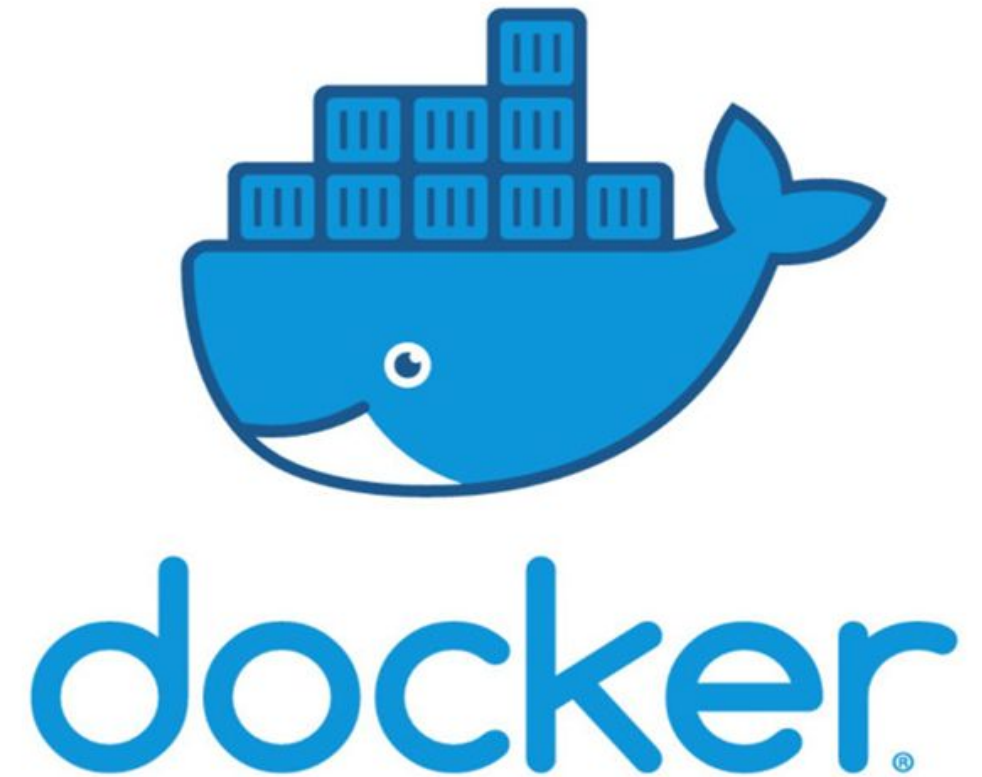
Private Cloud	IaaS Infrastructure as a Service	PaaS Platform as a Service	FaaS Function as a Service	SaaS Software as a Service
Function	Function	Function	Function	Function
Application	Application	Application	Application	Application
Runtime	Runtime	Runtime	Runtime	Runtime
Operating System	Operating System	Operating System	Operating System	Operating System
Virtualization	Virtualization	Virtualization	Virtualization	Virtualization
Server	Server	Server	Server	Server
Storage	Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking	Networking

Managed by the customer ■  
Managed by the provider ■



# ¿Qué es Docker?

---





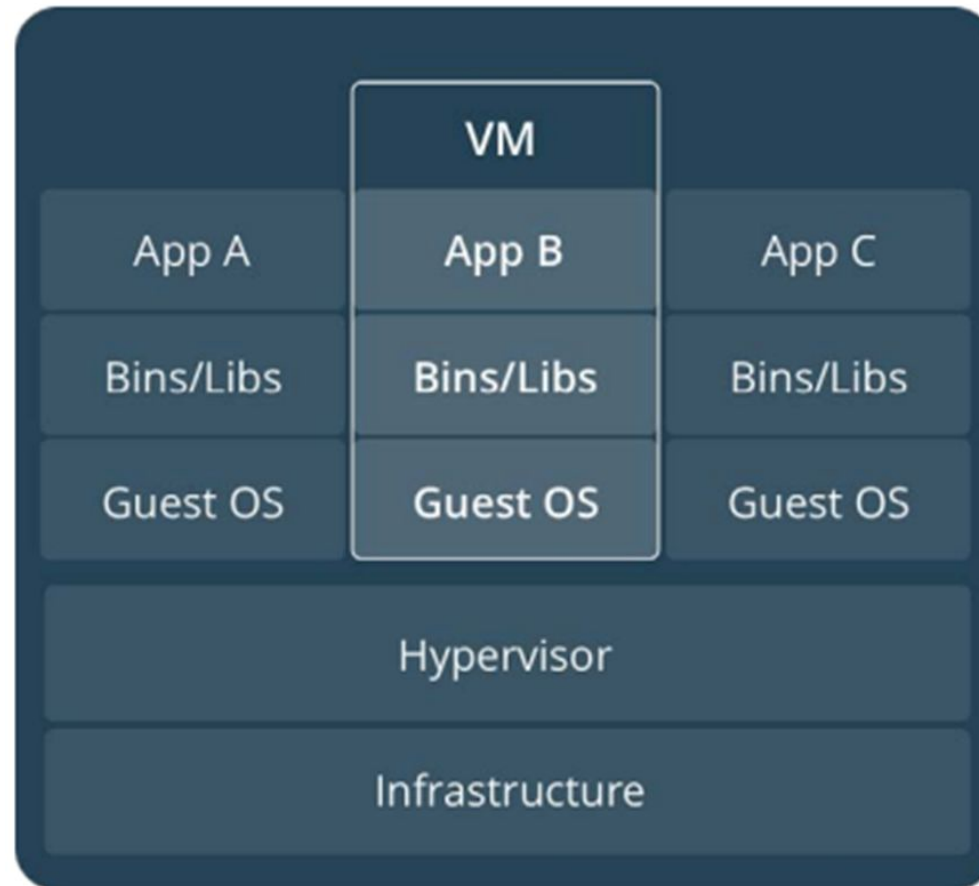
# ¿Qué es Docker?

