Red Hat Enterprise Linux 8

Configuração e gerenciamento de redes

Um guia para configuração e gerenciamento de redes no Red Hat Enterprise Linux 8

Last Updated: 2021-02-24

Red Hat Enterprise Linux 8 Configuração e gerenciamento de redes

Um guia para configuração e gerenciamento de redes no Red Hat Enterprise Linux 8

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

Nota Legal

Resumo

Este documento descreve como gerenciar o networking no Red Hat Enterprise Linux 8.

Índice

TORNANDO O CÓDIGO ABERTO MAIS INCLUSIVO

A Red Hat tem o compromisso de substituir a linguagem problemática em nosso código, documentação e propriedades da web. Estamos começando com estes quatro termos: master, slave, blacklist e whitelist. Por causa da enormidade deste esforço, estas mudanças serão implementadas gradualmente ao longo de vários lançamentos futuros. Para mais detalhes, veja a mensagem de nosso CTO Chris Wright.

FORNECENDO FEEDBACK SOBRE A DOCUMENTAÇÃO DA RED HAT

Agradecemos sua contribuição em nossa documentação. Por favor, diga-nos como podemos melhorá-la. Para fazer isso:

- Para comentários simples sobre passagens específicas:
  1. Certifique-se de que você está visualizando a documentação no formato *Multi-page HTML*. Além disso, certifique-se de ver o botão **Feedback** no canto superior direito do documento.
  2. Use o cursor do mouse para destacar a parte do texto que você deseja comentar.
  3. Clique no pop-up **Add Feedback** que aparece abaixo do texto destacado.
  4. Siga as instruções apresentadas.
- Para enviar comentários mais complexos, crie um bilhete Bugzilla:
  1. Ir para o site da Bugzilla.
  2. Como Componente, use **Documentation**.
  3. Preencha o campo **Description** com sua sugestão de melhoria. Inclua um link para a(s) parte(s) relevante(s) da documentação.
  4. Clique em **Submit Bug**.

CAPÍTULO 1. TÓPICOS GERAIS DA REDE RHEL

Esta seção fornece detalhes sobre tópicos gerais de redes.

1.1. A DIFERENÇA ENTRE REDES IP E NÃO IP

Uma rede é um sistema de dispositivos interligados que podem se comunicar compartilhando informações e recursos, tais como arquivos, impressoras, aplicações e conexão com a Internet. Cada um desses dispositivos tem um endereço IP único para enviar e receber mensagens entre dois ou mais dispositivos, usando um conjunto de regras chamado protocolo.

Categorias de comunicação em rede:

Redes IP

Redes que se comunicam através de endereços IP. Uma rede IP é implementada na Internet e na maioria das redes internas. Ethernet, redes sem fio e conexões VPN são exemplos típicos.

Redes não-IP

Redes que são utilizadas para se comunicar através de uma camada inferior, em vez da camada de transporte. Note que estas redes são raramente utilizadas. Por exemplo, a InfiniBand é uma rede não-IP.

1.2. A DIFERENÇA ENTRE ENDEREÇAMENTO IP ESTÁTICO E DINÂMICO

Endereçamento IP estático

Quando você atribui um endereço IP estático a um dispositivo, o endereço não muda com o tempo, a menos que você o altere manualmente. Use endereço IP estático, se desejar:

- Para garantir a consistência de endereços de rede para servidores como o DNS e servidores de autenticação.
- Utilizar dispositivos de gerenciamento fora da banda que funcionem independentemente de outras infra-estruturas de rede.

Endereçamento IP dinâmico

Quando você configura um dispositivo para usar um endereço IP dinâmico, o endereço pode mudar com o tempo. Por esta razão, os endereços dinâmicos são normalmente usados para dispositivos que se conectam à rede ocasionalmente porque o endereço IP pode ser diferente após reiniciar o host. Os endereços IP dinâmicos são mais flexíveis, mais fáceis de configurar e de administrar. O Protocolo de Controle Dinâmico de Host (DHCP) é um método tradicional de atribuição dinâmica de configurações de rede a hosts.

NOTA

Não há uma regra rígida que defina quando usar endereços IP estáticos ou dinâmicos. Depende das necessidades do usuário, das preferências e do ambiente de rede.

Recursos adicionais

Para detalhes sobre a instalação de um servidor DHCP, veja Capítulo 43, *Prestação de serviços de DHCP*.

1.3. FASES DE TRANSAÇÃO DO DHCP

O DHCP funciona em quatro fases: Descoberta, Oferta, Pedido, Reconhecimento, também chamado processo DORA. O DHCP usa este processo para fornecer endereços IP aos clientes.

Descoberta
O cliente DHCP envia uma mensagem para descobrir o servidor DHCP na rede. Esta mensagem é transmitida na rede e na camada de link de dados.

Oferta
O servidor DHCP recebe mensagens do cliente e oferece um endereço IP para o cliente DHCP. Esta mensagem é unicast na camada de link de dados, mas é transmitida na camada de rede.

Solicitação
O cliente DHCP solicita o servidor DHCP para o endereço IP oferecido. Esta mensagem é unicast na camada de link de dados, mas é transmitida na camada de rede.

Agradecimentos
O servidor DHCP envia um agradecimento ao cliente DHCP. Esta mensagem é unicast na camada de link de dados, mas transmitida na camada de rede. É a mensagem final do processo DHCP DORA.

1.4. REDES INFINIBAND E RDMA

Para obter detalhes sobre as redes InfiniBand e Remote Direct Memory Access (RDMA), consulte a documentação Configurando redes InfiniBand e RDMA.

1.5. SUPORTE A SCRIPTS DE REDE LEGADOS NA RHEL

Por padrão, a RHEL usa o NetworkManager para configurar e gerenciar conexões de rede, e os scripts **/usr/sbin/ifup** e **/usr/sbin/ifdown** usam o NetworkManager para processar **ifcfg** arquivos no diretório **/etc/sysconfig/network-scripts/**.

Entretanto, se você precisar dos scripts de rede depreciados que processam a configuração da rede sem usar o NetworkManager, você pode instalá-los:

```
# yum install network-scripts
```

Após a instalação dos scripts de rede legados, os scripts **/usr/sbin/ifup** e **/usr/sbin/ifdown** conectam-se aos scripts shell depreciados que gerenciam a configuração da rede.

IMPORTANTE

Os scripts legados são depreciados no RHEL 8 e serão removidos em uma futura versão principal do RHEL. Se você ainda usa os scripts de rede legados, por exemplo, porque você atualizou de uma versão anterior para a RHEL 8, a Red Hat recomenda que você migre sua configuração para o NetworkManager.

1.6. SELEÇÃO DE MÉTODOS DE CONFIGURAÇÃO DE REDE

- Para configurar uma interface de rede usando o NetworkManager, use uma das seguintes ferramentas:
  - a interface de usuário de texto, **nmtui**.

    - o utilitário de linha de comando , **nmcli**.
    - as ferramentas gráficas de interface com o usuário, **GNOME GUI**.

- Para configurar uma interface de rede sem utilizar ferramentas e aplicações NetworkManager:
    - editar os arquivos **ifcfg** manualmente. Note que mesmo que você edite os arquivos diretamente, o NetworkManager é o padrão no RHEL e processa estes arquivos. Somente se você instalou e habilitou os scrips de rede obsoletos, então estes scripts processam os arquivos **ifcfg**.

- Para configurar as configurações de rede quando o sistema de arquivo raiz não é local:
    - usar a linha de comando do kernel.

Recursos adicionais

- Capítulo 5, *Começando com nmtui*
- Capítulo 6, *Começando com nmcli*
- Capítulo 7, *Começando com a configuração de redes usando a GUI GNOME*
- Seção 1.5, "Suporte a scripts de rede legados na RHEL"

CAPÍTULO 2. NOME DE DISPOSITIVOS DE INTERFACE DE REDE CONSISTENTES

O Red Hat Enterprise Linux 8 fornece métodos para a nomeação consistente e previsível de dispositivos para interfaces de rede. Estas características ajudam a localizar e diferenciar as interfaces de rede.

O kernel atribui nomes às interfaces de rede concatenando um prefixo fixo e um número que aumenta à medida que o kernel inicializa os dispositivos de rede. Por exemplo, **eth0** representaria o primeiro dispositivo a ser sondado na inicialização. No entanto, estes nomes não correspondem necessariamente a etiquetas no chassi. Plataformas modernas de servidores com múltiplos adaptadores de rede podem encontrar nomes não determinísticos e contra-intuitivos destas interfaces. Isto afeta tanto os adaptadores de rede incorporados na placa do sistema quanto os adaptadores add-in.

No Red Hat Enterprise Linux 8, o gerente do dispositivo **udev** suporta uma série de diferentes esquemas de nomenclatura. Por default, **udev** atribui nomes fixos com base no firmware, topologia e informações de localização. Isto tem as seguintes vantagens:

- Os nomes dos dispositivos são totalmente previsíveis.
- Os nomes dos dispositivos permanecem fixos mesmo se você adicionar ou remover hardware, pois não há re-enumeração.
- O hardware defeituoso pode ser substituído sem problemas.

2.1. HIERARQUIA DE NOMES DE DISPOSITIVOS DE INTERFACE DE REDE

Se a nomeação consistente de dispositivos estiver ativada, que é o padrão no Red Hat Enterprise Linux 8, o gerenciador de dispositivos **udev** gera nomes de dispositivos com base nos seguintes esquemas:

Esquema	Descrição	Exemplo
1	Os nomes dos dispositivos incorporam firmware ou números de índice fornecidos pela BIOS para os dispositivos embarcados. Se esta informação não estiver disponível ou aplicável, **udev** utiliza o esquema 2.	**eno1**
2	Os nomes dos dispositivos incorporam o firmware ou os números de índice de hot slot PCI Express (PCIe) fornecidos pela BIOS. Se esta informação não estiver disponível ou aplicável, **udev** utiliza o esquema 3.	**ens1**

Esquema	Descrição	Exemplo
3	Os nomes dos dispositivos incorporam a localização física do conector do hardware. Se esta informação não estiver disponível ou aplicável, **udev** utiliza o esquema 5.	**enp2s0**
4	Os nomes dos dispositivos incorporam o endereço MAC. O Red Hat Enterprise Linux não usa este esquema por default, mas os administradores podem usá-lo opcionalmente.	**enx525400d5e0fb**
5	O tradicional e imprevisível esquema de nomenclatura do núcleo. Se **udev** não puder aplicar nenhum dos outros esquemas, o gerente do dispositivo usa este esquema.	**eth0**

Por default, o Red Hat Enterprise Linux seleciona o nome do dispositivo com base na configuração **NamePolicy** no arquivo **/usr/lib/systemd/network/99-default.link**. A ordem dos valores em **NamePolicy** é importante. O Red Hat Enterprise Linux usa o primeiro nome de dispositivo que é especificado no arquivo e que foi gerado em **udev**.

Se você configurou manualmente as regras **udev** para mudar o nome dos dispositivos do kernel, essas regras têm precedência.

2.2. COMO FUNCIONA A RENOMEAÇÃO DO DISPOSITIVO DE REDE

Por default, a nomeação consistente dos dispositivos é ativada no Red Hat Enterprise Linux 8. O gerente de dispositivos **udev** processa regras diferentes para renomear os dispositivos. A lista a seguir descreve a ordem na qual **udev** processa estas regras e por quais ações estas regras são responsáveis:

1. O arquivo **/usr/lib/udev/rules.d/60-net.rules** define que o utilitário helper **/lib/udev/rename_device** busca o parâmetro **HWADDR** nos arquivos **/etc/sysconfig/network-scripts/ifcfg-***. Se o valor definido na variável corresponder ao endereço MAC de uma interface, o utilitário helper renomeia a interface para o nome definido no parâmetro **DEVICE** do arquivo.

2. O arquivo **/usr/lib/udev/rules.d/71-biosdevname.rules** define que o utilitário **biosdevname** renomeia a interface de acordo com sua política de nomenclatura, desde que não tenha sido renomeado na etapa anterior.

3. O arquivo **/usr/lib/udev/rules.d/75-net-description.rules** define que **udev** examina o dispositivo de interface de rede e define as propriedades em **udev**- variáveis internas, que serão processadas na próxima etapa. Note que algumas destas propriedades podem estar indefinidas.

4. O arquivo **/usr/lib/udev/rules.d/80-net-setup-link.rules** chama o **net_setup_link udev** embutido que então aplica a política. A seguir está a política padrão que está armazenada no arquivo **/usr/lib/systemd/network/99-default.link**:

```
[Link]
NamePolicy=kernel database onboard slot path
MACAddressPolicy=persistent
```

   Com esta política, se o kernel usa um nome persistente, **udev** não renomeia a interface. Se o kernel não usar um nome persistente, **udev** renomeia a interface para o nome fornecido pelo banco de dados de hardware de **udev**. Se este banco de dados não estiver disponível, o Red Hat Enterprise Linux volta aos mecanismos descritos acima.

   Alternativamente, defina o parâmetro **NamePolicy** neste arquivo para **mac** para controle de acesso à mídia (MAC) nomes de interface baseados em endereços.

5. O arquivo **/usr/lib/udev/rules.d/80-net-setup-link.rules** define que **udev** renomeia a interface com base no **udev**- parâmetros internos na seguinte ordem:

   a. **ID_NET_NAME_ONBOARD**

   b. **ID_NET_NAME_SLOT**

   c. **ID_NET_NAME_PATH**

   Se um parâmetro não estiver definido, **udev** usa o próximo. Se nenhum dos parâmetros estiver definido, a interface não é renomeada.

Os passos 3 e 4 implementam os esquemas de nomenclatura 1 a 4 descritos em Seção 2.1, “Hierarquia de nomes de dispositivos de interface de rede”.

Recursos adicionais

- Para detalhes sobre a definição de prefixos personalizados para nomeação consistente, ver Seção 2.7, “Utilização de prefixo para nomeação de interfaces de rede Ethernet” .
- Para obter detalhes sobre o parâmetro **NamePolicy**, consulte a página de manual **systemd.link(5)**.

2.3. NOMES DE DISPOSITIVOS DE INTERFACE DE REDE PREVISÍVEIS NA PLATAFORMA X86_64 EXPLICADOS

Quando o recurso de nome consistente do dispositivo de rede é ativado, o gerenciador de dispositivos **udev** cria os nomes dos dispositivos com base em diferentes critérios. Esta seção descreve o esquema de nomenclatura quando o Red Hat Enterprise Linux 8 é instalado em uma plataforma x86_64.

O nome da interface começa com um prefixo de dois caracteres com base no tipo de interface:

- **en** para Ethernet
- **wl** para LAN sem fio (WLAN)
- **ww** para rede de área ampla sem fio (WWAN)

Além disso, um dos seguintes itens é anexado a um dos prefixos acima mencionados com base no esquema que o gerente do dispositivo **udev** aplica:

- **o*<on-board_index_number>***
- **s*<hot_plug_slot_index_number>*[f*<function>*][d*<device_id>*]**
  Note que todos os dispositivos PCI multi-função têm o **[f*<function>*]** número no nome do dispositivo, incluindo a função **0** dispositivo.
- **x*<MAC_address>***
- **[P*<domain_number>*]p*<bus>*s*<slot>*[f*<function>*][d*<device_id>*]**
  O **[P*<domain_number>*]** parte define a localização geográfica do PCI. Esta parte só é definida se o número de domínio não for **0**.
- **[P*<domain_number>*]p*<bus>*s*<slot>*[f*<function>*][u*<usb_port>*][...][c*<config>*][i*<interface>*]**
  Para dispositivos USB, a cadeia completa de números de portas de hubs é composta. Se o nome for maior do que o máximo (15 caracteres), o nome não é exportado. Se houver múltiplos dispositivos USB na cadeia, **udev** suprime os valores padrão para os descritores de configuração USB (**c1**) e os descritores de interface USB (**i0**).

2.4. NOMES DE DISPOSITIVOS DE INTERFACE DE REDE PREVISÍVEIS NA PLATAFORMA SYSTEM Z EXPLICADOS

Quando o recurso consistente de nome do dispositivo de rede é ativado, o gerenciador de dispositivos **udev** na plataforma System z cria os nomes dos dispositivos com base no ID do ônibus. O ID do barramento identifica um dispositivo no subsistema de canais s390.

Para um dispositivo de palavra de comando de canal (CCW), o ID do ônibus é o número do dispositivo com um prefixo principal **0.n** onde **n** é o ID do conjunto de sub-canais.

As interfaces Ethernet são nomeadas, por exemplo, **enccw0.0.1234**. Os dispositivos de rede Serial Line Internet Protocol (SLIP) channel-to-channel (CTC) são nomeados, por exemplo, **slccw0.0.1234**.

Use os comandos **znetconf -c** ou **lscss -a** para exibir os dispositivos de rede disponíveis e suas identificações de ônibus.

2.5. DESATIVAÇÃO DE NOMES CONSISTENTES DE DISPOSITIVOS DE INTERFACE DURANTE A INSTALAÇÃO

Esta seção descreve como desativar a nomeação consistente do dispositivo de interface durante a instalação.

ATENÇÃO

A Red Hat recomenda não desativar a nomeação consistente do dispositivo. A desativação de nomes consistentes de dispositivos pode causar diferentes tipos de problemas. Por exemplo, se você adicionar outra placa de interface de rede ao sistema, a atribuição dos nomes dos dispositivos do kernel, tais como **eth0**, não é mais corrigida. Conseqüentemente, após uma reinicialização, o Kernel pode nomear o dispositivo de maneira diferente.

Procedimento

1. Inicialize a mídia de instalação do Red Hat Enterprise Linux 8.
2. No gerenciador de boot, selecione **Install Red Hat Enterprise Linux 8**, e pressione a tecla **Tab** para editar a entrada.
3. Anexar o parâmetro **net.ifnames=0** à linha de comando do kernel:

   ```
   vmlinuz.. net.ifnames=0
   ```

4. Pressione **Enter** para iniciar a instalação.

Recursos adicionais

- É seguro definir net.ifnames=0 no RHEL 7 e RHEL 8?
- Como realizar uma atualização no local para o RHEL 8 ao usar os nomes NIC do kernel no RHEL 7

2.6. DESABILITANDO A NOMEAÇÃO CONSISTENTE DE DISPOSITIVOS DE INTERFACE EM UM SISTEMA INSTALADO

Esta seção descreve como desativar a nomeação consistente de dispositivos de interface em um sistema que já está instalado.

ATENÇÃO

A Red Hat recomenda não desativar a nomeação consistente do dispositivo. A desativação de nomes consistentes de dispositivos pode causar diferentes tipos de problemas. Por exemplo, se você adicionar outra placa de interface de rede ao sistema, a atribuição dos nomes dos dispositivos do kernel, tais como **eth0**, não é mais corrigida. Conseqüentemente, após uma reinicialização, o Kernel pode nomear o dispositivo de maneira diferente.

Pré-requisitos

- O sistema usa uma nomenclatura consistente de dispositivos de interface, que é o padrão.

Procedimento

1. Editar o arquivo **/etc/default/grub** e anexar o parâmetro **net.ifnames=0** à variável **GRUB_CMDLINE_LINUX**:

   ```
   GRUB_CMDLINE_LINUX="... *net.ifnames=0
   ```

2. Reconstruir o arquivo **grub.cfg**:
   - Em um sistema com modo de inicialização UEFI:
     -

```
# grub2-mkconfig -o /boot/efi/efi/EFI/redhat/grub.cfg
```

- Em um sistema com modo de inicialização herdado:

```
# grub2-mkconfig -o /boot/grub2/grub.cfg
```

3. Se você utiliza nomes de interface em arquivos de configuração ou scripts, você deve atualizá-los manualmente.

4. Reiniciar o anfitrião:

```
# reinicialização
```

2.7. UTILIZAÇÃO DE PREFIXO PARA NOMEAÇÃO DE INTERFACES DE REDE ETHERNET

Esta documentação descreve como definir os prefixos para nomeação consistente de interfaces de rede Ethernet caso você não queira usar o esquema de nomeação padrão de tais interfaces. No entanto, a Red Hat recomenda o uso do esquema de nomenclatura padrão. Para mais detalhes sobre este esquema, veja Capítulo 2, *Nome de dispositivos de interface de rede consistentes* .

2.7.1. Introdução ao prefixo

A ferramenta **prefixdevname** é um utilitário udev helper que permite definir seu próprio prefixo usado para nomear as interfaces de rede Ethernet.

2.7.2. Limitações do prefixo do nome

Existem certas limitações para os prefixos das interfaces de rede Ethernet.

O prefixo que você escolher deve atender aos seguintes requisitos

- Ser cordão ASCII
- Ser cadeia alfanumérica
- Ser menor que 16 caracteres

ATENÇÃO

O prefixo não pode entrar em conflito com nenhum outro prefixo bem conhecido utilizado para a nomenclatura da interface de rede no Linux. Especificamente, você não pode usar estes prefixos: **eth**, **eno**, **ens**, **em**.

2.7.3. Definição do prefixo do nome

A definição do prefixo com **prefixdevname** é feita durante a instalação do sistema.

Para definir e ativar o prefixo necessário para suas interfaces de rede Ethernet, use o seguinte procedimento.

Procedimento

- Adicione a seguinte seqüência na linha de comando do kernel:

```
net.ifnames.prefixo=<prefixo exigido>
```

ATENÇÃO

A Red Hat não suporta o uso do **prefixdevname** em sistemas já implantados.

Depois que o prefixo foi definido e o sistema operacional foi reinicializado, o prefixo é efetivo toda vez que uma nova interface de rede aparece. Ao novo dispositivo é atribuído um nome na forma de **<PREFIX><INDEX>**. Por exemplo, se seu prefixo selecionado é **net**, e as interfaces com **net0** e **net1** já existem no sistema, a nova interface é denominada **net2**. O utilitário **prefixdevname** gera então o novo arquivo **.link** no diretório **/etc/systemd/network** que aplica o nome à interface com o endereço MAC que acabou de aparecer. A configuração é persistente em todas as reinicializações.

2.8. INFORMAÇÕES RELACIONADAS

- Consulte a página de manual **udev(7)** para obter detalhes sobre o gerenciador de dispositivos **udev**.

CAPÍTULO 3. COMEÇANDO COM O NETWORKMANAGER

Por padrão, o RHEL 8 utiliza o NetworkManager para gerenciar a configuração e as conexões da rede.

3.1. BENEFÍCIOS DE USAR O NETWORKMANAGER

Os principais benefícios de utilizar o NetworkManager são:

- Oferecendo uma API através do D-Bus que permite consultar e controlar a configuração e o estado da rede. Desta forma, a rede pode ser verificada e configurada por múltiplas aplicações, garantindo um estado de rede sincronizado e atualizado. Por exemplo, o console web RHEL, que monitora e configura os servidores através de um navegador web, utiliza o **NetworkManager** Interface D-BUS para configurar a rede, assim como as ferramentas **Gnome GUI**, **nmcli** e **nm-connection-editor**. Cada mudança feita em uma destas ferramentas é detectada por todas as outras.

- Tornando a gestão da rede mais fácil **NetworkManager** garante que a conectividade de rede funcione. Quando detecta que não há configuração de rede em um sistema, mas que existem dispositivos de rede, **NetworkManager** cria conexões temporárias para fornecer conectividade.

- Proporcionando fácil configuração da conexão com o usuário **NetworkManager** oferece gerenciamento através de diferentes ferramentas - **GUI, nmtui, nmcli**.

- Apoio à flexibilidade de configuração. Por exemplo, a configuração de uma interface WiFi, **NetworkManager** escaneia e mostra as redes wifi disponíveis. Você pode selecionar uma interface, e **NetworkManager** exibe as credenciais necessárias para a conexão automática após o processo de reinicialização **NetworkManager** pode configurar aliases de rede, endereços IP, rotas estáticas, informações DNS e conexões VPN, assim como muitos parâmetros específicos de conexão. Você pode modificar as opções de configuração para refletir suas necessidades.

- Manter o estado dos dispositivos após o processo de reinicialização e assumir as interfaces que são colocadas em modo gerenciado durante o reinício.

- Dispositivos de manuseio que não são explicitamente definidos sem gerenciamento, mas controlados manualmente pelo usuário ou outro serviço de rede.

Recursos adicionais

- Para mais informações sobre a instalação e uso do console web RHEL 8, consulte Sistemas de gerenciamento usando o console web RHEL 8.

3.2. UMA VISÃO GERAL DAS UTILIDADES E APLICAÇÕES QUE VOCÊ PODE USAR PARA GERENCIAR AS CONEXÕES DO NETWORKMANAGER

Você pode usar as seguintes utilidades e aplicações para gerenciar as conexões do NetworkManager:

- **nmcli**: Um utilitário de linha de comando para gerenciar as conexões.

- **nmtui**: Uma interface de usuário de texto baseada em curses (TUI). Para utilizar este aplicativo, instale o pacote **NetworkManager-tui**.

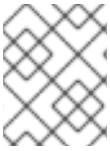
- **nm-connection-editor**: Uma interface gráfica do usuário (GUI) para tarefas relacionadas ao NetworkManager. Para iniciar este aplicativo, entre em **nm-connection-editor** em um terminal de uma sessão do GNOME.
- **control-center**: Uma GUI fornecida pela shell do GNOME para usuários desktop. Note que este aplicativo suporta menos recursos do que **nm-connection-editor**.
- O **network connection icon** na concha do GNOME: Este ícone representa estados de conexão de rede e serve como indicador visual para o tipo de conexão que você está usando.

Recursos adicionais

- Capítulo 5, *Começando com nmtui*
- Capítulo 6, *Começando com nmcli*
- Capítulo 7, *Começando com a configuração de redes usando a GUI GNOME*

3.3. UTILIZAÇÃO DE SCRIPTS DE DESPACHO NETWORKMANAGER

Por padrão, o diretório **/etc/NetworkManager/dispatcher.d/** existe e **NetworkManager** executa scripts lá, em ordem alfabética. Cada script deve ser um arquivo executável **owned by root** e deve ter **write permission** somente para o proprietário do arquivo.



NOTA

O NetworkManager executa os scripts do despachante em **/etc/NetworkManager/dispatcher.d/** em ordem alfabética.

Recursos adicionais

- Para um exemplo de um script de despachante, veja Como escrever um script de despachante NetworkManager para aplicar a solução de comandos ethtool.

3.4. CARREGAMENTO DE ARQUIVOS IFCFG CRIADOS MANUALMENTE NO NETWORKMANAGER

No Red Hat Enterprise Linux 8, se você editar um arquivo **ifcfg**, **NetworkManager** não está automaticamente ciente da mudança e tem que ser avisado da mudança. Se você usar uma das ferramentas para atualizar **NetworkManager** configurações de perfil, **NetworkManager** não implementa essas mudanças até que você se reconecte usando esse perfil. Por exemplo, se os arquivos de configuração tiverem sido alterados utilizando um editor, **NetworkManager** deve ler novamente os arquivos de configuração.

O diretório **/etc/sysconfig/** é um local para arquivos de configuração e scripts. A maioria das informações de configuração da rede é armazenada lá, com exceção das configurações VPN, banda larga móvel e PPPoE, que são armazenadas nos subdiretórios **/etc/NetworkManager/**. Por exemplo, informações específicas da interface são armazenadas nos arquivos **ifcfg** no diretório **/etc/sysconfig/network-scripts/**.

As informações para VPNs, banda larga móvel e conexões PPPoE são armazenadas em **/etc/NetworkManager/system-connections/**.

NOTA

Por padrão, a RHEL usa o NetworkManager para configurar e gerenciar conexões de rede, e os scripts **/usr/sbin/ifup** e **/usr/sbin/ifdown** usam o NetworkManager para processar **ifcfg** arquivos no diretório **/etc/sysconfig/network-scripts/**.

Se você precisar dos scripts de rede legados para gerenciar suas configurações de rede, você pode instalá-los manualmente. Para maiores detalhes, veja Seção 1.5, "Suporte a scripts de rede legados na RHEL". Entretanto, observe que os scripts de rede legados são depreciados e serão removidos em uma versão futura da RHEL.

Procedimento

1. Para carregar um novo arquivo de configuração:

```
# nmcli connection load /etc/sysconfig/network-scripts/ifcfg-connection_name
```

2. Se você atualizou um arquivo de conexão que já tenha sido carregado no NetworkManager, entre:

```
# nmcli connection up connection_name
```

Recursos adicionais

- **NetworkManager(8)** man page - Descreve o daemon de gestão da rede.
- **NetworkManager.conf(5)** man page - Descreve o arquivo de configuração **NetworkManager**.
- **/usr/share/doc/initscripts/sysconfig.txt** - Descreve **ifcfg** arquivos de configuração e suas diretrizes conforme entendidas pelo serviço de rede legado.
- **ifcfg(8)** man page - Descreve brevemente o comando **ifcfg**.

CAPÍTULO 4. CONFIGURANDO O NETWORKMANAGER PARA IGNORAR CERTOS DISPOSITIVOS

Por padrão, o NetworkManager gerencia todos os dispositivos, exceto o dispositivo **lo** (loopback). Entretanto, você pode definir certos dispositivos como **unmanaged** para configurar que o NetworkManager ignore estes dispositivos. Com esta configuração, você pode gerenciar manualmente estes dispositivos, por exemplo, usando um script.

4.1. CONFIGURAÇÃO PERMANENTE DE UM DISPOSITIVO COMO NÃO GERENCIADO NO NETWORKMANAGER

Você pode configurar dispositivos como **unmanaged** com base em vários critérios, como o nome da interface, endereço MAC ou tipo de dispositivo. Este procedimento descreve como configurar permanentemente a interface **enp1s0** como **unmanaged** no NetworkManager.

Para configurar temporariamente os dispositivos de rede como **unmanaged**, ver Seção 4.2, “Configuração temporária de um dispositivo como não gerenciado no NetworkManager”.

Procedimento

1. Opcional: Mostrar a lista de dispositivos para identificar o dispositivo que você deseja definir como **unmanaged**:

```
# nmcli device status
DEVICE  TYPE      STATE         CONNECTION
enp1s0  ethernet  disconnected  --
...
```

2. Crie o arquivo **/etc/NetworkManager/conf.d/99-unmanaged-devices.conf** com o seguinte conteúdo:

```
[keyfile]
unmanaged-devices=interface-name:enp1s0
```

   Para definir vários dispositivos como não gerenciados, separe as entradas no parâmetro **unmanaged-devices** com ponto-e-vírgula:

```
[keyfile]
unmanaged-devices=interface-name:interface_1;interface-name:interface_2;...
```

3. Recarregue o serviço **NetworkManager**:

```
# systemctl reload NetworkManager
```

Etapas de verificação

- Exibir a lista de dispositivos:

```
# nmcli device status
DEVICE  TYPE      STATE      CONNECTION
enp1s0  ethernet  unmanaged  --
...
```

O estado **unmanaged** ao lado do dispositivo **enp1s0** indica que o NetworkManager não gerencia este dispositivo.

Recursos adicionais

- Para uma lista de critérios que você pode usar para configurar dispositivos como não gerenciados e a sintaxe correspondente, consulte a seção **Device List Format** na página de manual **NetworkManager.conf(5)**.

4.2. CONFIGURAÇÃO TEMPORÁRIA DE UM DISPOSITIVO COMO NÃO GERENCIADO NO NETWORKMANAGER

Você pode configurar dispositivos como **unmanaged** com base em vários critérios, como o nome da interface, endereço MAC ou tipo de dispositivo. Este procedimento descreve como configurar temporariamente a interface **enp1s0** como **unmanaged** no NetworkManager.

Use este método, por exemplo, para fins de teste. Para configurar permanentemente os dispositivos de rede como **unmanaged**, ver Seção 4.1, "Configuração permanente de um dispositivo como não gerenciado no NetworkManager".

Use este método, por exemplo, para fins de teste. Para configurar permanentemente os dispositivos de rede como **unmanaged**, consulte a seção NetworkManager (Gerenciador de rede) na documentação **Configuring and managing networking**.

Procedimento

1. Opcional: Mostrar a lista de dispositivos para identificar o dispositivo que você deseja definir como **unmanaged**:

```
# nmcli device status
DEVICE  TYPE      STATE         CONNECTION
enp1s0  ethernet  disconnected  --
...
```

2. Configure o dispositivo **enp1s0** para o estado **unmanaged**:

```
# nmcli device set enp1s0 managed no
```

Etapas de verificação

- Exibir a lista de dispositivos:

```
# nmcli device status
DEVICE  TYPE      STATE      CONNECTION
enp1s0  ethernet  unmanaged  --
...
```

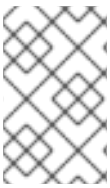
O estado **unmanaged** ao lado do dispositivo **enp1s0** indica que o NetworkManager não gerencia este dispositivo.

Recursos adicionais

- Para uma lista de critérios que você pode usar para configurar dispositivos como não gerenciados e a sintaxe correspondente, consulte a seção **Device List Format** na página de manual **NetworkManager.conf(5)**.

CAPÍTULO 5. COMEÇANDO COM NMTUI

O aplicativo **nmtui** é uma interface de usuário de texto (TUI) para **NetworkManager**. A seção a seguir fornece como você pode configurar uma interface de rede usando **nmtui**.



NOTA

O **nmtui** aplicação não suporta todos os tipos de conexão. Em particular, você não pode adicionar ou modificar conexões VPN ou conexões Ethernet que requerem autenticação 802.1X.

5.1. INICIANDO A UTILIDADE NMTUI

Este procedimento descreve como iniciar a interface de usuário de texto do NetworkManager, **nmtui**.

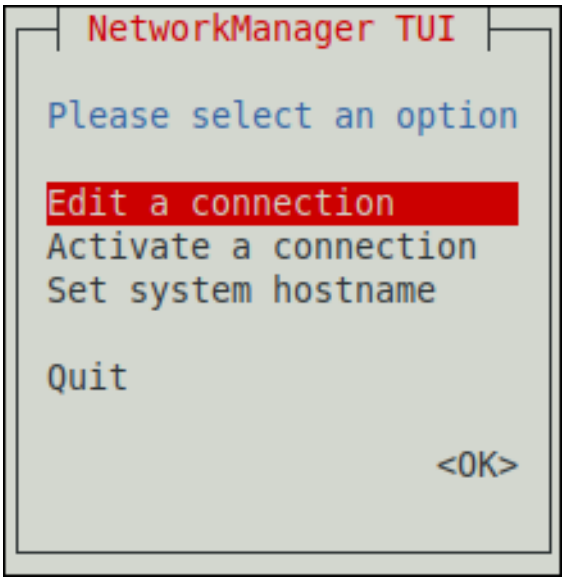
Pré-requisitos

- O pacote **NetworkManager-tui** está instalado.

Procedimento

1. Para iniciar **nmtui**, entre:

```
# nmtui
```



2. Para navegar:
   - Use os cursores ou pressione **Tab** para dar um passo à frente e pressione **Turno**+**Tab** para retroceder através das opções.
   - Use **Enter** para selecionar uma opção.
   - Use a barra de **espaço** para alternar o status das caixas de seleção.

5.2. ADICIONANDO UM PERFIL DE CONEXÃO USANDO NMTUI

O aplicativo **nmtui** fornece uma interface de usuário de texto para o NetworkManager. Este procedimento descreve como adicionar um novo perfil de conexão.

Pré-requisitos

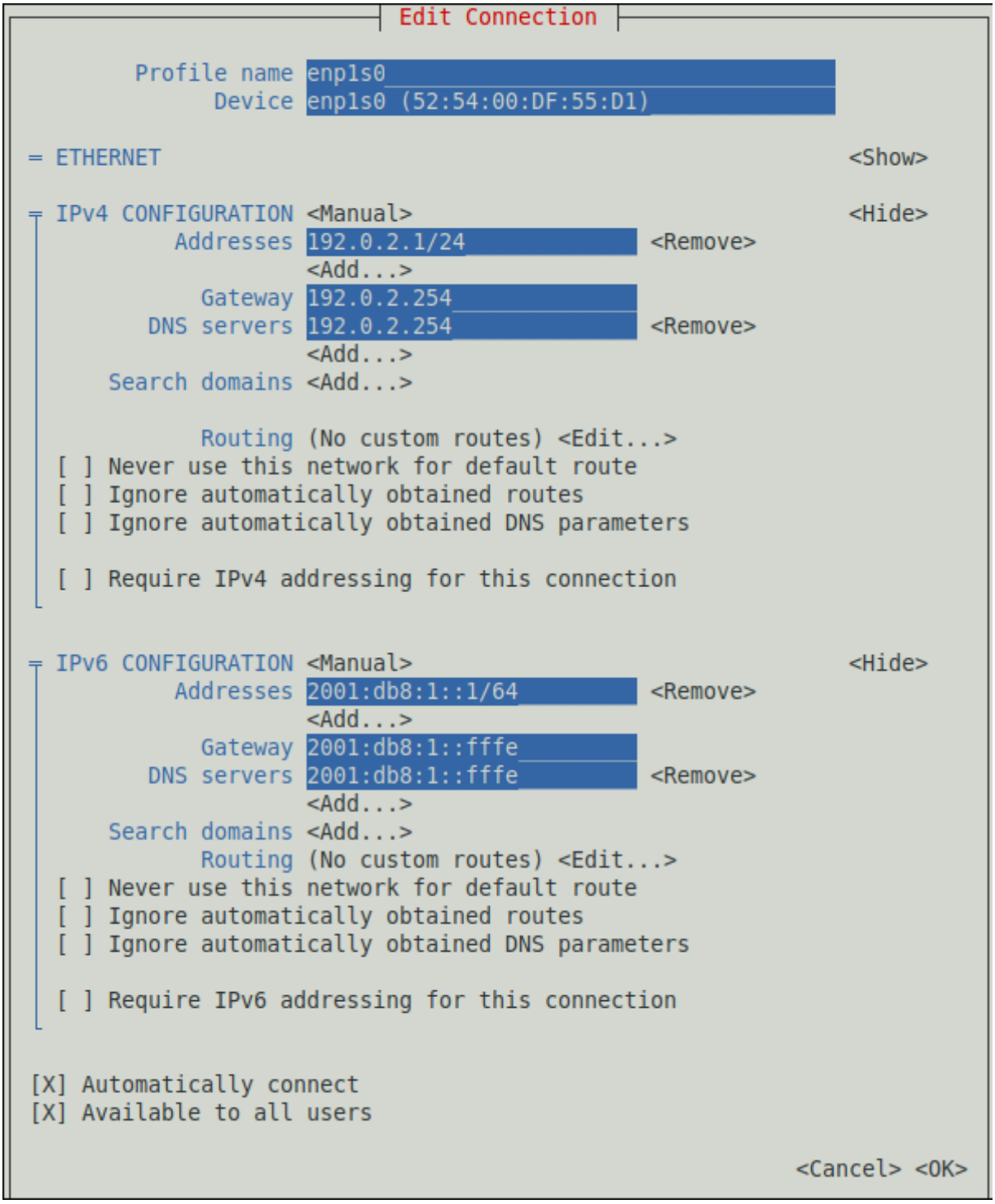
- O pacote **NetworkManager-tui** está instalado.

Procedimento

1. Inicie o utilitário de interface de usuário de texto NetworkManager:

```
# nmtui
```

2. Selecione a entrada do menu **Edit a connection**, e pressione **Enter**.
3. Selecione o botão **Adicionar**, e pressione **Enter**.
4. Selecione **Ethernet**, e pressione **Enter**.
5. Preencha os campos com os detalhes da conexão.



6. Selecione **OK** para salvar as mudanças.
7. Selecione **Back** para retornar ao menu principal.
8. Selecione **Activate a connection**, e pressione **Enter**.
9. Selecione a nova entrada de conexão, e pressione **Enter** para ativar a conexão.
10. Selecione **Voltar** para retornar ao menu principal.
11. Selecione **Quit**.

Etapas de verificação

1. Mostrar o status dos dispositivos e conexões:

```
# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
enp1s0      ethernet  connected  Example-Connection
```

2. Para exibir todas as configurações do perfil de conexão:

```
# nmcli connection show Example-Connection
connection.id:                Example-Connection
connection.uuid:              b6cdfa1c-e4ad-46e5-af8b-a75f06b79f76
connection.stable-id:         --
connection.type:              802-3-ethernet
connection.interface-name:    enp1s0
...
```

Recursos adicionais

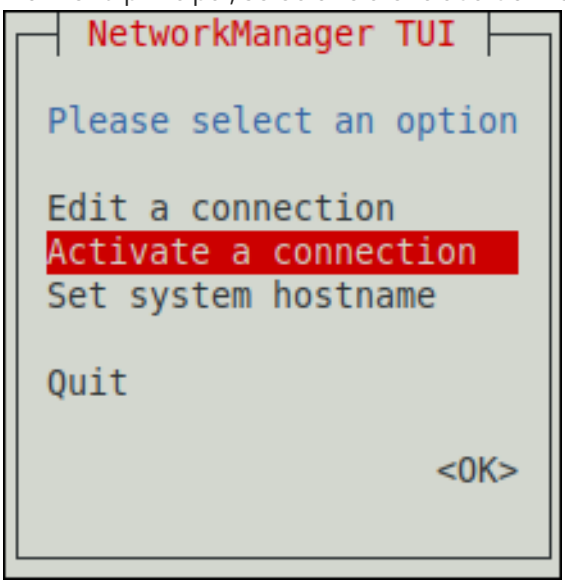
- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Para mais detalhes sobre a aplicação **nmtui**, consulte a página de manual **nmtui(1)**.
- Se a configuração no disco não corresponder à configuração no dispositivo, iniciar ou reiniciar o NetworkManager cria uma conexão in-memory que reflete a configuração do dispositivo. Para maiores detalhes e como evitar este problema, veja NetworkManager duplica uma conexão após o reinício do serviço NetworkManager.

5.3. APLICANDO MUDANÇAS EM UMA CONEXÃO MODIFICADA USANDO NMTUI

Depois de modificar uma conexão em **nmtui**, você deve reativar a conexão. Note que reativar uma conexão em **nmtui** desativa temporariamente a conexão.

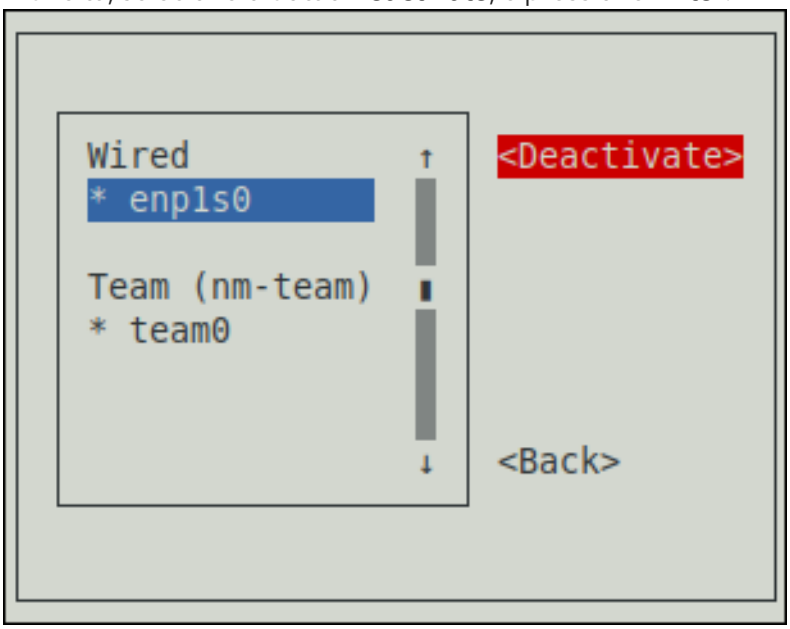
Procedimento

1. No menu principal, selecione a entrada do menu **Activate a connection**:



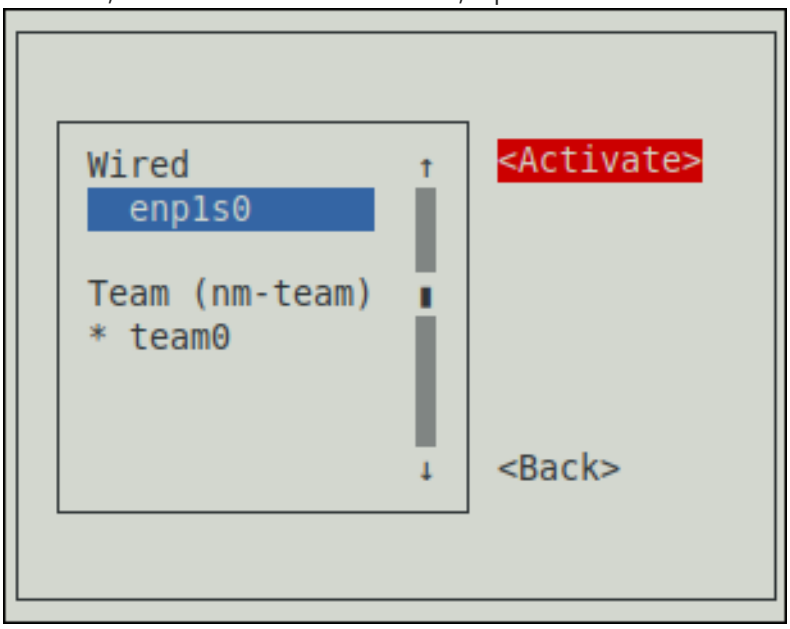
2. Selecione a conexão modificada.

3. À direita, selecione o botão **Deactivate**, e pressione **Enter**:



4. Selecione a conexão novamente.

5. À direita, selecione o botão **Activate**, e pressione **Enter**:



CAPÍTULO 6. COMEÇANDO COM NMCLI

Esta seção descreve informações gerais sobre a utilidade **nmcli**.

6.1. OS DIFERENTES FORMATOS DE SAÍDA DE NMCLI

O utilitário **nmcli** suporta diferentes opções para modificar a saída dos comandos **nmcli**. Usando estas opções, você pode exibir apenas as informações necessárias. Isto simplifica o processamento da saída em scripts.

Por padrão, o utilitário **nmcli** exibe sua saída em formato de tabela:

```
# nmcli device
DEVICE  TYPE      STATE      CONNECTION
enp1s0  ethernet  connected  enp1s0
lo      loopback  unmanaged  --
```

Usando a opção **-f**, você pode exibir colunas específicas em uma ordem personalizada. Por exemplo, para exibir apenas a coluna **DEVICE** e **STATE**, digite:

```
# nmcli -f DEVICE,STATE device
DEVICE  STATE
enp1s0  connected
lo      unmanaged
```

A opção **-t** permite exibir os campos individuais da saída em um formato separado por dois pontos:

```
# nmcli -t device
enp1s0:ethernet:connected:enp1s0
lo:loopback:unmanaged:
```

A combinação do **-f** e **-t** para exibir apenas campos específicos em formato de dois pontos pode ser útil quando se processa a saída em scripts:

```
# nmcli -f DEVICE,STATE -t device
enp1s0:connected
lo:unmanaged
```

6.2. USANDO PREENCHIMENTO DE TABULAÇÕES EM NMCLI

Se o pacote **bash-completion** estiver instalado em seu host, o utilitário **nmcli** suporta o preenchimento de guias. Isto permite que você complete automaticamente os nomes das opções e identifique possíveis opções e valores.

Por exemplo, se você digitar **nmcli con** e pressionar **Tab**, então a concha completa automaticamente o comando para **nmcli connection**.

Para a conclusão, as opções ou valor que você digitou devem ser únicos. Se não for único, então **nmcli** exibe todas as possibilidades. Por exemplo, se você digitar **nmcli connection d** e pressionar **Tab**, então o comando mostra o comando **delete** e **down** como opções possíveis.

Você também pode usar o preenchimento da aba para exibir todas as propriedades que você pode definir em um perfil de conexão. Por exemplo, se você digitar **nmcli connection modify *connection_name*** e pressione **Tab**, o comando mostra a lista completa das propriedades disponíveis.

6.3. COMANDOS NMCLI FREQÜENTES

A seguir, uma visão geral sobre os comandos **nmcli** freqüentemente utilizados.

- Para exibir os perfis de conexão da lista, entre:

```
# nmcli connection show
NAME    UUID                                  TYPE      DEVICE
enp1s0  45224a39-606f-4bf7-b3dc-d088236c15ee  ethernet  enp1s0
```

- Para exibir as configurações de um perfil de conexão específico, entre:

```
# nmcli connection show connection_name
connection.id:            enp1s0
connection.uuid:          45224a39-606f-4bf7-b3dc-d088236c15ee
connection.stable-id:     --
connection.type:          802-3-ethernet
...
```

- Para modificar as propriedades de uma conexão, entre:

```
# nmcli connection modify connection_name property value
```

  Você pode modificar várias propriedades usando um único comando se você passar vários ***property value*** combinações para o comando.

- Para exibir a lista de dispositivos de rede, seu estado, e quais perfis de conexão utilizam o dispositivo, entre:

```
# nmcli device
DEVICE  TYPE      STATE         CONNECTION
enp1s0  ethernet  connected     enp1s0
enp8s0  ethernet  disconnected  --
enp7s0  ethernet  unmanaged     --
...
```

- Para ativar uma conexão, entre:

```
# nmcli connection up connection_name
```

- Para desativar uma conexão, entre:

```
# nmcli connection down connection_name
```

CAPÍTULO 7. COMEÇANDO COM A CONFIGURAÇÃO DE REDES USANDO A GUI GNOME

Você pode gerenciar e configurar conexões de rede usando as seguintes maneiras no GNOME:

- o ícone de conexão de rede do GNOME Shell na parte superior direita da área de trabalho
- o GNOME **control-center** aplicação
- o GNOME **nm-connection-editor** aplicação

7.1. CONECTANDO-SE A UMA REDE USANDO O ÍCONE DE CONEXÃO DE REDE DO GNOME SHELL

Se você usar a GUI GNOME, você pode usar o ícone de conexão de rede do GNOME Shell para se conectar a uma rede.

Pré-requisitos

- O grupo de pacotes **GNOME** está instalado.
- Você está logado no GNOME.
- Se a rede requer uma configuração específica, como um endereço IP estático ou uma configuração 802.1x, já foi criado um perfil de conexão.

Procedimento

1. Clique no ícone de conexão de rede no canto superior direito de sua área de trabalho.

2. Dependendo do tipo de conexão, selecione a entrada **Wired** ou **Wi-Fi**.

- Para uma conexão com fio, selecione **Connect** para conectar-se à rede.
- Para uma conexão Wi-Fi, clique em **Select network**, selecione a rede à qual você deseja se conectar, e digite a senha.

CAPÍTULO 8. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET

Esta seção descreve diferentes maneiras de configurar uma conexão Ethernet com endereços IP estáticos e dinâmicos.

8.1. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET ESTÁTICA USANDO NMCLI

Este procedimento descreve a adição de uma conexão Ethernet com as seguintes configurações usando o utilitário **nmcli**:

- Um endereço IPv4 estático - **192.0.2.1** com uma máscara de sub-rede **/24**
- Um endereço IPv6 estático - **2001:db8:1::1** com uma máscara de sub-rede **/64**
- Um gateway padrão IPv4 - **192.0.2.254**
- Um gateway padrão IPv6 - **2001:db8:1::fffe**
- Um servidor DNS IPv4 - **192.0.2.200**
- Um servidor DNS IPv6 - **2001:db8:1::ffbb**
- Um domínio de busca DNS - **example.com**

Procedimento

1. Adicione um novo perfil de conexão NetworkManager para a conexão Ethernet:

   ```
   # nmcli connection add con-name Example-Connection ifname enp7s0 type ethernet
   ```

   Os próximos passos modificam o perfil de conexão **Example-Connection** que você criou.

2. Defina o endereço IPv4:

   ```
   # nmcli connection modify Example-Connection ipv4.addresses 192.0.2.1/24
   ```

3. Defina o endereço IPv6:

   ```
   # nmcli connection modify Example-Connection ipv6.addresses 2001:db8:1::1/64
   ```

4. Configure o método de conexão IPv4 e IPv6 para **manual**:

   ```
   # nmcli connection modify Example-Connection ipv4.method manual
   # nmcli connection modify Example-Connection ipv6.method manual
   ```

5. Defina os gateways padrão IPv4 e IPv6:

   ```
   # nmcli connection modify Example-Connection ipv4.gateway 192.0.2.254
   # nmcli connection modify Example-Connection ipv6.gateway 2001:db8:1::fffe
   ```

6. Configure os endereços dos servidores DNS IPv4 e IPv6:

```
# nmcli connection modify Example-Connection ipv4.dns "192.0.2.200"
# nmcli connection modify Example-Connection ipv6.dns "2001:db8:1::ffbb"
```

Para definir vários servidores DNS, especifique-os separados por espaço e entre aspas.

7. Definir o domínio de busca DNS para a conexão IPv4 e IPv6:

```
# nmcli connection modify Example-Connection ipv4.dns-search example.com
# nmcli connection modify Example-Connection ipv6.dns-search example.com
```

8. Ativar o perfil de conexão:

```
# nmcli connection up Example-Connection
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/13)
```

Etapas de verificação

1. Mostrar o status dos dispositivos e conexões:

```
# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
enp7s0      ethernet  connected  Example-Connection
```

2. Para exibir todas as configurações do perfil de conexão:

```
# nmcli connection show Example-Connection
connection.id:              Example-Connection
connection.uuid:            b6cdfa1c-e4ad-46e5-af8b-a75f06b79f76
connection.stable-id:       --
connection.type:            802-3-ethernet
connection.interface-name:  enp7s0
...
```

3. Use o utilitário **ping** para verificar se este host pode enviar pacotes para outros hosts.

- Pingar um endereço IP na mesma sub-rede.
  Para IPv4:

  ```
  # ping 192.0.2.3
  ```

  Para IPv6:

  ```
  # ping 2001:db8:2::1
  ```

  Se o comando falhar, verificar as configurações de IP e subrede.

- Pingar um endereço IP em uma sub-rede remota.
  Para IPv4:

  ```
  # ping 198.162.3.1
  ```

  Para IPv6:

```
# ping 2001:db8:2::1
```

- Se o comando falhar, pingar o gateway padrão para verificar as configurações.
  Para IPv4:

  ```
  # ping 192.0.2.254
  ```

  Para IPv6:

  ```
  # ping 2001:db8:1::fffe
  ```

4. Use o utilitário **host** para verificar se a resolução do nome funciona. Por exemplo:

   ```
   # host client.example.com
   ```

   Se o comando retornar algum erro, como **connection timed out** ou **no servers could be reached**, verifique suas configurações de DNS.

Passos para a solução de problemas

1. Se a conexão falhar ou se a interface de rede comutar entre um estado para cima e para baixo:
   - Certifique-se de que o cabo de rede esteja conectado ao host e a um switch.
   - Verifique se a falha do link só existe neste host ou também em outros hosts conectados ao mesmo switch ao qual o servidor está conectado.
   - Verificar se o cabo de rede e a interface de rede estão funcionando como esperado. Executar as etapas de diagnóstico do hardware e substituir os cabos de defeito e as placas de interface de rede.

Recursos adicionais

- Consulte a página de manual **nm-settings(5)** para mais informações sobre as propriedades do perfil de conexão e suas configurações.
- Para mais detalhes sobre a utilidade **nmcli**, consulte a página de manual **nmcli(1)**.
- Se a configuração no disco não corresponder à configuração no dispositivo, iniciar ou reiniciar o NetworkManager cria uma conexão in-memory que reflete a configuração do dispositivo. Para maiores detalhes e como evitar este problema, veja NetworkManager duplica uma conexão após o reinício do serviço NetworkManager.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, “Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão”.

8.2. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET ESTÁTICA USANDO O EDITOR INTERATIVO NMCLI

Este procedimento descreve a adição de uma conexão Ethernet com as seguintes configurações usando o modo interativo **nmcli**:

- Um endereço IPv4 estático - **192.0.2.1** com uma máscara de sub-rede **/24**

- Um endereço IPv6 estático - **2001:db8:1::1** com uma máscara de sub-rede **/64**
- Um gateway padrão IPv4 - **192.0.2.254**
- Um gateway padrão IPv6 - **2001:db8:1::fffe**
- Um servidor DNS IPv4 - **192.0.2.200**
- Um servidor DNS IPv6 - **2001:db8:1::ffbb**
- Um domínio de busca DNS - **example.com**

Procedimento

1. Para adicionar um novo perfil de conexão NetworkManager para a conexão Ethernet, e iniciar o modo interativo, entre:

```
# nmcli connection edit type ethernet con-name Example-Connection
```

2. Defina a interface de rede:

```
nmcli> set connection.interface-name enp7s0
```

3. Defina o endereço IPv4:

```
nmcli> set ipv4.addresses 192.0.2.1/24
```

4. Defina o endereço IPv6:

```
nmcli> set ipv6.addresses 2001:db8:1::1/64
```

5. Configure o método de conexão IPv4 e IPv6 para **manual**:

```
nmcli> set ipv4.method manual
nmcli> set ipv6.method manual
```

6. Defina os gateways padrão IPv4 e IPv6:

```
nmcli> set ipv4.gateway 192.0.2.254
nmcli> set ipv6.gateway 2001:db8:1::fffe
```

7. Configure os endereços dos servidores DNS IPv4 e IPv6:

```
nmcli> set ipv4.dns 192.0.2.200
nmcli> set ipv6.dns 2001:db8:1::ffbb
```

   Para definir vários servidores DNS, especifique-os separados por espaço e entre aspas.

8. Definir o domínio de busca DNS para a conexão IPv4 e IPv6:

```
nmcli> set ipv4.dns-search example.com
nmcli> set ipv6.dns-search example.com
```

9. Salvar e ativar a conexão:

```
nmcli> save persistent
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation of the connection.
Do you still want to save? (yes/no) [yes] yes
```

10. Abandonar o modo interativo:

```
nmcli> quit
```

Etapas de verificação

1. Mostrar o status dos dispositivos e conexões:

```
# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
enp7s0      ethernet  connected  Example-Connection
```

2. Para exibir todas as configurações do perfil de conexão:

```
# nmcli connection show Example-Connection
connection.id:                Example-Connection
connection.uuid:              b6cdfa1c-e4ad-46e5-af8b-a75f06b79f76
connection.stable-id:         --
connection.type:              802-3-ethernet
connection.interface-name:    enp7s0
...
```

3. Use o utilitário **ping** para verificar se este host pode enviar pacotes para outros hosts.

- Pingar um endereço IP na mesma sub-rede.
  Para IPv4:

```
# ping 192.0.2.3
```

  Para IPv6:

```
# ping 2001:db8:2::1
```

  Se o comando falhar, verificar as configurações de IP e subrede.

- Pingar um endereço IP em uma sub-rede remota.
  Para IPv4:

```
# ping 198.162.3.1
```

  Para IPv6:

```
# ping 2001:db8:2::1
```

  - Se o comando falhar, pingar o gateway padrão para verificar as configurações.

Para IPv4:

```
# ping 192.0.2.254
```

Para IPv6:

```
# ping 2001:db8:1::fffe
```

4. Use o utilitário **host** para verificar se a resolução do nome funciona. Por exemplo:

```
# host client.example.com
```

Se o comando retornar algum erro, como **connection timed out** ou **no servers could be reached**, verifique suas configurações de DNS.

Passos para a solução de problemas

1. Se a conexão falhar ou se a interface de rede comutar entre um estado para cima e para baixo:
   - Certifique-se de que o cabo de rede esteja conectado ao host e a um switch.
   - Verifique se a falha do link só existe neste host ou também em outros hosts conectados ao mesmo switch ao qual o servidor está conectado.
   - Verificar se o cabo de rede e a interface de rede estão funcionando como esperado. Executar as etapas de diagnóstico do hardware e substituir os cabos de defeito e as placas de interface de rede.

Recursos adicionais

- Consulte a página de manual **nm-settings(5)** para mais informações sobre as propriedades do perfil de conexão e suas configurações.
- Para mais detalhes sobre a utilidade **nmcli**, consulte a página de manual **nmcli(1)**.
- Se a configuração no disco não corresponder à configuração no dispositivo, iniciar ou reiniciar o NetworkManager cria uma conexão in-memory que reflete a configuração do dispositivo. Para maiores detalhes e como evitar este problema, veja NetworkManager duplica uma conexão após o reinício do serviço NetworkManager.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, “Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão”.

8.3. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET ESTÁTICA USANDO AS FUNÇÕES DO SISTEMA RHEL

Este procedimento descreve como usar as funções do Sistema RHEL para adicionar remotamente uma conexão Ethernet para a interface **enp7s0** com as seguintes configurações, executando um livro de exercícios possível:

- Um endereço IPv4 estático - **192.0.2.1** com uma máscara de sub-rede **/24**
- Um endereço IPv6 estático - **2001:db8:1::1** com uma máscara de sub-rede **/64**

- Um gateway padrão IPv4 - **192.0.2.254**
- Um gateway padrão IPv6 - **2001:db8:1::fffe**
- Um servidor DNS IPv4 - **192.0.2.200**
- Um servidor DNS IPv6 - **2001:db8:1::ffbb**
- Um domínio de busca DNS - **example.com**

Execute este procedimento no Nó de controle possível.

Pré-requisitos

- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.
- Se você usar um usuário remoto diferente de **root** ao executar o playbook, este usuário tem as permissões apropriadas **sudo** no nó gerenciado.
- O anfitrião usa o NetworkManager para configurar a rede.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo /**etc**/**ansible**/**hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook ~/**ethernet-static-IP.yml** com o seguinte conteúdo:

```
---
- name: Configure an Ethernet connection with static IP
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network

    vars:
      network_connections:
        - name: enp7s0
          type: ethernet
          autoconnect: yes
          ip:
            address:
              - 192.0.2.1/24
              - 2001:db8:1::1/64
            gateway4: 192.0.2.254
            gateway6: 2001:db8:1::fffe
            dns:
              - 192.0.2.200
              - 2001:db8:1::ffbb
            dns_search:
              - example.com
          state: up
```

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

     ```
     # ansible-playbook -u root ~/ethernet-static-IP.yml
     ```

   - Para conectar-se como usuário ao host administrado, entre:

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/ethernet-static-IP.yml
     ```

     A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definido no **-u *user_name*** opção.

   Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Recursos adicionais

- Para detalhes sobre os parâmetros usados em **network_connections** e para informações adicionais sobre o Sistema de Papel **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

8.4. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET DINÂMICA USANDO NMCLI

Este procedimento descreve a adição de uma conexão Ethernet dinâmica usando o utilitário **nmcli**. Com esta configuração, o NetworkManager solicita as configurações IP para esta conexão a partir de um servidor DHCP.

Pré-requisitos

- Um servidor DHCP está disponível na rede.

Procedimento

1. Adicione um novo perfil de conexão NetworkManager para a conexão Ethernet:

   ```
   # nmcli connection add con-name Example-Connection ifname enp7s0 type ethernet
   ```

2. Opcionalmente, mude o nome do host que o NetworkManager envia ao servidor DHCP ao usar o perfil **Example-Connection**:

   ```
   # nmcli connection modify Example-Connection ipv4.dhcp-hostname Example
   ipv6.dhcp-hostname Example
   ```

3. Opcionalmente, mude o NetworkManager de ID de cliente enviado para um servidor DHCP IPv4 ao usar o perfil **Example-Connection**:

   ```
   # nmcli connection modify Example-Connection ipv4.dhcp-client-id client-ID
   ```

Note que não há parâmetro **dhcp-client-id** para IPv6. Para criar um identificador para IPv6, configure o serviço **dhclient**.

Etapas de verificação

1. Mostrar o status dos dispositivos e conexões:

```
# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
enp7s0      ethernet  connected  Example-Connection
```

2. Para exibir todas as configurações do perfil de conexão:

```
# nmcli connection show Example-Connection
connection.id:                  Example-Connection
connection.uuid:                b6cdfa1c-e4ad-46e5-af8b-a75f06b79f76
connection.stable-id:           --
connection.type:                802-3-ethernet
connection.interface-name:      enp7s0
...
```

3. Use o utilitário **ping** para verificar se este host pode enviar pacotes para outros hosts.

- Pingar um endereço IP na mesma sub-rede.
  Para IPv4:

```
# ping 192.0.2.3
```

  Para IPv6:

```
# ping 2001:db8:2::1
```

  Se o comando falhar, verificar as configurações de IP e subrede.

- Pingar um endereço IP em uma sub-rede remota.
  Para IPv4:

```
# ping 198.162.3.1
```

  Para IPv6:

```
# ping 2001:db8:2::1
```

  - Se o comando falhar, pingar o gateway padrão para verificar as configurações.
    Para IPv4:

```
# ping 192.0.2.254
```

    Para IPv6:

```
# ping 2001:db8:1::fffe
```

4. Use o utilitário **host** para verificar se a resolução do nome funciona. Por exemplo:

```
# host client.example.com
```

Se o comando retornar algum erro, como **connection timed out** ou **no servers could be reached**, verifique suas configurações de DNS.

Recursos adicionais

- Para detalhes sobre a definição de um identificador de cliente para IPv6, consulte a página de manual **dhclient(8)**.
- Consulte a página de manual **nm-settings(5)** para mais informações sobre as propriedades do perfil de conexão e suas configurações.
- Para mais detalhes sobre a utilidade **nmcli**, consulte a página de manual **nmcli(1)**.
- Se a configuração no disco não corresponder à configuração no dispositivo, iniciar ou reiniciar o NetworkManager cria uma conexão in-memory que reflete a configuração do dispositivo. Para maiores detalhes e como evitar este problema, veja NetworkManager duplica uma conexão após o reinício do serviço NetworkManager.

8.5. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET DINÂMICA USANDO O EDITOR INTERATIVO NMCLI

Este procedimento descreve a adição de uma conexão Ethernet dinâmica usando o editor interativo do utilitário **nmcli**. Com esta configuração, o NetworkManager solicita as configurações IP para esta conexão a partir de um servidor DHCP.

Pré-requisitos

- Um servidor DHCP está disponível na rede.

Procedimento

1. Para adicionar um novo perfil de conexão NetworkManager para a conexão Ethernet, e iniciar o modo interativo, entre:

```
# nmcli connection edit type ethernet con-name Example-Connection
```

2. Defina a interface de rede:

```
nmcli> set connection.interface-name enp7s0
```

3. Opcionalmente, mude o nome do host que o NetworkManager envia ao servidor DHCP ao usar o perfil **Example-Connection**:

```
nmcli> set ipv4.dhcp-hostname Example
nmcli> set ipv6.dhcp-hostname Example
```

4. Opcionalmente, mude o NetworkManager de ID de cliente enviado para um servidor DHCP IPv4 ao usar o perfil **Example-Connection**:

```
nmcli> set ipv4.dhcp-client-id client-ID
```

Note que não há parâmetro **dhcp-client-id** para IPv6. Para criar um identificador para IPv6, configure o serviço **dhclient**.

5. Salvar e ativar a conexão:

```
nmcli> save persistent
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation of the connection.
Do you still want to save? (yes/no) [yes] yes
```

6. Abandonar o modo interativo:

```
nmcli> quit
```

Etapas de verificação

1. Mostrar o status dos dispositivos e conexões:

```
# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
enp7s0      ethernet  connected  Example-Connection
```

2. Para exibir todas as configurações do perfil de conexão:

```
# nmcli connection show Example-Connection
connection.id:                Example-Connection
connection.uuid:              b6cdfa1c-e4ad-46e5-af8b-a75f06b79f76
connection.stable-id:         --
connection.type:              802-3-ethernet
connection.interface-name:    enp7s0
...
```

3. Use o utilitário **ping** para verificar se este host pode enviar pacotes para outros hosts.

- Pingar um endereço IP na mesma sub-rede.
  Para IPv4:

```
# ping 192.0.2.3
```

  Para IPv6:

```
# ping 2001:db8:2::1
```

  Se o comando falhar, verificar as configurações de IP e subrede.

- Pingar um endereço IP em uma sub-rede remota.
  Para IPv4:

```
# ping 198.162.3.1
```

  Para IPv6:

```
# ping 2001:db8:2::1
```

- Se o comando falhar, pingar o gateway padrão para verificar as configurações.
  Para IPv4:

  ```
  # ping 192.0.2.254
  ```

  Para IPv6:

  ```
  # ping 2001:db8:1::fffe
  ```

4. Use o utilitário **host** para verificar se a resolução do nome funciona. Por exemplo:

   ```
   # host client.example.com
   ```

   Se o comando retornar algum erro, como **connection timed out** ou **no servers could be reached**, verifique suas configurações de DNS.

Recursos adicionais

- Para detalhes sobre a definição de um identificador de cliente para IPv6, consulte a página de manual **dhclient(8)**.
- Consulte a página de manual **nm-settings(5)** para mais informações sobre as propriedades do perfil de conexão e suas configurações.
- Para mais detalhes sobre a utilidade **nmcli**, consulte a página de manual **nmcli(1)**.
- Se a configuração no disco não corresponder à configuração no dispositivo, iniciar ou reiniciar o NetworkManager cria uma conexão in-memory que reflete a configuração do dispositivo. Para maiores detalhes e como evitar este problema, veja NetworkManager duplica uma conexão após o reinício do serviço NetworkManager.

8.6. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET DINÂMICA USANDO AS FUNÇÕES DO SISTEMA RHEL

Este procedimento descreve como usar as funções do Sistema RHEL para adicionar remotamente uma conexão Ethernet dinâmica para a interface **enp7s0**, executando um livro de exercícios possível. Com esta configuração, a conexão de rede solicita as configurações IP para esta conexão a partir de um servidor DHCP. Execute este procedimento no nó de controle do Ansible control.

Pré-requisitos

- Um servidor DHCP está disponível na rede.
- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.
- Se você usar um usuário remoto diferente de **root** ao executar o playbook, este usuário tem as permissões apropriadas **sudo** no nó gerenciado.
- O anfitrião usa o NetworkManager para configurar a rede.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo **/etc/ansible/hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook **~/ethernet-dynamic-IP.yml** com o seguinte conteúdo:

```
---
- name: Configure an Ethernet connection with dynamic IP
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network

    vars:
      network_connections:
        - name: enp7s0
          type: ethernet
          autoconnect: yes
          ip:
            dhcp4: yes
            auto6: yes
          state: up
```

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

```
# ansible-playbook -u root ~/ethernet-dynamic-IP.yml
```

   - Para conectar-se como usuário ao host administrado, entre:

```
# ansible-playbook -u user_name --ask-become-pass ~/ethernet-dynamic-IP.yml
```

     A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definida no **-u *user_name*** opção.

   Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Recursos adicionais

- Para detalhes sobre os parâmetros usados em **network_connections** e para informações adicionais sobre o Sistema de Papel **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

8.7. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET USANDO O CENTRO DE CONTROLE

As conexões Ethernet são os tipos de conexões mais freqüentemente utilizadas em máquinas físicas ou virtuais. Esta seção descreve como configurar este tipo de conexão no GNOME **control-center**:

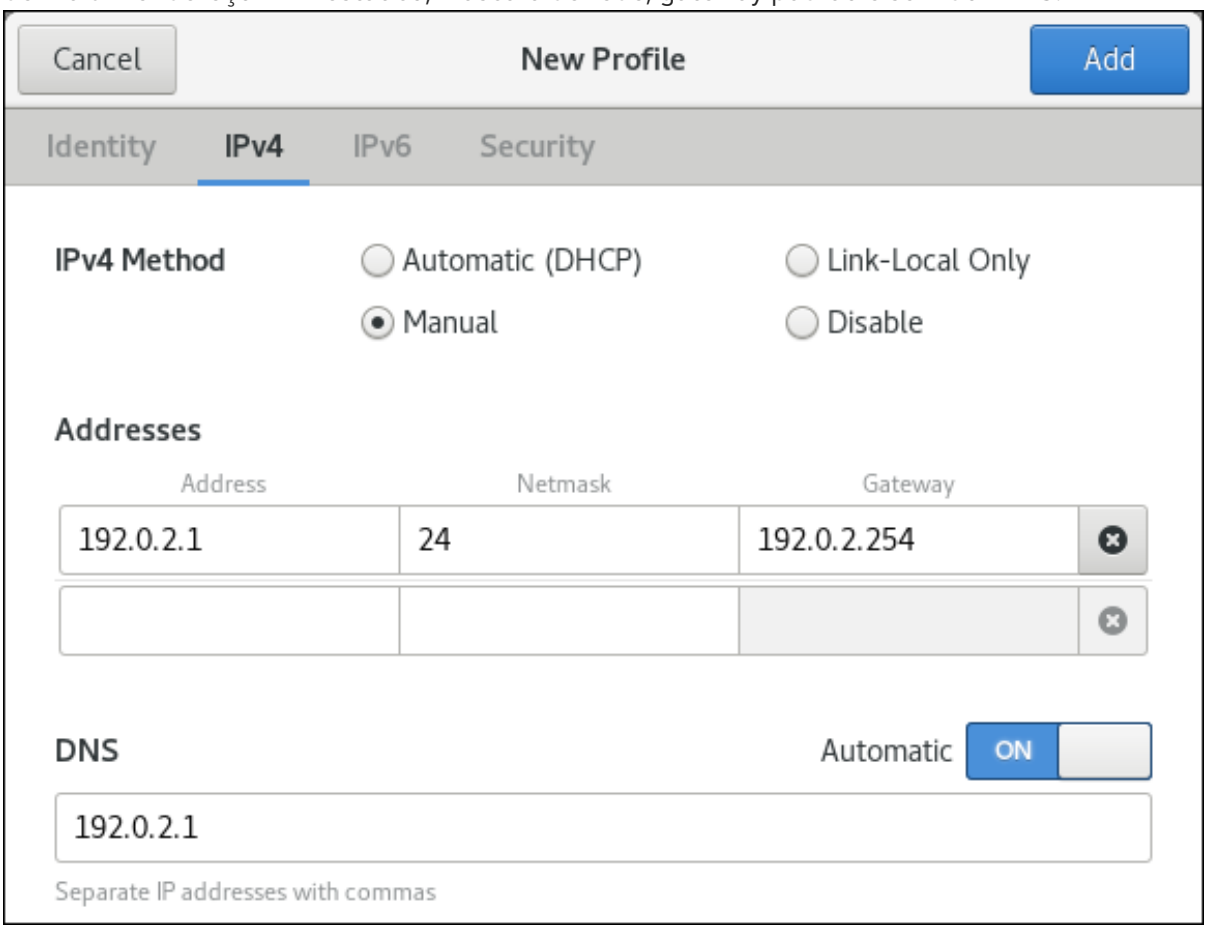
Note que **control-center** não suporta tantas opções de configuração como o aplicativo **nm-connection-editor** ou o utilitário **nmcli**.

Pré-requisitos

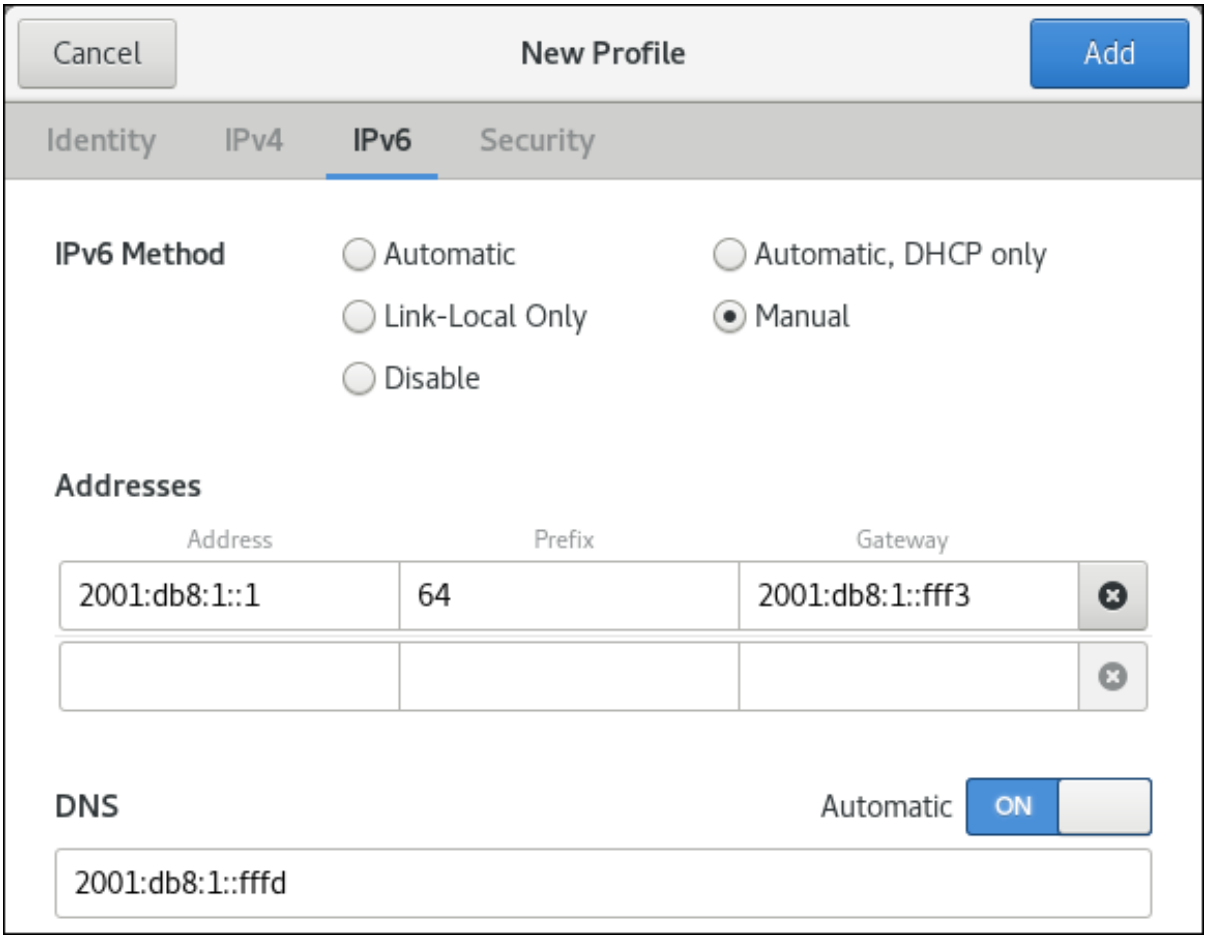
- Existe um dispositivo Ethernet físico ou virtual na configuração do servidor.
- O GNOME está instalado.

