



# **Adm. e Segurança de Redes de Computadores**

Docker  
C.T. Informática  
Prof. Vinícius Alves Hax



# Antes

- Instalação do servidor Apache no Linux



# Hoje

- Docker



# O que é Docker?

- Ferramenta de gerência de containers
- Mas o que são containers?
  - Containers são “máquinas virtuais leves”.  
Funcionam como máquinas virtuais (modo texto) mas usam muito menos recursos



# Para que serve um container?

- Criar e compartilhar servidores de desenvolvimento
- Gerenciar microsserviços (aplicações compostas por vários serviços/servidores com um propósito único)

# Popularidade de docker

● docker  
Termo de pesquisa

● virtual box  
Termo de pesquisa

+ Adicionar comparação

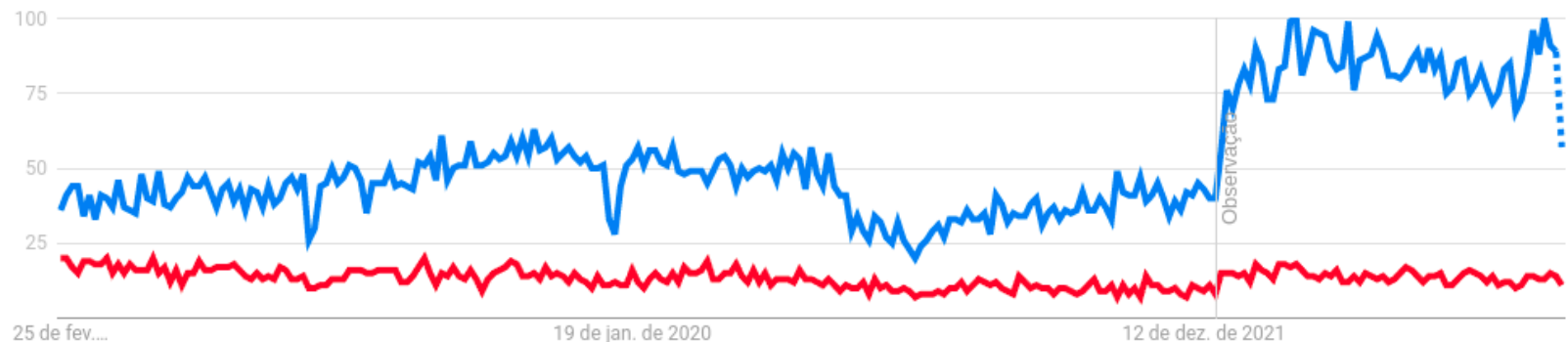
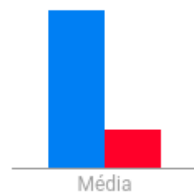
Brasil ▼

Nos últimos 5 anos ▼

Todas as categorias ▼

Pesquisa na Web ▼

Interesse ao longo do tempo ⓘ



# Restrições do docker

- Para poupar recursos o docker usa parte do sistema operacional do hospedeiro.
  - Por isso hospede/hospedeiro tem que ser sistemas operacionais próximos (Geralmente as imagens são Linux)

# Docker na prática

- “docker for beginners”
  - <https://docker-curriculum.com/>
- Atenção: dependendo da forma que o docker for instalado pode ser necessário adicionar sudo antes de cada comando docker. Ex: ao invés de  
  
docker run busybox  
  
execute  
  
sudo docker run busybox



- 
- Começando com docker

\$ docker pull busybox

- Esse comando baixa uma imagem docker chamada busybox




\$ docker images

- Lista as imagens disponíveis

\$ docker run busybox

- Executa a imagem docker busybox
- O espero é que o comando acima não produza nenhuma saída. Isto acontece pois o comando acima inicia um container docker mas o encerra logo depois

- 
- Mostrando algo na tela  
\$ docker run busybox echo "Ola"
  - Deve mostrar "Olá".
  - O que aconteceu acima?
    - Executamos um container usando a imagem busybox
    - Enviamos ao container um comando (echo "Ola")
    - A saída do comando é mostrada fora do container

- 
- Para visualizar os containers que estão executando

`$ docker ps`

- (Não deve aparecer nada pois nenhum container está sendo executado naquele instante)

`$ docker ps -a`

- Com o parâmetro -a aparecem também execuções recentes do docker

# Executando docker interativamente

- Para que o container não encerre após a execução de um comando podemos usar a flag -it
- `docker run -it busybox sh`

Obs: sh é o interpretador de comandos


# Exercício

- Use o comando `docker run` para iniciar uma imagem interativamente
- Dentro desse shell
  - Verifique em qual pasta você está (`pwd`)
  - Crie um novo arquivo (`touch arquivo`)
  - Saia do shell (o do container)
- Use o comando `docker run` para iniciar outro container interativo
- Dentro desse shell execute o comando `ls`

# Recuperando arquivos

- Onde está o arquivo recém criado?
  - O docker run, por padrão, cria um novo container a cada execução
- Podemos encerrar um container finalizado com o comando docker start
- Execute “docker ps” para descobrir o ID do container (primeira coluna)

```
$ docker start IDDOCONTAINER
```

- 
- Depois do docker start com o docker ps podemos ver os containers rodando  
\$ docker ps
  - E voltar a um terminal dentro do container com  
\$ docker exec -it IDDOCONTAINER sh





# Criando nossa imagem Docker

- Voltando ao tutorial
- <https://docker-curriculum.com/#our-first-image>