# Linux para Infraestrutura

Prof. André Nasserala

nasserala@gmail.com

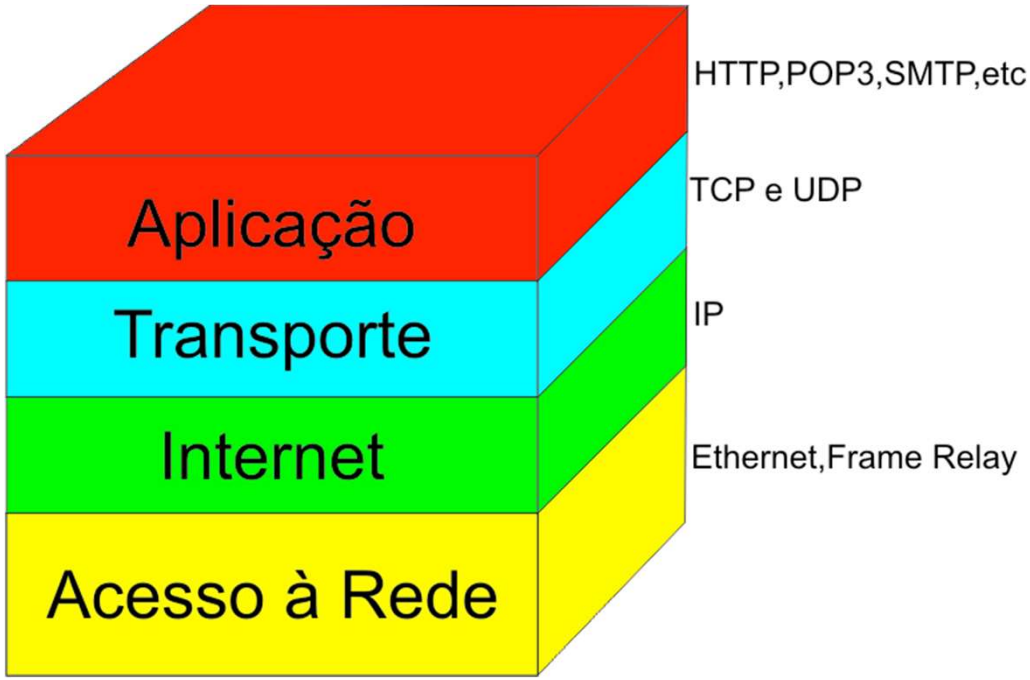
# TCP/IP

- **Modelos de Refêrencia**
- **Modelo OSI**
- Este modelo é dividido em camadas hierárquicas, ou seja, cada camada usa as funções da própria camada ou da camada anterior, para esconder a complexidade e transparecer as operações para o usuário, seja ele um programa ou uma outra camada.
- PDU's:
- Camada física: "Bit", Camada de enlace: "Quadro", Camada de rede: "Pacote", Camada de transporte: "Segmento" e nas camadas superiores "Dados";

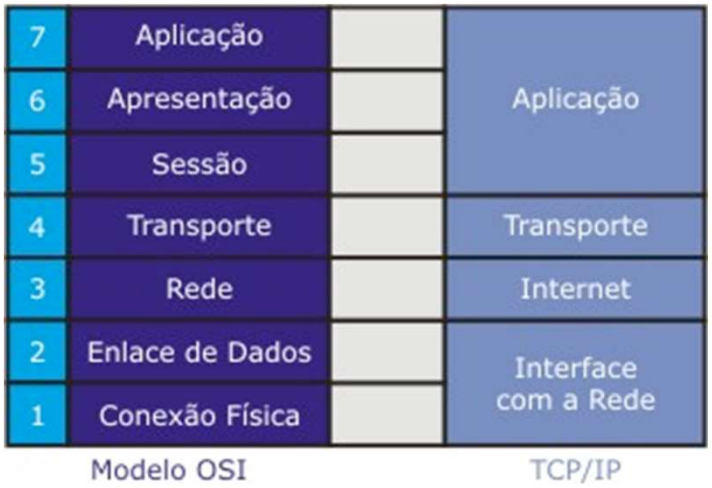

Notem que a ordem numérica das camadas e decrescente.

