

Capítulo 17 - Configurações de redes

17.1. Objetivos:

Entender e configurar uma rede com Linux.

17.2. Introdução

A configuração de rede em um sistema GNU/Linux é muito importante, pois esses sistemas são, intrinsecamente, sistemas de rede. Ou seja, mesmo que não haja nenhum tipo de interface de rede ou modem ou qualquer coisa do gênero, ainda sim uma máquina GNU/Linux será um sistema de rede.

Isso é possível, pois em todo GNU/Linux há uma interface lógica, chamada loopback (lo) cujo endereço IP será 127.0.0.1 e que sempre deve estar apropriadamente configurada.

Estando essa interface configurada, todo o tipo de serviço pode ser ativado na máquina, desde um simples servidor de ssh até um servidor de DNS e um servidor de páginas (Apache) por exemplo.

Resumindo, pode ser feito qualquer coisa que queira e não necessariamente precisa ter contato com o mundo exterior (internet).



Importante: Para treinar esse conteúdo, é interessante que use no mínimo dois computadores ou um computador só e mais uma máquina virtual.

Antes de mais nada, precisamos nos certificar que a placa esteja no ar. Para ver se a placa está no ar:



```
#lsmod  
Depois :  
#ifconfig -a
```

Onde o -a nessa caso, é de all , ou seja liste todas as placas ativas na máquina. Então, se der lsmod e o módulo da placa de rede estiver lá basta agora configurar a placa.

Verifique se há link nas interfaces:



```
# mii-tool  
eth0: negotiated 100baseTx-FD flow-control, link ok
```

Neste caso o link está OK.

Vamos agora para os arquivos de configuração de rede:



Debian



/etc/hostname
/etc/hosts
/etc/network/interfaces



Red Hat



/etc/hostname
/etc/hosts
/etc/sysconfig/network-scripts/ifcfg-ethX

17.3. O arquivo hostname

O primeiro arquivo que será mostrado é o */etc/hostname*. Sua função é apenas guardar o nome da máquina. E esse arquivo tem apenas uma única linha com o nome da minha máquina. Então para editá-lo, preciso apenas fazer:



```
#echo "trinity" > /etc/hostname
```

Ou seja, o nome da máquina está sendo jogado através do *echo* para dentro do arquivo. Usando o *>* ele irá apagar o conteúdo que estiver nesse arquivo. É essa a ideia, pois podemos apenas ter uma linha. Você pode alterar o nome da máquina dinamicamente através do comando *hostname*, exemplo:



```
# hostname leo
```

Verificando qual o hostname:



```
# hostname  
leo
```

Agora, próximo arquivo.

17.4. O arquivo /etc/hosts

Esse arquivo tem como objetivo resolver nome dentro de sua rede, ou seja, imagine que tem 5 máquinas em rede é preciso que a máquina resolva o nome de todas. É como se fosse um DNS primitivo. Colocamos a relação das máquinas no hosts:



```
IP----- - Nome  
192.168.0.1 - router  
192.168.0.2 - neo  
192.168.0.3 - trinity  
192.168.0.4 - morpheus  
192.168.0.5 - sentinelas
```

Então, seguindo a estrutura do arquivo:



```
# cat /etc/hosts  
127.0.0.1 localhost.localdomain localhost  
192.168.0.4 trinity.4linux.com.br trinity
```

Ou seja, IP Nome_Completo(FQDN) Hostname. E 4linux.com.br é o domínio da rede interna. Fully Qualified Domain Name. É o nome completo da máquina, ou seja, hostname + domínio.

Lembra-se que para que todas as máquinas resolvam nomes na rede, é necessário colocar nome em todas elas. Por isso não usa muito esse arquivo, pois dá muito trabalho. Uma opção ao hosts é configurar um servidor de DNS.

