**## Projeto de Redes**

# A máquina virtual contém o Endereço IP: 10.0.0.10

Foi criado um arquivo Vagrantfile:

Vagrant.configure("2") do |config|

config.vm.box = "ubuntu/bionic64"

config.vm.network "public\_network", ip: "10.0.0.10", type: "static"

config.vm.provision "docker" do |docker|

config.vm.provision "shell", inline: <<-SHELL

sudo apt update

sudo apt install nfs-common

sudo apt install -y bind9

sudo docker pull networkboot/dhcpd:latest

sudo docker pull ubuntu/bind9

sudo docker pull httpd:2.4

sudo docker pull fauria/vsftpd

sudo docker pull itsthenetwork/nfs-server-alpine

SHELL

end

end

* config.vm.box = "ubuntu/bionic64": **Define a imagem** base da máquina virtual como o Ubuntu 18.04 (bionic64).
* config.vm.network "public\_network", ip: "10.0.0.10", type: "static": Configura a interface de rede da máquina virtual para usar um endereço IP estático (10.0.0.10) na rede pública.
* config.vm.provision "docker": Define a configuração de provisionamento para o Docker.
* config.vm.provision "shell", inline: <<-SHELL ...: Executa comandos shell dentro da máquina virtual para instalar pacotes e puxar imagens Docker.

Inicialmente, é necessário subir a máquina virtual, entrando no diretório em que está incluído o Vagrant, digitando o comando “vagrant up” no terminal. Isso permitirá subir a imagem escolhida para a criação da VM. Logo após é possível entrar na máquina virtual digitando o comando “vagrant ssh default”. Default é o nome da VM. Em seguida entre na pasta vagrant com o comando “cd /vagrant” e o após o comando “sudo su” para conseguir as permissões necessárias. O próximo comando a ser digitado é o “./script.sh” que irá executar o script no sistema operacional. Por fim, digite o comando “docker ps -a” que irá listar todos os contêineres do docker, incluindo os que estiverem parados.

**Dockerfile para Servidor NFS**

Este Dockerfile cria um contêiner com um servidor NFS configurado para compartilhar o diretório /shared. Aqui estão as partes importantes do arquivo:

1. Imagem Base (FROM itsthenetwork/nfs-server-alpine:latest):
   * Utilizamos a imagem itsthenetwork/nfs-server-alpine:latest como base. Essa imagem já contém um servidor NFS configurado.
2. Criação do Diretório Compartilhado (RUN mkdir -p /shared):
   * Criamos um diretório chamado /shared dentro do contêiner. Esse diretório será usado para compartilhar arquivos via NFS.
3. Configuração das Exportações NFS (RUN echo "/shared \*(rw,fsid=0,sync,no\_subtree\_check,no\_auth\_nlm,insecure,no\_root\_squash)" > /etc/exports):
   * Definimos as configurações de exportação NFS para o diretório /shared.
   * Essas configurações permitem que qualquer cliente monte o diretório /shared com permissões de leitura e gravação.
4. Exposição das Portas (EXPOSE 2049, EXPOSE 111/udp, EXPOSE 111):
   * As instruções EXPOSE especificam quais portas do contêiner devem ser expostas para comunicação externa.
   * Aqui, estamos expondo as portas 2049 (NFS), 111 (RPC) e 111/udp (RPC via UDP).

# **DHCP**

O servidor DHCP, que é responsável por atribuir endereços IP às máquinas na rede, foi utilizado este serviço: https://hub.docker.com/r/networkboot/dhcpd/ para o conteiner em docker.

Após baixá-lo é necessário criar um arquivo de configuração de dhcp, igual ao arquivo de configurações básicas na pasta config/dhcp.

Utilizando o comando para criar o contêiner, automaticamente o arquivo de configuração será atribuído ao servidor.

Script do DHCP:

subnet 10.0.0.0 netmask 255.255.255.0 {

range 10.0.0.100 10.0.0.200;

option routers 10.0.0.1;

option domain-name-servers 10.0.0.10;

default-lease-time 600;

max-lease-time 7200;

authoritative;

}

Explicações para cada linha:

- subnet 10.0.0.0 netmask 255.255.255.0: Define a sub-rede com endereço IP 10.0.0.0 e máscara de sub-rede 255.255.255.0.