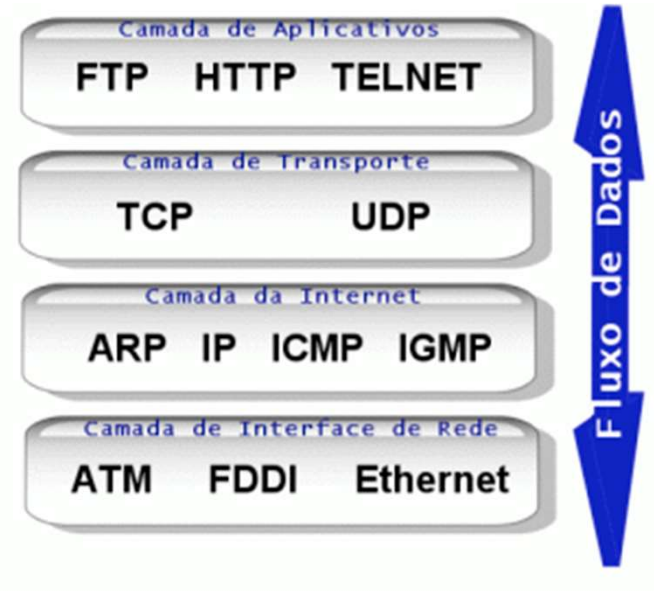
# TCP/IP

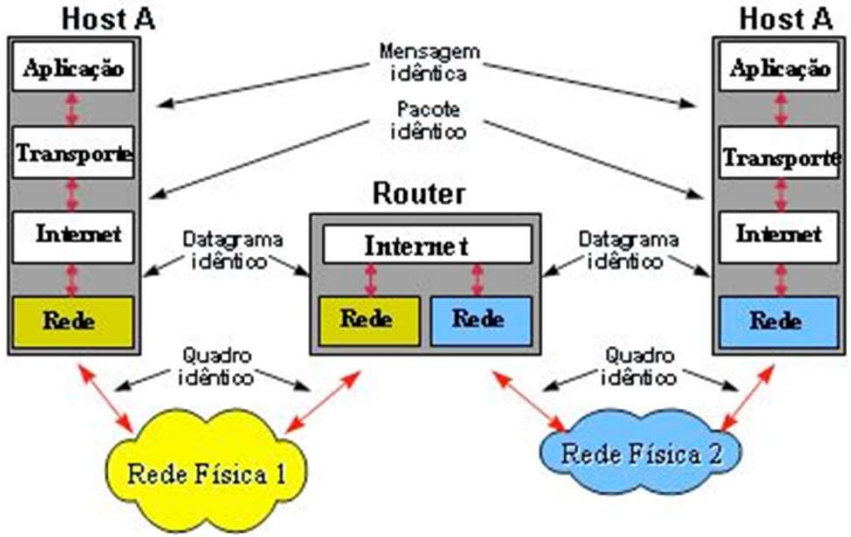
- **Modelos de Refêrencia**
- **Modelo TCP/IP**
- O TCP/IP é um conjunto de protocolos de comunicação entre computadores em rede (também chamado de pilha de protocolos TCP/IP). Seu nome vem de dois protocolos: o TCP (Transmission Control Protocol - Protocolo de Controle de Transmissão) e o IP (Internet Protocol - Protocolo de Interconexão)



# TCP/IP


7
Aplicação
6
Apresentação
5
Sessão
4
Transporte
3
Rede
2
Enlace de Dados
1
Conexão Física
Aplicação
Transporte
Internet
Interface com a Rede
Modelo OSI
TCP/IP


Camada de Aplicativos
FTP HTTP TELNET
Camada de Transporte
TCP UDP
Camada da Internet
ARP IP ICMP IGMP
Camada de Interface de Rede
ATM FDDI Ethernet
Fluxo de Dados


Host A
Aplicação
Transporte
Internet
Rede
Mensagem idêntica
Pacote idêntico
Router
Internet
Rede
Rede
Datagrama idêntico
Datagrama idêntico
Host A
Aplicação
Transporte
Internet
Rede
Quadro idêntico
Quadro idêntico
Rede Física 1
Rede Física 2

# TCP/IP

- **Endereço IP**
- O endereço IP (Internet Protocol), de forma genérica, é um endereço que indica o local de um determinado equipamento (normalmente computadores) em uma rede privada ou pública.
- O endereço IP, na versão 4 (IPv4), é um número de 32 bits escrito com quatro octetos representados no formato decimal, Exemplo:
- 200.103.16.13
- A primeira parte do endereço identifica uma rede específica na Inter-rede, a segunda parte identifica um host dentro dessa rede.
- Devemos notar que um endereço IP não identifica uma máquina individual, mas uma conexão à Inter-rede.

# TCP/IP

- **Endereço IP**
- Por convenção, um endereço de rede tem o campo identificador de host com todos os bits iguais a 0 (zero).
- Podemos também nos referir a todos os hosts de uma rede através de um endereço por difusão(Broadcast), quando, por convenção, o campo identificador de host deve ter todos os bits iguais a 1 (um).
- Um endereço com todos os 32 bits iguais a 1 é considerado um endereço de difusão para a rede do host de origem.
- O endereço 127.0.0.1 é reservado para teste (loopback) e comunicação entre processos da mesma máquina.

