**CURSO DOCKER**

**USUÁRIO**

1. **Para iniciar um container utilizamos o código abaixo:**  
   docker container run debian bash --version
2. **Para iniciar no modo interativo usamos o comando abaixo:**  
   docker container run -it debian bash
3. **Para dar um nome a um container fazemos o seguinte:**

docker container run --name mydeb2 -it debian bash

1. **Para dar start de um container nomeado (pois o run sempre cria um container novo a partir da imagem):**

docker container start mydeb2 -it Debian bash

1. **Parar um container:**

docker container stop mydeb2

1. **Mapear uma porta com nginx:**

docker container run -p **{{porta\_pc}}**:**{{porta\_container}}** nginx  
**ex:** docker container run -p 8080:80 nginx

1. **Mapear um volume com nginx:**

docker container run --name my-nginx-server -p 8080:80 -v '/mnt/e/Documentos Usuario/HARLLEM/curso-docker/ex-volume/html':/usr/share/nginx/html nginx

1. **Container em background (Daemon -d)**

docker container run -d --name ex-daemon-basic -p 8080:80 -v '/mnt/e/Documentos Usuario/HARLLEM/curso-docker/ex-volume/html':/usr/share/nginx/html nginx

1. **Remover container, imagem ou volume:**  
   docker image rm image-name

docker volume rm volume-name

docker container rm container-name

1. **Listar container, imagem ou volume:**

docker image ls

docker volume ls

docker container ls

1. **Inspecionar uma imagem:**

docker image inspect redis:latest

**DEIXANDO DE SER USUÁRIO**

1. **Tag em imagem:**

docker image tag redis:latest my-name

1. **Comandos usados após docker image:**

* pull (baixa imagem do docker-hub)
* ls (lista as imagens no repositório local)
* rm (remover uma imagem) {{nomeImage, tag, id...}}
* inspect (mostra um json sobre a imagem)
* tag imagem-origem tag-name (colocar uma tag em uma imagem)
* build (recebe um dockerFile para gerar uma imagem)
* push (publica no registry local ou registry do docker-hub)