Professor: Danilo Sibov

Microsserviços | Aula 2

Laboratórios

Este módulo aborda os seguintes tópicos:

- **Laboratório 5 - Primeiro Container**

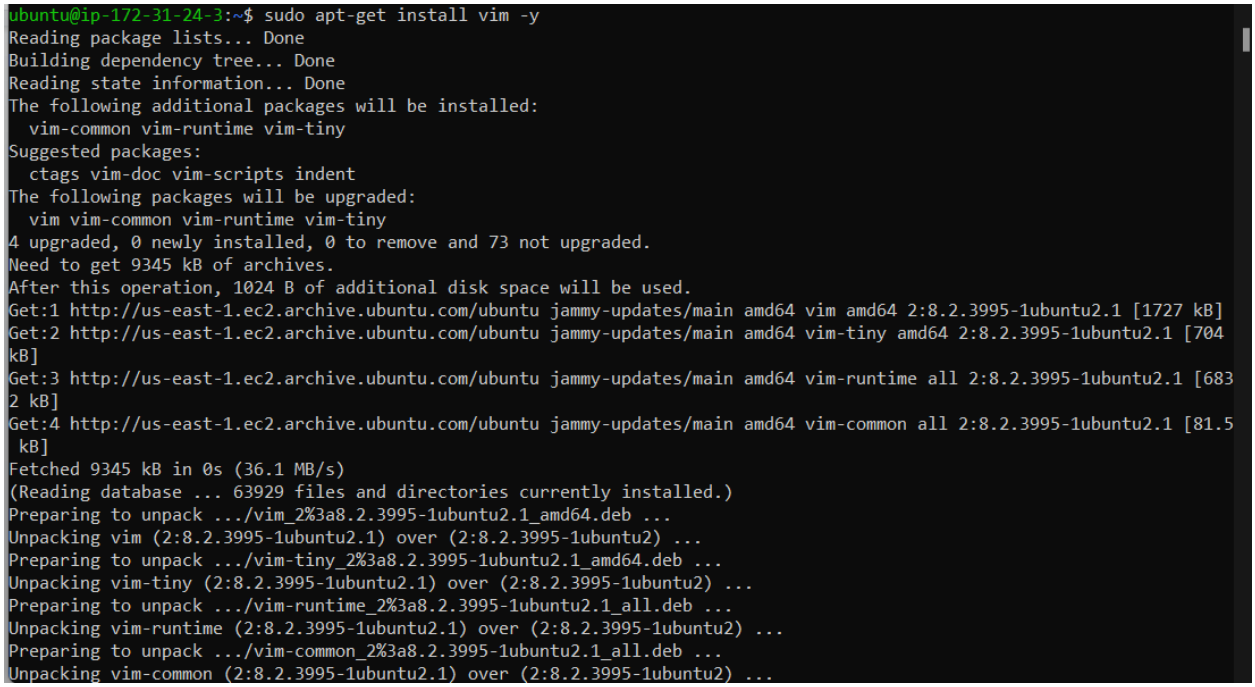
Neste laboratório você aprenderá a criar um Dockerfile simples, vai aprender a **construir** sua imagem a partir deste Dockerfile, executar um container com alguns parâmetros específicos e vai aprender a ver os Logs de um container.

Etapa 1 - Ensinar a criar um Dockerfile simples
Etapa 2 - Ensinar a buildar uma imagem
Etapa 3 - Ensinar a mostrar imagens do Docker
Etapa 4 - Ensinar a executar um container explicando
Etapa 5 - Ensinar a visualizar logs de um container
Etapa 6 - Acessar um Container

Etapa 1 - Como criar um Dockerfile simples

Para criarmos um Dockerfile simples, primeiro, vamos precisar de um recurso muito importante, um editor de texto, neste caso, estaremos utilizando o **vim**

1. **Para instalar o** vim **digite:** sudo apt-get install vim -y

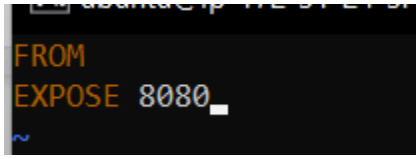


2. **Após instalar o** vim**, vamos criar nosso primeiro Dockerfile,**

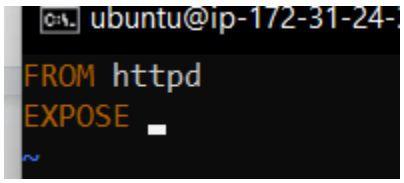
digite: vim Dockerfile

OBS: O nome precisa ser exatamente: Dockerfile

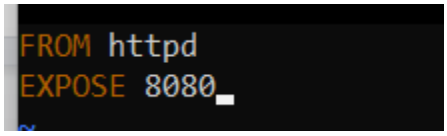
3. **Para começarmos a digitar no** vim **aperte a tecla “** i **“ e digite a seguinte sintaxe:**