# TCP/IP

- **Endereço IP**
- **Endereço de rede** é o primeiro endereço IP utilizado em uma rede, ele serve para identificar a rede para definição de roteamento.
- Exemplo: 192.168.0.0
- **Endereço de broadcast** é o ultimo endereço utilizado em uma rede, ele serve para transmissão em difusão.
- Exemplo: 192.168.0.255
- **Mascara de rede** é utilizado para divisão do endereço IP em hosts e redes.
- Exemplo: 255.255.255.0 (Parte da Rede: 255.255.255 e Parte do Host: 0)

# Cálculos de Rede

- Como uma estação(host) sabe se deve encaminhar seus pacotes ao seu gateway ou se pode transmitir diretamente(Roteamento Direto)?
- A resposta está no endereço de rede, pois se origem e destino possuírem o mesmo endereço de rede essa comunicação acontecerá diretamente, caso não possuam haverá roteamento via gateway.
- Exemplo:
- As estações 192.168.0.10 e 192.168.0.250 com mascaras 255.255.255.0, para se comunicarem, usarão gateway?
- R: Endereço de rede? **(192.168.0.0)** mas como sabemos?

# Cálculos de Rede

- O endereço ip 192.168.0.10 em binário é o seguinte:
- Comando: echo 'obase=2; ibase=10; 192' | bc
- **11000000**
- Comando: echo 'obase=2; ibase=10; 168' | bc
- **10101000**
- Comando: echo 'obase=2; ibase=10; 0' | bc
- **00000000**
- Comando: echo 'obase=2; ibase=10; 10' | bc
- **00001010**

# Cálculos de Rede

| A | B | AND | OR | XOR |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |

Figura 2: Tabela Verdade

- O Resultado:
- **11000000.10101000.0000000.00001010**
- A mascara 255.255.255.0:
- **11111111.11111111.11111111.00000000**
- Operação lógica(e) entre o IP e a Mascara resulta o endereço de rede:
- **11000000.10101000.0000000.00001010**
- **11111111.11111111.1111111.00000000**
- **11000000.10101000.00000000.0000000 (192.168.0.0) End. Rede**

# Cálculos de Rede

| A | B | AND | OR | XOR |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 1 | 1 | 1 | 0 |

Figura 2: Tabela Verdade

- Assim:
- **192.168.0.200**
- **11000000.10101000.0000000.11001000**
- Operação lógica(e) entre o IP(192.168.0.200) e a Mascara resulta o endereço de rede:
- **11000000.10101000.0000000. 11001000**
- **11111111.11111111.1111111.00000000**
- **11000000.10101000.00000000.0000000 (192.168.0.0) End. Rede**
- **Como ambos tem o mesmo endereço de rede dizemos que não haverá comunicação via gateway, pois eles estão na mesma rede.**

# Cálculos de Rede

- Podemos tirar mais conclusões sobre o endereço de rede de um host e sua mascara, por exemplo, a operação lógica “XAND” entre o endereço de rede e sua mascara, na parte que tem “0”, tem como resultado o endereço de difusão(Broadcast).
- Assim(192.168.0.200 e 255.255.255.0):
- **11000000.10101000.0000000.00000000**
- **11111111.11111111.1111111.00000000**
- **11000000.10101000.00000000.11111111**
- **Em decimal: 192.168.0.255(Endereço de Broadcast)**

**XAND ou XNOR**

| A | B | F |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

# Cálculos de Rede

- Um Exemplo: O endereço IP 192.168.0.214 com mascara 255.255.255.224 tem que endereço de rede e broadcast?
- 11000000.10101000.0000000. 11010110
- 11111111.11111111.11111111.11100000

**AND**

- **11000000.10101000.00000000.11000000(192.168.0.192)**
- **11111111.11111111.11111111.11100000**
- **11000000.10101000.00000000.11000000**

