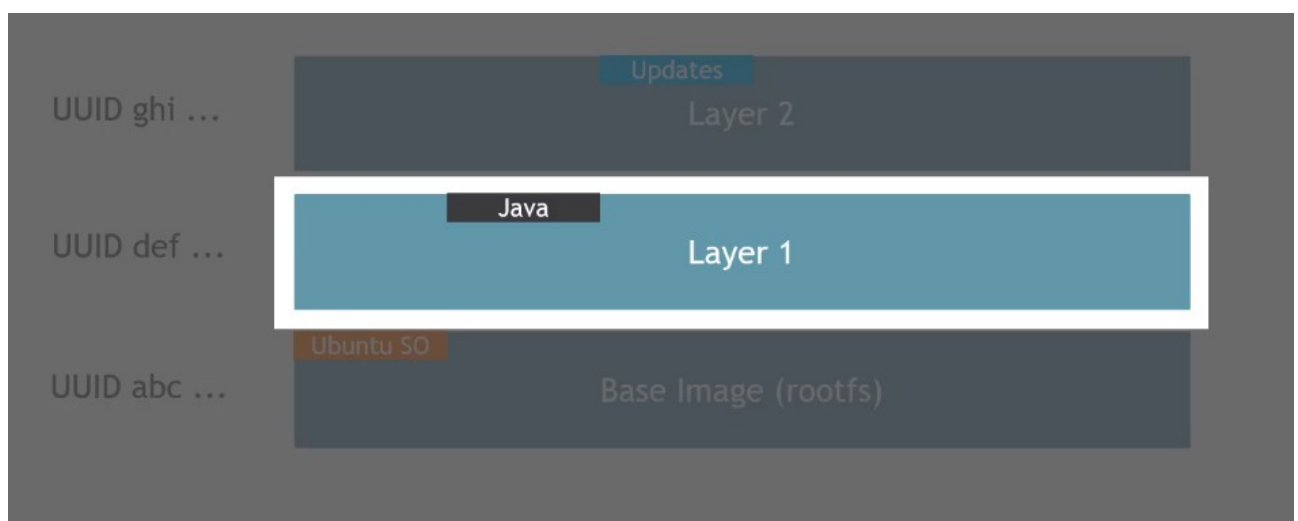
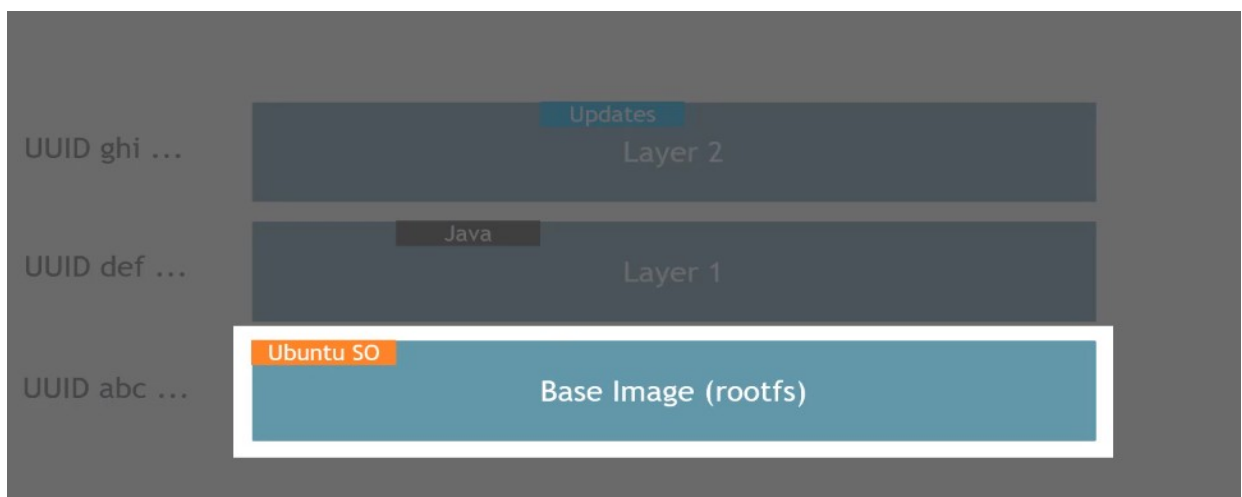
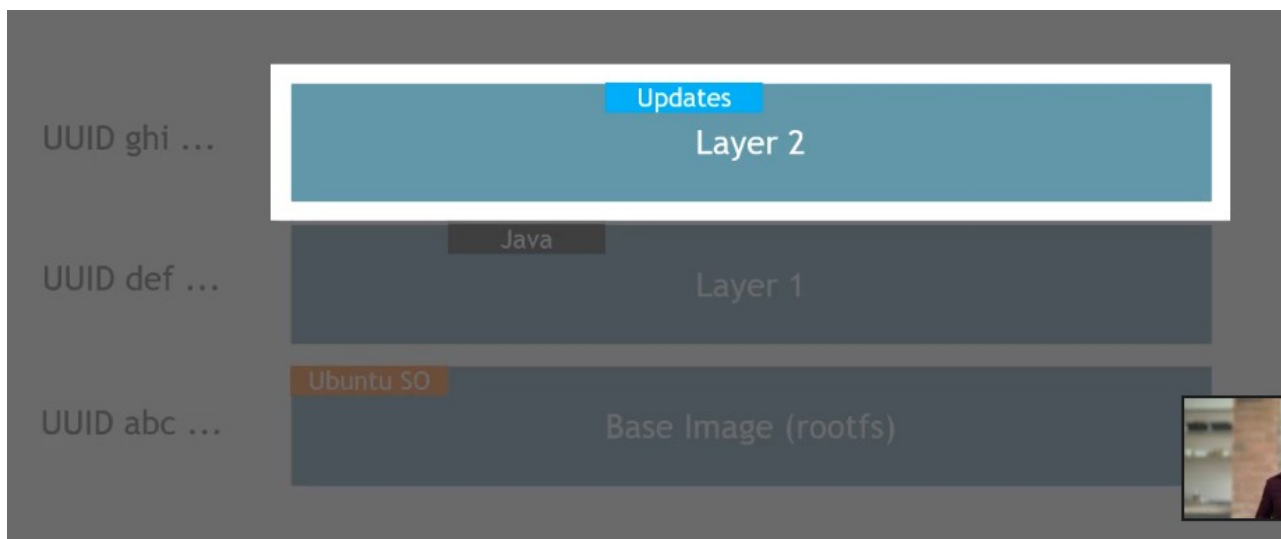


