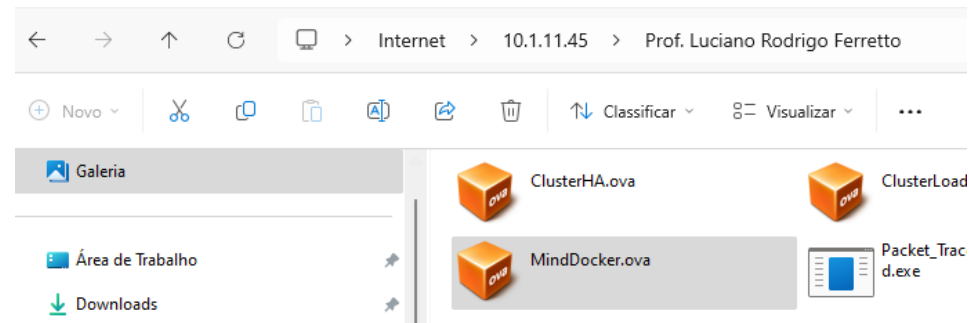


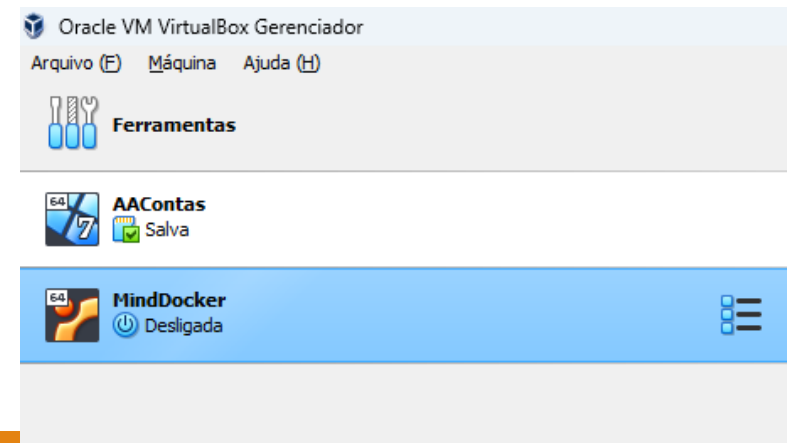
Paradigmas de Linguagem de Programação

Preparação ambiente

Arquivo: **MintDocker.ova**



Importar dentro do VirtualBox >> **Importar Appliance**



Virtualização - Benefícios

Isolamento completo

- Cada VM roda seu próprio sistema operacional (Windows, Linux...).
- Problemas numa VM não afetam as outras.

Melhor aproveitamento de hardware

- Um único servidor físico pode rodar várias VMs, usando os recursos de forma mais eficiente.



1. Benefícios da Virtualização (VMs)

Antes dos containers, a virtualização com VMs foi (e ainda é) uma revolução. Os principais benefícios:

Virtualização - Benefícios

Facilidade de gerenciamento

- VMs podem ser **ligadas/desligadas/salvas/clonadas** facilmente.
- Ideal pra ambientes de desenvolvimento, testes e produção.

Ambientes padronizados

- Você pode garantir que todos estão usando exatamente o mesmo sistema/configuração.

Segurança

- Como cada VM tem seu sistema isolado, o impacto de falhas ou invasões é limitado.



1. Benefícios da Virtualização (VMs)

Antes dos containers, a virtualização com VMs foi (e ainda é) uma revolução. Os principais benefícios:

Mas... (sempre tem um mas...)



2. Mas VMs têm limitações...

- São mais pesadas (em memória, disco, tempo de boot).
- Cada VM precisa de um sistema operacional completo.
- Leva tempo para iniciar.
- Redundância de bibliotecas/sistemas.

E aqui entra os Containers

MAIS ESPECIFICAMENTE, O DOCKER



Docker

Engine de Administração de Containers

É um sistema de virtualização, mas **NÃO** é um Sistema de Virtualização Tradicional como o VirtualBox, VmWare ESXi, Microsoft Hyper-V, etc.

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.

