

As imagens são construídas em tempo de build, os containers em tempo de execução. Para reforçar que é uma distinção importante e que devemos mantê-la em mente. As imagens são usadas para inicializar um contêiner, logo os contêineres são instância de imagens em tempo de execução.

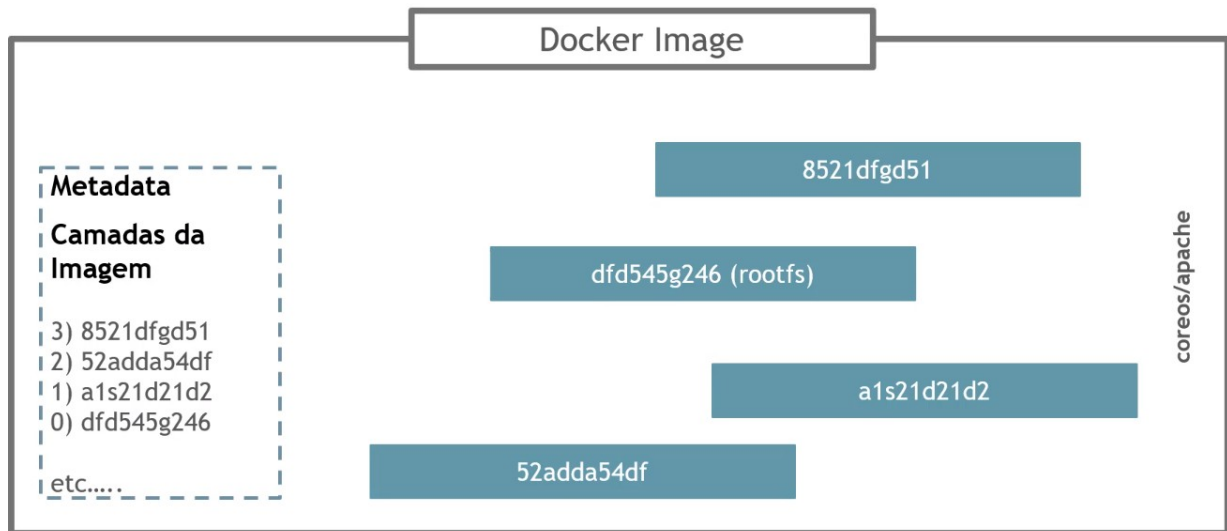


Figura 1: Quando inicializamos o contêiner com o comando "docker run", o docker engine lê a imagem e qualquer metadado p/ criar um contêiner.

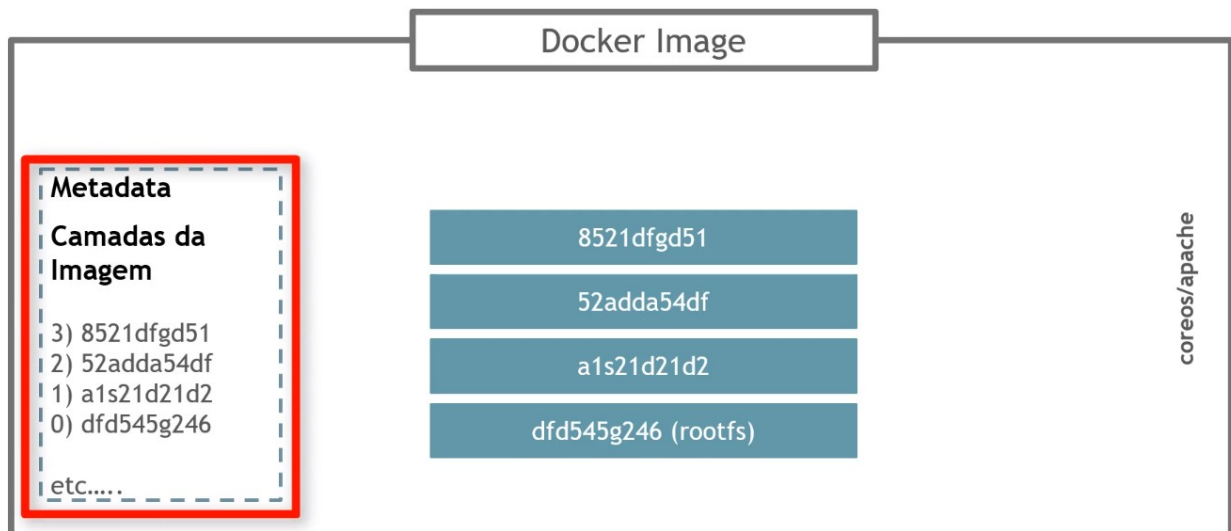


Figura 2: Empilhando as diferentes camadas da imagem, conforme as instruções dos metadados

