Sarthak Kaushik(101471600)

What K8s doesn’t do?

Kubernetes is a container orchestration engine for managing massively scalable applications and workloads.

K8s is not for/doesn’t do the following:

Application Agnostic: It can run any app given it is containerized, if an app can run in a container, it can run inside K8s. It does not impose restrictions as to what apps can run inside of K8s.

No Code Deployment or Build: Kubernetes does not handle the deployment of source code or building applications. CI/CD workflows are determined by organisation preferences and technical requirements.

No Application-level Services: Kubernetes doesn't provide application-level services like middleware, data-processing frameworks, databases, caches, or cluster storage systems as built-in services. These components can run on Kubernetes or be accessed through mechanisms like the Open Service Broker.

No Prescribed Logging or Monitoring: Kubernetes doesn't dictate logging, monitoring, or alerting solutions. It offers integrations as proof of concept and mechanisms to collect and export metrics.

No Mandated Configuration Language/System: Kubernetes doesn't enforce a specific configuration language or system. It provides a declarative API that can be targeted by various forms of declarative specifications.

No Built-in Machine Configuration or Maintenance Systems: Kubernetes doesn't provide comprehensive machine configuration, maintenance, management, or self-healing systems.

Not Just an Orchestration System: Kubernetes goes beyond traditional orchestration systems. It eliminates the need for explicit orchestration by providing a set of independent, composable control processes. Instead of a defined workflow (do A, then B, then C), it continuously drives the current state toward the desired state. Centralised control is not required, making it more user-friendly, powerful, robust, resilient, and extensible.

What other Orchestration tools are available other than Kubernetes?

Some other orchestration tools and their benefits are mentioned:

Docker Swarm:
Simple orchestration tool that comes bundled with Docker.
Good for smaller-scale deployments and projects.

Apache Mesos:

Efficient orchestration platform for managing distributed systems.
Scales well for large, complex applications.

Nomad (HashiCorp):

Lightweight and easy-to-use orchestrator.
Suitable for simpler deployment scenarios.

Amazon ECS(proprietary):

Amazon's container orchestration service.
Well-integrated with the AWS ecosystem.

OpenShift (Red Hat):

Kubernetes-based platform with additional features.
Streamlined for enterprise applications.

Docker Compose:

Simplifies the orchestration of multi-container Docker applications.
Great for local development and testing.

Rancher:

Provides a user-friendly interface for orchestrating containers.
Works well with various orchestration engines.

SwarmKit:

Native orchestration tool for Docker Swarm.
Part of the Docker toolkit.

Portainer:

Lightweight container management GUI.
Supports various orchestrators, including Docker Swarm.

Mesosphere DC/OS(proprietary):

Datacenter Operating System for managing containerized applications.
Offers simplified deployment and scaling.