

DESARROLLO DE APLICACIONES UTILIZANDO PYTHON Y ORQUESTACIÓN DE CONTENEDORES DOCKER

Universidad de Costa Rica

Instructor: Jose Pablo Ramírez Méndez

Semana III

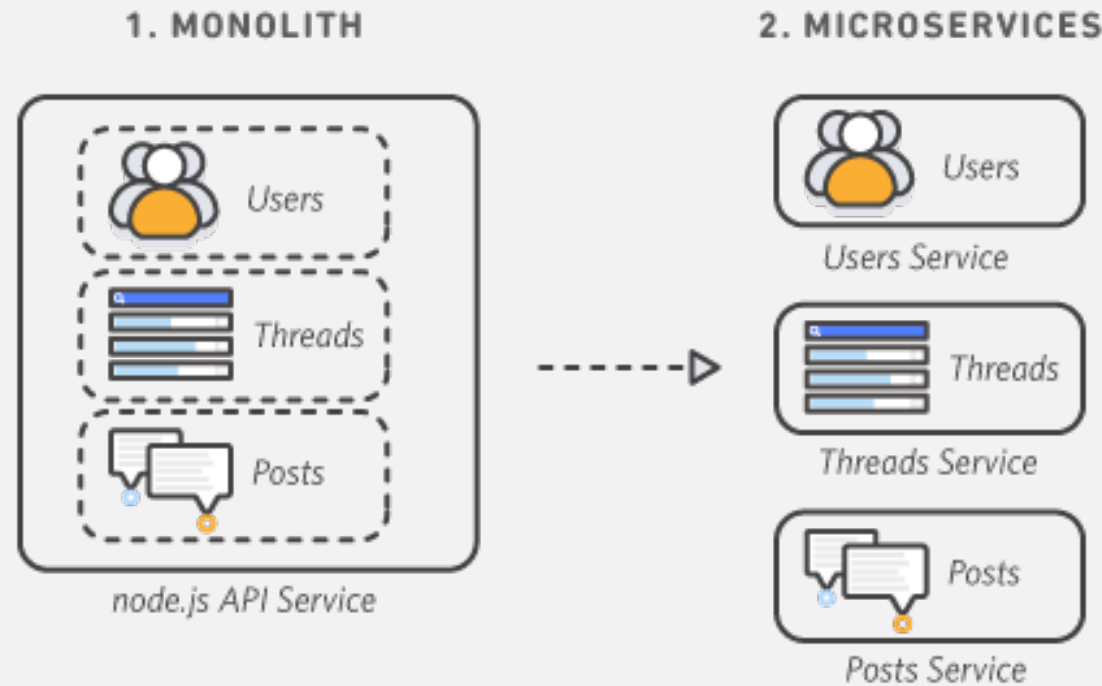
AGENDA

- Contexto de microservicios
- ¿Por qué los contenedores no son el final del camino?
- Kubernetes: Orquestación de contenedores
- Caso de estudio – Parte 2

OBJETIVOS

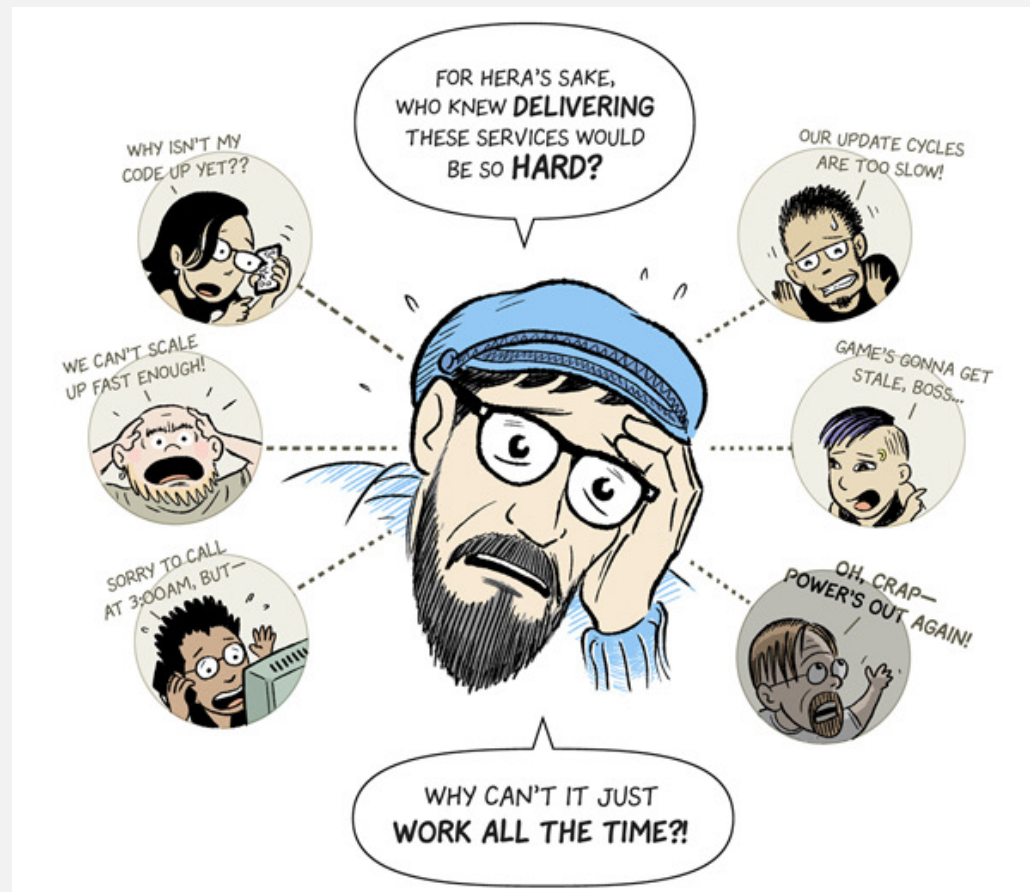
- Entender el concepto de orquestación de contenedores y su función
- Conocer la utilidad de Kubernetes y su filosofía.
- Aplicar el uso de Kubernetes en el contexto del contenedores docker.

