

Kubernetes

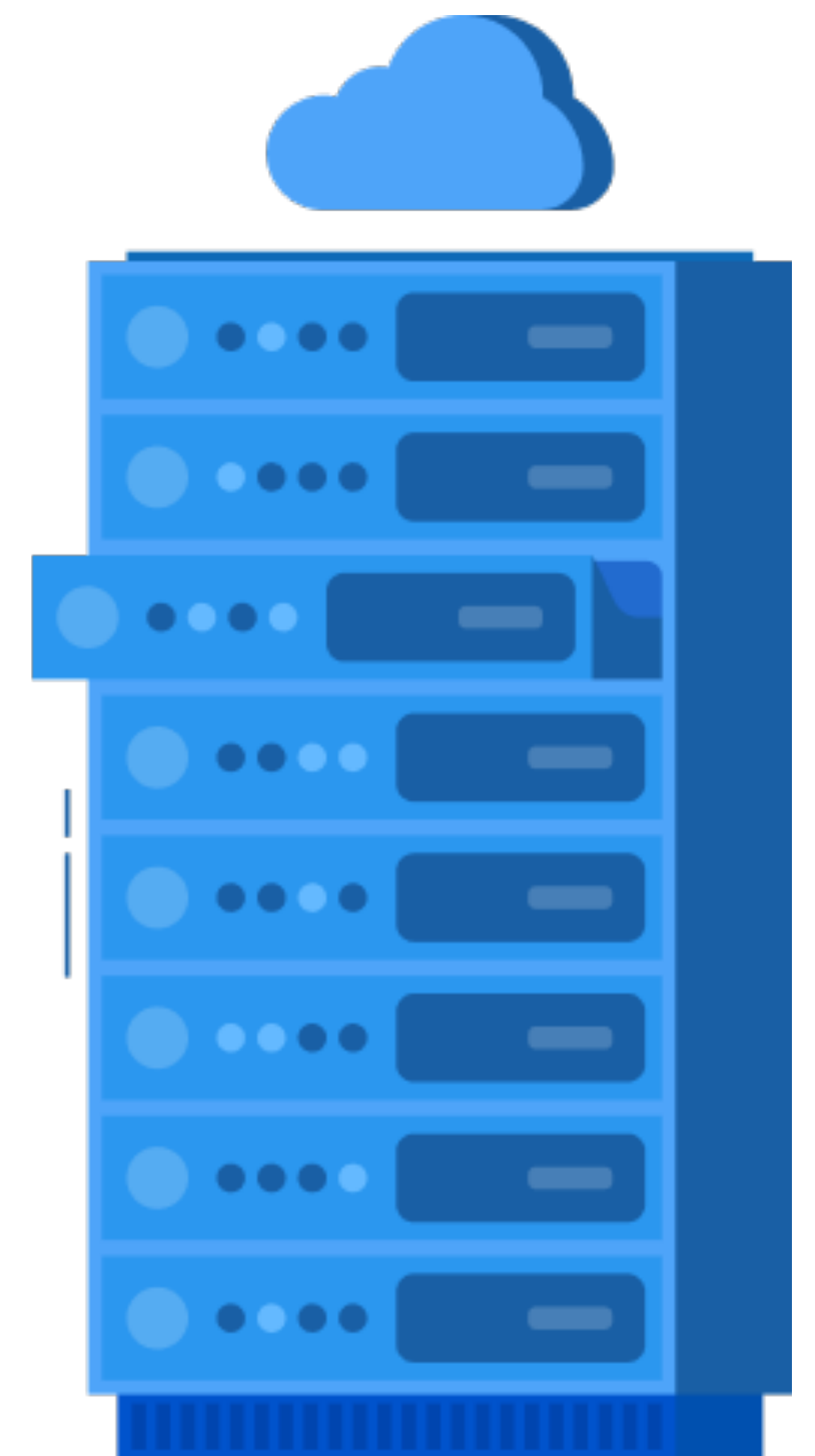
Clase 14 – 21/5/2024



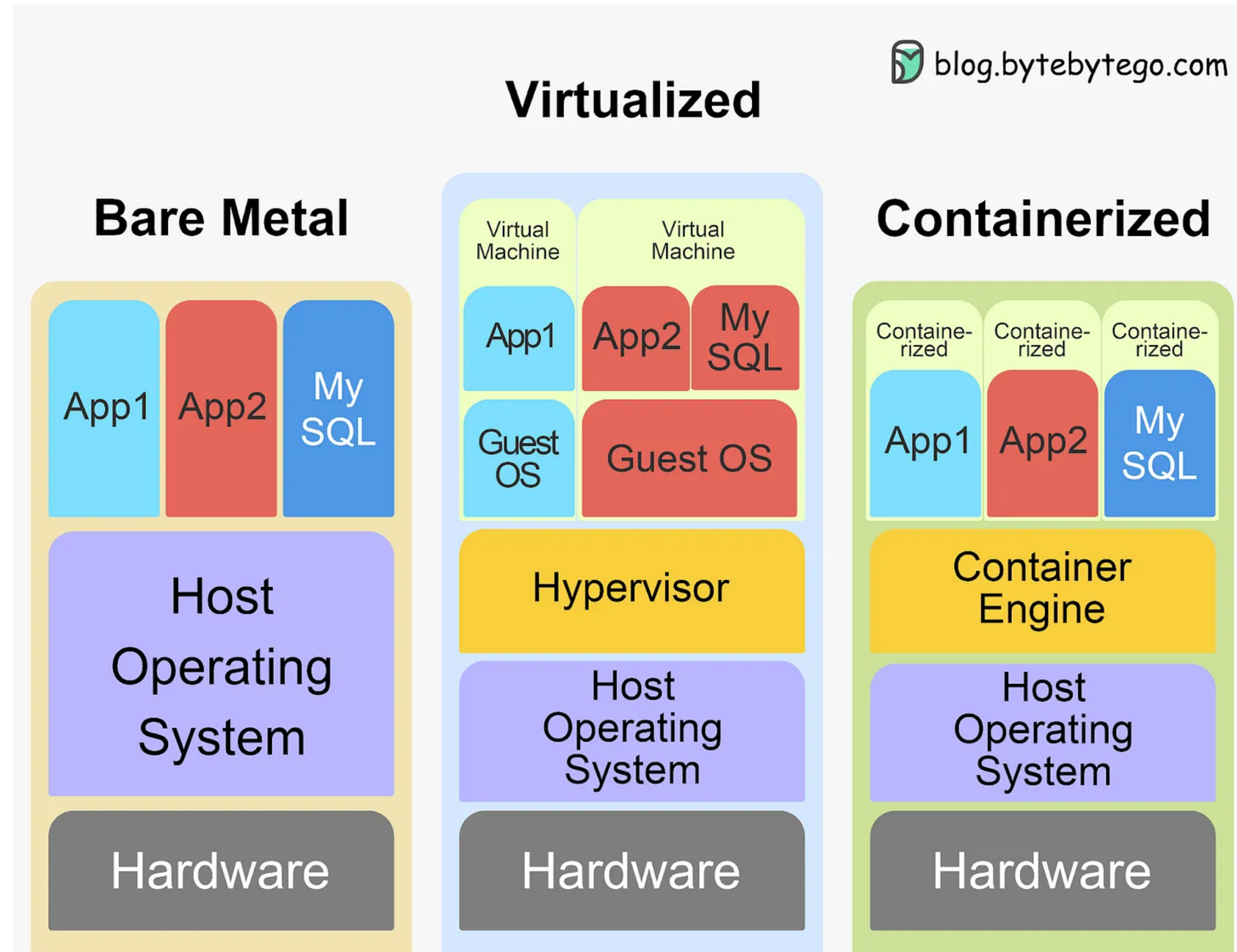
Bootcamp de Backend Avanzado @ Código Facilito

Kubernetes

- ¿Qué es Kubernetes?
 - Arquitectura
 - Componentes principales
 - Archivos de Configuración YAML
- Minikube y Kubectl
- Demo



Un poco de historia...



¿Qué es **Kubernetes**?



¿Qué es **Kubernetes**?



**CLOUD NATIVE
LANDSCAPE**

Google



**CLOUD NATIVE
LANDSCAPE**

<https://landscape.cncf.io/>

¿Qué es **Kubernetes**?

- plataforma
 - portable
 - extensible
 - de código abierto
- para administrar cargas de trabajo y servicios



¿Qué **problema** resuelve K8s?



¿Qué **problema** resuelve K8s?

