

Desafio B – Configuração de interfaces de rede do servidor

Introdução

O objetivo do desafio B é configurar as interfaces de rede do servidor no Ubuntu 22.04 a partir de um arquivo .yaml usando o software Netplan. O case técnico descreve que os servidores da GigaNuvem possuem uma arquitetura de rede específica, projetada para garantir acesso e clusterização com redundância e, por isso, as quatro interfaces são agregadas em pares utilizando o protocolo LACP. Dessa forma, há garantia da manutenção da conexão caso haja falha de alguma das interfaces.

Adaptação do ambiente de simulação

O desafio B envolve a configuração de interfaces de rede que não possuem acesso fisicamente e que não estão conectadas ao computador. Nesse sentido, foi necessário simular a existência delas a partir do módulo de kernel dummy. Ele permite criar interfaces de rede virtuais que não estão atreladas a nenhum hardware real facilitando a validação das configurações do Netplan. Primeiramente, carreguei o módulo dummy para garantir que estava ativo no sistema.

sudo modprobe dummy

Em seguida, criei as 4 interfaces:

- eno1:

sudo ip link add eno1 type dummy

- enp4s0:

sudo ip link add enp4s0 type dummy

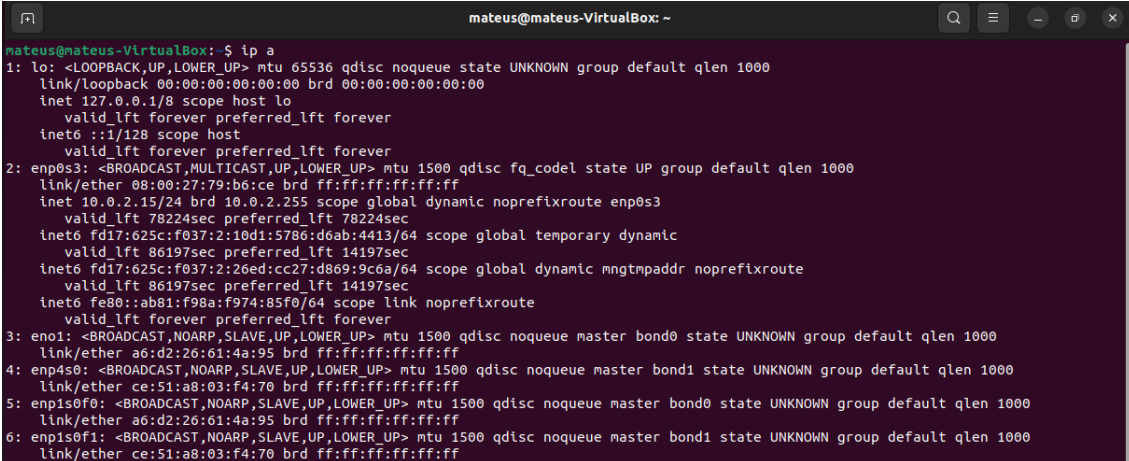
- enp1s0f0:

sudo ip link add enp1s0f0 type dummy

- enp1s0f1:

sudo ip link add enp1s0f1 type dummy

Por fim, verifiquei se as interfaces foram criadas no sistema pelo comando *ip a*.



```
mateus@mateus-VirtualBox: ~  
mateus@mateus-VirtualBox: $ ip a  
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000  
    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 fq_codel state UP group default qlen 1000  
    link/ether 08:00:27:79:b6:ce brd ff:ff:ff:ff:ff:ff  
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3  
        valid_lft 78224sec preferred_lft 78224sec  
    inet6 fd17:625c:f037:2:10d1:5786:d6ab:4413/64 scope global temporary dynamic  
        valid_lft 86197sec preferred_lft 14197sec  
    inet6 fd17:625c:f037:2:26ed:cc27:d869:9c6a/64 scope global dynamic mngtmpaddr noprefixroute  
        valid_lft 86197sec preferred_lft 14197sec  
    inet6 fe80::ab81:f98a:f974:85f0/64 scope link noprefixroute  
        valid_lft forever preferred_lft forever  
3: eno1: <BROADCAST,NOARP,SLAVE,UP,LOWER_UP> mtu 1500 qdisc noqueue master bond0 state UNKNOWN group default qlen 1000  
    link/ether a6:d2:26:61:4a:95 brd ff:ff:ff:ff:ff:ff  
4: enp4s0: <BROADCAST,NOARP,SLAVE,UP,LOWER_UP> mtu 1500 qdisc noqueue master bond1 state UNKNOWN group default qlen 1000  
    link/ether ce:51:a8:03:f4:70 brd ff:ff:ff:ff:ff:ff  
5: enp1s0f0: <BROADCAST,NOARP,SLAVE,UP,LOWER_UP> mtu 1500 qdisc noqueue master bond0 state UNKNOWN group default qlen 1000  
    link/ether a6:d2:26:61:4a:95 brd ff:ff:ff:ff:ff:ff  
6: enp1s0f1: <BROADCAST,NOARP,SLAVE,UP,LOWER_UP> mtu 1500 qdisc noqueue master bond1 state UNKNOWN group default qlen 1000  
    link/ether ce:51:a8:03:f4:70 brd ff:ff:ff:ff:ff:ff
```

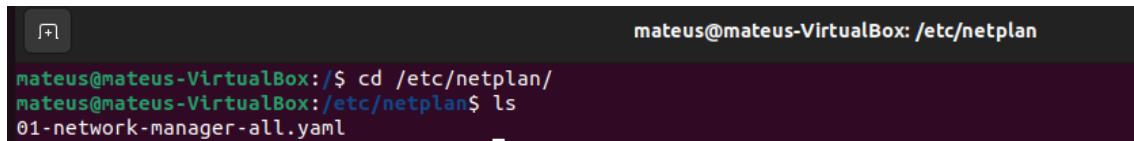
Configuração das interfaces

Primeiramente, é preciso acessar a pasta do sistema `/etc/netplan/` que contém os arquivos de configuração do netplan. Para isso, utilizamos o comando `cd` que é utilizado para alterar o diretório atual do sistema operacional (change directory).

`cd /etc/netplan/`

Com isso, acessamos a pasta do software Netplan que contém os arquivos `.yaml` que permitem a configuração. Em seguida, para ver se já tem algum arquivo criado utilizamos o comando `ls` (list) que listará todos os arquivos presentes na pasta (diretório).

`ls`



```
mateus@mateus-VirtualBox: /etc/netplan
mateus@mateus-VirtualBox:/$ cd /etc/netplan/
mateus@mateus-VirtualBox:/etc/netplan$ ls
01-network-manager-all.yaml
```

Por padrão o arquivo que aparece é `01-network-manager-all.yaml`. Caso não haja nenhum arquivo na pasta, basta utilizar o comando `touch` para criar um novo arquivo `.yaml` e nomeá-lo como desejar.

Exemplo: `touch network-config.yaml`

Considerando que o arquivo padrão já exista não há necessidade de criar um novo. Então, como a intenção é editar o texto presente nele, usaremos o comando `nano` que permite editar arquivos de texto simples.

`nano 01-network-manager-all.yaml`

No entanto, como algumas vezes o arquivo é acessível mesmo sem usar privilégios de administrador antes de começar a editar é interessante modificar a acessibilidade do arquivo para que apenas o usuário `root` consiga fazer alterações nas configurações, por segurança. Nesse sentido, o comando `chmod` é usado para alterar a permissão de acesso ao arquivo.

`sudo chmod 600 /etc/netplan/01-network-manager-all.yaml`

Assim, só será possível acessar o arquivo de texto que contém as configurações das interfaces de rede se o usuário for o administrador (`root`). Com isso, só com o comando `sudo nano` as edições poderão ser realizadas.

