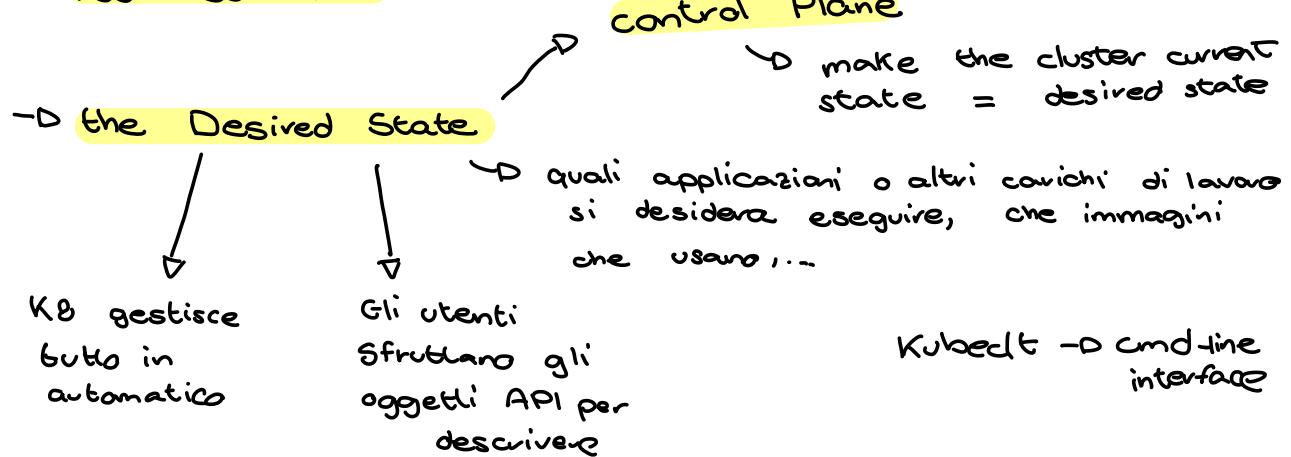
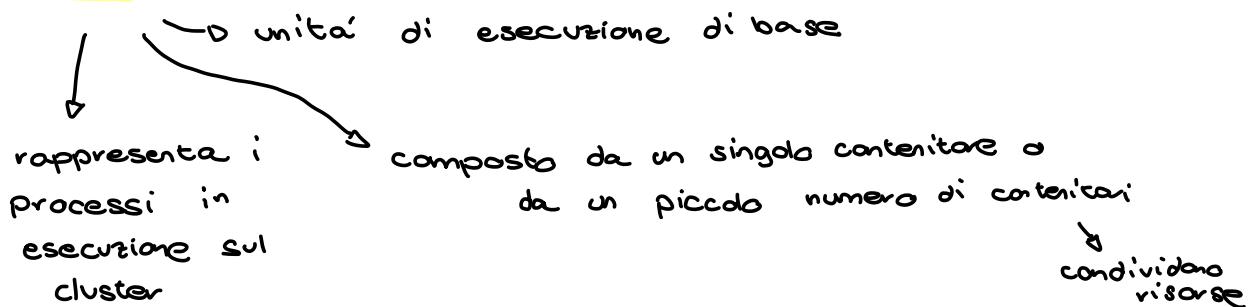


-> **Pod = Container**



-> **Pod** → unità più piccola e semplice

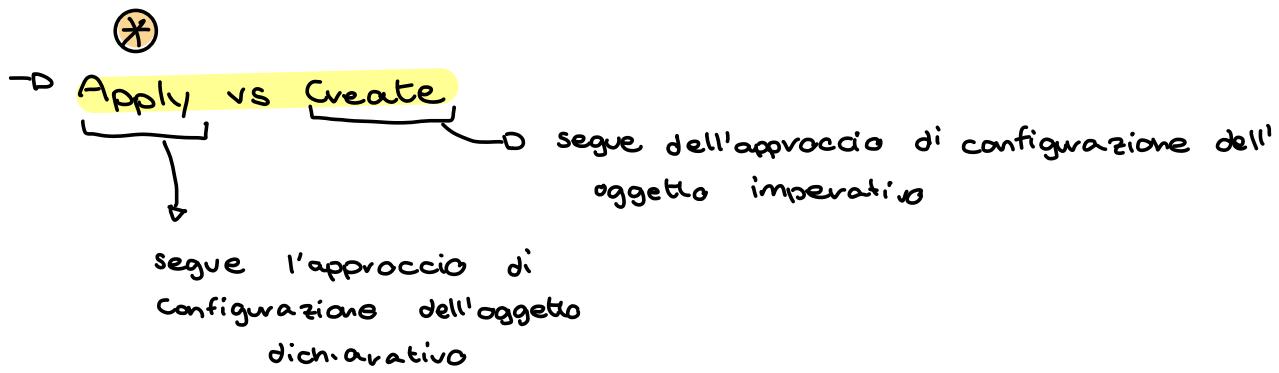


-> **Un Cluster Kubernetes**

→ è una collezione di macchine, chiamate nodi, contenitori in esecuzione.