4. **Feito isso, neste exemplo, vamos estar utilizando a imagem oficial do Apache, conhecida como** httpd, **portanto, no campo FROM digite:** http



5. **Depois, no campo EXPOSE, digite:** 8080



OBS: No campo EXPOSE, é onde definimos a porta que o Container irá rodar/escutar, ou seja, requisições do tipo http://localhost:8080 iram ser redirecionadas para este container.

6. **Após executar os dois últimos STEPS acima, aperte a tecla** ESC **e note que o** insert **saiu do campo inferior esquerdo**

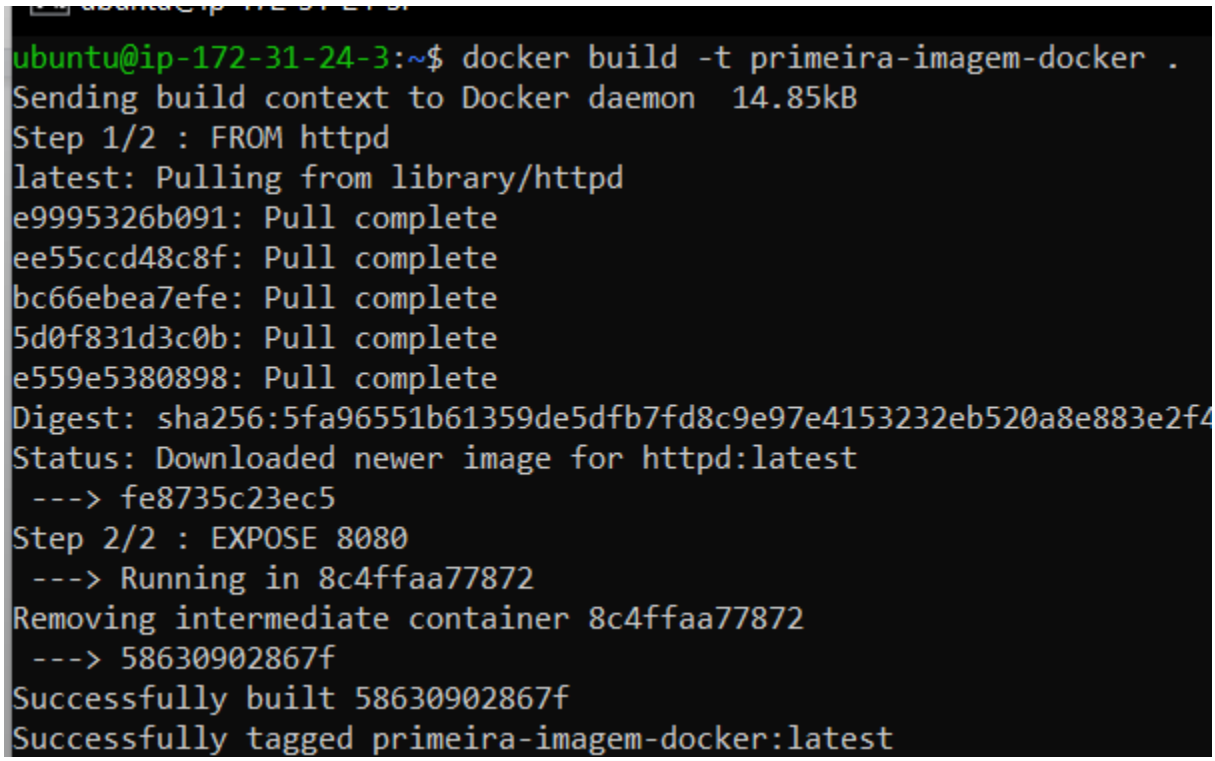
7. **Para salvarmos o nosso arquivo após ter apertado ESC, aperte** " : " **e depois digite** wq! **E depois aperte a tecla ENTER**

