

Erik Felipe Olinek de Castilho

## RUNC

Para runc, a primeira coisa a ser feita foi se fazer uma pasta /rootfs para rodar o container.

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS$ sudo mkdir /rootfs
```

Logo depois foram dados os comandos 'sudo mkdir /container-config', 'nano config.json', dentro do config.json foi mudado o nome do container para container1, dentro da pasta /rootfs foi dado o comando 'runc spec', e novamente fora dele foi dado o comando 'sudo runc run container1' para rodar o container.

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS$ sudo mkdir /container-config
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS$ runc spec
```

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS$ sudo runc run container1
/ #
/ #
/ #
```

## LXC

Para os containers em lxc foi dado o comando abaixo para a criação do container com o busybox

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~$ sudo lxc-create -n container2 -t busybox
```

Em seguida, os dois comandos abaixo foram executados para iniciar o container e colocar ele funcionável.

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~$ lxc-ls --active
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~$ sudo lxc-start -n container2
```

Após iniciado, foi executado o comando abaixo para que o container pudesse ser explorado e utilizado, para o busybox, para o debian, o comando de criação foi mudado para 'lxc-create -n c1 -t debian', e depois os mesmos passos foram seguidos.

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS$ sudo lxc-attach -n container2
BusyBox v1.30.1 (Ubuntu 1:1.30.1-7ubuntu3) built-in shell (ash)
Enter 'help' for a list of built-in commands.

/ #
```

E os dois comandos abaixo foram utilizados para explorar a imagem debian/12.

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS$ sudo lxc launch images:debian/12 container3
Creating container3
Starting container3
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS$ sudo lxc exec container3 -- /bin/bash
root@container3:~# echo Explorando o root do debian 12
Explorando o root do debian 12
```

## Docker

Para o Docker, a primeira coisa a se fazer foi configurar o sistema utilizando o comando

```
sudo systemctl start docker
```

Após isso entrei no Dockerhub para explorar, imagens, repositórios já feitos, entre outras coisas e conseguir instruções de configuração do Docker rootless, após isso foi criado o Dockerfile, para realização do docker build, o corpo do arquivo ficou da seguinte maneira:

```
FROM ubuntu:latest
|
RUN apt-get update && apt-get install -y \
    software-properties-common \
    && rm -rf /var/lib/apt/lists/*

# Comando a ser executado ao iniciar o contêiner
CMD ["bash"]
```

Abaixo segue os comandos para realização do docker build e verificação de que eles realmente foram criados, nessa ordem:

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS/Dockerfiles$ docker build -t imagem .

erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS/Dockerfiles$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
my-custom-image	latest	138ab058df34	8 minutes ago	207MB
imagem	latest	138ab058df34	8 minutes ago	207MB
ubuntu	latest	b6548eacb063	5 days ago	77.8MB
busybox	latest	a416a98b71e2	4 months ago	4.26MB
hello-world	latest	9c7a54a9a43c	7 months ago	13.3kB

Em seguida, com os comandos abaixo, presentes na imagem, o sistema rootless foi configurado e explorado com sucesso

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS/Dockerfile$ curl -o rootless-install.sh -fsSL 'https://get.docker.com/rootless'
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS/Dockerfile$ sh rootless-install.sh
# Installing stable version 24.0.7
# Executing docker rootless install script, commit: 98c7397
Aborting because rootful Docker is running and accessible. Set FORCE_ROOTLESS_INSTALL=1 to ignore.
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS/Dockerfile$ export PATH=$HOME/bin:$PATH
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS/Dockerfile$ docker run --rm -it ubuntu:latest
root@b264251256f5:/#
```

```
root@3e69ee84dfe9:/# echo Explorando o container
Explorando o container
root@3e69ee84dfe9:/#
```

Podman

Para o container em Podman foi usado o comando abaixo para instalá-lo.

```
erik@erik-VivoBook-ASUSLaptop-X540MAR-X540MAR:~/Área de Trabalho/Unioeste/TDS$ sudo apt-get update
sudo apt-get install -y podman
```

Após este, foi usado este abaixo para configurar o usuário para execução rootless

```
podman machine init
```

E este para explorar um container rootless, o Alpine Linux foi escolhido.

```
podman run --rm -it alpine
```

Link do git: <https://github.com/Erikfelipe011/TDS>