docker run -p 5001:5000 -m 512m --cpu-quota 5000 -d –restart=always in28min/todo-rest-api-h2:1.0.0.RELEASE

-p {hostPort}:{dockerPort} -> para redirecionar para a porta

-d -> para rodar em background

-m -> qtd de memória em megabytes

--cpu-quota -> qtd de CPU em % , 5000 significa 5%, 100000 significa 100%

--restart -> o padrão do restart é ‘no’. Se estiver ‘always’, este container executará quando o docker iniciar ou reiniciar

Docker stats -> ver qto de memoria e CPU cada container está utilizando no sistema

Docker container ls -a -> ver todos os containers que rodaram ou que estão rodando

-a -> para ver os que não estão rodando e foram fechados

Docker container stop {id} -> parar container específico

Id -> somente o inicio do ID é necessário

Docker container kill {id} -> mata o container

Docker images -> todas as imagens salvas localmente

Docker tag {repository}:{old-tag} {repository}:{new-tag} -> criar nova tag apontando para a mesma imagem

Docker pull mysql -> faz o pull da imagem do hub do Docker. E ele procura pela tag default “latest”, ou seja vai baixar o mysql:latest

Docker Search mysql -> busca tudo que tenha mysql

Docker image history {IMAGE ID} -> todos os comandos que são executados para criar a imagem

Docker image inspect {IMAGE ID} -> todas as configurações da imagem e suas tags

Docker container inspect {ID} -> todas as configurações do container

Docker image remove {IMAGE ID} -> deleta a imagem apenas localmente

Docker pause {ID} - > pausa o container, para despausar utilizar o comando “unpause”

Docker logs -f {ID} -> faz um tail nos logs do container

Docker container prune -> remove os container com status “exited” da lista

Docker events -> monitorar mount , unmount, start e stop das imagens/volumes

Docker top {ID} -> mostrar os processos que estão rodando nesse container

Docker system df -> ver as imagens e containers que estão sendo gerenciados pelo docker