---



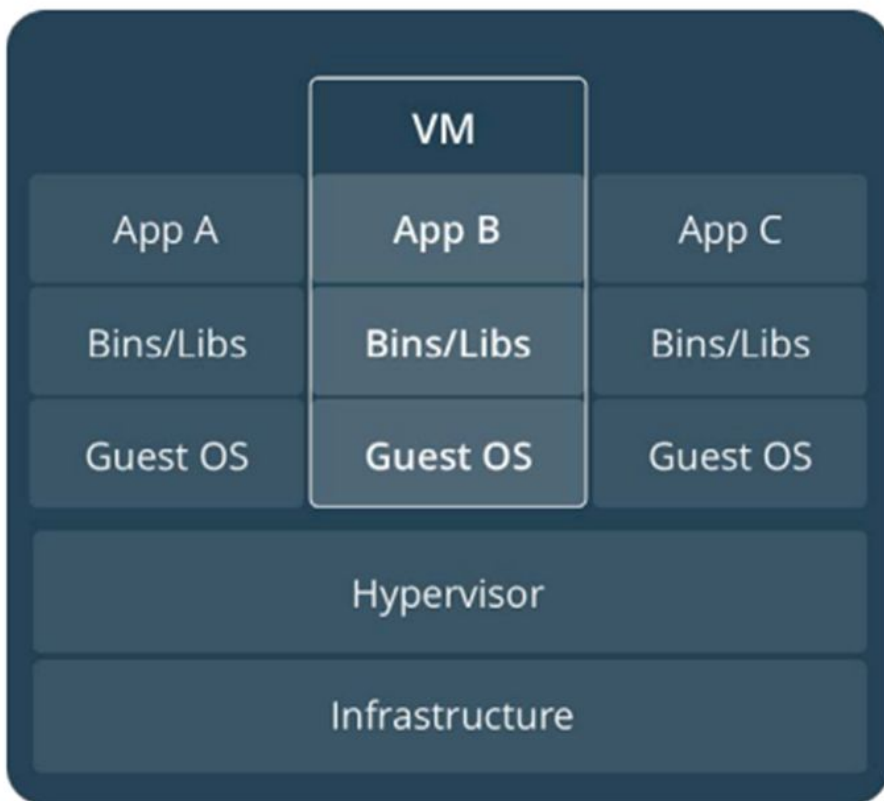
- Docker es una plataforma para que desarrolladores y administradores puedan desarrollar, desplegar y ejecutar aplicaciones en un entorno aislado denominado contenedor.
- Docker empaqueta software en unidades estandarizadas llamadas contenedores que incluyen todo lo necesario para que el software se ejecute (librerías, código, archivos de configuración, etc).

