

# Kubernetes

# Introducción a Kubernetes.

“Kubernetes es una plataforma portable y extensible de código abierto para administrar cargas de trabajo y servicios. Kubernetes facilita la automatización y la configuración declarativa”

- Orientada a contenedores
- Microservicios
- Híbrida / Nube

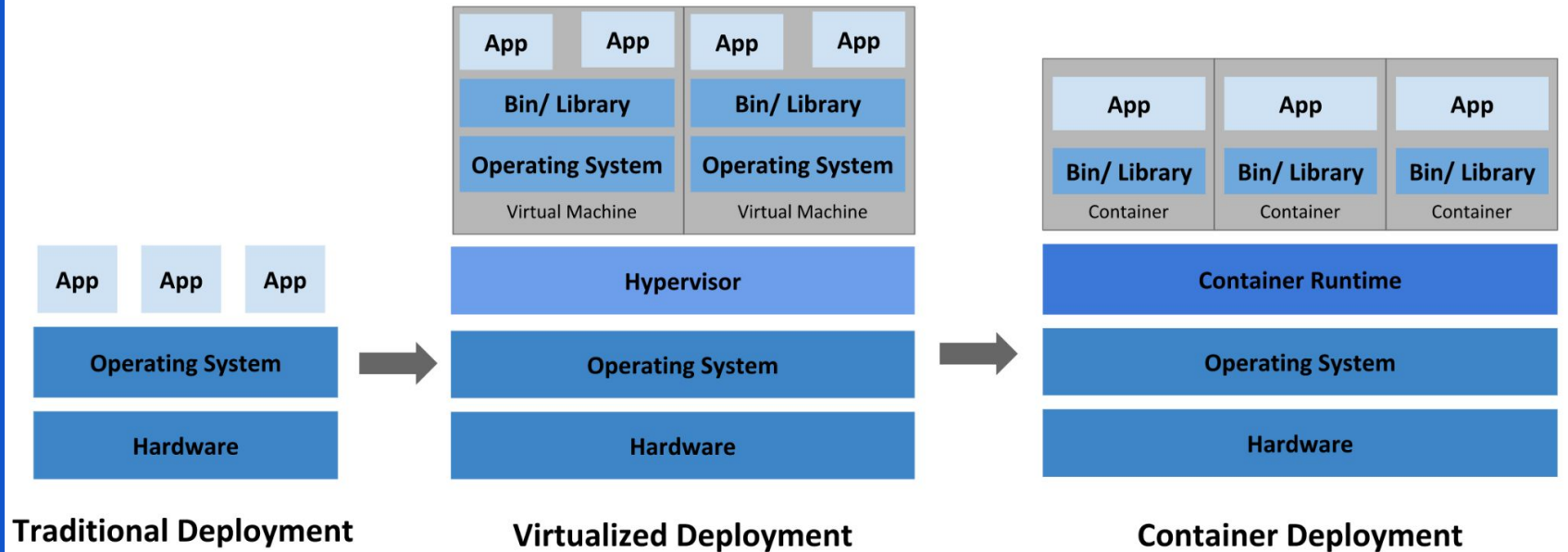
# Introducción a Kubernetes.

También llamado “K8s”

Historia:

- 2014 Kubernetes pasa a ser OpenSource (Cloud Native Computing Foundation). Google usa billones de contenedores al día.
- Versión 1: 1 Julio 2015.
- Escrito en Go
- Viene de los proyectos Borg y Omega (propietarios)
- K8s no es una versión libre de estos proyectos, pero está relacionado con ellos.

# Introducción a Kubernetes.



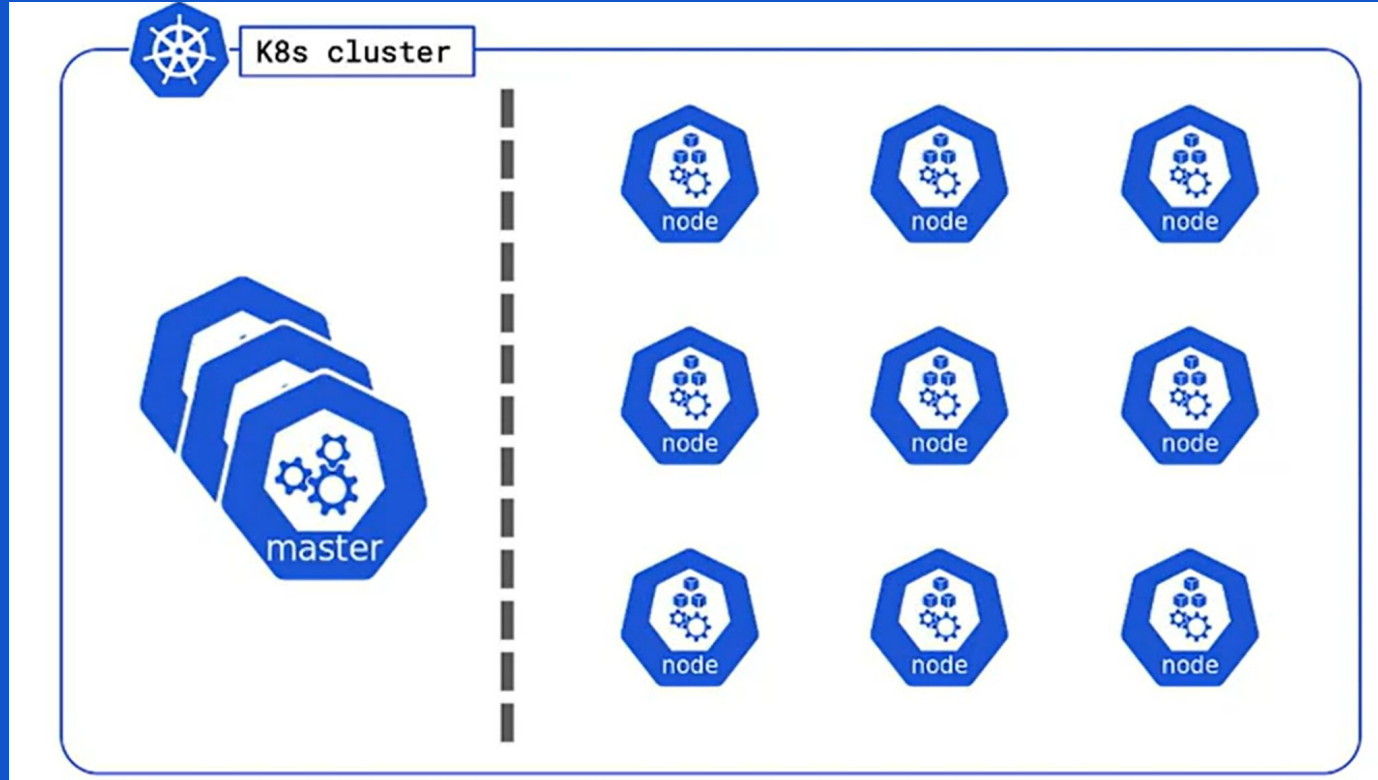
# Introducción a Kubernetes. Arquitectura

