Docker

# What is docker?

Docker is platform for building running and shipping applications in a consistent manner.

So, if your application works on your development machine.it can run and function same way on other machines.

# So, in which scenarios we docker?

## Scenario 1:

Let’s say a person. Working on a project on one machine. But he decided to invite a new member to work has team on the project. The real problem begins when new member wants to run the code. Now new member has to install all dependencies, libraries and setup database connection all manually. Which might take a lot of time and effort but sometimes it might not work.

This were docker comes into play. Docker uses container to run, create and deploy application.so new person doesn’t have to install all dependencies and libraries. And go through all setup process. Docker automates all the setup process.

## Scenario 2:

In second scenario. We have three friends that working on application. they each have different machine and different so i.e. Let’s friend1 has mac so, friend 2 has linux so and friend 3 has windows so. So, friend a has coded has basic application and run the application. Which is working. But when he has to send the code to other friend’s code does not work because they different operating system and they have setup everything from scratches it’s time consuming.

So, with the help of docker we can run your application or code.

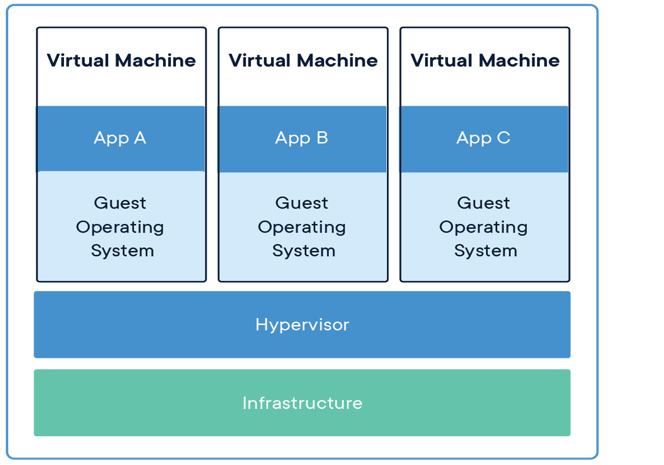
## To solve this problem we have multiple ways.

One way is to use virtual machines so that you don’t have to worry about different operating system. Or you can use containers to run your applications.

# what is virtual machine?

Virtual machine is where we can run two different operating system on same hardware also called dual booting. Instead of use different Computers for different operating system which

is costly. We can simply use one computer and two different so.



# What is a container?

A container is a lightweight, standalone, and executable software package that includes everything needed to run a piece of software, including the code, runtime, system, tools, libraries and operating system.

To simply put container is a package where operating system, code, software, tools and libraries in one place.so that you can run anywhere without any setup required.

Containers run on top of a shared OS provided by the host system.

# This is illustrated below:

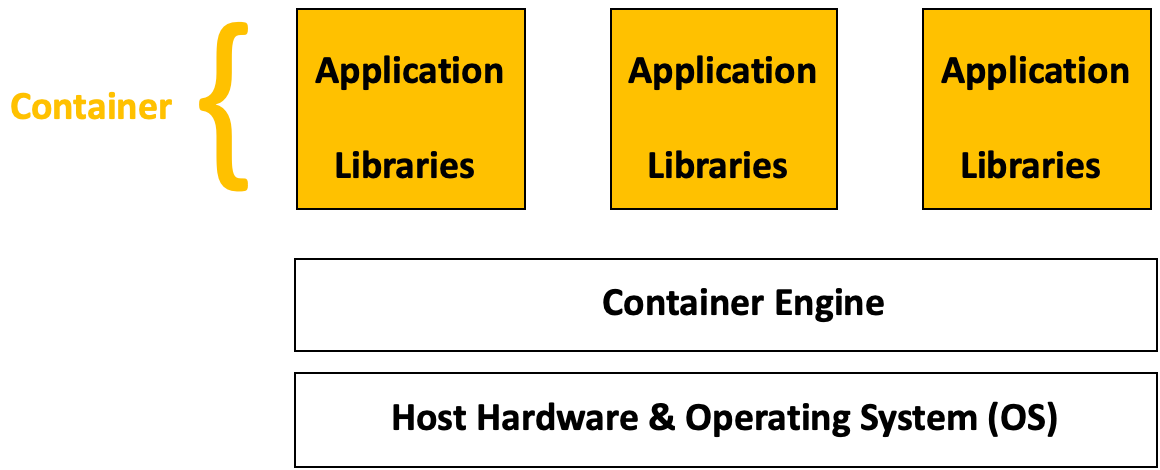


Image showing how containers work by virtualizing the OS.

