

embarcadero®

Delphi Academy

Dicas rápidas, truques e técnicas



Apresentando **Containers e Docker** para Devs

Fernando Rizzato
Lead Software Consultant, Latin America

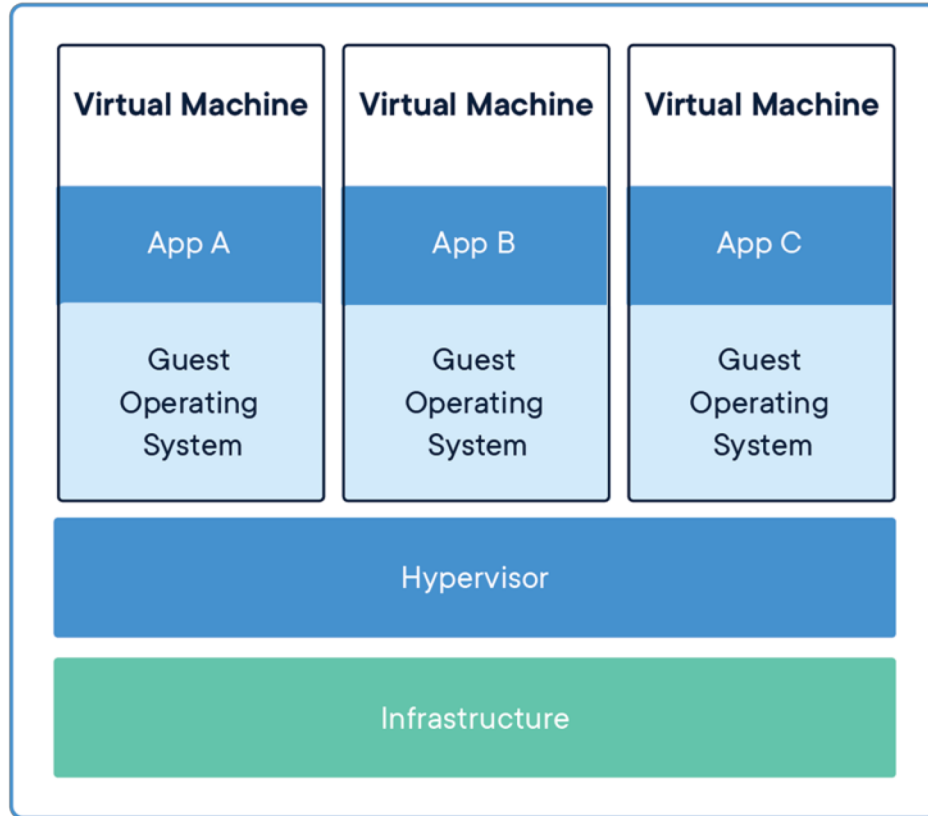
AGENDA

- Máquinas Virtuais vs Containers
- Tipos de Containers
- Docker
 - Conceitos básicos
 - Instalação e Configuração
 - Executando
- Recursos Adicionais

MÁQUINAS VIRTUAIS

- Máquinas Virtuais (VMs) são uma abstração de hardware físico, transformando um hardware em vários
- O hypervisor permite que várias VMs sejam executadas em uma única máquina
- Cada VM inclui uma cópia completa de um sistema operacional, aplicativos, etc., e são 100% isoladas
- Uma VM pode ocupar algumas dezenas de GBs
- As VMs também podem ser lentas para carregar
- VMWare, VirtualBox, Parallels, Xen, Hyper-V