Kubernetes es una plataforma de orquestación

Controla todos los componentes para ejecutar aplicaciones en contenedores

- Flujo de Trabajos
- Secretos (K/V)
- Balance de Carga
- Panel de Control
- API
- Automatización
- Almacenamiento

# Conceptos Básicos de Kubernetes. Arquitectura



# Introducción a Kubernetes. Master

Master Node o Panel de Control:

Detecta cambios en el cluster (nuevo despliegue, balance de carga, nuevo POD, nuevo nodo, replicas, etc)

Kube-apiserver: APi para interactuar

etcd: Almacenamiento persistente y distribuido.

kube-scheduler: Despliega “pods”

kube-controller: Ejecuta los controladores (independientes)

cloud-controller-manager: Interactúa con proveedores de la nube

# Introducción a Kubernetes. Master



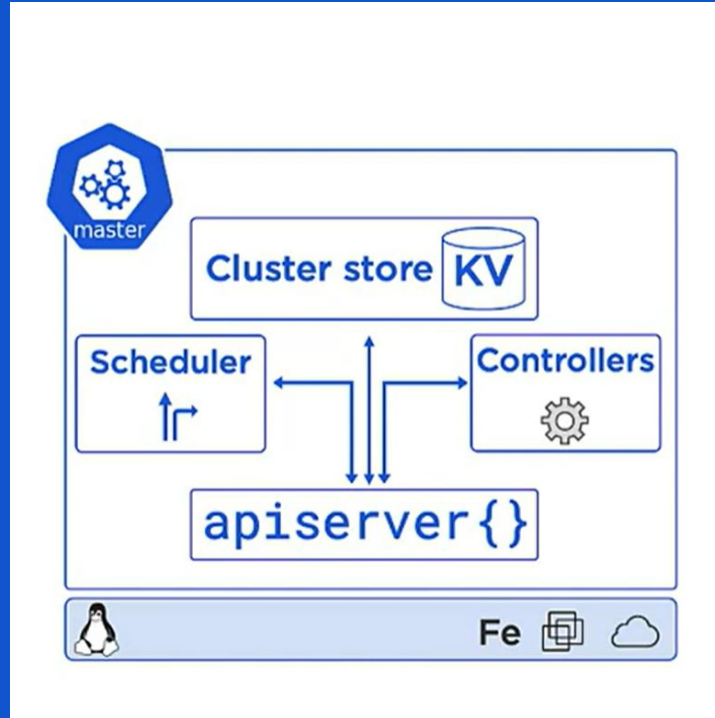
Lider



Seguidor



Seguidor





# Introducción a Kubernetes. Nodos

Mantienen los pods y proporcionan la ejecución de Kubernetes

Kubelet: Agente de Kubernetes, registra los nodos

kube-proxy: Networks, asigna IP, balance de carga

“Runtime”: docker, containerd, cri-o o cualquier implementación.

Otros componentes: Deployments, Controllers, ReplicaSets, DNS o dashboards.

# Kubernetes. Objetos

PODS: 1 o más container

Deployment: Escalar, “rolling updates”