Procedimento

1. Pressione a tecla **Super**, entre **Settings**, e pressione **Enter**.
2. Selecione **Network** na navegação à esquerda.
3. Clique no botão ao lado da entrada **Wired** para criar um novo perfil.
4. Opcional: Defina um nome para a conexão na guia **Identity**.
5. Na aba **IPv4**, configure as configurações do IPv4. Por exemplo, selecione o método **Manual**, defina um endereço IPv4 estático, máscara de rede, gateway padrão e servidor DNS:



6. Na aba **IPv6**, configure as configurações IPv6. Por exemplo, selecione o método **Manual**, defina um endereço IPv6 estático, máscara de rede, gateway padrão e servidor DNS:



7. Clique no botão **Adicionar** para salvar a conexão. O GNOME **control-center** ativa automaticamente a conexão.

Etapas de verificação

1. Mostrar o status dos dispositivos e conexões:

```
# nmcli device status
DEVICE      TYPE      STATE      CONNECTION
enp7s0      ethernet  connected  Example-Connection
```

2. Para exibir todas as configurações do perfil de conexão:

```
# nmcli connection show Example-Connection
connection.id:                  Example-Connection
connection.uuid:                b6cdfa1c-e4ad-46e5-af8b-a75f06b79f76
connection.stable-id:           --
connection.type:                802-3-ethernet
connection.interface-name:      enp7s0
...
```

3. Use o utilitário **ping** para verificar se este host pode enviar pacotes para outros hosts.

- Pingar um endereço IP na mesma sub-rede.
  Para IPv4:

```
# ping 192.0.2.3
```

Para IPv6:

```
# ping 2001:db8:2::1
```

Se o comando falhar, verificar as configurações de IP e subrede.

- Pingar um endereço IP em uma sub-rede remota.
  Para IPv4:

```
# ping 198.162.3.1
```

  Para IPv6:

```
# ping 2001:db8:2::1
```

  - Se o comando falhar, pingar o gateway padrão para verificar as configurações.
    Para IPv4:

```
# ping 192.0.2.254
```

    Para IPv6:

```
# ping 2001:db8:1::fffe
```

4. Use o utilitário **host** para verificar se a resolução do nome funciona. Por exemplo:

```
# host client.example.com
```

   Se o comando retornar algum erro, como **connection timed out** ou **no servers could be reached**, verifique suas configurações de DNS.

Passos para a solução de problemas

1. Se a conexão falhar ou se a interface de rede comutar entre um estado para cima e para baixo:
   - Certifique-se de que o cabo de rede esteja conectado ao host e a um switch.
   - Verifique se a falha do link só existe neste host ou também em outros hosts conectados ao mesmo switch ao qual o servidor está conectado.
   - Verificar se o cabo de rede e a interface de rede estão funcionando como esperado. Executar as etapas de diagnóstico do hardware e substituir os cabos de defeito e as placas de interface de rede.

Recursos adicionais

- Se a conexão não tiver um gateway padrão, veja Seção 18.8, “Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão”.

8.8. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET USANDO UM EDITOR DE CONEXÃO NM

As conexões Ethernet são os tipos de conexão mais freqüentemente utilizados em servidores físicos ou virtuais. Esta seção descreve como configurar este tipo de conexão usando a aplicação **nm-connection-editor**.

Pré-requisitos

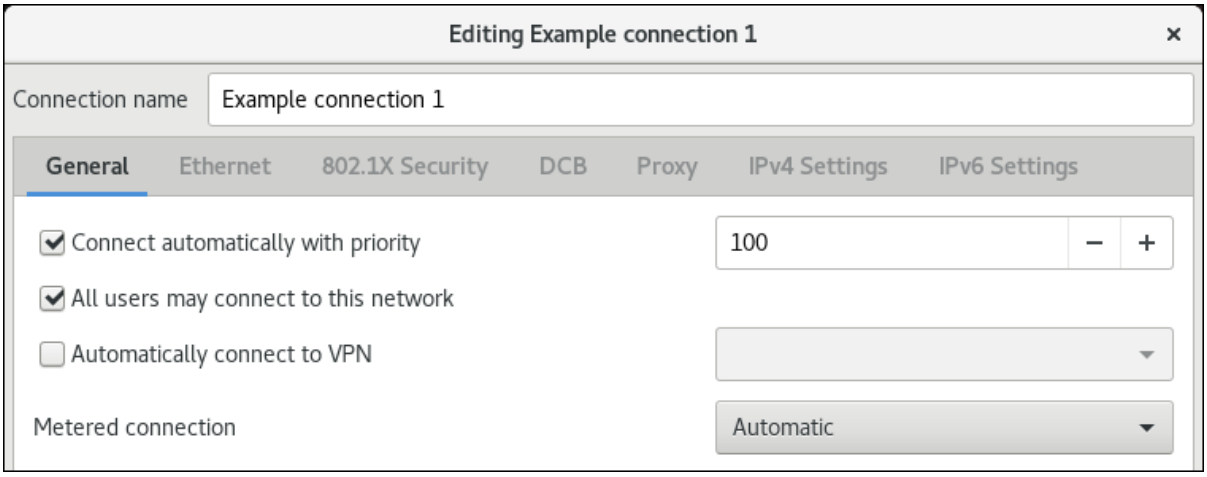
- Existe um dispositivo Ethernet físico ou virtual na configuração do servidor.
- O GNOME está instalado.

Procedimento

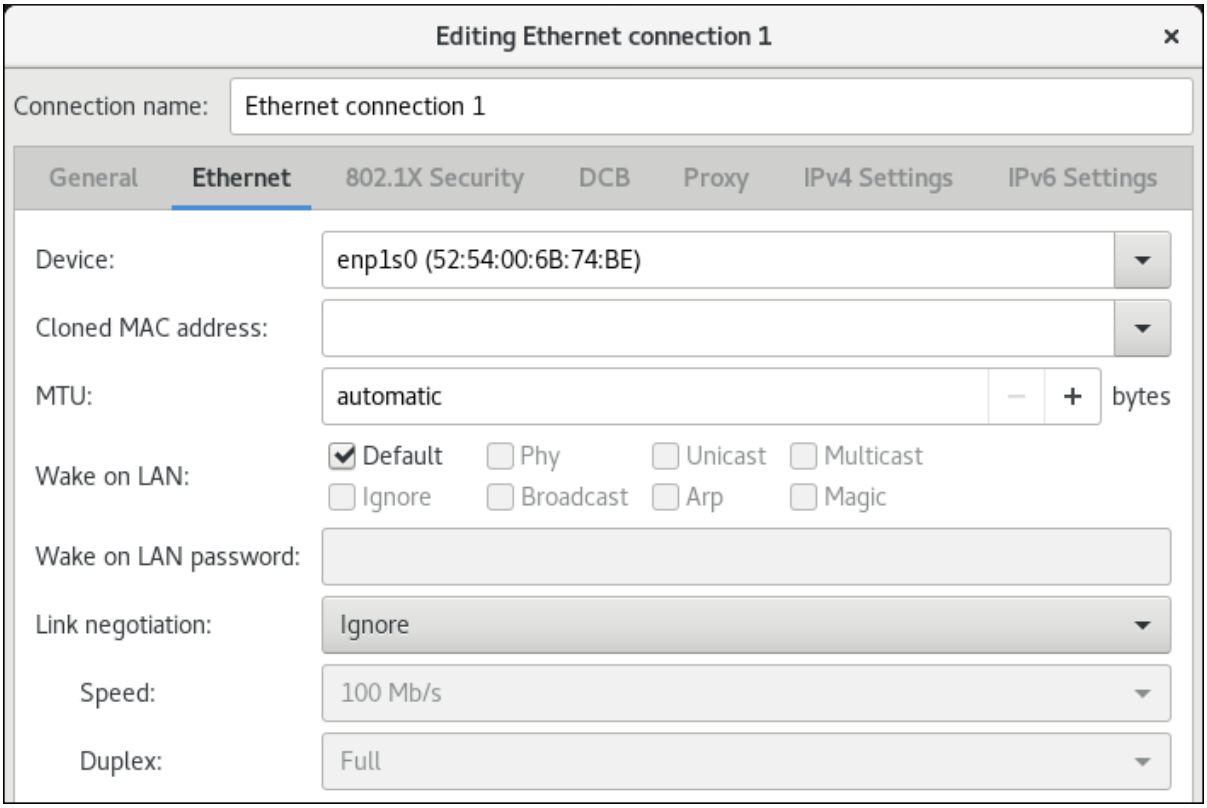
1. Abra um terminal, e entre:

```
Monitor de conexão de $ nm
```

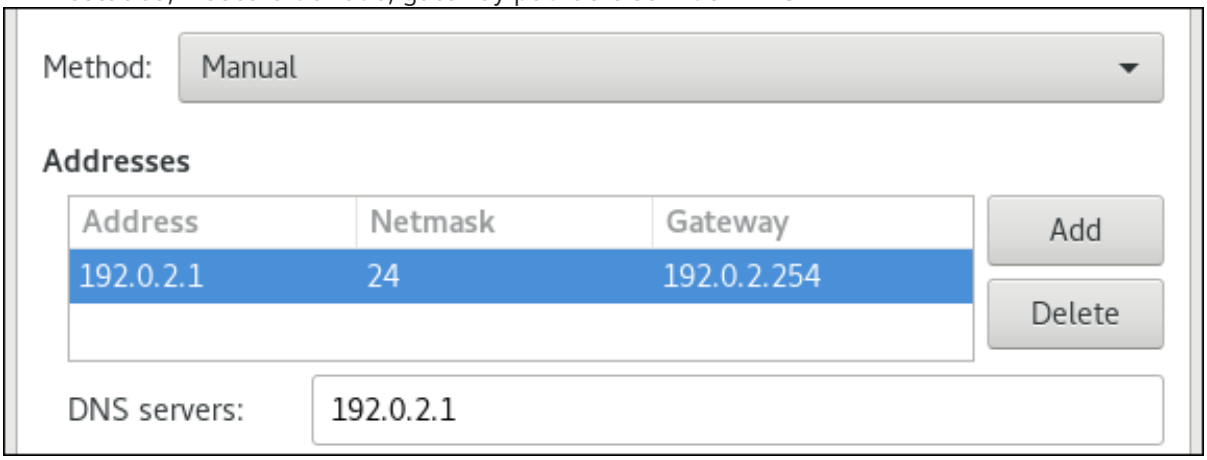
2. Clique no botão para adicionar uma nova conexão.
3. Selecione o tipo de conexão **Ethernet**, e clique em **Criar**.
4. Na aba **General**:
   a. Para ativar automaticamente esta conexão quando o sistema inicia ou quando você reinicia o serviço **NetworkManager**:
      i. Selecione **Connect automatically with priority**.
      ii. Opcional: Alterar o valor de prioridade ao lado de **Connect automatically with priority**. Se existirem vários perfis de conexão para o mesmo dispositivo, o NetworkManager permite apenas um perfil. Por padrão, o NetworkManager ativa o último perfil utilizado que tem a autoconexão ativada. Entretanto, se você definir valores de prioridade nos perfis, o NetworkManager ativa o perfil com a maior prioridade.
   b. Limpe a caixa de seleção **All users may connect to this network** se o perfil deve estar disponível somente para o usuário que criou o perfil de conexão.



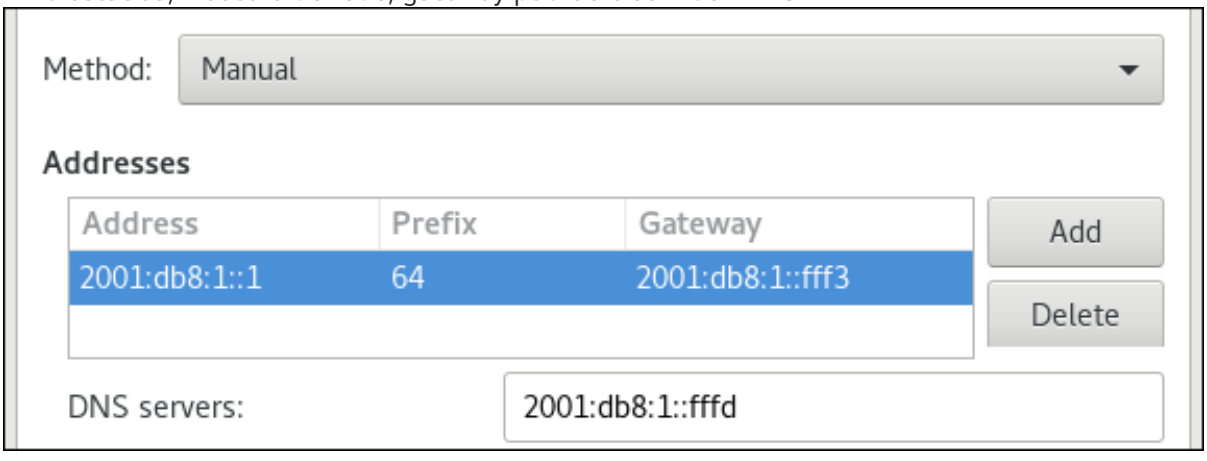
5. Na aba **Ethernet**, selecione um dispositivo e, opcionalmente, outras configurações relacionadas à Ethernet



6. Na aba **IPv4 Settings**, configure as configurações do IPv4. Por exemplo, defina um endereço IPv4 estático, máscara de rede, gateway padrão e servidor DNS



7. Na aba **IPv6 Settings**, configure as configurações IPv6. Por exemplo, configure um endereço IPv6 estático, máscara de rede, gateway padrão e servidor DNS



8. Salvar a conexão.
9. Fechar **nm-connection-editor**.

Etapas de verificação

1. Use o utilitário **ping** para verificar se este host pode enviar pacotes para outros hosts.
   - Pingar um endereço IP na mesma sub-rede.
     Para IPv4:

     ```
     # ping 192.0.2.3
     ```

     Para IPv6:

     ```
     # ping 2001:db8:2::1
     ```

     Se o comando falhar, verificar as configurações de IP e subrede.
   - Pingar um endereço IP em uma sub-rede remota.
     Para IPv4:

     ```
     # ping 198.162.3.1
     ```

     Para IPv6:

     ```
     # ping 2001:db8:2::1
     ```

     - Se o comando falhar, pingar o gateway padrão para verificar as configurações.
       Para IPv4:

       ```
       # ping 192.0.2.254
       ```

       Para IPv6:

       ```
       # ping 2001:db8:1::fff3
       ```
   - Use o utilitário **host** para verificar se a resolução do nome funciona. Por exemplo:

     ```
     # host client.example.com
     ```

     Se o comando retornar algum erro, como **connection timed out** ou **no servers could be reached**, verifique suas configurações de DNS.

Recursos adicionais

- Se a conexão não tiver um gateway padrão, veja Seção 18.8, “Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão”.

8.9. CONFIGURAÇÃO DO COMPORTAMENTO DO DHCP DE UMA CONEXÃO NETWORKMANAGER

Um cliente DHCP (Dynamic Host Configuration Protocol) solicita o endereço IP dinâmico e as informações de configuração correspondentes de um servidor DHCP cada vez que um cliente se conecta à rede.

Quando você configura uma conexão para recuperar um endereço IP de um servidor DHCP, o NetworkManager solicita um endereço IP de um servidor DHCP. Por padrão, o cliente espera 45 segundos para que esta solicitação seja completada. Quando uma conexão **DHCP** é iniciada, um cliente dhcp solicita um endereço IP a partir de um servidor **DHCP**.

Pré-requisitos

- Uma conexão que utiliza DHCP é configurada no host.

Procedimento

1. Defina as propriedades **ipv4.dhcp-timeout** e **ipv6.dhcp-timeout**. Por exemplo, para definir ambas as opções para **30** segundos, entre:

   ```
   # nmcli connection modify connection_name ipv4.dhcp-timeout 30 ipv6.dhcp-timeout 30
   ```

   Alternativamente, defina os parâmetros para **infinity** para configurar que o NetworkManager não pare de tentar solicitar e renovar um endereço IP até que seja bem sucedido.

2. Opcional: Configure o comportamento se o NetworkManager não receber um endereço IPv4 antes do timeout:

   ```
   # nmcli connection modify connection_name ipv4.may-fail value
   ```

   Se você definir a opção **ipv4.may-fail** para:

   - **yes**, o status da conexão depende da configuração IPv6:
     - Se a configuração IPv6 for ativada e bem sucedida, o NetworkManager ativa a conexão IPv6 e não tenta mais ativar a conexão IPv4.
     - Se a configuração IPv6 estiver desativada ou não configurada, a conexão falha.
   - **no**, a conexão está desativada. Neste caso:
     - Se a propriedade **autoconnect** da conexão estiver habilitada, o NetworkManager tenta novamente ativar a conexão tantas vezes quantas as definidas na propriedade **autoconnect-retries**. O padrão é **4**.
     - Se a conexão ainda não puder adquirir um endereço DHCP, a ativação automática falha. Observe que após 5 minutos, o processo de auto-conexão começa novamente para adquirir um endereço IP do servidor DHCP.

3. Opcional: Configure o comportamento se o NetworkManager não receber um endereço IPv6 antes do timeout:

   ```
   # nmcli connection modify connection_name ipv6.may-fail value
   ```

Recursos adicionais

- Para mais detalhes sobre as propriedades descritas nesta seção, consulte a página de manual **nm-settings(5)**.

CAPÍTULO 9. GERENCIANDO CONEXÕES WI-FI

Esta seção descreve como configurar e gerenciar as conexões Wi-Fi.

9.1. CONFIGURANDO O DOMÍNIO REGULATÓRIO SEM FIO

No Red Hat Enterprise Linux, o **crda** o pacote contém o Agente Regulatório Central de Domínio que fornece ao núcleo as regras regulatórias sem fio para uma determinada jurisdição. Ele é usado por certos **udev** scripts e não deve ser executado manualmente, a menos que seja feita uma depuração **udev** roteiros. O kernel roda **crda** enviando um **udev** em uma nova mudança de domínio regulatório. Mudanças no domínio regulatório são acionadas pelo subsistema sem fio Linux (IEEE-802.11). Este subsistema usa o arquivo **regulatory.bin** para manter suas informações de banco de dados regulamentares.

O utilitário **setregdomain** define o domínio regulatório para seu sistema. **Setregdomain** não aceita argumentos e é normalmente chamado através do script do sistema, tais como **udev** em vez de manualmente pelo administrador. Se um código de país falhar, o administrador do sistema pode definir a variável de ambiente **COUNTRY** no arquivo **/etc/sysconfig/regdomain**.

Recursos adicionais

Consulte as seguintes páginas de manual para obter mais informações sobre o domínio regulatório:

- **setregdomain(1)** man page - Define o domínio regulatório com base no código do país.
- **crda(8)** man page - Envia ao kernel um domínio regulador sem fio para uma determinada ISO ou IEC 3166 alpha2.
- **regulatory.bin(5)** man page - Mostra o banco de dados regulatório sem fio Linux.
- **iw(8)** man page - Mostra ou manipula os dispositivos sem fio e sua configuração.

9.2. CONFIGURAÇÃO DE UMA CONEXÃO WI-FI USANDO NMCLI

Este procedimento descreve como configurar um perfil de conexão Wi-fi usando nmcli.

Pré-requisitos

- O utilitário **nmcli** a ser instalado.
- Certifique-se de que o rádio WiFi esteja ligado (padrão):

```
~]$ nmcli radio wifi on
```

Procedimento

1. Para criar um perfil de conexão Wi-Fi com configuração estática **IP**:

```
~]$ nmcli con add con-name MyCafe ifname wlan0 type wifi ssid MyCafe ``ip4
192.168.100.101/24 gw4 192.168.100.1
```

2. Configurar um servidor DNS. Por exemplo, para definir **192.160.100.1** como o servidor DNS:

```
~]$ nmcli con modify con-name MyCafe ipv4.dns "192.160.100.1"
```

3. Opcionalmente, defina um domínio de busca DNS. Por exemplo, para definir o domínio de busca para **example.com**:

```
~]$ nmcli con modify con-name MyCafe ipv4.dns-search "example.com"
```

4. Para verificar uma propriedade específica, por exemplo **mtu**:

```
~]$ nmcli connection show id MyCafe | grep mtu
802-11-wireless.mtu:                    auto
```

5. Para mudar a propriedade de um ambiente:

```
~]$ nmcli connection modify id MyCafe 802-11-wireless.mtu 1350
```

6. Para verificar a mudança:

```
~]$ nmcli connection show id MyCafe | grep mtu
802-11-wireless.mtu:                    1350
```

Etapas de verificação

1. Use o utilitário **ping** para verificar se este host pode enviar pacotes para outros hosts.
   - Pingar um endereço IP na mesma sub-rede. Por exemplo, o endereço IP de uma sub-rede:

     ```
     # ping 192.168.100.103
     ```

     Se o comando falhar, verificar as configurações de IP e subrede.
   - Pingar um endereço IP em uma sub-rede remota. Por exemplo, o endereço IP de uma sub-rede remota:

     ```
     # ping 198.51.16.3
     ```

     - Se o comando falhar, pingar o gateway padrão para verificar as configurações.

       ```
       # ping 192.168.100.1
       ```

2. Use o utilitário **host** para verificar se a resolução do nome funciona. Por exemplo:

```
# host client.example.com
```

Se o comando retornar algum erro, como **connection timed out** ou **no servers could be reached**, verifique suas configurações de DNS.

Recursos adicionais

- Consulte a página de manual **nm-settings(5)** para mais informações sobre propriedades e suas configurações.
- Se a configuração no disco não corresponder à configuração no dispositivo, iniciar ou reiniciar o NetworkManager cria uma conexão in-memory que reflete a configuração do dispositivo. Para maiores detalhes e como evitar este problema, veja NetworkManager duplica uma conexão

após o reinício do serviço NetworkManager.

9.3. CONFIGURAÇÃO DE UMA CONEXÃO WI-FI USANDO O CENTRO DE CONTROLE

Quando você se conecta a um **Wi-Fi**, as configurações de rede são pré-preenchidas dependendo da conexão de rede atual. Isto significa que as configurações serão detectadas automaticamente quando a interface se conectar a uma rede.

Este procedimento descreve como usar **control-center** para configurar manualmente as configurações do **Wi-Fi**.

Procedimento

1. Pressione a tecla **Super** para entrar no **Activities Overview**, digite **Wi-Fi** e pressione **Enter**. Na entrada do menu do lado esquerdo, você vê a lista de redes disponíveis.

2. Selecione o ícone da roda dentada à direita do nome da conexão **Wi-Fi** que você deseja editar, e o diálogo de conexão de edição aparece. A janela do menu **Details** mostra os detalhes da conexão onde você pode fazer outras configurações.
   Options

   a. Se você selecionar **Connect automatically**, **NetworkManager** auto-conecta-se a esta conexão sempre que **NetworkManager** detecta que ela está disponível. Se você não quiser **NetworkManager** para se conectar automaticamente, desmarque a caixa de seleção. Note que quando a caixa de seleção estiver desmarcada, você tem que selecionar essa conexão manualmente no menu do ícone de conexão de rede para fazer com que ela se conecte.

   b. Para disponibilizar uma conexão a outros usuários, selecione a caixa de seleção **Make available to other users**.

   c. Você também pode controlar o uso dos dados de fundo. Se você deixar **Restrict background data usage** não especificado (padrão), então **NetworkManager** tenta fazer o download dos dados que você está usando ativamente. Caso contrário, selecione a caixa de seleção e **NetworkManager** define a conexão como medida, e aplica restrição ao uso de dados de fundo.

NOTA

Para excluir uma conexão **Wi-Fi**, clique na caixa vermelha **Forget Connection**.

3. Selecione a entrada do menu **Identity** para ver as opções básicas de configuração.
   SSID - O *Service Set Identifier* (SSID) do ponto de acesso (AP).

   BSSID - O *Basic Service Set Identifier* (BSSID) é o endereço MAC, também conhecido como *hardware address*, do ponto de acesso sem fio específico ao qual você está se conectando quando no modo **Infrastructure**. Este campo está em branco por padrão, e você é capaz de conectar-se a um ponto de acesso sem fio por **SSID** sem ter que especificar seu **BSSID**. Se o BSSID for especificado, ele forçará o sistema a se associar somente a um ponto de acesso específico. Para redes ad-hoc, o **BSSID** é gerado aleatoriamente pelo **mac80211** quando a rede ad-hoc é criada. Ela não é exibida por **NetworkManager**.

   MAC address - O *MAC address* permite associar um adaptador sem fio específico com uma conexão específica (ou conexões).

Cloned Address - Um endereço MAC clonado para usar no lugar do endereço de hardware real. Deixe em branco, a menos que seja necessário.

4. Para configuração posterior do endereço IP , selecione as entradas de menu **IPv4** e **IPv6**. Por padrão, tanto **IPv4** como **IPv6** estão definidos para configuração automática dependendo das configurações atuais da rede. Isto significa que endereços como o endereço IP local, endereço DNS e outras configurações serão detectados automaticamente quando a interface se conectar a uma rede. Se um servidor DHCP atribui a configuração IP nesta rede, isto é suficiente, mas você também pode fornecer a configuração estática nas configurações **IPv4** e **IPv6**. Nas entradas dos menus **IPv4** e **IPv6**, você pode ver as seguintes configurações:

   - **IPv4 Method**
     - **Automatic (DHCP)** - Escolha esta opção se a rede à qual você está se conectando usa anúncios de Roteador (RA) ou um servidor **DHCP** para atribuir endereços IP dinâmicos. Você pode ver o endereço IP atribuído na entrada do menu **Details**.
     - **Link-Local Only** - Escolha esta opção se a rede à qual você está se conectando não tiver um servidor **DHCP** e você não quiser atribuir endereços IP manualmente. Endereços aleatórios serão atribuídos de acordo com *RFC 3927* com prefixo **169.254/16**.
     - **Manual** - Escolha esta opção se você quiser atribuir endereços IP manualmente.
     - **Disable** - **IPv4** está desativado para esta conexão.
   - **DNS**
     Se **Automatic** for **ON**, e não houver um servidor DHCP disponível que atribua servidores DNS a esta conexão, mude-o para **OFF** para inserir o endereço IP de um servidor DNS que separe os IPs por vírgula.
   - **Routes**
     Observe que na seção **Routes**, quando **Automatic** é **ON**, são usadas rotas de Router Advertisements (RA) ou DHCP, mas você também pode adicionar rotas estáticas adicionais. Quando **OFF**, somente rotas estáticas são usadas.
     - **Address** - Digite o endereço **IP** de uma rede remota, subrede ou host.
     - **Netmask** - A máscara de rede ou o comprimento do prefixo do endereço IP inserido acima.
     - **Gateway** - O endereço IP do gateway que leva à rede remota, subrede ou host inserido acima.
     - **Metric** - Um custo de rede, um valor de preferência a dar a esta rota. Os valores mais baixos serão preferidos em relação aos valores mais altos.
   - **Use this connection only for resources on its network**
     Selecione esta caixa de seleção para evitar que a conexão se torne a rota padrão.

     Alternativamente, para configurar as configurações **IPv6** em uma conexão **Wi-Fi**, selecione a entrada do menu **IPv6**:

   - **IPv6 Method**
     - **Automatic** - Escolha esta opção para usar **IPv6** Endereço sem Estado AutoConfiguração (SLAAC) para criar uma configuração automática, sem Estado, baseada no endereço de hardware e Anúncios de Roteador (RA)

baseada no endereço de hardware e Anúncios de Roteador (RA).

- **Automatic, DHCP only** - Escolha esta opção para não usar a RA, mas solicite informações diretamente de **DHCPv6** para criar uma configuração estatal.
- **Link-Local Only** - Escolha esta opção se a rede à qual você está se conectando não tiver um servidor **DHCP** e você não quiser atribuir endereços IP manualmente. Endereços aleatórios serão atribuídos de acordo com *RFC 4862* com prefixo **FE80::0**.
- **Manual** - Escolha esta opção se você quiser atribuir endereços IP manualmente.
- **Disable** - **IPv6** está desativado para esta conexão.

- Os campos **DNS**, **Routes**, **Use this connection only for resources on its network** são comuns às configurações de **IPv4**.

5. Para configurar as configurações **Security** em uma conexão **Wi-Fi**, selecione a entrada do menu **Security**. As seguintes opções de configuração estão disponíveis:

   - Security
     - **None** - Não encripte a conexão Wi-Fi.
     - **WEP 40/128-bit Key** - Wired Equivalent Privacy (WEP), a partir da norma IEEE 802.11. Utiliza uma única chave pré-compartilhada (PSK).
     - **WEP 128-bit Passphrase** - Um hash MD5 da frase-chave para derivar uma chave WEP.

ATENÇÃO

Se o **Wi-Fi** não utiliza criptografia, **WEP**, ou **WPA**, não utilize a rede porque ela é insegura e todos podem ler os dados que você envia através desta rede.

     - **LEAP** - Lightweight Extensible Authentication Protocol, da Cisco Systems.
     - **Dynamic WEP (802.1X)** - As chaves WEP são alteradas dinamicamente.
     - **WPA & WPA2 Personal** – Wi-Fi Protected Access (WPA), from the draft IEEE 802.11i standard. A replacement for WEP. Wi-Fi Protected Access II (WPA2), from the 802.11i-2004 standard. Personal mode uses a pre-shared key (WPA-PSK).
     - **WPA & WPA2 Enterprise** – WPA for use with a RADIUS authentication server to provide IEEE 802.1X network access control.

   - **Password** - Digite a senha a ser utilizada no processo de autenticação.

6. Uma vez terminada a configuração, clique no botão **Aplicar** para salvá-la.

NOTA

Quando você adiciona uma nova conexão clicando no botão **mais**, **NetworkManager** cria um novo arquivo de configuração para essa conexão e depois abre o mesmo diálogo que é usado para editar uma conexão existente. A diferença entre estes diálogos é que um perfil de conexão existente tem uma entrada no menu **Details**.

9.4. CONECTANDO-SE A UMA REDE WI-FI COM NMCLI

Este procedimento descreve como se conectar a uma conexão **wireless** usando o utilitário **nmcli**.

Pré-requisitos

- O utilitário **nmcli** a ser instalado.
- Certifique-se de que o rádio WiFi esteja ligado (padrão):

```
~]$ nmcli radio wifi on
```

Procedimento

1. Para atualizar a lista de conexões Wi-Fi disponíveis:

```
~]$ nmcli device wifi rescan
```

2. Para visualizar os pontos de acesso Wi-Fi disponíveis:

```
~]$ nmcli dev wifi list

IN-USE  SSID      MODE   CHAN  RATE        SIGNAL  BARS  SECURITY
...
        MyCafe    Infra  3     405 Mbit/s  85      ▂▄▆_  WPA1 WPA2
```

3. Para se conectar a uma conexão Wi-Fi usando **nmcli**:

```
~]$ nmcli dev wifi connect SSID-Name password wireless-password
```

Por exemplo:

```
~]$ nmcli dev wifi connect MyCafe password wireless-password
```

Observe que se você quiser desativar o estado Wi-Fi:

```
~]$ nmcli radio wifi off
```

9.5. CONECTANDO-SE A UMA REDE WI-FI OCULTA USANDO NMCLI

Todos os pontos de acesso têm um Service Set Identifier (SSID) para identificá-los. Entretanto, um ponto de acesso pode ser configurado para não transmitir seu SSID, caso em que ele está oculto, e não aparecerá na lista **NetworkManager's** de redes disponíveis.

Este procedimento mostra como você pode se conectar a uma rede oculta usando a ferramenta **nmcli**.

Pré-requisitos

- O utilitário **nmcli** a ser instalado.
- Para conhecer o SSID, e a senha da conexão **Wi-Fi**.
- Certifique-se de que o rádio WiFi esteja ligado (padrão):

```
~]$ nmcli radio wifi on
```

Procedimento

- Conecte-se ao SSID que está oculto:

```
~]$ nmcli dev wifi connect SSID_Name password wireless_password hidden yes
```

9.6. CONEXÃO A UMA REDE WI-FI USANDO A GUI DO GNOME

Este procedimento descreve como você pode se conectar a uma rede sem fio para ter acesso à Internet.

Procedimento

1. Abra o menu de ícones de conexão de rede do GNOME Shell a partir do canto superior direito da tela.
2. Selecione **Wi-Fi Not Connected**.
3. Clique na opção **Select Network**.
4. Clique no nome da rede à qual você deseja se conectar, e depois clique em **Connect**.
   Note que se você não vir a rede, a rede pode estar escondida.
5. Se a rede for protegida por uma senha ou chaves de criptografia, digite a senha e clique em **Connect**.
   Observe que, se você não souber a senha, entre em contato com o administrador da rede Wi-Fi.
6. Se a conexão for bem sucedida, o nome da rede é visível no menu de ícones de conexão e o indicador sem fio está no canto superior direito da tela.

Recursos adicionais

- Configuração de uma conexão Wi-Fi usando o centro de controle .

CAPÍTULO 10. CONFIGURANDO A ETIQUETAGEM VLAN

Esta seção descreve como configurar a Virtual Local Area Network (VLAN). Uma VLAN é uma rede lógica dentro de uma rede física. Os pacotes de tags da interface VLAN com o ID da VLAN ao passar pela interface, e remove as tags dos pacotes de retorno.

Você cria uma interface VLAN em cima de outra interface, como uma Ethernet, bond, equipe ou dispositivo de ponte. Esta interface é chamada de **parent interface**.

10.1. CONFIGURANDO A MARCAÇÃO VLAN USANDO COMANDOS NMCLI

Esta seção descreve como configurar a etiquetagem da Rede Local Virtual (VLAN) usando o utilitário **nmcli**.

Pré-requisitos

- A interface que você planeja usar como pai para a interface VLAN virtual suporta tags VLAN.
- Se você configurar a VLAN em cima de uma interface de vínculo:
  - Os portos da ligação estão em alta.
  - O vínculo não é configurado com a opção **fail_over_mac=follow**. Um dispositivo virtual VLAN não pode mudar seu endereço MAC para combinar com o novo endereço MAC da matriz. Nesse caso, o tráfego ainda seria enviado com o endereço MAC de origem então incorreto.
- O switch ao qual o host está conectado é configurado para suportar tags VLAN. Para obter detalhes, consulte a documentação de seu switch.

Procedimento

1. Exibir as interfaces de rede:

```
# nmcli device status
DEVICE   TYPE      STATE         CONNECTION
enp1s0   ethernet  disconnected  enp1s0
bridge0  bridge    connected     bridge0
bond0    bond      connected     bond0
...
```

2. Criar a interface VLAN. Por exemplo, para criar uma interface VLAN chamada **vlan10** que usa **enp1s0** como sua interface pai e que tags pacotes com VLAN ID **10**, entre:

```
# nmcli connection add type vlan con-name vlan10 ifname vlan10 vlan.parent enp1s0
vlan.id 10
```

   Note que a VLAN deve estar dentro da faixa de **0** a **4094**.

3. Por padrão, a conexão VLAN herda a unidade de transmissão máxima (MTU) da interface pai. Opcionalmente, defina um valor MTU diferente:

```
# nmcli connection modify vlan10 802-3-ethernet.mtu 2000
```

4. Configurar as configurações de IP do dispositivo VLAN. Pular este passo se você quiser usar este dispositivo VLAN como porta de outros dispositivos.

   a. Configurar as configurações do IPv4. Por exemplo, para configurar um endereço IPv4 estático, máscara de rede, gateway padrão e servidor DNS para a conexão **vlan10**, entre:

   ```
   # nmcli connection modify vlan10 ipv4.addresses '192.0.2.1/24'
   # nmcli connection modify vlan10 ipv4.gateway '192.0.2.254'
   # nmcli connection modify vlan10 ipv4.dns '192.0.2.253'
   # nmcli connection modify vlan10 ipv4.method manual
   ```

   b. Configurar as configurações IPv6. Por exemplo, para configurar um endereço IPv6 estático, máscara de rede, gateway padrão e servidor DNS para a conexão **vlan10**, entre:

   ```
   # nmcli connection modify vlan10 ipv6.addresses '2001:db8:1::1/32'
   # nmcli connection modify vlan10 ipv6.gateway '2001:db8:1::fffe'
   # nmcli connection modify vlan10 ipv6.dns '2001:db8:1::fffd'
   # nmcli connection modify vlan10 ipv6.method manual
   ```

5. Ativar a conexão:

   ```
   # nmcli connection up vlan10
   ```

Etapas de verificação

1. Verificar as configurações:

   ```
   # ip -d addr show vlan10
   4: vlan10@enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
   state UP group default qlen 1000
       link/ether 52:54:00:d5:e0:fb brd ff:ff:ff:ff:ff:ff promiscuity 0
       vlan protocol 802.1Q id 10 <REORDER_HDR> numtxqueues 1 numrxqueues 1
   gso_max_size 65536 gso_max_segs 65535
       inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute vlan10
          valid_lft forever preferred_lft forever
       inet6 2001:db8:1::1/32 scope global noprefixroute
          valid_lft forever preferred_lft forever
       inet6 fe80::8dd7:9030:6f8e:89e6/64 scope link noprefixroute
          valid_lft forever preferred_lft forever
   ```

Recursos adicionais

- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, "Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão".
- Para exemplos em **nmcli**, veja a página de manual **nmcli-examples(7)**.
- Para todas as propriedades vlan que você pode definir, consulte a seção **vlan setting** na página de manual **nm-settings(5)**.

10.2. CONFIGURAÇÃO DA MARCAÇÃO DE VLAN USANDO O EDITOR DE NM-CONEXÃO

Esta seção descreve como configurar a etiquetagem da Rede Local Virtual (VLAN) usando a aplicação **nm-connection-editor**.

Pré-requisitos

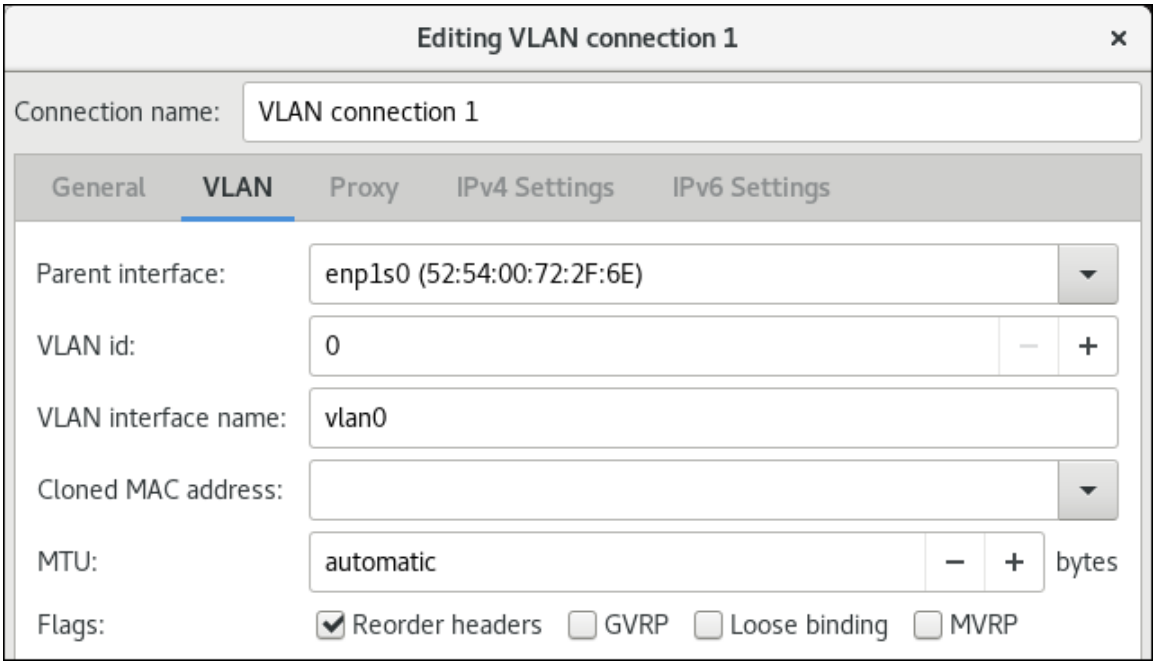
- A interface que você planeja usar como pai para a interface VLAN virtual suporta tags VLAN.
- Se você configurar a VLAN em cima de uma interface de vínculo:
  - Os portos da ligação estão em alta.
  - O vínculo não é configurado com a opção **fail_over_mac=follow**. Um dispositivo virtual VLAN não pode mudar seu endereço MAC para combinar com o novo endereço MAC da matriz. Nesse caso, o tráfego ainda seria enviado com o endereço MAC de origem então incorreto.
- O switch ao qual o host está conectado é configurado para suportar tags VLAN. Para obter detalhes, consulte a documentação de seu switch.

Procedimento

1. Abra um terminal e entre em **nm-connection-editor**:

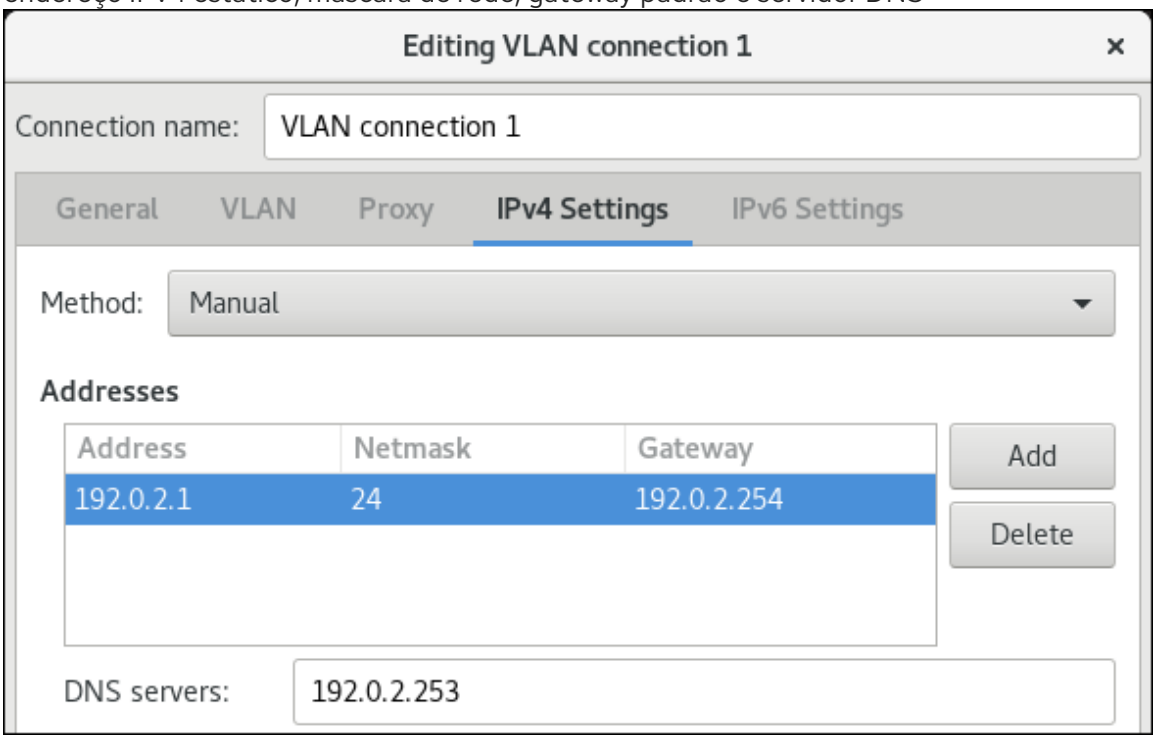
```
$ nm-connection-editor
```

2. Clique no botão para adicionar uma nova conexão.
3. Selecione o tipo de conexão **VLAN**, e clique em **Criar**.
4. Na aba **VLAN**:
   a. Selecione a interface dos pais.
   b. Selecione a identificação VLAN. Observe que a VLAN deve estar dentro da faixa de **0** a **4094**.
   c. Por padrão, a conexão VLAN herda a unidade de transmissão máxima (MTU) da interface pai. Opcionalmente, defina um valor MTU diferente.
   d. Opcionalmente, defina o nome da interface VLAN e outras opções específicas da VLAN.

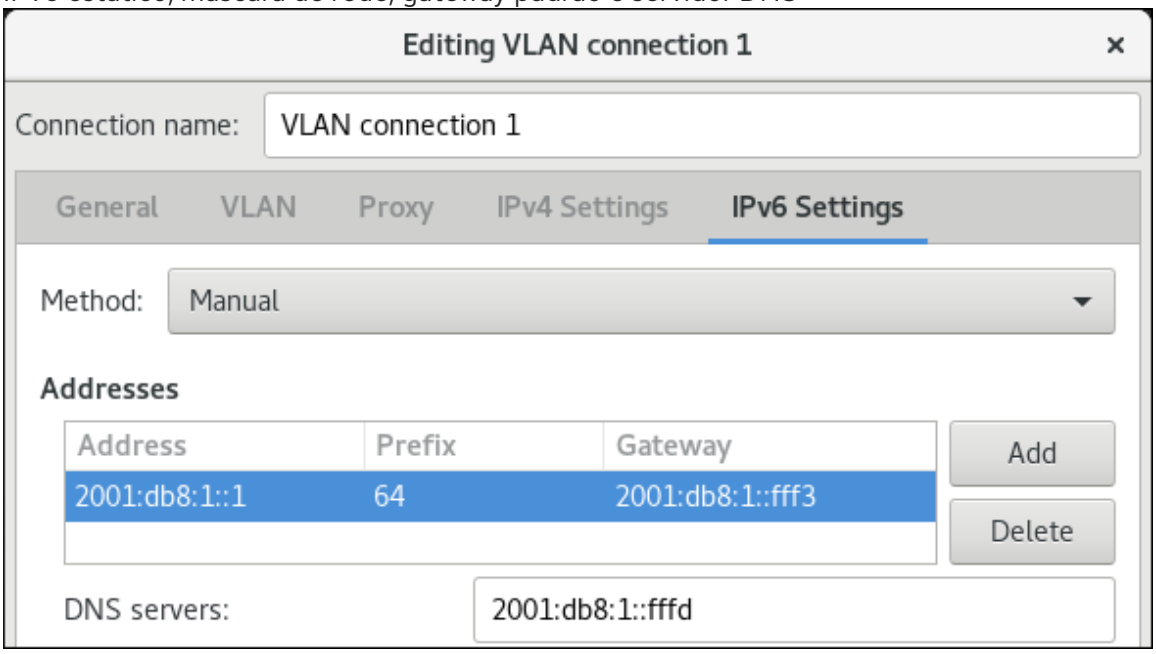


5. Configurar as configurações de IP do dispositivo VLAN. Pular este passo se você quiser usar este dispositivo VLAN como porta de outros dispositivos.

   a. Na aba **IPv4 Settings**, configure as configurações do IPv4. Por exemplo, defina um endereço IPv4 estático, máscara de rede, gateway padrão e servidor DNS



b. Na aba **IPv6 Settings**, configure as configurações IPv6. Por exemplo, defina um endereço IPv6 estático, máscara de rede, gateway padrão e servidor DNS



6. Clique em **Salvar** para salvar a conexão VLAN.

7. Fechar **nm-connection-editor**.

Etapas de verificação

1. Verificar as configurações:

```
# ip -d addr show vlan10
4: vlan10@enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default qlen 1000
    link/ether 52:54:00:d5:e0:fb brd ff:ff:ff:ff:ff:ff promiscuity 0
    vlan protocol 802.1Q id 10 <REORDER_HDR> numtxqueues 1 numrxqueues 1
gso_max_size 65536 gso_max_segs 65535
    inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute vlan10
       valid_lft forever preferred_lft forever
    inet6 2001:db8:1::1/32 scope global noprefixroute
       valid_lft forever preferred_lft forever
    inet6 fe80::8dd7:9030:6f8e:89e6/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
```

Recursos adicionais

- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.

- Se a conexão não tiver um gateway padrão, veja Seção 18.8, "Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão".

10.3. CONFIGURAÇÃO DA ETIQUETAGEM VLAN USANDO AS FUNÇÕES DO SISTEMA

Você pode usar a função do sistema **networking** RHEL para configurar a etiquetagem VLAN. Este procedimento descreve como adicionar uma conexão Ethernet e uma VLAN com ID **10** que usa esta

conexão Ethernet. Como o dispositivo pai, a conexão VLAN contém as configurações IP, gateway padrão e DNS.

Dependendo de seu ambiente, ajuste o jogo de acordo. Por exemplo:

- Para usar a VLAN como uma porta em outras conexões, como uma ligação, omitir o atributo **ip**, e definir a configuração IP na configuração pai.
- Para usar os dispositivos de equipe, ponte, ou bond na VLAN, adapte os atributos **interface_name** e **type** dos portos que você usa na VLAN.

Pré-requisitos

- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.
- Se você usar um usuário remoto diferente de **root** ao executar o playbook, este usuário tem as permissões apropriadas **sudo** no nó gerenciado.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo **/etc/ansible/hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook **~/vlan-ethernet.yml** com o seguinte conteúdo:

```
---
- name: Configure a VLAN that uses an Ethernet connection
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network

    vars:
      network_connections:
        # Add an Ethernet profile for the underlying device of the VLAN
        - name: enp1s0
          type: ethernet
  interface_name: enp1s0
  autoconnect: yes
          state: up
  ip:
    dhcp4: no
    auto6: no

        # Define the VLAN profile
        - name: vlan10
          type: vlan
          ip:
            address:
              - "192.0.2.1/24"
              - "2001:db8:1::1/64"
```

```
      gateway4: 192.0.2.254
      gateway6: 2001:db8:1::fffe
      dns:
        - 192.0.2.200
        - 2001:db8:1::ffbb
      dns_search:
        - example.com
    vlan_id: 10
parent: enp1s0
    state: up
```

O atributo **parent** no perfil da VLAN configura a VLAN para operar em cima do dispositivo **enp1s0**.

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

     ```
     # ansible-playbook -u root ~/vlan-ethernet.yml
     ```

   - Para conectar-se como usuário ao host administrado, entre:

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/vlan-ethernet.yml
     ```

     A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definido no **-u *user_name*** opção.

   Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Recursos adicionais

- Para detalhes sobre os parâmetros usados em **network_connections** e para informações adicionais sobre o Sistema de Papel **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

CAPÍTULO 11. CONFIGURAÇÃO DE UMA PONTE DE REDE

Uma ponte de rede é um dispositivo de camada de ligação que encaminha o tráfego entre redes com base em uma tabela de endereços MAC. A ponte constrói a tabela de endereços MAC ouvindo o tráfego da rede e, assim, aprendendo quais hosts estão conectados a cada rede. Por exemplo, você pode usar uma ponte de software em um host Red Hat Enterprise Linux 8 para emular uma ponte de hardware ou em ambientes de virtualização, para integrar máquinas virtuais (VM) à mesma rede que o host.

Uma ponte requer um dispositivo de rede em cada rede que a ponte deve conectar. Quando você configura uma ponte, a ponte é chamada **controller** e os dispositivos que ela usa **ports**.

Você pode criar pontes em diferentes tipos de dispositivos, como por exemplo:

- Dispositivos Ethernet físicos e virtuais
- Títulos de rede
- Equipes de rede
- Dispositivos VLAN

Devido ao padrão IEEE 802.11 que especifica o uso de quadros de 3 endereços em Wi-Fi para o uso eficiente do tempo de antena, não é possível configurar uma ponte sobre redes Wi-Fi operando nos modos Ad-Hoc ou Infra-estrutura.

11.1. CONFIGURAÇÃO DE UMA PONTE DE REDE USANDO COMANDOS NMCLI

Esta seção explica como configurar uma ponte de rede usando o utilitário **nmcli**.

Pré-requisitos

- Dois ou mais dispositivos físicos ou virtuais de rede são instalados no servidor.
- Para usar dispositivos Ethernet como portas da ponte, os dispositivos Ethernet físicos ou virtuais devem ser instalados no servidor.
- Para usar dispositivos de equipe, bond, ou VLAN como portas da ponte, você pode criar estes dispositivos enquanto cria a ponte ou pode criá-los antecipadamente como descrito em:
  - Seção 13.5, “Configuração de uma ligação em rede usando comandos nmcli”
  - Seção 12.6, “Configuração de uma equipe de rede usando comandos nmcli”
  - Seção 10.1, “Configurando a marcação VLAN usando comandos nmcli”

Procedimento

1. Criar uma interface de ponte:

   ```
   # nmcli connection add type bridge con-name bridge0 ifname bridge0
   ```

   Este comando cria uma ponte chamada **bridge0**, entre:

2. Mostre as interfaces de rede, e anote os nomes das interfaces que você deseja acrescentar à ponte:

```
# nmcli device status
DEVICE  TYPE      STATE         CONNECTION
enp7s0  ethernet  disconnected  --
enp8s0  ethernet  disconnected  --
bond0   bond      connected     bond0
bond1   bond      connected     bond1
...
```

Neste exemplo:

- **enp7s0** e **enp8s0** não estão configurados. Para usar estes dispositivos como portas, adicione perfis de conexão na próxima etapa.
- **bond0** e **bond1** têm perfis de conexão existentes. Para usar estes dispositivos como portas, modifique seus perfis na próxima etapa.

3. Atribuir as interfaces à ponte.

   a. Se as interfaces que você deseja atribuir à ponte não estiverem configuradas, crie novos perfis de conexão para elas:

```
# nmcli connection add type ethernet slave-type bridge con-name bridge0-port1 ifname enp7s0 master bridge0
# nmcli connection add type ethernet slave-type bridge con-name bridge0-port2 ifname enp8s0 master bridge0
```

   Estes comandos criam perfis para **enp7s0** e **enp8s0**, e os adicionam à conexão **bridge0**.

   b. Se você quiser atribuir um perfil de conexão existente à ponte, defina o parâmetro **master** dessas conexões para **bridge0**:

```
# nmcli connection modify bond0 master bridge0
# nmcli connection modify bond1 master bridge0
```

   Estes comandos atribuem os perfis de conexão existentes denominados **bond0** e **bond1** à conexão **bridge0**.

4. Configurar as configurações de IP da ponte. Pule este passo se você quiser usar esta ponte como porta de outros dispositivos.

   a. Configurar as configurações do IPv4. Por exemplo, para configurar um endereço IPv4 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS da conexão **bridge0**, digite:

```
# nmcli connection modify bridge0 ipv4.addresses '192.0.2.1/24'
# nmcli connection modify bridge0 ipv4.gateway '192.0.2.254'
# nmcli connection modify bridge0 ipv4.dns '192.0.2.253'
# nmcli connection modify bridge0 ipv4.dns-search 'example.com'
# nmcli connection modify bridge0 ipv4.method manual
```

   b. Configurar as configurações IPv6. Por exemplo, para configurar um endereço IPv6 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS da conexão **bridge0**, digite:

```
# nmcli connection modify bridge0 ipv6.addresses '2001:db8:1::1/64'
# nmcli connection modify bridge0 ipv6.gateway '2001:db8:1::fffe'
# nmcli connection modify bridge0 ipv6.dns '2001:db8:1::fffd'
# nmcli connection modify bridge0 ipv6.dns-search 'example.com'
# nmcli connection modify bridge0 ipv6.method manual
```

5. Opcional: Configurar outras propriedades da ponte. Por exemplo, para definir a prioridade do Protocolo Spanning Tree (STP) de **bridge0** a **16384**, entre:

```
# nmcli connection modify bridge0 bridge.priority '16384'
```

   Por padrão, o STP está habilitado.

6. Ativar a conexão:

```
# nmcli connection up bridge0
```

7. Verifique se os portos estão conectados, e a coluna **CONNECTION** mostra o nome da conexão do porto:

```
# nmcli device
DEVICE   TYPE      STATE      CONNECTION
...
enp7s0   ethernet  connected  bridge0-port1
enp8s0   ethernet  connected  bridge0-port2
```

   O Red Hat Enterprise Linux ativa o controlador e as portas quando o sistema inicia. Ativando qualquer conexão de porta, o controlador também é ativado. No entanto, neste caso, apenas uma conexão de porta é ativada. Por default, a ativação do controlador não ativa automaticamente as portas. No entanto, é possível ativar este comportamento através da configuração:

   a. Habilitar o parâmetro **connection.autoconnect-slaves** da conexão da ponte:

```
# nmcli connection modify bridge0 connection.autoconnect-slaves 1
```

   b. Reativar a ponte:

```
# nmcli connection up bridge0
```

Etapas de verificação

- Exibir o status do link dos dispositivos Ethernet que são portas de uma ponte específica:

```
# ip link show master bridge0
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bridge0 state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:62:61:0e brd ff:ff:ff:ff:ff:ff
4: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bridge0 state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:9e:f1:ce brd ff:ff:ff:ff:ff:ff
```

- Mostrar o status dos dispositivos Ethernet que são portas de qualquer dispositivo de ponte:

```
# bridge link show
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge0 state
forwarding priority 32 cost 100
4: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge0 state
listening priority 32 cost 100
5: enp9s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge1 state
forwarding priority 32 cost 100
6: enp11s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge1 state
blocking priority 32 cost 100
...
```

Para exibir o status de um dispositivo Ethernet específico, use o **bridge link show dev *ethernet_device_name*** comando.

Recursos adicionais

- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, "Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão".
- Para exemplos em **nmcli**, veja a página de manual **nmcli-examples(7)**.
- Para todas as propriedades da ponte que você pode definir, consulte a seção **bridge settings** na página de manual **nm-settings(5)**.
- Para todas as propriedades do porto da ponte que você pode definir, consulte a seção **bridge-port settings** na página de manual **nm-settings(5)**.
- Para obter detalhes sobre a utilidade **bridge**, consulte a página de manual **bridge(8)**.
- Se a configuração no disco não corresponder à configuração no dispositivo, iniciar ou reiniciar o NetworkManager cria uma conexão in-memory que reflete a configuração do dispositivo. Para maiores detalhes e como evitar este problema, veja NetworkManager duplica uma conexão após o reinício do serviço NetworkManager.

11.2. CONFIGURAÇÃO DE UMA PONTE DE REDE USANDO UM EDITOR DE CONEXÃO NM

Esta seção explica como configurar uma ponte de rede usando a aplicação **nm-connection-editor**.

Note que **nm-connection-editor** pode adicionar apenas novos portos a uma ponte. Para usar um perfil de conexão existente como porta, crie a ponte usando o utilitário **nmcli** como descrito em Seção 11.1, "Configuração de uma ponte de rede usando comandos nmcli".

Pré-requisitos

- Dois ou mais dispositivos físicos ou virtuais de rede são instalados no servidor.
- Para usar dispositivos Ethernet como portas da ponte, os dispositivos Ethernet físicos ou virtuais devem ser instalados no servidor.
- Para usar dispositivos de equipe, de ligação ou VLAN como portas da ponte, certifique-se de que estes dispositivos ainda não estejam configurados.

Procedimento

1. Abra um terminal e entre em **nm-connection-editor**:

```
$ nm-connection-editor
```

2. Clique no botão para adicionar uma nova conexão.

3. Selecione o tipo de conexão **Bridge**, e clique em **Criar**.

4. Na aba **Bridge**:

   a. Opcional: Defina o nome da interface da ponte no campo **Interface name**.

   b. Clique no botão **Adicionar** para criar um novo perfil de conexão para uma interface de rede e adicionar o perfil como uma porta para a ponte.

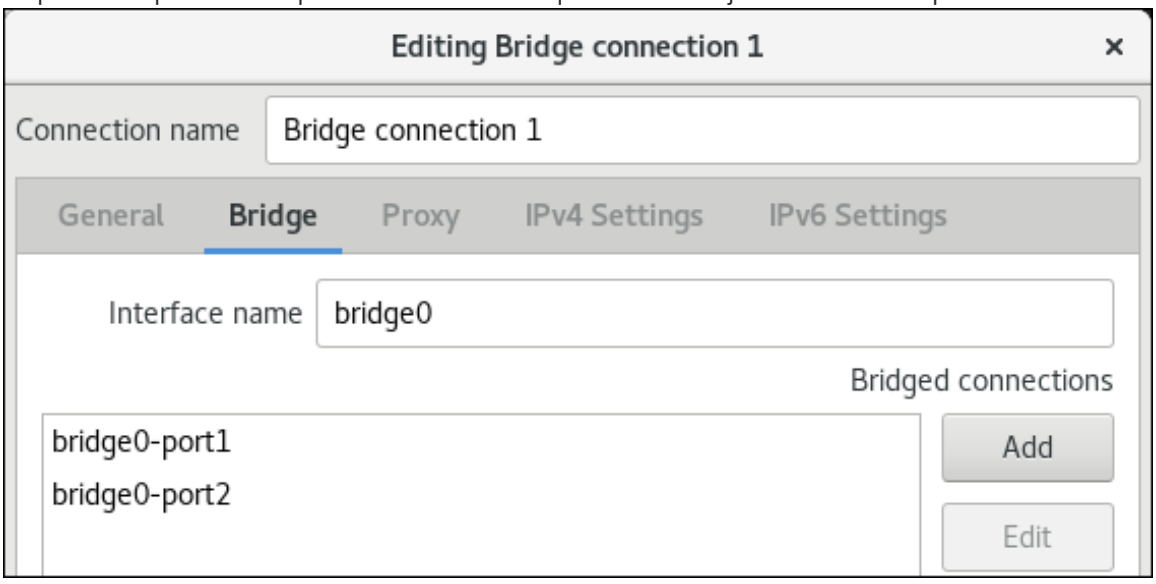
      i. Selecione o tipo de conexão da interface. Por exemplo, selecione **Ethernet** para uma conexão com fio.

      ii. Opcionalmente, defina um nome de conexão para o dispositivo de porta.

      iii. Se você criar um perfil de conexão para um dispositivo Ethernet, abra a aba **Ethernet**, e selecione no campo **Device** a interface de rede que você deseja adicionar como porta para a ponte. Se você selecionou um tipo de dispositivo diferente, configure-o de acordo.

      iv. Clique em **Salvar**.

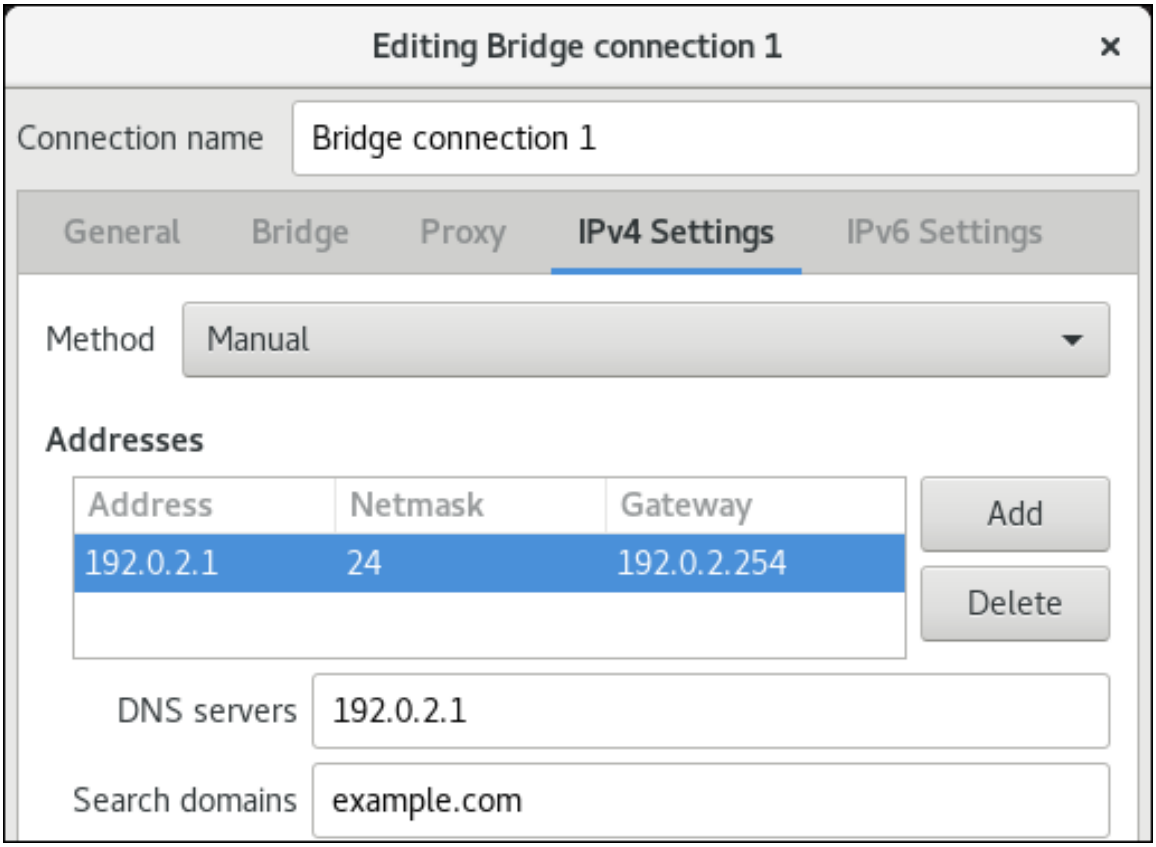
   c. Repita a etapa anterior para cada interface que você deseja acrescentar à ponte.



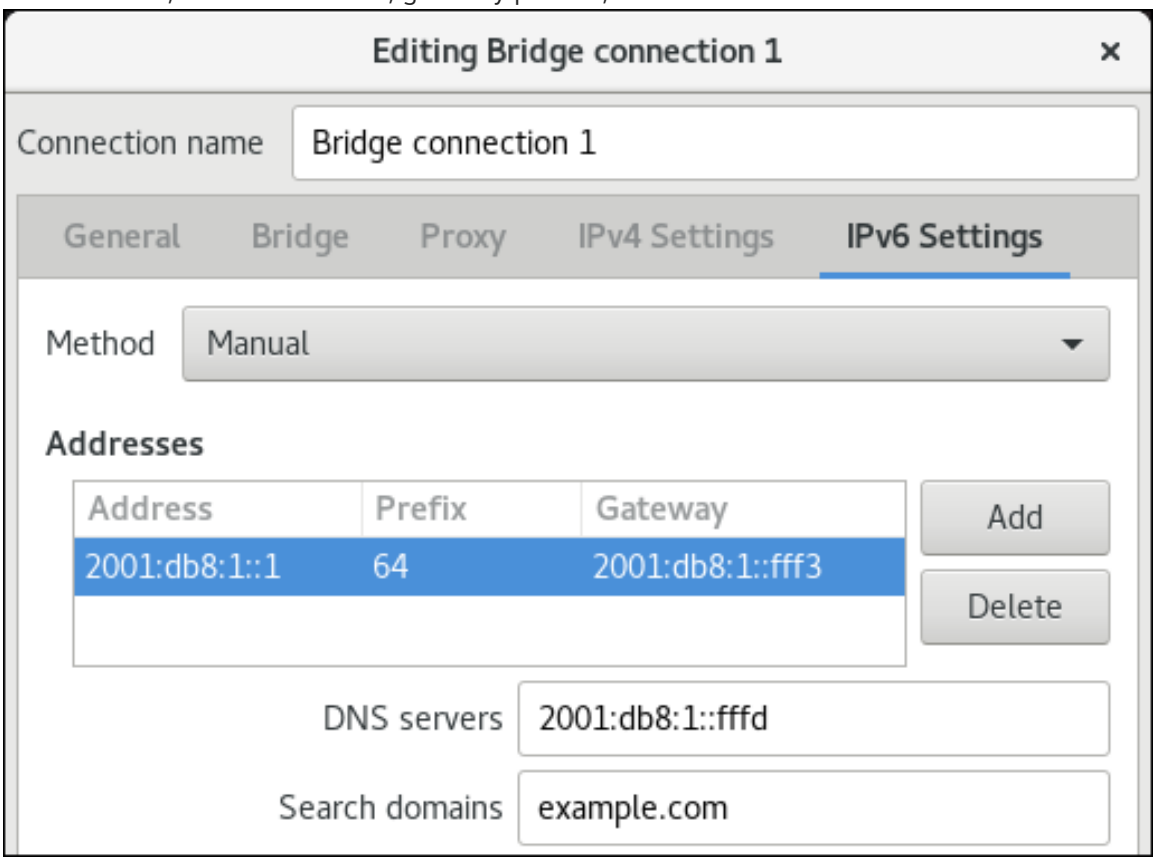
5. Opcional: Configurar outras configurações de ponte, tais como opções do Protocolo Spanning Tree (STP).

6. Configurar as configurações de IP da ponte. Pule esta etapa se você quiser usar esta ponte como porta de outros dispositivos.

   a. Na aba **IPv4 Settings**, configure as configurações do IPv4. Por exemplo, defina um endereço IPv4 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS:



b. Na aba **IPv6 Settings**, configure as configurações IPv6. Por exemplo, defina um endereço IPv6 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS:



7. Salvar a conexão da ponte.
8. Fechar **nm-connection-editor**.

Etapas de verificação

- Exibir o status do link dos dispositivos Ethernet que são portas de uma ponte específica.

```
# ip link show master bridge0
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bridge0 state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:62:61:0e brd ff:ff:ff:ff:ff:ff
4: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
bridge0 state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:9e:f1:ce brd ff:ff:ff:ff:ff:ff
```

- Mostrar o status dos dispositivos Ethernet que são portas em qualquer dispositivo de ponte:

```
# bridge link show
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge0 state
forwarding priority 32 cost 100
4: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge0 state
listening priority 32 cost 100
5: enp9s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge1 state
forwarding priority 32 cost 100
6: enp11s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge1 state
blocking priority 32 cost 100
...
```

  Para exibir o status de um dispositivo Ethernet específico, use o **bridge link show dev *ethernet_device_name*** comando.

Recursos adicionais

- Seção 13.6, “Configuração de uma ligação de rede usando um editor de nm-conexão”
- Seção 12.7, “Configuração de uma equipe de rede usando um editor de nm-conexão”
- Seção 10.2, “Configuração da marcação de VLAN usando o editor de nm-conexão”
- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, “Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão”.

11.3. CONFIGURAÇÃO DE UMA PONTE DE REDE USANDO AS FUNÇÕES DO SISTEMA RHEL

Você pode usar o sistema **networking** RHEL Role para configurar uma ponte Linux. Este procedimento descreve como configurar uma ponte de rede que utiliza dois dispositivos Ethernet e define endereços IPv4 e IPv6, gateways padrão e configuração DNS.

NOTA

Defina a configuração IP na ponte e não nas portas da ponte Linux.

Pré-requisitos

- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.

- Se você usar um usuário remoto diferente de **root** ao executar o playbook, este usuário tem as permissões apropriadas **sudo** no nó gerenciado.
- Dois ou mais dispositivos físicos ou virtuais de rede são instalados no servidor.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo **/etc/ansible/hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook **~/bridge-ethernet.yml** com o seguinte conteúdo:

```
---
- name: Configure a network bridge that uses two Ethernet ports
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network

    vars:
      network_connections:
        # Define the bridge profile
        - name: bridge0
          type: bridge
          interface_name: bridge0
          ip:
            address:
              - "192.0.2.1/24"
              - "2001:db8:1::1/64"
            gateway4: 192.0.2.254
            gateway6: 2001:db8:1::fffe
            dns:
              - 192.0.2.200
              - 2001:db8:1::ffbb
            dns_search:
              - example.com
          state: up

        # Add an Ethernet profile to the bridge
        - name: bridge0-port1
          interface_name: enp7s0
          type: ethernet
          master: bridge0
          slave_type: bridge
          state: up

        # Add a second Ethernet profile to the bridge
        - name: bridge0-port2
          interface_name: enp8s0
          type: ethernet
```

```
      master: bridge0
      slave_type: bridge
      state: up
```

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

     ```
     # ansible-playbook -u root ~/bridge-ethernet.yml
     ```

   - Para conectar-se como usuário ao host administrado, entre:

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/bridge-ethernet.yml
     ```

     A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definido no **-u *user_name*** opção.

   Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Recursos adicionais

- Para detalhes sobre os parâmetros usados em **network_connections** e para informações adicionais sobre o Sistema de Papel **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

CAPÍTULO 12. CONFIGURAÇÃO DA EQUIPE DA REDE

Esta seção descreve os conceitos básicos do trabalho em rede em equipe, as diferenças entre a união e o trabalho em equipe, e como configurar uma equipe de rede no Red Hat Enterprise Linux 8.

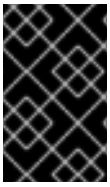
Você pode criar equipes de rede em diferentes tipos de dispositivos, como por exemplo:

- Dispositivos Ethernet físicos e virtuais
- Títulos de rede
- Pontes de rede
- Dispositivos VLAN

12.1. ENTENDENDO O TRABALHO EM EQUIPE EM REDE

A equipe de rede é uma característica que combina ou agrega interfaces de rede para fornecer uma interface lógica com maior rendimento ou redundância.

A equipe de rede usa um driver de kernel para implementar um manuseio rápido dos fluxos de pacotes, assim como bibliotecas de espaço do usuário e serviços para outras tarefas. Desta forma, o trabalho em equipe em rede é uma solução facilmente extensível e escalável para balanceamento de carga e requisitos de redundância.



IMPORTANTE

Certas características de equipe de rede, como o mecanismo de fail-over, não suportam conexões diretas de cabos sem um switch de rede. Para mais detalhes, veja Ligação com conexão direta usando cabos cruzados?

12.2. ENTENDENDO O COMPORTAMENTO PADRÃO DO CONTROLADOR E DAS INTERFACES DE PORTA

Considere o seguinte comportamento padrão, ao gerenciar ou solucionar problemas de equipe ou interfaces de portas de vínculo usando o serviço **NetworkManager**:

- O início da interface do controlador não inicia automaticamente as interfaces de porta.
- Iniciar uma interface de porta sempre inicia a interface do controlador.
- A parada da interface do controlador também pára a interface da porta.
- Um controlador sem portas pode iniciar conexões IP estáticas.
- Um controlador sem portas espera por portas ao iniciar as conexões DHCP.
- Um controlador com uma conexão DHCP à espera de portas se completa quando você adiciona uma porta com um transportador.
- Um controlador com uma conexão DHCP esperando por portas continua esperando quando você adiciona uma porta sem transportador.

12.3. COMPARAÇÃO ENTRE AS CARACTERÍSTICAS DE EQUIPE DE REDE E DE LIGAÇÃO

Conheça os recursos suportados em equipes de rede e vínculos de rede:

Destaque	Ligação em rede	Equipe da rede
Política de Tx Broadcast	Sim	Sim
Política do Tx Round-robin	Sim	Sim
Política de Tx backup ativo	Sim	Sim
Suporte LACP (802.3ad)	Sim (somente ativo)	Sim
Política Tx baseada em hastes	Sim	Sim
O usuário pode definir a função hash	Não	Sim
Tx load-balancing support (TLB)	Sim	Sim
LACP porto hash selecionar	Sim	Sim
Balanceamento de carga para suporte LACP	Não	Sim
Monitoramento do Ethtool link	Sim	Sim
Monitoramento do link ARP	Sim	Sim
Monitoramento de links NS/NA (IPv6)	Não	Sim
Atrasos nas portas para cima/para baixo	Sim	Sim
Prioridades portuárias e aderência (opção "primária" de melhoria)	Não	Sim
Configuração de monitoramento separado por link de porta	Não	Sim
Configuração de monitoramento de múltiplos links	Limitado	Sim
Caminho Tx/Rx sem fechadura	Não (rwlock)	Sim (RCU)

Destaque	Ligação em rede	Equipe da rede
Suporte VLAN	Sim	Sim
Controle do tempo de execução do espaço do usuário	Limitado	Sim
Lógica no espaço do usuário	Não	Sim
Extensibilidade	Difícil	Fácil
Design modular	Não	Sim
Despesas gerais de desempenho	Baixo	Muito baixo
Interface D-Bus	Não	Sim
Empilhamento de múltiplos dispositivos	Sim	Sim
Configuração zero usando LLDP	Não	(no planejamento)
Apoio ao NetworkManager	Sim	Sim

12.4. ENTENDENDO O SERVIÇO DA EQUIPE, CORREDORES E VIGILANTES DE LIGAÇÃO

O serviço de equipe, **teamd**, controla uma instância do motorista da equipe. Esta instância do driver acrescenta instâncias de um driver de dispositivo de hardware para formar uma equipe de interfaces de rede. O driver da equipe apresenta uma interface de rede, por exemplo **team0**, para o kernel.

O serviço **teamd** implementa a lógica comum a todos os métodos de trabalho em equipe. Essas funções são exclusivas dos diferentes métodos de compartilhamento de carga e backup, como o round-robin, e implementadas por unidades separadas de código referidas como **runners**. Os administradores especificam os executores no formato JavaScript Object Notation (JSON), e o código JSON é compilado em uma instância de **teamd** quando a instância é criada. Alternativamente, ao utilizar **NetworkManager**, você pode definir o runner no parâmetro **team.runner**, e **NetworkManager** auto-cria o código JSON correspondente.

Estão disponíveis os seguintes corredores:

- **broadcast**: Transmite dados sobre todos os portos.
- **roundrobin**: Transmite dados sobre todos os portos, por sua vez.
- **activebackup**: Transmite dados sobre uma porta enquanto as outras são mantidas como backup.
- **loadbalance**: Transmite dados sobre todas as portas com balanceamento de carga Tx ativo e seletores de porta Tx baseados em Berkeley Packet Filter (BPF).

- **random**: Transmite dados em uma porta selecionada aleatoriamente.
- **lacp**: implementa o Protocolo de Controle de Agregação de Links 802.3ad (LACP).

Os serviços **teamd** usam um link observador para monitorar o estado dos dispositivos subordinados. Os seguintes vigilantes de link estão disponíveis:

- **ethtool**: A biblioteca **libteam** usa o utilitário **ethtool** para observar as mudanças de estado dos links. Este é o link-observador padrão.
- **arp_ping**: A biblioteca **libteam** usa o utilitário **arp_ping** para monitorar a presença de um endereço de hardware remoto usando o Protocolo de Resolução de Endereços (ARP).
- **nsna_ping**: Nas conexões IPv6, a biblioteca **libteam** usa os recursos de Anúncio de Vizinhança e Solicitação de Vizinhança do protocolo IPv6 Neighbor Discovery para monitorar a presença da interface de um vizinho.

