

Introdução ao Docker

Professor: Pedro Horschulhack

Agenda

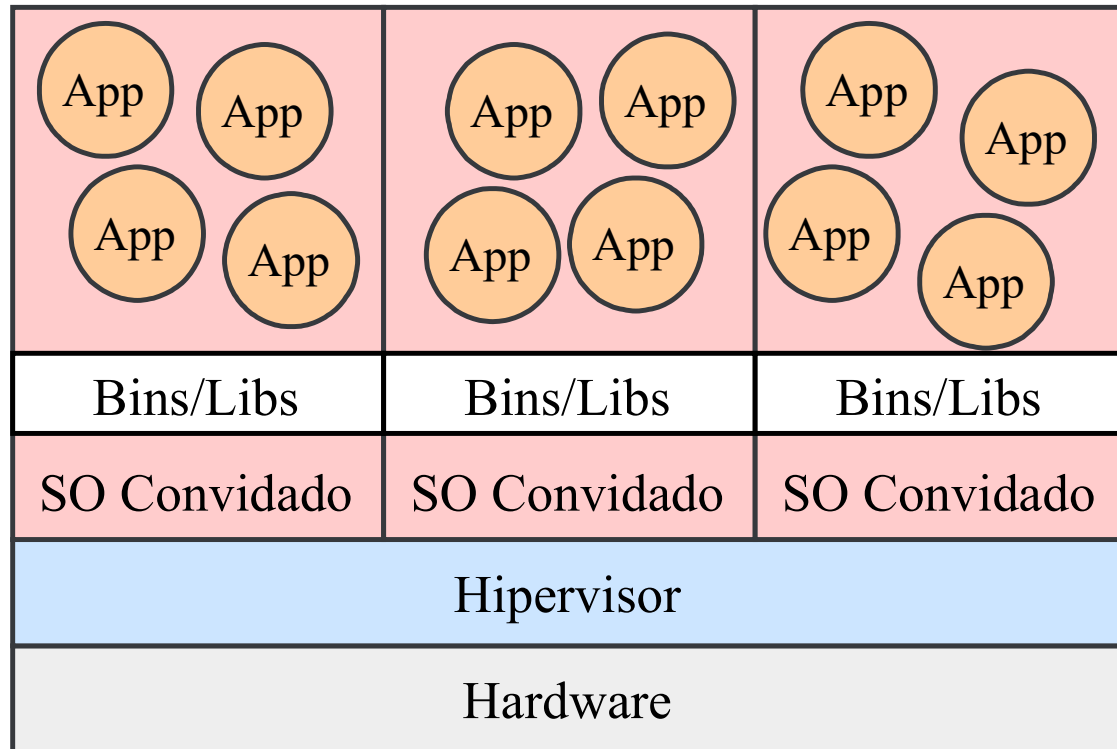
1. O que é o Docker?
2. Comandos básicos
3. Atividade

O que é o Docker?