Secrets

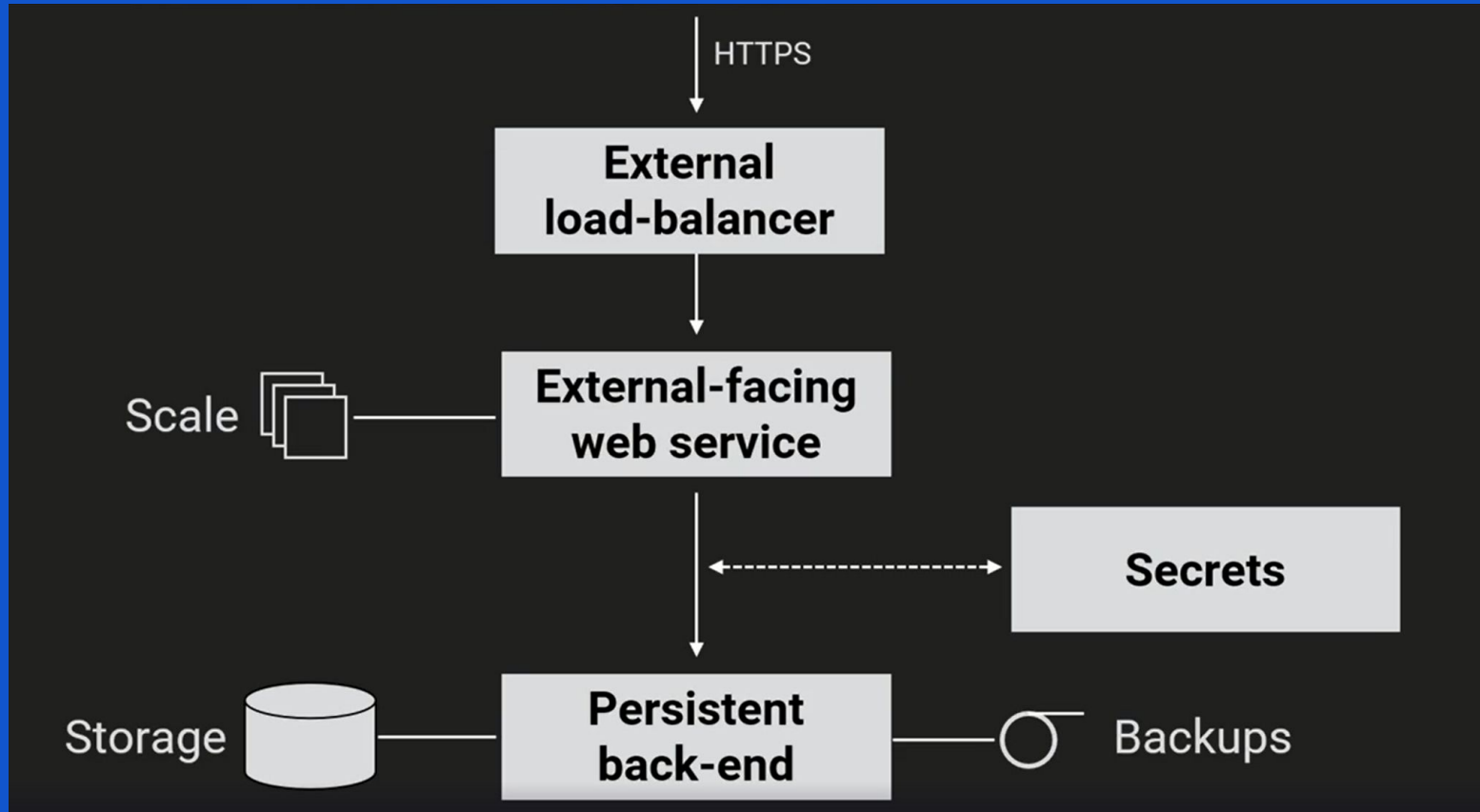
Network

Services

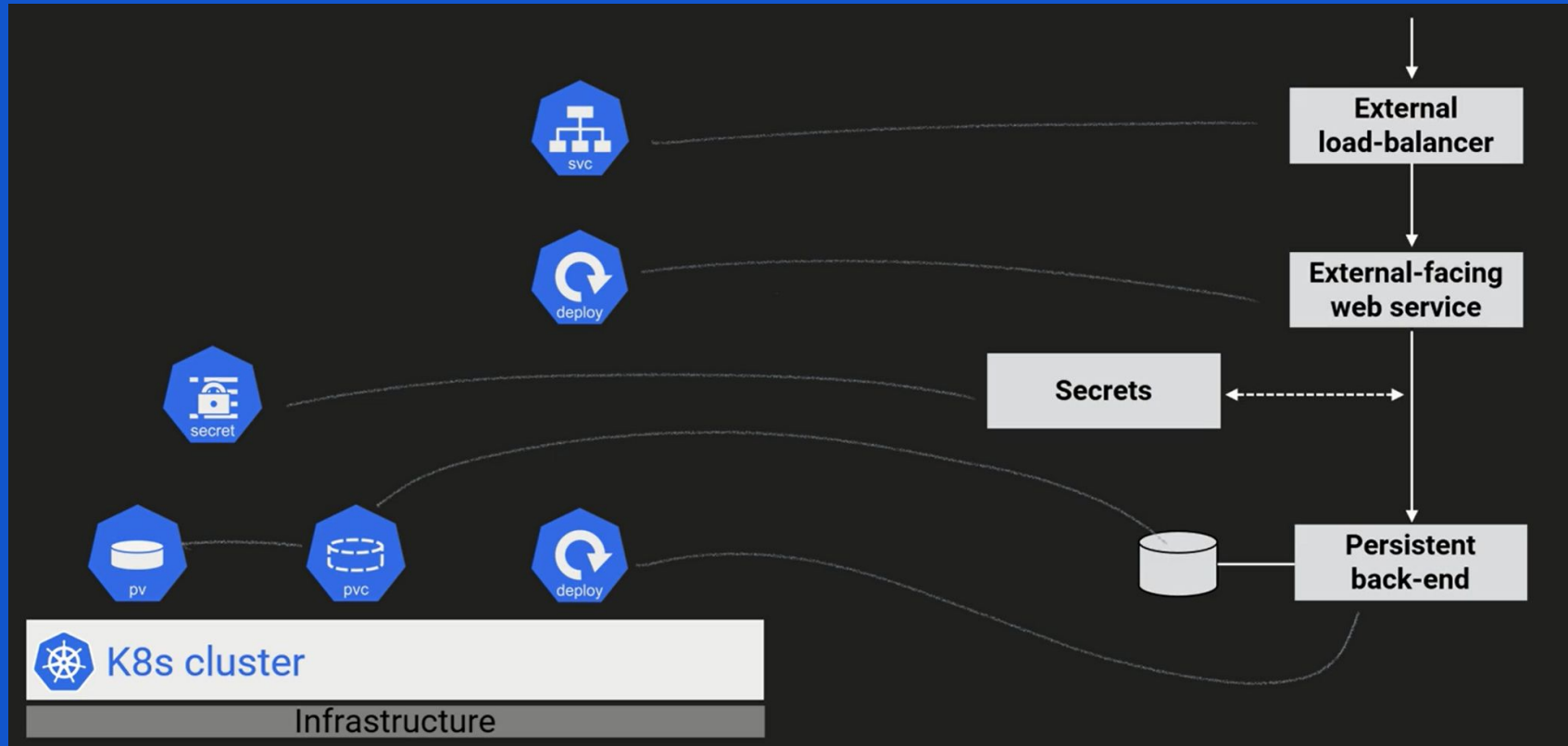
Almacenamiento

Load Balance

# Kubernetes. Aplicaciones



# Kubernetes. Aplicaciones

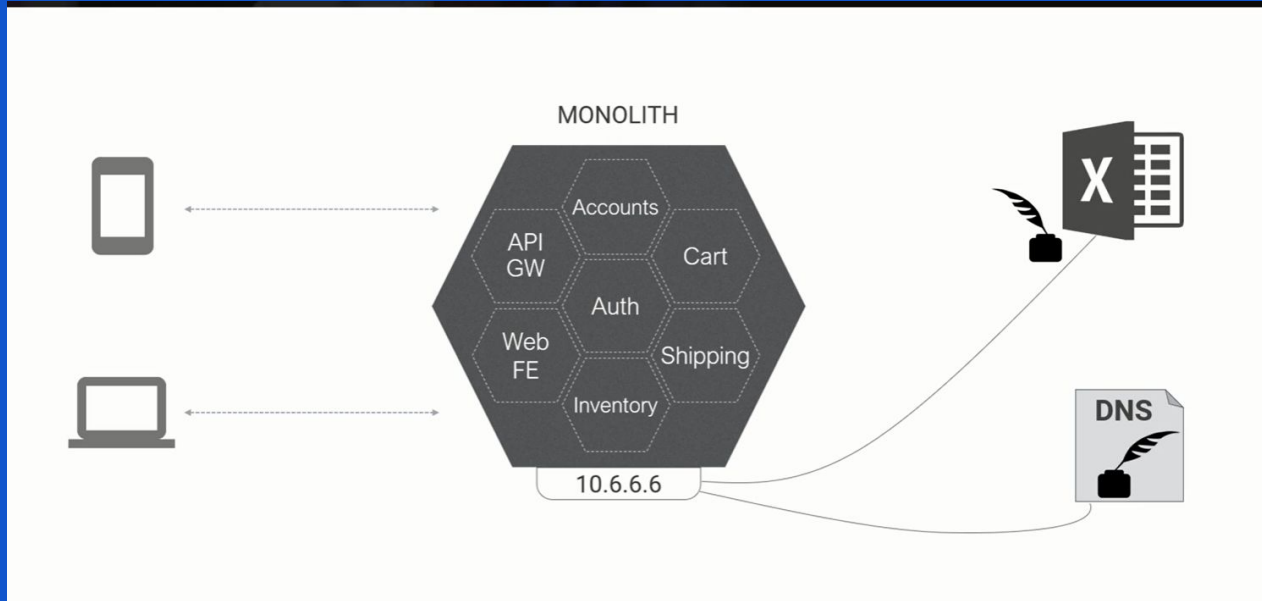


# Kubernetes. Namespaces

- Se puede entender como un cluster virtual que agrupa objetos de kubernetes.
- Ejemplo: Cuando hay muchos usuarios o equipos dentro de un mismo cluster.
- NOmbre único
- "resource quote"
- `kubectl .... --namespace`

