Red Hat Enterprise Linux 8

ネットワークの設定および管理

Red Hat Enterprise Linux 8 におけるネットワークの設定と管理に関するガイド

Last Updated: 2019-05-10

Red Hat Enterprise Linux 8 ネットワークの設定および管理

Red Hat Enterprise Linux 8 におけるネットワークの設定と管理に関するガイド

法律上の通知

概要

本書は、Red Hat Enterprise Linux 8 でネットワークを管理する方法を説明します。本バージョンのドキュメントには、厳選されたユーザーストーリーが含まれています。

目次

RED HAT ドキュメントへのフィードバック **7**

第1章 ネットワークトピックの概要 **8**
1.1. IP ネットワークと非 IP ネットワークの比較 8
ネットワーク通信のカテゴリー 8
1.2. 静的 IP アドレス指定と動的 IP アドレス指定の比較 8
1.3. DHCP クライアントの動作の設定 9
DHCP のタイムアウトの設定 9
リースの更新と有効期限 9
1.3.1. DHCPv4 の永続化 9
関連資料 10
1.4. ワイヤレス規制ドメインの設定 10
関連資料 10
1.5. NETCONSOLE の設定 10
前提条件 10
手順 10
関連資料 11
1.6. 送信元マシンの設定 11
前提条件 11
手順 11
関連資料 12
1.7. SYSCTL によるネットワークカーネル調整パラメーターの使用 12
1.8. NCAT ユーティリティーを使用したデータの管理 12
ncat のインストール 13
ncat ユースケースの簡単な例 13
関連資料 14

第2章 NETWORKMANAGER を使用したネットワーク管理の開始 **15**
2.1. NETWORKMANAGER の概要 15
2.1.1. NetworkManager を使用する利点 15
関連資料 15
2.2. NETWORKMANAGER のインストール 15
関連資料 16
2.3. NETWORKMANAGER のステータスの確認 16
2.4. NETWORKMANAGER の開始 16
2.5. NETWORKMANAGER のツール 16
関連資料 17
2.6. DISPATCHER スクリプトの実行 17
関連資料 17
2.7. SYSCONFIG ファイルによる NETWORKMANAGER の使用 17
2.7.1. 従来のネットワークスクリプトのサポート 18
関連資料 18

第3章 ネットワーク設定方法の概要 **20**
3.1. ネットワーク設定方法の選択 20
関連資料 20

第4章 NMTUI による IP ネットワークの設定 **21**
4.1. NMTUI の使用 21
前提条件 21
手順 21
4.1.1. nmtui による接続の編集 22

前提条件 22
4.1.2. nmtui による編集済み接続への変更の適用 22
前提条件 22
手順 22
関連資料 24

第5章 NMCLI によるネットワークの設定 25
5.1. NMCLI の使用 25
関連資料 27
5.2. NMCLI のプロパティー名とエイリアスの概要 28
前提条件 28
5.3. NMCLI コマンドの簡単な例 29
関連資料 32
5.4. NMCLI による管理対象または管理対象外のデバイスの設定 32
前提条件 32
関連資料 33
5.5. NMCLI による接続プロファイルの作成および修正 33
前提条件 33
関連資料 35
5.6. NMCLI インタラクティブ接続エディターの使用 35
5.7. NMCLI を使用した接続プロファイルの修正 36
前提条件 37
5.8. 静的イーサネット接続の設定 37
5.8.1. nmcli を使用した静的イーサネット接続の設定 37
前提条件 37
5.8.2. nmcli インタラクティブエディターを使用した静的イーサネット接続の設定 39
関連資料 39
5.9. 動的イーサネット接続の設定 39
5.9.1. nmcli を使用した動的イーサネット接続の設定 39
前提条件 39
5.9.2. インタラクティブエディターを使用した動的イーサネット接続の設定 40
関連資料 40
5.10. NMCLI コマンドを使用して、静的ルートを設定する方法 41
5.11. NMCLI コマンドを使用した静的ルートの設定 41
前提条件 41
手順 41
関連資料 42
5.12. NMCLI インタラクティブモードを使用した静的ルートの設定 42
前提条件 42
手順 43
関連資料 43
5.13. NMCLI を使用して、既存の接続でデフォルトのゲートウェイの設定 44
前提条件 44
手順 44
関連資料 45
5.14. NMCLI インタラクティブモードを使用して、既存の接続でデフォルトゲートウェイの設定 45
前提条件 45
手順 45
関連資料 46

第6章 GNOME GUI を使用したネットワークの設定 47
6.1. GNOME SHELL ネットワーク接続アイコンを使用したネットワーク接続 47
6.2. CONTROL-CENTER を使用したネットワーク接続の作成 47

手順 48
6.3. CONTROL-CENTER を使用したネットワーク接続の設定 48
6.3.1. control-center を使用した有線 (イーサネット) 接続の設定 48
手順 49
基本設定オプション 49
control-center を使用した有線用の IPv4 設定 50
control center を使用した有線用の IPv6 設定 51
control-center を使用した有線用の 802.1X セキュリティー設定 52
TLS の設定 53
PWD の設定 54
FAST の設定 54
Tunneled TLS の設定 55
Protected EAP (PEAP) の設定 55
6.4. CONTROL-CENTER を使用した静的ルートの設定 56
前提条件 56
手順 56
6.5. NM-CONNECTION-EDITOR を使用した静的ルートの設定 57
前提条件 57
手順 57
6.6. CONTROL-CENTER を使用して、既存の接続でフォルトのゲートウェイの設定 58
前提条件 58
手順 58
関連資料 59
6.7. USING NM-CONNECTION-EDITOR を使用して、既存の接続でデフォルトのゲートウェイの設定 59
前提条件 59
手順 59
関連資料 61

第7章 MACSEC の設定 62
7.1. MACSEC の概要 62
7.2. NMCLI ツールを使用した MACSEC の使用 62
前提条件 62
手順 62
7.3. WPA_SUPPLICANT を使用した MACSEC の使用 62
手順 62
7.4. 関連情報 63

第8章 IPVLAN の使用 64
8.1. IPVLAN の概要 64
8.2. IPVLAN モード 64

第9章 IPVLAN ネットワークの設定 65
9.1. IPROUTE2 を使用した IPVLAN デバイスの作成および設定 65
手順 65

第10章 VRF の使用 67
関連情報 67

第11章 FIREWALLD の使用および設定 68
11.1. FIREWALLD の使用 68
11.1.1. firewalld 68
関連資料 68
インストールされているドキュメント 69
オンラインのドキュメント 69

11.1.2. ゾーン 69
11.1.3. 事前定義サービス 70
11.1.4. ランタイムおよび永続化設定 71
11.1.4.1. CLI を使用したランタイムおよび永続設定の設定の変更 71
11.2. FIREWALL-CONFIG GUI 設定ツールのインストール 72
手順 72
11.3. FIREWALLD の現在のステータスおよび設定の表示 72
11.3.1. firewalld の現在のステータスの表示 72
手順 72
関連資料 73
11.3.2. 現在の firewalld 設定の表示 73
11.3.2.1. GUI を使用して許可されるサービスの表示 73
11.3.2.2. CLI を使用した firewalld 設定の表示 73
11.4. FIREWALLD の起動 74
手順 74
11.5. FIREWALLD の停止 74
手順 74
11.6. FIREWALLD を使用したネットワークトラフィックの制御 75
11.6.1. 緊急時に CLI を使用してすべてのトラフィックの無効化 75
手順 75
11.6.2. CLI を使用して事前定義されたサービスでトラフィックの制御 75
手順 75
11.6.3. GUI を使用した事前定義サービスでトラフィックの制御 76
11.6.4. 新しいサービスの追加 76
手順 77
11.6.5. CLI を使用したポートの制御 77
11.6.5.1. ポートを開く 77
手順 77
11.6.5.2. ポートを閉じる 78
手順 78
11.6.6. GUI を使用してポートを開く 78
11.6.7. GUI を使用してプロトコルを使用したトラフィックの制御 78
11.6.8. GUI を使用してソースポートを開く 79
11.7. ファイアウォールゾーンでの作業 79
11.7.1. ゾーンの一覧 79
手順 79
11.7.2. 特定ゾーンに対する firewalld 設定の修正 79
手順 79
11.7.3. デフォルトゾーンの変更 80
手順 80
11.7.4. ゾーンへのネットワークインターフェースの割り当て 80
手順 80
11.7.5. ネットワーク接続にデフォルトゾーンの割り当て 80
手順 81
11.7.6. 新しいゾーンの作成 81
手順 81
11.7.7. ゾーンの設定ファイル 81
関連資料 82
11.7.8. 着信トラフィックにデフォルトの動作を設定するゾーンターゲットの使用 82
手順 82
11.8. ゾーンを使用し、ソースに応じた着信トラフィックの管理 82
11.8.1. ゾーンを使用し、ソースに応じた着信トラフィックの管理 82
11.8.2. ソースの追加 82

手順 83
11.8.3. ソースの削除 83
手順 83
11.8.4. ソースポートの追加 83
手順 83
11.8.5. ソースポートの削除 83
手順 83
11.8.6. ゾーンおよびソースを使用して特定ドメインのみに対してサービスの許可 84
手順 84
11.8.7. プロトコルに基づいてゾーンが許可したトラフィックの設定 84
11.8.7.1. ゾーンへのプロトコルの追加 84
手順 84
11.8.7.2. ゾーンからプロトコルの削除 85
手順 85
11.9. IP アドレスのマスカレードの設定 85
手順 85
11.10. ICMP 要求の管理 85
11.10.1. ICMP 要求の一覧表示およびブロック 86
11.10.2. GUI を使用した ICMP フィルターの設定 87
11.11. FIREWALLD を使用した IP セットの設定および制御 88
11.11.1. CLI を使用した IP セットオプションの設定 88
11.12. ファイアウォールロックダウンの設定 90
11.12.1. CLI を使用したロックダウンの設定 90
11.12.2. CLI を使用したロックダウンホワイトリストオプションの設定 90
11.12.3. 設定ファイルを使用したロックダウンホワイトリストオプションの設定 92
11.13. 拒否されたパケットのログ 93

第12章 NFTABLES の使用 **95**

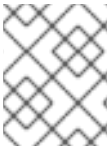
第13章 NFTABLES の概要 **96**
関連資料 96
13.1. 関連情報 96

RED HAT ドキュメントへのフィードバック

ドキュメントの改善に関するご意見やご要望をお聞かせください。

- 特定の文章に簡単なコメントを記入する場合は、ドキュメントが Multi-page HTML 形式になっているのを確認してください。コメントを追加する部分を強調表示し、そのテキストの下に表示される **Add Feedback** ポップアップをクリックし、表示された手順に従ってください。
- より詳細なフィードバックを行う場合は、Bugzilla のチケットを作成します。
    1. Bugzilla の Web サイトにアクセスします。
    2. Component で **Documentation** を選択します。
    3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも記入してください。
    4. **Submit Bug** をクリックします。

第1章 ネットワークトピックの概要



注記

本章では、実行すべきコマンドを紹介します。**root** 権限が必要なコマンドには **#** がついており、一般ユーザーが実行できるコマンドには **$** がついています。

1.1. IP ネットワークと非 IP ネットワークの比較

ネットワークとは、ファイル、プリンタ、アプリケーション、インターネット接続など、情報とリソースを共有して通信できる、相互接続されたデバイスのシステムです。これらの各デバイスには、プロトコルと呼ばれる一連の規則を使用して 2 つ以上のデバイス間でメッセージを送受信する固有のインターネットプロトコル (IP) アドレスがあります。

ネットワーク通信のカテゴリー

IP ネットワーク

インターネットプロトコルアドレスを介して通信するネットワーク。IP ネットワークは、インターネットおよびほとんどの内部ネットワークに実装されています。イーサネット、ケーブルモデム、DSL モデム、ダイヤルアップモデム、無線ネットワーク、VPN 接続などがその代表的な例です。

非 IP ネットワーク

トランスポート層ではなく下位層を介して通信するのに使用されるネットワーク。このネットワークはほとんど使用されないことに注意してください。InfiniBand は非 IP ネットワークです。

1.2. 静的 IP アドレス指定と動的 IP アドレス指定の比較

静的 IP アドレス指定

デバイスに静的 IP アドレスが割り当てられている場合、そのアドレスは手動で変更しない限り、時間の経過とともに変わることはありません。静的 **IP** アドレス指定の使用が推奨されるのは、次のような場合です。

- **DNS** などのサーバーや認証サーバーのネットワークアドレスの整合性を確保する。
- 他のネットワークインフラストラクチャから独立して動作する、帯域外管理デバイスを使用する。
  「ネットワーク設定方法の選択」に列挙されるすべての設定ツールでは、静的 **IP** アドレスを手動で割り当てることができます。

動的 IP アドレス指定

デバイスに動的 IP アドレスが割り当てられている場合、そのアドレスは時間の経過とともに変わります。このため、マシンの再起動後には IP アドレスが変わる可能性があるので、随時ネットワークに接続するデバイスに推奨されます。
動的 IP アドレスは、より柔軟で、設定と管理が簡単です。Dynamic Host Control Protocol (**DHCP**) は、ネットワーク設定をホストに動的に割り当てる従来の方法です。

注記

静的 IP アドレスまたは動的 IP アドレスをどのような場合に使用するかを定義する厳密な規則はありません。ユーザーのニーズ、設定、およびネットワーク環境によって異なります。

1.3. DHCP クライアントの動作の設定

DHCP (Dynamic Host Configuration Protocol) クライアントは、クライアントがネットワークに接続するたびに、動的 IP アドレスと対応する設定情報を DHCP サーバーに要求します。

DHCP のタイムアウトの設定

dhcp クライアントは、**DHCP** 接続の開始時に、**DHCP** サーバーに IP アドレスを要求します。dhcp クライアントがこの要求が完了するのを待つ時間は、デフォルトで 45 秒です。この手順では、**nmcli** ツールを使用して **ipv4.dhcp-timeout** プロパティーを設定する方法、または **/etc/sysconfig/network-scripts/ifcfg- インフレーム**ファイルの **IPV4_DHCP_TIMEOUT** オプションを設定する方法を説明します。たとえば、次のように **nmcli** を使用します。

```
~]# nmcli connection modify eth1 ipv4.dhcp-timeout 10
```

この間にアドレスを取得できない場合、IPv4 設定は失敗します。接続全体が失敗する可能性もありますが、これは **ipv4.may-fail** プロパティーにより異なります。

- **ipv4.may-fail** が **yes** (デフォルト) に設定されている場合、接続の状態は IPv6 設定に依存します。
  a. IPv6 設定が有効であり、これが成功した場合、接続はアクティブになりますが、IPv4 設定は再試行できません。
  b. IPv6 設定が無効であるか、または設定されていない場合、接続は失敗します。
- **ipv4.may-fail** が **no** に設定されている場合、接続は非アクティブになります。この場合は、以下のようになります。
  a. 接続の **autoconnect** プロパティーが有効になっている場合、**NetworkManager** は、**autoconnect-retries** プロパティーに設定された回数だけ、接続のアクティベーションを再試行します。デフォルトでは 4 回です。
  b. それでも接続が dhcp アドレスを取得できない場合、自動アクティベーションは失敗します。
  5 分後に自動接続プロセスが再開され、dhcp クライアントが dhcp サーバーからのアドレスの取得を再試行することに注意してください。

リースの更新と有効期限

DHCP リースが正常に取得された後、**NetworkManager** は、一定期間 DHCP サーバーから受信したパラメーターでインターフェースを設定し、定期的にリースを更新しようとします。リースの更新期間が切れても更新できない場合、**NetworkManager** は最大 8 分までサーバーへの接続を試み続けます。他方の IP 設定 (IPv4 または IPv6) が成功した場合は、接続がアクティブである限り DHCP 要求が続きます。

1.3.1. DHCPv4 の永続化

起動時とリース更新プロセス実行時の両方で DHCPv4 を永続化するには、**ipv4.dhcp-timeout** プロパティーを、32 ビットの整数の最大値 (MAXINT32) である **2147483647** か、または次の **infinity** 値に設定します。

```
~]$ nmcli connection modify eth1 ipv4.dhcp-timeout infinity
```

その結果、**NetworkManager** による DHCP サーバーからのリースの取得または更新の試行は、成功するまで停止しません。

リース更新プロセス中にのみ DHCP の永続的な動作を保証するには、**/etc/sysconfig/network-scripts/ifcfg-ethX** 設定ファイルで、または次のように **nmcli** を使用して、**IPADDR** プロパティーに静的 IP を手動で追加します。

```
~]$ nmcli connection modify eth0 ipv4.address 192.168.122.88/24
```

IP アドレスのリース期限が切れると、静的 IP は、設定済みあるいは一部設定済みの IP 状態を保持します。IP アドレスを持つことはできますが、インターネットには接続されていません。

関連資料

- 「静的 IP アドレス指定と動的 IP アドレス指定の比較」
- 「nmcli の使用」

1.4. ワイヤレス規制ドメインの設定

Red Hat Enterprise Linux では、**crda** パッケージに、特定地区のワイヤレス規制ルールをカーネルに提供する Central Regulatory Domain Agent が含まれています。これは特定の **udev** スクリプトで使用するもので、**udev** スクリプトをデバッグしない限り手動で実行しないでください。カーネルは、新しい規制ドメインの変更にあたり、**udev** イベントを送信することで **crda** を実行します。規制ドメインの変更は、Linux ワイヤレスサブシステム (IEEE-802.11) がトリガーとなります。このサブシステムでは、規制データベース情報の維持に **regulatory.bin** ファイルを使用します。

システムの規制ドメインは、**setregdomain** ユーティリティーで設定します。**Setregdomain** は引数が不要であり、通常は管理者が手動で呼び出すのではなく、**udev** などのシステムスクリプトを介して呼び出します。国コードの検索に失敗する場合、システム管理者は **/etc/sysconfig/regdomain** ファイルで **COUNTRY** 環境変数を定義できます。

関連資料
規制ドメインの詳細は、次の man ページを参照してください。

- **setregdomain(1)** の man ページ - 国コードに基づき規制ドメインを設定します。
- **crda(8)** の man ページ - 特定 ISO または IEC 3166 alpha2 のワイヤレス規制ドメインをカーネルに送信します。
- **regulatory.bin(5)** の man ページ - Linux ワイヤレス規制データベースを表示します。
- **iw(8)** の man ページ - ワイヤレスデバイスおよびその設定を表示または操作します。

1.5. NETCONSOLE の設定

netconsole カーネルモジュールにより、カーネルメッセージをネットワークを通じて他のコンピューターに記録することができます。これにより、ディスクへの記録に失敗した場合やシリアルコンソールを使用できない場合に、カーネルのデバッグを行うことができます。

前提条件
rsyslogd デーモンが 514/udp ポートをリッスンしている状態で、ネットワーク上に rsyslog サーバーを用意する必要があります。

手順

1. **/etc/rsyslog.conf** ファイルの **MODULES** セクションにある以下の行をアンコメントし、514/udp ポートをリッスンしてネットワークからのメッセージを受信するように **rsyslogd** デーモンを設定します。

```
$ModLoad imudp
$UDPServerRun 514
```

2. 変更を有効にするために **rsyslogd** サービスを再起動します。

```
]# systemctl restart rsyslog
```

3. **rsyslogd** が 514/udp ポートをリッスンしていることを確認します。

```
]# netstat -l | grep syslog
udp        0      0 0.0.0.0:syslog          0.0.0.0:*
udp6       0      0 [::]:syslog             [::]:*
```

netstat -l 出力の **0.0.0.0:syslog** および **[::]:syslog** の値は、**rsyslogd** が **/etc/services** ファイルで定義されたデフォルトの **netconsole** ポートをリッスンしていることを示しています。

```
]$ cat /etc/services | grep syslog
syslog          514/udp
syslog-conn     601/tcp              # Reliable Syslog Service
syslog-conn     601/udp              # Reliable Syslog Service
syslog-tls      6514/tcp             # Syslog over TLS
syslog-tls      6514/udp             # Syslog over TLS
syslog-tls      6514/dccp            # Syslog over TLS
```

Netconsole は、**/etc/sysconfig/netconsole** ファイルを使用して設定されます。このファイルは、**initscripts** パッケージの一部です。このパッケージはデフォルトでインストールされ、**netconsole** サービスも提供します。

注記

デフォルトでは、**rsyslogd** と **netconsole.service** はポート 514 を使用します。別のポートを使用するには、**/etc/rsyslog.conf** の次の行を必要なポート番号に変更します。

```
$UDPServerRun <PORT>
```

関連資料

netconsole の設定の詳細およびトラブルシューティングのヒントは、Netconsole カーネルのドキュメント を参照してください。

1.6. 送信元マシンの設定

この手順では、送信元マシンを設定する方法について説明します。

前提条件

「netconsole の設定」

手順

1. **/etc/sysconfig/netconsole** ファイルの **SYSLOGADDR** 変数の値を、**syslogd** サーバーの IP ア

ドレスと一致するように設定します。以下に例を示します。

```
SYSLOGADDR=192.168.0.1
```

2. 変更を有効にするために **netconsole** サービスを再起動します。

```
]# systemctl restart netconsole.service
```

3. システムを再起動した後に **netconsole.service** を実行できるようにします。

```
]# systemctl enable netconsole.service
```

4. クライアントからの **netconsole** メッセージを **/var/log/messages** ファイル (デフォルト) または **rsyslog.conf** で指定されたファイルで表示します。

```
]# cat /var/log/messages
```

注記

デフォルトでは、**rsyslogd** と **netconsole.service** はポート 514 を使用します。別のポートを使用するには、**/etc/rsyslog.conf** の次の行を必要なポート番号に変更します。

```
$UDPServerRun <PORT>
```

送信元マシンで、**/etc/sysconfig/netconsole** ファイルの以下の行をアンコメントして編集します。

```
SYSLOGPORT=514
```

関連資料

netconsole の設定の詳細およびトラブルシューティングのヒントは、Netconsole カーネルのドキュメント を参照してください。

1.7. SYSCTL によるネットワークカーネル調整パラメーターの使用

sysctl ユーティリティーを介して特定のカーネル調整パラメーターを使用すると、実行中のシステムでネットワーク設定を調整し、ネットワークパフォーマンスに直接影響を与えることができます。

ネットワーク設定を変更するには、**sysctl** コマンドを使用します。システムを再起動しても維持される永続的な変更を行うには、**/etc/sysctl.conf** ファイルにそれを設定する行を追加します。

利用可能なすべての **sysctl** パラメーターを表示するには、**root** で以下を実行します。

```
~]# sysctl -a
```

1.8. NCAT ユーティリティーを使用したデータの管理

ncat ネットワークユーティリティーは、Red Hat Enterprise Linux 7 で **netcat** に置き換わりました。**ncat** は信頼できるバックエンドツールで、他のアプリケーションやユーザーにネットワーク接続を提供します。ネットワークをまたいでコマンドラインからデータを読み書きし、Transmission Control Protocol (TCP)、User Datagram Protocol (UDP)、Stream Control Transmission Protocol

(SCTP)、または Unix ソケットを使用して通信します。**ncat** は、**IPv4** と **IPv6** の両方を処理し、接続を開き、パケットを送信し、ポートスキャンを実行することが可能であり、また **SSL** やコネクションブローカーなどの高度な機能をサポートします。

同じオプションを使用して、**nc** コマンドも **ncat** として入力できます。ncat オプションの詳細は、**『移行計画ガイド』の「新しいネットワーキングユーティリティ (ncat)」セクション** と、man ページの **ncat(1)** を参照してください。

ncat のインストール
ncat パッケージをインストールするには、**root** で以下を実行します。

```
~]# yum install nmap-ncat
```

ncat ユースケースの簡単な例

例1.1 クライアントとサーバー間の通信の有効化

1. TCP ポート 8080 で接続を待機するように、クライアントマシンを設定します。

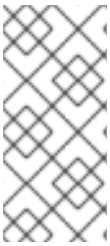
   ```
   ~]$ ncat -l 8080
   ```

2. サーバーマシンで、クライアントの IP アドレスを指定し、同じポート番号を使用します。

   ```
   ~]$ ncat 10.0.11.60 8080
   ```

   接続のどちら側でもメッセージを送信でき、それらはローカルマシンとリモートマシンの両方に表示されます。

3. **Ctrl+D** を押して、TCP 接続を閉じます。



注記

UDP ポートを確認するには、同じ **nc** コマンドで **–u** オプションを使用します。例を示します。

```
~]$ ncat -u -l 8080
```

例1.2 ファイルの送信

前の例で説明したように情報を画面に表示するのではなく、すべての情報をファイルに送信できます。たとえば、TCP ポート 8080 を介してクライアントからサーバーにファイルを送信するには、次の手順を実行します。

1. クライアントマシンで、ファイルをサーバーマシンに転送する特定のポートをリッスンするには、以下を実行します。

   ```
   ~]$  ncat -l 8080 > outputfile
   ```

2. サーバーマシンで、クライアントの IP アドレス、ポート、および転送するファイルを指定するには、以下を実行します。

   ```
   ~]$  ncat -l 10.0.11.60 8080 < inputfile
   ```

ファイルが転送された後、接続は自動的に閉じます。

注記

他の対象にファイルを転送することもできます。

```
~]$  ncat -l 8080 < inputfile
```

```
~]$  ncat -l 10.0.11.60 8080 > outputfile
```

