

Kubernetes: guida dettagliata per principianti

Cos'è Kubernetes?

Kubernetes (abbreviato anche come K8s) è una piattaforma open-source progettata da Google per automatizzare il deployment, la gestione, il bilanciamento del carico e il monitoraggio di applicazioni containerizzate. In parole semplici, Kubernetes è un "orchestratore": si occupa di far partire, fermare, spostare e gestire i container su un insieme di macchine (fisiche o virtuali) chiamato **cluster**.

Cos'è un container?

Un **container** è un ambiente isolato e leggero che contiene tutto il necessario per eseguire un'applicazione: il codice, le librerie, i file di configurazione, e il sistema di runtime. I container sono creati con strumenti come Docker e sono portabili: funzionano allo stesso modo su qualsiasi macchina, che sia il tuo PC, un server in cloud o un datacenter.

I vantaggi dei container:

- Leggeri e veloci da avviare
- Isolati tra loro
- Portabili tra ambienti diversi
- Scalabili facilmente

Perché usare Kubernetes?

Docker è perfetto per eseguire singoli container. Ma in un ambiente reale, un'app moderna è composta da decine o centinaia di container che devono:

- Comunicare tra loro
- Scalare in base al carico
- Essere monitorati continuamente
- Essere aggiornati senza downtime

- Essere ripristinati automaticamente in caso di errori

Kubernetes gestisce tutto questo **in modo automatico**, sicuro e coerente.

Architettura di Kubernetes

Kubernetes è formato da un **cluster**, cioè un insieme di macchine suddivise in due tipi:

- **Control Plane (master)**: è il cervello del sistema. Coordina il cluster, decide dove e come eseguire i container, monitora lo stato dei nodi.
- **Node (worker)**: sono i computer (fisici o virtuali) dove effettivamente girano i container.

Componenti principali del Control Plane:

- **kube-apiserver**: espone l'API di Kubernetes, è il punto d'accesso per interagire col cluster.
- **etcd**: un database chiave/valore che salva lo stato del cluster.
- **kube-scheduler**: decide su quale nodo lanciare nuovi container.
- **kube-controller-manager**: mantiene lo stato desiderato, eseguendo le logiche di controllo.
- **cloud-controller-manager**: interagisce con i servizi cloud (opzionale).

Componenti dei Node:

- **kubelet**: agente che comunica col Control Plane e gestisce i container sul nodo.
- **kube-proxy**: gestisce il networking sul nodo.
- **container runtime**: è il motore che esegue fisicamente i container (es: Docker, containerd, CRI-O).

Oggetti fondamentali di Kubernetes

Kubernetes utilizza alcuni oggetti chiave per gestire le applicazioni:

- **Pod:** l'unità più piccola di Kubernetes, contiene uno o più container che condividono rete e storage.
- **Deployment:** gestisce il numero di repliche di un'app, e lancia automaticamente i Pod.
- **Service:** espone un'app e gestisce il bilanciamento del carico tra i Pod.
- **Namespace:** suddivide logicamente il cluster in ambienti isolati (es. dev, test, prod).
- **ConfigMap & Secret:** gestiscono la configurazione e le credenziali in modo sicuro.

Cosa fa concretamente Kubernetes?

- Avvia container in modo automatico, in base alla configurazione dichiarata
- Monitora lo stato dei Pod: se un container si rompe, lo rimpiazza
- Bilancia il traffico tra i container
- Scala i container su richiesta (più utenti = più copie del container)
- Aggiorna l'applicazione senza interrompere il servizio (rolling update)
- Permette di definire tutto con file YAML, versionabili e gestibili in Git

Vantaggi principali

- **Alta disponibilità:** se un nodo si guasta, i Pod vengono ricreati su altri nodi.
- **Scalabilità:** automatica in base al carico o manuale.
- **Aggiornamenti sicuri:** rollout e rollback gestiti dal sistema.
- **Portabilità:** stesso cluster può girare su bare metal, cloud, virtual machine.
- **Gestione dichiarativa:** l'infrastruttura è descritta in file YAML, come codice (Infrastructure as Code).

Esempio di file YAML

yaml

CopiaModifica

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

Questo deployment crea 3 repliche di un container basato su `nginx`.

Dove viene usato Kubernetes?

Kubernetes è usato in quasi tutti i grandi ambienti cloud e aziendali:

- Google Cloud (GKE)
- Amazon (EKS)
- Microsoft Azure (AKS)
- OVH, Oracle Cloud, IBM Cloud
- Infrastrutture on-premise

È diventato lo **standard de facto** per il deployment e la gestione di microservizi e architetture cloud-native.

Conclusione

Kubernetes è un potente sistema di orchestrazione dei container che consente di gestire applicazioni moderne in modo affidabile, scalabile e automatico. Anche se inizialmente può sembrare complesso, conoscere Kubernetes è oggi **fondamentale per ogni sviluppatore backend, devops o cloud engineer**.

Inizia familiarizzando con i concetti di base (Pod, Deployment, Service) e poi prova a creare un piccolo cluster in locale (es. con Minikube o Kind). Pian piano potrai capire a fondo come funziona tutto il sistema e come sfruttarlo al massimo.