Capítulo 11: Redes

☑ Reviewed ☐

▼ ARP | DIG | SSH | CSP | PING | MTR | DNS | IP ROUTE | NETPLAN

▼ ***ARP | IP LINK | IP ADDR | IP ROUTE | PING | TRACEROUTE | NETPLAN***

1. arp

- O `arp` (Address Resolution Protocol) é usado para gerenciar a tabela ARP do sistema, que mapeia endereços IP para endereços MAC na rede local.

```
arp -a #Exibe a tabela ARP atual.


arp -s 10.144.177.22 00:16:3e:1e:4c:e6 #Adicionar uma
entrada ARP:



arp -d 10.144.177.1 #Apagar uma entrada ARP:
```

2. ip link

- O comando ip link é usado para gerenciar interfaces de rede, como ativar/desativar interfaces e configurar suas propriedades.

```
ip link show  ou  ip l show #Mostrar informações sobre
```

```
ip link set eth0 down #Desativar uma interface:


ip link set eth0 up #Ativar uma interface:


ip link set eth0 mtu 1500 #Configurar o MTU de uma inte
```

3. ip addr

- O comando `ip addr` é usado para gerenciar endereços IP nas interfaces de rede.

```
ip addr ou ip add show #Mostrar endereços IP:


ip -6 addr show #Mostrar endereços IPv6:


ip -4 addr show #Mostrar endereços IPv4:


ip -c addr #Mostrar endereços IP com cores (colorido):


ip addr add 192.168.122.22/24 dev eth0 #Adicionar um en


ip addr del 192.168.122.22/24 dev eth0 #Remover um ende
```

4. ip route

- O comando `ip route` é usado para visualizar e manipular a tabela de roteamento do sistema.

```
ip route show (modifica a tabela de rotas) #Mostrar a t
```

5. ping

- O `ping` é uma ferramenta de diagnóstico de rede que verifica a conectividade entre o host e um destino específico.

```
ping -c 5 192.168.1.1 #Enviar 5 pacotes para um IP e ex


ping -q -c 5 192.168.1.20 #Enviar 5 pacotes para um IP
```

6. traceroute

- O `traceroute` é uma ferramenta que mostra o caminho que um pacote percorre até um destino, incluindo todos os roteadores intermediários.

```
traceroute -n www.google.com
```

7. netplan

- O `netplan` é uma ferramenta para configurar interfaces de rede em sistemas Linux que usam o arquivo de configuração YAML.

```
ss -s -t

cd /etc/netplan #Acesse o diretório de configuração do

ls #Liste os arquivos de configuração:

vim 10-lxc.yaml #Edite o arquivo de configuração:

#Exemplo de configuração YAML:

network:
  version: 2
  ethernets:
    eth0:
      dhcp4: no
      addresses:
        - 192.168.1.1/24
      dhcp-identifier: mac

sudo netplan apply #Aplicar as mudanças:
```

▼

ROTA ESTÁTICA | DEFAULT | NETPLAN

Configuração de Rotas Usando o Comando `ip route`

- **Mostrar Rotas**

```
ip route show
```

- **Adicionar Rota**

```
ip route add [destino] via [gateway]
```

Exemplo: Adicionar uma rota para a rede `10.1.1.0/24` usando o gateway `192.168.1.1`:

```
ip route add 10.1.1.0/24 via 192.168.1.1
```

- **Deletar Rota**

```
ip route del [destino]
```

Exemplo: Remover a rota para a rede `10.1.1.0/24`:

```
ip route del 10.1.1.0/24
```

- **Modificar Rota**

```
ip route change [destino] via [gateway] dev [inter
face]
```

Exemplo: Modificar a rota para a rede `10.1.1.0/24` para usar o gateway `192.168.2.1` na interface `eth0`:

```
ip route change 10.1.1.0/24 via 192.168.2.1 dev et
h0
```

```

- **Configurar Gateway Padrão**

```
ip route add default via [endereço do gateway] dev
[interface]
```

Exemplo: Configurar o gateway padrão como `192.168.1.1` na interface `eth0`:

```
ip route add default via 192.168.1.1 dev eth0
```

Configuração de Rotas Usando o Netplan

- **Definir Gateway Padrão**
  Adicione a configuração do gateway padrão em um arquivo YAML de netplan:

```
yamlCopiar código
network:
  version: 2
  ethernets:
    eth0:
      dhcp4: no
      addresses:
        - 192.168.2.10/24
      gateway4: 192.168.2.1
```

- **Definir Outras Rotas**
  Adicione rotas específicas além do gateway padrão:

```
yamlCopiar código
network:
  version: 2
```
```

```
  ethernets:
    eth0:
      dhcp4: no
      addresses:
        - 192.168.2.10/24
      gateway4: 192.168.2.1
      routes:
        - to: 172.16.0.0/24
          via: 192.168.1.1
```

Nesse exemplo:

- `addresses`: Define o endereço IP estático da interface.
- `gateway4`: Define o gateway padrão para IPv4.
- `routes`: Define rotas adicionais, como uma rota para `172.16.0.0/24` via `192.168.1.1`.

Resumo

- `ip route` é um comando para adicionar, remover e modificar rotas diretamente no sistema.
- `netplan` é usado para configurar redes de maneira declarativa em arquivos YAML, o que inclui definir rotas estáticas e gateways padrão.

Para aplicar alterações feitas com o `netplan`, use o comando:

```
sudo netplan apply
```

Configurar NAT para Acesso à Internet

- O NAT (Network Address Translation) permite que máquinas na rede interna usem o IP público do gateway para acessar a Internet. Você pode configurar isso usando o `iptables`.

Comandos para Configurar NAT:

1. **Adicionar Regra de NAT:**

   Configure o NAT para mascarar (masquerade) os pacotes saindo pela interface `eth0`. Isso permite que a rede interna use o IP público do gateway.

   ```
   sudo iptables -t nat -A POSTROUTING -o eth0 -j MAS
   QUERADE
   ```

   - `t nat`: Especifica a tabela NAT.
   - `A POSTROUTING`: Adiciona uma regra na cadeia POSTROUTING.
   - `o eth0`: Aplica a regra à interface de saída `eth0`.
   - `j MASQUERADE`: Usa o masquerade para modificar os pacotes.

2. **Salvar Regras de iptables:**

   Para garantir que as regras de iptables sejam aplicadas após uma reinicialização, salve as regras atuais.

   ```
   sudo iptables-save | sudo tee /etc/iptables/rules.
   v4
   ```

   - `iptables-save`: Exclui as regras atuais.
   - `sudo tee /etc/iptables/rules.v4`: Salva as regras no arquivo `/etc/iptables/rules.v4`.

2. Configurar Rota Padrão nos Gateways Intermediários

Para garantir que o tráfego seja roteado corretamente através dos gateways intermediários, você pode definir a rota padrão nesses gateways.

Comando para Adicionar Rota Padrão:

```
sudo ip route add default via [IP_GATEWAY] dev [INTER
FACE]
```

- `default`: Especifica a rota padrão.
- `via [IP_GATEWAY]`: Define o IP do gateway para rotear o tráfego.
- `dev [INTERFACE]`: Especifica a interface a ser usada para a rota (por exemplo, `eth0`).

Exemplo:

Para configurar a rota padrão com o gateway `192.168.1.1` na interface `eth0`:

```
sudo ip route add default via 192.168.1.1 dev eth0
```

Resumo das Etapas

1. **Configurar NAT:**
   - Adicione uma regra de NAT usando `iptables` para permitir que a rede interna use o IP público para acessar a Internet.
   - Salve as regras para persistência após a reinicialização.
2. **Configurar Rota Padrão:**
   - Adicione uma rota padrão em gateways intermediários para garantir o roteamento correto do tráfego.

▼ *NSLOOKUP | DIG | SSH | CSP*

nslookup

- `nslookup` é uma ferramenta usada para encontrar informações sobre domínios na Internet. Por exemplo, pode transformar um nome de domínio em um endereço IP ou vice-versa.

Comando básico:

```
nslookup [hostname]
```

- `hostname` é o nome do domínio que você quer pesquisar, como `example.com`.

Opções úteis:

- `type=mx`: Encontra os servidores de e-mail associados a um domínio.
- `type=ns`: Mostra os servidores de nomes de um domínio.
- `timeout=10`: Define um tempo limite de 10 segundos para a consulta.

Modo Interativo:

Digite apenas `nslookup` no terminal para entrar no modo interativo e fazer múltiplas consultas.

---

2. dig

- `dig` é uma ferramenta mais poderosa para buscar informações DNS e diagnosticar problemas de configuração de servidores DNS.