# Máquinas Virtuales - VM

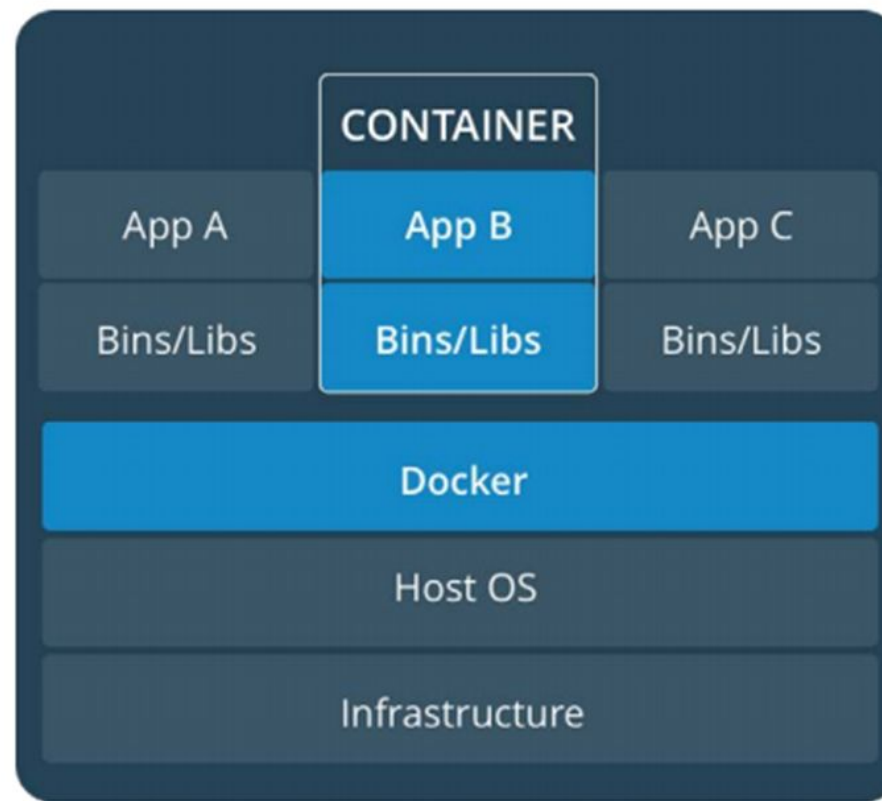


# VM vs Contenedores Docker

## Máquinas Virtuales



## Contenedores



# Contenedores - Ventajas

---

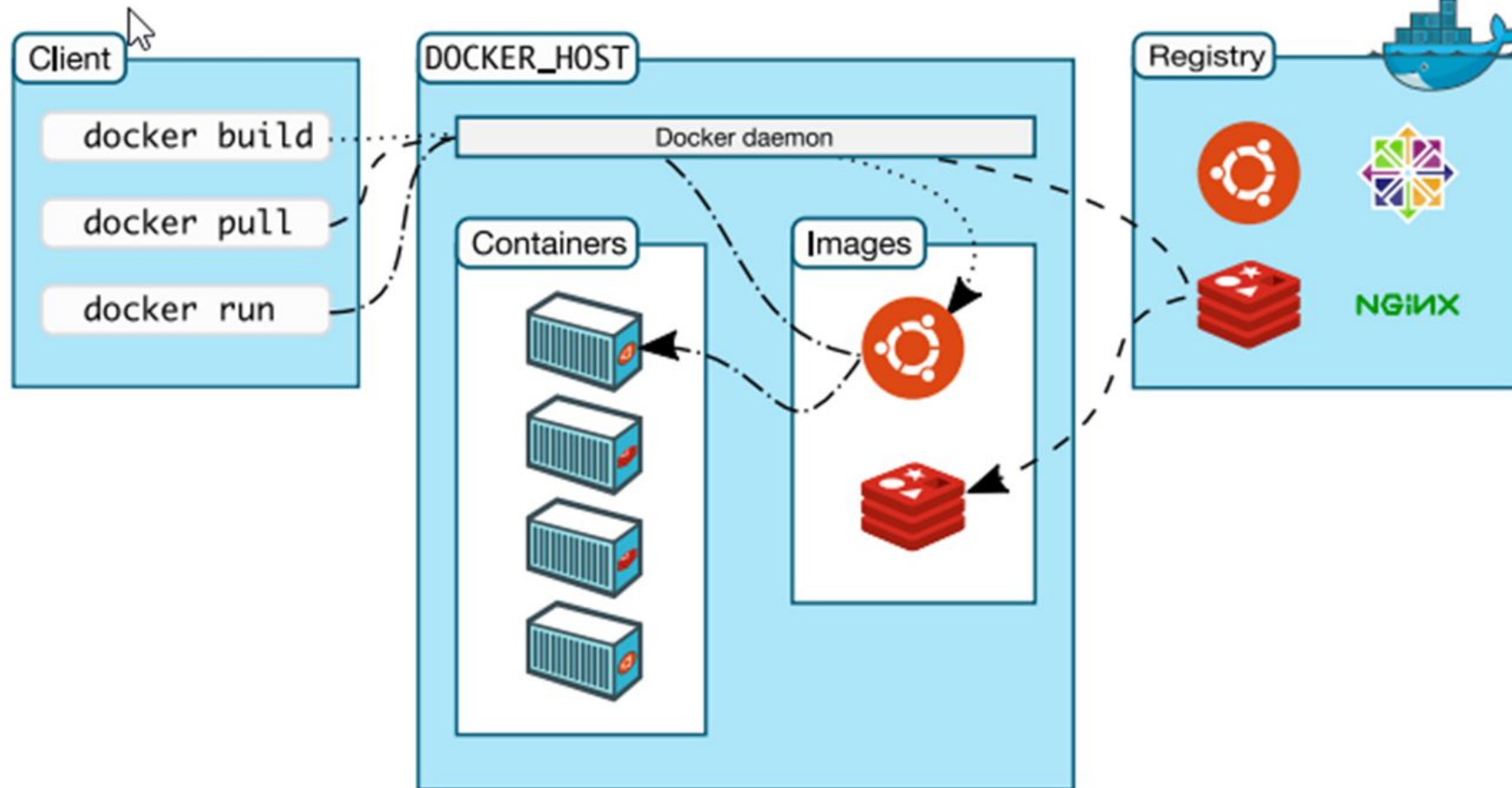
## En relación a las Máquinas Virtuales:

- Los contenedores son más ligeros que las máquinas virtuales porque comparten el kernel del host.
- Con el mismo hardware, es posible tener un mayor número de contenedores que de máquinas virtuales.
- Los contenedores se pueden ejecutar en hosts que sean máquinas virtuales.

