



# INTRODUÇÃO AO DOCKER



OLÁ!



Eu sou Marcelo Schirbel Gomes

Caso queira me encontrar:



[linkedin.com/in/mschirbel](https://www.linkedin.com/in/mschirbel)



[medium.com/@mschirbel](https://medium.com/@mschirbel)



## CONSIDERAÇÕES INICIAIS

### Perguntas:

Podem fazer perguntas quando quiserem.

Se eu não souber algo, farei questão de pesquisar e te enviar a resposta.

Se a resposta for um pouco longa, deixarei para o final

### Material:

Todo o material dessa aula é gratuito.

Podem distribuir com quem quiserem! Só deixem os créditos

Todo o código está no meu Github:  
[github.com/mschirbel/curso-docker](https://github.com/mschirbel/curso-docker)

1.

O QUE É DOCKER?



Docker é uma ferramenta feita  
para construir e implantar  
aplicações, usando *containers*.  
Também pode ser chamado de  
*Gerenciador de Containers*.



## ALGUMAS CARACTERÍSTICAS DO DOCKER

- ✗ Foi criado em 2013 pela Docker, Inc.
- ✗ Open source
- ✗ Pode ser usado em Linux, Windows ou MacOS

Código disponível em [github.com/docker](https://github.com/docker)



2.

COMO O DOCKER FUNCIONA?

# ARQUITETURA CLIENTE-SERVIDOR





# ARQUITETURA CLIENTE-SERVIDOR



# O QUE É UMA REST API?



REQUEST

RESPONSE



3.

# PROCESSOS DO DOCKER

## DOCKER CLIENT (DOCKER)





Socket é um arquivo  
que serve para trocar  
informações entre  
processos

# DOCKER DAEMON (DOCKERD)

DAEMON



OBJETO







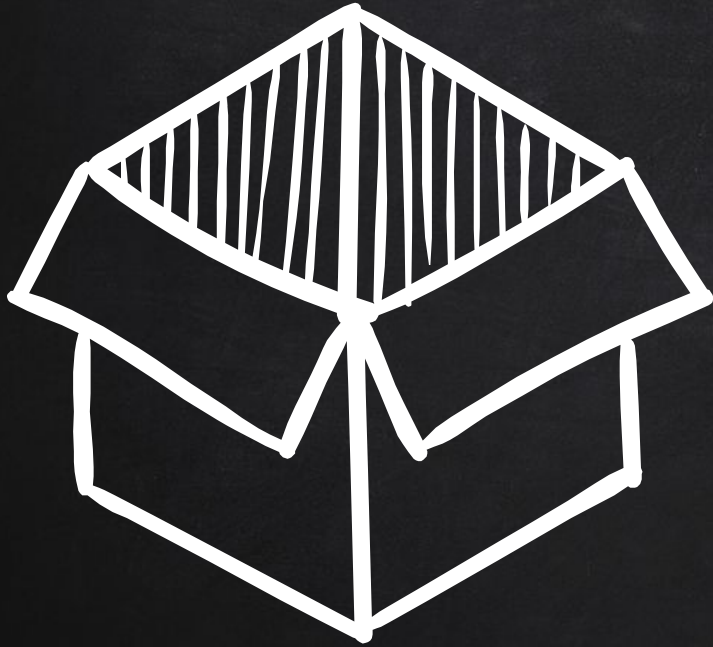
# IMAGENS



Imagens são  
templates para  
criação de  
containers. São  
composta de  
camadas.



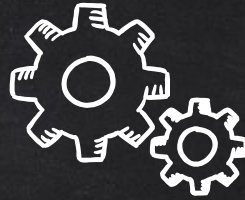
# CONTAINERS



Containers são  
instâncias de uma  
imagem. São, por  
padrão, isolados do  
host



IMAGE



DOCKER



CONTAINER



# REGISTRIES





Registries são  
serviços públicos,  
como o Docker Hub,  
ou privados, que  
armazenam imagens



A diagram illustrating the process of pushing a container image to a registry. On the left, a circle with concentric rings represents an image. On the right, a cloud shape represents a registry. A double-lined arrow points from the image to the registry.

