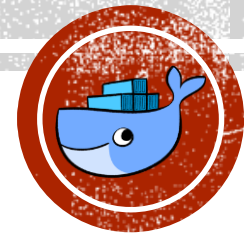


DOCKER E DOCKER-COMPOSSE

PARTE 1



Douglas Nassif Roma Junior

 /douglasjunior

 /in/douglasjunior

 smarppy.com

 douglas@smarppy.com

Slides: <https://git.io/fpgYO>

AGENDA

- Conceitos do Docker
- Imagens e Containers
- Containers e Máquinas Virtuais
- Preparando o Ambiente Docker
- Conclusão
- Referências

CONCEITOS DO DOCKER

CONCEITOS DO DOCKER

- Docker é uma plataforma para desenvolvedores e administradores de sistemas para auxiliar no desenvolvimento, implantação e execução de aplicações utilizando containers.
- O uso de containers Linux para implantar aplicações é chamado de “containerização”.
- Containers não são um conceito novo, mas a facilidade em implantar aplicações é.

CONCEITOS DO DOCKER

- A containerização está ficando cada vez mais popular, pois containers são:
 - **Flexíveis:** Até mesmo as aplicações mais complexas podem ser containerizadas.
 - **Leves:** Containers aproveitam e compartilham o mesmo kernel do host.
 - **Intercambiáveis:** Você pode atualizar um container *on-the-fly**.
 - **Portáveis:** Você pode construir localmente, implantar na nuvem e rodar em qualquer lugar.
 - **Escaláveis:** Você pode aumentar e distribuir automaticamente réplicas de seus containers.
 - **Empilháveis:** Você pode empilhar serviços verticalmente e *on-the-fly**.

* <https://www.maiovergara.com/on-the-fly-o-que-significa-esta-expressao/>

CONCEITOS DO DOCKER

- A containerização está ficando cada vez mais popular, pois containers são:
 - **Flexíveis:** Até mesmo as aplicações mais complexas podem ser containerizadas.
 - **Leves:** Containers aproveitam e compartilham o mesmo kernel do host.
 - **Intercambiáveis:** Você pode atualizar um container *on-the-fly**.
 - **Portáveis:** Você pode construir localmente, implantar na nuvem e rodar em qualquer lugar.
 - **Escaláveis:** Você pode aumentar e distribuir automaticamente réplicas de seus containers.
 - **Empilháveis:** Você pode empilhar serviços verticalmente e *on-the-fly**.



* <https://www.maiovergara.com/on-the-fly-o-que-significa-esta-expressao/>

IMAGENS E CONTAINERS

IMAGENS E CONTAINERS

- Uma **imagem** é um pacote executável que inclui tudo o que é necessário para executar o código da aplicação, um *runtime* (PHP, Node, Java, etc), bibliotecas, variáveis de ambiente e arquivos de configuração.
- Um **container** é iniciado quando você executa uma imagem, ou seja, **container** é uma instância de uma imagem – é o que a imagem se torna quando é executada.
- Você pode ver a lista de containers que estão sendo executados com o comando:

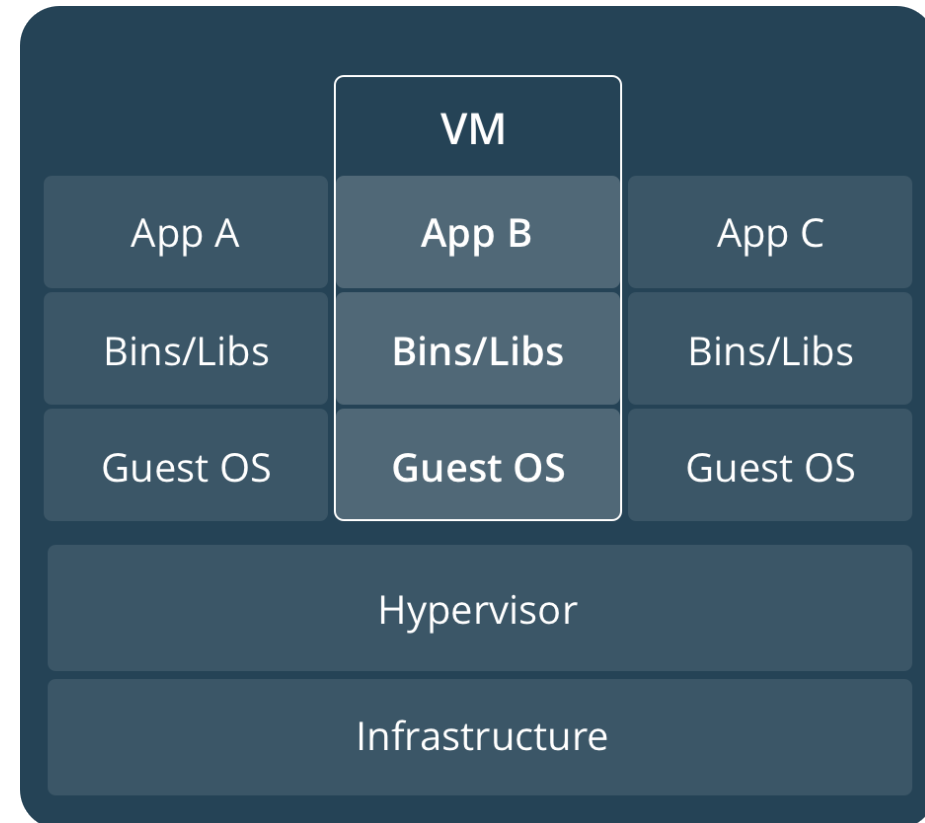
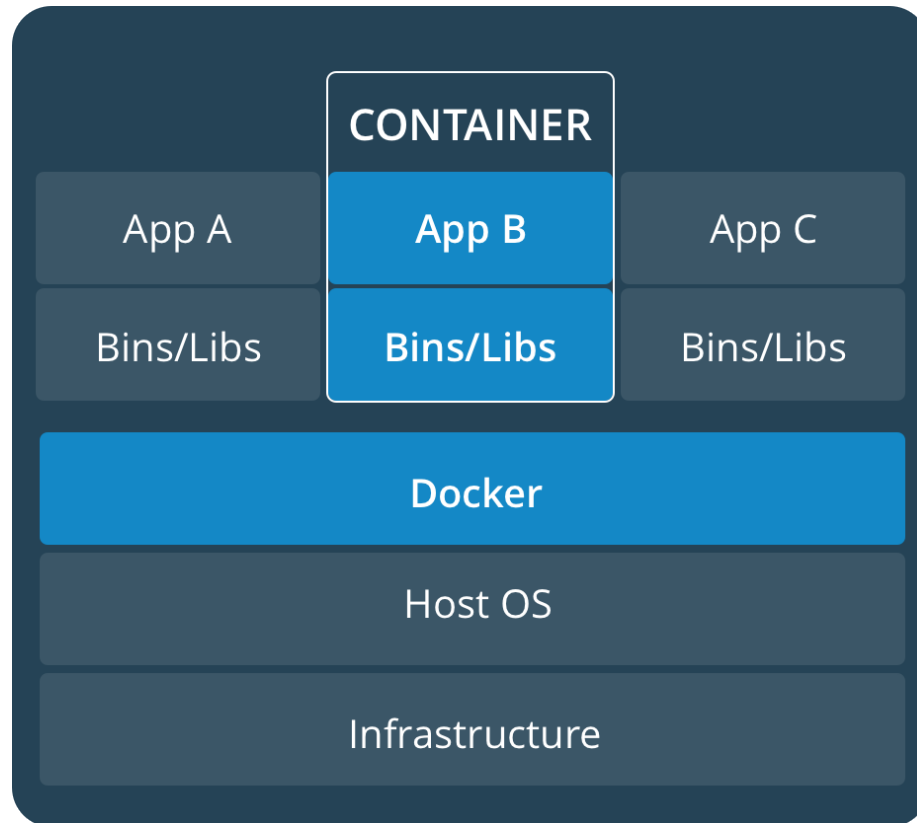
```
$ docker ps
```