Assim não precisa colocar em todas as máquinas da rede. Com isso, já é possível por exemplo pingar:



```
#ping trinity.4linux.com.br
```

Ou ainda:

```
#ping trinity
```

Na prática coloque nesse arquivo apenas a máquina para que ela possa saber quem ela é, ou seja, mesmo que tenha um DNS interno na rede, coloque apenas a linha que diz respeito a máquina para que ela não saia e pergunte para o DNS. (Quem é ela!).

17.5. O arquivo interfaces (Debian)

Partindo para o próximo arquivo. Veja agora sobre o arquivo responsável por armazenar as configurações de nossa placa.



Debian:



```
/etc/network/interfaces
```



RedHat:



```
/etc/sysconfig/network-scripts/ifcfg-ethX
```

Slackware



```
/etc/rc.d/rc.inet1.conf
```

É nesse arquivo que setamos nosso IP, MASK, GATEWAY. Lembre-se que na hora que editar esse arquivo um caractere errado vai fazer com que a placa não levante.

Diferente do Red Hat (e derivados), o Debian guarda em apenas um arquivo o IP (e demais configurações de rede como máscara, gateway) de todas as placas ativas no sistema. Já o Red Hat, cria um arquivo para cada placa. Reparem, que os arquivos do Linux costumam ter uma estrutura específica e não pode fugir dela.

Estrutura do Arquivo (Debian):



```
# vi /etc/network/interfaces
auto eth0
iface eth0 inet static
address 192.168.0.87
netmask 255.255.255.0
network 192.168.0.0
broadcast 192.168.0.255
gateway 192.168.0.1
```

Onde:

- auto eth0 - Significa que quero que essa placa seja levantada na hora que o serviço de rede iniciar. Caso não tenha essa linha, mesmo que o serviço networking iniciar, não irá carregar as configurações dessa placa em específico.

- iface eth0 inet static - Aqui apenas estabelece qual é a interface (eth0, eth1, etc) e que a mesma terá IP fixo. Caso a máquina for cliente de um servidor DHCP na rede, basta deletar todas as linhas de configuração da placa e deixar:



```
iface eth0 inet dhcp
```



address - o seu endereço ip

netmask - máscara de rede

network - endereço da rede

broadcast - ip que toda a rede responde

gateway - ip da máquina que fará a saída dos pacotes para internet

Após configurar o arquivo responsável por guardas configurações como ip, máscara de rede e etc., já é possível salvar o mesmo e levantar o serviço de rede. Para levantar o serviço de rede:



Debian:



```
#!/etc/init.d/networking stop
```

```
#!/etc/init.d/networking start
```



Red Hat:



```
# service network stop
```

```
# service network start
```

Verificando configurações feitas:



```
# ifconfig
```

Depois é possível pingar para broadcast por exemplo, para saber se esta vivo na rede.



**broadcast* - endereço de difusão, ou seja, um endereço que faz com que todas máquinas que estiverem ativas irão responder.

Detalhe, dependendo da versão do ping será necessário especificar o parâmetro -b :



```
# ping -b 192.168.0.255
```

Até agora foi mostrado como colocar a máquina na rede. Agora, se for uma máquina que estará na rede saindo para internet por meio de um roteador, é preciso configurar mais uma linha no arquivo da placa pensando em acesso a rede externa - internet). No arquivo /etc/network/interfaces foi especificado uma linha chamada:



```
gateway 192.168.0.1:
```

Essa linha só é necessária nesse arquivo, caso a rede tenha um roteador (router). Então, para acessar a NET por exemplo, terá primeiro que pedir para o Router para que ele mande a página de volta. Exemplo:



```
Leo (Rede Interna) ----- Router ----->> Internet  
Internet ----- Router ----->> Leo (Rede Interna)
```

Caso a rede não tenha essa estrutura, não haverá necessidade de colocar a linha gateway. Quando é especificado (eu máquina cliente) quem será o gateway, cria-se uma rota default. E rota default nada mais é que indicar o caminho que o pacote deverá percorrer na rede. Então, para visualizar isso, existe o comando:



```
# route -n
```

Pois o ifconfig apenas mostra o IP da máquina ou atribui ip; agora o comando route -n mostra a rota que os pacotes irão fazer.