W = Write
Q = quit
! = Force

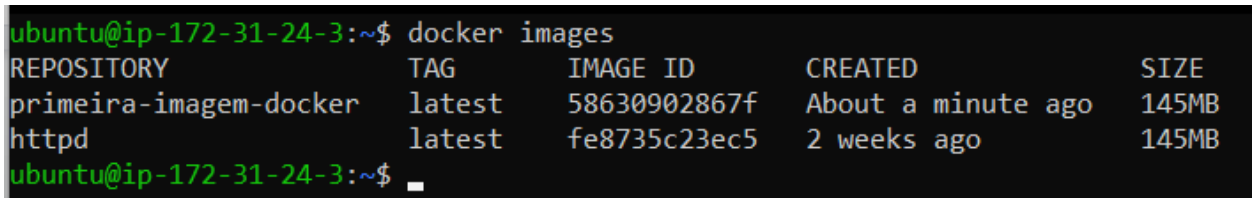
8. **Feito isso, nós temos o nosso primeiro Dockerfile**

Etapas 2, 3 e 4 - Como “ buildar “ uma imagem Docker

1. **Para buildar sua primeira imagem Docker, execute o seguinte comando:** docker build -t primeira-imagem-docker .



2. **Após você construir/buildar sua primeira imagem, digite:** docker images



Etapa 4 - Executando um Container

1. **Para executarmos um container, é necessário de uma Imagem Docker, anteriormente construímos uma, vamos utilizar ela, digite:** docker run -d -p 8080:8080 --name primeiro-container primeira-imagem-docker:latest

2. **Note que ao rodar o comando acima, um ID foi gerado automaticamente:**

```
ubuntu@ip-172-31-24-3:~$ docker run -d -p 8080:8080 --name primeiro-container primeira-imagem-docker:latest
5f77de296303288e36701aca0b8fc41bb32c69106968497aa9338a51ade9986c
ubuntu@ip-172-31-24-3:~$
```

3. **Vamos ver se o container está UP, digite:** docker container ls

```
ubuntu@ip-172-31-24-3:~$ docker container ls
CONTAINER ID   IMAGE                          COMMAND              CREATED
5f77de296303   primeira-imagem-docker:latest   "httpd-foreground"   About a minute ago
ubuntu@ip-172-31-24-3:~$
```

Se a saída do comando acima mostrar o seu container, significa que seu container está UP/Healthy

4. **Para podermos ver os LOGS de um container e identificar possíveis problemas, digite:** docker logs <nome/id do container>

```
ubuntu@ip-172-31-24-3:~$ docker logs primeiro-container
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, usi
AH00558: httpd: Could not reliably determine the server's fully qualified domain name, usi
[Thu Nov 10 18:44:41.637694 2022] [mpm_event:notice] [pid 1:tid 139661584325952] AH00489:
[Thu Nov 10 18:44:41.637899 2022] [core:notice] [pid 1:tid 139661584325952] AH00094: Comma
ubuntu@ip-172-31-24-3:~$
```

Etapa 6 - Como acessar um Container

1. **Para acessar um container Docker é simples, digite:** docker exec -it <nome/id do container> bash

```
ubuntu@ip-172-31-24-3:~$ docker exec -it primeiro-container bash
root@5f77de296303:/usr/local/apache2#
```

NOTE:

Exec: é um argumento do comando docker que pode ser utilizado para executar comandos ou entrar no container

-it: significa interative, ou seja, vamos acessar o modo iterativo do container

Bash: em conjunto com o -it, utilizamos o parâmetro bash, para dizer ao comando que queremos conectar com o container Docker

2. **Para sair do container, digite:** exit

```
ubuntu@ip-172-31-24-3:~$ docker exec -it primeiro-container bash
root@5f77de296303:/usr/local/apache2# exit
exit
ubuntu@ip-172-31-24-3:~$
```

3. **Para executar comandos remotamente em um container digite:** docker exec <nome/id do container> env

```
ubuntu@ip-172-31-24-3:~$ docker exec primeiro-container env
PATH=/usr/local/apache2/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin
HOSTNAME=5f77de296303
HTTPD_PREFIX=/usr/local/apache2
HTTPD_VERSION=2.4.54
HTTPD_SHA256=eb397feeefccaf254f8d45de3768d9d68e8e73851c49afd5b7176d1ecf80c340
HTTPD_PATCHES=
HOME=/root
ubuntu@ip-172-31-24-3:~$
```

Neste caso acima, não utilizamos o -it, apenas passamos um **comando** para o **exec** Executar dentro do container, neste caso, o comando foi **env** que lista todas as variáveis de ambiente do Linux.