**XAND**

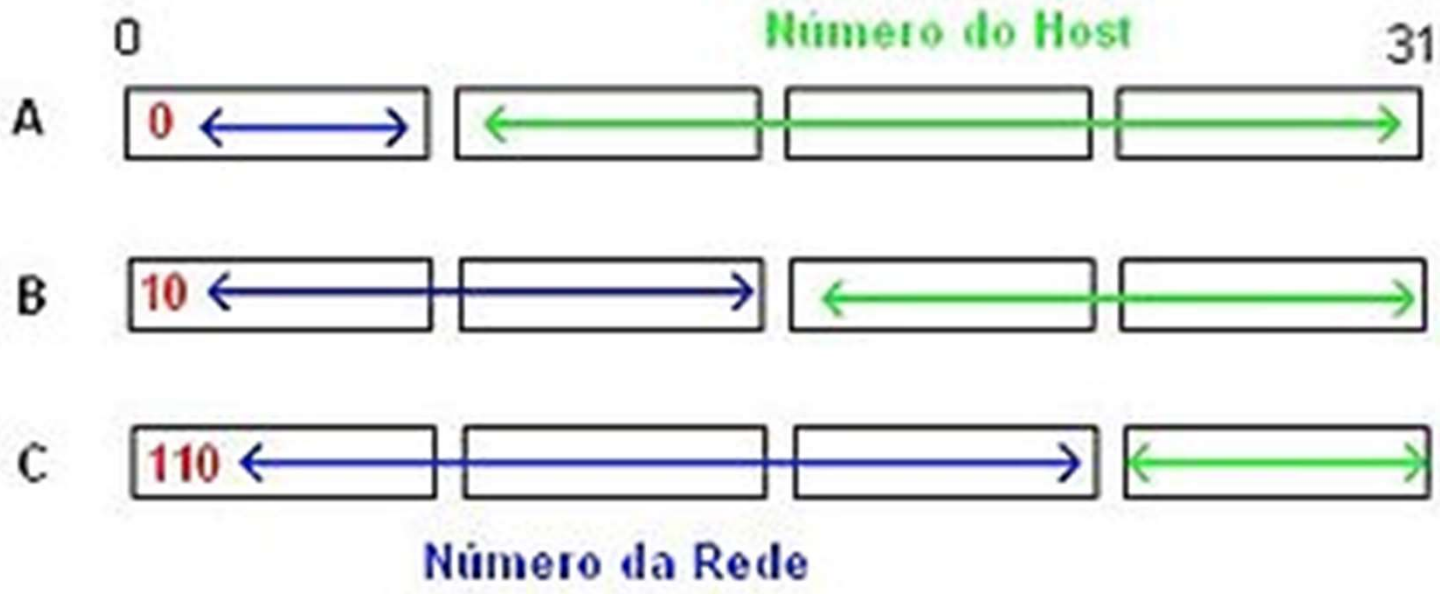
- **11000000110101000.00000000.11011111**
- **Endereço de Broadcast: 192.168.0.223**

# Classes de endereços

- Originalmente, o espaço do endereço IP foi dividido em poucas estruturas de tamanho fixo chamados de "classes de endereço".
- As três principais são a classe A, classe B e classe C. Examinando os primeiros bits de um endereço, o software do IP consegue determinar rapidamente qual a classe, e logo, a estrutura do endereço.
- Classe A: Primeiro bit é 0 (zero)
- Classe B: Primeiros dois bits são 10 (um, zero)
- Classe C: Primeiros três bits são 110 (um, um, zero)

# Classes de endereços

- Classe D: (endereço multicast): Primeiros quatro bits são: 1110 (um, um, um, zero)
- Classe E: (endereço especial reservado): Primeiros cinco bits são 11110 (um, um, um, um, zero)



# Classes de endereços

| Classe | Gama de Endereços | Nº de Endereços por Rede |
|---|---|---|
| A | 1.0.0.0 até 126.0.0.0 | 16 777 216 |
| B | 128.0.0.0 até 191.255.0.0 | 65 536 |
| C | 192.0.0.0 até 223.255.255.0 | 256 |
| D | 224.0.0.0 até 239.255.255.255 | Multicast |
| E | 240.0.0.0 até 255.255.255.254 | *Uso futuro; atualmente reservada a testes pela IETF* |

| Classe | Faixa de endereços de IP | Notação CIDR | Número de Redes | Número de IPs | IPs por rede |
|---|---|---|---|---|---|
| Classe A | 10.0.0.0 – 10.255.255.255 | 10.0.0.0/8 | 128 | 16.777.216 | 16.777.214 |
| Classe B | 172.16.0.0 – 172.31.255.255 | 172.16.0.0/16 | 16.384 | 1.048.576 | 65 534 |
| Classe C | 192.168.0.0 – 192.168.255.255 | 192.168.0.0/24 | 2.097.150 | 65.535 | 254 |

# Classes de endereços

- IP’s públicos são conhecidos na Internet e IP’s privados não são, assim, para se comunicar diretamente na Internet é necessário ter pelo menos um IP público na sua rede.
- **Sub-Rede**
- Uma sub-rede é uma divisão de uma rede de computadores. A divisão de uma rede grande em redes menores resulta num tráfego de rede reduzido, administração simplificada e melhor performance de rede.
- Para criar sub-redes, qualquer máquina tem que ter uma máscara de sub-rede que define que parte do seu endereço IP será usado como identificador da sub-rede e como identificador do host.