例1.3 HTTP プロキシーサーバーの作成

localhost ポート 8080 に HTTP プロキシーサーバーを作成するには、以下を実行します。

```
~]$  ncat -l --proxy-type http localhost 8080
```

例1.4 ポートスキャン

開いているポートを確認するには、**-z** オプションを使用して、スキャンするポートの範囲を指定します。

```
~]$  ncat -z 10.0.11.60 80-90
   Connection to 192.168.0.1 80 port [tcp/http] succeeded!
```

例1.5 SSL を使用した安全なクライアントサーバー通信の設定

サーバーに **SSL** を設定します。

```
~]$ ncat -e /bin/bash -k -l 8080 --ssl
```

クライアントマシンを設定します。

```
~]$ ncat --ssl 10.0.11.60 8080
```

注記

SSL 接続の完全な機密性を保証するには、サーバーでは **--ssl-cert** と **--ssl-key** オプションが、またクライアントでは **--ssl-verify** と **--ssl-trustfile** オプションが必要となります。

関連資料
他の例は、man ページの **ncat(1)** を参照してください。

第2章 NETWORKMANAGER を使用したネットワーク管理の開始

2.1. NETWORKMANAGER の概要

Red Hat Enterprise Linux 8 では、デフォルトのネットワークサービスである **NetworkManager** を使用します。これは動的ネットワークを制御および設定するデーモンで、使用可能な場合にはネットワークデバイスと接続が起動してアクティブな状態を維持します。従来の **ifcfg** タイプの設定ファイルは引き続きサポートされます。

各ネットワークデバイスは、**NetworkManager** デバイスに対応します。ネットワークデバイスの設定は、1つの **NetworkManager** 接続に完全に保存されます。**NetworkManager** 接続を **NetworkManager** デバイスに適用して、ネットワーク設定を実行できます。

2.1.1. NetworkManager を使用する利点

NetworkManager を使用する主な利点は、次の通りです。

- ネットワーク管理が容易になります。**NetworkManager** はネットワーク接続を確実に機能させます。システムにネットワーク設定はないがネットワークデバイスがあることを検出すると、**NetworkManager** は一時的な接続を作成して接続を提供します。
- ユーザーの接続設定が簡単になります。**NetworkManager** は、**GUI**、**nmtui**、**nmcli** など、さまざまなツールでの管理を提供します。
- 柔軟な設定に対応します。たとえば、WiFi インターフェースを設定すると、**NetworkManager** は使用可能な wifi ネットワークをスキャンして表示します。インターフェースを選択することができ、**NetworkManager** は再起動プロセス後の自動接続を提供するのに必要な資格情報を表示します。**NetworkManager** では、ネットワークエイリアス、IP アドレス、静的ルート、DNS 情報、VPN 接続の他、接続固有の多くのパラメーターを設定できます。設定オプションは必要に合わせて修正できます。
- ネットワーク設定と状態についてアプリケーションによるクエリーと制御を可能にする、D-Bus を介した API を提供します。この方法により、アプリケーションは D-BUS を介してネットワークを確認し、制御できます。たとえば、Web ベースインターフェース **Cockpit** は、Web ブラウザーを経由でサーバーを監視して設定するもので、**NetworkManager** D-BUS インターフェースを使用してネットワークを設定します。
- 再起動プロセス後もデバイスの状態を維持し、再起動中に管理モードに設定されているインターフェースを引き継ぎます。
- 明示的に管理対象外として設定されていないが、ユーザーまたは他のネットワークサービスによって手動で制御されているデバイスを処理します。

関連資料

- 「NetworkManager のツール」
- 「nmtui の使用」
- 「nmcli の使用」]
- RHEL 8 Web コンソールのインストールおよび使用に関する詳細は、Managing systems using the RHEL 8 web cosnole を参照してください。

2.2. NETWORKMANAGER のインストール

NetworkManager は、Red Hat Enterprise Linux 8 にはデフォルトでインストールされます。インストールされない場合は、**root** で以下のコマンドを入力します。

```
~]# yum install NetworkManager
```

関連資料

- 「NetworkManager の概要」
- 「NetworkManager を使用する利点」

2.3. NETWORKMANAGER のステータスの確認

NetworkManager が起動しているかどうかを確認するには、以下のコマンドを実行します。

```
~]$ systemctl status NetworkManager
NetworkManager.service - Network Manager
   Loaded: loaded (/lib/systemd/system/NetworkManager.service; enabled)
   Active: active (running) since Fri, 08 Mar 2013 12:50:04 +0100; 3 days ago
```

NetworkManager が起動していない場合は、**systemctl status** コマンドにより **Active: inactive (dead)** と表示されることに注意してください。

2.4. NETWORKMANAGER の開始

NetworkManager を開始するには、以下のコマンドを実行します。

```
~]# systemctl start NetworkManager
```

起動時に **NetworkManager** を自動的に有効にするには、以下のコマンドを実行します。

```
~]# systemctl enable NetworkManager
```

2.5. NETWORKMANAGER のツール

表2.1 NetworkManager のツールとアプリケーションの概要

アプリケーションおよびツール	説明
nmcli	コマンドラインツール。ユーザーとスクリプトが **NetworkManager** で対話できるようにします。**nmcli** を、サーバーなど GUI のないシステムで使用して、**NetworkManager** の全要素を制御できる点に注意。GUI ツールのようにさらに高度な機能を提供します。
nmtui	**NetworkManager** 向け curses ベースの単純なテキストユーザーインターフェース (TUI)。

アプリケーションおよびツール	説明
nm-connection-editor	ボンドの設定や接続のチーミングなど、まだ **control-center** ユーティリティーが対応していない特定のタスク用のグラフィカルユーザーインターフェース。**NetworkManager** で保存されたネットワーク接続を追加、削除、変更できます。これを開始するには、ターミナルに **nm-connection-editor** を入力します。
control-center	デスクトップユーザーが使用するユーザーインターフェース。GNOME Shell が提供し、ネットワーク設定ツールが含まれます。**Super** キーを押してアクティビティ画面を開き、**Network** と入力してから **Enter** キーを押すと開始して、ネットワーク設定ツールが表示されます。
ネットワーク接続アイコン	GNOME Shell が提供するグラフィカルユーザーインターフェースで、**NetworkManager** から報告されたネットワーク接続の状態を示します。アイコンには複数の状態があり、現在使用中の接続の種類を視覚的に表示します。

関連資料

- 「nmtui の使用」
- 「nmcli の使用」

2.6. DISPATCHER スクリプトの実行

NetworkManager には、接続状態に基づいてサービスを開始または停止する追加のカスタムスクリプトを実行する方法があります。デフォルトで **/etc/NetworkManager/dispatcher.d/** ディレクトリーが存在し、**NetworkManager** はそこでアルファベット順にスクリプトを実行します。各スクリプトは、**root** が**所有する**実行可能ファイルであること、また**書き込み権限**はファイル所有者のみが持っていることが必要です。

関連資料

- NetworkManager の dispatcher スクリプトの実行に関する詳細は、Red Hat ナレッジベースのソリューション「ethtool コマンドを適用するように NetworkManager の dispatcher スクリプトを記述する」を参照してください。

2.7. SYSCONFIG ファイルによる NETWORKMANAGER の使用

/etc/sysconfig/ ディレクトリーは、設定ファイルとスクリプト用の場所です。ほとんどのネットワーク設定情報がここに保存されます。例外は VPN、モバイルブロードバンド、および PPPoE の設定で、これらは **/etc/NetworkManager/** サブディレクトリーに保存されます。たとえば、インターフェース固有の情報は、**/etc/sysconfig/network-scripts/** ディレクトリーの **ifcfg** ファイルに保存されます。

グローバル設定には、**/etc/sysconfig/network** ファイルを使用します。VPN、モバイルブロードバンド、および PPPoE 接続の情報は、**/etc/NetworkManager/system-connections/** に保存されます。

Red Hat Enterprise Linux 8 では、**ifcfg** ファイルを編集しても、**NetworkManager** は自動的に変更を認識しないため、変更を通知する必要があります。**NetworkManager** のプロファイル設定を更新するツールを使用している場合、**NetworkManager** は、そのプロファイルを使用して再接続するまで変更を実行しません。たとえば、エディターを使用して設定ファイルを変更すると、**NetworkManager** がその設定ファイルを再度読み込む必要があります。

これを確実に行うには、**root** で以下を入力し、すべての接続プロファイルを再読み込みします。

```
~]# nmcli connection reload
```

または、変更したファイル **ifcfg-ifname** を **1つだけ** 再読み込みするには、以下を実行します。

```
~]# nmcli con load /etc/sysconfig/network-scripts/ifcfg-ifname
```

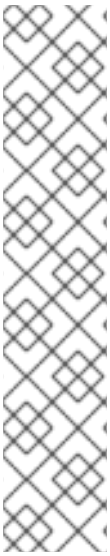
上記のコマンドを使用して複数のファイル名を指定できることに注意してください。

変更後に接続を再開するには、次のコマンドを使用します。

```
~]# nmcli con up connection-name
```

2.7.1. 従来のネットワークスクリプトのサポート

Red Hat Enterprise Linux 8 では、従来のネットワークスクリプトは非推奨となっており、デフォルトでは提供されません。基本インストールでは、**nmcli** ツールを介して **NetworkManager** を呼び出す **ifup** スクリプトおよび **ifdown** スクリプトの新しいバージョンが提供されます。Red Hat Enterprise Linux 8 で **ifup** スクリプトおよび **ifdown** スクリプトを実行するには、**NetworkManager** が起動している必要があります。



注記

/sbin/ifup-local スクリプト、**ifdown-pre-local** スクリプト、および**ifdown-local** スクリプトのカスタムコマンドは実行されません。

このスクリプトが必要な場合は、次のコマンドを使用すれば、システム内に非推奨のネットワークスクリプトをインストールできます。

```
~]# yum install network-scripts
```

ifup スクリプトおよび **ifdown** スクリプトが、インストールされた従来のネットワークスクリプトにリンクします。

従来のネットワークスクリプトを呼び出すと、そのスクリプトが非推奨であることを示す警告が表示されます。

関連資料

- **NetworkManager(8)** の man ページ - ネットワーク管理デーモンが説明されています。
- **NetworkManager.conf(5)** の man ページ - **NetworkManager** 設定ファイルが説明されています。
- **/usr/share/doc/initscripts/sysconfig.txt** - 従来のネットワークサービスで使用される **ifcfg** 設定ファイルとそのディレクトリーが説明されています。

- **ifcfg(8)** の man ページ - **ifcfg** コマンドついて簡単に説明されています。

第3章 ネットワーク設定方法の概要

本章では、Red Hat Enterprise Linux 8 で利用可能なネットワーク設定方法の概要を説明します。

3.1. ネットワーク設定方法の選択

- **NetworkManager** を使用してネットワークインターフェースを設定するには、次のいずかのツールを使用します。
  - テキストユーザーインターフェースツールの **nmtui**
  - コマンドラインツールの **nmcli**
  - グラフィカルユーザーインターフェースツールの **GNOME GUI**
- **NetworkManager** を使用**せずに**ネットワークインターフェースを設定する方法は、次のとおりです。
  - **ifcfg** ファイルを手動で編集する。
  - **ip** コマンドを使用する。インターフェースに IP アドレスを割り当てるために使用できますが、変更は再起動をまたぐ永続的なものではありません。再起動すると一切の変更は失われます。
- root ファイルシステムがローカルに **ない**場合にネットワーク設定を行う方法は、次のとおりです。
  - カーネルのコマンドラインを使用する。

関連資料

- 「nmtui の使用」
- 「nmcli の使用」

第4章 NMTUI による IP ネットワークの設定

本章では、**NetworkManager** のツールである **nmtui** を使用し、ネットワークインターフェースを設定する方法を説明します。

4.1. NMTUI の使用

nmtui は、**NetworkManager** 用のシンプルな curses ベースのテキストユーザーインターフェース (TUI) です。

この手順では、テキストユーザーインターフェースツール **nmtui** の起動方法について説明します。

前提条件

- 端末ウィンドウで **nmtui** ツールを使用する。**NetworkManager-tui** パッケージに含まれていますが、デフォルトで **NetworkManager** と一緒にインストールされるものではありません。**NetworkManager-tui** をインストールするには、次のコマンドを実行します。

```
~]# yum install NetworkManager-tui
```

- NetworkManager が起動していることを確認するには、「NetworkManager のステータスの確認」 を参照してください。

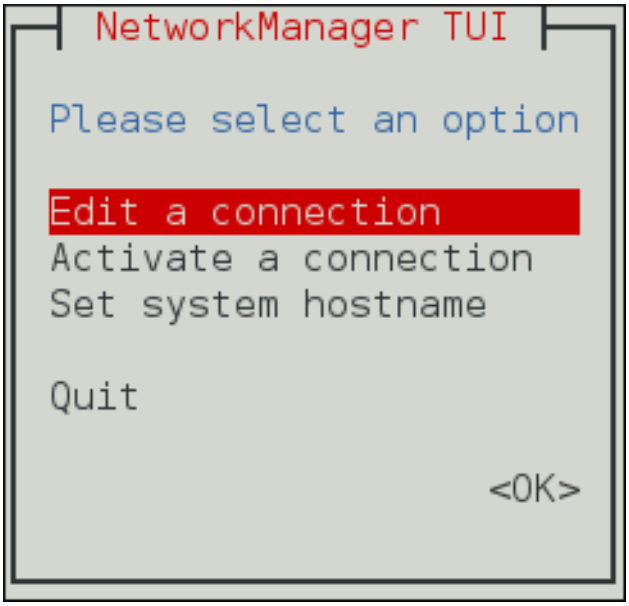
手順

1. **nmtui** ツールを起動します。

```
~]$ nmtui
```

テキストユーザーインターフェースが表示されます。

図4.1 NetworkManager のテキストユーザーインターフェースの開始メニュー



2. 移動するには矢印キーを使用するか、**Tab** を押して次に進むか **Shift**+**Tab** を押して前に戻ります。**Enter** を押してオプションを選びます。**Space** バーは、チェックボックスのステータスを切り替えます。

4.1.1. nmtui による接続の編集

前提条件

- 「nmtui の使用」

nmtui を使用して接続を編集するには、**NetworkManager TUI** メニューで **Edit a connection** オプションを選択し、**Enter** を押します。

4.1.2. nmtui による編集済み接続への変更の適用

すでにアクティブな接続に加えた変更を適用するには、接続の再アクティブ化が必要です。この場合は、以下の手順を実施します。

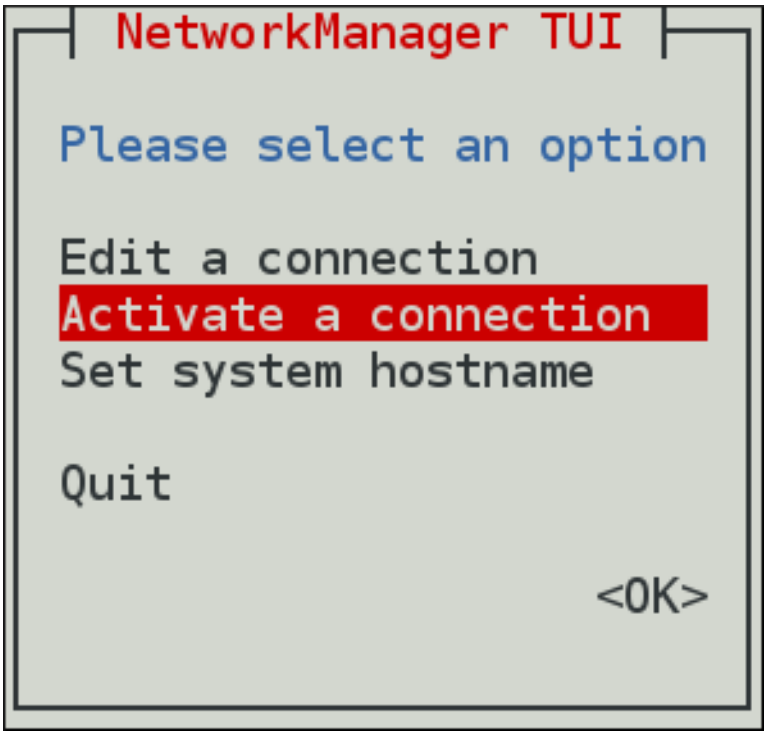
前提条件

- 「nmtui の使用」

手順

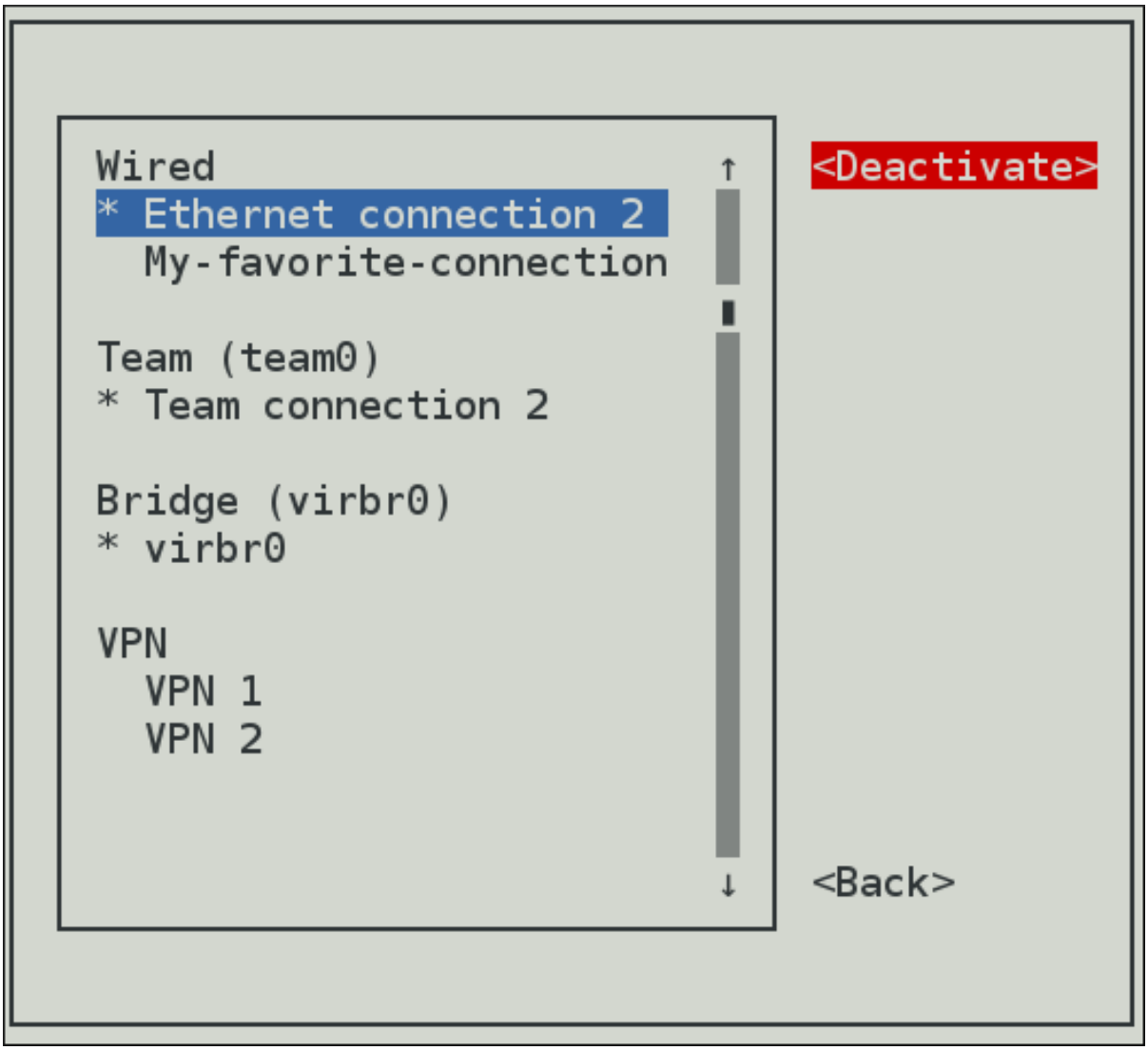
1. メニュー項目の **Activate a connection** を選択します。

   図4.2 nmtui による接続のアクティブ化



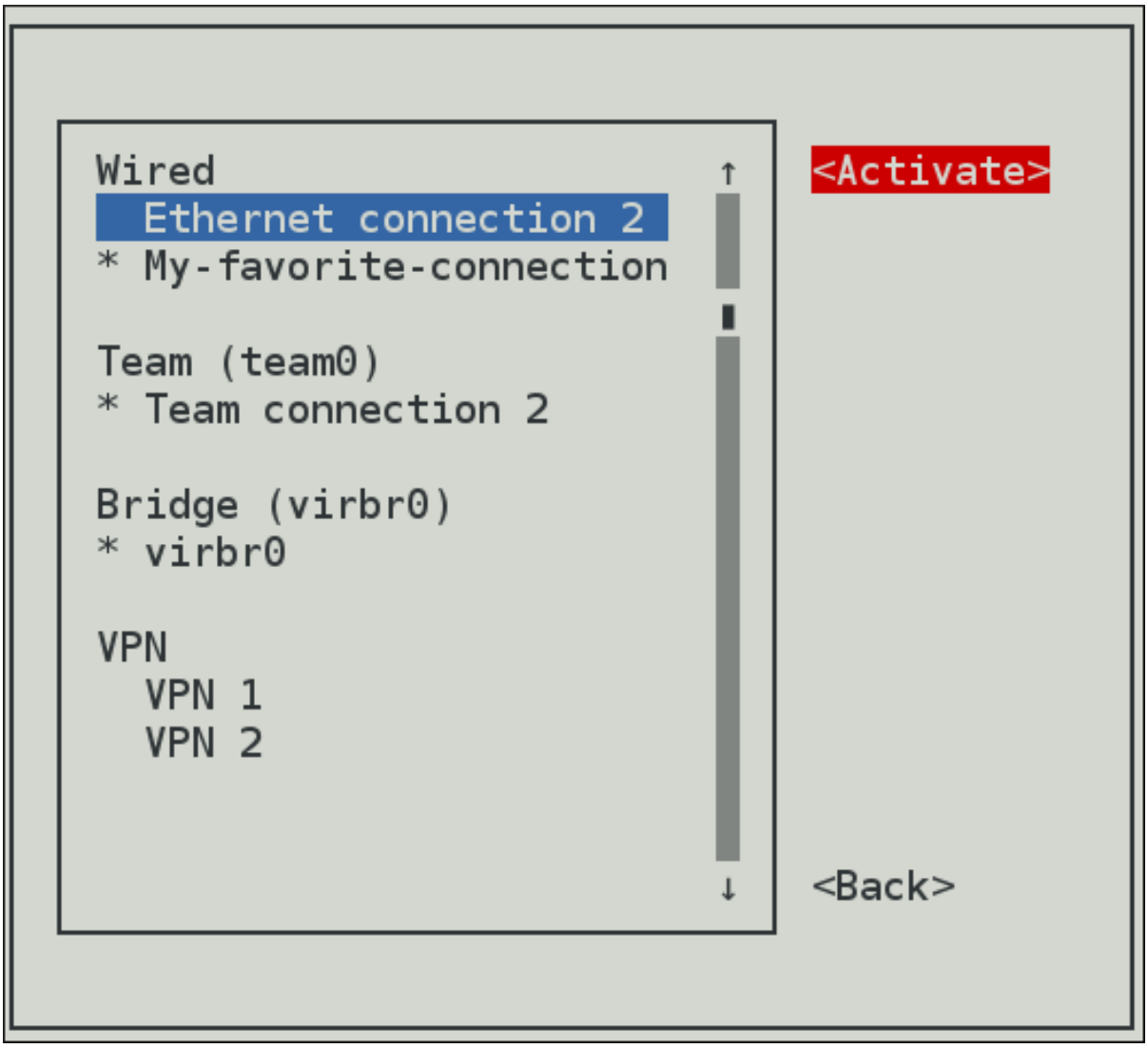
2. 修正した接続を選択します。右側の **Deactivate** ボタンをクリックします。

図4.3 nmtui による修正した接続の非アクティブ化



3. 接続を再度選択し、**Activate** ボタンをクリックします。

図4.4 nmtui による修正した接続の再アクティブ化



以下のコマンドも利用できます。

```
~]$ nmtui edit connection-name
```

接続名を指定していないと、選択メニューが表示されます。接続名が指定され、正しく特定されると、関連する **Edit connection (接続の編集)** 画面が表示されます。

```
~]$ nmtui connect connection-name
```

接続名を指定していないと、選択メニューが表示されます。接続名が指定されて正しく特定されると、関連する接続がアクティブ化されます。無効なコマンドがあると、使用方法に関するメッセージが表示されます。

nmtui は、すべての種類の接続をサポートしているわけではない点に注意してください。特に、VPN、WPA Enterprise を使用したワイヤレスネットワーク接続、**802.1X** を使用したイーサネット接続は編集できません。

