

DOCKER

Essa baleia vai te conquistar



docker

- **DOCKER**
- **DOCKER IMAGES**
- **DOCKER CONTAINERS**
- **CONTAINERS VS VIRTUAL MACHINES**
- **DOCKERFILE**
- **DOCKER-COMPOSE**
- **DOCKER-MACHINE**
- **COMANDOS MAIS UTILIZADOS**

DOCKER

Docker é uma plataforma Open Source escrito em Go, que é uma linguagem de programação de alto desempenho desenvolvida dentro do Google, que facilita a criação e administração de ambientes isolados. (MundoDocker)

Uma plataforma aberta para desenvolvedores e administradores de sistemas, usada para construir, executar e distribuir “máquinas”. Uma vez que máquina mesmo é somente o host onde o docker é executado, ou seja, tudo roda de forma isolada na mesma máquina. (VivaOLinux)

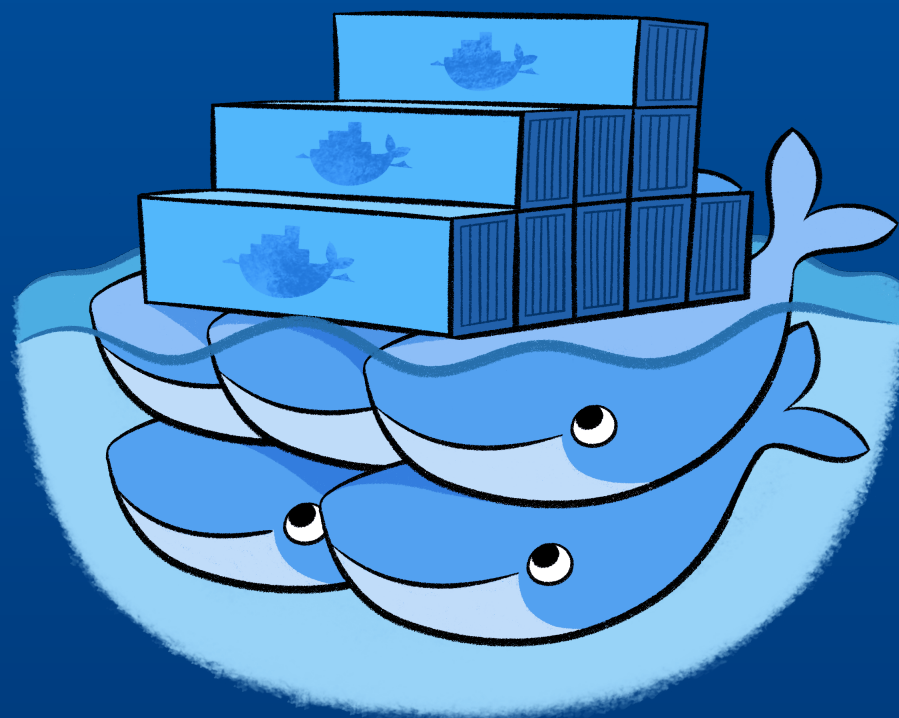


- DOCKER
- DOCKER IMAGES
- DOCKER CONTAINERS
- CONTAINERS VS VIRTUAL MACHINES
- DOCKERFILE
- DOCKER-COMPOSE
- DOCKER-MACHINE
- COMANDOS MAIS UTILIZADOS

DOCKER IMAGES

As docker images podem ser vistas como um template que é composto basicamente por um sistema operacional que pode possuir várias modificações antes de ser transformado em uma imagem.

As imagens são vistas como “**somente leitura**” e são utilizadas como base para a criação dos Docker Containers.



DOCKER IMAGES

O site “hub.docker.com” contém várias imagens já criadas e disponibilizadas com acesso gratuito.

- Exemplo: <https://hub.docker.com/u/culturagovbr/>

Com o comando “**docker pull [image:tag]**” é possível baixar uma imagem diretamente do hub para sua máquina e com o comando “**docker images**” é possível listar todas as imagens que já foram baixadas para sua máquina.

- Exemplo:
docker pull culturagovbr/salic-web
docker images



DOCKER IMAGES

Um dos pontos positivos de trabalhar com imagens é a possibilidade de ter containers sempre atualizados de acordo com as versões das imagens.

