

Kubernetes

K8S - Conceptos Básicos



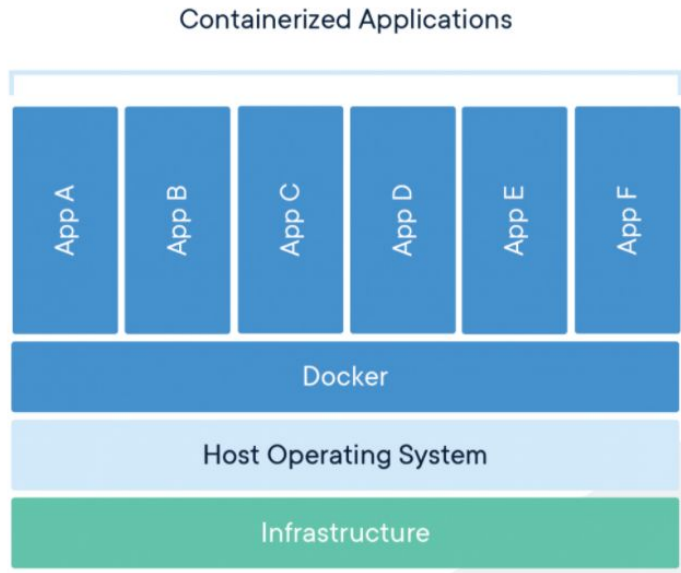
kubernetes

Contenedor

Un contenedor es una unidad estándar de software que **empaqueta el código y todas sus dependencias** para que la aplicación se ejecute de forma rápida y confiable de un entorno informático a otro.

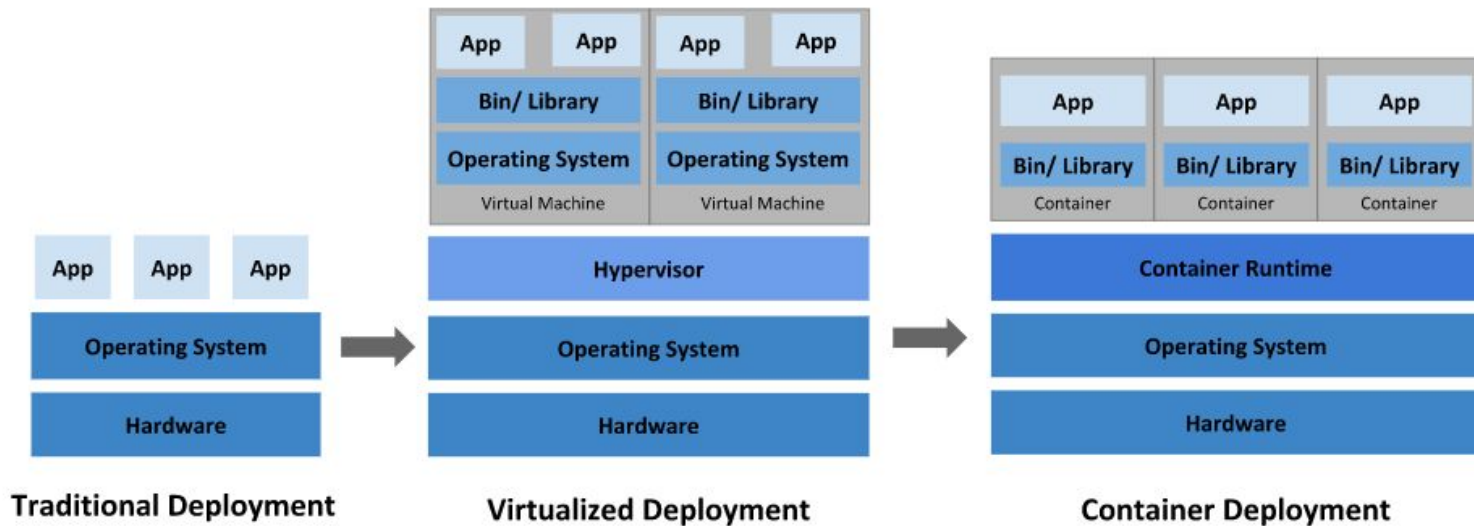
Docker

Es un proyecto de código abierto que **automatiza** el despliegue de aplicaciones dentro de contenedores de software.