Cada corredor pode usar qualquer link observador, com exceção do **lacp**. Este corredor só pode utilizar o link watcher **ethtool**.

12.5. INSTALANDO O SERVIÇO DA EQUIPE

Para configurar uma equipe de rede em **NetworkManager**, você precisa do serviço **teamd** e do plug-in da equipe para **NetworkManager**. Ambos são instalados no Red Hat Enterprise Linux 8 por default. Esta seção descreve como você instala os pacotes necessários caso você os remova.

Pré-requisitos

- Uma assinatura ativa da Red Hat é designada para o anfitrião.

Procedimento

1. Instale os pacotes **teamd** e **NetworkManager-team**:

```
# yum instalar equipe de NetworkManager teamd
```

12.6. CONFIGURAÇÃO DE UMA EQUIPE DE REDE USANDO COMANDOS NMCLI

Esta seção descreve como configurar uma equipe de rede usando o utilitário **nmcli**.

Pré-requisitos

- Dois ou mais dispositivos físicos ou virtuais de rede são instalados no servidor.
- Para usar dispositivos Ethernet como portas da equipe, os dispositivos Ethernet físicos ou virtuais devem ser instalados no servidor e conectados a um switch.
- Para usar dispositivos bond, bridge ou VLAN como portas da equipe, você pode criar estes dispositivos enquanto cria a equipe ou pode criá-los com antecedência, conforme descrito em:
  - Seção 13.5, “Configuração de uma ligação em rede usando comandos nmcli”
  - Seção 11.1, “Configuração de uma ponte de rede usando comandos nmcli”

- Seção 10.1, “Configurando a marcação VLAN usando comandos nmcli”

Procedimento

1. Criar uma interface de equipe:

```
# nmcli connection add type team con-name team0 ifname team0 team.runner activebackup
```

Este comando cria uma equipe de rede chamada **team0** que utiliza o corredor **activebackup**.

2. Opcionalmente, estabeleça um observador de ligação. Por exemplo, para definir o link watcher **ethtool** no perfil de conexão **team0**:

```
# nmcli connection modify team0 team.link-watchers "name=ethtool"
```

Os observadores de ligação suportam diferentes parâmetros. Para definir parâmetros para um observador de ligação, especifique-os separados por espaço na propriedade **name**. Observe que a propriedade do nome deve estar rodeada por aspas. Por exemplo, para usar o link watcher **ethtool** e definir seu parâmetro **delay-up** para **2500** milissegundos (2,5 segundos):

```
# nmcli connection modify team0 team.link-watchers "name=ethtool delay-up=2500"
```

Para definir vários vigilantes de ligação e cada um deles com parâmetros específicos, os vigilantes de ligação devem ser separados por uma vírgula. O exemplo a seguir define o link watcher **ethtool** com o parâmetro **delay-up** e o link watcher **arp_ping** com os parâmetros **source-host** e **target-host**:

```
# nmcli connection modify team0 team.link-watchers "name=ethtool delay-up=2, name=arp_ping source-host=192.0.2.1 target-host=192.0.2.2"
```

3. Mostre as interfaces de rede, e anote os nomes das interfaces que você deseja acrescentar à equipe:

```
# nmcli device status
DEVICE  TYPE      STATE         CONNECTION
enp7s0  ethernet  disconnected  --
enp8s0  ethernet  disconnected  --
bond0   bond      connected     bond0
bond1   bond      connected     bond1
...
```

Neste exemplo:

- **enp7s0** e **enp8s0** não estão configurados. Para usar estes dispositivos como portas, adicione perfis de conexão na próxima etapa. Observe que você só pode usar interfaces Ethernet em uma equipe que não esteja designada a nenhuma conexão.
- **bond0** e **bond1** têm perfis de conexão existentes. Para usar estes dispositivos como portas, modifique seus perfis na próxima etapa.

4. Atribuir as interfaces portuárias à equipe:

a. Se as interfaces que você deseja atribuir à equipe não estiverem configuradas, crie novos perfis de conexão para elas:

```
# nmcli connection add type ethernet slave-type team con-name team0-port1
ifname enp7s0 master team0
# nmcli connection add type ethernet slave-type team con-name team0-port2
ifname enp8s0 master team0
```

. Estes comandos criam perfis para **enp7s0** e **enp8s0**, e os adicionam à conexão **team0**.

b. Para atribuir um perfil de conexão existente à equipe, defina o parâmetro **master** dessas conexões para **team0**:

```
# nmcli connection modify bond0 master team0
# nmcli connection modify bond1 master team0
```

Estes comandos atribuem os perfis de conexão existentes denominados **bond0** e **bond1** à conexão **team0**.

5. Configurar as configurações de IP da equipe. Pule esta etapa se você quiser usar esta equipe como porta de outros dispositivos.

a. Configurar as configurações do IPv4. Por exemplo, para configurar um endereço IPv4 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS a conexão **team0**, digite:

```
# nmcli connection modify team0 ipv4.addresses '192.0.2.1/24'
# nmcli connection modify team0 ipv4.gateway '192.0.2.254'
# nmcli connection modify team0 ipv4.dns '192.0.2.253'
# nmcli connection modify team0 ipv4.dns-search 'example.com'
# nmcli connection modify team0 ipv4.method manual
```

b. Configurar as configurações IPv6. Por exemplo, para configurar um endereço IPv6 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS da conexão **team0**, digite:

```
# nmcli connection modify team0 ipv6.addresses '2001:db8:1::1/64'
# nmcli connection modify team0 ipv6.gateway '2001:db8:1::fffe'
# nmcli connection modify team0 ipv6.dns '2001:db8:1::fffd'
# nmcli connection modify team0 ipv6.dns-search 'example.com'
# nmcli connection modify team0 ipv6.method manual
```

6. Ativar a conexão:

```
# nmcli connection up team0
```

Etapas de verificação

- Mostrar o status da equipe:

```
# teamdctl team0 state
setup:
  runner: activebackup
ports:
  enp7s0
    link watches:
      link summary: up
```

```
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
    enp8s0
      link watches:
        link summary: up
        instance[link_watch_0]:
          name: ethtool
          link: up
          down count: 0
runner:
  active port: enp7s0
```

Neste exemplo, ambos os portos estão em alta.

Recursos adicionais

- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, "Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão".
- Seção 12.4, "Entendendo o serviço da equipe, corredores e vigilantes de ligação" .
- Para exemplos em **nmcli**, veja a página de manual **nmcli-examples(7)**.
- Para todas as propriedades da equipe que você pode definir, consulte a seção **team** na página de manual **nm-settings(5)**.
- Para parâmetros que você pode definir na configuração do JSON, assim como exemplos do JSON, veja a página de manual **teamd.conf(5)**.

12.7. CONFIGURAÇÃO DE UMA EQUIPE DE REDE USANDO UM EDITOR DE NM-CONEXÃO

Esta seção descreve como você configura uma equipe de rede usando o aplicativo **nm-connection-editor**.

Note que **nm-connection-editor** pode adicionar apenas novos portos a uma equipe. Para usar um perfil de conexão existente como uma porta, crie a equipe usando o utilitário **nmcli**, conforme descrito em Seção 12.6, "Configuração de uma equipe de rede usando comandos nmcli" .

Pré-requisitos

- Dois ou mais dispositivos físicos ou virtuais de rede são instalados no servidor.
- Para usar dispositivos Ethernet como portas da equipe, os dispositivos Ethernet físicos ou virtuais devem ser instalados no servidor.
- Para usar dispositivos de equipe, bond, ou VLAN como portas da equipe, garantir que estes dispositivos ainda não estejam configurados.

Procedimento

1. Abra um terminal e entre em **nm-connection-editor**:

```
$ nm-connection-editor
```

2. Clique no botão para adicionar uma nova conexão.

3. Selecione o tipo de conexão **Team**, e clique em **Criar**.

4. Na aba **Team**:

   a. Opcional: Defina o nome da interface da equipe no campo **Interface name**.

   b. Clique no botão **Adicionar** para adicionar um novo perfil de conexão para uma interface de rede e adicionar o perfil como uma porta para a equipe.

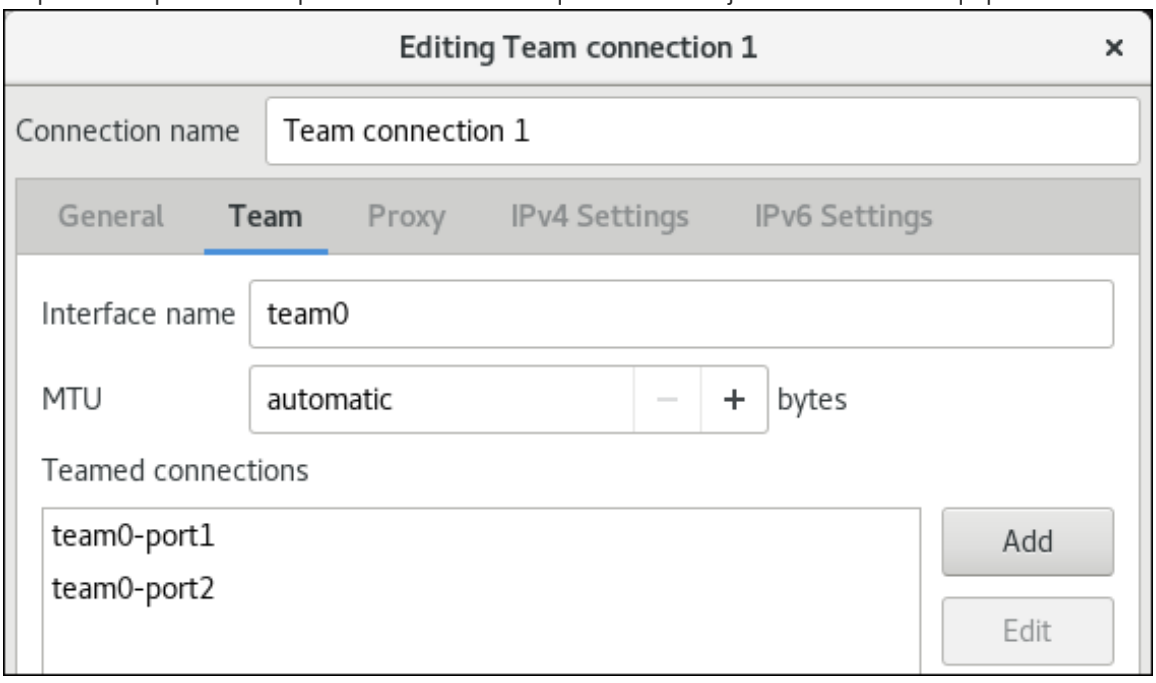
      i. Selecione o tipo de conexão da interface. Por exemplo, selecione **Ethernet** para uma conexão com fio.

      ii. Opcional: Defina um nome de conexão para o porto.

      iii. Se você criar um perfil de conexão para um dispositivo Ethernet, abra a aba **Ethernet**, e selecione no campo **Device** a interface de rede que você deseja adicionar como porta à equipe. Se você selecionou um tipo de dispositivo diferente, configure-o de acordo. Observe que você só pode usar interfaces Ethernet em uma equipe que não esteja designada a nenhuma conexão.

      iv. Clique em **Salvar**.

   c. Repita a etapa anterior para cada interface que você deseja acrescentar à equipe.



   d. Clique no botão **Avançado** para definir opções avançadas para a conexão da equipe.

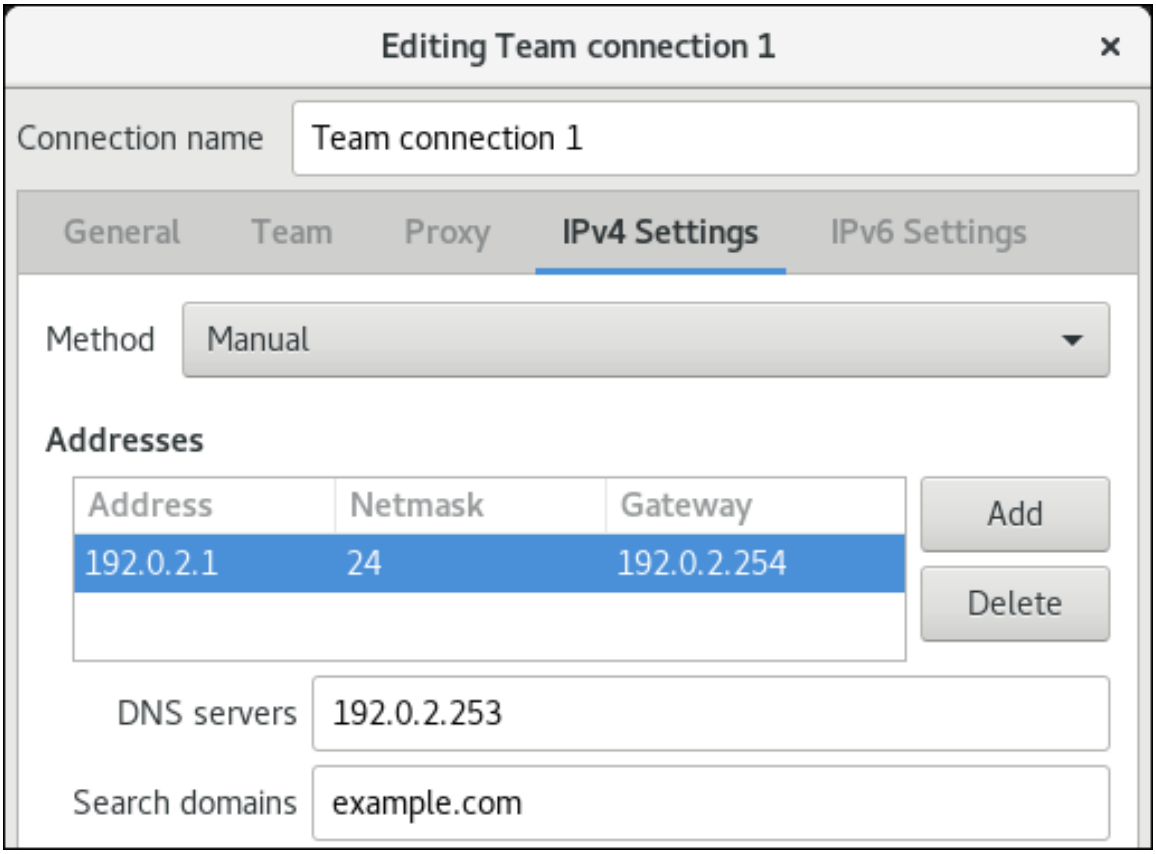
      i. Na aba **Runner**, selecione a corrediça.

      ii. Na aba **Link Watcher**, defina o link watcher e suas configurações opcionais.

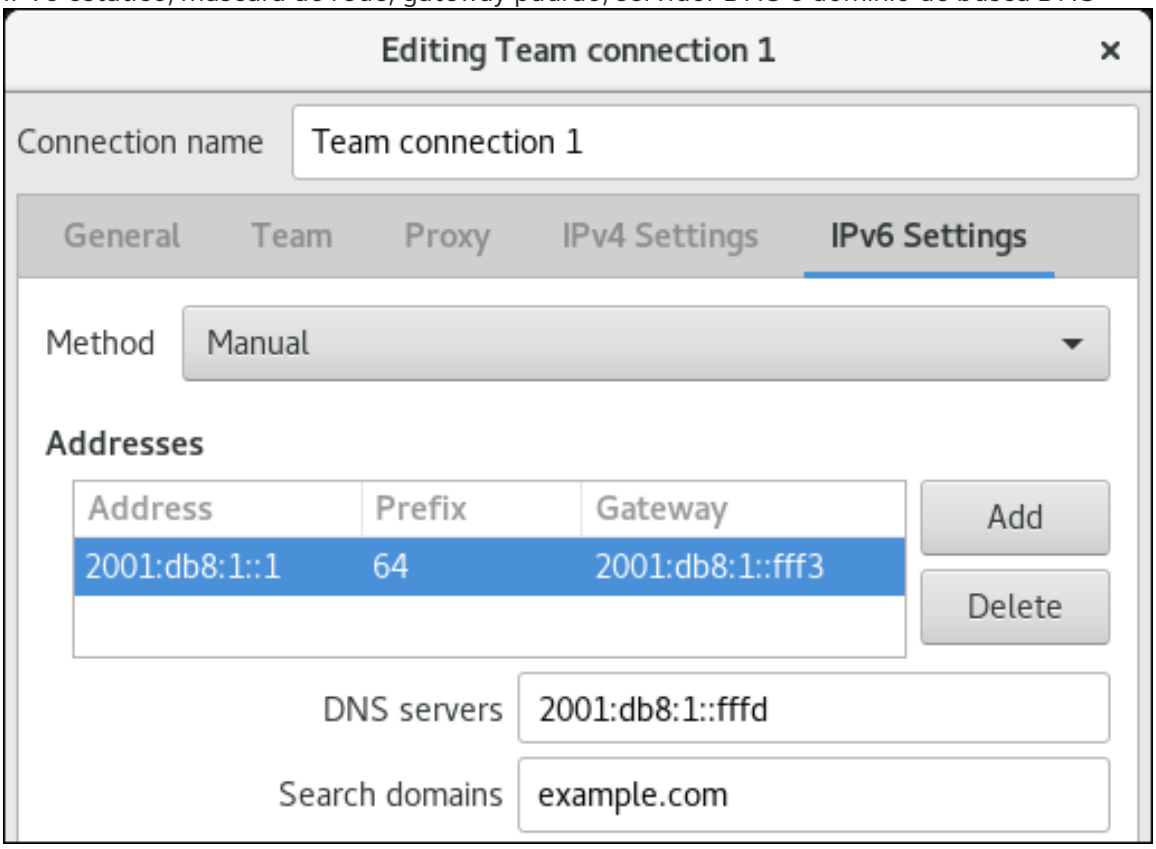
      iii. Clique **OK**.

5. Configurar as configurações de IP da equipe. Pule esta etapa se você quiser usar esta equipe como porta de outros dispositivos.

   a. Na aba **IPv4 Settings**, configure as configurações do IPv4. Por exemplo, defina um endereço IPv4 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS



   b. Na aba **IPv6 Settings**, configure as configurações IPv6. Por exemplo, defina um endereço IPv6 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS



6. Salvar a conexão da equipe.

7. Fechar **nm-connection-editor**.

Etapas de verificação

- Mostrar o status da equipe:

```
# teamdctl team0 state
setup:
  runner: activebackup
ports:
  enp7s0
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
  enp8s0
    link watches:
      link summary: up
      instance[link_watch_0]:
        name: ethtool
        link: up
        down count: 0
runner:
  active port: enp7s0
```

Recursos adicionais

- Seção 13.6, “Configuração de uma ligação de rede usando um editor de nm-conexão”
- Seção 11.2, “Configuração de uma ponte de rede usando um editor de conexão nm”
- Seção 10.2, “Configuração da marcação de VLAN usando o editor de nm-conexão”
- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, “Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão”.
- Seção 12.4, “Entendendo o serviço da equipe, corredores e vigilantes de ligação” .
- Se a configuração no disco não corresponder à configuração no dispositivo, iniciar ou reiniciar o NetworkManager cria uma conexão in-memory que reflete a configuração do dispositivo. Para maiores detalhes e como evitar este problema, veja NetworkManager duplica uma conexão após o reinício do serviço NetworkManager.

CAPÍTULO 13. CONFIGURANDO A LIGAÇÃO EM REDE

Esta seção descreve o básico da ligação em rede, as diferenças entre ligação e equipe, e como configurar uma ligação em rede no Red Hat Enterprise Linux 8.

Você pode criar vínculos em diferentes tipos de dispositivos, como por exemplo:

- Dispositivos Ethernet físicos e virtuais
- Pontes de rede
- Equipes de rede
- Dispositivos VLAN

13.1. ENTENDENDO A LIGAÇÃO EM REDE

A ligação de rede é um método para combinar ou agregar interfaces de rede para fornecer uma interface lógica com maior rendimento ou redundância.

Os modos **active-backup**, **balance-tlb**, e **balance-alb** não exigem nenhuma configuração específica do switch de rede. Entretanto, outros modos de ligação exigem a configuração do switch para agregar os links. Por exemplo, os switches Cisco requerem **EtherChannel** para os modos 0, 2, e 3, mas para o modo 4, o Protocolo de Controle de Agregação de Links (LACP) e **EtherChannel** são necessários.

Para mais detalhes, veja a documentação de seu switch e do Linux Ethernet Bonding Driver HOWTO .

IMPORTANTE

Certas características de ligação de rede, como o mecanismo de fail-over, não suportam conexões diretas de cabos sem um switch de rede. Para mais detalhes, veja a seção É a colagem suportada com conexão direta utilizando cabos cruzados? A solução KCS.

13.2. ENTENDENDO O COMPORTAMENTO PADRÃO DO CONTROLADOR E DAS INTERFACES DE PORTA

Considere o seguinte comportamento padrão, ao gerenciar ou solucionar problemas de equipe ou interfaces de portas de vínculo usando o serviço **NetworkManager**:

- O início da interface do controlador não inicia automaticamente as interfaces de porta.
- Iniciar uma interface de porta sempre inicia a interface do controlador.
- A parada da interface do controlador também pára a interface da porta.
- Um controlador sem portas pode iniciar conexões IP estáticas.
- Um controlador sem portas espera por portas ao iniciar as conexões DHCP.
- Um controlador com uma conexão DHCP à espera de portas se completa quando você adiciona uma porta com um transportador.
- Um controlador com uma conexão DHCP esperando por portas continua esperando quando você adiciona uma porta sem transportador.

13.3. COMPARAÇÃO ENTRE AS CARACTERÍSTICAS DE EQUIPE DE REDE E DE LIGAÇÃO

Conheça os recursos suportados em equipes de rede e vínculos de rede:

Destaque	Ligação em rede	Equipe da rede
Política de Tx Broadcast	Sim	Sim
Política do Tx Round-robin	Sim	Sim
Política de Tx backup ativo	Sim	Sim
Suporte LACP (802.3ad)	Sim (somente ativo)	Sim
Política Tx baseada em hastes	Sim	Sim
O usuário pode definir a função hash	Não	Sim
Tx load-balancing support (TLB)	Sim	Sim
LACP porto hash selecionar	Sim	Sim
Balanceamento de carga para suporte LACP	Não	Sim
Monitoramento do Ethtool link	Sim	Sim
Monitoramento do link ARP	Sim	Sim
Monitoramento de links NS/NA (IPv6)	Não	Sim
Atrasos nas portas para cima/para baixo	Sim	Sim
Prioridades portuárias e aderência (opção "primária" de melhoria)	Não	Sim
Configuração de monitoramento separado por link de porta	Não	Sim
Configuração de monitoramento de múltiplos links	Limitado	Sim
Caminho Tx/Rx sem fechadura	Não (rwlock)	Sim (RCU)

Destaque	Ligação em rede	Equipe da rede
Suporte VLAN	Sim	Sim
Controle do tempo de execução do espaço do usuário	Limitado	Sim
Lógica no espaço do usuário	Não	Sim
Extensibilidade	Difícil	Fácil
Design modular	Não	Sim
Despesas gerais de desempenho	Baixo	Muito baixo
Interface D-Bus	Não	Sim
Empilhamento de múltiplos dispositivos	Sim	Sim
Configuração zero usando LLDP	Não	(no planejamento)
Apoio ao NetworkManager	Sim	Sim

13.4. CONFIGURAÇÃO DO SWITCH UPSTREAM DEPENDENDO DOS MODOS DE LIGAÇÃO

A tabela a seguir descreve quais configurações você deve aplicar ao interruptor a montante, dependendo do modo de ligação:

Modo de colagem	Configuração no interruptor
0 - **balance-rr**	Requer Etherchannel estático ativado (não negociado com o LACP)
1 - **active-backup**	Requer portos autônomos
2 - **balance-xor**	Requer Etherchannel estático ativado (não negociado com o LACP)
3 - **broadcast**	Requer Etherchannel estático ativado (não negociado com o LACP)
4 - **802.3ad**	Requer LACP-negociado Etherchannel habilitado
5 - **balance-tlb**	Requer portos autônomos
6 - **balance-alb**	Requer portos autônomos

Para configurar estas configurações em seu switch, consulte a documentação do switch.

13.5. CONFIGURAÇÃO DE UMA LIGAÇÃO EM REDE USANDO COMANDOS NMCLI

Esta seção descreve como configurar um vínculo de rede usando os comandos **nmcli**.

Pré-requisitos

- Dois ou mais dispositivos físicos ou virtuais de rede são instalados no servidor.
- Para usar dispositivos Ethernet como portas da ligação, os dispositivos Ethernet físicos ou virtuais devem ser instalados no servidor.
- Para usar dispositivos de equipe, ponte ou VLAN como portas da ligação, você pode criar estes dispositivos enquanto cria a ligação ou pode criá-los com antecedência, conforme descrito em:
  - Seção 12.6, “Configuração de uma equipe de rede usando comandos nmcli”
  - Seção 11.1, “Configuração de uma ponte de rede usando comandos nmcli”
  - Seção 10.1, “Configurando a marcação VLAN usando comandos nmcli”

Procedimento

1. Criar uma interface de vínculo:

   ```
   # nmcli connection add type bond con-name bond0 ifname bond0 bond.options "mode=active-backup"
   ```

   Este comando cria um vínculo chamado **bond0** que usa o modo **active-backup**.

   Para definir adicionalmente um intervalo de monitoramento da Interface Independente de Mídia (MII), acrescente o **miimon=*interval*** opção para a propriedade **bond.options**. Por exemplo, para usar o mesmo comando mas, adicionalmente, definir o intervalo de monitoramento MII para **1000** milissegundos (1 segundo), entrar:

   ```
   # nmcli connection add type bond con-name bond0 ifname bond0 bond.options "mode=active-backup,miimon=1000"
   ```

2. Mostrar as interfaces de rede, e anotar os nomes das interfaces que você planeja acrescentar ao vínculo:

   ```
   # nmcli device status
   DEVICE   TYPE      STATE         CONNECTION
   enp7s0   ethernet  disconnected  --
   enp8s0   ethernet  disconnected  --
   bridge0  bridge    connected     bridge0
   bridge1  bridge    connected     bridge1
   ...
   ```

   Neste exemplo:

- **enp7s0** e **enp8s0** não estão configurados. Para usar estes dispositivos como portas, adicione perfis de conexão na próxima etapa.
- **bridge0** e **bridge1** têm perfis de conexão existentes. Para usar estes dispositivos como portas, modifique seus perfis na próxima etapa.

3. Atribuir interfaces para a ligação:

   a. Se as interfaces que você deseja atribuir ao vínculo não estiverem configuradas, crie novos perfis de conexão para elas:

   ```
   # nmcli connection add type ethernet slave-type bond con-name bond0-port1 ifname enp7s0 master bond0
   # nmcli connection add type ethernet slave-type bond con-name bond0-port2 ifname enp8s0 master bond0
   ```

   Estes comandos criam perfis para **enp7s0** e **enp8s0**, e os adicionam à conexão **bond0**.

   b. Para atribuir um perfil de conexão existente ao vínculo, defina o parâmetro **master** dessas conexões para **bond0**:

   ```
   # nmcli connection modify bridge0 master bond0
   # nmcli connection modify bridge1 master bond0
   ```

   Estes comandos atribuem os perfis de conexão existentes denominados **bridge0** e **bridge1** à conexão **bond0**.

4. Configurar as configurações de IP do vínculo. Pule esta etapa se você quiser usar este vínculo como porta de outros dispositivos.

   a. Configurar as configurações do IPv4. Por exemplo, para configurar um endereço IPv4 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS para a conexão **bond0**, digite:

   ```
   # nmcli connection modify bond0 ipv4.addresses '192.0.2.1/24'
   # nmcli connection modify bond0 ipv4.gateway '192.0.2.254'
   # nmcli connection modify bond0 ipv4.dns '192.0.2.253'
   # nmcli connection modify bond0 ipv4.dns-search 'example.com'
   # nmcli connection modify bond0 ipv4.method manual
   ```

   b. Configurar as configurações IPv6. Por exemplo, para configurar um endereço IPv6 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS para a conexão **bond0**, digite:

   ```
   # nmcli connection modify bond0 ipv6.addresses '2001:db8:1::1/64
   # nmcli connection modify bond0 ipv6.gateway '2001:db8:1::fffe'
   # nmcli connection modify bond0 ipv6.dns '2001:db8:1::fffd'
   # nmcli connection modify bond0 ipv6.dns-search 'example.com'
   # nmcli connection modify bond0 ipv6.method manual
   ```

5. Ativar a conexão:

   ```
   # nmcli connection up bond0
   ```

6. Verifique se os portos estão conectados, e a coluna **CONNECTION** mostra o nome da conexão do porto:

```
# nmcli device
DEVICE   TYPE     STATE      CONNECTION
...
enp7s0   ethernet  connected  bond0-port1
enp8s0   ethernet  connected  bond0-port2
```

O Red Hat Enterprise Linux ativa o controlador e os dispositivos de porta quando o sistema é inicializado. Ativando qualquer conexão de porta, o controlador também é ativado. Entretanto, neste caso, apenas uma conexão de porta é ativada. Por default, a ativação do controlador não ativa automaticamente as portas. No entanto, é possível ativar este comportamento através da configuração:

a. Habilitar o parâmetro **connection.autoconnect-slaves** da conexão do vínculo:

```
# nmcli connection modify bond0 connection.autoconnect-slaves 1
```

b. Reativar a ponte:

```
# nmcli connection up bond0
```

Etapas de verificação

1. Mostrar o status do vínculo:

```
# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: enp7s0
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: enp7s0
MII Status: up
Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 52:54:00:d5:e0:fb
Slave queue ID: 0

Slave Interface: enp8s0
MII Status: up
Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 52:54:00:b2:e2:63
Slave queue ID: 0
```

Neste exemplo, ambos os portos estão em alta.

2. Para verificar se o failover da colagem funciona:

   a. Remover temporariamente o cabo de rede do host. Note que não há nenhum método para testar adequadamente os eventos de falha do link usando a linha de comando.

   b. Mostrar o status do vínculo:

   ```
   # cat /proc/net/bonding/bond0
   ```

Recursos adicionais

- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, "Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão".
- Para exemplos em **nmcli**, veja a página de manual **nmcli-examples(7)**.
- Para obter uma lista de opções que você pode definir no parâmetro **bond.options** do comando **nmcli** ao criar um vínculo, consulte https://www.kernel.org/doc/Documentation/networking/bonding.txt.

13.6. CONFIGURAÇÃO DE UMA LIGAÇÃO DE REDE USANDO UM EDITOR DE NM-CONEXÃO

Esta seção descreve como configurar um vínculo de rede usando a aplicação **nm-connection-editor**.

Note que **nm-connection-editor** pode adicionar apenas novos portos a um vínculo. Para usar um perfil de conexão existente como uma porta, crie o vínculo usando o utilitário **nmcli** como descrito em Seção 13.5, "Configuração de uma ligação em rede usando comandos nmcli" .

Pré-requisitos

- Dois ou mais dispositivos físicos ou virtuais de rede são instalados no servidor.
- Para usar dispositivos Ethernet como portas da ligação, os dispositivos Ethernet físicos ou virtuais devem ser instalados no servidor.
- Para usar dispositivos de equipe, bond, ou VLAN como portas do bond, garantir que estes dispositivos ainda não estejam configurados.

Procedimento

1. Abra um terminal e entre em **nm-connection-editor**:

   ```
   $ nm-connection-editor
   ```

2. Clique no botão para adicionar uma nova conexão.
3. Selecione o tipo de conexão **Bond**, e clique em **Criar**.
4. Na aba **Bond**:

   a. Opcional: Defina o nome da interface do vínculo no campo **Interface name**.

b. Clique no botão **Adicionar** para adicionar uma interface de rede como uma porta para o vínculo.

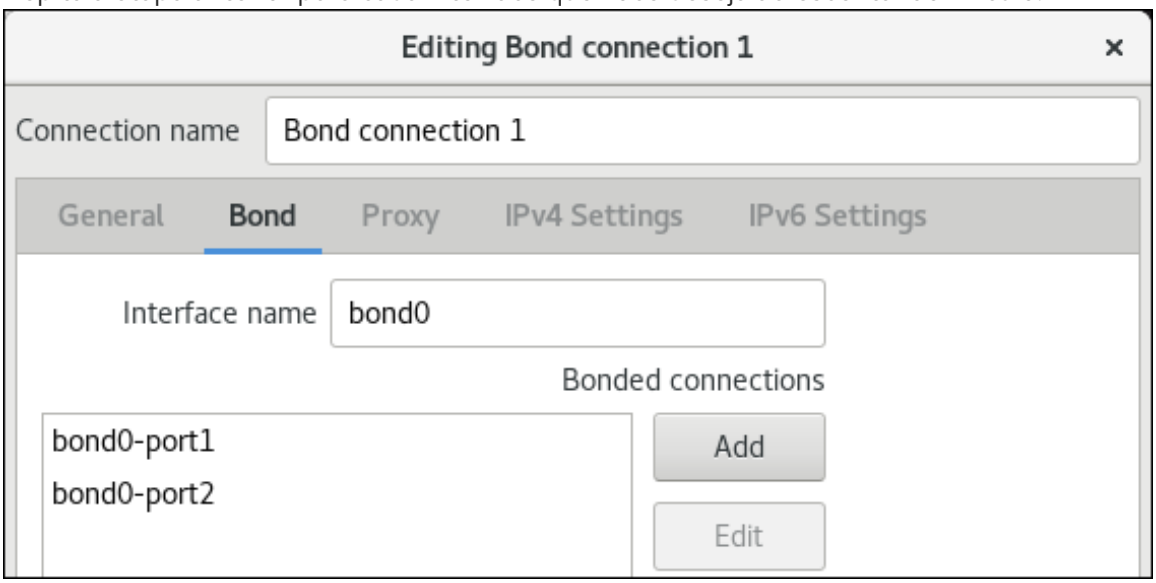
   i. Selecione o tipo de conexão da interface. Por exemplo, selecione **Ethernet** para uma conexão com fio.

   ii. Opcional: Defina um nome de conexão para o porto.

   iii. Se você criar um perfil de conexão para um dispositivo Ethernet, abra a aba **Ethernet**, e selecione no campo **Device** a interface de rede que você deseja adicionar como porta ao vínculo. Se você selecionou um tipo de dispositivo diferente, configure-o de acordo. Observe que você só pode usar interfaces Ethernet em um vínculo que não esteja configurado.

   iv. Clique em **Salvar**.

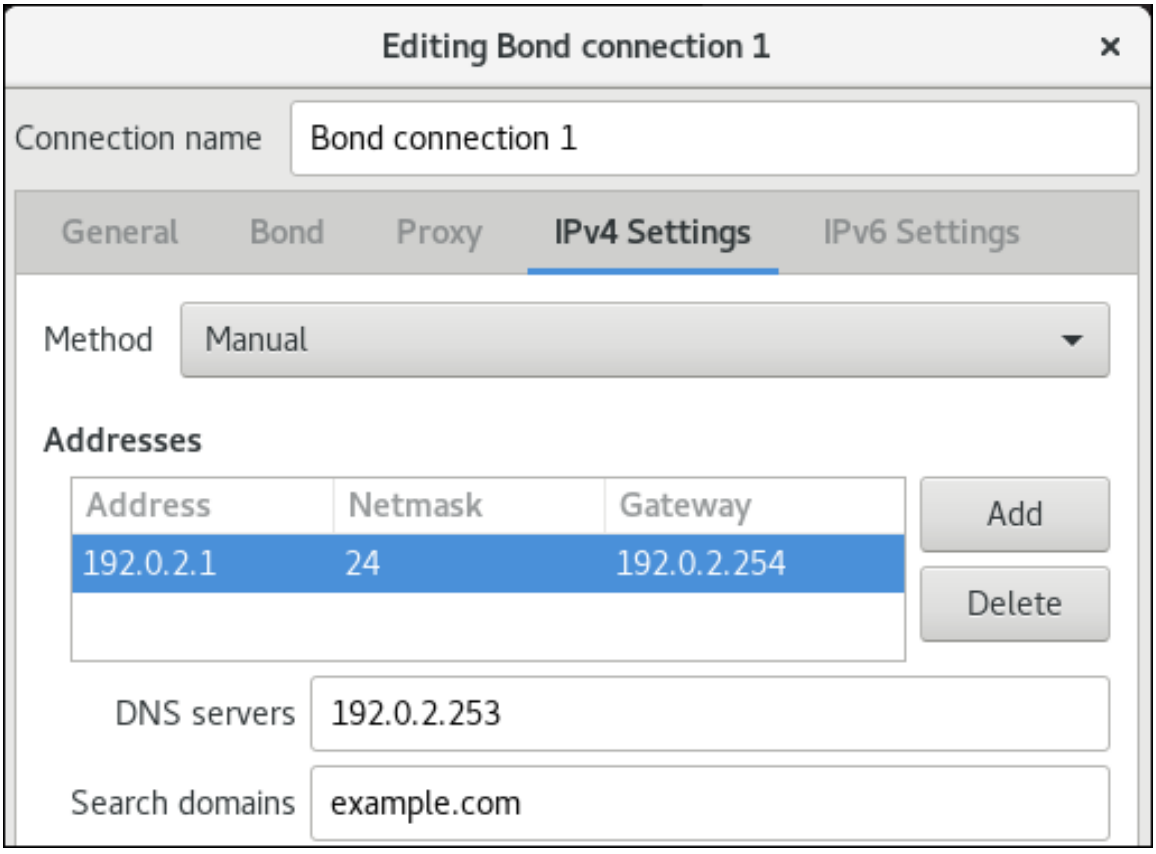
c. Repita a etapa anterior para cada interface que você deseja acrescentar ao vínculo:



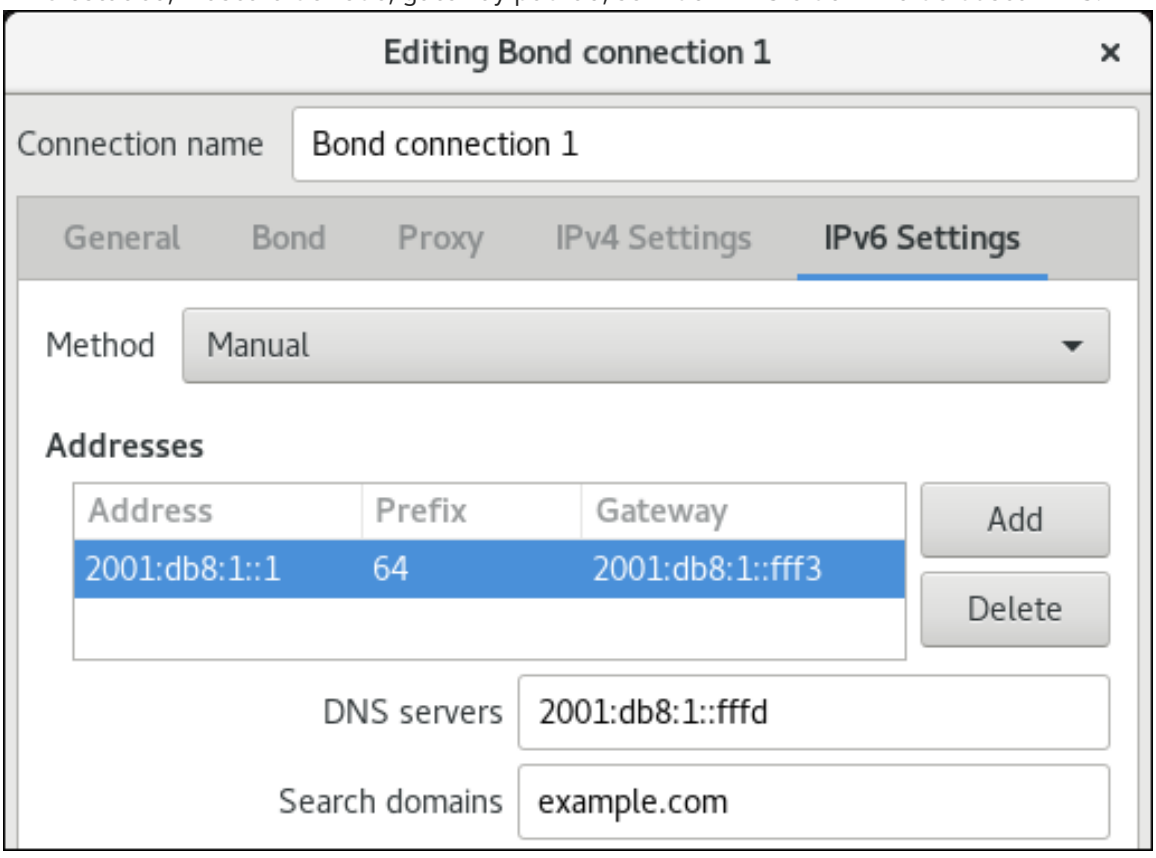
d. Opcional: Defina outras opções, tais como o intervalo de monitoramento da Interface Independente de Mídia (MII).

5. Configurar as configurações de IP do vínculo. Pule esta etapa se você quiser usar este vínculo como porta de outros dispositivos.

   a. Na aba **IPv4 Settings**, configure as configurações do IPv4. Por exemplo, defina um endereço IPv4 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS:



b. Na aba **IPv6 Settings**, configure as configurações IPv6. Por exemplo, defina um endereço IPv6 estático, máscara de rede, gateway padrão, servidor DNS e domínio de busca DNS:



6. Clique em **Salvar** para salvar a conexão do vínculo.
7. Fechar **nm-connection-editor**.

Etapas de verificação

- Veja o status do vínculo:

```
$ cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: fault-tolerance (active-backup)
Primary Slave: None
Currently Active Slave: enp7s0
MII Status: up
MII Polling Interval (ms): 100
Up Delay (ms): 0
Down Delay (ms): 0

Slave Interface: enp7s0
MII Status: up
Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 52:54:00:d5:e0:fb
Slave queue ID: 0

Slave Interface: enp8s0
MII Status: up
Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 52:54:00:b2:e2:63
Slave queue ID: 0
```

Neste exemplo, ambos os portos estão em alta.

Recursos adicionais

- Seção 12.7, "Configuração de uma equipe de rede usando um editor de nm-conexão"
- Seção 11.2, "Configuração de uma ponte de rede usando um editor de conexão nm"
- Seção 10.2, "Configuração da marcação de VLAN usando o editor de nm-conexão"
- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Se a conexão não tiver um gateway padrão, veja Seção 18.8, "Configuração do NetworkManager para evitar o uso de um perfil específico para fornecer um gateway padrão".

13.7. CONFIGURAÇÃO DE UM VÍNCULO DE REDE USANDO AS FUNÇÕES DO SISTEMA RHEL

Você pode usar a função do Sistema RHEL **network** para configurar um vínculo de rede. Este procedimento descreve como configurar um vínculo em modo de backup ativo que usa dois dispositivos Ethernet e define um endereço IPv4 e IPv6, gateways padrão e configuração DNS.

NOTA

Defina a configuração IP na ponte e não nas portas da ponte Linux.

Pré-requisitos

- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.
- Se você usar um usuário remoto diferente de **root** ao executar o playbook, este usuário tem as permissões apropriadas **sudo** no nó gerenciado.
- Dois ou mais dispositivos físicos ou virtuais de rede são instalados no servidor.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo **/etc/ansible/hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook **~/bond-ethernet.yml** com o seguinte conteúdo:

```
---
- name: Configure a network bond that uses two Ethernet ports
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network

    vars:
      network_connections:
        # Define the bond profile
        - name: bond0
          type: bond
          interface_name: bond0
          ip:
            address:
              - "192.0.2.1/24"
              - "2001:db8:1::1/64"
            gateway4: 192.0.2.254
            gateway6: 2001:db8:1::fffe
            dns:
              - 192.0.2.200
              - 2001:db8:1::ffbb
            dns_search:
              - example.com
          bond:
            mode: active-backup
          state: up

        # Add an Ethernet profile to the bond
        - name: bond0-port1
          interface_name: enp7s0
          type: ethernet
          master: bond0
          state: up

        # Add a second Ethernet profile to the bond
```

```
    - name: bond0-port2
      interface_name: enp8s0
      type: ethernet
      master: bond0
      state: up
```

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

     ```
     # ansible-playbook -u root ~/bond-ethernet.yml
     ```

   - Para conectar-se como usuário ao host administrado, entre:

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/bond-ethernet.yml
     ```

     A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definido no **-u *user_name*** opção.

   Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Recursos adicionais

- Para detalhes sobre os parâmetros usados em **network_connections** e para informações adicionais sobre o Sistema de Papel **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

13.8. CRIAÇÃO DE UMA LIGAÇÃO DE REDE PARA PERMITIR A COMUTAÇÃO ENTRE UMA CONEXÃO ETHERNET E SEM FIO SEM INTERROMPER A VPN

Os usuários da RHEL que conectam sua estação de trabalho à rede de sua empresa normalmente usam uma VPN para acessar recursos remotos. Entretanto, se a estação de trabalho comutar entre uma conexão Ethernet e Wi-Fi, por exemplo, se você liberar um laptop de uma estação de acoplamento com uma conexão Ethernet, a conexão VPN é interrompida. Para evitar este problema, você pode criar uma ligação de rede que usa a conexão Ethernet e Wi-Fi no modo **active-backup**.

Pré-requisitos

- O host contém uma Ethernet e um dispositivo Wi-Fi.
- Foi criado um perfil de conexão Ethernet e Wi-Fi NetworkManager e ambas as conexões funcionam independentemente.
  Este procedimento utiliza os seguintes perfis de conexão para criar um vínculo de rede chamado **bond0**:
  - **Docking_station** associado com o dispositivo Ethernet **enp11s0u1**
  - **Wi-Fi** associado com o dispositivo Wi-Fi **wlp61s0**

Procedimento

1. Criar uma interface de vínculo no modo **active-backup**:

```
# nmcli connection add type bond con-name bond0 ifname bond0 bond.options
"mode=active-backup"
```

   Este comando nomeia tanto a interface quanto o perfil de conexão **bond0**.

2. Configurar as configurações IPv4 do vínculo:

   - Se um servidor DHCP em sua rede atribui endereços IPv4 a hosts, nenhuma ação é necessária.
   - Se sua rede local requer endereços IPv4 estáticos, defina o endereço, máscara de rede, gateway padrão, servidor DNS, e domínio de busca DNS para a conexão **bond0**:

```
# nmcli connection modify bond0 ipv4.addresses '192.0.2.1/24'
# nmcli connection modify bond0 ipv4.gateway '192.0.2.254'
# nmcli connection modify bond0 ipv4.dns '192.0.2.253'
# nmcli connection modify bond0 ipv4.dns-search 'example.com'
# nmcli connection modify bond0 ipv4.method manual
```

3. Configurar as configurações IPv6 do vínculo:

   - Se seu roteador ou um servidor DHCP em sua rede atribui endereços IPv6 a hosts, nenhuma ação é necessária.
   - Se sua rede local requer endereços IPv6 estáticos, defina o endereço, máscara de rede, gateway padrão, servidor DNS, e domínio de busca DNS para a conexão **bond0**:

```
# nmcli connection modify bond0 ipv6.addresses '2001:db8:1::1/64'
# nmcli connection modify bond0 ipv6.gateway '2001:db8:1::fffe'
# nmcli connection modify bond0 ipv6.dns '2001:db8:1::fffd'
# nmcli connection modify bond0 ipv6.dns-search 'example.com'
# nmcli connection modify bond0 ipv6.method manual
```

4. Exibir os perfis de conexão:

```
# nmcli connection show
NAME            UUID                                  TYPE      DEVICE
Docking_station  256dd073-fecc-339d-91ae-9834a00407f9  ethernet  enp11s0u1
Wi-Fi           1f1531c7-8737-4c60-91af-2d21164417e8  wifi      wlp61s0
...
```

   Você requer os nomes dos perfis de conexão e o nome do dispositivo Ethernet nas próximas etapas.

5. Atribuir o perfil de conexão da conexão Ethernet ao vínculo:

```
# nmcli connection modify Docking_station master bond0
```

6. Atribuir o perfil de conexão da conexão Wi-Fi ao vínculo:

```
# nmcli connection modify Wi-Fi master bond0
```

7. Se sua rede Wi-Fi usa filtragem MAC para permitir somente endereços MAC em uma lista de permissão de acesso à rede, configure esse NetworkManager para atribuir dinamicamente o endereço MAC da porta ativa à ligação:

```
# nmcli connection modify bond0 bond.options fail_over_mac=1
```

   Com esta configuração, você deve definir apenas o endereço MAC do dispositivo Wi-Fi para a lista de permissão em vez do endereço MAC do dispositivo Ethernet e Wi-Fi.

8. Configure o dispositivo associado à conexão Ethernet como dispositivo principal da ligação:

```
# nmcli con modify bond0 bond.options "primary=enp11s0u1"
```

   Com este ajuste, a ligação sempre usa a conexão Ethernet, se ela estiver disponível.

9. Configure o NetworkManager para ativar automaticamente as portas quando o dispositivo **bond0** for ativado:

```
# nmcli connection modify bond0 connection.autoconnect-slaves 1
```

10. Ativar a conexão **bond0**:

```
# nmcli connection up bond0
```

Etapas de verificação

- Mostrar o dispositivo atualmente ativo, o status do vínculo e seus portos:

```
# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v3.7.1 (April 27, 2011)

Bonding Mode: fault-tolerance (active-backup) (fail_over_mac active)
Primary Slave: enp11s0u1 (primary_reselect always)
Currently Active Slave: enp11s0u1
MII Status: up
MII Polling Interval (ms): 1
Up Delay (ms): 0
Down Delay (ms): 0
Peer Notification Delay (ms): 0

Slave Interface: enp11s0u1
MII Status: up
Speed: 1000 Mbps
Duplex: full
Link Failure Count: 0
Permanent HW addr: 00:53:00:59:da:b7
Slave queue ID: 0

Slave Interface: wlp61s0
MII Status: up
Speed: Unknown
Duplex: Unknown
```

```
Link Failure Count: 2
Permanent HW addr: 00:53:00:b3:22:ba
Slave queue ID: 0
```

Recursos adicionais

- Capítulo 8, *Configuração de uma conexão Ethernet*
- Capítulo 9, *Gerenciando conexões Wi-Fi*
- Capítulo 13, *Configurando a ligação em rede*

CAPÍTULO 14. CONFIGURAÇÃO DE UMA CONEXÃO VPN

Esta seção explica como configurar uma conexão de rede privada virtual (VPN).

Uma VPN é uma forma de conexão a uma rede local pela Internet. **IPsec** fornecido por **Libreswan** é o método preferido para criar uma VPN. **Libreswan** é um espaço do usuário **IPsec** implementação para VPN. Uma VPN permite a comunicação entre sua LAN, e outra LAN remota, através da criação de um túnel através de uma rede intermediária, como a Internet. Por razões de segurança, um túnel VPN sempre usa autenticação e criptografia. Para operações criptográficas, **Libreswan** utiliza a biblioteca **NSS**.

14.1. CONFIGURAÇÃO DE UMA CONEXÃO VPN COM O CENTRO DE CONTROLE

Este procedimento descreve como configurar uma conexão VPN usando **control-center**.

Pré-requisitos

- O pacote **NetworkManager-libreswan-gnome** está instalado.

Procedimento

1. Pressione a tecla **Super**, digite **Settings**, e pressione **Enter** para abrir o aplicativo **control-center**.
2. Selecione a entrada **Network** à esquerda.
3. Clique no ícone .
4. Selecione **VPN**.
5. Selecione a entrada do menu **Identity** para ver as opções básicas de configuração:
   General

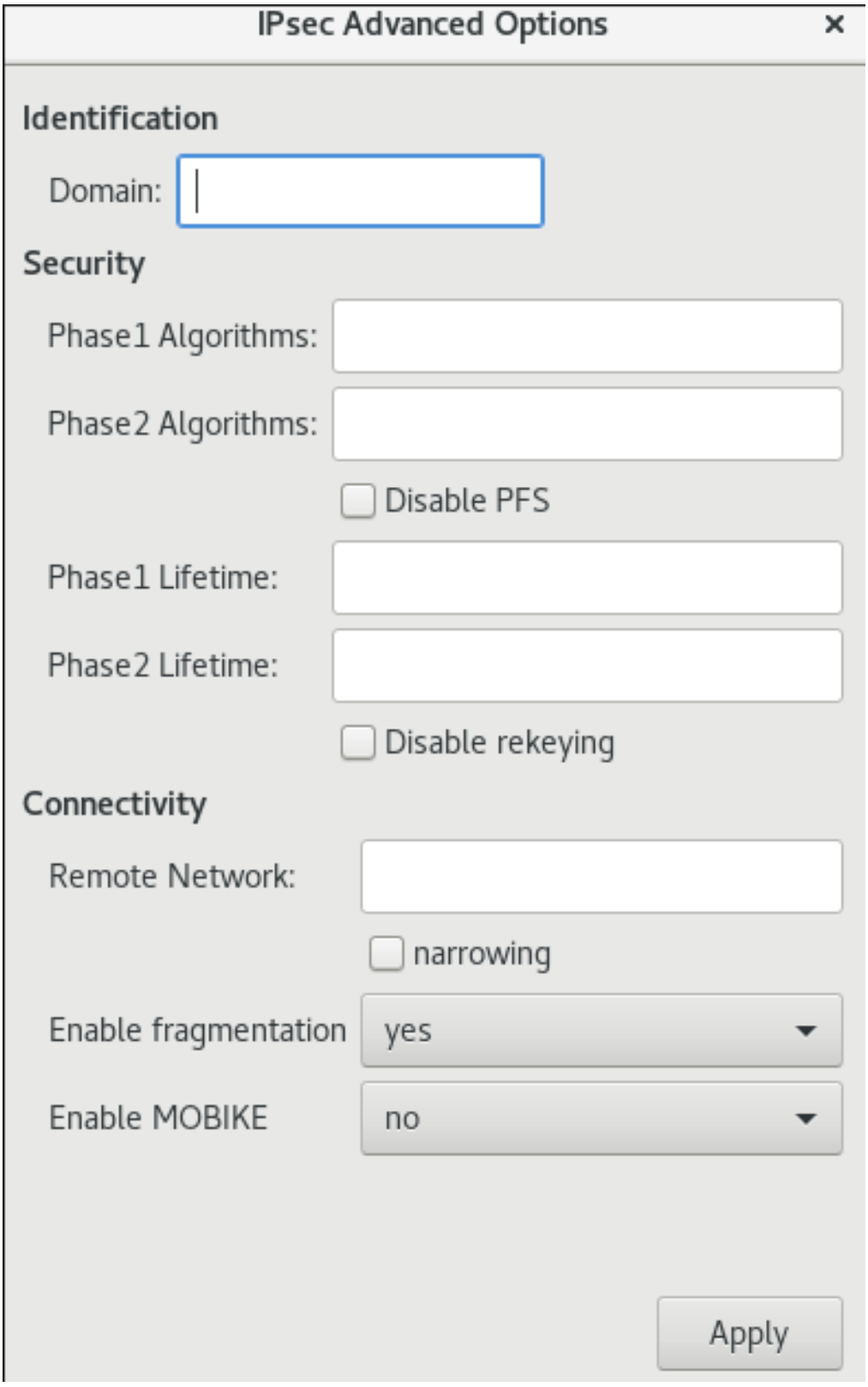
   Gateway - O nome ou endereço **IP** do gateway VPN remoto.

   Authentication

   Type

   - **IKEv2 (Certificate)**- o cliente é autenticado por certificado. É mais seguro (padrão).
   - **IKEv1 (XAUTH)** - o cliente é autenticado por nome de usuário e senha, ou por uma chave pré-compartilhada (PSK).
     Os seguintes ajustes de configuração estão disponíveis na seção **Advanced**:

Figura 14.1. Opções avançadas de uma conexão VPN



ATENÇÃO

Ao configurar uma conexão VPN baseada em IPsec usando o aplicativo **gnome-control-center**, o diálogo **Advanced** exibe a configuração, mas não permite nenhuma mudança. Como conseqüência, os usuários não podem alterar nenhuma opção IPsec avançada. Utilize as ferramentas **nm-connection-editor** ou **nmcli** para realizar a configuração das propriedades avançadas.

Identification

- **Domain** - Se necessário, digite o Nome do Domínio.

Security

- **Phase1 Algorithms** - corresponde ao parâmetro **ike** Libreswan - digite os algoritmos a serem utilizados para autenticar e configurar um canal criptografado.
- **Phase2 Algorithms** - corresponde ao parâmetro **esp** Libreswan - digite os algoritmos a serem utilizados para as negociações **IPsec**.
  Verifique o campo **Disable PFS** para desligar o Perfect Forward Secrecy (PFS) para garantir a compatibilidade com servidores antigos que não suportam PFS.
- **Phase1 Lifetime** - corresponde ao parâmetro **ikelifetime** Libreswan - quanto tempo a chave utilizada para criptografar o tráfego será válida.
- **Phase2 Lifetime** - corresponde ao parâmetro **salifetime** Libreswan - quanto tempo uma determinada instância de uma conexão deve durar antes de expirar.
  Observe que a chave de criptografia deve ser mudada de tempos em tempos por razões de segurança.
- **Remote network** - corresponde ao parâmetro **rightsubnet** Libreswan - a rede remota privada de destino que deve ser alcançada através da VPN.
  Verifique o campo **narrowing** para permitir o estreitamento. Observe que ele só é eficaz nas negociações do IKEv2.
- **Enable fragmentation** - corresponde ao parâmetro **fragmentation** Libreswan - permitir ou não a fragmentação do IKE. Os valores válidos são **yes** (padrão) ou **no**.
- **Enable Mobike** - corresponde ao parâmetro **mobike** Libreswan - se permitir a Mobilidade e o Protocolo Multihoming (MOBIKE, RFC 4555) para permitir uma conexão para migrar seu endpoint sem a necessidade de reiniciar a conexão do zero. Isto é usado em dispositivos móveis que comutam entre conexões com fio, sem fio, ou de dados móveis. Os valores são **no** (padrão) ou **yes**.

6. Selecione a entrada do menu **IPv4**:
   IPv4 Method

   - **Automatic (DHCP)** - Escolha esta opção se a rede à qual você está se conectando usar anúncios de Roteador (RA) ou um servidor **DHCP** para atribuir endereços dinâmicos **IP**.

- **Link-Local Only** - Escolha esta opção se a rede à qual você está se conectando não tiver um servidor **DHCP** e você não quiser atribuir endereços **IP** manualmente. Endereços aleatórios serão atribuídos de acordo com *RFC 3927* com prefixo **169.254/16**.

- **Manual** - Escolha esta opção se você quiser atribuir endereços **IP** manualmente.

- **Disable** - **IPv4** está desativado para esta conexão.
  DNS

  Na seção **DNS**, quando **Automatic** é **ON**, mude-o para **OFF** para inserir o endereço IP de um servidor DNS que você deseja usar separando os IPs por vírgula.

  Routes

  Note que na seção **Routes**, quando **Automatic** é **ON**, são usadas rotas do DHCP, mas também é possível acrescentar rotas estáticas adicionais. Quando **OFF**, somente rotas estáticas são usadas.

- **Address** - Digite o endereço **IP** de uma rede ou host remoto.

- **Netmask** - A máscara de rede ou o comprimento do prefixo do endereço **IP** inserido acima.

- **Gateway** - O endereço **IP** do gateway que conduz à rede remota ou ao host inserido acima.

- **Metric** - Um custo de rede, um valor de preferência a dar a esta rota. Os valores mais baixos serão preferidos em relação aos valores mais altos.
  Use this connection only for resources on its network

  Selecione esta caixa de seleção para evitar que a conexão se torne a rota padrão. Selecionar esta opção significa que somente o tráfego especificamente destinado às rotas aprendidas automaticamente sobre a conexão ou entradas aqui manualmente é roteado sobre a conexão.

7. Para configurar as configurações **IPv6** em uma conexão **VPN**, selecione a entrada do menu **IPv6**:
   IPv6 Method

   - **Automatic** - Escolha esta opção para usar **IPv6** Endereço sem Estado AutoConfiguração (SLAAC) para criar uma configuração automática, sem Estado, baseada no endereço de hardware e Anúncios de Roteador (RA).

   - **Automatic, DHCP only** - Escolha esta opção para não usar a RA, mas solicite informações diretamente de **DHCPv6** para criar uma configuração estatal.

   - **Link-Local Only** - Escolha esta opção se a rede à qual você está se conectando não tiver um servidor **DHCP** e você não quiser atribuir endereços **IP** manualmente. Endereços aleatórios serão atribuídos de acordo com *RFC 4862* com prefixo **FE80::0**.

   - **Manual** - Escolha esta opção se você quiser atribuir endereços **IP** manualmente.

   - **Disable** - **IPv6** está desativado para esta conexão.
     Note que **DNS**, **Routes**, **Use this connection only for resources on its network** são comuns às configurações de **IPv4**.

8. Uma vez terminada a edição da conexão **VPN**, clique no botão **Adicionar** para personalizar a configuração ou no botão **Aplicar** para salvá-la para a existente.

9. Mude o perfil para **ON** para ativar a conexão **VPN**.

Recursos adicionais

- Para mais detalhes sobre os parâmetros suportados **Libreswan**, consulte a página de manual **nm-settings-libreswan(5)**.

14.2. CONFIGURAÇÃO DE UMA CONEXÃO VPN USANDO UM EDITOR DE NM-CONEXÃO

Este procedimento descreve como configurar uma conexão VPN usando **nm-connection-editor**.

Pré-requisitos

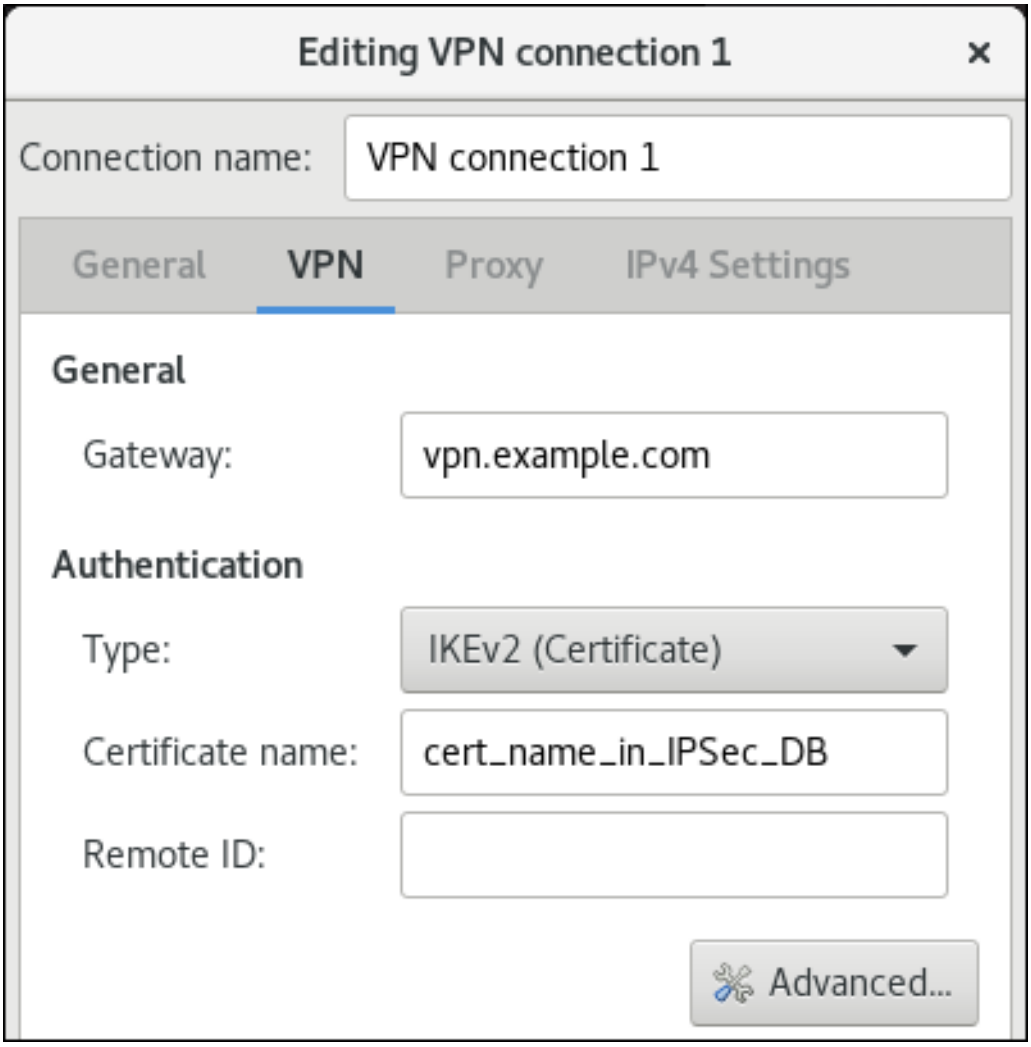
- O pacote **NetworkManager-libreswan-gnome** está instalado.
- Se você configurar uma conexão Internet Key Exchange versão 2 (IKEv2):
  - O certificado é importado para o banco de dados de serviços de segurança de rede IPsec (NSS).
  - O apelido do certificado no banco de dados do NSS é conhecido.

Procedimento

1. Abra um terminal, e entre:

```
$ nm-connection-editor
```

2. Clique no botão para adicionar uma nova conexão.
3. Selecione o tipo de conexão **IPsec based VPN**, e clique em **Criar**.
4. Na aba **VPN**:
   a. Digite o nome do host ou endereço IP do gateway VPN no campo **Gateway**, e selecione um tipo de autenticação. Com base no tipo de autenticação, você deve inserir informações adicionais diferentes:
      - **IKEv2 (Certifiate)** autentica o cliente usando um certificado, que é mais seguro. Esta configuração requer o apelido do certificado no banco de dados do IPsec NSS
      - **IKEv1 (XAUTH)** autentica o usuário usando um nome de usuário e uma senha (chave pré-compartilhada). Esta configuração requer que o usuário digite os seguintes valores:
        - Nome do usuário
        - Senha
        - Nome do grupo
        - Segredo
   b. Se o servidor remoto especificar um identificador local para a central IKE, digite a seqüência exata no campo **Remote ID**. No servidor remoto executa Libreswan, este valor é definido no parâmetro **leftid** do servidor.



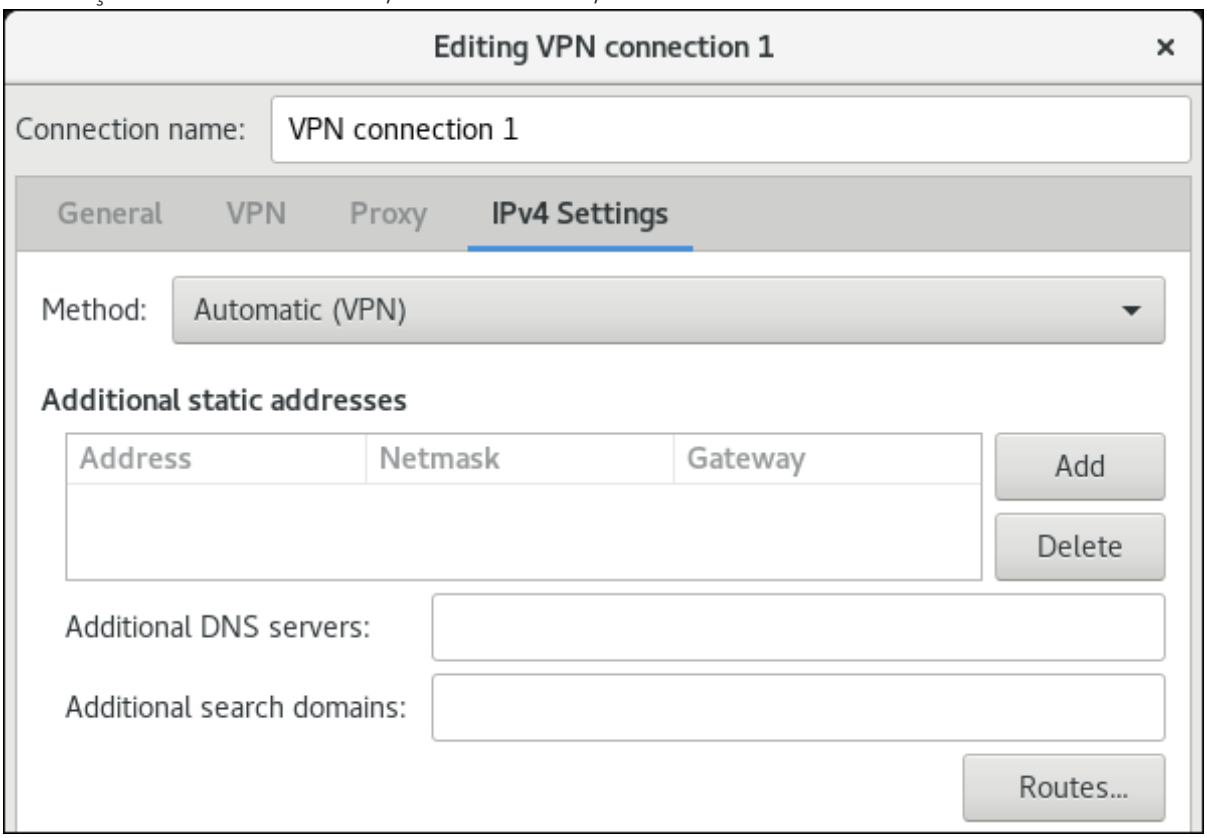
c. Opcionalmente, configure configurações adicionais clicando no botão **Avançado**. Você pode configurar as seguintes configurações:

- Identificação
  - **Domain** - Se necessário, digite o nome do domínio.
- Segurança
  - **Phase1 Algorithms** corresponde ao parâmetro **ike** Libreswan. Insira os algoritmos a serem usados para autenticar e configurar um canal criptografado.
  - **Phase2 Algorithms** corresponde ao parâmetro **esp** Libreswan. Digite os algoritmos a serem utilizados nas negociações de **IPsec**.
    Verifique o campo **Disable PFS** para desligar o Perfect Forward Secrecy (PFS) para garantir a compatibilidade com servidores antigos que não suportam PFS.
  - **Phase1 Lifetime** corresponde ao parâmetro **ikelifetime** Libreswan. Este parâmetro define quanto tempo a chave utilizada para criptografar o tráfego é válida.
  - **Phase2 Lifetime** corresponde ao parâmetro **salifetime** Libreswan. Este parâmetro define por quanto tempo uma associação de segurança é válida.
- Conectividade
  - **Remote network** corresponde ao parâmetro **rightsubnet** Libreswan e define a rede remota privada de destino que deve ser alcançada através da VPN.

Verifique o campo **narrowing** para permitir o estreitamento. Observe que ele só é eficaz na negociação do IKEv2.

- **Enable fragmentation** corresponde ao parâmetro **fragmentation** Libreswan e define se deve ou não permitir a fragmentação do IKE. Os valores válidos são **yes** (padrão) ou **no**.
- **Enable Mobike** corresponde ao parâmetro **mobike** Libreswan. O parâmetro define se permite a Mobilidade e o Protocolo Multihoming (MOBIKE) (RFC 4555) para permitir uma conexão para migrar seu ponto final sem a necessidade de reiniciar a conexão do zero. Isto é usado em dispositivos móveis que comutam entre conexões com fio, sem fio ou de dados móveis. Os valores são **no** (padrão) ou **yes**.

5. Na aba **IPv4 Settings**, selecione o método de atribuição de IP e, opcionalmente, defina endereços estáticos adicionais, servidores DNS, domínios de busca e rotas.



6. Salvar a conexão.
7. Fechar **nm-connection-editor**.

NOTA

Ao adicionar uma nova conexão clicando no botão , **NetworkManager** cria um novo arquivo de configuração para essa conexão e depois abre o mesmo diálogo que é usado para editar uma conexão existente. A diferença entre estes diálogos é que um perfil de conexão existente tem uma entrada no menu **Details**.

Recursos adicionais

- Para mais detalhes sobre os parâmetros IPsec suportados, consulte a página de manual **nm-settings-libreswan(5)**.

14.3. INFORMAÇÕES RELACIONADAS

- Para mais informações sobre a configuração de VPNs usando IPsec, consulte o capítulo Configuração de uma VPN com IPsec no documento Securing networks (Segurança de redes) .

CAPÍTULO 15. CONFIGURAÇÃO DE TÚNEIS IP

Similar a uma VPN, um túnel IP conecta diretamente duas redes através de uma terceira rede, como a Internet. Entretanto, nem todos os protocolos de túnel suportam criptografia.

Os roteadores em ambas as redes que estabelecem o túnel requerem pelo menos duas interfaces:

- Uma interface que está conectada à rede local
- Uma interface que é conectada à rede através da qual o túnel é estabelecido.

Para estabelecer o túnel, você cria uma interface virtual em ambos os roteadores com um endereço IP a partir da sub-rede remota.

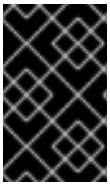
O NetworkManager suporta os seguintes túneis IP:

- Encapsulamento genérico de roteamento (GRE)
- Encapsulamento genérico de roteamento sobre IPv6 (IP6GRE)
- Ponto de Acesso ao Terminal de Encapsulamento de Roteamento Genérico (GRETAP)
- Ponto de Acesso Terminal de Encapsulamento de Roteamento Genérico sobre IPv6 (IP6GRETAP)
- IPv4 sobre IPv4 (IPIP)
- IPv4 sobre IPv6 (IPIP6)
- IPv6 sobre IPv6 (IP6IP6)
- Transição simples pela Internet (SIT)

Dependendo do tipo, estes túneis atuam ou na camada 2 ou 3 do modelo de Interconexão de Sistemas Abertos (OSI).

15.1. CONFIGURAÇÃO DE UM TÚNEL IPIP USANDO NMCLI PARA ENCAPSULAR O TRÁFEGO IPV4 EM PACOTES IPV4

Um túnel IP sobre IP (IPIP) opera na camada 3 do OSI e encapsula o tráfego IPv4 em pacotes IPv4, conforme descrito na RFC 2003.

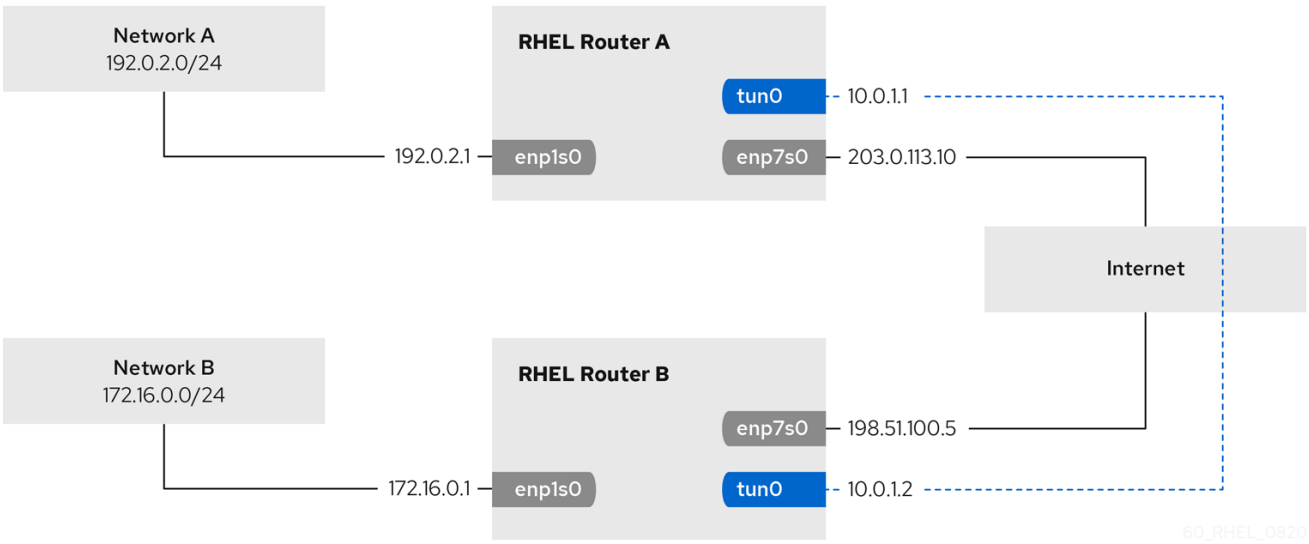


IMPORTANTE

Os dados enviados através de um túnel IPIP não são criptografados. Por razões de segurança, utilizar o túnel somente para dados que já estão criptografados, por exemplo, por outros protocolos, tais como HTTPS.

Note que os túneis IPIP suportam apenas pacotes unicast. Se você precisar de um túnel IPv4 que suporte multicast, veja Seção 15.2, “Configuração de um túnel GRE usando nmcli para encapsular o tráfego de camada-3 em pacotes IPv4”.

Este procedimento descreve como criar um túnel IPIP entre dois roteadores RHEL para conectar duas sub-redes internas através da Internet, como mostrado no diagrama a seguir:



Pré-requisitos

- Cada roteador RHEL tem uma interface de rede que é conectada à sua sub-rede local.
- Cada roteador RHEL tem uma interface de rede que está conectada à Internet.
- O tráfego que você deseja enviar através do túnel é IPv4 unicast.

