



Red Hat Enterprise Linux 9

Configuration et gestion des réseaux

Gestion des interfaces réseau et des fonctions réseau avancées

Red Hat Enterprise Linux 9 Configuration et gestion des réseaux

Gestion des interfaces réseau et des fonctions réseau avancées

Notice légale

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Résumé

En utilisant les capacités de mise en réseau de Red Hat Enterprise Linux (RHEL), vous pouvez configurer votre hôte pour qu'il réponde aux exigences de votre organisation en matière de réseau et de sécurité. Par exemple : Vous pouvez configurer des liens, des VLAN, des ponts, des tunnels et d'autres types de réseaux pour connecter l'hôte au réseau. IPSec et WireGuard fournissent des VPN sécurisés entre les hôtes et les réseaux. RHEL prend également en charge des fonctions réseau avancées, telles que le routage basé sur des règles et le protocole TCP MultiPath (MPTCP).

Table des matières

RENDRE L'OPEN SOURCE PLUS INCLUSIF	10
FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT	11
CHAPITRE 1. DÉNOMINATION COHÉRENTE DES PÉRIPHÉRIQUES D'INTERFACE RÉSEAU	12
1.1. HIÉRARCHIE DE DÉNOMINATION DES PÉRIPHÉRIQUES D'INTERFACE RÉSEAU	12
1.2. COMMENT FONCTIONNE LE RENOMMAGE DES PÉRIPHÉRIQUES RÉSEAU	13
1.3. EXPLICATION DES NOMS PRÉVISIBLES DES PÉRIPHÉRIQUES D'INTERFACE RÉSEAU SUR LA PLATE-FORME X86_64	14
1.4. EXPLICATION DES NOMS PRÉVISIBLES DES PÉRIPHÉRIQUES D'INTERFACE RÉSEAU SUR LA PLATE-FORME SYSTEM Z	15
1.5. PERSONNALISATION DU PRÉFIXE DES INTERFACES ETHERNET LORS DE L'INSTALLATION	15
1.6. ATTRIBUTION DE NOMS D'INTERFACE RÉSEAU DÉFINIS PAR L'UTILISATEUR À L'AIDE DES RÈGLES UDEV	16
1.7. ATTRIBUTION DE NOMS D'INTERFACE RÉSEAU DÉFINIS PAR L'UTILISATEUR À L'AIDE DES FICHIERS DE LIAISON DE SYSTEMD	17
1.8. ATTRIBUTION DE NOMS SUPPLÉMENTAIRES À L'INTERFACE RÉSEAU À L'AIDE DES FICHIERS DE LIAISON DE SYSTEMD	18
CHAPITRE 2. CONFIGURATION D'UNE CONNEXION ETHERNET	20
2.1. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE À L'AIDE DE NMCLI	20
2.2. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE À L'AIDE DE L'ÉDITEUR INTERACTIF NMCLI	22
2.3. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE À L'AIDE DE NMTUI	24
2.4. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE À L'AIDE DE NMSTATECTL	27
2.5. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE EN UTILISANT LE RÔLE DE SYSTÈME RHEL AVEC UN NOM D'INTERFACE	29
2.6. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE EN UTILISANT LE RÔLE DE SYSTÈME RHEL AVEC UN CHEMIN D'ACCÈS AUX PÉRIPHÉRIQUES	30
2.7. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE À L'AIDE DE NMCLI	32
2.8. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE À L'AIDE DE L'ÉDITEUR INTERACTIF NMCLI	33
2.9. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE À L'AIDE DE NMTUI	35
2.10. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE À L'AIDE DE NMSTATECTL	37
2.11. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE EN UTILISANT LE RÔLE DE SYSTÈME RHEL AVEC UN NOM D'INTERFACE	38
2.12. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE EN UTILISANT LE RÔLE DE SYSTÈME RHEL AVEC UN CHEMIN D'ACCÈS DE PÉRIPHÉRIQUE	39
2.13. CONFIGURATION D'UNE CONNEXION ETHERNET À L'AIDE DU CENTRE DE CONTRÔLE	41
2.14. CONFIGURATION D'UNE CONNEXION ETHERNET À L'AIDE DE NM-CONNECTION-EDITOR	43
2.15. CHANGEMENT DU CLIENT DHCP DU NETWORKMANAGER	46
2.16. CONFIGURATION DU COMPORTEMENT DHCP D'UNE CONNEXION NETWORKMANAGER	47
2.17. CONFIGURATION DE PLUSIEURS INTERFACES ETHERNET À L'AIDE D'UN SEUL PROFIL DE CONNEXION PAR NOM D'INTERFACE	48
2.18. CONFIGURATION D'UN PROFIL DE CONNEXION UNIQUE POUR PLUSIEURS INTERFACES ETHERNET À L'AIDE D'ID PCI	49
CHAPITRE 3. CONFIGURATION DE LA LIAISON RÉSEAU	51
3.1. COMPRENDRE LA LIAISON RÉSEAU	51

3.2. COMPRENDRE LE COMPORTEMENT PAR DÉFAUT DES INTERFACES DES CONTRÔLEURS ET DES PORTS	51
3.3. CONFIGURATION DU COMMUTATEUR EN AMONT EN FONCTION DES MODES DE LIAISON	52
3.4. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE NMCLI	52
3.5. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE LA CONSOLE WEB RHEL	55
3.6. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE NMTUI	59
3.7. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE NM-CONNECTION-EDITOR	62
3.8. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE NMSTATECTL	64
3.9. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DU RÔLE DE SYSTÈME RHEL RÉSEAU	66
3.10. CRÉATION D'UNE LIAISON RÉSEAU PERMETTANT DE PASSER D'UNE CONNEXION ETHERNET À UNE CONNEXION SANS FIL SANS INTERROMPRE LE VPN	68
3.11. LES DIFFÉRENTS MODES DE CONNEXION AU RÉSEAU	70
3.12. LE PARAMÈTRE DE LIAISON XMIT_HASH_POLICY	72
CHAPITRE 4. CONFIGURATION DU TEAMING RÉSEAU	76
4.1. MIGRATION D'UNE CONFIGURATION D'ÉQUIPE DE RÉSEAU VERS UN LIEN DE RÉSEAU	76
4.2. COMPRENDRE LE TRAVAIL EN ÉQUIPE EN RÉSEAU	79
4.3. COMPRENDRE LE COMPORTEMENT PAR DÉFAUT DES INTERFACES DES CONTRÔLEURS ET DES PORTS	79
4.4. COMPRENDRE LE SERVICE TEAMD, LES RUNNERS ET LES LINK-WATCHERS	80
4.5. CONFIGURATION D'UNE ÉQUIPE RÉSEAU À L'AIDE DE NMCLI	81
4.6. CONFIGURATION D'UNE ÉQUIPE RÉSEAU À L'AIDE DE LA CONSOLE WEB RHEL	84
4.7. CONFIGURATION D'UNE ÉQUIPE RÉSEAU À L'AIDE DE NM-CONNECTION-EDITOR	87
CHAPITRE 5. CONFIGURATION DU MARQUAGE VLAN	91
5.1. CONFIGURATION DU MARQUAGE DES VLAN À L'AIDE DE NMCLI	91
5.2. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DE LA CONSOLE WEB RHEL	93
5.3. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DE NMTUI	95
5.4. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DE NM-CONNECTION-EDITOR	99
5.5. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DE NMSTATECTL	101
5.6. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DU RÔLE DE SYSTÈME RHEL DU RÉSEAU	102
5.7. RESSOURCES SUPPLÉMENTAIRES	104
CHAPITRE 6. CONFIGURATION D'UN PONT RÉSEAU	105
6.1. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE NMCLI	105
6.2. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE LA CONSOLE WEB RHEL	108
6.3. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE NMTUI	110
6.4. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE NM-CONNECTION-EDITOR	114
6.5. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE NMSTATECTL	116
6.6. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DU RÔLE DE SYSTÈME RHEL DE RÉSEAU	118
CHAPITRE 7. CONFIGURATION D'UNE CONNEXION VPN	121
7.1. CONFIGURATION D'UNE CONNEXION VPN AVEC LE CENTRE DE CONTRÔLE	121
7.2. CONFIGURATION D'UNE CONNEXION VPN À L'AIDE DE NM-CONNECTION-EDITOR	125
7.3. CONFIGURATION DE LA DÉTECTION AUTOMATIQUE ET DE L'UTILISATION DE LA DÉCHARGE MATÉRIELLE ESP POUR ACCÉLÉRER UNE CONNEXION IPSEC	128
7.4. CONFIGURATION DE LA DÉCHARGE MATÉRIELLE ESP SUR UNE LIAISON POUR ACCÉLÉRER UNE CONNEXION IPSEC	129
CHAPITRE 8. MISE EN PLACE D'UN VPN WIREGUARD	131
8.1. PROTOCOLES ET PRIMITIVES UTILISÉS PAR WIREGUARD	131
8.2. COMMENT WIREGUARD UTILISE LES ADRESSES IP DES TUNNELS, LES CLÉS PUBLIQUES ET LES POINTS DE TERMINAISON DISTANTS	132
8.3. UTILISATION D'UN CLIENT WIREGUARD DERRIÈRE UN NAT ET DES PARE-FEUX	132
8.4. CRÉATION DE CLÉS PRIVÉES ET PUBLIQUES À UTILISER DANS LES CONNEXIONS WIREGUARD	133

