TS-NODE HELLO WORLD IN DOCKER CONTAINER

In the Backend 1 classes we did a node hello world with typescript and express, I'm going to use docker on a VM with Ubuntu in order to run that hellow world server that we did in a container.

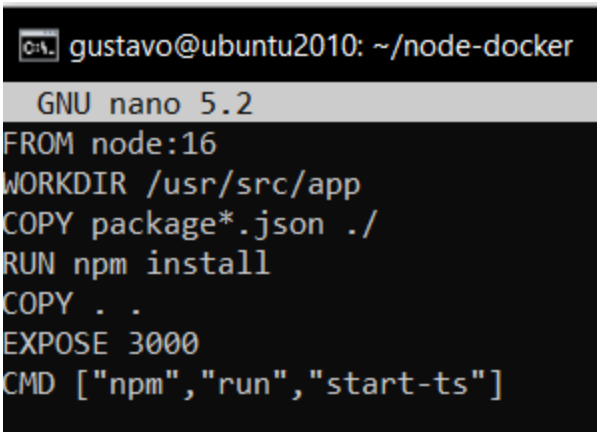
On Windows PS

>scp "C:/Users/your_username/Desktop/app/node.rar" user@ip:~/node-docker (First we need to send the files from windows to Ubuntu VM, I put package.json file, tsconfig.json file, tslint.json file and the src directory on that rar file and then send it to a folder in the user's dir called node-docker on the Ubuntu VM)

>ssh user@ip (Login with ssh keys to Ubuntu VM)

On Ubuntu Bash

>nano Dockerfile (Inside ~/node-docker create the Dockerfile)



>sudo docker build . -t gustavo/node-docker (The tag is optional)

```
gustavo@ubuntu2010:~/node-docker$ sudo docker build . -t gustavo/node-docker
Sending build context to Docker daemon  52.22kB
Step 1/7 : FROM node:16
 ---> 424bc28f998d
Step 2/7 : WORKDIR /usr/src/app
 ---> Using cache
 ---> a0d70e1899cf
Step 3/7 : COPY package*.json ./
 ---> Using cache
 ---> 61969b21bfa7
Step 4/7 : RUN npm install
 ---> Using cache
 ---> 32d1b1bd76bb
Step 5/7 : COPY . .
 ---> Using cache
 ---> 6221a1afb482
Step 6/7 : EXPOSE 3000
 ---> Using cache
 ---> 10595ef88136
Step 7/7 : CMD ["npm","run","start-ts"]
 ---> Using cache
 ---> 55a36386ecd5
Successfully built 55a36386ecd5
Successfully tagged gustavo/node-docker:latest
gustavo@ubuntu2010:~/node-docker$
```

>sudo docker images (Verify the image was created successfully)

```
gustavo@ubuntu2010:~/node-docker$ sudo docker images
REPOSITORY            TAG             IMAGE ID       CREATED          SIZE
gustavo/node-docker   latest          55a36386ecd5   15 minutes ago   1GB
```

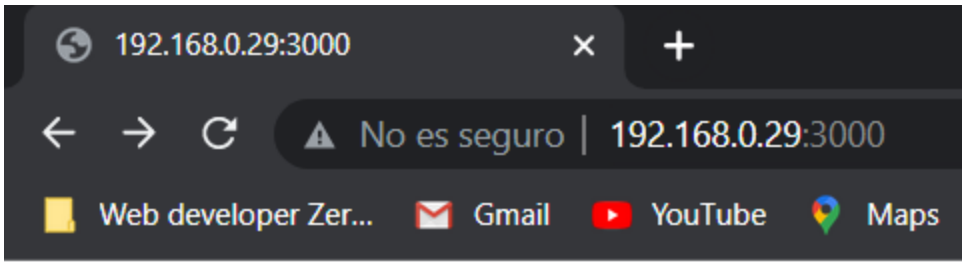
>sudo docker run -p 3000:3000 gustavo/node-docker (Run the image)

```
gustavo@ubuntu2010:~/node-docker$ sudo docker run -p 3000:3000 gustavo/node-docker

> backend@1.0.0 start-ts
> ts-node index.ts

Up on 3000
```

Now if you go from your Windows machine to http://ubuntu_ip:3000 you should see the “Hola mundo” message



Hola mundo

DOCKER VS PODMAN

Podman se diferencia principalmente por su arquitectura, no cuenta con un proceso de larga duracion para administrar contenedores, cuando se corre el comando podman se interactua directamente con el proceso que inicia el contenedor y recupera imagenes, docker depende de una conexion al daemon docker, la CLI de docker envia comandos al daemon y este actua de acuerdo a estos comandos.

Podman es mas seguro por que no cuenta con un daemon lo cual reduce la superficie de ataque a los contenedores a diferencia de docker.

En podman los contenedores pueden formar pods que funcionan juntas, similar a kubernetes Pod, docker por el contrario no cuenta con esto.