Docker

Principais comandos (sintaxe: docker “comando”)

run: criação do container. “-p” representa de qual porta sai do container para qual porta vai na máquina.

- **Docker run --name “nome-do-container” -p 80:80 -d nginx**

ps: Lista os containers;

info: informações do docker;

images: imagens utilizadas para criar o container;

exec: executa outro binário do container, que por exemplo, executa o ssh para possibilitar navegação no container.

stop/start;

logs: lista os outputs e logs do sistema;

inspect: lista configurações do container, como por exemplo, mapeamento de volume;

pull: baixar as imagens do repositório;

commit: comitar modificações nos containers;

tag: melhora versionamento;

login/logout: logar no repositório privado ou público;

push: armazena a imagem após buildada;

search: procurar imagem;

rm: remover container;

rmi: remover imagem;

export/import: exporta ou importa container gerando imagem (merge das camadas em apenas uma);

save/load: salvar ou carregar a imagem de docker em arquivo de texto.

Dockerfile

```
#Layers
FROM busybox
RUN echo hello > /hello
RUN echo world >> /hello
RUN touch remove_me /remove_me
RUN rm /remove_me
```

exemplo de uso dos principais comandos:

```
##Instalar explorador de imagem
#https://github.com/wagoodman/dive
wget
https://github.com/wagoodman/dive/releases/download/v0.9.2/dive_0.9.2_linux_amd
64.deb
sudo apt install ./dive_0.9.2_linux_amd64.deb

# Docker RUN
# Comando utilizado para criar um container
docker run --name newcontainer hello-world
docker run --name hello -d busybox sleep 3600
docker run --name site -d -p 80:80 nginx

#Docker PS
#Lista os container em execução, para listar os que não estão precisamos
colocar o parâmetro -a
docker ps -a

#Docker INFO
#Exibe um sumário dos nossos container, como também especificações do nosso
docker
docker info

#Docker EXEC
#Adiciona um processo a mais no container
#Vamos criar uma pasta dentro do container
docker exec hello mkdir teste
# Acessamos o container com o servico SH
docker exec -it hello sh

#Docker STOP, START
#Paramos nosso container
docker stop hello
#Iniciamos nosso container
docker start hello

#Docker LOGS
#coletamos o output do nosso container, ótimo para debugar uma aplicação
```

```
docker logs site

#Docker PULL
docker pull hello-world

#Docker COMMIT
docker commit --author="Diogo Sperandio" --message="Imagem com commit" hello
hello

#Docker TAG
#Preparando para docker hub
docker tag hello dsperax/hello:1.0
#Trocando um nome de um repositorio
docker tag hello-world ola-mundo

#Docker LOGIN,LOGOUT
#Logar no repositório local, ou público. Por padrão logamos no Docker HUB
docker login <usuário>

#Docker PUSH
#Vamos versionar nosso repositório/imagem ao docker hub
docker push dsperax/hello:1.0

#Docker SEARCH
#Procura por uma imagem no repositório
docker search <consulta>

#Docker RM
#Remove um container previamente parado
docker rm newcontainer
#para remover um container em execução é nesessário o parâmetro -f
docker rm -f site

#Docker RMI
#Remove um repositório/imagem local, se algum container estiver parado que
utiliza essa imagem deverá passar o parâmetro -f
docker rmi hello-world

#Docker EXPORT,IMPORT
#Exporta o container mergeando as suas camadas
docker export hello > export.tar
#Importa o arquivo gerado, criando uma imagem do container a partir dela
cat export.tar | docker import - hello-export:latest

#Docker SAVE,LOAD
#Exporta a imagem em um arquivo
docker save dsperax/hello:1.0 > save.tar
#Importa o arquivo gerado
docker load < save.tar
```