

COMPUTAÇÃO EM NUVEM PARA WEB I

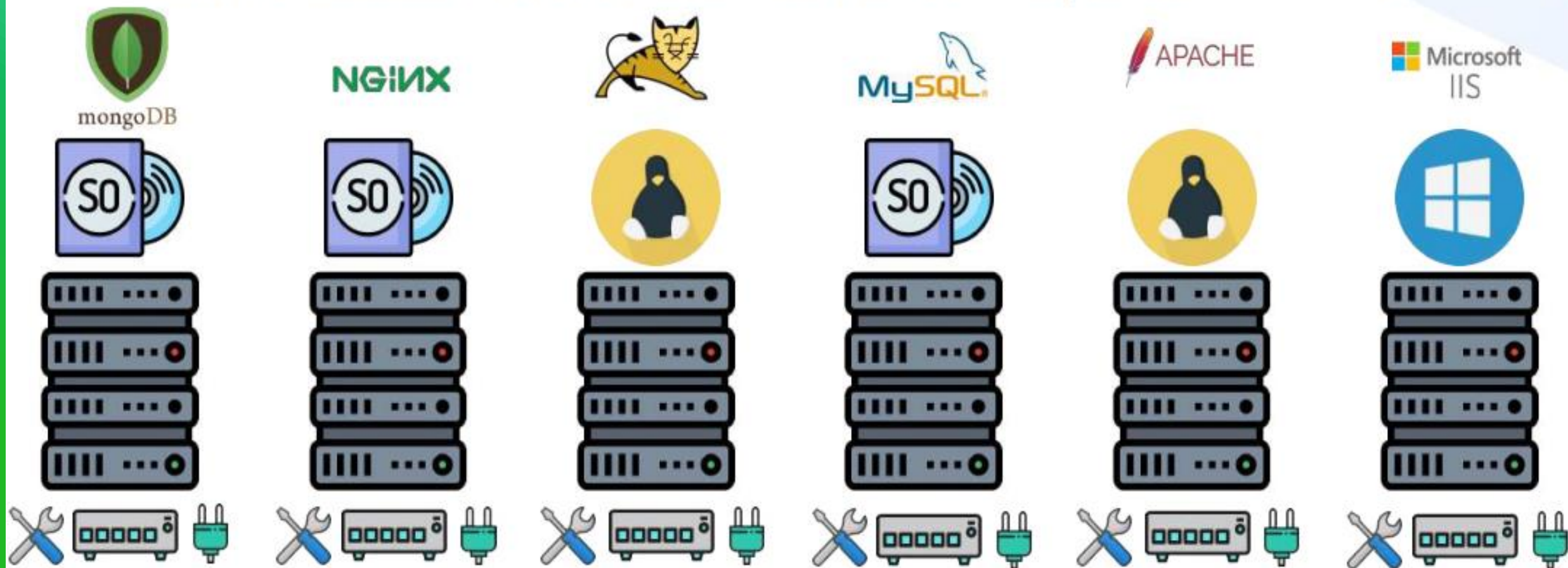
Aula 08 - Docker e Containers

Prof. Anderson Vanin



Como hospedamos aplicações anteriormente?

- ▶ Para vários serviços, utilizamos várias máquinas



Máquinas Virtuais

Capacidade subutilizada!