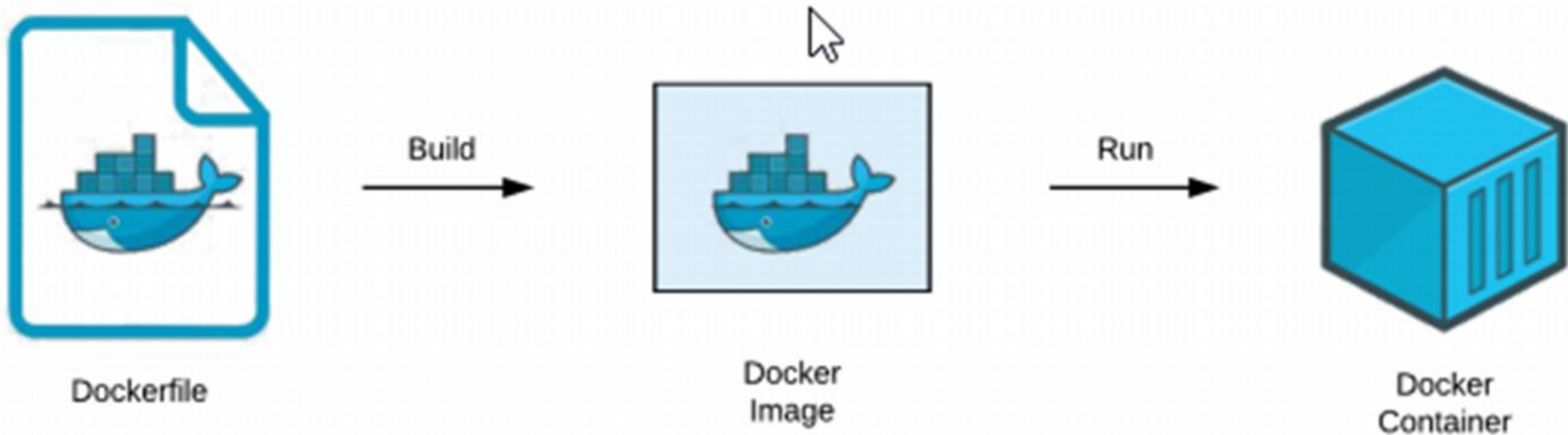
## En relación a tareas de desarrollo:

- Soluciona el problema "It works on my machine".
- Permite tener un entorno de desarrollo limpio, seguro y portátil.
- Permite la automatización de pruebas, integración y empaquetado.
- Permite empaquetar una aplicación con todas las dependencias que necesita (código fuente, librerías, configuración, etc.) para ser ejecutada en cualquier plataforma.
- Se eliminan inconsistencias entre los entornos de desarrollo, pruebas y producción.
- El proceso de despliegue es rápido y repetible.