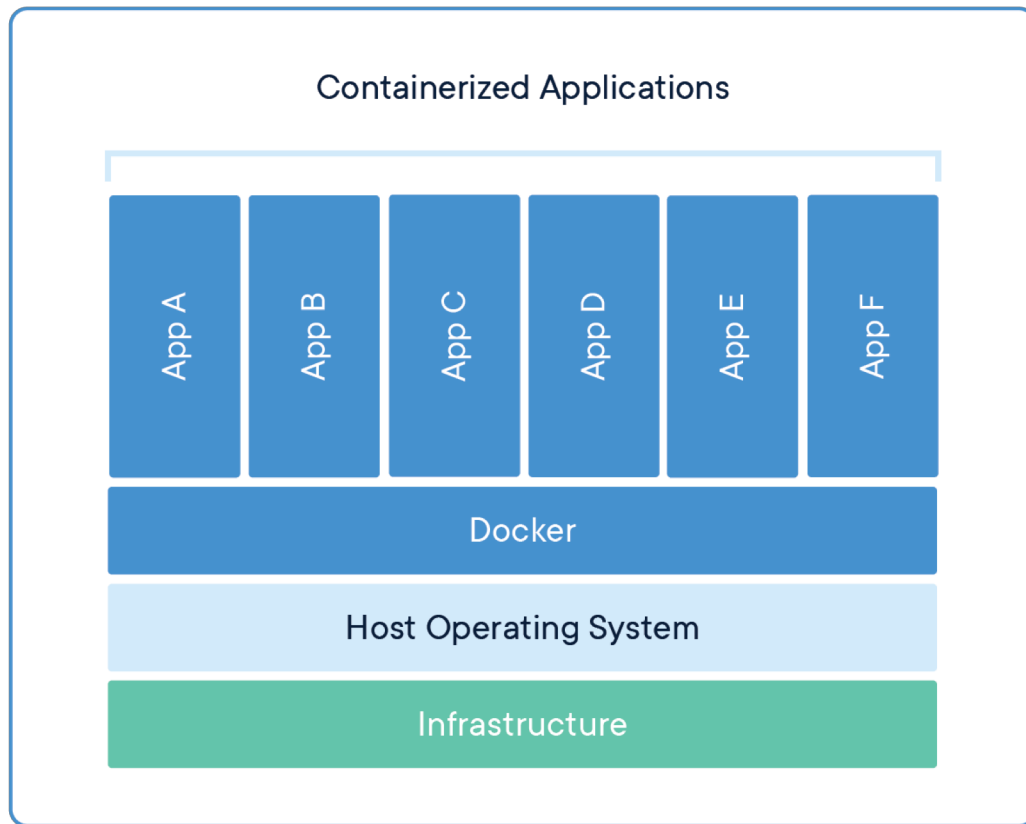
MÁQUINAS VIRTUAIS



CONTAINERS

- Containers são uma abstração na camada de aplicativo
- Vários containers podem ser executados no mesmo SO, e compartilhar o kernel do mesmo
- Cada container será executado como um processo isolado no espaço do usuário
- Os containers ocupam menos espaço que as VMs (dezenas de MBs)
- Containers podem ser executados dentro de VMs
- Docker, RKT, Mesos, LXC/LXD, OpenVZ

CONTAINERS



MÁQUINAS VIRTUAIS VS CONTAINERS

- Considere um sistema hipotético de 10 GB
- No modelo de VMs, teremos estes 10 GB multiplicados pelo número de instâncias necessárias
- No modelo de Containers você compartilha uma mesma imagem com centenas de instâncias
- Obviamente há prós e contras em cada modelo
- Se o requisito é isolamento completo, então VM parece ser o melhor caminho
- Se isolar processos é suficiente, então Containers são mais escaláveis e rápidos de configurar (ou provisionar)

TIPOS DE CONTAINERS

- Enquanto “Docker” virou sinônimo de containers, isso não significa que é o único tipo disponível.
- Containers de Sistema (Linux somente)
 - LXC, LXD - <https://linuxcontainers.org/>
 - Esse tipo de container faz com que um único sistema Linux execute como muitos
 - São mais “parecidos” com VMs com relação a gestão de recursos
- Containers de Aplicação (Linux, Windows, MAC)
 - Docker (<https://www.docker.com>), RTK (<https://coreos.com/rkt>)
 - A ideia básica é que cada aplicação rode em um container

DOCKER – CONCEITOS BÁSICOS

- Um **container** é iniciado executando uma imagem
- Uma **imagem** é um pacote que inclui tudo o que é necessário para executar um aplicativo
- Logo, um **container** é uma instância de uma **imagem**
- Usuários do Docker armazenam imagens em repositórios **privados** ou **públicos**
- Docker-Hub é um **registro** do Docker imagens publicas e privadas (<https://hub.docker.com>)

DOCKER – CONCEITOS BÁSICOS

- `docker run` – Runs a command in a new container
- `docker start` – Starts one or more stopped containers
- `docker stop` – Stops one or more running containers
- `docker build` – Builds an image from a Dockerfile
- `docker pull` – Pulls an image or a repository from a registry
- `docker push` – Pushes an image or a repository to a registry
- `docker export` – Exports a container's filesystem as a tar archive
- `docker exec` – Runs a command in a run-time container
- `docker search` – Searches the Docker Hub for images
- `docker attach` – Attaches to a running container
- `docker commit` – Creates a new image from a container's changes

DOCKER - INSTALAÇÃO

- Docker Desktop for Windows
 - <https://docs.docker.com/docker-for-windows/install/>
- Docker Desktop for macOS
 - <https://docs.docker.com/docker-for-mac/install/>
- Docker Server
 - <https://docs.docker.com/install/linux/docker-ce/ubuntu/>
 - <https://docs.docker.com/install/linux/docker-ce/fedora/>

DOCKER - PRIMEIRA IMAGEM

- `docker run --name my-redis -it ubuntu:latest bash`
- `apt-get update`
- `apt-get install wget`
- `apt-get install build-essential tcl8.5`
- `wget http://download.redis.io/releases/redis-stable.tar.gz`
- `tar xzf redis-stable.tar.gz`
- `cd redis-stable, make, make install`
- `./utils/install_server.sh`
- `service redis_6379 start`

DOCKER – PRIMEIRO DOCKERFILE

- Em um novo folder, criar um arquivo nomeado como Dockerfile com o seguinte conteúdo:

```
FROM ubuntu:latest
RUN ln -fs /usr/share/zoneinfo/America/Sao_Paulo /etc/localtime
RUN apt-get update
RUN apt-get install -y wget
RUN apt-get install -y build-essential tcl8.5
RUN wget http://download.redis.io/releases/redis-stable.tar.gz
RUN tar xzf redis-stable.tar.gz
RUN cd redis-stable && make && make install
RUN ./redis-stable/utils/install_server.sh
EXPOSE 6379
ENTRYPOINT ["redis-server"]
```

- `docker build -t redis .`
- `docker run -d -p 6379:6379 redis`

DEMOS

RECURSOS ADICIONAIS

Documentação

- <https://docs.docker.com/get-started/>
- <https://www.docker.com/resources/what-container>

Blogs

- <http://techgenix.com/getting-started-with-containers/>
- http://www.macoratti.net/19/01/intro_docker2.htm
- <http://prof.valiante.info/disciplinas/hardware/maquinas-virtuais-e-containers>
- <http://www.sinestec.com.br/blog/docker-fundamentos-conceito-e-arquitetura/>
- <https://blog.aquasec.com/a-brief-history-of-containers-from-1970s-chroot-to-docker-2016>
- <https://www.comparitech.com/net-admin/docker-vs-virtual-machines/>

OBRIGADO!

Perguntas?

Você pode me encontrar em:

@FernandoRizzato

fernando.rizzato@embarcadero.com

Siga-nos em

fb.com/DelphiBrasil

fb.com/EmbarcaderoBR