# Classes de endereços

- **Exemplo de uma sub-rede**
- Tomemos como exemplo um endereço de classe C e dois bits movidos para a direita para criar uma sub-rede:
- máscara: **11111111.11111111.11111111.11**000000
- Porque acrescentamos dois bits a 1 (um), podemos criar $2^2$=4 sub-redes. Sobram 6 zeros, logo esta sub-rede pode endereçar $2^6$ = 64 endereços por sub-rede, como temos que subtrair 2 endereços (o endereço de rede e de broadcast), temos um total de 62 endereços de hosts (64 - 2 = 62).
- A máscara a aplicar é **255.255.255.192**.

# Classes de endereços

- **Exemplo 1:**
- Um administrador de rede pretende pretende dividir sua rede classe B (172.16.0.0/16) em sub-redes que contenha 1022 hosts para otimizar seu tráfego.
- Ele pensa em fazer sub-redes conforme abaixo:
- 172.16.0.0-172.16.3.255 – Matriz Rio Branco
- 172.16.4.0-172.16.7.255 – Filial Cruzeiro do Sul
- 172.16.8.0-172.16.11.255 – Filial Sena Madureira
- Como devem ser os endereços de rede, mascara dessas rede? Prove que é possível ter 1022 hosts nelas.
- Quantas sub-redes classe B podem ser criadas?

# Classes de endereços

- Mascara classe B(255.255.0.0)
- 11111111.11111111.00000000.00000000
- 11111111.11111111.11111100.00000000 ($2^{10}$ = 1024 IPs)
- Assim a mascara de sub-rede será: 255.255.252.0
- O endereço de rede da matriz: **172.16.0.0** (Rio Branco)
- Da filial **172.16.4.0** (Cruzeiro do Sul)
- Da filial **172.16.8.0** (Sena Madureira)
- Hosts: $2^{10}$ -2 = **1022 hosts em cada sub-rede;**
- Sub-redes: $2^{6}$ = **64 sub-redes podem ser criadas**.

# Classes de endereços

- **Exemplo 2:**
- Uma rede classe C (192.168.10.0/24) deve ser divida em 8 sub-rede. Como deve ser a mascara dessas sub-redes e quantos hosts cada uma suportará?
- C: 11111111.11111111.11111111.00000000 (255.255.255.0)
- C: 11111111.11111111.11111111.**11100000** (255.255.255.224)
- Hosts: $2^5 - 2 = 30$ Hosts.
- 1ª Sub-Rede: 192.168.10.0/27
- 2ª Sub-Rede: 192.168.10.32/27
- 3ª Sub-Rede: 192.168.10.64/27
- 4ª Sub-Rede: 192.168.10.96/27
  - 5ª Sub-Rede: 192.168.10.128/27
  - 6ª Sub-Rede: 192.168.10.160/27
  - 7ª Sub-Rede: 192.168.10.192/27
  - 8ª Sub-Rede: 192.168.10.224/27

# Configuração de Rede - LINUX

- 3 Formas:
- **Ferramentas do pacote (net-tools):**
  - ifconfig;
  - route.

- **Utilizando comandos ip**
  - Iproute2.

- **Através dos arquivos de configuração:**
  - /etc/sysconfig/network-scripts/ifcfg-INT

# Comando ifconfig

- O comando ifconfig (presente no pacote net-tools) é utilizado para atribuir um endereço a uma interface de rede ou configurar parâmetros de interface de rede.
- **Alterando/Configurando IP/MAC:**
- O Mac Address da placa de rede teoricamente não pode ser alterado fisicamente, mas pode ser alterado virtualmente. Nos sistemas like-unix/Linux é utilizado o comando ifconfig, segue um exemplo.
- É necessário estar com a placa de rede desativada:
- **# ifconfig eth0 down (desativa)**
- **# ifconfig eth0 up (ativa)**

# Comando ifconfig

- Agora altere o Mac Address:
- **# ifconfig eth0 hw ether 00:D0:D0:67:2C:05**
- Agora ative a placa de rede e configure o endereço IP/Mask:
- **# ifconfig eth0 192.168.0.1 netmask 255.255.255.0 up**
- OBS: o parâmetro up é opcional se a placa já estiver ativa.
- Adicionando o segundo endereço (ip virtual):
- **# ifconfig eth0:1 10.0.0.5 netmask 255.255.255.0 up**

# Arquivo /etc/resolv.conf

- O /etc/resolv.conf é o arquivo de configuração principal do código do resolvedor de nomes. Seu formato é um arquivo texto simples com um parâmetro por linha e o endereço de servidores DNS externos são especificados nele. Existem três palavras chaves normalmente usadas que são:
- Domain: Especifica o nome do domínio local.
- Search: Especifica uma lista de nomes de domínio alternativos ao procurar por um computador, separados por espaços. A linha search pode conter no máximo 6 domínios ou 256 caracteres.
- **Nameserver**: Especifica o endereço IP de um servidor de nomes de domínio para resolução de nomes. Pode ser usado várias vezes.