# Componentes Docker

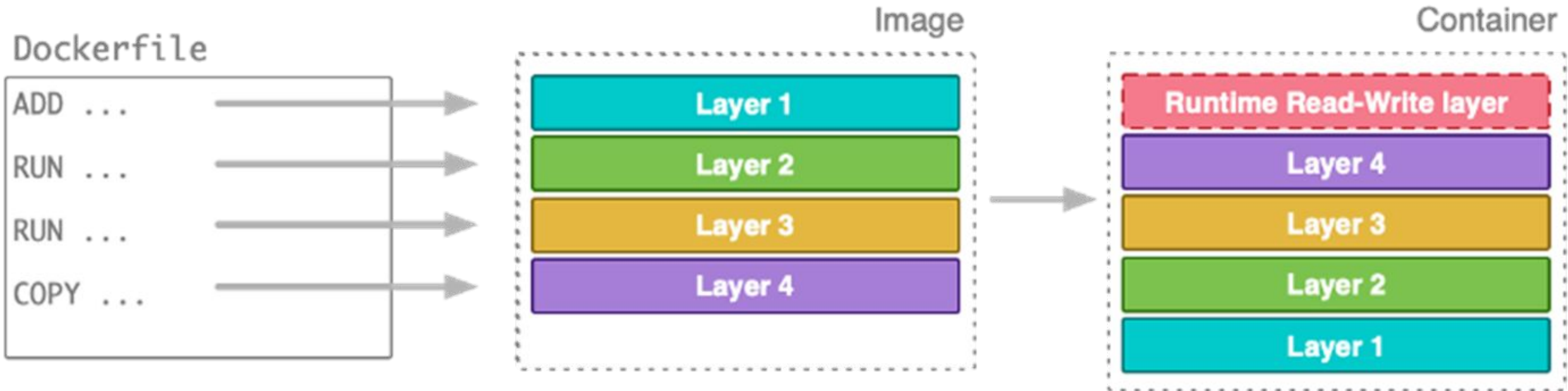


# Dockerfile & Imagen & Contenedor

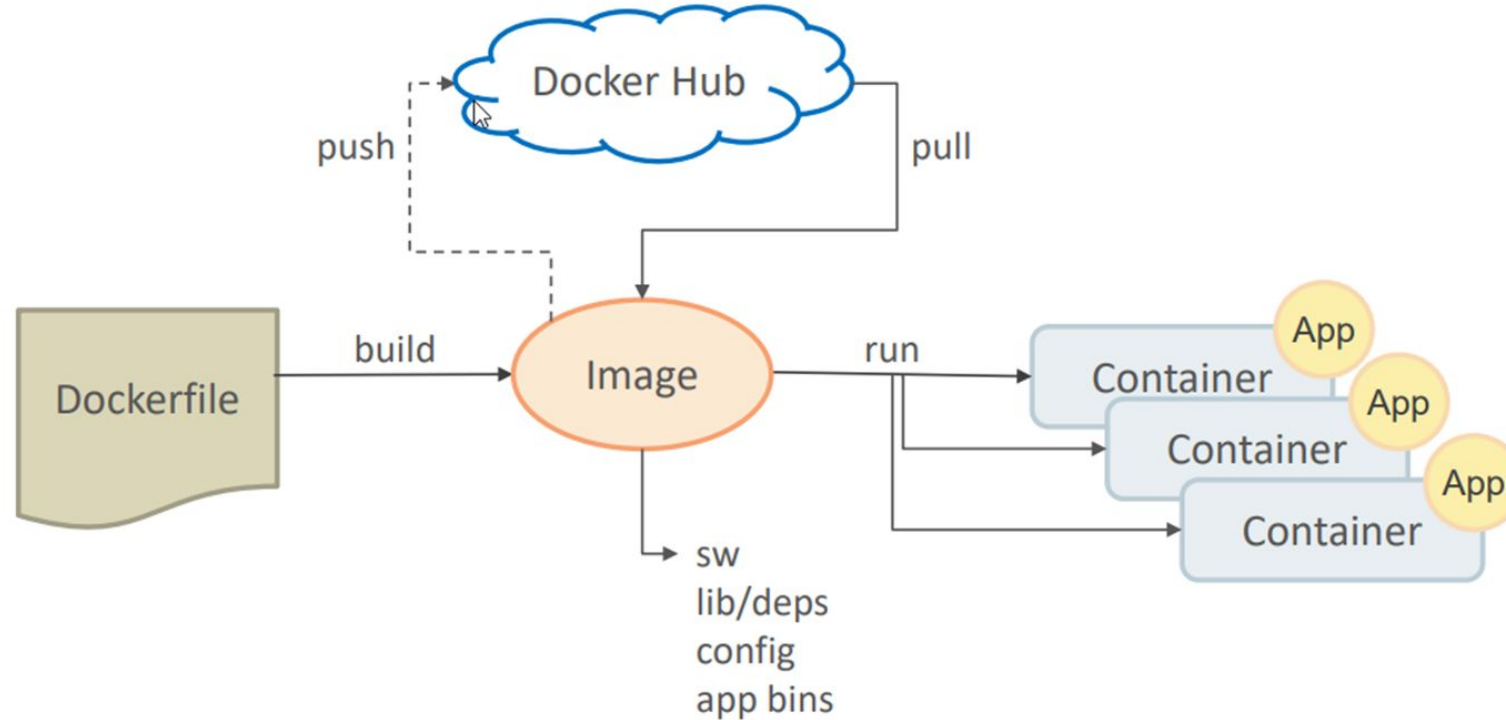




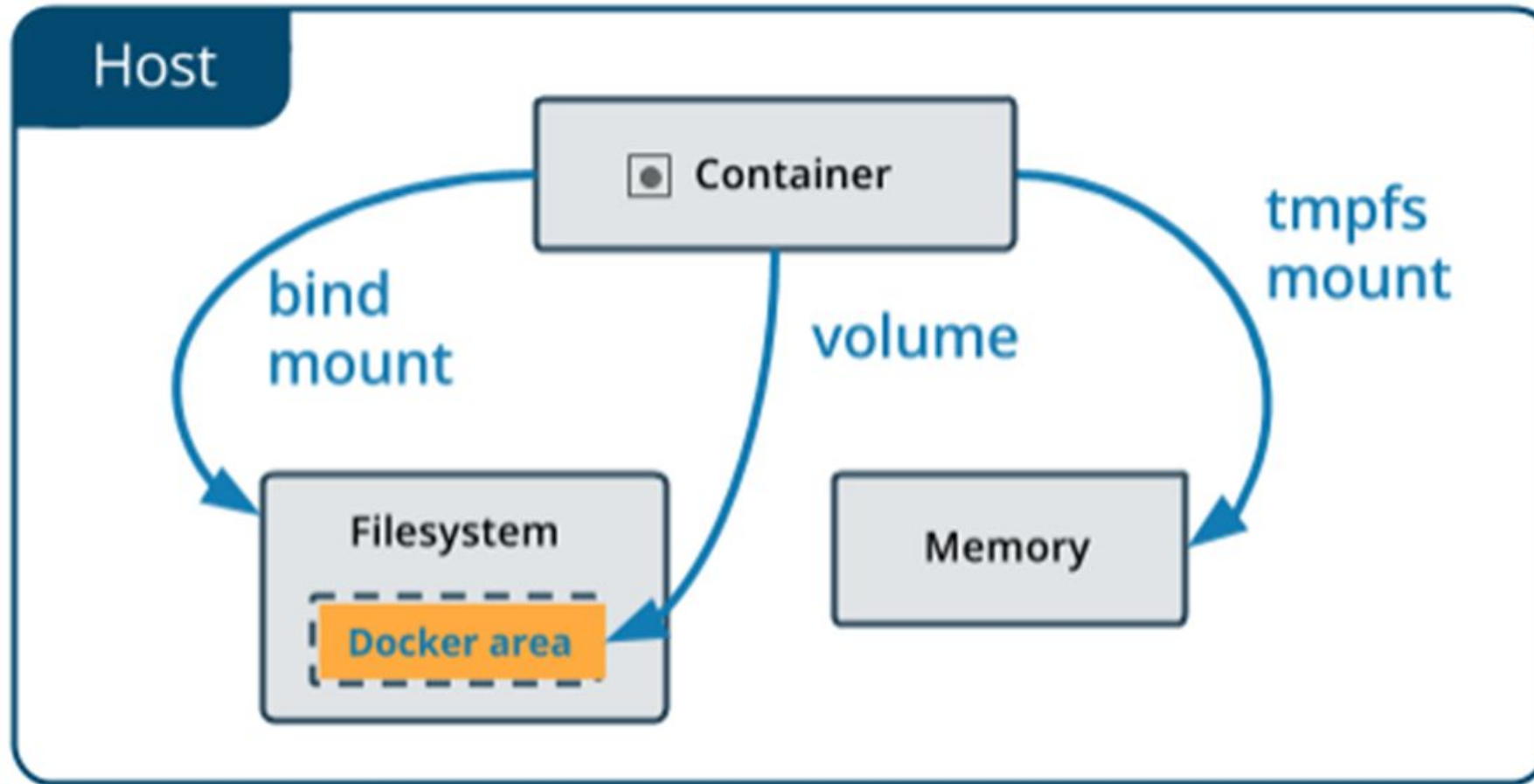
# Imágenes Docker - Capas



# ¿Cómo funciona?



# Almacenamiento en Docker

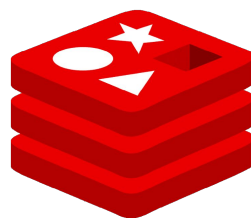


# CONTENEDORES MÁS POPULARES

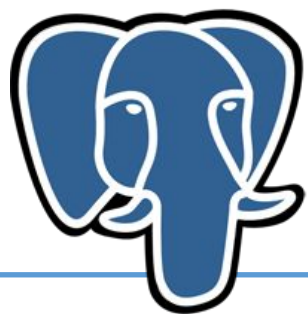
---



**Jenkins**



**redis**

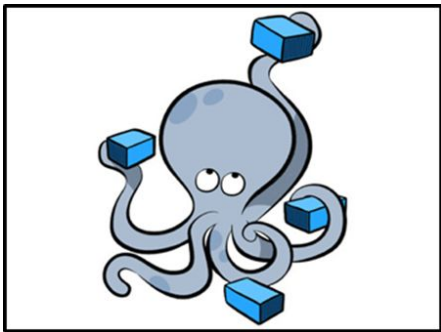


**PostgreSQL**

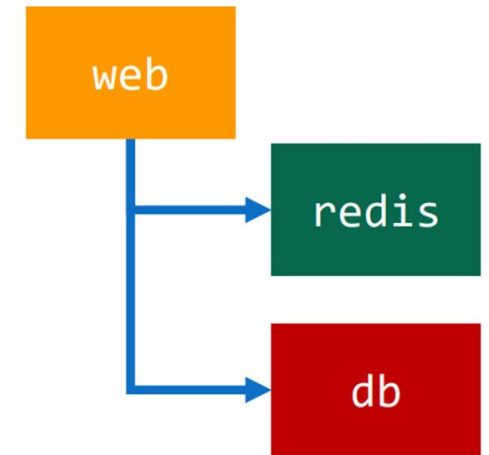


# Docker Compose - docker-compose.yml

- Docker Compose es una herramienta para definir y ejecutar aplicaciones **multi-contenedor** con Docker.
- Utiliza un **archivo YAML** para definir y configurar los servicios, los volúmenes y las redes que utilizará nuestra aplicación.
- El nombre del archivo que se utiliza por defecto es **docker-compose.yml**.



```
version: '2'
services:
  web:
    build: .
    depends_on:
      - db
      - redis
  redis:
    image: redis
  db:
    image: postgres
```



