**CRIAÇÃO DE CONTAINERS COM FIREWALL UTILIZANDO IPTABLES**

**Antonio Paulo da Silva Neto**

**Instituto Federal do Sertão Pernambucano – Campus Salgueiro**

antonio.paulo2@aluno.ifsertao-pe.edu.br

***Abstract.*** *This article presents a step-by-step guide to creating a Docker container that functions as a basic firewall using iptables. The process involves creating a project directory, a Dockerfile, a configuration script (firewall.sh), building the Docker image, and running the container with network permissions. As a practical example, blocking access to YouTube was included.*

***Resumo.*** *Este artigo apresenta um passo a passo para a criação de um container Docker que funciona como um firewall básico utilizando iptables. O processo envolve a criação de um diretório de projeto, um Dockerfile, um script de configuração (firewall.sh), a construção da imagem Docker e a execução do container com permissões de rede. Como exemplo prático, foi incluído o bloqueio de acesso ao YouTube.*

**1. Introdução**

Com o crescimento da virtualização e dos serviços em nuvem, o uso de containers tornou-se comum em ambientes de desenvolvimento e produção. O Docker oferece uma forma leve e eficiente de encapsular aplicações e serviços. Neste contexto, a implementação de funcionalidades de segurança, como firewalls, dentro de containers pode ser útil para simulações e testes.

https://github.com/Toin-Paulo/Firewall-container

**2. Estrutura do projeto**

A criação do container será dividida em cinco etapas:

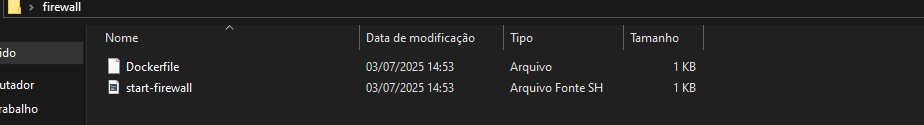
1. Criação do diretório do projeto
2. Criação do arquivo Dockerfile
3. Criação do script firewall.sh
4. Construção da imagem Docker
5. Execução do container

**3. Criação do Diretório**

No explorador de arquivos, crie uma pasta chamada firewall. Dentro dela, crie dois arquivos:

* - Dockerfile (sem extensão)
* - start-firewall.sh

Recomenda-se o uso do Visual Studio Code (VSCode) para criar os arquivos, certificando-se de que os nomes estejam corretos e sem extensões indesejadas como .txt.



(Certifique-se que o diretório esteja assim).

**4. Arquivo Dockerfile**

Abra o arquivo Dockerfile no VSCode e adicione o seguinte conteúdo:

FROM debian:stable

#Instalação de pacotes necessários

RUN apt update && apt install -y iptables iproute2 iputils-ping curl net-tools

#Ativação do encaminhamento de pacotes

RUN echo "net.ipv4.ip\_forward=1" >> /etc/sysctl.conf

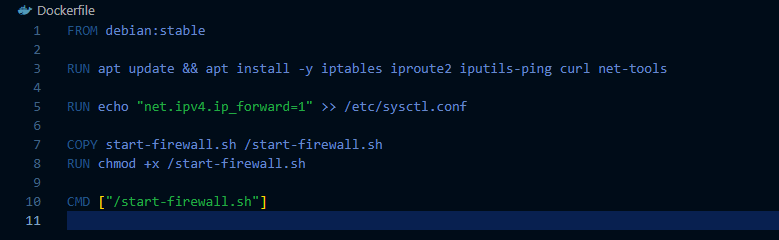
#Cópia e permissão do script de inicialização

COPY start-firewall.sh /start-firewall.sh

RUN chmod +x /[start-firewall.sh](http://start-firewall.sh)

#Execução do script como comando principal do container

CMD ["/[start-firewall.sh](http://start-firewall.sh)"]



**5. Script start-firewall.sh**

Abra o arquivo start-firewall.sh e insira o seguinte conteúdo:

#Ativa o roteamento IPv4 imediatamente

echo 1 > /proc/sys/net/ipv4/ip\_forward

#Regra NAT (masquerade) funcionando para compartilhamento de internet

iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE

iptables -A FORWARD -i eth0 -o eth0 -m state --state RELATED,ESTABLISHED -j ACCEPT

iptables -A FORWARD -s 172.28.0.3 -j ACCEPT

# Bloqueia IPs do YouTube (exemplo)

iptables -A FORWARD -s 172.28.0.3 -d 142.250.0.0/15 -j REJECT

iptables -A FORWARD -s 172.28.0.3 -d 142.251.0.0/16 -j REJECT



(Certifique-se de salvar como start-firewall.sh com permissão de execução.)

**6. Construção da Imagem Docker**

Abra o PowerShell.

Navegue até a pasta do projeto:



Utilize o comando cd e insira o caminho do diretório

Construa a imagem:



Utilize este comando para construir a imagen : docker build -t firewall-container .

**7. Execução do Container**

Para executar o container com permissão de gerenciamento de rede, utilize:



docker run -it --cap-add=NET\_ADMIN firewall-container

Se o processo ocorrer corretamente, o firewall estará ativo e o acesso ao YouTube será

bloqueado.

**References**

- Docker Documentation. Disponível em: https://docs.docker.com

- IPTables Tutorial. Disponível em: https://www.netfilter.org

- Ubuntu Packages. Disponível em: https://packages.ubuntu.com