Figura 1: Cada camada/imagem, recebe se uniqueId, esses ids são armazenados na imagem do docker juntos aos metadados, que informam ao docker como criar seus contêineres em tempo de execução

No exemplo:



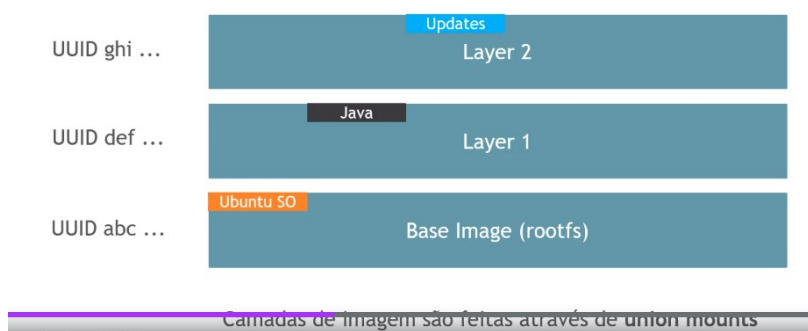


Os metadados diriam ao docker colocar unique id “abc” na camada base, na camada 1 “def”, na camada 2 “ghi”. Por fim, todas as camadas são combinadas em uma única visualização. Os dados da camadas superiores irão ocultar os dados das camadas inferiores