MICROSERVICIOS VS ARQ. MONOLÍTICA

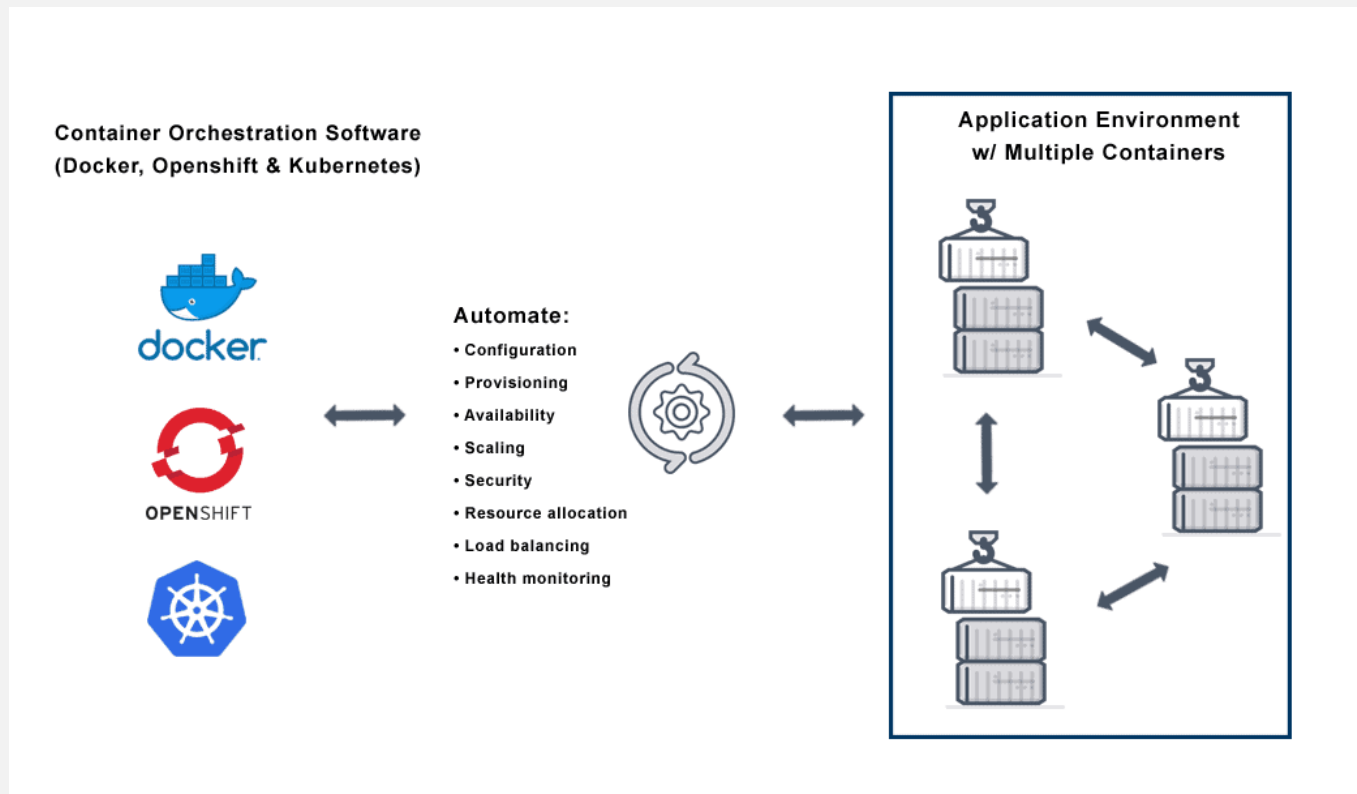


¿POR QUÉ LOS CONTENEDORES NO SON EL FINAL DEL CAMINO?

- Es difícil escalar rápidamente: Más carga => Más recursos
- Difícil de monitorear
- ¿Dónde desplegar los contenedores?
- Versionamiento de mis images.



¿QUE ES ORQUESTACIÓN DE CONTENEDORES?