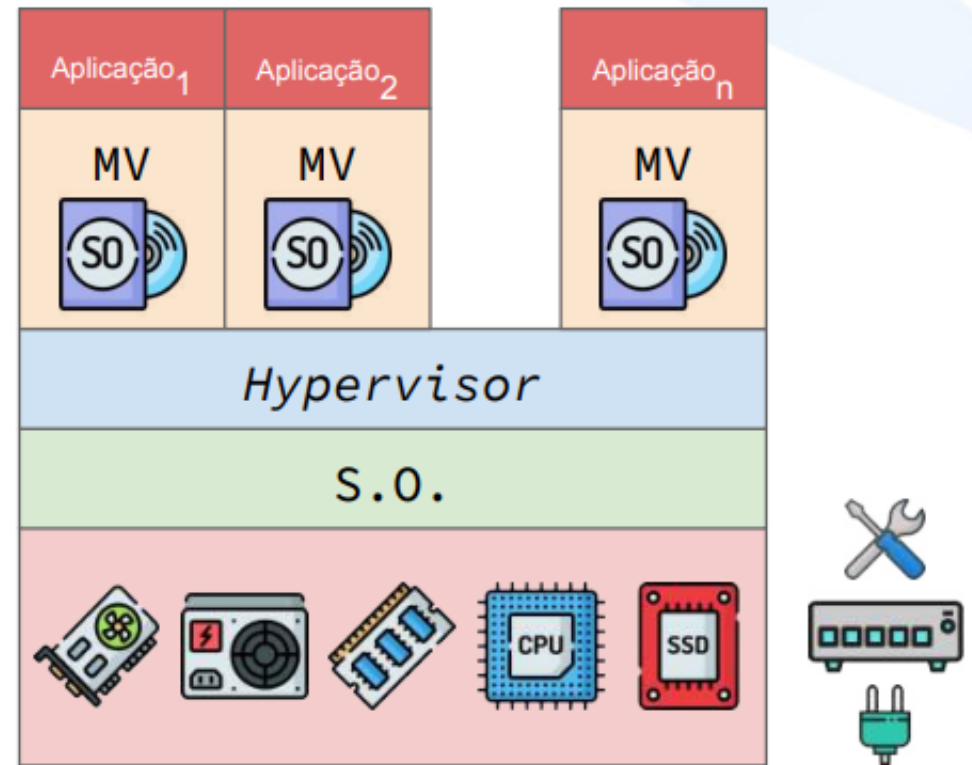


Máquinas subutilizadas na maior parte do tempo, desperdiçando poder computacional!

- ▶ Uso sazonal da capacidade máxima.

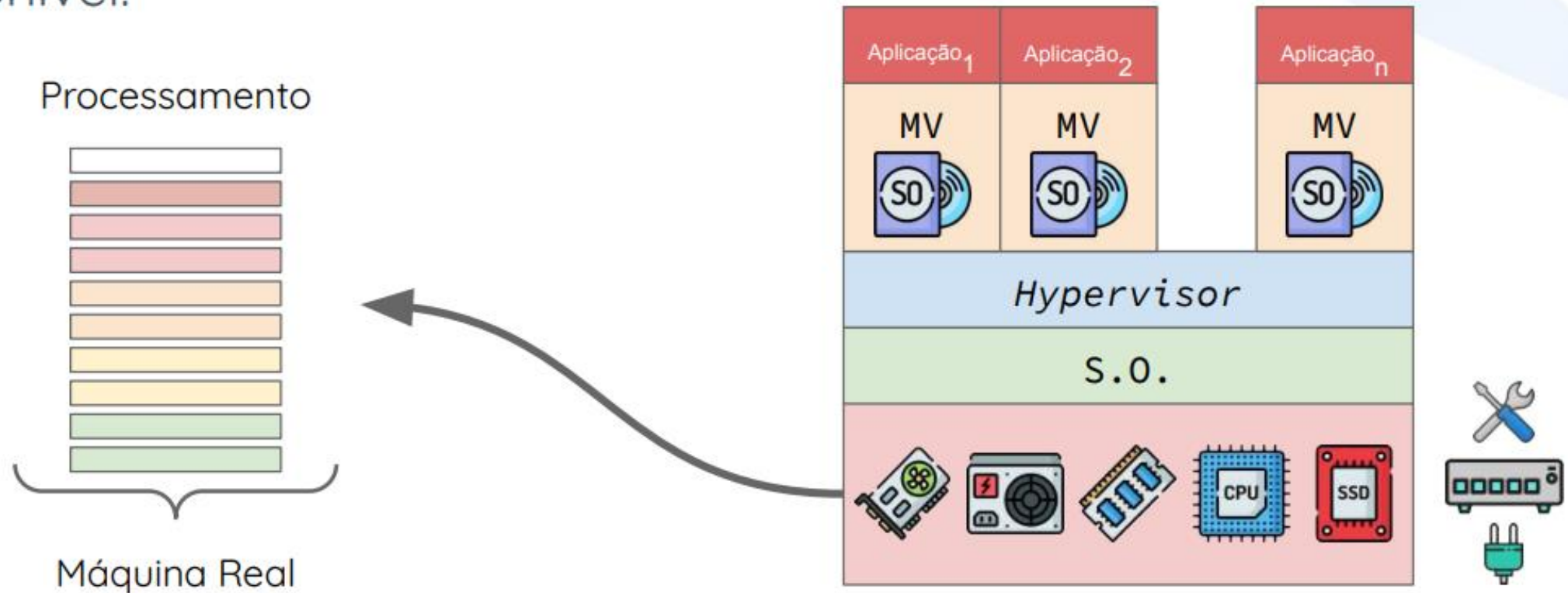
Máquinas Virtuais

Solução: vamos virtualizar nossos serviços!



