

## What is Kubernetes?

Kubernetes is a portable, extensible open-source platform for managing containerized workloads and services, that facilitates both declarative configuration and automation. It has a large, rapidly growing ecosystem. Kubernetes services, support, and tools are widely available

## Why do I need Kubernetes and what can it do?

Kubernetes has a number of features. It can be thought of as:

- a container platform
- a microservices platform
- a portable cloud platform and a lot more.

Kubernetes provides a **container-centric** management environment. It orchestrates computing, networking, and storage infrastructure on behalf of user workloads. This provides much of the simplicity of Platform as a Service (PaaS) with the flexibility of Infrastructure as a Service (IaaS), and enables portability across infrastructure providers.

## How is Kubernetes a platform?

Even though Kubernetes provides a lot of functionality, there are always new scenarios that would benefit from new features. Application-specific workflows can be streamlined to accelerate developer velocity. Ad hoc orchestration that is acceptable initially often requires robust automation at scale. This is why Kubernetes was also designed to serve as a platform for building an ecosystem of components and tools to make it easier to deploy, scale, and manage applications.

## Summary of benefits:

- **Agile application creation and deployment:** Increased ease and efficiency of container image creation compared to VM image use.
- **Continuous development, integration, and deployment:** Provides for reliable and frequent container image build and deployment with quick and easy rollbacks (due to image immutability).
- **Dev and Ops separation of concerns:** Create application container images at build/release time rather than deployment time, thereby decoupling applications from infrastructure.
- **Observability** Not only surfaces OS-level information and metrics, but also application health and other signals.
- **Environmental consistency across development, testing, and production:** Runs the same on a laptop as it does in the cloud.
- **Cloud and OS distribution portability:** Runs on Ubuntu, RHEL, CoreOS, on-prem, Google Kubernetes Engine, and anywhere else.
- **Application-centric management:** Raises the level of abstraction from running an OS on virtual hardware to run an application on an OS using logical resources.
- **Loosely coupled, distributed, elastic, liberated [micro-services](#):** Applications are broken into smaller, independent pieces and can be deployed and managed dynamically – not a fat monolithic stack running on one big single-purpose machine.
- **Resource isolation:** Predictable application performance.
- **Resource utilization:** High efficiency and density.