

# **Implementação de Sistemas em Containers RunC,LXC, Docker e Podman.**

**Aluno:** João Gabriel Fazio Pauli

<https://github.com/JoaoFazio/AP02>

## **\*Runc**

Criei um rootfs com **mkdir meu-rootfs**

**wget**

**[https://busybox.net/downloads/binaries/1.32.0-defconfig-multiarch/busybox-x86\\_64](https://busybox.net/downloads/binaries/1.32.0-defconfig-multiarch/busybox-x86_64) -O**

**meu-rootfs/busybox**

**chmod +x meu-rootfs/busybox**

após isso, criei um arquivo .json descrevendo a configuração básica do meu container com o seguinte conteúdo:

```
{  
  "ociVersion": "1.0.2",  
  "process": {  
    "terminal": true,  
    "user": {  
      "uid": 0,  
      "gid": 0  
    },  
    "args": [  
      "busybox",  
      "sh" ]  
  }  
}
```

```
    "/bin/sh"
  ]
},
"root": {
  "path": "meu-rootfs",
  "readonly": false
},
"hostname": "meu-container",
"mounts": [
  {
    "destination": "/proc",
    "type": "proc",
    "source": "proc"
  },
  {
    "destination": "/dev",
    "type": "tmpfs",
    "source": "tmpfs"
  },
  {
    "destination": "/dev/pts",
    "type": "devpts",
    "source": "devpts"
  },
  {
    "destination": "/dev/shm",
    "type": "tmpfs",
    "source": "shm"
  }
]
```

```
]
}
```

após isso executei o comando **runc run meu-container** para executar e criar o meu container.

Também fiz **mkdir runc** para criar uma pasta do container e **mkdir rootfs** para criar uma pasta rootfs. Depois, entrei no diretório do container com **cd runc**. instalei o SO no diretório do rootfs com **sudo debootstrap stable ./rootfs** e criei um arquivo .json com **runc spec** e executei o container com **sudo runc run hello** .

## \*LXC

Instalei o virt manager com **sudo apt-get install virt-manager**

Instalei o LXC com **sudo apt-get install lxc**

Criei o container **busybox** com **sudo lxc-create -t download -n "meu\_container" /home/TDS/AP02**

**Distribution: busybox**

**Release: 1.36.1**

**Architecture: amd64**

Iniciei o container com **sudo lxc-start -n "meu\_container"**

Baixei o ssh-server com **sudo apt-get install openssh-server**

Criei um container com **sudo lxc-create -t download -n "meu\_container1" /home/TDS/AP02**

**Distribution: debian**

**Release: bookworm**

**Architecture: amd64**

Iniciei o container com **sudo lxc-start "meu\_container1"**

## **LXD + LXC Debian 12 Bookworm**

**Usei: sudo apt install snapd**

**sudo snap install core**

**sudo snap install lxd**

**sudo lxd init**

**sudo lxc launch images:debian/12**

**sudo lxc list**

**sudo lxc start "nome\_do\_container"**

**sudo lxc exec "nome\_do\_container"**

**\*Docker**

Na parte do Docker foi requisitada a exploração do Docker Hub e a realização de um docker build.

### **\*Docker Hub**

Explorando o Docker Hub, consegui entender que o site é composto por um gigantesco repositório de imagens para a criação de contêineres, podendo criar sua própria imagem e postar no Docker Hub para disponibilizar seu uso a comunidade ou podendo utilizar a imagem de outras pessoas.

### **\*Docker build**

Para a criação do Docker build utilizei um editor de texto para a criação da imagem e construí utilizando o comando **docker build -t joaofazio/app-node:1.0 .**

Tendo feito a criação do docker utilizei **docker images** para encontrar no repositório de imagens a ID da minha imagem que é: 9bb56ad8c465

Depois disso, utilizei **docker run -d -p 8081:3000 joaofazio/app-node:1.0** para rodar o docker. Por fim, no navegador digitei **http://localhost:8081** e encontrei meu docker.

Para deixar meu docker rootless utilizei o comando  
**docker run -it --rm --privileged=false**  
**joaofazio/app-node 1.0**

E por fim, explorei um docker rootless da nginx  
utilizando **docker run -d -v**  
**\$PWD/nginx.conf:/etc/nginx/nginx.conf nginx**

## **\*Podman**

Para a parte do podman eu puxei uma imagem nginx  
usando o comando **podman search nginx**  
**--filter=is-official** e depois rodei ele utilizando **podman**  
**run -dt -p 8080:80 docker.io/library/nginx**

Após isso é possível ver o podman criado com **podman**  
**ps** e é hospedado no **localhost:8080** e o docker pode  
ser executado com **podman exec -ti 5061fed4bbfe**.  
Quanto a parte de acesso rootless, o podman não  
precisa inserir o usuário no grupo do podman para ele  
executa-lo.

Também entrei no podman com **podman run -it --run**  
**alpine /bin/sh** e adicionei arquivos do alpine com **apk**  
**add** e após isso verifiquei a versão do alpine com **cat**  
**/etc/os-release**

Por fim, instalei o **VIM** com **apk add vim** e adicionei um arquivo teste com **vim arquivoteste.txt**

Para fazer o upload no github, primeiramente foi criado o repositório no site do github, com o link do repositório sendo esse: **<https://github.com/JoaoFazio/AP02>**

Em seguida foi instalado o Git Bash, que é um terminal de linha de comando que permite aos usuários interagirem com o Git

Então os seguintes comandos foram rodados no Git Bash para realizar o upload no github:

1 - Foi clonado o repositório Git para termos uma cópia local de um repositório remoto

**git clone <https://github.com/JoaoFazio/AP02>.git**

2 - Foi feito as adições dos arquivos

**git add .**

3 - Foi dado um commit nessas alterações

**git commit -m "mensagem do commit"**

4 - E por fim foi dado um **push** no repositório para fazer upload desses arquivos no github

**git push**