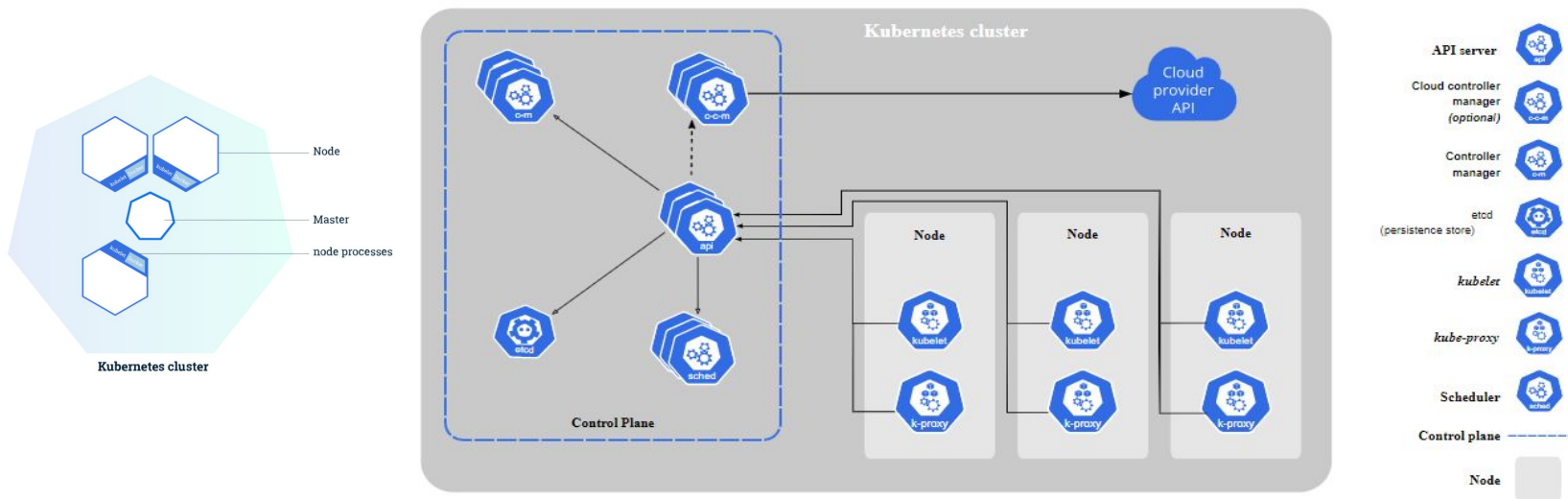
Kubernetes

Es un sistema para ejecutar y coordinar aplicaciones en contenedores en un **grupo de máquinas**.