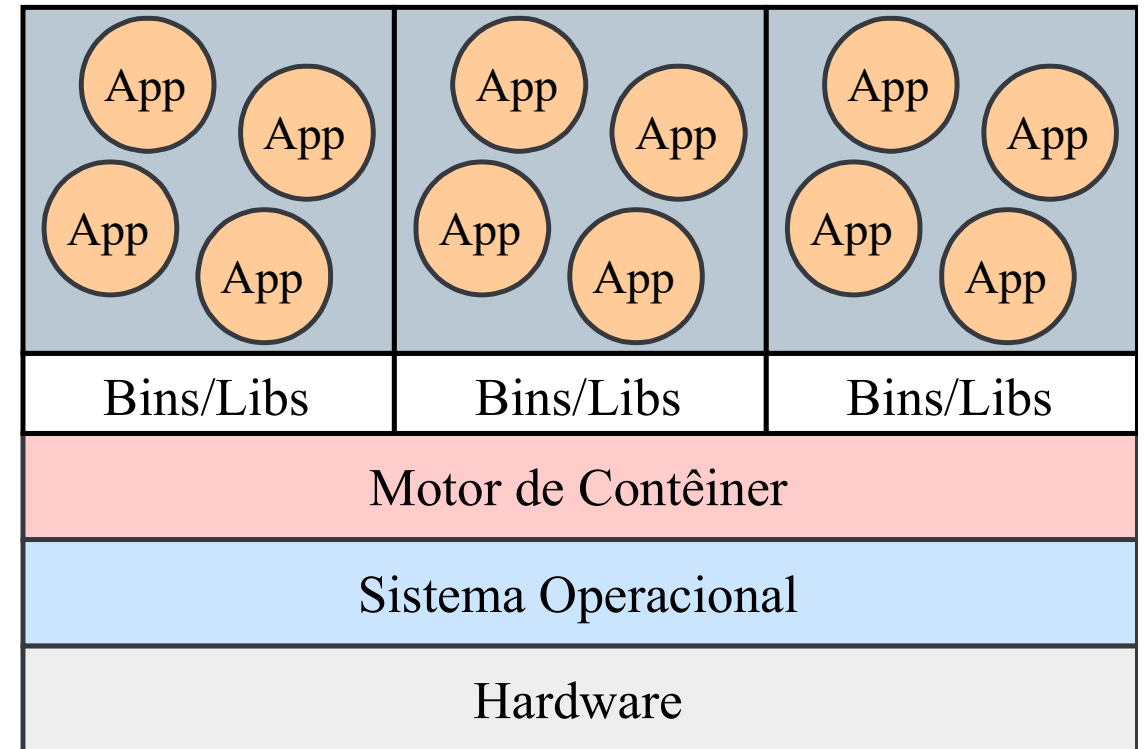
O que é o Docker?

- É uma plataforma de código aberto para o desenvolvimento, entrega e execução de aplicações, baseado em contêineres
- Permite você construir contêineres de forma simples e direta
 - Sem responsabilidade sobre a infraestrutura
- Conseguimos criar um ambiente todo dentro de um único processo
 - Bibliotecas
 - Arquivos
 - Ferramentas
 - etc

O que é o Docker?



Hypervisor nativo



Contêinerização

O que é o Docker?

- Como é feito o isolamento e a gestão de recursos (Linux)?
 - namespaces (isolamento)
 - cgroups (gestão de recursos)
- E no Windows?
 - Hyper-V

Como é a abstração e o isolamento?

- Namespaces é uma funcionalidade do kernel do Linux que isola processos
 - Recursos globais e suas alterações são visíveis somente para aqueles processos que **estão no mesmo namespace e invisíveis em outro namespace** [2]

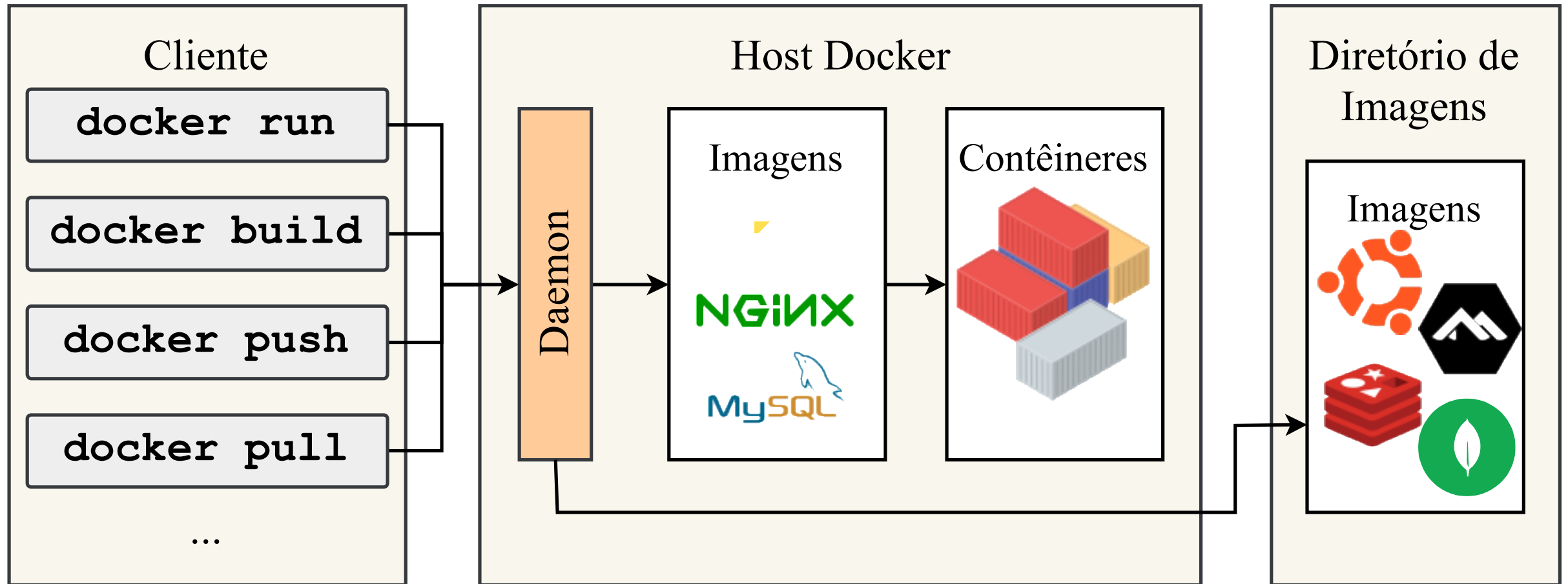
- Tipos:

- Cgroup
- IPC
- Rede
- Sistema de arquivos
- Tempo
- Usuário
- UTS

Como é a gestão de recursos?

- Cgroups é outra funcionalidade nativa do kernel do Linux para gerenciar os recursos entre os processos [1]
 - Estratégia de árvore hierárquica
- Controladores:
 - CPU
 - Memória
 - E/S
 - PID
 - *Cpuset*
 - Dispositivos
 - Misc

O que é o Docker?



O que é o Docker? – Importante lembrar

- **Imagem:** Conjunto de instruções que empacotam, de modo padronizado, como um contêiner deve executar (inclui bibliotecas, binários, arquivos auxiliares e configurações)
- **Container:** Processo isolado, em execução, contendo os itens empacotados por uma imagem específica
- **Diretório de imagens:** Repositório centralizado para armazenar e compartilhar imagens Docker

O que é o Docker? – Prática

- Vamos instalar o Docker na nossa máquina virtual
 - `sudo apt update && sudo apt install docker.io -y`
- Vamos permitir com que o nosso usuário consiga executar o docker sem ser usuário administrador
 - `sudo usermod -aG docker $USER`
 - Deslogue e logue novamente
- Agora vamos levantar um novo contêiner a partir da **imagem** `hello-world`
 - `docker run hello-world`

O que é o Docker? – Prática

```
ubuntu@arm:~$ docker run hello-world
```

```
Hello from Docker!
```

```
This message shows that your installation appears to be working correctly.
```

```
To generate this message, Docker took the following steps:
```

1. The Docker client contacted the Docker daemon.
2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
(arm64v8)
3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.
4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

```
To try something more ambitious, you can run an Ubuntu container with:
```

```
$ docker run -it ubuntu bash
```

```
Share images, automate workflows, and more with a free Docker ID:
```

```
https://hub.docker.com/
```

```
For more examples and ideas, visit:
```

```
https://docs.docker.com/get-started/
```

Comandos básicos

```
ubuntu@arm:~$ docker -h | head -n 18
Flag shorthand -h has been deprecated, please use --help

Usage:  docker [OPTIONS] COMMAND

A self-sufficient runtime for containers

Common Commands:
  run          Create and run a new container from an image
  exec         Execute a command in a running container
  ps           List containers
  build        Build an image from a Dockerfile
  pull         Download an image from a registry
  push         Upload an image to a registry
  images       List images
  login        Log in to a registry
  logout       Log out from a registry
  search       Search Docker Hub for images
  version      Show the Docker version information
  info         Display system-wide information
ubuntu@arm:~$
```

Comandos básicos

run: Cria e executa um contêiner a partir de uma **imagem**

Como executar: `docker run [OPÇÕES] <IMAGEM> [COMANDO]`

– Opções comuns (não restrito a):

- **i**: Modo interativo (mantém STDIN ativo mesmo sem estar atribuída ao processo)
- **t**: Aloca um pseudo-TTY a quem está executando o container
- **p**: Publica uma porta específica do container para o host (HOST:CONTAINER)
- **e**: Define variável de ambiente para o container
- **v**: Mapeia um arquivo/diretório no sistema de arquivos do host para o container
- **u**: Especifica o usuário do container
- **d**: Modo “detached” / não atribui o container à um TTY
- **-privileged**: Executa container em modo privilegiado (administrador/root)