関連資料

- **NetworkManager** のツールに関する詳細は、「NetworkManager のツール」 を参照してください。

第5章 NMCLI によるネットワークの設定

本章では、**nmcli** を使用してネットワークインターフェースを設定する方法を説明します。

5.1. NMCLI の使用

nmcli (NetworkManager Command Line Interface) は、 **NetworkManager** を介してネットワークを設定する、コマンドラインユーティリティーです。**nmcli** は、ネットワーク接続の作成、表示、編集、削除、有効化、無効化のほか、ネットワークデバイスのステータスの制御や表示に使用します。

nmcli ユーティリティーは、ユーザーとスクリプトの両方から使用できます。

- サーバー、ヘッドレスマシン、ターミナルの場合は、**nmcli** を使用すると、GUI なしで **NetworkManager** ディレクトリーを制御できます。
- スクリプトの場合、**nmcli** は、出力をよりスクリプト処理に適した形式に変更するオプションをサポートします。

各ネットワークデバイスは、**NetworkManager** デバイスに対応します。ネットワークデバイスの設定は、1 つの **NetworkManager** 接続に完全に保存されます。**NetworkManager** 接続を **NetworkManager** デバイスに適用して、ネットワーク設定を実行できます。

nmcli を使用するのに最も一般的な nmcli コマンドは、**nmcli device** と **nmcli connection** です。

- **nmcli device** コマンドは、システムで使用可能なネットワークデバイスを一覧表示します。

NetworkManager デバイスを表示するには、次のコマンドを実行します。

```
~]$ nmcli device
```

デバイスは以下の可能性があります。

1. **managed** - NetworkManager の管理下にある。**managed** デバイスは、**connected** (アクティブ化かつ設定済み) の場合と、**disconnected** (接続されていないが再アクティブ化の準備はできている) の場合とがある。
2. **unmanaged** - NetworkManager では管理していない。

managed または **unmanaged** デバイスの設定に関する詳細は、「nmcli による管理対象または管理対象外のデバイスの設定」 を参照してください。

nmcli device コマンドは、多くの引数を取ることができます。最も注目すべきものは、**status**、**show**、**set**、**connect**、**disconnect**、**modify**、**delete**、および **wifi** です。全リストを確認する場合は、**nmcli device help** コマンドを入力してください。

- **nmcli connection** コマンドは、NetworkManager で利用可能な接続のプロファイルの一覧を表示します。

NetworkManager の接続を表示するには、次のコマンドを実行します。

```
~]$ nmcli connection
```

アクティブな接続はすべて、リストの一番上に緑色で表示されます。非アクティブな接続は白で表示されます。DEVICE フィールドは、接続が適用されるデバイスを識別します。

nmcli connection コマンドは、接続プロファイルを管理するために多くの引数をとることができます。最も注目すべきものは、**show**、**up**、**down**、**add**、**modify**、および **delete** です。全リストを確認する場合は、**nmcli connection help** コマンドを入力してください。

重要

nmcli コマンドを使用する場合は、nmcli コマンドの一部を入力して **Tab** キーを押し、オートコンプリート機能でコマンドシーケンスを補完することをお勧めします。複数の補完候補がある場合は、**Tab** で一覧が表示されます。これにより、ユーザーはコマンド入力を速く簡単に行えるようになります。**nmcli** のオートコンプリート機能を有効にするには、**bash-completion** パッケージを忘れずにインストールしてください。

```
~]# yum install bash-completion
```

パッケージのインストール後、次回コンソールにログインしたときに **nmcli オートコンプリート**が利用できるようになります。これを現在のコンソールでも有効にするには、次のコマンドを入力します。

```
~]$ source /etc/profile.d/bash_completion.sh
```

nmcli を使用した基本的な書式は、以下のとおりです。

```
nmcli [OPTIONS] OBJECT { COMMAND | help }
```

- ここでの [OPTIONS] は、次のような任意のオプションになります。

  -t (terse)

  このモードは、コンピューターのスクリプト処理に使用される場合があり、値だけを表示する簡潔な出力を確認できます。

  例5.1 簡潔な出力の表示

  ```
  ~]$ nmcli -t device
  ens3:ethernet:connected:Profile 1
  lo:loopback:unmanaged:
  ```

  -f (field)

  このオプションでは、どのフィールドを出力に表示できるかを指定します。たとえば、NAME、UUID、TYPE、AUTOCONNECT、ACTIVE、DEVICE、STATE などです。フィールドは、1 つまたは複数使用できます。複数のフィールドを使用する場合は、フィールドを区切るコンマの後にスペースを入れないでください。

  例5.2 出力内のフィールドの指定

  ```
  ~]$ nmcli -f DEVICE,TYPE device
  DEVICE  TYPE
  ens3    ethernet
  lo      loopback
  ```

  また、次のようなスクリプトの記述に適しています。

```
~]$ nmcli -t -f DEVICE,TYPE device
ens3:ethernet
lo:loopback
```

-p (pretty)

このオプションでは、**nmcli** により人間が理解可能な出力を生成します。たとえば、値を揃え、ヘッダーを表示します。

例5.3 pretty モードで出力の表示

```
~]$ nmcli -p device
====================
 Status of devices
====================
DEVICE  TYPE      STATE      CONNECTION
------------------------------------------------------------
ens3    ethernet  connected  Profile 1
lo      loopback  unmanaged  --
```

-h (help)

ヘルプ情報を表示します。

- ここでの OBJECT は、**general**、**networking**、 **radio**、 **connection**、**device**、**agent**、**monitor** のいずれかのオプションになります。

注記

コマンドでは、上記のオプションの接頭辞を使用できます。たとえば、**nmcli con help**、**nmcli c help**、**nmcli connection help** は、同じ出力を生成します。

- ここでの COMMAND は、必須の **nmcli** コマンドです。
- この **help** では、次のように、指定されたオブジェクトに関連して使用可能なアクションを一覧表示します。

```
~]$ nmcli OBJECT help
```

以下に例を示します。

```
~]$ nmcli c help
```

関連資料

- 「NetworkManager のツール」
- **nmcli(1)** の man ページ
- 「**nmcli** コマンドの簡単な例」
- 「nmcli による接続プロファイルの作成および修正」

5.2. NMCLI のプロパティー名とエイリアスの概要

前提条件

プロパティー名は、NetworkManager が一般的なオプションの識別に使用する特定の名前である。重要な nmcli **プロパティー**名の例を以下に示します。

connection.type

特定の接続のタイプ。指定できる値は、**adsl**、**bond**、**bond-slave**、**bridge**、**bridge-slave**、**bluetooth**、**cdma**、**ethernet**、**gsm**、**infiniband**、**olpc-mesh**、**team**、**team-slave**、**vlan**、**wifi**、**wimax** です。各接続タイプには、タイプ固有のコマンドオプションがあります。**TYPE_SPECIFIC_OPTIONS** の一覧は、man ページの **nmcli(1)** で確認できます。たとえば、**gsm** 接続では、アクセスポイント名を **apn** に指定しておく必要があります。**wifi** デバイスは、サービスセット識別子を **ssid** に指定しておく必要があります。

connection.interface-name

接続に関連するデバイス名。**eth0** など。

connection.id

接続プロファイルに使用される名前。接続名を指定しないと、次のように接続名が生成されます。

> **connection.type -connection.interface-name**

connection.id は **接続プロファイル** の名前。デバイスを表すインターフェース名 (**wlan0**、**ens3**、**em1** など) と混同しないようにしてください。なお、ユーザーはインターフェースと同じ名前を接続に付けることができますが、これらは別のものです。1 つのデバイスに複数の接続プロファイルを利用することもできます。これは、モバイルデバイスの場合や異なるデバイス間でネットワークケーブルを切り替える場合に非常に便利です。必要に応じて、設定を編集するのではなく、異なるプロファイルを作成してインターフェースに適用します。**id** オプションも接続プロファイル名を参照します。

show、**up**、**down** などの nmcli コマンドで最も重要なオプションを以下に示します。

id

ユーザーが接続プロファイルに割り当てる識別用文字列。**nmcli** connection コマンドで、ID を使用して接続を指定できます。コマンド出力の NAME フィールドには、必ず 接続 ID が表示されます。con-name が参照するのと同じ接続プロファイル名が参照されます。

uuid

システムが接続プロファイルに割り当てる一意の識別用文字列。**nmcli connection** コマンドで、**uuid** を使用して接続を指定できます。

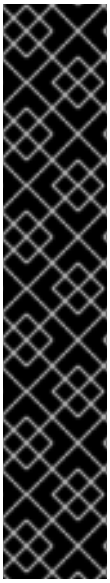
エイリアスとプロパティー名

エイリアスは、 **プロパティー**名の別名です。エイリアスは、**nmcli** で内部的にプロパティーに変換されます。**エイリアス** の方が読みやすいですが、一般に**プロパティー名**が利用されます。両者は交互に使用可能です。

エイリアス	例	プロパティー	例	定義

エイリアス	例	プロパティー	例	定義
type	**type bond**	connection.type	**connection.type bond**	特定の接続のタイプ。接続タイプの例 - **bond**、**bridge**、**ethernet**、**wifi**、**infiniband**、**vlan**
ifname	**ifname eth0**	connection.interface-name	**connection.interface-name eth0**	接続が属するデバイスの名前
con-name	**con-name "My Connection"**	connection.id	**connection.id "My Connection"**	接続名

5.3. NMCLI コマンドの簡単な例



重要

nmcli コマンドを使用する場合は、**nmcli** コマンドの一部を入力して **Tab** キーを押し、オートコンプリート機能でコマンドシーケンスを補完することをお勧めします。複数の補完候補がある場合は、**Tab** で一覧が表示されます。これにより、ユーザーはコマンド入力を速く簡単に行えるようになります。**nmcli** のオートコンプリート機能を有効にするには、**bash-completion** パッケージを忘れずにインストールしてください。

```
~]# yum install bash-completion
```

パッケージのインストール後、次回コンソールにログインしたときに **nmcli オートコンプリート**が利用できるようになります。これを現在のコンソールでも有効にするには、次のコマンドを入力します。

```
~]$ source /etc/profile.d/bash_completion.sh
```

特定のユースケースにおける **nmcli** の使用例を、以下に示します。

例5.4 すべての接続の表示

```
~]$ nmcli connection show
  NAME       UUID                                  TYPE      DEVICE
Profile 1  db1060e9-c164-476f-b2b5-caec62dc1b05  ethernet    ens3
bond0      aaf6eb56-73e5-4746-9037-eed42caa8a65  ethernet    --
```

例5.5 現在アクティブな接続のみを表示

```
~]$ nmcli connection show --active
  NAME       UUID                                  TYPE      DEVICE
Profile 1  db1060e9-c164-476f-b2b5-caec62dc1b05  ethernet    ens3
```

例5.6 接続のアクティブ化

接続をアクティブ化するには、**up** 引数を使用します。

```
~]$ nmcli connection show
  NAME       UUID                                  TYPE      DEVICE
Profile 1  db1060e9-c164-476f-b2b5-caec62dc1b05  ethernet    ens3
bond0      aaf6eb56-73e5-4746-9037-eed42caa8a65  ethernet    --
```

```
~]$ nmcli connection up id bond0
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/4)
```

```
~]$ nmcli connection show
NAME       UUID                                  TYPE      DEVICE
Profile 1  db1060e9-c164-476f-b2b5-caec62dc1b05  ethernet    ens3
bond0      aaf6eb56-73e5-4746-9037-eed42caa8a65  ethernet   bond0
```

例5.7 特定のアクティブな接続の非アクティブ化

特定のアクティブな接続を非アクティブ化するには、**down** 引数を使用します。

```
~]$ nmcli connection down id bond0
```

```
~]$ nmcli connection show
  NAME       UUID                                  TYPE      DEVICE
Profile 1  db1060e9-c164-476f-b2b5-caec62dc1b05  ethernet    ens3
bond0      aaf6eb56-73e5-4746-9037-eed42caa8a65  ethernet    --
```

例5.8 自動的に再起動できないようにデバイスの接続解除

```
~]$ nmcli device disconnect id bond0
```

注記

nmcli connection down コマンドはデバイスの接続を解除しますが、その後でデバイスが接続を自動的にアクティブ化するのを妨げるものではありません。**nmcli device disconnect** コマンドは、デバイスの接続を解除し、手動で操作しない限り、その後デバイスが接続を自動的にアクティブ化しないようにします。接続の **connection.autoconnect** フラグが **yes** に設定されている場合、接続は切断されたデバイスで自動的に再開します。この場合、**nmcli connection down** コマンドではなく、**nmcli device disconnect** コマンドを使用します。

例5.9 NetworkManager で認識されたデバイスとその状態のみ表示

```
~]$ nmcli device status
DEVICE  TYPE      STATE      CONNECTION
```

```
ens3    ethernet  connected  Profile 1
lo      loopback  unmanaged  --
```

例5.10 デバイスの一般情報の表示

```
~]$ nmcli device show
GENERAL.DEVICE:                         ens3
GENERAL.TYPE:                           ethernet
GENERAL.HWADDR:                         52:54:00:0A:2F:ED
GENERAL.MTU:                            1500
GENERAL.STATE:                          100 (connected)
GENERAL.CONNECTION:                     ens3
[...]
```

例5.11 NetworkManager の全体的な状態の確認

```
~]$ nmcli general status
STATE      CONNECTIVITY  WIFI-HW  WIFI     WWAN-HW  WWAN
connected  full          enabled  enabled  enabled  enabled
```

簡潔モードの場合:

```
~]$ nmcli -t -f STATE general
connected
```

例5.12 NetworkManager のログ記録の状態の表示

```
~]$ 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,DEVICE,OL
PC,
WIMAX,INFINIBAND,FIREWALL,ADSL,BOND,VLAN,BRIDGE,DBUS_PROPS,TEAM,CONCHECK
,DC
B,DISPATCH
```

nmcli コマンドは、以下の省略形を使用することもできます。

表5.1 nmcli コマンドの省略形の例

nmcli コマンド	省略形
nmcli general status	nmcli g
nmcli general logging	nmcli g log

nmcli コマンド	省略形
nmcli connection show	nmcli con show または nmcli c
nmcli connection show --active	nmcli con show -a または nmcli c -a
nmcli device status	nmcli dev または nmcli d
nmcli device show	nmcli dev show または nmcli d show

関連資料

- **nmcli** オプションの包括的な一覧に関する詳細は、man ページの **nmcli(1)** を参照してください。
- その他の例は、man ページの **nmcli-examples(5)** を参照してください。
- 「nmcli による接続プロファイルの作成および修正」

5.4. NMCLI による管理対象または管理対象外のデバイスの設定

前提条件

- 「nmcli の使用」
- 「nmcli のプロパティー名とエイリアスの概要」

現在利用可能なネットワーク接続を一覧表示するには、以下を実行します。

```
~]$ nmcli con show
NAME            UUID                                  TYPE           DEVICE
Auto Ethernet   9b7f2511-5432-40ae-b091-af2457dfd988  802-3-ethernet  --
ens3            fb157a65-ad32-47ed-858c-102a48e064a2  802-3-ethernet  ens3
MyWiFi          91451385-4eb8-4080-8b82-720aab8328dd  802-11-wireless wlan0
```

出力の **NAME** フィールドは常に **connection ID** (名前) を示すことに留意してください。これはインターフェース名と同じように見えますが、異なるものです。上記の 2 つ目の接続では、**NAME** フィールドの **ens3** が、ユーザーがインターフェース **ens3** に適用されるプロファイルに指定した **connection ID** です。最後の接続では、ユーザーが接続 ID **MyWiFi** をインターフェース **wlan0** に割り当てています。

イーサネット接続を追加すると、設定プロファイルが作成され、それがデバイスに割り当てられます。新規プロファイルを作成する前に、以下のように利用可能なデバイスを確認します。

```
~]$ nmcli device status
DEVICE  TYPE      STATE         CONNECTION
ens3    ethernet  disconnected  --
ens9    ethernet  disconnected  --
lo      loopback  unmanaged     --
```

デバイスを **NetworkManager** の管理対象外に設定するには、以下を実行します。

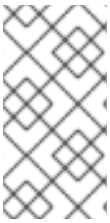
```
~]$ nmcli device set ifname managed no
```

eth2 を管理対象外に設定するには、次を実行します。

```
~]$ nmcli device status
DEVICE      TYPE      STATE      CONNECTION
bond0       bond      connected  bond0
virbr0      bridge    connected  virbr0
eth1        ethernet  connected  bond-slave-eth1
eth2        ethernet  connected  bond-slave-eth2
eth0        ethernet  unmanaged  --
```

```
~]$ nmcli device set eth2 managed no
```

```
~]$ nmcli device status
DEVICE      TYPE      STATE      CONNECTION
bond0       bond      connected  bond0
virbr0      bridge    connected  virbr0
eth1        ethernet  connected  bond-slave-eth1
eth2        ethernet  unmanaged  --
eth0        ethernet  unmanaged  --
```



注記

デバイスを管理対象外に設定すると、そのデバイスは NetworkManager で制御されません。設定しようとするデバイスが一覧に管理対象外として表示されている場合は、そのデバイスに対し、nmcli コマンドは一切の影響を与えません。ただし、デバイスは接続されたままです。

関連資料

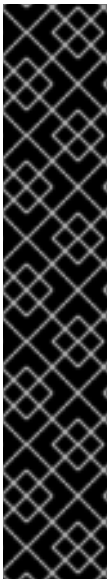
- 詳細は、man ページの nmcli(1) を参照してください。

5.5. NMCLI による接続プロファイルの作成および修正

デバイスに関連付ける接続プロファイルを作成できます。

前提条件

- 「nmcli の使用」
- 「nmcli のプロパティー名とエイリアスの概要」



重要

nmcli コマンドを使用する場合は、nmcli コマンドの一部を入力して **Tab** キーを押し、オートコンプリート機能でコマンドシーケンスを補完することをお勧めします。複数の補完候補がある場合は、**Tab** で一覧が表示されます。これにより、ユーザーはコマンド入力を速く簡単に行えるようになります。nmcli のオートコンプリート機能を有効にするには、**bash-completion** パッケージを忘れずにインストールしてください。

```
~]# yum install bash-completion
```

パッケージのインストール後、次回コンソールにログインしたときに **nmcli オートコンプリート**が利用できるようになります。これを現在のコンソールでも有効にするには、次のコマンドを入力します。

```
~]$ source /etc/profile.d/bash_completion.sh
```

nmcli を使用して、NetworkManager の新しいプロファイルを **作成** する基本書式は以下のとおりです。

nmcli c add {COMMON_OPTIONS} [IP_OPTIONS]/[NETMASK] [GATEWAY]

1. **{COMMON_OPTIONS}** は、エイリアスまたはプロパティー名です。エイリアスとプロパティー名を参照してください。
2. **[IP_OPTIONS]** は IP アドレスです。
   - IPv4 アドレスの場合 - **ip4**
   - IPv6 アドレスの場合 - **ip6**
3. **[NETMASK]** は、ネットワークマスクの幅です。たとえば、**255.255.255.0** は、プレフィックス **198.51.100.0/24** のネットワークマスクです。
4. **[GATEWAY]** は、ゲートウェイ情報です。
   - IPv4 アドレスの場合 - **gw4**
   - IPv6 アドレスの場合 - **gw6**

nmcli connection add type ethernet **con-name connection-name ifname interface-name ip4 address/network mask gw4 address**

例5.13 IPv4 アドレスで接続プロファイルの作成

```
~]$ nmcli c add type ethernet ifname eth0 con-name "My Connection" ip4 192.168.2.100/24 gw4 192.168.2.1
Connection 'new-ens33' (f0c23472-1aec-4e84-8f1b-be8a2ecbeade) successfully added.
```

作成した接続を有効にするには、以下を実行します。

```
~]$ nmcli c up _"My Connection"
```

作成した接続を表示するには、以下を実行します。

```
~]$ nmcli c show "My Connection"
```

nmcli c showcon-name コマンドを使用すると、空の値やデフォルト値のものも含め、接続内の全プロパティーが表示されることに留意してください。出力がターミナルページより長い場合は、出力での移動が簡単にできるよう **nmcli** によりページャーが生成されます。ページャーでは、矢印キーで上下に移動し、**q** キーで終了します。

接続をよりコンパクトに表示するには、**-o** オプションを使用します。

```
~]$ nmcli -o c show "My Connection"
```

nmcli -o c showcon-name コマンドでも接続の内容は表示されますが、空のプロパティーやデフォルト値が設定されたプロパティーは省略されます。これにより、通常は出力がより読みやすい短いものになります。

関連資料

- プロパティーとその設定に関する詳細は、man ページの **nm-settings(5)** を参照してください。

5.6. NMCLI インタラクティブ接続エディターの使用

nmcli ツールには、インタラクティブな接続エディターがあります。これにより、必要に応じて接続パラメーターを変更できます。使用するには、以下を実行します。

```
~]$ nmcli con edit
```

表示された一覧から、有効な**接続の種類**を入力してください。その後、パラメーターを修正できます。

```
~]$ nmcli con edit
Valid connection types: generic, 802-3-ethernet (ethernet), pppoe, 802-11-wireless (wifi), wimax,
gsm, cdma, infiniband, adsl, bluetooth, vpn, 802-11-olpc-mesh (olpc-mesh), vlan, bond, team, bridge,
bond-slave, team-slave, bridge-slave, no-slave, tun, ip-tunnel, macsec, macvlan, vxlan, dummy
Enter connection type: ethernet

===| nmcli interactive connection editor |===

Adding a new '802-11-wireless' connection

Type 'help' or '?' for available commands.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, 802-11-wireless (wifi), 802-11-wireless-security (wifi-
sec), 802-1x, ipv4, ipv6, proxy
nmcli>
```

イーサネット接続の設定を編集できるようになりました。使用可能なコマンドの一覧を見るには、**help** または **?** を入力します。

```
nmcli> ?
------------------------------------------------------------------------------
---[ Main menu ]---
goto     [<setting> | <prop>]        :: go to a setting or property
```

```
remove   <setting>[.<prop>] | <prop> :: remove setting or reset property value
set      [<setting>.<prop> <value>]  :: set property value
describe [<setting>.<prop>]          :: describe property
print    [all | <setting>[.<prop>]]  :: print the connection
verify   [all | fix]                 :: verify the connection
save     [persistent|temporary]      :: save the connection
activate [<ifname>] [/<ap>|<nsp>]    :: activate the connection
back                                 :: go one level up (back)
help/?   [<command>]                 :: print this help
nmcli    <conf-option> <value>       :: nmcli configuration
quit                                 :: exit nmcli
------------------------------------------------------------------------------
nmcli>
```

終了するには、**quit** コマンドを入力します。

例5.14 nmcli インタラクティブ接続エディターを使用した新しいイーサネット接続の追加

```
~]$ nmcli con edit
Valid connection types: generic, 802-3-ethernet (ethernet), pppoe, 802-11-wireless (wifi), wimax,
gsm, cdma, infiniband, adsl, bluetooth, vpn, 802-11-olpc-mesh (olpc-mesh), vlan, bond, team,
bridge, bond-slave, team-slave, bridge-slave, no-slave, tun, ip-tunnel, macsec, macvlan, vxlan,
dummy
Enter connection type: ethernet

===| nmcli interactive connection editor |===

Adding a new '802-3-ethernet' connection

Type 'help' or '?' for available commands.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-1x, dcb, ipv4, ipv6,
proxy
nmcli> set connection.id new_eth1
nmcli> set connection.interface-name eth1
nmcli> set connection.autoconnect yes
nmcli> save
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
Connection 'new_eth1' (34ac8f9a-e9d8-4e0b-9751-d5dc87cc0467) successfully saved.
nmcli> quit
```

新しいネットワークインターフェース設定ファイルは、**/etc/sysconfig/network-scripts** ディレクトリーに作成されます。

```
~]# ls -lrt /etc/sysconfig/network-scripts/ifcfg*
-rw-r--r--. 1 root root 254 Aug 15  2017 /etc/sysconfig/network-scripts/ifcfg-lo
-rw-r--r--. 1 root root 304 Apr 26 22:14 /etc/sysconfig/network-scripts/ifcfg-ens3
-rw-r--r--. 1 root root 266 Aug  6 11:03 /etc/sysconfig/network-scripts/ifcfg-new_eth1
```

5.7. NMCLI を使用した接続プロファイルの修正

接続プロファイルの既存の設定を修正できます。

前提条件

- 「nmcli の使用」
- 「nmcli のプロパティー名とエイリアスの概要」
- 「nmcli による接続プロファイルの作成および修正」

重要

nmcli コマンドを使用する場合は、nmcli コマンドの一部を入力して **Tab** キーを押し、オートコンプリート機能でコマンドシーケンスを補完することをお勧めします。複数の補完候補がある場合は、**Tab** で一覧が表示されます。これにより、ユーザーはコマンド入力を速く簡単に行えるようになります。nmcli のオートコンプリート機能を有効にするには、**bash-completion** パッケージを忘れずにインストールしてください。

```
~]# yum install bash-completion
```

パッケージのインストール後、次回コンソールにログインしたときに **nmcli オートコンプリート**が利用できるようになります。これを現在のコンソールでも有効にするには、次のコマンドを入力します。

```
~]$ source /etc/profile.d/bash_completion.sh
```