Comando básico:

```
dig [name] [type]
```

- `name` é o domínio que você quer pesquisar, como `example.com`.
- `type` é o tipo de informação que você quer obter.

Tipos de registros DNS que você pode pesquisar:

- `A`: Endereço IPv4.
- `MX`: Servidores de e-mail.
- `TXT`: Texto associado ao domínio.
- `NS`: Servidores de nomes.
- `AAAA`: Endereço IPv6.

Opções úteis:

- `+short`: Retorna uma resposta mais curta e direta.

- `+trace`: Mostra cada passo da resolução do nome, útil para diagnosticar problemas.

---

3. ssh

- `ssh` (Secure Shell) é um protocolo para acessar computadores remotamente de forma segura.

Comando básico:

```
ssh [username@IP]
```

- `username` é o nome de usuário no computador remoto.
- `IP` é o endereço IP do computador remoto.

Opções úteis:

- `p [port]`: Conectar usando uma porta específica, se a porta padrão (22) não for usada.
- `i [keyfile]`: Usar uma chave SSH específica para autenticação.
- `o StrictHostKeyChecking=no`: Desativa a verificação de autenticidade do host, mas pode ser arriscado.

---

4. scp

- `scp` (Secure Copy) é usado para copiar arquivos entre computadores de forma segura, utilizando o protocolo SSH.

Comando básico:

```
scp [source] [username@IP]:[destination]
```

- `source` é o caminho do arquivo ou diretório que você quer copiar.
- `username@IP` é o nome de usuário e o endereço IP do computador remoto.
- `destination` é o local onde o arquivo será copiado no computador remoto.

Exemplo:

Para copiar um arquivo chamado `file.txt` para o diretório `/home/user/` em um servidor remoto com IP `192.168.1.1`:

```
scp file.txt username@192.168.1.1:/home/user/
```

▼ ***PING | TRACEROUTE e MTR | DNS***

PING E INTERFACE

Listar os ips e interfaces

```
ip addr ou ip add show
ip -6 addr show
ip -4 addr show
```

Lista apenas as interfaces de rede

```
ip add show dev enp0s3
```

Teste de ping | Exibindo interfaces

```
#Ipv4
ping 192.168.20.10

#Ipv6
ping6 xxxxxxxx:xxxxxxxxx:xxxxxxxxxx:xxxxxxxxx

#Exibindo apenas as interfaces IPV6
ip -6 add show dev enp0s3

#Exibindo apenas as interfaces IPV4
ip -4 add show dev enp0s3
```

Adicionando IP

```
#IPV4
ip addr add 192.168.0.215/24 dev enp0s3

#IPV6
ip -6 addr add fd00:f0ca:f0ca:f0ca:f0ca::d40/64 dev enp
```

TRACEROUTE e MTR

- **Traceroute**: Ela exibe uma lista dos roteadores pelos quais os pacotes passam até alcançar o destino.
- **mtr**: combina as funcionalidades de `traceroute` e `ping` Ele mostra o caminho e também as estatísticas de perda de pacotes e latência em cada hop.
- Shift + L ele vai reiniciar as estatísticas (no caso do mtr)

```
#traceroute
traceroute -n www.google.com

#mtr
mtr exemplo.com
```

DNS

- o cliente DNS é resolvido no `“vi /etc/resolf.conf”` onde voce pode configurar para adicionar um dns de maneira, que, se cair um, o outro resolve (ex: nameserver 8.8.8.8)

```
domain gms.com.br
search gms.com.br
nameserver 8.8.8.8
nameserver 8.8.4.4
```

DIAGNOSTICO DE COMO ESTÁ FUNCIONANDO A RESOLUÇÃO DE NOMES

- Ele mostra mais de um endereço, o que indica a redundância nos links

```
host www.google.com

host -t A #para ver o endereço ipv4
host -t AAAA www.google.com #para ver o endereço ipv6
host -t mx www.google.com #para ver o alias para
```

▼ Rotas-Calculo-Máscara-DNS-Interface-Serviço de Rede

Rotas-Calculo-Máscara

LISTAR AS ROTAS

```
ip route ls
```

PARA CALCULAR O CALCULO DE MASCARA DE REDE

- Voce pode usar o seguinte comando para fazer calculo de mascara de rede:

```
apt-get install subnetcal


#exemplo com IPV4
subnetcal 192.168.0.211/24

#exemplo com IPV6
subectcalc fd0:f0ca:f0ca::1/64
```

