

Preparação do Ambiente

Advanced Institute for Artificial Intelligence – AI2

<https://advancedinstitute.ai>



Contêineres Docker

- ❑ <https://docker-curriculum.com/>
- ❑ <https://docs.microsoft.com/pt-br/visualstudio/docker/tutorials/docker-tutorial>

Agenda

- ☐ Conceitos e definições
- ☐ Arquitetura Docker
- ☐ Criação e manipulação de contêineres
- ☐ Construindo imagens personalizadas

Definição

- ☐ Segregação de processos no mesmo kernel;
- ☐ Isolamento máximo possível de todo o resto do ambiente;
- ☐ File Systems, criados a partir de uma “imagem”;
- ☐ Torna a reprodutibilidade muito mais fácil
- ☐ Conceitualmente semelhante a máquinas virtuais

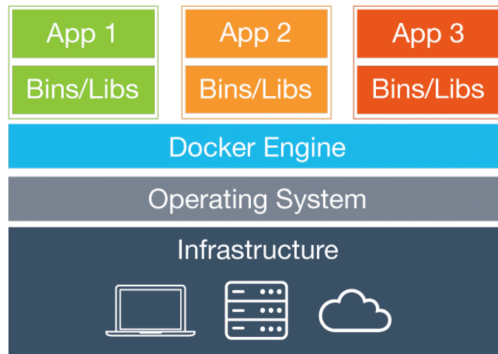


Figure: Aplicações utilizando Contêineres

Vantagens

- ☐ Leves porque não contêm um sistema operacional;
- ☐ Têm o mesmo desempenho que o código executado no sistema operacional host:
 - Até três vezes mais desempenho do que as máquinas virtuais, quando executados no mesmo hardware).
- ☐ Tempo de inicialização em milissegundos (em comparação com minutos para uma máquina virtual).
- ☐ Requerem pouca memória RAM
- ☐ São definidos usando código
 - Podemos tirar vantagem de sistemas de controle de código como o Git.
- ☐ Incentivam o reúso, para que você possa minimizar o retrabalho dispendioso.

Desvantagens

- Sempre são executados no sistema operacional Linux (os contêineres compartilham o sistema operacional do host):
 - Máquinas virtuais podem executar um sistema operacional diferente para cada máquina virtual
- Os contêineres usam isolamento no nível do processo
 - potencialmente menos seguro do que as máquinas virtuais totalmente isoladas

```
1
2 > docker run hello-world
3
4 Unable to find image 'hello-world:latest' locally
5 latest: Pulling from library/hello-world
6 b8dfde127a29: Pull complete
7 Digest: sha256:f2266cbfc127c960fd30e76b7c792dc23b588c0db76233517e1891a4e35
   7d519
8 Status: Downloaded newer image for hello-world:latest
9
10 Hello from Docker!
11 This message shows that your installation appears to be working correctly.
12
13 ...
14
15 Share images, automate workflows, and more with a free Docker ID:
16   https://hub.docker.com/
17
18 For more examples and ideas, visit:
19   https://docs.docker.com/get-started/
```


Conceitos

- ❑ **Imagem:** um conjunto estático de arquivos binários que armazenam todas as informações necessárias para iniciar um contêiner
- ❑ **Contêiner:** um sistema operacional isolado usando containerização (neste caso via Docker) para rodar em um sistema operacional host (neste caso MacOS)

Importante

É importante ressaltar que um contêiner é uma instância em execução de uma imagem.

Imagens

- ❑ Materialização de um modelo de um sistema de arquivos
- ❑ Produzido através de um processo de build;
- ❑ Representada por um ou mais arquivos e pode ser armazenada em um repositório como Github;
- ❑ O Docker utiliza file systems especiais para otimizar o uso, transferência e armazenamento das imagens, containers e volumes.
 - O principal é o AUFS, que armazena os dados em camadas sobrepostas, e somente a camada mais recente é gravável.

Arquitetura

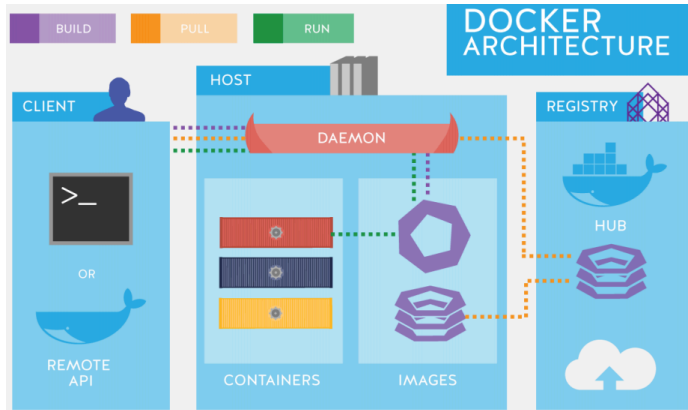


Figure: Arquitetura Docker

Parâmetros de Execução

❑ Execução Interativa:

- `--interactive` - isso nos permitirá digitar comandos de forma interativa no contêiner
- `--tty` - aloca um pseudo-TTY que permitirá que o contêiner imprima sua saída na tela.

```
1 > docker run --interactive --tty ubuntu:20.04 bash
2
3 Unable to find image 'ubuntu:20.04' locally
4 20.04: Pulling from library/ubuntu
5 a70d879fa598: Pull complete
6 c4394a92d1f8: Pull complete
7 10e6159c56c0: Pull complete
8 Digest: sha256:3c9c713e0979e9bd6061ed52ac1e9e1f246c9495aa063619d9d695fb803
   9aa1f
9 Status: Downloaded newer image for ubuntu:20.04
10 root@e2dd9abd3559:/#
```