CONTAINERS E MÁQUINAS VIRTUAIS

CONTAINERS E MÁQUINAS VIRTUAIS

- Um **container** roda “nativamente” em Linux e compartilha o kernel da máquina host com outros containers.
- Ele roda como um processo discreto, sem ocupar mais memória do que qualquer outro executável, fazendo com que seja leve.
- Em contraste, uma **máquina virtual (VM)** executa um novo sistema operacional, com acessos virtuais para a máquina host através de um *hypervisor*.
- Em geral, VMs fornecem um ambiente com mais recursos do que uma aplicação realmente precisa.

CONTAINERS E MÁQUINAS VIRTUAIS



PREPARANDO O AMBIENTE DOCKER

PREPARANDO O AMBIENTE DOCKER

- Instale a versão atual do **Docker Community Edition (CE)** ou **Enterprise Edition (EE)** através do site oficial, para a plataforma suportada.
- Mac OSX: <https://docs.docker.com/docker-for-mac/install/>
- Windows 10: <https://docs.docker.com/docker-for-windows/install/>
- Ubuntu: <https://docs.docker.com/install/linux/docker-ce/ubuntu/>
- CentOS: <https://docs.docker.com/install/linux/docker-ce/centos/>
- Outros: <https://docs.docker.com/install/>

PREPARANDO O AMBIENTE DOCKER

- Verificando a versão do docker:

```
$ docker --version
```

- Veja mais detalhes sobre a sua instalação do docker:

```
$ docker info
```

- No Linux, para evitar erros com permissões (e a necessidade do `sudo`), adicione o seu usuário no grupo docker: <https://docs.docker.com/install/linux/linux-postinstall/>

PREPARANDO O AMBIENTE DOCKER

- **Teste se a sua instalação funciona, rodando uma simples imagem `hello-world`.**

```
$ docker run hello-world
```

- **Liste as imagens que foram baixadas em sua máquina:**

```
$ docker image ls (ou docker images)
```

- **Liste os containers que foram criados em sua máquina. (para exibir os containers que estão parados utilize o `--all`)**

```
$ docker container ls -all (ou docker ps -a)
```

PREPARANDO O AMBIENTE DOCKER

Lista os comandos Docker CLI

docker

docker container **--help**

Exibe a versão do Docker

docker **--version**

docker version

docker info

Executa uma imagem Docker

docker run hello-world

Lista as imagens do Docker

docker image **ls**

Lista os containers do Docker (em execução, todos)

docker container **ls**

docker container **ls --all**

docker container **ls -aq**

CONCLUSÃO

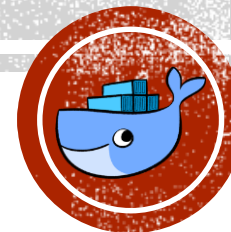
CONCLUSÃO

- **Containerização** torna o desenvolvimento e implantação de aplicações mais simples.
 - Sua aplicação não depende do sistema operacional
 - Atualizações podem ser enviadas para qualquer parte de uma aplicação distribuída
 - A densidade dos recursos pode ser otimizada
- Com **Docker**, escalar sua aplicação é uma questão de rodar novos executáveis, e não rodar pesadas máquinas virtuais inteiras.

REFERÊNCIAS

- Docker - <https://www.docker.com/>
- Documentação - <https://docs.docker.com/>
- Get Started - <https://docs.docker.com/get-started/>
- Instalação - <https://docs.docker.com/install/>

DÚVIDAS?



Douglas Nassif Roma Junior

 /douglasjunior

 /in/douglasjunior

 smarppy.com

 douglas@smarppy.com

Slides: <https://git.io/fpgYO>