



Docker

SISTEMAS OPERACIONAIS - UFOP



Grupo



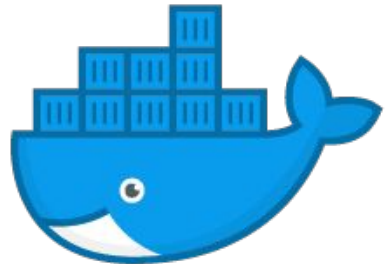
Gustavo Lucas



Philippe Lemos



O que é **Docker**



O Docker é uma tecnologia Open Source que permite criar, executar, testar e implantar aplicações distribuídas dentro de containers de software. Ele permite que você empacote um software de uma padronizada para o desenvolvimento de software, contendo tudo que é necessário para a execução: código, runtime, ferramentas, bibliotecas, etc.

O que são **containers**

Os containers são um método de virtualização em nível de sistema operacional que permite executar uma aplicação e suas dependências em processos com recursos isolados. Os containers permitem empacotar facilmente o código, as configurações e as dependências de uma aplicação em elementos fundamentais que oferecem consistência ambiental, eficiência operacional, produtividade de desenvolvedores e controle de versões.

Virtual Machine vs Docker

VM

O objetivo desse modelo é compartilhar os recursos físicos entre vários ambientes isolados, sendo que cada um deles tem sob sua tutela uma máquina inteira, com memória, disco, processador, rede e outros periféricos, todos entregues via abstração de virtualização.

É como se dentro de uma máquina física criasse máquinas menores e independentes entre si. Cada máquina dessa tem seu próprio sistema operacional completo, que por sua vez interage com todos os hardwares virtuais que lhe foi entregue pelo modelo de virtualização a nível de máquina.

Vale ressaltar que o sistema operacional instalado dentro de uma máquina virtual fará interação com os hardwares virtuais e não com o hardware real.