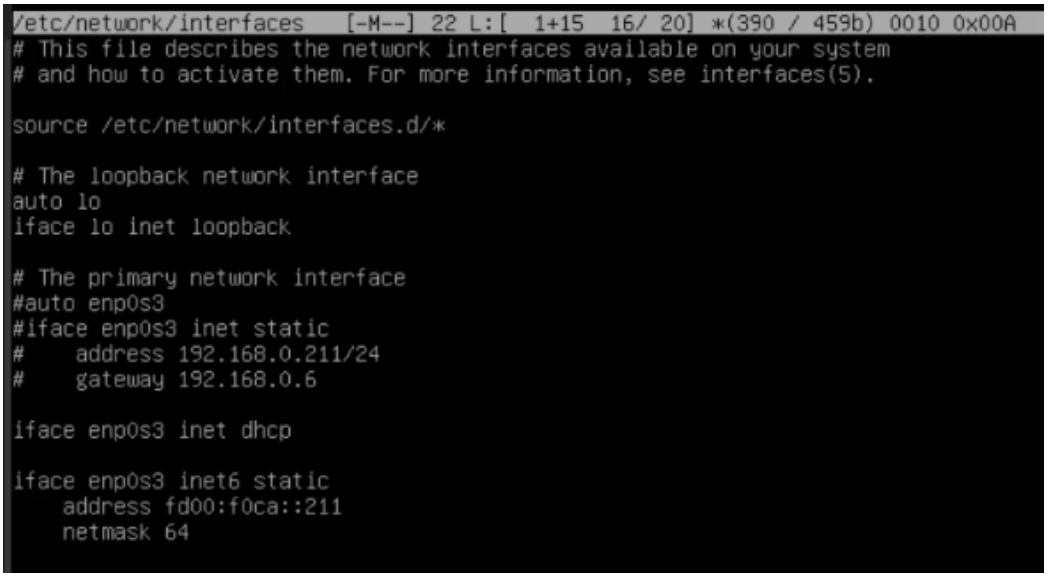
Criando um endereço apontando para o gateway

```
ip route add coloqueoip/mascara via ip
```

COLOCANDO ENDEREÇO ESTÁTICO NO PC

- encontra-se no `/etc/network/interfaces`
- Isso só funciona se o servidor estiver usando DHCP DNS.

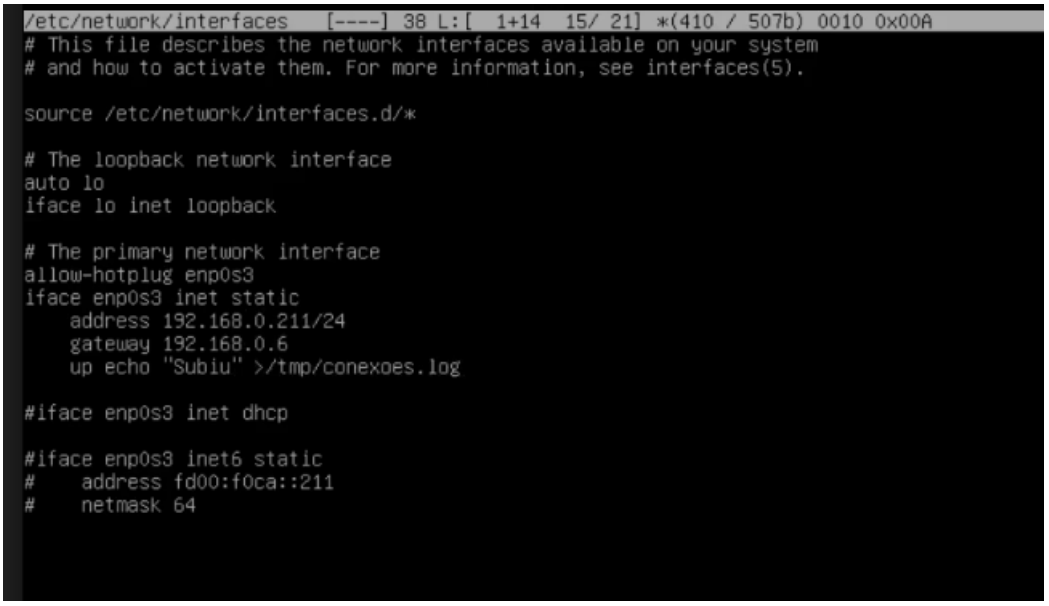
```
#adicione, em algum lugar, o comando "iface enp0s3 inet dh

iface enp0s3 inet dhcp
```