8.5. CONFIGURATION D'UN SERVEUR WIREGUARD À L'AIDE DE NMCLI	134
8.6. CONFIGURATION D'UN SERVEUR WIREGUARD À L'AIDE DE NMTUI	136
8.7. CONFIGURATION D'UN SERVEUR WIREGUARD À L'AIDE DE NM-CONNECTION-EDITOR	139
8.8. CONFIGURATION D'UN SERVEUR WIREGUARD À L'AIDE DU SERVICE WG-QUICK	141
8.9. CONFIGURER FIREWALLD SUR UN SERVEUR WIREGUARD EN UTILISANT LA LIGNE DE COMMANDE	143
8.10. CONFIGURATION DE FIREWALLD SUR UN SERVEUR WIREGUARD À L'AIDE DE L'INTERFACE GRAPHIQUE	144
8.11. CONFIGURATION D'UN CLIENT WIREGUARD À L'AIDE DE NMCLI	145
8.12. CONFIGURATION D'UN CLIENT WIREGUARD À L'AIDE DE NMTUI	148
8.13. CONFIGURATION D'UN CLIENT WIREGUARD À L'AIDE DE NM-CONNECTION-EDITOR	151
8.14. CONFIGURER UN CLIENT WIREGUARD EN UTILISANT LE SERVICE WG-QUICK	154
CHAPITRE 9. CONFIGURATION DES TUNNELS IP	157
9.1. CONFIGURATION D'UN TUNNEL IPIP À L'AIDE DE NMCLI POUR ENCAPSULER LE TRAFIC IPV4 DANS DES PAQUETS IPV4	157
9.2. CONFIGURATION D'UN TUNNEL GRE À L'AIDE DE NMCLI POUR ENCAPSULER LE TRAFIC DE COUCHE 3 DANS DES PAQUETS IPV4	160
9.3. CONFIGURATION D'UN TUNNEL GREAP POUR TRANSFÉRER DES TRAMES ETHERNET SUR IPV4	162
9.4. RESSOURCES SUPPLÉMENTAIRES	165
CHAPITRE 10. UTILISATION D'UN VXLAN POUR CRÉER UN DOMAINE VIRTUEL DE COUCHE 2 POUR LES MACHINES VIRTUELLES	166
10.1. AVANTAGES DES VXLAN	166
10.2. CONFIGURATION DE L'INTERFACE ETHERNET SUR LES HÔTES	167
10.3. CRÉATION D'UN PONT RÉSEAU AVEC UN VXLAN ATTACHÉ	168
10.4. CRÉER UN RÉSEAU VIRTUEL DANS LIBVIRT AVEC UN PONT EXISTANT	169
10.5. CONFIGURATION DES MACHINES VIRTUELLES POUR L'UTILISATION DE VXLAN	170
CHAPITRE 11. GESTION DES CONNEXIONS WIFI	172
11.1. TYPES DE SÉCURITÉ WIFI PRIS EN CHARGE	172
11.2. CONNEXION À UN RÉSEAU WIFI À L'AIDE DE NMCLI	173
11.3. SE CONNECTER À UN RÉSEAU WIFI EN UTILISANT LE MENU SYSTÈME GNOME	174
11.4. SE CONNECTER À UN RÉSEAU WIFI EN UTILISANT L'APPLICATION DE PARAMÉTRAGE GNOME	175
11.5. CONFIGURATION D'UNE CONNEXION WIFI À L'AIDE DE NMTUI	177
11.6. CONFIGURER UNE CONNEXION WIFI À L'AIDE DE NM-CONNECTION-EDITOR	179
11.7. CONFIGURER UNE CONNEXION WIFI AVEC L'AUTHENTIFICATION RÉSEAU 802.1X EN UTILISANT LE RÔLE RÉSEAU RHEL SYSTEM ROLE	180
11.8. CONFIGURER UNE CONNEXION WIFI AVEC L'AUTHENTIFICATION RÉSEAU 802.1X DANS UN PROFIL EXISTANT EN UTILISANT NMCLI	182
11.9. CONFIGURATION MANUELLE DU DOMAINE DE RÉGULATION SANS FIL	183
CHAPITRE 12. CONFIGURER RHEL COMME POINT D'ACCÈS WIFI	185
12.1. IDENTIFIER SI UN APPAREIL WIFI SUPPORTE LE MODE POINT D'ACCÈS	185
12.2. CONFIGURATION DE RHEL EN TANT QUE POINT D'ACCÈS PERSONNEL WPA2 OU WPA3	185
CHAPITRE 13. MODIFIER UN NOM D'HÔTE	188
13.1. MODIFIER UN NOM D'HÔTE À L'AIDE DE NMCLI	188
13.2. MODIFIER UN NOM D'HÔTE À L'AIDE DE HOSTNAMECTL	188
CHAPITRE 14. MISE EN MIROIR DES PORTS	190
14.1. MISE EN MIROIR D'UNE INTERFACE RÉSEAU À L'AIDE DE NMCLI	190
CHAPITRE 15. CONFIGURER NETWORKMANAGER POUR QU'IL IGNORE CERTAINS PÉRIPHÉRIQUES	192
15.1. CONFIGURATION DE L'INTERFACE DE BOUCLAGE À L'AIDE DE NMCLI	192
15.2. CONFIGURATION PERMANENTE D'UN PÉRIPHÉRIQUE COMME NON GÉRÉ DANS NETWORKMANAGER	

	193
15.3. CONFIGURER TEMPORAIREMENT UN PÉRIPHÉRIQUE COMME NON GÉRÉ DANS NETWORKMANAGER	194
CHAPITRE 16. CONFIGURER LES APPAREILS DU RÉSEAU POUR QU'ILS ACCEPTENT LE TRAFIC PROVENANT DE TOUTES LES ADRESSES MAC	196
16.1. CONFIGURER TEMPORAIREMENT UN APPAREIL POUR QU'IL ACCEPTE TOUT LE TRAFIC	196
16.2. CONFIGURATION PERMANENTE D'UN PÉRIPHÉRIQUE RÉSEAU POUR QU'IL ACCEPTE TOUT LE TRAFIC À L'AIDE DE NMCLI	197
16.3. CONFIGURATION PERMANENTE D'UN PÉRIPHÉRIQUE RÉSEAU POUR QU'IL ACCEPTE TOUT LE TRAFIC À L'AIDE DE NMSTATECTL	197
CHAPITRE 17. MISE EN PLACE D'UN SERVICE D'AUTHENTIFICATION RÉSEAU 802.1X POUR LES CLIENTS DU RÉSEAU LOCAL EN UTILISANT HOSTAPD AVEC FREERADIUS BACKEND	199
17.1. CONDITIONS PRÉALABLES	199
17.2. MISE EN PLACE DE LA PASSERELLE SUR L'AUTHENTIFICATEUR	199
17.3. EXIGENCES EN MATIÈRE DE CERTIFICAT PAR FREERADIUS	200
17.4. CRÉATION D'UN ENSEMBLE DE CERTIFICATS SUR UN SERVEUR FREERADIUS À DES FINS DE TEST	201
17.5. CONFIGURATION DE FREERADIUS POUR AUTHENTIFIER LES CLIENTS DU RÉSEAU EN TOUTE SÉCURITÉ À L'AIDE D'EAP	203
17.6. CONFIGURER HOSTAPD COMME AUTHENTIFICATEUR DANS UN RÉSEAU CÂBLÉ	207
17.7. TEST DE L'AUTHENTIFICATION EAP-TTLS CONTRE UN SERVEUR OU UN AUTHENTIFICATEUR FREERADIUS	209
17.8. TEST DE L'AUTHENTIFICATION EAP-TLS CONTRE UN SERVEUR OU UN AUTHENTIFICATEUR FREERADIUS	211
17.9. BLOQUER ET AUTORISER LE TRAFIC EN FONCTION DES ÉVÉNEMENTS D'AUTHENTIFICATION HOSTAPD	212
CHAPITRE 18. AUTHENTIFICATION D'UN CLIENT RHEL AU RÉSEAU À L'AIDE DE LA NORME 802.1X AVEC UN CERTIFICAT STOCKÉ SUR LE SYSTÈME DE FICHIERS	215
18.1. CONFIGURATION DE L'AUTHENTIFICATION RÉSEAU 802.1X SUR UNE CONNEXION ETHERNET EXISTANTE À L'AIDE DE NMCLI	215
18.2. CONFIGURATION D'UNE CONNEXION ETHERNET STATIQUE AVEC AUTHENTIFICATION RÉSEAU 802.1X À L'AIDE DE NMSTATECTL	216
18.3. CONFIGURATION D'UNE CONNEXION ETHERNET STATIQUE AVEC AUTHENTIFICATION RÉSEAU 802.1X À L'AIDE DU RÔLE DE SYSTÈME RHEL RÉSEAU	218
18.4. CONFIGURER UNE CONNEXION WIFI AVEC L'AUTHENTIFICATION RÉSEAU 802.1X EN UTILISANT LE RÔLE RÉSEAU RHEL SYSTEM ROLE	220
CHAPITRE 19. GESTION DE LA PASSERELLE PAR DÉFAUT	223
19.1. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DE NMCLI	223
19.2. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DU MODE INTERACTIF NMCLI	224
19.3. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DE NM- CONNECTION-EDITOR	225
19.4. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DU CENTRE DE CONTRÔLE	227
19.5. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DE NMSTATECTL	228
19.6. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DU RÔLE DE RÉSEAU RHEL SYSTEM ROLE	229
19.7. COMMENT NETWORKMANAGER GÈRE PLUSIEURS PASSERELLES PAR DÉFAUT	231
19.8. CONFIGURER NETWORKMANAGER POUR ÉVITER D'UTILISER UN PROFIL SPÉCIFIQUE POUR FOURNIR UNE PASSERELLE PAR DÉFAUT	232
19.9. CORRECTION D'UN COMPORTEMENT DE ROUTAGE INATTENDU DÙ À LA PRÉSENCE DE PLUSIEURS	

