

# Containers - Trabalho TDS

## Criação do repositório

1. Criar repositório remoto, pelo website <https://codeberg.org>
2. Em máquina local, instalar `git`, com o comando (em Fedora 38):  

```
$ sudo dnf install git
```
3. E clonar o repositório remoto com comando  

```
$ git clone https://codeberg.org/gabrielsantos46/trabalho-tds-containers.git
```
4. Após navegar até o diretório do repositório, podemos criar o arquivo `RELATORIO.md`, e gerar `RELATORIO.pdf` com o comando  

```
$ pandoc RELATORIO.md -o RELATORIO.pdf
```
5. Podemos verificar que o repositório remoto está funcionando corretamente, realizando o primeiro `push`, com os seguintes comandos:  

```
$ git add .  
$ git commit -m 'criacao de RELATORIO'  
$ git push
```

## Container RunC

1. Instalar o pacote `RunC`:  

```
$ sudo dnf install runc
```
2. Para configurar o `rootfs`, podemos utilizar a ferramenta `debootstrap`, que instala a base de um sistema Debian em um subdiretório de outro sistema já instalado  

```
$ sudo dnf install debootstrap  
$ mkdir -p runc/rootfs && cd runc  
$ sudo debootstrap stable ./rootfs http://deb.debian.org/debian
```
3. Agora podemos gerar o arquivo de configuração `config.json`, executando:  

```
[runc]$ runc spec --rootless
```

As configurações sobre o container são especificadas neste arquivo. Podemos observar, por exemplo, o caminho autodetectado do diretório `/`, o `hostname`, entre outros. Segue porção do arquivo `config.json`:

```
(...)  
"root": {  
  "path": "rootfs",  
  "readonly": true  
},  
"hostname": "runc",
```

```

"mounts": [
  {
    "destination": "/proc",
    "type": "proc",
    "source": "proc"
  },
  (...)

```

4. Finalmente, podemos executar o container, com o comando `runc run runc` (o nome do container foi automaticamente definido como o nome do diretório, `runc`). Como resultado, devemos obter um shell root dentro do container:

```

$ runc run runc
# test

```

5. O diretório `runc/rootfs` ocupa 301M, e consiste apenas de um sistema mínimo Debian, então não vamos adicioná-lo ao repositório. Para isso vamos voltar ao diretório raiz, e adicionar uma regra ao `.gitignore` deste repositório, assim como criar um arquivo shell que realiza a instalação do container.

```

$ cat >> .gitignore << EOF
runc/config.json
runc/rootfs/*
$ cat >> runc/create.sh << EOF
mkdir rootfs &&
sudo debootstrap stable ./rootfs http://deb.debian.org/debian &&
runc spec --rootless

```

Agora, para recriar o container RunC basta executar `sh create.sh` no diretório `runc`.

6. Ao executar `git add . && git status`, pode-se observar que apenas os arquivos `.gitignore` e `runc/create.sh` foram adicionados como arquivos novos

```

On branch main
Your branch is up to date with 'origin/main'.

```

```

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
    new file:   .gitignore
    modified:   RELATORIO.md
    modified:   RELATORIO.pdf
    new file:   runc/create.sh

```

Container LXC

Container Podman

Container Docker