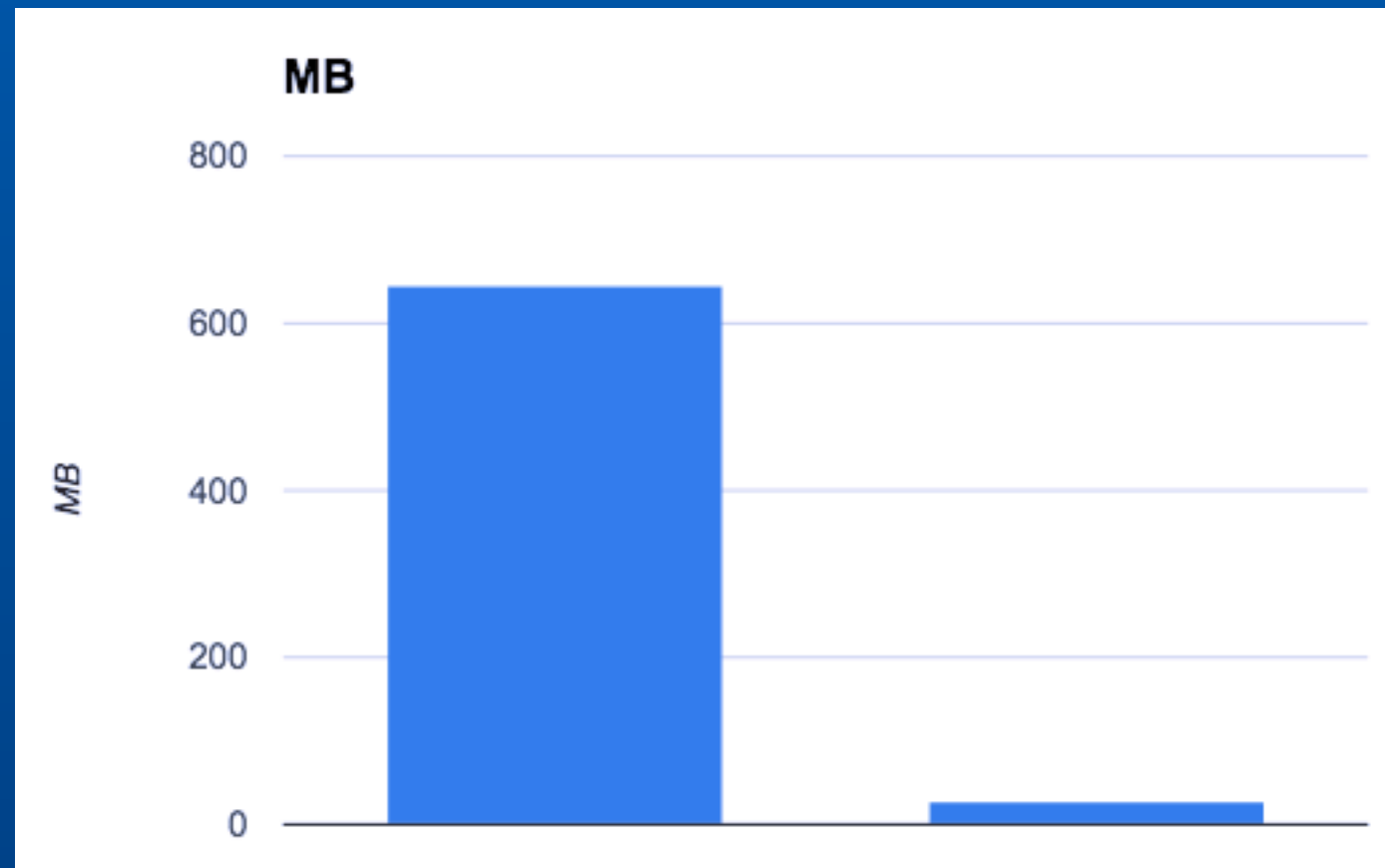
Outro ponto interessante é que o tamanho das imagens, mesmo sendo baseadas em sistemas operacionais, possuem o tamanho **menor**, comparado com outras plataformas como Vagrant, por exemplo.

Algumas imagens com o foco em micro-serviços para a geração de micro containers como a imagem “alpine”, por exemplo.



DOCKER IMAGES

Veja a diferença de uma imagem comum e uma micro imagem, somente com o NodeJS instalado.