PASSERELLES PAR DÉFAUT	233
CHAPITRE 20. CONFIGURATION DES ROUTES STATIQUES	235
20.1. EXEMPLE DE RÉSEAU NÉCESSITANT DES ROUTES STATIQUES	235
20.2. COMMENT UTILISER LA COMMANDE NMCLI POUR CONFIGURER UNE ROUTE STATIQUE ?	237
20.3. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DE NMCLI	238
20.4. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DE NMTUI	239
20.5. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DU CENTRE DE CONTRÔLE	241
20.6. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DE NM-CONNECTION-EDITOR	243
20.7. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DU MODE INTERACTIF NMCLI	244
20.8. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DE NMSTATECTL	246
20.9. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DU RÔLE RÉSEAU RHEL SYSTEM ROLE	247
CHAPITRE 21. CONFIGURATION DU ROUTAGE BASÉ SUR DES RÈGLES POUR DÉFINIR DES ITINÉRAIRES ALTERNATIFS	250
21.1. ROUTAGE DU TRAFIC D'UN SOUS-RÉSEAU SPÉCIFIQUE VERS UNE PASSERELLE PAR DÉFAUT DIFFÉRENTE À L'AIDE DE NMCLI	250
21.2. ROUTAGE DU TRAFIC D'UN SOUS-RÉSEAU SPÉCIFIQUE VERS UNE PASSERELLE PAR DÉFAUT DIFFÉRENTE EN UTILISANT LE RÔLE DE SYSTÈME RHEL DU RÉSEAU	254
CHAPITRE 22. CRÉATION D'UNE INTERFACE FICTIVE	258
22.1. CRÉATION D'UNE INTERFACE FICTIVE AVEC UNE ADRESSE IPV4 ET IPV6 À L'AIDE DE NMCLI	258
CHAPITRE 23. UTILISATION DE NMSTATE-AUTOCONF POUR CONFIGURER AUTOMATIQUÉMENT L'ÉTAT DU RÉSEAU À L'AIDE DE LLDP	259
23.1. UTILISATION DE NMSTATE-AUTOCONF POUR CONFIGURER AUTOMATIQUÉMENT LES INTERFACES RÉSEAU	259
CHAPITRE 24. UTILISATION DE LLDP POUR DÉBOGUEUR LES PROBLÈMES DE CONFIGURATION DU RÉSEAU	262
24.1. DÉBOGAGE D'UNE CONFIGURATION VLAN INCORRECTE À L'AIDE D'INFORMATIONS LLDP	262
CHAPITRE 25. CRÉATION MANUELLE DE PROFILS NETWORKMANAGER AU FORMAT KEYFILE	265
25.1. FORMAT DU FICHIER CLÉ DES PROFILS NETWORKMANAGER	265
25.2. CRÉATION D'UN PROFIL NETWORKMANAGER AU FORMAT KEYFILE	266
25.3. MIGRATION DES PROFILS NETWORKMANAGER DU FORMAT IFCFG AU FORMAT KEYFILE	267
25.4. UTILISATION DE NMCLI POUR CRÉER DES PROFILS DE CONNEXION DE FICHIERS CLÉS EN MODE DÉCONNECTÉ	268
CHAPITRE 26. CIBLES ET SERVICES RÉSEAU SYSTEMD	272
26.1. DIFFÉRENCES ENTRE LA CIBLE SYSTEMD RÉSEAU ET LA CIBLE SYSTEMD RÉSEAU EN LIGNE	272
26.2. APERÇU DE NETWORKMANAGER-WAIT-ONLINE	272
26.3. CONFIGURER UN SERVICE SYSTEMD POUR QU'IL DÉMARRE APRÈS LE DÉMARRAGE DU RÉSEAU	273
CHAPITRE 27. CONTRÔLE DU TRAFIC SOUS LINUX	274
27.1. VUE D'ENSEMBLE DES DISCIPLINES DE LA FILE D'ATTENTE	274
27.2. INTRODUCTION AU SUIVI DES CONNEXIONS	275
27.3. INSPECTION DES QDISCS D'UNE INTERFACE RÉSEAU À L'AIDE DE L'UTILITAIRE TC	275
27.4. MISE À JOUR DU QDISC PAR DÉFAUT	276
27.5. DÉFINITION TEMPORAIRE DU QDISK ACTUEL D'UNE INTERFACE RÉSEAU À L'AIDE DE L'UTILITAIRE TC	277
27.6. FIXER DE FAÇON PERMANENTE LE QDISK COURANT D'UNE INTERFACE RÉSEAU EN UTILISANT NETWORKMANAGER	277
27.7. CONFIGURATION DE LA LIMITATION DU DÉBIT DES PAQUETS À L'AIDE DE L'UTILITAIRE TC-CTINFO	278
27.8. DISQUES DURS DISPONIBLES DANS RHEL	282

CHAPITRE 28. PREMIERS PAS AVEC TCP MULTIPATH	285
28.1. COMPRENDRE MPTCP	285
28.2. PRÉPARATION DE RHEL À LA PRISE EN CHARGE DE MPTCP	285
28.3. UTILISATION DE IPROUTE2 POUR CONFIGURER ET ACTIVER TEMPORAIREMENT PLUSIEURS CHEMINS POUR LES APPLICATIONS MPTCP	286
28.4. CONFIGURATION PERMANENTE DE CHEMINS MULTIPLES POUR LES APPLICATIONS MPTCP	288
28.5. SURVEILLANCE DES SOUS-FLUX MPTCP	290
28.6. DÉSACTIVATION DE TCP MULTIPATH DANS LE NOYAU	293
CHAPITRE 29. GESTION DU SERVICE MPTCPD	294
29.1. CONFIGURATION DE MPTCPD	294
29.2. GESTION DES APPLICATIONS AVEC L'OUTIL MPTCPIZE	294
29.3. ACTIVATION DES SOCKETS MPTCP POUR UN SERVICE À L'AIDE DE L'UTILITAIRE MPTCPIZE	295
CHAPITRE 30. CONFIGURATION DE L'ORDRE DES SERVEURS DNS	296
30.1. COMMENT NETWORKMANAGER ORDONNE LES SERVEURS DNS DANS /ETC/RESOLV.CONF	296
30.2. DÉFINITION D'UNE VALEUR DE PRIORITÉ DU SERVEUR DNS PAR DÉFAUT POUR L'ENSEMBLE DU NETWORKMANAGER	297
30.3. DÉFINITION DE LA PRIORITÉ DNS D'UNE CONNEXION NETWORKMANAGER	298
CHAPITRE 31. UTILISATION DE NETWORKMANAGER POUR DÉSACTIVER IPV6 POUR UNE CONNEXION SPÉCIFIQUE	299
31.1. DÉSACTIVATION D'IPV6 SUR UNE CONNEXION À L'AIDE DE NMCLI	299
CHAPITRE 32. AUGMENTATION DES TAMPONS DE L'ANNEAU POUR RÉDUIRE UN TAUX ÉLEVÉ DE PERTE DE PAQUETS	301
CHAPITRE 33. CONFIGURATION DES PARAMÈTRES DE LIAISON 802.3	303
33.1. COMPRENDRE L'AUTO-NÉGOCIATION	303
33.2. CONFIGURATION DES PARAMÈTRES DE LIAISON 802.3 À L'AIDE DE L'UTILITAIRE NMCLI	303
CHAPITRE 34. CONFIGURATION DES FONCTIONS DE DÉLESTAGE D'ETHTOOL	305
34.1. FONCTIONNALITÉS D'OFFLOAD SUPPORTÉES PAR NETWORKMANAGER	305
34.2. CONFIGURATION D'UNE FONCTION DE DÉLESTAGE ETHTOOL À L'AIDE DE NMCLI	307
34.3. CONFIGURATION D'UNE FONCTION DE DÉLESTAGE ETHTOOL À L'AIDE DU RÔLE DE SYSTÈME RHEL DE RÉSEAU	308
CHAPITRE 35. CONFIGURATION DES PARAMÈTRES ETHTOOL COALESCE	310
35.1. PARAMÈTRES DE COALESCENCE SUPPORTÉS PAR NETWORKMANAGER	310
35.2. CONFIGURATION DES PARAMÈTRES ETHTOOL COALESCE À L'AIDE DE NMCLI	311
35.3. CONFIGURATION DES PARAMÈTRES DE COALESCENCE D'UN OUTIL ETHTOOL À L'AIDE DU RÔLE DE SYSTÈME RHEL DE RÉSEAU	311
CHAPITRE 36. UTILISATION DE MACSEC POUR CRYPTER LE TRAFIC DE COUCHE 2 DANS LE MÊME RÉSEAU PHYSIQUE	314
36.1. CONFIGURATION D'UNE CONNEXION MACSEC À L'AIDE DE NMCLI	314
36.2. RESSOURCES SUPPLÉMENTAIRES	316
CHAPITRE 37. UTILISER DES SERVEURS DNS DIFFÉRENTS POUR DES DOMAINES DIFFÉRENTS	317
37.1. ENVOI DE REQUÊTES DNS POUR UN DOMAINE SPÉCIFIQUE À UN SERVEUR DNS SÉLECTIONNÉ	317
CHAPITRE 38. PREMIERS PAS AVEC IPVLAN	319
38.1. MODES IPVLAN	319
38.2. COMPARAISON ENTRE IPVLAN ET MACVLAN	319
38.3. CRÉATION ET CONFIGURATION DU DISPOSITIF IPVLAN À L'AIDE DE IPROUTE2	320
CHAPITRE 39. RÉUTILISATION DE LA MÊME ADRESSE IP SUR DIFFÉRENTES INTERFACES	322

