

## O que aprendemos?

Aprendemos neste capítulo:

- Comandos básicos do Docker para podermos baixar imagens e interagir com o container.
- Imagens do Docker possuem um sistema de arquivos em camadas (Layered File System) e os benefícios dessa abordagem principalmente para o download de novas imagens.
- Imagens são *Read-Only* sempre (apenas para leitura)
- Containers representam uma instância de uma imagem
- Como imagens são *Read-Only* os containers criam nova camada (layer) para guardar as alterações
- O comando Docker `run` e as possibilidades de execução de um container

Segue também uma breve lista dos comandos utilizados:

- `docker ps` - exibe todos os containers em execução no momento.
- `docker ps -a` - exibe todos os containers, independente de estarem em execução ou não.
- `docker run -it NOME_DA_IMAGEM` - conecta o terminal que estamos utilizando com o do container.
- `docker start ID_CONTAINER` - inicia o container com id em questão.
- `docker stop ID_CONTAINER` - interrompe o container com id em questão.
- `docker start -a -i ID_CONTAINER` - inicia o container com id em questão e integra os terminais, além de permitir interação entre ambos.
- `docker rm ID_CONTAINER` - remove o container com id em questão.
- `docker container prune` - remove todos os containers que estão parados.
- `docker rmi NOME_DA_IMAGEM` - remove a imagem passada como parâmetro.
- `docker run -d -P --name NOME dockersamples/static-site` - ao executar, dá um nome ao container.
- `docker run -d -p 12345:80 dockersamples/static-site` - define uma porta específica para ser atribuída à porta 80 do container, neste caso 12345.