-> **The Worker Node** → host the Pods

-> **The Control Plane Node** → controlla tutto ciò che succede in the cluster → To ensure high availability



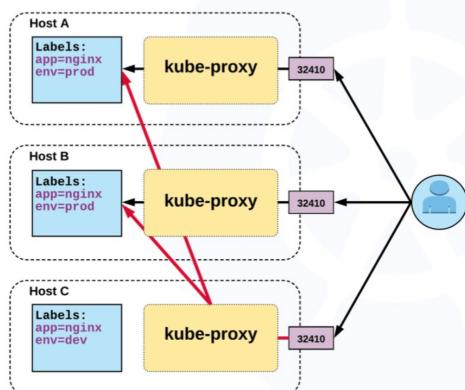
→ **Servizio** → E' un astrazione che definisce un insieme logico di Pods e una politica per accederci!

- Load balance
- Controllo traffico
- Service discovery

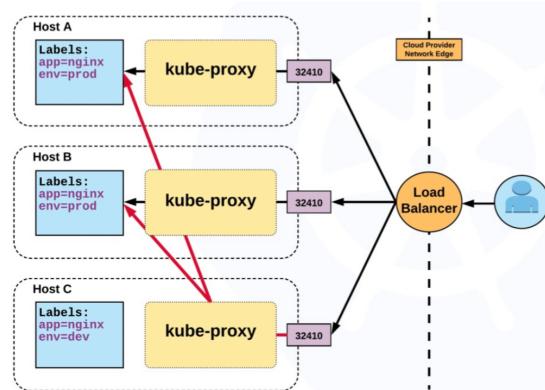
→ ClusterIP → No external access
 NodePort → expose port
 Load balance
 External Name

→ **NodePort vs LoadBalancer**

NodePort service



LoadBalancer service



→ Un servizio offre un punto di accesso astratto ai Pod, consentendo la scoperta del servizio e bilanciamento del carico tra i Pod.

→ Deployment + ReplicaSet

- ↳ Stateless
- ↳ E' un'astrazione gestire più repliche dello stesso Pod tramite ReplicaSet

Always ensure the desired number of pod are running = control

pianificazione, ridimensionamento e cancellazione



→ Stateful

↳ Gestire più repliche dello stesso Pod è un'astrazione

- ↳ L'identità del pod, inclusi nome host, rete e spazio di archiviazione, sarà mantenuta in un controller dedicato
- ↳ utile quando ho:
 - ↳ identificatori di rete univoci
 - ↳ archiviazione persistente
 - ↳ distribuzione ordinata

→ Storage

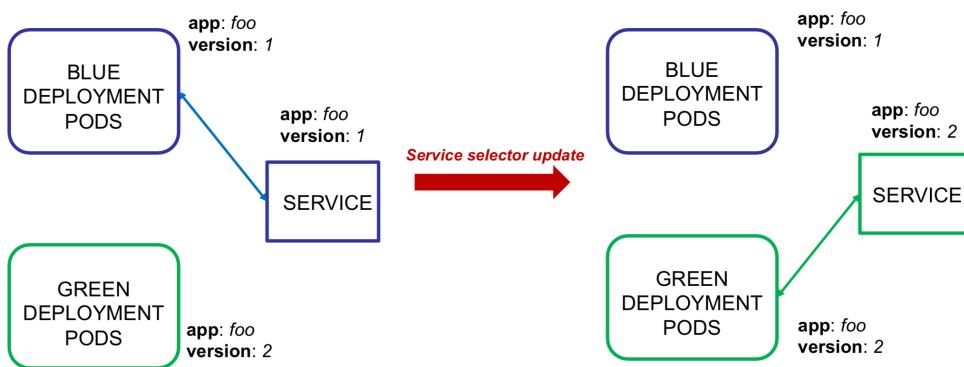
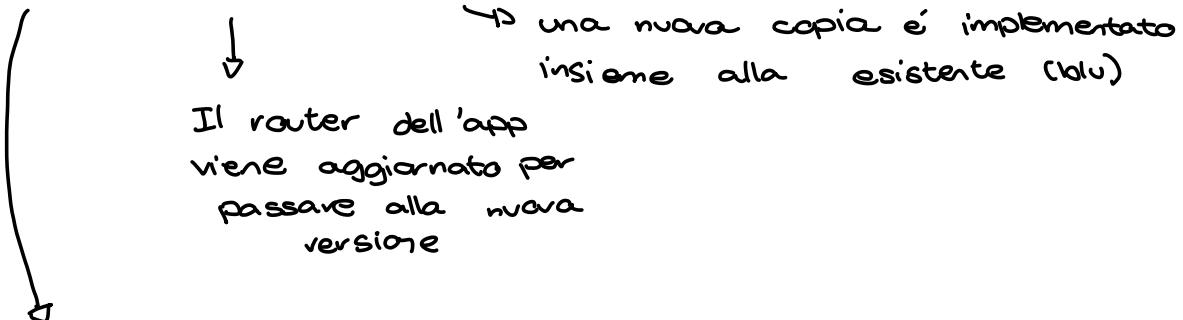
↳ Persistent VolumeClaims

