

Redes de Computadores

Prof. Jean Carlos

Aluno
Rafael Augusto Campos Plinio

DOCKER

Introdução

Com o aumento da tecnologia em todos os lugares do mundo, há uma crescente demanda para empresas de todos os tamanhos e até mesmo para programadores solo ou freelancer. Existe também o aumento de dados que trafegam na rede, este aumento se dá por diversos fatores, como por exemplo a melhoria na qualidade das imagens de fotos e vídeos, na melhoria do áudio, a maior quantidade de pessoas que tem acesso à internet, e por aí vai.

Alguns anos atrás, para atender a alta demanda era necessário a implantação de uma infraestrutura física, era trabalhoso e caro mas agora com o Docker e outras tecnologias similares isso mudou.

A tecnologia Docker nos oferece um ambiente virtual com recursos que podemos usar diretamente em aplicações. Estes recursos são separados por containers e através desses containers, um ambiente ou uma aplicação inteira torna-se portátil para outro host, desde que tenha o Docker instalado.

Instalação

Detector de Visitas – NodeJS

Criando as imagens

Haproxy

No diretório onde estão os arquivos dockerfile e haproxy.cfg.

`docker build -t my-haproxy .`

```
haproxy — rafaelplinio@Rafaels-Mini — ..aster/haproxy — zsh — 80x24
[+] haproxy ls
Dockerfile haproxy.cfg
[+] haproxy docker build -t my-haproxy .
Sending build context to Docker daemon 3.584kB
Step 1/2 : FROM haproxy
latest: Pulling from library/haproxy
d121f8d1c412: Pull complete
1cae796857a0: Pull complete
718fed174203: Pull complete
Digest: sha256:a24b18b09c223e606fe170ac83916f6cfb707d2c97db4b9eb8aa9eaab3649acb
Status: Downloaded newer image for haproxy:latest
----> 8e202ffaa1a8
Step 2/2 : COPY haproxy.cfg /usr/local/etc/haproxy/haproxy.cfg
----> af3864489325
Successfully built af3864489325
Successfully tagged my-haproxy:latest
[+] haproxy
```

Nodejs

Voltar uma pasta e acessar a pasta do nodejs (local onde está o dockerfile).

```
Cd ../nodejs
```

```
Docker build -t my-nodejs .
```

```
nodejs — rafaelplinio@Rafaels-Mini — ..master/nodejs — -zsh — 80x41
[→ haproxy cd ../nodejs
[→ nodejs ls
Dockerfile  index.js  package.json
[→ nodejs docker build -t my-nodejs .
Sending build context to Docker daemon  4.608kB
Step 1/6 : 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 2/6 : WORKDIR "/app"
----> Running in 5c9b4bf46dcb
Removing intermediate container 5c9b4bf46dcb
----> 9410f7256746
Step 3/6 : COPY package.json .
----> e44207c5d5fc
Step 4/6 : RUN npm install
----> Running in 83692c45185c
npm notice created a lockfile as package-lock.json. You should commit this file.
npm WARN app No description
npm WARN app No repository field.
npm WARN app No license field.

added 54 packages from 41 contributors and audited 54 packages in 2.968s
found 0 vulnerabilities

Removing intermediate container 83692c45185c
----> 35cb754417e8
Step 5/6 : COPY . .
----> 930fd3abf81b
Step 6/6 : CMD ["npm", "start"]
----> Running in 9933cb76650f
Removing intermediate container 9933cb76650f
----> 10b6acc5faa2
Successfully built 10b6acc5faa2
Successfully tagged my-nodejs:latest
[→ nodejs
```

As duas imagens estão criadas (localmente): `docker images`

```
nodejs — rafaelplinio@Rafaels-Mini — ../master/nodejs — zsh — 80x41
[→ nodejs docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
my-nodejs            latest             10b6acc5faa2        9 minutes ago
120MB
my-haproxy           latest             af3864489325        12 minutes ago
93.3MB
<none>               <none>             219b7e573589        45 hours ago
357MB
node                 alpine             b5e2b4c5089e        2 days ago
117MB
haproxy              latest             8e202ffaa1a8        3 days ago
93.3MB
golang               1.11-alpine        e116d2efa2ab        12 months ago
312MB
→ nodejs
```

O Redis não teve uma imagem criada porque a imagem padrão `redis` é o suficiente para a aplicação.

Criando e conectando os containers.

Iniciando o Redis Container.