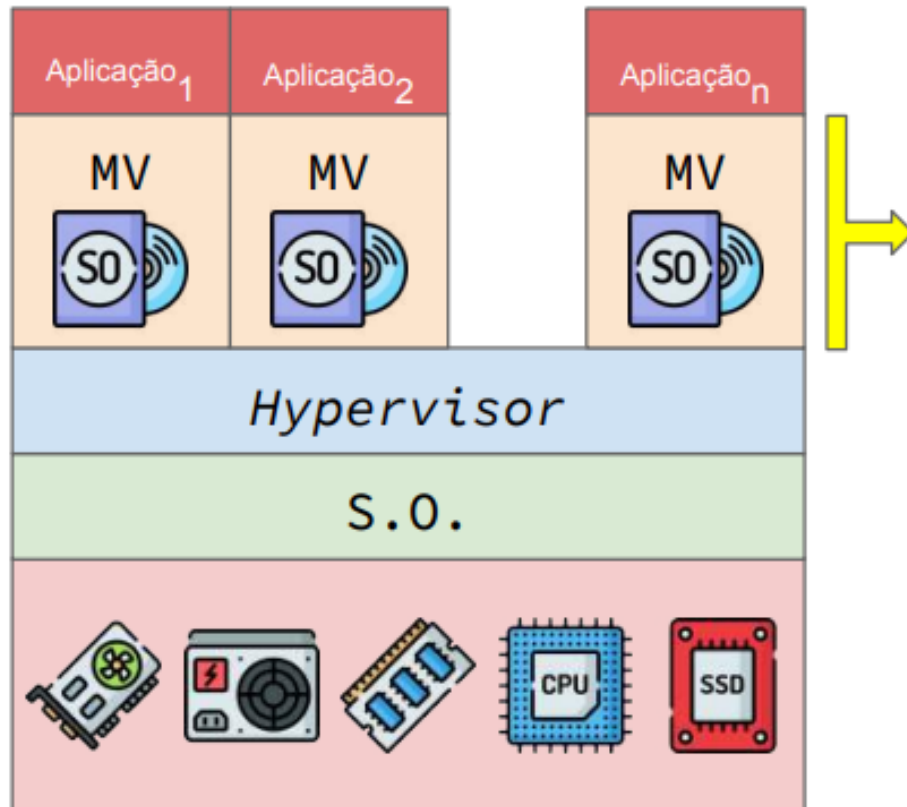
Máquinas Virtuais

As MV otimizaram o uso da capacidade de processamento disponível!



Máquinas Virtuais

Mas quais são as desvantagens das MV?



Custos de um Sistema Operacional

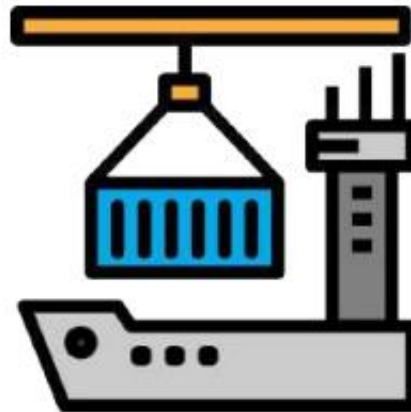
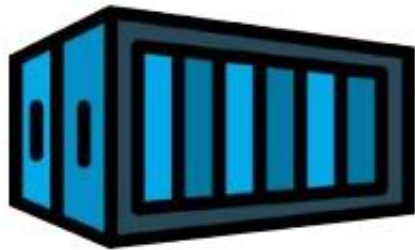
- ▶ Configuração
- ▶ Atualização
- ▶ Segurança
- ▶ Para cada máquina virtual
 - ▶ Armazenamento
 - ▶ Processamento
 - ▶ Memória

Máquinas Virtuais

Já resolvemos os problemas das máquinas físicas com as máquinas virtuais.

- ▶ E agora, como resolver o problema das máquinas virtuais?

Utilizando...



O que é Docker?

É uma ferramenta que se apoia em recursos existentes no kernel, inicialmente Linux, para isolar a execução de processos. As ferramentas que o Docker traz são basicamente uma camada de administração de containers, baseado originalmente no LXC.

O que é Docker?

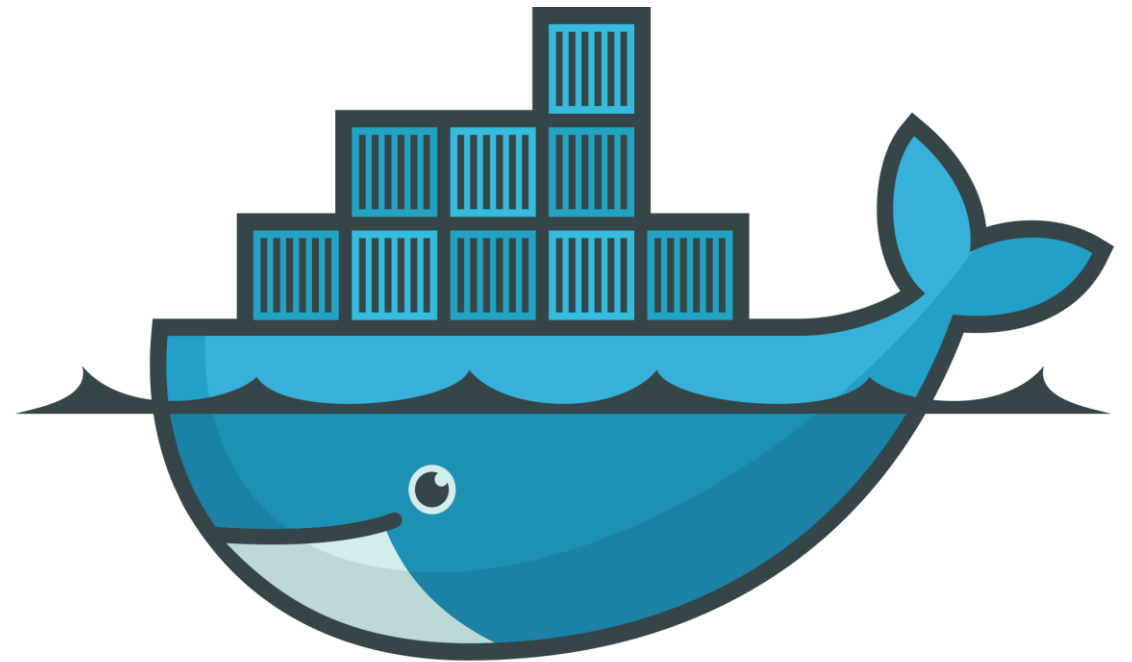
Alguns isolamentos possíveis:

- Limites de uso de memória
- Limites de uso de CPU
- Limites de uso de I/O
- Limites de uso de rede
- Isolamento da rede (que redes e portas são acessíveis)
- Isolamento do file system
- Permissões e Políticas
- Capacidades do kernel

Definição Oficial

Containers Docker empacotam componentes de software em um sistema de arquivos completo, que contém tudo necessário para a execução: código, runtime, ferramentas de sistema - qualquer coisa que possa ser instalada em um servidor.

Isto garante que o software sempre irá executar da mesma forma, independente do seu ambiente.



docker

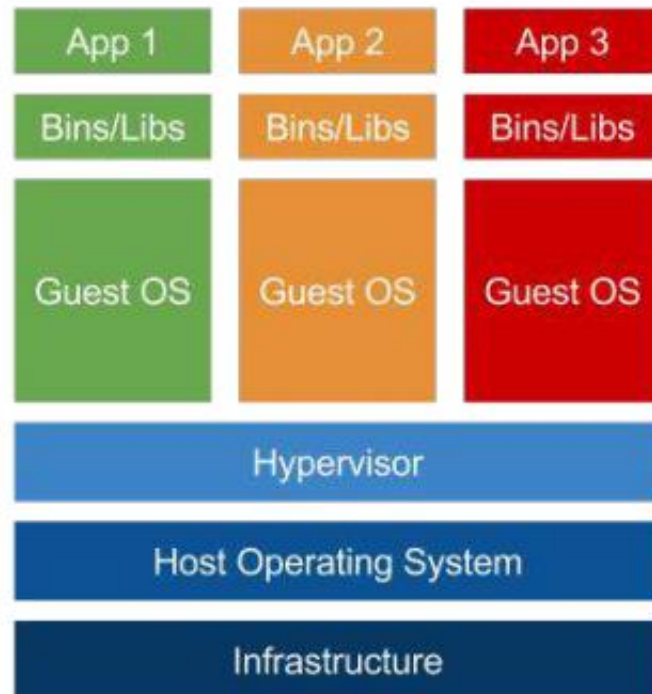
Porque não uma VM?

O Docker tende a utilizar menos recursos que uma VM tradicional, um dos motivos é não precisar de uma pilha completa como vemos na figura a seguir. O Docker utiliza o mesmo kernel do host, e ainda pode compartilhar bibliotecas.

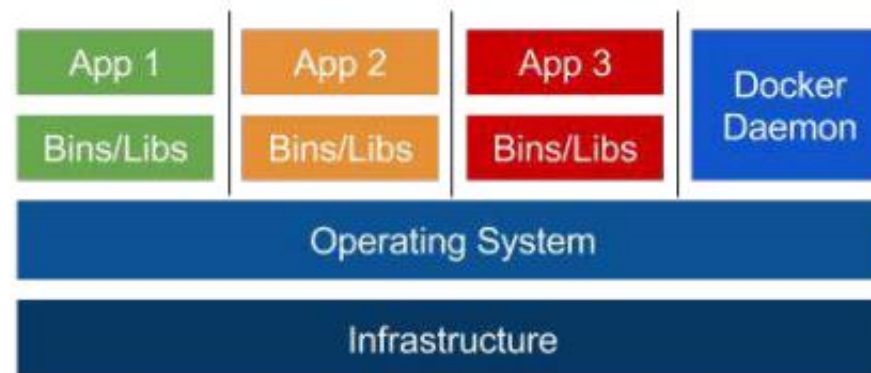
Mesmo utilizando o mesmo kernel é possível utilizar outra distribuição com versões diferentes das bibliotecas e aplicativos.

Porque não uma VM?

