

LUIZ ZENHA

POSTECH

SOFTWARE ARCHITECTURE

DOCKERIZAÇÃO

# AULA 01

---

## SUMÁRIO

O QUE VEM POR AÍ? .....	3
HANDS ON .....	4
SAIBA MAIS .....	5
O QUE VOCÊ VIU NESTA AULA? .....	11
REFERÊNCIAS .....	12
PALAVRAS-CHAVE .....	13

EMSE

## O QUE VEM POR AÍ?

Vamos aprender o que é o Docker, containers e virtualização, e entender como funciona essa ferramenta que facilita tanto o nosso ambiente de desenvolvimento, como também a publicação das nossas aplicações em ambiente de produção, simplificando principalmente a questão do provisionamento de infraestrutura.

Além de entender como funciona a parte teórica, vamos praticar criando e aprender os principais comandos para interagir com os containers.

## HANDS ON

Vamos para o nosso Hands On! Cada comando que executarmos nos ajudará a aprender ainda mais na prática.

Na primeira aula, veremos o setup e configuração inicial do Docker e os primeiros comandos que, com certeza, serão a base para as próximas aulas.

EMAND

## SAIBA MAIS

O Docker é um sistema para desenvolvimento, distribuição ou execução de aplicações. Ele possui um conjunto de funcionalidades que permite que você separe sua aplicação da dependência de infraestrutura, e para isso utiliza-se do conceito de virtualização.

Nas virtualizações convencionais, instalamos um software como o VirtualBox ou VMWare, que gerenciará as máquinas virtuais, e para cada máquina, temos que instalar todo sistema operacional, provisionar o hardware virtual desejado e, se sua aplicação precisar de bibliotecas ou configurações específicas, é preciso realizar a instalação e configuração para cada máquina virtual.

Já no Docker, usamos o conceito de containers. A grande diferença dos containers é que eles compartilham o kernel do sistema operacional da máquina Host e, mesmo com esse compartilhamento, a construção dos containers faz com que cada execução seja isolada, garantindo que arquivos e processamentos não sejam compartilhados com outros containers, tornando a execução segura para sistemas e aplicações em ambientes produtivos.

Outra vantagem dos containers é que, por não precisar de um sistema operacional completo, eles são mais leves para serem executados e iniciam mais rapidamente. Isso permite que você possa executar mais serviços com uma mesma máquina, principalmente comparando as virtualizações tradicionais citadas anteriormente.

Na figura 1 – “Comparativo entre Virtual Machines e Containers”, podemos visualizar a diferença da arquitetura e estrutura entre virtual machines e containers:

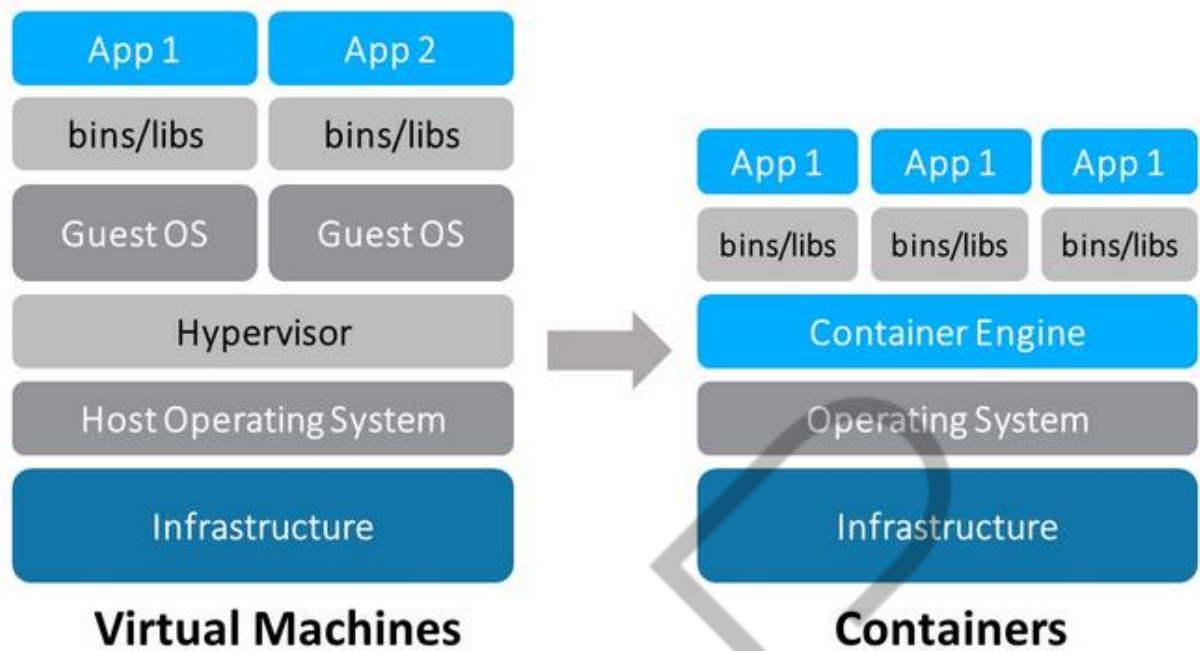


Figura 1– Comparativo entre Virtual Machines e Containers  
 Fonte: <https://medium.com/@IgorDuarte17/arquitetura-docker-c76cb14ffac6> (2022)

Existem diversas ferramentas para execução de containers. Vamos olhar mais a fundo como funciona o Docker, onde basicamente temos a seguinte separação entre Client, Host e Registry:

- Client: seria a forma de interação com o Docker para, por exemplo, iniciar ou parar um container via linha de comando. O Docker possui também a versão Docker Desktop que, por meio de uma interface, é possível interagir e fazer a gestão de container, imagens e configurações.
- Server: é o ambiente completo para executar os containers, imagens, rede, volumes e o daemon, que é responsável por todas as ações com o container, recebendo os comandos a partir do Client.
- Registry: aqui podemos definir como um repositório de imagens usadas pelo container. No caso do Docker, o registry é o Docker Hub, mas existem diversas outras opções como, por exemplo, algumas providas pelas próprias clouds, como o AWS ECR.

Como vimos, o Docker funciona recebendo comandos na máquina Host (Server) por meio do Client, podendo usar um Host local em nossas máquinas ou executar em uma máquina virtual, por exemplo, em uma instância na AWS.

## INSTALANDO O DOCKER

Para instalar o Docker localmente temos um processo de instalação específico para cada plataforma.

### Windows

No Windows, é necessário ter configurado o WSL 2 ou ter o Hyper-V habilitado para que seja possível virtualizar os containers. Recomenda-se, para uma melhor performance, a instalação e configuração do WSL 2.

**Para saber mais sobre a instalação do docker no windows, acesse a página:**  
<<https://learn.microsoft.com/pt-br/windows/wsl/install>>

Após a configuração, basta realizar o download do executável Docker Desktop diretamente no site do Docker <<https://www.docker.com>> e instalar clicando duas vezes sobre o arquivo exe baixado.

### Mac OS

Para o sistema Mac OS, fazemos o download do Docker Desktop no site do Docker <<https://www.docker.com>>. Atente-se que, para máquinas com chips Apple Silicon, é necessário baixar uma versão específica e, no caso de chips Intel, também existe a versão correta. Ao concluir o download, basta executar a instalação do arquivo dmg.

### Linux

No caso das diferentes distribuições Linux, existem os pacotes corretos para download. Oficialmente, no site do Docker, temos suporte para Debian, Ubuntu e Fedora. Como requisitos, é necessário que o módulo KVM esteja ativo e em execução.

**Para saber mais sobre a instalação do docker no Linux, acesse a página:**  
<<https://docs.docker.com/desktop/install/linux-install/>>

## COMANDOS BÁSICO DE DOCKER

Agora, vamos ver como utilizar o Docker em nosso ambiente local. Para isso, precisamos que o Docker Desktop esteja em execução; utilizaremos o terminal ou powershell para explorar alguns dos principais comandos do Docker, que nos ajudam a iniciar, parar e manipular os containers.