Procedimento

1. No roteador RHEL da rede A:
   a. Criar uma interface de túnel IPIP chamada **tun0**:

   ```
   # nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name tun0 ifname tun0 remote 198.51.100.5 local 203.0.113.10
   ```

   Os parâmetros **remote** e **local** definem os endereços IP públicos dos roteadores remotos e locais.

   b. Defina o endereço IPv4 para o dispositivo **tun0**:

   ```
   # nmcli connection modify tun0 ipv4.addresses '10.0.1.1/30'
   ```

   Note que uma sub-rede **/30** com dois endereços IP utilizáveis é suficiente para o túnel.

   c. Configure a conexão **tun0** para usar uma configuração IPv4 manual:

   ```
   # nmcli connection modify tun0 ipv4.method manual
   ```

   d. Adicionar uma rota estática que encaminhe o tráfego para a rede **172.16.0.0/24** para o IP do túnel no roteador B:

   ```
   # nmcli connection modify tun0 ipv4.routes "172.16.0.0/24 10.0.1.2"
   ```

   e. Habilitar a conexão **tun0**.

```
# nmcli connection up tun0
```

f. Habilitar o envio de pacotes:

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

2. No roteador RHEL da rede B:

a. Criar uma interface de túnel IPIP chamada **tun0**:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name tun0 ifname
tun0 remote 203.0.113.10 local 198.51.100.5
```

Os parâmetros **remote** e **local** definem os endereços IP públicos dos roteadores remotos e locais.

b. Defina o endereço IPv4 para o dispositivo **tun0**:

```
# nmcli connection modify tun0 ipv4.addresses '10.0.1.2/30'
```

c. Configure a conexão **tun0** para usar uma configuração IPv4 manual:

```
# nmcli connection modify tun0 ipv4.method manual
```

d. Adicione uma rota estática que encaminha o tráfego para a rede **192.0.2.0/24** para o IP do túnel no roteador A:

```
# nmcli connection modify tun0 ipv4.routes "192.0.2.0/24 10.0.1.1"
```

e. Habilitar a conexão **tun0**.

```
# nmcli connection up tun0
```

f. Habilitar o envio de pacotes:

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

Etapas de verificação

1. A partir de cada roteador RHEL, pingando o endereço IP da interface interna do outro roteador:

a. No Router A, ping **172.16.0.1**:

```
# ping 172.16.0.1
```

b. No Router B, ping **192.0.2.1**:

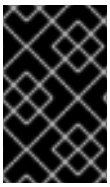
```
# ping 192.0.2.1
```

Recursos adicionais

- Para mais detalhes sobre o uso de **nmcli**, consulte a página de manual **nmcli**.
- Para detalhes sobre as configurações do túnel você pode definir com **nmcli**, veja a seção **ip-tunnel settings** na página de manual **nm-settings(5)**.

15.2. CONFIGURAÇÃO DE UM TÚNEL GRE USANDO NMCLI PARA ENCAPSULAR O TRÁFEGO DE CAMADA-3 EM PACOTES IPV4

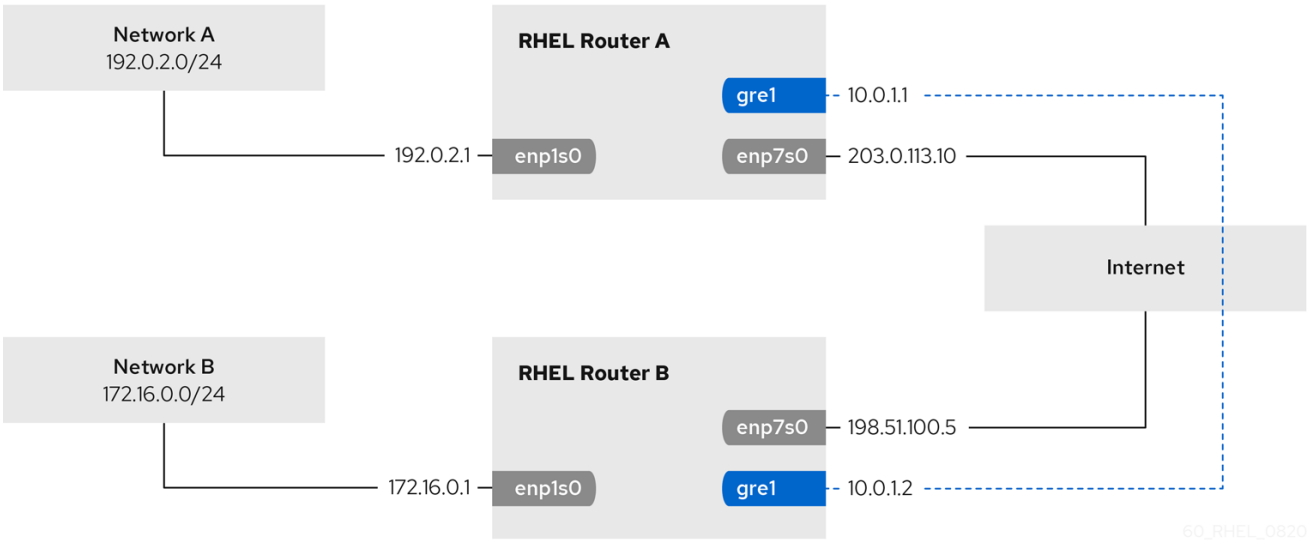
Um túnel de Encapsulamento de Roteamento Genérico (GRE) encapsula o tráfego de camada-3 em pacotes IPv4, conforme descrito no RFC 2784. Um túnel GRE pode encapsular qualquer protocolo da camada 3 com um tipo de Ethernet válido.

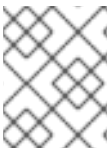


IMPORTANTE

Os dados enviados através de um túnel GRE não são criptografados. Por razões de segurança, utilizar o túnel somente para dados que já estão criptografados, por exemplo, por outros protocolos, tais como HTTPS.

Este procedimento descreve como criar um túnel GRE entre dois roteadores RHEL para conectar duas sub-redes internas através da Internet, como mostrado no diagrama a seguir:





NOTA

O nome do dispositivo **gre0** é reservado. Use **gre1** ou um nome diferente para o dispositivo.

Pré-requisitos

- Cada roteador RHEL tem uma interface de rede que é conectada à sua sub-rede local.
- Cada roteador RHEL tem uma interface de rede que está conectada à Internet.

Procedimento

1. No roteador RHEL da rede A:

a. Criar uma interface de túnel GRE chamada **gre1**:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode gre con-name gre1 ifname
gre1 remote 198.51.100.5 local 203.0.113.10
```

Os parâmetros **remote** e **local** definem os endereços IP públicos dos roteadores remotos e locais.

b. Defina o endereço IPv4 para o dispositivo **gre1**:

```
# nmcli connection modify gre1 ipv4.addresses '10.0.1.1/30'
```

Note que uma sub-rede /**30** com dois endereços IP utilizáveis é suficiente para o túnel.

c. Configure a conexão **gre1** para usar uma configuração IPv4 manual:

```
# nmcli connection modify gre1 ipv4.method manual
```

d. Adicionar uma rota estática que encaminhe o tráfego para a rede **172.16.0.0/24** para o IP do túnel no roteador B:

```
# nmcli connection modify tun0 ipv4.routes "172.16.0.0/24 10.0.1.2"
```

e. Habilitar a conexão **gre1**.

```
# nmcli connection up gre1
```

f. Habilitar o envio de pacotes:

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

2. No roteador RHEL da rede B:

a. Criar uma interface de túnel GRE chamada **gre1**:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name gre1 ifname
gre1 remote 203.0.113.10 local 198.51.100.5
```

Os parâmetros **remote** e **local** definem os endereços IP públicos dos roteadores remotos e locais.

b. Defina o endereço IPv4 para o dispositivo **gre1**:

```
# nmcli connection modify gre1 ipv4.addresses '10.0.1.2/30'
```

c. Configure a conexão **gre1** para usar uma configuração IPv4 manual:

```
# nmcli connection modify gre1 ipv4.method manual
```

d. Adicione uma rota estática que encaminha o tráfego para a rede **192.0.2.0/24** para o IP do túnel no roteador A:

```
# nmcli connection modify tun0 ipv4.routes "192.0.2.0/24 10.0.1.1"
```

e. Habilitar a conexão **gre1**.

```
# nmcli connection up gre1
```

f. Habilitar o envio de pacotes:

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

Etapas de verificação

1. A partir de cada roteador RHEL, pingando o endereço IP da interface interna do outro roteador:

   a. No Router A, ping **172.16.0.1**:

   ```
   # ping 172.16.0.1
   ```

   b. No Router B, ping **192.0.2.1**:

   ```
   # ping 192.0.2.1
   ```

Recursos adicionais

- Para mais detalhes sobre o uso de **nmcli**, consulte a página de manual **nmcli**.
- Para detalhes sobre as configurações do túnel você pode definir com **nmcli**, veja a seção **ip-tunnel settings** na página de manual **nm-settings(5)**.

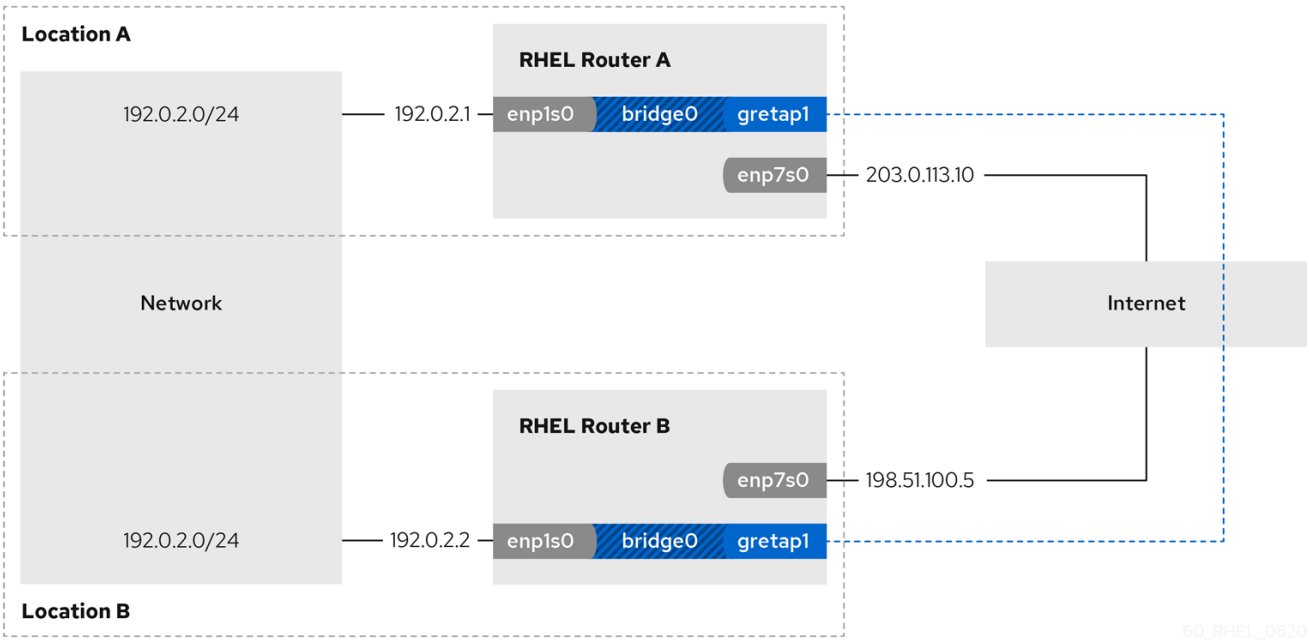
15.3. CONFIGURAÇÃO DE UM TÚNEL GRETAP PARA TRANSFERIR QUADROS ETHERNET SOBRE IPV4

Um túnel de Encapsulamento Terminal de Encapsulamento Genérico (GRETAP) opera no nível 2 do OSI e encapsula o tráfego Ethernet em pacotes IPv4, conforme descrito no RFC 2784.

IMPORTANTE

Os dados enviados através de um túnel GRETAP não são criptografados. Por razões de segurança, estabelecer o túnel através de uma VPN ou uma conexão criptografada diferente.

Este procedimento descreve como criar um túnel GRETAP entre dois roteadores RHEL para conectar duas redes usando uma ponte, como mostrado no diagrama a seguir:



NOTA

O nome do dispositivo **gretap0** é reservado. Use **gretap1** ou um nome diferente para o dispositivo.

Pré-requisitos

- Cada roteador RHEL tem uma interface de rede que é conectada à sua rede local, e a interface não tem nenhuma configuração IP atribuída.
- Cada roteador RHEL tem uma interface de rede que está conectada à Internet.

Procedimento

1. No roteador RHEL da rede A:
   a. Criar uma interface de ponte chamada **bridge0**:

```
# nmcli connection add type bridge con-name bridge0 ifname bridge0
```

   b. Configurar as configurações de IP da ponte:

```
# nmcli connection modify bridge0 ipv4.addresses '192.0.2.1/24'
# nmcli connection modify bridge0 ipv4.method manual
```

   c. Adicionar um novo perfil de conexão para a interface que está conectada à rede local à ponte:

```
# nmcli connection add type ethernet slave-type bridge con-name bridge0-port1
ifname enp1s0 master bridge0
```

   d. Acrescentar um novo perfil de conexão para a interface do túnel GRETAP à ponte:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode gretap slave-type bridge
con-name bridge0-port2 ifname gretap1 remote 198.51.100.5 local 203.0.113.10
master bridge0
```

Os parâmetros **remote** e **local** definem os endereços IP públicos dos roteadores remotos e locais.

e. Opcional: Desabilite o Protocolo Spanning Tree (STP) se você não precisar dele:

```
# nmcli connection modify bridge0 bridge.stp no
```

Por padrão, o STP é ativado e causa um atraso antes que você possa usar a conexão.

f. Configurar que a ativação da conexão **bridge0** ativa automaticamente as portas da ponte:

```
# nmcli connection modify bridge0 connection.autoconnect-slaves 1
```

g. Ative a conexão **bridge0**:

```
# nmcli connection up bridge0
```

2. No roteador RHEL da rede B:

a. Criar uma interface de ponte chamada **bridge0**:

```
# nmcli connection add type bridge con-name bridge0 ifname bridge0
```

b. Configurar as configurações de IP da ponte:

```
# nmcli connection modify bridge0 ipv4.addresses '192.0.2.2/24'
# nmcli connection modify bridge0 ipv4.method manual
```

c. Adicionar um novo perfil de conexão para a interface que está conectada à rede local à ponte:

```
# nmcli connection add type ethernet slave-type bridge con-name bridge0-port1
ifname enp1s0 master bridge0
```

d. Acrescentar um novo perfil de conexão para a interface do túnel GRETAP à ponte:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode gretap slave-type bridge
con-name bridge0-port2 ifname gretap1 remote 203.0.113.10 local 198.51.100.5
master bridge0
```

Os parâmetros **remote** e **local** definem os endereços IP públicos dos roteadores remotos e locais.

e. Opcional: Desabilite o Protocolo Spanning Tree (STP) se você não precisar dele:

```
# nmcli connection modify bridge0 bridge.stp no
```

f. Configurar que a ativação da conexão **bridge0** ativa automaticamente as portas da ponte:

```
# nmcli connection modify bridge0 connection.autoconnect-slaves 1
```

g. Ative a conexão **bridge0**:

```
# nmcli connection up bridge0
```

Etapas de verificação

1. Em ambos os roteadores, verifique se as conexões **enp1s0** e **gretap1** estão conectadas e se a coluna **CONNECTION** exibe o nome da conexão da porta:

```
# nmcli device
nmcli device
DEVICE   TYPE      STATE      CONNECTION
...
bridge0  bridge    connected  bridge0
enp1s0   ethernet  connected  bridge0-port1
gretap1  iptunnel  connected  bridge0-port2
```

2. A partir de cada roteador RHEL, pingando o endereço IP da interface interna do outro roteador:

   a. No Router A, ping **192.0.2.2**:

```
# ping 192.0.2.2
```

   b. No Router B, ping **192.0.2.1**:

```
# ping 192.0.2.1
```

Recursos adicionais

- Para mais detalhes sobre o uso de **nmcli**, consulte a página de manual **nmcli**.
- Para detalhes sobre as configurações do túnel você pode definir com **nmcli**, veja a seção **ip-tunnel settings** na página de manual **nm-settings(5)**.

15.4. RECURSOS ADICIONAIS

- Para uma lista de interfaces de túneis e na configuração temporária de túneis usando o utilitário **ip**, consulte a página de manual **ip-link(8)**.

CAPÍTULO 16. CONFIGURAÇÃO DO CANAL DE FIBRA SOBRE ETHERNET

Baseado na norma IEEE T11 FC-BB-5, o Fibre Channel over Ethernet (FCoE) é um protocolo para transmitir quadros de Fibre Channel sobre redes Ethernet. Normalmente, os centros de dados possuem uma rede LAN e uma rede de área de armazenamento (SAN) dedicadas que são separadas umas das outras com sua própria configuração específica. O FCoE combina essas redes em uma estrutura de rede única e convergente. Os benefícios do FCoE são, por exemplo, custos mais baixos de hardware e energia.

16.1. USANDO HARDWARE FCOE HBAS EM RHEL

No Red Hat Enterprise Linux você pode usar o hardware FCoE Host Bus Adapter (HBA) suportado pelos seguintes drivers:

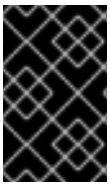
- **qedf**
- **bnx2fc**
- **fnic**

Se você usar tal HBA, você configura as configurações do FCoE na configuração do HBA. Para obter detalhes, consulte a documentação do adaptador.

Após configurar o HBA em sua configuração, os Números de Unidade Lógica (LUN) exportados da Rede de Área de Armazenamento (SAN) estão automaticamente disponíveis para a RHEL como dispositivos **/dev/sd***. Você pode usar estes dispositivos de forma similar aos dispositivos de armazenamento local.

16.2. INSTALAÇÃO DE UM DISPOSITIVO DE SOFTWARE FCOE

Um dispositivo de software FCoE permite acessar os Números de Unidade Lógica (LUN) sobre o FCoE usando um adaptador Ethernet que suporta parcialmente o descarregamento do FCoE.



IMPORTANTE

A RHEL não suporta os dispositivos de software FCoE que requerem o módulo de kernel **fcoe.ko**. Para detalhes, consulte a remoção do software FCoE na documentação **Considerations in adopting RHEL 8**.

Após completar este procedimento, os LUNs exportados da Rede de Área de Armazenamento (SAN) estão automaticamente disponíveis para a RHEL como dispositivos **/dev/sd***. Você pode usar estes dispositivos de forma similar aos dispositivos de armazenamento locais.

Pré-requisitos

- O Host Bus Adapter (HBA) usa o driver **qedf**, **bnx2fc**, ou **fnic** e não requer o módulo do kernel **fcoe.ko**.
- A SAN usa uma VLAN para separar o tráfego de armazenamento do tráfego Ethernet normal.
- O switch de rede foi configurado para suportar a VLAN.
- O HBA do servidor é configurado em sua BIOS. Para obter detalhes, consulte a documentação de seu HBA.

- A HBA está conectada à rede e o link está pronto.

Procedimento

1. Instale o pacote **fcoe-utils**:

```
# yum install fcoe-utils
```

2. Copie o arquivo modelo **/etc/fcoe/cfg-ethx** para **/etc/fcoe/cfg-*interface_name***. Por exemplo, se você quiser configurar a interface **enp1s0** para usar o FCoE, entre:

```
# cp /etc/fcoe/cfg-ethx /etc/fcoe/cfg-enp1s0
```

3. Habilite e inicie o serviço **fcoe**:

```
# systemctl enable --now fcoe
```

4. Descubra o FCoE VLAN ID, inicie o iniciador e crie um dispositivo de rede para a VLAN descoberta:

```
# fipvlan -s -c enp1s0
Created VLAN device enp1s0.200
Starting FCoE on interface enp1s0.200
Fibre Channel Forwarders Discovered
interface       | VLAN | FCF MAC
------------------------------------------
enp1s0          | 200  | 00:53:00:a7:e7:1b
```

5. Opcional: Para exibir detalhes sobre os alvos descobertos, os LUNs, e os dispositivos associados com os LUNs, entre:

```
# fcoeadm -t
Interface:        enp1s0.200
Roles:            FCP Target
Node Name:        0x500a0980824acd15
Port Name:        0x500a0982824acd15
Target ID:        0
MaxFrameSize:     2048 bytes
OS Device Name:   rport-11:0-1
FC-ID (Port ID):  0xba00a0
State:            Online

LUN ID  Device Name   Capacity   Block Size  Description
------  -----------  ----------  ----------  ----------------------
    0  sdb           28.38 GiB      512     NETAPP LUN (rev 820a)
    ...
```

Este exemplo mostra que o LUN 0 da SAN foi anexado ao host como o dispositivo **/dev/sdb**.

Etapas de verificação

- Use o comando **fcoeadm -i** para exibir informações sobre todas as interfaces FCoE ativas:

```
# fcoeadm -i
```

```
Description:      BCM57840 NetXtreme II 10 Gigabit Ethernet
Revision:         11
Manufacturer:     Broadcom Inc. and subsidiaries
Serial Number:    000AG703A9B7

Driver:           bnx2x Unknown
Number of Ports:  1

    Symbolic Name:     bnx2fc (QLogic BCM57840) v2.12.13 over enp1s0.200
    OS Device Name:    host11
    Node Name:         0x2000000af70ae935
    Port Name:         0x2001000af70ae935
    Fabric Name:       0x20c8002a6aa7e701
    Speed:             10 Gbit
    Supported Speed:   1 Gbit, 10 Gbit
    MaxFrameSize:      2048 bytes
    FC-ID (Port ID):   0xba02c0
    State:             Online
```

Recursos adicionais

- Para mais detalhes sobre a utilidade **fcoeadm**, consulte a página de manual **fcoeadm(8)**.
- Para detalhes sobre como montar o armazenamento conectado através de um software FCoE quando o sistema inicia, consulte o arquivo **/usr/share/doc/fcoe-utils/README**.

16.3. RECURSOS ADICIONAIS

- Para detalhes sobre o uso de dispositivos Fibre Channel, consulte a seção Usando dispositivos Fibre Channel no guia **Managing storage devices**.

CAPÍTULO 17. AUTENTICAÇÃO DE UM CLIENTE RHEL PARA A REDE USANDO A NORMA 802.1X

Os administradores freqüentemente usam o Controle de Acesso à Rede (NAC) baseado no padrão IEEE 802.1X para proteger uma rede contra clientes LAN e Wi-Fi não autorizados. Os procedimentos nesta seção descrevem diferentes opções para configurar a autenticação da rede.

17.1. CONFIGURAÇÃO DA AUTENTICAÇÃO DE REDE 802.1X EM UMA CONEXÃO ETHERNET EXISTENTE USANDO NMCLI

Usando o utilitário **nmcli**, é possível configurar o cliente para se autenticar na rede. Este procedimento descreve como configurar a autenticação do Protocolo de Autenticação Extensível Protegida (PEAP) com o Microsoft Challenge-Handshake Authentication Protocol versão 2 (MSCHAPv2) em um perfil de conexão Ethernet NetworkManager existente chamado **enp1s0**.

Pré-requisitos

1. A rede deve ter uma autenticação de rede 802.1X.
2. O perfil de conexão Ethernet existe no NetworkManager e tem uma configuração IP válida.
3. Se o cliente for obrigado a verificar o certificado do autenticador, o certificado da Autoridade Certificadora (CA) deve ser armazenado no diretório **/etc/pki/ca-trust/source/anchors/**.
4. O pacote **wpa_supplicant** está instalado.

Procedimento

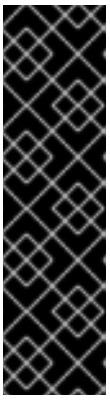
1. Defina o Protocolo de Autenticação Extensível (EAP) para **peap**, o protocolo de autenticação interna para **mschapv2**, e o nome do usuário:

```
# nmcli connection modify enp1s0 802-1x.eap peap 802-1x.phase2-auth mschapv2
802-1x.identity user_name
```

   Observe que você deve definir os parâmetros **802-1x.eap**, **802-1x.phase2-auth**, e **802-1x.identity** em um único comando.

2. Opcionalmente, armazenar a senha na configuração:

```
# nmcli connection modify enp1s0 802-1x.password password
```



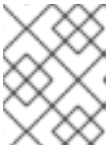
IMPORTANTE

Por padrão, o NetworkManager armazena a senha em texto claro no **/etc/sysconfig/network-scripts/keys-*connection_name*** que só pode ser lido pelo usuário do **root**. Entretanto, senhas de texto claras em um arquivo de configuração podem ser um risco de segurança.

Para aumentar a segurança, defina o parâmetro **802-1x.password-flags** para **0x1**. Com esta configuração, em servidores com o ambiente desktop GNOME ou o **nm-applet** em execução, o NetworkManager recupera a senha destes serviços. Em outros casos, o NetworkManager solicita a senha.

3. Se o cliente for obrigado a verificar o certificado do autenticador, defina o parâmetro **802-1x.ca-cert** no perfil de conexão para o caminho do certificado da CA:

```
# nmcli connection modify enp1s0 802-1x.ca-cert /etc/pki/ca-trust/source/anchors/ca.crt
```



NOTA

Por razões de segurança, a Red Hat recomenda o uso do certificado do autenticador para permitir que os clientes validem a identidade do autenticador.

4. Ativar o perfil de conexão:

```
# nmcli connection up enp1s0
```

Etapas de verificação

- Acessar recursos na rede que requerem autenticação da rede.

Recursos adicionais

- Para detalhes sobre como adicionar um perfil de conexão NetworkManager Ethernet, veja Capítulo 8, *Configuração de uma conexão Ethernet*.
- Para mais parâmetros relacionados a 802.1X e suas descrições, consulte a seção **802-1x settings** na página de manual **nm-settings(5)**.
- Para mais detalhes sobre a utilidade **nmcli**, consulte a página de manual **nmcli(1)**.

17.2. CONFIGURAÇÃO DE UMA CONEXÃO ETHERNET ESTÁTICA COM AUTENTICAÇÃO DE REDE 802.1X USANDO AS FUNÇÕES DO SISTEMA RHEL

Usando as funções do Sistema RHEL, você pode automatizar a criação de uma conexão Ethernet que usa o padrão 802.1X para autenticar o cliente. Este procedimento descreve como adicionar remotamente uma conexão Ethernet para a interface **enp1s0** com as seguintes configurações, executando um Livro de Jogadas Possível:

- Um endereço IPv4 estático - **192.0.2.1** com uma máscara de sub-rede **/24**
- Um endereço IPv6 estático - **2001:db8:1::1** com uma máscara de sub-rede **/64**
- Um gateway padrão IPv4 - **192.0.2.254**
- Um gateway padrão IPv6 - **2001:db8:1::fffe**
- Um servidor DNS IPv4 - **192.0.2.200**
- Um servidor DNS IPv6 - **2001:db8:1::ffbb**
- Um domínio de busca DNS - **example.com**
- 802.1X autenticação de rede usando o protocolo **TLS** Extensible Authentication Protocol (EAP)

Execute este procedimento no Nó de controle possível.

Pré-requisitos

- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.
- Se você usar um usuário remoto diferente de **root** ao executar o playbook, você deve ter as permissões apropriadas **sudo** no nó gerenciado.
- A rede suporta autenticação de rede 802.1X.
- O nó gerenciado utiliza o NetworkManager.
- Os seguintes arquivos necessários para autenticação TLS existem no nó de controle:
  - A chave do cliente armazenada no arquivo **/srv/data/client.key**.
  - O certificado do cliente armazenado no arquivo **/srv/data/client.crt**.
  - O certificado da Autoridade Certificadora (CA) armazenado no arquivo **/srv/data/ca.crt**.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo **/etc/ansible/hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook **~/enable-802.1x.yml** com o seguinte conteúdo:

```
---
- name: Configure an Ethernet connection with 802.1X authentication
  hosts: node.example.com
  become: true
  tasks:
    - name: Copy client key for 802.1X authentication
      copy:
        src: "/srv/data/client.key"
        dest: "/etc/pki/tls/private/client.key"
        mode: 0600

    - name: Copy client certificate for 802.1X authentication
      copy:
        src: "/srv/data/client.crt"
        dest: "/etc/pki/tls/certs/client.crt"

    - name: Copy CA certificate for 802.1X authentication
      copy:
        src: "/srv/data/ca.crt"
        dest: "/etc/pki/ca-trust/source/anchors/ca.crt"

    - include_role:
        name: linux-system-roles.network
      vars:
        network_connections:
```

```
    - name: enp1s0
      type: ethernet
      autoconnect: yes
      ip:
        address:
          - 192.0.2.1/24
          - 2001:db8:1::1/64
        gateway4: 192.0.2.254
        gateway6: 2001:db8:1::fffe
        dns:
          - 192.0.2.200
          - 2001:db8:1::ffbb
        dns_search:
          - example.com
      ieee802_1x:
        identity: user_name
        eap: tls
        private_key: "/etc/pki/tls/private/client.key"
        private_key_password: "password"
        client_cert: "/etc/pki/tls/certs/client.crt"
        ca_cert: "/etc/pki/ca-trust/source/anchors/ca.crt"
        domain_suffix_match: example.com
      state: up
```

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

     ```
     # ansible-playbook -u root ~/enable-802.1x.yml
     ```

   - Para conectar-se como usuário ao host administrado, entre:

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/ethernet-static-IP.yml
     ```

     A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definido no **-u *user_name*** opção.

   Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Recursos adicionais

- Para detalhes sobre os parâmetros usados em **network_connections** e para informações adicionais sobre o Sistema de Papel **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre os parâmetros 802.1X, consulte a seção **ieee802_1x** no arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

17.3. CONFIGURAÇÃO DA AUTENTICAÇÃO DA REDE 802.1X EM UMA CONEXÃO WI-FI EXISTENTE USANDO NMCLI

Usando o utilitário **nmcli**, é possível configurar o cliente para se autenticar na rede. Este procedimento descreve como configurar a autenticação do Protocolo de Autenticação Extensível Protegida (PEAP) com o Microsoft Challenge-Handshake Authentication Protocol versão 2 (MSCHAPv2) em um perfil de conexão Wi-Fi NetworkManager existente chamado **wlp1s0**.

Pré-requisitos

1. A rede deve ter uma autenticação de rede 802.1X.
2. O perfil de conexão Wi-Fi existe no NetworkManager e tem uma configuração IP válida.
3. Se o cliente for obrigado a verificar o certificado do autenticador, o certificado da Autoridade Certificadora (CA) deve ser armazenado no diretório **/etc/pki/ca-trust/source/anchors/**.
4. O pacote **wpa_supplicant** está instalado.

Procedimento

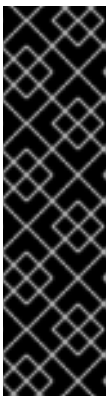
1. Configurar o modo de segurança Wi-Fi para **wpa-eap**, o Protocolo de Autenticação Extensível (EAP) para **peap**, o protocolo de autenticação interna para **mschapv2**, e o nome do usuário:

```
# nmcli connection modify wpl1s0 802-11-wireless-security.key-mgmt wpa-eap 802-1x.eap peap 802-1x.phase2-auth mschapv2 802-1x.identity user_name
```

   Observe que você deve definir os parâmetros **802-11-wireless-security.key-mgmt**, **802-1x.eap**, **802-1x.phase2-auth**, e **802-1x.identity** em um único comando.

2. Opcionalmente, armazenar a senha na configuração:

```
# nmcli connection modify wpl1s0 802-1x.password password
```



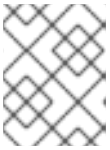
IMPORTANTE

Por padrão, o NetworkManager armazena a senha em texto claro no **/etc/sysconfig/network-scripts/keys-*connection_name*** que só pode ser lido pelo usuário do **root**. Entretanto, senhas de texto claras em um arquivo de configuração podem ser um risco de segurança.

Para aumentar a segurança, defina o parâmetro **802-1x.password-flags** para **0x1**. Com esta configuração, em servidores com o ambiente desktop GNOME ou o **nm-applet** em execução, o NetworkManager recupera a senha destes serviços. Em outros casos, o NetworkManager solicita a senha.

3. Se o cliente for obrigado a verificar o certificado do autenticador, defina o parâmetro **802-1x.ca-cert** no perfil de conexão para o caminho do certificado da CA:

```
# nmcli connection modify wpl1s0 802-1x.ca-cert /etc/pki/ca-trust/source/anchors/ca.crt
```



NOTA

Por razões de segurança, a Red Hat recomenda o uso do certificado do autenticador para permitir que os clientes validem a identidade do autenticador.

4. Ativar o perfil de conexão:

```
# nmcli connection up wpl1s0
```

Etapas de verificação

- Acessar recursos na rede que requerem autenticação da rede.

Recursos adicionais

- Para detalhes sobre como adicionar um perfil de conexão NetworkManager Ethernet, veja Capítulo 9, *Gerenciando conexões Wi-Fi*.
- Para mais parâmetros relacionados a 802.1X e suas descrições, consulte a seção **802-1x settings** na página de manual **nm-settings(5)**.
- Para mais detalhes sobre a utilidade **nmcli**, consulte a página de manual **nmcli(1)**.

CAPÍTULO 18. GERENCIANDO A CONFIGURAÇÃO PADRÃO DO GATEWAY

O gateway padrão é um roteador que encaminha pacotes de rede quando nenhuma outra rota corresponde ao destino de um pacote. Em uma rede local, o gateway padrão é tipicamente o host que está um salto mais próximo da Internet.

18.1. CONFIGURANDO O GATEWAY PADRÃO EM UMA CONEXÃO EXISTENTE USANDO NMCLI

Na maioria das situações, os administradores definem o gateway padrão quando criam uma conexão, como explicado, por exemplo, em Seção 8.1, “Configuração de uma conexão Ethernet estática usando nmcli”.

Esta seção descreve como definir ou atualizar o gateway padrão em uma conexão criada anteriormente usando o utilitário **nmcli**.

Pré-requisitos

- Pelo menos um endereço IP estático deve ser configurado na conexão na qual o gateway padrão será configurado.
- Se o usuário estiver logado em um console físico, as permissões de usuário são suficientes. Caso contrário, o usuário deve ter as permissões do **root**.

Procedimento

1. Defina o endereço IP do gateway padrão.
   Por exemplo, para definir o endereço IPv4 do gateway padrão no ***example*** conexão a **192.0.2.1**:

   ```
   $ sudo nmcli connection modify example ipv4.gateway "192.0.2.1"
   ```

   Por exemplo, para definir o endereço IPv6 do gateway padrão no ***example*** conexão a **2001:db8:1::1**:

   ```
   $ sudo nmcli connection modify example ipv6.gateway "2001:db8:1::1"
   ```

2. Reinicie a conexão de rede para que as mudanças entrem em vigor. Por exemplo, para reiniciar a ***example*** conexão usando a linha de comando:

   ```
   $ sudo nmcli connection up example
   ```

ATENÇÃO

Todas as conexões que atualmente utilizam esta conexão de rede são temporariamente interrompidas durante o reinício.

3. Opcionalmente, verificar se a rota está ativa.
   Para exibir o gateway padrão IPv4:

   ```
   $ ip -4 route
   default via 192.0.2.1 dev example proto static metric 100
   ```

   Para exibir o gateway padrão IPv6:

   ```
   $ ip -6 route
   default via 2001:db8:1::1 dev example proto static metric 100 pref medium
   ```

Recursos adicionais

- Seção 8.1, “Configuração de uma conexão Ethernet estática usando nmcli”

18.2. CONFIGURANDO O GATEWAY PADRÃO EM UMA CONEXÃO EXISTENTE USANDO O MODO INTERATIVO NMCLI

Na maioria das situações, os administradores definem o gateway padrão quando criam uma conexão, como explicado, por exemplo, em Seção 8.5, “Configuração de uma conexão Ethernet dinâmica usando o editor interativo nmcli”.

Esta seção descreve como definir ou atualizar o gateway padrão em uma conexão criada anteriormente, usando o modo interativo do utilitário **nmcli**.

Pré-requisitos

- Pelo menos um endereço IP estático deve ser configurado na conexão na qual o gateway padrão será configurado.
- Se o usuário estiver logado em um console físico, as permissões de usuário são suficientes. Caso contrário, o usuário deve ter permissões em **root**.

Procedimento

1. Abra o modo interativo **nmcli** para a conexão necessária. Por exemplo, para abrir o modo interativo **nmcli** para a conexão *example*:

   ```
   $ sudo nmcli connection edit example
   ```

2. Defina a porta de entrada padrão.
   Por exemplo, para definir o endereço IPv4 do gateway padrão no ***example*** conexão a **192.0.2.1**:

   ```
   nmcli> set ipv4.gateway 192.0.2.1
   ```

   Por exemplo, para definir o endereço IPv6 do gateway padrão no ***example*** conexão a **2001:db8:1::1**:

   ```
   nmcli> set ipv6.gateway 2001:db8:1::1
   ```

3. Opcionalmente, verificar se o gateway padrão foi configurado corretamente:

```
nmcli> print
...
ipv4.gateway:                192.0.2.1
...
ipv6.gateway:                2001:db8:1::1
...
```

4. Salvar a configuração:

```
nmcli> save persistent
```

5. Reinicie a conexão de rede para que as mudanças entrem em vigor:

```
nmcli> activate example
```

> **ATENÇÃO**
>
> Todas as conexões que atualmente utilizam esta conexão de rede são temporariamente interrompidas durante o reinício.

6. Deixe o modo interativo **nmcli**:

```
nmcli> quit
```

7. Opcionalmente, verificar se a rota está ativa.
   Para exibir o gateway padrão IPv4:

```
$ ip -4 route
default via 192.0.2.1 dev example proto static metric 100
```

   Para exibir o gateway padrão IPv6:

```
$ ip -6 route
default via 2001:db8:1::1 dev example proto static metric 100 pref medium
```

Recursos adicionais

- Seção 8.2, "Configuração de uma conexão Ethernet estática usando o editor interativo nmcli"

18.3. CONFIGURAÇÃO DO GATEWAY PADRÃO EM UMA CONEXÃO EXISTENTE USANDO O EDITOR DE CONEXÃO NM

Na maioria das situações, os administradores definem o gateway padrão quando criam uma conexão. Esta seção descreve como definir ou atualizar o gateway padrão em uma conexão previamente criada usando a aplicação **nm-connection-editor**.

Pré-requisitos

- Pelo menos um endereço IP estático deve ser configurado na conexão na qual o gateway padrão será configurado.

Procedimento

1. Abra um terminal e entre em **nm-connection-editor**:

```
$ nm-connection-editor
```

2. Selecione a conexão para modificar, e clique no ícone da roda dentada para editar a conexão existente.

3. Defina o gateway padrão IPv4. Por exemplo, para definir o endereço IPv4 do gateway padrão na conexão a **192.0.2.1**:

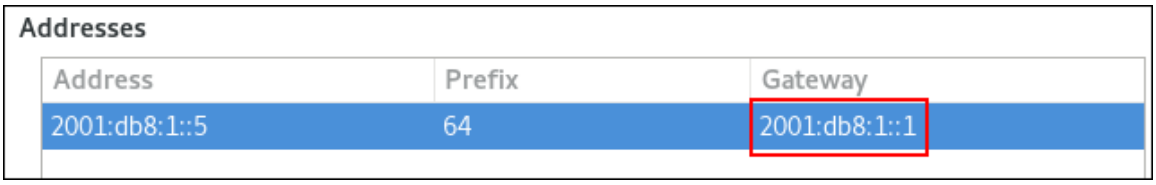
   a. Abra a aba **IPv4 Settings**.

   b. Digite o endereço no campo **gateway** ao lado da faixa de IP que o endereço do gateway está dentro:

4. Defina o gateway padrão IPv6. Por exemplo, para definir o endereço IPv6 do gateway padrão na conexão a **2001:db8:1::1**:

   a. Abra a aba **IPv6**.

   b. Digite o endereço no campo **gateway** ao lado da faixa de IP que o endereço do gateway está dentro:



5. Clique **OK**.

6. Clique em **Salvar**.

7. Reinicie a conexão de rede para que as mudanças entrem em vigor. Por exemplo, para reiniciar a ***example*** conexão usando a linha de comando:

```
$ sudo nmcli connection up example
```

> **ATENÇÃO**
>
> Todas as conexões que atualmente utilizam esta conexão de rede são temporariamente interrompidas durante o reinício.

8. Opcionalmente, verificar se a rota está ativa.
   Para exibir o gateway padrão IPv4:

```
$ ip -4 route
default via 192.0.2.1 dev example proto static metric 100
```

   Para exibir o gateway padrão IPv6:

```
$ ip -6 route
default via 2001:db8:1::1 dev example proto static metric 100 pref medium
```

Recursos adicionais

- Seção 8.8, "Configuração de uma conexão Ethernet usando um editor de conexão nm"

18.4. CONFIGURANDO O GATEWAY PADRÃO EM UMA CONEXÃO EXISTENTE USANDO O CENTRO DE CONTROLE

Na maioria das situações, os administradores definem o gateway padrão quando criam uma conexão. Esta seção descreve como definir ou atualizar o gateway padrão em uma conexão previamente criada usando a aplicação **control-center**.

Pré-requisitos

- Pelo menos um endereço IP estático deve ser configurado na conexão na qual o gateway padrão será configurado.
- A configuração de rede da conexão está aberta no aplicativo **control-center**.

Procedimento

1. Defina o gateway padrão IPv4. Por exemplo, para definir o endereço IPv4 do gateway padrão na conexão a **192.0.2.1**:
   a. Abra a aba **IPv4**.
   b. Digite o endereço no campo **gateway** ao lado da faixa de IP que o endereço do gateway está dentro:

2. Defina o gateway padrão IPv6. Por exemplo, para definir o endereço IPv6 do gateway padrão na conexão a **2001:db8:1::1**:

   a. Abra a aba **IPv6**.

   b. Digite o endereço no campo **gateway** ao lado da faixa de IP que o endereço do gateway está dentro:

3. Clique em **Aplicar**.

4. De volta à janela **Network**, desabilite e reative a conexão, mudando o botão para a conexão para **Desligado** e de volta para **Ligado** para que as mudanças tenham efeito.

> **ATENÇÃO**
>
> Todas as conexões que atualmente utilizam esta conexão de rede são temporariamente interrompidas durante o reinício.

5. Opcionalmente, verificar se a rota está ativa.
   Para exibir o gateway padrão IPv4:

```
$ ip -4 route
default via 192.0.2.1 dev example proto static metric 100
```

   Para exibir o gateway padrão IPv6:

```
$ ip -6 route
default via 2001:db8:1::1 dev example proto static metric 100 pref medium
```

Recursos adicionais

- Seção 8.8, “Configuração de uma conexão Ethernet usando um editor de conexão nm”

18.5. CONFIGURANDO O GATEWAY PADRÃO EM UMA CONEXÃO EXISTENTE USANDO AS FUNÇÕES DO SISTEMA

Você pode usar a função do Sistema RHEL **networking** para definir o gateway padrão.

IMPORTANTE

Quando você executa uma peça que usa o Sistema Função **networking** RHEL, o Sistema Função substitui um perfil de conexão existente com o mesmo nome se as configurações não coincidirem com as especificadas na peça. Portanto, sempre especifique toda a configuração do perfil de conexão de rede na peça, mesmo que, por exemplo, a configuração IP já exista. Caso contrário, o papel redefine estes valores com seus padrões.

Dependendo se já existe, o procedimento cria ou atualiza o perfil de conexão **enp1s0** com as seguintes configurações:

- Um endereço IPv4 estático - **198.51.100.20** com uma máscara de sub-rede **/24**
- Um endereço IPv6 estático - **2001:db8:1::1** com uma máscara de sub-rede **/64**
- Um gateway padrão IPv4 - **198.51.100.254**
- Um gateway padrão IPv6 - **2001:db8:1::fffe**
- Um servidor DNS IPv4 - **198.51.100.200**
- Um servidor DNS IPv6 - **2001:db8:1::ffbb**
- Um domínio de busca DNS - **example.com**

Pré-requisitos

- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.
- Se você usar um usuário remoto diferente de **root** ao executar o playbook, este usuário tem as permissões apropriadas **sudo** no nó gerenciado.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo **/etc/ansible/hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook **~/ethernet-connection.yml** com o seguinte conteúdo:

```
---
- name: Configure an Ethernet connection with static IP and default gateway
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network

    vars:
      network_connections:
        - name: enp1s0
          type: ethernet
```

```
    autoconnect: yes
    ip:
      address:
        - 198.51.100.20/24
        - 2001:db8:1::1/64
      gateway4: 198.51.100.254
      gateway6: 2001:db8:1::fffe
      dns:
        - 198.51.100.200
        - 2001:db8:1::ffbb
      dns_search:
        - example.com
    state: up
```

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

     ```
     # ansible-playbook -u root ~/ethernet-connection.yml
     ```

   - Para conectar-se como usuário ao host administrado, entre:

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/ethernet-connection.yml
     ```

     A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definido no **-u *user_name*** opção.

   Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Recursos adicionais

- Para detalhes sobre os parâmetros usados em **network_connections** e para informações adicionais sobre o Sistema de Papel **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

18.6. CONFIGURANDO O GATEWAY PADRÃO EM UMA CONEXÃO EXISTENTE AO UTILIZAR OS SCRIPTS DE REDE LEGADOS

Este procedimento descreve como configurar um gateway padrão quando você utiliza os scripts de rede legados. O exemplo define o gateway padrão para **192.0.2.1** que pode ser acessado através da interface **enp1s0**.

Pré-requisitos

- O pacote **NetworkManager** não está instalado, ou o serviço **NetworkManager** está desativado.
- O pacote **network-scripts** está instalado.

Procedimento

1. Defina o parâmetro **GATEWAY** no arquivo **/etc/sysconfig/network-scripts/ifcfg-enp1s0** para **192.0.2.1**:

```
GATEWAY=192.0.2.1
```

2. Adicione a entrada **default** no arquivo **/etc/sysconfig/network-scripts/route-enp0s1**:

```
padrão via 192.0.2.1
```

3. Reinicie a rede:

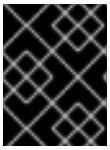
```
# rede de reinicialização systemctl
```

18.7. COMO O NETWORKMANAGER GERENCIA VÁRIOS GATEWAYS PADRÃO

Em certas situações, por exemplo, por razões de emergência, você define vários gateways padrão em um host. Entretanto, para evitar problemas de roteamento assíncrono, cada gateway padrão do mesmo protocolo requer um valor métrico separado. Observe que a RHEL só usa a conexão com o gateway padrão que tem o menor valor de métrica definido.

Você pode definir a métrica para o gateway IPv4 e IPv6 de uma conexão usando o seguinte comando:

```
# nmcli connection modify connection-name ipv4.route-metric value ipv6.route-metric value
```



IMPORTANTE

Não defina o mesmo valor métrico para o mesmo protocolo em vários perfis de conexão para evitar problemas de roteamento.

Se você definir um gateway padrão sem um valor métrico, o NetworkManager define automaticamente o valor métrico com base no tipo de interface. Para isso, o NetworkManager atribui o valor padrão deste tipo de rede à primeira conexão que é ativada, e define um valor incremental para uma conexão do mesmo tipo na ordem em que são ativadas. Por exemplo, se existirem duas conexões Ethernet com um gateway padrão, o NetworkManager define uma métrica de **100** na rota para o gateway padrão da conexão que você ativar primeiro. Para a segunda conexão, o NetworkManager define **101**.

A seguir, uma visão geral dos tipos de rede freqüentemente utilizados e suas métricas padrão:

Tipo de conexão	Valor métrico padrão
VPN	50
Ethernet	100
MACsec	125
InfiniBand	150
Bond	300

Tipo de conexão	Valor métrico padrão
Equipe	350
VLAN	400
Ponte	425
TUN	450
Wi-Fi	600
Túnel IP	675

Recursos adicionais

- Para detalhes sobre roteamento baseado em políticas, veja Capítulo 20, *Configuração de rotas baseadas em políticas para definir rotas alternativas*.
- Para detalhes sobre o Multipath TCP, veja Capítulo 25, *Começando com o Multipath TCP* .

18.8. CONFIGURAÇÃO DO NETWORKMANAGER PARA EVITAR O USO DE UM PERFIL ESPECÍFICO PARA FORNECER UM GATEWAY PADRÃO

Você pode configurar que o NetworkManager nunca utilize um perfil específico para fornecer o gateway padrão. Siga este procedimento para perfis de conexão que não estejam conectados ao gateway padrão.

Pré-requisitos

- O perfil de conexão NetworkManager para a conexão que não está conectada ao gateway padrão existe.

Procedimento

1. Se a conexão utiliza uma configuração IP dinâmica, configure que o NetworkManager não utilize a conexão como a rota padrão para conexões IPv4 e IPv6:

   ```
   # nmcli connection modify connection_name ipv4.never-default yes ipv6.never-default yes
   ```

   Observe que a configuração **ipv4.never-default** e **ipv6.never-default** para **yes**, remove automaticamente o endereço IP padrão do gateway para o protocolo correspondente do perfil de conexão.

2. Ativar a conexão:

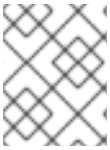
   ```
   # nmcli connection up connection_name
   ```

Etapas de verificação

- Utilize os comandos **ip -4 route** e **ip -6 route** para verificar se a RHEL não utiliza a interface de rede para a rota padrão para o protocolo IPv4 e IPv6.

18.9. CORRIGINDO COMPORTAMENTOS INESPERADOS DE ROTEAMENTO DEVIDO A MÚLTIPLOS GATEWAYS PADRÃO

Há apenas alguns cenários, como quando se usa TCP multipath, nos quais são necessários vários gateways padrão em um host. Na maioria dos casos, você configura apenas um único gateway padrão para evitar comportamento de roteamento inesperado ou problemas de roteamento assíncrono.



NOTA

Para rotear o tráfego para diferentes provedores de Internet, use roteamento baseado em políticas em vez de vários gateways padrão.

Pré-requisitos

- O host usa o NetworkManager para gerenciar as conexões de rede, que é o padrão.
- O host tem múltiplas interfaces de rede.
- O host tem vários gateways padrão configurados.

Procedimento

1. Exibir a tabela de roteamento:
   - Para IPv4, entre:

     ```
     # ip -4 route
     default via 192.0.2.1 dev enp1s0 proto static metric 101
     default via 198.51.100.1 dev enp7s0 proto static metric 102
     ...
     ```

   - Para IPv6, entre:

     ```
     # ip -6 route
     default via 2001:db8:1::1 dev enp1s0 proto static metric 101 pref medium
     default via 2001:db8:2::1 dev enp7s0 proto static metric 102 pref medium
     ...
     ```

   As entradas que começam com **default** indicam uma rota padrão. Observe os nomes das interfaces destas entradas exibidas ao lado de **dev**.

2. Use os seguintes comandos para exibir as conexões do NetworkManager que utilizam as interfaces que você identificou na etapa anterior:

   ```
   # nmcli -f GENERAL.CONNECTION,IP4.GATEWAY,IP6.GATEWAY device show enp1s0
   GENERAL.CONNECTION:      Corporate-LAN
   IP4.GATEWAY:             192.168.122.1
   IP6.GATEWAY:             2001:db8:1::1

   # nmcli -f GENERAL.CONNECTION,IP4.GATEWAY,IP6.GATEWAY device show enp7s0
   ```

```
GENERAL.CONNECTION:      Internet-Provider
IP4.GATEWAY:             198.51.100.1
IP6.GATEWAY:             2001:db8:2::1
```

Nestes exemplos, os perfis denominados **Corporate-LAN** e **Internet-Provider** têm os gateways padrão definidos. Como, em uma rede local, o gateway padrão é normalmente o host que está mais próximo da Internet, o restante deste procedimento assume que os gateways padrão no **Corporate-LAN** estão incorretos.

3. Configure que o NetworkManager não utilize a conexão **Corporate-LAN** como a rota padrão para conexões IPv4 e IPv6:

```
# nmcli connection modify Corporate-LAN ipv4.never-default yes ipv6.never-default yes
```

Observe que a configuração **ipv4.never-default** e **ipv6.never-default** para **yes**, remove automaticamente o endereço IP padrão do gateway para o protocolo correspondente do perfil de conexão.

4. Ativar a conexão **Corporate-LAN**:

```
# nmcli connection up Corporate-LAN
```

Etapas de verificação

- Exibir as tabelas de roteamento IPv4 e IPv6 e verificar se apenas um gateway padrão está disponível para cada protocolo:
  - Para IPv4, entre:

```
# ip -4 route
default via 192.0.2.1 dev enp1s0 proto static metric 101
...
```

  - Para IPv6, entre:

```
# ip -6 route
default via 2001:db8:1::1 dev enp1s0 proto static metric 101 pref medium
...
```

Recursos adicionais

- Para detalhes sobre roteamento baseado em políticas, veja Capítulo 20, *Configuração de rotas baseadas em políticas para definir rotas alternativas*.
- Para detalhes sobre o Multipath TCP, veja Capítulo 25, *Começando com o Multipath TCP* .

CAPÍTULO 19. CONFIGURAÇÃO DE ROTAS ESTÁTICAS

Por default, e se um gateway default for configurado, o Red Hat Enterprise Linux encaminha o tráfego para redes que não estão diretamente conectadas ao host para o gateway default. Usando uma rota estática, você pode configurar que o Red Hat Enterprise Linux encaminhe o tráfego para um host ou rede específica para um roteador diferente do gateway default. Esta seção descreve diferentes opções de como configurar rotas estáticas.

19.1. COMO USAR O COMANDO NMCLI PARA CONFIGURAR UMA ROTA ESTÁTICA

Para configurar uma rota estática, use o utilitário **nmcli** com a seguinte sintaxe:

```
$ nmcli connection modify connection_name ipv4.routes "ip[/prefix] [next_hop] [metric] [attribute=value] [attribute=value] ..."
```

O comando suporta os seguintes atributos de rota:

- **table=*n***
- **src=*address***
- **tos=*n***
- **onlink=true|false**
- **window=*n***
- **cwnd=*n***
- **mtu=*n***
- **lock-window=true|false**
- **lock-cwdn=true|false**
- **lock-mtu=true|false**

Se você usar o sub-comando **ipv4.routes**, **nmcli** anula todas as configurações atuais deste parâmetro. Para adicionar uma rota adicional, use o **nmcli connection modify *connection_name* ipv4.routes "..."** comando. De maneira semelhante, você pode usar **nmcli connection modify *connection_name* - ipv4.routes "..."** para remover uma rota específica.

19.2. CONFIGURAÇÃO DE UMA ROTA ESTÁTICA USANDO UM COMANDO NMCLI

Você pode adicionar uma rota estática à configuração de uma conexão de rede usando o comando **nmcli connection modify**.

O procedimento nesta seção descreve como adicionar uma rota à rede **192.0.2.0/24** que utiliza o gateway que funciona em **198.51.100.1**, que é acessível através da conexão **example**.

Pré-requisitos

- A rede é configurada
- A porta de entrada para a rota estática deve ser acessível diretamente na interface.
- Se o usuário estiver logado em um console físico, as permissões de usuário são suficientes. Caso contrário, o comando requer as permissões do **root**.

Procedimento

1. Acrescente a rota estática à conexão **example**:

```
$ sudo nmcli connection modify example ipv4.routes "192.0.2.0/24 198.51.100.1"
```

Para definir várias rotas em uma única etapa, passe as rotas individuais, separadas por vírgula, para o comando. Por exemplo, para adicionar uma rota para as redes **192.0.2.0/24** e **203.0.113.0/24**, ambas encaminhadas através do portal **198.51.100.1**, entre:

```
$ sudo nmcli connection modify example ipv4.routes "192.0.2.0/24 198.51.100.1,
203.0.113.0/24 198.51.100.1"
```

2. Opcionalmente, verificar se as rotas foram adicionadas corretamente à configuração:

```
$ nmcli connection show example
...
ipv4.routes:       { ip = 192.0.2.1/24, nh = 198.51.100.1 }
...
```

3. Reiniciar a conexão de rede:

```
$ sudo nmcli connection up example
```

ATENÇÃO

A reinicialização da conexão interrompe brevemente a conectividade nessa interface.

4. Opcionalmente, verificar se a rota está ativa:

```
$ ip route
...
192.0.2.0/24 via 198.51.100.1 dev example proto static metric 100
```

Recursos adicionais

- Para mais detalhes sobre **nmcli**, consulte a página de manual **nmcli(1)**.

19.3. CONFIGURAÇÃO DE UMA ROTA ESTÁTICA USANDO O CENTRO DE CONTROLE

Você pode usar **control-center** no GNOME para adicionar uma rota estática à configuração de uma conexão de rede.

O procedimento nesta seção descreve como adicionar uma rota à rede **192.0.2.0/24** que utiliza o gateway que funciona no site **198.51.100.1**.

Pré-requisitos

- A rede está configurada.
- A porta de entrada para a rota estática deve ser acessível diretamente na interface.
- A configuração de rede da conexão é aberta no aplicativo **control-center**. Ver Seção 8.8, "Configuração de uma conexão Ethernet usando um editor de conexão nm".

Procedimento

1. Abra a aba **IPv4**.
2. Opcionalmente, desabilite as rotas automáticas clicando no botão **On** na seção **Routes** da guia **IPv4** para usar apenas rotas estáticas. Se as rotas automáticas estiverem habilitadas, o Red Hat Enterprise Linux usa rotas estáticas e rotas recebidas de um servidor DHCP.
3. Digite o endereço, a máscara de rede, o gateway e, opcionalmente, um valor métrico:

4. Clique em **Aplicar**.
5. De volta à janela **Network**, desabilite e reative a conexão, mudando o botão para a conexão para **Desligado** e de volta para **Ligado** para que as mudanças tenham efeito.

> **ATENÇÃO**
>
> A reinicialização da conexão interrompe brevemente a conectividade nessa interface.

6. Opcionalmente, verificar se a rota está ativa:

```
$ ip route
...
192.0.2.0/24 via 198.51.100.1 dev example proto static metric 100
```

19.4. CONFIGURAÇÃO DE UMA ROTA ESTÁTICA USANDO UM EDITOR DE NM-CONEXÃO

Você pode usar o aplicativo **nm-connection-editor** para adicionar uma rota estática à configuração de uma conexão de rede.

O procedimento nesta seção descreve como adicionar uma rota à rede **192.0.2.0/24** que utiliza o gateway que funciona em **198.51.100.1**, que é acessível através da conexão **example**.

Pré-requisitos

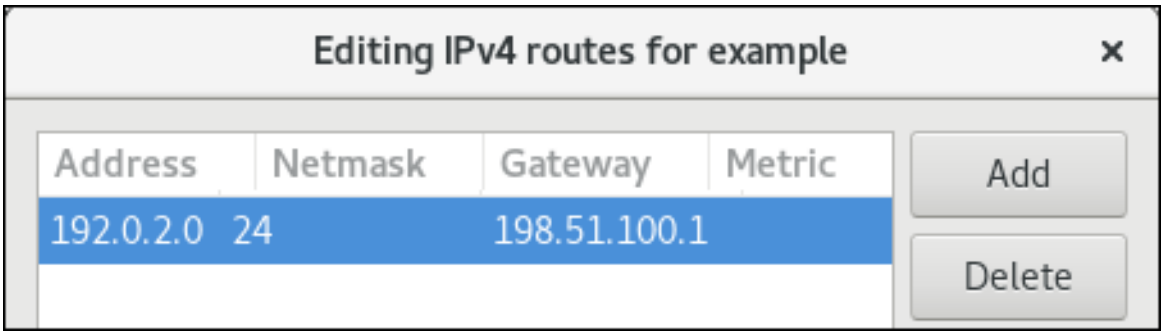
- A rede está configurada.
- A porta de entrada para a rota estática deve ser acessível diretamente na interface.

Procedimento

1. Abra um terminal e entre em **nm-connection-editor**:

```
$ nm-connection-editor
```

2. Selecione a conexão **example** e clique no ícone da roda dentada para editar a conexão existente.
3. Abra a aba **IPv4**.
4. Clique no botão **Routes (Rotas)**.
5. Clique no botão **Adicionar** e digite o endereço, a máscara de rede, o gateway e, opcionalmente, um valor métrico.



6. Clique **OK**.
7. Clique em **Salvar**.
8. Reinicie a conexão de rede para que as mudanças entrem em vigor. Por exemplo, para reiniciar a conexão **example** usando a linha de comando:

```
$ sudo nmcli connection up example
```

9. Opcionalmente, verificar se a rota está ativa:

```
$ ip route
...
192.0.2.0/24 via 198.51.100.1 dev example proto static metric 100
```

19.5. CONFIGURAÇÃO DE UMA ROTA ESTÁTICA USANDO O MODO INTERATIVO NMCLI

Você pode usar o modo interativo do utilitário **nmcli** para adicionar uma rota estática à configuração de uma conexão de rede.

O procedimento nesta seção descreve como adicionar uma rota à rede **192.0.2.0/24** que utiliza o gateway que funciona em **198.51.100.1**, que é acessível através da conexão **example**.

Pré-requisitos

- A rede é configurada
- A porta de entrada para a rota estática deve ser acessível diretamente na interface.
- Se o usuário estiver logado em um console físico, as permissões de usuário são suficientes. Caso contrário, o comando requer as permissões do **root**.

Procedimento

1. Abra o modo interativo **nmcli** para a conexão **example**:

```
$ sudo nmcli connection edit example
```

2. Acrescente a rota estática:

```
nmcli> set ipv4.routes 192.0.2.0/24 198.51.100.1
```

3. Opcionalmente, verificar se as rotas foram adicionadas corretamente à configuração:

```
nmcli> print
...
ipv4.routes:       { ip = 192.0.2.1/24, nh = 198.51.100.1 }
...
```

   O atributo **ip** exibe a rede para rotear e o atributo **nh** atribui o gateway (próximo salto).

4. Salvar a configuração:

```
nmcli> save persistent
```

5. Reiniciar a conexão de rede:

```
nmcli> activate example
```

ATENÇÃO

Quando você reiniciar a conexão, todas as conexões que atualmente utilizam esta conexão serão temporariamente interrompidas.

6. Deixe o modo interativo **nmcli**:

```
nmcli> quit
```

7. Opcionalmente, verificar se a rota está ativa:

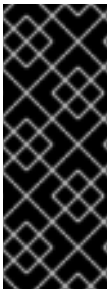
```
$ ip route
...
192.0.2.0/24 via 198.51.100.1 dev example proto static metric 100
```

Recursos adicionais

- Para obter a lista de comandos disponíveis no modo interativo, digite **help** na shell interativa.

19.6. CONFIGURAÇÃO DE UMA ROTA ESTÁTICA USANDO AS FUNÇÕES DO SISTEMA RHEL

Você pode usar o sistema **networking** RHEL Role para configurar rotas estáticas.



IMPORTANTE

Quando você executa uma peça que usa o Sistema Função **networking** RHEL, o Sistema Função substitui um perfil de conexão existente com o mesmo nome se as configurações não coincidirem com as especificadas na peça. Portanto, sempre especifique toda a configuração do perfil de conexão de rede na peça, mesmo que, por exemplo, a configuração IP já exista. Caso contrário, o papel redefine estes valores com seus padrões.

Dependendo se já existe, o procedimento cria ou atualiza o perfil de conexão **enp7s0** com as seguintes configurações:

- Um endereço IPv4 estático - **198.51.100.20** com uma máscara de sub-rede **/24**
- Um endereço IPv6 estático - **2001:db8:1::1** com uma máscara de sub-rede **/64**
- Um gateway padrão IPv4 - **198.51.100.254**
- Um gateway padrão IPv6 - **2001:db8:1::fffe**
- Um servidor DNS IPv4 - **198.51.100.200**
- Um servidor DNS IPv6 - **2001:db8:1::ffbb**
- Um domínio de busca DNS - **example.com**
- Rotas estáticas:
  - **192.0.2.0/24** com gateway **198.51.100.1**
  - **203.0.113.0/24** com gateway **198.51.100.2**

Pré-requisitos

- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.

- Se você usar um usuário remoto diferente de **root** ao executar o playbook, este usuário tem as permissões apropriadas **sudo** no nó gerenciado.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo **/etc/ansible/hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook **~/add-static-routes.yml** com o seguinte conteúdo:

```
---
- name: Configure an Ethernet connection with static IP and additional routes
  hosts: node.example.com
  become: true
  tasks:
  - include_role:
      name: linux-system-roles.network

    vars:
      network_connections:
        - name: enp7s0
          type: ethernet
          autoconnect: yes
          ip:
            address:
              - 198.51.100.20/24
              - 2001:db8:1::1/64
            gateway4: 198.51.100.254
            gateway6: 2001:db8:1::fffe
            dns:
              - 198.51.100.200
              - 2001:db8:1::ffbb
            dns_search:
              - example.com
            route:
              - network: 192.0.2.0
                prefix: 24
                gateway: 198.51.100.1
              - network: 203.0.113.0
                prefix: 24
                gateway: 198.51.100.2
          state: up
```

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

```
# ansible-playbook -u root ~/add-static-routes.yml
```

   - Para conectar-se como usuário ao host administrado, entre:

```
# ansible-playbook -u user_name --ask-become-pass ~/add-static-routes.yml
```

-

A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definido no **-u *user_name*** opção.

Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Etapas de verificação

- Exibir a tabela de roteamento:

```
# ip -4 route
default via 198.51.100.254 dev enp7s0 proto static metric 100
192.0.2.0/24 via 198.51.100.1 dev enp7s0 proto static metric 100
203.0.113.0/24 via 198.51.100.2 dev enp7s0 proto static metric 100
...
```

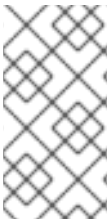
Recursos adicionais

- Para detalhes sobre os parâmetros usados em **network_connections** e para informações adicionais sobre o Sistema de Papel **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

19.7. CRIAÇÃO DE ARQUIVOS DE CONFIGURAÇÃO DE ROTAS ESTÁTICAS EM FORMATO DE VALOR CHAVE AO UTILIZAR OS SCRIPTS DE REDE LEGADOS

Este procedimento descreve como criar manualmente um arquivo de configuração de roteamento para uma rota IPv4 para a rede **192.0.2.0/24** quando você utiliza os scripts de rede legados em vez do NetworkManager. Neste exemplo, o gateway correspondente com o endereço IP **198.51.100.1** pode ser acessado através da interface **enp1s0**.

O exemplo neste procedimento utiliza entradas de configuração em formato de valores-chave.



NOTA

Os scripts de rede legados suportam o formato do valor chave somente para rotas IPv4 estáticas. Para rotas IPv6, use o formato **ip**-command. Veja Seção 19.8, “Criação de arquivos de configuração de rotas estáticas em formato ip-command ao utilizar os scripts de rede legados”.

Pré-requisitos

- A porta de entrada para a rota estática deve ser acessível diretamente na interface.
- O pacote **NetworkManager** não está instalado, ou o serviço **NetworkManager** está desativado.
- O pacote **network-scripts** está instalado.

Procedimento

1. Adicione a rota estática IPv4 ao arquivo **/etc/sysconfig/network-scripts/route-enp0s1**:

```
ADDRESS0=192.0.2.0
NETMASK0=255.255.255.0
GATEWAY0=198.51.100.1
```

   - A variável **ADDRESS0** define a rede da primeira entrada de roteamento.
   - A variável **NETMASK0** define a máscara da rede da primeira entrada de roteamento.
   - A variável **GATEWAY0** define o endereço IP do gateway para a rede remota ou host para a primeira entrada de roteamento.
     Se você adicionar múltiplas rotas estáticas, aumente o número nos nomes das variáveis. Observe que as variáveis para cada rota devem ser numeradas seqüencialmente. Por exemplo, **ADDRESS0**, **ADDRESS1**, **ADDRESS3**, e assim por diante.

2. Reinicie a rede:

```
# rede de reinicialização systemctl
```

Recursos adicionais

- Para mais detalhes sobre a configuração de scripts de rede legados, consulte o arquivo **/usr/share/doc/network-scripts/sysconfig.txt**.

19.8. CRIAÇÃO DE ARQUIVOS DE CONFIGURAÇÃO DE ROTAS ESTÁTICAS EM FORMATO IP-COMMAND AO UTILIZAR OS SCRIPTS DE REDE LEGADOS

Este procedimento descreve como criar manualmente um arquivo de configuração de roteamento para as seguintes rotas estáticas quando você utiliza scripts de rede legados:

- Uma rota IPv4 para a rede **192.0.2.0/24**. O gateway correspondente com o endereço IP **198.51.100.1** é acessível através da interface **enp1s0**.
- Uma rota IPv6 para a rede **2001:db8:1::/64**. O gateway correspondente com o endereço IP **2001:db8:2::1** é acessível através da interface **enp1s0**.

O exemplo neste procedimento utiliza entradas de configuração no formato **ip**-command-format.

Pré-requisitos

- A porta de entrada para a rota estática deve ser acessível diretamente na interface.
- O pacote **NetworkManager** não está instalado, ou o serviço **NetworkManager** está desativado.
- O pacote **network-scripts** está instalado.

Procedimento

1. Adicione a rota estática IPv4 ao arquivo **/etc/sysconfig/network-scripts/route-enp0s1**:

```
192.0.2.0/24 via 198.51.100.1 dev enp0s1
```

2. Adicione a rota IPv6 estática ao arquivo **/etc/sysconfig/network-scripts/route6-enp0s1**:

```
2001:db8:1::/64 via 2001:db8:2::1 dev enp0s1
```

3. Reinicie a rede:

```
# rede de reinicialização systemctl
```

Recursos adicionais

- Para mais detalhes sobre a configuração de scripts de rede legados, consulte o arquivo **/usr/share/doc/network-scripts/sysconfig.txt**.

CAPÍTULO 20. CONFIGURAÇÃO DE ROTAS BASEADAS EM POLÍTICAS PARA DEFINIR ROTAS ALTERNATIVAS

Por padrão, o kernel na RHEL decide onde encaminhar os pacotes de rede com base no endereço de destino usando uma tabela de roteamento. O roteamento baseado em políticas permite a configuração de cenários complexos de roteamento. Por exemplo, você pode encaminhar pacotes com base em vários critérios, como o endereço de origem, metadados de pacotes ou protocolo.

Esta seção descreve como configurar o roteamento baseado em políticas usando o NetworkManager.

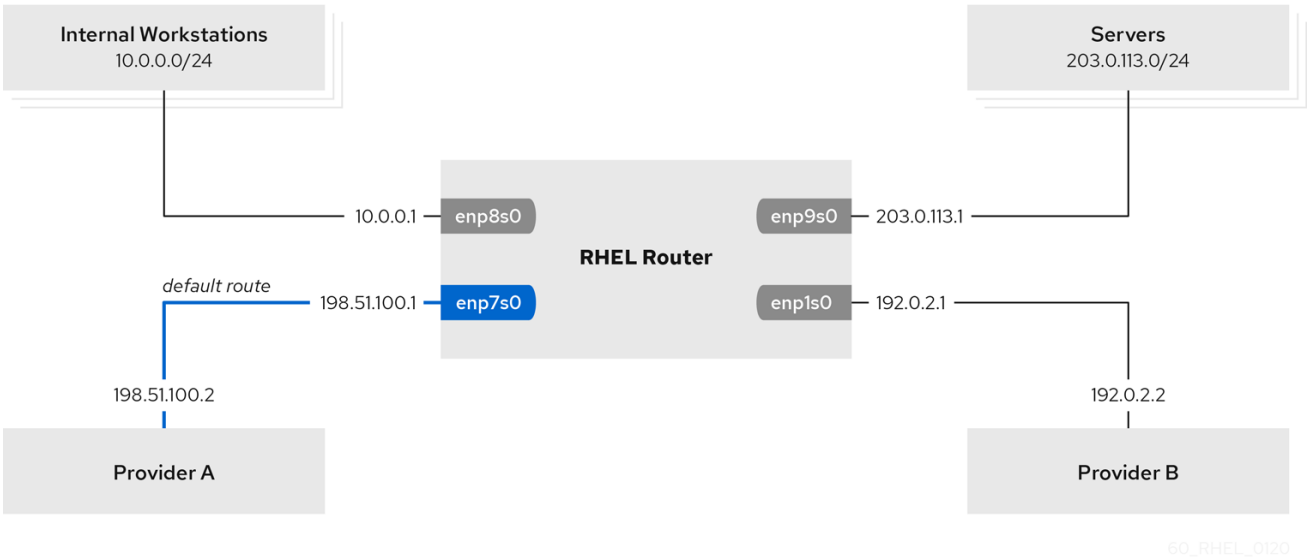
NOTA

Em sistemas que utilizam o NetworkManager, apenas o utilitário **nmcli** suporta a definição de regras de roteamento e a atribuição de rotas a tabelas específicas.

20.1. ROTEAMENTO DO TRÁFEGO DE UMA SUB-REDE ESPECÍFICA PARA UM GATEWAY PADRÃO DIFERENTE USANDO O NETWORKMANAGER

Esta seção descreve como configurar o RHEL como um roteador que, por padrão, encaminha todo o tráfego para o provedor de Internet A usando a rota padrão. Usando o roteamento baseado em políticas, a RHEL encaminha o tráfego recebido da sub-rede interna das estações de trabalho para o provedor B.

O procedimento assume a seguinte topologia de rede:



Pré-requisitos

- O sistema usa **NetworkManager** para configurar a rede, que é o padrão no RHEL 8.
- O roteador RHEL que você deseja instalar no procedimento tem quatro interfaces de rede:
  - A interface **enp7s0** está conectada à rede do provedor A. O gateway IP da rede do provedor é **198.51.100.2**, e a rede usa uma máscara de rede **/30**.

- A interface **enp1s0** está conectada à rede do provedor B. O gateway IP da rede do provedor é **192.0.2.2**, e a rede usa uma máscara de rede **/30**.
- A interface **enp8s0** está conectada à sub-rede **10.0.0.0/24** com estações de trabalho internas.
- A interface **enp9s0** está conectada à sub-rede **203.0.113.0/24** com os servidores da empresa.

- Os anfitriões na sub-rede interna das estações de trabalho utilizam **10.0.0.1** como o gateway padrão. No procedimento, você atribui este endereço IP à interface de rede **enp8s0** do roteador.
- Os anfitriões na sub-rede do servidor utilizam **203.0.113.1** como porta de entrada padrão. No procedimento, você atribui este endereço IP para a interface de rede **enp9s0** do roteador.
- O serviço **firewalld** está habilitado e ativo.

Procedimento

1. Configurar a interface de rede para o provedor A:

```
# nmcli connection add type ethernet con-name Provider-A ifname enp7s0
ipv4.method manual ipv4.addresses 198.51.100.1/30 ipv4.gateway 198.51.100.2
ipv4.dns 198.51.100.200 connection.zone external
```

O comando **nmcli connection add** cria um perfil de conexão NetworkManager. A lista a seguir descreve as opções do comando:

- **type ethernet** : Define que o tipo de conexão é Ethernet.
- **con-name*connection_name***: Define o nome do perfil. Use um nome significativo para evitar confusão.
- **ifname*network_device***: Define a interface da rede.
- **ipv4.method manual** : Permite configurar um endereço IP estático.
- **ipv4.addresses*IP_address*/*subnet_mask***: Define os endereços IPv4 e a máscara de sub-rede.
- **ipv4.gateway*IP_address***: Define o endereço padrão do gateway.
- **ipv4.dns*IP_of_DNS_server***: Define o endereço IPv4 do servidor DNS.
- **connection.zone*firewalld_zone***: Atribui a interface de rede à zona **firewalld** definida. Note que **firewalld** permite automaticamente o mascaramento para as interfaces atribuídas à zona **external**.

2. Configurar a interface de rede para o provedor B:

```
# nmcli connection add type ethernet con-name Provider-B ifname enp1s0
ipv4.method manual ipv4.addresses 192.0.2.1/30 ipv4.routes "0.0.0.0/0 192.0.2.2
table=5000" connection.zone external
```

Este comando usa o parâmetro **ipv4.routes** ao invés de **ipv4.gateway** para definir o gateway padrão. Isto é necessário para atribuir o gateway padrão para esta conexão a uma tabela de

roteamento diferente (**5000**) do que o padrão. O NetworkManager cria automaticamente esta nova tabela de roteamento quando a conexão é ativada.

3. Configurar a interface da rede para a subrede interna das estações de trabalho:

```
# nmcli connection add type ethernet con-name Internal-Workstations ifname enp8s0
ipv4.method manual ipv4.addresses 10.0.0.1/24 ipv4.routes "10.0.0.0/24 src=192.0.2.1
table=5000" ipv4.routing-rules "priority 5 from 10.0.0.0/24 table 5000" connection.zone
internal
```

Este comando usa o parâmetro **ipv4.routes** para adicionar uma rota estática à tabela de roteamento com ID **5000**. Esta rota estática para a sub-rede **10.0.0.0/24** usa o IP da interface da rede local para o provedor B (**192.0.2.1**) como próximo salto.

Além disso, o comando usa o parâmetro **ipv4.routing-rules** para adicionar uma regra de roteamento com prioridade **5** que roteia o tráfego da subrede **10.0.0.0/24** para a tabela **5000**. Valores baixos têm uma prioridade alta.

Observe que a sintaxe no parâmetro **ipv4.routing-rules** é a mesma de um comando **ip route add**, exceto que **ipv4.routing-rules** sempre requer a especificação de uma prioridade.

4. Configurar a interface de rede para a sub-rede do servidor:

```
# nmcli connection add type ethernet con-name Servers ifname enp9s0 ipv4.method
manual ipv4.addresses 203.0.113.1/24 connection.zone internal
```

Etapas de verificação

1. Em um host RHEL na sub-rede interna da estação de trabalho:
   a. Instale o pacote **traceroute**:

```
# yum install traceroute
```

   b. Use o utilitário **traceroute** para exibir a rota para um host na Internet:

```
# traceroute redhat.com
traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)     0.337 ms  0.260 ms  0.223 ms
 2  192.0.2.1 (192.0.2.1)   0.884 ms  1.066 ms  1.248 ms
 ...
```

   A saída do comando mostra que o roteador envia pacotes sobre **192.0.2.1**, que é a rede do provedor B.

2. Em um host RHEL na sub-rede do servidor:
   a. Instale o pacote **traceroute**:

```
# yum install traceroute
```

   b. Use o utilitário **traceroute** para exibir a rota para um host na Internet:

```
# traceroute redhat.com
traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
```

```
1  203.0.113.1 (203.0.113.1)    2.179 ms  2.073 ms  1.944 ms
2  198.51.100.2 (198.51.100.2)  1.868 ms  1.798 ms  1.549 ms
...
```

A saída do comando mostra que o roteador envia pacotes através de **198.51.100.2**, que é a rede do provedor A.

Passos para a solução de problemas

No roteador RHEL:

1. Exibir a lista de regras:

```
# ip rule list
0: from all lookup local
5: from 10.0.0.0/24 lookup 5000
32766: from all lookup main
32767: from all lookup default
```

Por padrão, a RHEL contém regras para as tabelas **local**, **main**, e **default**.

2. Mostrar as rotas na tabela **5000**:

```
# ip route list table 5000
0.0.0.0/0 via 192.0.2.2 dev enp1s0 proto static metric 100
10.0.0.0/24 dev enp8s0 proto static scope link src 192.0.2.1 metric 102
```

3. Exibir as interfaces e as zonas de firewall:

```
# firewall-cmd --get-active-zones
external
  interfaces: enp1s0 enp7s0
internal
  interfaces: enp8s0 enp9s0
```

4. Verifique se a zona **external** tem o mascaramento ativado:

```
# firewall-cmd --info-zone=external
external (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp1s0 enp7s0
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: yes
  ...
```

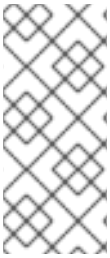
Recursos adicionais

- Para mais detalhes sobre os parâmetros **ipv4.*** você pode definir no comando **nmcli connection add**, veja a seção **IPv4 settings** na página de manual **nm-settings(5)**.

- Para mais detalhes sobre os parâmetros **connection.*** você pode definir no comando **nmcli connection add**, veja a seção **Connection settings** na página de manual **nm-settings(5)**.
- Para mais detalhes sobre o gerenciamento de conexões NetworkManager usando **nmcli**, consulte a seção **Connection management commands** na página de manual **nmcli(1)**.

20.2. VISÃO GERAL DOS ARQUIVOS DE CONFIGURAÇÃO ENVOLVIDOS NO ROTEAMENTO BASEADO EM POLÍTICAS AO UTILIZAR OS SCRIPTS DE REDE LEGADOS

Se você usar os scripts de rede legados em vez do NetworkManager para configurar sua rede, você também pode configurar o roteamento baseado em políticas.



NOTA

A configuração da rede usando os scripts de rede legados fornecidos pelo pacote **network-scripts** é depreciada no RHEL 8. A Red Hat recomenda que você use o NetworkManager para configurar o roteamento baseado em políticas. Para um exemplo, veja Seção 20.1, “Roteamento do tráfego de uma sub-rede específica para um gateway padrão diferente usando o NetworkManager”.

Os seguintes arquivos de configuração estão envolvidos no roteamento baseado em políticas quando você utiliza os scripts de rede legados:

- **/etc/sysconfig/network-scripts/route-*interface***: Este arquivo define as rotas IPv4. Use a opção **table** para especificar a tabela de roteamento. Por exemplo:

```
192.0.2.0/24 via 198.51.100.1 table 1
203.0.113.0/24 via 198.51.100.2 table 2
```

- **/etc/sysconfig/network-scripts/route6-*interface***: Este arquivo define as rotas IPv6.
- **/etc/sysconfig/network-scripts/rule-*interface***: Este arquivo define as regras para as redes de origem IPv4 para as quais o kernel encaminha o tráfego para tabelas de roteamento específicas. Por exemplo, este arquivo define as regras para redes IPv4:

```
from 192.0.2.0/24 lookup 1
from 203.0.113.0/24 lookup 2
```

- **/etc/sysconfig/network-scripts/rule6-*interface***: Este arquivo define as regras para redes de origem IPv6 para as quais o kernel encaminha o tráfego para tabelas de roteamento específicas.
- **/etc/iproute2/rt_tables**: Este arquivo define os mapeamentos se você quiser usar nomes em vez de números para se referir a tabelas de roteamento específicas. Por exemplo, o arquivo de mapeamento:

```
1    Provider_A
2    Provider_B
```

Recursos adicionais

- Para mais detalhes sobre roteamento IP, consulte a página de manual **ip-route(8)**.

- Para mais detalhes sobre as regras de roteamento, consulte a página de manual **ip-rule(8)**.

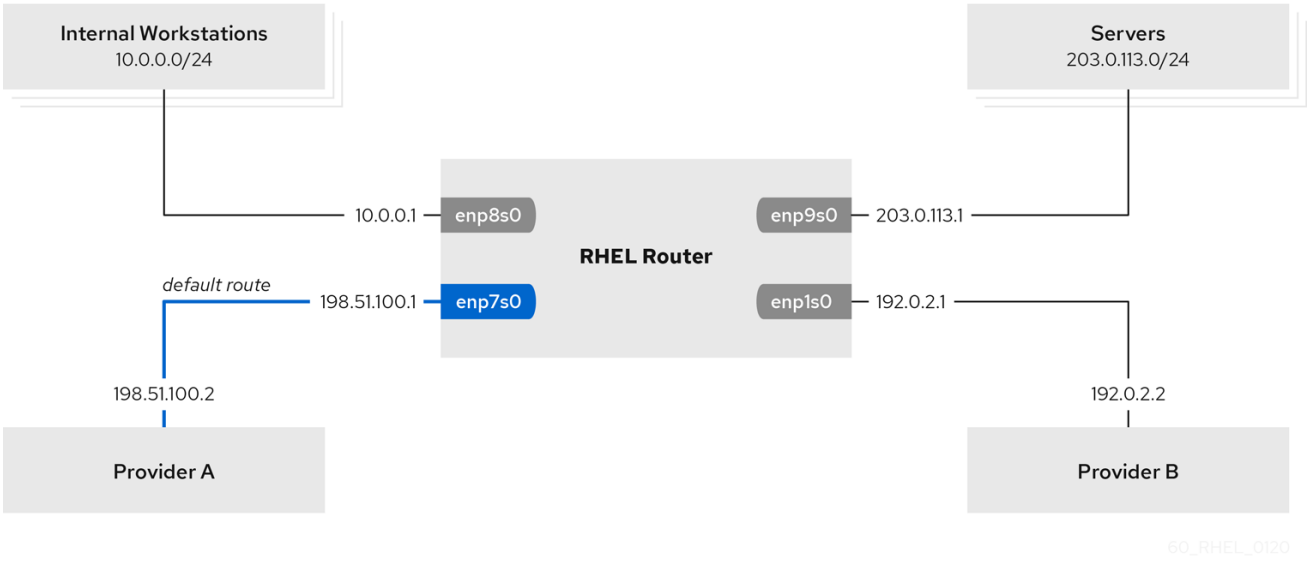
20.3. ROTEAMENTO DO TRÁFEGO DE UMA SUBREDE ESPECÍFICA PARA UM GATEWAY PADRÃO DIFERENTE USANDO OS SCRIPTS DE REDE LEGADOS

Esta seção descreve como configurar o RHEL como um roteador que, por padrão, encaminha todo o tráfego para o provedor de Internet A usando a rota padrão. Usando o roteamento baseado em políticas, a RHEL encaminha o tráfego recebido da sub-rede interna das estações de trabalho para o provedor B.

IMPORTANTE

A configuração da rede usando os scripts de rede legados fornecidos pelo pacote **network-scripts** é depreciada no RHEL 8. Siga o procedimento desta seção somente se você usar os scripts de rede legados em vez do NetworkManager em seu host. Se você usar o NetworkManager para gerenciar suas configurações de rede, veja Seção 20.1, “Roteamento do tráfego de uma sub-rede específica para um gateway padrão diferente usando o NetworkManager”.

O procedimento assume a seguinte topologia de rede:



NOTA

Os scripts de rede legados processam os arquivos de configuração em ordem alfabética. Portanto, é necessário nomear os arquivos de configuração de forma a garantir que uma interface, que é utilizada em regras e rotas de outras interfaces, esteja pronta quando uma interface dependente a requer. Para realizar a ordem correta, este procedimento utiliza números nos arquivos **ifcfg-***, **route-***, e **rules-***.

Pré-requisitos

- O pacote **NetworkManager** não está instalado, ou o serviço **NetworkManager** está desativado.
- O pacote **network-scripts** está instalado.

- O roteador RHEL que você deseja instalar no procedimento tem quatro interfaces de rede:
  - A interface **enp7s0** está conectada à rede do provedor A. O gateway IP da rede do provedor é **198.51.100.2**, e a rede usa uma máscara de rede **/30**.
  - A interface **enp1s0** está conectada à rede do provedor B. O gateway IP da rede do provedor é **192.0.2.2**, e a rede usa uma máscara de rede **/30**.
  - A interface **enp8s0** está conectada à sub-rede **10.0.0.0/24** com estações de trabalho internas.
  - A interface **enp9s0** está conectada à sub-rede **203.0.113.0/24** com os servidores da empresa.
- Os anfitriões na sub-rede interna das estações de trabalho utilizam **10.0.0.1** como o gateway padrão. No procedimento, você atribui este endereço IP à interface de rede **enp8s0** do roteador.
- Os anfitriões na sub-rede do servidor utilizam **203.0.113.1** como porta de entrada padrão. No procedimento, você atribui este endereço IP para a interface de rede **enp9s0** do roteador.
- O serviço **firewalld** está habilitado e ativo.

Procedimento

1. Adicione a configuração da interface de rede ao provedor A, criando o arquivo **/etc/sysconfig/network-scripts/ifcfg-1_Provider-A** com o seguinte conteúdo:

```
TYPE=Ethernet
IPADDR=198.51.100.1
PREFIX=30
GATEWAY=198.51.100.2
DNS1=198.51.100.200
DEFROUTE=yes
NAME=1_Provider-A
DEVICE=enp7s0
ONBOOT=yes
ZONE=external
```

   A lista a seguir descreve os parâmetros usados no arquivo de configuração:

   - **TYPE**=**Ethernet**: Define que o tipo de conexão é Ethernet.
   - **IPADDR**=***IP_address***: Define o endereço IPv4.
   - **PREFIX**=***subnet_mask***: Define a máscara de sub-rede.
   - **GATEWAY**=***IP_address***: Define o endereço padrão do gateway.
   - **DNS1**=***IP_of_DNS_server***: Define o endereço IPv4 do servidor DNS.
   - **DEFROUTE**=***yes***|***no***: Define se a conexão é ou não uma rota padrão.
   - **NAME**=***connection_name***: Define o nome do perfil de conexão. Use um nome significativo para evitar confusão.
   - **DEVICE**=***network_device***: Define a interface da rede.

- **ONBOOT**=**yes**: Define que a RHEL inicia esta conexão quando o sistema inicia.
- **ZONE**=***firewalld_zone***: Atribui a interface de rede à zona **firewalld** definida. Note que **firewalld** permite automaticamente o mascaramento para as interfaces atribuídas à zona **external**.

2. Adicionar a configuração para a interface de rede ao provedor B:

   a. Crie o arquivo /**etc**/**sysconfig**/**network-scripts**/**ifcfg-2_Provider-B** com o seguinte conteúdo:

   ```
   TYPE=Ethernet
   IPADDR=192.0.2.1
   PREFIX=30
   DEFROUTE=no
   NAME=2_Provider-B
   DEVICE=enp1s0
   ONBOOT=yes
   ZONE=external
   ```

   Observe que o arquivo de configuração para esta interface não contém uma configuração padrão de gateway.

   b. Atribuir o gateway para a conexão **2_Provider-B** a uma tabela de roteamento separada. Portanto, crie o arquivo /**etc**/**sysconfig**/**network-scripts**/**route-2_Provider-B** com o seguinte conteúdo:

   ```
   0.0.0.0/0 via 192.0.2.2 tabela 5000
   ```

   Esta entrada atribui a porta de entrada e o tráfego de todas as sub-redes roteadas através desta porta de entrada à mesa **5000.**

3. Criar a configuração da interface da rede para a sub-rede interna das estações de trabalho:

   a. Crie o arquivo /**etc**/**sysconfig**/**network-scripts**/**ifcfg-3_Internal-Workstations** com o seguinte conteúdo:

   ```
   TYPE=Ethernet
   IPADDR=10.0.0.1
   PREFIX=24
   DEFROUTE=no
   NAME=3_Internal-Workstations
   DEVICE=enp8s0
   ONBOOT=yes
   ZONE=internal
   ```

   b. Adicionar a configuração da regra de roteamento para a sub-rede interna da estação de trabalho. Portanto, crie o arquivo /**etc**/**sysconfig**/**network-scripts**/**rule-3_Internal-Workstations** com o seguinte conteúdo:

   ```
   pri 5 a partir de 10.0.0.0/24 tabela 5000
   ```

   Esta configuração define uma regra de roteamento com prioridade **5** que encaminha todo o tráfego da sub-rede **10.0.0.0**/**24** para a tabela **5000**. Valores baixos têm uma prioridade alta.

c. Crie o arquivo **/etc/sysconfig/network-scripts/route-3_Internal-Workstations** com o seguinte conteúdo para adicionar uma rota estática à tabela de roteamento com ID **5000**:

```
10.0.0.0/24 via 192.0.2.1 tabela 5000
```

Esta rota estática define que a RHEL envia tráfego da sub-rede **10.0.0.0/24** para o IP da interface da rede local para o provedor B (**192.0.2.1**). Esta interface é para a tabela de roteamento **5000** e usada como o próximo salto.

4. Adicione a configuração para a interface de rede à sub-rede do servidor criando o arquivo **/etc/sysconfig/network-scripts/ifcfg-4_Servers** com o seguinte conteúdo:

```
TYPE=Ethernet
IPADDR=203.0.113.1
PREFIX=24
DEFROUTE=no
NAME=4_Servers
DEVICE=enp9s0
ONBOOT=yes
ZONE=internal
```

5. Reinicie a rede:

```
# systemctl restart network
```

Etapas de verificação

1. Em um host RHEL na sub-rede interna da estação de trabalho:

a. Instale o pacote **traceroute**:

```
# yum install traceroute
```

b. Use o utilitário **traceroute** para exibir a rota para um host na Internet:

```
# traceroute redhat.com
traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
 1  10.0.0.1 (10.0.0.1)     0.337 ms  0.260 ms  0.223 ms
 2  192.0.2.1 (192.0.2.1)   0.884 ms  1.066 ms  1.248 ms
 ...
```

A saída do comando mostra que o roteador envia pacotes sobre **192.0.2.1**, que é a rede do provedor B.

2. Em um host RHEL na sub-rede do servidor:

a. Instale o pacote **traceroute**:

```
# yum install traceroute
```

b. Use o utilitário **traceroute** para exibir a rota para um host na Internet:

```
# traceroute redhat.com
traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
```

```
1  203.0.113.1 (203.0.113.1)    2.179 ms  2.073 ms  1.944 ms
2  198.51.100.2 (198.51.100.2)  1.868 ms  1.798 ms  1.549 ms
...
```

A saída do comando mostra que o roteador envia pacotes através de **198.51.100.2**, que é a rede do provedor A.

Passos para a solução de problemas

No roteador RHEL:

1. Exibir a lista de regras:

```
# ip rule list
0:      from all lookup local
5:      from 10.0.0.0/24 lookup 5000
32766:  from all lookup main
32767:  from all lookup default
```

Por padrão, a RHEL contém regras para as tabelas **local**, **main**, e **default**.

2. Mostrar as rotas na tabela **5000**:

```
# ip route list table 5000
default via 192.0.2.2 dev enp1s0
10.0.0.0/24 via 192.0.2.1 dev enp1s0
```

3. Exibir as interfaces e as zonas de firewall:

```
# firewall-cmd --get-active-zones
external
  interfaces: enp1s0 enp7s0
internal
  interfaces: enp8s0 enp9s0
```

4. Verifique se a zona **external** tem o mascaramento ativado:

```
# firewall-cmd --info-zone=external
external (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp1s0 enp7s0
  sources:
  services: ssh
  ports:
  protocols:
  masquerade: yes
  ...
```

Recursos adicionais

- Seção 20.2, “Visão geral dos arquivos de configuração envolvidos no roteamento baseado em políticas ao utilizar os scripts de rede legados”
- A página do homem **ip-route(8)**

- A página do homem **ip-rule(8)**
- Para mais detalhes sobre os scripts de rede legados, veja o arquivo **/usr/share/doc/network-scripts/sysconfig.txt**

CAPÍTULO 21. CRIANDO UMA INTERFACE FICTÍCIA

Como usuário do Red Hat Enterprise Linux, você pode criar e usar interfaces de rede fictícias para fins de depuração e teste. Uma interface fictícia fornece um dispositivo para rotear os pacotes sem realmente transmiti-los. Ela permite que você crie dispositivos adicionais do tipo loopback gerenciados pelo NetworkManager e faz com que um endereço SLIP (Serial Line Internet Protocol) inativo pareça um endereço real para programas locais.

21.1. CRIAÇÃO DE UMA INTERFACE FICTÍCIA COM UM ENDEREÇO IPV4 E IPV6 USANDO NMCLI

Você pode criar uma interface fictícia com várias configurações. Este procedimento descreve como criar uma interface fictícia com um endereço IPv4 e IPv6. Após criar a interface fictícia, o NetworkManager a atribui automaticamente à zona padrão do firewall **public**.

NOTA

Para configurar uma interface fictícia sem endereço IPv4 ou IPv6, defina os parâmetros **ipv4.method** e **ipv6.method** para **disabled**. Caso contrário, a auto-configuração do IP falha e o NetworkManager desativa a conexão e remove o dispositivo dummy.

Procedimento

1. Para criar uma interface fictícia chamada *dummy0* com endereços IPv4 e IPv6 estáticos, entre:

```
# nmcli connection add type dummy ifname dummy0 ipv4.method manual
ipv4.addresses 192.0.2.1/24 ipv6.method manual ipv6.addresses 2001:db8:2::1/64
```

2. Opcional: Para visualizar a interface do manequim, entre:

```
# nmcli connection show
NAME            UUID                                  TYPE      DEVICE
enp1s0          db1060e9-c164-476f-b2b5-caec62dc1b05  ethernet  ens3
dummy-dummy0    aaf6eb56-73e5-4746-9037-eed42caa8a65  dummy     dummy0
```

Recursos adicionais

- A página de homem de nm(5)

CAPÍTULO 22. USANDO O NETCONSOLE PARA REGISTRAR MENSAGENS DO KERNEL ATRAVÉS DE UMA REDE

Usando o módulo do kernel **netconsole** e o serviço com o mesmo nome, você pode registrar mensagens do kernel através de uma rede para depurar o kernel quando o registro em disco falha ou quando o uso de um console serial não é possível.

22.1. CONFIGURAÇÃO DO SERVIÇO NETCONSOLE PARA REGISTRAR MENSAGENS DO KERNEL EM UM HOST REMOTO

Usando o módulo do kernel **netconsole**, você pode registrar as mensagens do kernel em um serviço de registro remoto do sistema.

Pré-requisitos

- Um serviço de registro de sistema, como o **rsyslog**, está instalado no host remoto.
- O serviço de registro remoto do sistema é configurado para receber entradas de registro de entrada deste host.

Procedimento

1. Instale o pacote **netconsole-service**:

```
# yum install netconsole-service
```

2. Edite o arquivo **/etc/sysconfig/netconsole** e configure o parâmetro **SYSLOGADDR** para o endereço IP do host remoto:

```
# SYSLOGADDR=192.0.2.1
```

3. Habilite e inicie o serviço **netconsole**:

```
# systemctl enable --now netconsole
```

Etapas de verificação

- Exibir o arquivo **/var/log/messages** no servidor de logs do sistema remoto.

Recursos adicionais

- Para detalhes sobre como permitir que o host remoto receba as mensagens de registro, consulte a seção Configurando uma solução de registro remoto na documentação **Configuring basic system settings**.

CAPÍTULO 23. METAS E SERVIÇOS DE REDE DO SISTEMA

O NetworkManager configura a rede durante o processo de inicialização do sistema. Entretanto, ao inicializar com uma raiz remota (/), como se o diretório raiz fosse armazenado em um dispositivo iSCSI, as configurações de rede são aplicadas no disco RAM inicial (**initrd**) antes que a RHEL seja iniciada. Por exemplo, se a configuração de rede for especificada na linha de comando do kernel usando **rd.neednet=1** ou se uma configuração for especificada para montar sistemas de arquivos remotos, então as configurações de rede são aplicadas em **initrd**.

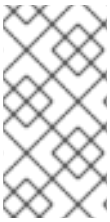
Esta seção descreve diferentes alvos como **network**, **network-online** e **NetworkManager-wait-online** serviço que são usados durante a aplicação de configurações de rede, e como configurar o serviço **systemd** para iniciar após o serviço **network-online** ser iniciado.

23.1. DIFERENÇAS ENTRE A REDE E O ALVO DO SISTEMA EM REDE

Systemd mantém as unidades-alvo **network** e **network-online**. As unidades especiais, como **NetworkManager-wait-online.service**, têm parâmetros **WantedBy=network-online.target** e **Before=network-online.target**. Se ativadas, estas unidades começam com **network-online.target** e atrasam o alvo a ser alcançado até que alguma forma de conectividade de rede seja estabelecida. Elas atrasam o alvo **network-online** até que a rede seja conectada.

A meta **network-online** inicia um serviço, o que acrescenta atrasos substanciais à execução posterior. O Systemd adiciona automaticamente as dependências com parâmetros **Wants** e **After** para esta unidade alvo a todas as unidades de serviço do Sistema V (SysV) **init** com um cabeçalho Linux Standard Base (LSB) referente à instalação **$network**. O cabeçalho da LSB é metadados para scripts **init**. Você pode usá-lo para especificar as dependências. Isto é similar ao objetivo **systemd**.

A meta **network** não retarda significativamente a execução do processo de inicialização. Atingir a meta **network** significa que o serviço responsável pela instalação da rede já começou. Entretanto, isso não significa que um dispositivo de rede foi configurado. Este alvo é importante durante o desligamento do sistema. Por exemplo, se você tiver um serviço que foi pedido após o alvo **network** durante o boot, então esta dependência é revertida durante o desligamento. A rede não é desconectada até que seu serviço tenha sido interrompido. Todas as unidades de montagem para sistemas de arquivos remotos de rede iniciam automaticamente a unidade de destino **network-online** e se encomendam após a mesma.



NOTA

A unidade alvo **network-online** só é útil durante o início do sistema. Após a inicialização completa do sistema, este alvo não rastreia o estado on-line da rede. Portanto, não é possível usar **network-online** para monitorar a conexão da rede. Este alvo fornece um conceito único de inicialização do sistema.

23.2. VISÃO GERAL DO NETWORKMANAGER-WAIT-ONLINE

Os scripts de rede legados síncronos iteram através de todos os arquivos de configuração para configurar os dispositivos. Eles aplicam todas as configurações relacionadas à rede e garantem que a rede esteja on-line.

O serviço **NetworkManager-wait-online** espera com um timeout para que a rede seja configurada. Esta configuração de rede envolve a conexão de um dispositivo Ethernet, a busca de um dispositivo Wi-Fi, e assim por diante. O NetworkManager ativa automaticamente os perfis adequados que são configurados para iniciar automaticamente. A falha do processo de ativação automática devido a um timeout DHCP ou evento similar pode manter o NetworkManager ocupado por um longo período de tempo. Dependendo da configuração, o NetworkManager volta a ativar o mesmo perfil ou um perfil diferente.

Quando a partida é concluída, ou todos os perfis estão em um estado desconectado ou são ativados com sucesso. Você pode configurar os perfis para auto-conexão. A seguir, alguns exemplos de parâmetros que definem os intervalos de tempo ou definem quando a conexão é considerada ativa:

- **connection.wait-device-timeout** - define o tempo limite para que o motorista detecte o dispositivo
- **ipv4.may-fail** e **ipv6.may-fail** - define a ativação com uma família de endereços IP pronta, ou se uma determinada família de endereços deve ter completado a configuração.
- **ipv4.gateway-ping-timeout** - retarda a ativação.

Recursos adicionais

- A página do homem **nm-settings(5)**

23.3. CONFIGURAÇÃO DE UM SERVIÇO DE SISTEMA PARA INICIAR DEPOIS QUE A REDE FOR INICIADA

O Red Hat Enterprise Linux instala os arquivos de serviço **systemd** no diretório **/usr/lib/systemd/system/**. Este procedimento cria um snippet para um arquivo de serviço no diretório **/etc/systemd/system/*service_name*.service.d/** que é usado junto com o arquivo de serviço em **/usr/lib/systemd/system/** para iniciar um determinado *service* depois que a rede estiver on-line. Tem uma prioridade mais alta se as configurações no snippet se sobrepuserem às do arquivo de serviço em **/usr/lib/systemd/system/**.

Procedimento

1. Para abrir o arquivo de serviço no editor, entre:
   # **systemctl edit service_name**
2. Digite o seguinte, e salve as mudanças:

   ```
   [Unit]
   After=network-online.target
   ```

3. Recarregue o serviço **systemd**.
   # **systemctl daemon-reload**

CAPÍTULO 24. CONTROLE DE TRÁFEGO LINUX

O Linux oferece ferramentas para gerenciar e manipular a transmissão de pacotes. O subsistema de Controle de Tráfego Linux (TC) ajuda no policiamento, classificação, modelagem e agendamento do tráfego da rede. O TC também modifica o conteúdo dos pacotes durante a classificação, utilizando filtros e ações. O subsistema TC consegue isso utilizando disciplinas de enfileiramento (**qdisc**), um elemento fundamental da arquitetura do TC.

O mecanismo de programação organiza ou rearranja os pacotes antes que eles entrem ou saiam de diferentes filas. O programador mais comum é o programador First-In-First-Out (FIFO). Você pode fazer as operações do **qdiscs** temporariamente usando o utilitário **tc** ou permanentemente usando o NetworkManager.

Esta seção explica as disciplinas de enfileiramento e descreve como atualizar o padrão **qdiscs** em RHEL.

24.1. VISÃO GERAL DAS DISCIPLINAS DE ENFILEIRAMENTO

As disciplinas de enfileiramento (**qdiscs**) ajudam com o enfileiramento e, posteriormente, o agendamento da transmissão de tráfego por uma interface de rede. Um **qdisc** tem duas operações;

- solicita para que um pacote possa ser enfileirado para posterior transmissão e
- dequeue solicita para que um dos pacotes enfileirados possa ser escolhido para transmissão imediata.

Cada **qdisc** tem um número de identificação hexadecimal de 16 bits chamado **handle**, com um cólon anexo, como **1:** ou **abcd:**. Este número é chamado de **qdisc** número principal. Se um **qdisc** tem classes, então os identificadores são formados como um par de dois números com o número maior antes do menor, **<major>:<minor>**, por exemplo **abcd:1**. O esquema de numeração para os números menores depende do tipo **qdisc**. Às vezes a numeração é sistemática, onde a primeira classe tem o ID **<major>:1**, a segunda **<major>:2**, e assim por diante. Alguns **qdiscs** permitem ao usuário definir arbitrariamente os números menores de classe ao criar a classe.

Classificado qdiscs
Existem diferentes tipos de **qdiscs** e ajudam na transferência de pacotes de e para uma interface de rede. Você pode configurar **qdiscs** com classes raiz, pai, ou criança. O ponto onde as crianças podem ser anexadas é chamado classes. As classes em **qdisc** são flexíveis e podem sempre conter várias classes infantis ou uma única criança, **qdisc**. Não há proibição contra uma classe que contenha uma classe em si **qdisc**, o que facilita cenários complexos de controle de tráfego. A classe **qdiscs** não armazena nenhum pacote em si. Em vez disso, eles fazem consultas e fazem pedidos de filiação a um de seus filhos de acordo com critérios específicos do site **qdisc**. Eventualmente, esta passagem recursiva de pacotes termina onde os pacotes são armazenados (ou pegos no caso de desqueue).

Sem classe qdiscs
Alguns **qdiscs** não contêm aulas para crianças e são chamados de classless **qdiscs**. Sem classes **qdiscs** requerem menos personalização em comparação com a classe **qdiscs**. Geralmente é suficiente anexá-los a uma interface.

Recursos adicionais

- Para informações detalhadas sobre os sem classe e os sem classe **qdiscs**, consulte a página de manual **tc(8)**.
- Para informações detalhadas sobre as ações, consulte as páginas de manual **actions** e **tc-actions.8**.

24.2. QDISCS DISPONÍVEIS NA RHEL

Cada **qdisc** trata de questões exclusivas relacionadas à rede. A seguir está a lista de **qdiscs** disponível em RHEL. Você pode usar qualquer uma das seguintes **qdisc** para moldar o tráfego de rede com base em suas necessidades de rede.

Tabela 24.1. Agendadores disponíveis na RHEL

qdisc nome	Incluído em	Suporte de descarregamento
Modo de Transferência Assíncrona (ATM)	**kernel-modules-extra**	
Enfileiramento baseado em classe	**kernel-modules-extra**	
Modelador baseado em crédito	**kernel-modules-extra**	Sim
Escolher e Manter para fluxos responsivos, Escolher e Matar para fluxos não responsivos (CHOKE)	**kernel-modules-extra**	
Atraso Controlado (CoDel)	**kernel-core**	
Déficit Round Robin (DRR)	**kernel-modules-extra**	
Marcador de Serviços Diferenciados (DSMARK)	**kernel-modules-extra**	
Seleção de Transmissão Aprimorada (ETS)	**kernel-modules-extra**	Sim
Fila da Feira (FQ)	**kernel-core**	
Atraso controlado por filas justas (FQ_CODel)	**kernel-core**	
Detecção precoce aleatória generalizada (GRED)	**kernel-modules-extra**	
Curva Hierárquica de Serviço Justo (HSFC)	**kernel-core**	
Filtro Heavy-Hitter (HHF)	**kernel-core**	
Balde de Hierarquia (HTB)	**kernel-core**	
INGRESS	**kernel-core**	Sim

qdisc nome	Incluído em	Suporte de descarregamento
Prioridade de múltiplas filas (MQPRIO)	**kernel-modules-extra**	Sim
Multiqueue (MULTIQ)	**kernel-modules-extra**	Sim
Emulador de Rede (NETEM)	**kernel-modules-extra**	
Proporcional Integral-controller Enhanced (PIE)	**kernel-core**	
PLUG	**kernel-core**	
Enfileiramento Rápido de Feira (QFQ)	**kernel-modules-extra**	
Detecção precoce aleatória (VERMELHO)	**kernel-modules-extra**	Sim
Stochastic Fair Blue (SFB)	**kernel-modules-extra**	
Enfileiramento Estocástico Justo (SFQ)	**kernel-core**	
Filtro Token Bucket (TBF)	**kernel-core**	Sim
Equalizador de Elo Trivial (TEQL)	**kernel-modules-extra**	

IMPORTANTE

A descarga do **qdisc** requer suporte de hardware e drivers no NIC.

Recursos adicionais

- Para informações completas dos parâmetros e filtros usados para configurar o **qdiscs**, consulte as páginas man **tc(8)**, **cbq**, **cbs**, **choke**, **CoDel**, **drr**, **fq**, **htb**, **mqprio**, **netem**, **pie**, **sfb**, **pfifo**, , **tc-red**, **sfq**, **tbf**, e **prio**.

24.3. INSPEÇÃO DE QDISCS DE UMA INTERFACE DE REDE USANDO O UTILITÁRIO TC

Por default, os sistemas Red Hat Enterprise Linux usam **fq_codel qdisc** . Este procedimento descreve como inspecionar os balcões **qdisc**.

Procedimento

1. Opcional: Veja seu atual **qdisc**:
   # **tc qdisc show dev *enp0s1***

2. Inspecione os atuais balcões **qdisc**:

```
# tc -s qdisc show dev enp0s1
qdisc fq_codel 0: root refcnt 2 limit 10240p flows 1024 quantum 1514 target 5.0ms interval
100.0ms memory_limit 32Mb ecn
Sent 1008193 bytes 5559 pkt (dropped 233, overlimits 55 requeues 77)
backlog 0b 0p requeues 0
....
```

- **dropped** - o número de vezes que um pacote é descartado porque todas as filas estão cheias
- **overlimits** - o número de vezes que a capacidade do link configurado é preenchida
- **sent** - o número de dequeues

24.4. ATUALIZAÇÃO DO QDISC PADRÃO

Se você observar perdas de pacotes de rede com o atual **qdisc**, você pode alterar o **qdisc** com base em seus requisitos de rede. Você pode selecionar o **qdisc**, que atende às exigências de sua rede. Este procedimento descreve como alterar o padrão **qdisc** no Red Hat Enterprise Linux.

Procedimento

1. Veja o padrão atual **qdisc**:

```
# sysctl -a | grep qdisc
net.core.default_qdisc = fq_codel
```

2. Veja o **qdisc** da atual conexão Ethernet:

```
# tc -s qdisc show dev enp0s1
qdisc fq_codel 0: root refcnt 2 limit 10240p flows 1024 quantum 1514 target 5.0ms interval
100.0ms memory_limit 32Mb ecn
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
backlog 0b 0p requeues 0
maxpacket 0 drop_overlimit 0 new_flow_count 0 ecn_mark 0
new_flows_len 0 old_flows_len 0
```

3. Atualize o existente **qdisc**:
   # **sysctl -w net.core.default_qdisc=pfifo_fast**

4. Para aplicar as mudanças, recarregue o driver da rede:
   # **rmmod *NETWORKDRIVERNAME***

   # **modprobe *NETWORKDRIVERNAME***

5. Iniciar a interface da rede:
   # **ip link set *enp0s1* up**

Etapas de verificação

- Veja o **qdisc** da conexão Ethernet:

```
# tc -s qdisc show dev enp0s1
qdisc pfifo_fast 0: root refcnt 2 bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
 Sent 373186 bytes 5333 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0b 0p requeues 0
....
```

Recursos adicionais

- Para mais informações sobre como tornar estas mudanças persistentes, veja Como definir variáveis em **sysctl** no artigo do Red Hat Enterprise Linux .

24.5. CONFIGURAÇÃO TEMPORÁRIA DO QDISK ATUAL DE UMA INTERFACE DE REDE USANDO O UTILITÁRIO TC

Você pode atualizar o atual **qdisc** sem alterar o padrão. Este procedimento descreve como alterar o atual **qdisc** no Red Hat Enterprise Linux.

Procedimento

1. Opcional: Veja o atual **qdisc**:
   # tc -s qdisc show dev *enp0s1*

2. Atualize o atual **qdisc**:
   # tc qdisc replace dev *enp0s1* root *htb*

Etapa de verificação

- Veja a atualização atual **qdisc**:

```
# tc -s qdisc show dev enp0s1
qdisc htb 8001: root refcnt 2 r2q 10 default 0 direct_packets_stat 0 direct_qlen 1000
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
backlog 0b 0p requeues 0
```

24.6. CONFIGURAÇÃO PERMANENTE DO QDISK ATUAL DE UMA INTERFACE DE REDE USANDO O NETWORKMANAGER

Você pode atualizar o valor atual **qdisc** de uma conexão NetworkManager.

Procedimento

1. Opcional: Veja o atual **qdisc**:

```
# tc qdisc show dev enp0s1
  qdisc fq_codel 0: root refcnt 2
```

2. Atualize o atual **qdisc**:
   # nmcli connection modify *enp0s1* tc.qdiscs 'root pfifo_fast'

3. Opcional: Para adicionar outro **qdisc** sobre o existente **qdisc**, use a opção **tc.qdisc**:

 # nmcli connection modify *enp0s1* tc.qdisc 'ingress handle ffff:'

4. Ativar as mudanças:
 # nmcli connection up *enp0s1*

Etapas de verificação

- Veja atualmente **qdisc** a interface da rede:

```
# tc qdisc show dev _enp0s1_
qdisc _pfifo_fast_ 8001: root refcnt 2 bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
qdisc ingress ffff: parent ffff:fff1 ----------------
```

Recursos adicionais

- Para mais informações, consulte a página de manual **nm-settings(5)**.

CAPÍTULO 25. COMEÇANDO COM O MULTIPATH TCP

IMPORTANTE

O TCP Multipath é fornecido apenas como uma Visualização de Tecnologia. Os recursos de Technology Preview não são suportados com os Acordos de Nível de Serviço (SLAs) de produção da Red Hat, podem não ser completos funcionalmente e a Red Hat não recomenda o seu uso para produção. Estas prévias fornecem acesso antecipado às próximas características do produto, permitindo aos clientes testar a funcionalidade e fornecer feedback durante o processo de desenvolvimento.

Veja Technology Preview Features Support Scope no Portal do Cliente da Red Hat para obter informações sobre o escopo de suporte para os recursos de Technology Preview.

O TCP Multipath (MPTCP) é uma extensão do Protocolo de Controle de Transmissão (TCP). Usando o Protocolo de Internet (IP), um host pode enviar pacotes para um destino. O TCP garante a entrega confiável dos dados através da Internet e ajusta automaticamente sua largura de banda em resposta à carga da rede.

As vantagens do MPTCP são as seguintes:

- Ele permite TCP para uso em dispositivos equipados com duas ou mais interfaces de rede.
- Ela permite que os usuários utilizem simultaneamente diferentes interfaces de rede ou mudem sem problemas de uma conexão para outra.
- Ela melhora o uso de recursos dentro da rede e a resiliência à falha da rede.

Esta seção descreve como fazê-lo:

- criar uma nova conexão MPTCP,
- usar **iproute2** para adicionar novos subfluxos e endereços IP às conexões MPTCP, e
- desabilitar o MPTCP no kernel para evitar aplicações que utilizam conexões MPTCP.

25.1. PREPARANDO A RHEL PARA PERMITIR O APOIO AO MPTCP

Poucas aplicações suportam nativamente o MPTCP. Na maioria das vezes, a conexão e os soquetes orientados a fluxo solicitam o protocolo TCP no soquete() chamada para o sistema operacional. Você pode habilitar o suporte a MPTCP na RHEL usando a ferramenta **sysctl** para programas suportados nativamente por MPTCP. A implementação do MPTCP também foi projetada para permitir o uso do protocolo MPTCP para aplicações que solicitem **IPPROTO_TCP** call to the kernel.

Este procedimento descreve como habilitar o suporte a MPTCP e preparar a RHEL para habilitar todo o sistema MPTCP usando um script SystemTap.

Pré-requisitos

Os seguintes pacotes estão instalados:

- **kernel-debuginfo**
- **kernel-debuginfo-common**
- **systemtap**

- **systemtap-devel**
- **kernel-devel**
- **nmap-ncat**

Procedimento

1. Habilitar soquetes MPTCP no núcleo:

```
# echo "net.mptcp.enabled=1" > /etc/sysctl.d/90-enable-MPTCP.conf
# sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf
```

2. Crie um arquivo **mptcp.stap** com o seguinte conteúdo:

```
#! /usr/bin/env stap

%{
#include <linux/in.h>
#include <linux/ip.h>
%}

/* according to [1], RSI contains 'type' and RDX
 * contains 'protocol'.
 * [1] https://github.com/torvalds/linux/blob/master/arch/x86/entry/entry_64.S#L79
 */

function mptcpify () %{
    if (CONTEXT->kregs->si == SOCK_STREAM &&
        (CONTEXT->kregs->dx == IPPROTO_TCP ||
         CONTEXT->kregs->dx == 0)) {
            CONTEXT->kregs->dx = IPPROTO_MPTCP;
            STAP_RETVALUE = 1;
    } else {
          STAP_RETVALUE = 0;
    }
%}

probe kernel.function("__sys_socket") {
        if (mptcpify() == 1) {
                printf("command %16s mptcpified\n", execname());
        }
}
```

3. Substitua o soquete TCP por MPTCP:
   # **stap -vg mptcp.stap**

   Nota: Use **Ctrl**+**C** para converter a conexão de volta para TCP a partir de MPTCP.

4. Inicie um servidor que escute a porta TCP 4321:
   # **ncat -4 -l *4321***

5. Conecte-se ao servidor e troque tráfego. Por exemplo, o cliente aqui escreve "Olá mundo" para o servidor 5 vezes, depois ele encerra a conexão.

```
# ncat -4 192.0.2.1 4321
Hello world 1
Hello world 2
Hello world 3
Hello world 4
Hello world 5
```

Imprensa **Ctrl**+**D** para desistir.

Etapas de verificação

1. Verificar se o MPTCP está habilitado no núcleo:

```
# sysctl -a | grep mptcp.enabled
net.mptcp.enabled = 1
```

2. Após o script **mptcp.stap** instalar a sonda do kernel, os seguintes avisos aparecem na saída do kernel **dmesg**

```
# dmesg
...
[ 1752.694072] Kprobes globally unoptimized
[ 1752.730147] stap_1ade3b3356f3e68765322e26dec00c3d_1476: module_layout: kernel
tainted.
[ 1752.732162] Disabling lock debugging due to kernel taint
[ 1752.733468] stap_1ade3b3356f3e68765322e26dec00c3d_1476: loading out-of-tree
module taints kernel.
[ 1752.737219] stap_1ade3b3356f3e68765322e26dec00c3d_1476: module verification
failed: signature and/or required key missing - tainting kernel
```

3. Depois que a conexão for estabelecida, verifique a saída **ss** para ver o status específico do subfluxo:

```
# ss -nti '( dport :4321 )' dst 192.0.2.1
State Recv-Q Send-Q Local Address:Port    Peer Address:Port Process

ESTAB 0      0          192.0.2.2:60874      192.0.2.1:4321
cubic wscale:7,7 rto:201 rtt:0.042/0.017 mss:1448 pmtu:1500 rcvmss:536 advmss:1448
cwnd:10 bytes_sent:64 bytes_$cked:65 segs_out:6 segs_in:5 data_segs_out:4 send
2758095238bps lastsnd:57 lastrcv:3054 lastack:57 pacing_rate 540361516$bps
delivery_rate 413714280bps delivered:5 rcv_space:29200 rcv_ssthresh:29200 minrtt:0.009
tcp-ulp-mptcp flags:Mmec token:0000(id:0)/4bffe73d(id:0) seq:c11f40d6c5337463 sfseq:1
ssnoff:f7455705 maplen:0
```

4. Capture o tráfego usando **tcpdump** e verifique o uso de sub-opções MPTCP:

```
# tcpdump -tnni interface tcp port 4321
client Out IP 192.0.2.2.60802 > 192.0.2.1.4321: Flags [S], seq 3420255622, win 29200,
options [mss 1460,sackOK,TS val 411 4539945 ecr 0,nop,wscale 7,mptcp capable v1],
length 0
client In  IP 192.0.2.1.4321 > 192.0.2.2.60802: Flags [S.], seq 2619315374, ack 3420255623,
win 28960, options [mss 1460 sackOK,TS val 3241564233 ecr 4114539945,nop,wscale
7,mptcp capable v1 {0xb6f8dc721aee7f64}], length 0
client Out IP 192.0.2.2.60802 > 192.0.2.1.4321: Flags [.], ack 1, win 229, options [nop,nop,TS
```

```
val 4114539945 ecr 3241564 233,mptcp capable v1
{0xcc58d5d632a32d13,0xb6f8dc721aee7f64}], length 0
client Out IP 192.0.2.2.60802 > 192.0.2.1.4321: Flags [P.], seq 1:17, ack 1, win 229, options
[nop,nop,TS val 4114539945 ecr 3241564233,mptcp capable v1
{0xcc58d5d632a32d13,0xb6f8dc721aee7f64},nop,nop], length 16
client In  IP 192.0.2.1.4321 > 192.0.2.2.60802: Flags [.], ack 17, win 227, options [nop,nop,TS
val 3241564233 ecr 411459945,mptcp dss ack 1105509586894558345], length 0
client Out IP 192.0.2.2.60802 > 192.0.2.1.4321: Flags [P.], seq 17:33, ack 1, win 229, options
[nop,nop,TS val 4114540939 ecr 3241564233,mptcp dss ack 13265586846326199424 seq
105509586894558345 subseq 17 len 16,nop,nop], length 16
```

O pacote **tcpdump** é necessário para executar este comando.

Recursos adicionais

- Para mais informações veja, Como posso baixar ou instalar pacotes de debuginfo para sistemas RHEL? artigo.
- Para mais informações em **IPPROTO_TCP**, consulte as páginas de manual **tcp(7)**.

25.2. USANDO O IPROUTE2 PARA NOTIFICAR APLICAÇÕES SOBRE MÚLTIPLOS CAMINHOS DISPONÍVEIS

Por padrão, o soquete MPTCP começa com um único subfluxo, mas você pode adicionar novos subfluxos e endereços IP à conexão uma vez que você a crie pela primeira vez. Este procedimento descreve como atualizar os limites por conexão para subfluxos e endereços IP, e adicionar novos endereços IP (endpoints) à conexão MPTCP.

Note que o MPTCP ainda não suporta terminais mistos IPv6 e IPv4 para o mesmo soquete. Use endpoints que pertençam à mesma família de endereços.

Procedimento

1. Defina os limites por conexão e endereço IP para *1* no servidor:
   # **ip mptcp limits set subflow *1***
2. Estabelecer os limites por conexão e endereço IP para *1* no cliente:
   # **ip mptcp limits set subflow *1* add_addr_accepted *1***
3. Adicione o endereço IP *198.51.100.1* como um novo endpoint MPTCP no servidor:
   # **ip mptcp endpoint add *198.51.100.1* dev *enp1s0 signal***

IMPORTANTE

Você pode definir os seguintes valores para as bandeiras: **subflow**, **backup**, **signal**. Colocando a bandeira em;

- **signal**, envia um pacote **ADD_ADDR** depois que o aperto de mão triplo for concluído
- **subflow**, envia um **MP_JOIN SYN** pelo cliente
- **backup**, define o ponto final como um endereço de backup

4. Iniciar a ligação do servidor a 0.0.0.0 com o argumento **-k** para evitar que o [systemitem]'ncat' feche a tomada de escuta após aceitar a primeira conexão e fazer o servidor rejeitar **MP_JOIN SYN** feito pelo cliente.
   # **ncat -4 0.0.0.0 -k -l *4321***

5. Inicie o cliente e conecte-se ao servidor para trocar tráfego. Por exemplo, o cliente aqui escreve "Olá mundo" para o servidor 5 vezes, depois ele encerra a conexão.

   ```
   # ncat -4 192.0.2.1 4321
   Hello world 1
   Hello world 2
   Hello world 3
   Hello world 4
   Hello world 5
   ```

   Imprensa **Ctrl**+**D** para desistir.

Etapas de verificação

1. Verificar a conexão e o limite de endereço IP:
   # **ip mptcp limit show**

2. Verificar o ponto final recém-adicionado:
   # **ip mptcp endpoint show**

3. Capture o tráfego usando **tcpdump** e verifique o uso de sub-opções MPTCP:

   ```
   # tcpdump -tnni interface tcp port 4321
   client Out IP 192.0.2.2.56868 > 192.0.2.1.4321: Flags [S], seq 3107783947, win 29200,
   options [mss 1460,sackOK,TS val 2568752336 ecr 0,nop,wscale 7,mptcp capable v1], length
   0
   client In  IP 192.0.2.1.4321 > 192.0.2.2.56868: Flags [S.], seq 4222339923, ack 3107783948,
   win 28960, options [mss 1460,sackOK,TS val 1713130246 ecr 2568752336,nop,wscale
   7,mptcp capable v1 {0xf51c07a47cc2ba75}], length 0
   client Out IP 192.0.2.2.56868 > 192.0.2.1.4321: Flags [.], ack 1, win 229, options [nop,nop,TS
   val 2568752336 ecr 1713130246,mptcp capable v1
   {0xb243376cc5af60bd,0xf51c07a47cc2ba75}], length 0
   client Out IP 192.0.2.2.56868 > 192.0.2.1.4321: Flags [P.], seq 1:17, ack 1, win 229, options
   [nop,nop,TS val 2568752336 ecr 1713130246,mptcp capable v1
   {0xb243376cc5af60bd,0xf51c07a47cc2ba75},nop,nop], length 16
   client In  IP 192.0.2.1.4321 > 192.0.2.2.56868: Flags [.], ack 17, win 227, options [nop,nop,TS
   val 1713130246 ecr 2568752336,mptcp add-addr id 1 198.51.100.1 hmac
   0xe445335073818837,mptcp dss ack 5562689076006296132], length 0
   client Out IP 198.51.100.2.42403 > 198.51.100.1.4321: Flags [S], seq 3356992178, win
   29200, options [mss 1460,sackOK,TS val 4038525523 ecr 0,nop,wscale 7,mptcp join backup
   id 0 token 0xad58df1 nonce 0x74a8137f], length 0
   client In  IP 198.51.100.1.4321 > 198.51.100.2.42403: Flags [S.], seq 1680863152, ack
   3356992179, win 28960, options [mss 1460,sackOK,TS val 4213669942 ecr
   4038525523,nop,wscale 7,mptcp join backup id 0 hmac 0x9eff7a1bf4e65937 nonce
   0x77303fd8], length 0
   client Out IP 198.51.100.2.42403 > 198.51.100.1.4321: Flags [.], ack 1, win 229, options
   [nop,nop,TS val 4038525523 ecr 4213669942,mptcp join hmac
   0xdfdc0129424f627ea774c094461328ce49d195bc], length 0
   client In  IP 198.51.100.1.4321 > 198.51.100.2.42403: Flags [.], ack 1, win 227, options
   [nop,nop,TS val 4213669942 ecr 4038525523,mptcp dss ack 5562689076006296132],
   length 0
   ```

-

O pacote **tcpdump** é necessário para executar este comando.

Recursos adicionais

- Para mais informações sobre as bandeiras de endpoint disponíveis, consulte a página de manual **ip-mptcp(8)**.

25.3. DESABILITANDO O TCP MULTIPATH NO KERNEL

Este procedimento descreve como desativar a opção MPTCP no kernel.

Procedimento

- Desativar a opção **mptcp.enabled**.

```
# echo "net.mptcp.enabled=0" > /etc/sysctl.d/90-enable-MPTCP.conf
# sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf
```

Etapas de verificação

- Verificar se o **mptcp.enabled** está desativado no núcleo.

```
# sysctl -a | grep mptcp.enabled
net.mptcp.enabled = 0
```

CAPÍTULO 26. CONFIGURANDO A ORDEM DOS SERVIDORES DNS

A maioria das aplicações usa a função **getaddrinfo()** da biblioteca **glibc** para resolver solicitações DNS. Por padrão, **glibc** envia todas as solicitações DNS para o primeiro servidor DNS especificado no arquivo **/etc/resolv.conf**. Se este servidor não responder, o Red Hat Enterprise Linux usa o próximo servidor deste arquivo.

Esta seção descreve como personalizar a ordem dos servidores DNS.

26.1. COMO O NETWORKMANAGER ORDENA SERVIDORES DNS EM /ETC/RESOLV.CONF

O NetworkManager ordena servidores DNS no arquivo **/etc/resolv.conf** com base nas seguintes regras:

- Se existir apenas um perfil de conexão, o NetworkManager utiliza a ordem do servidor DNS IPv4 e IPv6 especificada nessa conexão.
- Se vários perfis de conexão forem ativados, o NetworkManager ordena servidores DNS com base em um valor de prioridade DNS. Se você definir prioridades DNS, o comportamento do NetworkManager dependerá do valor definido no parâmetro **dns**. Você pode definir este parâmetro na seção **[main]** no arquivo **/etc/NetworkManager/NetworkManager.conf**:
  - **dns=default** ou se o parâmetro **dns** não estiver definido:
    O NetworkManager ordena os servidores DNS a partir de diferentes conexões com base no parâmetro **ipv4.dns-priority** e **ipv6.dns-priority** em cada conexão.

    Se você não definir nenhum valor ou se você definir **ipv4.dns-priority** e **ipv6.dns-priority** para **0**, o NetworkManager utiliza o valor padrão global. Ver "Valores padrão dos parâmetros de prioridade DNS".
  - **dns=dnsmasq** ou **dns=systemd-resolved**:
    Quando você usa uma dessas configurações, o NetworkManager configura **127.0.0.1** para **dnsmasq** ou **127.0.0.53** como entrada **nameserver** no arquivo **/etc/resolv.conf**.

    Os serviços **dnsmasq** e **systemd-resolved** encaminham as consultas para o domínio de busca definido em uma conexão NetworkManager ao servidor DNS especificado nessa conexão, e encaminha as consultas para outros domínios para a conexão com a rota padrão. Quando múltiplas conexões têm o mesmo conjunto de domínios de busca, **dnsmasq** e **systemd-resolved** encaminham as consultas para este domínio ao servidor DNS definido na conexão com o menor valor de prioridade.

Valores padrão dos parâmetros de prioridade DNS

O NetworkManager utiliza os seguintes valores padrão para conexões:

- **50** para conexões VPN
- **100** para outras conexões

Valores de prioridade DNS válidos:

Você pode definir tanto o padrão global quanto parâmetros específicos de conexão **ipv4.dns-priority** e **ipv6.dns-priority** para um valor entre **-2147483647** e **2147483647**.

- Um valor menor tem uma prioridade maior.

- Os valores negativos têm o efeito especial de excluir outras configurações com um valor maior. Por exemplo, se existe pelo menos uma conexão com um valor de prioridade negativa, o NetworkManager utiliza apenas os servidores DNS especificados no perfil de conexão com a prioridade mais baixa.

- Se várias conexões tiverem a mesma prioridade DNS, o NetworkManager prioriza o DNS na seguinte ordem:

  a. Conexões VPN

  b. Conexão com uma rota padrão ativa. A rota padrão ativa é a rota padrão a mais baixa métrica.

Recursos adicionais

- Para mais detalhes sobre como o NetworkManager ordena as entradas do servidor DNS no arquivo **/etc/resolv.conf**, consulte a descrição dos parâmetros **dns-priority** nas seções **ipv4** e **ipv6** na página de manual **nm-settings(5)**.

- Para detalhes sobre o uso de **systemd-resolved** para usar diferentes servidores DNS para diferentes domínios, veja Capítulo 34, *Usando diferentes servidores DNS para diferentes domínios*.

26.2. DEFINIÇÃO DE UM VALOR DE PRIORIDADE DE SERVIDOR DNS PADRÃO DO NETWORKMANAGER

O NetworkManager utiliza os seguintes valores padrão de prioridade DNS para conexões:

- **50** para conexões VPN

- **100** para outras conexões

Esta seção descreve como substituir esses padrões de todo o sistema com um valor padrão personalizado para conexões IPv4 e IPv6.

Procedimento

1. Edite o arquivo **/etc/NetworkManager/NetworkManager.conf**:

   a. Adicione a seção **[connection]**, se ela não existir:

   ```
   [conexão]
   ```

   b. Adicione os valores padrão personalizados à seção **[connection]**. Por exemplo, para definir o novo padrão tanto para IPv4 quanto para IPv6 para **200**, adicionar:

   ```
   ipv4.dns-priority=200
   ipv6.dns-priority=200
   ```

   Você pode definir os parâmetros para um valor entre **-2147483647** e **2147483647**. Note que a definição dos parâmetros para **0** permite os padrões embutidos (**50** para conexões VPN e **100** para outras conexões).

2. Recarregue o serviço **NetworkManager**:

```
# systemctl reload NetworkManager
```

Recursos adicionais

- Para detalhes adicionais sobre a definição de valores padrão para todas as conexões NetworkManager, consulte **Connection Section** na página de manual **NetworkManager.conf(5)**.

26.3. DEFININDO A PRIORIDADE DNS DE UMA CONEXÃO NETWORKMANAGER

Esta seção descreve como definir a ordem dos servidores DNS quando o NetworkManager cria ou atualiza o arquivo **/etc/resolv.conf**.

Observe que a definição de prioridades DNS só faz sentido se você tiver múltiplas conexões com diferentes servidores DNS configurados. Se você tiver apenas uma conexão com múltiplos servidores DNS configurados, configure manualmente os servidores DNS na ordem preferida no perfil de conexão.

Pré-requisitos

- O sistema tem múltiplas conexões NetworkManager configuradas.
- O sistema ou não tem nenhum parâmetro **dns** definido no arquivo **/etc/NetworkManager/NetworkManager.conf** ou o parâmetro está definido para **default**.

Procedimento

1. Opcionalmente, exibir as conexões disponíveis:

```
# nmcli connection show
NAME          UUID                                  TYPE      DEVICE
Example_con_1  d17ee488-4665-4de2-b28a-48befab0cd43  ethernet  enp1s0
Example_con_2  916e4f67-7145-3ffa-9f7b-e7cada8f6bf7  ethernet  enp7s0
...
```

2. Estabelecer os parâmetros **ipv4.dns-priority** e **ipv6.dns-priority**. Por exemplo, para definir os dois parâmetros para **10** para a conexão **Example_con_1**:

```
# nmcli connection modify Example_con_1 ipv4.dns-priority 10 ipv6.dns-priority 10
```

3. Opcionalmente, repetir a etapa anterior para outras conexões.
4. Reative a conexão que você atualizou:

```
# nmcli connection up Example_con_1
```

Etapas de verificação

- Exibir o conteúdo do arquivo **/etc/resolv.conf** para verificar se a ordem do servidor DNS está correta:

```
# cat /etc/resolv.conf
```

CAPÍTULO 27. CONFIGURAÇÃO DE REDE IP COM ARQUIVOS IFCFG

Esta seção descreve como configurar uma interface de rede manualmente através da edição dos arquivos **ifcfg**.

Os arquivos de configuração de interface (ifcfg) controlam as interfaces de software para dispositivos de rede individuais. Como o sistema inicia, ele usa esses arquivos para determinar quais interfaces devem ser criadas e como configurá-las. Estes arquivos são normalmente nomeados **ifcfg-*name***, onde o sufixo *name* se refere ao nome do dispositivo que o arquivo de configuração controla. Por convenção, o sufixo **ifcfg** é o mesmo que a seqüência dada pela diretiva **DEVICE** no próprio arquivo de configuração.

27.1. CONFIGURAÇÃO DE UMA INTERFACE COM CONFIGURAÇÕES DE REDE ESTÁTICA USANDO ARQUIVOS IFCFG

Este procedimento descreve como configurar uma interface de rede usando os arquivos **ifcfg**.

Procedimento

- Para configurar uma interface com configurações de rede estática usando arquivos **ifcfg**, para uma interface com o nome **enp1s0**, crie um arquivo com o nome **ifcfg-enp1s0** no diretório **/etc/sysconfig/network-scripts/** que contém:
  - Para configuração em **IPv4**:

```
DEVICE=enp1s0
BOOTPROTO=none
ONBOOT=yes
PREFIX=24
IPADDR=10.0.1.27
GATEWAY=10.0.1.1
```

  - Para configuração em **IPv6**:

```
DEVICE=enp1s0
BOOTPROTO=none
ONBOOT=yes
IPV6INIT=yes
IPV6ADDR=2001:db8:1::2/64
```

Recursos adicionais

- Para mais informações sobre as conexões de teste, ver Capítulo 39, *Teste de configurações básicas de rede*.
- Para mais **IPv6** ifcfg opções de configuração, veja ***nm-settings-ifcfg-rh***(5) página do homem.

27.2. CONFIGURAÇÃO DE UMA INTERFACE COM CONFIGURAÇÕES DINÂMICAS DE REDE USANDO ARQUIVOS IFCFG

Este procedimento descreve como configurar uma interface de rede com configurações dinâmicas de rede usando arquivos **ifcfg**.

Procedimento

1. Para configurar uma interface chamada *em1* com configurações dinâmicas de rede usando arquivos **ifcfg**, crie um arquivo com o nome **ifcfg-em1** no diretório **/etc/sysconfig/network-scripts/** que contém:

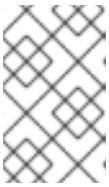
```
DEVICE=em1
BOOTPROTO=dhcp
ONBOOT=yes
```

2. Para configurar uma interface para enviar um nome de host diferente para o servidor **DHCP**, adicione a seguinte linha ao arquivo **ifcfg**:

```
DHCP_HOSTNAME=hostname
```

3. Para configurar uma interface para enviar um outro nome de domínio totalmente qualificado (FQDN) para o servidor **DHCP**, adicione a seguinte linha ao arquivo **ifcfg**:

```
DHCP_FQDN=fully.qualified.domain.name
```



NOTA

Apenas uma diretriz, **DHCP_HOSTNAME** ou **DHCP_FQDN**, deve ser usada em um determinado arquivo **ifcfg**. No caso de serem especificados **DHCP_HOSTNAME** e **DHCP_FQDN**, somente esta última é utilizada.

4. Para configurar uma interface para utilizar determinados servidores **DNS**, adicione as seguintes linhas ao arquivo **ifcfg**:

```
PEERDNS=no
DNS1=ip-address
DNS2=ip-address
```

onde *ip-address* é o endereço de um servidor **DNS**. Isto fará com que o serviço de rede atualize **/etc/resolv.conf** com os servidores **DNS** especificados. Apenas um **DNS** endereço de servidor é necessário, o outro é opcional.

27.3. GERENCIAMENTO DE TODO O SISTEMA E PERFIS DE CONEXÃO PRIVADA COM ARQUIVOS IFCFG

Este procedimento descreve como configurar os arquivos **ifcfg** para gerenciar os perfis de conexão privada e de todo o sistema.

Procedimento

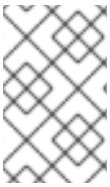
As permissões correspondem à diretriz **USERS** nos arquivos **ifcfg**. Se a diretiva **USERS** não estiver presente, o perfil da rede estará disponível para todos os usuários.

1. Como exemplo, modifique o arquivo **ifcfg** com a seguinte linha, que tornará a conexão disponível apenas para os usuários listados:

```
USUÁRIOS="joe bob alice
```

CAPÍTULO 28. USANDO O NETWORKMANAGER PARA DESATIVAR O IPV6 PARA UMA CONEXÃO ESPECÍFICA

Esta seção descreve como desativar o protocolo IPv6 em um sistema que usa o NetworkManager para gerenciar as interfaces de rede. Se você desabilitar o IPv6, o NetworkManager define automaticamente os valores correspondentes de **sysctl** no Kernel.



NOTA

O serviço NetworkManager estabelece certos valores **sysctl** quando inicia uma conexão. Para evitar comportamentos inesperados, não defina manualmente os valores **sysctl** para desativar o IPv6.

Pré-requisitos

- O sistema usa o NetworkManager para gerenciar as interfaces de rede, que é o padrão no Red Hat Enterprise Linux 8.
- O sistema roda o Red Hat Enterprise Linux 8.1 ou posterior.

28.1. DESABILITANDO IPV6 EM UMA CONEXÃO USANDO NMCLI

Use esta seção para desativar o protocolo IPv6 usando o utilitário **nmcli**.

Procedimento

1. Opcionalmente, exibir a lista de conexões de rede:

```
# nmcli connection show
NAME    UUID                                  TYPE      DEVICE
Example 7a7e0151-9c18-4e6f-89ee-65bb2d64d365  ethernet  enp1s0
...
```

2. Definir o parâmetro **ipv6.method** da conexão para **disabled**:

```
# nmcli connection modify Example ipv6.method {\i1}"disabled
```

3. Reiniciar a conexão de rede:

```
# nmcli conexão acima Example
```

Etapas de verificação

1. Digite o comando **ip address show** para exibir as configurações de IP do dispositivo:

```
# ip address show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:6b:74:be brd ff:ff:ff:ff:ff:ff
    inet 192.0.2.1/24 brd 192.10.2.255 scope global noprefixroute enp1s0
       valid_lft forever preferred_lft forever
```

Se não for exibida a entrada **inet6**, o IPv6 é desativado no dispositivo.

2. Verifique se o /**proc**/**sys**/**net**/**ipv6**/**conf**/***enp1s0***/**disable_ipv6** contém agora o valor **1**:

```
# cat /proc/sys/net/ipv6/conf/enp1s0/disable_ipv6
1
```

O valor **1** significa que o IPv6 está desabilitado para o dispositivo.

CAPÍTULO 29. CONFIGURAÇÃO MANUAL DO ARQUIVO /ETC/RESOLV.CONF

Por default, o NetworkManager no Red Hat Enterprise Linux (RHEL) 8 atualiza dinamicamente o arquivo **/etc/resolv.conf** com as configurações DNS dos perfis de conexão ativos do NetworkManager. Esta seção descreve diferentes opções sobre como desativar este recurso para configurar manualmente as configurações do DNS em **/etc/resolv.conf**.

29.1. DESATIVAÇÃO DO PROCESSAMENTO DNS NA CONFIGURAÇÃO DO NETWORKMANAGER

Esta seção descreve como desativar o processamento DNS na configuração do NetworkManager para configurar manualmente o arquivo **/etc/resolv.conf**.

Procedimento

1. Como usuário root, crie o arquivo **/etc/NetworkManager/conf.d/90-dns-none.conf** com o seguinte conteúdo, utilizando um editor de texto:

```
[main]
dns=none
```

2. Recarregue o serviço **NetworkManager**:

```
# systemctl reload NetworkManager
```

NOTA

Depois de recarregar o serviço, o NetworkManager não atualiza mais o arquivo **/etc/resolv.conf**. Entretanto, o último conteúdo do arquivo é preservado.

3. Opcionalmente, remova o comentário **Generated by NetworkManager** de **/etc/resolv.conf** para evitar confusão.

Etapas de verificação

1. Editar o arquivo **/etc/resolv.conf** e atualizar manualmente a configuração.
2. Recarregue o serviço **NetworkManager**:

```
# systemctl reload NetworkManager
```

3. Exibir o arquivo **/etc/resolv.conf**:

```
# cat /etc/resolv.conf
```

Se você desativou com sucesso o processamento DNS, o NetworkManager não anulou as configurações manuais.

Recursos adicionais

- Para mais detalhes, consulte a descrição do parâmetro **dns** na página de manual **NetworkManager.conf(5)**.

29.2. SUBSTITUINDO /ETC/RESOLV.CONF POR UM LINK SIMBÓLICO PARA CONFIGURAR MANUALMENTE AS CONFIGURAÇÕES DO DNS

O NetworkManager não atualiza automaticamente a configuração do DNS se **/etc/resolv.conf** for um link simbólico. Esta seção descreve como substituir **/etc/resolv.conf** por um link simbólico para um arquivo alternativo com a configuração do DNS.

Pré-requisitos

- A opção **rc-manager** não está definida para **file**. Para verificar, use o comando **NetworkManager --print-config**.

Procedimento

1. Crie um arquivo, como **/etc/resolv.conf.manually-configured**, e adicione a configuração DNS para seu ambiente a ele. Use os mesmos parâmetros e sintaxe que no original **/etc/resolv.conf**.
2. Remover o arquivo **/etc/resolv.conf**:

```
# rm /etc/resolv.conf
```

3. Criar um link simbólico chamado **/etc/resolv.conf** que se refere a **/etc/resolv.conf.manually-configured**:

```
# ln -s /etc/resolv.conf.manualmente-configurado /etc/resolv.conf
```

Recursos adicionais

- Para detalhes sobre os parâmetros que você pode definir em **/etc/resolv.conf**, consulte a página de manual **resolv.conf(5)**.
- Para mais detalhes sobre por que o NetworkManager não processa configurações DNS se **/etc/resolv.conf** é um link simbólico, veja a descrição do parâmetro **rc-manager** na página de manual **NetworkManager.conf(5)**.

CAPÍTULO 30. CONFIGURAÇÃO DAS CONFIGURAÇÕES DO LINK 802.3

Você pode configurar as configurações de conexão 802.3 de uma conexão Ethernet modificando os seguintes parâmetros de configuração:

- **802-3-ethernet.auto-negotiate**
- **802-3-ethernet.speed**
- **802-3-ethernet.duplex**

Você pode configurar as configurações do link 802.3 para os seguintes modos principais:

- Ignorar a negociação do link
- Fazer valer a ativação da auto-negociação
- Configurar manualmente os links **speed** e **duplex**

30.1. CONFIGURAÇÃO DAS CONFIGURAÇÕES DE LIGAÇÃO 802.3 COM A FERRAMENTA NMCLI

Este procedimento descreve como configurar as configurações do link 802.3 usando a ferramenta **nmcli**.

Pré-requisitos

- O **NetworkManager** deve ser instalado e em funcionamento.

Procedimento

1. Para ignorar a negociação do link, defina os seguintes parâmetros:

```
~]# conexão nmcli modificar connection_name 802-3-ethernet.auto-negociate no 802-3-
ethernet.speed 0 802-3-ethernet.duplex \i
```

   Observe que o parâmetro de auto-negociação não é desativado mesmo que os parâmetros de velocidade e duplex não estejam definidos e o parâmetro de auto-negociação esteja definido como não.

2. Para impor a ativação da auto-negociação, digite o seguinte comando:

```
~]# conexão nmcli modificar connection_name 802-3-ethernet.auto-negociar sim 802-3-
ethernet.speed 0 802-3-ethernet.duplex ""
```

   Isso permite negociar todos os modos de velocidade e duplex disponíveis suportados pelo NIC.

   Você também pode habilitar a auto-negociação enquanto faz publicidade e permitir apenas um modo velocidade/duplex. Isto pode ser útil se você quiser aplicar a configuração de links **1000BASE-T** e **10GBASE-T** Ethernet, uma vez que estas normas exigem a auto-negociação ativada. Para fazer cumprir a norma **1000BASE-T**:

```
~]# conexão nmcli modificar connection_name 802-3-ethernet.auto-negociate sim 802-3-
ethernet.speed 1000 802-3-ethernet.duplex full
```

3. Para definir manualmente as configurações de velocidade e link duplex, digite o seguinte comando:

```
~]# conexão nmcli modificar connection_name 802-3-ethernet.auto-negociar sem 802-3-
ethernet.speed [velocidade em Mbit/s] 802-3-ethernet.duplex [full|half]
```

CAPÍTULO 31. CONFIGURAÇÃO DOS RECURSOS DE DESCARGA DE ETOOL

As placas de interface de rede podem usar o motor de descarga TCP (TOE) para descarregar o processamento de certas operações para o controlador da rede para melhorar o rendimento da rede.

Esta seção descreve como ativar os recursos de descarga.

31.1. RECURSOS DE DESCARREGAMENTO SUPORTADOS PELO NETWORKMANAGER

Você pode configurar os seguintes recursos para descarregar **ethtool** usando o NetworkManager:

- ethtool.feature-esp-hw-offload
- ethtool.feature-esp-tx-csum-hw-offload
- ethtool.feature-fcoe-mtu
- ethtool.feature-gro
- ethtool.feature-gso
- ethtool.feature-highdma
- ethtool.feature-hw-tc-offload
- ethtool.feature-l2-fwd-offload
- ethtool.feature-loopback
- ethtool.feature-lro
- ethtool.feature-ntuple
- ethtool.feature-rx
- ethtool.feature-rx-all
- ethtool.feature-rx-fcs
- ethtool.feature-rx-gro-hw
- ethtool.feature-rx-udp_tunnel-port-offload
- ethtool.feature-rx-vlan-filter
- ethtool.feature-rx-vlan-stag-filter
- ethtool.feature-rx-vlan-stag-hw-parse
- etool.extirpar características
- ethtool.feature-rxvlan
- ethtool.feature-sg

- ethtool.feature-tls-hw-record
- ethtool.feature-tls-hw-tx-offload
- ethtool.feature-tso
- ethtool.feature-tx
- ethtool.feature-tx-checksum-fcoe-crc
- ethtool.feature-tx-checksum-ip-generic
- ethtool.feature-tx-checksum-ipv4
- ethtool.feature-tx-checksum-ipv6
- ethtool.feature-tx-checksum-sctp
- ethtool.feature-tx-esp-segmentation
- ethtool.feature-tx-fcoe-segmentation
- ethtool.feature-tx-gre-csum-segmentação
- ethtool.feature-tx-gregmentation
- ethtool.feature-tx-gso-parcial
- ethtool.feature-tx-gso-robust
- ethtool.feature-tx-ipxip4-segmentation
- ethtool.feature-tx-ipxip6-segmentação
- ethtool.feature-tx-nocache-copy
- ethtool.feature-tx-scatter-gather
- ethtool.feature-tx-scatter-gather-fraglist
- ethtool.feature-tx-sctp-segmentation
- ethtool.feature-tx-tcp-ecn-segmentation
- ethtool.feature-tx-tcp-mangleid-segmentation
- ethtool.feature-tx-tcp-segmentation
- ethtool.feature-tx-tcp6-segmentação
- ethtool.feature-tx-udp-segmentation
- ethtool.feature-tx-udp_tnl-csum-segmentation
- ethtool.feature-tx-udp_tnl-segmentation
- ethtool.feature-tx-vlan-stag-hw-insert

- ethtool.feature-txvlan

Para detalhes sobre os recursos individuais de descarga, veja a documentação do utilitário **ethtool** e a documentação do kernel.

31.2. CONFIGURAÇÃO DE UM RECURSO DE DESCARGA DE ETOOL USANDO O NETWORKMANAGER

Esta seção descreve como habilitar e desabilitar os recursos de descarregar **ethtool** usando o NetworkManager, bem como como remover a configuração de um recurso de um perfil de conexão NetworkManager.

Procedimento

1. Por exemplo, para ativar o recurso de descarga RX e desativar a descarga TX no perfil de conexão **enp1s0**, entre:

```
# nmcli con modificar enp1s0 ethtool.feature-rx em ethtool.feature-tx off
```

   Este comando permite explicitamente a descarga do RX e desativa a descarga do TX.

2. Para remover a configuração de um recurso de descarga que você ativou ou desativou anteriormente, defina o parâmetro do recurso para **ignore**. Por exemplo, para remover a configuração para TX offload, entre:

```
# nmcli con modificar enp1s0 ethtool.feature-tx ignore
```

3. Reativar o perfil da rede:

```
# nmcli conexão acima enp1s0
```

Etapas de verificação

1. Use o comando **ethtool -k** para exibir as características atuais de descarga de um dispositivo de rede:

```
# etool -k network_device
```

Recursos adicionais

- Para obter uma lista de recursos de descarregamento do **ethtool** que o NetworkManager suporta, consulte Seção 31.1, “Recursos de descarregamento suportados pelo NetworkManager”.

31.3. UTILIZAÇÃO DE FUNÇÕES DO SISTEMA PARA DEFINIR AS CARACTERÍSTICAS DO ETOOL

Você pode usar a função do Sistema RHEL **networking** para configurar **ethtool** recursos de uma conexão NetworkManager.

IMPORTANTE

Quando você executa uma peça que usa o Sistema Função **networking** RHEL, o Sistema Função substitui um perfil de conexão existente com o mesmo nome se as configurações não coincidirem com as especificadas na peça. Portanto, sempre especifique toda a configuração do perfil de conexão de rede na peça, mesmo que, por exemplo, a configuração IP já exista. Caso contrário, o papel redefine estes valores com seus padrões.

Dependendo se já existe, o procedimento cria ou atualiza o perfil de conexão **enp1s0** com as seguintes configurações:

- Um endereço IPv4 estático - **198.51.100.20** com uma máscara de sub-rede **/24**
- Um endereço IPv6 estático - **2001:db8:1::1** com uma máscara de sub-rede **/64**
- Um gateway padrão IPv4 - **198.51.100.254**
- Um gateway padrão IPv6 - **2001:db8:1::fffe**
- Um servidor DNS IPv4 - **198.51.100.200**
- Um servidor DNS IPv6 - **2001:db8:1::ffbb**
- Um domínio de busca DNS - **example.com**
- **ethtool** apresenta:
  - Genéricos recebem descarregar (GRO): desativado
  - Segmentação genérica de descarga (GSO): habilitada
  - Segmentação do TX Stream Control Transmission Protocol (SCTP): desativado

Pré-requisitos

- Os pacotes **ansible** e **rhel-system-roles** estão instalados no nó de controle.
- Se você usar um usuário remoto diferente do root ao executar o playbook, este usuário tem as permissões apropriadas **sudo** no nó gerenciado.

Procedimento

1. Se o anfitrião no qual você deseja executar as instruções no playbook ainda não estiver inventariado, adicione o IP ou nome deste anfitrião ao arquivo **/etc/ansible/hosts** Inventário possível:

```
node.example.com
```

2. Crie o playbook **~/configure-ethernet-device-with-ethtool-features.yml** com o seguinte conteúdo:

```
---
- name. Configure an Ethernet connection with ethtool features
  hosts: node.example.com
  become: true
```

```
tasks:
- include_role:
    name: linux-system-roles.network

  vars:
    network_connections:
      - name: enp1s0
        type: ethernet
        autoconnect: yes
        ip:
          address:
            - 198.51.100.20/24
            - 2001:db8:1::1/64
          gateway4: 198.51.100.254
          gateway6: 2001:db8:1::fffe
          dns:
            - 198.51.100.200
            - 2001:db8:1::ffbb
          dns_search:
            - example.com
        ethtool:
          feature:
            gro: "no"
            gso: "yes"
            tx_sctp_segmentation: "no"
        state: up
```

3. Execute o livro de brincadeiras:

   - Para se conectar como usuário **root** ao host gerenciado, entre:

     ```
     # ansible-playbook -u root ~/configure-ethernet-device-with-ethtool-features.yml
     ```

   - Para conectar-se como usuário ao host administrado, entre:

     ```
     # ansible-playbook -u user_name --ask-become-pass ~/configure-ethernet-device-with-ethtool-features.yml
     ```

     A opção **--ask-become-pass** garante que o comando **ansible-playbook** solicita a senha **sudo** do usuário definido no **-u *user_name*** opção.

   Se você não especificar o **-u *user_name* ansible-playbook** se conecta ao host gerenciado como o usuário que está atualmente conectado ao nó de controle.

Recursos adicionais

- **ethtool** Para uma lista completa dos recursos e detalhes sobre os parâmetros usados em **network_connections**, e para informações adicionais sobre a função do sistema **network**, consulte o arquivo **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.
- Para obter detalhes sobre o comando **ansible-playbook**, consulte a página de manual **ansible-playbook(1)**.

CAPÍTULO 32. CONFIGURAÇÃO DE COALESCEAMENTO DE ETOOL

Usando a coalescência de interrupção, o sistema coleta pacotes de rede e gera uma única interrupção para vários pacotes. Isto aumenta a quantidade de dados enviados ao kernel com uma interrupção de hardware, o que reduz a carga de interrupção, e maximiza a produção.

Esta seção fornece diferentes opções para definir as configurações de coalescência do **ethtool**.

32.1. COALESCE CONFIGURAÇÕES SUPORTADAS PELO NETWORKMANAGER

Você pode definir as seguintes configurações **ethtool** coalescer usando o NetworkManager:

- [parâmetro]`coalescer-adaptativo-rx'
- [parâmetro]`coalescer-adaptativo-tx
- [parâmetro]`coalesce-pkt-taxa alta''
- [parâmetro]`coalesce-pkt-rate-low
- [parâmetro]`coalesce-rx-frames''
- [parâmetro]`coalesce-rx-frames-elevados''
- [parâmetro]`coalesce-rx-frames-irq'
- [parâmetro]`coalesce-rx-frames-low'
- [parâmetro]`coalesce-rx-usecs'
- [parâmetro]`coalesce-rx-usecs-high''
- [parâmetro]`coalesce-rx-usecs-irq'
- [parâmetro]`coalesce-rx-usecs-low'
- parâmetro]`coalescer-intervalo da amostra''
- [parâmetro]`coalesce-stats-block-usecs'
- [parâmetro]`coalesce-tx-frames''
- [parâmetro]`coalesce-tx-frames-high
- [parâmetro]`coalesce-tx-frames-irq'
- [parâmetro]`coalesce-tx-frames-low'
- [parâmetro]`coalesce-tx-usecs'
- [parâmetro]`coalesce-tx-usecs-high
- [parâmetro]`coalesce-tx-usecs-irq'

- [parâmetro]`coalesce-tx-usecs-low'

32.2. CONFIGURAÇÃO DE COALESCEAMENTO DE ETHTOOL USANDO O NETWORKMANAGER

Esta seção descreve como configurar **ethtool** coalesce as configurações usando o NetworkManager, bem como como remover a configuração de um perfil de conexão NetworkManager.

Procedimento

1. Por exemplo, para definir o número máximo de pacotes recebidos para adiar para **128** no perfil de conexão **enp1s0**, entre:

```
# conexão nmcli modificar enp1s0 ethtool.coalesce-rx-frames 128
```

2. Para remover um ajuste de coalescência, defina o ajuste para **ignore**. Por exemplo, para remover a configuração **ethtool.coalesce-rx-frames**, entre:

```
# conexão nmcli modificar enp1s0 ethtool.coalesce-rx-frames ignorar
```

3. Para reativar o perfil da rede:

```
# nmcli conexão acima enp1s0
```

Etapas de verificação

1. Use o comando **ethtool -c** para exibir as características atuais de descarga de um dispositivo de rede:

```
# ettool -c network_device
```

Recursos adicionais

- Para uma lista de configurações de coalescência **ethtool** que o NetworkManager suporta, veja Seção 32.1, “Coalesce configurações suportadas pelo NetworkManager”

CAPÍTULO 33. CONFIGURAÇÃO DE MACSEC

A seção a seguir fornece informações sobre como configurar **Media Control Access Security** (**MACsec**), que é uma tecnologia de segurança padrão 802.1AE IEEE para comunicação segura em todo o tráfego em links Ethernet.

33.1. INTRODUÇÃO AO MACSEC

Media Access Control Security (**MACsec**, IEEE 802.1AE) codifica e autentica todo o tráfego em LANs com o algoritmo GCM-AES-128. **MACsec** pode proteger não apenas **IP** mas também o Protocolo de Resolução de Endereços (ARP), Neighbor Discovery (ND), ou **DHCP**. Enquanto **IPsec** opera na camada de rede (camada 3) e **SSL** ou **TLS** na camada de aplicação (camada 7), **MACsec** opera na camada de link de dados (camada 2). Combine **MACsec** com protocolos de segurança para outras camadas de rede para tirar proveito das diferentes características de segurança que estes padrões oferecem.

33.2. USANDO MACSEC COM A FERRAMENTA NMCLI

Este procedimento mostra como configurar **MACsec** com a ferramenta **nmcli**.

Pré-requisitos

- O **NetworkManager** deve estar funcionando.
- Você já tem um CAK hexadecimal de 16 bytes (**$MKA_CAK**) e um CKN hexadecimal de 32 bytes (**$MKA_CKN**).

Procedimento

1. Para adicionar uma nova conexão usando **nmcli**, entre:

```
~]# nmcli connection add type macsec \
  con-name test-macsec+ ifname macsec0 \
  connection.autoconnect no \
  macsec.parent enp1s0 macsec.mode psk \
  macsec.mka-cak $MKA_CAK \
  macsec.mka-ckn $MKA_CKN
```

   Substitua *macsec0* pelo nome do dispositivo que você deseja configurar.

2. Para ativar a conexão, entre:

```
~]# nmcli conexão up test-macsec
```

Após esta etapa, o dispositivo *macsec0* é configurado e pode ser utilizado para a criação de redes.

33.3. USANDO MACSEC COM WPA_SUPPLICANT

Este procedimento mostra como habilitar **MACsec** com um switch que realiza a autenticação usando um par pré-partilhado de Chave de Conectividade de Associação/CAK Name (CAK/CKN).

Procedimento

1. Criar um par CAK/CKN. Por exemplo, o seguinte comando gera uma chave de 16 bytes em notação hexadecimal:

```
~]$ dd if=/dev/urandom count=16 bs=1 2> /dev/null | hexdump -e '1/2 "%02x"'
```

2. Crie o arquivo de configuração **wpa_supplicant.conf** e acrescente as seguintes linhas a ele:

```
ctrl_interface=/var/run/wpa_supplicant
eapol_version=3
ap_scan=0
fast_reauth=1

network={
    key_mgmt=NONE
    eapol_flags=0
    macsec_policy=1

    mka_cak=0011... # 16 bytes hexadecimal
    mka_ckn=2233... # 32 bytes hexadecimal
}
```

   Use os valores da etapa anterior para completar as linhas **mka_cak** e **mka_ckn** no arquivo de configuração **wpa_supplicant.conf**.

   Para mais informações, consulte a página de manual **wpa_supplicant.conf(5)**.

3. Assumindo que você está usando *wlp61s0* para se conectar à sua rede, comece **wpa_supplicant** usando o seguinte comando:

```
~]# wpa_supplicant -i wlp61s0 -Dmacsec_linux -c wpa_supplicant.conf
```

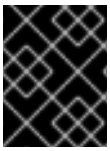
33.4. INFORMAÇÕES RELACIONADAS

Para mais detalhes, veja o artigo O que há de novo em MACsec: criação de MACsec usando wpa_supplicant e (opcionalmente) NetworkManager. Além disso, veja o artigo MACsec: uma solução diferente para criptografar o tráfego de rede para mais informações sobre a arquitetura de uma rede **MACsec**, cenários de caso de uso e exemplos de configuração.

CAPÍTULO 34. USANDO DIFERENTES SERVIDORES DNS PARA DIFERENTES DOMÍNIOS

Por default, o Red Hat Enterprise Linux (RHEL) envia todas as solicitações DNS para o primeiro servidor DNS especificado no arquivo **/etc/resolv.conf**. Se este servidor não responder, a RHEL usa o próximo servidor neste arquivo.

Em ambientes onde um servidor DNS não pode resolver todos os domínios, os administradores podem configurar a RHEL para enviar solicitações DNS de um domínio específico para um servidor DNS selecionado. Por exemplo, você pode configurar um servidor DNS para resolver consultas para **example.com** e outro servidor DNS para resolver consultas para **example.net**. Para todas as outras solicitações DNS, a RHEL utiliza o servidor DNS configurado na conexão com o gateway padrão.



IMPORTANTE

No RHEL 8, a Red Hat fornece **systemd-resolved** como uma prévia tecnológica sem suporte.

34.1. ENVIO DE SOLICITAÇÕES DNS PARA UM DOMÍNIO ESPECÍFICO PARA UM SERVIDOR DNS SELECIONADO

Esta seção configura o serviço **systemd-resolved** e o NetworkManager para enviar consultas ao DNS de um domínio específico para um servidor DNS selecionado.

Se você completar o procedimento nesta seção, a RHEL utiliza o serviço DNS fornecido por **systemd-resolved** no arquivo **/etc/resolv.conf**. O serviço **systemd-resolved** inicia um serviço DNS que escuta na porta **53** endereço IP **127.0.0.53**. O serviço encaminha dinamicamente as solicitações DNS para os servidores DNS correspondentes especificados no NetworkManager.

NOTA

O endereço **127.0.0.53** só é acessível a partir do sistema local e não a partir da rede.

Pré-requisitos

- O sistema tem múltiplas conexões NetworkManager configuradas.
- Um servidor DNS e um domínio de busca são configurados nas conexões NetworkManager, responsáveis pela resolução de um domínio específico
  Por exemplo, se o servidor DNS especificado em uma conexão VPN deve resolver as consultas para o domínio **example.com**, o perfil da conexão VPN deve ter:
  - Configurou um servidor DNS que pode resolver **example.com**
  - Configurou o domínio de busca para **example.com** nos parâmetros **ipv4.dns-search** e **ipv6.dns-search**

Procedimento

1. Inicie e habilite o serviço **systemd-resolved**:

```
# systemctl - agora habilita sistema-resolvido
```

2. Edite o arquivo **/etc/NetworkManager/NetworkManager.conf**, e defina a seguinte entrada na seção **[main]**:

```
dns=resolvido pelo sistema
```

3. Recarregue o serviço **NetworkManager**:

```
# systemctl reload NetworkManager
```

Etapas de verificação

1. Verifique se a entrada **nameserver** no arquivo **/etc/resolv.conf** se refere a **127.0.0.53**:

```
# cat /etc/resolv.conf
nameserver 127.0.0.53
```

2. Verifique se o serviço **systemd-resolved** ouve na porta **53** no endereço IP local **127.0.0.53**:

```
# netstat -tulpn | grep "127.0.0.53:53"
tcp   0   0 127.0.0.53:53   0.0.0.0:*   LISTEN   1050/systemd-resolv
udp   0   0 127.0.0.53:53   0.0.0.0:*            1050/systemd-resolv
```

Recursos adicionais

- Para mais detalhes, consulte a descrição do parâmetro **dns** na página de manual **NetworkManager.conf(5)**.

CAPÍTULO 35. COMEÇANDO COM IPVLAN

Este documento descreve o driver IPVLAN.

35.1. VISÃO GERAL DO IPVLAN

IPVLAN é um driver para um dispositivo de rede virtual que pode ser usado em ambiente de contêineres para acessar a rede anfitriã. IPVLAN expõe um único endereço MAC à rede externa, independentemente do número de dispositivos IPVLAN criados dentro da rede host. Isto significa que um usuário pode ter vários dispositivos IPVLAN em vários contêineres e o switch correspondente lê um único endereço MAC. O driver IPVLAN é útil quando o switch local impõe restrições ao número total de endereços MAC que ele pode gerenciar.

35.2. MODOS IPVLAN

Os seguintes modos estão disponíveis para IPVLAN:

- **L2 mode**
  No IPVLAN **L2 mode**, os dispositivos virtuais recebem e respondem às solicitações do Protocolo de Resolução de Endereços (ARP). A estrutura **netfilter** funciona apenas dentro do contêiner que possui o dispositivo virtual. No **netfilter** as cadeias são executadas no espaço de nomes padrão no tráfego do contêiner. O uso de **L2 mode** proporciona bom desempenho, mas menos controle sobre o tráfego da rede.

- **L3 mode**
  Em **L3 mode**, os dispositivos virtuais processam apenas **L3** tráfego e acima. Os dispositivos virtuais não respondem à solicitação da ARP e os usuários devem configurar manualmente as entradas vizinhas para os endereços IPVLAN nos pares relevantes. O tráfego de saída de um contêiner relevante é desembarcado na cadeia **netfilter** POSTROUTING e OUTPUT no namespace padrão, enquanto o tráfego de entrada é rosqueado da mesma forma que **L2 mode**. O uso de **L3 mode** proporciona um bom controle, mas diminui o desempenho do tráfego de rede.

- **L3S mode**
  Em **L3S mode**, os dispositivos virtuais processam da mesma forma que em **L3 mode**, exceto que tanto a saída como a entrada de um contêiner relevante são desembarcadas na cadeia **netfilter** no namespace padrão. **L3S mode** comporta-se de forma semelhante a **L3 mode**, mas proporciona maior controle da rede.

NOTA

O dispositivo virtual IPVLAN não recebe tráfego broadcast e multicast no caso de **L3** e **L3S modes**.

35.3. VISÃO GERAL DO MACVLAN

O driver MACVLAN permite criar múltiplos dispositivos de rede virtual sobre um único NIC, cada um deles identificado por seu próprio endereço MAC único. Os pacotes que aterrissam no NIC físico são desmultiplexados em direção ao dispositivo MACVLAN relevante através do endereço MAC do destino. Os dispositivos MACVLAN não adicionam nenhum nível de encapsulamento.

35.4. COMPARAÇÃO ENTRE IPVLAN E MACVLAN

A tabela a seguir mostra as principais diferenças entre MACVLAN e IPVLAN.

MACVLAN	IPVLAN
Utiliza endereço MAC para cada dispositivo MACVLAN. O excesso de endereços MAC da tabela MAC no switch pode causar a perda da conectividade.	Utiliza um único endereço MAC que não limita o número de dispositivos IPVLAN.
As regras do Netfilter para namespace global não podem afetar o tráfego para ou do dispositivo MACVLAN em um namespace infantil.	É possível controlar o tráfego de ou para o dispositivo IPVLAN em **L3 mode** e **L3S mode**.

Note que tanto IPVLAN quanto MACVLAN não exigem nenhum nível de incapsulação.

35.5. CRIAÇÃO E CONFIGURAÇÃO DO DISPOSITIVO IPVLAN USANDO IPROUTE2

Este procedimento mostra como configurar o dispositivo IPVLAN usando o iproute2.

Procedimento

1. Para criar um dispositivo IPVLAN, digite o seguinte comando:

```
~]# link ip adicionar link real_NIC_device nome IPVLAN_device tipo ipvlan mode l2
```

   Note que o controlador de interface de rede (NIC) é um componente de hardware que conecta um computador a uma rede.

   Exemplo 35.1. Criação de um dispositivo IPVLAN

```
~]# ip link add link enp0s31f6 name my_ipvlan type ipvlan mode l2
~]# ip link
47: my_ipvlan@enp0s31f6: <BROADCAST,MULTICAST> mtu 1500 qdisc noop state
DOWN mode DEFAULT group default qlen 1000 link/ether e8:6a:6e:8a:a2:44 brd
ff:ff:ff:ff:ff:ff
```

2. Para atribuir um endereço **IPv4** ou **IPv6** à interface, digite o seguinte comando:

```
~]# ip address add dev IPVLAN_device IP_address/subnet_mask_prefix
```

3. No caso de configurar um dispositivo IPVLAN em **L3 mode** ou **L3S mode**, faça as seguintes configurações:

   a. Configurar a configuração do vizinho para o colega remoto no host remoto:

```
~]# ip neigh add dev peer_device IPVLAN_device_IP_address lladdr MAC_address
```

   onde *MAC_address* é o endereço MAC do verdadeiro NIC no qual se baseia um dispositivo IPVLAN.

b. Configure um dispositivo IPVLAN para **L3 mode** com o seguinte comando:

```
~]# ip neigh add dev real_NIC_device peer_IP_address lladdr peer_MAC_address
```

Para **L3S mode**:

```
~]# ip route dev adicionar real_NIC_device peer_IP_address/32
```

em que o endereço IP representa o endereço do par remoto.

4. Para colocar um dispositivo IPVLAN ativo, digite o seguinte comando:

```
~]# ip link set dev IPVLAN_device up
```

5. Para verificar se o dispositivo IPVLAN está ativo, execute o seguinte comando no host remoto:

```
~]# ping IP_address
```

onde o *IP_address* usa o endereço IP do dispositivo IPVLAN.

CAPÍTULO 36. CONFIGURAÇÃO DE ENCAMINHAMENTO E ENCAMINHAMENTO VIRTUAL (VRF)

Com o encaminhamento e encaminhamento virtual (VRF), os administradores podem usar várias tabelas de encaminhamento simultaneamente no mesmo host. Para isso, o VRF separa uma rede na camada 3. Isto permite que o administrador isole o tráfego usando tabelas de roteamento separadas e independentes por domínio VRF. Esta técnica é similar às LANs virtuais (VLAN), que dividem uma rede na camada 2, onde o sistema operacional usa diferentes tags VLAN para isolar o tráfego compartilhando o mesmo meio físico.

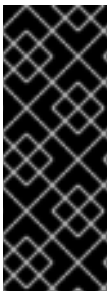
Um benefício do VRF sobre a partição na camada 2 é que as escalas de roteamento são melhores considerando o número de pares envolvidos.

O Red Hat Enterprise Linux usa um dispositivo virtual **vrt** para cada domínio VRF e adiciona rotas a um domínio VRF ao adicionar dispositivos de rede existentes a um dispositivo VRF. Endereços e rotas previamente anexados ao dispositivo original serão movidos dentro do domínio VRF.

Observe que cada domínio VRF é isolado um do outro.

36.1. REUTILIZAÇÃO PERMANENTE DO MESMO ENDEREÇO IP EM INTERFACES DIFERENTES

Este procedimento descreve como usar permanentemente o mesmo endereço IP em diferentes interfaces em um servidor, usando a função VRF.



IMPORTANTE

Para que os pares remotos possam contatar ambas as interfaces VRF enquanto reutilizam o mesmo endereço IP, as interfaces de rede devem pertencer a domínios de transmissão diferentes. Um domínio de transmissão em uma rede é um conjunto de nós, que recebem o tráfego de transmissão enviado por qualquer um deles. Na maioria das configurações, todos os nós conectados ao mesmo switch pertencem ao mesmo domínio de radiodifusão.

Pré-requisitos

- Você está logado como usuário do **root**.
- As interfaces de rede não são configuradas.

Procedimento

1. Criar e configurar o primeiro dispositivo VRF:
   a. Criar uma conexão para o dispositivo VRF e atribuí-la a uma tabela de roteamento. Por exemplo, para criar um dispositivo VRF chamado **vrf0** que é atribuído à tabela de roteamento **1001**:

```
# nmcli connection add type vrf ifname vrf0 con-name vrf0 tabela 1001 ipv4.method disabled ipv6.method disabled
```

   b. Habilite o dispositivo **vrf0**:

```
# nmcli conexão acima vrf0
```

c. Atribuir um dispositivo de rede ao VRF recém-criado. Por exemplo, para adicionar o dispositivo **enp1s0** Ethernet ao dispositivo **vrf0** VRF e atribuir um endereço IP e a máscara de sub-rede a **enp1s0**, entre:

```
# nmcli connection add type ethernet con-name vrf.enp1s0 ifname enp1s0 master vrf0
ipv4.method manual ipv4.address 192.0.2.1/24
```

d. Ativar a conexão **vrf.enp1s0**:

```
# nmcli conexão acima vrf.enp1s0
```

2. Criar e configurar o próximo dispositivo VRF:

a. Criar o dispositivo VRF e atribuí-lo a uma tabela de roteamento. Por exemplo, para criar um dispositivo VRF chamado **vrf1** que é atribuído à tabela de roteamento **1002**, entre:

```
# nmcli connection add type vrf ifname vrf1 con-name vrf1 tabela 1002 ipv4.method
disabled ipv6.method disabled
```

b. Ativar o dispositivo **vrf1**:

```
# nmcli conexão acima vrf1
```

c. Atribuir um dispositivo de rede ao VRF recém-criado. Por exemplo, para adicionar o dispositivo **enp7s0** Ethernet ao dispositivo **vrf1** VRF e atribuir um endereço IP e a máscara de sub-rede a **enp7s0**, entre:

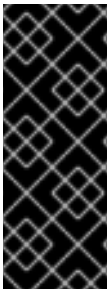
```
# nmcli connection add type ethernet con-name vrf.enp7s0 ifname enp7s0 master vrf1
ipv4.method manual ipv4.address 192.0.2.1/24
```

d. Ativar o dispositivo **vrf.enp7s0**:

```
# nmcli conexão acima vrf.enp7s0
```

36.2. REUTILIZAÇÃO TEMPORÁRIA DO MESMO ENDEREÇO IP EM INTERFACES DIFERENTES

O procedimento nesta seção descreve como usar temporariamente o mesmo endereço IP em diferentes interfaces em um servidor, usando o recurso de encaminhamento e encaminhamento virtual (VRF). Use este procedimento apenas para fins de teste, pois a configuração é temporária e perdida após o reinício do sistema.



IMPORTANTE

Para que os pares remotos possam contatar ambas as interfaces VRF enquanto reutilizam o mesmo endereço IP, as interfaces de rede devem pertencer a domínios de transmissão diferentes. Um domínio de difusão em uma rede é um conjunto de nós que recebem o tráfego de difusão enviado por qualquer um deles. Na maioria das configurações, todos os nós conectados ao mesmo switch pertencem ao mesmo domínio de radiodifusão.

Pré-requisitos

- Você está logado como usuário do **root**.
- As interfaces de rede não são configuradas.

Procedimento

1. Criar e configurar o primeiro dispositivo VRF:
   a. Criar o dispositivo VRF e atribuí-lo a uma tabela de roteamento. Por exemplo, para criar um dispositivo VRF chamado **blue** que é atribuído à tabela de roteamento **1001**:

```
# ip link add dev azul tipo vrf tabela 1001
```

   b. Habilite o dispositivo **blue**:

```
# ip link set dev blue up
```

   c. Atribuir um dispositivo de rede ao dispositivo VRF. Por exemplo, para adicionar o dispositivo **enp1s0** Ethernet ao dispositivo **blue** VRF:

```
# ip link set dev enp1s0 master blue
```

   d. Habilite o dispositivo **enp1s0**:

```
# ip link set dev enp1s0 up
```

   e. Atribuir um endereço IP e uma máscara de sub-rede ao dispositivo **enp1s0**. Por exemplo, para configurá-la para **192.0.2.1/24**:

```
# ip address add dev enp1s0 192.0.2.1/24
```

2. Criar e configurar o próximo dispositivo VRF:
   a. Criar o dispositivo VRF e atribuí-lo a uma tabela de roteamento. Por exemplo, para criar um dispositivo VRF chamado **red** que é atribuído à tabela de roteamento **1002**:

```
# ip link add dev vermelho tipo vrf tabela 1002
```

   b. Habilite o dispositivo **red**:

```
# ip link set dev red up
```

   c. Atribuir um dispositivo de rede ao dispositivo VRF. Por exemplo, para adicionar o dispositivo **enp7s0** Ethernet ao dispositivo **red** VRF:

```
# ip link set dev enp7s0 master red
```

   d. Habilite o dispositivo **enp7s0**:

```
# ip link set dev enp7s0 up
```

e. Atribua ao dispositivo **enp7s0** o mesmo endereço IP e máscara de sub-rede que você usou para **enp1s0** no domínio **blue** VRF:

```
# ip address add dev enp7s0 192.0.2.1/24
```

3. Opcionalmente, criar outros dispositivos VRF, como descrito acima.

36.3. INFORMAÇÕES RELACIONADAS

- https://www.kernel.org/doc/Documentation/networking/vrf.txt

CAPÍTULO 37. DEFININDO OS PROTOCOLOS DE ROTEAMENTO PARA SEU SISTEMA

Esta seção descreve como usar o **Free Range Routing** (**FRRouting**, ou **FRR**) para habilitar e definir os protocolos de roteamento necessários para seu sistema.

37.1. INTRODUÇÃO AO FRROUTING

Free Range Routing (**FRRouting**, ou **FRR**) é uma pilha de protocolos de roteamento, que é fornecida pelo pacote **frr**, disponível no repositório **AppStream**.

FRR substitui **Quagga** que era usado em versões anteriores da RHEL. Como tal, **FRR** fornece serviços de roteamento baseados em TCP/IP com suporte a múltiplos protocolos de roteamento IPv4 e IPv6.

Os protocolos suportados são:

- Protocolo Border Gateway (**BGP**)
- Sistema Intermediário a Sistema Intermediário (**IS-IS**)
- Abrir primeiro o caminho mais curto (**OSPF**)
- Multicast Independente de Protocolo (**PIM**)
- Protocolo de Informação de Roteamento (**RIP**)
- Routing Information Protocol next generation (**RIPng**)
- Protocolo de roteamento de portal interior aprimorado (**EIGRP**)
- Próximo Protocolo de Resolução do Lúpulo (**NHRP**)
- Detecção de Encaminhamento Bidirecional (**BFD**)
- Roteamento baseado em políticas (**PBR**)

FRR é um conjunto dos seguintes serviços:

- zebra
- bgpd
- isisd
- ospfd
- ospf6d
- pimd
- ripd
- ripngd
- eigrpd

- nhrpd
- bfdd
- pbrd
- estáticod
- fabricd

Se **frr** estiver instalado, o sistema pode atuar como um roteador dedicado, que troca informações de roteamento com outros roteadores tanto na rede interna quanto externa, utilizando os protocolos de roteamento.

37.2. ESTABELECENDO O FRROUTING

Pré-requisitos

- Certifique-se de que o pacote **frr** esteja instalado em seu sistema:

```
# yum instalar frr
```

Procedimento

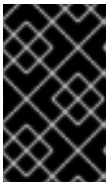
1. Edite o arquivo de configuração **/etc/frr/daemons**, e habilite os daemons necessários para seu sistema.
   Por exemplo, para habilitar o daemon **ripd**, inclua a seguinte linha:

   ```
   ripd=yes
   ```

   ATENÇÃO

   O daemon **zebra** deve estar sempre habilitado, de modo que você deve definir **zebra=yes** para poder usar **FRR**.

   IMPORTANTE

   Por padrão, **/etc/frr/daemons** contém **[daemon_name]=no** entradas para todos os daemons. Portanto, todos os demônios estão desativados, e a partir de **FRR** após uma nova instalação do sistema não tem efeito.

2. Iniciar o serviço **frr**:

   ```
   # systemctl start frr
   ```

3. Opcionalmente, você também pode definir **FRR** para iniciar automaticamente na inicialização:

   ```
   # systemctl habilita frr
   ```

-

37.3. MODIFICANDO A CONFIGURAÇÃO DO FRR

Esta seção descreve:

- Como ativar um daemon adicional após a instalação **FRR**
- Como desativar um daemon depois de ter se instalado **FRR**

Possibilitando um daemon adicional

Pré-requisitos

- **FRR** é criado como descrito em Seção 37.2, “Estabelecendo o FRRouting” .

Procedimento

Para habilitar um ou mais demônios adicionais:

1. Edite o arquivo de configuração **/etc/frr/daemons** e modifique a linha dos daemons necessários para declarar **yes** em vez de **no**.
   Por exemplo, para habilitar o daemon **ripd**:

   ```
   ripd=yes
   ```

2. Recarregue o serviço **frr**:

   ```
   # systemctl reload frr
   ```

Desabilitando um daemon

Pré-requisitos

- **FRR** é criado como descrito em Seção 37.2, “Estabelecendo o FRRouting” .

Procedimento

Para desativar um ou mais demônios:

1. Edite o arquivo de configuração **/etc/frr/daemons** e modifique a linha dos daemons necessários para declarar **no** em vez de **yes**.
   Por exemplo, para desativar o daemon **ripd**:

   ```
   ripd=não
   ```

2. Recarregue o serviço **frr**:

   ```
   # systemctl reload frr
   ```

37.4. MODIFICAR UMA CONFIGURAÇÃO DE UM DETERMINADO DAEMON

Com a configuração padrão, cada daemon de roteamento em **FRR** só pode atuar como um roteador simples.

Para qualquer configuração adicional de um daemon, use o seguinte procedimento.

Procedimento

1. Dentro do diretório **/etc/frr/**, crie um arquivo de configuração para o daemon necessário, e nomeie o arquivo da seguinte forma:

   ```
   [daemon_name].conf
   ```

   Por exemplo, para configurar ainda mais o daemon **eigrpd**, crie o arquivo **eigrpd.conf** no diretório mencionado.

2. Povoar o novo arquivo com o conteúdo necessário.
   Para exemplos de configuração de **FRR** daemons, consulte o diretório **/usr/share/doc/frr/**.

3. Recarregue o serviço **frr**:

   ```
   # systemctl reload frr
   ```

CAPÍTULO 38. MONITORAMENTO E AJUSTE DO BUFFER DE ANÉIS RX

Os buffers de anel de recepção (RX) são buffers compartilhados entre o driver do dispositivo e a placa de interface de rede (NIC), e armazenam os pacotes recebidos até que o driver do dispositivo possa processá-los.

Você pode aumentar o tamanho do dispositivo Ethernet RX ring buffer se a taxa de queda de pacotes fizer com que as aplicações sejam relatadas:

- uma perda de dados,
- cerca de cacho,
- desempenho lento,
- tempo esgotado, e
- backups falhados.

Esta seção descreve como identificar o número de pacotes descartados e aumentar o buffer do anel RX para reduzir uma alta taxa de queda de pacotes.

38.1. EXIBINDO O NÚMERO DE PACOTES DESCARTADOS

O utilitário **ethtool** permite que os administradores consultem, configurem ou controlem as configurações do driver da rede.

A exaustão do buffer do anel RX causa um incremento nos contadores, como "descarte" ou "queda" na saída de **ethtool -S *interface_name***. Os pacotes descartados indicam que o buffer disponível está se enchendo mais rápido do que o kernel pode processar os pacotes.

Este procedimento descreve como exibir contadores de queda usando **ethtool**.

Procedimento

- Para exibir contadores de queda para o ***enp1s0*** interface, entre:

```
US$ etool -S enp1s0
```

38.2. AUMENTAR O BUFFER DO ANEL RX PARA REDUZIR UMA ALTA TAXA DE QUEDA DE PACOTES

A utilidade **ethtool** ajuda a aumentar o buffer RX para reduzir uma alta taxa de queda de pacotes.

Procedimento

1. Para visualizar o tamanho máximo do tampão de anel RX:

```
# ethtool -g enp1s0
 Ring parameters for enp1s0:
 Pre-set maximums:
 RX:             4080
```

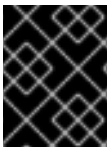
```
RX Mini:        0
RX Jumbo:       16320
TX:             255
Current hardware settings:
RX:             255
RX Mini:        0
RX Jumbo:       0
TX:             255
```

2. Se os valores na seção **Pre-set maximums** forem superiores aos da seção **Current hardware settings**, aumente o buffer de anéis RX:

   - Para mudar temporariamente o buffer de anéis RX do dispositivo **enp1s0** para **4080**, entre:

     ```
     # ethtool -G enp1s0 rx 4080
     ```

   - Para alterar permanentemente o buffer de anéis RX, crie um script de despacho NetworkManager.
     Para obter detalhes, veja o artigo Como fazer com que as configurações de etool NIC sejam persistentes (aplicar automaticamente na inicialização) e criar um script do despachante.



IMPORTANTE

Dependendo do driver que sua placa de interface de rede utiliza, a mudança no buffer de anel pode interromper em breve a conexão de rede.

Recursos adicionais

- Para maiores informações sobre estatísticas que cobrem mais razões para descarte de pacotes indesejados, veja o *ifconfig and ip commands report packet drops in RHEL7* artigo.
- *Should I be concerned about a 0.05% packet drop rate?*
- A página do homem **ethtool(8)**.

CAPÍTULO 39. TESTE DE CONFIGURAÇÕES BÁSICAS DE REDE

Esta seção descreve como realizar testes básicos de rede.

39.1. USANDO O UTILITÁRIO PING PARA VERIFICAR A CONEXÃO IP COM OUTROS HOSTS

O utilitário **ping** envia pacotes ICMP para um host remoto. Você pode usar esta funcionalidade para testar se a conexão IP a um host diferente funciona.

Procedimento

- Pingando o endereço IP de um host na mesma sub-rede, tal como seu gateway padrão:

```
# ping 192.0.2.3
```

  Se o comando falhar, verifique as configurações padrão do gateway.

- Pingar um endereço IP de um host em uma sub-rede remota:

```
# ping 198.162.3.1
```

  Se o comando falhar, verifique as configurações padrão do gateway e certifique-se de que o gateway encaminha os pacotes entre as redes conectadas.

39.2. USANDO O UTILITÁRIO HOSPEDEIRO PARA VERIFICAR A RESOLUÇÃO DO NOME

Este procedimento descreve como verificar a resolução do nome no Red Hat Enterprise Linux 8.

Procedimento

- Use o utilitário **host** para verificar se a resolução do nome funciona. Por exemplo, para resolver o **client.example.com** hostname para um endereço IP, digite:

```
# host client.example.com
```

  Se o comando retornar um erro, tal como **connection timed out** ou **no servers could be reached**, verifique suas configurações de DNS.

CAPÍTULO 40. INTRODUÇÃO AO NETWORKMANAGER DEBUGGING

O aumento dos níveis de registro para todos ou certos domínios ajuda a registrar mais detalhes das operações que o NetworkManager realiza. Os administradores podem usar estas informações para solucionar problemas. O NetworkManager fornece diferentes níveis e domínios para produzir informações de registro. O arquivo **/etc/NetworkManager/NetworkManager.conf** é o principal arquivo de configuração do NetworkManager. Os logs são armazenados no diário.

Esta seção fornece informações sobre como permitir o registro de depuração para o NetworkManager e usar diferentes níveis de registro e domínios para configurar a quantidade de detalhes de registro.

40.1. NÍVEIS E DOMÍNIOS DE DEPURAÇÃO

Você pode usar os parâmetros **levels** e **domains** para gerenciar a depuração para o NetworkManager. O nível define o nível de verbosidade, enquanto que os domínios definem a categoria das mensagens para registrar os logs com determinada severidade (**level**).

Níveis de log	Descrição
OFF	Não registra nenhuma mensagem sobre o NetworkManager
ERR	Registra apenas erros críticos
WARN	Avisos de registros que podem refletir a operação
INFO	Registra várias mensagens informativas que são úteis para rastrear o estado e as operações
DEBUG	Permite o registro verboso para fins de depuração
TRACE	Possibilita mais extração verbosa do que o nível **DEBUG**

Observe que níveis subsequentes registram todas as mensagens de níveis anteriores. Por exemplo, definir o nível de log para **INFO** também registra as mensagens contidas nos níveis de log **ERR** e **WARN**.

Recursos adicionais

- Para obter detalhes em **domains**, consulte a página de manual **NetworkManager.conf(5)**.

40.2. DEFINIÇÃO DO NÍVEL DE REGISTRO DO NETWORKMANAGER

Por padrão, todos os domínios de log são definidos para registrar o nível de log **INFO**. Desabilite a limitação da taxa antes de coletar os logs de debug. Com a limitação de taxa, **systemd-journald** deixa cair mensagens se houver muitas delas em um curto espaço de tempo. Isto pode ocorrer quando o nível de log é **TRACE**.

Este procedimento desativa a limitação da taxa e permite o registro de logs de depuração para todos os domínios (TODOS).

Procedimento

Procedimento

1. Para desativar a limitação de taxas, edite o arquivo **/etc/systemd/journald.conf**, descomente o parâmetro **RateLimitBurst** na seção **[Journal]**, e defina seu valor como **0**:

```
RateLimitBurst=0
```

2. Reinicie o serviço **systemd-journald**.

```
# systemctl restart systemd-journald
```

3. Crie o arquivo **/etc/NetworkManager/conf.d/95-nm-debug.conf** com o seguinte conteúdo:

```
[logging]
domains=ALL:DEBUG
```

   O parâmetro **domains** pode conter múltiplos pares separados por vírgula **domain:level**.

4. Reinicie o serviço NetworkManager.

```
# systemctl restart NetworkManager
```

40.3. AJUSTE TEMPORÁRIO DOS NÍVEIS DE REGISTRO EM TEMPO DE EXECUÇÃO USANDO NMCLI

Você pode alterar o nível de registro em tempo de execução usando **nmcli**. Entretanto, a Red Hat recomenda permitir a depuração usando arquivos de configuração e reiniciar o NetworkManager. A atualização da depuração **levels** e **domains** usando o arquivo **.conf** ajuda a depurar problemas de inicialização e captura todos os logs a partir do estado inicial.

Procedimento

1. Opcional: Exibir as configurações atuais de registro:

```
# nmcli general logging
  LEVEL  DOMAINS
  INFO
PLATFORM,RFKILL,ETHER,WIFI,BT,MB,DHCP4,DHCP6,PPP,WIFI_SCAN,IP4,IP6,A
UTOIP4,DNS,VPN,SHARING,SUPPLICANT,AGENTS,SETTINGS,SUSPEND,CORE,DEVIC
E,OLPC,
WIMAX,INFINIBAND,FIREWALL,ADSL,BOND,VLAN,BRIDGE,DBUS_PROPS,TEAM,CONC
HECK,DC
B,DISPATCH
```

2. Para modificar o nível de registro e os domínios, use as seguintes opções:

   - Para definir o nível de log para todos os domínios para o mesmo **LEVEL**, entre:

```
# nmcli general logging level LEVEL domains ALL
```

   - Para mudar o nível para domínios específicos, entre:

```
# nmcli general logging level LEVEL domains DOMAINS
```

Observe que atualizar o nível de registro usando este comando desabilita o registro para todos os outros domínios.

- Para alterar o nível de domínios específicos e preservar o nível de todos os outros domínios, entre:

```
# nmcli general logging level KEEP domains DOMAIN:LEVEL,DOMAIN:LEVEL
```

40.4. VISUALIZAÇÃO DOS LOGS DO NETWORKMANAGER

Você pode visualizar os logs do NetworkManager para a solução de problemas.

Procedimento

- Para ver os logs, entre:

```
# journalctl -u NetworkManager -b
```

Recursos adicionais

- A página do homem **NetworkManager.conf(5)**
- A página do homem **journalctl**

CAPÍTULO 41. CAPTURA DE PACOTES DE REDE

Para depurar problemas de rede e comunicações, você pode capturar pacotes de rede. As seções seguintes fornecem instruções e informações adicionais sobre a captura de pacotes de rede.

41.1. USANDO O XDPDUMP PARA CAPTURAR PACOTES DE REDE, INCLUINDO PACOTES DESCARTADOS POR PROGRAMAS XDP

O utilitário **xdpdump** captura os pacotes da rede. Ao contrário do utilitário **tcpdump**, **xdpdump** usa um programa estendido de Filtro de Pacotes de Berkeley (eBPF) para esta tarefa. Isto permite que **xdpdump** também capture pacotes descartados por programas Express Data Path (XDP). Os utilitários de espaço do usuário, como **tcpdump**, não são capazes de capturar estes pacotes descartados, assim como os pacotes originais modificados por um programa XDP.

Você pode usar **xdpdump** para depurar programas XDP que já estão anexados a uma interface. Portanto, o utilitário pode capturar pacotes antes que um programa XDP seja iniciado e depois que ele tenha terminado. No último caso, **xdpdump** também captura a ação do XDP. Por padrão, **xdpdump** captura os pacotes recebidos na entrada do programa XDP.

IMPORTANTE

A Red Hat fornece o site **xdpdump** como uma prévia tecnológica sem suporte.

Note que **xdpdump** não tem capacidade de filtragem de pacotes ou decodificação. Entretanto, você pode usá-lo em combinação com **tcpdump** para a decodificação de pacotes.

O procedimento descreve como capturar todos os pacotes na interface **enp1s0** e escrevê-los no arquivo **/root/capture.pcap**.

Pré-requisitos

- Um driver de rede que suporta programas XDP.
- Um programa XDP é carregado para a interface **enp1s0**. Se nenhum programa for carregado, **xdpdump** captura os pacotes de forma semelhante **tcpdump** o faz, para compatibilidade retroativa.

Procedimento

1. Para capturar os pacotes na interface **enp1s0** e escrevê-los no arquivo **/root/capture.pcap**, entre:

```
# xdpdump -i enp1s0 -w /root/capture.pcap
```

2. Para parar de capturar os pacotes, pressione **Ctrl**+**C**.

Recursos adicionais

- Para mais detalhes sobre **xdpdump**, consulte a página de manual **xdpdump(8)**.
- Se você é um desenvolvedor e está interessado no código fonte de **xdpdump**, baixe e instale o RPM (SRPM) de origem correspondente no Portal do Cliente da Red Hat.

41.2. RECURSOS ADICIONAIS

- Como capturar pacotes de rede com a solução de base de conhecimento tcpdump?

CAPÍTULO 42. USANDO UMA VERSÃO ESPECÍFICA DO KERNEL NA RHEL

O kernel é um componente central de um sistema operacional Linux que gerencia os recursos do sistema, e fornece a interface entre aplicações de hardware e software. Em alguns casos, o kernel pode afetar a funcionalidade da rede, por isso é sempre recomendável usar a versão mais recente do kernel. Se necessário, também é possível rebaixar a versão do kernel para uma versão anterior do mesmo kernel x-stream e selecionar uma versão específica enquanto o sistema é inicializado para a solução de problemas.

Esta seção explica como selecionar um kernel no carregador de inicialização do GRUB caso você atualize ou diminua o kernel.

42.1. INICIANDO A RHEL USANDO UMA VERSÃO ANTERIOR DO KERNEL

Por padrão, após a atualização, o sistema inicia a última versão do kernel. O Red Hat Enterprise Linux permite ter três versões do kernel instaladas ao mesmo tempo. Isto está definido no arquivo **/etc/dnf/dnf.conf** (**installonly_limit=3**).

Se você observar qualquer problema quando o sistema for carregado com o novo kernel, você pode reiniciá-lo com o kernel anterior e restaurar a máquina de produção. Entre em contato com o suporte para solucionar o problema.

Procedimento

1. Inicie o sistema.
2. No carregador de inicialização GRUB, você vê os kernels instalados. Use as teclas ↑ e ↓ para selecionar um kernel, e pressione **Enter** para inicializá-lo.

Recursos adicionais

- Alterando o kernel padrão para inicializar usando a ferramenta **grubby**.
- Para mais informações sobre a instalação e atualização do Kernel, consulte a seção Atualização do Kernel com yum.

CAPÍTULO 43. PRESTAÇÃO DE SERVIÇOS DE DHCP

O Dynamic Host Configuration Protocol (DHCP) é um protocolo de rede que atribui automaticamente informações IP aos clientes.

Esta seção explica informações gerais sobre o serviço **dhcpd**, bem como como configurar um servidor DHCP e DHCP relay.

Se um procedimento requer diferentes passos para fornecer DHCP em redes IPv4 e IPv6, as seções deste capítulo contêm procedimentos para ambos os protocolos.

43.1. AS DIFERENÇAS AO USAR O DHCPD PARA DHCPV4 E DHCPV6

O serviço **dhcpd** suporta o fornecimento de DHCPv4 e DHCPv6 em um servidor. Entretanto, você precisa de uma instância separada de **dhcpd** com arquivos de configuração separados para fornecer DHCP para cada protocolo.

DHCPv4

- Arquivo de configuração **/etc/dhcp/dhcpd.conf**
- Nome do serviço Systemd **dhcpd**

DHCPv6

- Arquivo de configuração **/etc/dhcp/dhcpd6.conf**
- Nome do serviço Systemd **dhcpd6**

43.2. O BANCO DE DADOS DE LOCAÇÃO DO SERVIÇO DHCPD

Um aluguel DHCP é o período de tempo para o qual o serviço **dhcpd** aloca um endereço de rede a um cliente. O serviço **dhcpd** armazena os aluguéis de DHCP nos seguintes bancos de dados:

- Para DHCPv4 **/var/lib/dhcpd/dhcpd.leases**
- Para DHCPv6 **/var/lib/dhcpd/dhcpd6.leases**

ATENÇÃO

A atualização manual dos arquivos do banco de dados pode corromper os bancos de dados.

Os bancos de dados do arrendamento contêm informações sobre os arrendamentos atribuídos, tais como o endereço IP atribuído a um endereço MAC (Media Access Control) ou o carimbo de tempo quando o arrendamento expira. Note que todos os carimbos de tempo nos bancos de dados de arrendamento estão em Tempo Universal Coordenado (UTC).

O serviço **dhcpd** recria as bases de dados periodicamente:

1. O serviço renomeia os arquivos existentes:
   - **/var/lib/dhcpd/dhcpd.leases** para **/var/lib/dhcpd/dhcpd.leases~**
   - **/var/lib/dhcpd/dhcpd6.leases** para **/var/lib/dhcpd/dhcpd6.leases~**
2. O serviço escreve todos os arrendamentos conhecidos para os arquivos recém-criados **/var/lib/dhcpd/dhcpd.leases** e **/var/lib/dhcpd/dhcpd6.leases**.

Recursos adicionais

- Para mais detalhes sobre o que está armazenado no banco de dados do arrendamento, consulte a página de manual **dhcpd.leases(5)**.
- Seção 43.10, “Restaurando um banco de dados de arrendamento corrupto”

43.3. COMPARAÇÃO DO DHCPV6 COM O RADVD

Em uma rede IPv6, somente mensagens publicitárias de roteadores fornecem informações sobre um gateway padrão IPv6. Como conseqüência, se você quiser usar DHCPv6 em sub-redes que requerem uma configuração padrão de gateway, você deve configurar adicionalmente um serviço de propaganda de roteador, como o Router Advertisement Daemon (**radvd**).

O serviço **radvd** usa bandeiras em pacotes de propaganda de roteadores para anunciar a disponibilidade de um servidor DHCPv6.

Esta seção compara DHCPv6 e **radvd**, e fornece informações sobre a configuração **radvd**.

	DHCPv6	radvd
Fornece informações sobre o gateway padrão	não	sim
Garante endereços aleatórios para proteger a privacidade	sim	não
Envia outras opções de configuração de rede	sim	não
Mapas de endereços de controle de acesso de mídia (MAC) para endereços IPv6	sim	não

43.4. CONFIGURAÇÃO DO SERVIÇO RADVD PARA ROTEADORES IPV6

O daemon de propaganda do roteador (**radvd**) envia mensagens de propaganda do roteador que são necessárias para a autoconfiguração do IPv6 sem estado. Isto permite que os usuários configurem automaticamente seus endereços, configurações, rotas e escolham um roteador padrão com base nestes anúncios.

O procedimento nesta seção explica como configurar **radvd**.

Pré-requisitos

- Você está logado como usuário do **root**.

Procedimento

Procedimento

1. Instale o pacote **radvd**:

```
# yum instalar radvd
```

2. Edite o arquivo **/etc/radvd.conf**, e adicione a seguinte configuração:

```
interface enp1s0
{
  AdvSendAdvert on;
  AdvManagedFlag on;
  AdvOtherConfigFlag on;

  prefix 2001:db8:0:1::/64 {
  };
};
```

   Estas configurações configuram **radvd** para enviar mensagens de propaganda de roteador no dispositivo **enp1s0** para a sub-rede **2001:db8:0:1::/64**. A configuração **AdvManagedFlag on** define que o cliente deve receber o endereço IP de um servidor DHCP, e o parâmetro **AdvOtherConfigFlag** definido para **on** define que os clientes também devem receber informações sem endereço do servidor DHCP.

3. Opcionalmente, configure o **radvd** para iniciar automaticamente quando o sistema for inicializado:

```
# systemctl habilita radvd
```

4. Iniciar o serviço **radvd**:

```
# systemctl start radvd
```

5. Opcionalmente, exibir o conteúdo dos pacotes de publicidade do roteador e os valores configurados que **radvd** envia:

```
# radvdump
```

Recursos adicionais

- Para mais detalhes sobre a configuração **radvd**, consulte a página de manual **radvd.conf(5)**.
- Para um exemplo de configuração de **radvd**, veja o arquivo **/usr/share/doc/radvd/radvd.conf.example**.

43.5. CONFIGURAÇÃO DE INTERFACES DE REDE PARA OS SERVIDORES DHCP

Por padrão, o serviço **dhcpd** processa solicitações apenas em interfaces de rede que tenham um endereço IP na sub-rede definida no arquivo de configuração do serviço.

Por exemplo, no cenário a seguir, **dhcpd** ouve apenas na interface de rede **enp0s1**:

- Você tem apenas uma definição **subnet** para a rede 192.0.2.0/24 no arquivo **/etc/dhcp/dhcpd.conf**.
- A interface de rede **enp0s1** está conectada à sub-rede 192.0.2.0/24.
- A interface **enp7s0** está conectada a uma sub-rede diferente.

Somente siga o procedimento desta seção se o servidor DHCP contiver várias interfaces de rede conectadas à mesma rede, mas o serviço deve escutar somente em interfaces específicas.

Dependendo se você deseja fornecer DHCP para IPv4, IPv6, ou ambos os protocolos, veja o procedimento a seguir:

- Redes IPv4
- Redes IPv6

Pré-requisitos

- Você está logado como usuário do **root**.
- O pacote **dhcp-server** está instalado.

Procedimento

- Para redes IPv4:
  1. Copie o arquivo **/usr/lib/systemd/system/dhcpd.service** para o diretório **/etc/systemd/system/**:

     ```
     # cp /usr/lib/systemd/systemd/system/dhcpd.service /etc/systemd/systemd/system/
     ```

     Não edite o arquivo **/usr/lib/systemd/system/dhcpd.service**. Futuras atualizações do pacote **dhcp-server** podem anular as mudanças.
  2. Edite o arquivo **/etc/systemd/system/dhcpd.service**, e anexe os nomes da interface, que **dhcpd** deve ouvir o comando no parâmetro **ExecStart**:

     ```
     ExecStart=/usr/sbin/dhcpd -f -cf /etcc/dhcp/dhcpd.conf -user dhcpd -group dhcpd --no-pid $DHCPDARGS enp0s1 enp7s0
     ```

     Este exemplo configura que **dhcpd** ouve apenas no ***enp0s1*** e interfaces **enp7s0**.
  3. Recarregar a configuração do gerenciador **systemd**:

     ```
     # systemctl daemon-reload
     ```
  4. Reinicie o serviço **dhcpd**:

     ```
     # systemctl restart dhcpd.service
     ```
- Para redes IPv6:
  1. Copie o arquivo **/usr/lib/systemd/system/dhcpd6.service** para o diretório **/etc/systemd/system/**:

```
# cp /usr/lib/systemd/systemd/system/dhcpd6.service /etc/systemd/systemd/system/
```

Não edite o arquivo **/usr/lib/systemd/system/dhcpd6.service**. Futuras atualizações do pacote **dhcp-server** podem anular as mudanças.

2. Edite o arquivo **/etc/systemd/system/dhcpd6.service**, e anexe os nomes da interface, que **dhcpd** deve ouvir o comando no parâmetro **ExecStart**:

```
ExecStart=/usr/sbin/dhcpd -f -6 -cf /etcc/dhcp/dhcpd6.conf -user dhcpd -group dhcpd --
no-pid $DHCPDARGS enp0s1 enp7s0
```

Este exemplo configura que **dhcpd** ouve apenas no ***enp0s1*** e interfaces **enp7s0**.

3. Recarregar a configuração do gerenciador **systemd**:

```
# systemctl daemon-reload
```

4. Reinicie o serviço **dhcpd6**:

```
# systemctl restart dhcpd6.service
```

43.6. CONFIGURAÇÃO DO SERVIÇO DHCP PARA SUB-REDES DIRETAMENTE CONECTADAS AO SERVIDOR DHCP

Use o seguinte procedimento se o servidor DHCP estiver diretamente conectado à sub-rede para a qual o servidor deve responder às solicitações DHCP. Este é o caso se uma interface de rede do servidor tiver um endereço IP desta subrede atribuído.

Dependendo se você deseja fornecer DHCP para IPv4, IPv6, ou ambos os protocolos, veja o procedimento a seguir:

- Redes IPv4
- Redes IPv6

Pré-requisitos

- Você está logado como usuário do **root**.
- O pacote **dhcpd-server** está instalado.

Procedimento

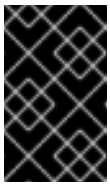
- Para redes IPv4:
  1. Edite o arquivo **/etc/dhcp/dhcpd.conf**:
     a. Opcionalmente, adicionar parâmetros globais que **dhcpd** usa como padrão se nenhuma outra diretriz contiver essas configurações:

```
option domain-name "example.com";
default-lease-time 86400;
```

Este exemplo define o nome de domínio padrão para a conexão com **example.com**, e o tempo de locação padrão para **86400** segundos (1 dia).

b. Adicione a declaração **authoritative** em uma nova linha:

```
autoritária;
```



IMPORTANTE

Sem a declaração **authoritative**, o serviço **dhcpd** não responde **DHCPREQUEST** mensagens com **DHCPNAK** se um cliente pedir um endereço que esteja fora do pool.

c. Para cada sub-rede IPv4 conectada diretamente a uma interface do servidor, acrescente uma declaração em **subnet**:

```
subnet 192.0.2.0 netmask 255.255.255.0 {
  range 192.0.2.20 192.0.2.100;
  option domain-name-servers 192.0.2.1;
  option routers 192.0.2.1;
  option broadcast-address 192.0.2.255;
  max-lease-time 172800;
}
```

Este exemplo acrescenta uma declaração de sub-rede para a rede 192.0.2.0/24. Com esta configuração, o servidor DHCP atribui as seguintes configurações a um cliente que envia uma solicitação DHCP a partir desta sub-rede:

- Um endereço IPv4 livre a partir da faixa definida no parâmetro **range**
- IP do servidor DNS para esta sub-rede **192.0.2.1**
- Porta de entrada padrão para esta sub-rede **192.0.2.1**
- Endereço de transmissão para esta sub-rede **192.0.2.255**
- O tempo máximo de locação, após o qual os clientes nesta sub-rede liberam o IP e enviam uma nova solicitação para o servidor: **172800** segundos (2 dias)

2. Opcionalmente, configure que o site **dhcpd** seja iniciado automaticamente quando o sistema for inicializado:

```
# systemctl habilita o dhcpd
```

3. Iniciar o serviço **dhcpd**:

```
# systemctl start dhcpd
```

- Para redes IPv6:

1. Edite o arquivo **/etc/dhcp/dhcpd6.conf**:

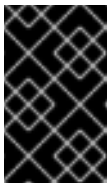
a. Opcionalmente, adicionar parâmetros globais que **dhcpd** usa como padrão se nenhuma outra diretriz contiver essas configurações:

```
option dhcp6.domain-search "example.com";
default-lease-time 86400;
```

Este exemplo define o nome de domínio padrão para a conexão com **example.com**, e o tempo de locação padrão para **86400** segundos (1 dia).

b. Adicione a declaração **authoritative** em uma nova linha:

```
autoritária;
```



IMPORTANTE

Sem a declaração **authoritative**, o serviço **dhcpd** não responde **DHCPREQUEST** mensagens com **DHCPNAK** se um cliente pedir um endereço que esteja fora do pool.

c. Para cada sub-rede IPv6 conectada diretamente a uma interface do servidor, adicione uma declaração em **subnet**:

```
subnet6 2001:db8:0:1::/64 {
  range6 2001:db8:0:1::20 2001:db8:0:1::100;
  option dhcp6.name-servers 2001:db8:0:1::1;
  max-lease-time 172800;
}
```

Este exemplo acrescenta uma declaração de sub-rede para a rede 2001:db8:0:1::/64. Com esta configuração, o servidor DHCP atribui as seguintes configurações a um cliente que envia uma solicitação DHCP a partir desta sub-rede:

- Um endereço IPv6 gratuito da faixa definida no parâmetro **range6**.
- O IP do servidor DNS para esta sub-rede é **2001:db8:0:1::1**.
- O tempo máximo de locação, após o qual os clientes nesta sub-rede liberam o IP e enviam uma nova solicitação ao servidor é de **172800** segundos (2 dias).
  Note que IPv6 requer o uso de mensagens publicitárias de roteador para identificar o gateway padrão.

2. Opcionalmente, configure que o site **dhcpd6** seja iniciado automaticamente quando o sistema for inicializado:

```
# systemctl habilita o dhcpd6
```

3. Iniciar o serviço **dhcpd6**:

```
# systemctl start dhcpd6
```

Recursos adicionais

- Para obter uma lista de todos os parâmetros que você pode definir em **/etc/dhcp/dhcpd.conf** e **/etc/dhcp/dhcpd6.conf**, consulte a página de manual **dhcp-options(5)**.

- Para mais detalhes sobre a declaração **authoritative**, consulte a seção **The authoritative statement** na página de manual **dhcpd.conf(5)**.
- Por exemplo, configurações, consulte os arquivos **/usr/share/doc/dhcp-server/dhcpd.conf.example** e **/usr/share/doc/dhcp-server/dhcpd6.conf.example**.

43.7. CONFIGURAÇÃO DO SERVIÇO DHCP PARA SUB-REDES QUE NÃO ESTÃO DIRETAMENTE CONECTADAS AO SERVIDOR DHCP

Use o seguinte procedimento se o servidor DHCP não estiver diretamente conectado à sub-rede para a qual o servidor deve responder as solicitações DHCP. Este é o caso se um agente de retransmissão DHCP encaminhar solicitações ao servidor DHCP, porque nenhuma das interfaces do servidor DHCP está diretamente conectada à subrede que o servidor deve servir.

Dependendo se você deseja fornecer DHCP para IPv4, IPv6, ou ambos os protocolos, veja o procedimento a seguir:

- Redes IPv4
- Redes IPv6

Pré-requisitos

- Você está logado como usuário do **root**.
- O pacote **dhcpd-server** está instalado.

Procedimento

- Para redes IPv4:
  1. Edite o arquivo **/etc/dhcp/dhcpd.conf**:
     a. Opcionalmente, adicionar parâmetros globais que **dhcpd** usa como padrão se nenhuma outra diretriz contiver essas configurações:

     ```
     option domain-name "example.com";
     default-lease-time 86400;
     ```

     Este exemplo define o nome de domínio padrão para a conexão com **example.com**, e o tempo de locação padrão para **86400** segundos (1 dia).

     b. Adicione a declaração **authoritative** em uma nova linha:

     ```
     autoritária;
     ```

     IMPORTANTE

     Sem a declaração **authoritative**, o serviço **dhcpd** não responde **DHCPREQUEST** mensagens com **DHCPNAK** se um cliente pedir um endereço que esteja fora do pool.

     c. Adicione uma declaração em **shared-network**, como a seguinte, para sub-redes IPv4 que não estão diretamente conectadas a uma interface do servidor:

```
shared-network example {
  option domain-name-servers 192.0.2.1;
  ...

  subnet 192.0.2.0 netmask 255.255.255.0 {
    range 192.0.2.20 192.0.2.100;
    option routers 192.0.2.1;
  }

  subnet 198.51.100.0 netmask 255.255.255.0 {
    range 198.51.100.20 198.51.100.100;
    option routers 198.51.100.1;
  }
  ...
}
```

Este exemplo acrescenta uma declaração de rede compartilhada, que contém uma declaração **subnet** tanto para as redes 192.0.2.0/24 como para as 198.51.100.0/24. Com esta configuração, o servidor DHCP atribui as seguintes configurações a um cliente que envia uma solicitação DHCP de uma destas sub-redes:

- O IP do servidor DNS para clientes de ambas as sub-redes é: **192.0.2.1**.
- Um endereço IPv4 livre da faixa definida no parâmetro **range**, dependendo de qual sub-rede o cliente enviou a solicitação.
- O gateway padrão é **192.0.2.1** ou **198.51.100.1**, dependendo de qual sub-rede o cliente enviou a solicitação.

d. Adicione uma declaração **subnet** para a sub-rede à qual o servidor está diretamente conectado e que é usada para alcançar as sub-redes remotas especificadas em **shared-network** acima:

```
subnet 203.0.113.0 netmask 255.255.255.0 {
}
```

NOTA

Se o servidor não fornecer serviço DHCP para esta sub-rede, a declaração **subnet** deve estar vazia, como mostrado no exemplo. Sem uma declaração para a subrede diretamente conectada, **dhcpd** não inicia.

2. Opcionalmente, configure que o site **dhcpd** seja iniciado automaticamente quando o sistema for inicializado:

```
# systemctl habilita o dhcpd
```

3. Iniciar o serviço **dhcpd**:

```
# systemctl start dhcpd
```

- Para redes IPv6:

1. Edite o arquivo **/etc/dhcp/dhcpd6.conf**:

i. Edite o arquivo **/etc/dhcp/dhcpd6.conf**:

a. Opcionalmente, adicionar parâmetros globais que **dhcpd** usa como padrão se nenhuma outra diretriz contiver essas configurações:

```
option dhcp6.domain-search "example.com";
default-lease-time 86400;
```

Este exemplo define o nome de domínio padrão para a conexão com **example.com**, e o tempo de locação padrão para **86400** segundos (1 dia).

b. Adicione a declaração **authoritative** em uma nova linha:

```
autoritária;
```

IMPORTANTE

Sem a declaração **authoritative**, o serviço **dhcpd** não responde **DHCPREQUEST** mensagens com **DHCPNAK** se um cliente pedir um endereço que esteja fora do pool.

c. Adicione uma declaração em **shared-network**, como a seguinte, para sub-redes IPv6 que não estão diretamente conectadas a uma interface do servidor:

```
shared-network example {
  option domain-name-servers 2001:db8:0:1::1:1
  ...

  subnet6 2001:db8:0:1::1:0/120 {
    range6 2001:db8:0:1::1:20 2001:db8:0:1::1:100
  }

  subnet6 2001:db8:0:1::2:0/120 {
    range6 2001:db8:0:1::2:20 2001:db8:0:1::2:100
  }
  ...
}
```

Este exemplo acrescenta uma declaração de rede compartilhada que contém uma declaração **subnet6** tanto para as redes 2001:db8:0:1::1:0/120 como para 2001:db8:0:1::2:0/120. Com esta configuração, o servidor DHCP atribui as seguintes configurações a um cliente que envia uma solicitação DHCP de uma destas sub-redes:

- O IP do servidor DNS para clientes de ambas as sub-redes é **2001:db8:0:1::1:1**.
- Um endereço IPv6 gratuito da faixa definida no parâmetro **range6**, dependendo de qual sub-rede o cliente enviou a solicitação.
  Note que IPv6 requer o uso de mensagens publicitárias de roteador para identificar o gateway padrão.

d. Adicione uma declaração **subnet6** para a sub-rede à qual o servidor está diretamente conectado e que é usada para alcançar as sub-redes remotas especificadas em **shared-network** acima:

```
subnet6 2001:db8:0:1::50:0/120 {
}
```

NOTA

Se o servidor não fornecer serviço DHCP para esta sub-rede, a declaração **subnet6** deve estar vazia, como mostrado no exemplo. Sem uma declaração para a subrede diretamente conectada, **dhcpd** não inicia.

2. Opcionalmente, configure que o site **dhcpd6** seja iniciado automaticamente quando o sistema for inicializado:

```
# systemctl habilita o dhcpd6
```

3. Iniciar o serviço **dhcpd6**:

```
# systemctl start dhcpd6
```

Recursos adicionais

- Para obter uma lista de todos os parâmetros que você pode definir em **/etc/dhcp/dhcpd.conf** e **/etc/dhcp/dhcpd6.conf**, consulte a página de manual **dhcp-options(5)**.
- Para mais detalhes sobre a declaração **authoritative**, consulte a seção **The authoritative statement** na página de manual **dhcpd.conf(5)**.
- Por exemplo, configurações, consulte os arquivos **/usr/share/doc/dhcp-server/dhcpd.conf.example** e **/usr/share/doc/dhcp-server/dhcpd6.conf.example**.
- Seção 43.11, “Instalação de um agente de relé DHCP”

43.8. ATRIBUIÇÃO DE UM ENDEREÇO ESTÁTICO A UM HOST USANDO DHCP

Usando uma declaração **host**, você pode configurar o servidor DHCP para atribuir um endereço IP fixo a um endereço MAC (Media Access Control) de um host. Por exemplo, use este método para sempre atribuir o mesmo endereço IP a um servidor ou dispositivo de rede.

IMPORTANTE

Se você configurar um endereço IP fixo para um endereço MAC, o endereço IP deve estar fora do grupo de endereços que você especificou nos parâmetros **fixed-address** e **fixed-address6**.

Dependendo se você deseja configurar endereços fixos para IPv4, IPv6, ou ambos os protocolos, veja o procedimento a seguir:

- Redes IPv4
- Redes IPv6

Pré-requisitos

- O serviço **dhcpd** está configurado e funcionando.
- Você está logado como usuário do **root**.

Procedimento

- Para redes IPv4:
  1. Edite o arquivo **/etc/dhcp/dhcpd.conf**:
     a. Adicione uma declaração em **host**:

```
host server.example.com {
 hardware ethernet 52:54:00:72:2f:6e;
 fixed-address 192.0.2.130;
}
```

     Este exemplo configura o servidor DHCP para sempre atribuir o endereço IP **192.0.2.130** ao host com o endereço MAC **52:54:00:72:2f:6e**.

     O serviço **dhcpd** identifica os sistemas pelo endereço MAC especificado no parâmetro **fixed-address**, e não pelo nome na declaração **host**. Como conseqüência, você pode definir este nome para qualquer string que não corresponda a outras declarações de **host**. Para configurar o mesmo sistema para múltiplas redes, use um nome diferente, caso contrário, **dhcpd** não inicia.

     b. Opcionalmente, adicione outros ajustes à declaração **host** que são específicos para este host.

  2. Reinicie o serviço **dhcpd**:

```
# systemctl start dhcpd
```

- Para redes IPv6:
  1. Edite o arquivo **/etc/dhcp/dhcpd6.conf**:
     a. Adicione uma declaração em **host**:

```
host server.example.com {
 hardware ethernet 52:54:00:72:2f:6e;
 fixed-address6 2001:db8:0:1::200;
}
```

     Este exemplo configura o servidor DHCP para sempre atribuir o endereço IP **2001:db8:0:1::20** ao host com o endereço MAC **52:54:00:72:2f:6e**.

     O serviço **dhcpd** identifica os sistemas pelo endereço MAC especificado no parâmetro **fixed-address6**, e não pelo nome na declaração **host**. Como conseqüência, você pode definir este nome para qualquer string, desde que seja exclusivo para outras declarações **host**. Para configurar o mesmo sistema para múltiplas redes, use um nome diferente porque, caso contrário, **dhcpd** não inicia.

     b. Opcionalmente, adicione outros ajustes à declaração **host** que são específicos para este host.

2. Reinicie o serviço **dhcpd6**:

```
# systemctl start dhcpd6
```

Recursos adicionais

- Para obter uma lista de todos os parâmetros que você pode definir em **/etc/dhcp/dhcpd.conf** e **/etc/dhcp/dhcpd6.conf**, consulte a página de manual **dhcp-options(5)**.
- Por exemplo, configurações, consulte os arquivos **/usr/share/doc/dhcp-server/dhcpd.conf.example** e **/usr/share/doc/dhcp-server/dhcpd6.conf.example**.

43.9. UTILIZAÇÃO DE UMA DECLARAÇÃO DE GRUPO PARA APLICAR PARÂMETROS A MÚLTIPLOS HOSTS, SUB-REDES E REDES COMPARTILHADAS AO MESMO TEMPO

Usando uma declaração em **group**, você pode aplicar os mesmos parâmetros a vários hosts, sub-redes e redes compartilhadas.

Observe que o procedimento nesta seção descreve o uso de uma declaração **group** para anfitriões, mas os passos são os mesmos para sub-redes e redes compartilhadas.

Dependendo se você deseja configurar um grupo para IPv4, IPv6, ou ambos os protocolos, veja o procedimento a seguir:

- Redes IPv4
- Redes IPv6

Pré-requisitos

- O serviço **dhcpd** está configurado e funcionando.
- Você está logado como usuário do **root**.

Procedimento

- Para redes IPv4:
  1. Edite o arquivo **/etc/dhcp/dhcpd.conf**:
     a. Adicione uma declaração em **group**:

```
group {
  option domain-name-servers 192.0.2.1;

  host server1.example.com {
    hardware ethernet 52:54:00:72:2f:6e;
    fixed-address 192.0.2.130;
  }

  host server2.example.com {
    hardware ethernet 52:54:00:1b:f3:cf;
```

```
    fixed-address 192.0.2.140;
  }
}
```

Esta definição **group** agrupa duas entradas **host**. O serviço **dhcpd** aplica o valor definido no parâmetro **option domain-name-servers** aos dois anfitriões do grupo.

b. Opcionalmente, adicione outros ajustes à declaração **group** que são específicos para estes anfitriões.

2. Reinicie o serviço **dhcpd**:

```
# systemctl start dhcpd
```

- Para redes IPv6:

1. Edite o arquivo **/etc/dhcp/dhcpd6.conf**:

a. Adicione uma declaração em **group**:

```
group {
  option dhcp6.domain-search "example.com";

  host server1.example.com {
    hardware ethernet 52:54:00:72:2f:6e;
    fixed-address 2001:db8:0:1::200;
  }

  host server2.example.com {
    hardware ethernet 52:54:00:1b:f3:cf;
    fixed-address 2001:db8:0:1::ba3;
  }
}
```

Esta definição **group** agrupa duas entradas **host**. O serviço **dhcpd** aplica o valor definido no parâmetro **option dhcp6.domain-search** aos dois anfitriões do grupo.

b. Opcionalmente, adicione outros ajustes à declaração **group** que são específicos para estes anfitriões.

2. Reinicie o serviço **dhcpd6**:

```
# systemctl start dhcpd6
```

Recursos adicionais

- Para obter uma lista de todos os parâmetros que você pode definir em **/etc/dhcp/dhcpd.conf** e **/etc/dhcp/dhcpd6.conf**, consulte a página de manual **dhcp-options(5)**.

- Por exemplo, configurações, consulte os arquivos **/usr/share/doc/dhcp-server/dhcpd.conf.example** e **/usr/share/doc/dhcp-server/dhcpd6.conf.example**.

43.10. RESTAURANDO UM BANCO DE DADOS DE ARRENDAMENTO CORRUPTO

Se o servidor DHCP registrar um erro relacionado ao banco de dados do arrendamento, como **Corrupt lease file - possible data loss!**, você pode restaurar o banco de dados do arrendamento a partir da cópia do serviço **dhcpd** criado. Note que esta cópia pode não refletir o último status do banco de dados.

ATENÇÃO

Se você remover o banco de dados do arrendamento ao invés de substituí-lo por um backup, você perde todas as informações sobre os arrendamentos atualmente atribuídos. Como conseqüência, o servidor DHCP poderia atribuir arrendamentos a clientes que tenham sido previamente atribuídos a outros anfitriões e que ainda não tenham expirado. Isto leva a conflitos de IP.

Dependendo se você deseja restaurar o DHCPv4, DHCPv6, ou ambos os bancos de dados, veja o procedimento a seguir:

- Restaurando o banco de dados de arrendamento DHCPv4
- Restaurando o banco de dados de arrendamento DHCPv6

Pré-requisitos

- Você está logado como usuário do **root**.
- O banco de dados do arrendamento é corrupto.

Procedimento

- Restaurando o banco de dados de arrendamento DHCPv4:

1. Pare o serviço **dhcpd**:

```
# sistemactl stop dhcpd
```

2. Renomear o banco de dados de arrendamento corrupto:

```
# mv /var/lib/dhcpd/dhcpd.leases /var/lib/dhcpd/dhcpd.leases.corrupt
```

3. Restaurar a cópia do banco de dados de arrendamento que o serviço **dhcp** criou quando atualizou o banco de dados de arrendamento:

```
# cp -p /var/lib/dhcpd/dhcpd.leases~ /var/lib/dhcpd/dhcpd.leases
```

IMPORTANTE

Se você tiver um backup mais recente do banco de dados do arrendamento, restaure este backup em seu lugar.

4. Iniciar o serviço **dhcpd**:

```
# systemctl start dhcpd
```

- Restaurando o banco de dados de arrendamento DHCPv6:

  1. Pare o serviço **dhcpd6**:

     ```
     # systemctl stop dhcpd6
     ```

  2. Renomear o banco de dados de arrendamento corrupto:

     ```
     # mv /var/lib/dhcpd/dhcpd6.leases /var/lib/dhcpd/dhcpd6.leases.corrupt
     ```

  3. Restaurar a cópia do banco de dados de arrendamento que o serviço **dhcp** criou quando atualizou o banco de dados de arrendamento:

     ```
     # cp -p /var/lib/dhcpd/dhcpd6.leases~ /var/lib/dhcpd/dhcpd6.leases
     ```

     IMPORTANTE

     Se você tiver um backup mais recente do banco de dados do arrendamento, restaure este backup em seu lugar.

  4. Iniciar o serviço **dhcpd6**:

     ```
     # systemctl start dhcpd6
     ```

Recursos adicionais

- Seção 43.2, "O banco de dados de locação do serviço dhcpd"

43.11. INSTALAÇÃO DE UM AGENTE DE RELÉ DHCP

O DHCP Relay Agent (**dhcrelay**) permite o retransmissão de solicitações DHCP e BOOTP de uma sub-rede sem servidor DHCP para um ou mais servidores DHCP em outras sub-redes. Quando um cliente DHCP solicita informações, o Agente de Relay DHCP encaminha a solicitação para a lista de servidores DHCP especificada. Quando um servidor DHCP retorna uma resposta, o Agente de Relay DHCP encaminha esta solicitação para o cliente.

Dependendo se você deseja configurar um relé DHCP para IPv4, IPv6, ou ambos os protocolos, veja o procedimento a seguir:

- Redes IPv4
- Redes IPv6

Pré-requisitos

- Você está logado como usuário do **root**.

Procedimento

- Para redes IPv4:

1. Instale o pacote **dhcp-relay**:

```
# yum instalar dhcp-relay
```

2. Copie o arquivo **/lib/systemd/system/dhcrelay.service** para o diretório **/etc/systemd/system/**:

```
# cp /lib/systemd/systemd/system/dhcrelay.service /etc/systemd/systemd/system/
```

Não edite o arquivo **/usr/lib/systemd/system/dhcrelay.service**. Atualizações futuras do pacote **dhcp-relay** podem anular as mudanças.

3. Editar o arquivo **/etc/systemd/system/dhcrelay.service**, e anexar o **-i *interface*** juntamente com uma lista de endereços IP dos servidores DHCPv4 responsáveis pela sub-rede:

```
ExecStart=/usr/sbin/sbin/dhcrelay -d --no-pid -i enp1s0 192.0.2.1
```

Com estes parâmetros adicionais, **dhcrelay** ouve os pedidos de DHCPv4 na interface **enp1s0** e os encaminha para o servidor DHCP com o IP **192.0.2.1**.

4. Recarregar a configuração do gerenciador **systemd**:

```
# systemctl daemon-reload
```

5. Opcionalmente, configure que o serviço **dhcrelay** seja iniciado quando o sistema inicia:

```
# systemctl habilita o dhcrelay.service
```

6. Iniciar o serviço **dhcrelay**:

```
# systemctl start dhcrelay.service
```

- Para redes IPv6:

1. Instale o pacote **dhcp-relay**:

```
# yum instalar dhcp-relay
```

2. Copie o arquivo **/lib/systemd/system/dhcrelay.service** para o diretório **/etc/systemd/system/** e nomeie o arquivo **dhcrelay6.service**:

```
# cp /lib/systemd/system/dhcrelay.service /etc/systemd/system/dhcrelay6.service
```

Não edite o arquivo **/usr/lib/systemd/system/dhcrelay.service**. Atualizações futuras do pacote **dhcp-relay** podem anular as mudanças.

3. Editar o arquivo **/etc/systemd/system/dhcrelay6.service**, e anexar o **-l *receiving_interface*** e **-u *outgoing_interface*** parâmetros:

```
ExecStart=/usr/sbin/sbin/dhcrelay -d --no-pid -l enp1s0 -u enp7s0
```

Com estes parâmetros adicionais, **dhcrelay** ouve os pedidos de DHCPv6 na interface **enp1s0** e os encaminha para a rede conectada à interface **enp7s0**.

4. Recarregar a configuração do gerenciador **systemd**:

```
# systemctl daemon-reload
```

5. Opcionalmente, configure que o serviço **dhcrelay6** seja iniciado quando o sistema inicia:

```
# systemctl habilita o dhcrelay6.service
```

6. Iniciar o serviço **dhcrelay6**:

```
# systemctl start dhcrelay6.service
```

Recursos adicionais

- Para mais detalhes sobre **dhcrelay**, consulte a página de manual **dhcrelay(8)**.

Recursos adicionais

- Seção 43.1, "As diferenças ao usar o dhcpd para DHCPv4 e DHCPv6"

CAPÍTULO 44. USANDO E CONFIGURANDO O FIREWALLD

A *firewall* é uma forma de proteger as máquinas de qualquer tráfego indesejado do exterior. Ele permite aos usuários controlar o tráfego de entrada da rede nas máquinas host, definindo um conjunto de *firewall rules*. Estas regras são usadas para ordenar o tráfego de entrada e ou bloqueá-lo ou permitir a passagem.

Note que **firewalld** com **nftables** backend não suporta a passagem das regras personalizadas **nftables** para **firewalld**, usando a opção **--direct**.

44.1. QUANDO USAR FIREWALLD, NFTABLES, OU IPTABLES

A seguir, uma breve visão geral em que cenário você deve usar uma das seguintes utilidades:

- **firewalld**: Use o utilitário **firewalld** para casos simples de uso de firewall. O utilitário é fácil de usar e cobre os casos de uso típico para estes cenários.
- **nftables**: Use o utilitário **nftables** para criar firewalls complexos e de desempenho crítico, como para toda uma rede.
- **iptables**: O utilitário **iptables** no Red Hat Enterprise Linux 8 usa a API do kernel **nf_tables** ao invés do back end **legacy**. A API **nf_tables** fornece compatibilidade retroativa para que scripts que usam os comandos **iptables** ainda funcionem no Red Hat Enterprise Linux 8. Para novos scripts de firewall, a Red Hat recomenda usar **nftables**.

IMPORTANTE

Para evitar que os diferentes serviços de firewall influenciem uns aos outros, execute apenas um deles em um host RHEL, e desabilite os outros serviços.

44.2. COMEÇANDO COM FIREWALLD

44.2.1. firewalld

firewalld é um daemon de serviço de firewall que fornece um firewall dinâmico personalizável baseado em host com uma interface **D-Bus**. Sendo dinâmico, ele permite criar, alterar e apagar as regras sem a necessidade de reiniciar o daemon de firewall cada vez que as regras são alteradas.

firewalld utiliza os conceitos de *zones* e *services*, que simplificam a gestão do tráfego. As zonas são conjuntos de regras pré-definidas. As interfaces e fontes de rede podem ser atribuídas a uma zona. O tráfego permitido depende da rede à qual seu computador está conectado e do nível de segurança que esta rede é atribuída. Os serviços de firewall são regras predefinidas que cobrem todas as configurações necessárias para permitir o tráfego de entrada para um serviço específico e se aplicam dentro de uma zona.

Os serviços utilizam um ou mais *ports* ou *addresses* para comunicação em rede. Os firewalls filtram a comunicação com base em portas. Para permitir o tráfego de rede para um serviço, suas portas devem ser *open*. **firewalld** bloqueia todo o tráfego nas portas que não estão explicitamente definidas como abertas. Algumas zonas, tais como *trusted*, permitem todo o tráfego por padrão.

Recursos adicionais

- **firewalld(1)** página do homem

44.2.2. Zonas

firewalld pode ser usado para separar as redes em diferentes zonas de acordo com o nível de confiança que o usuário decidiu colocar nas interfaces e no tráfego dentro daquela rede. Uma conexão só pode ser parte de uma zona, mas uma zona pode ser usada para muitas conexões de rede.

NetworkManager notifica **firewalld** sobre a zona de uma interface. Você pode atribuir zonas para interfaces com:

- **NetworkManager**
- ferramenta**firewall-config**
- **firewall-cmd** ferramenta de linha de comando
- O console web RHEL

Os três últimos só podem editar os arquivos de configuração **NetworkManager** apropriados. Se você mudar a zona da interface usando o console web, **firewall-cmd** ou **firewall-config**, a solicitação é encaminhada para **NetworkManager** e não é tratada por **firewalld**.

As zonas pré-definidas são armazenadas no diretório **/usr/lib/firewalld/zones/** e podem ser aplicadas instantaneamente a qualquer interface de rede disponível. Estes arquivos são copiados para o diretório **/etc/firewalld/zones/** somente após serem modificados. As configurações padrão das zonas pré-definidas são as seguintes:

block

Qualquer conexão de rede que chegue é rejeitada com uma mensagem proibida para **IPv4** e para **IPv6**. Somente conexões de rede iniciadas de dentro do sistema são possíveis.

dmz

Para computadores em sua zona desmilitarizada que são de acesso público com acesso limitado à sua rede interna. Somente conexões de entrada selecionadas são aceitas.

drop

Qualquer pacote de rede recebido é descartado sem nenhuma notificação. Somente as conexões de rede de saída são possíveis.

external

Para uso em redes externas com mascaramento habilitado, especialmente para roteadores. Você não confia nos outros computadores da rede para não danificar seu computador. Somente conexões de entrada selecionadas são aceitas.

home

Para uso em casa quando você confia principalmente nos outros computadores da rede. Somente as conexões de entrada selecionadas são aceitas.

internal

Para uso em redes internas quando você confia principalmente nos outros computadores da rede. Somente as conexões de entrada selecionadas são aceitas.

public

Para uso em áreas públicas onde você não confia em outros computadores na rede. Somente conexões de entrada selecionadas são aceitas.

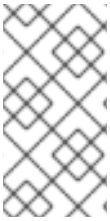
trusted

Todas as conexões de rede são aceitas.

work

Para uso no trabalho, onde você confia principalmente nos outros computadores da rede. Somente as conexões de entrada selecionadas são aceitas.

Uma dessas zonas é definida como a zona *default*. Quando as conexões de interface são adicionadas a **NetworkManager**, elas são atribuídas à zona padrão. Na instalação, a zona padrão em **firewalld** é definida como a zona **public**. A zona padrão pode ser alterada.



NOTA

Os nomes das zonas de rede devem ser auto-explicativos e permitir que os usuários tomem rapidamente uma decisão razoável. Para evitar quaisquer problemas de segurança, revisar a configuração padrão da zona e desativar quaisquer serviços desnecessários de acordo com suas necessidades e avaliações de risco.

Recursos adicionais

- **firewalld.zone(5)** página do homem

44.2.3. Serviços pré-definidos

Um serviço pode ser uma lista de portas locais, protocolos, portas de origem e destinos, bem como uma lista de módulos de ajuda de firewall carregados automaticamente se um serviço for ativado. O uso de serviços economiza tempo dos usuários porque eles podem realizar várias tarefas, tais como abrir portas, definir protocolos, permitir o envio de pacotes e mais, em uma única etapa, em vez de configurar tudo, um após o outro.

As opções de configuração de serviço e informações genéricas do arquivo estão descritas na página de manual **firewalld.service(5)**. Os serviços são especificados por meio de arquivos de configuração XML individuais, que são nomeados no formato a seguir ***service-name*.xml**. Os nomes dos protocolos são preferidos aos nomes dos serviços ou aplicativos em **firewalld**.

Os serviços podem ser adicionados e removidos usando a ferramenta gráfica **firewall-config**, **firewall-cmd**, e **firewall-offline-cmd**.

Alternativamente, você pode editar os arquivos XML no diretório **/etc/firewalld/services/**. Se um serviço não for adicionado ou alterado pelo usuário, então nenhum arquivo XML correspondente é encontrado em **/etc/firewalld/services/**. Os arquivos no diretório **/usr/lib/firewalld/services/** podem ser usados como modelos se você quiser adicionar ou alterar um serviço.

Recursos adicionais

- **firewalld.service(5)** página do homem

44.3. INSTALANDO A FERRAMENTA DE CONFIGURAÇÃO FIREWALL-CONFIG GUI

Para usar a ferramenta de configuração **firewall-config** GUI, instale o pacote **firewall-config**.

Procedimento

1. Digite o seguinte comando como **root**:

```
# yum instalar firewall-configurar
```

Alternativamente, em **GNOME, use the Super key and type `Software** para lançar o aplicativo **Software Sources**. Digite **firewall** na caixa de busca, que aparece após selecionar o botão de busca no canto superior direito. Selecione o item **Firewall** nos resultados da busca e clique no botão **Instalar**.

2. Para executar **firewall-config**, use o comando **firewall-config** ou pressione a tecla **Super** para entrar no **Activities Overview**, digite **firewall**, e pressione **Enter**.

44.4. VISUALIZANDO O STATUS ATUAL E AS CONFIGURAÇÕES DE FIREWALLD

44.4.1. Visualizando o status atual de firewalld

O serviço de firewall, **firewalld**, é instalado no sistema por padrão. Use a interface **firewalld** CLI para verificar se o serviço está sendo executado.

Procedimento

1. Para ver o status do serviço:

```
# firewall-cmd --state
```

2. Para mais informações sobre o status do serviço, use o sub-comando **systemctl status**:

```
# systemctl status firewalld
firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/usr/lib/systemd/system/firewalld.service; enabled; vendor pr
   Active: active (running) since Mon 2017-12-18 16:05:15 CET; 50min ago
     Docs: man:firewalld(1)
 Main PID: 705 (firewalld)
    Tasks: 2 (limit: 4915)
   CGroup: /system.slice/firewalld.service
           └─705 /usr/bin/python3 -Es /usr/sbin/firewalld --nofork --nopid
```

Recursos adicionais

É importante saber como **firewalld** está configurado e quais regras estão em vigor antes de tentar editar as configurações. Para exibir as configurações do firewall, veja Seção 44.4.2, “Visualizando os ajustes firewalld atuais”

44.4.2. Visualizando os ajustes firewalld atuais

44.4.2.1. Visualização de serviços permitidos usando GUI

Para visualizar a lista de serviços utilizando o gráfico **firewall-config** pressione a tecla **Super** para entrar na Visão Geral das Atividades, digite **firewall**, e pressione **Enter**. O **firewall-config** aparece a ferramenta. Agora você pode visualizar a lista de serviços na guia **Services**.

Alternativamente, para iniciar a ferramenta gráfica de configuração de firewall usando a linha de comando, digite o seguinte comando:

```
$ firewall-config
```

A janela **Firewall Configuration** se abre. Note que este comando pode ser executado como um usuário normal, mas ocasionalmente você é solicitado a obter uma senha de administrador.

44.4.2.2. Visualizando as configurações firewalld usando CLI

Com o cliente CLI, é possível obter diferentes visões das configurações atuais do firewall. A opção **--list-all** mostra uma visão completa das configurações do **firewalld**.

firewalld utiliza zonas para gerenciar o tráfego. Se uma zona não for especificada pela opção **--zone**, o comando é efetivo na zona padrão atribuída à interface de rede ativa e à conexão.

Para listar todas as informações relevantes para a zona padrão:

```
# firewall-cmd --list-all
public
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh dhcpv6-client
  ports:
  protocols:
  masquerade: no
  forward-ports:
  source-ports:
  icmp-blocks:
  rich rules:
```

Para especificar a zona para a qual devem ser exibidas as configurações, acrescente o **--zone=*zone-name*** argumento para o comando **firewall-cmd --list-all**, por exemplo:

```
# firewall-cmd --list-all --zone=home
home
  target: default
  icmp-block-inversion: no
  interfaces:
  sources:
  services: ssh mdns samba-client dhcpv6-client
... [trimmed for clarity]
```

Para ver as configurações para determinadas informações, tais como serviços ou portos, use uma opção específica. Consulte as páginas do manual **firewalld** ou obtenha uma lista das opções usando a ajuda do comando:

```
# firewall-cmd --help

Usage: firewall-cmd [OPTIONS...]