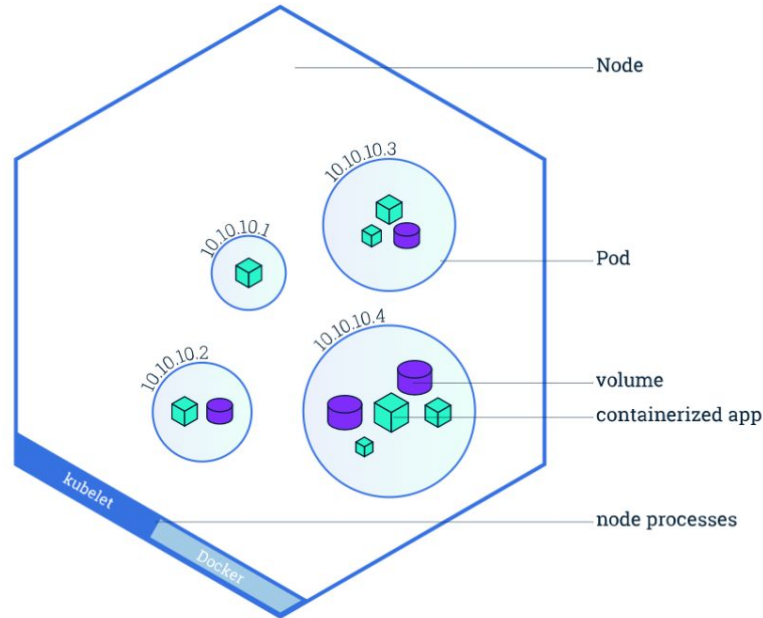
Arquitectura de Kubernetes

Un clúster de Kubernetes consta de un **master** y un **conjunto de workers**, llamados **nodos**, que ejecutan aplicaciones en contenedores. Cada clúster tiene al menos un nodo worker.



Nodo o Worker

Es la entidad encargada de contener los componentes desplegados.

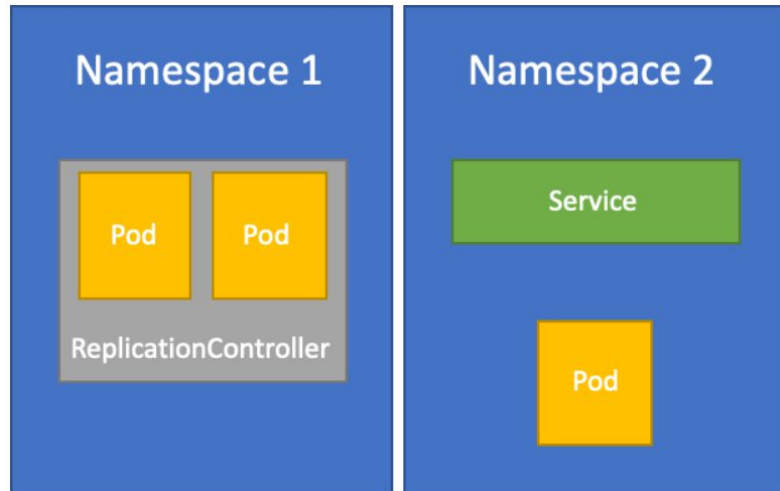


Namespaces

Los Namespaces, están pensados para su uso en entornos con muchos usuarios distribuidos en varios equipos o proyectos.

```
kubectl get namespace
```

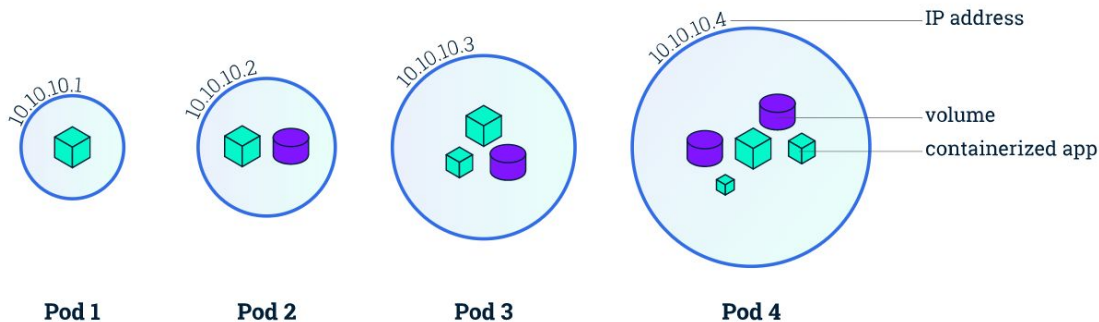
NAME	STATUS	AGE
default	Active	1d
kube-node-lease	Active	1d
kube-public	Active	1d
kube-system	Active	1d



Pods

Los pods son las unidades de computación implementables más pequeñas que puede crear y administrar en Kubernetes.