Para verificar a versão do docker, podemos executar o comando a seguir e, reparem que na saída deste comando, veremos a versão do Client e do Server, citados anteriormente.

```
$ docker version
```

Comando de prompt 1 – Verificando versão do Docker  
Fonte: Elaborado pelo autor (2023)

Como acabamos de instalar, não temos nenhum container em execução, mas para confirmar, executaremos o comando a seguir, para listar os containers:

```
$ docker ps
```

Comando de prompt 2 – Listar containers em execução  
Fonte: Elaborado pelo autor (2023)

Agora executaremos nosso primeiro container Linux e interagiremos com ele via linha de comando. O parâmetro -it permite acessar o container. Ubuntu é a imagem do container que vamos usar, latest é a tag da versão e /bin/bash é o ponto de entrada que queremos, no caso, abrir o terminal bash.



```
$ docker run -it ubuntu:latest /bin/bash
```

Comando de prompt 3 – Executar um container Ubuntu e interagir com ele via linha de comando  
Fonte: Elaborado pelo autor (2023)

Agora que abrimos o container, podemos visualizar a versão do sistema que estamos utilizando:

```
$ cat /etc/os-release
```

Comando de prompt 4 – Verificar a versão do SO do container em execução  
Fonte: Elaborado pelo autor (2023)

Temos total liberdade para instalar o que quisermos dentro de um container, pois estamos dentro de um Linux, e não em nossa máquina.

Em outra janela do terminal, podemos executar novamente o comando para listar os containers em execução e teremos a seguinte saída:

```
$ docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
STATUS	PORTS	NAMES	
dbd8f7a33be7	ubuntu:latest	"/bin/bash"	2 seconds ago
Up 2 seconds		admiring_blackwell	

Comando de prompt 5 – Listar containers em execução  
Fonte: Elaborado pelo autor (2023)

Para parar o container, precisamos pegar o CONTAINER ID que vimos anteriormente e executar o seguinte comando:

```
$ docker stop dbd8f7a33be7
```

Comando de prompt 6 – Parar um container  
Fonte: Elaborado pelo autor (2023)

Ao dar 'stop', o container apenas é desligado, mas o espaço que foi alocado ainda continua em disco. Para liberar e apagar o container, podemos executar o comando para listar os containers, com o parâmetro -a para exibir containers parados:

```
$ docker ps -a
```

Comando de prompt 7 – Listar containers parados  
Fonte: Elaborado pelo autor (2023)

E com o CONTAINER ID podemos excluir o container:

```
$ docker rm dbd8f7a33be7
```

Comando de prompt 8 – Apagando container  
Fonte: Elaborado pelo autor (2023)

E, para conferir a exclusão, executamos o Docker ps -a e vemos que mais nenhum container é listado.

## O QUE VOCÊ VIU NESTA AULA?

Nesta aula aprendemos sobre Docker, o que é, como usamos, a estrutura básica dos containers e suas diferenças com a virtualização de máquinas virtuais.

Vimos também como o instalar em diversos sistemas operacionais e os comandos básicos que são utilizados frequentemente para quem utiliza containers diariamente.

Não se esqueça de que estamos disponíveis no Discord para conversar sobre o tema desta aula, tirar dúvidas, fazer networking e muito mais! Aproveite!

## REFERÊNCIAS

DOCKER. **Documentação Get-Started**. 2023. Disponível em: <<https://docs.docker.com/get-started/overview/>>. Acesso em: 13 mar. 2023.

DUARTE, I. Arquitetura Docker. **Medium**. 27 de fevereiro de 2020. Disponível em: <<https://medium.com/@IgorDuarte17/arquitetura-docker-c76cb14ffac6>>. Acesso em: 13 mar. 2023.

ESPINOSA, A. **Docker High Performance**. [S.l]: Packt Publishing, 2019.

ROMERO, D. **Containers com Docker: Do desenvolvimento à produção**. [S.l]: Editora Casa do Código, 2015.

## **PALAVRAS-CHAVE**

Docker. Container. Arquitetura Container. Virtualização. Comandos Docker.

EMSE



POSTECH