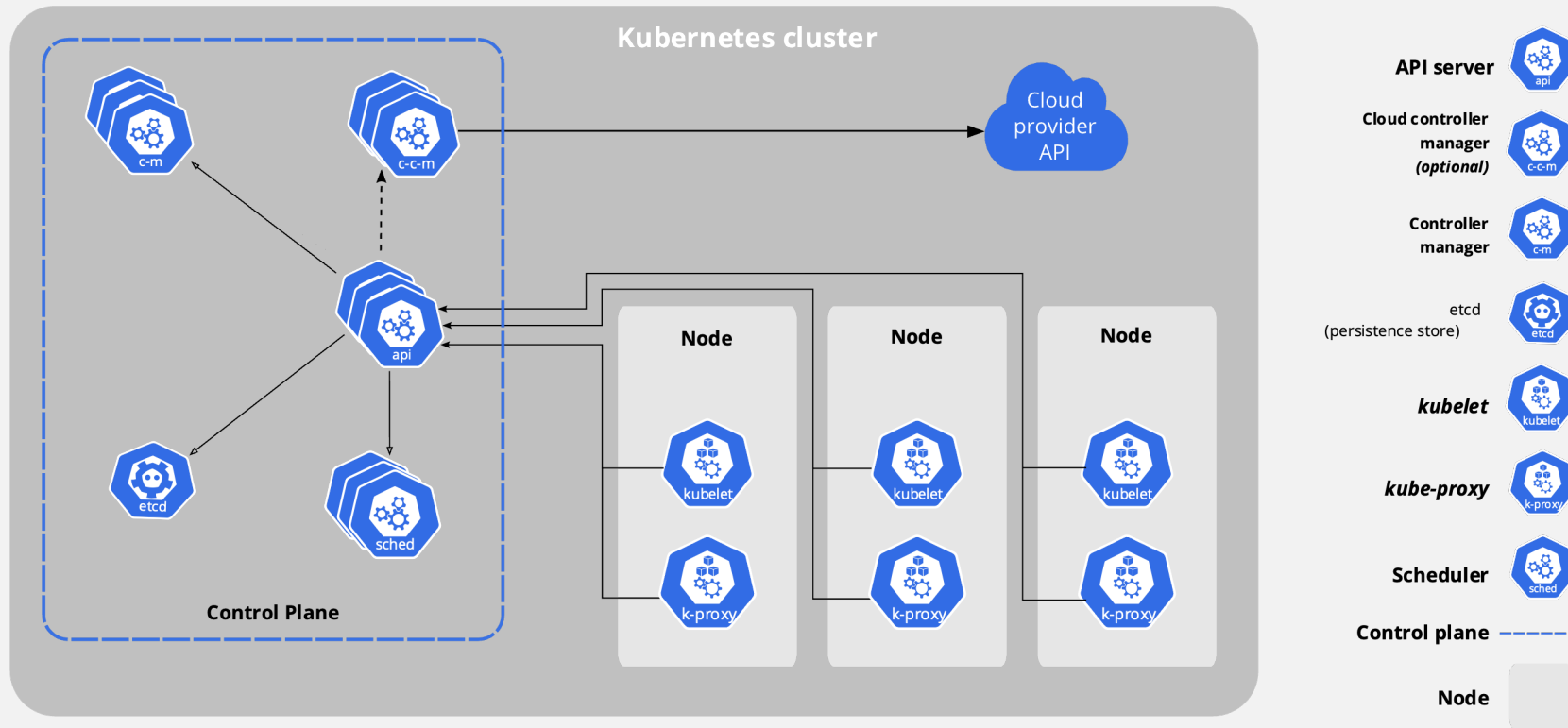


KUBERNETES

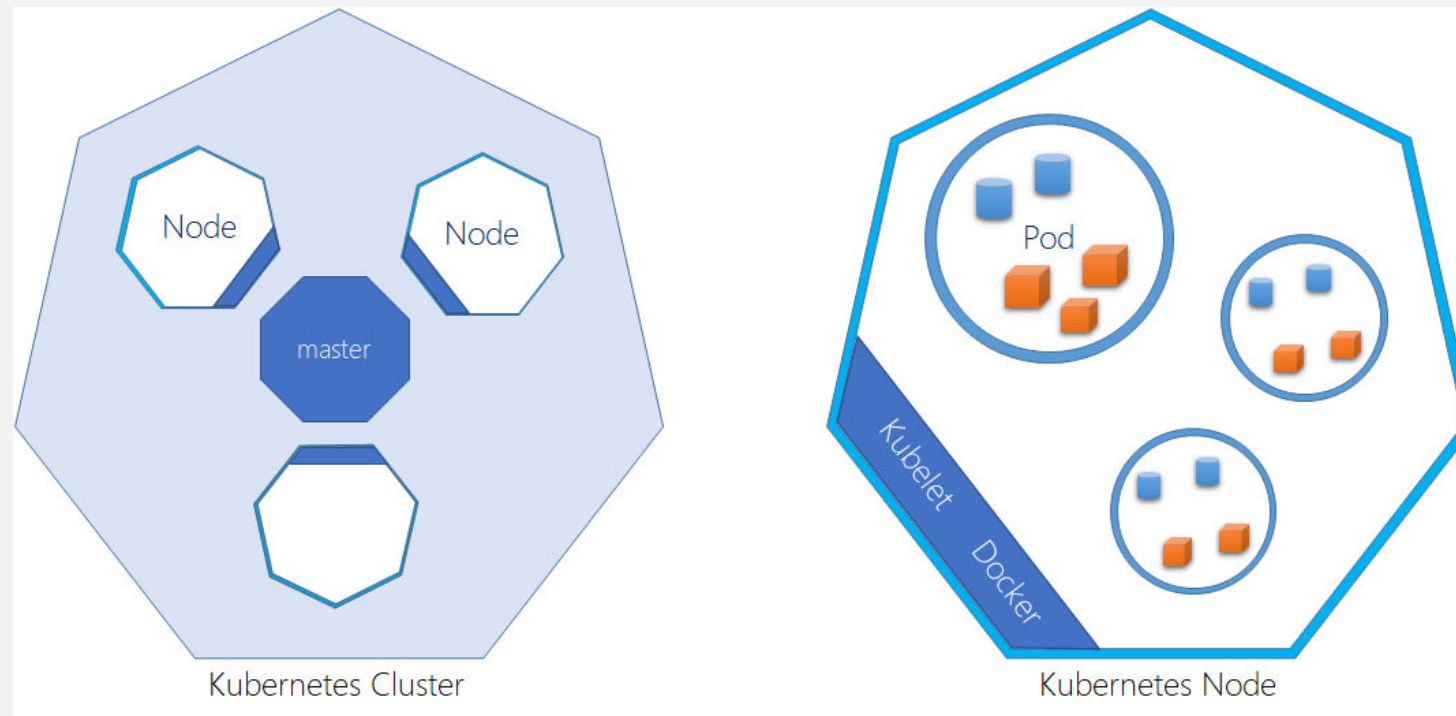
- Orquestador de contenedores
 - Configuración
 - Seguridad
 - Balanceo de carga
 - Escalabilidad
 - Redundancia
 - Versionamiento
 - Y más...