接続プロファイルの 1 つまたは複数のプロパティーを修正するには、次のコマンドを使用します。

```
nmcli c modify
```

たとえば、connection.id を "My Connection" から "My favorite connection" に、connection.interface-name を eth1 に変更する場合は、次を実行します。

```
~]$ nmcli c modify "My Connection" connection.id "My favorite connection"
connection.interface-name eth1
```

接続の修正後、nmcliを使用して変更を**適用**するには、次のコマンドを入力して接続を再度アクティブにします。

```
~]$ [command]~]$nmcli con up "My favorite connection"
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/16)
```

修正した接続を表示するには、**nmcli con show con-name** コマンドを入力します。

5.8. 静的イーサネット接続の設定

5.8.1. nmcli を使用した静的イーサネット接続の設定

前提条件

- 「nmcli の使用」

- 「nmcli による接続プロファイルの作成および修正」

例5.15 2 つのIPv4 DNS サーバーアドレスの設定

```
~]$ nmcli con mod test-lab ipv4.dns "8.8.8.8 8.8.4.4"
```

これにより、以前設定された **DNS** サーバーが置き換えられる点に留意してください。

または、2 つの **IPv6 DNS** サーバーアドレスを設定します。

```
~]$ nmcli con mod test-lab ipv6.dns "2001:4860:4860::8888 2001:4860:4860::8844"
```

これにより、以前設定された **DNS** サーバーが置き換えられる点に留意してください。

以前の設定に他の **DNS** サーバーを追加する場合は、**+** プレフィックスを使用します。

```
~]$ nmcli con mod test-lab +ipv4.dns "8.8.8.8 8.8.4.4"
```

```
~]$ nmcli con mod test-lab +ipv6.dns "2001:4860:4860::8888 2001:4860:4860::8844"
```

新しい**イーサネット**接続をアクティブ化するには、次を実行します。

```
~]$ nmcli con up test-lab ifname ens9
Connection successfully activated (D-Bus active path:
/org/freedesktop/NetworkManager/ActiveConnection/6)
```

デバイスおよび接続のステータスを確認するには、次を実行します。

```
~]$ nmcli device status
DEVICE  TYPE      STATE      CONNECTION
ens3    ethernet  connected  my-office
ens9    ethernet  connected  test-lab
lo      loopback  unmanaged  --
```

新規に設定した接続の詳細情報を表示するには、次を実行します。

```
~]$ nmcli -p con show test-lab
===============================================================================

                    Connection profile details (test-lab)
===============================================================================

connection.id:                          test-lab
connection.uuid:                        05abfd5e-324e-4461-844e-8501ba704773
connection.interface-name:                ens9
connection.type:                        802-3-ethernet
connection.autoconnect:                   yes
connection.timestamp:                     1410428968
connection.read-only:                     no
connection.permissions:
connection.zone:                        --
connection.master:                      --
```

```
connection.slave-type:                  --
connection.secondaries:
connection.gateway-ping-timeout:        0
[output truncated]
```

-p, --pretty オプションを使用すると、出力にタイトルバナーとセクション区切りが追加されます。

5.8.2. nmcli インタラクティブエディターを使用した静的イーサネット接続の設定

nmcli インタラクティブエディターを使用して静的イーサネット接続を設定するには、次を実行します。

```
~]$ nmcli con edit type ethernet con-name ens3

===| nmcli interactive connection editor |===

Adding a new '802-3-ethernet' connection

Type 'help' or '?' for available commands.
Type 'describe [>setting<.>prop<]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-1x, ipv4, ipv6, dcb
nmcli> set ipv4.addresses 192.168.122.88/24
Do you also want to set 'ipv4.method' to 'manual'? [yes]: yes
nmcli> save temporary
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation of the
connection.
Do you still want to save? [yes] yes
Connection 'ens3' (704a5666-8cbd-4d89-b5f9-fa65a3dbc916) successfully saved.
nmcli> quit
```

デフォルトの動作では、接続プロファイルが永続的に保存されます。必要な場合は、**save temporary** コマンドで、プロファイルを次回の再起動時までメモリーにだけ保持できます。

関連資料

- プロパティーとその設定に関する詳細は、man ページの **nm-settings(5)** を参照してください。

5.9. 動的イーサネット接続の設定

5.9.1. nmcli を使用した動的イーサネット接続の設定

前提条件

- 「nmcli の使用」
- 「nmcli による接続プロファイルの作成および修正」

ホストから **DHCP** サーバーに送信されたホスト名を変更するには、**dhcp-hostname** プロパティーを修正します。

```
~]$ nmcli con modify my-office my-office ipv4.dhcp-hostname host-name ipv6.dhcp-hostname
host-name
```

ホストから **DHCP** サーバーに送信された **IPv4** クライアント ID を変更するには、 **dhcp-client-id** プロパティーを修正します。

```
~]$ nmcli con modify my-office my-office ipv4.dhcp-client-id client-ID-string
```

IPv6には、**IPv6** の識別子を作成する **dhclient**、つまり **dhcp-client-id** プロパティーはありません。詳細は、man ページの **dhclient(8)** を参照してください。

DHCP サーバーからホストに送信された **DNS** サーバーを無視するには、**ignore-auto-dns** プロパティーを修正します。

```
~]$ nmcli con modify my-office my-office ipv4.ignore-auto-dns yes ipv6.ignore-auto-dns yes
```

5.9.2. インタラクティブエディターを使用した動的イーサネット接続の設定

インタラクティブエディターを使用して動的イーサネット接続を設定するには、次を実行します。

```
~]$ nmcli con edit type ethernet con-name ens3

===| nmcli interactive connection editor |===

Adding a new '802-3-ethernet' connection

Type 'help' or '?' for available commands.
Type 'describe [<setting>.<prop>]' for detailed property description.

You may edit the following settings: connection, 802-3-ethernet (ethernet), 802-1x, ipv4, ipv6, dcb
nmcli> describe ipv4.method

=== [method] ===
[NM property description]
IPv4 configuration method.  If 'auto' is specified then the appropriate automatic method (DHCP, PPP,
etc) is used for the interface and most other properties can be left unset.  If 'link-local' is specified,
then a link-local address in the 169.254/16 range will be assigned to the interface.  If 'manual' is
specified, static IP addressing is used and at least one IP address must be given in the 'addresses'
property.  If 'shared' is specified (indicating that this connection will provide network access to other
computers) then the interface is assigned an address in the 10.42.x.1/24 range and a DHCP and
forwarding DNS server are started, and the interface is NAT-ed to the current default network
connection.  'disabled' means IPv4 will not be used on this connection.  This property must be set.

nmcli> set ipv4.method auto
nmcli> save
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation of the
connection.
Do you still want to save? [yes] yes
Connection 'ens3' (090b61f7-540f-4dd6-bf1f-a905831fc287) successfully saved.
nmcli> quit
```

デフォルトの動作では、接続プロファイルが永続的に保存されます。必要な場合は、**save temporary** コマンドで、プロファイルを次回の再起動時までメモリーにだけ保持できます。

関連資料

- プロパティーとその設定に関する詳細は、man ページの **nm-settings(5)** を参照してください。

5.10. NMCLI コマンドを使用して、静的ルートを設定する方法

静的ルートを設定するには、次の構文で **nmcli** ユーティリティーを使用します。

```
$ nmcli connection modify connection_name ipv4.routes "ip[/prefix] [next_hop] [metric]
[attribute=value] [attribute=value] ..."
```

このコマンドは、次のルート属性に対応します。

- **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**

サブコマンド **ipv4.routes** を使用する場合は、**nmcli** が、このパラメーターの現在の設定をすべて上書きします。ルートを追加するには、**nmcli connection modify connection_name +ipv4.routes "…"** コマンドを使用します。同様に、**nmcli connection modify connection_name -ipv4.routes "…"** を使用して、特定ルートを削除できます。

5.11. NMCLI コマンドを使用した静的ルートの設定

nmcli connection modify コマンドを使用して、ネットワーク接続の設定に静的ルートを追加できます。

本セクションの手順では、**198.51.100.1** で実行しているゲートウェイを使用する **192.0.2.0/24** ネットワークにルートを追加する方法を説明します。これは、**example** 接続から到達可能です。

前提条件

- ネットワークが設定されている。
- 静的ルートのゲートウェイが、インターフェース上で直接到達できる。
- **root** 権限が必要 (物理コンソールにログインしている場を除く)。

手順

1. 静的ルートを **example** 接続に追加します。

```
$ sudo nmcli connection modify example +ipv4.routes "192.0.2.0/24 198.51.100.1"
```

1回で複数のルートを設定するには、個々のルートをコンマで区切ってコマンドに渡す必要があります。たとえば、ルートを **192.0.2.0/24** および **203.0.113.0/24** のネットワーク追加して、両方のルートが **198.51.100.1** ゲートウェイを通るには、以下のコマンドを実行します。

```
$ 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. 必要に応じて、ルートが設定に正しく追加されたことを確認します。

```
$ nmcli connection show example
...
ipv4.routes:       { ip = 192.0.2.1/24, nh = 198.51.100.1 }
...
```

3. ネットワーク接続が再起動します。

```
$ sudo nmcli connection up example
```

警告

接続を再起動すると、そのインターフェースの接続が一時的に中断します。

4. 必要に応じて、ルートがアクティブであることを確認します。

```
$ ip route
...
192.0.2.0/24 via 198.51.100.1 dev example proto static metric 100
```

関連資料

- **nmcli** の詳細は、man ページの **nmcli(1)** を参照してください。

5.12. NMCLI インタラクティブモードを使用した静的ルートの設定

nmcli ユーティリティーのインタラクティブモードを使用して、ネットワーク接続の設定に静的ルートを追加できます。

本セクションの手順では、**198.51.100.1** で実行しているゲートウェイを使用する **192.0.2.0/24** ネットワークにルートを追加する方法を説明します。これは、**example** 接続から到達可能です。

前提条件

- ネットワークが設定されている。
- 静的ルートのゲートウェイが、インターフェース上で直接到達できる。
- **root** 権限が必要 (物理コンソールにログインしている場を除く)。

手順

1. **example** 接続の **nmcli** インタラクティブモードを開きます。

```
$ sudo nmcli connection edit example
```

2. 静的ルートを追加します。

```
nmcli> set ipv4.routes 192.0.2.0/24 198.51.100.1
```

3. 必要に応じて、ルートが設定に正しく追加されたことを確認します。

```
nmcli> print
...
ipv4.routes:       { ip = 192.0.2.1/24, nh = 198.51.100.1 }
...
```

ip 属性には、転送するネットワークと、ゲートウェイの **nh** 属性 (次のホップ) が表示されます。

4. 設定を保存します。

```
nmcli> save persistent
```

5. ネットワーク接続が再起動します。

```
nmcli> activate example
```

警告

接続を再起動すると、この接続を現在使用している接続がすべて一時的に中断されます。

6. **nmcli** インタラクティブモードは残します。

```
nmcli> quit
```

7. 必要に応じて、ルートがアクティブであることを確認します。

```
$ ip route
...
192.0.2.0/24 via 198.51.100.1 dev example proto static metric 100
```

関連資料

- インタラクティブモードで使用できるコマンドの一覧は、インタラクティブシェルに **help** を入力します。

5.13. NMCLI を使用して、既存の接続でデフォルトのゲートウェイの設定

ほとんどの場合、管理者は、たとえば 「nmcli を使用した静的イーサネット接続の設定」 の説明に従って、接続を作成する際にデフォルトのゲートウェイを設定します。

本セクションでは、**nmcli** ユーティリティーを使用して作成されている接続で、デフォルトのゲートウェイを設定または更新する方法を説明します。

前提条件

- デフォルトゲートウェイが設定される接続で、静的 IP アドレスを少なくとも 1 つ設定している。
- **root** 権限が必要 (物理コンソールにログインしている場を除く)。

手順

1. デフォルトゲートウェイの IP アドレスを設定します。
   たとえば、**example** 接続のデフォルトゲートウェイの IPv4 アドレスを **192.0.2.1** に設定します。

   ```
   $ sudo nmcli connection modify example ipv4.gateway "192.0.2.1"
   ```

   たとえば、**example** 接続のデフォルトゲートウェイの IPv6 アドレスを **2001:db8::1** に設定するには、以下を実行します。

   ```
   $ sudo nmcli connection modify example ipv6.gateway "2001:db8::1"
   ```

2. ネットワーク接続を再起動して、変更を有効にします。たとえば、コマンドラインで **example** 接続を再起動するには、以下を実行します。

   ```
   $ sudo nmcli connection up example
   ```

   警告

   このネットワーク接続を使用しているすべての接続が、再起動時に一時的に中断されます。

3. 必要に応じて、ルートがアクティブであることを確認します。
   IPv4 デフォルトゲートウェイを表示するには、以下を実行します。

   ```
   $ ip -4 route
   default via 192.0.2.1 dev example proto static metric 100
   ```

   IPv6 デフォルトゲートウェイを表示するには、以下を実行します。

   ```
   $ ip -6 route
   default via 2001:db8::1 dev example proto static metric 100 pref medium
   ```

関連資料

- 「nmcli を使用した静的イーサネット接続の設定」

5.14. NMCLI インタラクティブモードを使用して、既存の接続でデフォルトゲートウェイの設定

ほとんどの場合、管理者は、たとえば 「nmcli インタラクティブエディターを使用した静的イーサネット接続の設定」 の説明に従って、接続を作成する際にデフォルトのゲートウェイを設定します。

本セクションでは、**nmcli** ユーティリティーのインタラクティブモードを使用して作成されている接続で、デフォルトのゲートウェイを設定または更新する方法を説明します。

前提条件

- デフォルトゲートウェイが設定される接続で、静的 IP アドレスを少なくとも 1 つ設定している。
- **root** 権限が必要 (物理コンソールにログインしている場を除く)。

手順

1. 必要な接続に対して **nmcli** インタラクティブモードを開きます。たとえば、**example** 接続の **nmcli** インタラクティブモードを開くには、以下を実行します。

```
$ sudo nmcli connection edit example
```

2. デフォルトのゲートウェイを設定します。
たとえば、**example** 接続のデフォルトゲートウェイの IPv4 アドレスを **192.0.2.1** に設定します。

```
nmcli> set ipv4.gateway 192.0.2.1
```

たとえば、**example** 接続のデフォルトゲートウェイの IPv6 アドレスを **2001:db8::1** に設定するには、以下を実行します。

```
nmcli> set ipv6.gateway 2001:db8::1
```

3. 必要に応じて、デフォルトゲートウェイが正しく設定されていることを確認します。

```
nmcli> print
...
ipv4.gateway:                192.0.2.1
...
ipv6.gateway:                2001:db8::1
...
```

4. 設定を保存します。

```
nmcli> save persistent
```

5. ネットワーク接続を再起動して、変更を有効にします。

```
nmcli> activate example
```

警告

このネットワーク接続を使用しているすべての接続が、再起動時に一時的に中断されます。

6. **nmcli** インタラクティブモードは残します。

```
nmcli> quit
```

7. 必要に応じて、ルートがアクティブであることを確認します。
IPv4 デフォルトゲートウェイを表示するには、以下を実行します。

```
$ ip -4 route
default via 192.0.2.1 dev example proto static metric 100
```

IPv6 デフォルトゲートウェイを表示するには、以下を実行します。