Docker NÃO é um Hypervisor



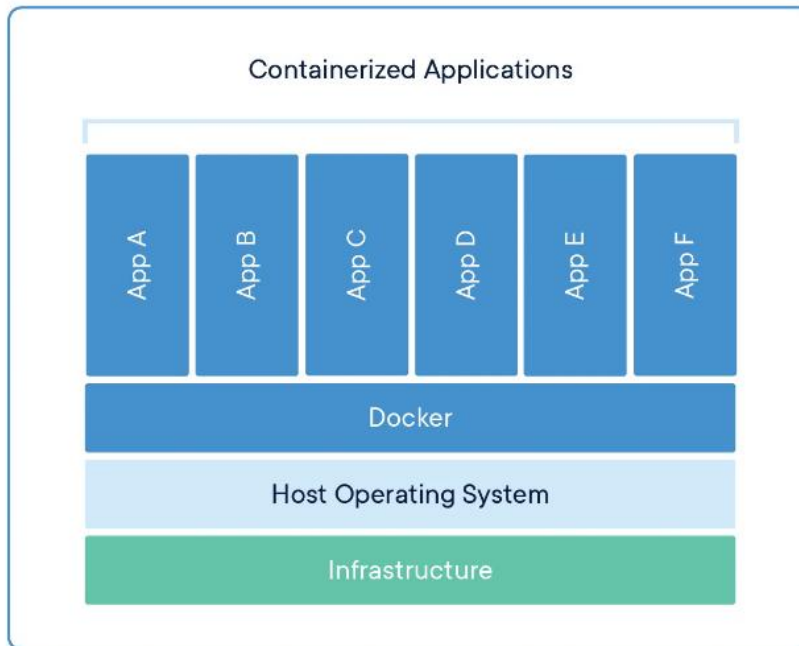
- Docker Daemon está fora do caminho de execução;
- Apps possuem “paredes” entre elas;
- Docker Daemon é apenas mais um processo.



apps DO NOT sit “on top” of the container engine

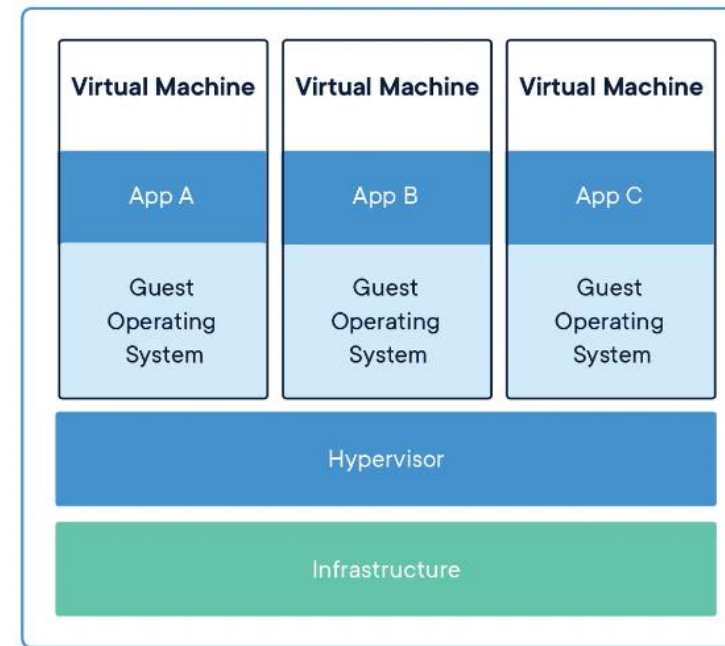
[Veja mais detalhes aqui](#)

Comparação VM X Containers



CONTAINERS

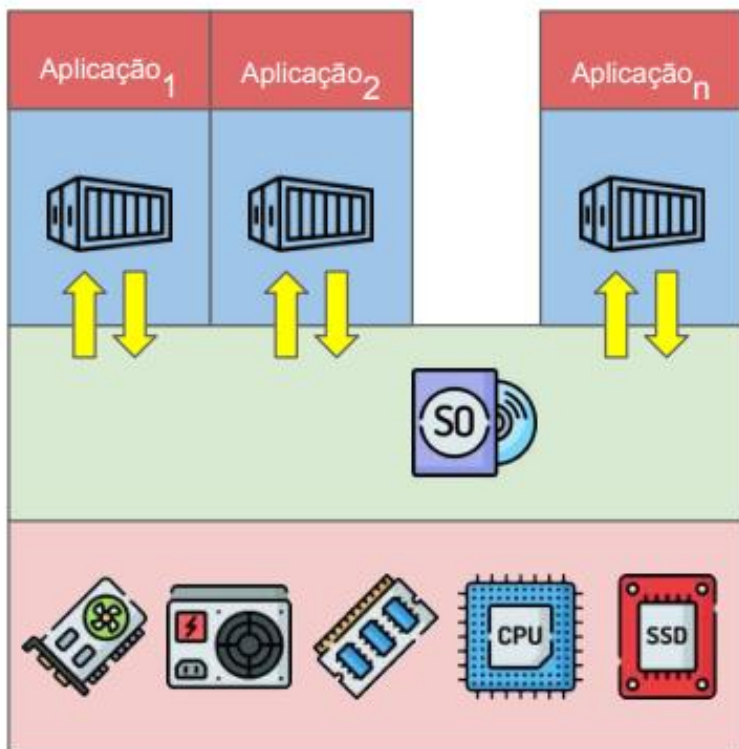
Os contêineres são uma abstração na camada do aplicativo que empacota código e dependências. Vários contêineres podem ser executados na mesma máquina e compartilhar o kernel do sistema operacional com outros contêineres, cada um executando como processos isolados no espaço do usuário. Os contêineres ocupam menos espaço que as VMs (as imagens dos contêineres geralmente têm dezenas de MBs), podem lidar com mais aplicativos e exigem menos VMs e sistemas operacionais.



