

kubernetes presenta diversi vantaggi e feature che facilitano molto il lavoro, come :

1. L'orchestrazione automatica dei container

- Kubernetes gestisce il ciclo di vita dei container (avvio, arresto, riavvio).

2. La scalabilità automatica

- permette di scalare orizzontalmente (più repliche) le applicazioni automaticamente in base al carico (CPU, memoria, ecc.)

3. Bilanciamento del carico (Load Balancing)

- Kubernetes distribuisce automaticamente il traffico tra i vari container/pod.

4. Alta disponibilità (High Availability)

- Se un container o un nodo fallisce, Kubernetes lo sostituisce automaticamente.

5. Self-healing (Auto-riparazione)

- Kubernetes riavvia i container che falliscono, li sostituisce, inoltre elimina temporaneamente quelli che non rispondono, e non li esporrà fino a che non sono pronti.

6. Aggiornamenti senza downtime (Rolling Updates & Rollback)

- permette di aggiornare le applicazioni senza causare interruzioni del servizio.

nonostante presenti molti vantaggi presenta anche ulteriori svantaggi, i quali possono essere :

1. Curva di apprendimento ripida

- Kubernetes **non è semplice**: anche concetti base come Pod , Service , Ingress , ConfigMap , Deployment richiedono tempo per essere compresi. essendo necessario capire anche Docker, reti, volumi, sicurezza, YAML, ecc.
-

2. Configurazione complessa

- Anche un'app semplice richiede **molti manifesti** (file `.yaml`) e risorse da configurare (Pod, Service, Namespace, ecc.).
-

3. Overhead infrastrutturale

- Kubernetes introduce **una complessità architetturale** che potrebbe non essere giustificata per progetti piccoli. Richiedendo **più risorse hardware** per eseguire il cluster (master, worker, ecc.).
-

4. Difficoltà di debug

- Il **debug** su Kubernetes può essere difficile in quanto:
 - bisogna capire cosa sta succedendo tra pod, servizi, ingress, DNS interni.
 - la diagnostica è distribuita: occorre usare `kubectl` , log, eventi, metriche.
-

5. Gestione dello storage persistente

- I volumi e lo storage **non sono banali** da configurare, specialmente su cloud o ambienti ibridi.
 - Serve scegliere e integrare **StorageClass**, volumi PVC/PV, e spesso provider esterni.
-

6. Richiede DevOps esperti

- Kubernetes è uno strumento potente **pensato per team DevOps o SRE esperti**.
- Per farlo girare in produzione servono competenze elevate come:
 - networking
 - sicurezza
 - CI/CD

- monitoring/logging (Prometheus, Grafana, etc.)