

[PLATFORM](#)[PRICING](#)[SOLUTIONS ▾](#)[RESOURCES ▾](#)[BLOG](#)[GET A DEMO](#)



APP DEPLOYMENT

Docker Advantages and Disadvantages: What You Need to Know Before You Switch

Docker's broad compatibility, flexibility, and scalability are appealing, but there are also some downsides developers should consider first

Containerization promises to increase efficiency, improve compatibility, and offer flexibility at scale. But how can developers know which container-based software is right for their needs? Docker is one of the most popular containerized development tools on the market today, but what are the advantages and disadvantages of Docker? Would another platform be a better fit? In this article, we'll explore the Docker advantages and disadvantages that developers need to consider before onboarding the platform or making a switch to new tech.

Jump to a section...

[Docker Advantages and Disadvantages at a Glance](#)

[Advantages of Docker Containers](#)

[Cross-platform Consistency](#)

[Serverless Storage](#)

[High Speed Build & Deployment](#)

[Flexibility & Scalability](#)

[Disadvantages of Docker](#)

[Outdated Documentation](#)

[Steep Learning Curve](#)

[Security Issues](#)

[Limited Orchestration](#)

[DuploCloud Makes Container Configuration Easy](#)

Docker Advantages and Disadvantages at a Glance

Docker is an open source software platform designed to help developers build, test, deploy, and run containerized applications. When Docker originally launched in 2008 (evolving out of its origins as dotCloud), it combined the convenience of a command line interface with a server daemon, library of pre-built images, and registry server to make it easy to build and share layered containers. The container approach means that multiple application environments can exist isolated from one another within a host system.

Let's take a quick look at the advantages and disadvantages of Docker before we dive deeper:

Docker Pros	Docker Cons
<small>Cross-platform consistency: Compatibility across a range of systems and architectures.</small>	<small>Outdated documentation: Docker's extensive documentation doesn't always reflect the latest features and best practices.</small>

active memory to run reliably.	find Docker easy to begin but hard to master.
High-speed deployment: Eliminating redundant installations and configurations makes deployment fast and easy.	Security issues: The lack of segmentation means that multiple containers can be vulnerable to host system attacks.
Flexibility and scalability: Developers can use any programming language and scale container resources up and down as needed.	Limited orchestration: It can be difficult to effectively manage multiple containers at once without orchestration capabilities.

Advantages of Docker Containers

Docker is one of the most popular containerized application development platforms for a reason. It may not be the best fit for every development team or organization, but its strengths have made it a favorite in the space. Here are the primary Docker advantages you should know:

Cross-platform Consistency

The fact that Docker works across multiple environments, platforms, and operating systems helps ensure consistency across development teams. Engineers get to work in a consistent and unified way instead of bouncing back and forth between servers or machines. Docker's cross-platform compatibility eliminates the risk of deployment panic when dealing with Mac, Windows, and Linux deployments simultaneously; when images transform into containers through Docker, they deploy automatically and remain stable on any system or device.

Serverless Storage

Docker containers eliminate the need for sprawling server farms built to contain virtual machines. Because virtual machines split each server into several environments, they require both extensive server space and active memory. By contrast, Docker containers can run completely in the cloud, without installing an operating system on each one. Containers also make the most of memory and storage space by reusing components in virtualized images.

High Speed Build & Deployment

One of the most important advantages of Docker containers is that it's easy to deploy them quickly. The ability to schedule and automate specific actions across multiple environments takes some of those mundane, repetitive tasks off of developers' plates. Docker containers can also have multiple layers representing a log of changes to the image base, which speeds up the build phase and enables version control and CI/CD support. And without having to install or configure operating systems each time, developers can deploy applications in just a few seconds or with just [a few lines of code](#).

Flexibility & Scalability

In order to fulfill its mission of making developers more efficient, Docker prioritizes flexibility and scalability. Developers can use any programming language or library on a server, and can easily scale up or scale down resources depending on the specific needs of a given container. In addition to that vertical scaling, horizontal scaling is also easy with Docker. Developers can deploy multiple containers within a unified overlay network in order to load balance and distribute performance.

Getting the most out of Docker's scalability features requires transitioning your on-premise infrastructure to the cloud. DuploCloud can help you get there. [Download a free copy of our Essential Cloud Migration Checklist](#), and follow our step-by-step guide to plan and execute a seamless migration.

G U I D E

The Essential Cloud Migration Checklist

DOWNLOAD NOW



Disadvantages of Docker

Outdated Documentation

The open-source culture behind Docker helps ensure that the software is constantly evolving. While that rapid-fire pace of change is positive in most respects, it can mean that the community can sometimes get ahead of themselves. Docker is known for its [expansive documentation](#) library, but new documentation can't always keep up with the pace of new releases and updates to the software. When developers need answers about changes in Docker, they can sometimes be hard or even impossible to find until the relevant documentation is ready.