Docker – Características básicas

1. Não é um sistema de virtualização tradicional
2. Sistema de Virtualização Baseado em Software
3. Engine de Administração de Containers
4. Inicialmente baseado no LXC (Linux Containers)
 - Ainda utiliza alguns recursos chaves do LXC como namespaces e cgroups
5. Open Source escrito em Go
6. Host e Containers compartilham o Kernel
7. Empacota o software com vários níveis de isolamento

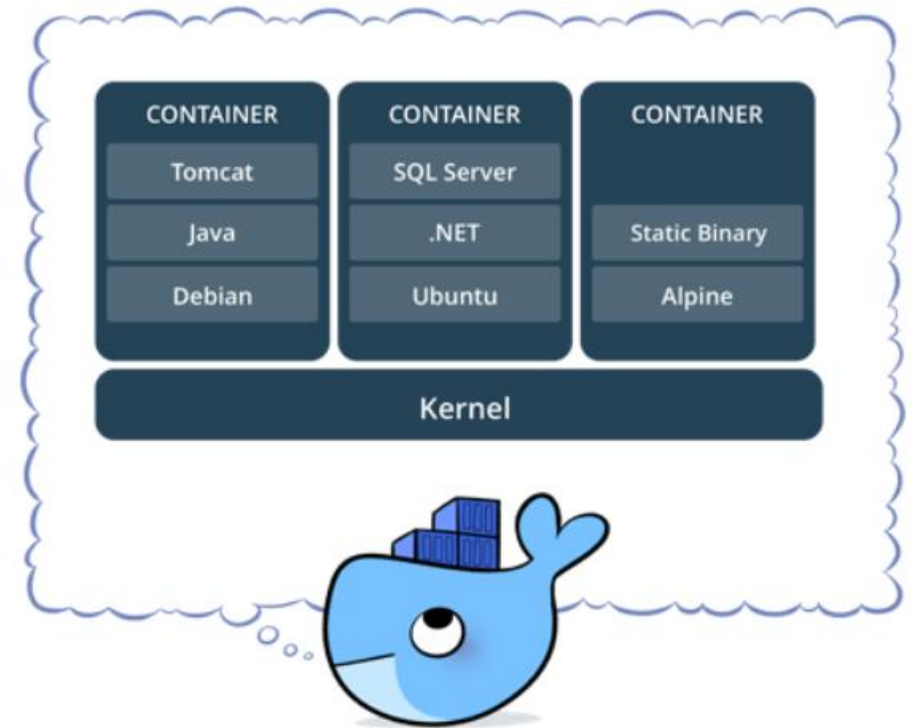
O que é um Container?

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.