`sudo nano 01-network-manager-all.yaml`

```
mateus@mateus-VirtualBox: /etc/netplan
GNU nano 6.2                                01-network-manager-all.yaml
network:
  version: 2
  renderer: networkd #define o uso do systemd-networkd como renderizador de rede

ethernets:
  enol:
    dhcp4: no
    dhcp6: no
  enp4s0:
    dhcp4: no
    dhcp6: no
  enp1s0f0:
    dhcp4: no
    dhcp6: no
  enp1s0f1:
    dhcp4: no
    dhcp6: no
  # dhcp desativado em todas as interfaces
  # desativando tanto para IPv4 quanto para IPv6 para evitar conflitos

bonds:
  bond0:
    interfaces:
      - enol
      - enp1s0f0
    addresses:
      - 10.1.0.1/24
    routes:
      - to: default
        via: 10.1.0.0
        table: 100 # tabela de roteamento para bond0
    routing-policy:
      - from: 10.1.0.0/24
        table: 100 # politica de roteamento para bond0
    nameservers:
      addresses: [10.1.0.100] # servidor DNS
    parameters:
      mode: 802.3ad # agregação das interfaces por LACP
      mii-monitor-interval: 100
      lacp-rate: fast

bondi:
  interfaces:
    - enp4s0

Ajudar  Salir  Gravar  Onde está?  Recortar  Executar  Local  Desfazer  Marcar  Parênteses  Anterior
Onde está?  Substituir  Colar  Justificar  Ir p/ linha  Refazer  Copiar  Onde estava  Próxima
```

Tendo acesso ao conteúdo do arquivo é possível editar e colocar em texto as configurações necessárias para as interfaces de rede. Depois de colocar o texto de configuração no editor, é preciso salvar (CTRL+O) e sair da edição (CTRL+X).

Aplicação da configuração e teste das interfaces

Ao voltar ao terminal, é utilizado o comando `sudo netplan generate` para validar a sintaxe e converter as configurações de rede escritas em formato .yaml para os arquivos de configuração específicos que os serviços de rede do sistema entendem.

`sudo netplan generate`

Por fim, falta apenas aplicar as configurações definitivamente através do comando `netplan apply`.