Comandos básicos

exec: Executa um **comando** num contêiner em execução

Como executar: `docker exec [OPÇÕES] <NOME OU ID DO CONTÊINER> COMANDO`

– Opções comuns (não restrito a):

- **i**: Modo interativo (mantém STDIN ativo mesmo sem estar atribuída ao processo)
- **t**: Aloca um pseudo-TTY a quem está executando o container
- **e**: Define variável de ambiente para o container
- **u**: Especifica o usuário do container
- **-privileged**: Executa container em modo privilegiado (administrador/root)

Comandos básicos

ps: Lista contêineres que foram criados ou que estão em execução

Como executar: `docker ps [OPÇÕES]`

– Opções comuns (não restrito a):

- a: Lista todos os contêineres (criados e em execução)
- q: Lista somente os IDs dos contêineres
- n: Mostra os último n contêineres criados (inclui todos os estados)

Comandos básicos

build: Constrói uma nova imagem a partir de um arquivo Dockerfile

Como executar: `docker build [OPÇÕES] [CONTEXTO]`

– Opções comuns (não restrito a):

- t: Especifica a **tag** que a imagem terá
- f: Especifica o caminho onde está seu arquivo Dockerfile

Comandos básicos

kill: Mata um ou mais contêineres

Como executar: docker **kill** [OPÇÕES] [**NOME OU ID DO CONTÊINER...**]

Comandos básicos

top: Lista processos do contêiner

Como executar: `docker top [NOME OU ID DO CONTÊINER]`

Comandos básicos

stats: Informações de consumo de contêineres em execução

Como executar: `docker stats [OPÇÕES] [NOME OU ID DO CONTÊINER...]`

– Opções comuns (não restrito a):

- a: Mostra todos os contêineres (em execução e parados)
- format: Especifica formato da saída

Comandos básicos

images: Lista as imagens disponíveis no seu host

Como executar: `docker images [OPÇÕES] [NOME OU ID DO CONTÊINER...]`

– Opções comuns (não restrito a):

- a: Mostra todos os contêineres (em execução e parados)
- format: Especifica formato da saída

Comandos básicos

image: Operações sobre as imagens disponíveis no seu host

Como executar: `docker image [COMANDO]`

– Comandos mais frequentes:

- **history:** Mostra histórico de uma imagem
- **prune:** Remove imagens não utilizadas
- **pull:** Baixa imagem de um diretório
- **push:** Envia imagem à um diretório
- **ls:** Lista as imagens disponíveis no seu host
- **rm:** Remove uma ou mais imagens
- **inspect:** Mostra informações detalhadas sobre uma imagem

Atividade

Vamos criar um contêiner de uma aplicação web

1. A partir da imagem **nginx**, crie e execute um novo contêiner. Junto a isso, considere as seguintes opções:
 1. Modo “detached”
 2. Publique a porta 80 do contêiner à porta 8080 do host (8080:80)
2. Mostre o contêiner em execução e seus atributos (docker ps)
3. Mostre o consumo de recursos do contêiner (docker stats)
4. Acesse o seu contêiner através do IP da sua VM

Atividade

Vamos criar um contêiner de uma aplicação em shell

1. A partir da imagem **python:3.10**, crie e execute um novo contêiner. Junto a isso, considere as seguintes opções:
 1. Modo “detached”
 2. Publique a porta 80 do contêiner à porta 8080 do host (8080:80)
 3. Escolha um nome para o contêiner (--name)
2. Mostre o id/nome do contêiner
3. Acesse o shell do contêiner (docker exec)
4. Instale o pacote vim (apt install) e a biblioteca Flask (pip install)
 1. Crie uma aplicação web junto com o professor
5. Acesse o seu contêiner através do IP da sua VM

Referências

1. HEO, Tejun. **Control Group v2**. The Linux Kernel, 2015. Disponível em <<https://docs.kernel.org/admin-guide/cgroup-v2.html>>. Acesso em 30 de abril de 2024.
2. KERRISK, Michael. Namespaces(7) – Linux Manual Page. Linux Manual Page, 2023. Disponível em <<https://man7.org/linux/man-pages/man7/namespaces.7.html>>. Acesso em 30 de abril de 2024.

Contato: p.horchulhack@pucpr.br