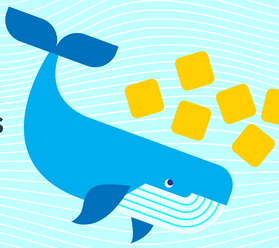

Chapter 4/Multi-Host-Environments

Multi-host environments

Now that we have 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 introduced 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 it 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. We have also the course DevOps with Kubernetes available until end of January 2026 and starting again in summer 2026!

Rather than maintaining a cluster 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. Kubernetes

TRIES 1 POINTS 0/1

After you submit this exercise, you will need to self review your own answer. Following that, course staff will review your submission before you earn any points.

Instructions

Familiarize yourself with Kubernetes terminology and draw a diagram describing what "parts" the Kubernetes contains 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 Chapter 3.

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, would understand it. The diagram should contain at least four of the following labels: Pod, Cluster, Container, Service, and Volume. I prefer to use draw.io for diagrams, but you can use whichever tool you want.

See the Kubernetes Glossary, and have a look to some helpful diagrams

As the answer submit the URL where the diagram can be found.

Answer

Word count:
0 words Min words: 0 Max words: 150

Submit

↑ Chapter front page

You've reached the end of this topic.

Proceed to the next topic

Next page:
Summary