# Arquivo /etc/resolv.conf

- Como exemplo, o /etc/resolv.conf se parece com isto:
- **domain nasserala.org**
- **search dns.nasserala.org**
- **nameserver 192.168.10.1**
- **nameserver 192.168.12.1**
- Este exemplo especifica que o nome de domínio a adicionar ao nome não qualificado (i.e. hostnames sem o domínio) é nasserala.org e que se o computador não for encontrado naquele domínio então a procura segue para o domínio dns.nasserala.org diretamente. Duas linhas de nomes de servidores foram especificadas, cada uma pode ser chamada pelo código resolvedor de nomes para resolver o nome.

# Comando Route

- **# route add default gw 192.168.10.10** (configura um gateway)
- **# route -n** (lista a tabela de roteamento)
- Podemos utilizar o comando 'route' para adicionar rotas estáticas. Por exemplo, imaginemos que a rede da filial seja 192.168.2.0 com máscara 255.255.255.0 e o roteador que da acesso a essa rede seja o ip 192.168.0.200:
- **# route add -net 192.168.2.0 mask 255.255.255.0 gw 192.168.0.200**
- Para remover uma rota:
- **# route del -net 192.168.2.0 mask 255.255.255.0 gw 192.168.0.200**
- **# route del default** (remove a rota default)

# Configuração Rede Linux

- **<u>Exemplo prático</u>**
- Vamos supor que eu tenha uma rede com 3 computadores, 1 sendo o gateway do meu sistema, que possui, claro, duas placas de rede.
- IPs dos 3 computadores: 10.0.0.1 , 10.0.0.2, 10.0.0.3
- Máscara: 255.255.255.0
- DNS: 8.8.8.8
- Gateway: 10.0.0.1
- **Pré requisito: desativar a trava de pacotes**:
- echo “1” > /proc/sys/net/ipv4/ip_forward

# Configuração Rede Linux

- **1° PC**
- No terminal como root:
- # ifconfig eth0 10.0.0.3 netmask 255.255.0.0 up
- # route add default gw 10.0.0.1

- **2° PC**
- No terminal como root:
- # ifconfig eth0 10.0.0.2 netmask 255.255.0.0 up
- # route add default gw 10.0.0.1

# Configuração Rede Linux

- **3° PC - Gateway com duas placas de rede**
- No terminal como root:

- # ifconfig eth0 10.0.0.1 netmask 255.255.0.0 up
- # ifconfig eth1 200.17.11.1 netmask 255.255.255.0 up

- # route add default gw 10.0.0.1 eth0
- # route add default gw 200.17.11.2 eth1

# Configuração Rede Linux

- **Configurações do DNS**
- Por último, para configurar o DNS, edite o arquivo /etc/resolv.conf de cada máquina e preencha da seguinte maneira:
- **nameserver 8.8.8.8**
- Repare que você pode configurar infinitos DNS, tantos quantos queira.
- Para adicionar mais um por exemplo, só adicionamos 1 linha:
- **nameserver 8.8.8.8**
- **nameserver 8.8.4.4**

# Comando ip

- Os comandos ip são usados para mostrar e configurar parâmetros de rede para as interfaces em uma máquina.
- Os programas do pacote iproute2 são o sistema de configuração de rede atual do Linux.
- O iproute2 consiste em uma série de utilitários, dos quais o ip é o principal, mas além do ip, outros utilitários que compõem o pacote iproute2 são:
  - Ss
  - bridge
  - rtmon
  - nstat
  - tc
  - devlink

# Comando ip

- A tabela a seguir mostra os principais objetos disponíveis para uso com os comandos ip, suas abreviações e função:

| Objeto | Abreviação | Função |
|---|---|---|
| address | a, addr | Endereço em um dispositivo (IPv4 ou IPv6) |
| addrlabel | addrl | Configuração de rótulo para seleção de endereço |
| link | l | Dispositivo de rede |
| maddress | m, maddr | Endereço multicast |
| mroute | mr | Entrada de cache de roteamento multicast |
| neighbour | n, neigh | Entrada de cache ARP ou NDISC |
| rule | ru | Regra no banco de políticas de roteamento |
| tunnel | t | Túnel sobre IP |

# Comando ip

- No dia-a-dia de trabalho do administrador de redes Linux, é mais comum o uso dos objetos address e link, por se tratarem de objetos usados na configuração direta de endereçamento e interfaces de rede físicas, mas é importante conhecer todos eles, pois em algum momento eles podem ser necessários.
- Vejamos alguns exemplos de aplicação do comando ip para gerenciamento de parâmetros de rede em um servidor (ou estação).
- Note que para executar a maior parte dos comandos é necessário possuir privilégios de administrador.