General Options
  -h, --help          Prints a short help text and exists
  -V, --version       Print the version string of firewalld
  -q, --quiet         Do not print status messages

Status Options
```

```
  --state             Return and print firewalld state
  --reload            Reload firewall and keep state information
... [trimmed for clarity]
```

Por exemplo, para ver quais serviços são permitidos na zona atual:

```
# firewall-cmd --list-services
ssh dhcpv6-client
```

NOTA

Listar as configurações para uma determinada subparte usando a ferramenta CLI pode, às vezes, ser difícil de interpretar. Por exemplo, você permite o serviço **SSH** e **firewalld** abre a porta necessária (22) para o serviço. Mais tarde, se você listar os serviços permitidos, a lista mostra o serviço **SSH**, mas se você listar as portas abertas, ela não mostra nenhuma. Portanto, recomenda-se usar a opção **--list-all** para garantir que você receba uma informação completa.

44.5. INICIANDO O FIREWALLD

Procedimento

1. Para iniciar **firewalld**, digite o seguinte comando como **root**:

```
# systemctl unmask firewalld
# systemctl start firewalld
```

2. Para garantir que **firewalld** comece automaticamente no início do sistema, digite o seguinte comando como **root**:

```
# systemctl habilita firewalld
```

44.6. PARANDO A FIREWALLD

Procedimento

1. Para parar **firewalld**, digite o seguinte comando como **root**:

```
# systemctl stop firewalld
```

2. Para evitar que o **firewalld** comece automaticamente no início do sistema:

```
# systemctl desativar firewalld
```

3. Para garantir que o firewalld não seja iniciado, acesse a interface **firewalld D-Bus** e também se outros serviços exigirem **firewalld**:

```
# Systemctl máscara firewalld
```

44.7. TEMPO DE EXECUÇÃO E AJUSTES PERMANENTES

Quaisquer mudanças comprometidas no modo *runtime* só se aplicam enquanto **firewalld** estiver em funcionamento. Quando **firewalld** é reiniciado, as configurações revertem para seus valores *permanent*.

Para tornar as mudanças persistentes através de reinicializações, aplicá-las novamente usando a opção **--permanent**. Alternativamente, para fazer alterações persistentes enquanto **firewalld** estiver em execução, use a opção **--runtime-to-permanent firewall-cmd** .

Se você definir as regras enquanto **firewalld** estiver funcionando usando apenas a opção **--permanent**, elas não se tornam efetivas antes de **firewalld** ser reiniciado. Entretanto, reiniciar **firewalld** fecha todas as portas abertas e pára o tráfego da rede.

Modificando configurações em tempo de execução e configuração permanente usando CLI

Usando o CLI, você não modifica as configurações do firewall em ambos os modos ao mesmo tempo. Você modifica apenas o tempo de execução ou o modo permanente. Para modificar as configurações do firewall no modo permanente, use a opção **--permanent** com o comando **firewall-cmd**.

```
# firewall-cmd --permanente <outras opções>
```

Sem esta opção, o comando modifica o modo de tempo de execução.

Para alterar as configurações em ambos os modos, você pode usar dois métodos:

1. Alterar as configurações de tempo de execução e depois torná-las permanentes como a seguir:

```
# firewall-cmd <other options>
# firewall-cmd --runtime-to-permanent
```

2. Definir configurações permanentes e recarregar as configurações no modo tempo de execução:

```
# firewall-cmd --permanent <other options>
# firewall-cmd --reload
```

O primeiro método permite testar as configurações antes de aplicá-las no modo permanente.

NOTA

É possível, especialmente em sistemas remotos, que uma configuração incorreta resulte em um bloqueio do usuário fora de uma máquina. Para evitar tais situações, use a opção **--timeout**. Após um determinado período de tempo, qualquer mudança reverte para seu estado anterior. O uso desta opção exclui a opção **--permanent**.

Por exemplo, para adicionar o serviço **SSH** por 15 minutos:

```
# firewall-cmd --add-service=ssh --timeout 15m
```

44.8. VERIFICAÇÃO DA CONFIGURAÇÃO FIREWALLD PERMANENTE

Em certas situações, por exemplo, após editar manualmente os arquivos de configuração **firewalld**, os administradores querem verificar se as mudanças estão corretas. Esta seção descreve como verificar a configuração permanente do serviço **firewalld**.

Pré-requisitos

- O serviço **firewalld** está funcionando.

Procedimento

1. Verificar a configuração permanente do serviço **firewalld**:

```
# firewall-cmd --check-config
success
```

Se a configuração permanente for válida, o comando retorna **success**. Em outros casos, o comando retorna um erro com mais detalhes, tais como os seguintes:

```
# firewall-cmd --check-config
Error: INVALID_PROTOCOL: 'public.xml': 'tcpx' not from {'tcp'|'udp'|'sctp'|'dccp'}
```

44.9. CONTROLE DO TRÁFEGO DA REDE USANDOFIREWALLD

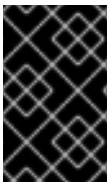
44.9.1. Desabilitação de todo o tráfego em caso de emergência usando CLI

Em uma situação de emergência, como um ataque ao sistema, é possível desativar todo o tráfego da rede e cortar o atacante.

Procedimento

1. Para desativar imediatamente o tráfego em rede, ligue o modo de pânico:

```
# firewall-cmd --panic-on
```



IMPORTANTE

A ativação do modo de pânico interrompe todo o tráfego em rede. Por este motivo, ele deve ser usado somente quando você tiver acesso físico à máquina ou se estiver logado usando um console serial.

A desativação do modo de pânico reverte o firewall para suas configurações permanentes. Para desativar o modo de pânico:

```
# firewall-cmd --panic-off
```

Para ver se o modo de pânico está ligado ou desligado, use:

```
# firewall-cmd --query-panic
```

44.9.2. Controle de tráfego com serviços pré-definidos usando CLI

O método mais simples para controlar o tráfego é adicionar um serviço pré-definido a **firewalld**. Isto abre todas as portas necessárias e modifica outras configurações de acordo com o *service definition file* .

Procedimento

1. Verifique se o serviço já não é permitido:

```
# firewall-cmd --list-services
ssh dhcpv6-client
```

2. Liste todos os serviços pré-definidos:

```
# firewall-cmd --get-services
RH-Satellite-6 amanda-client amanda-k5-client bacula bacula-client bitcoin bitcoin-rpc
bitcoin-testnet bitcoin-testnet-rpc ceph ceph-mon cfengine condor-collector ctdb dhcp dhcpv6
dhcpv6-client dns docker-registry ...
[trimmed for clarity]
```

3. Acrescente o serviço aos serviços permitidos:

```
# firewall-cmd --add-service=<service-namee>
```

4. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

44.9.3. Controle de tráfego com serviços pré-definidos usando GUI

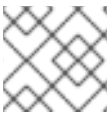
Para ativar ou desativar um serviço pré-definido ou personalizado:

1. Comece o **firewall-config** e selecionar a zona de rede cujos serviços devem ser configurados.
2. Selecione a guia **Services**.
3. Selecione a caixa de seleção para cada tipo de serviço em que você deseja confiar ou desmarque a caixa de seleção para bloquear um serviço.

Para editar um serviço:

1. Comece o **firewall-config** ferramenta.
2. Selecione **Permanent** a partir do menu etiquetado **Configuration**. Ícones adicionais e botões de menu aparecem na parte inferior da janela **Serviços**.
3. Selecione o serviço que você deseja configurar.

As guias **Ports**, **Protocols**, e **Source Port** permitem adicionar, alterar e remover portas, protocolos e porta de origem para o serviço selecionado. A aba módulos é para configurar **Netfilter** módulos auxiliares. A aba **Destination** permite limitar o tráfego a um endereço de destino específico e ao Protocolo Internet (**IPv4** ou **IPv6**).



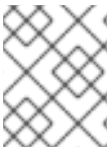
NOTA

Não é possível alterar as configurações de serviço no modo **Runtime**.

44.9.4. Adicionando novos serviços

Os serviços podem ser adicionados e removidos usando o gráfico **firewall-config** **firewall-cmd** , e **firewall-offline-cmd**. Alternativamente, você pode editar os arquivos XML em **/etc/firewalld/services/**. Se um serviço não for adicionado ou alterado pelo usuário, então nenhum arquivo XML correspondente

é encontrado em **/etc/firewalld/services/**. Os arquivos **/usr/lib/firewalld/services/** podem ser usados como modelos se você quiser adicionar ou alterar um serviço.



NOTA

Os nomes dos serviços devem ser alfanuméricos e podem, adicionalmente, incluir apenas os caracteres _ (sublinhado) e **-** (traço).

Procedimento

Para adicionar um novo serviço em um terminal, use **firewall-cmd**, ou **firewall-offline-cmd** no caso de não estar ativo **firewalld**.

1. Digite o seguinte comando para adicionar um serviço novo e vazio:

```
$ firewall-cmd --new-service=service-name --permanent
```

2. Para adicionar um novo serviço usando um arquivo local, use o seguinte comando:

```
$ firewall-cmd --new-service-from-file=service-name.xml --permanent
```

   Você pode mudar o nome do serviço com o **--name=*service-name*** opção.

3. Assim que as configurações do serviço são alteradas, uma cópia atualizada do serviço é colocada em **/etc/firewalld/services/**.
   Como **root**, você pode digitar o seguinte comando para copiar um serviço manualmente:

```
# cp /usr/lib/firewalld/services/services/service-name.xml /etc/firewalld/services/service-name.xml
```

firewalld carrega arquivos de **/usr/lib/firewalld/services** em primeiro lugar. Se os arquivos forem colocados em **/etc/firewalld/services** e forem válidos, então estes substituirão os arquivos correspondentes de **/usr/lib/firewalld/services**. Os arquivos anulados em **/usr/lib/firewalld/services** são usados assim que os arquivos correspondentes em **/etc/firewalld/services** forem removidos ou se **firewalld** tiver sido solicitado a carregar os padrões dos serviços. Isto se aplica somente ao ambiente permanente. Uma recarga é necessária para obter estas falhas também no ambiente de tempo de execução.

44.9.5. Controle de portos usando CLI

Os portos são dispositivos lógicos que permitem a um sistema operacional receber e distinguir o tráfego da rede e encaminhá-lo de acordo com os serviços do sistema. Estes são normalmente representados por um daemon que escuta no porto, ou seja, espera por qualquer tráfego que chegue a este porto.

Normalmente, os serviços de sistema escutam nos portos padrão que lhes são reservados. O daemon **httpd**, por exemplo, ouve no porto 80. Entretanto, os administradores de sistema, por padrão, configuram daemons para ouvir em diferentes portas para aumentar a segurança ou por outras razões.

44.9.5.1. Abertura de um porto

Através de portas abertas, o sistema é acessível do exterior, o que representa um risco de segurança. Geralmente, mantenha as portas fechadas e só as abra se elas forem necessárias para determinados serviços.

Procedimento

Para obter uma lista de portos abertos na zona atual:

1. Liste todos os portos permitidos:

```
# firewall-cmd --list-ports
```

2. Adicione uma porta aos portos permitidos para abri-la para o tráfego de entrada:

```
# firewall-cmd --add-port=port-number/port-type
```

3. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

Os tipos de portos são **tcp**, **udp**, **sctp**, ou **dccp**. O tipo deve corresponder ao tipo de comunicação em rede.

44.9.5.2. Fechamento de um porto

Quando uma porta aberta não for mais necessária, feche essa porta em **firewalld**. É altamente recomendado fechar todas as portas desnecessárias assim que elas não forem utilizadas, pois deixar uma porta aberta representa um risco à segurança.

Procedimento

Para fechar um porto, removê-lo da lista de portos permitidos:

1. Liste todos os portos permitidos:

```
# firewall-cmd --list-ports
[WARNING]
====
This command will only give you a list of ports that have been opened as ports. You will not be able to see any open ports that have been opened as a service. Therefore, you should consider using the --list-all option instead of --list-ports.
====
```

2. Retirar o porto dos portos permitidos para fechá-lo para o tráfego de entrada:

```
# firewall-cmd --remove-port=port-number/port-type
```

3. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

44.9.6. Abertura de portos usando GUI

Para permitir o tráfego através do firewall até uma determinada porta:

1. Comece o **firewall-config** e selecione a zona de rede cujas configurações você deseja alterar.

2. Selecione a aba **Ports** e clique no botão **Adicionar**, no lado direito. A janela **Port and Protocol** se abre.
3. Digite o número da porta ou intervalo de portas a permitir.
4. Selecione **tcp** ou **udp** a partir da lista.

44.9.7. Controle de tráfego com protocolos usando GUI

Para permitir o tráfego através do firewall usando um determinado protocolo:

1. Comece o **firewall-config** e selecione a zona de rede cujas configurações você deseja alterar.
2. Selecione a aba **Protocols** e clique no botão **Add** no lado direito. A janela **Protocol** se abre.
3. Selecione um protocolo da lista ou selecione a caixa de seleção **Other Protocol** e digite o protocolo no campo.

44.9.8. Abertura de portas de origem usando GUI

Permitir o tráfego através do firewall a partir de uma determinada porta:

1. Inicie a ferramenta de configuração de firewall e selecione a zona de rede cujas configurações você deseja alterar.
2. Selecione a aba **Source Port** e clique no botão **Add** no lado direito. A janela **Source Port** se abre.
3. Digite o número da porta ou intervalo de portas a permitir. Selecione **tcp** ou **udp** da lista.

44.10. TRABALHANDO COM ZONAS FIREWALLD

As zonas representam um conceito para gerenciar o tráfego de entrada de forma mais transparente. As zonas são conectadas a interfaces de rede ou designadas a uma gama de endereços de origem. As regras de firewall são gerenciadas independentemente para cada zona, o que permite definir configurações complexas de firewall e aplicá-las ao tráfego.

44.10.1. Listagem de zonas

Procedimento

1. Para ver quais zonas estão disponíveis em seu sistema:

```
# firewall-cmd --get-zones
```

   O comando **firewall-cmd --get-zones** exibe todas as zonas que estão disponíveis no sistema, mas não mostra nenhum detalhe para zonas específicas.

2. Para ver informações detalhadas para todas as zonas:

```
# firewall-cmd --list-all-zonas
```

3. Para ver informações detalhadas para uma zona específica:

```
# firewall-cmd --zone=zone-name --list-all
```

44.10.2. Modificação de configurações firewalld para uma determinada zona

Os sites Seção 44.9.2, "Controle de tráfego com serviços pré-definidos usando CLI" e Seção 44.9.5, "Controle de portos usando CLI" explicam como adicionar serviços ou modificar portos no escopo da zona de trabalho atual. S vezes, é necessário estabelecer regras em uma zona diferente.

Procedimento

1. Para trabalhar em uma zona diferente, use o **--zone=*zone-name*** opção. Por exemplo, para permitir o serviço **SSH** na zona *public*:

```
# firewall-cmd --add-service=ssh --zone=public
```

44.10.3. Mudando a zona padrão

Os administradores de sistema atribuem uma zona a uma interface de rede em seus arquivos de configuração. Se uma interface não for atribuída a uma zona específica, ela será atribuída à zona padrão. Após cada reinício do serviço **firewalld**, **firewalld** carrega as configurações para a zona padrão e a torna ativa.

Procedimento

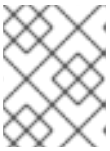
Para configurar a zona padrão:

1. Exibir a zona padrão atual:

```
# firewall-cmd --get-default-zone
```

2. Defina a nova zona padrão:

```
# firewall-cmd --set-default-zone zone-nome
```



NOTA

Seguindo este procedimento, a configuração é permanente, mesmo sem a opção **--permanent**.

44.10.4. Atribuição de uma interface de rede a uma zona

É possível definir diferentes conjuntos de regras para diferentes zonas e, em seguida, alterar as configurações rapidamente alterando a zona da interface que está sendo utilizada. Com várias interfaces, uma zona específica pode ser definida para cada uma delas para distinguir o tráfego que está passando por elas.

Procedimento

Para atribuir a zona a uma interface específica:

1. Relacione as zonas ativas e as interfaces atribuídas a elas:

```
# firewall-cmd --get-active-zones
```

2. Atribuir a interface a uma zona diferente:

```
# firewall-cmd --zone=zone_name --change-interface=interface_name --permanente
```

44.10.5. Atribuição de uma zona a uma conexão usando nmcli

Este procedimento descreve como adicionar uma zona firewalld a uma conexão NetworkManager usando o utilitário **nmcli**.

Procedimento

1. Atribuir a zona ao perfil de conexão do NetworkManager:

```
# nmcli conexão modificar profile connection.zone zone_name
```

2. Recarregue a conexão:

```
# nmcli conexão acima profile
```

44.10.6. Atribuição manual de uma zona a uma conexão de rede em um arquivo ifcfg

Quando a conexão é gerenciada por **NetworkManager**deve estar ciente de uma zona que utiliza. Para cada conexão de rede, uma zona pode ser especificada, o que proporciona a flexibilidade de várias configurações de firewall de acordo com a localização do computador com dispositivos portáteis. Assim, as zonas e configurações podem ser especificadas para diferentes locais, como empresa ou residência.

Procedimento

1. Para definir uma zona para uma conexão, edite o **/etc/sysconfig/network-scripts/ifcfg-*connection_name*** e acrescentar uma linha que atribua uma zona a esta conexão:

```
ZONA=zone_name
```

44.10.7. Criando uma nova zona

Para usar zonas personalizadas, criar uma nova zona e usá-la como uma zona pré-definida. Novas zonas requerem a opção **--permanent**, caso contrário o comando não funciona.

Procedimento

Para criar uma nova zona:

1. Criar uma nova zona:

```
# firewall-cmd --new-zone=zone-name
```

2. Verifique se a nova zona é adicionada a seus ajustes permanentes:

```
# firewall-cmd --get-zones
```

3. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

44.10.8. Arquivos de configuração de zona

Zonas também podem ser criadas usando um *zone configuration file*. Esta abordagem pode ser útil quando você precisa criar uma nova zona, mas quer reutilizar as configurações de uma zona diferente e apenas alterá-las um pouco.

Um arquivo de configuração de zona **firewalld** contém as informações para uma zona. Estas são a descrição da zona, serviços, portos, protocolos, icmp-blocks, mascarada, forward-ports e regras de linguagem rica em um formato de arquivo XML. O nome do arquivo tem que ser ***zone-name*.xml** onde o comprimento de *zone-name* é atualmente limitado a 17 caracteres. Os arquivos de configuração da zona estão localizados nos diretórios **/usr/lib/firewalld/zones/** e **/etc/firewalld/zones/**.

O exemplo a seguir mostra uma configuração que permite um serviço (**SSH**) e uma faixa de portas, tanto para os protocolos **TCP** como para **UDP**:

```
<?xml version="1.0" encoding="utf-8"?>
<zone>
  <short>My zone</short>
  <description>Here you can describe the characteristic features of the zone.</description>
  <service name="ssh"/>
  <port port="1025-65535" protocol="tcp"/>
  <port port="1025-65535" protocol="udp"/>
</zone>
```

Para alterar as configurações dessa zona, adicionar ou remover seções para adicionar portos, encaminhar portos, serviços, e assim por diante.

Recursos adicionais

- Para mais informações, consulte as páginas do manual **firewalld.zone**.

44.10.9. Utilização de metas de zona para definir o comportamento padrão para o tráfego de entrada

Para cada zona, você pode definir um comportamento padrão que trata do tráfego de entrada que não é especificado. Tal comportamento é definido pela definição do objetivo da zona. Há quatro opções - **default**, **ACCEPT**, **REJECT**, e **DROP**. Ao definir o alvo para **ACCEPT**, você aceita todos os pacotes de entrada, exceto aqueles desabilitados por uma regra específica. Se você definir a meta para **REJECT** ou **DROP**, você desabilita todos os pacotes de entrada, exceto aqueles que você permitiu em regras específicas. Quando os pacotes são rejeitados, a máquina de origem é informada sobre a rejeição, enquanto não há informação enviada quando os pacotes são descartados.

Procedimento

Estabelecer uma meta para uma zona:

1. Liste as informações para a zona específica para ver o alvo padrão:

```
$ firewall-cmd --zone=zone-name --list-all
```

2. Estabelecer uma nova meta na zona:

```
# firewall-cmd --permanent --zone=zone-name --set-target=
<default|ACCEPT|REJECT|DROP>
```

44.11. UTILIZAÇÃO DE ZONAS PARA GERENCIAR O TRÁFEGO DE ENTRADA, DEPENDENDO DE UMA FONTE

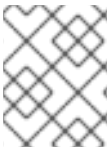
44.11.1. Utilização de zonas para gerenciar o tráfego de entrada, dependendo de uma fonte

Você pode usar zonas para gerenciar o tráfego de entrada com base em sua fonte. Isso permite classificar o tráfego de entrada e encaminhá-lo através de diferentes zonas para permitir ou não serviços que podem ser alcançados por esse tráfego.

Se você adicionar uma fonte a uma zona, a zona se torna ativa e qualquer tráfego de entrada dessa fonte será direcionado através dela. Você pode especificar configurações diferentes para cada zona, que são aplicadas ao tráfego de acordo com as fontes dadas. Você pode usar mais zonas mesmo que você tenha apenas uma interface de rede.

44.11.2. Adicionando uma fonte

Para encaminhar o tráfego de entrada para uma fonte específica, acrescente a fonte a essa zona. A fonte pode ser um endereço IP ou uma máscara IP na notação Classless Inter-domain Routing (CIDR).



NOTA

Caso você acrescente múltiplas zonas com uma faixa de rede sobreposta, elas são ordenadas alfanumericamente pelo nome da zona e somente a primeira é considerada.

- Para definir a fonte na zona atual:

```
# firewall-cmd --add-source=<source>
```

- Para definir o endereço IP de origem para uma zona específica:

```
# firewall-cmd --zone=zone-name --add-source=<source>
```

O procedimento a seguir permite todo o tráfego de entrada do site *192.168.2.15* na zona **trusted**:

Procedimento

1. Liste todas as zonas disponíveis:

```
# firewall-cmd --get-zones
```

2. Adicione a fonte IP à zona de confiança no modo permanente:

```
# firewall-cmd --zone=trusted --add-source=192.168.2.15
```

3. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

-

44.11.3. Remoção de uma fonte

A remoção de uma fonte da zona corta o tráfego proveniente da mesma.

Procedimento

1. Liste as fontes permitidas para a zona requerida:

```
# firewall-cmd --zone=zone-name --list-fontes
```

2. Remover a fonte da zona permanentemente:

```
# firewall-cmd --zone=zone-name --remove-source=<source>
```

3. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

44.11.4. Adicionando uma porta de origem

Para permitir a ordenação do tráfego com base em um porto de origem, especifique um porto de origem usando a opção **--add-source-port**. Você também pode combinar isto com a opção **--add-source** para limitar o tráfego a um determinado endereço IP ou faixa IP.

Procedimento

1. Para adicionar uma porta de origem:

```
# firewall-cmd --zone=zone-name --add-source-port=<port-name>/<tcp|udp|sctp|dccp>
```

44.11.5. Remoção de uma porta de origem

Ao remover um porto de origem, você desabilita a ordenação do tráfego com base em um porto de origem.

Procedimento

1. Para remover um porto de origem:

```
# firewall-cmd --zone=zone-name --remove-source-port=<port-name>/<tcp|udp|sctp|dccp>
```

44.11.6. Usando zonas e fontes para permitir um serviço apenas para um domínio específico

Para permitir que o tráfego de uma rede específica utilize um serviço em uma máquina, utilize zonas e fonte. O seguinte procedimento permite que o tráfego de *192.168.1.0/24* possa chegar ao serviço *HTTP* enquanto qualquer outro tráfego é bloqueado.

Procedimento

1. Liste todas as zonas disponíveis:

```
# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

2. Adicione a fonte à zona de confiança para encaminhar o tráfego proveniente da fonte através da zona:

```
# firewall-cmd --zone=trusted --add-source=192.168.1.0/24
```

3. Adicione o serviço *http* na zona de confiança:

```
# firewall-cmd --zone=trusted --add-service=http
```

4. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

5. Verifique se a zona de confiança está ativa e se o serviço é permitido nela:

```
# firewall-cmd --zone=trusted --list-all
trusted (active)
target: ACCEPT
sources: 192.168.1.0/24
services: http
```

44.11.7. Configuração do tráfego aceito por uma zona com base em um protocolo

Você pode permitir que o tráfego de entrada seja aceito por uma zona com base em um protocolo. Todo o tráfego usando o protocolo especificado é aceito por uma zona, na qual você pode aplicar outras regras e filtragem.

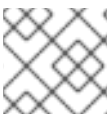
44.11.7.1. Adicionando um protocolo a uma zona

Ao acrescentar um protocolo a uma determinada zona, você permite que todo tráfego com este protocolo seja aceito por esta zona.

Procedimento

1. Para acrescentar um protocolo a uma zona:

```
# firewall-cmd --zone=zone-name --add-protocol=port-name/tcp|udp|sctp|dccp|igmp
```



NOTA

Para receber tráfego multicast, use o valor **igmp** com a opção **--add-protocol**.

44.11.7.2. Remoção de um protocolo de uma zona

Ao remover um protocolo de uma determinada zona, você deixa de aceitar todo o tráfego com base neste protocolo pela zona.

Procedimento

1. Para remover um protocolo de uma zona:

```
# firewall-cmd --zone=zone-name --remove-protocol=port-name/tcp|udp|sctp|dccp|igmp
```

44.12. CONFIGURAÇÃO DE MASCARAMENTO DE ENDEREÇOS IP

O procedimento a seguir descreve como habilitar o mascaramento de IP em seu sistema. O mascaramento de IP esconde máquinas individuais atrás de um gateway ao acessar a Internet.

Procedimento

1. Para verificar se o mascaramento IP está habilitado (por exemplo, para a zona **external**), digite o seguinte comando como **root**:

```
# firewall-cmd --zone=externo --query-masquerade
```

   O comando imprime **yes** com status de saída **0** se habilitado. Ele imprime **no** com status de saída **1** caso contrário. Se **zone** for omitido, será usada a zona padrão.

2. Para ativar o mascaramento de IP, digite o seguinte comando como **root**:

```
# firewall-cmd --zone=external --add-masquerade
```

3. Para tornar esta configuração persistente, repita o comando adicionando a opção **--permanent**.

Para desativar o mascaramento de IP, digite o seguinte comando como **root**:

```
# firewall-cmd --zone=externo --remove-masquerade --permanente
```

44.13. ENCAMINHAMENTO DE PORTAS

A redirecionamento de portas usando este método só funciona para tráfego baseado em IPv4. Para a configuração do redirecionamento IPv6, é preciso usar regras ricas.

Para redirecionar para um sistema externo, é necessário permitir o mascaramento. Para mais informações, consulte Configuração de mascaramento de endereços IP .

44.13.1. Adicionando uma porta para redirecionar

Usando **firewalld**, você pode configurar o redirecionamento de portos para que qualquer tráfego de entrada que chegue a um determinado porto em seu sistema seja entregue a outro porto interno de sua escolha ou a um porto externo em outra máquina.

Pré-requisitos

- Antes de redirecionar o tráfego de um porto para outro porto, ou outro endereço, você tem que saber três coisas: qual porta os pacotes chegam, qual protocolo é usado e onde você quer redirecioná-los.

Procedimento

Para redirecionar um porto para outro porto:

```
# firewall-cmd --add-forward-port=port=port-number:proto=tcp|udp|sctp|dccp:toport=port-number
```

Para redirecionar uma porta para outra porta em um endereço IP diferente:

1. Acrescentar o porto a ser encaminhado:

```
# firewall-cmd --add-forward-port=port=port-number:proto=tcp|udp:toport=port-number:toaddr=IP
```

2. Habilitar o mascaramento:

```
# firewall-cmd --add-masquerade
```

44.13.2. Redirecionando a porta TCP 80 para a porta 88 na mesma máquina

Siga os passos para redirecionar a porta TCP 80 para a porta 88.

Procedimento

1. Redirecionar a porta 80 para a porta 88 para tráfego TCP:

```
# firewall-cmd --add-forward-port=port=80:proto=tcp:toport=88
```

2. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

3. Verifique se o porto está redirecionado:

```
# firewall-cmd --list-all
```

44.13.3. Remoção de um porto redirecionado

Para remover um porto redirecionado:

```
# firewall-cmd --remove-forward-port=port=port-number:proto=<tcp|udp>:toport=port-number:toaddr=<IP>
```

Para remover um porto redirecionado para um endereço diferente, use o seguinte procedimento.

Procedimento

1. Retirar o porto encaminhado:

```
# firewall-cmd --remove-forward-port=port=port-number:proto=<tcp|udp>:toport=port-number:toaddr=<IP>
```

2. Desabilitar o mascaramento:

```
# firewall-cmd --remove-masquerade
```

44.13.4. Remoção da porta TCP 80 encaminhada para a porta 88 na mesma máquina

Para remover o redirecionamento do porto:

Procedimento

1. Listar portos redirecionados:

```
~]# firewall-cmd --list-forward-ports
port=80:proto=tcp:toport=88:toaddr=
```

2. Remover a porta redirecionada do firewall::

```
~]# firewall-cmd  --remove-forward-port=port=80:proto=tcp:toport=88:toaddr=
```

3. Faça com que as novas configurações sejam persistentes:

```
~]# firewall-cmd --operabilidade a permanente
```

44.14. GERENCIAMENTO DE SOLICITAÇÕES DO ICMP

O **Internet Control Message Protocol** (**ICMP**) é um protocolo de suporte que é usado por vários dispositivos de rede para enviar mensagens de erro e informações operacionais indicando um problema de conexão, por exemplo, que um serviço solicitado não está disponível. **ICMP** difere dos protocolos de transporte como TCP e UDP porque não é usado para trocar dados entre sistemas.

Infelizmente, é possível usar as mensagens **ICMP**, especialmente **echo-request** e **echo-reply**, para revelar informações sobre sua rede e usar indevidamente tais informações para vários tipos de atividades fraudulentas. Portanto, **firewalld** permite bloquear as solicitações **ICMP** para proteger as informações de sua rede.

44.14.1. Listagem e bloqueio de pedidos do ICMP

Listagem ICMP solicitações

As solicitações **ICMP** estão descritas em arquivos XML individuais que estão localizados no diretório **/usr/lib/firewalld/icmptypes/**. Você pode ler estes arquivos para ver uma descrição da solicitação. O comando **firewall-cmd** controla a manipulação das solicitações **ICMP**.

- Para listar todos os tipos disponíveis em **ICMP**:

```
# firewall-cmd --get-icmptypes
```

- A solicitação **ICMP** pode ser usada por IPv4, IPv6 ou por ambos os protocolos. Para ver para qual protocolo a solicitação **ICMP** é utilizada:

```
# firewall-cmd --info-icmptype=<icmptype>
```

- O status de uma solicitação **ICMP** mostra **yes** se a solicitação estiver atualmente bloqueada ou **no** se não estiver. Para ver se uma solicitação **ICMP** está bloqueada no momento:

```
# firewall-cmd --query-icmp-block=<icmptype>
```

Bloqueio ou desbloqueio ICMP solicitações

Quando seu servidor bloqueia solicitações do **ICMP**, ele não fornece as informações que normalmente forneceria. No entanto, isso não significa que nenhuma informação seja dada. Os clientes recebem informações de que o pedido específico **ICMP** está sendo bloqueado (rejeitado). O bloqueio das solicitações **ICMP** deve ser considerado cuidadosamente, pois pode causar problemas de comunicação, especialmente com o tráfego IPv6.

- Para ver se uma solicitação **ICMP** está atualmente bloqueada:

```
# firewall-cmd --query-icmp-block=<icmptype>
```

- Para bloquear um pedido em **ICMP**:

```
# firewall-cmd --add-icmp-block=<icmptype>
```

- Para remover o bloco para um pedido em **ICMP**:

```
# firewall-cmd --remove-icmp-block=<icmptype>
```

Bloqueio de solicitações ICMP sem fornecer qualquer tipo de informação

Normalmente, se você bloquear solicitações do **ICMP**, os clientes sabem que você está bloqueando. Portanto, um potencial atacante que está farejando endereços IP ao vivo ainda é capaz de ver que seu endereço IP está online. Para esconder completamente estas informações, você tem que descartar todas as solicitações **ICMP**.

- Para bloquear e abandonar todas as solicitações **ICMP**:

1. Defina o objetivo de sua zona para **DROP**:

```
# firewall-cmd --permanent --set-target=DROP
```

Agora, todo o tráfego, incluindo os pedidos de **ICMP**, é descartado, exceto o tráfego que você permitiu explicitamente.

- Para bloquear e abandonar certas solicitações **ICMP** e permitir outras:

1. Defina o objetivo de sua zona para **DROP**:

```
# firewall-cmd --permanent --set-target=DROP
```

2. Adicionar a inversão de bloco ICMP para bloquear todas as solicitações **ICMP** de uma só vez:

```
# firewall-cmd --add-icmp-inversion-block-inversion
```

3. Adicione o bloco ICMP para aqueles pedidos do site **ICMP** que você deseja permitir:

```
# firewall-cmd --add-icmp-block=<icmptype>
```

4. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

O *block inversion* inverte a configuração dos bloqueios de solicitações **ICMP**, de modo que todas as solicitações, que não estavam anteriormente bloqueadas, são bloqueadas por causa do alvo de suas mudanças de zona para **DROP**. As solicitações que foram bloqueadas não são bloqueadas. Isto significa que se você deseja desbloquear uma solicitação, deve usar o comando de bloqueio.

- Para reverter a inversão de bloco para um ajuste totalmente permissivo:

1. Defina a meta de sua zona para **default** ou **ACCEPT**:

```
# firewall-cmd --permanente --set-target=default
```

2. Remover todos os blocos adicionados para pedidos em **ICMP**:

```
# firewall-cmd --remove-icmp-block=<icmptype>
```

3. Remova a inversão de bloco **ICMP**:

```
# firewall-cmd --remove-icmp-block-inversion
```

4. Faça com que as novas configurações sejam persistentes:

```
# Firewall-cmd - tempo de execução a permanente
```

44.14.2. Configuração do filtro ICMP usando o GUI

- Para ativar ou desativar um filtro **ICMP**, inicie o **firewall-config** e selecionar a zona de rede cujas mensagens devem ser filtradas. Selecione a aba **ICMP Filter** e selecione a caixa de seleção para cada tipo de mensagem **ICMP** que você deseja filtrar. Desmarque a caixa de seleção para desativar um filtro. Esta configuração é por direção e o padrão permite tudo.

- Para editar um tipo **ICMP**, inicie o **firewall-config** e selecione o modo **Permanent** a partir do menu etiquetado **Configuration**. Ícones adicionais aparecem na parte inferior da janela **Serviços**. Selecione **Sim** no diálogo seguinte para habilitar o mascaramento e para fazer o encaminhamento para outra máquina funcionando.

- Para ativar a inversão do **ICMP Filter**, clique na caixa de seleção **Invert Filter**, à direita. Somente os tipos marcados com **ICMP** são agora aceitos, todos os outros são rejeitados. Em uma zona utilizando o alvo DROP, eles são descartados.

44.15. CONFIGURAÇÃO E CONTROLE DE CONJUNTOS IP USANDO FIREWALLD

Para ver a lista de tipos de conjunto IP suportados por **firewalld**, digite o seguinte comando como root.

```
~]# firewall-cmd --get-ipset-types
hash:ip hash:ip,mark hash:ip,port hash:ip,port,ip hash:ip,port,net hash:mac hash:net hash:net,iface
hash:net,net hash:net,port hash:net,port,net
```

44.15.1. Configuração das opções do conjunto IP usando CLI

Os conjuntos IP podem ser usados nas zonas **firewalld** como fontes e também como fontes em regras ricas. No Red Hat Enterprise Linux, o método preferido é usar os conjuntos de IPs criados com **firewalld** em uma regra direta.

- Para listar os conjuntos de IPs conhecidos por **firewalld** no ambiente permanente, use o seguinte comando como **root**:

```
# firewall-cmd --permanente --get-ipsets
```

- Para adicionar um novo conjunto IP, use o seguinte comando usando o ambiente permanente como **root**:

```
# firewall-cmd --permanent --new-ipset=test --type=hash:net
success
```

O comando anterior cria um novo conjunto IP com o nome *test* e o tipo **hash:net** para **IPv4**. Para criar um conjunto de IP para uso com **IPv6**, adicione a opção **--option=family=inet6**. Para tornar o novo ajuste efetivo no ambiente de tempo de execução, recarregue **firewalld**.

- Liste o novo conjunto IP com o seguinte comando: **root**:

```
# firewall-cmd --permanent --get-ipsets
test
```

- Para obter mais informações sobre o conjunto IP, use o seguinte comando como **root**:

```
# firewall-cmd --permanent --info-ipset=test
test
type: hash:net
options:
entries:
```

Observe que o conjunto IP não tem nenhuma entrada no momento.

- Para adicionar uma entrada ao conjunto IP *test*, use o seguinte comando como **root**:

```
# firewall-cmd --permanent --ipset=test --add-entry=192.168.0.1
success
```

O comando anterior adiciona o endereço IP *192.168.0.1* ao conjunto IP.

- Para obter a lista de entradas atuais no conjunto IP, use o seguinte comando como **root**:

```
# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
```

- Gerar um arquivo contendo uma lista de endereços IP, por exemplo:

```
# cat > iplist.txt <<EOL
192.168.0.2
192.168.0.3
192.168.1.0/24
192.168.2.254
EOL
```

O arquivo com a lista de endereços IP para um conjunto IP deve conter uma entrada por linha. Linhas começando com um hash, um ponto e vírgula, ou linhas vazias são ignoradas.

- Para adicionar os endereços do arquivo *iplist.txt*, use o seguinte comando como **root**:

```
# firewall-cmd --permanent --ipset=test --add-entries-from-file=iplist.txt
success
```

- Para ver a lista ampliada de entradas do conjunto IP, use o seguinte comando como **root**:

```
# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
192.168.0.2
192.168.0.3
192.168.1.0/24
192.168.2.254
```

- Para remover os endereços do conjunto IP e verificar a lista de entradas atualizada, use os seguintes comandos como **root**:

```
# firewall-cmd --permanent --ipset=test --remove-entries-from-file=iplist.txt
success
# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
```

- Você pode adicionar o conjunto IP como fonte a uma zona para lidar com todo o tráfego vindo de qualquer um dos endereços listados no conjunto IP com uma zona. Por exemplo, para adicionar o conjunto de IP *test* como fonte à zona *drop* para descartar todos os pacotes vindos de todas as entradas listadas no conjunto de IP *test*, use o seguinte comando como **root**:

```
# firewall-cmd --permanent --zone=drop --add-source=ipset:test
success
```

O prefixo **ipset:** na fonte mostra **firewalld** que a fonte é um conjunto IP e não um endereço IP ou uma faixa de endereços.

Apenas a criação e remoção de conjuntos IP é limitada ao ambiente permanente, todas as outras opções de conjuntos IP podem ser usadas também no ambiente de tempo de execução sem a opção **--permanent**.

ATENÇÃO

A Red Hat não recomenda o uso de conjuntos IP que não são gerenciados através de **firewalld**. Para utilizar tais conjuntos IP, é necessária uma regra direta permanente para referenciar o conjunto, e um serviço personalizado deve ser adicionado para criar estes conjuntos IP. Este serviço precisa ser iniciado antes do firewalld iniciar, caso contrário **firewalld** não é capaz de adicionar as regras diretas usando estes conjuntos. Você pode adicionar as regras diretas permanentes com o arquivo **/etc/firewalld/direct.xml**.

44.16. PRIORIZANDO REGRAS RICAS

Por padrão, as regras ricas são organizadas com base em sua ação de regras. Por exemplo, as regras **deny** têm precedência sobre as regras **allow**. O parâmetro **priority** nas regras ricas fornece aos administradores um controle granulado fino sobre as regras ricas e sua ordem de execução.

44.16.1. Como o parâmetro prioritário organiza as regras em diferentes cadeias

Você pode definir o parâmetro **priority** em uma regra rica para qualquer número entre **-32768** e **32767**, e valores mais baixos têm maior precedência.

O serviço **firewalld** organiza regras com base em seu valor prioritário em diferentes cadeias:

- Prioridade inferior a 0: a regra é redirecionada para uma corrente com o sufixo **_pre**.
- Prioridade maior que 0: a regra é redirecionada para uma cadeia com o sufixo **_post**.
- Prioridade igual a 0: com base na ação, a regra é redirecionada para uma cadeia com o **_log**, **_deny**, ou **_allow** a ação.

Dentro destas subdivisões, **firewalld** ordena as regras com base em seu valor prioritário.

44.16.2. Estabelecendo a prioridade de uma regra rica

O procedimento descreve um exemplo de como criar uma regra rica que usa o parâmetro **priority** para registrar todo o tráfego que não é permitido ou negado por outras regras. Você pode usar esta regra para sinalizar tráfego inesperado.

Procedimento

1. Acrescente uma regra rica com uma precedência muito baixa para registrar todo o tráfego que não tenha sido igualado por outras regras:

```
# firewall-cmd --add-rich-rule='rule priority=32767 log prefix="UNEXPECTED: "\valor limite="5/m""
```

   O comando limita adicionalmente o número de entradas de registro a **5** por minuto.

2. Opcionalmente, exibir a regra **nftables** que o comando na etapa anterior criou:

```
# nft list chain inet firewalld filter_IN_public_post
table inet firewalld {
  chain filter_IN_public_post {
    log prefix "UNEXPECTED: " limit rate 5/minute
  }
}
```

44.17. CONFIGURAÇÃO DO BLOQUEIO DO FIREWALL

Aplicações ou serviços locais são capazes de alterar a configuração do firewall se estiverem rodando como **root** (por exemplo, **libvirt**). Com este recurso, o administrador pode bloquear a configuração do firewall para que nenhuma aplicação ou apenas as aplicações que são adicionadas à lista de bloqueio

permitam solicitar mudanças no firewall. As configurações de bloqueio padrão são desabilitadas. Se ativada, o usuário pode ter certeza de que não há alterações indesejadas na configuração do firewall feitas por aplicações ou serviços locais.

44.17.1. Configuração de bloqueio usando CLI

- Para consultar se o bloqueio está ativado, use o seguinte comando como **root**:

```
# firewall-cmd --query-lockdown
```

  O comando imprime **yes** com status de saída **0** se o bloqueio estiver ativado. Ele imprime **no** com status de saída **1** caso contrário.

- Para ativar o bloqueio, digite o seguinte comando como **root**:

```
# firewall-cmd --lockdown-on
```

- Para desativar o bloqueio, use o seguinte comando como **root**:

```
# firewall-cmd --lockdown-off
```

44.17.2. Configuração das opções de listas de bloqueio usando CLI

A lista de permissão de bloqueio pode conter comandos, contextos de segurança, usuários e IDs de usuários. Se uma entrada de comando na lista de permissões terminar com um asterisco "*", então todas as linhas de comando que começam com esse comando serão iguais. Se o {\i1}"*" não estiver lá, então o comando absoluto, incluindo os argumentos, deve coincidir.

- O contexto é o contexto de segurança (SELinux) de uma aplicação ou serviço em execução. Para obter o contexto de uma aplicação ou serviço em execução, use o seguinte comando:

```
$ ps -e --context
```

  Esse comando retorna todas as aplicações em execução. Encaneie a saída através do **grep** ferramenta para obter a aplicação de interesse. Por exemplo:

```
$ ps -e --contextos | grep example_program
```

- Para listar todas as linhas de comando que estão na lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --list-lockdown-whitelist-commands
```

- Para adicionar um comando *command* à lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --add-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
```

- Para remover um comando *command* da lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --remove-lockdown-whitelist-command='/usr/bin/python3 -Es
/usr/bin/command'
```

- Para saber se o comando *command* está na lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --query-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
```

O comando imprime **yes** com status de saída **0** se for verdade. Ele imprime **no** com status de saída **1** caso contrário.

- Para listar todos os contextos de segurança que estão na lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --list-lockdown-whitelist-contexts
```

- Para adicionar um contexto *context* à lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --add-lockdown-whitelist-context=contexto
```

- Para remover um contexto *context* da lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --remove-lockdown-whitelist-context=contexto
```

- Para saber se o contexto *context* está na lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --query-lockdown-whitelist-context=contexto
```

Imprime **yes** com status de saída **0**, se verdadeiro, imprime **no** com status de saída **1** caso contrário.

- Para listar todos os IDs de usuário que estão na lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --list-lockdown-whitelist-uids
```

- Para adicionar um ID de usuário *uid* à lista de permissões, digite o seguinte comando como **root**:

```
# firewall-cmd --add-lockdown-whitelist-uid=uid
```

- Para remover um ID de usuário *uid* da lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --remove-lockdown-whitelist-uid=uid
```

- Para consultar se o ID do usuário *uid* está na lista de permissão, digite o seguinte comando:

```
$ firewall-cmd --query-lockdown-whitelist-uid=uid
```

Imprime **yes** com status de saída **0**, se verdadeiro, imprime **no** com status de saída **1** caso contrário.

- Para listar todos os nomes de usuários que estão na lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --list-lockdown-whitelist-usuários
```

- Para adicionar um nome de usuário *user* à lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --add-lockdown-whitelist-user=usuário
```

- Para remover um nome de usuário *user* da lista de permissão, digite o seguinte comando como **root**:

```
# firewall-cmd --remove-lockdown-whitelist-user=usuário
```

- Para consultar se o nome do usuário *user* está na lista de permissão, digite o seguinte comando:

```
$ firewall-cmd --query-lockdown-whitelist-user=user
```

  Imprime **yes** com status de saída **0**, se verdadeiro, imprime **no** com status de saída **1** caso contrário.

44.17.3. Configuração de opções de lista de bloqueio usando arquivos de configuração

O arquivo de configuração padrão da lista de permissão contém o contexto **NetworkManager** e o contexto padrão de **libvirt**. O ID de usuário 0 também está na lista.

```
<?xml version="1.0" encoding="utf-8"?>
 <whitelist>
   <selinux context="system_u:system_r:NetworkManager_t:s0"/>
   <selinux context="system_u:system_r:virtd_t:s0-s0:c0.c1023"/>
   <user id="0"/>
 </whitelist>
```

A seguir, um exemplo de arquivo de configuração de lista de permissão que permite todos os comandos para o utilitário **firewall-cmd**, para um usuário chamado *user* cujo ID de usuário é **815**:

```
<?xml version="1.0" encoding="utf-8"?>
 <whitelist>
   <command name="/usr/libexec/platform-python -s /bin/firewall-cmd*"/>
   <selinux context="system_u:system_r:NetworkManager_t:s0"/>
   <user id="815"/>
   <user name="user"/>
 </whitelist>
```

Este exemplo mostra tanto **user id** como **user name**, mas apenas uma opção é necessária. Python é o intérprete e está preparado para a linha de comando. Você também pode usar um comando específico, por exemplo:

```
/usr/bin/python3 /bin/firewall-cmd --lockdown-on
```

Nesse exemplo, somente o comando **--lockdown-on** é permitido.

No Red Hat Enterprise Linux, todos os utilitários são colocados no diretório **/usr/bin/** e o diretório **/bin/** está vinculado simbolicamente ao diretório **/usr/bin/**. Em outras palavras, embora o caminho para **firewall-cmd** quando inserido como **root** possa ser resolvido para **/bin/firewall-cmd**, **/usr/bin/firewall-cmd** pode agora ser usado. Todos os novos scripts devem usar o novo local. Mas esteja ciente de que se os scripts que rodam como **root** forem escritos para usar o caminho **/bin/firewall-cmd**, então esse caminho de comando deve ser adicionado na lista de permissão, além do caminho **/usr/bin/firewall-cmd** tradicionalmente usado apenas para usuários não**root**.

O * no final do atributo do nome de um comando significa que todos os comandos que começam com esta string correspondem. Se o * não estiver lá, então o comando absoluto, incluindo argumentos, deve coincidir.

44.18. LOG PARA PACOTES NEGADOS

Com a opção **LogDenied** no site **firewalld**, é possível adicionar um mecanismo simples de registro para pacotes negados. Estes são os pacotes que são rejeitados ou descartados. Para alterar a configuração do registro, edite o arquivo **/etc/firewalld/firewalld.conf** ou use a linha de comando ou a ferramenta de configuração GUI.

Se **LogDenied** estiver habilitado, as regras de registro são adicionadas logo antes das regras de rejeição e desistência nas cadeias INPUT, FORWARD e OUTPUT para as regras padrão e também as regras finais de rejeição e desistência nas zonas. Os valores possíveis para esta configuração são: **all** , **unicast**, **broadcast**, **multicast**, e **off**. A configuração padrão é **off**. Com a configuração **unicast**, **broadcast**, e **multicast**, a correspondência **pkttype** é usada para combinar com o tipo de pacote de camada de link. Com **all**, todos os pacotes são registrados.

Para listar a configuração real **LogDenied** com firewall-cmd, use o seguinte comando como **root**:

```
# firewall-cmd --get-log-denied
off
```

Para alterar a configuração **LogDenied**, use o seguinte comando como **root**:

```
# firewall-cmd --set-log-denied=all
success
```

Para alterar a configuração **LogDenied** com a ferramenta de configuração **firewalld** GUI, inicie **firewall-config**clique no menu **Options** e selecione **Change Log Denied**. A janela **LogDenied** aparece. Selecione a nova configuração **LogDenied** no menu e clique em OK.

44.19. INFORMAÇÕES RELACIONADAS

As seguintes fontes de informação fornecem recursos adicionais em relação a **firewalld**.

Documentação instalada

- **firewalld(1)** página man - descreve opções de comando para **firewalld**.
- **firewalld.conf(5)** página man - contém informações para configurar **firewalld**.
- **firewall-cmd(1)** man page - descreve opções de comando para o cliente de linha de comando **firewalld**.
- **firewall-config(1)** man page - descreve as configurações para o **firewall-config** ferramenta.

- **firewall-offline-cmd(1)** man page - descreve as opções de comando para o cliente de linha de comando offline **firewalld**.
- **firewalld.icmptype(5)** man page - descreve arquivos de configuração XML para **ICMP** filtragem.
- **firewalld.ipset(5)** man page - descreve arquivos de configuração XML para os conjuntos **firewalld IP** .
- **firewalld.service(5)** man page - descreve arquivos de configuração XML para **firewalld service**.
- **firewalld.zone(5)** man page - descreve os arquivos de configuração XML para a configuração da zona **firewalld**.
- **firewalld.direct(5)** página man - descreve o arquivo de configuração da interface direta **firewalld**.
- **firewalld.lockdown-whitelist(5)** man page - descreve o arquivo de configuração da lista **firewalld** lockdown allowlist.
- **firewalld.richlanguage(5)** man page - descreve a sintaxe da regra da linguagem rica **firewalld**.
- **firewalld.zones(5)** man page - descrição geral do que são zonas e como configurá-las.
- **firewalld.dbus(5)** página man - descreve a interface **D-Bus** de **firewalld**.

Documentação on-line

- http://www.firewalld.org/ - **firewalld** página inicial.

CAPÍTULO 45. COMEÇANDO COM NFTABLES

A estrutura **nftables** oferece facilidades de classificação de pacotes e é o sucessor designado para as ferramentas **iptables**, **ip6tables**, **arptables**, e **ebtables**. Ela oferece inúmeras melhorias em conveniência, características e desempenho em relação às ferramentas de filtragem de pacotes anteriores, mais notadamente:

- tabelas de pesquisa em vez de processamento linear
- uma estrutura única para ambos os protocolos **IPv4** e **IPv6**
- regras todas aplicadas atomicamente em vez de buscar, atualizar e armazenar um conjunto completo de regras
- suporte para depuração e rastreamento no conjunto de regras (**nftrace**) e monitoramento de eventos de rastreamento (na ferramenta **nft**)
- sintaxe mais consistente e compacta, sem extensões específicas de protocolo
- uma API Netlink para aplicações de terceiros

Da mesma forma que **iptables**, **nftables** utiliza tabelas para o armazenamento de correntes. As cadeias contêm regras individuais para a realização de ações. A ferramenta **nft** substitui todas as ferramentas das estruturas anteriores de filtragem de pacotes. A biblioteca **libnftnl** pode ser usada para interação de baixo nível com **nftables** Netlink API sobre a biblioteca **libmnl**.

O efeito dos módulos sobre o conjunto de regras **nftables** pode ser observado usando o comando **nft** list rule set. Como estas ferramentas adicionam tabelas, correntes, regras, conjuntos e outros objetos ao conjunto de regras **nftables**, esteja ciente de que **nftables** operações do conjunto de regras, como o comando **nft flush ruleset**, podem afetar os conjuntos de regras instalados usando os comandos herdados anteriormente separados.

45.1. MIGRANDO DE IPTABLES PARA NFTABLES

Se você atualizou seu servidor para o RHEL 8 ou sua configuração de firewall ainda usa as regras **iptables**, você pode migrar suas regras **iptables** para **nftables**.

45.1.1. Quando usar firewalld, nftables, ou iptables

A seguir, uma breve visão geral em que cenário você deve usar uma das seguintes utilidades:

- **firewalld**: Use o utilitário **firewalld** para casos simples de uso de firewall. O utilitário é fácil de usar e cobre os casos de uso típico para estes cenários.
- **nftables**: Use o utilitário **nftables** para criar firewalls complexos e de desempenho crítico, como para toda uma rede.
- **iptables**: O utilitário **iptables** no Red Hat Enterprise Linux 8 usa a API do kernel **nf_tables** ao invés do back end **legacy**. A API **nf_tables** fornece compatibilidade retroativa para que scripts que usam os comandos **iptables** ainda funcionem no Red Hat Enterprise Linux 8. Para novos scripts de firewall, a Red Hat recomenda usar **nftables**.

IMPORTANTE

Para evitar que os diferentes serviços de firewall influenciem uns aos outros, execute apenas um deles em um host RHEL, e desabilite os outros serviços.

45.1.2. Conversão de regras iptables em regras nftables

O Red Hat Enterprise Linux 8 fornece as ferramentas **iptables-translate** e **ip6tables-translate** para converter as regras existentes **iptables** ou **ip6tables** em regras equivalentes para **nftables**.

Observe que algumas extensões carecem de suporte de tradução. Se tal extensão existir, a ferramenta imprime a regra não traduzida prefixada com o sinal **#**. Por exemplo:

```
# iptables-translate -A INPUT -j CHECKSUM --checksum-fill
nft # -A INPUT -j CHECKSUM --checksum-fill
```

Além disso, os usuários podem usar as ferramentas **iptables-restore-translate** e **ip6tables-restore-translate** para traduzir um lixão de regras. Note que antes disso, os usuários podem usar os comandos **iptables-save** ou **ip6tables-save** para imprimir um dump das regras atuais. Por exemplo:

```
# iptables-save >/tmp/iptables.dump
# iptables-restore-translate -f /tmp/iptables.dump

# Translated by iptables-restore-translate v1.8.0 on Wed Oct 17 17:00:13 2018
add table ip nat
...
```

Para mais informações e uma lista de opções e valores possíveis, digite o comando **iptables-translate --help**.

45.2. ESCREVER E EXECUTAR SCRIPTS NFTABLES

A estrutura **nftables** fornece um ambiente de script nativo que traz um grande benefício sobre o uso de scripts shell para manter as regras de firewall: a execução de scripts é atômica. Isto significa que o sistema ou aplica o script inteiro ou impede a execução se ocorrer um erro. Isto garante que o firewall esteja sempre em um estado consistente.

Além disso, o ambiente de script **nftables** permite que os administradores o façam:

- adicionar comentários
- definir variáveis
- incluir outros arquivos do conjunto de regras

Esta seção explica como utilizar estes recursos, assim como a criação e execução de scripts **nftables**.

Quando você instala o pacote **nftables**, o Red Hat Enterprise Linux cria automaticamente ***.nft** scripts no diretório **/etc/nftables/**. Estes scripts contêm comandos que criam tabelas e cadeias vazias para diferentes propósitos. Você pode estender estes arquivos ou escrever seus scripts.

45.2.1. O cabeçalho do script necessário em nftables script

Semelhante a outros scripts, **nftables** scripts requerem uma seqüência de shebang na primeira linha do script que define a diretiva do intérprete.

Um script **nftables** deve sempre começar com a seguinte linha:

```
#!/usr/sbin/nft -f
```

IMPORTANTE

Se você omitir o parâmetro **-f**, o utilitário **nft** não lê o script e exibe **Error: syntax error, unexpected newline, expecting string**.

45.2.2. Formatos de scripts nftables suportados

O ambiente **nftables** suporta scripts nos seguintes formatos:

- Você pode escrever um script no mesmo formato que o comando **nft list ruleset** exibe o conjunto de regras:

```
#!/usr/sbin/nft -f

# Flush the rule set
flush ruleset

table inet example_table {
  chain example_chain {
    # Chain for incoming packets that drops all packets that
    # are not explicitly allowed by any rule in this chain
    type filter hook input priority 0; policy drop;

    # Accept connections to port 22 (ssh)
    tcp dport ssh accept
  }
}
```

- Você pode usar a mesma sintaxe para comandos como em **nft** comandos:

```
#!/usr/sbin/nft -f

# Flush the rule set
flush ruleset

# Create a table
add table inet example_table

# Create a chain for incoming packets that drops all packets
# that are not explicitly allowed by any rule in this chain
add chain inet example_table example_chain { type filter hook input priority 0 ; policy drop ; }

# Add a rule that accepts connections to port 22 (ssh)
add rule inet example_table example_chain tcp dport ssh accept
```

45.2.3. Executando nftables scripts

Para executar um script **nftables**, o script deve ser executável. Somente se o script for incluído em outro script, ele não precisa ser executável. O procedimento descreve como tornar um script executável e executar o script.

Pré-requisitos

- O procedimento desta seção pressupõe que você tenha armazenado um script **nftables** no arquivo **/etc/nftables/example_firewall.nft**.

Procedimento

1. Passos que são necessários apenas uma vez:
   a. Opcionalmente, defina o dono do roteiro para **root**:

```
# raiz de enxada /etc/nftables/example_firewall.nft
```

   b. Tornar o roteiro executável para o proprietário:

```
# chmod u x /etc/nftables/example_firewall.nft
```

2. Execute o roteiro:

```
# /etc/nftables/example_firewall.nft
```

   Se nenhuma saída for exibida, o sistema executou o script com sucesso.

IMPORTANTE

Mesmo se **nft** executar o script com sucesso, regras colocadas incorretamente, parâmetros ausentes ou outros problemas no script podem causar que o firewall não se comporte como esperado.

Recursos adicionais

- Para detalhes sobre como definir o proprietário de um arquivo, consulte a página de manual **chown(1)**.
- Para detalhes sobre a definição de permissões de um arquivo, consulte a página de manual **chmod(1)**.
- Seção 45.2.7, "Carregamento automático das regras nftables quando o sistema inicia"

45.2.4. Usando comentários em scripts nftables

O ambiente **nftables** interpreta tudo à direita de um personagem **#** como um comentário.

Exemplo 45.1. Comentários em um roteiro nftables

Os comentários podem começar no início de uma linha, assim como ao lado de um comando:

```
...
# Flush the rule set
flush ruleset

add table inet example_table  # Create a table
...
```

–

45.2.5. Usando variáveis em um script nftables

Para definir uma variável em um script **nftables**, use a palavra-chave **define**. Você pode armazenar valores individuais e conjuntos anônimos em uma variável. Para cenários mais complexos, use conjuntos ou mapas de veredictos.

Variáveis com um único valor
O exemplo a seguir define uma variável chamada **INET_DEV** com o valor **enp1s0**:

```
define INET_DEV = enp1s0
```

Você pode usar a variável no script escrevendo o sinal **$** seguido do nome da variável:

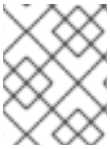
```
...
add rule inet example_table example_chain iifname $INET_DEV tcp dport ssh accept
...
```

Variáveis que contêm um conjunto anônimo
O exemplo a seguir define uma variável que contém um conjunto anônimo:

```
define DNS_SERVERS = { 192.0.2.1, 192.0.2.2 }
```

Você pode usar a variável no script escrevendo o sinal **$** seguido do nome da variável:

```
adicionar exemplo de regra inet exemplo de tabela_chain ip daddr $DNS_SERVERS aceitar
```



NOTA

Observe que os suportes encaracolados têm uma semântica especial quando você os usa em uma regra, pois indicam que a variável representa um conjunto.

Recursos adicionais

- Para detalhes sobre os conjuntos, ver Seção 45.5, “Usando conjuntos em comandos nftables” .
- Para detalhes sobre os mapas de veredictos, veja Seção 45.6, “Usando mapas de veredictos em comandos nftables”.

45.2.6. Incluindo arquivos em um script nftables

O ambiente **nftables** permite que os administradores incluam outros scripts usando a declaração **include**.

Se você especificar apenas um nome de arquivo sem um caminho absoluto ou relativo, **nftables** inclui arquivos do caminho de busca padrão, que está definido para **/etc** no Red Hat Enterprise Linux.

Exemplo 45.2. Incluindo arquivos do diretório de busca padrão

Para incluir um arquivo do diretório de busca padrão:

```
incluem "exemplo.nft"
```

Exemplo 45.3. Incluindo todos os arquivos *.nft de um diretório

Para incluir todos os arquivos que terminam em ***.nft** que estão armazenados no diretório **/etc/nftables/rulesets/**:

```
incluem "/etc/nftables/rulesets/*.nft\"
```

Observe que a declaração **include** não corresponde a arquivos que começam com um ponto.

Recursos adicionais

- Para mais detalhes, consulte a seção **Include files** na página de manual **nft(8)**.

45.2.7. Carregamento automático das regras nftables quando o sistema inicia

O serviço **nftables** systemd carrega scripts de firewall que estão incluídos no arquivo **/etc/sysconfig/nftables.conf**. Esta seção explica como carregar as regras de firewall quando o sistema inicia.

Pré-requisitos

- Os scripts **nftables** são armazenados no diretório **/etc/nftables/**.

Procedimento

1. Edite o arquivo **/etc/sysconfig/nftables.conf**.
   - Se você melhorar ***.nft** scripts criados em **/etc/nftables/** ao instalar o pacote **nftables**, descomente a declaração **include** para estes scripts.
   - Se você escrever scripts a partir do zero, adicione declarações em **include** para incluir estes scripts. Por exemplo, para carregar o **/etc/nftables/*example*.nft** quando o serviço **nftables** for iniciado, acrescente:

     ```
     incluem "/etc/nftables/example.nft\"
     ```

2. Habilite o serviço **nftables**.

   ```
   # systemctl habilita nftables
   ```

3. Opcionalmente, inicie o serviço **nftables** para carregar as regras de firewall sem reiniciar o sistema:

   ```
   # systemctl start nftables
   ```

Recursos adicionais

- Seção 45.2.2, “Formatos de scripts nftables suportados”

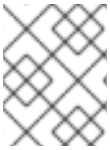
45.3. CRIAÇÃO E GERENCIAMENTO DE TABELAS, CORRENTES E REGRAS NFTABLES

Esta seção explica como exibir os conjuntos de regras **nftables**, e como gerenciá-los.

45.3.1. Valores padrão de prioridade da cadeia e nomes textuais

Quando você cria uma cadeia, o **priority** pode definir um valor inteiro ou um nome padrão que especifica a ordem na qual as cadeias com o mesmo valor **hook** atravessam.

Os nomes e valores são definidos com base em quais prioridades são utilizados pelo **xtables** ao registrar suas cadeias padrão.



NOTA

O comando **nft list chains** exibe valores de prioridade textual por padrão. Você pode visualizar o valor numérico passando a opção **-y** para o comando.

Exemplo 45.4. Usando um valor textual para definir a prioridade

O seguinte comando cria uma cadeia chamada **example_chain** em **example_table** usando o valor de prioridade padrão **50**:

 # nft add chain inet *example_table example_chain* { type *filter* hook *input* priority *50* \; policy accept \; }

Como a prioridade é um valor padrão, você pode, alternativamente, usar o valor textual:

 # nft add chain inet *example_table example_chain* { type *filter* hook *input* priority *security* \; policy accept \; }

Tabela 45.1. Nomes de prioridade padrão, família e matriz de compatibilidade de ganchos

Nome	Valor	Famílias	Anzóis
raw	-300	**ip**, **ip6**, **inet**	todos
mangle	-150	**ip**, **ip6**, **inet**	todos
dstnat	-100	**ip**, **ip6**, **inet**	pré-encaminhamento
filter	0	**ip**, **ip6**, **inet**, **arp**, **netdev**	todos
security	50	**ip**, **ip6**, **inet**	todos
srcnat	100	**ip**, **ip6**, **inet**	pós-transplante

Todas as famílias utilizam os mesmos valores, mas a família **bridge** utiliza os seguintes valores:

Tabela 45.2. Nomes de prioridade padrão, e compatibilidade de ganchos para a família bridge

Nome	Valor	Anzóis
dstnat	-300	pré-encaminhamento
filter	-200	todos
out	100	saída
srcnat	300	pós-transplante

Recursos adicionais

- Para obter detalhes sobre outras ações que você pode executar em cadeias, consulte a seção **Chains** na página de manual **nft(8)**.

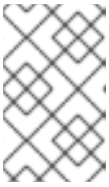
45.3.2. Exibição de conjuntos de regras nftables

Os conjuntos de regras do **nftables** contêm tabelas, correntes e regras. Esta seção explica como exibir esses conjuntos de regras.