MÁQUINAS VIRTUAIS

Máquinas virtuais (VMs) são uma abstração de hardware físico que transforma um servidor em vários servidores. O hipervisor permite que várias VMs sejam executadas em uma única máquina. Cada VM inclui uma cópia completa de um sistema operacional, o aplicativo, os binários e bibliotecas necessários – ocupando dezenas de GBs. As VMs também podem demorar para inicializar.

Caracterizando um Container



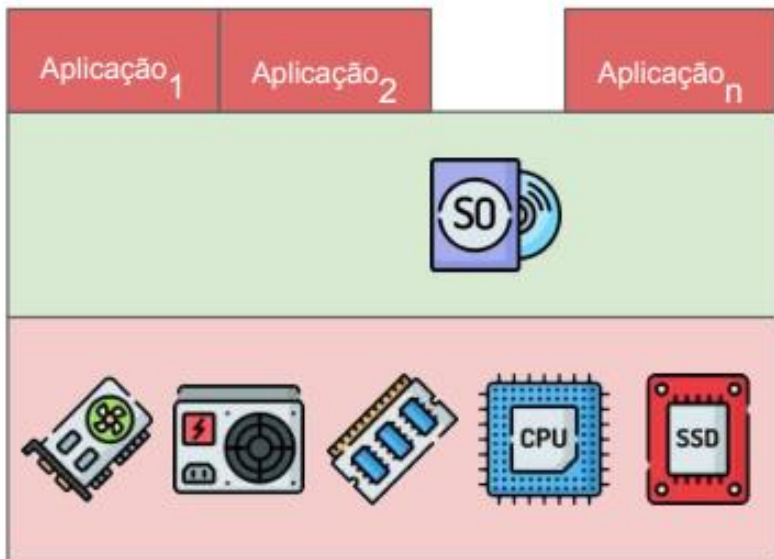
Um container será um ambiente de virtualização

- ▶ Com menos consumo de processamento,
- ▶ Com menor custo de manutenção/configuração,
- ▶ Com menor tempo de inicialização/finalização.

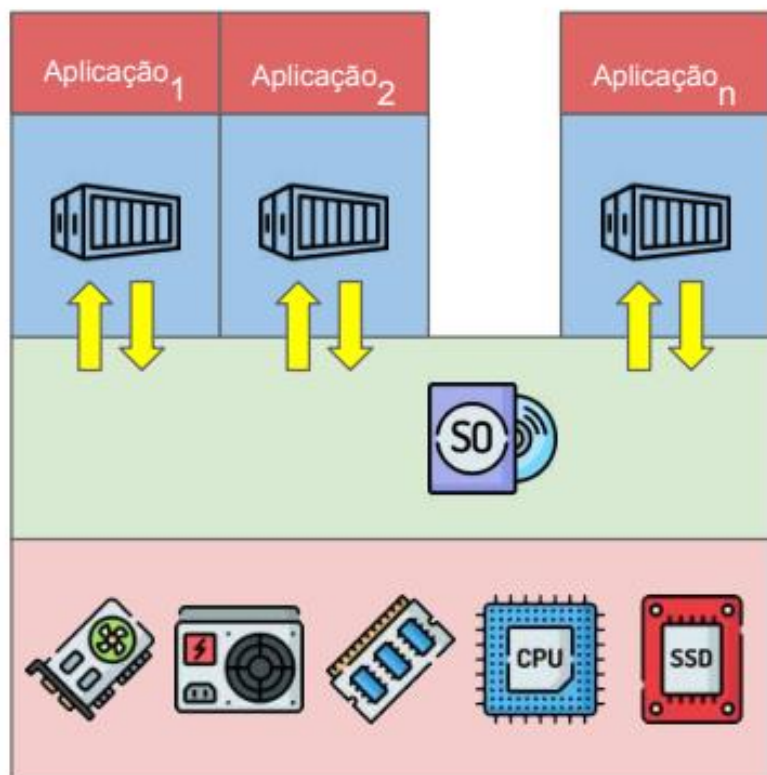
Caracterizando um Container

Por que não utilizar as aplicações diretamente no SO principal?