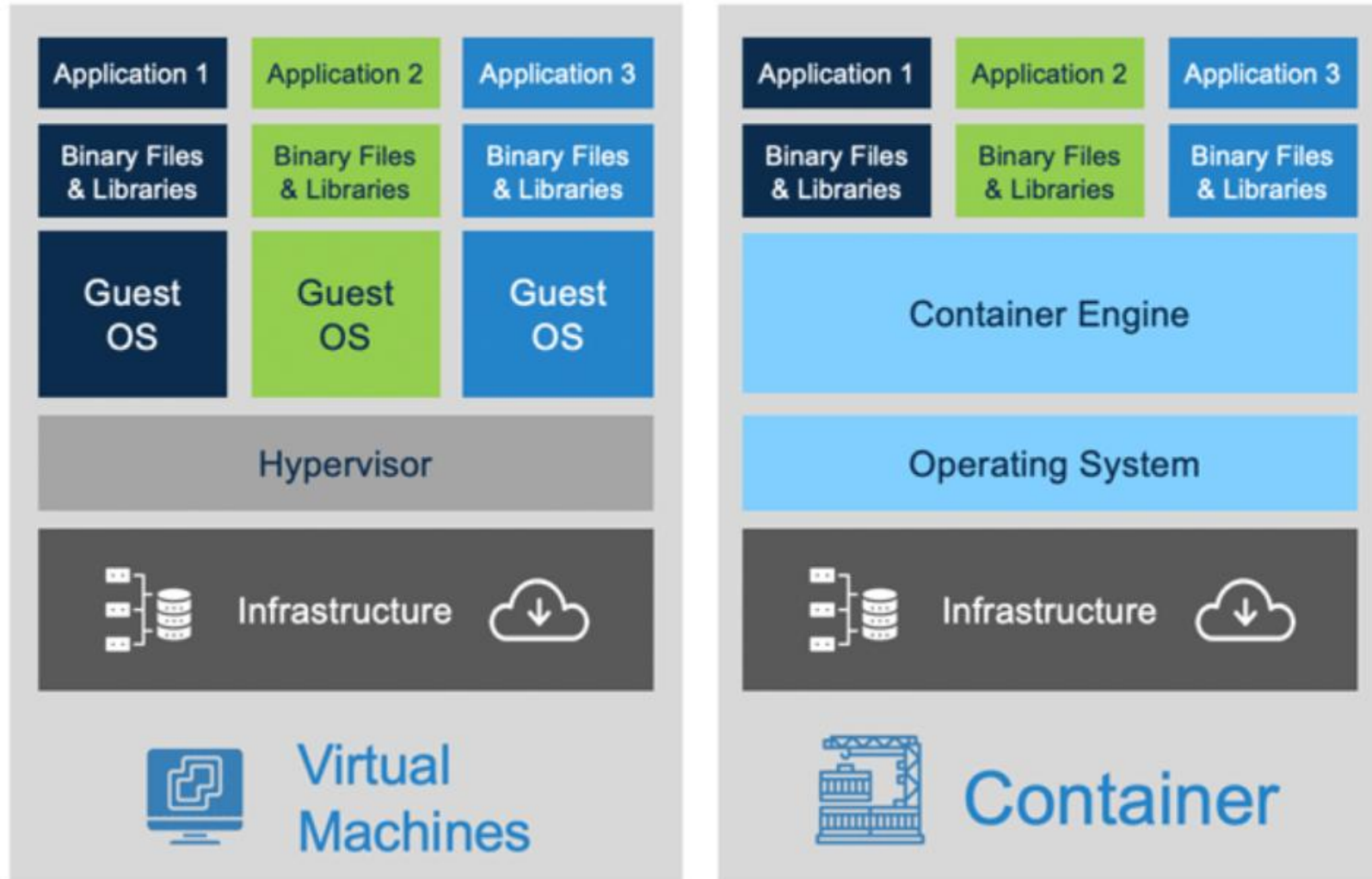
Sistema de Arquivos criados a partir de uma "*imagem*"

Ambiente leve e portátil no qual aplicações são executadas

Encapsula todos os binários e bibliotecas necessárias para a execução de uma aplicação.