- DOCKER
- DOCKER IMAGES
- DOCKER CONTAINERS
- CONTAINERS VS VIRTUAL MACHINES
- DOCKERFILE
- DOCKER-COMPOSE
- DOCKER-MACHINE
- COMANDOS MAIS UTILIZADOS

DOCKER CONTAINERS

Um docker container utiliza uma **docker image** como base e permite “**ler / escrever**”, possibilitando aplicar customizações e execução de comandos de acordo com a necessidade.

À partir de um container também é possível gerar **docker images**, basta executar o comando “**docker commit**” após realizar as mudanças no container.

Exemplo:

```
$ docker commit -m "Adição do nginx + dependências" -a "Vinícius Feitosa da Silva <viniciusfesil@gmail.com>" 0b2616b0e5a8 vinnyfs89/nginx_web_server:v2
```



DOCKER CONTAINERS

O comando “**docker ps**” serve para listar todos os containers que estão em execução. Já o comando “**docker ps -a**” lista todos os containers existentes, independente se estão em execução ou não.

Um container também pode ser gerado e executado a partir do comando “**docker run**”. Com esse comando é possível, definir parâmetros para a criação do container tais como :

- Compartilhamento de pastas
- Compartilhamento de portas
- Execução de comandos dentro do container à partir do host
- Entre outras coisas.



DOCKER CONTAINERS

A habilitação/desabilitação de um container pode ser feita usando os respectivos comandos “`docker start {id_container}`” ou “`docker stop {id_container}`”.

A execução de comandos diretamente dentro de um container pode ser realizada com o comando “`docker exec {id_container} {comando}`”.

Exemplo: `docker exec 123asd32mogsr echo “hello hello!”`



- DOCKER
- DOCKER IMAGES
- DOCKER CONTAINERS
- CONTAINERS VS VIRTUAL MACHINES
- DOCKERFILE
- DOCKER-COMPOSE
- DOCKER-MACHINE
- COMANDOS MAIS UTILIZADOS

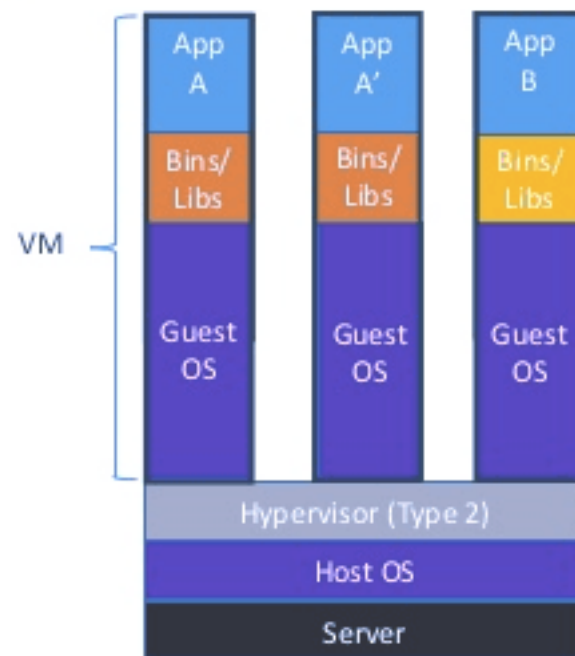
CONTAINERS VS VIRTUAL MACHINES

O Docker executa suas operações e trabalha diretamente com sistema operacional para gerir as imagens e containers, possibilitando mais flexibilidade para customização de ambientes e disponibilidade de recursos.

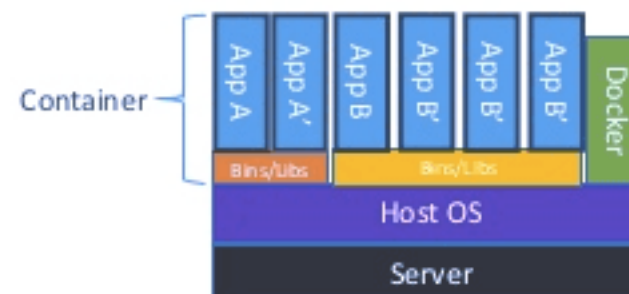


CONTAINERS VS VIRTUAL MACHINES

Containers vs. VMs



Containers are isolated, but share OS and, where appropriate, bins/libraries



- DOCKER
- DOCKER IMAGES
- DOCKER CONTAINERS
- CONTAINERS VS VIRTUAL MACHINES
- DOCKERFILE
- DOCKER-COMPOSE
- DOCKER-MACHINE
- COMANDOS MAIS UTILIZADOS

DOCKERFILE

O Dockerfile é um arquivo que contém um conjunto de instruções que possibilitam a criação de novas imagens e execução de atividades durante a geração das imagens.