```
$ ip -6 route
default via 2001:db8::1 dev example proto static metric 100 pref medium
```

関連資料

- 「nmcli インタラクティブエディターを使用した静的イーサネット接続の設定」

第6章 GNOME GUI を使用したネットワークの設定

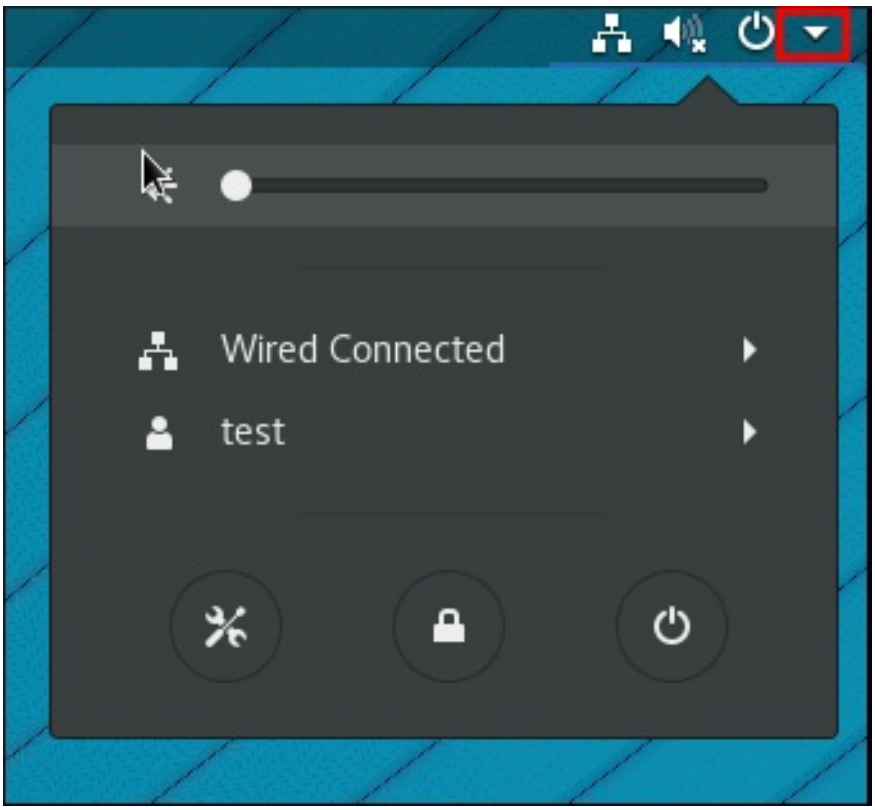
グラフィカルユーザーインターフェース (**GUI**) を使用してネットワークインターフェースを変更する方法を、以下に示します。

- デスクトップ右上の GNOME Shell **ネットワーク接続アイコン**
- GNOME **control-center** アプリケーション
- GNOME **nm-connection-editor** アプリケーション

6.1. GNOME SHELL ネットワーク接続アイコンを使用したネットワーク接続

ネットワーク 設定にアクセスするには、画面右上隅の GNOME Shell **ネットワーク接続アイコン**をクリックし、メニューを開きます。

図6.1 ネットワーク接続アイコンメニュー



GNOME Shell ネットワーク接続アイコンをクリックすると、以下が表示されます。

- 現在接続しているカテゴリ別のネットワーク一覧 (**Wired** や **Wi-Fi** など)。
- **NetworkManager** で検知されている、すべての**利用可能なネットワーク**の一覧。ネットワークに接続している場合は、接続名の左側に表示される。
- 設定済みの仮想プライベートネットワーク (VPN) への接続オプション。
  ならびに
- **ネットワーク設定** メニューエントリーの選択オプション。

6.2. CONTROL-CENTER を使用したネットワーク接続の作成

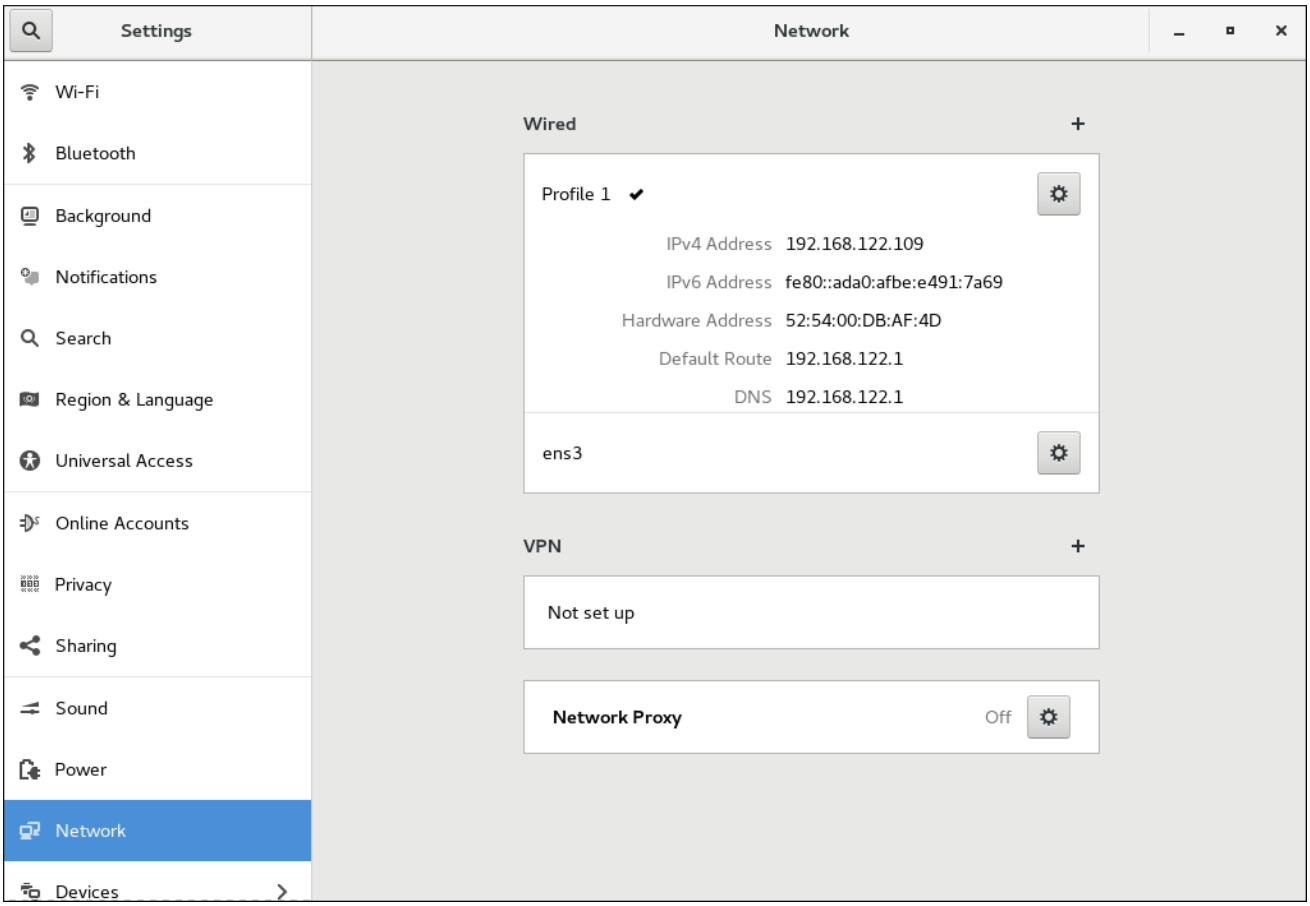
GNOME **control-center** アプリケーションを介してネットワーク接続を作成できます。このアプリケーションは、利用可能なネットワークデバイスとその現在の設定を表示する、グラフィカルユーザーインターフェースです。

この手順では、**control-center** を使用して、新しく**有線**、**無線**、 **VPN** 接続を作成する方法を説明します。

手順

1. **Super** キーを押してアクティビティ画面を表示し、 **Settings** と入力して **Enter** キーを押します。次に、左側の **Network** タブを選択すると、**ネットワーク**設定ツールが表示されます。

図6.2 ネットワーク設定ウィンドウの表示



1. 新しい接続を追加するには、プラスボタンをクリックします。

   - **有線接続**を追加する場合は、**Wired** エントリーの横のプラスボタンを押して、接続を設定します。
   - **VPN 接続**を追加する場合は、**VPN** エントリーの横のプラスボタンを押して、接続を設定します。
   - **Wi-Fi 接続**を追加する場合は、**Settings** メニューの **Wi-fi** エントリーをクリックして、接続を設定します。

6.3. CONTROL-CENTER を使用したネットワーク接続の設定

GNOME **control-center** アプリケーションを介してネットワーク接続を設定できます。

6.3.1. control-center を使用した有線 (イーサネット) 接続の設定

手順

1. **Super** キーを押してアクティビティ画面を表示し、**Settings** と入力して**Enter** キーを押します。次に、左側の **Network** メニューエントリーを選択すると、**Network** 設定ツールが表示されます。 ネットワーク設定ウィンドウの表示を参照してください。

2. **Wired** ネットワークインターフェースを選択します
システムに、デフォルトで1つの有線**接続プロファイル**が作成されて設定されます。名前は、**Wired** とです。1つのインターフェースに対して複数のプロファイルを作成し、必要に応じて適用することができます。デフォルトのプロファイルは削除できませんが、設定は変更できます。

3. 既存の接続を編集する歯車アイコンをクリックして、デフォルトの **Wired** プロファイルを編集します。あるいは、プラスボタンをクリックして、新しい接続用の設定オプションを設定します。

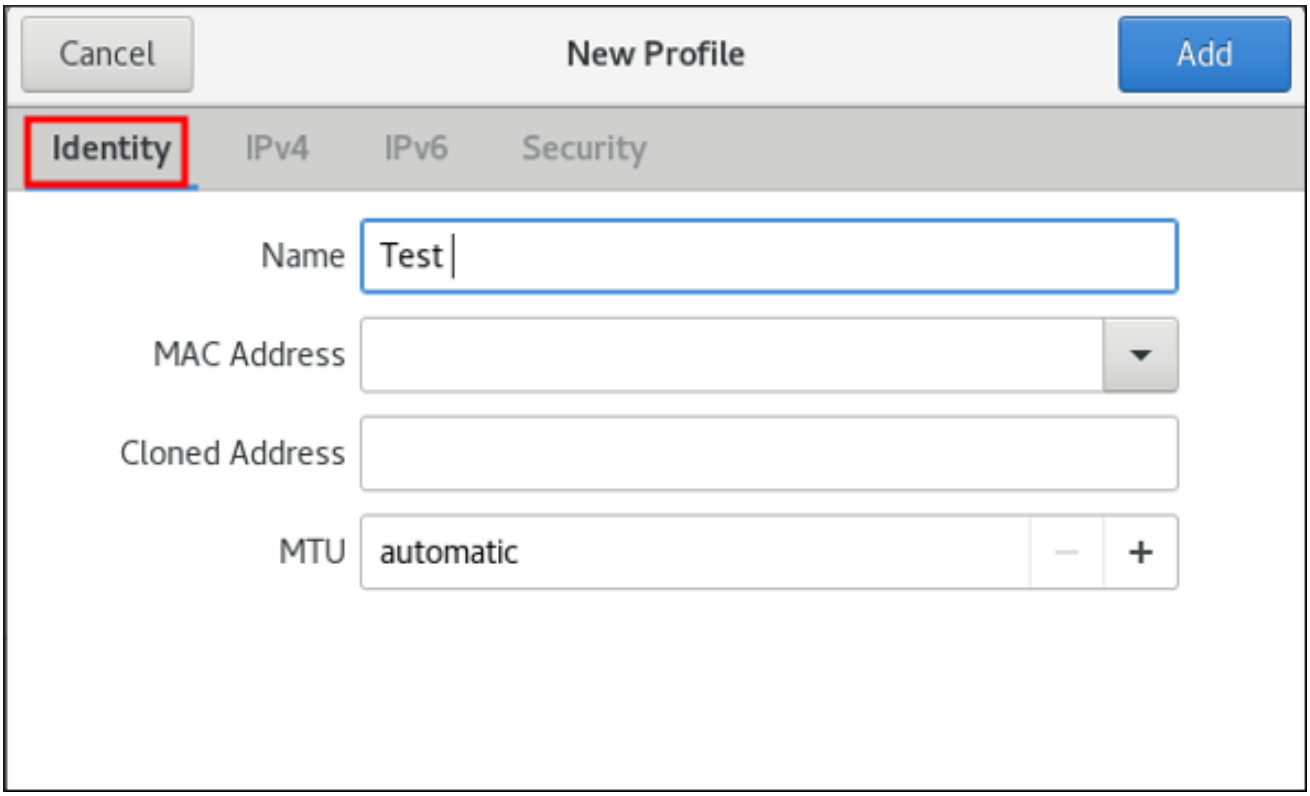
注記

プラスボタンをクリックして新しい接続を追加する場合は、NetworkManager により、その接続用の新しい設定が作成され、既存の接続の編集に使用するのと同じダイアログが表示されます。これらのダイアログの違いは、既存の接続プロファイルに Details メニューエントリーがあることです。

基本設定オプション

Wired ダイアログでは、**Identity** メニューエントリーを選択すると、次の設定が表示されます。

図6.3 有線接続に関する基本設定オプション



- **Name** - ネットワーク接続のわかりやすい名前を入力します。この名前は、**Network** のメニューでこの接続を一覧に表示する際に使用されます。

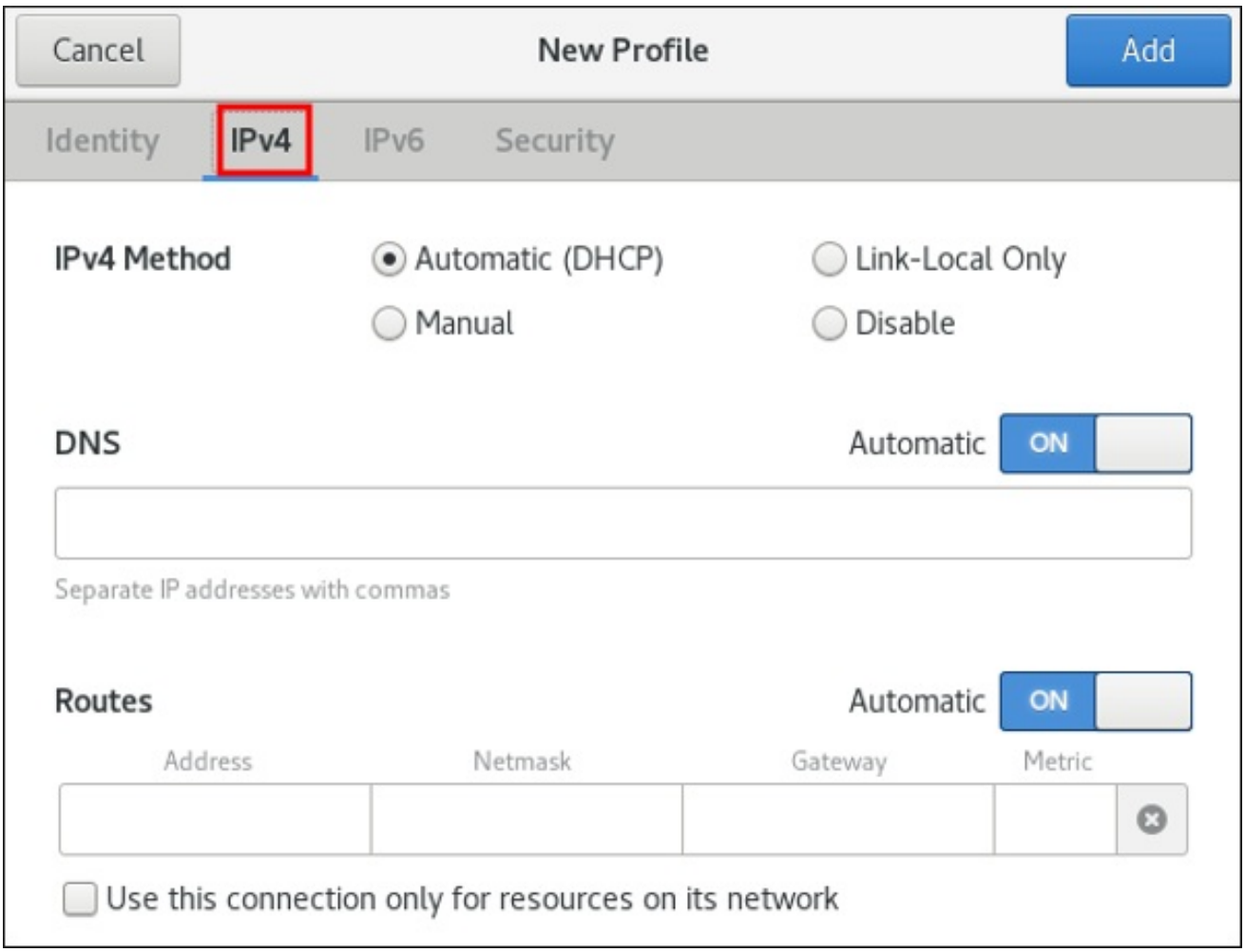
- **MAC Address** - このプロファイルを適用する必要があるインターフェースの MAC アドレスを選択します。

- **Cloned Address** - 必要な場合は、使用する別な MAC アドレスを入力します。
- **MTU** - 必要な場合は、使用する特定の **最大転送単位** (**MTU**) を入力します。MTU 値は、リンク層が転送する最大パケットサイズをバイト単位で表したものです。この値のデフォルトは **1500** で、通常は指定したり変更したりする必要はありません。

control-center を使用した有線用の IPv4 設定

有線接続での **IPv4** 設定は、さらに設定できます。**Wired** ダイアログで、**IPv4** メニューエントリーをクリックします。

図6.4 IPv4 セッティングの設定



IPv4 メニューエントリーでは、以下を設定できます。

- ネットワークへの接続に使用する **IPv4 Method**
- **DNS**
- **Routes**

IPv4 Method

Automatic (自動) (DHCP) - 接続しているネットワークがルーター通知 (RA) または **DHCP** サーバーを使用して動的 **IP** アドレスを割り当てる場合は、このオプションを選択します。

Link-Local Only (リンクローカルのみ) - 接続しているネットワークに **DHCP** サーバーがなく、**IP** アドレスを手動で割り当てない場合は、このオプションを選択します。プレフィックス **169.254/16** 付きのランダムなアドレスが、RFC 3927 に従って割り当てられます。

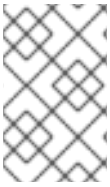
Manual (手動) - **IP** アドレスを手動で割り当てたい場合は、このオプションを選択します。

Disable (無効) - この接続では **IPv4** は無効です。

DNS

DNS セクションでは、 **Automatic (自動)** が **ON** になっている場合、Automatic を **OFF** に切り替えて、使用したい DNS サーバーの IP アドレスを入力します。IP アドレスはコンマで区切ります。

Routes



注記

Routes セクションでは、 **Automatic (自動)** が **ON** になっている場合は、ルート通知 (RA) または使用する DHCP からのルートが使用されますが、他の静的ルートを追加することもできます。**OFF** の場合は、静的ルートだけが使用されます。

Address - リモートネットワーク、サブネット、またはホストの **IP** アドレスを入力します。

Netmask - 上に入力した **IP** アドレスのネットマスクまたはプレフィックス長。

Gateway - 上に入力したリモートネットワーク、サブネット、またはホストにつながるゲートウェイの **IP** アドレス。

Metric - このルートに与える優先値であるネットワークコスト。数値が低い方が有線されます。

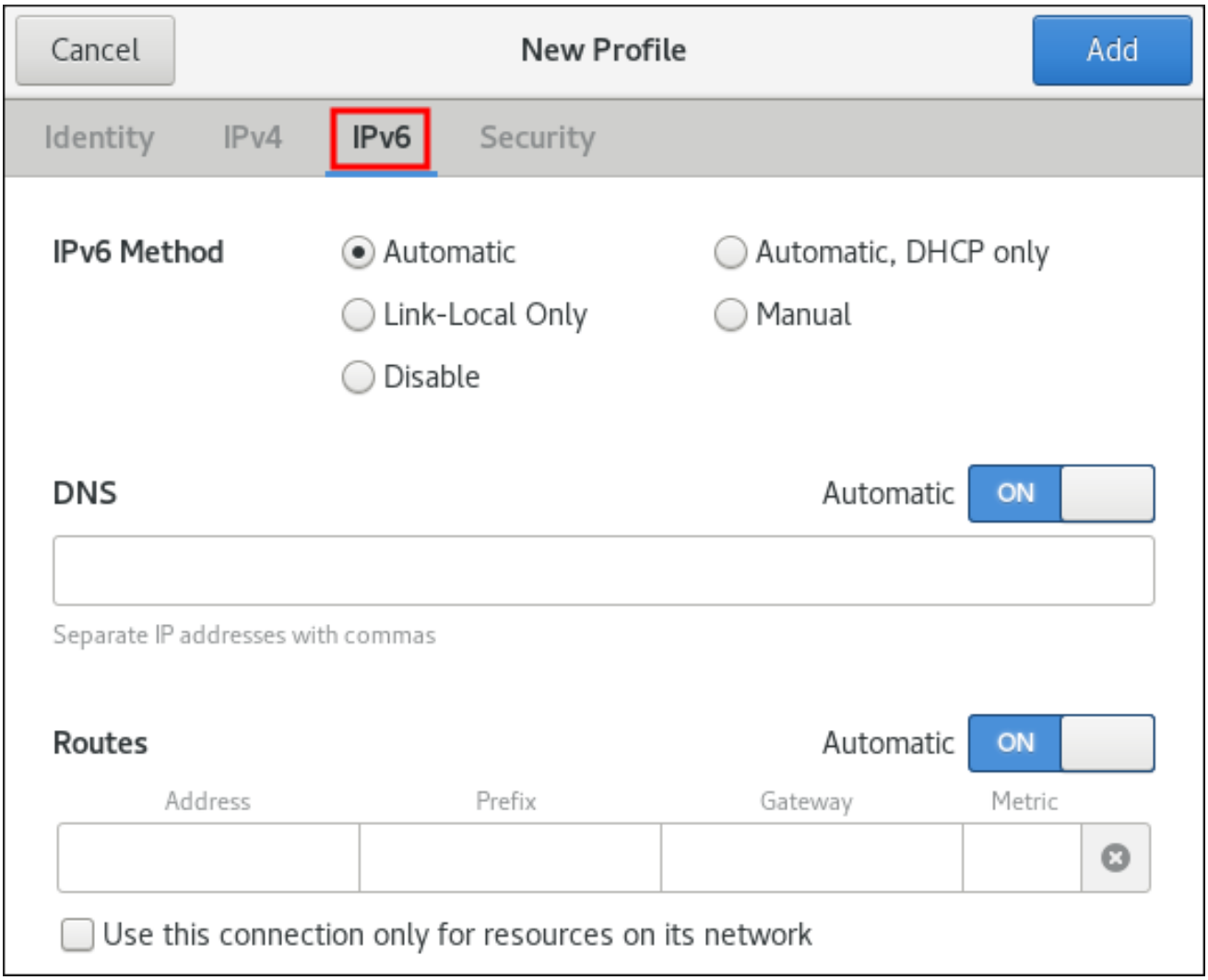
Use this connection only for resources on its network (**この接続はネットワーク上のリソースのためだけに使用**)

このチェックボックスを選択すると、この接続はデフォルトルートになりません。よくある例は、ヘッドオフィスへの接続が VPN トンネルや専用線で、インターネット向けトラフィックにこの接続を使用しない場合です。このオプションを選択すると、この接続で自動的に学習されたルートを使用することが明確なトラフィックか、手動で入力されたトラフィックのみがこの接続を経由します。

control center を使用した有線用の IPv6 設定

また、有線接続で **IPv6** 設定を設定するには、**IPv6** メニューエントリーをクリックします。

図6.5 IPv6 セッティングの設定



IPv6 メニューエントリーでは、以下を設定できます。

- ネットワークへの接続に使用する **IPv6 Method**
- **DNS**
- **Routes**

IPv6 Method

Automatic (自動) - **IPv6** ステートレスアドレス自動設定 (SLAAC) を使用してハードウェアのアドレスとルーター通知 (RA) に基づくステートレス自動設定を作成するには、このオプションを選択します。

Automatic, DHCP only (自動、DHCP のみ) - RA を使用せず、直接 **DHCPv6** に情報を要求してステートフルな設定を作成する場合は、このオプションを選択します。

Link-Local Only (リンクローカルのみ) - 接続しているネットワークに **DHCP** サーバーがなく、**IP** アドレスを手動で割り当てない場合は、このオプションを選択します。プレフィックス **FE80::0** 付きのランダムなアドレスが RFC 4862 に従って割り当てられます。

Manual (手動) - **IP** アドレスを手動で割り当てたい場合は、このオプションを選択します。

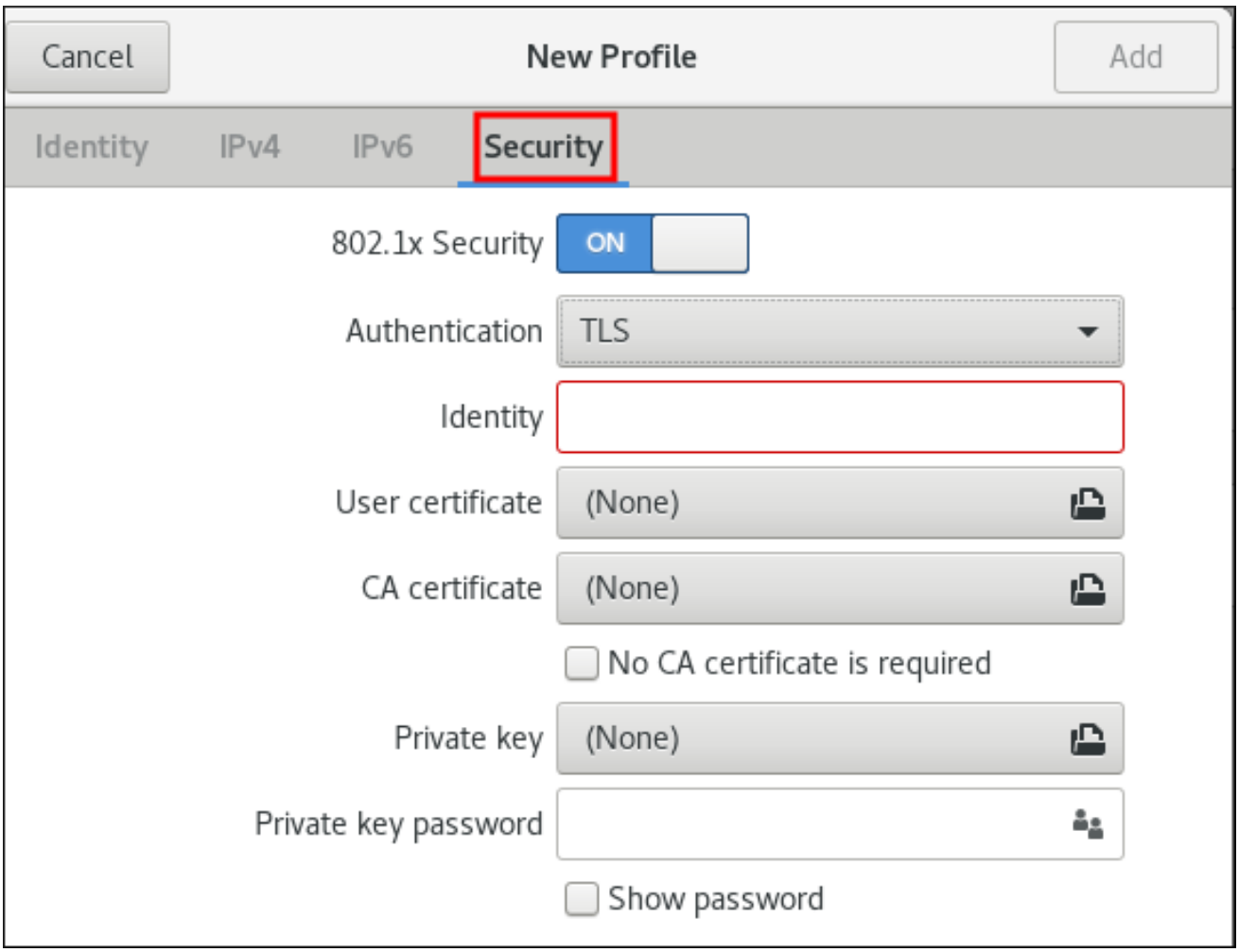
Disable (無効) - この接続では **IPv6** は無効です。

control-center を使用した有線用の 802.1X セキュリティー設定

802.1X セキュリティーとは、**ポートベースのネットワークアクセス制御** (**PNAC**) 用の IEEE 基準の名前です。これは、**WPA エンタープライズ** とも呼ばれます。802.1X セキュリティーは、物理ネットワークから **論理ネットワーク** へのアクセスを制御する手段です。論理ネットワークに参加するクライアントはすべて、正しい 802.1X 認証方法を使用して、ルーターなどのサーバーで認証を行う必要があります。

有線接続で **802.1X セキュリティー**を設定するには、**Security** メニューエントリーをクリックします。

図6.6 control-center を使用した有線用の 802.1X セキュリティー設定



設定を有効にするには、ボタンを **ON** に設定して、以下の認証方法から 1 つ選択します。

- **TLS** は Transport Layer Security (**トランスポート層セキュリティ**) の略です。TLS の設定に進みます
- **PWD** は Password (**パスワード**) の略です。PWD の設定に進みます
- **FAST** は Flexible Authentication through Secure Tunneling (**セキュアトンネリングを介したフレキシブル認証**) の略です。FAST の設定に進みます
- **Tunneled TLS** を選択します。これは Tunneled Transport Layer Security (**トンネル化トランスポート層セキュリティ**) の略で、TTLS、あるいは EAP-TTLS とも呼ばれます。Tunneled TLS の設定に進みます
- **Protected EAP (PEAP)** (Protected Extensible Authentication Protocol (**保護された拡張認証プロトコル**) を表す) を選択して、Protected EAP (PEAP) の設定に進みます

TLS の設定

トランスポート層セキュリティ（TLS）では、クライアントとサーバーは、TLS プロトコルを使用した相互認証が行われます。

TLS セキュリティーを使用すると、証明書を管理する公開鍵インフラストラクチャー (PKI) のオーバーヘッドが必要になります。TLS セキュリティーを使用する利点は、侵害されたパスワードでは (W)LAN へのアクセスが許可されないことです。侵入者は、認証するクライアントの秘密鍵にもアクセスしなければなりません。

NetworkManager は、対応する TLS のバージョンを決定しません。**NetworkManager** は、ユーザーが入力するパラメーターを集め、手順を処理するデーモンである **wpa_supplicant** にこれらを渡します。このデーモンは、OpenSSL を使用して TLS トンネルを確立します。OpenSSL 自体は、SSL/TLS プロトコルバージョンを処理します。両端が対応する一番高いバージョンが使用されます。

TLS を設定するには、「control-center を使用した有線 (イーサネット) 接続の設定」 で説明されている手順に従います。以下の設定が可能です。

Identity

このサーバーの識別子を入力します。

User certificate

クリックして、**Distinguished Encoding Rules**(DER) または **Privacy Enhanced Mail**(PEM) でエンコードされた個人の X.509 証明書ファイルを参照し、選択します。

CA certificate

クリックして、**Distinguished Encoding Rules**(DER) または **Privacy Enhanced Mail**(PEM) でエンコードされた X.509 の **認証局** 認証ファイルを参照し、選択します。

Private key

クリックして、**Distinguished Encoding Rules**(DER)、**Privacy Enhanced Mail**(PEM)、または **Personal Information Exchange Syntax Standard**(PKCS #12) でエンコードされた**秘密鍵** ファイルを参照し、選択します。

Private key password

Private key フィールドの秘密鍵のパスワードを入力します。**Show password** を選択すると、入力時にパスワードが表示されます。

PWD の設定

パスワード (PWD) により、ユーザー名とパスワードを指定できます。

Username

認証プロセスで使用するユーザー名を入力します。

Password

認証プロセスで使用するパスワードを入力します。

FAST の設定

FAST を設定するには、「control-center を使用した有線 (イーサネット) 接続の設定」 で説明される手順に従います。以下の設定が可能です。

Anonymous Identity

このサーバーの識別子を入力します。

Allow automatic PAC provisioning (自動 PAC プロビジョニングを許可する)

チェックボックスをオンにして有効にし、Anonymous (匿名)、Authenticated (認証済み)、および Both (両方) から選択します。

PAC file

クリックして、**Protected Access Credential**(**PAC**) ファイルを参照し、選択します。

Inner authentication

GTC - Generic Token Card (汎用トークンカード)
MSCHAPv2 - Microsoft Challenge Handshake Authentication Protocol version 2 (Microsoft チャレンジハンドシェイク認証プロトコル version 2)

Username

認証プロセスで使用するユーザー名を入力します。

Password

認証プロセスで使用するパスワードを入力します。

Tunneled TLS の設定

Tunneled TLS を設定するには、「control-center を使用した有線 (イーサネット) 接続の設定」 で説明されている手順に従います。以下の設定が可能です。

Anonymous identity

この値は、非暗号化 ID として使用されます。

CA certificate

クリックして、認証局の証明書を参照し、選択します。

Inner authentication

PAP - パスワード認証プロトコル
MSCHAP - チャレンジハンドシェイク認証プロトコル

MSCHAPv2 - Microsoft Challenge Handshake Authentication Protocol version 2 (Microsoft チャレンジハンドシェイク認証プロトコル version 2)

MSCHAPv2 (no EAP) - 拡張認証プロトコルなしの Microsoft Challenge Handshake Authentication Protocol version 2

CHAP - チャレンジハンドシェイク認証プロトコル

MD5 - Message Digest 5 (暗号学的ハッシュ関数)

GTC - Generic Token Card (汎用トークンカード)

Username

認証プロセスで使用するユーザー名を入力します。

Password

認証プロセスで使用するパスワードを入力します。

Protected EAP (PEAP) の設定

Protected EAP (PEAP) を設定するには、「control-center を使用した有線 (イーサネット) 接続の設定」 で説明されている手順に従います。以下の設定が可能です。

Anonymous Identity

この値は、非暗号化 ID として使用されます。

CA certificate

クリックして、認証局の証明書を参照し、選択します。

PEAP version

使用する、保護された EAP のバージョン。Automatic、0、1 のいずれか。

Inner authentication

MSCHAPv2 - Microsoft Challenge Handshake Authentication Protocol version 2 (Microsoft チャレンジハンドシェイク認証プロトコル version 2)
MD5 - Message Digest 5 (暗号学的ハッシュ関数)

GTC - Generic Token Card (汎用トークンカード)

Username

認証プロセスで使用するユーザー名を入力します。

Password

認証プロセスで使用するパスワードを入力します。

6.4. CONTROL-CENTER を使用した静的ルートの設定

GNOME で **control-center** を使用して、ネットワーク接続の設定に静的ルートを追加します。

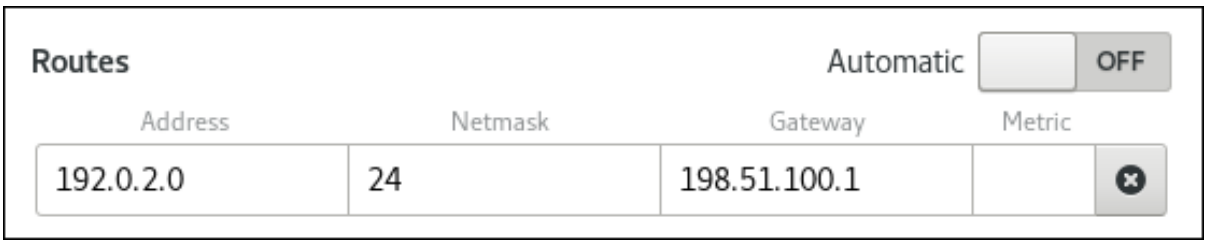
本セクションの手順では、**198.51.100.1** で実行しているゲートウェイを使用する **192.0.2.0/24** ネットワークにルートを追加する方法を説明します。

前提条件

- ネットワークが設定されている。
- 静的ルートのゲートウェイが、インターフェース上で直接到達できる。
- **control-center** アプリケーションで接続のネットワーク設定が開いている。「control-center を使用したネットワーク接続の設定」を参照してください。

手順

1. **IPv4** タブを開きます。
2. 必要に応じて、**IPv4** タブの **Routes** セクションの **On** ボタンをクリックして自動ルートを無効にし、静的ルートのみを使用します。自動ルートが有効になっている場合は、Red Hat Enterprise Linux が静的ルートと、DHCP サーバーから受け取ったルートを使用します。
3. アドレス、ネットマスク、ゲートウェイを入力します。必要に応じて、メトリック値を入力します。



4. **Apply** をクリックします。
5. **Network** ウィンドウに戻り、接続のボタンを **Off** に切り替えて **On** に戻して、接続を無効にして再度有効にし、変更を適用します。

警告

接続を再起動すると、そのインターフェースの接続が一時的に中断します。

6. 必要に応じて、ルートがアクティブであることを確認します。