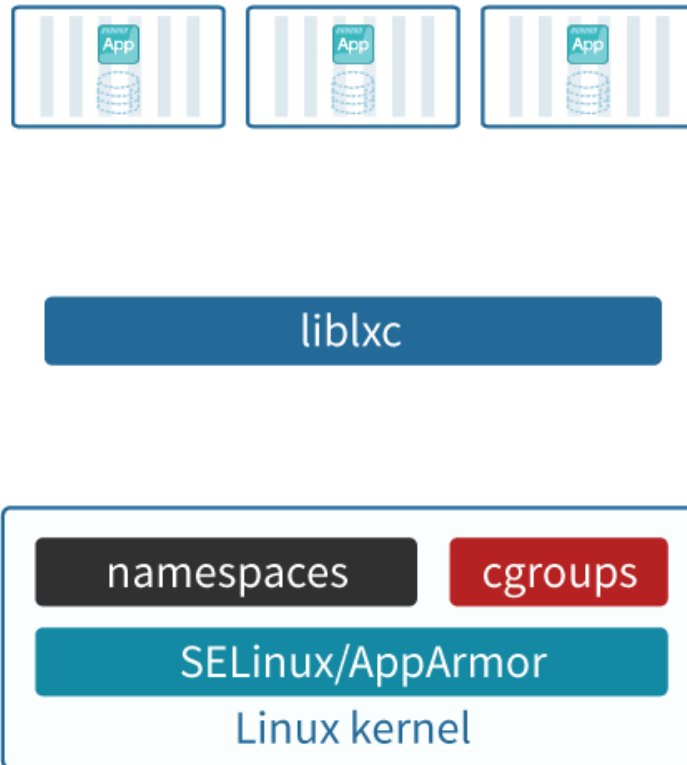
VMs x Containers



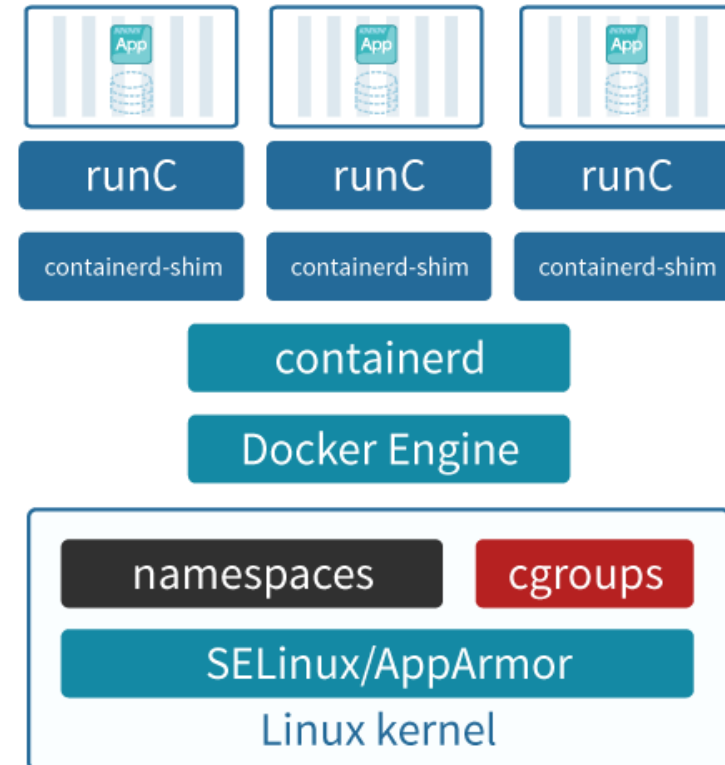
LXC x Docker



Linux Containers



Docker 1.10 and later



O que são Imagens Docker?