# Exemplos

- 1. Mostrar as interfaces de rede disponíveis no computador:
- # ip link show
- 2. Ver os ips das interfaces de rede:
- # ip addr show
- Ou simplesmente:
- # ip a

```
fabio@debian:~$ ip addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1000
    link/ether 08:00:27:0b:aa:3b brd ff:ff:ff:ff:ff:ff
    inet 10.24.80.60/24 brd 10.24.80.255 scope global dynamic enp0s3
       valid_lft 27887sec preferred_lft 27887sec
    inet6 fe80::a00:27ff:fe0b:aa3b/64 scope link
       valid_lft forever preferred_lft forever
```

# Exemplos

- 3. Ver somente informações sobre o protocolo IPv4 nas interfaces:
- # ip -4 a

- Para o protocolo IPv6:
- # ip -6 a

- 4. Habilitar uma interface de rede, como a enp0s3:
- # ip link set enp0s3 up

- Reiniciar a máquina ou o serviço de rede após:
- #systemctl restart network

# Exemplos

- 5. Desabilitar uma interface de rede, como a enp0s3:
- ip link set enp0s3 down

- 6. Ver o ip de uma interface específica, como a enp0s3:
- # ip addr ls enp0s3

- ou

- # ip addr show enp0s3

# Exemplos

- 7. Ver estatísticas de comunicação (tx e rx) de uma interface específica (opção -s):
- # ip -s link show enp0s3

- 8. Atribuir um endereço IP a uma interface específica:
- # ip addr add 192.168.12.100/24 dev enp0s3

- Obs. Esta configuração é perdida ao reiniciar o sistema.
- Para mantê-la, editar o arquivo de configuração /etc/network/interfaces (Debian e derivados) ou os arquivos em /etc/sysconfig/network-scripts/ (Red Hat e derivados)

# Exemplos

- 9. Excluir um endereço IP de uma interface específica:
- # ip addr del 192.168.12.100/24 dev enp0s3

- 10. Ajustar o nome da interface de rede enp0s3 para eth0:
- # ip link set enp0s3 name eth0

- 11. Verificar as rotas de rede (tabela de roteamento):
- # ip route show

# Exemplos

- 12. Adicionar uma rota estática:
- # ip route add 10.20.30.0/24 via 192.168.100.10 dev enp0s3
- Obs. Esta configuração é perdida ao reiniciar o sistema. Para mantê-la, editar o arquivo de configuração /etc/network/interfaces (Debian e derivados) ou os arquivos em /etc/sysconfig/network-scripts/ (Red Hat e derivados).

- Por exemplo, no debian, adicionamos a linha:
- # ip route add 10.20.30.0/24 via 192.168.100.10 dev enp0s3

- ao arquivo /etc/network/interfaces para adicionar a rota estática de forma permanente.

# Exemplos

- 13. Remover uma rota estática:
- # ip route del 10.20.30.0/24

- 14. Adicionar um endereço de gateway padrão geral
- # ip route add default via 192.168.100.10

- Se o gateway padrão para a rede já existir, será emitida uma mensagem de erro.

- 15. Configurar um endereço MAC em uma interface:
- # ip link set dev enp0s3 address 00:0a:75:20:f5:bd

# Exemplos

- 16. Alterar o MTU em uma interface. Por exemplo, aplicar um MTU de 9000 na interface enp0s3:
- # ip link set mtu 9000 dev enp0s3
- Geralmente alteramos o MTU em redes gigabit para permitir o tráfego de Jumbo Frames (quadros jumbo), de modo a aumentar a performance de transmissão da rede.