Steep Learning Curve

Even for developers who are familiar with virtual machines and containerized infrastructure, switching to Docker can sometimes be a difficult task. Learning the basics isn't necessarily out of reach, but becoming proficient or achieving excellence with Docker often requires a lot of dedicated time and effort. [Docker Extensions](#) and other additional tools that Docker supports are helpful in many ways, but they also make the software even more complex to learn. And as with Docker documentation, the constant pace of updates can make it hard to stay on top of platform mastery.

Security Issues

While one of the main advantages of Docker containers is that they are lightweight and don't require tons of resources, sharing a common operating system also introduces security issues. Isolation or segmentation are important principles in modern network architecture, specifically to prevent the risk of several containers or environments being impacted at the same time when an attacker breaches the host system. So, while virtual machines require server space and more memory to run, the fact that each one uses its own operating system supports a stronger security posture. It's possible to combat these security issues with containers by integrating them into existing infrastructures and inheriting their security standards, but that introduces even further complexity.

Limited Orchestration

Although Docker does offer some automation features, its capabilities for automation and orchestration are not as robust as other containerized platforms like [Kubernetes](#). Without extensive orchestration, it can be difficult to manage multiple containers and environments at the same time. DevOps teams rely on orchestration to be effective, so using Docker would necessitate third-party or external tools.

DuploCloud Makes Container Configuration Easy

In addition to Kubernetes, a containerized application tool focused specifically on automation and orchestration, there are other alternatives to Docker worth considering. [OpenShift](#) is Red Hat's suite of containerized application tools (and it also happens to be built on Kubernetes). [Podman](#) is a daemon-less container platform, while [LXD](#) is an application engine designed specifically for Linux containers.

At the end of the day, the right containerization platform for your team will depend entirely on your existing environments, preferred programming languages, and development needs. For DevOps professionals, Docker remains a popular choice thanks to its CI/CD capabilities, broad compatibility, and speed and efficiency in every phase from build to testing to deployment. No matter what containerization platform you choose, manually configuring containers cancels out some of the most important benefits of the tech.

Instead of time-consuming, error-prone, and hard-to-scale manual configuration, use DuploCloud's DevOps Automation Platform. It streamlines crucial aspects of the DevOps pipeline, including container configuration and orchestration, and provides 24/7 monitoring and reporting functionality to keep all your cloud infrastructure secure and compliant. And with a no-code/low-code approach, the DevOps Automation Platform speeds up deployment times by 10x. [Contact us today](#) for a free demo to learn more.



← PREV POST

NEXT POST →

Related Posts



Docker Container Deployment: The Complete Checklist for DevOps Leaders

Disadvantages: What You Need to Know Before You Switch

to Run Containers at Scale with Docker Swarm

Practices Every DevOps Leader Needs to Know

Search... >

Category

[App Deployment \(19\)](#)

[Audit & Reporting \(2\)](#)

[CI/CD \(3\)](#)

[Cloud \(44\)](#)

[Cloud Services \(56\)](#)

[Compliance \(57\)](#)

[DevOps \(69\)](#)

[DevSecOps \(21\)](#)

[Healthcare \(17\)](#)

[Migration \(17\)](#)

[Platform Engineering \(11\)](#)

[Security \(18\)](#)

Recent Posts

No posts were found for provided query parameters.

Tag

alerting, api, automation, aws, azure, backups, compliance reports, conferences, cto, database, data migration, deploy, edge deployments, elasticsearch, gcp, gdpr, hipaa, hitrust, iaas, infrastructure-as-code, internal developer platform, iso-27001, kubernetes, lambda, leadership, linux, logging, low-code, microservices, no-code, on-premise, paas, pci-dss, private cloud, public cloud, serverless, smb, soa, soc 2, tracing, vpc

Follow us





Want to work with us?

USE THIS EMAIL

careers@duplocloud.net

Want to say hi?

GENERAL ENQUIRIES

info@duplocloud.net

Find us on social media.

SOCIAL MEDIA

**SOLUTIONS**

DevOps Automation

Out of the Box Compliance

SOC 2 Certification

Deploy on the Edge

Cloud Migration

Internal Developer Platform

DevOps-as-a-Service

AWS Solutions

RESOURCES

Documentation

Customer Stories

Videos

Webinars

White Papers

COMPANY

About us

Blog

Contact us

Careers

Events

News and Press

Privacy Policy

Cookie Policy

HEADQUARTERS

2150 N 1st St

San Jose, CA 95131

+1 (866) 830-6588

Copyright © 2023 DuploCloud, Inc.



Read our reviews on