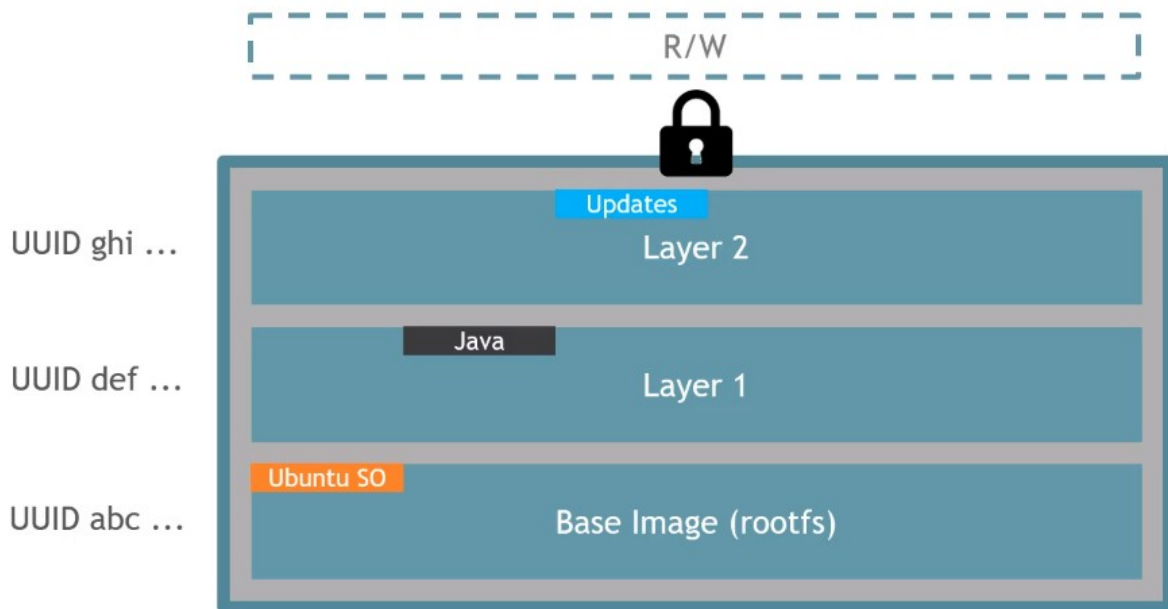


Caso ocorra algum conflitos, por exemplo : alterações em etc/ na camada superior e isso sobrescreve o que está definido na camada base. O docker resolve com conflito definindo o que tá na camada superior. LEMBRE SE DISSO, AS CAMADAS SUPERIORES SEMPRE VENCEM QUANDO HÁ CONFLITOS.

Por baixo dos panos, o docker utiliza toda tecnologia provida pelo “union mounts”. O union mounts permite montar vários sistemas de arquivos um sobre outros, se comportando como se fosse um único sistema de arquivos, assim, com union mounts em vez de montar um sistema de arquivos c/ diretórios como /root ou home. *É possível montar sistemas de arquivos um sobre outros, combinando todas as camadas em uma única visualização. O union mounts exhibe ao sistema, a aplicação e a nós um único sistema de arquivos*



O docker lida com union mounts da seguinte maneira, ele monta todas as imagens como somente leitura, e então adicionar uma nova camada no topo. Na prática, quando iniciamos um contêiner, somente a camada superior permite gravação. Lembre-se que na base na parte inferior, tem nosso boot e sistema de arquivos raiz.



Por isso, a terminologia mais adequada é dizer que estamos iniciando contêineres, em vez de dizer que estamos “bootando”. O processo de iniciar o contêiner é igual o processo de iniciar uma vm linux completa, como se houvesse código bios, ou código de nível de máquina, localizado no kernel. Na prática, o kernel já foi inicializado na máquina hospedeira, então compartilhamos o mesmo kernel comum entre os contêineres que estão em execução. O Docker usa Union Mounts e AWFS.

Em tempo de execução, as alterações são commitadas na camada r/w, se precisarmos atualizar um dos arquivos em uma das camadas inferiores que são somente leitura, primeiro precisamos copiá-lo p/ camada superior e em seguida precisamos armazenar as alterações feitas na mesma camada