`sudo netplan apply`

```
mateus@mateus-VirtualBox: /etc/netplan
mateus@mateus-VirtualBox:~$ cd /etc/netplan/
mateus@mateus-VirtualBox:~$ ls
01-network-manager-all.yaml
mateus@mateus-VirtualBox:~$ nano 01-network-manager-all.yaml
[sudo] senha para mateus:
mateus@mateus-VirtualBox:~$ sudo netplan generate
mateus@mateus-VirtualBox:~$ sudo netplan apply
mateus@mateus-VirtualBox:~$ sudo netplan status
Online state: online
DNS Addresses: 127.0.0.53 (stub)
DNS Search: .
● 1: lo ethernet UNKNOWN/UP (unmanaged)
  MAC Address: 00:00:00:00:00:00
  Addresses: 127.0.0.1/8
  Routes: ::1/128
  Routes: ::1 metric 256
● 2: enp3s3 ethernet UP (unmanaged)
  MAC Address: 08:00:27:30:b6:c0 (Intel Corporation)
  Addresses: 10.0.2.15/24
  fe80::625c:f037:2:1801:5786:a6b:d413/64
  fe80::625c:f037:2:1801:5786:a6b:d413/64
  fe80::625c:f037:2:1801:5786:a6b:d413/64
  fe80::625c:f037:2:1801:5786:a6b:d413/64
  DNS Addresses: 10.0.2.3
  fe80::625c:f037:2:1801:5786:a6b:d413/64
  Routes: default via 10.0.2.2 metric 100 (dhcp)
  10.0.2.0/24 from 10.0.2.15 metric 100 (link)
  169.254.0.0/16 metric 1000 (boot, link)
  fe80::625c:f037:2:1801:5786:a6b:d413/64 metric 100 (ra)
  fe80::625c:f037:2:1801:5786:a6b:d413/64 metric 100 (ra)
  default via fe80::625c:f037:2:1801:5786:a6b:d413/64 metric 100 (ra)
● 3: enol ethernet UNKNOWN/UP (networkd: enol)
  MAC Address: a6:d2:26:61:4a:95
● 4: enp4s0 ethernet UNKNOWN/UP (networkd: enp4s0)
  MAC Address: ce:51:a8:03:f4:70
● 5: enp1s0f0 ethernet UNKNOWN/UP (networkd: enp1s0f0)
  MAC Address: a6:d2:26:61:4a:95
● 6: enp1s0f1 ethernet UNKNOWN/UP (networkd: enp1s0f1)
  MAC Address: ce:51:a8:03:f4:70
● 7: bond0 bond UP (networkd: bond0)
  MAC Address: a6:d2:26:61:4a:95
  Addresses: 10.1.0.1/24
  fe80::a6d2:26ff:fe61:4a95/64 (link)
  DNS Addresses: 10.1.0.100
  Routes: 10.1.0.0/24 from 10.1.0.1 (link)
  fe80::a6d2:26ff:fe61:4a95/64 metric 256
● 8: bondi bond UP (networkd: bondi)
  MAC Address: ce:51:a8:03:f4:70
  Addresses: 10.2.0.1/24
  fe80::ccc5:a8ff:fe03:f470/64 (link)
  Routes: 10.2.0.0/24 from 10.2.0.1 (link)
  fe80::ccc5:a8ff:fe03:f470/64 metric 256
```

Depois da aplicação das configurações no netplan, as interfaces estão conforme as especificações das tabelas do desafio B. Podemos ver o estado atual das configurações aplicadas pelo comando *netplan status* que resulta na listagem das interfaces eno1, enp4s0, enp1s0f0 e enp1s0f1 com seus respectivos MAC address e os bonds bond0 e bond1 com os endereços IP, DNS e gateways definidos.

Netplan status

```
mateus@mateus-VirtualBox: /etc/netplan$ netplan status
Online state: online
DNS Addresses: 127.0.0.53 (stub)
DNS Search: .

● 1: lo ethernet UNKNOWN/UP (unmanaged)
  MAC Address: 00:00:00:00:00:00
  Addresses: 127.0.0.1/8
             ::1/128
  Routes: ::1 metric 256

● 2: enp0s3 ethernet UP (unmanaged)
  MAC Address: 08:00:27:79:b6:ce (Intel Corporation)
  Addresses: 10.0.2.15/24
             fd17:625c:f037:2:10d1:5786:d6ab:4413/64
             fd17:625c:f037:2:26ed:cc27:d869:9c6a/64
             fe80::ab81:f98a:f974:85f0/64 (link)
  DNS Addresses: 10.0.2.3
                 fd17:625c:f037:2::3
  Routes: default via 10.0.2.2 metric 100 (dhcp)
          10.0.2.0/24 from 10.0.2.15 metric 100 (link)
          169.254.0.0/16 metric 1000 (boot, link)
          fd17:625c:f037:2::/64 metric 100 (ra)
          fe80::/64 metric 1024
          default via fe80::2 metric 100 (ra)

● 3: eno1 ethernet UNKNOWN/UP (networkd: eno1)
  MAC Address: a6:d2:26:61:4a:95

● 4: enp4s0 ethernet UNKNOWN/UP (networkd: enp4s0)
  MAC Address: ce:51:a8:03:f4:70

● 5: enp1s0f0 ethernet UNKNOWN/UP (networkd: enp1s0f0)
  MAC Address: a6:d2:26:61:4a:95

● 6: enp1s0f1 ethernet UNKNOWN/UP (networkd: enp1s0f1)
  MAC Address: ce:51:a8:03:f4:70

● 7: bond0 bond UP (networkd: bond0)
  MAC Address: a6:d2:26:61:4a:95
  Addresses: 10.1.0.1/24
             fe80::a4d2:26ff:fe61:4a95/64 (link)
  DNS Addresses: 10.1.0.100
                 fe80::/64 metric 256
  Routes: 10.1.0.0/24 from 10.1.0.1 (link)
          fe80::/64 metric 256

● 8: bond1 bond UP (networkd: bond1)
  MAC Address: ce:51:a8:03:f4:70
  Addresses: 10.2.0.1/24
             fe80::cc51:a8ff:fe03:f470/64 (link)
  Routes: 10.2.0.0/24 from 10.2.0.1 (link)
          fe80::/64 metric 256

mateus@mateus-VirtualBox: /etc/netplan$
```