Ele pode ser utilizado tanto diretamente com o “**docker build**” ou mesmo através do comando “**docker-compose**”, informando parâmetros importantes para sua geração.

Exemplo: `docker build -t imagemCultural:v1 .`

Exemplo2: `docker-compose up - - build -d`



DOCKERFILE

O Dockerfile é muito importante para garantir a clareza e rastreabilidade do que foi feito para ser gerada determinada imagem.

As chances de algo não ser replicado, ou executado, ou configurado na troca de uma imagem é muito grande quando essas configurações são realizadas diretamente no container para que no futuro vire uma imagem.

É uma boa prática versionar o Dockerfile no Github, pois no hub.docker.com é possível fazer uma referência dele para a determinada imagem/tag, mantendo todos atualizados.



DOCKERFILE

Exemplo de um Dockerfile do projeto GOG:

- <https://github.com/culturagovbr/docker-GOG/blob/master/Dockerfile>



- DOCKER
- DOCKER IMAGES
- DOCKER CONTAINERS
- CONTAINERS VS VIRTUAL MACHINES
- DOCKERFILE
- DOCKER-COMPOSE
- DOCKER-MACHINE
- COMANDOS MAIS UTILIZADOS

DOCKER-COMPOSE

O docker-compose é uma ferramenta do docker que possibilita gerir, vincular e customizar imagens e containers com mais facilidade.

Com ele podemos definir que um container “n1” depende ou tem ligação com um ou vários outros containers. Essas configurações são definidas em um arquivo chamado “**docker-compose.yml**” e executadas através do comando “**docker compose {ação}**”.



DOCKER-COMPOSE

Exemplo de um arquivo “**docker-compose.yml**”:

build:

context: ./dir

dockerfile: ./caminho/para/meu/Dockerfile-alternativo

hostname: meu_ambiente_local

ports:

- "80:80"

command: echo "/usr/local/apache2/bin/apachectl -f /usr/local/apache2/conf/httpd.conf" >> /etc/bash.bashrc

No mesmo local onde está o arquivo “**docker-compose.yml**”
basta executar o comando “**docker-compose {acao}**”

Exemplo: docker-compose up



- DOCKER
- DOCKER IMAGES
- DOCKER CONTAINERS
- CONTAINERS VS VIRTUAL MACHINES
- DOCKERFILE
- DOCKER-COMPOSE
- DOCKER-MACHINE
- COMANDOS MAIS UTILIZADOS

DOCKER-MACHINE

O docker-machine é um dos recursos mais úteis e talvez um dos mais importantes que fazem parte do docker.

Com o docker-machine é possível:

- Definir drivers de conexão
- Trabalhar com a virtualização de ambientes
- Alocação de containers e imagens diretamente no ambiente desejado



DOCKER-MACHINE

Deploy de container utilizando docker-machine:

- <https://youtu.be/clJxND1Riql>



- DOCKER
- DOCKER IMAGES
- DOCKER CONTAINERS
- CONTAINERS VS VIRTUAL MACHINES
- DOCKERFILE
- DOCKER-COMPOSE
- DOCKER-MACHINE
- COMANDOS MAIS UTILIZADOS

COMANDOS MAIS UTILIZADOS

- docker ps -a - lista a situação dos containers atuais
- docker images - lista as imagens disponíveis
- docker rm - remove um container
- docker rmi - remove uma imagem
- docker run - baixa, cria container e executa uma imagem local ou à partir do hub.docker.com
- docker start - inicia um container existente
- docker stop - para um container existente
- docker attach (image_id) --sig-proxy=false
 - * CTRL + P + Q = sair do terminal sem derrubar o container
 - * docker attach --sig-proxy=true
- docker-compose up -d --build
- docker-compose stop
- docker-compose build



DÚVIDAS?



MINISTÉRIO DA CULTURA – FUNDEP/ABC



VINÍCIUS FEITOSA DA SILVA



VINICIUSFESIL@GMAIL.COM



@VINNYFS89