39.1. RÉUTILISATION PERMANENTE DE LA MÊME ADRESSE IP SUR DIFFÉRENTES INTERFACES	322
39.2. RÉUTILISATION TEMPORAIRE DE LA MÊME ADRESSE IP SUR DIFFÉRENTES INTERFACES	323
39.3. RESSOURCES SUPPLÉMENTAIRES	325
CHAPITRE 40. DÉMARRAGE D'UN SERVICE DANS UN RÉSEAU VRF ISOLÉ	326
40.1. CONFIGURATION D'UN DISPOSITIF VRF	326
40.2. DÉMARRAGE D'UN SERVICE DANS UN RÉSEAU VRF ISOLÉ	328
CHAPITRE 41. EXÉCUTION DES CROCHETS DE SORTIE DE DHCLIENT À L'AIDE D'UN SCRIPT DE DISTRIBUTION DE NETWORKMANAGER	330
41.1. LE CONCEPT DE SCRIPTS DE DISTRIBUTION DU NETWORKMANAGER	330
41.2. CRÉATION D'UN SCRIPT DE DISTRIBUTION DE NETWORKMANAGER QUI EXÉCUTE LES CROCHETS DE SORTIE DE DHCLIENT	331
CHAPITRE 42. INTRODUCTION AU DÉBOGAGE DE NETWORKMANAGER	332
42.1. INTRODUCTION À LA MÉTHODE DE RÉAPPLICATION DU NETWORKMANAGER	332
42.2. CONFIGURATION DU NIVEAU DE JOURNALISATION DE NETWORKMANAGER	334
42.3. RÉGLAGE TEMPORAIRE DES NIVEAUX DE JOURNALISATION AU MOMENT DE L'EXÉCUTION EN UTILISANT NMCLI	335
42.4. VISUALISATION DES JOURNAUX DE NETWORKMANAGER	336
42.5. NIVEAUX ET DOMAINES DE DÉBOGAGE	336
CHAPITRE 43. INTRODUCTION À NMSTATE	338
43.1. UTILISATION DE LA BIBLIOTHÈQUE LIBNMSTATE DANS UNE APPLICATION PYTHON	338
43.2. MISE À JOUR DE LA CONFIGURATION ACTUELLE DU RÉSEAU À L'AIDE DE NMSTATECTL	338
43.3. ÉTATS DU RÉSEAU POUR LE RÔLE DE SYSTÈME RHEL	339
43.4. RESSOURCES SUPPLÉMENTAIRES	340
CHAPITRE 44. CAPTURER DES PAQUETS RÉSEAU	341
44.1. UTILISATION DE XDPDUMP POUR CAPTURER DES PAQUETS RÉSEAU, Y COMPRIS LES PAQUETS ABANDONNÉS PAR LES PROGRAMMES XDP	341
44.2. RESSOURCES SUPPLÉMENTAIRES	342
CHAPITRE 45. DÉMARRER AVEC DPDK	343
45.1. INSTALLATION DU PAQUET DPDK	343
45.2. RESSOURCES SUPPLÉMENTAIRES	343
CHAPITRE 46. COMPRENDRE LES FONCTIONNALITÉS DE MISE EN RÉSEAU EBPf DANS RHEL 9	344
46.1. VUE D'ENSEMBLE DES FONCTIONNALITÉS DE MISE EN RÉSEAU EBPf DANS RHEL 9	344
46.2. VUE D'ENSEMBLE DES FONCTIONNALITÉS XDP DANS RHEL 9 PAR CARTES RÉSEAU	347
CHAPITRE 47. TRAÇAGE DE RÉSEAUX À L'AIDE DE LA COLLECTION DE COMPILATEURS BPF	350
47.1. INSTALLATION DU PAQUETAGE BCC-TOOLS	350
47.2. AFFICHAGE DES CONNEXIONS TCP AJOUTÉES À LA FILE D'ATTENTE D'ACCEPTATION DU NOYAU	350
47.3. TRAÇAGE DES TENTATIVES DE CONNEXION TCP SORTANTES	351
47.4. MESURE DE LA LATENCE DES CONNEXIONS TCP SORTANTES	352
47.5. AFFICHAGE DES DÉTAILS CONCERNANT LES PAQUETS ET LES SEGMENTS TCP QUI ONT ÉTÉ ABANDONNÉS PAR LE NOYAU	352
47.6. TRAÇAGE DES SESSIONS TCP	353
47.7. SUIVI DES RETRANSMISSIONS TCP	354
47.8. AFFICHAGE DES INFORMATIONS SUR LES CHANGEMENTS D'ÉTAT DE TCP	354
47.9. RÉSUMER ET AGRÉGER LE TRAFIC TCP ENVOYÉ À DES SOUS-RÉSEAUX SPÉCIFIQUES	355
47.10. AFFICHAGE DU DÉBIT DU RÉSEAU PAR ADRESSE IP ET PAR PORT	356
47.11. TRAÇAGE DES CONNEXIONS TCP ÉTABLIES	356
47.12. TRAÇAGE DES TENTATIVES D'ÉCOUTE IPV4 ET IPV6	357

47.13. RÉSUMÉ DU TEMPS DE SERVICE DES INTERRUPTIONS DOUCES	358
47.14. RÉSUMÉ DE LA TAILLE ET DU NOMBRE DE PAQUETS SUR UNE INTERFACE RÉSEAU	358
47.15. RESSOURCES SUPPLÉMENTAIRES	359
CHAPITRE 48. DÉMARRER AVEC LE TIPC	360
48.1. L'ARCHITECTURE DU TIPC	360
48.2. CHARGEMENT DU MODULE TIPC AU DÉMARRAGE DU SYSTÈME	360
48.3. CRÉER UN RÉSEAU TIPC	361
48.4. RESSOURCES SUPPLÉMENTAIRES	362
CHAPITRE 49. CONFIGURATION AUTOMATIQUE DES INTERFACES RÉSEAU DANS LES NUAGES PUBLICS À L'AIDE DE NM-CLOUD-SETUP	364
49.1. CONFIGURATION ET PRÉ-DÉPLOIEMENT DE NM-CLOUD-SETUP	364

RENDRE L'OPEN SOURCE PLUS INCLUSIF

Red Hat s'engage à remplacer les termes problématiques dans son code, sa documentation et ses propriétés Web. Nous commençons par ces quatre termes : master, slave, blacklist et whitelist. En raison de l'ampleur de cette entreprise, ces changements seront mis en œuvre progressivement au cours de plusieurs versions à venir. Pour plus de détails, voir le [message de notre directeur technique Chris Wright](#).

FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT

Nous apprécions vos commentaires sur notre documentation. Faites-nous savoir comment nous pouvons l'améliorer.

Soumettre des commentaires sur des passages spécifiques

1. Consultez la documentation au format **Multi-page HTML** et assurez-vous que le bouton **Feedback** apparaît dans le coin supérieur droit après le chargement complet de la page.
2. Utilisez votre curseur pour mettre en évidence la partie du texte que vous souhaitez commenter.
3. Cliquez sur le bouton **Add Feedback** qui apparaît près du texte en surbrillance.
4. Ajoutez vos commentaires et cliquez sur **Submit**.

Soumettre des commentaires via Bugzilla (compte requis)

1. Connectez-vous au site Web de [Bugzilla](#).
2. Sélectionnez la version correcte dans le menu **Version**.
3. Saisissez un titre descriptif dans le champ **Summary**.
4. Saisissez votre suggestion d'amélioration dans le champ **Description**. Incluez des liens vers les parties pertinentes de la documentation.
5. Cliquez sur **Submit Bug**.

