

- Getting Started
- Frequently Asked Questions
- Part 1 >
- Part 2 >
- Part 3 >
 - Introduction to Part 3
 - Official Images and trust
 - Deployment pipelines
 - Using a non-root user
 - Optimizing the image size
 - Multi-host environments**
 - End

Multi-host environments

Now that we've mastered containers in small systems with Docker Compose it's time to look beyond what the tools we practiced are capable of. In situations where we have more than a single host machine we cannot rely solely on Docker. However, Docker does contain other tools to help us with automatic deployment, scaling and management of dockerized applications.

In the scope of this course, we cannot go into how to use the tools in this section, but leaving them without a mention would be a disservice.

Docker swarm mode is built into Docker. It turns a pool of Docker hosts into a single virtual host. You can read the feature highlights [here](#). You can run right away with `docker swarm`. Docker swarm mode is the lightest way of utilizing multiple hosts.

Kubernetes is the de facto way of orchestrating your containers in large multi-host environments. The reason being it's customizability, large community and robust features. However, the drawback is the higher learning curve compared to Docker swarm mode. You can read their introduction [here](#).

It is always good to remember that a single tool is rarely an optimal solution for all possible scenarios. In a 2-3 host environment for a hobby project, the gains from Kubernetes might not be as large compared to an environment where you need to orchestrate hundreds of hosts with multiple containers each.

You can get to know Kubernetes with `k3s` a lightweight Kubernetes distribution that you can run inside containers with `k3d`. Another similar solution is `kind`. These are a great way to get started as you don't have to worry about complicated setup or any credit limits that the cloud providers always have.

Rather than maintaining one yourself the most common way to use Kubernetes is by using a managed service by a cloud provider. Such as Google Kubernetes Engine (GKE) or Amazon Elastic Kubernetes Service (Amazon EKS) which are both offering some credits to get started.

Exercise 3.11

EXERCISE 3.11: KUBERNETES

Familiarize yourself with Kubernetes terminology and draw a diagram describing what "parts" the Kubernetes contain and how those are related to each other.

You should draw a diagram of at least three host machines in a Kubernetes cluster. In the diagram assume that the cluster is running two applications. The applications can be anything you want. An example could be a video game server and a blog website.

You may take inspiration from the diagrams of part 2.

The applications may utilize other machines or APIs that are not part of the cluster. At least three of the machines should be utilized. Include "your own computer" in the diagram as the one sending instructions via `kubectl` to deploy an application. In addition, include a HTTP message coming from the internet to your Kubernetes cluster and how it may reach an application.

Make sure to label the diagram so that anyone else who has completed this exercise, and read the glossary, would understand it. The diagram should contain at least four of the following labels: Pod, Cluster, Container, Service and Volume.

Glossary. And some helpful diagrams

I prefer to use [draw.io](#) but you can use whichever tool you want.

If you are interested, later this year the course DevOps With Kubernetes will provide a comprehensive treatment on using Kubernetes.

Edit this page

Previous
« Optimizing the image size

Next
End »

Help

Discord [↗](#)
Report an issue

More

About
GitHub [↗](#)

In collaboration

University of Helsinki [↗](#)
Eficode [↗](#)
Unity [↗](#)