- altamente disponible
- un ecosistema de componentes y herramientas
 - que hacen más fácil el desplegar,
- escalar y
- administrar aplicaciones.

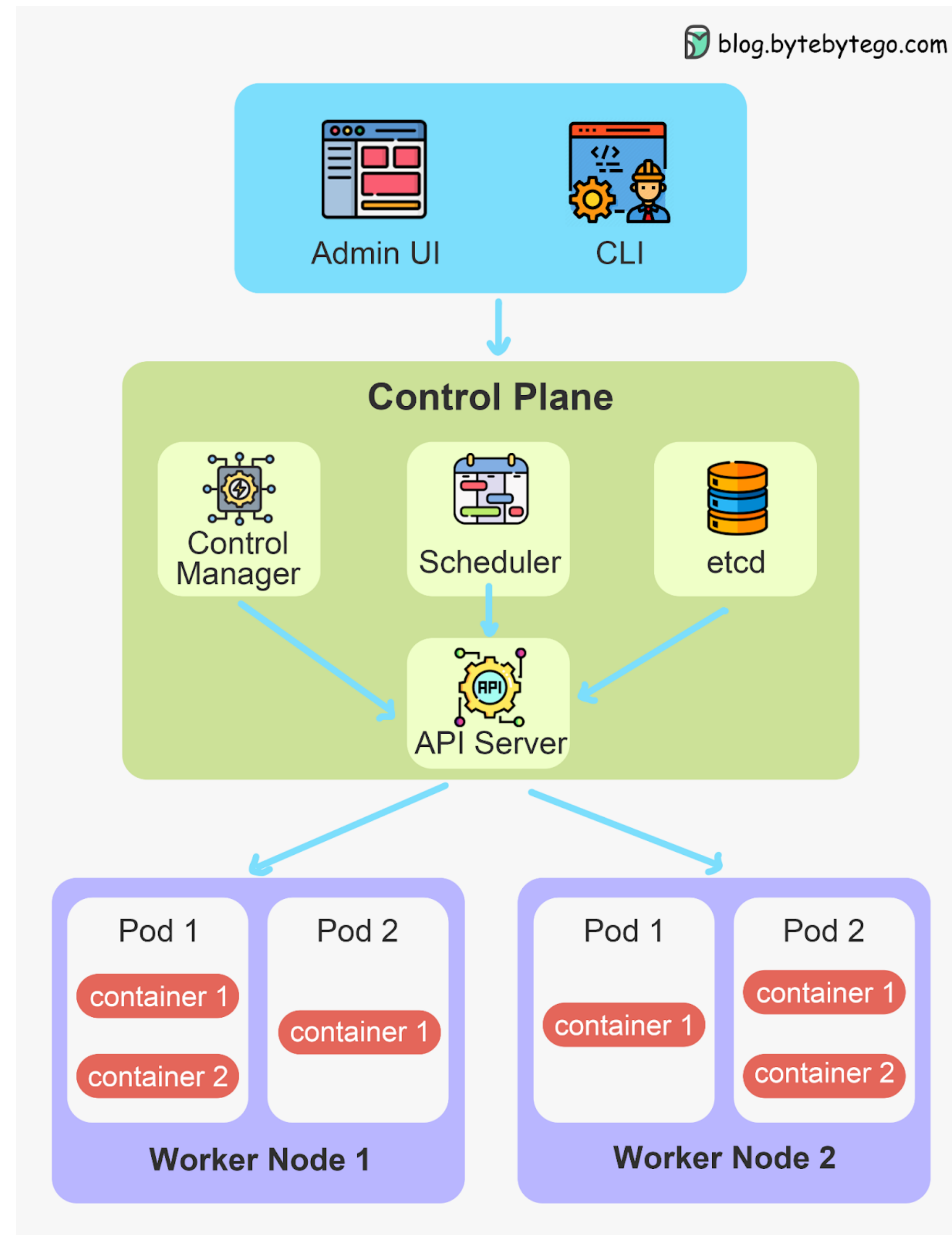


¿Qué **beneficios** tiene K8s?