KUBERNETES: ARQUITECTURA

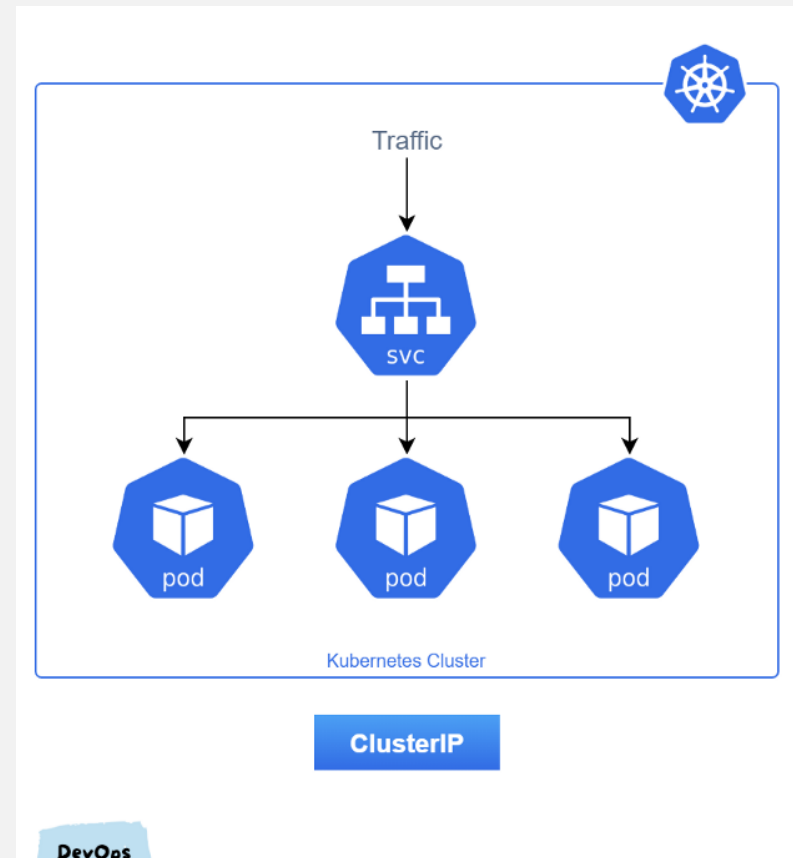


KUBERNETES: POD



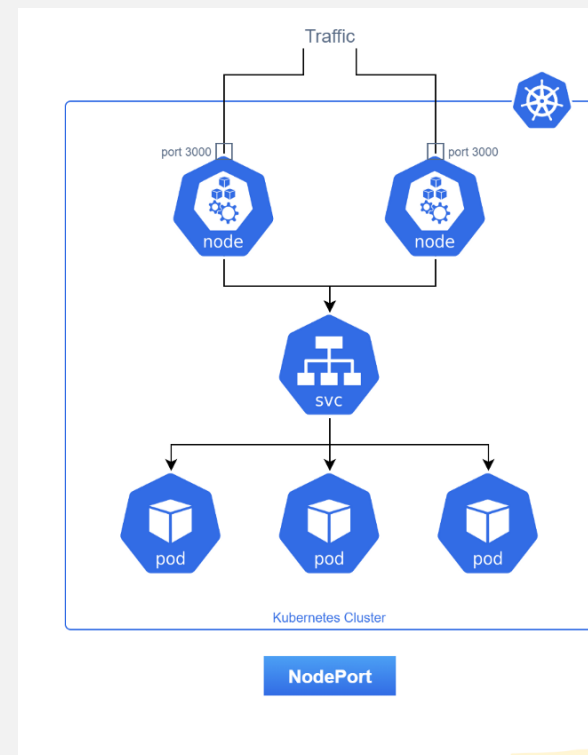
KUBERNETES: SERVICES

- **Servicios (svc):**
 - son el punto de entrada al pod.
 - Interfaz de acceso al los pods.
- **ClusterIP:** Cada **pod** tiene un ip que lo identifica y direcciona internamente en el cluster. Un svc de tipo ClusterIP es el default y direcciona **internamente** hacia los pods.