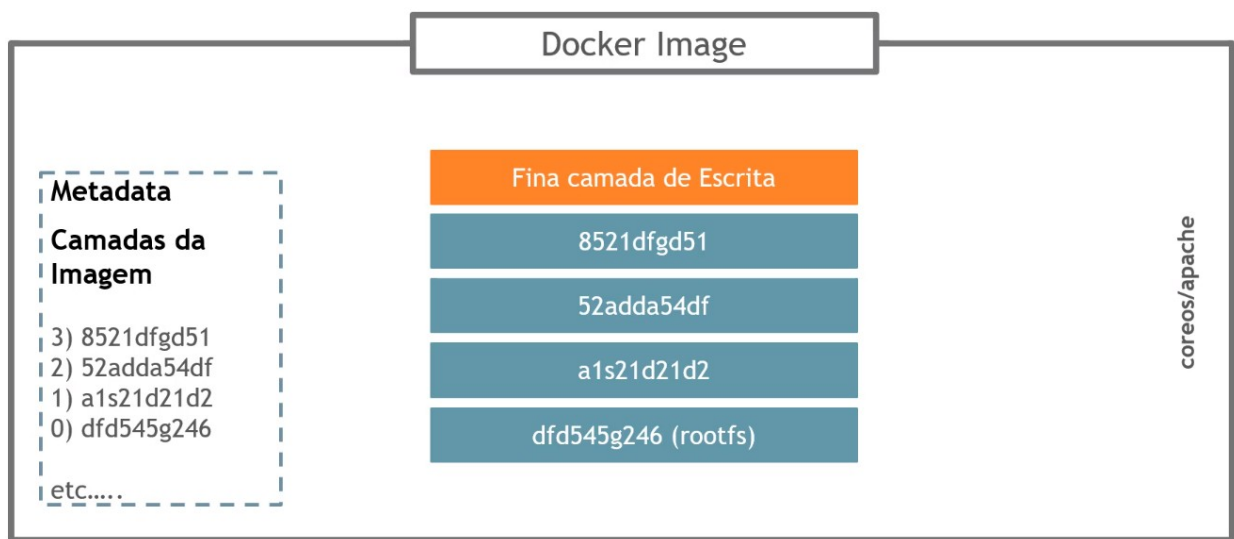
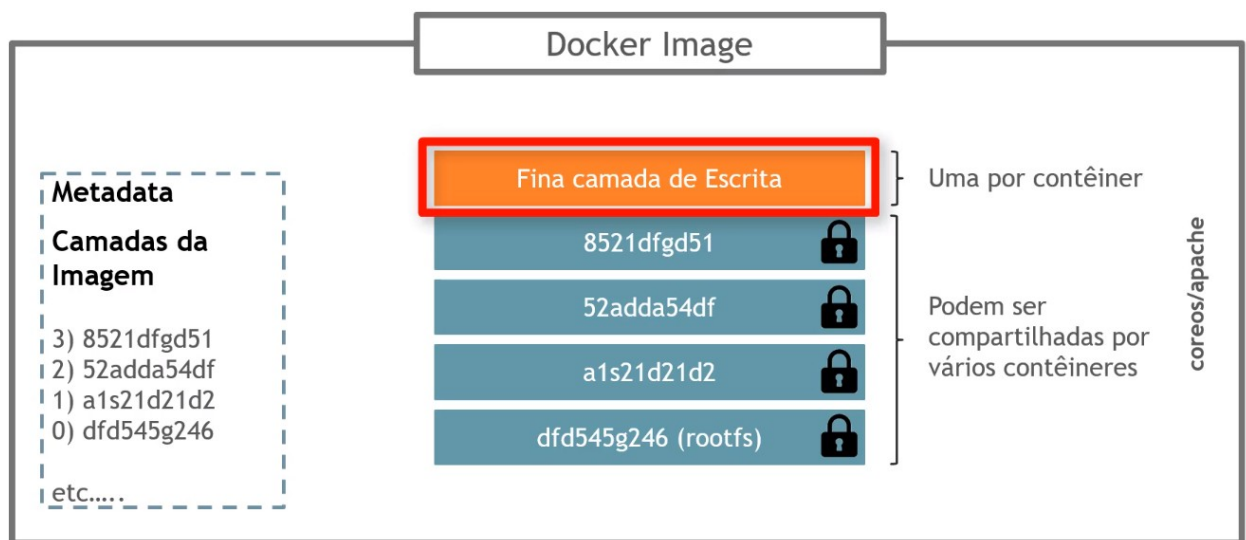


Figura 3: Então, todos contêneires obtém sua própria camada de escrita no topo



Todo contêiner obtém a sua própria camada de escrita

Figura 4: As camadas de baixo são somente leitura, enquanto a camada superior é onde todas as alterações são feitas

Os contêineres são statefull, ou seja, mantém seu estado após encerrar a conexão. Vale destacar que a última camada vai vim por padrão vazia, esse espaço é consumido sempre que fazemos alterações em nosso contêiner, portanto, um contêiner que sofre grande quantidade de alterações, essa camada irá crescer rapidamente e consumirá mais espaço. Já em contêiner que não grava muito, essa camada permanecerá eficiente no quesito armazenamento.

**IMPORTANTE:** O rootfs de um contêiner não se torna realmente gravável, em um cenário normal de inicialização do linux: o rootfs é montado como somente leitura e depois montado novamente como gravável. No mundo dos containers, isso nunca acontece.