1. Concorrência entre portas,
2. Consumo unilateral da CPU,
3. Dependências entre diferentes recursos.



Caracterizando um Container



Ganhamos:

- ▶ Controle sobre o uso de cada recurso,
- ▶ Velocidade para criação/destruição de serviços,
- ▶ Maior gerenciamento de dependências de bibliotecas,
- ▶ Menor consumo de processamento do que as MV.

Então devo trocar?

Devo trocar então minha VM por um container? Nem sempre, os containers Docker possuem algumas limitações em relação as VMs:

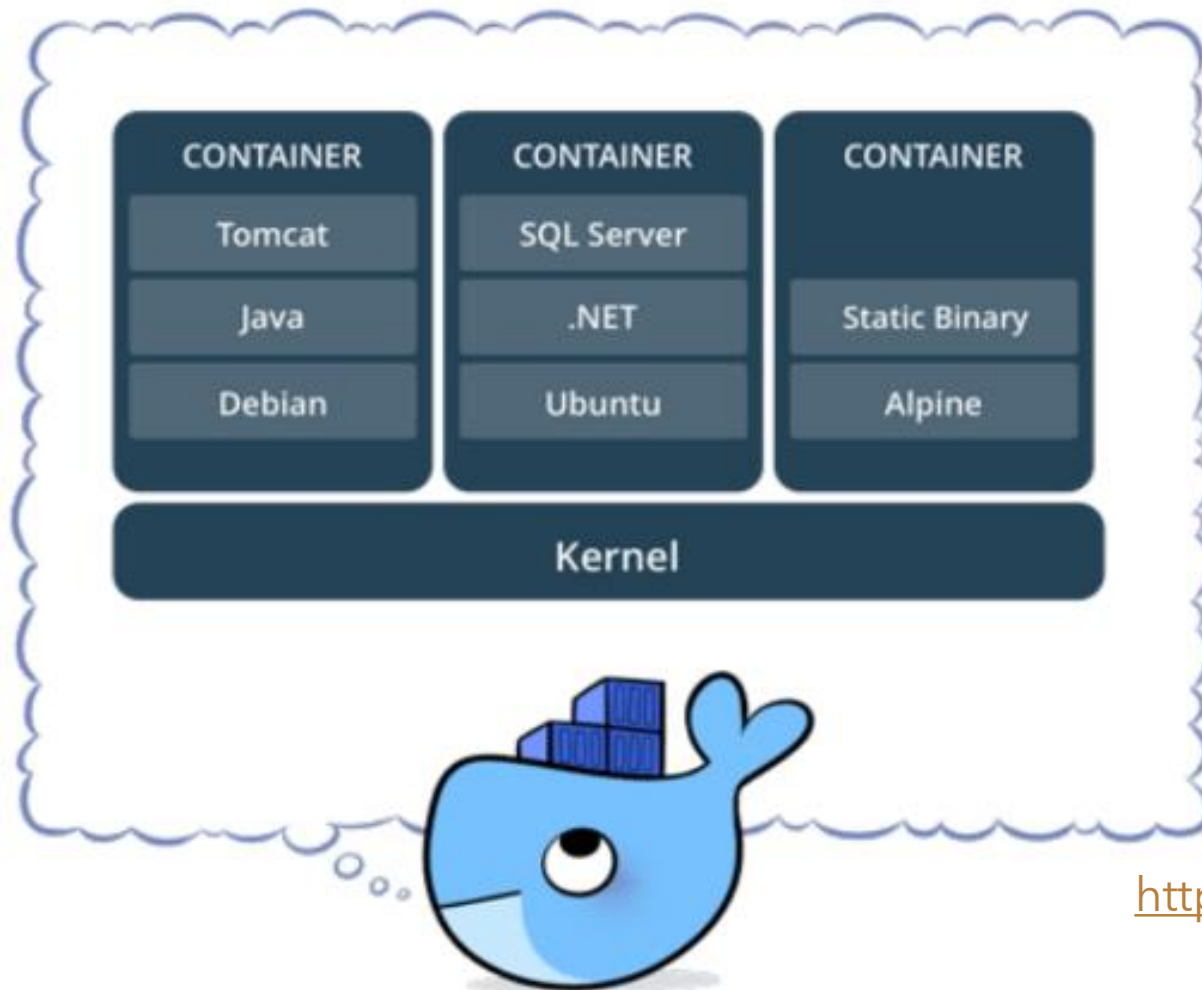
- Todas as imagens são linux, apesar do host poder ser qualquer SO que use ou emule um kernel linux, as imagens em si serão baseadas em linux.
- Não é possível usar um kernel diferente do host, o Docker Engine estará executando sob uma determinada versão (ou emulação) do kernel linux, e não é possível executar uma versão diferente, pois as imagens não possuem kernel.

O que são Containers?

Container é o nome dado para a segregação de processos no mesmo kernel, de forma que o processo seja isolado o máximo possível de todo o resto do ambiente.

