**SUMÁRIO**

[1. WHAT IS DOCKER? 1](#_Toc109352223)

[2. DOCKER UTILITY 1](#_Toc109352224)

[2.1. DOCKER IS ONLY USED TO RUN APPLICATIONS! 2](#_Toc109352225)

[3. IMAGE AND CONTAINER 2](#_Toc109352226)

[3.1. DOCKER IMAGE 2](#_Toc109352227)

[3.1.1. DOCKER HUB 2](#_Toc109352228)

[3.2. DOCKERFILE 2](#_Toc109352229)

[3.2.1. COMMANDS 3](#_Toc109352230)

[3.3. DOCKER CONTAINER 3](#_Toc109352231)

[3.3.1. COMMANDS 3](#_Toc109352232)

[4. PERSISTING DATA WITH VOLUMES 3](#_Toc109352233)

[4.1. NAMED VOLUME 3](#_Toc109352234)

[4.2. BIND MOUNT 3](#_Toc109352235)

[5. NETWORK 3](#_Toc109352236)

[6. DOCKER COMPOSE: WORKING WITH MULTIPLE CONTAINERS 3](#_Toc109352237)

# WHAT IS DOCKER?

The documentation says: “docker is a set of service platform products that use operational-level virtualization to deliver software in packages called containers. Containers are isolated from each other and have their own software, libraries, and configuration files”.

Thus, containers, as a virtualized environment, are like virtual machines, with the difference that it – a virtual machine - has an entirely operational system and a container has only the necessary binary and resources to run an application.

Graphical user interface

Description automatically generated

Beside that, docker has two ways to work: with Desktop application and with command line. They can work together too. For some things command line is much more efficient, and for others, a graphical interface is preferable.

# DOCKER UTILITY

A container can be used in two circumstances: for development or production. **Using it for development** guarantees an environment that can be shared with other people, like your teammates, and the problems with different machines, hardware and any configuration are surpassed.

**On the other side, the benefits for production,** and considering the utility for development, is already deductible: the final environment, in production, can be the same as development. In this case, like the previously, the problems related to different machines running the same application are eliminated.

## DOCKER IS ONLY USED TO RUN APPLICATIONS!

To be clear, the original application code always stays on the host machine to be modified. Code is created and typed locally. We create a folder, named “project”, for example, create the files and type the code. This do not change at all. So, what changes? The running phase.

Instead of running the application absolutely in your environment, with your personal software configurations, server, etc, you will run it in a virtualized environment. So, yes, we always code locally but when the test moment comes out, we transfer the application for a virtualized and shareable environment (docker container) and run it.

# IMAGE AND CONTAINER

## DOCKER IMAGE

Being objective, an image is a file or a recipe for build a container. By this way, for example, an image for node will be used for the construction of a future container that will run a node environment.

So, in other words, an image is a recipe for a future container, and not just one, but for an unlimited number of containers. An image can be used infinite times for build infinite containers and they can run at same time, because each constructed container with a same image X has a different identification.

### DOCKER HUB

Docker Hub is the official website with all the public images that can be used for build containers. It’s like the npm library sharing, but, in this case, if for docker images.

Additionality, like npm, composer, and another similarity stuffs, docker hub has officially and not officially images. For example, Apache has an official image named “http”, Node has “node”, MySQL has “mysql” and exists a lot of other images made by common people.

## DOCKERFILE

Docker can build images automatically by reading the instructions from a Dockerfile. A Dockerfile is a text document that contains all the commands a user could call on the command line to assemble an image.

### COMMANDS

## DOCKER CONTAINER

### COMMANDS

# PERSISTING DATA WITH VOLUMES

## NAMED VOLUME

## BIND MOUNT

# NETWORK

# DOCKER COMPOSE: WORKING WITH MULTIPLE CONTAINERS