KodeKloud

Docker

Kubernetes vs. Docker Swarm: A Comprehensive Comparison (2023)

Containers make application deployment and scaling easy and fast, and that makes them a perfect fit for modern applications. To get the most out of containerization, you need a good container orchestration tool. Currently, two of the most popular orchestration tools are **Kubernetes** and **Docker Swarm**.

In this article, we look at each of these tools. We shall then examine the similarities and differences and discuss when to use each.

What is Kubernetes?

Kubernetes is an open-source container orchestration tool that automates the deployment, scaling, and management of containerized applications. It was originally developed by Google and is now maintained by the Cloud Native Computing Foundation (CNCF).

Kubernetes allows users to easily manage and deploy containerized applications across a cluster of nodes. It provides features such as load balancing, storage orchestration, automated rollouts and rollbacks, and self-healing of containers, making it an ideal tool for m

Subscribe

containerized workloads at scale.

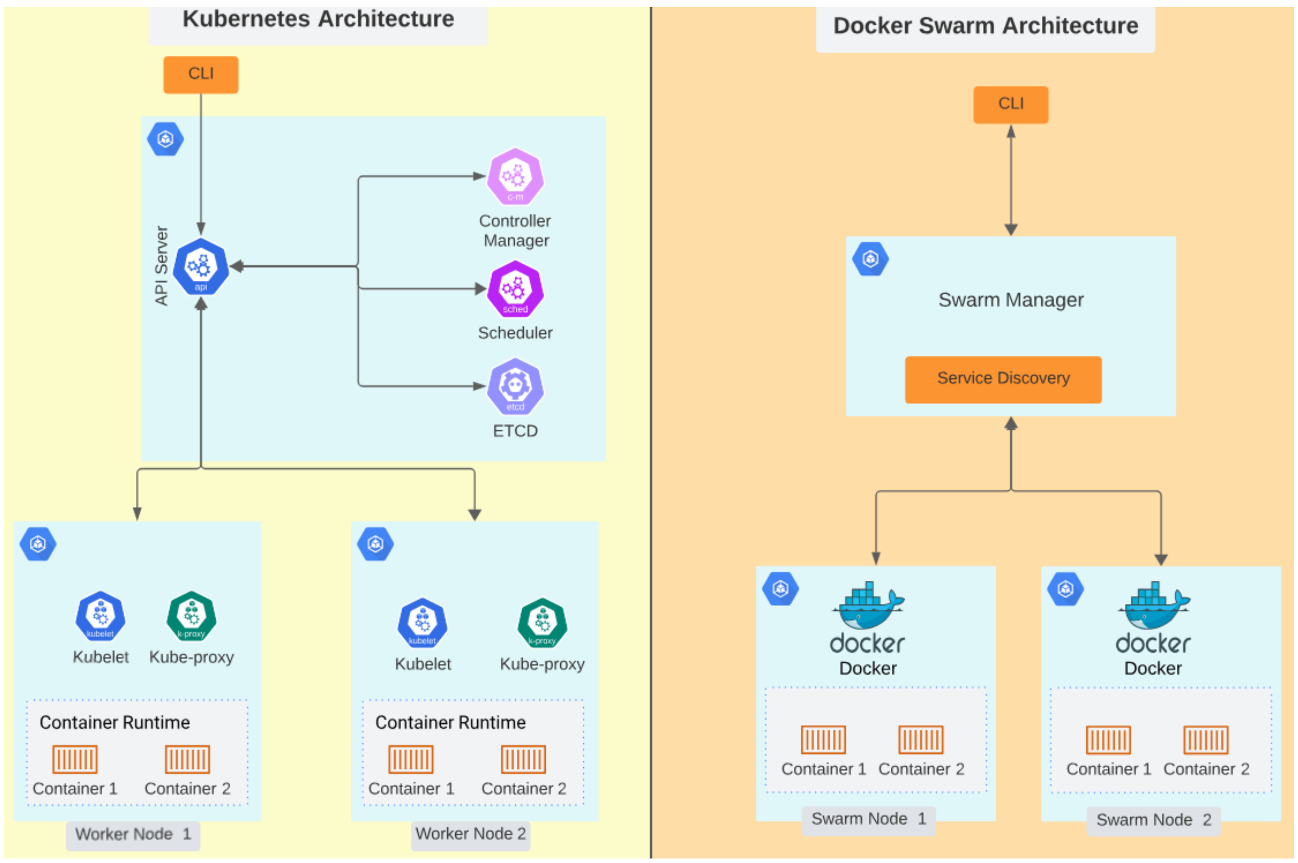
To learn more about how Kubernetes works, check out this blog: Kubernetes Concepts Explained!

What is Docker Swarm?

Docker Swarm is an open-source container orchestration tool that allows users to manage and deploy Docker containers with ease. Docker Swarm provides features such as load balancing, service discovery, and rolling updates, among others. It is the native clustering and scheduling tool for Docker containers.

