* docker ps **-** exibe todos os containers em execução no momento.
* docker ps -a **-** exibe todos os containers, independentemente 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 -P -e AUTHOR="Fulano" dockersamples/static-site **-** define uma variável de ambiente AUTHOR com o valor *Fulano* no container criado.
* docker network create –driver bridge minha-rede (comando para criar a própria rede.)
* docker run -it - -name meu-container – network minha-rede (subindo container na rede criada)
* docker network ls (ver a lista da rede)
* docker network inspect minha-rede (ver os container da rede criada)

Exemplo de projeto com 2 containers se comunicando (app e bd)

docker pull douglasq/alura-books:cap05

docker pull mongo

docker run –d - - name meu-mongo - - network minha-rede mongo

docker run - - network minha-rede –d –p 8080:3000 douglasq/alura-books:cap05

(instalar ping ubuntu: apt-get install –y iputils-ping)

(hostname –i: mostra ip atribuído pelo docker)