CHAPITRE 1. DÉNOMINATION COHÉRENTE DES PÉRIPHÉRIQUES D'INTERFACE RÉSEAU

Le noyau Linux attribue des noms aux interfaces réseau en combinant un préfixe fixe et un nombre qui augmente au fur et à mesure que le noyau initialise les périphériques réseau. Par exemple, **eth0** représente le premier périphérique interrogé au démarrage. Si vous ajoutez une autre carte d'interface réseau au système, l'attribution des noms de périphériques du noyau n'est plus fixe. Par conséquent, après un redémarrage, le noyau peut nommer le périphérique différemment.

Pour résoudre ce problème, le gestionnaire de périphériques **udev** prend en charge plusieurs schémas de dénomination différents. Par défaut, **udev** attribue des noms fixes basés sur le micrologiciel, la topologie et les informations de localisation. Cela présente les avantages suivants :

- Les noms des appareils sont entièrement prévisibles.
- Les noms de périphériques restent fixes même si vous ajoutez ou supprimez du matériel, car il n'y a pas de nouvelle énumération.
- Le matériel défectueux peut être remplacé sans problème.



AVERTISSEMENT

Red Hat ne prend pas en charge les systèmes dont le nommage cohérent des périphériques est désactivé. Pour plus de détails, voir [Est-il sûr de définir net.ifnames=0 ?](#)

1.1. HIÉRARCHIE DE DÉNOMINATION DES PÉRIPHÉRIQUES D'INTERFACE RÉSEAU

Si la dénomination cohérente des périphériques est activée, ce qui est le cas par défaut dans Red Hat Enterprise Linux, le gestionnaire de périphériques **udev** génère des noms de périphériques basés sur les schémas suivants :

Régime	Description	Exemple :
1	Les noms des périphériques intègrent les numéros d'index fournis par le firmware ou le BIOS pour les périphériques embarqués. Si cette information n'est pas disponible ou applicable, udev utilise le schéma 2.	eno1
2	Les noms des dispositifs intègrent les numéros d'index de l'emplacement du hot plug PCI Express (PCIe) fournis par le micrologiciel ou le BIOS. Si cette information n'est pas disponible ou applicable, udev utilise le schéma 3.	ens1
3	Les noms de dispositifs intègrent l'emplacement physique du connecteur du matériel. Si cette information n'est pas disponible ou applicable, udev utilise le schéma 5.	enp2s0

Régime	Description	Exemple :
4	Les noms de périphériques intègrent l'adresse MAC. Red Hat Enterprise Linux n'utilise pas ce schéma par défaut, mais les administrateurs peuvent l'utiliser de manière optionnelle.	enx525400d5e0fb
5	Le schéma traditionnel de dénomination imprévisible du noyau. Si udev ne peut appliquer aucun des autres schémas, le gestionnaire de périphériques utilise ce schéma.	eth0

Par défaut, Red Hat Enterprise Linux sélectionne le nom du périphérique en fonction du paramètre **NamePolicy** dans le fichier **/usr/lib/systemd/network/99-default.link**. L'ordre des valeurs dans **NamePolicy** est important. Red Hat Enterprise Linux utilise le premier nom de périphérique qui est à la fois spécifié dans le fichier et généré par **udev**.

Si vous avez configuré manuellement les règles **udev** pour modifier le nom des périphériques du noyau, ces règles sont prioritaires.

1.2. COMMENT FONCTIONNE LE RENOMMAGE DES PÉRIPHÉRIQUES RÉSEAU

Par défaut, la dénomination cohérente des périphériques est activée dans Red Hat Enterprise Linux. Le gestionnaire de périphériques **udev** traite différentes règles pour renommer les périphériques. Le service **udev** traite ces règles dans l'ordre suivant :

1. Le fichier **/usr/lib/udev/rules.d/60-net.rules** définit que l'utilitaire d'aide **/lib/udev/rename_device** doit rechercher le paramètre **HWADDR** dans les fichiers **/etc/sysconfig/network-scripts/ifcfg-***. Si la valeur définie dans la variable correspond à l'adresse MAC d'une interface, l'utilitaire d'aide renomme l'interface avec le nom défini dans le paramètre **DEVICE** du fichier. Ce fichier n'existe qu'après l'installation du paquetage **initscripts**.
2. Le fichier **/usr/lib/udev/rules.d/71-biosdevname.rules** définit que l'utilitaire **biosdevname** renomme l'interface conformément à sa politique de dénomination, à condition qu'elle n'ait pas été renommée à l'étape précédente.
3. Le fichier **/usr/lib/udev/rules.d/75-net-description.rules** définit que **udev** examine le périphérique d'interface réseau et définit les propriétés dans **udev-** variables internes qui seront traitées à l'étape suivante. Notez que certaines de ces propriétés peuvent être indéfinies.
4. Le fichier **/usr/lib/udev/rules.d/80-net-setup-link.rules** appelle le fichier **net_setup_link udev** intégré qui applique alors la politique. Voici la politique par défaut qui est stockée dans le fichier **/usr/lib/systemd/network/99-default.link**:

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

Avec cette politique, si le noyau utilise un nom persistant, **udev** ne renomme pas l'interface. Si le noyau n'utilise pas de nom persistant, **udev** renomme l'interface avec le nom fourni par la base de données matérielle de **udev**. Si cette base de données n'est pas disponible, Red Hat Enterprise Linux se rabat sur les mécanismes décrits ci-dessus.

Il est également possible de définir le paramètre **NamePolicy** de ce fichier sur **mac** pour les noms d'interface basés sur l'adresse MAC (Media Access Control).

5. Le fichier `/usr/lib/udev/rules.d/80-net-setup-link.rules` définit que **udev** renomme l'interface en fonction des paramètres internes **udev** dans l'ordre suivant :
 - a. **ID_NET_NAME_ONBOARD**
 - b. **ID_NET_NAME_SLOT**
 - c. **ID_NET_NAME_PATH**

Si l'un des paramètres n'est pas défini, **udev** utilise le suivant. Si aucun des paramètres n'est défini, l'interface n'est pas renommée.

Les étapes 3 et 4 mettent en œuvre les schémas de dénomination 1 à 4 décrits dans [Hiérarchie de dénomination des périphériques d'interface réseau](#).

Ressources supplémentaires

- [Personnalisation du préfixe des interfaces Ethernet lors de l'installation](#)
- [Pourquoi les noms des interfaces réseau de systemd diffèrent-ils entre les principales versions de RHEL ? solution](#)
- **systemd.link(5)** page de manuel

1.3. EXPLICATION DES NOMS PRÉVISIBLES DES PÉRIPHÉRIQUES D'INTERFACE RÉSEAU SUR LA PLATE-FORME X86_64

Lorsque la fonction de nom cohérent des périphériques réseau est activée, le gestionnaire de périphériques **udev** crée les noms des périphériques en fonction de différents critères. Le nom de l'interface commence par un préfixe de deux caractères en fonction du type d'interface :

- **en** pour Ethernet
- **wl** pour le réseau local sans fil (WLAN)
- **ww** pour les réseaux étendus sans fil (WWAN)

En outre, l'un des éléments suivants est ajouté à l'un des préfixes susmentionnés en fonction du schéma appliqué par le gestionnaire de périphériques **udev**:

- **o<on-board_index_number>**
- **s<hot_plug_slot_index_number>[f<function>][d<device_id>]**
Notez que tous les périphériques PCI multifonctions ont le numéro **[f<function>]** dans le nom du périphérique, y compris le périphérique de fonction **0**.
- **x<MAC_address>**
- **[P<domain_number>]p<bus>s<slot>[f<function>][d<device_id>]**
La partie **[P<domain_number>]** définit l'emplacement géographique du PCI. Cette partie n'est définie que si le numéro de domaine n'est pas **0**.

- **[P<domain_number>]p<bus>s<slot>[f<function>][u<usb_port>][...][c<config>][i<interface>]**

Pour les dispositifs USB, la chaîne complète des numéros de port des concentrateurs est composée. Si le nom est plus long que le maximum (15 caractères), il n'est pas exporté. S'il y a plusieurs dispositifs USB dans la chaîne, **udev** supprime les valeurs par défaut des descripteurs de configuration USB (**c1**) et des descripteurs d'interface USB (**i0**).

1.4. EXPLICATION DES NOMS PRÉVISIBLES DES PÉRIPHÉRIQUES D'INTERFACE RÉSEAU SUR LA PLATE-FORME SYSTEM Z

Lorsque la fonction de nom de périphérique réseau cohérent est activée, le gestionnaire de périphériques **udev** sur la plate-forme System z crée les noms de périphériques en fonction de l'ID de bus. L'ID de bus identifie un périphérique dans le sous-système de canaux s390.

Pour un dispositif à mot de commande de canal (CCW), l'ID de bus est le numéro de dispositif avec un préfixe **0.n** où **n** est l'ID de l'ensemble de sous-canaux.

Les interfaces Ethernet sont nommées, par exemple, **enccw0.0.1234**. Les périphériques réseau de type canal à canal (CTC) du protocole Internet de ligne série (SLIP) sont nommés, par exemple, **slccw0.0.1234**.

Utilisez les commandes **znetconf -c** ou **lscss -a** pour afficher les périphériques réseau disponibles et leurs ID de bus.