Procedimento

- Para exibir todos os conjuntos de regras, entre:

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport http accept
    tcp dport ssh accept
  }
}
```



NOTA

Por padrão, **nftables** não pré-cria tabelas. Como conseqüência, exibindo a regra definida em um host sem nenhuma tabela, o comando **nft list ruleset** não mostra nenhuma saída.

45.3.3. Criando uma tabela nftables

Uma tabela em **nftables** é um espaço de nomes que contém uma coleção de correntes, regras, conjuntos e outros objetos. Esta seção explica como criar uma tabela.

Cada tabela deve ter uma família de endereços definida. A família de endereços de uma tabela define que tipos de endereços a tabela processa. Você pode definir uma das seguintes famílias de endereços ao criar uma tabela:

- **ip**: Combina somente pacotes IPv4. Este é o padrão se você não especificar uma família de endereços.
- **ip6**: Combina apenas pacotes IPv6.

- **inet**: Combina pacotes IPv4 e IPv6.
- **arp**: Corresponde aos pacotes do protocolo de resolução de endereços IPv4 (ARP).
- **bridge**: Combina pacotes que atravessam um dispositivo de ponte.
- **netdev**: Combina pacotes de entrada.

Procedimento

1. Use o comando **nft add table** para criar uma nova tabela. Por exemplo, para criar uma tabela chamada **example_table** que processa pacotes IPv4 e IPv6:

```
# nft adicionar tabela inet exemplo_tabela
```

2. Opcionalmente, liste todas as tabelas do conjunto de regras:

```
# nft list tables
table inet example_table
```

Recursos adicionais

- Para mais detalhes sobre famílias de endereços, consulte a seção **Address families** na página de manual **nft(8)**.
- Para detalhes sobre outras ações que você pode executar em tabelas, consulte a seção **Tables** na página de manual **nft(8)**.

45.3.4. Criando uma cadeia nftables

As correntes são recipientes para regras. Existem os dois tipos de regras a seguir:

- Cadeia base: Você pode usar cadeias de base como um ponto de entrada para pacotes da pilha de rede.
- Corrente regular: Você pode usar correntes regulares como um alvo **jump** e para organizar melhor as regras.

O procedimento descreve como adicionar uma cadeia de base a uma tabela existente.

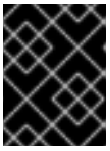
Pré-requisitos

- A tabela à qual se deseja acrescentar a nova cadeia existe.

Procedimento

1. Use o comando **nft add chain** para criar uma nova cadeia. Por exemplo, para criar uma cadeia chamada **example_chain** em **example_table**:

```
# nft add chain inet example_table example_chain { type filter hook input priority 0 {\i1}; policy accept {\i1}
```



IMPORTANTE

Para evitar que a casca interprete os ponto-e-vírgula como o fim do comando, você deve escapar dos pontos-e-vírgula com uma barra invertida.

Esta corrente filtra os pacotes de entrada. O parâmetro **priority** especifica a ordem na qual **nftables** processa cadeias com o mesmo valor de gancho. Um valor de prioridade mais baixo tem precedência sobre os mais altos. O parâmetro **policy** define a ação padrão para as regras nesta cadeia. Observe que se você estiver conectado remotamente ao servidor e definir a política padrão para **drop**, você será desconectado imediatamente se nenhuma outra regra permitir o acesso remoto.

2. Opcionalmente, exibir todas as correntes:

```
# nft list chains
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
  }
}
```

Recursos adicionais

- Para mais detalhes sobre famílias de endereços, consulte a seção **Address families** na página de manual **nft(8)**.
- Para obter detalhes sobre outras ações que você pode executar em cadeias, consulte a seção **Chains** na página de manual **nft(8)**.

45.3.5. Adicionando uma regra a uma cadeia de nftables

Esta seção explica como adicionar uma regra a uma cadeia **nftables** existente. Por padrão, o comando **nftables add rule** acrescenta uma nova regra ao final da cadeia.

Se você quiser inserir uma regra no início da cadeia, veja Seção 45.3.6, “Inserindo uma regra em uma cadeia de nftables”.

Pré-requisitos

- A cadeia à qual se deseja acrescentar a regra existe.

Procedimento

1. Para adicionar uma nova regra, use o comando **nft add rule**. Por exemplo, para adicionar uma regra ao **example_chain** no **example_table** que permite o tráfego TCP na porta 22:

```
# nft adicionar regra inet example_table example_chain tcp dport 22 accept
```

Em vez do número da porta, você pode, alternativamente, especificar o nome do serviço. No exemplo, você poderia usar **ssh** em vez do número da porta **22**. Observe que um nome de serviço é resolvido para um número de porta com base em sua entrada no arquivo **/etc/services**.

2. Opcionalmente, exibir todas as correntes e suas regras em **example_table**:

```
# nft list table inet example_table
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    ...
    tcp dport ssh accept
  }
}
```

Recursos adicionais

- Para mais detalhes sobre famílias de endereços, consulte a seção **Address families** na página de manual **nft(8)**.
- Para detalhes sobre outras ações que você pode executar sobre regras, consulte a seção **Rules** na página de manual **nft(8)**.

45.3.6. Inserindo uma regra em uma cadeia de nftables

Esta seção explica como inserir uma regra no início de uma cadeia existente **nftables** usando o comando **nftables insert rule**. Se você quiser, ao invés disso, adicionar uma regra ao final de uma cadeia, veja Seção 45.3.5, “Adicionando uma regra a uma cadeia de nftables” .

Pré-requisitos

- A cadeia à qual se deseja acrescentar a regra existe.

Procedimento

1. Para inserir uma nova regra, use o comando **nft insert rule**. Por exemplo, para inserir uma regra no **example_chain** no **example_table** que permite o tráfego TCP na porta 22:

   ```
   # nft inserir regra inet example_table example_chain tcp dport 22 accept
   ```

   Você pode, alternativamente, especificar o nome do serviço em vez do número da porta. No exemplo, você poderia usar **ssh** ao invés do número da porta **22**. Observe que um nome de serviço é resolvido para um número de porta com base em sua entrada no arquivo **/etc/services**.

2. Opcionalmente, exibir todas as correntes e suas regras em **example_table**:

   ```
   # nft list table inet example_table
   table inet example_table {
     chain example_chain {
       type filter hook input priority filter; policy accept;
       tcp dport ssh accept
       ...
     }
   }
   ```

Recursos adicionais

- Para mais detalhes sobre famílias de endereços, consulte a seção **Address families** na página de manual **nft(8)**.

- Para detalhes sobre outras ações que você pode executar sobre regras, consulte a seção **Rules** na página de manual **nft(8)**.

45.4. CONFIGURAÇÃO DE NAT USANDO NFTABLES

Com **nftables**, você pode configurar os seguintes tipos de tradução de endereços de rede (NAT):

- Mascarading
- Fonte NAT (SNAT)
- Destino NAT (DNAT)

45.4.1. Os diferentes tipos de NAT: mascaramento, NAT de origem e NAT de destino

Estes são os diferentes tipos de tradução de endereços de rede (NAT):

Mascaramento e fonte NAT (SNAT)

Use um desses tipos de NAT para alterar o endereço IP de origem dos pacotes. Por exemplo, os provedores de Internet não roteiam faixas IP reservadas, tais como **10.0.0.0/8**. Se você utiliza faixas de IP reservadas em sua rede e os usuários devem ser capazes de alcançar servidores na Internet, mapeie o endereço IP de origem dos pacotes a partir dessas faixas para um endereço IP público. Tanto o mascaramento quanto o SNAT são muito semelhantes. As diferenças são:

- O mascaramento utiliza automaticamente o endereço IP da interface de saída. Portanto, use o mascarading se a interface de saída usar um endereço IP dinâmico.
- SNAT define o endereço IP de origem dos pacotes para um IP especificado e não procura dinamicamente o IP da interface de saída. Portanto, o SNAT é mais rápido que o mascaramento. Use o SNAT se a interface de saída usar um endereço IP fixo.

Destino NAT (DNAT)

Use este tipo de NAT para encaminhar o tráfego de entrada para um host diferente. Por exemplo, se seu servidor web usa um endereço IP de uma faixa IP reservada e, portanto, não é diretamente acessível da Internet, você pode definir uma regra DNAT no roteador para redirecionar o tráfego de entrada para este servidor.

45.4.2. Configuração de mascaramento usando nftables

O mascaramento permite que um roteador altere dinamicamente o IP de origem dos pacotes enviados através de uma interface para o endereço IP da interface. Isto significa que se a interface recebe um novo IP atribuído, **nftables** usa automaticamente o novo IP ao substituir o IP de origem.

O procedimento a seguir descreve como substituir o IP de origem dos pacotes que saem do host através da interface **ens3** para o conjunto IP em **ens3**.

Procedimento

1. Criar uma mesa:

```
# nft adicionar tabela nat
```

2. Acrescente as cadeias **prerouting** e **postrouting** à tabela:

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

IMPORTANTE

Mesmo que você não acrescente uma regra à cadeia **prerouting**, a estrutura **nftables** exige que esta cadeia corresponda às respostas dos pacotes recebidos.

Observe que você deve passar a opção **--** para o comando **nft** para evitar que o shell interprete o valor de prioridade negativa como uma opção do comando **nft**.

3. Adicione uma regra à cadeia **postrouting** que combine com os pacotes de saída na interface **ens3**:

```
# nft add rule nat postrouting oifname"ens3" mascarada
```

45.4.3. Configuração da fonte NAT usando nftables

Em um roteador, Source NAT (SNAT) permite alterar o IP dos pacotes enviados através de uma interface para um endereço IP específico.

O procedimento a seguir descreve como substituir o IP de origem dos pacotes que deixam o roteador através da interface **ens3** para **192.0.2.1**.

Procedimento

1. Criar uma mesa:

```
# nft adicionar tabela nat
```

2. Acrescente as cadeias **prerouting** e **postrouting** à tabela:

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

IMPORTANTE

Mesmo que você não acrescente uma regra à cadeia **postrouting**, a estrutura **nftables** exige que esta cadeia combine as respostas dos pacotes de saída.

Observe que você deve passar a opção **--** para o comando **nft** para evitar que o shell interprete o valor de prioridade negativa como uma opção do comando **nft**.

3. Adicione uma regra à cadeia **postrouting** que substitui o IP de origem dos pacotes de saída através de **ens3** por **192.0.2.1**:

```
# nft add rule nat postrouting oifname"ens3" snat to 192.0.2.1
```

Recursos adicionais

- Seção 45.7.2, “Encaminhamento de pacotes de entrada em uma porta local específica para um host diferente”

45.4.4. Configuração do NAT de destino usando nftables

O NAT de destino permite redirecionar o tráfego em um roteador para um host que não é diretamente acessível a partir da Internet.

O procedimento a seguir descreve como redirecionar o tráfego de entrada enviado para a porta **80** e **443** do roteador para o host com o endereço IP **192.0.2.1**.

Procedimento

1. Criar uma mesa:

```
# nft adicionar tabela nat
```

2. Acrescente as cadeias **prerouting** e **postrouting** à tabela:

```
# nft -- add chain nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain nat postrouting { type nat hook postrouting priority 100 \; }
```

IMPORTANTE

Mesmo que você não acrescente uma regra à cadeia **postrouting**, a estrutura **nftables** exige que esta cadeia combine as respostas dos pacotes de saída.

Observe que você deve passar a opção **--** para o comando **nft** para evitar que o shell interprete o valor de prioridade negativa como uma opção do comando **nft**.

3. Adicione uma regra à cadeia **prerouting** que redireciona o tráfego de entrada na interface **ens3** enviada para a porta **80** e **443** para o host com o IP **192.0.2.1**:

```
# nft adicionar regra nat prerouting iifname ens3 tcp dport { 80, 443 } dnat a 192.0.2.1
```

4. Dependendo de seu ambiente, adicione uma regra SNAT ou mascarada para alterar o endereço de origem:

   a. Se a interface **ens3** utilizava endereços IP dinâmicos, acrescente uma regra de mascaramento:

```
# nft adicionar regra nat postrouting oifname {\i1}"ens3} mascarada
```

   b. Se a interface **ens3** usa um endereço IP estático, adicione uma regra SNAT. Por exemplo, se o **ens3** usa o endereço IP **198.51.100.1**:

```
nft adicionar a regra nat postrouting oifname {\i1}"ens3} snat a 198.51.100.1
```

Recursos adicionais

- Seção 45.4.1, “Os diferentes tipos de NAT: mascaramento, NAT de origem e NAT de destino”

45.5. USANDO CONJUNTOS EM COMANDOS NFTABLES

A estrutura **nftables** suporta nativamente conjuntos. Você pode usar conjuntos, por exemplo, se uma regra deve corresponder a múltiplos endereços IP, números de porta, interfaces ou qualquer outro critério de correspondência.

45.5.1. Utilização de conjuntos anônimos em nftables

Um conjunto anônimo contém valores separados por vírgulas entre parênteses, como **{ 22, 80, 443 }**, que você usa diretamente em uma regra. Você também pode usar conjuntos anônimos também para endereços IP ou qualquer outro critério de correspondência.

A desvantagem dos conjuntos anônimos é que, se você quiser mudar o conjunto, você deve substituir a regra. Para uma solução dinâmica, use os conjuntos nomeados como descrito em Seção 45.5.2, "Usando conjuntos nomeados em nftables".

Pré-requisitos

- A cadeia **example_chain** e a tabela **example_table** da família **inet** existe.

Procedimento

1. Por exemplo, para adicionar uma regra a **example_chain** em **example_table** que permite o tráfego de entrada para a porta **22**, **80** e **443**:

```
# nft adicionar regra inet example_table example_chain tcp dport { 22, 80, 443 } aceitar
```

2. Opcionalmente, exibir todas as correntes e suas regras em **example_table**:

```
# nft list table inet example_table
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport { ssh, http, https } accept
  }
}
```

45.5.2. Usando conjuntos nomeados em nftables

A estrutura **nftables** suporta conjuntos de nomes mutáveis. Um conjunto nomeado é uma lista ou gama de elementos que você pode usar em múltiplas regras dentro de uma tabela. Outro benefício sobre os conjuntos anônimos é que você pode atualizar um conjunto nomeado sem substituir as regras que utilizam o conjunto.

Quando você cria um conjunto nomeado, você deve especificar o tipo de elementos que o conjunto contém. Você pode definir os seguintes tipos:

- **ipv4_addr** para um conjunto que contenha endereços ou faixas IPv4, como **192.0.2.1** ou **192.0.2.0/24**.
- **ipv6_addr** para um conjunto que contenha endereços ou faixas IPv6, como **2001:db8:1::1** ou **2001:db8:1::1/64**.
- **ether_addr** para um conjunto que contém uma lista de endereços de controle de acesso à mídia (MAC), tais como **52:54:00:6b:66:42**.

- **inet_proto** para um conjunto que contém uma lista de tipos de protocolos de Internet, como **tcp**.
- **inet_service** para um conjunto que contém uma lista de serviços de Internet, tais como **ssh**.
- **mark** para um conjunto que contém uma lista de marcas de pacotes. As marcas de pacotes podem ser qualquer valor inteiro positivo de 32 bits (**0** a **2147483647**].

Pré-requisitos

- A cadeia **example_chain** e a tabela **example_table** existem.

Procedimento

1. Criar um conjunto vazio. Os exemplos a seguir criam um conjunto para endereços IPv4:
   - Para criar um conjunto que possa armazenar múltiplos endereços IPv4 individuais:

     ```
     # nft add set inet example_table example_set { type ipv4_addr }; }
     ```

   - Para criar um conjunto que possa armazenar faixas de endereços IPv4:

     ```
     # nft add set inet example_table example_set { type ipv4_addr \; flags interval \; }
     ```

     IMPORTANTE

     Para evitar que a casca interprete os ponto-e-vírgula como o fim do comando, você deve escapar dos pontos-e-vírgula com uma barra invertida.

2. Opcionalmente, criar regras que utilizem o conjunto. Por exemplo, o seguinte comando adiciona uma regra ao **example_chain** no site **example_table** que irá descartar todos os pacotes de endereços IPv4 em **example_set**.

   ```
   # nft add rule inet example_table example_chain ip saddr @example_set drop
   ```

   Como **example_set** ainda está vazio, a regra atualmente não tem efeito.

3. Adicionar endereços IPv4 a **example_set**:
   - Se você criar um conjunto que armazene endereços IPv4 individuais, entre:

     ```
     # nft adicionar elemento inet example_table example_set { 192.0.2.1, 192.0.2.2 }
     ```

   - Se você criar um conjunto que armazene faixas IPv4, entre:

     ```
     # nft adicionar elemento inet example_table example_set { 192.0.2.0-192.0.2.255 }
     ```

     Quando você especifica uma faixa de endereços IP, você pode alternativamente usar a notação Classless Inter-Domain Routing (CIDR), como por exemplo **192.0.2.0/24** no exemplo acima.

45.5.3. Informações relacionadas

- Para mais detalhes sobre os conjuntos, consulte a seção **Sets** na página de manual **nft(8)**.

45.6. USANDO MAPAS DE VEREDICTOS EM COMANDOS NFTABLES

Os mapas verídicos, que também são conhecidos como dicionários, permitem que **nft** execute uma ação baseada em informações de pacotes, mapeando critérios de correspondência a uma ação.

45.6.1. Usando mapas literais em nftables

Um mapa literal é um **{ *match_criteria* : *action* }** declaração de que você usa diretamente em uma regra. A declaração pode conter vários mapeamentos separados por vírgula.

A desvantagem de um mapa literal é que se você quiser mudar o mapa, você deve substituir a regra. Para uma solução dinâmica, use mapas de veredictos nomeados, como descrito em Seção 45.6.2, “Usando mapas de veredictos mutáveis em nftables”.

O exemplo descreve como usar um mapa literal para encaminhar tanto os pacotes TCP e UDP do protocolo IPv4 e IPv6 para diferentes cadeias a fim de contar separadamente os pacotes TCP e UDP que chegam.

Procedimento

1. Crie o **example_table**:

```
# nft adicionar tabela inet exemplo_tabela
```

2. Criar a cadeia **tcp_packets** em **example_table**:

```
# nft add chain inet exemplo_tabela tcp_packets
```

3. Adicione uma regra a **tcp_packets** que conta o tráfego nesta cadeia:

```
# nft add rule inet example_table tcp_packets counter
```

4. Crie a cadeia **udp_packets** em **example_table**

```
# nft add chain inet exemplo_tabela udp_packets
```

5. Adicione uma regra a **udp_packets** que conta o tráfego nesta cadeia:

```
# nft add rule inet example_table udp_packets counter
```

6. Criar uma cadeia para o tráfego de entrada. Por exemplo, para criar uma cadeia chamada **incoming_traffic** em **example_table** que filtra o tráfego de entrada:

```
# nft add chain inet example_table incoming_traffic { type filter hook input priority 0 { type filter hook input priority 0}; }
```

7. Adicione uma regra com um mapa literal a **incoming_traffic**:

```
# nft add rule inet example_table incoming_traffic ip protocol vmap { tcp : jump tcp_packets, udp : jump udp_packets }
```

O mapa literal distingue os pacotes e os envia para as diferentes cadeias de contadores com base em seu protocolo.

8. Para listar os balcões de trânsito, exibir **example_table**:

```
# nft list table inet example_table
table inet example_table {
  chain tcp_packets {
    counter packets 36379 bytes 2103816
  }

  chain udp_packets {
    counter packets 10 bytes 1559
  }

  chain incoming_traffic {
    type filter hook input priority filter; policy accept;
    ip protocol vmap { tcp : jump tcp_packets, udp : jump udp_packets }
  }
}
```

Os balcões da cadeia **tcp_packets** e **udp_packets** exibem tanto o número de pacotes recebidos quanto o número de bytes.

45.6.2. Usando mapas de veredictos mutáveis em nftables

A estrutura **nftables** suporta mapas de veredictos mutáveis. Você pode usar estes mapas em várias regras dentro de uma tabela. Outro benefício sobre os mapas literais é que você pode atualizar um mapa mutável sem substituir as regras que o utilizam.

Quando você cria um mapa de veredicto mutável, você deve especificar o tipo de elementos

- **ipv4_addr** para um mapa cuja parte correspondente contém um endereço IPv4, tal como **192.0.2.1**.
- **ipv6_addr** para um mapa cuja parte correspondente contém um endereço IPv6, tal como **2001:db8:1::1**.
- **ether_addr** para um mapa cuja parte correspondente contém um endereço de controle de acesso à mídia (MAC), tal como **52:54:00:6b:66:42**.
- **inet_proto** para um mapa cuja parte correspondente contém um tipo de protocolo Internet, tal como **tcp**.
- **inet_service** para um mapa cuja parte correspondente contém um número de porta do nome dos serviços da Internet, como **ssh** ou **22**.
- **mark** para um mapa cuja parte correspondente contém uma marca de pacote. Uma marca de pacote pode ser qualquer valor inteiro positivo de 32 bits (**0** a **2147483647**.
- **counter** para um mapa cuja parte correspondente contém um contravalor. O valor do contador pode ser qualquer valor inteiro positivo de 64 bits.
- **quota** para um mapa cuja parte correspondente contém um valor de cota. O valor da cota pode ser qualquer valor inteiro positivo de 64 bits.

O exemplo descreve como permitir ou largar pacotes de entrada com base em seu endereço IP de origem. Usando um mapa de veredicto mutável, é necessária apenas uma única regra para configurar este cenário enquanto os endereços IP e ações são armazenados dinamicamente no mapa. O

procedimento também descreve como adicionar e remover entradas do mapa.

Procedimento

1. Criar uma mesa. Por exemplo, para criar uma tabela chamada **example_table** que processa pacotes IPv4:

```
# nft adicionar tabela ip exemplo_tabela
```

2. Criar uma corrente. Por exemplo, para criar uma cadeia chamada **example_chain** em **example_table**:

```
# nft add chain ip example_table example_chain { type filter hook input priority 0 {\i1}; {\i1}
```

IMPORTANTE

Para evitar que a casca interprete os ponto-e-vírgula como o fim do comando, você deve escapar dos pontos-e-vírgula com uma barra invertida.

3. Criar um mapa vazio. Por exemplo, para criar um mapa para endereços IPv4:

```
# nft add map ip example_table example_map { type ipv4_addr : veredicto }
```

4. Criar regras que utilizem o mapa. Por exemplo, o seguinte comando adiciona uma regra a **example_chain** em **example_table** que aplica ações a endereços IPv4 que são ambos definidos em **example_map**:

```
# nft add rule example_table example_chain ip saddr vmap @example_map
```

5. Adicionar endereços IPv4 e ações correspondentes a **example_map**:

```
# nft adicionar elemento ip example_table example_map { 192.0.2.1 : aceitar, 192.0.2.2 :
largar }
```

Este exemplo define os mapeamentos de endereços IPv4 para ações. Em combinação com a regra criada acima, o firewall aceita pacotes de **192.0.2.1** e deixa cair pacotes de **192.0.2.2**.

6. Opcionalmente, melhore o mapa adicionando outro endereço IP e declaração de ação:

```
# nft adicionar elemento ip example_table example_map { 192.0.2.3 : aceitar }
```

7. Opcionalmente, remova uma entrada do mapa:

```
# nft apagar elemento ip example_table example_map { 192.0.2.1 }
```

8. Opcionalmente, exibir o conjunto de regras:

```
# nft list ruleset
table ip example_table {
  map example_map {
    type ipv4_addr : verdict
    elements = { 192.0.2.2 : drop, 192.0.2.3 : accept }
```

```
  }

  chain example_chain {
    type filter hook input priority filter; policy accept;
    ip saddr vmap @example_map
  }
}
```

45.6.3. Informações relacionadas

- Para mais detalhes sobre os mapas de veredictos, consulte a seção **Maps** na página de manual **nft(8)**.

45.7. CONFIGURAÇÃO DO ENCAMINHAMENTO DE PORTAS USANDO NFTABLES

O redirecionamento de portas permite aos administradores encaminhar pacotes enviados a uma porta de destino específica para uma porta local ou remota diferente.

Por exemplo, se seu servidor web não tiver um endereço IP público, você pode definir uma regra de encaminhamento de porta em seu firewall que encaminha os pacotes recebidos na porta **80** e **443** no firewall para o servidor web. Com esta regra de firewall, os usuários na Internet podem acessar o servidor web usando o IP ou o nome do host do firewall.

45.7.1. Encaminhamento de pacotes de entrada para uma porta local diferente

Esta seção descreve um exemplo de como encaminhar pacotes IPv4 recebidos na porta **8022** para a porta **22** no sistema local.

Procedimento

1. Criar uma tabela com o nome **nat** com a família de endereços **ip**:

```
# nft adicionar tabela ip nat
```

2. Acrescente as cadeias **prerouting** e **postrouting** à tabela:

```
# nft -- adicionar prerouting de corrente ip nat { tipo nat hook prerouting priority -100 }; { tipo nat hook prerouting priority -100 }; { tipo nat hook prerouting priority -100 }
```

NOTA

Passe a opção **--** para o comando **nft** para evitar que a casca interprete o valor de prioridade negativa como uma opção do comando **nft**.

3. Adicionar uma regra à cadeia **prerouting** que redireciona os pacotes recebidos na porta **8022** para a porta local **22**:

```
# nft adicionar regra ip nat prerouting tcp dport 8022 redirecionar para :22
```

45.7.2. Encaminhamento de pacotes de entrada em uma porta local específica para um host diferente

Você pode usar uma regra de tradução de endereço de rede de destino (DNAT) para encaminhar pacotes de entrada em uma porta local para um host remoto. Isto permite aos usuários na Internet acessar um serviço que roda em um host com um endereço IP privado.

O procedimento descreve como encaminhar os pacotes IPv4 recebidos na porta local **443** para o mesmo número de porta no sistema remoto com o endereço IP **192.0.2.1**.

Pré-requisito

- Você está logado como o usuário **root** no sistema que deve encaminhar os pacotes.

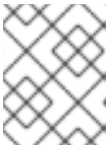
Procedimento

1. Criar uma tabela com o nome **nat** com a família de endereços **ip**:

```
# nft adicionar tabela ip nat
```

2. Acrescente as cadeias **prerouting** e **postrouting** à tabela:

```
# nft -- add chain ip nat prerouting { type nat hook prerouting priority -100 \; }
# nft add chain ip nat postrouting { type nat hook postrouting priority 100 \; }
```



NOTA

Passe a opção **--** para o comando **nft** para evitar que a casca interprete o valor de prioridade negativa como uma opção do comando **nft**.

3. Adicione uma regra à cadeia **prerouting** que redireciona os pacotes recebidos na porta **443** para a mesma porta em **192.0.2.1**:

```
# nft adicionar regra ip nat prerouting tcp dport 443 dnat a 192.0.2.1
```

4. Adicione uma regra à cadeia **postrouting** para disfarçar o tráfego de saída:

```
# nft adicionar regra ip daddr 192.0.2.1 mascarada
```

5. Habilitar o envio de pacotes:

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

45.8. UTILIZAÇÃO DE NFTABLES PARA LIMITAR A QUANTIDADE DE CONEXÕES

Você pode usar **nftables** para limitar o número de conexões ou para bloquear endereços IP que tentam estabelecer uma determinada quantidade de conexões para evitar que elas usem muitos recursos do sistema.

45.8.1. Limitando o número de conexões usando nftables

O parâmetro **ct count** do utilitário **nft** permite aos administradores limitar o número de conexões. O procedimento descreve um exemplo básico de como limitar as conexões de entrada.

Pré-requisitos

- A base **example_chain** em **example_table** existe.

Procedimento

1. Adicione uma regra que permite apenas duas conexões simultâneas à porta SSH (22) a partir de um endereço IPv4 e rejeita todas as outras conexões a partir do mesmo IP:

```
# nft add rule ip example_table example_chain tcp dport ssh meter example_meter { ip saddr ct count over 2 } counter reject
```

2. Opcionalmente, exibir o medidor criado na etapa anterior:

```
# nft list meter ip example_table example_meter
table ip example_table {
  meter example_meter {
    type ipv4_addr
    size 65535
    elements = { 192.0.2.1 : ct count over 2 , 192.0.2.2 : ct count over 2  }
  }
}
```

A entrada **elements** exibe endereços que atualmente correspondem à regra. Neste exemplo, **elements** lista os endereços IP que têm conexões ativas com a porta SSH. Observe que a saída não exibe o número de conexões ativas ou se as conexões foram rejeitadas.

45.8.2. Bloqueio de endereços IP que tentam mais de dez novas conexões TCP de entrada em um minuto

A estrutura **nftables** permite que os administradores atualizem dinamicamente os conjuntos. Esta seção explica como usar esta funcionalidade para bloquear temporariamente hosts que estão estabelecendo mais de dez conexões TCP IPv4 dentro de um minuto. Após cinco minutos, **nftables** remove automaticamente o endereço IP da lista de negação.

Procedimento

1. Criar a tabela **filter** com a família de endereços **ip**:

```
# nft adicionar tabela ip filter
```

2. Acrescente a cadeia **input** à tabela **filter**:

```
# nft add chain ip filter input { type filter hook input priority 0 { type hook input priority 0}; { type filter hook input priority 0}; { type filter hook input priority 0
```

3. Adicione um conjunto chamado **denylist** à tabela **filter**:

```
# nft add set ip filter denylist { type ipv4_addr }; flags dynamic, timeout; timeout 5m; timeout
```

Este comando cria um conjunto dinâmico para endereços IPv4. O parâmetro **timeout 5m** define que **nftables** remove automaticamente as entradas após 5 minutos do conjunto.

4. Adicionar uma regra que automaticamente adiciona o endereço IP de origem dos hosts que tentam estabelecer mais de dez novas conexões TCP dentro de um minuto ao conjunto **denylist**:

```
# nft add rule ip filter input ip protocol tcp ct state new, unracked limit rate over 10/minute add @denylist { ip saddr }
```

5. Acrescente uma regra que abandone todas as conexões de endereços IP no conjunto **denylist**:

```
# nft add rule ip filter input ip saddr @denylist drop
```

Recursos adicionais

- Seção 45.5.2, "Usando conjuntos nomeados em nftables"

45.9. REGRAS DE DEPURAÇÃO DE NFTABLES

A estrutura **nftables** oferece diferentes opções para os administradores depurarem as regras e se os pacotes corresponderem a elas. Esta seção descreve estas opções.

45.9.1. Criando uma regra com um contador

Para identificar se uma regra é igualada, você pode usar um contador. Esta seção descreve como criar uma nova regra com um contador.

Para um procedimento que acrescenta um contrário a uma regra existente, ver Seção 45.9.2, "Adicionando um contador a uma regra existente".

Pré-requisitos

- A cadeia à qual se deseja acrescentar a regra existe.

Procedimento

1. Adicione uma nova regra com o parâmetro **counter** à cadeia. O exemplo a seguir adiciona uma regra com um contador que permite o tráfego TCP na porta 22 e conta os pacotes e o tráfego que correspondem a esta regra:

```
# nft adicionar regra inet example_table example_chain tcp dport 22 counter accept
```

2. Para exibir os valores do contador:

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
```

```
    tcp dport ssh counter packets 6872 bytes 105448565 accept
  }
}
```

45.9.2. Adicionando um contador a uma regra existente

Para identificar se uma regra é igualada, você pode usar um contador. Esta seção descreve como adicionar um contador a uma regra existente.

Para um procedimento para adicionar uma nova regra com um contador, ver Seção 45.9.1, "Criando uma regra com um contador".

Pré-requisitos

- A regra à qual se deseja acrescentar o contador existe.

Procedimento

1. Mostrar as regras da corrente incluindo seus cabos:

```
# nft --handle list chain inet example_table example_chain
table inet example_table {
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport ssh accept # handle 4
  }
}
```

2. Adicione o contador substituindo a regra, mas com o parâmetro **counter**. O exemplo a seguir substitui a regra exibida na etapa anterior e adiciona um contador:

```
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 counter accept
```

3. Para exibir os valores do contador:

```
# nft list ruleset
table inet example_table {
  chain example_chain {
    type filter hook input priority filter; policy accept;
    tcp dport ssh counter packets 6872 bytes 105448565 accept
  }
}
```

45.9.3. Pacotes de monitoramento que correspondem a uma regra existente

O recurso de rastreamento em **nftables** em combinação com o comando **nft monitor** permite que os administradores exibam pacotes que correspondem a uma regra. O procedimento descreve como permitir o rastreamento de uma regra, bem como o monitoramento de pacotes que correspondam a esta regra.

Pré-requisitos

- A regra à qual se deseja acrescentar o contador existe.

Procedimento

1. Mostrar as regras da corrente incluindo seus cabos:

```
# nft --handle list chain inet example_table example_chain
table inet example_table {
  chain example_chain { # handle 1
    type filter hook input priority filter; policy accept;
    tcp dport ssh accept # handle 4
  }
}
```

2. Adicione o recurso de rastreamento substituindo a regra, mas com os parâmetros **meta nftrace set 1**. O exemplo a seguir substitui a regra exibida na etapa anterior e permite o rastreamento:

```
# nft replace rule inet example_table example_chain handle 4 tcp dport 22 meta nftrace set 1 accept
```

3. Use o comando **nft monitor** para exibir o rastreamento. O seguinte exemplo filtra a saída do comando para exibir somente as entradas que contenham **inet example_table example_chain**:

```
# nft monitor | grep "inet example_table example_chain"
trace id 3c5eb15e inet example_table example_chain packet: iif "enp1s0" ether saddr 52:54:00:17:ff:e4 ether daddr 52:54:00:72:2f:6e ip saddr 192.0.2.1 ip daddr 192.0.2.2 ip dscp cs0 ip ecn not-ect ip ttl 64 ip id 49710 ip protocol tcp ip length 60 tcp sport 56728 tcp dport ssh tcp flags == syn tcp window 64240
trace id 3c5eb15e inet example_table example_chain rule tcp dport ssh nftrace set 1 accept (verdict accept)
...
```

ATENÇÃO

Dependendo do número de regras com rastreamento habilitado e da quantidade de tráfego correspondente, o comando **nft monitor** pode exibir uma grande quantidade de resultados. Use **grep** ou outras utilidades para filtrar a saída.

45.10. APOIO E RESTAURAÇÃO DOS CONJUNTOS DE REGRAS NFTABLES

Esta seção descreve como fazer backup das regras **nftables** em um arquivo, assim como restaurar as regras de um arquivo.

Os administradores podem usar um arquivo com as regras para, por exemplo, transferir as regras para um servidor diferente.

45.10.1. Cópia de segurança dos conjuntos de regras nftables para um arquivo

Esta seção descreve como fazer backup do conjunto de regras **nftables** para um arquivo.

Procedimento

- Para fazer backup das regras **nftables**:
  - No formato **nft list ruleset**:

```
# nft list ruleset > file.nft
```

  - No formato JSON:

```
# nft -j list ruleset > file.json
```

45.10.2. Restauração de conjuntos de regras nftables a partir de um arquivo

Esta seção descreve como restaurar os conjuntos de regras **nftables**.

Procedimento

- Para restaurar as regras do **nftables**:
  - Se o arquivo a ser restaurado estiver no formato **nft list ruleset** ou contiver comandos **nft**:

```
# nft -f file.nft
```

  - Se o arquivo a ser restaurado estiver no formato JSON:

```
# nft -j -f file.json
```

45.11. INFORMAÇÕES RELACIONADAS

- O post Usando nftables no blog Red Hat Enterprise Linux 8 fornece uma visão geral sobre o uso dos recursos do **nftables**.
- O que vem depois do iptables? Seu sucessor, é claro: o artigo nftables explica porque **nftables** substitui **iptables**.
- O Firewalld: The Future is nftables article provides additional information on **nftables** as a default back end for **firewalld**.

CAPÍTULO 46. USANDO O XDP-FILTER PARA FILTRAGEM DE TRÁFEGO DE ALTO DESEMPENHO PARA EVITAR ATAQUES DDOS

Comparado aos filtros de pacotes, como **nftables**, o Express Data Path (XDP) processa e descarrega os pacotes da rede diretamente na interface da rede. Portanto, o XDP determina o próximo passo para o pacote antes que ele chegue a um firewall ou outras aplicações. Como resultado, os filtros XDP requerem menos recursos e podem processar pacotes de rede a uma taxa muito maior do que os filtros de pacotes convencionais para se defender contra ataques de Negação de Serviço Distribuída (DDoS). Por exemplo, durante os testes, a Red Hat caiu 26 milhões de pacotes de rede por segundo em um único núcleo, o que é significativamente maior do que a taxa de queda de **nftables** no mesmo hardware.

O utilitário **xdp-filter** permite ou deixa cair os pacotes de rede que chegam usando XDP. Você pode criar regras para filtrar o tráfego de ou para o específico:

- Endereços IP
- Endereços MAC
- Portos

Note que, mesmo que **xdp-filter** tenha uma taxa significativamente maior de processamento de pacotes, ele não tem as mesmas capacidades que, por exemplo, **nftables**. Considere **xdp-filter** uma utilidade conceitual para demonstrar a filtragem de pacotes usando o XDP. Além disso, você pode usar o código do utilitário para uma melhor compreensão de como escrever suas próprias aplicações XDP.

IMPORTANTE

A Red Hat fornece o utilitário **xdp-filter** como uma prévia tecnológica não suportada.

46.1. ELIMINAÇÃO DE PACOTES DE REDE QUE CORRESPONDEM A UMA REGRA DO FILTRO DE XDP

Esta seção descreve como usar **xdp-filter** para soltar pacotes de rede:

- Para um porto de destino específico
- A partir de um endereço IP específico
- A partir de um endereço MAC específico

A política **allow** de **xdp-filter** define que todo o tráfego é permitido e o filtro deixa cair apenas os pacotes de rede que correspondem a uma determinada regra. Por exemplo, use este método se você souber os endereços IP de origem dos pacotes que você deseja descartar.

Pré-requisitos

- O pacote **xdp-tools** está instalado.
- Um driver de rede que suporta programas XDP.

Procedimento

1. Carregar **xdp-filter** para processar os pacotes recebidos em uma determinada interface, como **enp1s0**:

```
# xdp-filter load enp1s0
```

Por padrão, **xdp-filter** usa a política **allow**, e a concessionária só deixa cair o tráfego que corresponde a qualquer regra.

Opcionalmente, use o **-f *feature*** opção para ativar apenas características particulares, tais como **tcp**, **ipv4**, ou **ethernet**. O carregamento apenas das características exigidas em vez de todas elas aumenta a velocidade de processamento da embalagem. Para habilitar múltiplas características, separá-las com uma vírgula.

Se o comando falhar com um erro, o driver da rede não suporta programas XDP.

2. Acrescentar regras para deixar cair pacotes que correspondam a elas. Por exemplo:

- Para deixar os pacotes recebidos na porta **22**, entre:

```
# xdp-filter port 22
```

Este comando acrescenta uma regra que combina com o tráfego TCP e UDP. Para corresponder apenas a um determinado protocolo, use o **-p *protocol*** opção.

- Para soltar os pacotes recebidos de **192.0.2.1**, entre:

```
# xdp-filter ip 192.0.2.1 -m src
```

Note que **xdp-filter** não suporta faixas de IP.

- Para soltar os pacotes recebidos do endereço MAC **00:53:00:AA:07:BE**, entre:

```
# xdp-filter ether 00:53:00:AA:07:BE -m src
```

Etapas de verificação

- Use o seguinte comando para exibir estatísticas sobre pacotes descartados e permitidos:

```
# xdp-filter status
```

Recursos adicionais

- Para mais detalhes sobre **xdp-filter**, consulte a página de manual **xdp-filter(8)**.
- Se você é um desenvolvedor e está interessado no código de **xdp-filter**, baixe e instale o RPM (SRPM) de origem correspondente no Portal do Cliente da Red Hat.

46.2. SOLTAR TODOS OS PACOTES DE REDE, EXCETO OS QUE CORRESPONDEM A UMA REGRA DO FILTRO XDP

Esta seção descreve como usar **xdp-filter** para permitir apenas os pakets de rede:

- De e para um porto de destino específico

- De e para um endereço IP específico
- De e para o endereço MAC específico

Para isso, utilize a política **deny** de **xdp-filter**, que define que o filtro deixa cair todos os pacotes de rede, exceto aqueles que correspondem a uma determinada regra. Por exemplo, use este método se você não souber os endereços IP de origem dos pacotes que você deseja descartar.

ATENÇÃO

Se você definir a política padrão para **deny** ao carregar **xdp-filter** em uma interface, o kernel imediatamente deixa cair todos os pacotes desta interface até que você crie regras que permitam certo tráfego. Para evitar ser bloqueado fora do sistema, digite os comandos localmente ou conecte-se através de uma interface de rede diferente com o host.

Pré-requisitos

- O pacote **xdp-tools** está instalado.
- Você está logado no host localmente ou usando uma interface de rede para a qual você não planeja filtrar o tráfego.
- Um driver de rede que suporta programas XDP.

Procedimento

1. Carregue **xdp-filter** para processar pacotes em uma determinada interface, como **enp1s0**:

   ```
   # xdp-filter load enp1s0 -p deny
   ```

   Opcionalmente, use o **-f *feature*** opção para ativar apenas características particulares, tais como **tcp**, **ipv4**, ou **ethernet**. O carregamento apenas das características exigidas em vez de todas elas aumenta a velocidade de processamento da embalagem. Para habilitar múltiplas características, separá-las com uma vírgula.

   Se o comando falhar com um erro, o driver da rede não suporta programas XDP.

2. Adicionar regras para permitir pacotes que correspondam a elas. Por exemplo:

   - Para permitir pacotes de e para a porta **22**, entre:

     ```
     # xdp-filter port 22
     ```

     Este comando acrescenta uma regra que combina com o tráfego TCP e UDP. Para corresponder apenas a um determinado protocolo, passe o **-p *protocol*** opção para o comando.

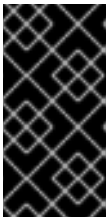
   - Para permitir pacotes de e para **192.0.2.1**, entre:

     ```
     # xdp-filter ip 192.0.2.1
     ```

Note que **xdp-filter** não suporta faixas de IP.

- Para permitir pacotes de e para o endereço MAC **00:53:00:AA:07:BE**, entre:

```
# xdp-filter ether 00:53:00:AA:07:BE
```



IMPORTANTE

A utilidade **xdp-filter** não suporta a inspeção estadual de pacotes. Isto requer que você não defina um modo usando o **-m *mode*** ou você adiciona regras explícitas para permitir o tráfego de entrada que a máquina recebe em resposta ao tráfego de saída.

Etapas de verificação

- Use o seguinte comando para exibir estatísticas sobre pacotes descartados e permitidos:

```
# xdp-filter status
```

Recursos adicionais

- Para mais detalhes sobre **xdp-filter**, consulte a página de manual **xdp-filter(8)**.
- Se você é um desenvolvedor e está interessado no código de **xdp-filter**, baixe e instale o RPM (SRPM) de origem correspondente no Portal do Cliente da Red Hat.

CAPÍTULO 47. COMEÇANDO COM DPDK

O Data Plane Development Kit (DPDK) fornece bibliotecas e drivers de rede para acelerar o processamento de pacotes no espaço do usuário.

Os administradores usam DPDK, por exemplo, em máquinas virtuais para usar a Virtualização de E/S de raiz única (SR-IOV) para reduzir as latências e aumentar a produção de E/S.

NOTA

A Red Hat não suporta APIs DPDK experimentais.

47.1. INSTALANDO O PACOTE DPDK

Esta seção descreve como instalar o pacote **dpdk**.

Pré-requisitos

- O Red Hat Enterprise Linux está instalado.
- Uma assinatura válida é designada para o anfitrião.

Procedimento

1. Use o utilitário **yum** para instalar o pacote **dpdk**:

```
# yum instalar dpdk
```

47.2. INFORMAÇÕES RELACIONADAS

- Para uma lista de adaptadores de rede que suportam SR-IOV no Red Hat Enterprise Linux 8, veja Network Adapter Fast Datapath Feature Support Matrix .

CAPÍTULO 48. ENTENDENDO AS CARACTERÍSTICAS DA REDE EBPF NA RHEL

O Filtro de Pacotes Berkeley estendido (eBPF) é uma máquina virtual no kernel que permite a execução do código no espaço do kernel. Este código é executado em um ambiente restrito de sandbox com acesso apenas a um conjunto limitado de funções.

Em rede, você pode usar o eBPF para complementar ou substituir o processamento de pacotes de kernel. Dependendo do gancho usado, os programas eBPF têm, por exemplo:

- Acesso de leitura e escrita aos dados e metadados dos pacotes
- Pode procurar soquetes e rotas
- Pode definir opções de soquetes
- Pode redirecionar pacotes

48.1. VISÃO GERAL DAS CARACTERÍSTICAS DE REDE EBPF NA RHEL

Você pode anexar programas ampliados de rede Berkeley Paket Filter (eBPF) aos seguintes ganchos no RHEL:

- eXpress Data Path (XDP): Fornece acesso antecipado aos pacotes recebidos antes que a pilha do kernel em rede os processe.
- **tc** Classificador eBPF com bandeira de ação direta: Fornece um poderoso processamento de pacotes na entrada e na saída.
- Grupos de controle versão 2 (cgroup v2): Permite filtrar e sobrepor operações baseadas em soquetes realizadas por programas em um grupo de controle.
- Filtragem de soquetes: Permite a filtragem de pacotes recebidos de soquetes. Esta característica também estava disponível no clássico Berkeley Packet Filter (cBPF), mas foi estendida para suportar programas eBPF.
- Stream parser: Permite dividir os fluxos em mensagens individuais, filtrá-las e redirecioná-las para soquetes.
- **SO_REUSEPORT** seleção do soquete: Fornece uma seleção programável de um soquete receptor de um grupo de soquetes **reuseport**.
- Dissecador de fluxo: Permite anular o modo como os cabeçalhos de pacotes de análise do núcleo em determinadas situações.
- Controle de congestionamento TCP: Permite a implementação de um algoritmo personalizado de controle de congestionamento TCP.
- Rotas com encapsulamento: Permite criar encapsulamento personalizado de túneis.

Note que a Red Hat não suporta todas as funcionalidades do eBPF que estão disponíveis no RHEL e descritas aqui. Para mais detalhes e o status de suporte dos ganchos individuais, consulte as Notas de Lançamento do RHEL 8 e a seguinte visão geral.

XDP

Você pode anexar programas do tipo **BPF_PROG_TYPE_XDP** a uma interface de rede. O kernel então

executa o programa nos pacotes recebidos antes que a pilha de rede do kernel comece a processá-los. Isto permite o encaminhamento rápido de pacotes em determinadas situações, tais como queda rápida de pacotes para evitar ataques de Negação de Serviço Distribuída (DDoS) e redirecionamentos rápidos de pacotes para cenários de balanceamento de carga.

Você também pode usar o XDP para diferentes formas de monitoramento e amostragem de pacotes. O kernel permite que os programas XDP modifiquem os pacotes e os passem para processamento posterior na pilha de rede do kernel.

Os seguintes modos XDP estão disponíveis:

- Nativo (motorista) XDP: O kernel executa o programa a partir do ponto mais próximo possível durante a recepção do pacote. Neste momento, o kernel não analisou o pacote e, portanto, nenhum metadado fornecido pelo kernel está disponível. Este modo requer que o driver da interface de rede suporte XDP, mas nem todos os drivers suportam este modo nativo.

- Genéricos XDP: A pilha de rede do núcleo executa o programa XDP no início do processamento. Naquele momento, as estruturas de dados do kernel foram alocadas, e o pacote foi pré-processado. Se um pacote deve ser descartado ou redirecionado, ele requer uma sobrecarga significativa em comparação com o modo nativo. Entretanto, o modo genérico não requer suporte de driver de interface de rede e funciona com todas as interfaces de rede.

- XDP descarregado: O kernel executa o programa XDP na interface de rede ao invés de na CPU do host. Note que isto requer hardware específico, e somente certas características do eBPF estão disponíveis neste modo.

Na RHEL, carregue todos os programas XDP usando a biblioteca **libxdp**. Esta biblioteca permite o uso controlado pelo sistema do XDP.

NOTA

Atualmente, há algumas limitações na configuração do sistema para programas XDP. Por exemplo, é necessário desativar certos recursos de descarregamento de hardware na interface de recepção. Além disso, nem todas as características estão disponíveis com todos os drivers que suportam o modo nativo.

No RHEL 8.3, a Red Hat suporta o recurso XDP somente se todas as seguintes condições se aplicarem:

- Você carrega o programa XDP em uma arquitetura AMD ou Intel de 64 bits.

- Você usa a biblioteca **libxdp** para carregar o programa no kernel.

- O programa XDP utiliza um dos seguintes códigos de retorno: **XDP_ABORTED**, **XDP_DROP**, ou **XDP_PASS**.

- O programa XDP não utiliza a descarga de hardware XDP.

Além disso, a Red Hat fornece o seguinte uso das características do XDP como uma visualização tecnológica não suportada:

- Carregamento de programas XDP em arquiteturas que não AMD e Intel 64 bits. Note que a biblioteca **libxdp** não está disponível para outras arquiteturas que não AMD e Intel 64 bits.

- Os códigos de retorno **XDP_TX** e **XDP_REDIRECT**.

- A descarga de hardware do XDP.

AF_XDP

Usando um programa XDP que filtra e redireciona os pacotes para um determinado soquete **AF_XDP**, você pode usar um ou mais soquetes da família de protocolos **AF_XDP** para copiar rapidamente os pacotes do kernel para o espaço do usuário.

No RHEL 8.3, a Red Hat fornece esta característica como uma prévia tecnológica não suportada.

Controle de tráfego

O subsistema de Controle de Tráfego (**tc**) oferece os seguintes tipos de programas de eBPF:

- **BPF_PROG_TYPE_SCHED_CLS**
- **BPF_PROG_TYPE_SCHED_ACT**

Estes tipos permitem escrever classificadores personalizados **tc** e ações **tc** no eBPF. Juntamente com as partes do ecossistema **tc**, isto proporciona a capacidade de processamento poderoso de pacotes e é a parte central de várias soluções de orquestração em rede de contêineres.

Na maioria dos casos, somente o classificador é usado, como com a bandeira de ação direta, o classificador eBPF pode executar ações diretamente do mesmo programa eBPF. A Disciplina de Enfileiramento **clsact** (**qdisc**) foi projetada para permitir isso no lado da entrada.

Note que o uso de um programa de dissecador de fluxo eBPF pode influenciar a operação de alguns outros classificadores **qdiscs** e **tc**, tais como **flower**.

O recurso eBPF para **tc** é totalmente suportado no RHEL 8.2 e posteriores.

Filtro de soquetes

Vários utilitários usam ou já usaram o clássico Berkeley Packet Filter (cBPF) para filtrar os pacotes recebidos em um soquete. Por exemplo, o utilitário **tcpdump** permite que o usuário especifique expressões, que **tcpdump** então se traduz em código cBPF.

Como alternativa ao cBPF, o kernel permite programas eBPF do tipo **BPF_PROG_TYPE_SOCKET_FILTER** para o mesmo propósito.

No RHEL 8.3, a Red Hat fornece esta característica como uma prévia tecnológica não suportada.

Grupos de controle

Na RHEL, você pode usar vários tipos de programas eBPF que você pode anexar a um cgroup. O kernel executa estes programas quando um programa no cgroup em questão executa uma operação. Note que você pode usar apenas a versão 2 do cgroups.

Os seguintes programas de cgroup eBPF relacionados à rede estão disponíveis na RHEL:

- **BPF_PROG_TYPE_SOCK_OPS**: O kernel chama este programa durante um TCP **connect** e permite a configuração de operações TCP por soquete.
- **BPF_PROG_TYPE_CGROUP_SOCK_ADDR**: O kernel chama este programa durante **connect**, **bind**, **sendto**, e **recvmsg** operações. Este programa permite a mudança de endereços IP e portas.
- **BPF_PROG_TYPE_CGROUP_SOCKOPT**: O kernel chama este programa durante as operações em **setsockopt** e **getsockopt** e permite mudar as opções.
- **BPF_PROG_TYPE_CGROUP_SOCK**: O kernel chama este programa durante a criação de soquetes e a vinculação a endereços. Você pode usar estes programas para permitir ou negar a operação, ou apenas para inspecionar a criação de soquetes para estatísticas.

- **BPF_PROG_TYPE_CGROUP_SKB**: Este programa filtra pacotes individuais na entrada e na saída, e pode aceitar ou rejeitar pacotes.
- **BPF_PROG_TYPE_CGROUP_SYSCTL**: Este programa permite a filtragem do acesso aos controles do sistema (**sysctl**).

No RHEL 8.3, a Red Hat fornece esta característica como uma prévia tecnológica não suportada.

Stream Parser

Um analisador de fluxo opera em um grupo de tomadas que são adicionadas a um mapa especial de eBPF. O programa eBPF processa então os pacotes que o kernel recebe ou envia nesses soquetes.

Os seguintes programas de análise de fluxos eBPF estão disponíveis na RHEL:

- **BPF_PROG_TYPE_SK_SKB**: Um programa eBPF analisa os pacotes recebidos do soquete em mensagens individuais, e instrui o kernel a soltar essas mensagens ou enviá-las para outro soquete do grupo.
- **BPF_PROG_TYPE_SK_MSG**: Este programa filtra as mensagens de saída. Um programa eBPF analisa os pacotes em mensagens individuais e os aprova ou rejeita.

No RHEL 8.3, a Red Hat fornece esta característica como uma prévia tecnológica não suportada.

SO_REUSEPORT seleção de soquetes

Usando esta opção de soquete, você pode ligar vários soquetes ao mesmo endereço IP e porta. Sem o eBPF, o kernel seleciona o soquete receptor com base em um hash de conexão. Com o programa **BPF_PROG_TYPE_SK_REUSEPORT**, a seleção do soquete receptor é totalmente programável.

No RHEL 8.3, a Red Hat fornece esta característica como uma prévia tecnológica não suportada.

Dissecador de fluxo

Quando o kernel precisa processar cabeçalhos de pacotes sem passar pela decodificação do protocolo completo, eles são **dissected**. Por exemplo, isto acontece no subsistema **tc**, em roteamento multipath, em bonding, ou ao calcular um hash de pacote. Nesta situação o kernel analisa os cabeçalhos dos pacotes e preenche as estruturas internas com as informações dos cabeçalhos dos pacotes. Você pode substituir este analisador interno usando o programa **BPF_PROG_TYPE_FLOW_DISSECTOR**. Note que você só pode dissecar TCP e UDP sobre IPv4 e IPv6 no eBPF no RHEL.

No RHEL 8.3, a Red Hat fornece esta característica como uma prévia tecnológica não suportada.

Controle de Congestionamento TCP

Você pode escrever um algoritmo de controle de congestionamento TCP personalizado usando um grupo de programas **BPF_PROG_TYPE_STRUCT_OPS** que implementam **struct tcp_congestion_oops** callbacks. Um algoritmo que é implementado desta forma está disponível para o sistema junto com os algoritmos de kernel embutidos.

No RHEL 8.3, a Red Hat fornece esta característica como uma prévia tecnológica não suportada.

Rotas com encapsulamento

Você pode anexar um dos seguintes tipos de programa eBPF a rotas na tabela de rotas como um atributo de encapsulamento de túnel:

- **BPF_PROG_TYPE_LWT_IN**
- **BPF_PROG_TYPE_LWT_OUT**
- **BPF_PROG_TYPE_LWT_XMIT**

A funcionalidade de tal programa eBPF é limitada a configurações específicas de túneis e não permite criar uma solução genérica de encapsulamento ou decapsulamento.

No RHEL 8.3, a Red Hat fornece esta característica como uma prévia tecnológica não suportada.

CAPÍTULO 49. RASTREAMENTO DE REDE USANDO A COLEÇÃO DE COMPILADORES BPF

Esta seção explica o que é a Coleção de Compiladores BPF (BCC), como instalar o BCC, bem como realizar diferentes operações de rastreamento de rede usando os scripts pré-criados fornecidos pelo pacote **bcc-tools**. Todos estes scripts suportam o parâmetro **--ebpf** para exibir o código eBPF que o utilitário carrega para o kernel. Você pode usar o código para aprender mais sobre como escrever scripts eBPF.

49.1. UMA INTRODUÇÃO AO BCC

BPF Compiler Collection (BCC) é uma biblioteca, que facilita a criação dos programas ampliados de Filtro de Pacotes Berkeley (eBPF). A principal utilidade dos programas eBPF é analisar o desempenho do sistema operacional e o desempenho da rede sem ter problemas de overhead ou de segurança.

BCC elimina a necessidade de os usuários conhecerem detalhes técnicos profundos do eBPF, e fornece muitos pontos de partida prontos para uso, tais como o pacote **bcc-tools** com programas eBPF pré-criados.

NOTA

Os programas eBPF são acionados em eventos, tais como E/S de disco, conexões TCP e criações de processo. É improvável que os programas causem o colapso, loop ou tornem-se insensíveis por funcionarem em uma máquina virtual segura no kernel.

49.2. INSTALANDO O PACOTE BCC-TOOLS

Esta seção descreve como instalar o pacote **bcc-tools**, que também instala a biblioteca BPF Compiler Collection (BCC) como uma dependência.

Pré-requisitos

- Um ativo *Red Hat Enterprise Linux subscription*
- Um *enabled repository* contendo o pacote **bcc-tools**
- Kernel atualizado
- Permissões de raiz.

Procedimento

1. Instale **bcc-tools**:

```
# yum install bcc-tools
```

   As ferramentas BCC estão instaladas no diretório **/usr/share/bcc/tools/**.

2. Opcionalmente, inspecionar as ferramentas:

```
# ll /usr/share/bcc/tools/
...
-rwxr-xr-x. 1 root root  4198 Dec 14 17:53 dcsnoop
```

```
-rwxr-xr-x. 1 root root  3931 Dec 14 17:53 dcstat
-rwxr-xr-x. 1 root root 20040 Dec 14 17:53 deadlock_detector
-rw-r--r--. 1 root root  7105 Dec 14 17:53 deadlock_detector.c
drwxr-xr-x. 3 root root  8192 Mar 11 10:28 doc
-rwxr-xr-x. 1 root root  7588 Dec 14 17:53 execsnoop
-rwxr-xr-x. 1 root root  6373 Dec 14 17:53 ext4dist
-rwxr-xr-x. 1 root root 10401 Dec 14 17:53 ext4slower
...
```

O diretório **doc** na lista acima contém documentação para cada ferramenta.

49.3. EXIBIÇÃO DAS CONEXÕES TCP ADICIONADAS À FILA DE ACEITAÇÃO DO KERNEL

Após o kernel receber o pacote **ACK** em um aperto de mão TCP de 3 vias, o kernel move a conexão da fila **SYN** para a fila **accept** depois que o estado da conexão muda para **ESTABLISHED**. Portanto, somente as conexões TCP bem sucedidas são visíveis nesta fila.

O utilitário **tcpaccept** usa os recursos do eBPF para exibir todas as conexões que o kernel adiciona à fila **accept**. O utilitário é leve porque rastreia a função **accept()** do kernel em vez de capturar os pacotes e filtrá-los. Por exemplo, use **tcpaccept** para solução de problemas gerais para exibir as novas conexões que o servidor aceitou.

Procedimento

1. Digite o seguinte comando para iniciar o rastreamento do kernel **accept** fila:

```
# /usr/share/bcc/tools/tcpaccept
PID   COMM      IP RADDR         RPORT  LADDR    LPORT
843   sshd      4  192.0.2.17    50598  192.0.2.1  22
1107  ns-slapd  4  198.51.100.6  38772  192.0.2.1  389
1107  ns-slapd  4  203.0.113.85  38774  192.0.2.1  389
...
```

   Cada vez que o núcleo aceita uma conexão, **tcpaccept** exibe os detalhes das conexões.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcpaccept(8)**.
- Para mais detalhes sobre **tcpaccept** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcpaccept_example.txt**.
- Para exibir o script eBPF **tcpaccept8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcpaccept --ebpf**.

49.4. RASTREAMENTO DE TENTATIVAS DE CONEXÃO TCP DE SAÍDA

O utilitário **tcpconnect** usa recursos do eBPF para rastrear tentativas de conexão TCP de saída. A saída do utilitário também inclui conexões que falharam.

O utilitário **tcpconnect** é leve porque traça, por exemplo, a função connect() do kernel em vez de capturar os pacotes e filtrá-los.

Procedimento

1. Digite o seguinte comando para iniciar o processo de rastreamento que exibe todas as conexões de saída:

```
# /usr/share/bcc/tools/tcpconnect
PID    COMM         IP SADDR      DADDR          DPORT
31346  curl         4  192.0.2.1  198.51.100.16  80
31348  telnet       4  192.0.2.1  203.0.113.231  23
31361  isc-worker00 4  192.0.2.1  192.0.2.254    53
...
```

   Cada vez que o núcleo processa uma conexão de saída, **tcpconnect** exibe os detalhes das conexões.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcpconnect(8)**.
- Para mais detalhes sobre **tcpconnect** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcpconnect_example.txt**.
- Para exibir o script eBPF **tcpconnect(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcpconnect --ebpf**.

49.5. MEDINDO A LATÊNCIA DAS CONEXÕES TCP DE SAÍDA

A latência da conexão TCP é o tempo necessário para estabelecer uma conexão. Isto normalmente envolve o processamento TCP/IP do kernel e o tempo de viagem de ida e volta da rede, e não o tempo de execução da aplicação.

O utilitário **tcpconnlat** usa recursos do eBPF para medir o tempo entre um pacote **SYN** enviado e o pacote de resposta recebida.

Procedimento

1. Comece a medir a latência das conexões de saída:

```
# /usr/share/bcc/tools/tcpconnlat
PID    COMM         IP SADDR      DADDR          DPORT LAT(ms)
32151  isc-worker00 4  192.0.2.1  192.0.2.254    53    0.60
32155  ssh          4  192.0.2.1  203.0.113.190  22    26.34
32319  curl         4  192.0.2.1  198.51.100.59  443   188.96
...
```

   Cada vez que o kernel processa uma conexão de saída, **tcpconnlat** exibe os detalhes da conexão depois que o kernel recebe o pacote de resposta.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcpconnlat(8)**.
- Para mais detalhes sobre **tcpconnlat** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcpconnlat_example.txt**.
- Para exibir o script eBPF **tcpconnlat(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcpconnlat --ebpf**.

49.6. EXIBINDO DETALHES SOBRE PACOTES TCP E SEGMENTOS QUE FORAM DESCARTADOS PELO KERNEL

O utilitário **tcpdrop** permite aos administradores exibir detalhes sobre pacotes TCP e segmentos que foram descartados pelo kernel. Use este utilitário para depurar altas taxas de pacotes descartados que podem fazer com que o sistema remoto envie retransmissões baseadas em temporizadores. Altas taxas de pacotes e segmentos descartados podem impactar o desempenho de um servidor.

Em vez de capturar e filtrar pacotes, que consome muitos recursos, o utilitário **tcpdrop** usa recursos do eBPF para recuperar as informações diretamente do kernel.

Procedimento

1. Digite o seguinte comando para começar a exibir detalhes sobre pacotes e segmentos TCP descartados:

```
# /usr/share/bcc/tools/tcpdrop
TIME     PID    IP SADDR:SPORT       > DADDR:DPORT   STATE (FLAGS)
13:28:39 32253  4  192.0.2.85:51616  > 192.0.2.1:22  CLOSE_WAIT (FIN|ACK)
 b'tcp_drop+0x1'
 b'tcp_data_queue+0x2b9'
 ...

13:28:39 1      4  192.0.2.85:51616  > 192.0.2.1:22   CLOSE (ACK)
 b'tcp_drop+0x1'
 b'tcp_rcv_state_process+0xe2'
 ...
```

   Cada vez que o kernel deixa cair pacotes TCP e segmentos, **tcpdrop** exibe os detalhes da conexão, incluindo o traço da pilha do kernel que levou ao pacote deixado cair.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcpdrop(8)**.
- Para mais detalhes sobre **tcpdrop** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcpdrop_example.txt**.
- Para exibir o script eBPF **tcpdrop(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcpdrop --ebpf**.

49.7. RASTREAMENTO DE SESSÕES TCP

O utilitário **tcplife** utiliza o eBPF para rastrear sessões TCP que abrem e fecham, e imprime uma linha de saída para resumir cada uma delas. Os administradores podem usar **tcplife** para identificar as conexões e a quantidade de tráfego transferido.

O exemplo nesta seção descreve como exibir as conexões à porta **22** (SSH) para recuperar as seguintes informações:

- O ID do processo local (PID)
- O nome do processo local
- O endereço IP local e o número da porta
- O endereço IP remoto e o número da porta
- A quantidade de tráfego recebido e transmitido em KB.
- O tempo em milissegundos a conexão estava ativa

Procedimento

1. Digite o seguinte comando para iniciar o rastreamento das conexões para a porta local **22**:

```
/usr/share/bcc/tools/tcplife -L 22
PID   COMM    LADDR      LPORT RADDR       RPORT TX_KB  RX_KB      MS
19392 sshd    192.0.2.1  22    192.0.2.17  43892    53     52 6681.95
19431 sshd    192.0.2.1  22    192.0.2.245 43902    81 249381 7585.09
19487 sshd    192.0.2.1  22    192.0.2.121 43970  6998      7 16740.35
...
```

   Cada vez que uma conexão é fechada, **tcplife** exibe os detalhes das conexões.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcplife(8)**.
- Para mais detalhes sobre **tcplife** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcplife_example.txt**.
- Para exibir o script eBPF **tcplife(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcplife --ebpf**.

49.8. RASTREAMENTO DE RETRANSMISSÕES TCP

O utilitário **tcpretrans** exibe detalhes sobre as retransmissões TCP, tais como o endereço IP local e remoto e o número da porta, bem como o estado TCP no momento das retransmissões.

O utilitário utiliza recursos do eBPF e, portanto, tem uma sobrecarga muito baixa.

Procedimento

1. Use o seguinte comando para começar a exibir os detalhes da retransmissão TCP:

```
# /usr/share/bcc/tools/tcpretrans
TIME     PID  IP LADDR:LPORT   T> RADDR:RPORT        STATE
00:23:02 0    4  192.0.2.1:22  R> 198.51.100.0:26788  ESTABLISHED
00:23:02 0    4  192.0.2.1:22  R> 198.51.100.0:26788  ESTABLISHED
00:45:43 0    4  192.0.2.1:22  R> 198.51.100.0:17634  ESTABLISHED
...
```

Cada vez que o núcleo chama a função de retransmissão TCP, **tcpretrans** exibe os detalhes da conexão.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcpretrans(8)**.
- Para mais detalhes sobre **tcpretrans** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcpretrans_example.txt**.
- Para exibir o script eBPF **tcpretrans(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcpretrans --ebpf**.

49.9. EXIBIÇÃO DAS INFORMAÇÕES DE MUDANÇA DE ESTADO DO TCP

Durante uma sessão de TCP, o estado do TCP muda. O utilitário **tcpstates** usa funções eBPF para rastrear essas mudanças de estado, e imprime detalhes incluindo a duração em cada estado. Por exemplo, use **tcpstates** para identificar se as conexões gastam muito tempo no estado de inicialização.

Procedimento

1. Use o seguinte comando para começar a rastrear as mudanças de estado do TCP:

```
# /usr/share/bcc/tools/tcpstates
SKADDR           C-PID C-COMM     LADDR     LPORT RADDR       RPORT OLDSTATE    ->
NEWSTATE    MS
ffff9cd377b3af80 0     swapper/1  0.0.0.0   22    0.0.0.0     0     LISTEN      -> SYN_RECV
0.000
ffff9cd377b3af80 0     swapper/1  192.0.2.1 22    192.0.2.45  53152 SYN_RECV    ->
ESTABLISHED 0.067
ffff9cd377b3af80 818   sssd_nss   192.0.2.1 22    192.0.2.45  53152 ESTABLISHED ->
CLOSE_WAIT  65636.773
ffff9cd377b3af80 1432  sshd       192.0.2.1 22    192.0.2.45  53152 CLOSE_WAIT  ->
LAST_ACK    24.409
ffff9cd377b3af80 1267  pulseaudio 192.0.2.1 22    192.0.2.45  53152 LAST_ACK    ->
CLOSE       0.376
...
```

Cada vez que uma conexão muda de estado, **tcpstates** exibe uma nova linha com detalhes de conexão atualizados.

Se várias conexões mudarem seu estado ao mesmo tempo, use o endereço da tomada na primeira coluna (**SKADDR**) para determinar quais entradas pertencem à mesma conexão.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcpstates(8)**.
- Para mais detalhes sobre **tcpstates** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcpstates_example.txt**.
- Para exibir o script eBPF **tcpstates(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcpstates --ebpf**.

49.10. RESUMINDO E AGREGANDO O TRÁFEGO TCP ENVIADO PARA SUB-REDES ESPECÍFICAS

O utilitário **tcpsubnet** resume e agrega o tráfego TCP IPv4 que o host local envia para sub-redes e exibe a saída em um intervalo fixo. O utilitário usa as características do eBPF para coletar e resumir os dados a fim de reduzir as despesas gerais.

Por padrão, **tcpsubnet** resume o tráfego para as seguintes sub-redes:

- **127.0.0.1/32**
- **10.0.0.0/8**
- **172.16.0.0/12**
- **192.0.2.0/24/16**
- **0.0.0.0/0**

Note que a última subrede (**0.0.0.0/0**) é uma opção de captura de todos. O utilitário **tcpsubnet** conta todo o tráfego para sub-redes diferente das quatro primeiras nesta entrada de "catch-all".

Siga o procedimento para contar o tráfego para as sub-redes **192.0.2.0/24** e **198.51.100.0/24**. O tráfego para outras sub-redes será rastreado na entrada da sub-rede **0.0.0.0/0** catch-all.

Procedimento

1. Comece a monitorar a quantidade de tráfego enviada para **192.0.2.0/24**, **198.51.100.0/24**, e outras sub-redes:

```
# /usr/share/bcc/tools/tcpsubnet 192.0.2.0/24,198.51.100.0/24,0.0.0.0/0
Tracing... Output every 1 secs. Hit Ctrl-C to end
[02/21/20 10:04:50]
192.0.2.0/24            856
198.51.100.0/24        7467
[02/21/20 10:04:51]
192.0.2.0/24           1200
198.51.100.0/24        8763
0.0.0.0/0               673
...
```

   Este comando exibe o tráfego em bytes para as sub-redes especificadas uma vez por segundo.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcpsubnet(8)**.
- Para mais detalhes sobre **tcpsubnet** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcpsubnet.txt**.
- Para exibir o script eBPF **tcpsubnet(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcpsubnet --ebpf**.

49.11. EXIBIÇÃO DA TAXA DE TRANSFERÊNCIA DA REDE POR ENDEREÇO IP E PORTA

O utilitário **tcptop** exibe o tráfego TCP que o host envia e recebe em kilobytes. O relatório atualiza automaticamente e contém apenas as conexões TCP ativas. O utilitário utiliza as características do eBPF e, portanto, tem apenas uma sobrecarga muito baixa.

Procedimento

1. Para monitorar o tráfego enviado e recebido, entre:

```
# /usr/share/bcc/tools/tcptop
13:46:29 loadavg: 0.10 0.03 0.01 1/215 3875

PID    COMM       LADDR           RADDR             RX_KB   TX_KB
3853   3853       192.0.2.1:22    192.0.2.165:41838 32      102626
1285   sshd       192.0.2.1:22    192.0.2.45:39240  0       0
...
```

   A saída do comando inclui apenas conexões TCP ativas. Se o sistema local ou remoto fecha uma conexão, a conexão não é mais visível na saída.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcptop(8)**.
- Para mais detalhes sobre **tcptop** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcptop.txt**.
- Para exibir o script eBPF **tcptop(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcptop --ebpf**.

49.12. RASTREAMENTO DE CONEXÕES TCP ESTABELECIDAS

O utilitário **tcptracer** rastreia as funções do kernel que conectam, aceitam e fecham as conexões TCP. O utilitário utiliza as características do eBPF e, portanto, tem uma sobrecarga muito baixa.

Procedimento

1. Use o seguinte comando para iniciar o processo de rastreamento:

```
# /usr/share/bcc/tools/tcptracer
Tracing TCP established connections. Ctrl-C to end.
T  PID    COMM         IP SADDR        DADDR       SPORT  DPORT
A  1088   ns-slapd     4  192.0.2.153  192.0.2.1   0      65535
A  845    sshd         4  192.0.2.1    192.0.2.67  22     42302
X  4502   sshd         4  192.0.2.1    192.0.2.67  22     42302
...
```

Cada vez que o núcleo se conecta, aceita ou fecha uma conexão, **tcptracer** exibe os detalhes das conexões.

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **tcptracer(8)**.
- Para mais detalhes sobre **tcptracer** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/tcptracer_example.txt**.
- Para exibir o script eBPF **tcptracer(8)** uploads para o kernel, use o comando **/usr/share/bcc/tools/tcptracer --ebpf**.

49.13. RASTREAMENTO DE TENTATIVAS DE ESCUTA IPV4 E IPV6

O utilitário **solisten** rastreia todas as tentativas de escuta de IPv4 e IPv6. Ele rastreia as tentativas de escuta, incluindo as que falham ou o programa de escuta que não aceita a conexão. O utilitário traça a função que o kernel chama quando um programa quer escutar as conexões TCP.

Procedimento

1. Digite o seguinte comando para iniciar o processo de rastreamento que exibe todas as tentativas de escuta TCP:

```
# /usr/share/bcc/tools/solisten
PID    COMM          PROTO       BACKLOG     PORT     ADDR
3643   nc            TCPv4       1           4242     0.0.0.0
3659   nc            TCPv6       1           4242     2001:db8:1::1
4221   redis-server  TCPv6       128         6379     ::
4221   redis-server  TCPv4       128         6379     0.0.0.0
....
```

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **solisten**.
- Para mais detalhes sobre **solisten** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/solisten_example.txt**.
- Para exibir o script eBPF **solisten** uploads para o kernel, use o comando **/usr/share/bcc/tools/solisten --ebpf**.

49.14. RESUMINDO O TEMPO DE SERVIÇO DAS INTERRUPÇÕES SUAVES

O utilitário **softirqs** resume o tempo gasto no serviço de interrupções suaves (soft IRQs) e mostra este tempo como totais ou distribuições de histogramas. Este utilitário usa os pontos de rastreamento do kernel **irq:softirq_enter** e **irq:softirq_exit**, que é um mecanismo de rastreamento estável.

Procedimento

1. Digite o seguinte comando para iniciar o rastreamento **soft irq** hora do evento:

```
# /usr/share/bcc/tools/softirqs
Tracing soft irq event time... Hit Ctrl-C to end.
^C
SOFTIRQ          TOTAL_usecs
tasklet                  166
block                   9152
net_rx                 12829
rcu                    53140
sched                 182360
timer                 306256
```

2. Pressione **Ctrl C** para interromper o processo de rastreamento.

Recursos adicionais

- Para mais detalhes, consulte a página de manual **softirqs**.
- Para mais detalhes sobre **softirqs** e exemplos, veja o arquivo **/usr/share/bcc/tools/doc/softirqs_example.txt**.
- Para exibir o script eBPF **solisten** uploads para o kernel, use o comando **/usr/share/bcc/tools/softirqs --ebpf**.
- Para mais detalhes sobre como **mpstat** utiliza estas informações, consulte a página de manual **mpstat(1)**.

49.15. RECURSOS ADICIONAIS

- Para maiores informações sobre a BCC, consulte o arquivo **/usr/share/doc/bcc/README.md**.

CAPÍTULO 50. COMEÇANDO COM O TIPC

Transparent Inter-process Communication (TIPC), que também é conhecido como **Cluster Domain Sockets**, é um serviço de comunicação inter-processo (IPC) para operação em cluster.

As aplicações que estão sendo executadas em um ambiente de cluster dinâmico e de alta disponibilidade têm necessidades especiais. O número de nós em um cluster pode variar, os roteadores podem falhar e, devido a considerações de equilíbrio de carga, a funcionalidade pode ser movida para diferentes nós no cluster. O TIPC minimiza o esforço dos desenvolvedores de aplicações para lidar com tais situações, e maximiza a chance de que elas sejam tratadas de forma correta e otimizada. Além disso, o TIPC proporciona uma comunicação mais eficiente e tolerante a falhas do que os protocolos gerais, como o TCP.

50.1. A ARQUITETURA DO TIPC

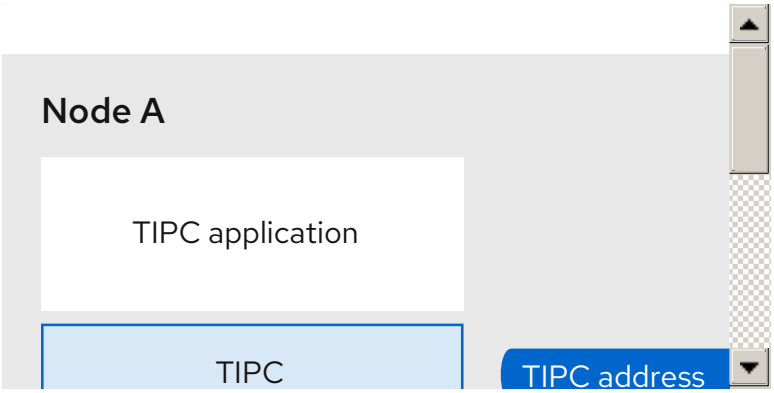
TIPC é uma camada entre aplicações que utilizam TIPC e um serviço de transporte de pacotes (**bearer**), e abrange o nível de transporte, a rede e as camadas de ligação de sinalização. Entretanto, o TIPC pode usar um protocolo de transporte diferente como portador, de modo que, por exemplo, uma conexão TCP pode servir como portador para um link de sinalização TIPC.

O TIPC suporta os seguintes portadores:

- Ethernet
- InfiniBand
- Protocolo UDP

TIPC fornece uma transferência confiável de mensagens entre as portas TIPC, que são os pontos finais de todas as comunicações TIPC.

O diagrama a seguir é um diagrama da arquitetura TIPC:



50.2. CARREGANDO O MÓDULO TIPC QUANDO O SISTEMA INICIA

Antes de poder usar o protocolo TIPC, carregue o módulo do kernel **tipc**. Esta seção explica como configurar que a RHEL carregue este módulo automaticamente quando o sistema inicia.

Procedimento

1. Crie o arquivo **/etc/modules-load.d/tipc.conf** com o seguinte conteúdo:

```
tipc
```

2. Reinicie o serviço **systemd-modules-load** para carregar o módulo sem reiniciar o sistema:

```
# systemctl start systemd-modules-load
```

Etapas de verificação

1. Use o seguinte comando para verificar se a RHEL carregou o módulo **tipc**:

```
# lsmod | grep tipc
tipc    311296  0
```

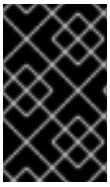
Se o comando não mostra nenhuma entrada para o módulo **tipc**, a RHEL falhou em carregá-lo.

Recursos adicionais

- Para mais detalhes sobre o carregamento de módulos quando o sistema inicia, consulte a página de manual **modules-load.d(5)**.

50.3. CRIAÇÃO DE UMA REDE TIPC

Esta seção descreve como criar uma rede TIPC.



IMPORTANTE

Os comandos configuram a rede TIPC apenas temporariamente. Para configurar permanentemente o TIPC em um nó, use os comandos deste procedimento em um script, e configure o RHEL para executar esse script quando o sistema iniciar.

Pré-requisitos

- O módulo **tipc** foi carregado. Para detalhes, veja Seção 50.2, “Carregando o módulo Tipc quando o sistema inicia”

Procedimento

1. Opcional: Definir uma identidade única do nó, como um UUID ou o nome do host do nó:

```
# tipc node set identity host_name
```

A identidade pode ser qualquer cadeia única composta de no máximo 16 letras e números.

2. Acrescente um portador. Por exemplo, para usar a Ethernet como mídia e o dispositivo **enp0s1** como dispositivo físico portador, entre:

```
# tipc bearer enable media eth device enp1s0
```

3. Opcional: Para redundância e melhor desempenho, anexar mais portadores usando o comando da etapa anterior. Você pode configurar até três portadores, mas não mais que dois na mesma mídia.

4. Repita todos os passos anteriores em cada nó que deve entrar na rede TIPC.

Etapas de verificação

1. Exibir o status do link para os membros do cluster:

```
# tipc link list
broadcast-link: up
5254006b74be:enp1s0-525400df55d1:enp1s0: up
```

Esta saída indica que o link entre o portador **enp1s0** no nó **5254006b74be** e o portador **enp1s0** no nó **525400df55d1** é **up**.

2. Exibir a tabela de publicação TIPC:

```
# tipc nametable show
Type       Lower      Upper      Scope    Port       Node
0          1795222054 1795222054 cluster  0          5254006b74be
0          3741353223 3741353223 cluster  0          525400df55d1
1          1          1          node     2399405586 5254006b74be
2          3741353223 3741353223 node     0          5254006b74be
```

- As duas entradas com o tipo de serviço **0** indicam que dois nós são membros deste agrupamento.
- A entrada com o tipo de serviço **1** representa o serviço de rastreamento de serviço de topologia embutido.
- A entrada com tipo de serviço **2** exibe o link como visto a partir do nó emissor. O limite de faixa **3741353223** representa o endereço do ponto final (um valor único de hash de 32 bits baseado na identidade do nó) em formato decimal.

Recursos adicionais

- Para detalhes sobre outros portadores que você pode usar e os parâmetros de linha de comando correspondentes, consulte a página de manual **tipc-bearer(8)**.
- Para mais detalhes sobre o comando **tipc namespace**, consulte a página de manual **tipc-namespace(8)**.

50.4. RECURSOS ADICIONAIS

- A Red Hat recomenda usar outros protocolos de nível portador para criptografar a comunicação entre os nós com base nos meios de transporte. Por exemplo, a Red Hat recomenda o uso de outros protocolos de nível portador:
  - MACSec: Para maiores detalhes, ver Capítulo 33, *Configuração de MACsec*.
  - IPsec: Para detalhes, consulte a seção Configurando uma VPN com IPsec no guia **Securing networks**.
- Para exemplos de como usar o TIPC, clone o repositório GIT upstream usando o comando **git clone git://git.code.sf.net/p/tipc/tipcutils**. Este repositório contém o código fonte das demonstrações e dos programas de teste que utilizam as funcionalidades do TIPC. Note que este repositório não é fornecido pela Red Hat.
- Para obter detalhes sobre o protocolo TIPC, consulte http://tipc.io/protocol.html.
- Para detalhes sobre a programação TIPC, consulte http://tipc.io/protocol.html.