Modelo de sistema de arquivo **somente-leitura** usado para criar containers.

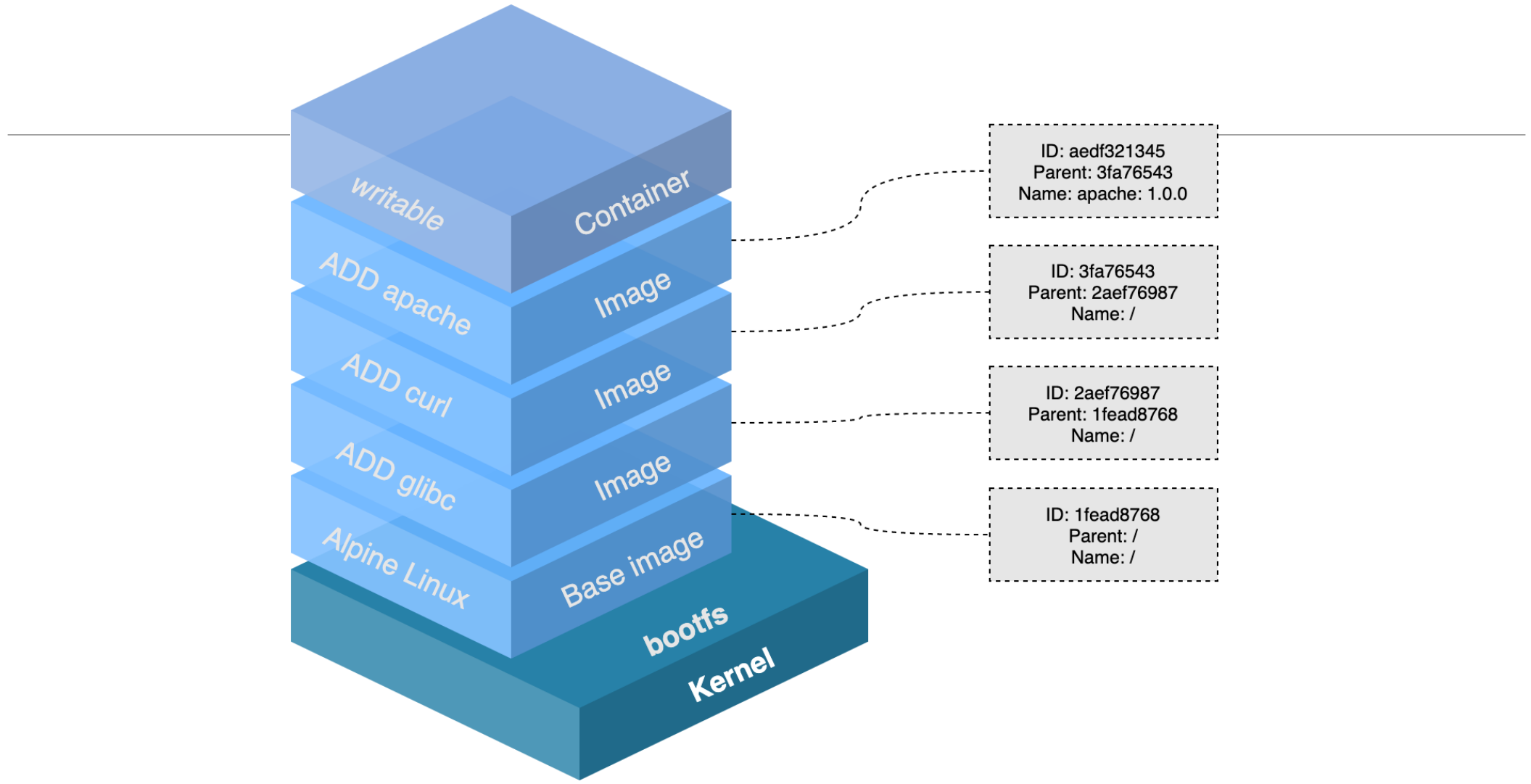
São armazenados em repositórios no Registry.

