Início de projeto node com docker

Iniciar projeto em node

Criando o arquivo package.json

```
$ npm init -y
```

Criando arquivos

```
$ touch inde.js Dockerfile
```

Instalação do express

```
$ npm install express
```

Criando .dockerignore

arquivos ao qual o docker deve ignorar

```
$ touch .dockerignore
```

Alterando o arquivo package.json

adicione a linha "start": "node index.js"

```
"scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "node index.js"
  }
```

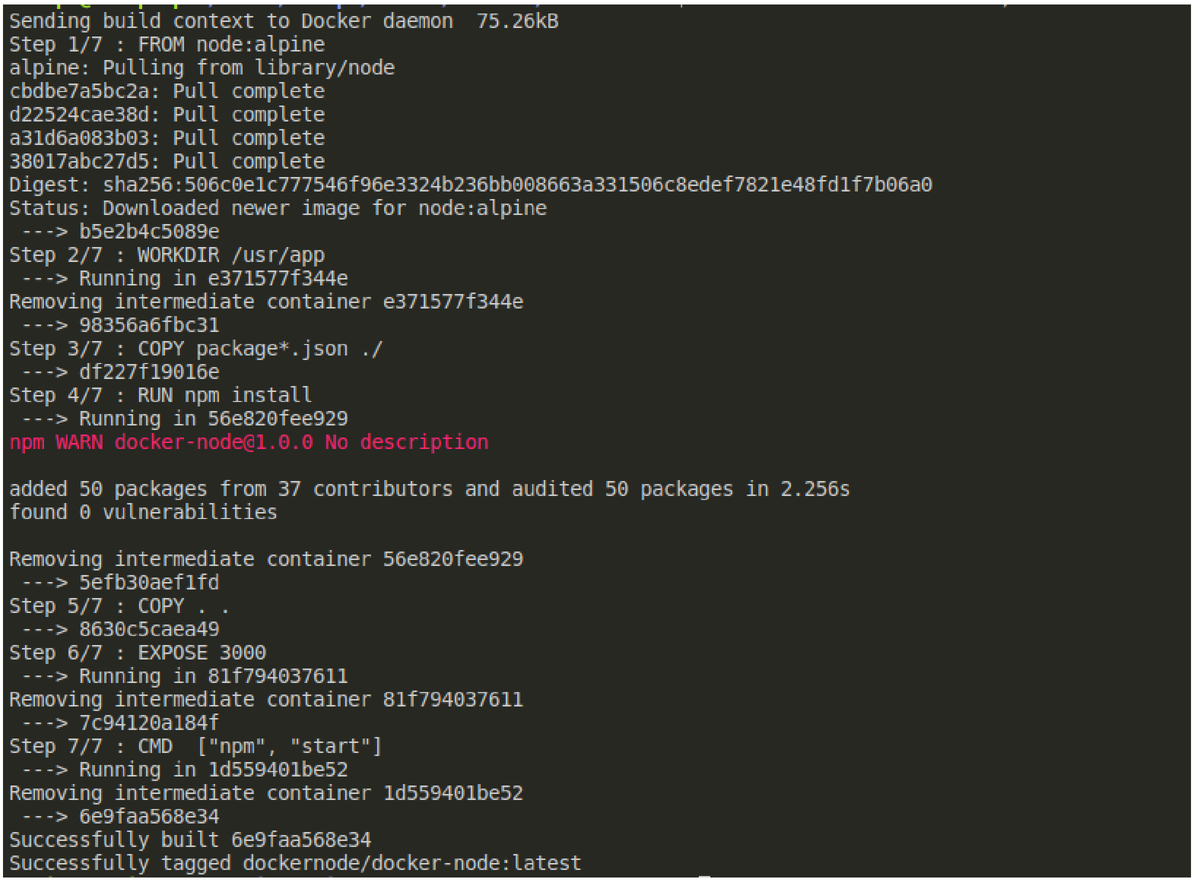
Criando uma imagem

```
$ docker build -t dockernode/docker-node .
```

Rodando o container

docker run -p < portaacesso:portacontainer > -d < imagem a ser utilizada>

```
$ docker run -p 3000:3000 -d dockernode/docker-node
```