Red Hat Enterprise Linux prend également en charge les noms d'interface prévisibles et persistants pour les fonctions PCI RDMA over Converged Ethernet (RoCE) Express. Deux identifiants fournissent des noms d'interface prévisibles : l'identifiant de l'utilisateur (UID) et l'identifiant de la fonction (FID). Pour obtenir des noms d'interface prévisibles basés sur l'UID, un système doit imposer l'unicité de l'UID, qui est le schéma de dénomination préféré. Si aucun UID unique n'est disponible, RHEL utilise les FID pour définir les noms d'interface prévisibles.

1.5. PERSONNALISATION DU PRÉFIXE DES INTERFACES ETHERNET LORS DE L'INSTALLATION

Vous pouvez personnaliser le préfixe des noms d'interface Ethernet lors de l'installation de Red Hat Enterprise Linux.



IMPORTANT

Red Hat ne prend pas en charge la personnalisation du préfixe à l'aide de l'utilitaire **prefixdevname** sur des systèmes déjà déployés.

Après l'installation de RHEL, le service **udev** nomme les périphériques Ethernet **<prefix>.<index>**. Par exemple, si vous sélectionnez le préfixe **net**, RHEL nomme les interfaces Ethernet **net0**, **net1**, etc.

Conditions préalables

- Le préfixe que vous souhaitez définir répond aux exigences suivantes :
 - Il se compose de caractères ASCII.
 - Il s'agit d'une chaîne alphanumérique.
 - Il est inférieur à 16 caractères.

- Il n'entre pas en conflit avec d'autres préfixes bien connus utilisés pour nommer les interfaces réseau, tels que **eth**, **eno**, **ens** et **em**.

Procédure

1. Démarrez le support d'installation de Red Hat Enterprise Linux.
2. Dans le gestionnaire de démarrage :
 - a. Sélectionnez l'entrée **Install Red Hat Enterprise Linux <version>** et appuyez sur **Tab** pour modifier l'entrée.
 - b. Ajouter **net.ifnames.prefix=<prefix>** aux options du noyau.
 - c. Appuyez sur **Entrée** pour lancer le programme d'installation.
3. Installer Red Hat Enterprise Linux.

Vérification

- Après l'installation, affichez les interfaces Ethernet :

```
# ip link show
...
2: net0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
   link/ether 00:53:00:c5:98:1c brd ff:ff:ff:ff:ff:ff
3: net1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
   link/ether 00:53:00:c2:39:9e brd ff:ff:ff:ff:ff:ff
...
```

1.6. ATTRIBUTION DE NOMS D'INTERFACE RÉSEAU DÉFINIS PAR L'UTILISATEUR À L'AIDE DES RÈGLES UDEV

Le gestionnaire de périphériques **udev** prend en charge un ensemble de règles permettant de personnaliser les noms d'interface.

Procédure

1. Affiche toutes les interfaces réseau et leurs adresses MAC :

```
# ip link list

enp6s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
   link/ether b4:96:91:14:ae:58 brd ff:ff:ff:ff:ff:ff
enp6s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
   link/ether b4:96:91:14:ae:5a brd ff:ff:ff:ff:ff:ff
enp4s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
   link/ether 00:90:fa:6a:7d:90 brd ff:ff:ff:ff:ff:ff
```

2. Créez le fichier `/etc/udev/rules.d/70-custom-ifnames.rules` avec le contenu suivant :

```
SUBSYSTEM=="net",ACTION=="add",ATTR{address}=="b4:96:91:14:ae:58",ATTR{type}=="1",NAME="provider0"
SUBSYSTEM=="net",ACTION=="add",ATTR{address}=="b4:96:91:14:ae:5a",ATTR{type}=="1",NAME="provider1"
SUBSYSTEM=="net",ACTION=="add",ATTR{address}=="00:90:fa:6a:7d:90",ATTR{type}=="1",NAME="dmz"
```

Ces règles correspondent à l'adresse MAC des interfaces réseau et les renomment en fonction du nom indiqué dans la propriété **NAME**. Dans ces exemples, la valeur du paramètre **ATTR{type} 1** définit que l'interface est de type Ethernet.

Vérification

1. Redémarrer le système.

```
# reboot
```

2. Vérifiez que les noms d'interface de chaque adresse MAC correspondent à la valeur définie dans le paramètre **NAME** du fichier de règles :

```
# ip link show

provider0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DEFAULT group default qlen 1000
    link/ether b4:96:91:14:ae:58 brd ff:ff:ff:ff:ff:ff
    altname enp6s0f0
provider1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DEFAULT group default qlen 1000
    link/ether b4:96:91:14:ae:5a brd ff:ff:ff:ff:ff:ff
    altname enp6s0f1
dmz: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DEFAULT group default qlen 1000
    link/ether 00:90:fa:6a:7d:90 brd ff:ff:ff:ff:ff:ff
    altname enp4s0f0
```

Ressources supplémentaires

- [udev\(7\)](#) page de manuel
- [udevadm\(8\)](#) page de manuel
- `/usr/src/kernels/<kernel_version>/include/uapi/linux/if_arp.h` fournie par le paquet **kernel-doc**

1.7. ATTRIBUTION DE NOMS D'INTERFACE RÉSEAU DÉFINIS PAR L'UTILISATEUR À L'AIDE DES FICHIERS DE LIAISON DE SYSTEMD

Créez un schéma de dénomination en renommant les interfaces réseau en **provider0**.

Procédure

1. Affiche les noms de toutes les interfaces et leurs adresses MAC :

```
# ip link show
```

```
enp6s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
    link/ether b4:96:91:14:ae:58 brd ff:ff:ff:ff:ff:ff
enp6s0f1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
    link/ether b4:96:91:14:ae:5a brd ff:ff:ff:ff:ff:ff
enp4s0f0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
    link/ether 00:90:fa:6a:7d:90 brd ff:ff:ff:ff:ff:ff
```

2. Pour nommer l'interface avec l'adresse MAC **b4:96:91:14:ae:58** à **provider0**, créez le fichier `/etc/systemd/network/70-custom-ifnames.link` avec le contenu suivant :

```
[Match]
MACAddress=b4:96:91:14:ae:58

[Link]
Name=provider0
```

Ce fichier de liaison correspond à une adresse MAC et renomme l'interface réseau en fonction du nom défini dans le paramètre **Name**.

Vérification

1. Redémarrer le système :

```
# reboot
```

2. Vérifiez que l'appareil portant l'adresse MAC que vous avez spécifiée dans le fichier de liaison a été attribué à **provider0**:

```
# ip link show
```

```
provider0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UP mode
DEFAULT group default qlen 1000
    link/ether b4:96:91:14:ae:58 brd ff:ff:ff:ff:ff:ff
```

Ressources supplémentaires

- [systemd.link\(5\)](#) page de manuel

1.8. ATTRIBUTION DE NOMS SUPPLÉMENTAIRES À L'INTERFACE RÉSEAU À L'AIDE DES FICHIERS DE LIAISON DE SYSTEMD

Le nommage alternatif des interfaces permet au noyau de définir d'autres noms pour les interfaces réseau. Par défaut, il fournit le même schéma de dénomination que le paramètre de dénomination d'interface normal - **NamePolicy**. Vous pouvez écrire votre fichier de lien **systemd** personnalisé à l'aide des directives **AlternativeNamesPolicy** ou **AlternativeName** pour donner des noms alternatifs aux interfaces réseau de votre choix.

La dernière mise en œuvre de la dénomination alternative des interfaces vous permet de.. :

- Créer des noms alternatifs de longueur arbitraire.
- Avoir un ou plusieurs noms alternatifs pour la même interface réseau.
- Utiliser des noms alternatifs pour les commandes.

Conditions préalables

- Vous connaissez l'adresse MAC (Media Access Control) ou un autre identifiant d'interface réseau. Pour plus de détails, voir la partie OPTIONS DE SECTION [MATCH] à l'adresse **systemd.link(5)**.

Procédure

1. Créez le fichier **/etc/systemd/network/10-altnames.link** avec le contenu suivant :

```
[Match]
MACAddress=52:54:00:76:e0:2a

[Link]
AlternativeName=production_alias_of_arbitrary_length
AlternativeName=PRD
```

