

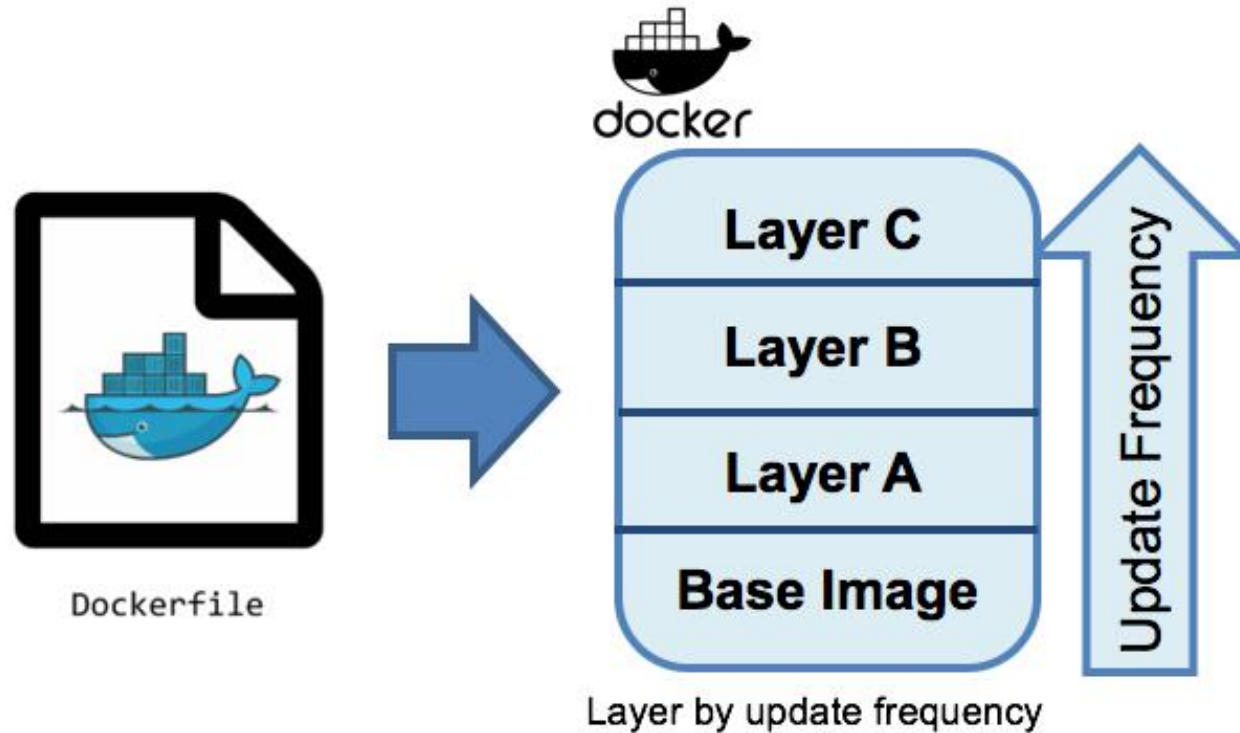


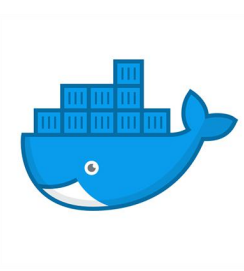
Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Sistema de Arquivos em Camadas



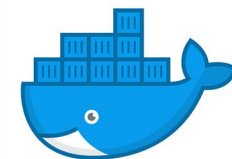


Sistema de Arquivos em Camadas

Um dos tópicos mais importantes do **Docker** é o *filesystem*, ou seja, o sistema de arquivos utilizado pela ferramenta.