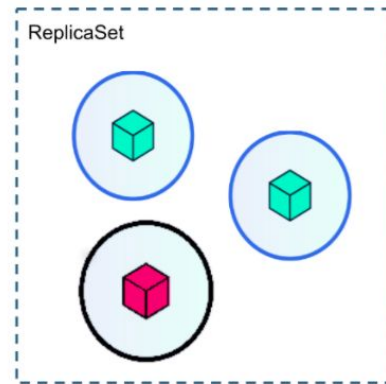
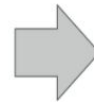
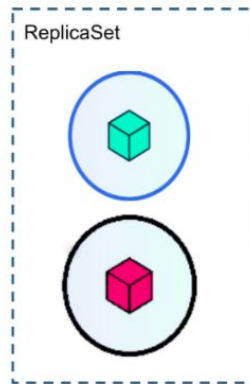
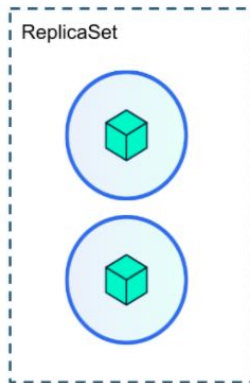
```
apiVersion: v1
kind: Pod
metadata:
  name: mongodb
spec:
  containers:
  - image: mongo
    name: mongodb
```



ReplicaSet

El propósito de un ReplicaSet es mantener un conjunto estable de Pods de réplica en ejecución en cualquier momento. Como tal, a menudo se usa para garantizar la disponibilidad de un número específico de Pods idénticos.

```
apiVersion: extensions/v1beta1
kind: ReplicaSet
metadata:
  name: nginx
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
```

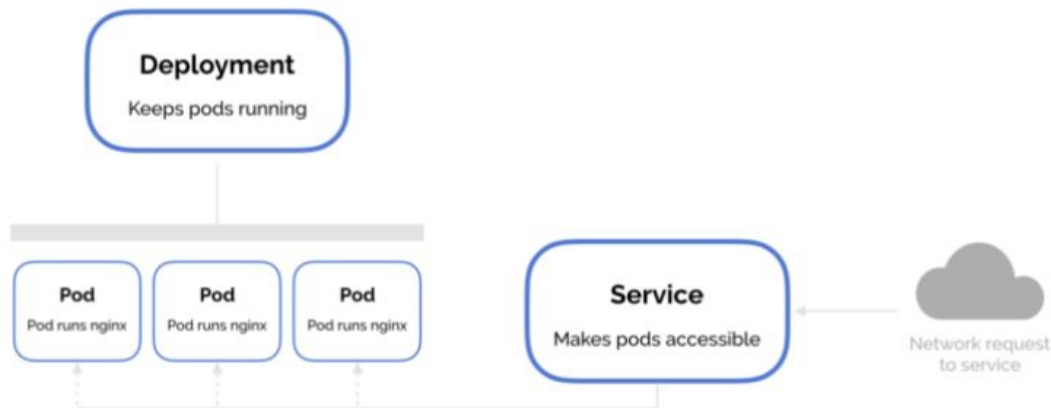


Services

Es una forma abstracta de exponer una aplicación que se ejecuta en un conjunto de Pods como un servicio de red.

Tipos:

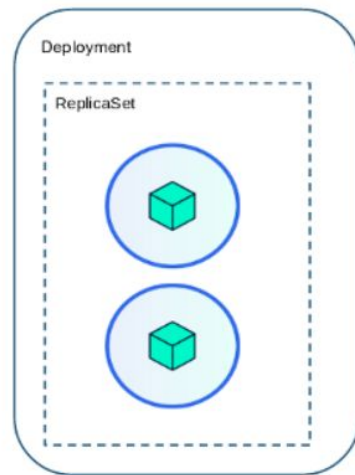
- ClusterIP
- NodePort
- LoadBalancer
- Ingress, que aunque no es un servicio propiamente, tiene una gran relación