IMAGE

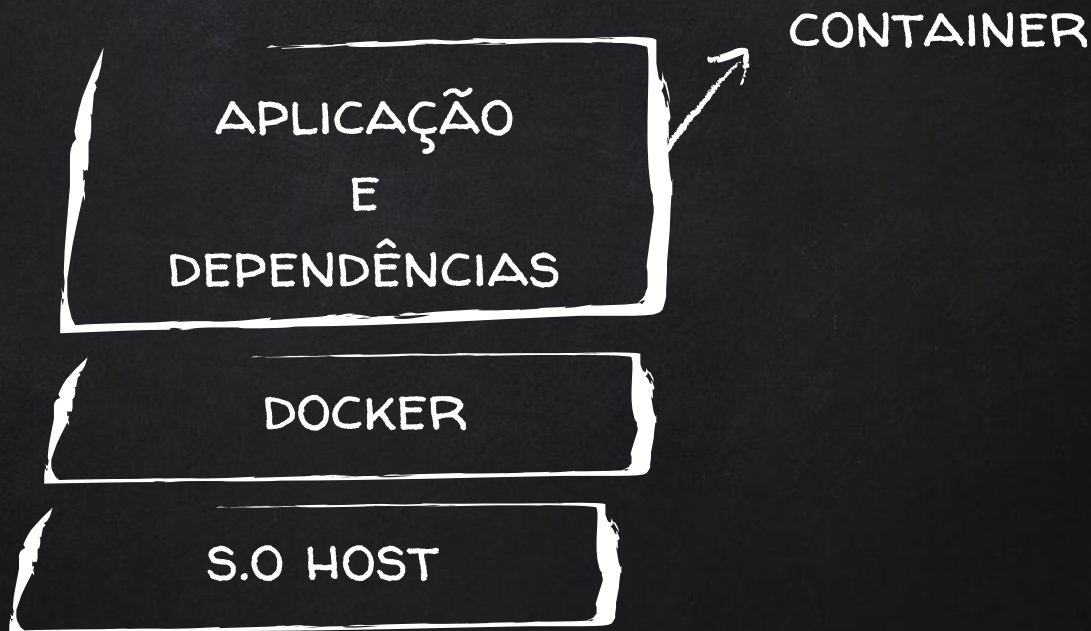


REGISTRY

# QUAL A DIFERENÇA ENTRE UMA MÁQUINA VIRTUAL E UM CONTAINER?



# QUAL A DIFERENÇA ENTRE UMA MÁQUINA VIRTUAL E UM CONTAINER?





# INSTALANDO O DOCKER

## TESTANDO A INSTALAÇÃO (CLIENT E DAEMON)

Para testar o client,  
precisamos saber se  
temos permissão na  
linha de comando:

```
$ docker -v
```

Para testar o  
daemon, precisamos  
saber se temos  
acesso a API:

```
$ docker ps
```





# NOSSO PRIMEIRO CONTAINER

DOCKER RUN HELLO-WORLD

9.

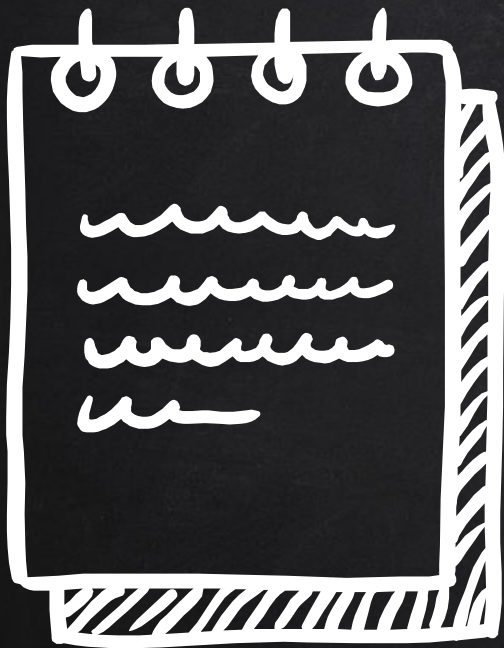
# MYSQL CONTAINER

# MYSQL CONTAINER

- Criar o container com:
  1. Um nome especial
  2. Senha de root
- Entrar no container
- Executar alguns comandos de MySQL
- Remover o container



# DOCKERFILE



Dockerfile é um  
arquivo onde  
podemos definir e  
preparar a imagem  
com um script de  
execução. Como se  
fosse uma receita de  
bolo.



## HTTPD CONTAINER

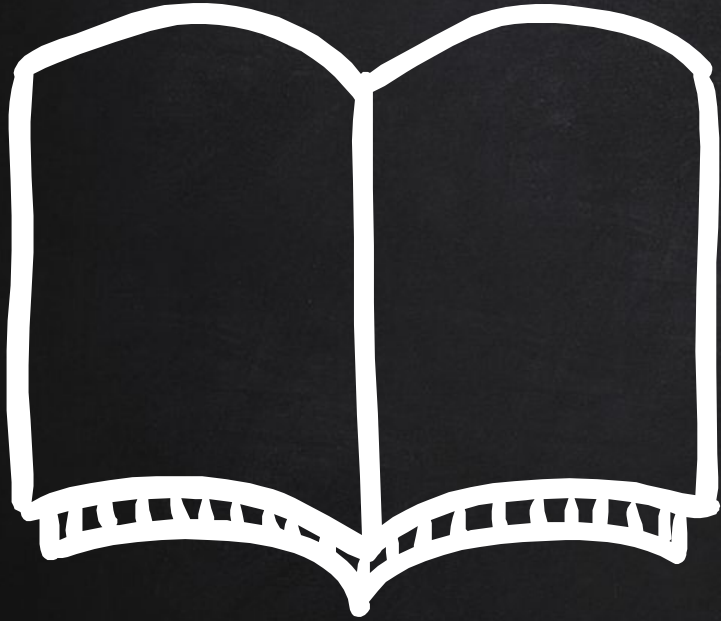
- Criar uma imagem com:
  1. Uma porta exposta
  2. Nosso index.html
- Criar o container a partir da nova imagem
- Verificar a página usando o IP do Host
- Verificar a página de dentro do container