```
$ ip route
...
192.0.2.0/24 via 198.51.100.1 dev example proto static metric 100
```

6.5. NM-CONNECTION-EDITOR を使用した静的ルートの設定

nm-connection-editor アプリケーションを使用して、ネットワーク接続の設定に静的ルートを追加できます。

本セクションの手順では、**198.51.100.1** で実行しているゲートウェイを使用する **192.0.2.0/24** ネットワークにルートを追加する方法を説明します。これは、**example** 接続から到達可能です。

前提条件

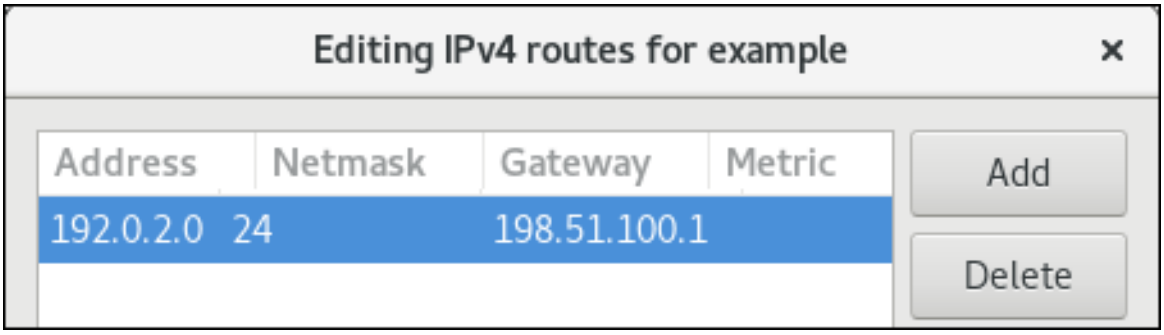
- ネットワークが設定されている。
- 静的ルートのゲートウェイが、インターフェース上で直接到達できる。

手順

1. ターミナルを開き、**nm-connection-editor** と入力します。

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

2. **example** 接続を選択し、歯車アイコンをクリックして、既存の接続を変更します。
3. **IPv4** タブを開きます。
4. **Routes** ボタンをクリックします。
5. **Add** ボタンをクリックして、アドレス、ネットマスク、ゲートウェイを入力します。必要に応じてメトリック値を入力します。



6. **OK** をクリックします。

7. **Save** をクリックします。

8. ネットワーク接続を再起動して、変更を有効にします。たとえば、コマンドラインで **example** を再起動するには、以下を実行します。

```
$ sudo nmcli connection up example
```

9. 必要に応じて、ルートがアクティブであることを確認します。

```
$ ip route
...
192.0.2.0/24 via 198.51.100.1 dev example proto static metric 100
```

6.6. CONTROL-CENTER を使用して、既存の接続でフォルトのゲートウェイの設定

ほとんどの場合、管理者は、たとえば 「control-center を使用したネットワーク接続の設定」 の説明に従って、接続を作成する際にデフォルトのゲートウェイを設定します。

本セクションでは、**control-center** アプリケーションを使用して作成されている接続で、デフォルトのゲートウェイを設定または更新する方法を説明します。

前提条件

- デフォルトゲートウェイが設定される接続で、静的 IP アドレスを少なくとも 1 つ設定している。
- 接続のネットワーク設定は、**control-center** アプリケーションで開きます。「control-center を使用したネットワーク接続の設定」を参照してください。

手順

1. IPv4 デフォルトゲートウェイを設定します。たとえば、その接続のデフォルトゲートウェイの IPv4 アドレスを **192.0.2.1** に設定します。

   a. **IPv4** タブを開きます。

   b. そのゲートウェイのアドレスが含まれる IP アドレスの範囲の隣の **gateway** フィールドにアドレスを入力します。

2. IPv6 デフォルトゲートウェイを設定します。たとえば、接続のデフォルトゲートウェイの IPv6 アドレスを **2001:db8::1** に設定するには、以下を実行します。

   a. **IPv6** タブを開きます。

   b. そのゲートウェイのアドレスが含まれる IP アドレスの範囲の隣の **gateway** フィールドにアドレスを入力します。

3. **Apply** をクリックします。

4. **Network** ウィンドウに戻り、接続のボタンを **Off** に切り替えて **On** に戻して、接続を無効にして再度有効にし、変更を適用します。

> **警告**
>
> このネットワーク接続を使用しているすべての接続が、再起動時に一時的に中断されます。

5. 必要に応じて、ルートがアクティブであることを確認します。
IPv4 デフォルトゲートウェイを表示するには、以下を実行します。

```
$ ip -4 route
default via 192.0.2.1 dev example proto static metric 100
```

IPv6 デフォルトゲートウェイを表示するには、以下を実行します。

```
$ ip -6 route
default via 2001:db8::1 dev example proto static metric 100 pref medium
```

関連資料

- 「control-center を使用したネットワーク接続の設定」

6.7. USING NM-CONNECTION-EDITOR を使用して、既存の接続でデフォルトのゲートウェイの設定

ほとんどの場合、管理者は、たとえば 「control-center を使用したネットワーク接続の設定」 の説明に従って、接続を作成する際にデフォルトのゲートウェイを設定します。

本セクションでは、**nm-connection-editor** アプリケーションを使用して作成されている接続で、デフォルトのゲートウェイを設定または更新する方法を説明します。

前提条件

- デフォルトゲートウェイが設定される接続で、静的 IP アドレスを少なくとも 1 つ設定している。

手順

1. ターミナルを開き、**nm-connection-editor** と入力します。

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

2. 変更する接続を選択し、既存の接続を編集する歯車のアイコンをクリックします。

3. IPv4 デフォルトゲートウェイを設定します。たとえば、その接続のデフォルトゲートウェイの IPv4 アドレスを **192.0.2.1** に設定します。

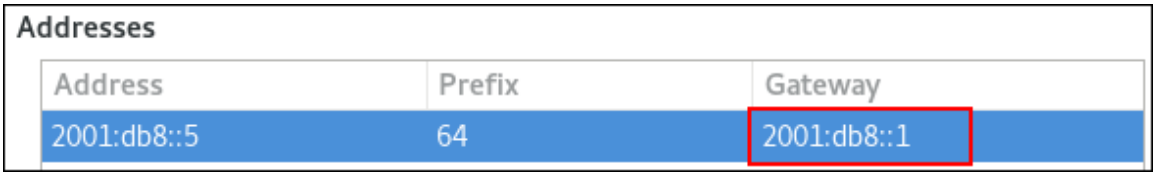
   a. **IPv4 Settings** タブを開きます。

   b. そのゲートウェイのアドレスが含まれる IP アドレスの範囲の隣の **gateway** フィールドにアドレスを入力します。

4. IPv6 デフォルトゲートウェイを設定します。たとえば、接続のデフォルトゲートウェイの IPv6 アドレスを **2001:db8::1** に設定するには、以下を実行します。

   a. **IPv6** タブを開きます。

   b. そのゲートウェイのアドレスが含まれる IP アドレスの範囲の隣の **gateway** フィールドにアドレスを入力します。



5. **OK** をクリックします。

6. **Save** をクリックします。

7. ネットワーク接続を再起動して、変更を有効にします。たとえば、コマンドラインで **example** 接続を再起動するには、以下を実行します。

```
$ sudo nmcli connection up example
```

警告

このネットワーク接続を使用しているすべての接続が、再起動時に一時的に中断されます。

8. 必要に応じて、ルートがアクティブであることを確認します。
   IPv4 デフォルトゲートウェイを表示するには、以下を実行します。

```
$ ip -4 route
default via 192.0.2.1 dev example proto static metric 100
```

   IPv6 デフォルトゲートウェイを表示するには、以下を実行します。

```
$ ip -6 route
default via 2001:db8::1 dev example proto static metric 100 pref medium
```

関連資料

- 「control-center を使用したネットワーク接続の設定」

第7章 MACSEC の設定

次のセクションでは、イーサネットリンクのすべてのトラフィックで、安全な通信のために、 802.1AE IEEE 標準セキュリティー技術である **MACsec** (**Media Control Access Security**) を設定する方法を説明します。

7.1. MACSEC の概要

MACsec (**Media Access Control Security**) (IEEE 802.1AE) は、GCM-AES-128 アルゴリズムを使用して LAN 上のすべてのトラフィックを暗号化して認証します。**MACsec** は、**IP** だけでなく、ARP (Address Resolution Protocol)、ND (Neighbor Discovery)、または **DHCP** も保護します。**IPsec** はネットワーク層 (レイヤー 3) で機能しますが、**SSL** または **TLS** はアプリケーション層 (レイヤー 7) で機能し、**MACsec** はデータリンク層 (レイヤー 2) で機能します。**MACsec** を、その他のネットワーク層のセキュリティープロトコルと組み合わせて、これらの標準が提供する異なるセキュリティー機能を活用します。

7.2. NMCLI ツールを使用した MACSEC の使用

この手順は、**nmcli** ツールを使用して **MACsec** を設定する方法を説明します。

前提条件

- NetworkManager が実行している。
- 16 バイトの 16 進数表記 CAK (**$MKA_CAK**) と、32 バイトの 16 進数表記 CKN (**$MKA_CKN**) がある。

手順

```
~]# nmcli connection add type macsec \
  con-name test-macsec+ ifname macsec0 \
  connection.autoconnect no \
  macsec.parent eth0 macsec.mode psk \
  macsec.mka-cak $MKA_CAK \
  macsec.mka-ckn $MKA_CKN

~]# nmcli connection up test-macsec+
```

これにより、macsec0 デバイスが設定され、ネットワークに使用できます。

7.3. WPA_SUPPLICANT を使用した MACSEC の使用

この手順は、事前に共有された CAK/CKN (Connectivity Association Key/CAK Name) のペアを使用して認証を実行するスイッチを使用して **MACsec** を有効にする方法を説明します。

手順

1. CAK/CKN ペアを作成します。たとえば、次のコマンドにより、16 バイトのキーが 16 進数表記で生成されます。

   ```
   ~]$ dd if=/dev/urandom count=16 bs=1 2> /dev/null | hexdump -e '1/2 "%02x"'
   ```

2. **wpa_supplicant.conf** 設定ファイルを作成し、次の行を追加します。

```
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
}
```

wpa_supplicant.conf 設定ファイルの **mka_cak** 行および **mka_ckn** 行には、前のステップの値を使用します。

詳細は man ページの **wpa_supplicant.conf(5)** を参照してください。

3. ネットワークに接続するには、**eth0** を使用し、次のコマンドを実行して **wpa_supplicant** を起動します。

```
~]# wpa_supplicant -i eth0 -Dmacsec_linux -c wpa_supplicant.conf
```

7.4. 関連情報

詳細は「What's new in MACsec: setting up MACsec using wpa_supplicant and (optionally) NetworkManager」を参照してください。また、**MACsec** ネットワークのアーキテクチャー、ユースケースシナリオ、設定例の詳細は「MACsec: a different solution to encrypt network traffic」 を参照してください。

第8章 IPVLAN の使用

ここでは、IPVLAN ドライバーについて説明します。

8.1. IPVLAN の概要

IPVLAN は、仮想ネットワークデバイス用のドライバーで、コンテナ環境でホストネットワークにアクセスするのに使用できます。IPVLAN は外部ネットワークに対し、ホストネットワーク内で作成された IPVLAN デバイスの数に関わらず、単一の MAC アドレスを公開します。つまり、ユーザーは複数コンテナに複数の IPVLAN デバイスを持つことができますが、対応するスイッチは MAC アドレスを 1 つ読み込むということです。IPVLAN ドライバーは、ローカルスイッチで管理できる MAC アドレスの数に制限がある場合に役立ちます。

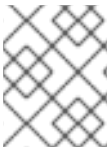
8.2. IPVLAN モード

IPVLANでは、次のモードを使用できます。

- **L2 モード**
  IPVLAN の **L2 モード** では、仮想デバイスは アドレス解決プロトコル (ARP) リクエストを受信して応答します。**netfilter** フレームワークは、仮想デバイスを所有するコンテナ内でのみ動作します。**netfilter** チェーンは、コンテナ化したトラッフィクにあるデフォルトの名前空間では実行されません。**L2 モード**を使用すると、高いパフォーマンスが得られますが、ネットワークトラフィックの制御性は低下します。

- **L3 モード**
  L3 モードでは、仮想デバイスは **L3** 以上のトラフィックのみを処理します。仮想デバイスは ARP リクエストに応答せず、関連するピアの IPVLAN IP アドレスは、隣接エントリーをユーザーが手動で設定する必要があります。関連するコンテナの送信トラフィックはデフォルトの名前空間の **netfilter** POSTROUTING および OUTPUT チェーンに到達する一方、受信トラフィックは **L2 モード**と同様にスレッド化されます。**L3 モード** を使用すると、高い制御性が得られますが、ネットワークトラフィックのパフォーマンスは低下します。

- **L3S モード**
  L3S モードでは、仮想デバイスは **L3 モード**と同様の処理をしますが、関連するコンテナの送信トラフィックと受信トラフィックの両方がデフォルトの名前空間にある **netfilter** チェーンに到達する点が異なります。**L3S モード** は、**L3 モード** と類似した動作を行いますが、ネットワークの制御性に優れています。



注記

IPVLAN 仮想デバイスは、**L3 モード** および **L3S モード**では、ブロードキャストトラフィックおよびマルチキャストトラフィックを受信しません。

第9章 IPVLAN ネットワークの設定

9.1. IPROUTE2 を使用した IPVLAN デバイスの作成および設定

この手順では、iproute2 を使用して IPVLAN デバイスを設定する方法を説明します。

手順

1. IPVLAN デバイスを作成するには、以下のコマンドを実行します。

```
~]# ip link add link real_NIC_device name IPVLAN_device type ipvlan mode l2
```

ネットワークインターフェースコントローラー (NIC) は、コンピューターをネットワークに接続するハードウェアコンポーネントです。

例9.1 IPVLAN デバイスの作成

```
~]# 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. **IPv4** または **IPv6** アドレスをインターフェースに割り当てるには、以下のコマンドを実行します。

```
~]# ip addr add dev IPVLAN_device IP_address/subnet_mask_prefix
```

3. **L3 モード** または **L3S モード** の IPVLAN デバイスを設定する場合は、以下の設定を行います。

   a. リモートホスト上のリモートピアのネイバー設定を行います。

   ```
   ~]# ip neigh add dev peer_device IPVLAN_device_IP_address lladdr MAC_address
   ```

   MAC_address は、IPVLAN デバイスのベースである実際の NIC の MAC アドレスになります。

   b. **L3 モード** の IPVLAN デバイスを設定する場合は、以下のコマンドを実行します。

   ```
   ~]# ip neigh add dev real_NIC_device peer_IP_address lladdr peer_MAC_address
   ```

   L3S モード の場合は、以下のコマンドを実行します。

   ```
   ~]# ip route dev add real_NIC_device peer_IP_address/32
   ```

   IP アドレスは、リモートピアのアドレスを使用します。

4. IPVLAN デバイスをアクティブに設定するには、以下のコマンドを実行します。

```
~]# ip link set dev IPVLAN_device up
```

5. IPVLAN デバイスがアクティブであることを確認するには、リモートホストで以下のコマンドを実行します。

```
~]# ping IP_address
```

IP_address には、IPVLAN デバイスの IP アドレスを使用します。

第10章 VRF の使用

記載	アセンブリーの一覧
ユーザーストーリー	ネットワーク管理者として、このようなマルチテナントサーバーなどの特定のルートおよびアドレスとのネットワークトラフィックを分離できるようにする必要があります。
Jira	https://projects.engineering.redhat.com/browse/RHELPLAN-8072
BZ	https://bugzilla.redhat.com/show_bug.cgi?id=1688125
SME	Paolo Abeni <pabeni@redhat.com>
SME Ack	はい
Peer Ack	はい

VRF (Virtual Routing and Forwarding) を使用すると、管理者は、同じホストで複数のルーティングテーブルを同時に使用できます。このため、VRF は、レイヤー 3 でネットワークをパーティションで区切ります。これにより、管理者は、VRF ドメインごとに個別の独立したルートテーブルを使用してトラフィックを分離できるようになります。この技術は、レイヤー 2 でネットワークのパーティションを作成する仮想 LAN (VLAN) に類似しており、ここではオペレーティングシステムシステムが異なる VLAN タグを使用して、同じ物理メディアを共有するトラフィックを分離させます。

レイヤー 2 のパーティション上にある VRF の利点は、ルーティングが関与するピアの数に対して、適切にスケーリングすることです。

Red Hat Enterprise Linux は、各 VRF ドメインに仮想 **vrt** デバイスを使用し、既存のネットワークデバイスを VRF デバイスにスレーブにして、VRF ドメインにルートを追加します。スレーブになったデバイスに接続していたアドレスとルートは、VRF ドメイン内に移動します。

各 VRF ドメインが互いに分離されていることに注意してください。

関連情報

- https://www.kernel.org/doc/Documentation/networking/vrf.txt

第11章 FIREWALLD の使用および設定

記載	セキュリティーの設定および管理
ユーザーストーリー	* システム管理者として、着信ネットワークトラフィックを守る必要があります。* システム管理者として、システムで特定のアプリケーションを許可したりブロックしたりできるように、ポートのブロックやブロック解除ができるようにしたいです。* 熟練のシステム管理者として、パフォーマンスを最適化し、パケットに対しブロックしたり遅延をかけたりできるよう、着信トラフィックをモニターして制御したいです。* システム管理者として、特定のサービス (Web サービス と SSH など) のみを許可し、その他のすべてのアプリケーションをブロックしたいです。* 熟練のシステム管理者として、ネットワークトラフィックを詳細に制御する設定を行う必要があります。
JIRA	https://projects.engineering.redhat.com/browse/RHELPLAN-2918 https://projects.engineering.redhat.com/browse/RHELPLAN-9159
BZ	https://bugzilla.redhat.com/show_bug.cgi?id=1601541
SME	egarver@redhat.com, todoleza@redhat.com, psutter@redhat.com
SME Ack	firewalld: YES, nftables: YES
Peer Ack	なし

firewall は、外部からの不要なトラフィックからマシンを保護する方法です。**ファイアウォールルール** セットを定義することで、ホストマシンに着信ネットワークトラフィックを制御できます。このようなルールは、着信トラフィックを分類して、拒否または許可するために使用されます。

11.1. FIREWALLD の使用

11.1.1. firewalld

firewalld は、**D-Bus** インターフェースを使用して、動的にカスタマイズできるホストベースのファイアウォールを提供するファイアウォールサービスデーモンです。ルールが変更するたびに、ファイアウォールデーモンを再起動しなくても、ルールの作成、変更、および削除を動的に可能にします。

firewalld は、**ゾーン** および **サービス** の概念を使用し、トラフィック管理を簡素化します。ゾーンは、事前定義したルールセットです。ネットワークインターフェースおよびソースはゾーンに割り当てることができます。許可されているトラフィックは、コンピューターが接続するネットワークと、このネットワークが割り当てられているセキュリティーレベルに従います。ファイアウォールサービスは、特定のサービスに着信トラフィックを許可するのに必要な設定を扱う事前定義のルールで、ゾーンに適用されます。

サービスは、ネットワーク接続に 1 つ以上の **ポート** または **アドレス** を使用します。ファイアウォールは、ポートに基づいて接続のフィルターを設定します。サービスに対してネットワークトラフィックを許可するには、そのポートを **開く** 必要があります。**firewalld** は、明示的に開いていないポートのトラフィックをすべてブロックします。**trusted** などのゾーンで、デフォルトですべてのトラフィックを許可します。

関連資料
以下の資料は、**firewalld** に関する関連資料を提供します。

インストールされているドキュメント

- **firewalld(1)** の man ページ - **firewalld** のコマンドオプションが説明されています。
- **firewalld.conf(5)** の man ページ - **firewalld** を設定する情報が含まれます。
- **firewall-cmd(1)** の man ページ - **firewalld** コマンドラインクライアントのコマンドオプションが説明されています。
- **firewall-config(1)** の man ページ - firewall-config ツールの設定が説明されています。
- **firewall-offline-cmd(1)** の man ページ - **firewalld** オフラインコマンドラインクライアントのコマンドオプションが説明されています。
- **firewalld.icmptype(5)** の man ページ - **ICMP** フィルタリングの XML 設定ファイルが説明されています。
- **firewalld.ipset(5)** の man ページ - **firewalld IP** セットの XML 設定ファイルが説明されています。
- **firewalld.service(5)** の man ページ - firewalld service 用の XML 設定ファイルが説明されています。
- **firewalld.zone(5)** の man ページ - **firewalld** ゾーン設定の XML 設定ファイルが説明されています。
- **firewalld.direct(5)** の man ページ - **firewalld** ダイレクトインターフェース設定ファイルが説明されています。
- **firewalld.lockdown-whitelist(5)** の man ページ - **firewalld** ロックダウンホワイトリスト設定ファイルが説明されています。
- **firewalld.richlanguage(5)** の man ページ - **firewalld** リッチ言語ルール構文が説明されています。
- **firewalld.zones(5)** の man ページ - ゾーンの全般的な説明と設定方法が説明されています。
- **firewalld.dbus(5)** の man ページ - **firewalld** の **D-Bus** インターフェースが説明されています。

オンラインのドキュメント

- http://www.firewalld.org/ - **firewalld** ホームページ

11.1.2. ゾーン

firewalld は、インターフェースに追加する信頼レベルと、そのネットワークのトラフィックに従って、複数のネットワークを複数のゾーンに分類できます。接続は、1 つのゾーンにしか指定できませんが、ゾーンは多くのネットワーク接続に使用できます。

NetworkManager は、インターフェースのゾーンに **firewalld** を通知します。インターフェースにゾーンを割り当てるには、NetworkManager と、firewall-config ツールまたは **firewall-cmd** コマンドラインツールを使用して割り当てられます。後者の 2 つは、適切な NetworkManager 設定ファイルの編集のみを行います。**firewall-cmd** または firewall-config を使用してインターフェースのゾーンを変更する場合、リクエストは NetworkManager に転送され、**firewalld** には処理されません。

事前定義したゾーンは **/usr/lib/firewalld/zones/** ディレクトリーに保存され、利用可能なネットワークインターフェースに即座に適用されます。このファイルは、修正しないと **/etc/firewalld/zones/** ディレクトリーにコピーされません。事前定義したゾーンのデフォルト設定は以下のようになります。

block

IPv4 の場合は icmp-host-prohibited メッセージ、 **IPv6** の場合は icmp6-adm-prohibited メッセージで、すべての着信ネットワーク接続が拒否されます。システム内で開始したネットワーク接続のみが可能です。

dmz

公開アクセスは可能ですが、内部ネットワークへのアクセスに制限がある非武装地帯にあるコンピューター用。選択された着信接続のみが許可されます。

drop

着信ネットワークパケットは、通知なしで遮断されます。発信ネットワーク接続だけが可能です。

external

マスカレードを特別にルーター用に有効にした外部ネットワーク上での使用向けです。自分のコンピューターを保護するため、ネットワーク上の他のコンピューターを信頼しません。選択された着信接続のみが許可されます。

home

そのネットワークでその他のコンピューターをほぼ信頼できる自宅での使用。選択した着信接続のみが許可されます。

internal

そのネットワークでその他のコンピューターをほぼ信頼できる内部ネットワークでの使用。選択した着信接続のみが許可されます。

public

そのネットワークでその他のコンピューターを信頼できないパブリックエリアでの使用。選択した着信接続のみが許可されます。

trusted

すべてのネットワーク接続が許可されます。

work

そのネットワークで、その他のコンピューターをほぼ信頼できる職場での使用。選択した着信接続のみが許可されます。

これらのゾーンのいずれかを **デフォルト** に設定できます。インターフェース接続を NetworkManager に追加すると、デフォルトゾーンに割り当てられます。**firewalld** のデフォルトゾーンは、インストール時に **public** ゾーンに設定されます。デフォルトゾーンは変更できます。