Deployments

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  name: nginx
  namespace: default
  labels:
    app: nginx
spec:
  revisionHistoryLimit: 2
  strategy:
    type: RollingUpdate
  replicas: 2
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
        - image: nginx
          name: nginx
          ports:
            - name: http
              containerPort: 80
```

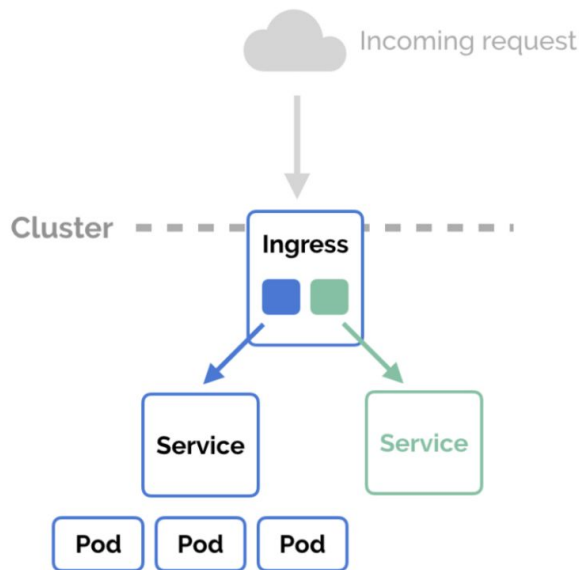
Los Deployments representan un conjunto de varios pods idénticos sin identidades únicas. Un Deployment ejecuta varias réplicas de la aplicación y reemplaza automáticamente cualquier instancia que falle o deje de responder.



Ingress

Objeto API que administra el acceso externo a los servicios en un clúster, generalmente HTTP.

Ingress puede proporcionar equilibrio de carga, terminación SSL y alojamiento virtual basado en nombres.

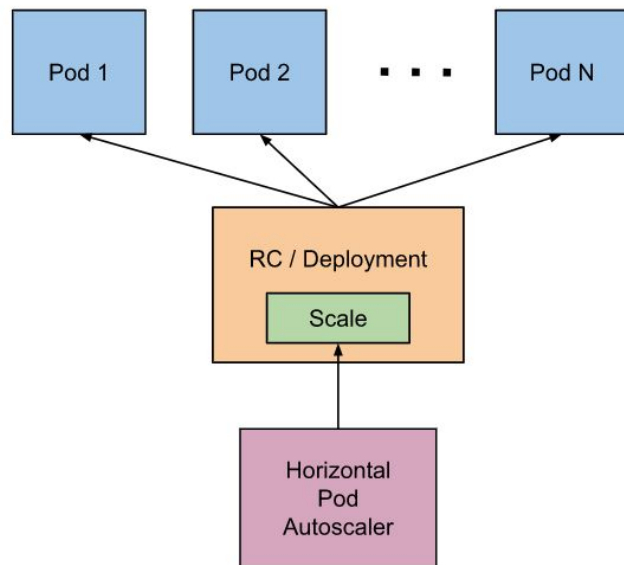