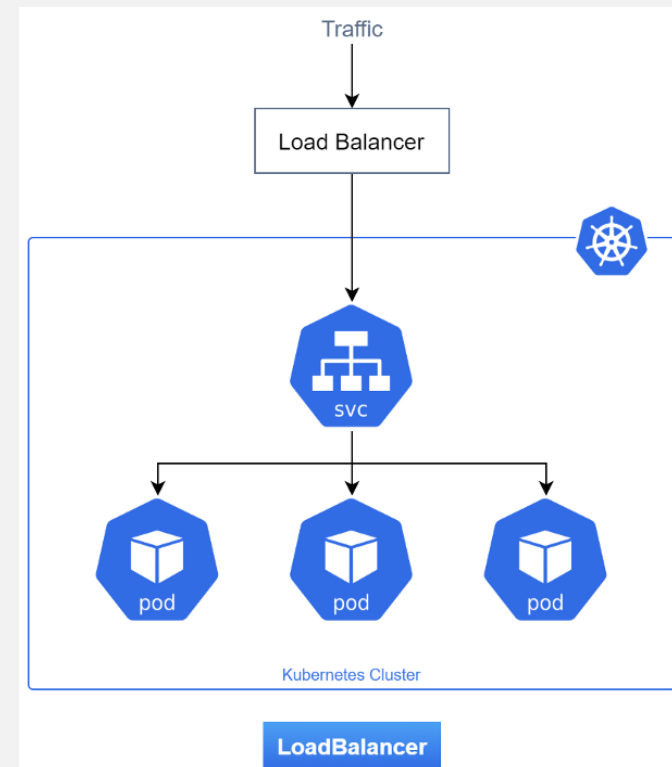
KUBERNETES: SERVICES

- **NodePort:** Un svc de tipo NodePort.
Rango de 30000–32767.