- Ágil creación y despliegue de aplicaciones
- Desarrollo, integración y despliegue continuo
- Aislamiento de recursos
- Utilización de recursos



Arquitectura



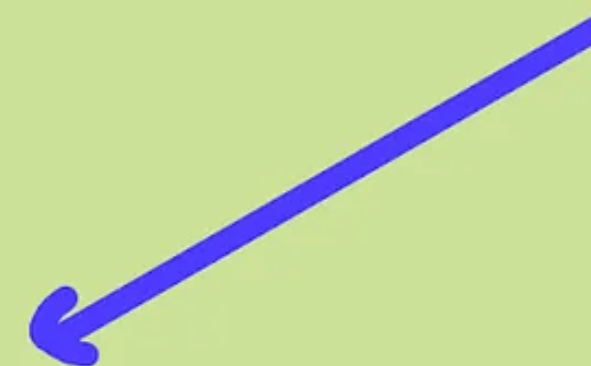
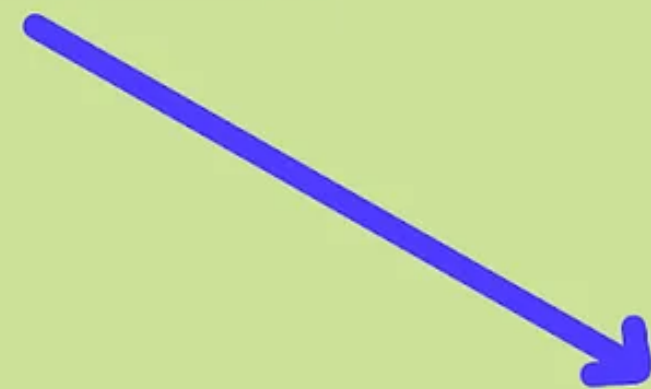
Control Plane