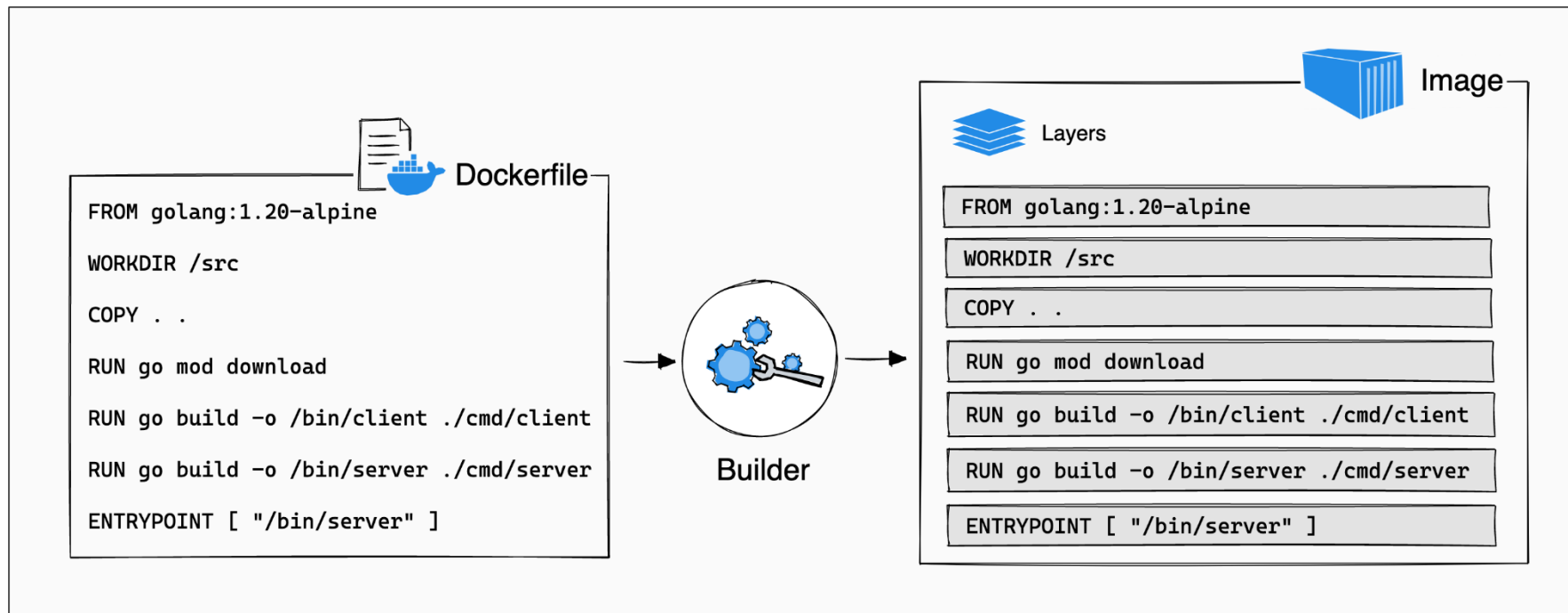
São compostos por uma ou mais camadas (layers)

