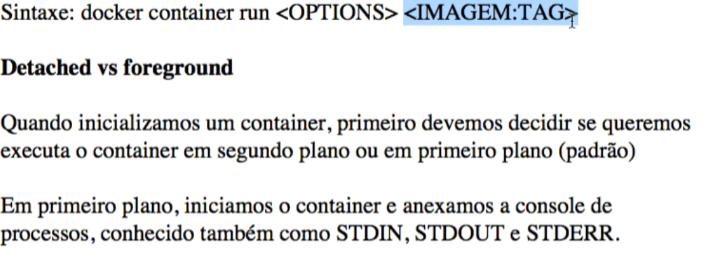
**DOCKER**

# Comandos CMD

# Sintaxe da CLI

**- docker <comando> <sub-comando> <opcoes>**

****

# Versão do docker

**- docker Version**

# Criar container com nome

**docker container run --name web01 -p 80:80 nginx**

# Executar/criar container em primeiro plano

**docker container run -p 80:80 nginx**

# Criar container em segundo plano

**docker container run -d -p 80:80 nginx**

# Parar container criado

**docker stop web01**

# Startar container criado

**docker start web01**

# Remover containers

**docker container rm <nome ou id do container>**

# Remover containers forçadamente

**docker container rm -f <nome ou id do container>**

# Listar containers em execução

**docker container ls**

# Listar todos containers

**docker container ls – a**

# Informações Gerais da docker

**docker info**

# Visualizar logs da docker

**docker container logs <nome ou id do container>**

# Acompanhar logs da docker

**docker container logs -f <nome ou id do container>**

# Visualizar processos que o container está rodando

**docker container top <nome ou id do container>**

# Visualizar status de utilização que os containers estão rodando

**docker container stats**

# Informações de rede do container

**docker container inspect <nome ou id do container>**

# Criar nova Rede

**docker network create <nome da rede>**

# Remover uma rede

**docker network rm <nome da rede>**

# Listar Redes

**docker network ls**

# Informações de rede

**docker container inspect <nome da rede>**

# Informações de rede especificas do container

**docker container inspect -f {{.NetworkSettings.Networks}} <nome ou id do container>**

# Criar container passando rede

**docker container run --name <nome ou id do container> -d --network <nome da rede> nginx:alpine**

# Criar container rede bridge

**docker container run --name h\_bridge -d nginx:alpine**

# Criar container rede none

**docker container run -d --name h\_none -network none -d nginx:alpine**

# Acessar terminal de um container

**docker container exec -it <nome container> sh**

# Criar rede com subnet e gateway especifico

**docker network create my\_custom\_net2 --subnet 192.168.134.0/24 --gateway 192.168.134.1**

# Criar container associando a rede

**docker container run --name <nome do container> -d --network <nome da rede> nginx**

# Remover todas as redes não utilizadas

**docker network prune**

# Fazer ping entre duas containers

**docker container exec -it <nome do container 1>ping <nome do container 2>**

# Linkar duas containers

**docker container run -d –name <nome do container 1> –link <nome do container 2>nginx:alpine**