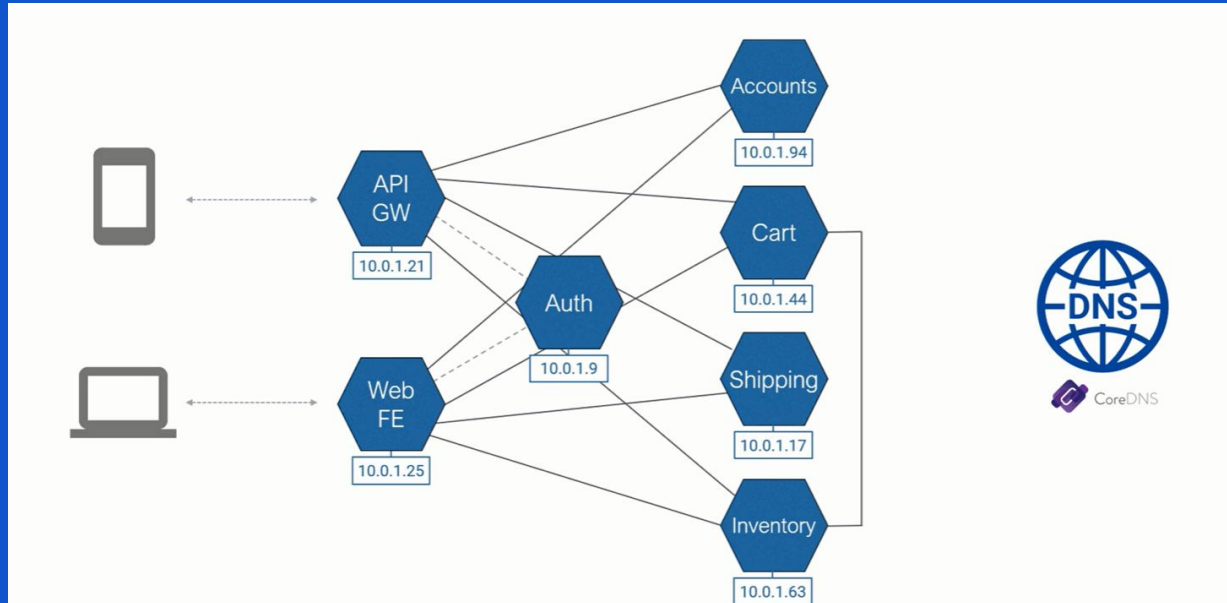
# Kubernetes. Redes

En aplicaciones “monolíticas”, normalmente no hay que preocuparse por networking

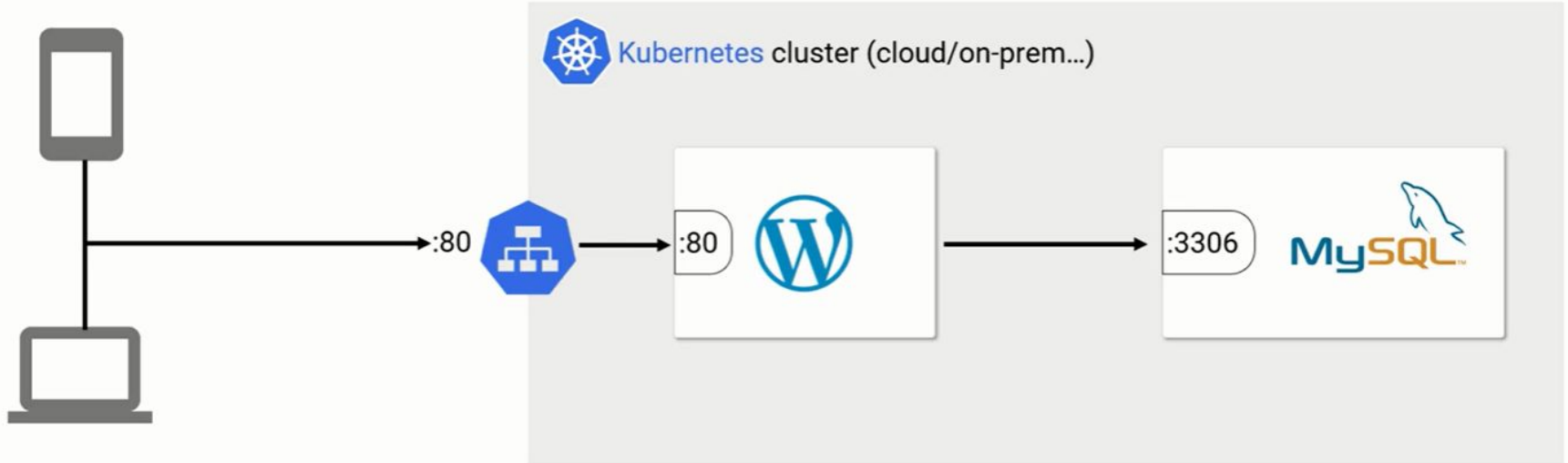


# Kubernetes. Redes

Todo cambia con contenedores. “Dynamic Network / Service Discovery”  
network es software en K8s: Cisco, Cilium, Contrail, Flannel, Calico, Etc.