Para testes adicionais é necessário baixar o pacote net-tools no Ubuntu que contém ferramentas básicas de rede, para isso basta usar o comando seguinte: *sudo apt install net-tools*.

```
mateus@mateus-VirtualBox: ~$ sudo apt install net-tools
Lendo listas de pacotes... Pronto
Construindo árvore de dependências... Pronto
Lendo informação de estado... Pronto
Os NOVOS pacotes a seguir serão instalados:
  net-tools
0 pacotes atualizados, 1 pacotes novos instalados, 0 a serem removidos e 0 não atualizados.
É preciso baixar 204 kB de arquivos.
Depois desta operação, 819 kB adicionais de espaço em disco serão usados.
Obter:1 http://br.archive.ubuntu.com/ubuntu jammy-updates/main amd64 net-tools amd64 1.60+git20181103.0eebece-1ubuntu5.4 [204 kB]
Baixados 204 kB em 6s (32,3 kB/s)
A seleccionar pacote anteriormente não seleccionado net-tools.
(Lendo banco de dados ... 183792 ficheiros e directórios actualmente instalados.)
A preparar para descompactar .../net-tools_1.60+git20181103.0eebece-1ubuntu5.4_amd64.deb ...
A descompactar net-tools (1.60+git20181103.0eebece-1ubuntu5.4) ...
Configurando net-tools (1.60+git20181103.0eebece-1ubuntu5.4) ...
A processar 'triggers' para man-db (2.10.2-1) ...
mateus@mateus-VirtualBox: ~$
```

