Kubernetes vs. Docker: What Does It Really Mean?

“Kubernetes vs. Docker” is a phrase that you hear more and more these days as Kubernetes becomes ever more popular as a container orchestration solution.

However, “Kubernetes vs. Docker” is also a somewhat misleading phrase. When you break it down, these words don’t mean what many people intend them to mean, because Docker and Kubernetes aren’t direct competitors.

This post aims to clear up some common confusion surrounding Kubernetes and Docker, and explain what people really mean when they talk about “Docker vs. Kubernetes.”

The Rise of Containerization and Docker

Containers and container platforms provide a lot more advantages over traditional virtualization. Isolation is done on the kernel level without the need for a guest operating system, so containers are much more efficient, fast, and lightweight. Allowing for applications to become encapsulated in self-contained environments comes with a slew of advantages, such as quicker deployments, scalability, and closer parity between development environments.

Docker is currently the most popular container platform. Although the idea of isolating environments dates quite far back, and there has been other container software in the past, Docker appeared on the market at the right time, and was open source from the beginning, which likely led to its current market domination.

Docker features the Docker Engine, which is a runtime and allows you to build and run containers, and includes Docker Hub, a service for storing and sharing images.

The Need for Orchestration Systems

While Docker provided an open standard for packaging and distributing containerized applications, there arose a new problem. How would all of these containers be coordinated and scheduled? How do all the different containers in your application communicate with each other? How can container instances be scaled?

Solutions for orchestrating containers soon emerged. Kubernetes, Mesos, and Docker Swarm are some of the more popular options for providing an abstraction to make a cluster of machines behave like one big machine, which is vital in a large-scale environment.

When most people talk about “Kubernetes vs. Docker,” what they really mean is “Kubernetes vs. Docker Swarm.” The latter is Docker’s own native clustering solution for Docker containers, which has the advantage of being tightly integrated into the ecosystem of Docker, and uses its own API. Like most schedulers, Docker Swarm provides a way to administer a large number of containers spread across clusters of servers. Its filtering and scheduling system enables the selection of optimal nodes in a cluster to deploy containers.

Kubernetes is the container orchestrator that was developed at Google which has been donated to the CNCF and is now open source. It has the advantage of leveraging Google’s years of expertise in container management. It is a comprehensive system for automating deployment, scheduling and scaling of containerized applications, and supports many containerization tools such as Docker.

For now, Kubernetes is the market leader and the standardized means of orchestrating containers and deploying distributed applications. Kubernetes can be run on a public cloud service or on-premises, is highly modular, open source, and has a vibrant community. Companies of all sizes are investing into it, and many cloud providers offer Kubernetes as a service. Sumo Logic provides support for all orchestration technologies, including Kubernetes-powered applications.

How Does Kubernetes Relate to Docker?

Kubernetes and Docker are both comprehensive de-facto solutions to intelligently manage containerized applications and provide powerful capabilities, and from this some confusion has emerged. “Kubernetes” is now sometimes used as a shorthand for an entire container environment based on Kubernetes. In reality, they are not directly comparable, have different roots, and solve for different things.

Docker is a platform and tool for building, distributing, and running Docker containers. It offers its own native clustering tool that can be used to orchestrate and schedule containers on machine clusters. Kubernetes is a container orchestration system for Docker containers that is more extensive than Docker Swarm and is meant to coordinate clusters of nodes at scale in production in an efficient manner. It works around the concept of pods, which are scheduling units (and can contain one or more containers) in the Kubernetes ecosystem, and they are distributed among nodes to provide high availability. One can easily run a Docker build on a Kubernetes cluster, but Kubernetes itself is not a complete solution and is meant to include custom plugins.

Kubernetes and Docker are both fundamentally different technologies but they work very well together, and both facilitate the management and deployment of containers in a distributed architecture.

Additional Resources

- What is Kubernetes?
- Monitoring k8s-powered Apps