Código 1: Depois que fizer isso, vá no terminal e coloque "`ifup enp0s3`"

SE A INTERFACE CAIR OU SUBIR, GERAR ALERTA

- encontra-se no `/etc/network/interfaces`



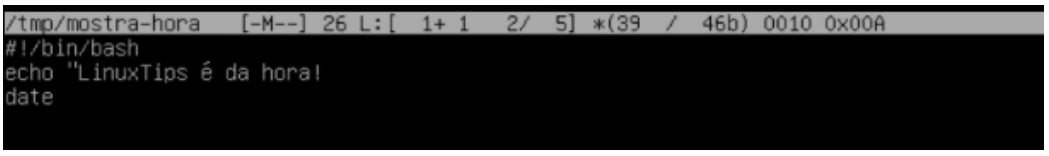
Código 1: Se a interface subir, ele vai gerar um log em "/tmp/conexões.temp. se voce colocar o ">>" ele faz um append, o ">" ele só adiciona uma vez

Código 2: Se a interface cair, ele vai gerar um log em "/tmp/conexões.temp

SERVIÇO DE REDE

- São serviços que prever uma comunicação em rede, que tem porta
- Existem dois tipos:
  - Deamon: grande quantidade de conexão simultaneamente
  - xinetd (ou Super-Server Daemon): Binário só é chamado no momento que é executado

```
#Verifica qual serviço de rede estou executando no momento
ss -int
```



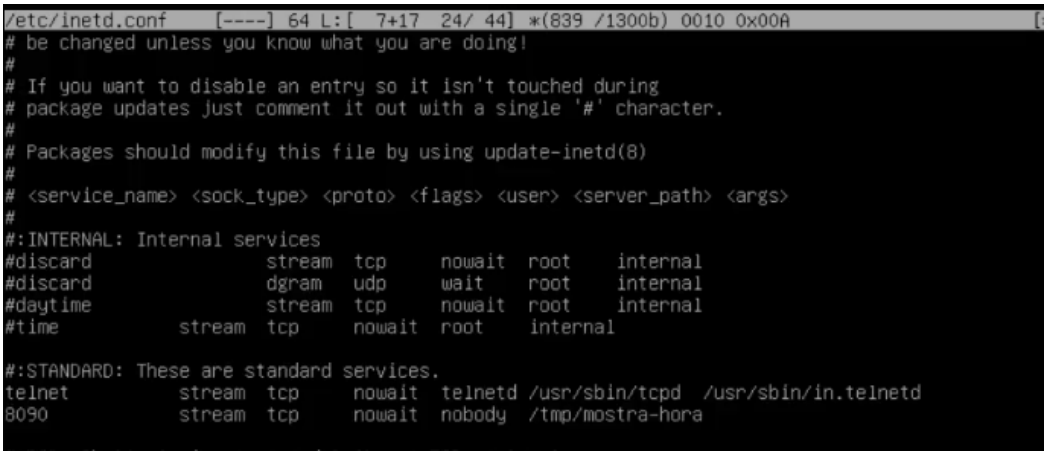
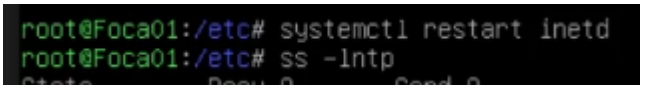


Figura 2: Em /etc/inetd.conf, coloque: “8090 stream tcp nowait nobody /tmp/mostra-hora



NETCAT

- ampla gama de funcionalidades para transferência de arquivos, diagnóstico de rede, depuração e muito mais

```
apt-get install netcat
netcat -l -p 4545

netcat localhost 4545
```

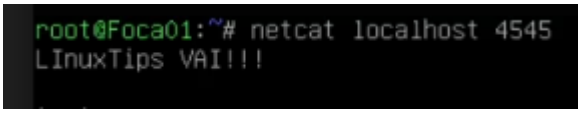


Figura 1: Comunicação entre cliente e servidor