# **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**

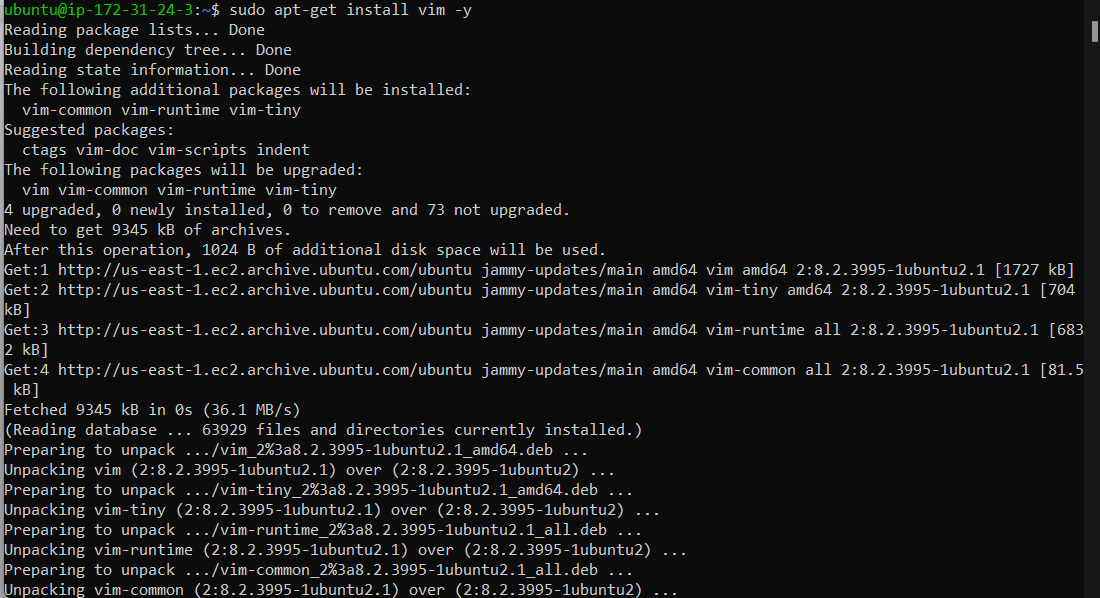
**Etapa 4 - Ensinar a executar um container explicando**

**Etapa 5 - Ensinar a visualizar logs de um container**

**Etapa 6 - Acessar um Container**

**Etapa 1 - Como criar um Dockerfile simples**

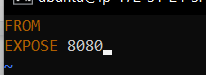
Para criarmos um Dockerfile simples, primeiro, vamos precisar de um recurso muito importante, um editor de texto, neste caso, estaremos utilizando o **vim**

1. **Para instalar o** vim **digite:** sudo apt-get install vim -y
2. **Após instalar o** vim**, vamos criar nosso primeiro Dockerfile,**

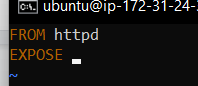
**digite**: vim Dockerfile

**OBS**: O nome precisa ser exatamente: Dockerfile

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

****

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



1. **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.

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

****

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

****

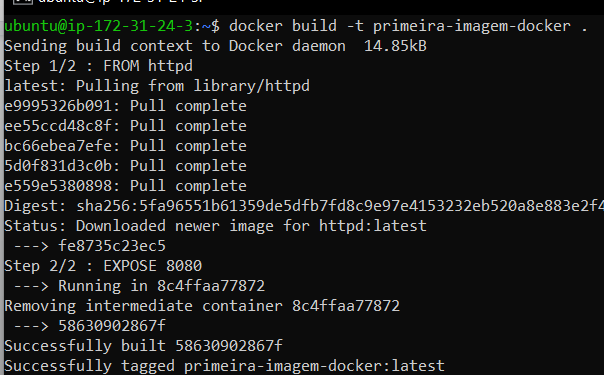
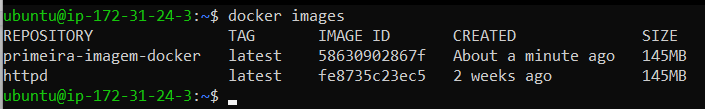
*W = Write*

*Q = quit*

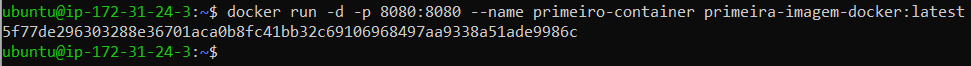
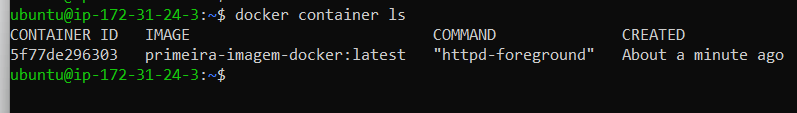
*! = Force*

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

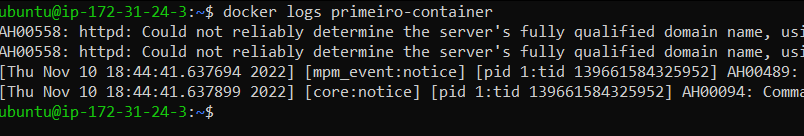
**Etapas 2, 3 e 4 - Como “ buildar “ uma imagem Docker**

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

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

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

**Etapa 6 - Como acessar um Container**

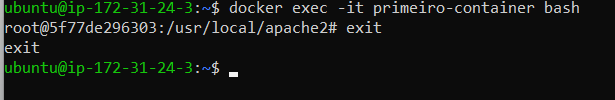
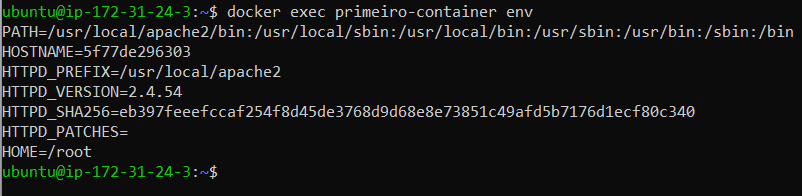
1. **Para acessar um container Docker é simples, digite:** docker exec -it <nome/id do container> bash

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

1. **Para sair do container, digite:** exit
2. **Para executar comandos remotamente em um container digite:** docker exec <nome/id do container> env

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.