# Bueno, Vamo a Codea!!!

---

- Iniciar API Productos usando DockerCompose





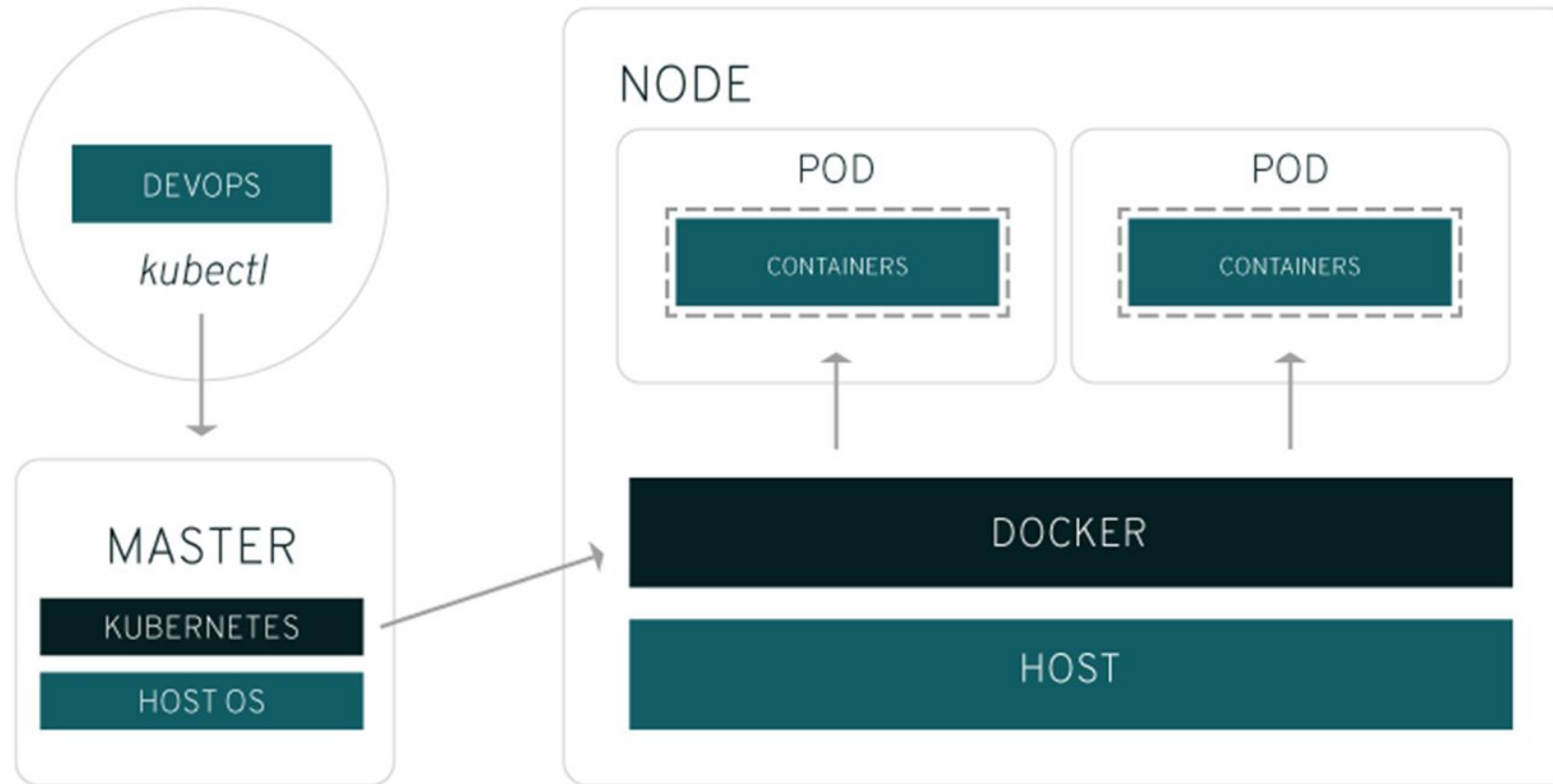
# Kubernetes: Orquestando Contenedores

---

- Es una plataforma de automatización de containers, que permite simplificar la gestión de clústeres de grupos de hosts que ejecutan containers.
- Agrupamiento lógico (aplicación)
- Clustering de servidores
- Deploy automático
- Balanceo de carga
- Escalamiento automático
- Self-healing (Recupero ante fallos)
- Rollouts y rollback (Despliegues / Vuelta a atrás)
- Run Anywhere (Funciona en cualquier lugar)