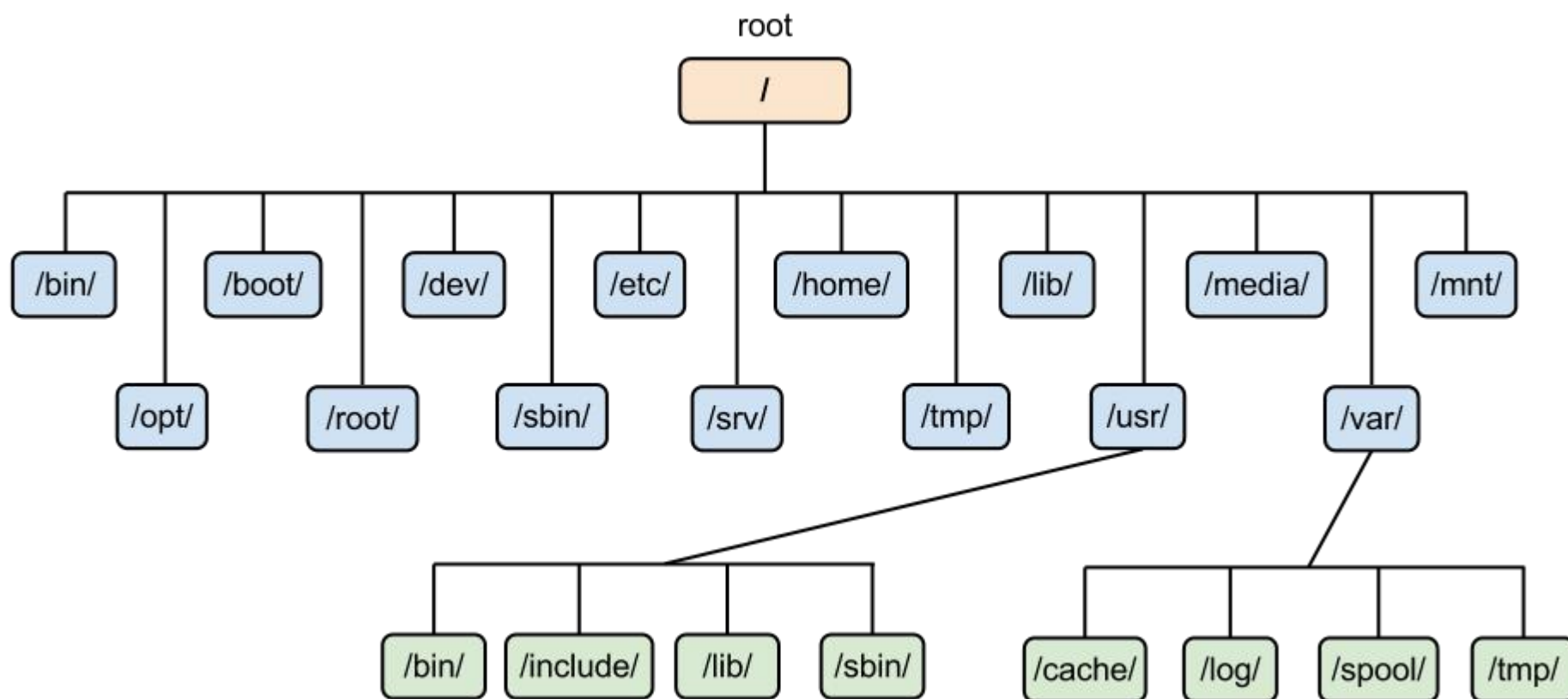
O sistema de arquivos do **Docker** é chamado de **Layered**, ou seja, é um sistema de arquivos em camadas.

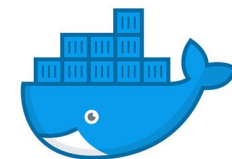
Mas o que isso quer dizer?



Sistema de Arquivos em Camadas

Sistemas de Arquivos





Sistema de Arquivos em Camadas

Sistemas de Arquivos

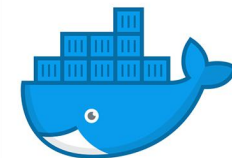
Um sistema de arquivos comum (Linux/Unix), possui basicamente duas camadas:

bootfs: Onde fica o sistema de Boot do sistema operacional e o Kernel.

rootfs: Inclui o sistema de arquivos do sistema operacional, incluindo a arquitetura de diretórios, como /dev, /proc, /bin, /etc, /lib, /usr e /tmp assim como os arquivos de configuração e binários do sistema operacional.

Quando o sistema operacional é iniciado ele carrega o ***rootfs*** primeiramente em modo somente leitura, verifica sua integridade e em seguida remonta-o como leitura/escrita e assim ficando disponível para o usuário e aplicações.

Sistema de Arquivos em Camadas

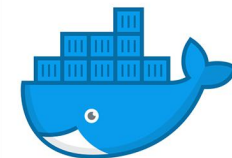


Sistemas de Arquivos no Docker

No Docker temos esta arquitetura também, mas cum um pequeno diferencial:

A camada de escrita que o processo/aplicação visualiza não é o mesmo rootfs base do sistema mas sim uma camada de abstração do rootfs.

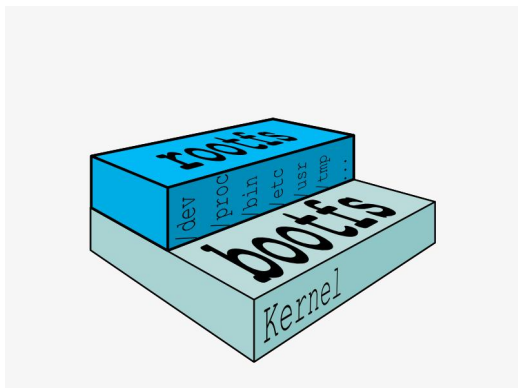
Isso faz com que um container torne-se portátil, pois as modificações realizadas não são aplicadas ao sistema origem do container e sim na camada a qual o sistema visualiza.