## PROBLEMAS ENCONTRADOS ATÉ AGORA

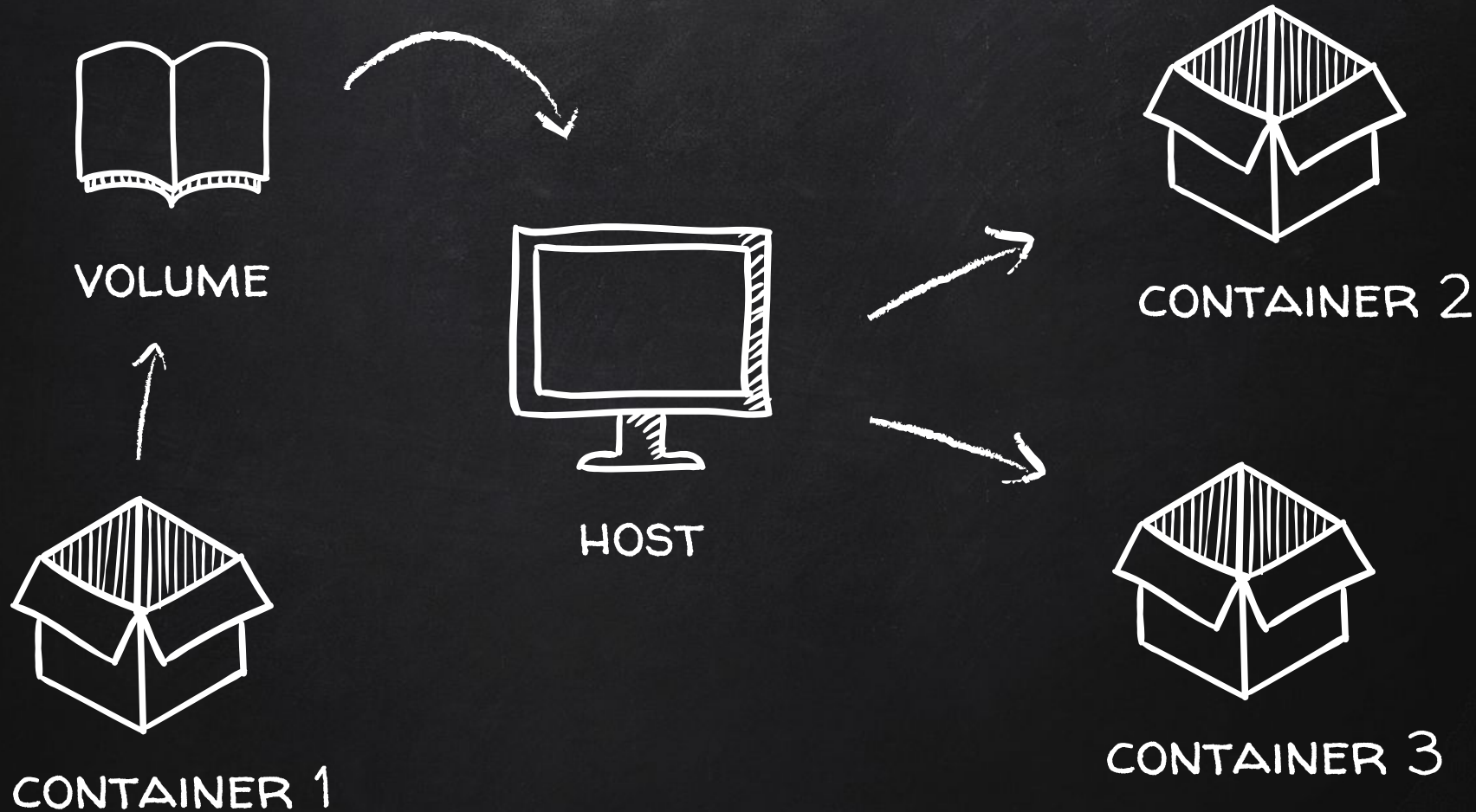
- A. PERDEMOS DADOS AO APAGAR CONTAINERS
- B. CONTAINERS ISOLADOS



VOLUMES



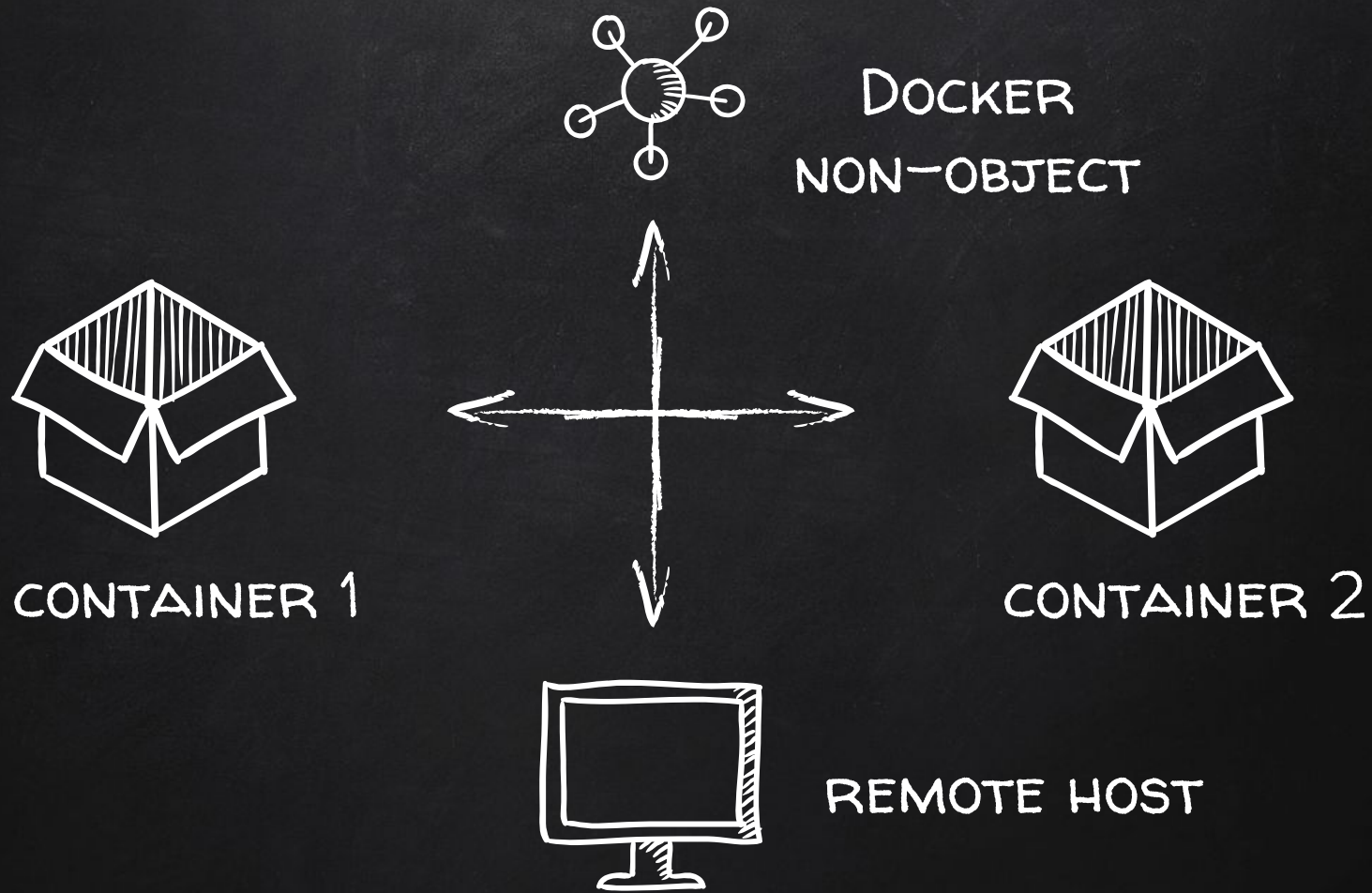
Volumes é o  
mecanismo que o  
Docker usa para  
persistir dados que  
estão em containers





NETWORK





13.

# DOCKER-COMPOSE



Docker Compose é  
uma ferramenta para  
auxiliar na  
construção de  
múltiplos containers.  
É escrito em YAML.

14.

LAMP STACK

# LAMP STACK

- Criar uma imagem para nosso Apache Web Server
- Criar uma imagem para nosso MySQL
- Criar um compose interligando os dois containers
- Subir nossa stack
- Verificar se nossos dados estão no container
- Remover nossa stack

15.

EXTRAS



DÚVIDAS?