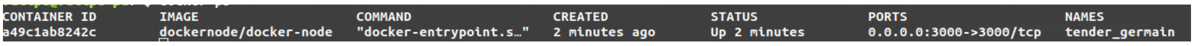
```
Sending build context to Docker daemon  75.26kB
Step 1/7 : FROM node:alpine
alpine: Pulling from library/node
cbdbe7a5bc2a: Pull complete
d22524cae38d: Pull complete
a31d6a083b03: Pull complete
38017abc27d5: Pull complete
Digest: sha256:506c0e1c777546f96e3324b236bb008663a331506c8edef7821e48fd1f7b06a0
Status: Downloaded newer image for node:alpine
 ---> b5e2b4c5089e
Step 2/7 : WORKDIR /usr/app
 ---> Running in e371577f344e
Removing intermediate container e371577f344e
 ---> 98356a6fbc31
Step 3/7 : COPY package*.json ./
 ---> df227f19016e
Step 4/7 : RUN npm install
 ---> Running in 56e820fee929
npm WARN docker-node@1.0.0 No description

added 50 packages from 37 contributors and audited 50 packages in 2.256s
found 0 vulnerabilities

Removing intermediate container 56e820fee929
 ---> 5efb30aef1fd
Step 5/7 : COPY . .
 ---> 8630c5caea49
Step 6/7 : EXPOSE 3000
 ---> Running in 81f794037611
Removing intermediate container 81f794037611
 ---> 7c94120a184f
Step 7/7 : CMD  ["npm", "start"]
 ---> Running in 1d559401be52
Removing intermediate container 1d559401be52
 ---> 6e9faa568e34
Successfully built 6e9faa568e34
Successfully tagged dockernode/docker-node:latest
```

Ver imagens que estão rodando

```
$ docker ps
```



```
CONTAINER ID        IMAGE                    COMMAND                  CREATED             STATUS              PORTS                    NAMES
a49c1ab8242c        dockernode/docker-node   "docker-entrypoint.s…"   2 minutes ago       Up 2 minutes        0.0.0.0:3000->3000/tcp   tender_germain
```

Usando docker compose

criando arquivo docker-compose.yml

```
$ touch docker-compose.yml
```

definindo versão do composer e servicos

```
version: "3"

services:
    app:
        build: .                    ( indicação de onde está o arquivo
Dockerfile)
        command: npm start          ( comando a ser executado quando a aplicação
subir )
        ports:
            - "3000:3000"           ( redirecionamento de portas )
        volumes:
            - .:/usr/app            ( pasta aonde serão refletidas as alterações
)
```

instalando nodemom

```
$ npm install nodemon
```

altere novamente o package.json

```
adicione a linha "start": "node index.js" para "start": "nodemon index.js"

"scripts": {
    "test": "echo \"Error: no test specified\" && exit 1",
    "start": "nodemon index.js"

  }
```

Parando o container para modificações

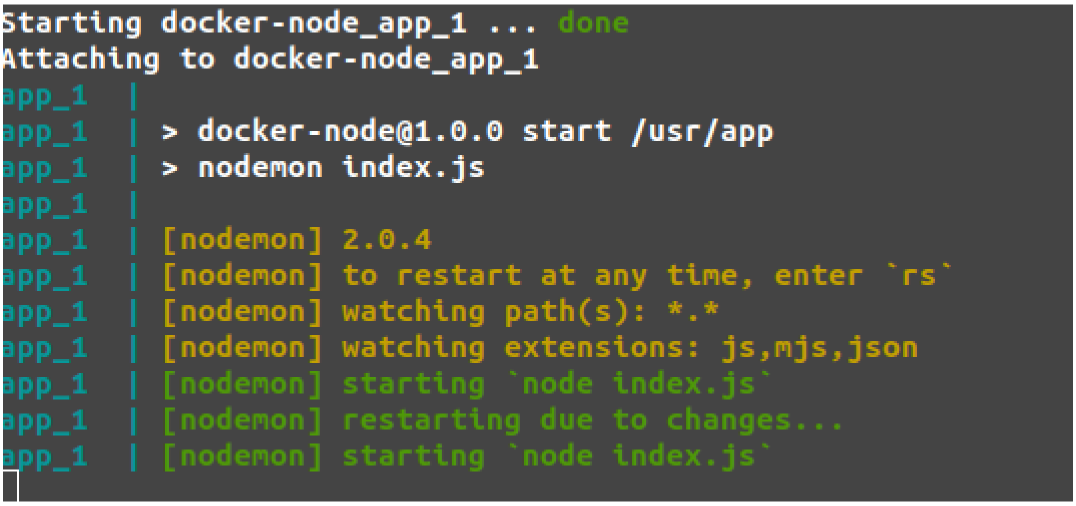
docker ps (para listar as imagens)

docker stop < CONTAINER ID NAMES >

```
$ docker stop a49c1ab8242c (no meu caso)
```

subindo as alterações

```
$ docker-compose up
```



```
Starting docker-node_app_1 ... done
Attaching to docker-node_app_1
app_1  |
app_1  | > docker-node@1.0.0 start /usr/app
app_1  | > nodemon index.js
app_1  |
app_1  | [nodemon] 2.0.4
app_1  | [nodemon] to restart at any time, enter `rs`
app_1  | [nodemon] watching path(s): *.*
app_1  | [nodemon] watching extensions: js,mjs,json
app_1  | [nodemon] starting `node index.js`
app_1  | [nodemon] restarting due to changes...
app_1  | [nodemon] starting `node index.js`
```