- 17. Consultar a tabela ARP:
- # ip neigh

- 18. Consultar a tabela ARP de um interface específica:
- # ip neigh show dev enp0s3

# Exemplos

- 19. Visualizar a ajuda dos comandos ip:
- # ip help

- Ou ainda, visualizar a ajuda apenas dos comandos de endereçamento:
- # ip addr help

- 20. Habilitar o modo promíscuo na interface enp0s3:
- # ip link set enp0s3 promisc on

# Arquivos ifcfg

- A placa física que normalmente se chama eth0, eth1, p1p2 ou enp0s3. Veja com comando: ip a
- Primeiramente edite o arquivo:

- vi /etc/sysconfig/network-scripts/ifcfg-nomedaplaca
- ou seja
- vi /etc/sysconfig/network-scripts/ifcfg-enp0s3

- Para configurar a placa em modo DHCP para que a mesma obtenha um endereço dinâmico automaticamente basta editar o arquivo /etc/sysconfig/network-script/ifcfg-placa e deixar as configurações semelhantes as:

# Arquivos ifcfg

- TYPE=Ethernet
- BOOTPROTO=**dhcp**
- DEFROUTE=yes
- PEERDNS=yes
- PEERROUTES=yes
- IPV4_FAILURE_FATAL=no
- IPV6INIT=yes
- IPV6_AUTOCONF=yes
- IPV6_DEFROUTE=yes
- IPV6_PEERDNS=yes
- IPV6_PEERROUTES=yes
- IPV6_FAILURE_FATAL=no
- NAME=enp0s3
- UUID=610254c7-6b51-48de-9e5b-c07dff952408
- DEVICE=enp0s3
- ONBOOT=**yes**

# Arquivos ifcfg

- **Configurando a placa com um endereço estático**
- Para configurar a placa com um IP estático, primeiro vamos editar o arquivo /etc/sysconfig/network-script/ifcfg-placa e colocar as seguintes configurações adicionais no arquivo.
- Passaremos como argumento para o parâmetro IPADDR o endereço de IP que queremos colocar na nossa placa e no NETMASK nosso endereço de sub rede.

- TYPE=Ethernet
- **BOOTPROTO=static**
- **IPADDR=192.168.1.66**
- **NETMASK=255.255.255.0**
- DEFROUTE=yes
- PEERDNS=yes
- PEERROUTES=yes
- IPV4_FAILURE_FATAL=no
- IPV6INIT=yes
- IPV6_AUTOCONF=yes
- IPV6_DEFROUTE=yes
- IPV6_PEERDNS=yes
- IPV6_PEERROUTES=yes
- IPV6_FAILURE_FATAL=no
- NAME=enp0s3
- UUID=610254c7-6b51-48de-9e5b-c07dff952408
- DEVICE=enp0s3
- **ONBOOT=yes**

# Arquivos ifcfg

```
# Created by anaconda
NETWORKING=yes
HOSTNAME=servidordeteste
GATEWAY=192.168.1.1
```

- **Configurando o Hostname e o Gateway**
- Para definir o nome do host, que por default vem definido como localhost precisamos editar o arquivo /etc/hostname
- vi /etc/hostname
- As configurações de host e domínio do Rocky Linux e do Red Hat vem no padrão hostname.dominio.
- No nosso caso irei configurar somente o hostname, alterando o parâmetro localhost.localdomain para servidorteste.localdomain.
- Agora vamos editar o arquivo /etc/sysconfig/network
- vi /etc/sysconfig/network
- Caso não existem, crie as linhas acima, de forma que o HOSTNAME seja o nome da sua máquina e o GATEWAY o IP do seu Gateway e/ou roteador.

# Comando ss

- Para verificar estatísticas de rede:
- [root@fwnasser ~]# ss -s
- Total: 224
- TCP:  12 (estab 4, closed 4, orphaned 0, timewait 0)

- Transport Total  IP  IPv6
- RAW  1  1  0
- UDP  10  8  2
- TCP  8  5  3
- INET  19  14  5
- FRAG  0  0  0

# Comando ss

- Para ver as portas TCP abertas (em LISTENING):
- # ss -ltn

- Para mostrar as portas TCP e UDP abertas e os processos donos do sockets.
- Para mostrar os processos, o usuário precisa ser o administrador root:
- # ss -ltpu

- Mostra todas as conexões estabelecidas na porta (22) do ssh:
- # ss -o state established '( dport = :22 or sport = :22 )'

# Comando ss

- Suas opções mais comuns são:
- -a: lista todos sockets;
- -r: resolve os endereços IPs e portas por nomes de serviços;
- -n: não resolve os endereços IPs e portas para serviços;
- -l: lista somente as portas abertas (LISTEN);
- -e: mostra informações detalhadas sobre o socket;
- -m: mostra a alocação de memória do socket;
- -p: mostra o processo dono do socket;
- -i: mostra estatísticas do TCP sobre o socket;
- -K: força o fechamento de um socket;
- -s: mostra as estatísticas da rede;
- -t: filtra somente os pacotes TCP;
- -u: filtra somente os pacotes UDP;
- -4: filtra somente os pacotes IPv4;
- -6: filtra somente os pacotes IPv6;