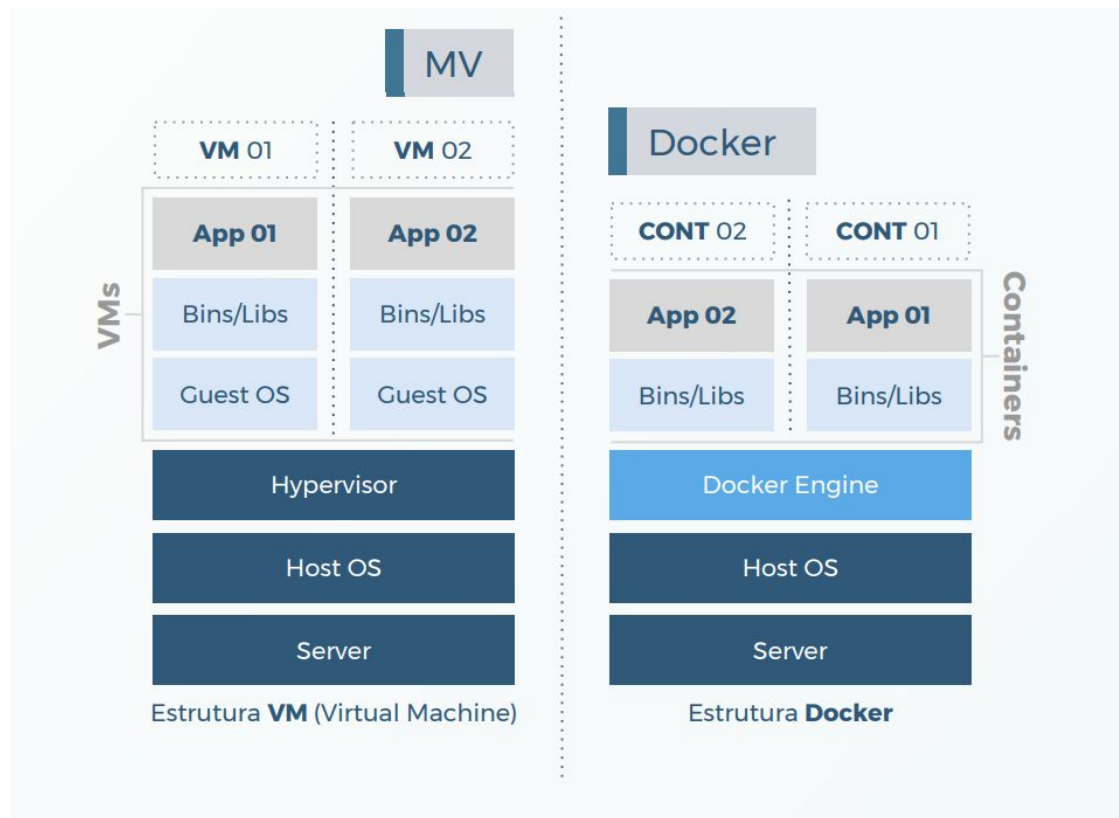
VM vs Docker

Docker

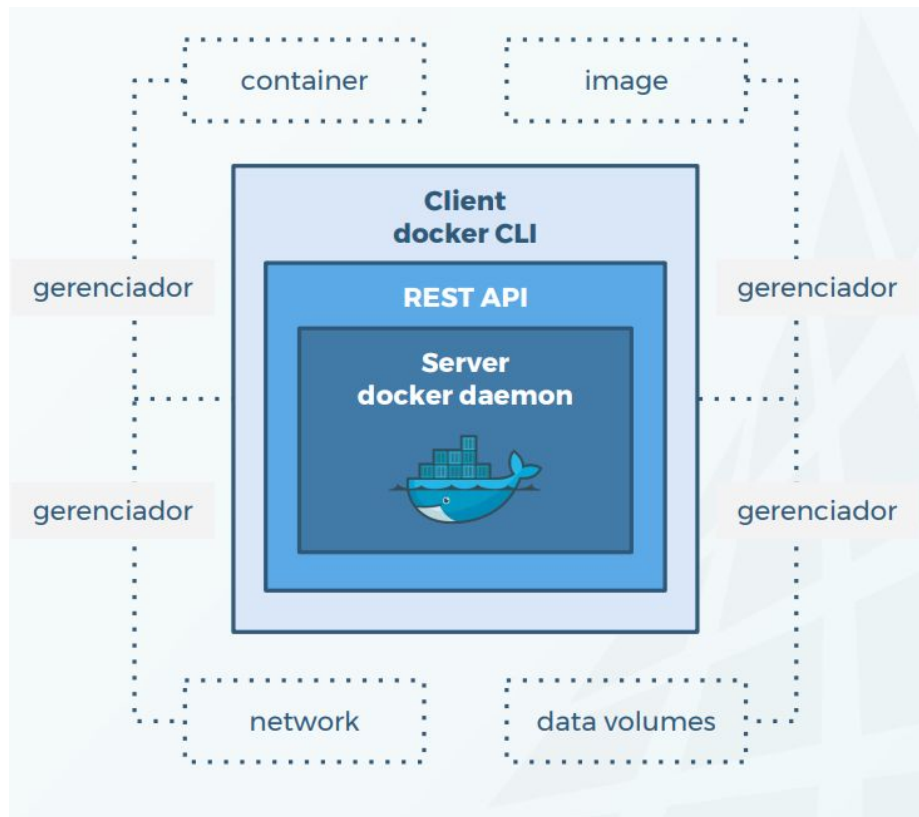
Esse modelo de virtualização está no nível de sistema operacional, ou seja, ao contrário da máquina virtual um container não tem visão de uma máquina inteira, ele é apenas um processo em execução em um kernel compartilhado entre todos os outros containers.

Ele utiliza o namespace para prover o devido isolamento de memória RAM, processamento, disco e acesso a rede, mesmo compartilhamento o mesmo kernel, esse processo em execução tem a visão de estar usando um sistema operacional dedicado.

Virtual Machine vs Docker



Partes principais do **Docker**



Docker usa uma arquitetura cliente-servidor.

A parte cliente fala com o Docker daemon, que faz o trabalho pesado de construção, execução e distribuição de seus containers e imagens Docker, também controla os recursos executados