Controller
Manager


Scheduler


API Server


etcd



Pod 1

container 1

container 2

Pod 2

container 1

container runtime



kubelet

kube-proxy

Worker Node 1



Kubernetes Objects



Pod



Service



Controllers



System Resources



Container



Volume



iptables rules

Update

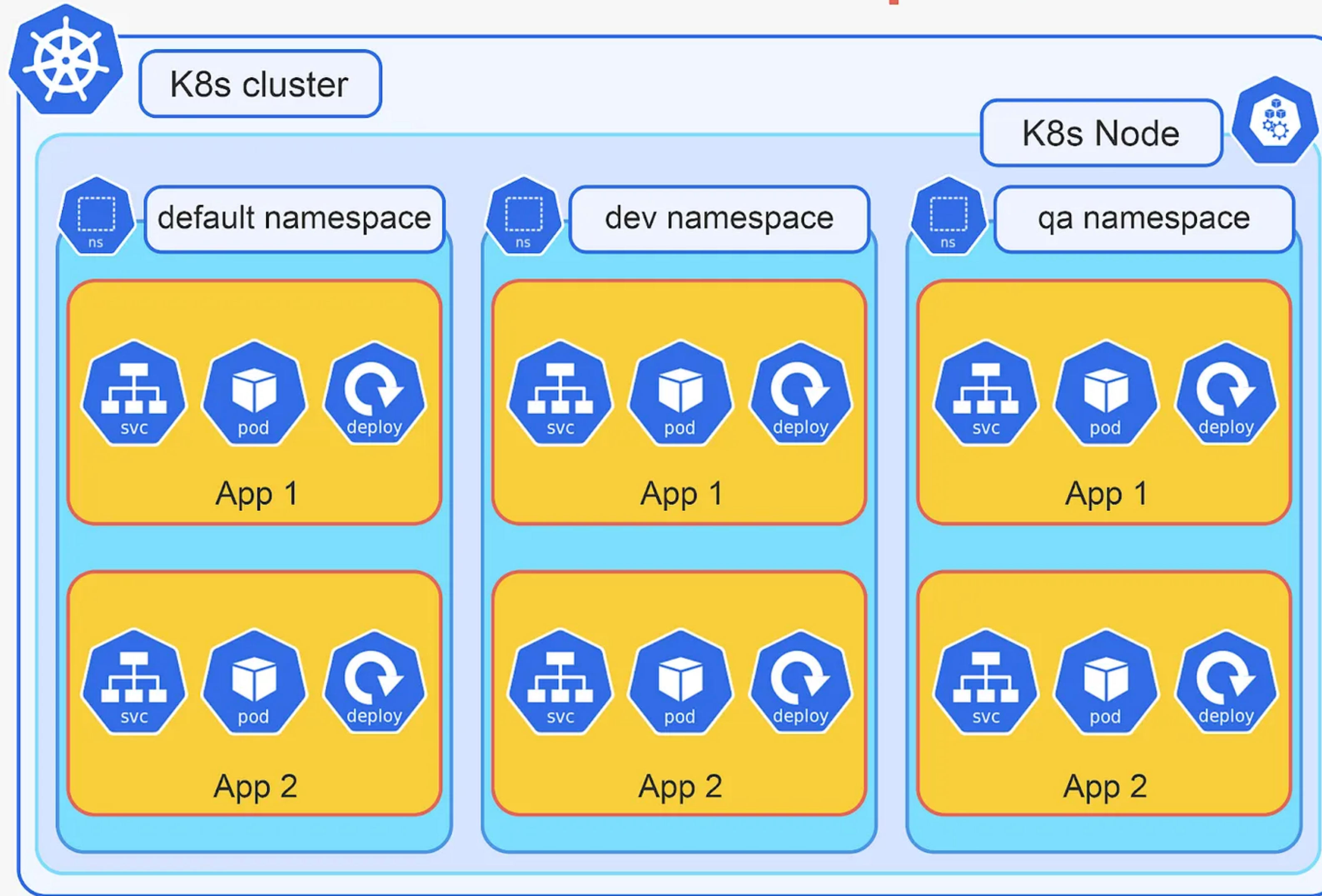
Watch

Watch

Update



Kubernetes - Namespaces



Componentes principales



Pod

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx
spec:
  containers:
  - name: nginx
    image: nginx:1.14.2
    ports:
    - containerPort: 80
```



```
kubectl apply -f https://
k8s.io/examples/pods/
simple-pod.yaml
```

ReplicaSet

```
apiVersion: apps/v1
kind: ReplicaSet
metadata:
  name: frontend
  labels:
    app: guestbook
    tier: frontend
spec:
  # modify replicas according to your case
  replicas: 3
  selector:
    matchLabels:
      tier: frontend
  template:
    metadata:
      labels:
        tier: frontend
    spec:
      containers:
        - name: php-redis
          image: us-docker.pkg.dev/google-
samples/containers/gke/gb-frontend:v5
```



```
kubectl apply -f https://
kubernetes.io/examples/
controllers/frontend.yaml
```


Deployment

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx-deployment
  labels:
    app: nginx
spec:
  replicas: 3
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9
        ports:
        - containerPort: 80
```



```
kubectl apply -f https://
k8s.io/examples/
controllers/nginx-
deployment.yaml
```

StatefulSet

- Identificadores de red estables, únicos.
- Almacenamiento estable, persistente.
- Despliegue y escalado ordenado, controlado.
- Actualizaciones en línea ordenadas, automatizadas.



StatefulSet

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  labels:
    app: nginx
spec:
  ports:
  - port: 80
    name: web
  clusterIP: None
  selector:
    app: nginx
```



```
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: web
spec:
  serviceName: "nginx"
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: registry.k8s.io/nginx-slim:0.8
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: www
          mountPath: /usr/share/nginx/html
  volumeClaimTemplates:
  - metadata:
      name: www
    spec:
      accessModes: [ "ReadWriteOnce" ]
      resources:
        requests:
          storage: 1Gi
```

Volúmenes Persistentes



```
apiVersion: v1
kind: PersistentVolume
metadata:
  name: task-pv-volume
  labels:
    type: local
spec:
  storageClassName: manual
  capacity:
    storage: 10Gi
  accessModes:
    - ReadWriteOnce
  hostPath:
    path: "/mnt/data"
```

