



# Docker Cheat Sheet

## Comandos Básicos



**João Neto**

**Github:** j040n3t0

**Skype:** joaoneto.ti

**E-mail:** joaojose.ti@gmail.com

**## Baixar e executar o container "Hello World" (Hello World?! Quem nunca??)**

```
# docker run hello-world
```

**## Listar os contêineres que estão executando**

```
# docker ps
```

**## Listar todos os contêineres independente do estado (em execução, parados)**

```
# docker ps -a
```

**## Baixar uma imagem (Ex. kali-linux)**

```
# docker pull kalilinux/kali-linux-docker
```

**## Executar um comando com base em uma imagem (Obs. Caso você não tenha baixado a imagem, a mesma será baixada antes de executar o comando. Neste caso logo após executar o comando irá aparecer no terminal a palavra "H4ck3d!")**

```
# docker run ubuntu echo "H4ck3d!"
```

**## Executar um container passando um comando para ser executado assim que o mesmo for iniciado.**

```
# docker run ubuntu bash -c "apt-get -y install nginx"
```

**## Executar um container no modo interativo.**

```
# docker run -it ubuntu
```

→ CTRL + D (sair do container, porém o mesmo é encerrado)

→ CTRL + P ; CTRL + Q (sair do container, mantendo o estado de execução)

**## Retornar ao console do container**

```
# docker attach CONTAINER_ID
```

**## Iniciar um container**

```
# docker start ID_Container
```

**## Iniciar um container no modo interativo**

```
# docker start -a -i ID_Container
```

**## Parar um container**

```
# docker stop ID_Container
```

**## Parar um container sem esperar o tempo default de 10 segundos. (-t tempo em segundos)**

```
# docker stop -t 0 ID_Container
```

**## Rodar um container no modo dettach (-d), isso serve para iniciar o container sem travar o terminal.**

```
# docker run -d --name kali kalilinux/kali-linux-docker
```

### ## Renomear um container

```
# docker rename discourse_app disc_app
```

### ## Apagar um container

```
# docker container rm cc3f2ff51cab cd20b396a061
```

### ## Apagar todos os containers parados

```
# docker container prune
```

### ## Apagar todos os containers parados com mais de 12 horas.

```
# docker container prune --filter "until=12h"
```

### ## Listar imagens baixadas

```
# docker image ls
```

### ## Remover imagens

```
# docker rmi <ID>
```

```
# docker image rm 75835a67d134 2a4cca5ac898
```

### ## Rodar um container no modo interativo, alterando o hostname e o nome do container.

```
# docker run -it --hostname kali --name kali kalilinux/kali-linux-docker
```

### ## A opção "-P" cria um NAT da máquina física para o container (Modo aleatório de portas)

```
# docker run -d -P dockersamples/static-site
```

### ## A opção "-p" cria um NAT da máquina física para o container, porém é possível definir a porta

```
# docker run -d -p 80:80 --name meu-site dockersamples/static-site
```

### ## Lista as portas abertas pelo container

```
# docker port ID
```

### ## A opção "-e" cria uma variável de ambiente de ambiente para ser exibido no site.

```
# docker run -d -p 80:80 -e AUTHOR="João Neto - LoL" --name meu-site dockersamples/static-site
```

### ## Utiliza o retorno dentro de " \$( ) " como parâmetro do primeiro comando

```
# docker stop -t 0 $(docker ps -q)
```

### ## Docker inspect fornece informações detalhadas sobre o container (Endereço IP, Volumes, hostname, etc)

```
# docker inspect ID_Container
```

### ## Docker log mostra os logs gerados pelo Container

```
# docker log ID_Container
```

## VOLUME

### ## A opção "-v" cria um link simbólico da máquina física com o container, isso fazer com que os dados sejam preservados caso o container seja removido, ou caso haja a necessidade de compartilhar dados entre containers.

```
# docker run -it -p 80:8080 -v "/home/joaoneto/docker-projects/simpleweb:/var/www" -w "/var/www" --hostname kali --name "web-kali" kalilinux/kali-linux-docker
```