2. Redémarrez votre système pour que les modifications soient prises en compte.

Vérification

- Vous pouvez utiliser le nom alternatif pour afficher l'état de l'interface réseau :

```
# ip address show production_alias_of_arbitrary_length
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
    link/ether 52:54:00:76:e0:2a brd ff:ff:ff:ff:ff:ff
    altname production_alias_of_arbitrary_length
    altname PRD
    inet 192.0.2.1/24 brd 192.0.2.255 scope global dynamic noprefixroute enp1s0
        valid_lft 2760sec preferred_lft 2760sec
    inet6 2001:db8::/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- [Hiérarchie de dénomination des périphériques d'interface réseau](#)
- [Comment fonctionne le renommage des périphériques réseau](#)
- [Que signifie AlternativeNamesPolicy dans le schéma de dénomination de l'interface ?](#)

CHAPITRE 2. CONFIGURATION D'UNE CONNEXION ETHERNET

Red Hat Enterprise Linux offre aux administrateurs différentes options pour configurer les connexions Ethernet. Par exemple :

- Utilisez **nmcli** pour configurer les connexions sur la ligne de commande.
- Utilisez **nmtui** pour configurer les connexions dans une interface utilisateur textuelle.
- Utilisez les rôles système RHEL pour automatiser la configuration des connexions sur un ou plusieurs hôtes.
- Utilisez le menu Paramètres de GNOME ou l'application **nm-connection-editor** pour configurer les connexions dans une interface graphique.
- Utilisez **nmstatectl** pour configurer les connexions via l'API Nmstate.



NOTE

Si vous souhaitez configurer manuellement les connexions Ethernet sur des hôtes fonctionnant dans le nuage Microsoft Azure, désactivez le service **cloud-init** ou configurez-le de manière à ce qu'il ignore les paramètres réseau récupérés dans l'environnement du nuage. Sinon, **cloud-init** remplacera au prochain redémarrage les paramètres réseau que vous avez configurés manuellement.

2.1. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE À L'AIDE DE NMCLI

Pour configurer une connexion Ethernet sur la ligne de commande, utilisez l'utilitaire **nmcli**.

Par exemple, la procédure ci-dessous crée un profil de connexion NetworkManager pour le périphérique **enp7s0** avec les paramètres suivants :

- Une adresse IPv4 statique - **192.0.2.1** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **192.0.2.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::ffe**
- Un serveur DNS IPv4 - **192.0.2.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.

Procédure

1. Ajouter un nouveau profil de connexion NetworkManager pour la connexion Ethernet :

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

Les étapes suivantes modifient le profil de connexion **Example-Connection** que vous avez créé.

2. Définir l'adresse IPv4 :

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

3. Définir l'adresse IPv6 :

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

4. Réglez la méthode de connexion IPv4 et IPv6 sur **manual**:

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

5. Définir les passerelles par défaut IPv4 et IPv6 :

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

6. Définissez les adresses des serveurs DNS IPv4 et IPv6 :

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

Pour définir plusieurs serveurs DNS, indiquez-les en les séparant par des espaces et en les plaçant entre guillemets.

7. Définir le domaine de recherche DNS pour la connexion IPv4 et IPv6 :

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

8. Activer le profil de connexion :

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

Vérification

1. Affiche l'état des appareils et des connexions :

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

2. Utilisez l'utilitaire **ping** pour vérifier que cet hôte peut envoyer des paquets à d'autres hôtes :

```
# ping host_name_or_IP_address
```

Résolution de problèmes

- Vérifiez que le câble réseau est branché sur l'hôte et sur un commutateur.
- Vérifiez si la défaillance de la liaison existe uniquement sur cet hôte ou également sur d'autres hôtes connectés au même commutateur.
- Vérifiez que le câble réseau et l'interface réseau fonctionnent comme prévu. Effectuez les étapes de diagnostic du matériel et remplacez les câbles et les cartes d'interface réseau défectueux.
- Si la configuration du disque ne correspond pas à celle du périphérique, le démarrage ou le redémarrage de NetworkManager crée une connexion en mémoire qui reflète la configuration du périphérique. Pour plus de détails et pour savoir comment éviter ce problème, voir [NetworkManager duplique une connexion après le redémarrage du service NetworkManager](#) .

Ressources supplémentaires

- **nm-settings(5)** page de manuel
- **nmcli(1)** page de manuel
- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

2.2. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE À L'AIDE DE L'ÉDITEUR INTERACTIF NMCLI

Vous pouvez configurer une connexion Ethernet sur la ligne de commande en utilisant le mode interactif de l'utilitaire **nmcli**.

Par exemple, la procédure ci-dessous crée un profil de connexion NetworkManager pour le périphérique **enp7s0** avec les paramètres suivants :

- Une adresse IPv4 statique - **192.0.2.1** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **192.0.2.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::fffe**
- Un serveur DNS IPv4 - **192.0.2.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.

Procédure

1. Pour ajouter un nouveau profil de connexion NetworkManager pour la connexion Ethernet et démarrer le mode interactif, entrez :

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

2. Définir l'interface réseau :

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

3. Définir l'adresse IPv4 :

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

4. Définir l'adresse IPv6 :

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

5. Réglez la méthode de connexion IPv4 et IPv6 sur **manual**:

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

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

6. Définir les passerelles par défaut IPv4 et IPv6 :

```
nmcli> set ipv4.gateway 192.0.2.254
```

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

7. Définissez les adresses des serveurs DNS IPv4 et IPv6 :

```
nmcli> set ipv4.dns 192.0.2.200
```

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

Pour définir plusieurs serveurs DNS, indiquez-les en les séparant par des espaces et en les plaçant entre guillemets.

8. Définir le domaine de recherche DNS pour la connexion IPv4 et IPv6 :

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

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

9. Sauvegarder et activer la connexion :

```
nmcli> save persistent
```

```
Saving the connection with 'autoconnect=yes'. That might result in an immediate activation of the connection.
```

```
Do you still want to save? (yes/no) [yes] yes
```

10. Quitter le mode interactif :

```
nmcli> quit
```

Vérification

1. Affiche l'état des appareils et des connexions :

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

2. Utilisez l'utilitaire **ping** pour vérifier que cet hôte peut envoyer des paquets à d'autres hôtes :

```
# ping host_name_or_IP_address
```

Résolution de problèmes

- Vérifiez que le câble réseau est branché sur l'hôte et sur un commutateur.
- Vérifiez si la défaillance de la liaison existe uniquement sur cet hôte ou également sur d'autres hôtes connectés au même commutateur.
- Vérifiez que le câble réseau et l'interface réseau fonctionnent comme prévu. Effectuez les étapes de diagnostic du matériel et remplacez les câbles et les cartes d'interface réseau défectueux.
- Si la configuration du disque ne correspond pas à celle du périphérique, le démarrage ou le redémarrage de NetworkManager crée une connexion en mémoire qui reflète la configuration du périphérique. Pour plus de détails et pour savoir comment éviter ce problème, voir [NetworkManager duplique une connexion après le redémarrage du service NetworkManager](#) .

Ressources supplémentaires

- **nm-settings(5)** page de manuel
- **nmcli(1)** page de manuel
- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

2.3. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE À L'AIDE DE NMTUI

L'application **nmtui** fournit une interface utilisateur textuelle pour NetworkManager. Vous pouvez utiliser **nmtui** pour configurer une connexion Ethernet avec une adresse IP statique sur un hôte sans interface graphique.



NOTE

Sur **nmtui**:

- Naviguer à l'aide des touches du curseur.
- Appuyez sur un bouton en le sélectionnant et en appuyant sur **Entrée**.
- Sélectionnez et désélectionnez les cases à cocher en utilisant l'**espace**.

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.

Procédure

1. Si vous ne connaissez pas le nom du périphérique réseau que vous souhaitez utiliser pour la connexion, affichez les périphériques disponibles :

```
# nmcli device status
DEVICE  TYPE    STATE      CONNECTION
enp7s0  ethernet unavailable  --
...
```

2. Démarrer **nmtui**:

```
# nmtui
```

3. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
4. Appuyez sur le bouton **Add**.
5. Sélectionnez **Ethernet** dans la liste des types de réseaux et appuyez sur **Entrée**.
6. Optionnel : Entrez un nom pour le profil NetworkManager à créer.
7. Saisissez le nom de l'appareil réseau dans le champ **Device**.
8. Configurez les paramètres des adresses IPv4 et IPv6 dans les zones **IPv4 configuration** et **IPv6 configuration**:
 - a. Appuyez sur la touche **Automatic** et sélectionnez **Manual** dans la liste affichée.
 - b. Appuyez sur le bouton **Show** en regard du protocole que vous souhaitez configurer pour afficher des champs supplémentaires.
 - c. Appuyez sur le bouton **Add** à côté de **Addresses**, et entrez l'adresse IP et le masque de sous-réseau au format CIDR (Classless Inter-Domain Routing).
Si vous ne spécifiez pas de masque de sous-réseau, NetworkManager définit un masque de sous-réseau **/32** pour les adresses IPv4 et **/64** pour les adresses IPv6.
 - d. Saisissez l'adresse de la passerelle par défaut.
 - e. Appuyez sur la touche **Add** à côté de **DNS servers**, et entrez l'adresse du serveur DNS.
 - f. Appuyez sur la touche **Add** à côté de **Search domains**, et entrez le domaine de recherche DNS.

Figure 2.1. Exemple d'une connexion Ethernet avec des paramètres d'adresse IP statiques

Edit Connection

Profile name `Example-Connection`
 Device `enp7s0`

= ETHERNET <Show>

= IPv4 CONFIGURATION `<Manual>` <Hide>

Addresses `192.0.2.1/24` <Remove>
<Add...>

Gateway `192.0.2.254`

DNS servers `192.0.2.200` <Remove>
<Add...>

Search domains `example.com` <Remove>
<Add...>

Routing (No custom routes) <Edit...>

Never use this network for default route
 Ignore automatically obtained routes
 Ignore automatically obtained DNS parameters

Require IPv4 addressing for this connection

= IPv6 CONFIGURATION `<Manual>` <Hide>

Addresses `2001:db8:1::1/64` <Remove>
<Add...>

Gateway `2001:db8:1::fffe`

DNS servers `2001:db8:1::ffbb` <Remove>
<Add...>

Search domains `example.com` <Remove>
<Add...>

Routing (No custom routes) <Edit...>

Never use this network for default route
 Ignore automatically obtained routes
 Ignore automatically obtained DNS parameters

Require IPv6 addressing for this connection

Automatically connect
 Available to all users

<Cancel> <OK>

9. Appuyez sur le bouton **OK** pour créer et activer automatiquement la nouvelle connexion.
10. Appuyez sur le bouton **Back** pour revenir au menu principal.
11. Sélectionnez **Quit** et appuyez sur **Entrée** pour fermer l'application **nmtui**.

Vérification

1. Affiche l'état des appareils et des connexions :

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

- Utilisez l'utilitaire **ping** pour vérifier que cet hôte peut envoyer des paquets à d'autres hôtes :