Elementos do **Docker**

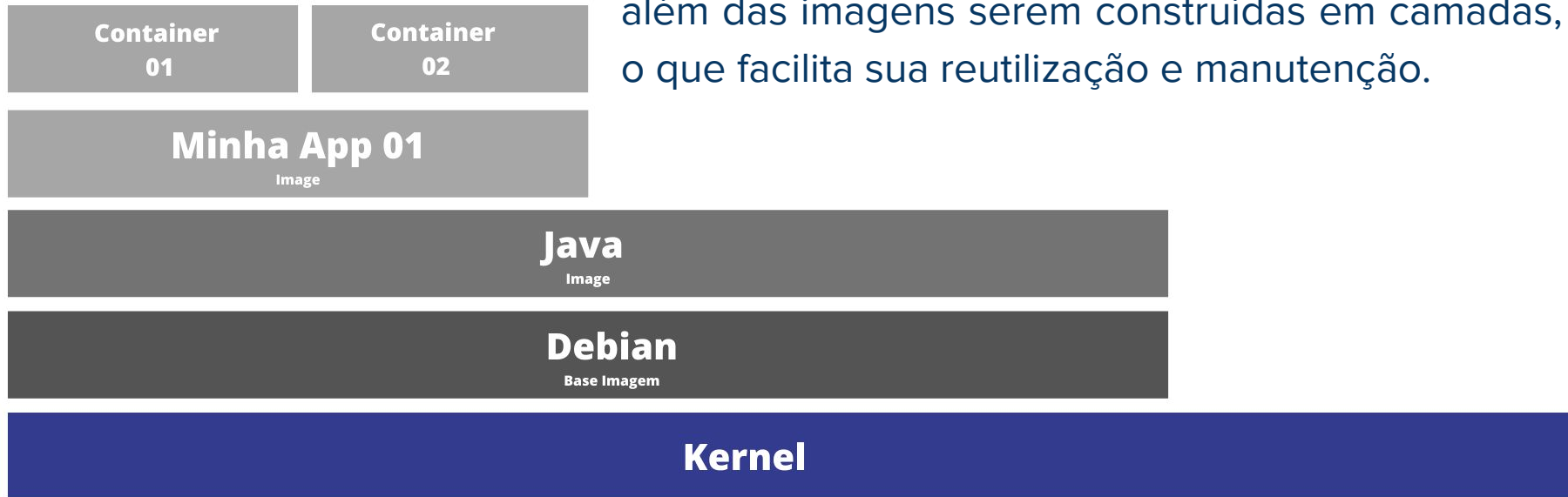
Containers docker - Containers tem como base sempre uma imagem, pense como na seguinte analogia do mundo Java, uma imagem é uma classe e um container é como um objeto instância dessa classe, então podemos através da imagem “instanciar” vários containers.

Algumas característica dos containers

- Portabilidade de aplicação
- Isolamento de processos
- Prevenção de violação externa
- Gerenciamento de consumo de recursos

Elementos do Docker

Imagens docker - Imagens são templates para criação de containers, para executá-las é necessário criar uma instância dela o “container”, além das imagens serem construídas em camadas, o que facilita sua reutilização e manutenção.



Elementos do Docker

Dockerfile - São scripts com uma série de comandos para criação de uma imagem, nesses scripts podemos fazer uma série de coisas como executar comandos sh, criar variáveis de ambiente, copiar arquivos e pastas do host para dentro da imagem etc.

Exemplo:

```
FROM gcc:4.9
MAINTAINER Gustavo Lucas
COPY . /usr/src/server
WORKDIR /usr/src/server
RUN g++ -o server *.cpp -lpthread -std=c++11
CMD ["/server"]
```

Elementos do **Docker**

Docker Registry - É como um repositório GIT, onde as imagens podem ser versionadas, comitadas, “puxadas” etc, quando recuperamos uma imagem, usando o comando `docker pull` por exemplo, estamos normalmente baixando a imagem de um registro Docker, o repositório oficial do Docker é o Docker HUB, onde é possível hospedar e versionar imagens públicas e privadas.

Elementos do **Docker**

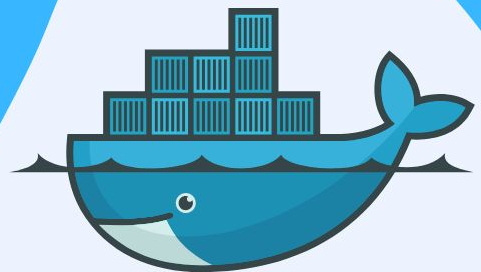
Docker Registry - É como um repositório GIT, onde as imagens podem ser versionadas, comitadas, “puxadas” etc, quando recuperamos uma imagem, usando o comando `docker pull` por exemplo, estamos normalmente baixando a imagem de um registro Docker, o repositório oficial do Docker é o Docker HUB, onde é possível hospedar e versionar imagens públicas e privadas.

Kubernetes vs. Docker

Docker é uma solução para containerização. A principal proposta dos containers é fazer isolamento entre processos.

Kubernetes é uma solução para orquestração de aplicações baseadas em *containers*. O Kubernetes utiliza uma solução de *containers* (como o Docker) para fazer o isolamento das aplicações especificadas nos serviços ou *pods* que podem ser constituídos por um ou mais *containers*.

Como o Kubernetes é um serviço de orquestração preocupações com escalabilidade, *deployment* e disponibilidade veem à tona, dessa forma existe o suporte à uma série de funcionalidades como escalonamento automático, balanceamento de carga, serviços de DNS e entre outros.



docker



Obrigado