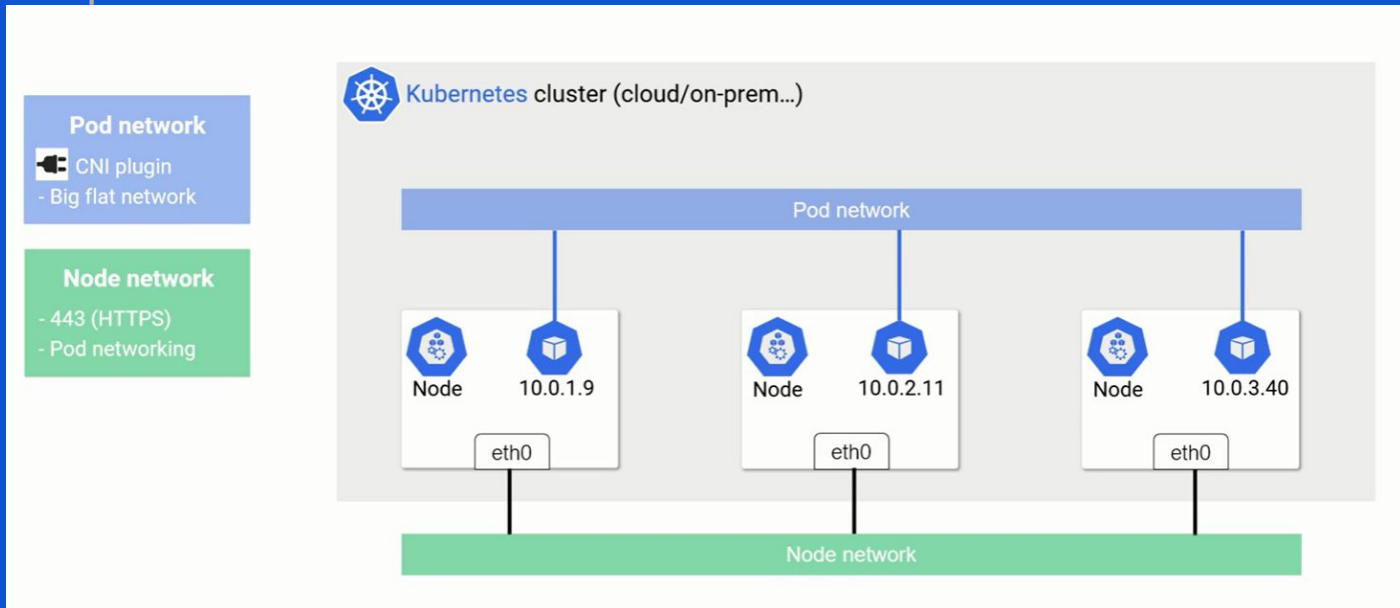
# Kubernetes. Ejemplo





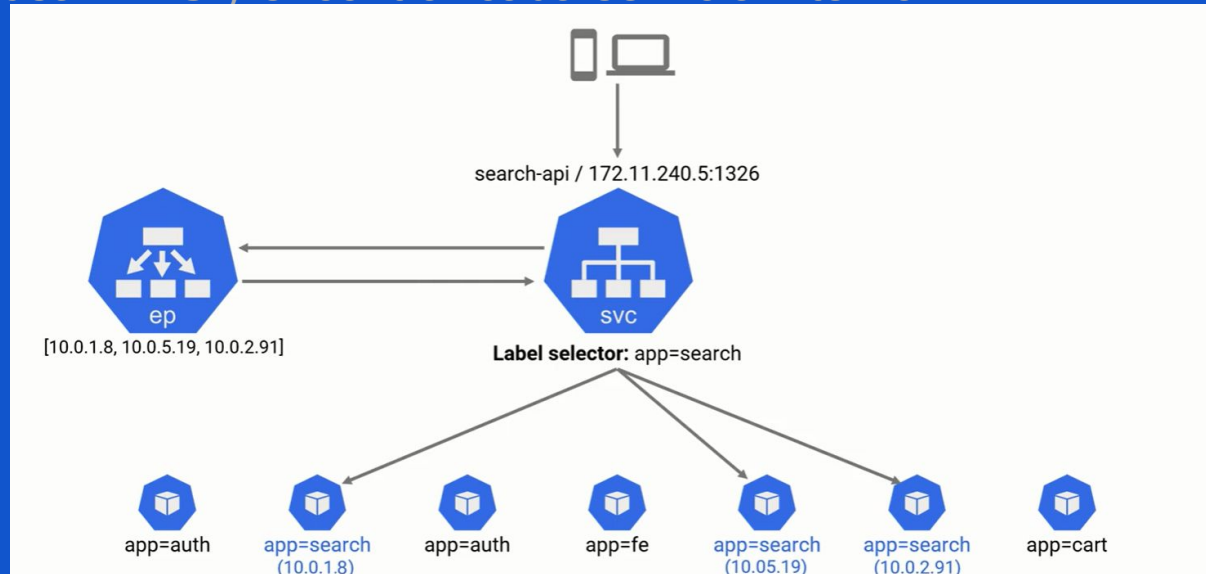
# Kubernetes. Fundamentos de Redes

- Flat network
- Todos los pods se pueden comunicar
- Todos los pods tienen IP



# Kubernetes. Servicios

- Pods son efímeros, como recordar IP?
- Usar “Servicios” y no Pods directamente. Ejemplo: search-api, database-local
- Servicio / IP / Name, registrado en DNS.
- Cada Pod puede usar DNS y encontrar cada servicio Interno



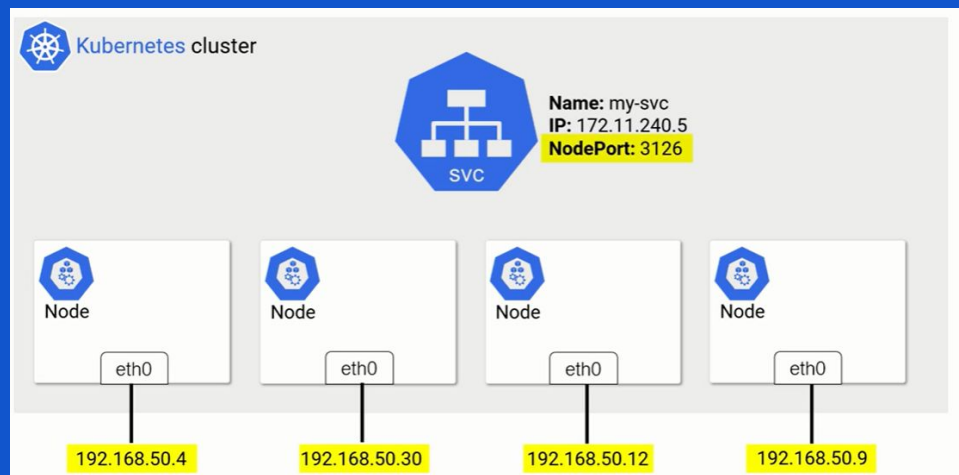
# Kubernetes. Servicios. ClusterIP y NodePort

## ClusterIP

- Tiene su propio IP
- Solo se accede “desde el cluster”

## NodePort

- Su propio puerto
- Se puede acceder fuera del Cluster, con la IP del NodePort



# Kubernetes. Servicios. LoadBalancer

## Balaneo de Carga

- Por Defecto Kubernetes configura NodePort con el LoadBalancer (cloud)
- External Load Balancer / K8s LoadBalancer objeto