注記

ネットワークゾーンは、分かりやすく、ユーザーが妥当な決定をすばやく下せるような名前が付けられています。セキュリティー問題を回避するために、ユーザーのニーズおよびリスク評価に合わせて、デフォルトゾーンの設定の見直しを行ったり、不要なサービスを無効にしてください。

11.1.3. 事前定義サービス

サービスは、ローカルポート、プロトコル、ソースポート、宛先、そしてサービスが有効になると自動的にロードされるファイアウォールヘルパーモジュールの一覧になります。サービスを使用すると、ポートのオープン、プロトコルの定義、パケット転送などを 1 つ 1 つ行うのではなく、1 回のステップで定義できます。

サービス設定オプションと、一般的なファイル情報は、man ページの **firewalld.service(5)** で説明されています。サービスは、個々の XML 設定ファイルを使用して指定し、名前は、**service-name.xml** のような形式になります。プロトコル名は、**firewalld** のサービス名またはアプリケーション名よりも優先されます。

グラフィカルな **firewall-config** ツール、**firewall-cmd**、および **firewall-offline-cmd** を使用してサービスを追加および削除できます。

または、**/etc/firewalld/services/** ディレクトリーの XML ファイルを変更できます。ユーザーがサービスを追加または変更しない場合は、対応する XML ファイルが **/etc/firewalld/services/** では見つかりません。**/usr/lib/firewalld/services/** ディレクトリーのファイルは、サービスを追加または変更する場合にテンプレートとして使用できます。

11.1.4. ランタイムおよび永続化設定

runtime モードで行った変更は、**firewalld** が実行している間しか適用されません。**firewalld** を再起動すると、設定内容は **永続的な** 値に戻ります。

変更した内容を再起動後も持続させるには、**--permanent** オプションを使用します。**firewalld** が実行している間だけ変更を持続させる場合は、**--runtime-to-permanent firewall-cmd** オプションを実行します。

--permanent オプションのみを使用して **firewalld** を実行している場合にルールを設定するには、**firewalld** が再起動するまで有効にはなりません。ただし、**firewalld** を再起動すると、開いているポートがすべて閉じ、ネットワーキングトラフィックを停止します。

11.1.4.1. CLI を使用したランタイムおよび永続設定の設定の変更

CLI では、2 つのモードのファイアウォール設定を同時に修正することはできません。CLI では、ランタイムまたは永続モードを修正します。永続化設定でファイアウォール設定を修正するには、**firewall-cmd** コマンドで **--permanent** オプションを使用します。

```
# firewall-cmd --permanent <other options>
```

このオプションを使用しないと、コマンドはランタイムモードを変更します。

両方のモードで設定を変更するには、2 つの方法を使用できます。

1. 以下のように、ランタイム設定を変更して、永続化します。

```
# firewall-cmd <other options>
# firewall-cmd --runtime-to-permanent
```

2. 永続的な設定を行い、ランタイムモードで設定を再ロードします。

```
# firewall-cmd --permanent <other options>
# firewall-cmd --reload
```

最初の方法では、永続化モードで設定を適用する前に、設定をテストできます。

注記

特にリモートシステムでは、設定を誤ると、ユーザーが自身をロックする結果となります。そのような状況を回避するには、**--timeout** オプションを使用します。指定した時間が経つと、変更は元に戻ります。このオプションを使用した場合は、**--permanent** オプションが無効になります。

たとえば、15 分間 **SSH** サービスを追加するには、以下のコマンドを実行します。

```
# firewall-cmd --add-service=ssh --timeout 15m
```

11.2. FIREWALL-CONFIG GUI 設定ツールのインストール

firewall-config GUI 設定ツールを使用するには、**firewall-config** パッケージをインストールします。

手順

1. **root** で以下のコマンドを実行します。

```
# yum install firewall-config
```

また、GNOME では、**Super** キーを使用して **Software** と入力し、Software Sources アプリケーションを起動します。右上端で検索ボタンを選択すると表示される検索ボックスに **firewall** を入力します。検索結果から **Firewall** アイテムを選択し、**Install** ボタンをクリックします。

2. **firewall-config** を実行するために、**firewall-config** コマンドを実行するか、**Super** キーを押して **Activities Overview** を開き、**firewall** 入力して **Enter** を押します。

11.3. FIREWALLD の現在のステータスおよび設定の表示

11.3.1. firewalld の現在のステータスの表示

ファイアウォールサービス **firewalld** がデフォルトでシステムにインストールされています。**firewalld** CLI インターフェースを使用して、サービスが実行していることを確認します。

手順

1. サービスのステータスを表示するには、以下のコマンドを実行します。

```
# firewall-cmd --state
```

2. サービスステータスの詳細は **systemctl status** サブコマンドを実行します。

```
# 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
```

関連資料
設定を編集する前に、**firewalld** の設定方法と、強制するルールを確認することが重要です。ファイアウォール設定を表示するには、「現在の firewalld 設定の表示」 を参照してください。

11.3.2. 現在の firewalld 設定の表示

11.3.2.1. GUI を使用して許可されるサービスの表示

グラフィカルの **firewall-config** ツールを使用してサービスの一覧を表示するには、**Super** キーを押して Activities Overview を開き、**firewall** と入力して **Enter** を押します。**firewall-config** ツールが表示され、**サービス** タブでサービスの一覧を確認できます。

もしくは、コマンドラインを使用してグラフィカルなファイアウォール設定ツールを開始するには、以下のコマンドを入力します。

```
$ firewall-config
```

Firewall Configuration ウィンドウが開きます。このコマンドは通常ユーザーとして実行できますが、監理者パスワードが求められる場合もあります。

11.3.2.2. CLI を使用した firewalld 設定の表示

CLI クライアントで、現在のファイアウォール設定を、複数の方法で表示できます。**--list-all** オプションは、**firewalld** 設定の完全概要を表示します。

firewalld は、ゾーンを使用してトラフィックを管理します。**--zone** オプションでゾーンを指定しないと、コマンドは、アクティブネットワークインターフェース及び接続に割り当てたデフォルトゾーンに対して有効になります。

デフォルトゾーンに関連する情報をすべて表示するには、以下のコマンドを実行します。

```
# 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:
```

設定を表示するゾーンを指定するには、たとえば、**--zone=zone-name** 引数を **firewall-cmd --list-all** コマンドに追加します。

```
# firewall-cmd --list-all --zone=home
home
  target: default
  icmp-block-inversion: no
  interfaces:
```

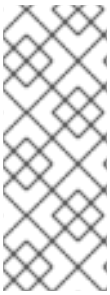
```
  sources:
  services: ssh mdns samba-client dhcpv6-client
... [output truncated]
```

サービス、ポートなど、特定情報の設定を確認するには、特定のオプションを使用します。man ページの **firewalld** か、コマンドヘルプを使用してオプションの一覧を表示します。

```
# 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
... [output truncated]
```

たとえば、現在のゾーンで許可されているサービスを表示します。

```
# firewall-cmd --list-services
ssh dhcpv6-client
```



注記

CLI ツールを使用して一覧表示した特定のサブパートの設定は、解釈が難しいことがしばしばあります。たとえば、**firewalld** で **SSH** サービスを許可し、そのサービスに必要なポート (22) を開したあと、許可されたサービスを一覧表示すると、一覧には **SSH** サービスが表示されますが、開いているポートを一覧表示しても、何も表示されません。したがって、**--list-all** オプションを使用して、完全な情報を取得することが推奨されます。

11.4. FIREWALLD の起動

手順

1. **firewalld** を開始するには、**root** で以下のコマンドを実行します。

```
# systemctl unmask firewalld
# systemctl start firewalld
```

2. システムの起動時に **firewalld** を自動的に起動するように設定するには、**root** で以下のコマンドを実行します。

```
# systemctl enable firewalld
```

11.5. FIREWALLD の停止

手順

1. **firewalld** を停止するには、**root** で以下のコマンドを実行します。

1. **firewalld** を停止するには、**root** で以下のコマンドを実行します。

```
# systemctl stop firewalld
```

2. システムの起動時に **firewalld** を自動的に起動しないように設定するには、以下を行います。

```
# systemctl disable firewalld
```

3. **firewalld D-Bus** インターフェースにアクセスして firewalld を起動していないこと、そしてその他のサービスが **firewalld** を求めているかどうかを確認するには、以下を行います。

```
# systemctl mask firewalld
```

11.6. FIREWALLD を使用したネットワークトラフィックの制御

11.6.1. 緊急時に CLI を使用してすべてのトラフィックの無効化

システムへの攻撃など、緊急な状態では、すべてのネットワークトラフィックを無効にし、攻撃を遮断できます。

手順

1. ネットワークトラフィックを直ちに無効にするには、パニックモードをオンにします。

```
# firewall-cmd --panic-on
```

重要

パニックモードを有効にすると、ネットワークトラフィックがすべて停止します。したがって、そのマシンへの物理アクセスがある場合、またはシリアルコンソールを使用してログインする場合に限り使用してください。

パニックモードをオフにし、ファイアウォールを永続設定に戻します。パニックモードを無効にするには、以下のコマンドを実行します。

```
# firewall-cmd --panic-off
```

パニックモードを有効または無効にするには、以下のコマンドを実行します。

```
# firewall-cmd --query-panic
```

11.6.2. CLI を使用して事前定義されたサービスでトラフィックの制御

トラフィックを制御する最も簡単な方法は、事前定義したサービスを **firewalld** に追加する方法です。これは、必要なすべてのポートを開き、**service definition file** に従ってその他の設定を変更します。

手順

1. サービスが許可されていないことを確認します。

```
# firewall-cmd --list-services
ssh dhcpv6-client
```

2. 事前定義したサービスの一覧を表示します。

```
# 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 ...
[output truncated]
```

3. サービスを、許可されたサービスに追加します。

```
# firewall-cmd --add-service=<service-name>
```

4. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

11.6.3. GUI を使用した事前定義サービスでトラフィックの制御

事前定義したサービスまたはカスタマイズしたサービスを有効または無効にするには、以下を行います。

1. **firewall-config** ツールを起動して、サービスを設定するネットワークゾーンを選択します。
2. **Services** タブを選択します。
3. 信頼するサービスのチェックボックスを選択してください。ブロックするサービスのチェックボックスは選択を解除してください。

サービスを編集するには、以下を行います。

1. **firewall-config** ツールを起動します。
2. **Configuration** メニューから **Permanent** を選択します。**Services** ウィンドウの下部に、その他のアイコンおよびメニューボタンが表示されます。
3. 設定するサービスを選択します。

ポート、**プロトコル**、**ソースポート** のタブでは、選択したサービスのポート、プロトコル、およびソースポートの追加、変更、削除が可能です。モジュールタブは、**Netfilter** ヘルパーモジュールの設定を行います。**Destination** タブは、特定の送信先アドレスとインターネットプロトコル (**IPv4** または **IPv6**) へのトラフィックが制限できます。

注記

実行時 モードでは、サービス設定を変更できません。

11.6.4. 新しいサービスの追加

サービスは、グラフィカルな **firewall-config** ツールと、**firewall-cmd** および **firewall-offline-cmd** を使用して追加または削除できます。または、**/etc/firewalld/services/** にある XML ファイルを編集できます。ユーザーがサービスを追加または変更しないと、対応する XML が **/etc/firewalld/services/** に作成されません。**/usr/lib/firewalld/services/** のファイルは、サービスを追加または変更する際にテンプレートとして使用できます。

手順
firewalld がアクティブでない場合に、ターミナルで新しいサービスを追加するには、**firewall-cmd** または **firewall-offline-cmd** を使用します。

1. 新しい、空のサービスを追加するには、次のコマンドを実行します。

```
$ firewall-cmd --new-service=service-name
```

2. ローカルファイルを使用して新規サービスを追加するには、以下のコマンドを使用します。

```
$ firewall-cmd --new-service-from-file=service-name.xml
```

--name=service-name オプションを指定して、サービス名を変更できます。

3. サービス設定を変更すると、直ちにサービスの更新コピーが **/etc/firewalld/services/** に作成できます。
root で次のコマンドを実行して、サービスを手動でコピーします。

```
# cp /usr/lib/firewalld/services/service-name.xml /etc/firewalld/services/service-name.xml
```

firewalld は、最初に **/usr/lib/firewalld/services** からファイルをロードします。**/etc/firewalld/services** にファイルが置かれ、そのファイルが有効な場合は、**/usr/lib/firewalld/services** で一致するファイルを上書きします。**/usr/lib/firewalld/services** で上書きしたファイルは、**/etc/firewalld/services** で一致するファイルが削除されるとすぐに、もしくはサービスのデフォルトをロードするように **firewalld** が求められた場合に使用されます。これに該当するのは永続環境のみです。ランタイム環境でフォールバックさせるには、再読み込みが必要です。

11.6.5. CLI を使用したポートの制御

ポートは、オペレーティングシステムが、ネットワークトラフィックを受信し、区別し、システムサービスに従って転送する論理デバイスです。これは、通常、ポートをリッスンするデーモンにより示されますが、このポートに入るトラフィックを待ちます。

通常、システムサービスは、サービスに予約されている標準ポートでリッスンします。**httpd** デーモンでは、たとえばポート 80 をリッスンします。ただし、デフォルトでは、システム管理者は、セキュリティーを強化するため、またはその他の理由により、別のポートをリッスンするようにデーモンを設定します。

11.6.5.1. ポートを開く

開かれたポートを介して、システムが外部からアクセスできます。これはセキュリティーリスクでもあります。一般的に、ポートを閉じたままにし、特定サービスに要求される場合に限り開きます。

手順
現在のゾーンで開かれたポートの一覧を表示するには、以下を行います。

1. 許可されているポートを一覧表示します。

```
# firewall-cmd --list-ports
```

2. 許可されているポートにポートを追加して、着信トラフィックに対して開きます。

```
# firewall-cmd --add-port=port-number/port-type
```

3. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

ポートタイプは、**tcp**、**udp**、**sctp**、または **dccp** になります。このタイプは、ネットワーク接続の種類と一致させる必要があります。

11.6.5.2. ポートを閉じる

開かれたポートが必要なくなった場合に、**firewalld** のポートを閉じます。ポートをそのままにするとセキュリティーリスクとなるため、使用されなくなったらすぐに不要なポートを閉じることが強く推奨されます。

手順
ポートを閉じるには、許可されているポートの一覧からそれを削除します。

1. 許可されているポートを一覧表示します。

```
# 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. 「許可されているポート」からポートを削除し、着信トラフィックに対して閉じます。

```
# firewall-cmd --remove-port=port-number/port-type
```

3. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

11.6.6. GUI を使用してポートを開く

ファイアウォールを経由して特定のポートに向かうトラフィックを許可するには、以下を行います。

1. **firewall-config** ツールを起動して、設定を変更するネットワークゾーンを選択します。
2. 右側の **Ports** タブを選択し、**Add** ボタンをクリックします。**Port and Protocol** ウィンドウが開きます。
3. 許可するポート番号またはポートの範囲を入力します。
4. リストから **tcp** または **udp** を選択します。

11.6.7. GUI を使用してプロトコルを使用したトラフィックの制御

特定のプロトコルを使用してファイアウォールを経由したトラフィックを許可するには、以下を行います。

1. **firewall-config** ツールを起動して、設定を変更するネットワークゾーンを選択します。

2. 右側で **Protocols** タブを選択し、**Add** ボタンをクリックします。**Protocol** ウィンドウが開きます。

3. リストからプロトコルを選択するか、**Other Protocol** チェックボックスを選択し、そのフィールドにプロトコルを入力します。

11.6.8. GUI を使用してソースポートを開く

特定ポートからファイアウォールを経由したトラフィックを許可するには、以下を行います。

1. firewall-config ツールを起動し、設定を変更するネットワークゾーンを選択します。

2. 右側の **Source Port** タブを選択し、**Add** ボタンをクリックします。**Source Port** ウィンドウが開きます。

3. 許可するポート番号またはポート範囲を入力します。リストから **tcp** または **udp** を選択します。

11.7. ファイアウォールゾーンでの作業

ゾーンは、着信トラフィックをより透過的な管理をする概念を表しています。ゾーンはネットワークインターフェースに接続されているか、ソースアドレスの範囲に割り当てられます。各ゾーンは個別にファイアウォールルールを管理しますが、これにより、複雑なファイアウォール設定を定義してトラフィックに割り当てることができます。

11.7.1. ゾーンの一覧

手順

1. システムで利用可能なゾーンを確認するには、以下のコマンドを実行します。

```
# firewall-cmd --get-zones
```

firewall-cmd --get-zones コマンドは、システムで利用可能な全てのゾーンを表示し、特定ゾーンの詳細は表示しません。

2. すべてのゾーンで詳細情報を表示する場合は、以下のコマンドを実行します。

```
# firewall-cmd --list-all-zones
```

3. 特定ゾーンに関する詳細情報を表示する場合は、以下のコマンドを実行します。

```
# firewall-cmd --zone=zone-name --list-all
```

11.7.2. 特定ゾーンに対する firewalld 設定の修正

「CLI を使用して事前定義されたサービスでトラフィックの制御」 および 「CLI を使用したポートの制御」 は、現在作業中のゾーンの範囲にサービスを追加するか、またはゾーンの範囲にあるポートを修正する方法を説明します。別のゾーンにルールを設定しないといけない場合もあります。

手順

1. 別のゾーンに指定するには、**--zone=zone-name** オプションを使用します。たとえば、**public** ゾーンで **SSH** サービスを許可するには、以下のコマンドを実行します。

```
# firewall-cmd --add-service=ssh --zone=public
```

11.7.3. デフォルトゾーンの変更

システム管理者は、設定ファイルのネットワークインターフェースにゾーンを割り当てます。特定のゾーンに割り当てられないインターフェースは、デフォルトゾーンに割り当てられます。**firewalld** サービスを再起動するたびに、**firewalld** は、デフォルトゾーンの設定を読み込み、それをアクティブにします。

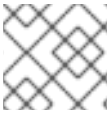
手順
デフォルトゾーンを設定するには、以下を行います。

1. 現在のデフォルトゾーンを表示します。

```
# firewall-cmd --get-default-zone
```

2. 新しいデフォルトゾーンを設定します。

```
# firewall-cmd --set-default-zone zone-name
```



注記

この手順では、**--permanent** オプションを使用しなくても、設定は永続化します。

11.7.4. ゾーンへのネットワークインターフェースの割り当て

複数のゾーンに複数のルールセットを定義して、使用されているインターフェースのゾーンを変更することで、迅速に設定を変更できます。複数のインターフェースを使用して、その各インターフェースに、トラフィックを通過する特定のゾーンを設定できます。

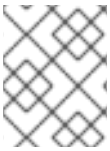
手順
特定インターフェースにゾーンを割り当てるには、以下を行います。

1. アクティブゾーン、およびそのゾーンに割り当てられているインターフェースを一覧表示します。

```
# firewall-cmd --get-active-zones
```

2. 別のゾーンにインターフェースを割り当てます。

```
# firewall-cmd --zone=zone-name --change-interface=<interface-name>
```



注記

再起動後も設定を持続させる **--permanent** オプションを使用する必要はありません。新しいデフォルトゾーンを設定すると、設定は永続化されます。

11.7.5. ネットワーク接続にデフォルトゾーンの割り当て

接続が NetworkManager により管理されると、使用するゾーンを認識する必要があります。すべてのネットワーク接続に、ゾーンを指定できます。これにより、ポータブルデバイスを使用したコンピューターの場所に従って、さまざまなファイアウォール設定の柔軟性を提供します。したがって、ゾーンお

よび設定には、会社または自宅など、さまざまな場所を指定できます。

手順

1. インターネット接続にデフォルトゾーンを設定するには、NetworkManager GUI を使用するか、**/etc/sysconfig/network-scripts/ifcfg-connection-name** ファイルを変更して、この接続にゾーンを割り当てる行を追加します。

```
ZONE=zone-name
```

11.7.6. 新しいゾーンの作成

カスタムゾーンを使用するには、新しいゾーンを作成したり、事前定義したゾーンなどを使用したりします。新しいゾーンには **--permanent** オプションが必要となり、このオプションがなければコマンドは動作しません。

手順
新しいゾーンを作成するには、以下を行います。

1. 新しいゾーンを作成します。

```
# firewall-cmd --new-zone=zone-name
```

2. 作成したゾーンが永続化設定に追加されたかどうかを確認します。

```
# firewall-cmd --get-zones
```

3. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

11.7.7. ゾーンの設定ファイル

また、**ゾーンの設定ファイル** を使用してゾーンを作成できます。このアプローチは、新しいゾーンを作成する必要がある場合に、別のゾーンの設定を変更して利用する場合に便利です。

firewalld ゾーン設定ファイルには、ゾーンに対する情報があります。これは、XML ファイルフォーマットで、ゾーンの説明、サービス、ポート、プロトコル、icmp-block、マスカレード、転送ポート、およびリッチ言語ルールです。ファイル名は **zone-name.xml** となります。**zone-name** の長さは 17 文字に制限されます。ゾーンの設定ファイルは **/usr/lib/firewalld/zones/** ディレクトリーおよび **/etc/firewalld/zones/** ディレクトリーです。

以下の例は、**TCP** プロトコルまたは **UDP** プロトコルの両方に、1 つのサービス (**SSH**) および 1 つのポート範囲を許可する設定を示します。

```
<?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>
```

そのゾーンの設定を変更するには、セクションを追加または削除して、ポート、転送ポート、サービスなどを追加します。

関連資料

- 詳細は、man ページの **firewalld.zone** を参照してください。

11.7.8. 着信トラフィックにデフォルトの動作を設定するゾーンターゲットの使用

すべてのゾーンに対して、特に指定されていない着信トラフィックを処理するデフォルト動作を設定できます。そのような動作は、ゾーンのターゲットを設定することで定義されます。オプションは、**default**、**ACCEPT**、**REJECT**、および **DROP** の 3 つになります。ターゲットを **ACCEPT** に設定すると、特定ルールで無効にした着信パケット以外のパケットをすべて許可します。**REJECT** または **DROP** にターゲットを設定すると、特定のルールで許可したパケット以外の着信パケットがすべて無効になります。パケットが拒否されると、拒否についてソースマシンに通知しますが、パケットが破棄される時に送られる情報はありません。

手順
ゾーンにターゲットを設定するには、以下を行います。

1. 特定ゾーンに対する情報を一覧表示して、デフォルトゾーンを確認します。

```
$ firewall-cmd --zone=zone-name --list-all
```

2. ゾーンに新しいターゲットを設定します。

```
# firewall-cmd --zone=zone-name --set-target=<default|ACCEPT|REJECT|DROP>
```

11.8. ゾーンを使用し、ソースに応じた着信トラフィックの管理

11.8.1. ゾーンを使用し、ソースに応じた着信トラフィックの管理

ゾーンを使用して、そのソースに基づいて着信トラフィックを管理するゾーンを使用できます。これにより、着信トラフィックを仕分けし、複数のゾーンに向け、トラフィックにより到達できるサービスを許可または拒否できます。

ソースをゾーンに追加する場合は、ゾーンがアクティブになり、そのソースからの着信トラフィックは、それを介して行われます。各ゾーンに異なる設定を指定できますが、それは指定したソースから順次トラフィックに適用されます。ネットワークインターフェースが 1 つしかない場合でも、複数のゾーンを使用できます。

11.8.2. ソースの追加

着信トラフィックを特定のソースにルートするには、そのゾーンにソースを追加します。ソースは、CIDR (Classless Inter-domain Routing) 表記法の IP アドレスまたは IP マスクになります。

- 現在のゾーンにソースを設定するには、以下のコマンドを実行します。

```
# firewall-cmd --add-source=<source>
```

- 特定ゾーンのソース IP アドレスを設定するには、以下のコマンドを実行します。

```
# firewall-cmd --zone=zone-name --add-source=<source>
```

以下の手順は、**信頼される** ゾーンで 192.168.2.15 からのすべての着信トラフィックを許可します。

手順

1. 利用可能なゾーンの一覧を表示します。

```
# firewall-cmd --get-zones
```

2. 永続化モードで、信頼ゾーンにソース IP を追加します。

```
# firewall-cmd --zone=trusted --add-source=192.168.2.15
```

3. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

11.8.3. ソースの削除

ゾーンからソースを削除すると、そのゾーンからのトラフィックを遮断します。

手順

1. 必要なゾーンに対して許可されているソースを一覧表示します。