# Términos en Kubernetes



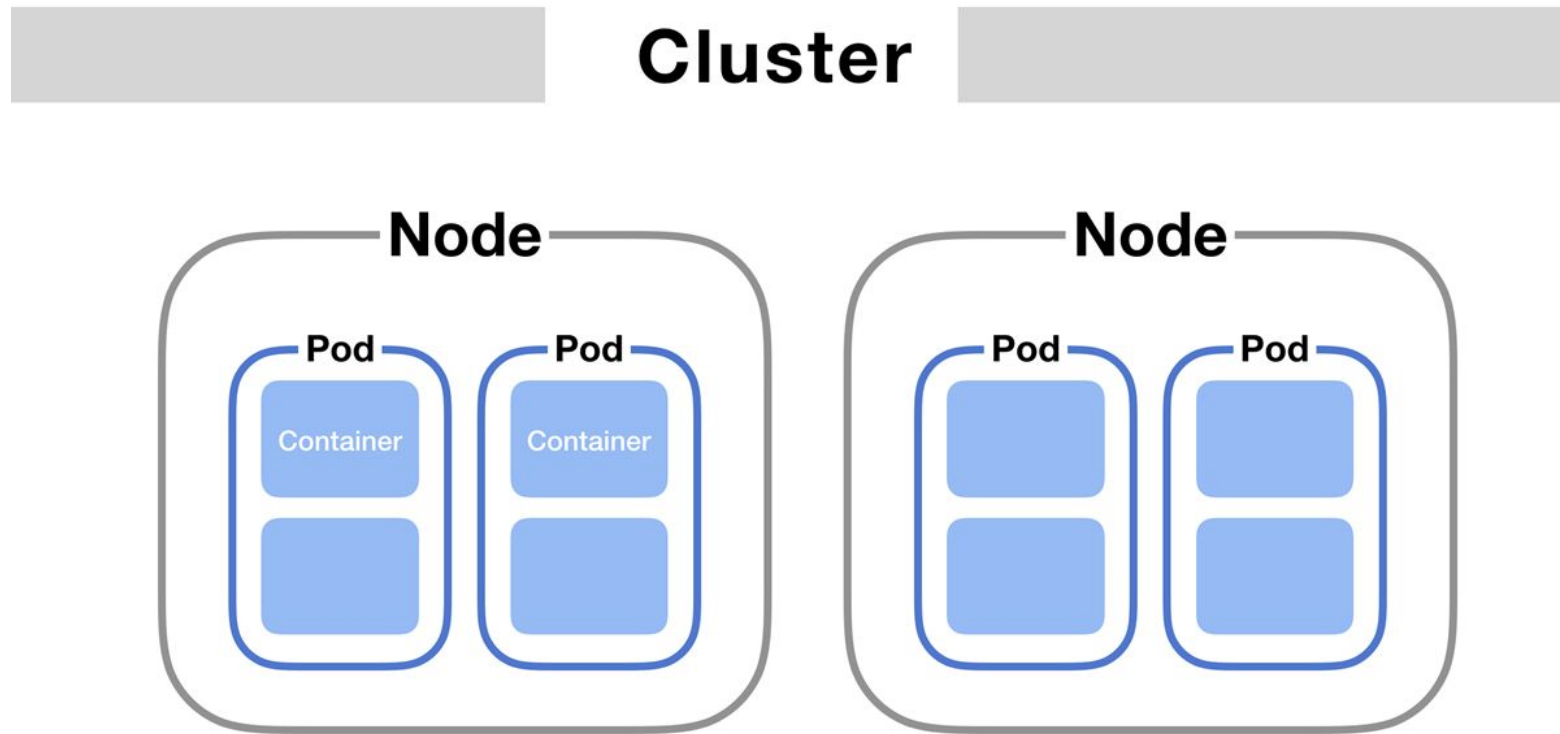
# Términos en Kubernetes

---

- **Master:** el Master es la máquina que gestiona los nodos de Kubernetes, asignando tareas a cada nodo.
- **Nodo:** un Nodo es una máquina que realiza las tareas que le indica el Master.
- **Pod:** un grupo de uno o más contenedores implementados en un nodo único. Todos los contenedores de un pod comparten la dirección IP, la IPC Socket, el nombre del host y otros recursos. Los pods abstraen la red y el almacenamiento del contenedor subyacente. Esto le permite mover los contenedores por el clúster con mayor facilidad.
- **Controlador de replicación:** controla cuántas copias idénticas de un pod deben estar ejecutándose en algún lugar del clúster.
- **Servicio:** separa las definiciones de tareas de los pods. Los proxies de servicios de Kubernetes envían automáticamente las solicitudes de servicio al pod correspondiente, sin importar adónde se traslada en el clúster, o incluso si está siendo reemplazado.
- **Kubelet:** es un servicio que se ejecuta en cada nodo y revisa si los containers están iniciados y ejecutándose.
- **kubectl:** herramienta para la configuración de la línea de comandos de Kubernetes.

# Cluster Kubernetes

---



# Despliegue de la API: Opciones en AWS



- **Elastic Beanstalk:** Ideal para proyectos sin mucha experiencia en administración de infraestructura y que requieran una plataforma gestionada.
- **Amazon ECS (Elastic Container Service):** Recomendado para aplicaciones que necesiten alta flexibilidad y escalabilidad con contenedores Docker.
- **Lightsail:** Perfecto para proyectos más pequeños que necesiten una solución rápida y sencilla con recursos preconfigurados.
- **Lambda (Serverless):** Adecuado para funciones pequeñas y eventos específicos, con una escalabilidad automática y un modelo de pago por uso.

# CloudFormation e Infraestructura como Código (IaC)

**Infraestructura como Código (IaC)** es una práctica que consiste en describir y administrar la infraestructura mediante código en lugar de realizar configuraciones manuales.

**CloudFormation** es un servicio de AWS que permite crear y gestionar recursos de manera automatizada en la nube.

Ventajas de CloudFormation y IaC:

- **Automatización:** Permite automatizar la creación y configuración de recursos en AWS, ahorrando tiempo y evitando errores manuales.
- **Consistencia:** Garantiza que la infraestructura se implemente de manera coherente y replicable en diferentes entornos.
- **Control de Versiones:** El código de infraestructura se puede gestionar con herramientas de control de versiones, lo que facilita el seguimiento de cambios y la colaboración en equipos.
- **Escalabilidad:** Admite el escalado automático de recursos en función de la demanda de la aplicación.
- **Seguridad:** Proporciona una forma segura y auditable para administrar recursos en la nube.





# Bueno, Vamo a Desplega!!!

---

Despliegue con Amazon ECS (Elastic Container Service)

1. Preparamos el entorno.
2. Generamos Imagen Docker
3. Ejecutamos el Despliegue
4. Monitoreamos que todo funcione bien.



# Actividad 8: Paso a Paso Despliegue API

- Seguir las instrucciones de la actividad publicada en la UVE.



# MUCHAS GRACIAS

---

**ANDÉN**  
Centro de Innovación  
y Emprendimientos Tecnológicos

SECRETARÍA DE  
EXTENSIÓN  
UNIVERSITARIA  
UTN - FRC

**SEU**

**UTN**  
Facultad Regional Córdoba

Agencia  
**CÓRDOBA  
JOVEN**

 **CÓRDOBA**  
entre todos