## ServiceNetwork

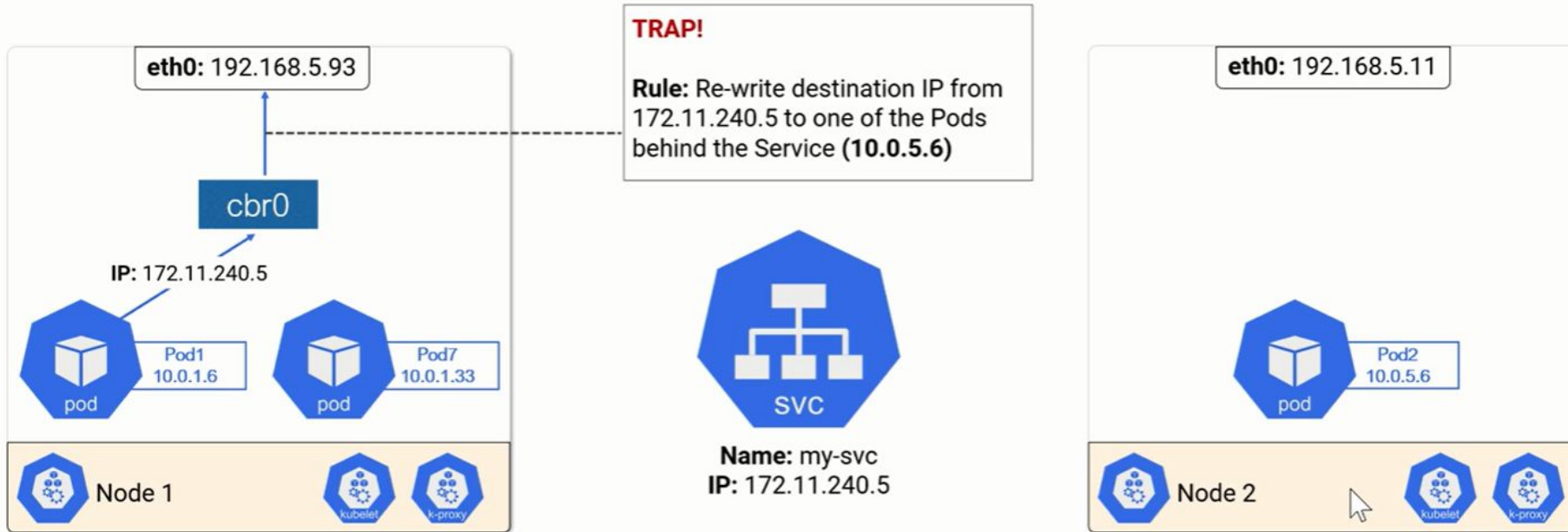
- Service siempre tienen una nueva IP / Nueva Network. Se crea en “ServiceNetwork”. No es una network física (no interface)
- KubeProxy: IPTABLES rules en cada Nodo (rutas)

# Kubernetes. Service Network

## ServiceNetwork

- Service siempre tienen una nueva IP / Nueva Network. Se crea en “ServiceNetwork”. No es una network física (no interface)
- KubeProxy: IPTABLES/IPVS rules en cada Nodo (rutas)

Pod network (10.0.0.0/16)