- range 10.0.0.100 10.0.0.200;: Especifica o intervalo de endereços IP que o servidor DHCP pode atribuir às máquinas na rede (de 10.0.0.100 a 10.0.0.200).

- option routers 10.0.0.1;: Define o endereço IP do gateway padrão (roteador) para a rede como 10.0.0.1.

- option domain-name-servers 10.0.0.10;: Especifica o endereço IP do servidor DNS (10.0.0.10) que será fornecido aos clientes DHCP.

- default-lease-time 600;: Define o tempo de concessão padrão para endereços IP (em segundos).

- max-lease-time 7200;: Define o tempo máximo de concessão para endereços IP (em segundos).

- authoritative;: Indica que este servidor DHCP é a fonte autoritativa para informações de configuração na rede.

## Teste do DHCP

Crie uma nova máquina na rede e utilize os comandos "sudo dhclient -r" liberar o endereço IP atual "sudo dhclient" e solicitar um novo endereço IP.

# **DNS**

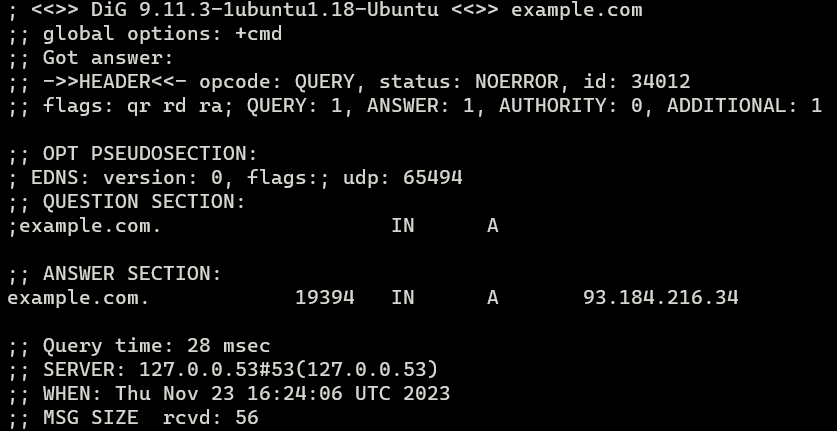
O servidor DNS é responsável por resolver os dominios para IP, foi utilizado este serviço: https://hub.docker.com/r/ubuntu/bind9

Após baixá-lo é necessário criar alguns arquivos de configuração de dns, igual aos arquivos de configurações básicas na pasta config/dns.

Utilizando o comando para criar o contêiner, automaticamente os arquivos de configuração serão atribuídos ao servidor.

## Teste do DNS

Utilize o comando dig example.com, já o domínio usado para criar o servidor foi esse de exemplo. Resultado do comando de teste:



# **APACHE**

O servidor APACHE é responsável por criar um servidor WEB, foi utilizado este serviço: https://hub.docker.com/\_/httpd

Após baixá-lo é necessário criar um arquivo de index.html, semelhante ao arquivo de na pasta web.

Utilizando o comando para criar o contêiner, automaticamente o arquivo index.html será atribuído ao servidor.

## Teste do apache

Abra o navegador nesta pagina "http://10.0.0.10:8080" e verá o seu arquivo de index.html

# **FTP**

O servidor FTP é um protocolo padrão de rede usado para transferir arquivos de um host para outro em uma rede, foi utilizado este serviço: https://hub.docker.com/r/fauria/vsftpd

## Teste do FTP

Abra o terminal e digite ftp admin@10.0.0.10, usuário e senha são admin, admin respectivamente.

# **NFS**

O servidor NFS é responsável pelo compartilhamento de arquivos que permite computadores na rede compartilharem diretorios e arquivos, foi utilizado este serviço: https://hub.docker.com/r/itsthenetwork/nfs-server-alpine

## Teste do NFS

Após executá-lo crie uma pasta para receber o compartilhamento, mkdir /mnt/nfs, depois monte o compartilhamento na pasta, mount -v 10.0.0.10:/nfsshare /mnt/nfs, crie um arquivo para testar, docker exec -it server-nfs touch /nfsshare/teste.txt