**Docker and Kubernetes**

Docker and Kubernetes, 2 amazing technologies which make managing and deploying and also developing more complex applicationsway easier.

We are going to learn What Docker exactly is and what it is about?

What Exactly is Docker and why might we wanna use it?

What’s its advantage?

Docker is a container Technology. A tool for creating and managing containers.

What exactly does it mean?

What is a container in software development ? and why might we want to use it?

A container in a software development is a standardized unit of software.which basically means it is a package of code and that. That’s important dependencies and tools required to run that code.

The advantage is that , the same container with same code with the same software version, will give you the same exact behavior and same result.

A Container?

Containers are self contained and isolated.

We have our units of software, our packages with code and with the dependencies to run this code. And we can take it anywhere where the docker runs. And we will then be able to run exactly the same application with the same environment, whatever it is.

We don’t need to worry about installing any extra tools in that place where we wanna run our application. Because its all in the Container. That is what containers are and that is what Docker is about.

Docker is just the tool for building these containers.

Why Containers?

Why would we want independent, standardized “application packages”?

One of the main usecases of Docker is

**“Different Development and Production Environment”**

We want to build and test in exactly the same environment as we later run our app in.

1. We want to have the exact same environment for development and production.

This ensures that the app works exactly as tested.

This is where Docker and containers can help. You can to lock a specific version of software into docker container and therefore ensure that our code is executed with same exact version.

By this the potential problem of different software versions in test and prod is gone.

Because your applications will run in the container which brings its own software version.

**“Different Development environments within a team or a company.”**

Every team member should have the exactly same environment when working on the same project. 1 might write latest version code which might not run in other team member machine.

**“Clashing tools or Versions Between Different projects”**

**Online learning community:** [**Academind Community**](https://academind.com/community/)

**Virtual Machines Vs Docker Containers:**

**\*add image here**

**Docker Playground**

If you can't install Docker on your system, you can also look into this online playground: <https://labs.play-with-docker.com/>

For the most part, you should be able to follow along without issues with this playground.

**An Overview of Docker Tools:**