```
kubectl apply -f https://
k8s.io/examples/pods/
storage/pv-volume.yaml
```

Servicio



ClusterIP

NodePort

LoadBalancer

ClusterIP

- Expone el Service en una dirección IP interna del clúster.
- Al escoger este valor el Service solo es alcanzable desde el clúster.
- Este es el ServiceType por defecto.



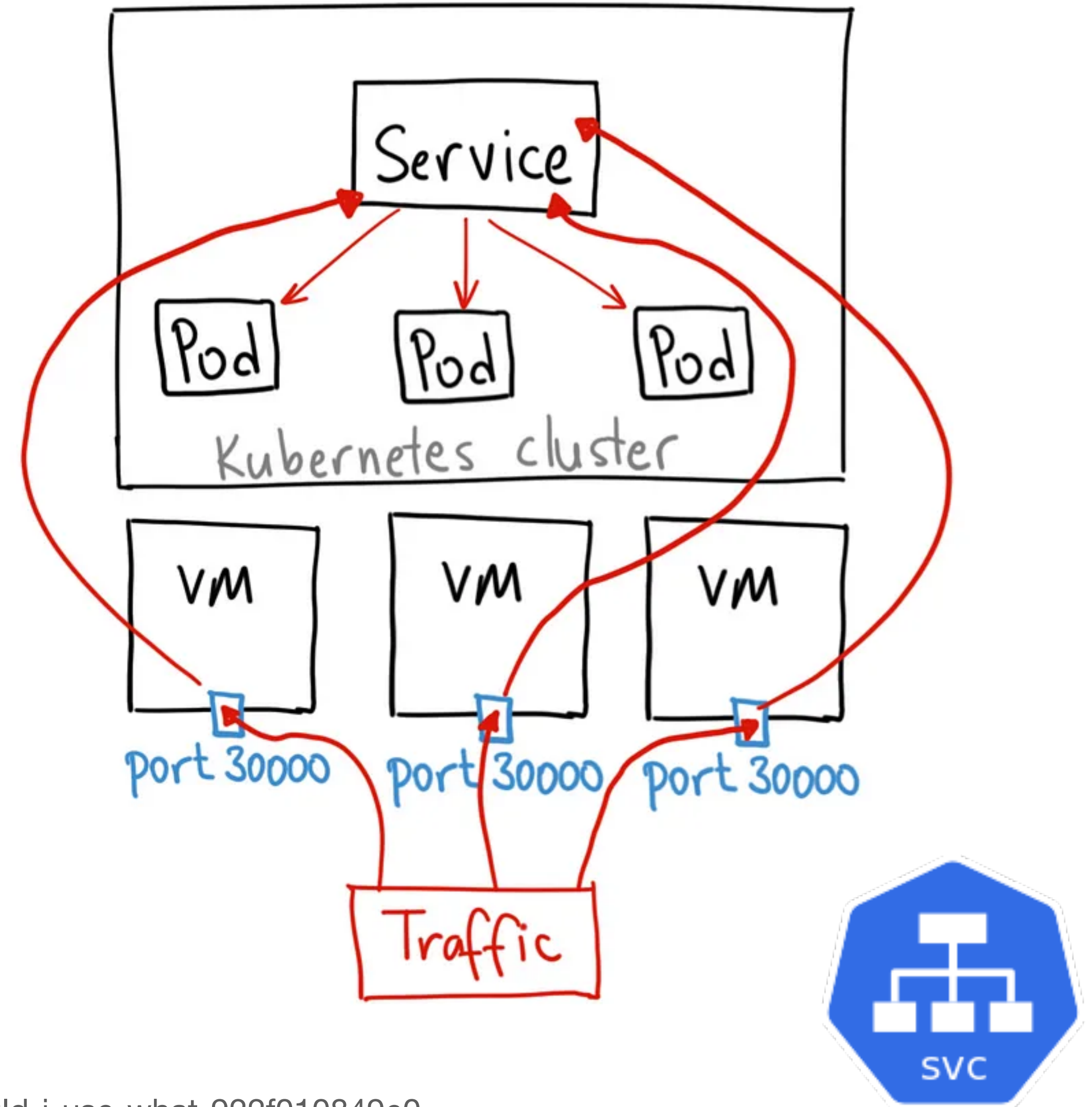
NodePort

- Expone el Service en cada IP del nodo en un puerto estático (el NodePort).
- Automáticamente se crea un Service ClusterIP, al cual enruta el NodePort del Service.
- Podrás alcanzar el Service NodePort desde fuera del clúster, haciendo una petición a `<NodeIP>:<NodePort>`.



NodePort

