Docker vs Kubernetes vs Openshift: A Comparison

Docker

**What is it:**[Docker](https://www.docker.com/) first arrived on the market in 2013 and quickly became one of the most widely used containerization applications in software development. Docker enables developers to build, test, deploy, and manage containerized applications in open-source and commercially available versions.

**Pros:**

* **Easy to use:**Installing Docker Desktop is a breeze and gives you everything you need to start with the command line interface (CLI) in just a few minutes. Docker APIs provide simplified access and control of the daemon, enabling commands or scripts to automate containerization. Docker also provides software development kits (SDKs) for programming languages like Go or Python, further facilitating container use.
* **Easy to share:** Docker operates through the use of Dockerfiles, which define the configurations necessary to run the image file housed within the Docker container. Developers can create a single Dockerfile with the required configuration, build an image with the application, and then push this image to the cloud. That way, developers can ensure the application runs as intended within a Docker container, no matter which device they’re running on.
* **Open source and enterprise options:**Whether you want to create and manage a few Docker containers or you’re looking for official, enterprise-level support, Docker has options available to fit your needs.
* **Additional software available:**Docker provides a suite of products to facilitate development. Docker Scout scans container images for potential vulnerabilities, Docker Buildx delivers various image-building tools, and Docker Extensions allows for integrating third-party tools, such as Kubernetes.

**Cons:**

* **Difficult to scale:**While Docker includes a swarm mode for managing clusters of containers, it does not offer the same level of automation and orchestration capabilities as Kubernetes, making it challenging to manage large volumes of containers without other tools. This scalability is one of the most significant considerations when comparing Docker orchestration vs Kubernetes.

**Best used for:**

Docker is best for building and sharing single or small groups of containers. Its versatility and end-to-end containerization development tools make it worthwhile for startups and enterprises. However, organizations looking for deeper orchestration management will likely need to integrate other tools into their workflow.

One such tool Docker pairs well with is DuploCloud, a DevOps Automation platform that helps to automate container configuration and orchestration. It fills in Docker’s gaps and, when paired with Kubernetes, transforms into a powerful containerization platform that can help maintain security and compliance at scale.

Kubernetes

**What is it:**[Kubernetes](https://kubernetes.io/) (also known as K8s) is an open-source container orchestration system. Initially developed by Google to automate deployment and resource scaling within its own network of servers and applications, the company made Kubernetes open source in 2014 and it is now managed by the Cloud Native Computing Foundation. Numerous organizations, including Box, IBM, Spotify, and Squarespace, use Kubernetes to manage their containers.

**Pros:**

* **Highly scalable:**Kubernetes uses horizontal scaling to manage resource loads. Instead of increasing the memory or compute power assigned to already-existing workloads (through vertical scaling), Kubernetes deploys additional Pods (i.e., additional copies of container clusters) to ensure operations continue running smoothly. Kubernetes also automates this horizontal scaling to ensure maximum efficiency. The sheer volume of containers Kubernetes can manage is unmatched when comparing Kubernetes vs Docker.
* **Self-healing and rollback:**If containers fail or don’t respond to health checks, Kubernetes will automatically close and restart containers for you.
* **Portable and uniform:**Kubernetes can run in a variety of environments, from public and private cloud environments within AWS, Google Cloud, or Microsoft Azure, to on-premises infrastructure, and even hybrid environments. It can also provide a uniform way to build and deploy containers across these environments en masse.
* **Open-source:**Development is fully open-source and transparent, and a large community provides extensive support for the platform.

**Cons:**

* **Highly complex:**Setting up Kubernetes environments is extremely difficult and requires extensive knowledge of numerous components. Teams must know how to build and run the control plane, which manages Kubernetes container pods. Setting these pods up in a way that ensures maximum efficiency requires an investment in specialized team members or Kubernetes management services provided by the major cloud platforms — both of which can incur high costs.
* **Resource intensive:**Kubernetes’ robust scaling and redundancy capabilities require far more resources than standard containerization applications, which requires a more significant investment in cloud networking components.

**Best used for:**

Because of its complexity and high cost of operation, Kubernetes is best for large organizations that need extensive container orchestration and scaling capabilities — typically cloud-native applications that need to ensure speed and stability on a global scale. If your business is just starting with the cloud and isn’t sure whether it has the resources necessary to get the most out of Kubernetes, other orchestration tools will likely be a better fit.

OpenShift

**What is it:**[OpenShift](https://www.redhat.com/en/technologies/cloud-computing/openshift) is a suite of containerization applications built on Red Hat Linux platforms and powered by Kubernetes. Designed initially around custom container technologies in 2011, Red Hat transitioned the platform to a Docker/Kubernetes base in 2015.

**Pros:**

* **Simplified Kubernetes platform:**Kubernetes can be overwhelming for many organizations just starting to transition to cloud development. OpenShift helps to simplify Kubernetes development through a suite of cluster management and containerization programs managed through a central ecosystem.
* **Automated security features:**Upgrading security in Kubernetes and Docker requires manual integration of tools. Because OpenShift is a full subscription-based containerization suite, it handles security and compliance coverage for you.

**Cons:**

* **Subscription-based service:**While RedHat offers a free trial for up to 60 days, the trial provides limited access to important CloudShift features. Unlocking the full benefits of the platform (along with upgraded access to modern Kubernetes updates and customer support) requires subscribing to a pay-as-you-go plan.
* **Exists downstream of Kubernetes:**OpenShift relies on Kubernetes as its orchestration platform, so OpenShift platform updates typically fall behind Kubernetes’ update schedule — leading to a potential gap between OpenShift vs Kubernetes.
* **Onboarding can be more difficult:**While Kubernetes is difficult to use, it’s a widely available free and open platform, making its use common among experienced DevOps engineers. OpenShift is less common, and its unique approach to Kubernetes makes finding and onboarding OpenShift-specific developers a potentially more difficult task.

**Best used for:**

OpenShift is suited for organizations already comfortable with Red Hat products that are looking to expand their reach into containerized development and orchestration.