MR.PLANB FIELD GUIDE · CONTAINERS

Why Teams Move Away From Docker: Quick Guide

What is changing, what is not, and how to think about runtimes versus tooling.

Updated **2026-10-29** · Full guide: www.mrplanb.com/what-is/moving-away-from-docker

> Docker is still widely used, especially for local development. In production, many teams prefer Kubernetes-native runtimes and alternative desktop tooling, for reasons including Docker Desktop licensing for larger organizations, security preferences and standardization on runtimes like containerd and CRI-O. Containers stay central. What changes is the tooling used to build, run and orchestrate them.

3
reasons named: licensing, security preference, standardization

2
Kubernetes-native runtimes named: containerd and CRI-O

Dev
where Docker is still most widely used

Containers
still central, only the tooling changes

Short answer

- Docker remains widely used, especially for local development.
- In production, many teams now prefer Kubernetes-native runtimes.
- Teams also pick alternative desktop tooling.
- Containers are still central. The tooling around them keeps changing.

Why teams move

Licensing
Docker Desktop licensing for larger organizations.

Security preferences
Some teams prefer runtimes and tools with a different security approach.

Standardization
Operational standardization around runtimes like containerd and CRI-O.

Different interfaces
Teams keep containers and switch to different container interfaces and runtimes.

Docker versus the runtime layer

General background beyond the page's short answer.

Layer	Examples	Role
Developer tooling	Docker Desktop, Podman Desktop	Build and run containers locally
Container runtime	containerd, CRI-O	Run containers on Kubernetes nodes
Orchestrator	Kubernetes	Schedule and manage containers at scale
Image format	OCI images	Portable across tools and runtimes

Questions to ask before switching

- [ ] Is the pain on developer laptops (desktop tooling) or on production nodes (runtime)?
- [ ] Does Docker Desktop licensing affect your organization size?
- [ ] Does your Kubernetes platform already use containerd or CRI-O on nodes?
- [ ] Do your images and pipelines depend on Docker-specific features?
- [ ] Is standardizing on one runtime worth the retraining cost?

Sources and notes: Mr.PlanB What-is answer: Why are people moving away from Docker? · General background on containerd, CRI-O and OCI images