```
apiVersion: v1
kind: Service
metadata:
  name: mi-servicio
spec:
  type: NodePort
  selector:
    app: MiApp
  ports:
    # Por defecto y por comodidad, el
    # `TargetPort` tiene el mismo valor que el
    # campo `port`.
    - port: 80
      targetPort: 80
      # Campo opcional
      # Por defecto y por comodidad, el
      # plano de control de Kubernetes asignará el
      # puerto desde un rango (por defecto:
      # 30000-32767)
      nodePort: 30007
```



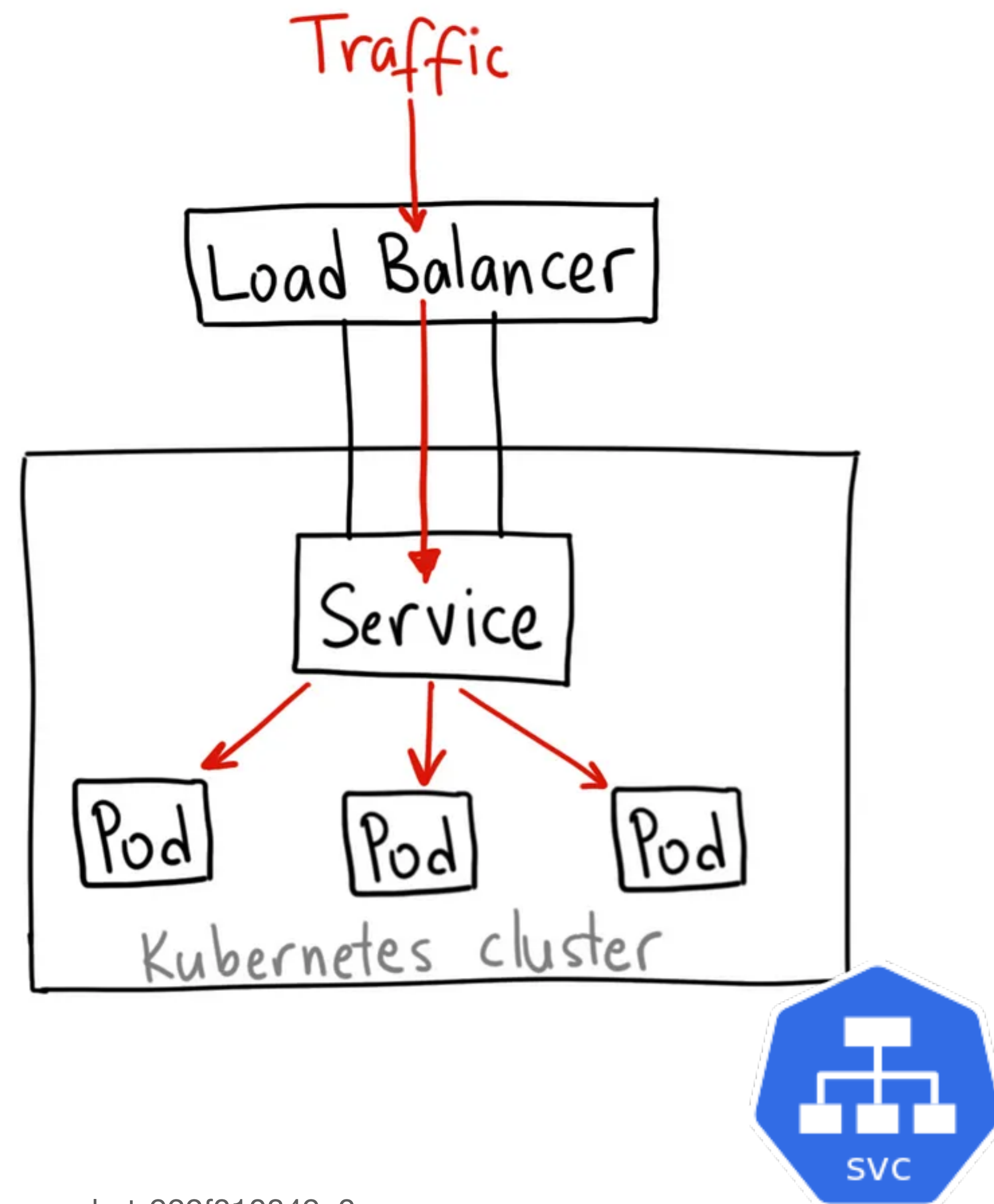
LoadBalancer

- Expone el Service externamente usando el balanceador de carga del proveedor de la nube.
- Son creados automáticamente Services NodePort y ClusterIP, a los cuales el apuntará el balanceador externo.



LoadBalancer