Parâmetros de Execução

```
1 root@e2dd9abd3559:/# uname -r
2 4.19.121-linuxkit
3
4 root@e2dd9abd3559:/# cat /etc/lsb-release
5 DISTRIB_ID=Ubuntu
6 DISTRIB_RELEASE=20.04
7 DISTRIB_CODENAME=focal
8 DISTRIB_DESCRIPTION="Ubuntu 20.04.2 LTS"
```

Principais comandos:

- ❑ Listar Imagens:
 - `docker image ls`
- ❑ Listar Contêineres
 - `docker ps -a`
- ❑ Remover Contêineres/Imagens
 - `docker [image] rm 6f0684d58dca`
- ❑ Mapeamento de Volumes
 - `docker run -it -v [host]:[container] ubuntu bash`

Imagens Personalizadas

- ☐ A imagem é a abstração da infraestrutura em estado somente leitura, de onde será instanciado o contêiner.
- ☐ Imagens podem ser oficiais ou não oficiais
- ☐ As imagens oficiais são mantidas pela empresa docker e disponibilizadas no [docker hub](#);

Importante

- ☐ Todo contêiner é iniciado a partir de uma imagem
- ☐ Nunca teremos uma imagem em execução
- ☐ Um contêiner só pode ser iniciado a partir de uma única imagem

Imagens Oficiais



Figure: Imagens oficiais no Docker Hub

Imagens Personalizadas

- ❑ O objetivo das imagens oficiais é prover um ambiente básico;
 - um ponto de partida para criação de imagens pelos usuários
- ❑ As imagens não oficiais são mantidas pelos usuários que as criaram
- ❑ Nomeclatura:
 - Nome de uma imagem é composto por duas partes: *repositório* e *tag*;
 - `ubuntu:20.04`: repositório ubuntu e tag 20.04
 - Cada dupla `repositório:tag` representa uma imagem diferente;

Utilizando o comando `commit` para criação de imagem:

- ❑ Criando um contêiner e realizando alguma modificação na imagem padrão:

```
1 > docker container run -it --name ubuntu-fun ubuntu:20.04 bash
2 root@fb4cdc61c273:/# apt update
3 root@fb4cdc61c273:/# apt install cowsay fortune -y
4 root@fb4cdc61c273:/# exit
```

Utilizando o comando `commit` para criação de imagem:

- Agora vamos criar uma nova imagem utilizando o comando `commit`:

```
1 > docker container commit ubuntu-fun ubuntu:fun
2 sha256:d88fd6d76bcba0de4c4bb2304330d0b662a5cadd4db447ba83ae54fc6bba848c
3 > docker image ls
4 REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
5 ubuntu        fun       d88fd6d76bcb   4 seconds ago  149MB
```

Utilizando o comando `commit` para criação de imagem:

- ❑ Vamos testar a nova imagem:

```
1 > docker run --rm ubuntu:fun bash -c "/usr/games/fortune | /usr/games/  
    cowsay"  
2  
3 / Q: What's tan and black and looks great \  
4 \ on a lawyer? A: A doberman. /  
5 -----  
6      \      ^__^  
7      \      (oo)\_____  
8      (__)\"        )\\/\  
9           ||----w |  
10          ||     ||
```

Dockerfiles:

- ❑ Conjunto de instruções aplicadas em uma imagem para geração de outra;
- ❑ Controle de diferenças entre uma imagem (base), e a imagem que se deseja criar;

```
1 > cat Dockerfile
2 FROM ubuntu:20.04
3 RUN apt-get update && apt-get install cowsay fortune -y
4 COPY arquivo_teste /tmp/arquivo_teste
5 CMD bash
```

Diretivas do Dockerfile:

- ☐ **FROM:** informa qual a imagem base
- ☐ **RUN:** informa quais comandos serão executados durante a criação da imagem
- ☐ **COPY:** copia arquivos do *host* para a imagem;
- ☐ **CMD:** informa qual comando será executado por padrão, caso nenhum seja informado na inicialização do contêiner

Construção da imagem:

```
1  > docker image build -t ubuntu:fun .
2  Sending build context to Docker daemon    2.56kB
3  Step 1/4 : FROM ubuntu:20.04
4  ----> 26b77e58432b
5  Step 2/4 : RUN apt-get update && apt-get install cowsay fortune -y
6
7  ...
8
9  Step 3/4 : COPY arquivo_teste /tmp/arquivo_teste
10  ----> dea0a06e75d4
11  Step 4/4 : CMD bash
12  ----> Running in 9e38da969f42
13  Removing intermediate container 9e38da969f42
14  ----> 5a1f0278b1c3
15  Successfully built 5a1f0278b1c3
16  Successfully tagged ubuntu:fun
```

Outras Diretivas do Dockerfile:

- ❑ **ADD**: faz o mesmo que o **COPY**, porém permite que a cópia seja feita de uma URL
- ❑ **ENTRYPOINT**: Define um executável (e argumentos padrão) a ser executado quando o contêiner é iniciado - não é redefinido pela linha de comando
- ❑ **ENV**: Define variáveis de ambiente dentro da imagem;
- ❑ **WORKDIR**: Define o diretório de trabalho para qualquer **RUN**, **CMD**, **ENTRYPOINT**, **ADD** ou **COPY** subsequente;