

# vsupalov

[Courses](#)[More Docker](#)[More Deployment](#)[About](#)[Start Here](#)[Articles](#)

## Is Docker-Compose Suited For Production?

Docker-compose is just a tool for handling and configuring Docker containers.

If you would be comfortable working with plain Docker commands in your environment, you can use docker-compose as well. The [docker-compose docs](#) even have a section about using docker-compose in production.

But let's look deeper than this easy answer. Why do some people recommend otherwise? I think it's important to understand why some people mean when they say "don't use docker-compose in production" and what assumptions are behind this recommendation.

### Hidden Assumptions

Production means [different things to different people](#). If somebody has a strong opinion whether something is "production ready" or "not suited for production", the best thing one can do is ask **why**.

Maybe, the person making the statement is of the strong opinion that any “production” environment needs to be distributed across multiple machines, and that containers should be distributed across those machines dynamically. Docker-compose is not built for this use case, just as Docker by itself wouldn’t be able to take care of this - it focuses on [other responsibilities](#). You’d need to use Docker Swarm, Kubernetes or another orchestrator to achieve this. But! You can have a perfectly fine production environment without falling back to these tools and without dynamically distributing containers across a varying number of machines.

Maybe the statement is motivated by the assumption that even a small amount of downtime in between deploys has to be avoided. If you are deploying in a simple fashion without putting work into it, your app will be unavailable for a bit, but depending on the application this is far from critical. As always, it’s a tradeoff to consider, what you are getting in return for keeping it simple is reduced complexity.

## Valid Use Cases

If you’re deploying to a single machine, and want to use docker-compose to build up your containers, you’re fine. Docker Compose is just a tool which reads the configuration file you pass to it and then talks to the Docker daemon according to those configs. As long as the containers are used on a single machine, you are fine.

If you need high availability, you can choose the right architecture to handle the load balancing and redundancy by having multiple identical machines behind a load balancer. This setup does not care about how each machine is provisioned, and what is running on them as long as the load balancer can pass traffic to them.

You can use docker-compose, plain Docker or more traditional ways of provisioning an environment. The workload on each machine exists isolated from its siblings, and you don't need to orchestrate containers across them.

## In Conclusion

Using docker-compose is fine if you are working on a single machine or don't need to distribute containers across multiple inter-connected machines. If you would be alright with just using Docker by itself, you can use docker-compose as well. It's a cool tool to make handling container configuration or multiple interconnected containers a bit easier.

The “don't use docker-compose in production” statement is motivated by hidden assumptions which are not necessarily valid for everybody, and propagated by unclear communication. After all, production means [different things to different people](#).

Whether you choose to use docker-compose or not - make sure that you're using Docker [carefully](#), your images are well-built, you're following best practices and that you're taking care of your servers.

If you want your containerized deployment to be distributed across multiple machines, you might want to look beyond docker-compose. Docker Swarm and Kubernetes are both orchestration solutions, you could also consider whether AWS ECS might be a good fit for you. If you just want to manage containers on single machines (even if there are multiple ones), docker-compose is a great tool.

**Get Better With Docker**

Subscribe to get useful Docker tips and tricks via email.

Send me Docker tips!

[\(About the content, privacy, analytics and revocation\).](#)

We won't send you spam. Unsubscribe at any time.

Powered By ConvertKit

## See Also

[Using Docker At Your Own Pace](#)

[Should You Run Your Database in Docker?](#)

[What's Wrong With The Docker :latest Tag?](#)

[Should Built Docker Images Be Used in a Development Environment?](#)

[Should Dockerfiles Be Used in a Production Environment?](#)

[Overriding One Single Value in Your Docker-Compose .env File](#)

© 2019 vsupalov.com. All rights reserved. | [Imprint](#) | [Privacy Policy](#) | [RSS](#) | [@vsupalov](#)