Here we have container engine which allows use to run containers.

# ***Note:***

Container is different from virtual machines. Virtual machines create a complete virtualized environment, including a full operating system, hardware emulation, and application software. This provides a high degree of isolation but can be resource-intensive. On other hand container use share host operating system’s kernel and only package the application and its dependencies.

You might get a doubt like “In above note you say it only package the application and dependencies. But in previously you say that it packs operating system also?”

Yes it’s true but here we say that it uses host operating system’s kernel.to understand

We first need to know about kernel. A kernel act has a bridge between hardware and software. Which mean we can also use package stripped downed version operating system that uses same kernel. So that it is not resource intensive like virtual machines.

Docker is tool for creating and managing containers.

# Main reason to use docker in your workflow.

if you have been developing software for a while you've probably come across this situation where your application works on your development machine but doesn't somewhere else.

This why we use docker. so that your application works on every machine irrespective of development machine.

# Benefits of using docker.

* You can run your application on any machine
* Using docker you can run two different application using different version of same programing languages, frameworks, etc.
* When new person joins the team, they don’t have to sit all day setting up their system. They can simply tell docker to set up your application and docker will automatically download all the dependencies and run this application in an isolated environment called containers.

# How does docker works?

To understand how docker works we need to know two concepts:

1. Docker file
2. Docker images

## Docker file

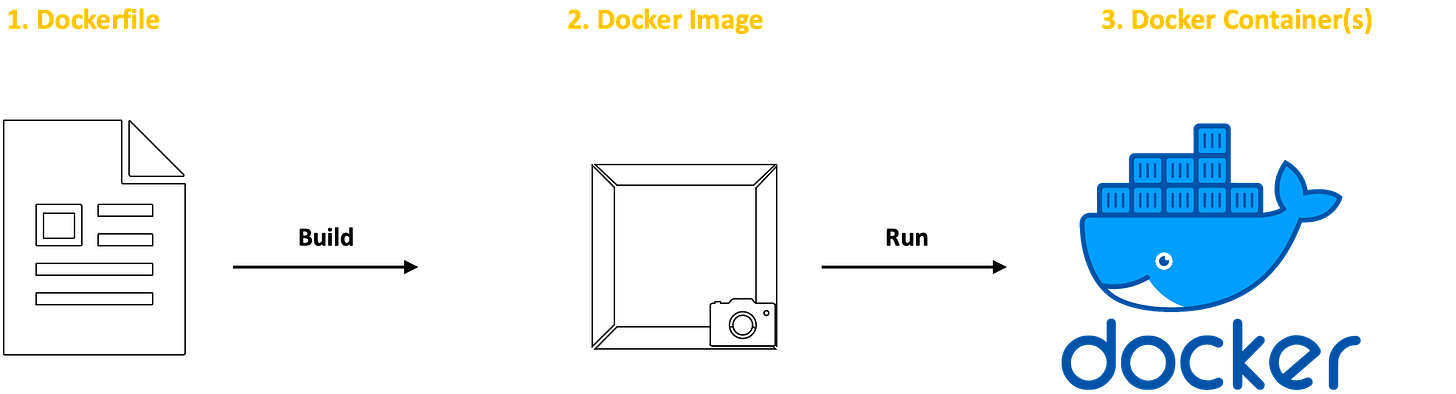
A Docker file contains the set of instructions for building a Docker Image.

In a nutshell a docker file contains which programming language to use which operating system to use what are commands to run and what dependencies to install and how to execute your application.

## Docker images

A Docker image is created when docker file is build. Docker image is executable file which contains all the necessary code, runtime, system tools, libraries, and settings required to run a software application.

A Docker Image serves as a template for creating Docker containers.

So, a Docker file is used to build a Docker Image which is then used as the template for creating one or more Docker containers. This is illustrated below. 

First, we create a docker file which is used to build docker image which is finally used to run docker container.

Imagine you're designing a new line of bicycles. You begin with a detailed document outlining the desired features and specifications. This includes factors like frame size, wheel size, gear ratios, braking system, and any specific components (like electric motors for e-bikes). we call this document has requirements file.

This document then serves as the blueprint for creating a detailed bike design, complete with engineering drawings that show the precise dimensions and specifications for each component.

From this design, physical prototypes or production models can be built. A single design template can be used to produce multiple bikes, ensuring they are all identical and meet the exact specifications outlined in the original design document.

So, think requirements file has docker file. Which is simply a set of instruction for building blueprint for bicycle.

Blueprint has docker image.

Based on this blueprint we manufacture different types of bicycles this step has docker container.

Docker container is basically a running instance of docker image.