Kubernetes vs. Docker Swarm: Architecture Comparison

Below is the simplified architecture of two tools:



Kubernetes vs. Docker Swarm

From the diagram above, you can see that Kubernetes has more components. On the other hand, you can see the simplicity of Docker Swarm. Let's now look at their similarities.

Similarities Between Kubernetes and Docker Swarm

Below are some similarities between Docker Swarm and Kubernetes:

I. Open-Source

Both tools are open source. Kubernetes is maintained by the CNCF and can be freely used, modified, and distributed under the Apache 2.0 license. Docker Swarm is part of the Docker ecosystem and can be freely used, modified, and distributed under the Apache 2.0 license.

II. Community

Kubernetes and Docker Swarm have large and active communities, with thousands of manifests available for common tasks. The communities provide learning materials, test projects, and answers to users' questions.

Differences Between Kubernetes and Docker Swarm

Below are some of the differences between Kubernetes and Docker Swarm:

I. Autoscaling

Kubernetes autoscaling feature allows for the automatic adjustment of the number of replicas in a deployment or replication controller based on the resource utilization of the running pods. Autoscaling in Kubernetes can be done both horizontally and vertically, depending on the type of resource being scaled. Horizontal autoscaling adds or removes replicas of the same pod, while vertical autoscaling changes the resource allocation of a single pod.

See Vertical Pod Autoscaler (VPA) in Kubernetes Explained through an Example.

Unlike Kubernetes, Docker Swarm does not have an autoscaler. It uses commands to implement scaling.

II. Load Balancing

Docker Swarm provides automatic load balancing that handles the routing and distribution of traffic.

When using Kubernetes, you have to configure load balancing manually. Both tools support multiple load balancing strategies, such as round-robin, session-based, and IP-based load balancing.

III. Monitoring

Kubernetes provides a comprehensive set of monitoring tools and metrics that allow you to keep a close eye on your cluster's health and performance. Additionally, it has a larger and more active community that contributes to the development of monitoring solutions.

Docker Swarm, on the other hand, provides only basic metrics for containers and nodes, which can make it difficult to get a complete picture of your cluster's health and performance. Secondly, unlike Kubernetes, it does not have many third-party monitoring and logging integrations.

IV. Learning Curve

The learning curve for Kubernetes can be steep, especially for those new to container orchestration and cloud-native technologies. That’s because it has a large set of commands that one needs to understand to effectively manage and secure a cluster.

Docker Swarm is easier to learn than Kubernetes due to its simpler architecture and fewer features. It also has fewer commands for managing and securing a cluster.

V. Orchestration

Docker Swarm uses a manager-worker architecture to create a cluster of nodes that can run containers in a distributed manner. The manager nodes are responsible for managing the cluster, while the worker nodes are responsible for running the containers.

Kubernetes also uses a master-worker architecture. However, unlike Docker Swarm, a cluster of nodes is managed by a central Kubernetes master. The master is responsible for scheduling containers onto the worker nodes and monitoring and managing the state of the cluster.