- Uma camada representa uma ou mais mudanças no sistema de arquivo
- Uma camada é também chamada de imagem intermediária
- A junção dessas camadas formam a imagem

Quando o container é iniciado, é criado uma última camada que é a única que pode ser alterada.

AUFS (Advanced multi-layered unification filesystem)

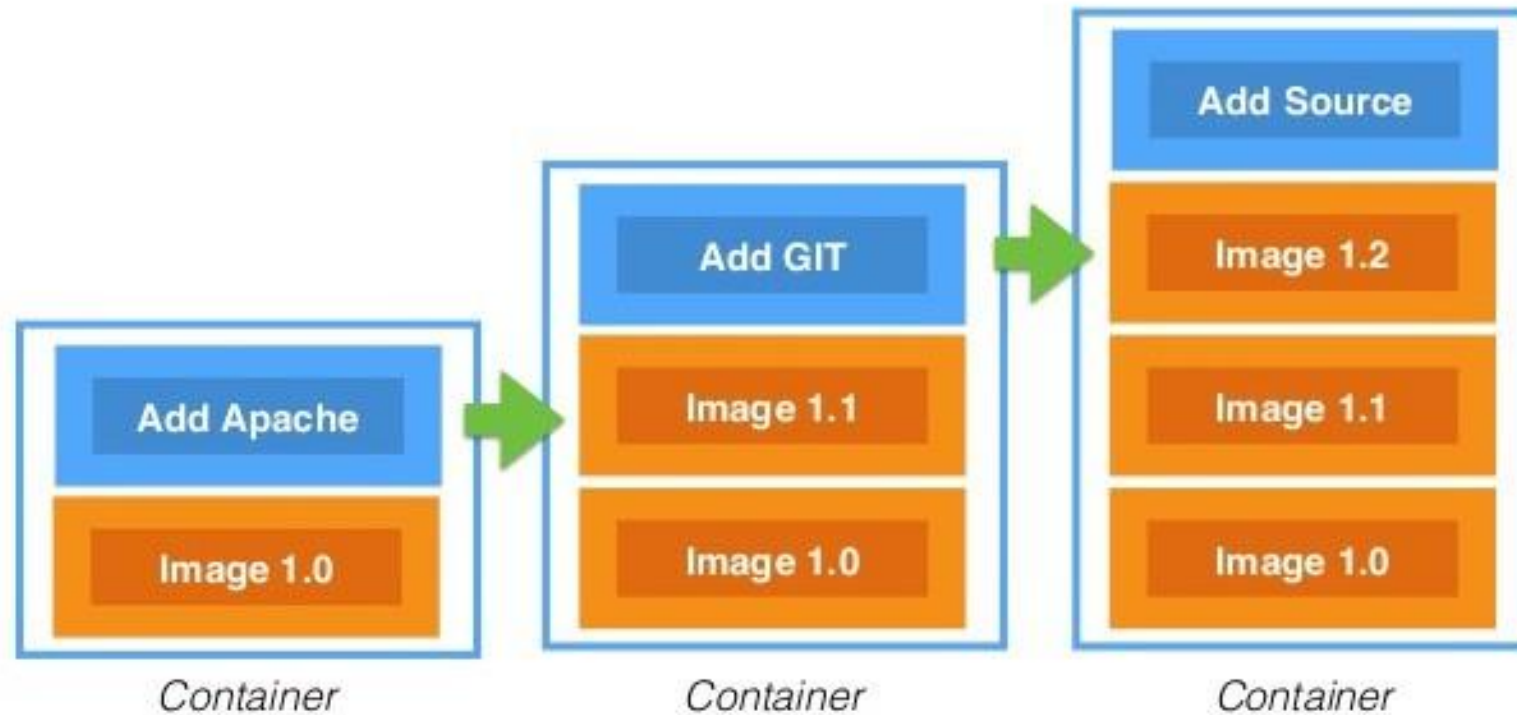




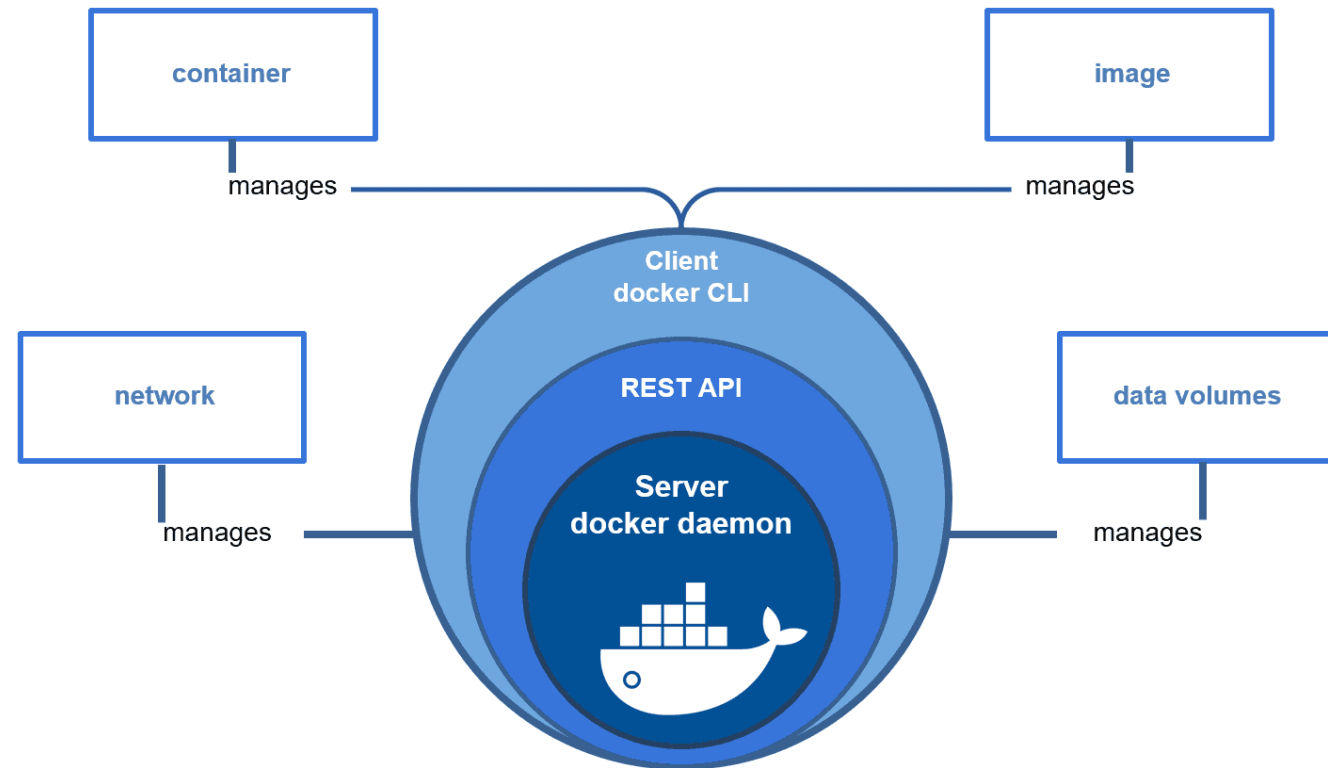
AuFS

Layered Filesystem

2019



Arquitura Docker



DOCKER ARCHITECTURE