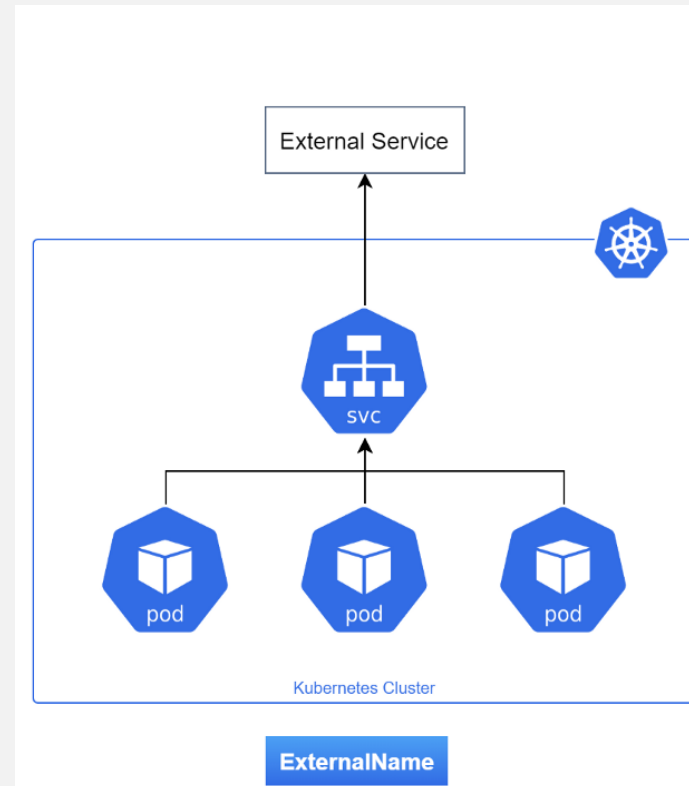
KUBERNETES: SERVICES

- **LoadBalancer:** IP público



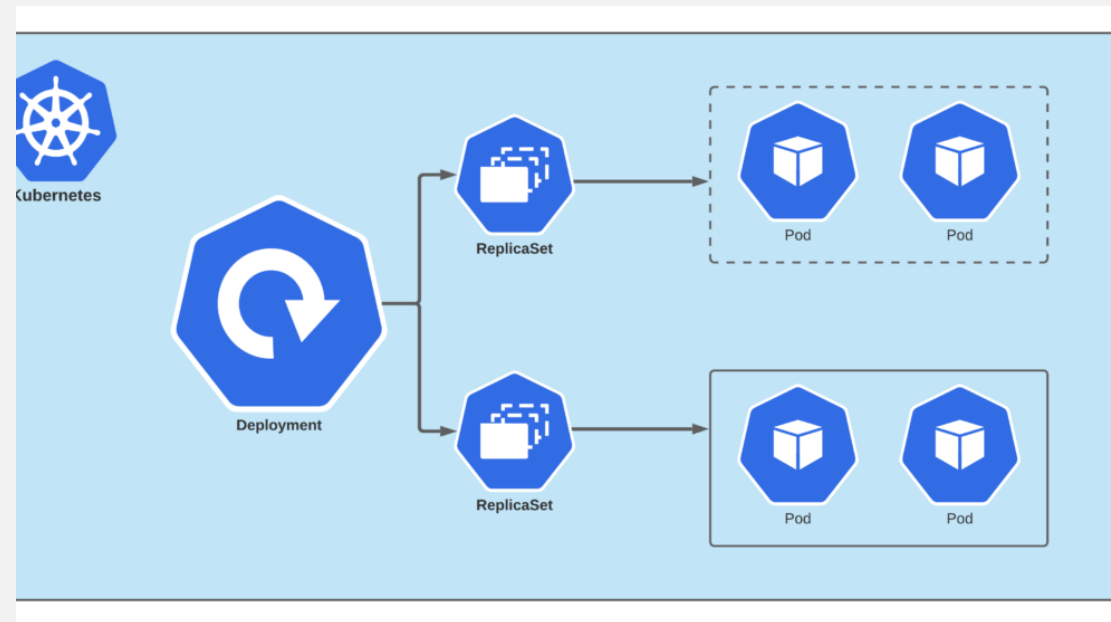
KUBERNETES: SERVICES

- **ExternalService:** Un svc que mapea comunicación hacia un servidor externo al clúster como un servicio interno



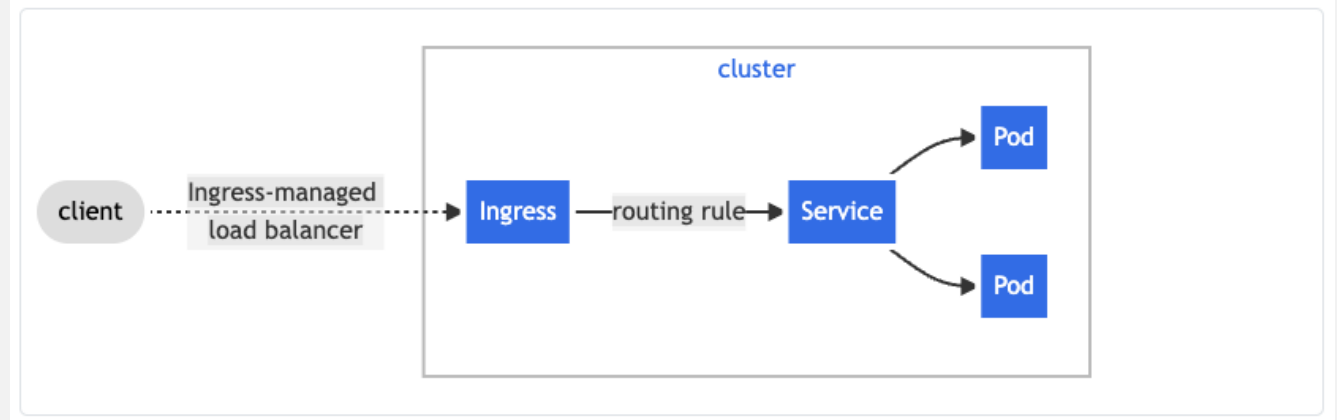
KUBERNETES: DEPLOYMENTS

- Un **deployment** es una instrucción declarativa del **estado** de una serie pods y sus réplicas.
- Controla cuantos **Pods** se crean con un **ReplicaSet**.
- En realidad **NO** creamos manualmente replicas ni pods; el deployment cumple dicha tarea



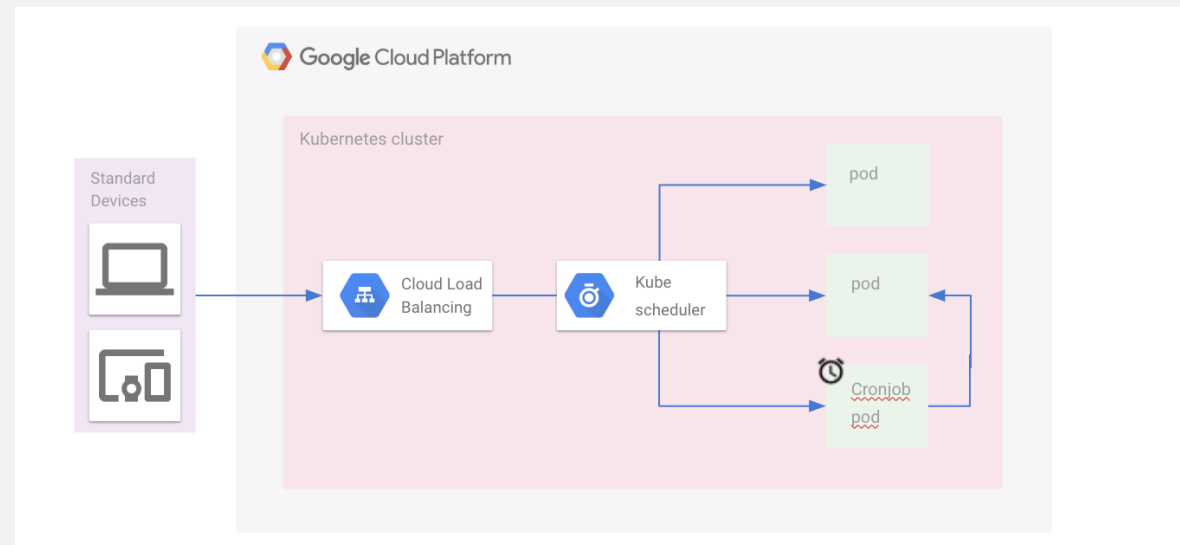
KUBERNETES: INGRESS

- Es una serie de recursos en Kubernetes
- El ingress expone comunicación HTTP y HTTPS fuera del clúster hacia los servicios.
- Se basa en reglas de enrutamiento.
- Más popular: nginx-ingress-controller