Em termos práticos são File Systems, criados a partir de uma "imagem" e que podem possuir também algumas características próprias.

O que são Containers?



<https://www.docker.com/what-container>

O que são imagens Docker?

Uma imagem Docker é a materialização de um modelo de um sistema de arquivos, modelo este produzido através de um processo chamado build.

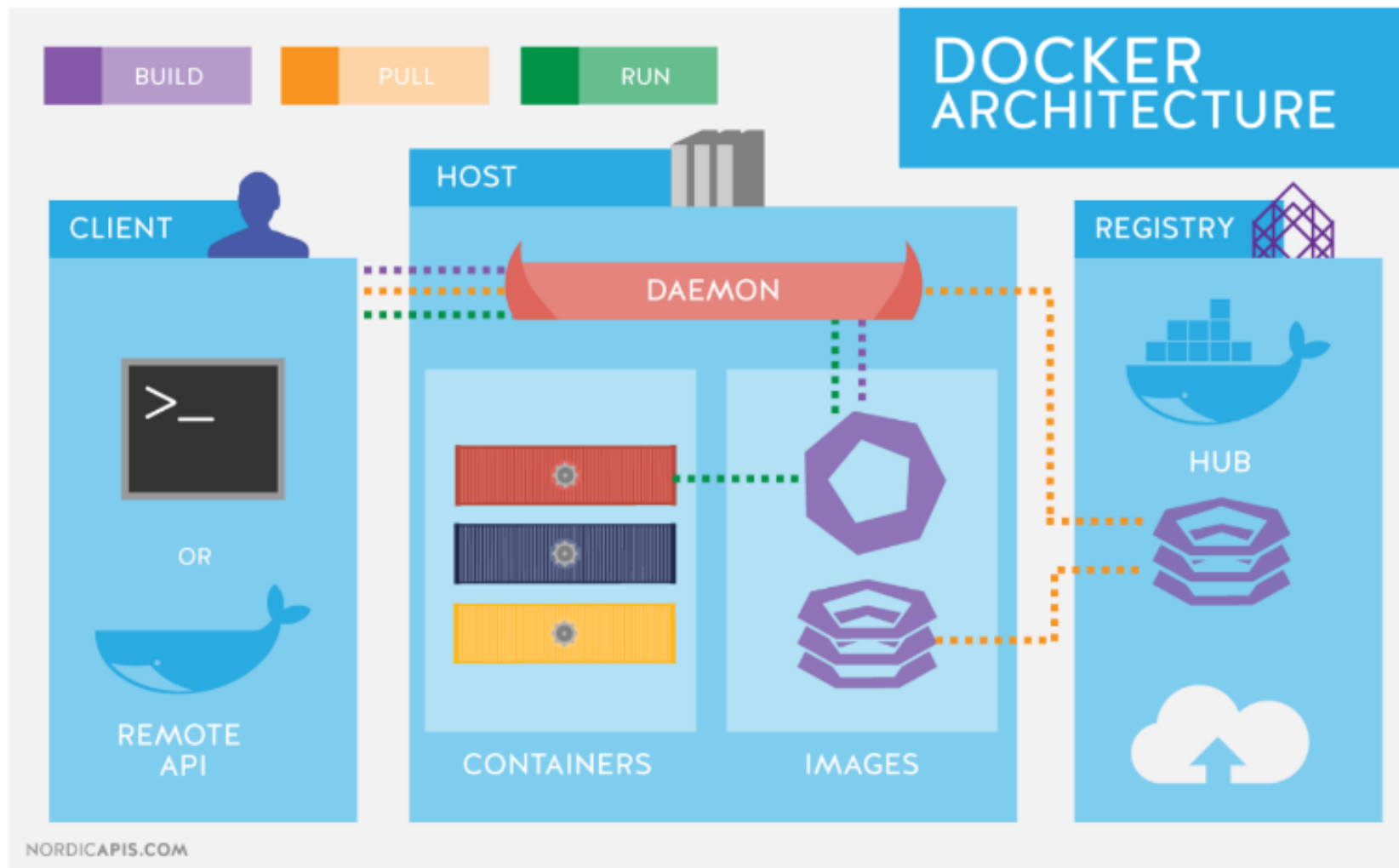
Esta imagem é representada por um ou mais arquivos e pode ser armazenada em um repositório.

Arquitetura

De maneira simplificada podemos dizer que o uso mais básico do Docker consiste em:

- Ter o serviço Docker Engine rodando
- Ter acesso a API Rest do Docker Engine, normalmente através do Docker Client
- Baixar uma imagem do Docker Registry, normalmente do registry público oficial: <https://hub.docker.com>
- Instanciar um container a partir da imagem baixada

Arquitetura



Vídeo

Aula Resumida

+ <https://www.youtube.com/watch?v=Kzcz-EVKBEQ>

Aula Prática Completa sobre Docker (Node e MySql)

+ <https://www.youtube.com/watch?v=obtKdOW1pWQ>