* 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.
* docker run -d -e AUTHOR="Fulano" dockersamples/static-site **-** define uma variável de ambiente AUTHOR com o valor *Fulano* no container criado.
* docker run -v "CAMINHO\_VOLUME" NOME\_DA\_IMAGEM - cria um volume no respectivo caminho do container.
* docker inspect ID\_CONTAINER - retorna diversas informações sobre o container.
* docker build -f Dockerfile **-** cria uma imagem a partir de um Dockerfile.
* docker build -f CAMINHO\_DOCKERFILE/Dockerfile -t NOME\_USUARIO/NOME\_IMAGEM **-**constrói e nomeia uma imagem não-oficial informando o caminho para o Dockerfile.
* docker login **-** inicia o processo de login no Docker Hub.
* docker push NOME\_USUARIO/NOME\_IMAGEM **-** envia a imagem criada para o Docker Hub.
* docker pull NOME\_USUARIO/NOME\_IMAGEM **-** baixa a imagem desejada do Docker Hub.
* hostname -i **-** mostra o ip atribuído ao container pelo docker (funciona apenas dentro do container).
* docker network create --driver bridge NOME\_DA\_REDE **-** cria uma rede especificando o driver desejado.
* docker run -it --name NOME\_CONTAINER --network NOME\_DA\_REDE NOME\_IMAGEM **-**cria um container especificando seu nome e qual rede deverá ser usada.

Resumo final:

Segue a lista com os principais comandos utilizados durante o curso:

* Comandos relacionados à informações
  + docker version **-** exibe a versão do docker que está instalada.
  + docker inspect ID\_CONTAINER **-** retorna diversas informações sobre o container.
  + docker ps **-** exibe todos os containers em execução no momento.
  + docker ps -a **-** exibe todos os containers, independente de estarem em execução ou não.
* Comandos relacionados à execução
  + docker run NOME\_DA\_IMAGEM **-** cria um container com a respectiva imagem passada como parâmetro.
  + docker run -it NOME\_DA\_IMAGEM **-** conecta o terminal que estamos utilizando com o do container.
  + 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.
  + docker run -v "CAMINHO\_VOLUME" NOME\_DA\_IMAGEM **-** cria um volume no respectivo caminho do container.
  + docker run -it --name NOME\_CONTAINER --network NOME\_DA\_REDE NOME\_IMAGEM **-** cria um container especificando seu nome e qual rede deverá ser usada.
* Comandos relacionados à inicialização/interrupção
  + docker start ID\_CONTAINER **-** inicia 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 stop ID\_CONTAINER **-** interrompe o container com id em questão.
* Comandos relacionados à remoção
  + 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.
* Comandos relacionados à construção de Dockerfile
  + docker build -f Dockerfile **-** cria uma imagem a partir de um Dockerfile.
  + docker build -f Dockerfile -t NOME\_USUARIO/NOME\_IMAGEM **-** constrói e nomeia uma imagem não-oficial.
  + docker build -f Dockerfile -t NOME\_USUARIO/NOME\_IMAGEM CAMINHO\_DOCKERFILE **-** constrói e nomeia uma imagem não-oficial informando o caminho para o Dockerfile.
* Comandos relacionados ao Docker Hub
  + docker login **-** inicia o processo de login no Docker Hub.
  + docker push NOME\_USUARIO/NOME\_IMAGEM **-** envia a imagem criada para o Docker Hub.
  + docker pull NOME\_USUARIO/NOME\_IMAGEM **-** baixa a imagem desejada do Docker Hub.
* Comandos relacionados à rede
  + hostname -i **-** mostra o ip atribuído ao container pelo docker (funciona apenas dentro do container).
  + docker network create --driver bridge NOME\_DA\_REDE **-** cria uma rede especificando o driver desejado.
* docker-compose up - sobe os serviços criados
* docker-compose down - para os serviços criados.
* docker-compose ps - lista os serviços que estão rodando.
* docker exec -it alura-books-1 ping node2- executa o comando ping node2 dentro do container alura-books-1