```
# firewall-cmd --zone=zone-name --list-sources
```

2. ゾーンからソースを永続的に削除します。

```
# firewall-cmd --zone=zone-name --remove-source=<source>
```

3. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

11.8.4. ソースポートの追加

発信源となるポートに基づいてトラフィックの仕分けを有効にするには、**--add-source-port** オプションを使用してソースポートを指定します。**--add-source** オプションと組み合わせて、トラフィックを特定の IP アドレスまたは IP 範囲に制限できます。

手順

1. ソースポートを追加するには、以下のコマンドを実行します。

```
# firewall-cmd --zone=zone-name --add-source-port=<port-name>/<tcp|udp|sctp|dccp>
```

11.8.5. ソースポートの削除

ソースポートを削除して、送信元ポートに基づいてトラフィックの仕分けを無効にします。

手順

1. ソースポートを削除するには、以下のコマンドを実行します。

```
# firewall-cmd --zone=zone-name --remove-source-port=<port-name>/<tcp|udp|sctp|dccp>
```

11.8.6. ゾーンおよびソースを使用して特定ドメインのみに対してサービスの許可

特定のネットワークからのトラフィックを許可してマシンのサービスを使用するには、ゾーンおよびソースを使用します。以下の手順では、その他のトラフィックをブロックしつつ、192.168.1.0/24 からのトラフィックを許可し、HTTP サービスに到達できるようにします。

手順

1. 利用可能なゾーンの一覧を表示します。

```
# firewall-cmd --get-zones
block dmz drop external home internal public trusted work
```

2. ソースを信頼されるゾーンに追加して、ゾーンを経由してソースから発信するトラフィックに転送します。

```
# firewall-cmd --zone=trusted --add-source=192.168.1.0/24
```

3. 信頼ゾーンに **http** サービスを追加するには、以下を行います。

```
# firewall-cmd --zone=trusted -add-service=http
```

4. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

5. 信頼されるゾーンがアクティブで、サービスが許可されているのを確認します。

```
# firewall-cmd --zone=trusted --list-all
trusted (active)
target: ACCEPT
sources: 192.168.1.0/24
services: http
```

11.8.7. プロトコルに基づいてゾーンが許可したトラフィックの設定

プロトコルに基づいて、ゾーンが着信トラフィックを許可できます。指定したプロトコルを使用したすべてのトラフィックがゾーンにより許可されていますが、そこにさらにルールおよびフィルタリングを適用できます。

11.8.7.1. ゾーンへのプロトコルの追加

特定ゾーンへプロトコルを追加すると、このゾーンが許可するこのプロトコルを使用するすべてのトラフィックを許可します。

手順

1. プロトコルをゾーンに追加するには、以下のコマンドを実行します。

```
# firewall-cmd --zone=zone-name --add-protocol=port-name/tcp|udp|sctp|dccp|igmp
```

注記

マルチキャストトラフィックを受けるには、 **--add-protocol** オプションで **igmp** 値を使用します。

11.8.7.2. ゾーンからプロトコルの削除

特定ゾーンからプロトコルを削除するには、ゾーンにより、このプロトコルに基づいたすべてのトラフィックの許可を停止します。

手順

1. ゾーンからプロトコルを削除するには、以下のコマンドを削除します。

```
# firewall-cmd --zone=zone-name --remove-protocol=port-name/tcp|udp|sctp|dccp|igmp
```

11.9. IP アドレスのマスカレードの設定

以下の手順では、システムで IP マスカレードを有効にする方法を説明します。IP マスカレードは、インターネットにアクセスする際にゲートウェイの向こう側にある個々のマシンを隠します。

手順

1. **external** ゾーンなどで IP マスカレーディングが有効かどうかを確認するには、 **root** で以下のコマンドを実行します。

```
# firewall-cmd --zone=external --query-masquerade
```

このコマンドでは、有効な場合は終了ステータスが **0** で **yes** が出力され、無効の場合は終了ステータスが **1** で **no** が出力されます。**zone** を省略すると、デフォルトのゾーンが使用されます。

2. IP マスカレードを有効にするには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --zone=external --add-masquerade
```

3. この設定を永続的にするには、**--permanent** オプションを追加してコマンドを繰り返します。

IP マスカレードを無効にするには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --zone=external --remove-masquerade --permanent
```

11.10. ICMP 要求の管理

Internet Control Message Protocol (**ICMP**) は、接続問題 (要求されているサービスが利用できないなど) を示すエラーメッセージと運用情報を送信するために、さまざまなネットワークデバイスにより使用されている、サポート対象のプロトコルです。**ICMP** は、システム間のデータ交換するには使用されていないため、TCP、UDP などの転送プロトコルとは異なります。

ただし、**ICMP** メッセージ (特に **echo-request** および **echo-reply**) を利用して、ネットワークに関する情報を明らかにし、その情報をさまざまな不正行為に悪用することが可能です。したがって、**firewalld** は、ネットワーク情報を保護するため、**ICMP** リクエストをブロックできます。

11.10.1. ICMP 要求の一覧表示およびブロック

ICMP リクエストの一覧表示

ICMP リクエストについては、**/usr/lib/firewalld/icmptypes/** ディレクトリー内の各 XML ファイルで説明されています。このファイルを読み、リクエストの説明を確認します。**firewall-cmd** コマンドは、**ICMP** リクエストの操作を制御します。

- 利用可能な **ICMP** タイプを一覧表示するには、以下を行います。

```
# firewall-cmd --get-icmptypes
```

- **ICMP** リクエストは、IPv4、IPv6、またはその両方のプロトコルで使用できます。**ICMP** リクエストが使用されてるプロトコルを表示するには、以下のコマンドを実行します。

```
# firewall-cmd --info-icmptype=<icmptype>
```

- **ICMP** リクエストのステータスは、リクエストが現在ブロックされている場合は **yes**、ブロックされていない場合は **no** となります。**ICMP** リクエストが現在ブロックされているかどうかを確認するには、以下のコマンドを実行します。

```
# firewall-cmd --query-icmp-block=<icmptype>
```

ICMP リクエストのブロックまたはブロック解除

サーバーが **ICMP** リクエストをブロックした場合は、通常の情報が提供されません。ただし、情報が全く提供されないというわけではありません。クライアントは、特定の **ICMP** リクエストがブロックされている (拒否されている) 情報を受け取ります。**ICMP** リクエストは、特に IPv6 トラフィックを使用すると、接続問題が発生することがあるため、注意深く検討する必要があります。

- **ICMP** リクエストが現在ブロックされているかどうかを確認するには、以下を行います。

```
# firewall-cmd --query-icmp-block=<icmptype>
```

- **ICMP** リクエストをブロックするには、以下を行います。

```
# firewall-cmd --add-icmp-block=<icmptype>
```

- **ICMP** リクエストのブロックを削除するには、以下を行います。

```
# firewall-cmd --remove-icmp-block=<icmptype>
```

情報を提供せずに ICMP 要求のブロック

通常、**ICMP** リクエストをブロックすると、ブロックしていることをクライアントは認識します。したがって、ライブの IP アドレスを傍受している潜在的な攻撃者は、IP アドレスがオンラインであることを見ることができます。この情報を完全に非表示にするには、**ICMP** リクエストをすべて破棄する必要があります。

- すべての **ICMP** リクエストをブロックして破棄するには、以下のコマンドを実行します。

1. ゾーンのターゲットを **DROP** に設定します。

```
# firewall-cmd --set-target=DROP
```

2. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

これで、明示的に許可されるトラフィックを除き、**ICMP** リクエストを含むすべてのトラフィックが破棄されます。

- 特定の **ICMP** 要求をブロックして破棄し、その他の要求は許可するには、以下を行います。

1. ゾーンのターゲットを **DROP** に設定します。

```
# firewall-cmd --set-target=DROP
```

2. すべての **ICMP** リクエストを一度にブロックする ICMP ブロックの反転を追加します。

```
# firewall-cmd --add-icmp-block-inversion
```

3. 許可する **ICMP** リクエストに ICMP ブロックを追加するには、以下を行います。

```
# firewall-cmd --add-icmp-block=<icmptype>
```

4. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

ブロックの反転 は、**ICMP** リクエストブロックの設定を反転するため、すでにブロックしていないリクエストをすべてブロックします。ブロックされているものはブロックされません。したがって、リクエストのブロックを解除する必要がある場合は、ブロックコマンドを使用してください。

- ブロックの反転を完全許可の設定に戻すには、以下を行います。

1. ゾーンのターゲットを **default** または **ACCEPT** に戻すには、以下のコマンドを設定します。

```
# firewall-cmd --set-target=default
```

2. **ICMP** リクエストに追加したすべてのブロックを削除します。

```
# firewall-cmd --remove-icmp-block=<icmptype>
```

3. **ICMP** ブロックの反転を削除します。

```
# firewall-cmd --remove-icmp-block-inversion
```

4. 新しい設定を永続化します。

```
# firewall-cmd --runtime-to-permanent
```

11.10.2. GUI を使用した ICMP フィルターの設定

- **ICMP** フィルターを有効または無効にするには、**firewall-config** ツールを起動して、フィルターをかけるメッセージのネットワークゾーンを選択します。**ICMP フィルター** タブを選択し、フィルターをかける **ICMP** メッセージの各タイプのチェックボックスを選択します。フィ

ルターを無効にするには、チェックボックスの選択を外します。これは方向ごとに設定され、デフォルトではすべてが許可されます。

- **ICMP** タイプを編集するには、**firewall-config** ツールを起動してから **設定** ラベルのあるメニューで **永続** モードを選択します。**サービス** ウィンドウの下部に新たなアイコンが表示されます。以下のダイアログで「**はい**」を選択し、マスカレーディングを有効にし、動作している別のマシンに転送します。

- **ICMP フィルター** の反転を有効にするには、右側の **フィルターの反転** チェックボックスをクリックします。マークがついた **ICMP** タイプだけが許可され、その他はすべて拒否されます。DROP ターゲットを使用するゾーンでは破棄されます。

11.11. FIREWALLD を使用した IP セットの設定および制御

firewalld でサポートする IP セットタイプの一覧を表示するには、root で以下のコマンドを実行します。

```
~]# 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
```

11.11.1. CLI を使用した IP セットオプションの設定

IP セットは、**firewalld** ゾーンでソースとして使用でき、リッチルールでソースとして使用できます。Red Hat Enterprise Linux で推奨される方法は、ダイレクトルールで **firewalld** を使用して作成した IP セットを使用する方法です。

- 永続的な環境で **firewalld** に認識されている IP セットを一覧表示するには、次のコマンドを **root** で実行します。

  ```
  # firewall-cmd --permanent --get-ipsets
  ```

- 新しい IP セットを追加するには、永続化環境を使用した以下のコマンドを **root** で実行します。

  ```
  # firewall-cmd --permanent --new-ipset=test --type=hash:net
  success
  ```

  上記のコマンドは、**IPv4** の名前 **test** とタイプ **hash:net** で新しい IP セットを作成します。**IPv6** と使用する IP セットを作成するには、**--option=family=inet6** オプションを追加します。ランタイム環境で新しい設定を有効にするには、**firewalld** を再ロードします。

- **root** で以下のコマンドを実行し、新しい IP セットを一覧表示します。

  ```
  # firewall-cmd --permanent --get-ipsets
  test
  ```

- IP セットの詳細は、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --permanent --info-ipset=test
  test
  type: hash:net
  ```

```
options:
entries:
```

この時点では IP セットにエントリーがありません。

- **test** IP セットにエントリーを追加するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --permanent --ipset=test --add-entry=192.168.0.1
success
```

上記のコマンドは、IP アドレス **192.168.0.1** を IP セットに追加します。

- IP セットの現在のエントリーを一覧表示するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
```

- IP アドレスの一覧を含むファイルを生成します。以下のコマンドを実行します。

```
# cat > iplist.txt <<EOL
192.168.0.2
192.168.0.3
192.168.1.0/24
192.168.2.254
EOL
```

IP セットの IP アドレスの一覧が含まれるファイルには、行ごとにエントリーが含まれる必要があります。ハッシュ、セミコロン、また空の行から始まる行は無視されます。

- **iplist.txt** ファイルからアドレスを追加するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --permanent --ipset=test --add-entries-from-file=iplist.txt
success
```

- IP セットの拡張エントリーの一覧を表示するには、**root** で以下のコマンドを実行します。

```
# 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
```

- IP セットからアドレスを削除し、更新したエントリー一覧を確認するには、以下のコマンドを **root** で実行します。

```
# firewall-cmd --permanent --ipset=test --remove-entries-from-file=iplist.txt
success
# firewall-cmd --permanent --ipset=test --get-entries
192.168.0.1
```

- IP セットをゾーンへのソースとして追加し、ゾーンを使用して、IP セットに記載されるアドレスから受信するすべてのトラフィックを処理します。たとえば、**test** IP セットをソースとして **drop** ゾーンに追加し、**test** の IP セット一覧に表示されるすべてエントリーから発信されるパ

ケットをすべて破棄するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --permanent --zone=drop --add-source=ipset:test
success
```

ソースの **ipset:** 接頭辞は、ソースが IP セットで、IP アドレスまたはアドレス範囲ではない **firewalld** を示しています。

IP セットの作成および削除は、永続環境に限定されており、その他の IP セットオプションは、**--permanent** オプションを使用しないランタイム環境で使用できます。

警告

Red Hat は、**firewalld** を介して管理していない IP セットを使用することは推奨しません。このような IP セットを使用すると、そのセットを参照する永続的なダイレクトルールが必要で、IP セットを作成するカスタムサービスを追加する必要があります。このサービスは、firewalld を起動する前に起動する必要があります。先に起動しておかないと、**firewalld** が、このセットを使用してダイレクトルールを追加できません。**/etc/firewalld/direct.xml** ファイルを使用して、永続的なダイレクトルールを追加できます。

11.12. ファイアウォールロックダウンの設定

ローカルのアプリケーションやサービスは、**root** で実行していれば、ファイアウォール設定を変更できます (たとえば **libvirt**)。管理者は、この機能を使用してファイアウォール設定をロックし、どのアプリケーションもファイアウォール変更を要求できなくするか、ロックダウンのホワイトリストに追加されたアプリケーションのみがファイアウォール変更を要求できるようにすることが可能になります。ロックダウン設定はデフォルトで無効になっています。これを有効にすると、ローカルのアプリケーションやサービスによるファイアウォールの望ましくない設定変更を確実に防ぐことができます。

11.12.1. CLI を使用したロックダウンの設定

- ロックダウンが有効になっているかを確認するには、**root** で以下のコマンドを使用します。

  ```
  # firewall-cmd --query-lockdown
  ```

  ロックダウンが有効な場合は終了ステータスが **0** で **yes** が出力され、無効の場合は終了ステータスが **1** で **no** が出力されます。

- ロックダウンを有効にするには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --lockdown-on
  ```

- ロックダウンを無効にするには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --lockdown-off
  ```

11.12.2. CLI を使用したロックダウンホワイトリストオプションの設定

ロックダウンのホワイトリストには、コマンド、セキュリティーのコンテキスト、ユーザー、および ユーザー ID を追加できます。ホワイトリストのコマンドエントリーがアスタリスク「*」で終了している場合は、そのコマンドで始まるすべてのコマンドラインが一致することになります。「*」がない場合は、コマンドと引数が完全に一致する必要があります。

- ここでのコンテキストは、実行中のアプリケーションやサービスのセキュリティー (SELinux) コンテキストです。実行中のアプリケーションのコンテキストを確認するには、以下のコマンドを実行します。

  ```
  $ ps -e --context
  ```

  このコマンドで、実行中のアプリケーションがすべて返されます。**grep** ツールを使用して、出力から目的のアプリケーションを以下のようにパイプ処理します。

  ```
  $ ps -e --context | grep example_program
  ```

- ホワイトリストにあるコマンドラインを一覧表示するには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --list-lockdown-whitelist-commands
  ```

- ホワイトリストにコマンド **command** を追加するには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --add-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
  ```

- ホワイトリストからコマンド **command** を削除するには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --remove-lockdown-whitelist-command='/usr/bin/python3 -Es
  /usr/bin/command'
  ```

- ホワイトリストにコマンド **command** があるかどうかを確認するには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --query-lockdown-whitelist-command='/usr/bin/python3 -Es /usr/bin/command'
  ```

  True の場合は終了ステータスが **0** で **yes** が出力され、False の場合は終了ステータスが **1** で **no** が出力されます。

- ホワイトリストにあるセキュリティーコンテキストを一覧表示するには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --list-lockdown-whitelist-contexts
  ```

- ホワイトリストにコンテキスト **context** を追加するには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --add-lockdown-whitelist-context=context
  ```

- ホワイトリストからコンテキスト **context** を削除するには、**root** で以下のコマンドを実行します。

  ```
  # firewall-cmd --remove-lockdown-whitelist-context=context
  ```

- ホワイトリストにコンテキスト **context** があるかどうかを確認するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --query-lockdown-whitelist-context=context
```

コマンドがある場合は終了ステータスが **0** で **yes** が出力され、ない場合は終了ステータスが **1** で **no** が出力されます。

- ホワイトリストにあるユーザー ID を一覧表示するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --list-lockdown-whitelist-uids
```

- ホワイトリストにユーザー ID (**uid**) を追加するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --add-lockdown-whitelist-uid=uid
```

- ホワイトリストからユーザー ID (**uid**) を削除するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --remove-lockdown-whitelist-uid=uid
```

- ホワイトリストにユーザー ID (**uid**) があるかどうかを確認するには、以下のコマンドを実行します。

```
$ firewall-cmd --query-lockdown-whitelist-uid=uid
```

コマンドがある場合は終了ステータスが **0** で **yes** が出力され、ない場合は終了ステータスが **1** で **no** が出力されます。

- ホワイトリストにあるユーザー名を一覧表示するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --list-lockdown-whitelist-users
```

- ホワイトリストにユーザー名 (**user**) を追加するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --add-lockdown-whitelist-user=user
```

- ホワイトリストからユーザー名 (**user**) を削除するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --remove-lockdown-whitelist-user=user
```

- ホワイトリストにユーザー名 (**user**) があるかどうかを確認するには、以下のコマンドを実行します。

```
$ firewall-cmd --query-lockdown-whitelist-user=user
```

コマンドがある場合は終了ステータスが **0** で **yes** が出力され、ない場合は終了ステータスが **1** で **no** が出力されます。

11.12.3. 設定ファイルを使用したロックダウンホワイトリストオプションの設定

デフォルトのホワイトリスト設定ファイルには、**NetworkManager** コンテキストと、**libvirt** のデフォルトコンテキストが含まれます。リストには、ユーザー ID (0) もあります。

-

```
<?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>
```

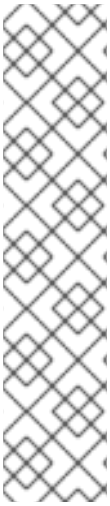
以下のホワイトリスト設定ファイルの例では、**firewall-cmd** ユーティリティーのコマンドと、ユーザー ID が **815** である **user** のコマンドをすべて有効にしています。

```
<?xml version="1.0" encoding="utf-8"?>
 <whitelist>
   <command name="/usr/bin/python3 -Es /bin/firewall-cmd*"/>
   <selinux context="system_u:system_r:NetworkManager_t:s0"/>
   <user id="815"/>
   <user name="user"/>
 </whitelist>
```

この例では、**user id** と **user name** の両方が使用されていますが、実際にはどちらか一方のオプションだけが必要です。Python はインタープリターとしてコマンドラインに追加されています。または、以下のような明確なコマンドも使用できます。

```
/usr/bin/python3 /bin/firewall-cmd --lockdown-on
```

この例では、**--lockdown-on** コマンドだけが許可されます。



注記

Red Hat Enterprise Linux では、すべてのユーティリティーが **/usr/bin/** ディレクトリーに格納されており、**/bin/** ディレクトリーは **/usr/bin/** ディレクトリーのシンボリックリンクとなります。つまり、**root** で **firewall-cmd** のパスを実行すると **/bin/firewall-cmd** に対して解決しますが、**/usr/bin/firewall-cmd** が使用できるようになっています。新たなスクリプトはすべて新しい格納場所を使用する必要がありますが、**root** で実行するスクリプトが **/bin/firewall-cmd** のパスを使用するようになっているのであれば、これまでは **root** 以外のユーザーにのみ使用されていた **/usr/bin/firewall-cmd** パスに加え、このコマンドのパスもホワイトリストに追加する必要があります。

コマンドの名前属性の最後にある「*」は、その名前で始まるすべてのコマンドが一致することを意味します。「*」がなければ、コマンドと引数が完全に一致する必要があります。

11.13. 拒否されたパケットのログ

firewalld で **LogDenied** オプションを使用して、拒否したパケットに簡易ロギングメカニズムを追加できます。対象となるのは、拒否または破棄されるパケットになります。ログ設定を変更するには、**/etc/firewalld/firewalld.conf** ファイルを変更するか、コマンドラインまたは GUI 設定ツールを使用します。

LogDenied を有効にすると、デフォルトルールの INPUT、FORWARD、および OUTPUT チェインの reject ルールおよび drop ルールと、ゾーンの最後の reject ルールおよび drop ルールの直前に、ロギングルールが追加されます。ここに設定できる値は、**all**、**unicast**、**broadcast**、**multicast**、および **off** です。デフォルト設定は **off** です。**unicast**、**broadcast**、**multicast** の設定では、リンク層のパケットタイプを一致させるのに **pkttype** 一致を使用します。**all** を使用すると、すべてのパケットがログに記録されます。

firewall-cmd で実際の **LogDenied** 設定を一覧表示するには、**root** で以下のコマンドを使用します。

```
# firewall-cmd --get-log-denied
off
```

LogDenied 設定を変更するには、**root** で以下のコマンドを実行します。

```
# firewall-cmd --set-log-denied=all
success
```

firewalld GUI 設定ツールを使用して **LogDenied** 設定を変更するには、**firewall-config** を起動し、**Options** メニューをクリックし、**Change Log Denied** を選択します。**LogDenied** ウィンドウが表示されます。メニューから新しい **LogDenied** 設定を選択し、OK をクリックします。

第12章 NFTABLES の使用

記載	アセンブリーの一覧
ユーザーストーリー	ユーザーストーリー
Jira	JIRA のリンク
BZ	BUGZILLA のリンク
SME	SME の名前
SME Ack	はい/いいえ
Peer Ack	はい/いいえ

管理者は、**nftables** フレームワークを使用すると、Linux カーネルのファイアウォールで使用されるパケットフィルタリングルールを設定できます。

第13章 NFTABLES の概要

nftables フレームワークは、パケットの分離機能を提供し、**iptables** ツール、**ip6tables** ツール、**arptables** ツール、および **ebtables** ツールの後継となります。利便性、機能、パフォーマンスにおいて、以前のパケットフィルタリングツールに多くの改良が追加されました。以下に例を示します。

- 線形処理の代わりにルックアップテーブル
- **IPv4** プロトコルおよび **IPv6** プロトコルの両方に対する単一のフレームワーク
- 完全ルールセットのフェッチ、更新、および保存を行わず、すべてアトミックに適用されるルール
- ルールセットにおけるデバッグおよびトレースのサポート (**nftrace**) およびトレースイベントの監視 (**nft** ツール)
- より一貫性のあるコンパクトな構文、プロトコル固有の拡張なし
- サードパーティーのアプリケーション用 Netlink API

iptables と同様、**nftables** は、チェーンを保存するテーブルを使用します。このチェーンは、アクションを実行する個々のルールが含まれます。**nft** ツールは、以前のパケットフィルタリングフレームワークのツールをすべて置き換えます。**libnftnl** ライブラリーは、**libmnl** ライブラリーの **nftables** Netlink API で、低レベルの対話のために使用できます。

RHEL 8 では、**nftables** は、デフォルトの **firewalld** バックエンドとして動作します。**nftables** バックエンドは、ファイアウォール設定で以前の **iptables** バックエンドと後方互換性がありますが、**/etc/firewalld/firewalld.conf** ファイルの **FirewallBackend** オプションに **iptables** を設定して、バックエンドを **iptables** に戻すことができます。

nftables ルールセットに対するモジュールの効果は、**nft** list ruleset コマンドを使用して確認できます。このツールは、テーブル、チェーン、およびルールを **nftables** ルールセットに追加するため、**nft flush ruleset** などの **nftables** ルールセット操作は、以前は別々のコマンドを使用してインストールしたルールセットに影響を及ぼす可能性があることに注意してください。

どの種類のツールが存在するかをすばやく識別するために、バックエンド名を追加するようにバージョン情報が更新されました。RHEL 8 では、nftables ベースの **iptables** ツールで、次のバージョン文字列が出力されます。

```
$ iptables --version
iptables v1.8.0 (nf_tables)
```

一方、従来の iptables ツールが存在する場合は、次のバージョン情報が出力されます。

```
$ iptables --version
iptables v1.8.0 (legacy)
```

関連資料

- man ページの **nft(8)** は、**nft** コマンドラインツールで nftables を使用して、パケットのフィルタリングを設定および検査するための包括的な参考資料を提供します。

13.1. 関連情報

- 「What comes after iptables? Its successor, of course: nftables」 のブログ投稿では、**nftables** が **iptables** の代替になった理由が説明されています。
- 「Firewalld: The Future is nftables」 の記事では、**firewalld** でデフォルトのバックエンドとなる **nftables** に関する追加情報が提供されます。