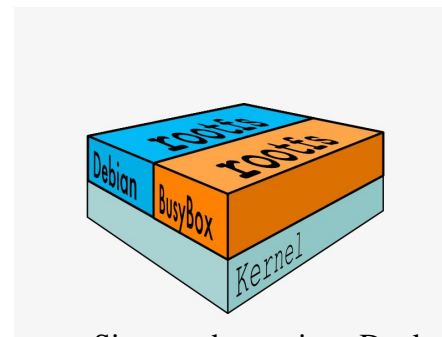
Sistema de Arquivos em Camadas

Sistema de Arquivos vs Sistemas de Arquivos no Docker

Para termos um entendimento mais claro do funcionamento do sistema de arquivos:



Sistema de arquivos comum

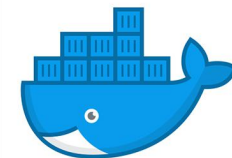


Sistema de arquivos Docker

Note que o **bootfs** em vez de ser único por sistema/container é compartilhado entre eles.

Já o **rootfs** é isolado por camadas, ou seja, o que deve ser em comum entre o host e o container é compartilhado via **AUFS***, que monta uma camada de leitura/escrita em cima do sistema de arquivos que é somente leitura. Isso garante que modificações feitas dentro do container não afetem o sistema de arquivos do host.

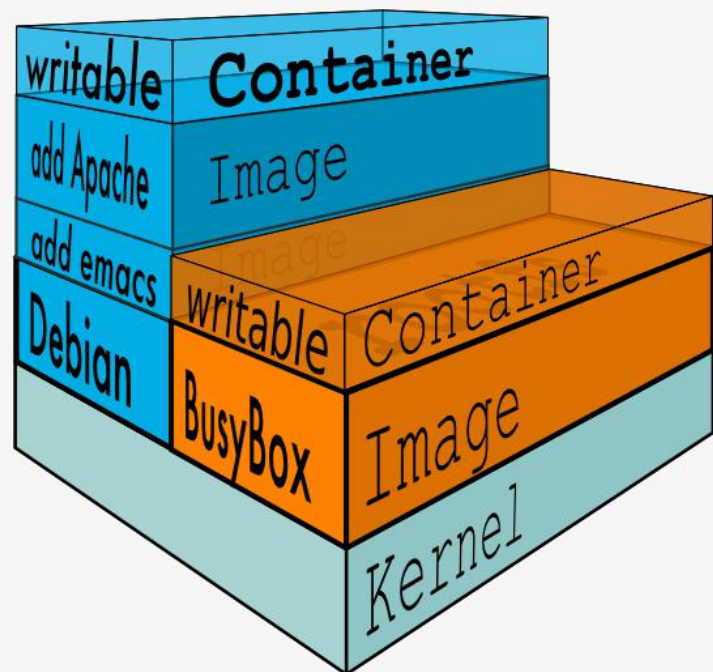
* <https://en.wikipedia.org/wiki/Aufs>



Sistema de Arquivos em Camadas

Sistemas de Arquivos no Docker

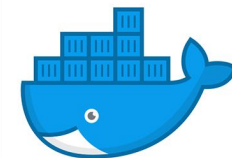
Note um exemplo do funcionamento da arquitetura do sistema de arquivos do Docker:



A cada modificação do container é gerada uma nova camada.

Por exemplo, imagine que iniciamos um container a partir de uma imagem do Debian, e então instalamos o Emacs no container. Então instalamos o servidor web Apache.

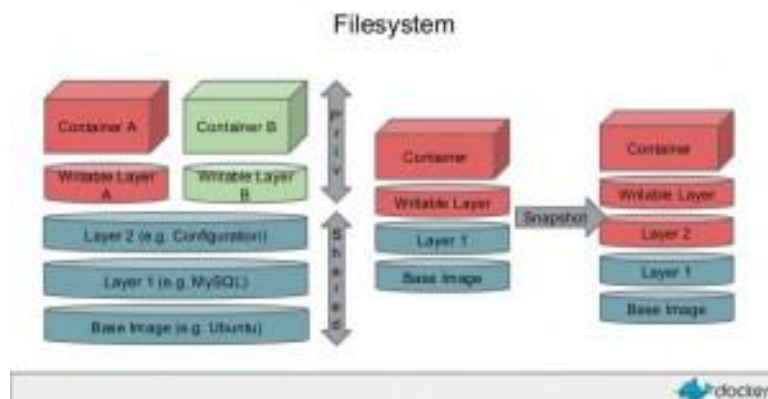
Se formos transformar este container em uma imagem, esta imagem possuirá a primeira camada que é do Debian e mais duas camadas adicionais, do Emacs e do Apache.



Sistema de Arquivos em Camadas

Imagens são compartilhadas entre containers Docker

Este tipo de sistema de arquivos em camadas busca sempre a eficiência. Desta forma, containers compartilham imagens semelhantes.



Ou seja, imaginando que temos 2 containers, sendo que: No Container 1 instalamos o Debian, instalamos o Nginx e No Container 2 instalamos o Debian, instalamos o Redis

A imagem do Debian será compartilhada entre os dois containers.



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br