`docker container run -it -d --name redisnode01 redis`

```
nodejs — rafaelplinio@Rafaels-Mini — ../redis/nodejs — zsh — 80x24
[→ nodejs docker container run -it -d --name redisnode01 redis
6f518ddd355c2eeae135ff507b8af39ff59954613a88de7931f0f9ac578fa31a
→ nodejs
```

Iniciando o NodeJS Container e conectando ao Redis Container.

`docker container run -it -d --name nodejsserver01 --link redisnode01 my-nodejs`

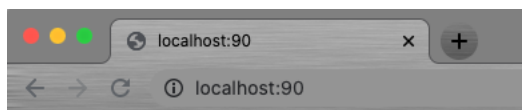
```
nodejs — rafaelplinio@Rafaels-Mini — ../redis/nodejs — zsh — 80x24
[→ nodejs docker container run -it -d --name nodejsserver01 --link redisnode01 m
y-nodejs
11aded5e4523694b80ef90cc23fbe6dac87fcb5ce33ed9327b1ace3e0e8797ab
→ nodejs
```

Iniciando o Haproxy Container e conectando ao NodeJS Container.

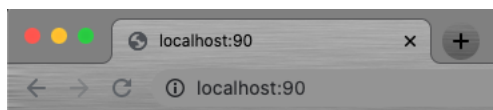
`docker container run -it -d -name haproxyserver --publish 90:80 -link nodejsserver01 my-haproxy`

```
nodejs — rafaelplinio@Rafaels-Mini — ../redis/nodejs — zsh — 80x24
[→ nodejs docker container run -it -d --name haproxyserver --publish 90:80 --lin
k nodejsserver01 my-haproxy
2b3164b581f2d1c7ec5eff5ce20874e54172e0a096be59548974fabfcba88bc2
→ nodejs
```

Para testar o funcionamento dos dockers integrados, basta acessar a URL <http://localhost:90/>



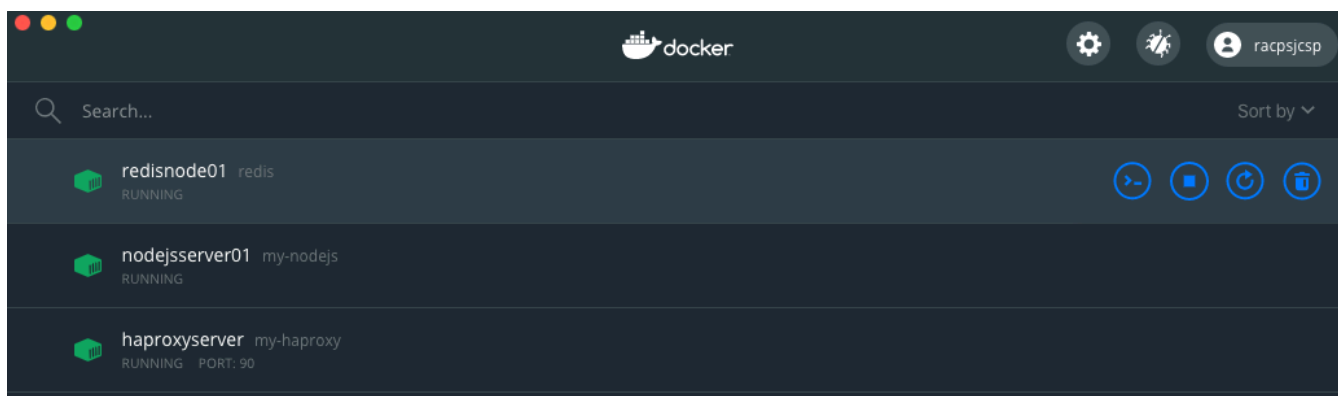
Number of Visits 3



Number of Visits 4

Toda vez que o usuário acessar a página (ou atualizar: ⌘ + R (MacOS) ou F5 (Windows)), o contador aumentará, provando que a integração está completa e funcionando.

Containers em execução (Docker Desktop).



Conclusão

Utilizar as novas tecnologias de infraestrutura em nuvem tem uma escalabilidade dinâmica e oferece agilidade e outros benefícios no desenvolvimento de software.

Os benefícios de se utilizar o Docker são: criar suas próprias imagens (containers prontos) através dos Dockerfiles, redução no tempo de deploy de uma aplicação ou infraestrutura e definição de recursos utilizados por container (CPU, memória, etc).