Nota: Ingress es un servicio que debe habilitarse, no viene por defecto configurado en una instalación de Kubernetes, ejemplo para microk8s se debe aplicar el "microk8s enable ingress"



HPA Horizontal Pod Autoscaler

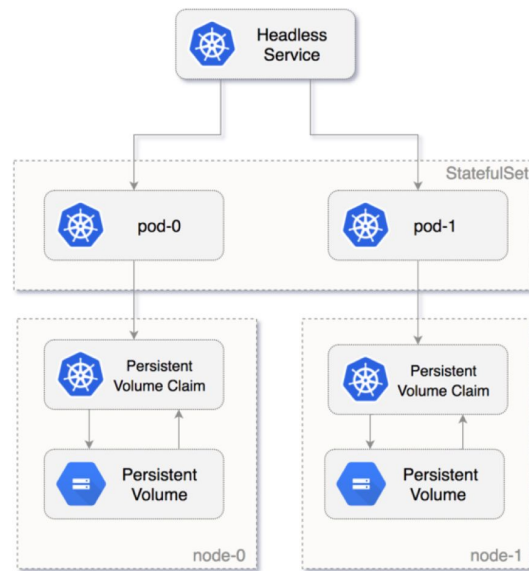
Escala automáticamente la cantidad de pods en un controlador de replicación, implementación, conjunto de réplicas , generalmente en función al uso de CPU.



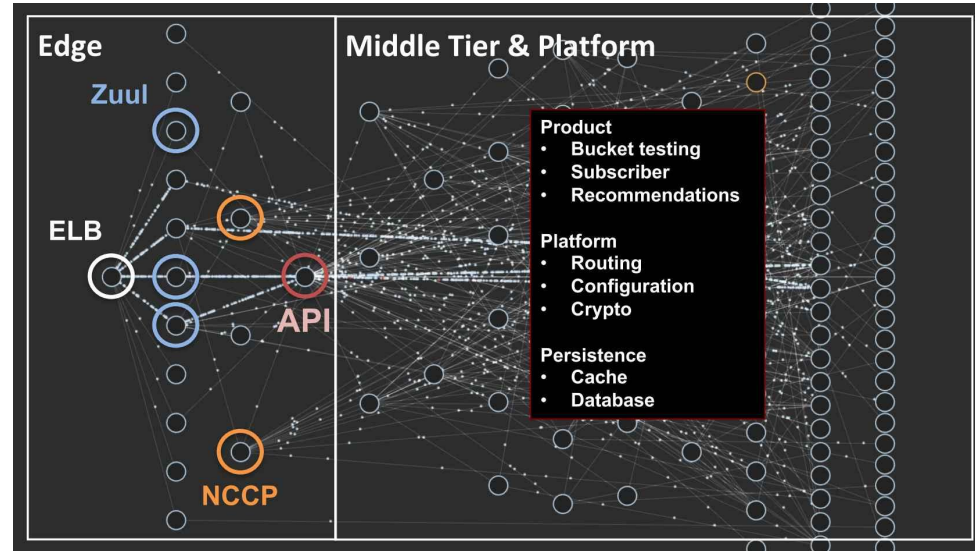
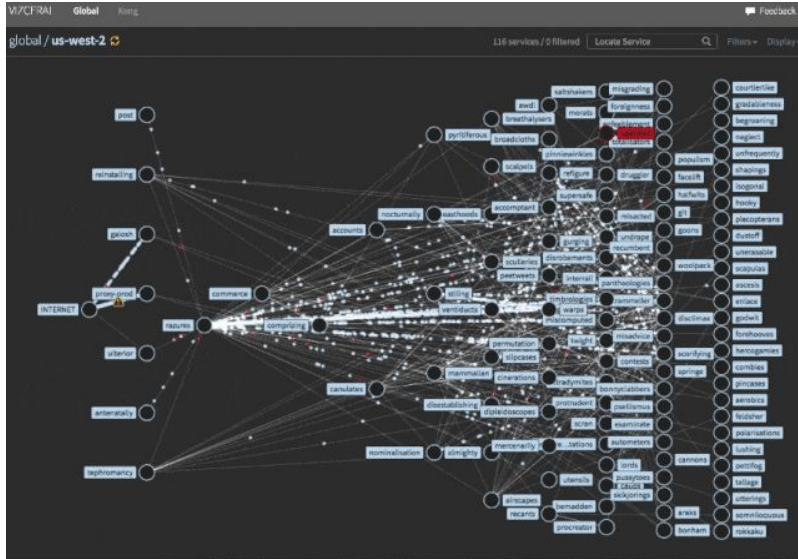
Volumes

Un PersistentVolume (PV) es una pieza de almacenamiento en el clúster que ha sido aprovisionada por un administrador o aprovisionada dinámicamente mediante clases de almacenamiento.

Un PersistentVolumeClaim (PVC) es una solicitud de almacenamiento por parte de un usuario. Es similar a un Pod. Pods consumen recursos de los nodos y los PVCs consumen los recursos del PV.



K8S en Producción a gran escala



https://www.youtube.com/watch?v=CZ3wluvmHeM&ab_channel=InfoQ



Por FIN