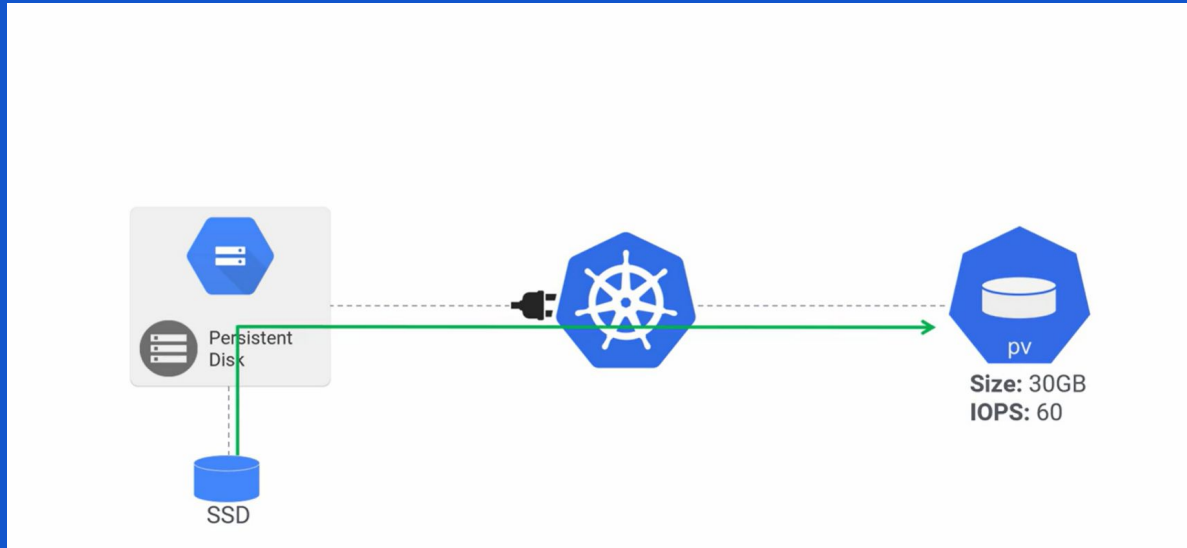


# Kubernetes. Almacenamiento

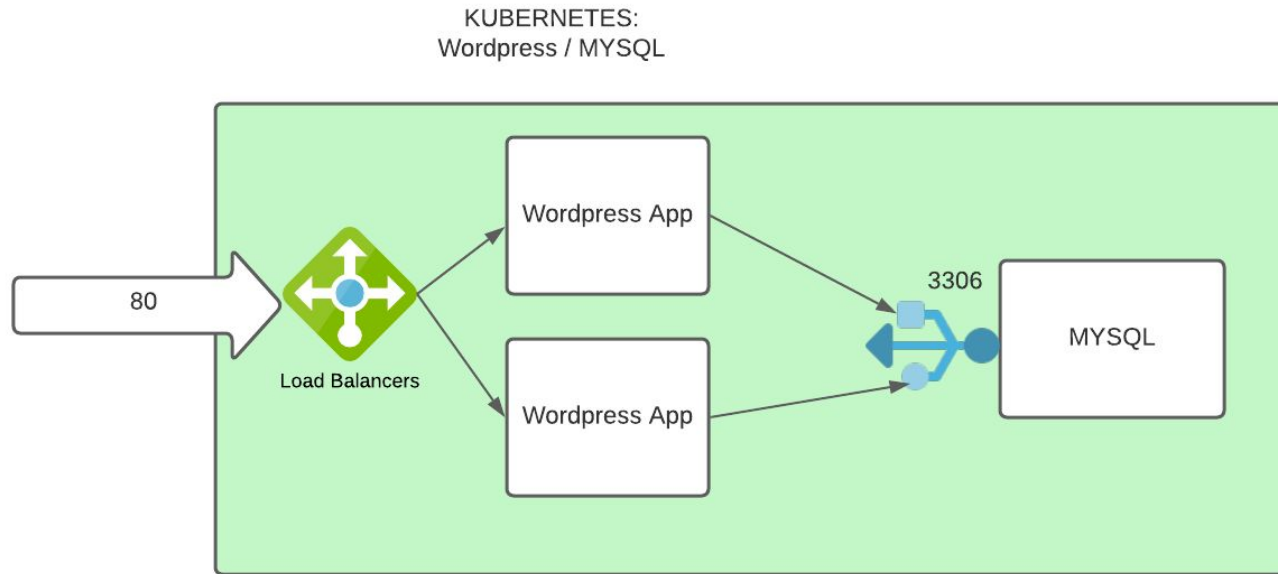
Volumen: Abstraer almacenamiento de los pods

Persistent Volume (pv): Recurso de Almacenamiento (disk, cloud disk).

Persistent Volumen Claim: Ticket para usar pv.



# Kubernetes. Ejercicio





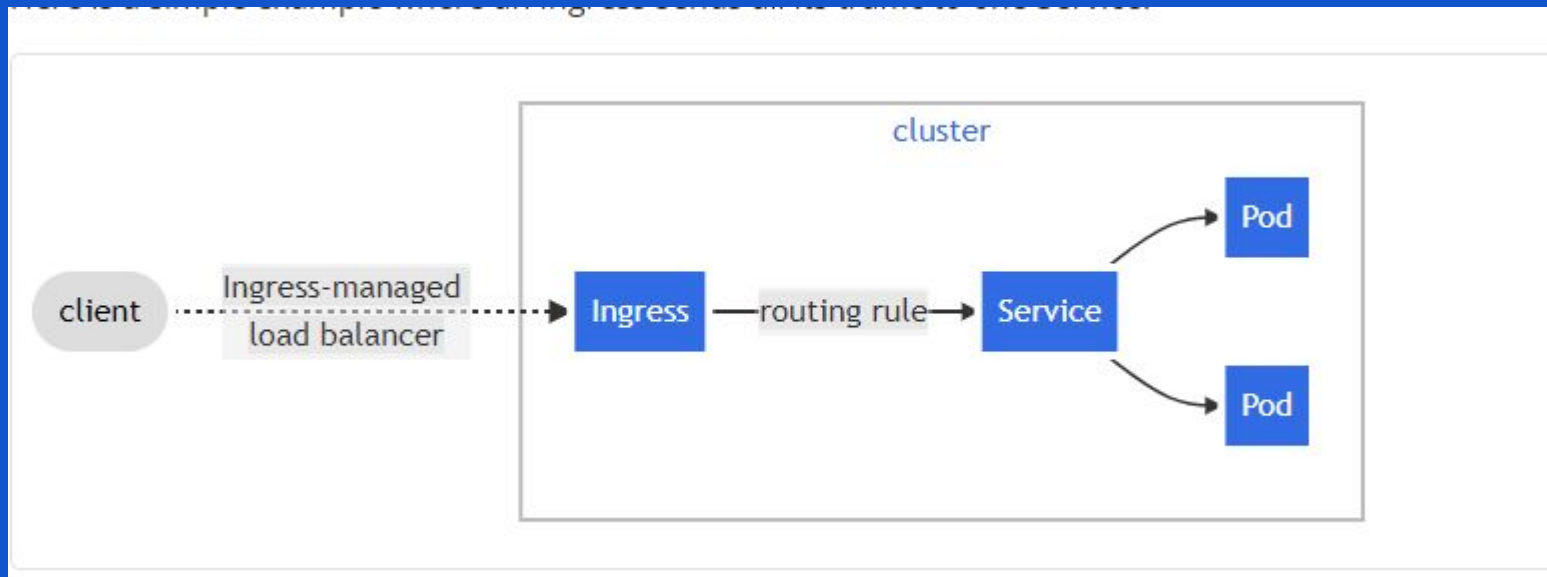
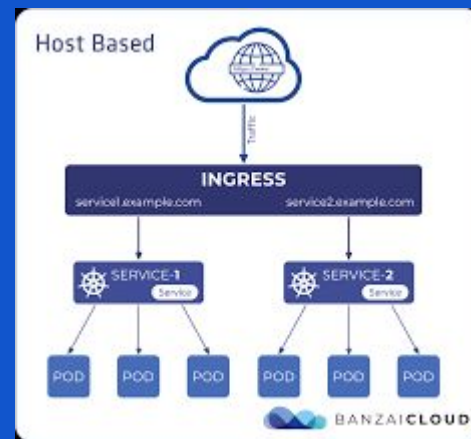
# Kubernetes. Ejercicio Solucion

<https://github.com/andrescolodrero/CursoImagina-openshift/tree/main/sesiones/Sesion4-Kubernetes/Stateful-app>

Solucion: `kubectl apply -f .`

# Kubernetes. Ingress

Ingress expone HTTP/S fuera del cluster a servicios internos.  
Dependiendo del cluster K8s, se necesita un tipo de “controllador”



# Kubernetes. Despliegues

