Docker Command

Every commands use in the course.

* docker container run hello-world

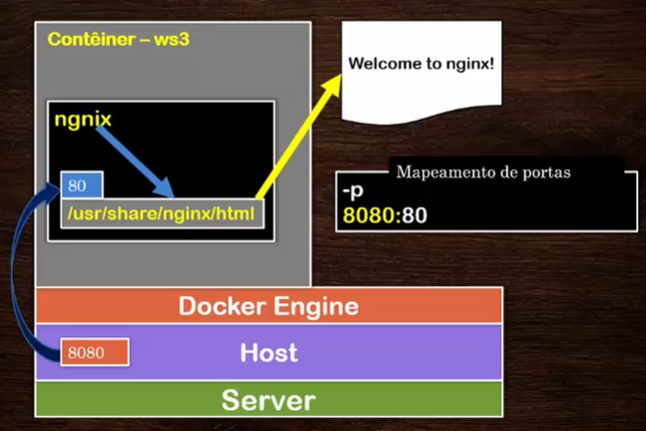
docker image pull

run docker container create

docker container start

docker container exec

* docker container run alpine ls -l
* docker container run -it alpine bin/sh
* docker container exec container\_id ls -l
* docker container stop container\_id
* docker container run –name novoNome imagem
* docker container rm container\_id
* docker container run --name ws1 -p 8080:80 -d(Segundo plano) nginx



* docker container run --name wsTeste -p 8080:80 -v c:/users/Jefferson.ricardo/downloads/html:/usr/share/nginx/html -d nginx
* docker container inspect container\_id

Criando container com image dotNet

* docker container run --name demoNet -it mcr.microsoft.com/dotnet/sdk

Criando appConsole

* - dotnet new console

Build appConsole

* - dotnet build

Executando appConsole

* - dotnet run

Entrar no container após criado e parado.

* docker container start -ia dotnet

Interface gráfica do usuário

Descrição gerada automaticamente

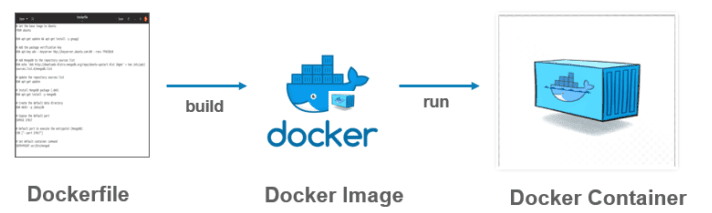
* docker container prune
* docker image prune

Criando Imagens

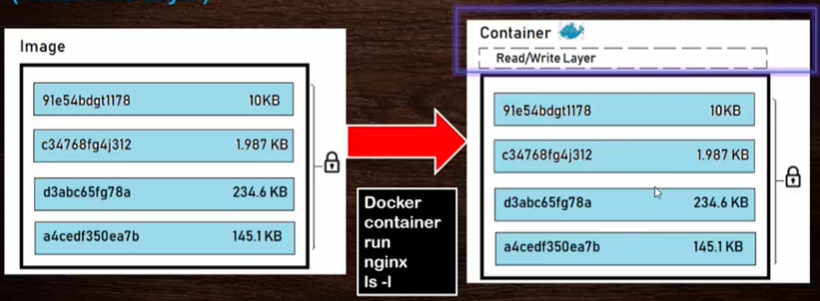
O dockerfile é um arquivo texto com instruções, comandos e passos que é executado através do comando build para gerar uma imagem.

Uma imagem é um binário que inclui todos os requisitos para a criação e execução de único contêiner do docker, bem como metadados que descrevem suas necessidades e capacidades, incluindo o código do aplicativo dentro do contêiner e suas configurações.

Docker build -t <imagem>



Ao criar o container, o Docker cria uma nova camada acima da última camada da imagem, e nessa camada podemos ler e escrever.



Instruções e Argumentos são interpretados linha a linha pelo engine do Docker para criação da imagem.

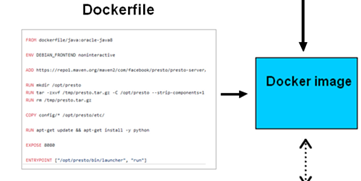


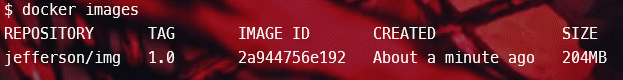
Figura : Processo de builder

Definindo as etapas que vamos usar para criar a imagem

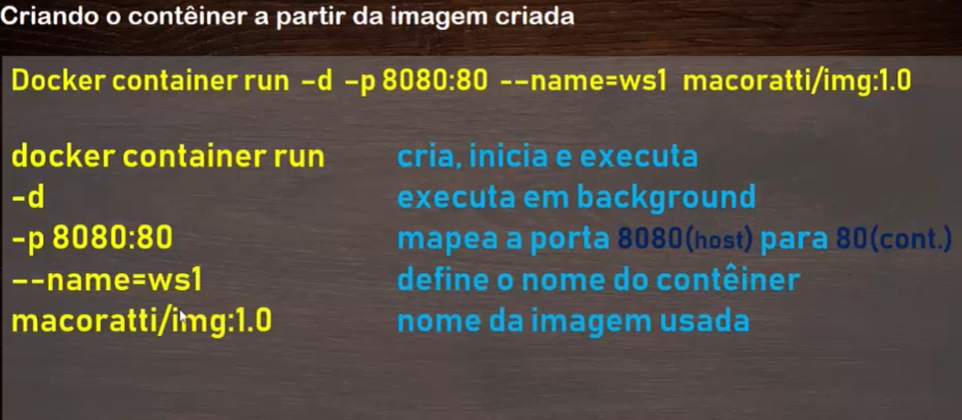
1. Definir uma imagem base.
2. Definir informações para a imagem.
3. Executar comandos para instalar e iniciar o nginx.
4. Expor qual porta o servidor vai atender.
5. Definir o ponto de entrada a aplicação
6. Definir a execução de um comando para inicializar o servidor nginx.

docker build -t jefferson/img:1.0 .

|  |  |
| --- | --- |
| Docker build | O commando que constroi a imagem |
| -t | Usado para informar que a imagem pertence ao meu usuário |
| jefferson/img:1.0 | O nome da imagem e a tag atribuída a imagem |
| . | Significa o diretório atual, onde está o dockerfile. |
|  |  |



* docker container run -d -p 8080:80 --name=ws1 jefferson/img:1.0



* docker login
* docker tag 2a944756e192(da img) 40509477(use dockerhub)/debian:1.0(img name and version)
* docker push 40509477/debian:1.0