Em seguida, podemos testar a aplicação das configurações nas interfaces utilizando os comandos do pacote net tools como, por exemplo, *ifconfig* para ver os endereços IP, máscaras de rede e outras informações da interface de rede, *route -n* para verificar as rotas da rede e *resolvectl status* para verificar as configurações do servidor DNS.

```
mateus@mateus-VirtualBox:~$ ifconfig
bond0: flags=5187<UP,BROADCAST,RUNNING,MASTER,MULTICAST> mtu 1500
    inet 10.1.0.1 netmask 255.255.255.0 broadcast 10.1.0.255
    inet6 fe80::a402:26ff:fe61:4a95 prefixlen 64 scopeid 0x20<link>
    ether a5:d2:26:61:4a:95 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 158 bytes 10922 (10.9 KB)
    TX errors 0 dropped 687 overruns 0 carrier 0 collisions 0

bond1: flags=5187<UP,BROADCAST,RUNNING,MASTER,MULTICAST> mtu 1500
    inet 10.2.0.1 netmask 255.255.255.0 broadcast 10.2.0.255
    inet6 fe80::cc51:a8ff:fe03:f470 prefixlen 64 scopeid 0x20<link>
    ether ce:51:a8:03:f4:70 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 54 bytes 6388 (6.3 KB)
    TX errors 0 dropped 89 overruns 0 carrier 0 collisions 0

eno1: flags=2243<UP,BROADCAST,RUNNING,NOARP,SLAVE> mtu 1500
    ether a5:d2:26:61:4a:95 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::ab81:f98a:f974:85f0 prefixlen 64 scopeid 0x20<link>
    inet6 fd17:625c:f037:2:18d1:5786:d6ab:4413 prefixlen 64 scopeid 0x0<global>
    inet6 fd17:625c:f037:2:26ed:cc27:d869:9c5a prefixlen 64 scopeid 0x0<global>
    ether 08:00:27:79:b5:c6 txqueuelen 1000 (Ethernet)
    RX packets 41115 bytes 50315931 (50.3 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 9230 bytes 1775315 (1.7 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp1s0f0: flags=2243<UP,BROADCAST,RUNNING,NOARP,SLAVE> mtu 1500
    ether a5:d2:26:61:4a:95 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 158 bytes 10922 (10.9 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp1s0f1: flags=2243<UP,BROADCAST,RUNNING,NOARP,SLAVE> mtu 1500
    ether ce:51:a8:03:f4:70 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 54 bytes 6388 (6.3 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp4s0: flags=2243<UP,BROADCAST,RUNNING,NOARP,SLAVE> mtu 1500
    ether ce:51:a8:03:f4:70 txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Loopback Local)
    RX packets 2590 bytes 268723 (268.7 KB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2590 bytes 268723 (268.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```
mateus@mateus-VirtualBox:~$ resolvectl status
Global
    Protocols: -LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    resolv.conf mode: stub

Link 2 (enp0s3)
    Current Scopes: DNS
    Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    Current DNS Server: fd17:625c:f037:2::3
    DNS Servers: 10.0.2.3 fd17:625c:f037:2::3

Link 3 (eno1)
    Current Scopes: none
    Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported

Link 4 (enp4s0)
    Current Scopes: none
    Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported

Link 5 (enp1s0f0)
    Current Scopes: none
    Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported

Link 6 (enp1s0f1)
    Current Scopes: none
    Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported

Link 7 (bond0)
    Current Scopes: DNS
    Protocols: +DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported
    Current DNS Server: 10.1.0.100
    DNS Servers: 10.1.0.100

Link 8 (bond1)
    Current Scopes: none
    Protocols: -DefaultRoute +LLMNR -mDNS -DNSOverTLS DNSSEC=no/unsupported

mateus@mateus-VirtualBox:~$ route -n
Tabela de Roteamento IP do Kernel

```

Destino	Roteador	MáscaraGen.	Opções	Métrica	Ref	Uso	Iface
0.0.0.0	10.0.2.2	0.0.0.0	UG	100	0	0	enp0s3
10.0.2.0	0.0.0.0	255.255.255.0	U	100	0	0	enp0s3
10.1.0.0	0.0.0.0	255.255.255.0	U	0	0	0	bond0
10.2.0.0	0.0.0.0	255.255.255.0	U	0	0	0	bond1
169.254.0.0	0.0.0.0	255.255.0.0	U	1000	0	0	enp0s3

```
mateus@mateus-VirtualBox:~$
```

Conclusão

A configuração de interfaces de rede para o servidor via Netplan foi concluída com sucesso, seguindo o desenho da arquitetura e as tabelas propostas. Para validar o seu funcionamento, a configuração foi testada em uma máquina virtual, resultando em um arquivo netplan pronto para implementação no servidor final, atendendo a todos os requisitos do desafio.

Para a fase de testes, optou-se por utilizar o Ubuntu desktop em vez do Ubuntu server. Essa escolha se baseia na conveniência da interface gráfica, que simplifica a manipulação de arquivos e a visualização do ambiente. O Ubuntu desktop se mostrou suficiente para a simulação, pois o Netplan permite o uso do mesmo renderizador (systemd-networkd) em ambas as plataformas. Com isso, o comportamento de rede do Ubuntu server é replicado com fidelidade, garantindo a validade técnica da simulação sem abrir mão da praticidade do ambiente gráfico.