-n para mostrar numericamente (por IP).

Ou seja, o caminho que eles vão percorrer para sair da máquina para a Net e da Net para máquina.



Para criar a rota default na mão, ou seja, não especificar o gateway no arquivo, execute este comando (atenção isso é cobrado na LPI):



```
#route add default gw 192.168.0.1
```

Que fará a mesma coisa que deixar a linha no arquivo. a diferença é que no arquivo é definitivo. Onde add seria para adicionar uma rota, e gw (gateway) para especificar o gateway. Agora que foi definido o gateway, e visualize através do comando:



```
# route -n
```

Dicas para o comando ifconfig:

Para derrubar a interface de rede:



```
# ifconfig eth0 down
```

Certifique-se que você a derrubou:



```
# ifconfig
```

Para levantar a interface:



```
# ifconfig eth0 up
```

Agora precisamos atribuir novamente um endereço IP:



```
# ifconfig eth0 192.168.0.1 netmask 255.255.255.0
```

Se caso precisar atribuir mais um IP para mesma placa de rede, poderá utilizar um recurso que o Linux oferece. Supondo que você só tenha uma placa de rede na máquina e você quer que a máquina faça parte de duas redes diferentes. A lógica seria ter duas placas de redes, mas na prática não é necessário:



```
# ifconfig eth0:1 192.168.1.1
```

Se eu quisesse colocar mais um ip de uma outra rede:



```
# ifconfig eth0:2 192.168.2.1
```

17.6. O arquivo /etc/resolv.conf

Agora, tenho apenas que editar um arquivo chamado:



```
/etc/resolv.conf
```

Nesse arquivo apontamos os servidores de DNS usado pela nossa máquina. No arquivo resolv.conf tenho apenas que colocar o IP do meu DNS:



```
#echo "domain 4linux.com.br" > /etc/resolv.conf  
#echo "nameserver 200.170.203.70" >> /etc/resolv.conf
```

Aqui foi jogado apenas 2 linhas no arquivo, mas pode editar com o VI também.



Onde:

domain - Nome do meu Domínio (Não é obrigatório)

nameserver - IP do DNS (obrigatório)

E caso tenha mais de um endereço de DNS pode apenas ir acrescentado as linhas nameserver, uma abaixo da outra. Um problema que ocorre quando não tem nada no arquivo /etc/resolv.conf é que a internet parece que está OK (pois você consegue pingar sites externos através do endereço ip), mas você não consegue acessar sites utilizando nomes como por exemplo <http://www.4linux.com.br>

Mais alguns comandos interessantes de rede:

Para descobrir qual é o FQDN da sua máquina e não apenas o nome do host:



```
# hostname -f
```

Para você descobrir o endereço MAC de outras máquinas na rede, faça:



```
# arp -n
```

-n para que o resultado seja numérico



Address	HWtype	HWaddress	Flags Mask	Iface
192.168.1.1	ether	00:19:5b:56:de:cd	C	eth0

Ou utilize o comando ifconfig para descobrir o seu endereço MAC:



```
eth0    Link encap:Ethernet Endereço de HW 00:1d:7d:fe:12:fb
        inet end.: 192.168.1.100 Bcast:192.168.1.255 Masc:255.255.255.0
        endereço inet6: fe80::21d:7dff:fefe:12fb/64 Escopo:Link
        UP BROADCASTRUNNING MULTICAST MTU:1500 Métrica:1
        RX packets:221661 errors:0 dropped:0 overruns:0 frame:0
        TX packets:137664 errors:0 dropped:0 overruns:0 carrier:0
        colisões:0 txqueuelen:1000
        RX bytes:302731190 (288.7 MiB) TX bytes:14765158 (14.0 MiB)
        IRQ:220 Endereço de E/S:0x2000
```