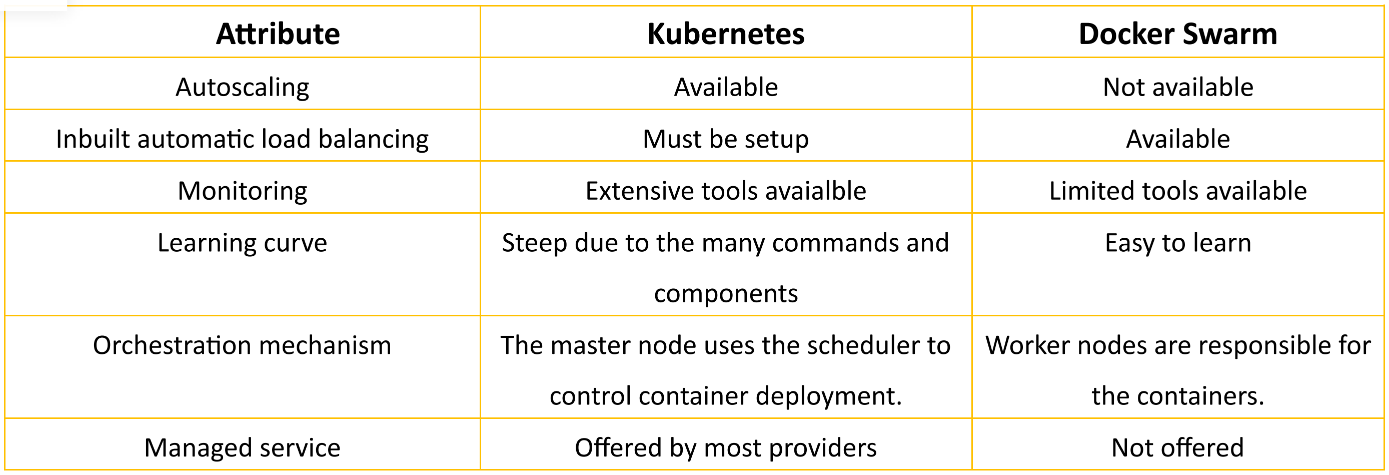
VI. Managed Service

Most cloud service providers offer managed Kubernetes services. A managed Kubernetes service is a cloud-based offering that provides a managed environment for deploying, scaling, and managing containerized applications on Kubernetes. It typically includes features such as automated upgrades, monitoring, logging, and security, which make it easier for organizations to focus on developing their applications rather than managing the underlying infrastructure.

Examples of managed Kubernetes services include Amazon Elastic Kubernetes Service (EKS), Microsoft Azure Kubernetes Service (AKS), and Google Kubernetes Engine (GKE).

Unlike Kubernetes, popular cloud providers do not offer a managed Docker Swarm service. This means that when you deploy a Swarm cluster, you have to also actively manage the underlying infrastructure.

Below is a summary of the differences between the two tools:



Attribute	Kubernetes	Docker Swarm
Autoscaling	Available	Not available
Inbuilt automatic load balancing	Must be setup	Available
Monitoring	Extensive tools avaialble	Limited tools available
Learning curve	Steep due to the many commands and components	Easy to learn
Orchestration mechanism	The master node uses the scheduler to control container deployment.	Worker nodes are responsible for the containers.
Managed service	Offered by most providers	Not offered

Both Kubernetes and Docker Swarm are great tools for container orchestration and have their respective strengths.

When to use Kubernetes

Kubernetes is a more mature and feature-rich platform with a vast ecosystem and a vibrant community. It offers more advanced features such as auto-scaling and advanced monitoring. It is also highly customizable and can be easily integrated with other tools and platforms.

When to use Docker Swarm

Docker Swarm, on the other hand, is simpler to set up and use, making it a good choice for smaller projects or teams that don't require the complexity of Kubernetes. It is also a good choice for those who are already familiar with Docker and prefer a single tool for container management.

Ultimately, the choice between Kubernetes and Docker Swarm depends on the specific needs of the project and the level of expertise of the team.

Conclusion

This article compares two popular container orchestration tools, Kubernetes and Docker Swarm. Both tools are open-source and have active communities, but there are differences in their features.

Kubernetes is a powerful tool that simplifies the management of complex clusters, making it an ideal tool for managing containerized workloads at scale. Docker Swarm is easier to learn and set up than Kubernetes due to its simpler architecture. However, it lacks features such as autoscaling and comprehensive monitoring tools, making it suitable for smaller projects.

Ultimately, the choice of tool depends on the specific needs of the user.

Check out the following Kubernetes course from KodeKloud:

Kubernetes for the Absolute Beginners – Hands-on Tutorial | KodeKloud

Learn Kubernetes with simple, easy lectures and hands-on labs

KodeKloud logo

You can also watch our whole video on YouTube here:

KodeKloud

Jan 5, 2024 • 5 min read

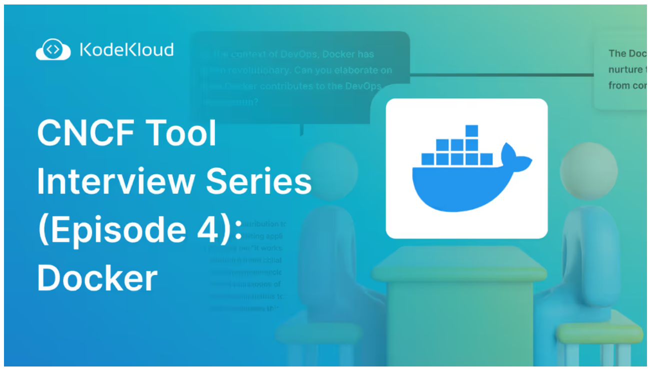
Recommended

DevOps

10 Essential DevOps Tools You Should Learn in 2024

The DevOps landscape is constantly evolving, with new tools and technologies emerging at ...

May 29, 2024 — 13 min read



cncf

CNCF Tool Interview Series(Episode 04): Docker

The Interview Interviewer: Welcome back to our CNCF Tool Interview Series Episode 04, wher...

Apr 12, 2024 — 5 min read

LEARNING PATHS

DevOps

Kubernetes

Docker

COURSES

Certified Kubernetes Administrator

Certified Kubernetes Application Developer

Certified Kubernetes

COMMUNITY

Join our community

Teach with Us

Write with Us

ABOUT

About Us

Success Stories

Our Values

Careers at KodeKloud

HELP

Contact Us

Support

Give us feedback

YOUR ACCOUNT

Sign In

Register