

## Seção 6 - Criando e Personalizando Aplicações Simples

### 1. Criação de imagens personalizadas a partir de um container



#### Iniciar container

```
# docker run -it --name criandocontainer_web debian /bin/bash
```

#### Realizar mudança

```
# apt-get update && apt-get install apache2
# apt-get install nano
# rm -f /var/www/html/index.html
# nano /var/www/html/index.html
```

#### Persistir mudança

```
# docker commit criandocontainer_web debian_web
```

- `docker commit [options] [container] [repository]`



## Iniciar novo container

```
# docker run -d --name minhaaplicacao -p 8080:80
debian_web /usr/sbin/apache2ctl -D FOREGROUND
```

## “Criando aplicações simples”

Listar todos os containers (ativos e inativos):

```
sudo docker ps -a
```

**A partir do container, com imagem base debian, criar uma aplicação web personalizada:**

```
sudo docker exec -it [CONTAINER_NAME] /bin/bash
sudo apt-get update && apt-get install apache2
```

*\* Editar arquivos do apache, página e outros, conforme sua necessidade*

*\* Caso container com seu nome não esteja iniciado, inicio-o com docker start*

**Depois de feitas as alterações, dar commit:**

```
sudo docker commit [CONTAINER_NAME_ALTERADO] debian_web_[SEUNOME]
```

**Visualizar resultado do commit, listando todas as imagens criadas:**

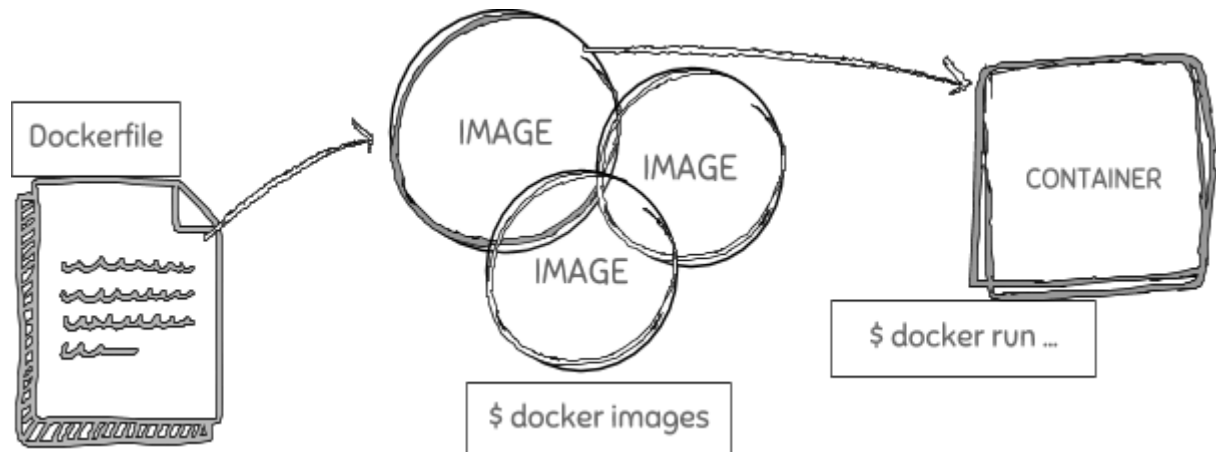
```
sudo docker images
```

**Iniciar aplicação web personalizada:**

```
sudo docker run -itd --name minhaaplicacao -p 8081:80 debian_web_[SEUNOME]
/usr/sbin/apache2ctl -D FOREGROUND
```



## 2. Uso do Dockerfile



- **Dockerfile** ⇒ arquivo de configuração que contém instruções para a criação de uma imagem Docker

- Prover uma maneira eficiente para criar ou personalizar aplicações;
- Solução mais completa do que o uso do “docker commit”;
- A melhor forma é criar/modificar um Dockerfile;

## INSTRUÇÕES

<b>FROM</b>	=	Indica a imagem base;
<b>MAINTAINER</b>	=	Informa o autor da imagem;
<b>RUN</b>	=	Indica o que será executado para criação da imagem;
<b>EXPOSE</b>	=	Informa qual porta o container docker irá "escutar". Para expor essas portas, é preciso usar "-p" ou "-P" no “docker run”;
<b>CMD</b>	=	define um comando padrão que será executado quando o container é criado;
<b>ENTRYPOINT</b>	=	Especifica o que o que será executado ao iniciar o container. Precede a instrução CMD;
<b>ADD</b>	=	Copia um arquivo do host para dentro da imagem;

- Instruções ⇒ o que você quer fazer!
- Essas instruções são seguidas de comandos shell script;



Nome do arquivo: Dockerfile

```
FROM debian
MAINTAINER Ricardo <ricardo@teste.net>
RUN apt-get update && apt-get install -y nginx
ADD index.html /var/www/html
EXPOSE 80
CMD ["nginx", "-g", "daemon off;"]
```

Criação da Imagem

```
# docker build -t ricardo/nginx .
```

- Build ⇒ responsável por criar a imagem.
- -t ⇒ parâmetro para informar a 'tag' da imagem.
- nome da imagem ⇒ ricardo/nginx
- "." ⇒ diretório atual. Dentro da pasta onde se encontra arquivo Dockerfile
- **AGUARDAR O PROCESSO DE BUILD**  
pode demorar dependendo de sua conexão, quantos programas você colocou para instalar, etc.



## “Usando o Arquivo Dockerfile”

**Criar diretório docker para salvar o arquivo Dockerfile de exemplo:**

```
mkdir ~/docker
```

**Criar arquivo Dockerfile e index.html nesse diretório**

```
touch ~/docker/Dockerfile
```

```
touch ~/docker/index.html
```

```
<h1>Minha primeira imagem em Docker usando o Dockerfile</h1>
```

**Editar arquivo Dockerfile conforme exemplo:**

```
FROM debian
```

```
FROM ubuntu:12.04
```

```
MAINTAINER Kimbro Staken version: 0.1
```

```
RUN apt-get update && apt-get install -y apache2 && apt-get clean && rm -rf  
/var/lib/apt/lists/*
```

```
ENV APACHE_RUN_USER www-data
```

```
ENV APACHE_RUN_GROUP www-data
```

```
ENV APACHE_LOG_DIR /var/log/apache2
```

```
ADD index.html /var/www/html
```

```
EXPOSE 80
```

```
CMD ["/usr/sbin/apache2", "-D", "FOREGROUND"]
```

**FONTE:**

<https://github.com/kstaken/dockerfile-examples/blob/master/apache/Dockerfile>

**Criar a imagem personalizada:**

```
sudo docker build -t seunome/apache:1.0 ~/docker
```

**Iniciar aplicação web personalizada:**

```
sudo docker run -itd --name minhaaplicacaodockerfile -p 9090:80  
seunome/apache:1.0
```

*\* Depois, acessar o navegador web no endereço <http://localhost:9090>*



## Referências

- ▶ <https://docs.docker.com/engine/reference/commandline/commit/>
- ▶ <https://docs.docker.com/engine/reference/builder/>
- ▶ [https://docs.docker.com/engine/userguide/eng-image/dockerfile\\_best-practices/](https://docs.docker.com/engine/userguide/eng-image/dockerfile_best-practices/)