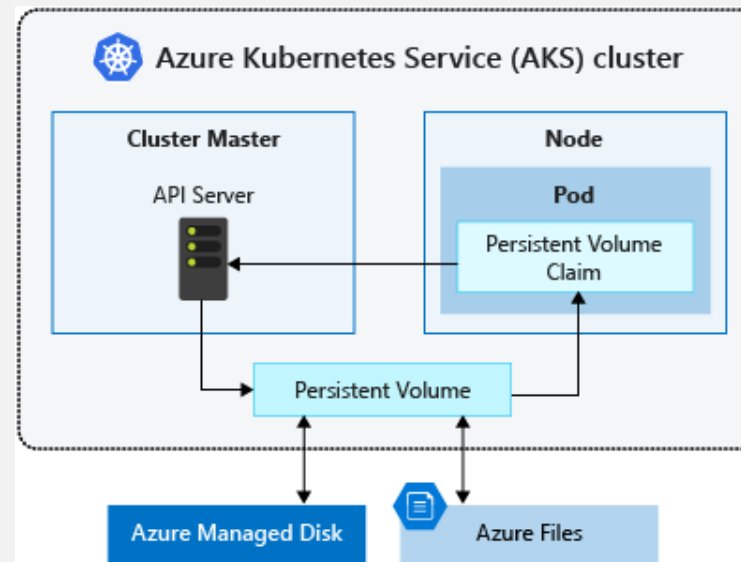
KUBERNETES: JOB Y CRONJOBS

- Un **job** crea uno o más job y continua la ejecución hasta que se alcance un número de intentos o finalice correctamente. Corren en paralelo.
- **Cronjobs:** Jobs que se repiten periódicamente.



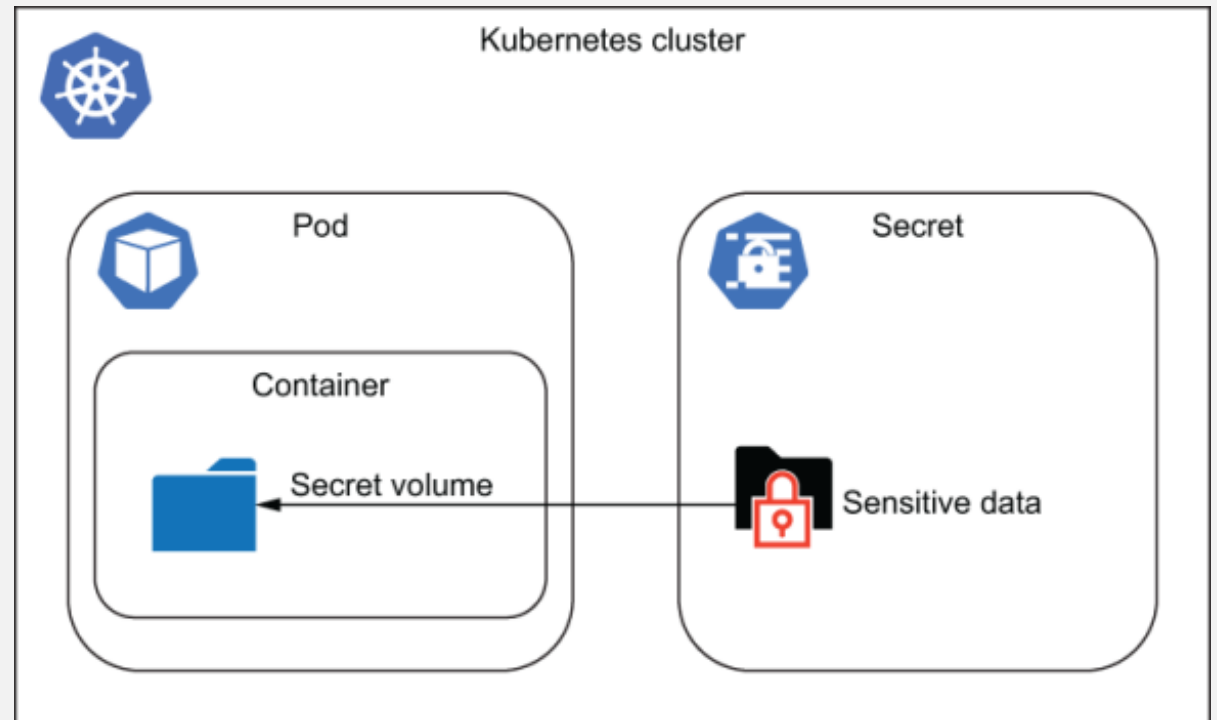
KUBERNETES: PERSISTENT VOLUMEN Y PERSISTENT VOLUMEN CLAIM

- Persistent Volumen (PV)
 - Almacenamiento físico (externo o local al clúster)
- Persistent Volumen Claim (PVC)
 - Solicitud para crear un volumen para pod de X espacio.