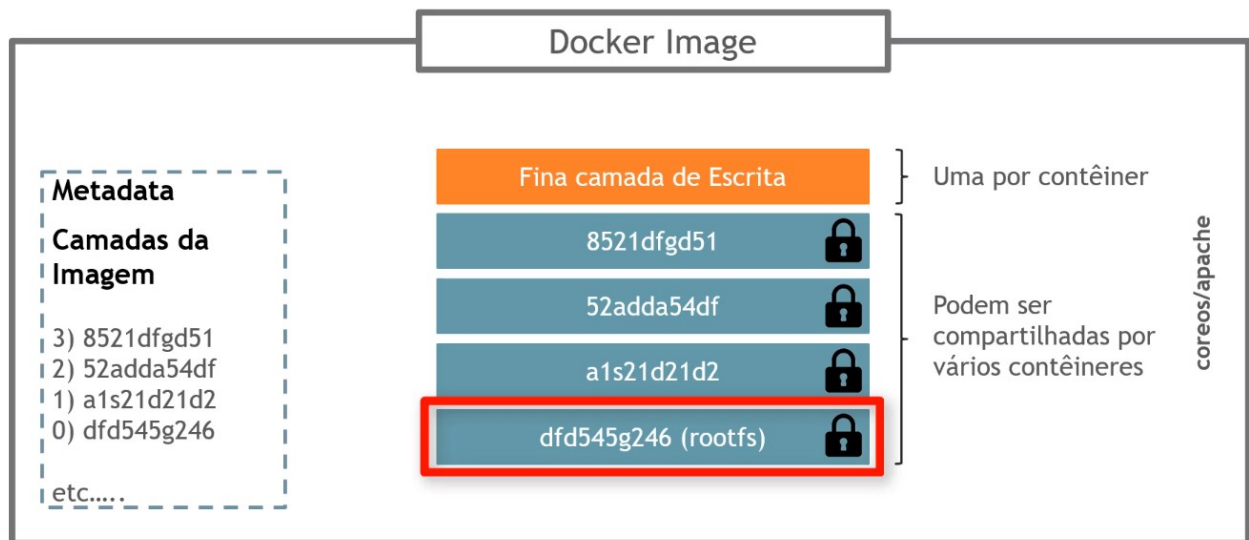


Figura 5: O rootfs está na camada inferior das docker images, portanto só pode ser acessada, somente p/ leitura, mas graças a tecnologia de union mounts, o sistema grava na camada superior, como se fosse um sistema de arquivos normal.

Lembre- se apesar de contêineres serem ambientes totalmente isolados p/ execução de aplicação e do Kernel linux, existem algumas diferenças sutis entre contêineres e sistema operacionais completos.

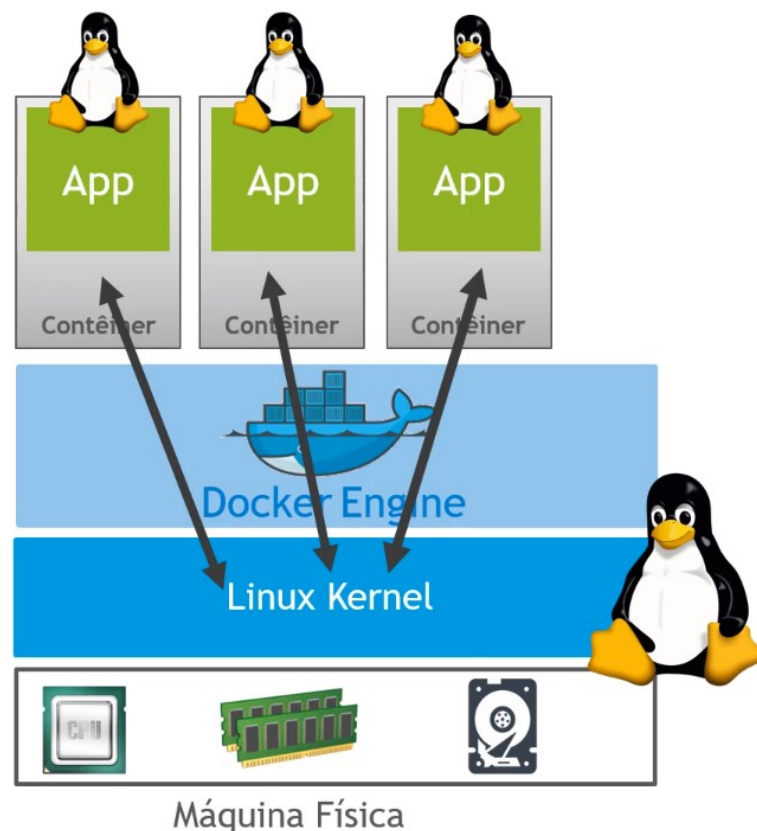


Figura 6: Todos contêineres de uma máquina compartilham o kernel do sistema operacional hospedeiro

O processo de inicialização também tem algumas diferenças, mas p/ nossas aplicações nada disso importa. Porque elas irão ter uma ambiente de runtime linux .  
Para nossa visão de desenvolvedor, todos os containeres terão um ambiente de execução isolado, onde nossas aplicações executam