**Docker Section 2**

**Summary:**

* Why Docker
* Dockerfile
* Image and containers
* Docker commands

# Why Docker

A new developer in joining a company, the dev needs to install all the technologies that requires to work in the application:

* It will take a lot of time.
* Installation Errors.
* Problem of versions.

Deploy your app in testing, staging and production, you need to do the same process when you install it in the first time....

Docker comes to solve this problem.

The idea of docker is to transfer the application with all of it dependencies to another machine or another environment.

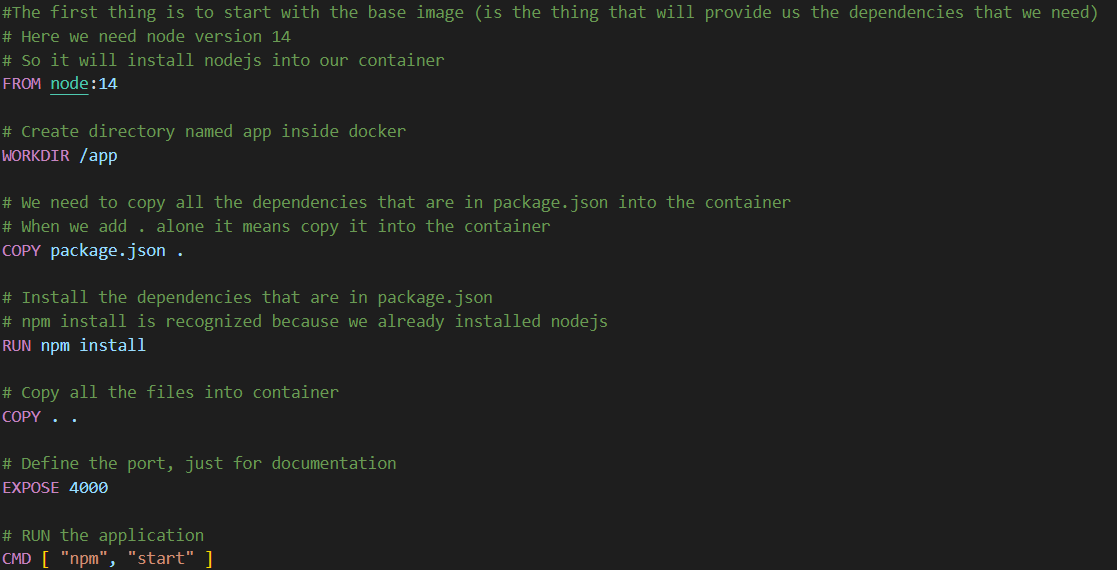
# Dockerfile

To dockerise the application we need a Dockerfile.

Dockerfile is a file that tell the application how it can be run and what's the dependencies that need to be used.

RUN Dockerfile : Look **.Dockerfile** file

**docker build -t image-name .** : Run the Dockerfile and install all the dependencies.



# Docker Commands:

|  |  |
| --- | --- |
| **Commands** | **Description** |
| **docker version** | Show what version of docker we are working on |
| **docker search** | The “docker search” command searches for specific images through the Docker hub. This command returns the specific information, including image name, description, automated, official stars, etc |
| **docker pull** | this command pulls a specific image from the Docker Hub  **docker pull MySQL** |
| **docker run** | This command is used to create a container from an image. |
| **docker ps** | List all the running containers. |
| **docker stop** | The ‘docker stop’ command stops a container using the container name or its id |
| **docker restart** | This command is used to restart the stopped container |
| **docker kill** | This command is used to stop the container immediately by killing its execution |
| **docker login** | This command helps you to log into your docker hub |
| **docker logs** | This command is used to check the logs of all the docker containers with the corresponding contained id mentioned in the command. |
| **docker logout** | This command will log you out of the docker hub. |

# Image and containers:

**docker ps**: show all the running containers.

**docker ps -a**: show all the running and stopped containers.

**docker image ls**: list images

**docker stop container\_name**: stop the container: container\_name

**docker rm container\_name -f**: remove the container: container\_name, -f: forced

**How to run the application?**

**docker run --name node-app-container -d -p 1000:4000 node-app**

Run the image node-app that is existing in the container node-app-container.

-d : to not show the logs in the console.

-p 1000:4000 (run the port 4000 that is existing in the container into the port 1000 in the local ==> so we need to write localhost:1000)

**Image vs Container:**

In dockerfile we start by the base image, docker goes to dockerhub that store all the images that can be used in (FROM): (https://hub.docker.com)

When we build the docker file, he creates the docker image.

We can create from one image one or more containers with the command RUN

To enter to the command line of the application in docker

**docker exec -it node-app-container bash**

-it: interactive

**Questions:**

*Do I need to copy all the files?*

No, so we add files that aren't needed like (. dockerfile, node\_modules ...) to .dockerignore

*Why did we split "package.json" copy command?*

Because when we run dockerfile, it will all times run npm install.