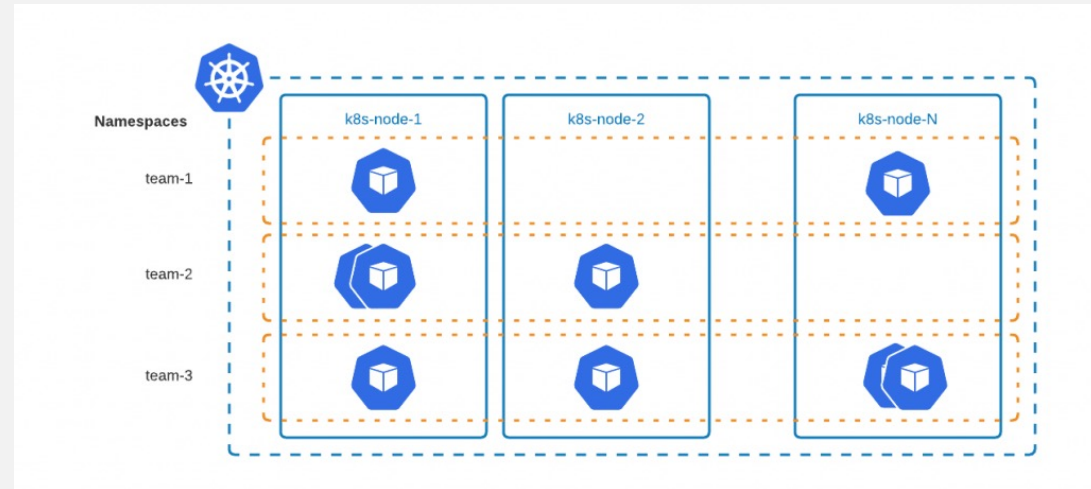
KUBERNETES: SECRETS

- Almacenan información **sensible** que será accedida más adelante por otros recursos, como **pods** por ejemplo.
- Valores encriptados



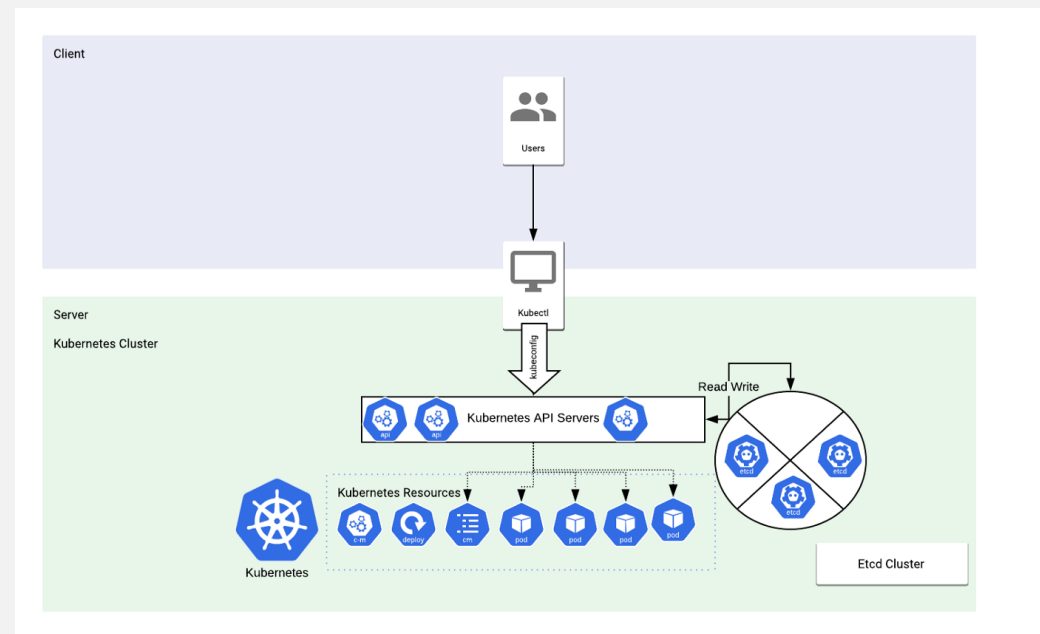
KUBERNETES: NAMESPACES

- Los **namespaces** permiten la separación lógica de recursos
- Evitar conflictos en nombres, limitar acceso a recursos y más



KUBERNETES: KUBECTL

- La línea de comandos para acceder a un clúster de K8 es a través del comando **kubectl**



KUBECTL: DEMO

CASO DE ESTUDIO: KUBERNETES

¿CÓMO SEGUIR?



<https://www.docker.com/blog/tag/docker-certified-associate/>



<https://www.cncf.io/certification/cka/>

¿CÓMO SEGUIR?

<https://kubernetes.io/es/docs/home/>

<https://www.youtube.com/watch?v=X48VuDVv0do>

<https://www.udemy.com/course/learn-kubernetes/>

https://helm.sh/docs/chart_template_guide/