```
apiVersion: v1
kind: Service
metadata:
  name: mi-servicio
spec:
  selector:
    app: MiApp
  ports:
    - protocol: TCP
      port: 80
      targetPort: 9376
  clusterIP: 10.0.171.239
  type: LoadBalancer
status:
  loadBalancer:
    ingress:
      - ip: 192.0.2.127
```



Minikube y Kubectl



Instalación

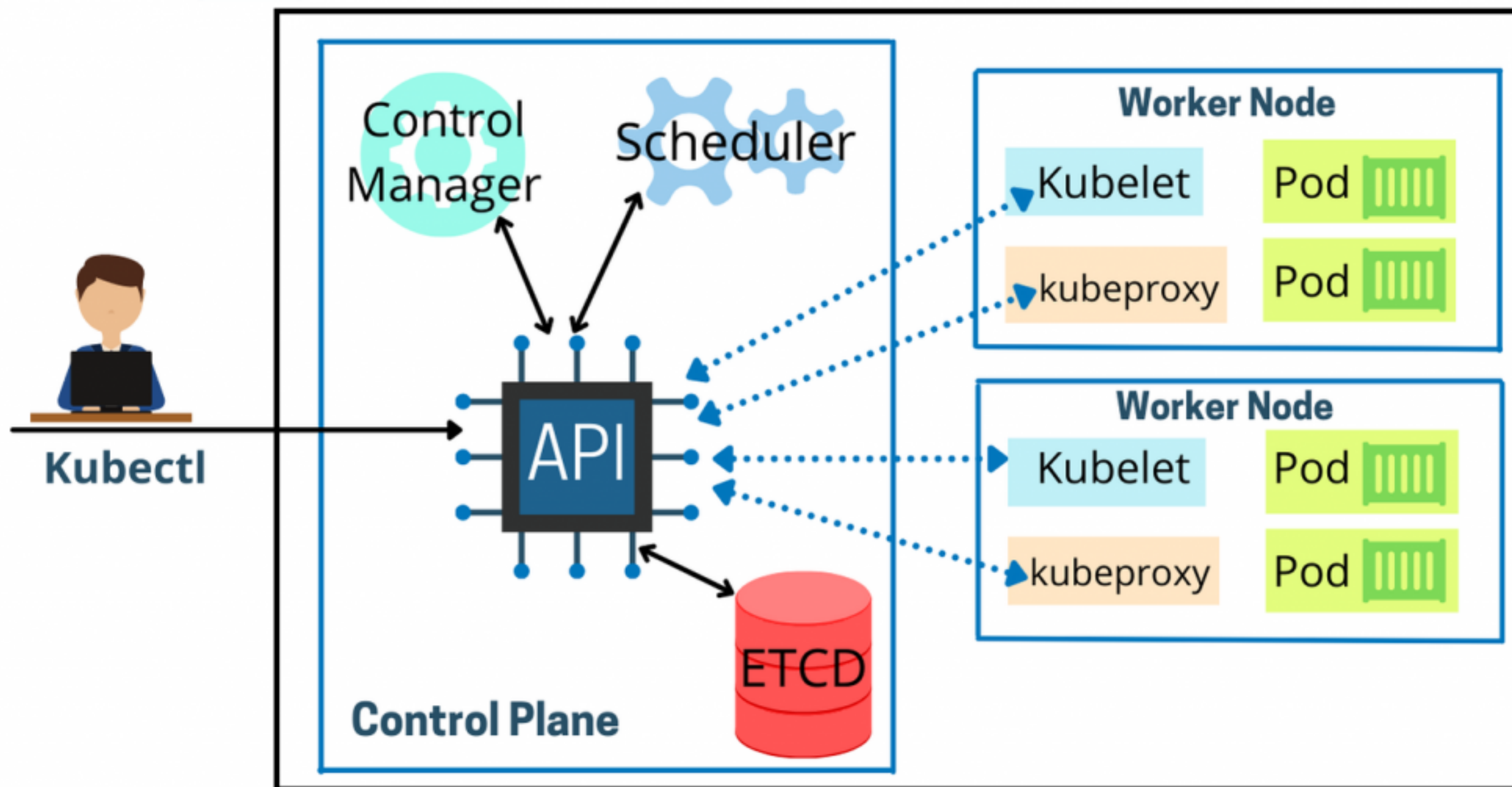
<https://minikube.sigs.k8s.io/docs/start/>





Kubernetes Cluster

Minikube instance- All in 1



¡ Gracias !

