Not all the machines were updated sometimes.

User buy something🡪Error 🡪 Refresh 🡪 Works

The reason is that 1 of 50 servers had a problem!

Solution:

Graphical user interface, application, Teams

Description automatically generated

Docker 3 Phases:

Graphical user interface

Description automatically generated

CI/CD – Different Versions – Roll Forward

Text

Description automatically generated

Just Download Docker: (iteresting)

<https://docs.docker.com/get-started/> (watched 16/25)

Docker File looks like this:

Text

Description automatically generated

It is the list of stuff that Docker Machine understands and apply

docker build --help

docker build --tag hello-world .

docker images

docker run --help

docker run hello-world

docker ps

docker ps -a

docker start frosty\_blackwell

docker stop frosty\_blackwell

docker rm frosty\_blackwell

// Internet Connection

########################

docker run -d -p 80:80 docker/getting-started

OR

docker run -p 8080:80 --name hello -d hello-world

docker ps

// Logs:

#########

docker start hello

docker logs hello

docker logs --help

Container is a box containing everything starting from source code, dependencies (versions) & everything needed to run the application.

A blue screen with white text

Description automatically generated with low confidence

Diagram, treemap chart

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Images & Containers

Docker Images: Images are ReadOnly (if you wanna change something create a new copy…)

Graphical user interface, text, application, chat or text message

Description automatically generated

Treemap chart

Description automatically generated

Diagram

Description automatically generated with medium confidence

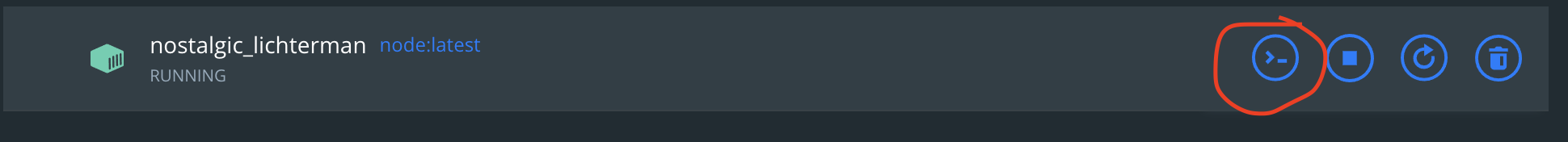
So the main Process is how to create Images & how to run Containers on them

Images are made up from several Layers:

Parent image: Graphical user interface

Description automatically generated

* Docker pull node (Search for images in the App docker)
* After downloading you will find a new image in Docker app 🡪 Run a container on it
* After that you will see a new Container which is using the image click on command (red)



* It will open a Terminal where you can interact with the environment:

Text

Description automatically generated

* This image (till now) is the parent image!
* To build the different Layers of the Image we need the Docker File ! it is a list of construction.
* When an app is downloaded don’t npm install 🡪 Create an Image

Now in the Docker file do these:

1. FROM node:17-alpine where alpine is linux version and 17 is node version
2. WORKDIR /app (When we run commands on the images in the future do it from /app dir)
3. Copy . . (copy the files in “.” The same dir as dockerFile to the “.” Of image which is “/app” directory)
4. RUN npm install (run this on the Image “In Workdir “/app” ”)
5. EXPOSE 4000 (just to give which port will be export for the image)
6. CMD[“node”, “app.js”]. (When the container runs on the image !)

To build the image now go to the dir od Dockerfile & type “ Docker build -t jadoApp . ”

We will not have node\_modules in rep cause its already in the Image.

“.dockerignore” is like .gitignore

Commands:

* Docker image
* Docker run –name myapp\_c1 myapp
* Docker ps
* Docker stop myapp\_c1
* Docker run –name myapp\_c1 myapp -p 4000 :4000 -d myapp

When I change anything in the Source Code we should create a new image to pick up these changes! (it will take less time cause docker uses the cache which are in this case FROM & WORKDIR )

In the foto below the new layer Copy Pacakage.json . help us take the runnpm install from cache tooA screenshot of a computer

Description automatically generated with medium confidence

Continue here <https://www.youtube.com/watch?v=4XsjXscp70o&list=PL4cUxeGkcC9hxjeEtdHFNYMtCpjNBm3h7&index=9>