- ↳ Volumes → Archiviazione legata al ciclo di vita del pod
- ↳ Persistent Volumes → esiste indipendentemente dai pod
 - ↳ b
 - ↳ Se il pod muore, lo spazio di archiviazione rimane intatto
- ↳ PVC, richiesta di archiviazione namespace
- ↳ soddisfa dei requisiti

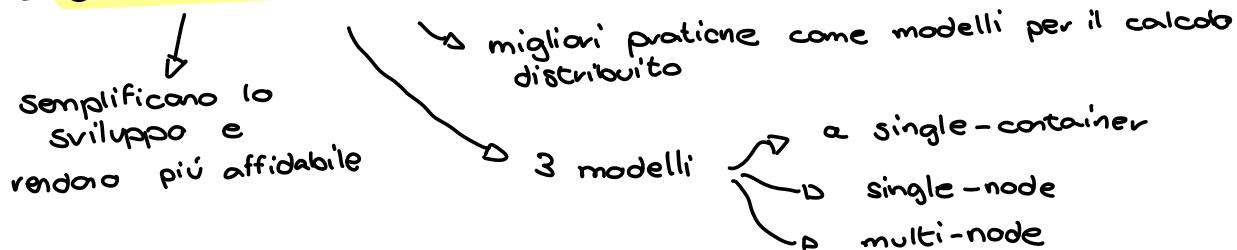
→ Secret

- ↳ Oggetto che contiene una piccola quantità di dati sensibili

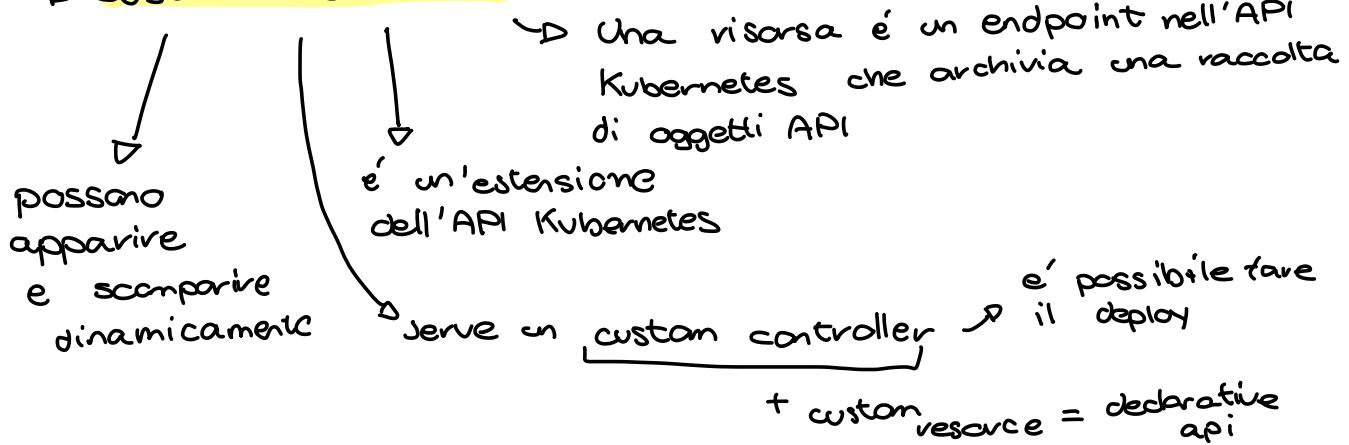
→ Blue / Green Deployments



→ Container Patterns



→ Custom Resources



Controller Kubernetes mantiene sincronizzato lo stato corrente con lo stato desiderato.

→ Config Map

lo scopo principale è che un programma in esecuzione su un pod sul cluster utilizzi il file per configurarsi

vs

Custom Resource

- Supporto di primo livello
- Nuova automazione che controlla gli aggiornamenti
- L'oggetto è un' astrazione su un insieme di risorse controllate

Rolling update
I clienti preferiscono un file rispetto alle API

→ Operator Pattern

existing software extensions

→ Catturare l'obiettivo chiave di un operatore umano che gestisce un servizio o un insieme di servizi scrivendo codice al di fuori di Kubernetes



→ Namespace → Supports multiple virtual clusters

divide cluster resource between multiple users

unique

→ Config map → Fine-grained info, separate your configuration from Pod and component

useful for storing - non sensitive, un-encrypted conf.

*- Deployment