```
# ping host_name_or_IP_address
```

Résolution de problèmes

- Vérifiez que le câble réseau est branché sur l'hôte et sur un commutateur.
- Vérifiez si la défaillance de la liaison existe uniquement sur cet hôte ou également sur d'autres hôtes connectés au même commutateur.
- Vérifiez que le câble réseau et l'interface réseau fonctionnent comme prévu. Effectuez les étapes de diagnostic du matériel et remplacez les câbles et les cartes d'interface réseau défectueux.
- Si la configuration du disque ne correspond pas à celle du périphérique, le démarrage ou le redémarrage de NetworkManager crée une connexion en mémoire qui reflète la configuration du périphérique. Pour plus de détails et pour savoir comment éviter ce problème, voir [NetworkManager duplique une connexion après le redémarrage du service NetworkManager](#) .

Ressources supplémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

2.4. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE À L'AIDE DE NMSTATECTL

Pour configurer une connexion Ethernet à l'aide de l'API Nmstate, utilisez l'utilitaire **nmstatectl**.

Par exemple, la procédure ci-dessous crée un profil de connexion NetworkManager pour le périphérique **enp7s0** avec les paramètres suivants :

- Une adresse IPv4 statique - **192.0.2.1** avec le masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec le masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **192.0.2.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::fffe**
- Un serveur DNS IPv4 - **192.0.2.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**

L'utilitaire **nmstatectl** s'assure qu'après avoir défini la configuration, le résultat correspond au fichier de configuration. En cas d'échec, **nmstatectl** annule automatiquement les modifications pour éviter de laisser le système dans un état incorrect.

La procédure définit la configuration de l'interface au format YAML. Vous pouvez également spécifier la configuration au format JSON.

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Le paquet **nmstate** est installé.

Procédure

1. Créez un fichier YAML, par exemple `~/create-ethernet-profile.yml`, avec le contenu suivant :

```
---
interfaces:
- name: enp7s0
  type: ethernet
  state: up
  ipv4:
    enabled: true
    address:
    - ip: 192.0.2.1
      prefix-length: 24
    dhcp: false
  ipv6:
    enabled: true
    address:
    - ip: 2001:db8:1::1
      prefix-length: 64
    autoconf: false
    dhcp: false
routes:
  config:
  - destination: 0.0.0.0/0
    next-hop-address: 192.0.2.254
    next-hop-interface: enp7s0
  - destination: ::0
    next-hop-address: 2001:db8:1::fffe
    next-hop-interface: enp7s0
dns-resolver:
  config:
  search:
  - example.com
  server:
  - 192.0.2.200
  - 2001:db8:1::ffbb
```

2. Appliquer les paramètres au système :

```
# nmstatectl apply ~/create-ethernet-profile.yml
```

Vérification

1. Affiche l'état des appareils et des connexions :

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


- Affiche tous les paramètres du profil de connexion :

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

- Affiche les paramètres de connexion au format YAML :

```
# nmstatectl show enp7s0
```

Ressources supplémentaires

- [nmstatectl\(8\)](#) page de manuel
- [/usr/share/doc/nmstate/examples/](#) répertoire

2.5. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE EN UTILISANT LE RÔLE DE SYSTÈME RHEL AVEC UN NOM D'INTERFACE

Vous pouvez configurer à distance une connexion Ethernet à l'aide de **network** RHEL System Role.

Par exemple, la procédure ci-dessous crée un profil de connexion NetworkManager pour le périphérique **enp7s0** avec les paramètres suivants :

- Une adresse IPv4 statique - **192.0.2.1** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **192.0.2.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::fffe**
- Un serveur DNS IPv4 - **192.0.2.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.

- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Les nœuds gérés utilisent NetworkManager pour configurer le réseau.

Procédure

1. Créez un fichier playbook, par exemple `~/ethernet-static-IP.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
  - name: Configure an Ethernet connection with static IP
    include_role:
      name: rhel-system-roles.network

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

2. Exécutez le manuel de jeu :

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

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

2.6. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP STATIQUE EN UTILISANT LE RÔLE DE SYSTÈME RHEL AVEC UN CHEMIN D'ACCÈS AUX PÉRIPHÉRIQUES

Vous pouvez configurer à distance une connexion Ethernet à l'aide de **network** RHEL System Role.

Vous pouvez identifier le chemin d'accès à l'appareil à l'aide de la commande suivante :

■

```
# udevadm info /sys/class/net/<device_name> | grep ID_PATH=
```

Par exemple, la procédure ci-dessous crée un profil de connexion NetworkManager avec les paramètres suivants pour le périphérique qui correspond à l'expression PCI ID **0000:00:0[1-3].0**, mais pas **0000:00:02.0**:

- Une adresse IPv4 statique - **192.0.2.1** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **192.0.2.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::ffe**
- Un serveur DNS IPv4 - **192.0.2.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Les nœuds gérés utilisent NetworkManager pour configurer le réseau.

Procédure

1. Créez un fichier playbook, par exemple `~/ethernet-static-IP.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure an Ethernet connection with static IP
      include_role:
        name: rhel-system-roles.network

  vars:
    network_connections:
      - name: example
        match:
          path:
```

```

- pci-0000:00:0[1-3].0
- &!pci-0000:00:02.0
type: ethernet
autoconnect: yes
ip:
  address:
    - 192.0.2.1/24
    - 2001:db8:1::1/64
  gateway4: 192.0.2.254
  gateway6: 2001:db8:1::fffe
  dns:
    - 192.0.2.200
    - 2001:db8:1::ffbb
  dns_search:
    - example.com
state: up

```

Le paramètre **match** dans cet exemple définit qu'Ansible applique la pièce aux périphériques qui correspondent à l'ID PCI **0000:00:0[1-3].0**, mais pas à **0000:00:02.0**. Pour plus de détails sur les modificateurs spéciaux et les jokers que vous pouvez utiliser, consultez la description du paramètre **match** dans le fichier `/usr/share/ansible/roles/rhel-system-roles.network/README.md`.

2. Exécutez le manuel de jeu :

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

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

2.7. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE À L'AIDE DE NMCLI

Pour configurer une connexion Ethernet sur la ligne de commande, utilisez l'utilitaire **nmcli**. Pour les connexions avec des paramètres d'adresse IP dynamiques, NetworkManager demande les paramètres IP pour la connexion à partir d'un serveur DHCP.

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Un serveur DHCP est disponible dans le réseau.

Procédure

1. Ajouter un nouveau profil de connexion NetworkManager pour la connexion Ethernet :

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

2. Optionnellement, changer le nom d'hôte que NetworkManager envoie au serveur DHCP lors de l'utilisation du profil **Example-Connection**:

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

- Optionnellement, changer l'ID client que NetworkManager envoie à un serveur DHCP IPv4 lors de l'utilisation du profil **Example-Connection**:

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

Notez qu'il n'y a pas de paramètre **dhcp-client-id** pour IPv6. Pour créer un identifiant pour IPv6, configurez le service **dhclient**.

Vérification

- Affiche l'état des appareils et des connexions :

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

- Utilisez l'utilitaire **ping** pour vérifier que cet hôte peut envoyer des paquets à d'autres hôtes :

```
# ping host_name_or_IP_address
```

Résolution de problèmes

- Vérifiez que le câble réseau est branché sur l'hôte et sur un commutateur.
- Vérifiez si la défaillance de la liaison existe uniquement sur cet hôte ou également sur d'autres hôtes connectés au même commutateur.
- Vérifiez que le câble réseau et l'interface réseau fonctionnent comme prévu. Effectuez les étapes de diagnostic du matériel et remplacez les câbles et les cartes d'interface réseau défectueux.
- Si la configuration du disque ne correspond pas à celle du périphérique, le démarrage ou le redémarrage de NetworkManager crée une connexion en mémoire qui reflète la configuration du périphérique. Pour plus de détails et pour savoir comment éviter ce problème, voir [NetworkManager duplique une connexion après le redémarrage du service NetworkManager](#) .

Ressources supplémentaires

- **dhclient(8)** page de manuel
- **nm-settings(5)**
- **nmcli(1)** page de manuel
- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

2.8. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE À L'AIDE DE L'ÉDITEUR INTERACTIF NMCLI

Vous pouvez configurer une connexion Ethernet sur la ligne de commande en utilisant le mode interactif de l'utilitaire **nmcli**. Pour les connexions avec des paramètres d'adresse IP dynamiques, NetworkManager demande les paramètres IP pour la connexion à partir d'un serveur DHCP.

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Un serveur DHCP est disponible dans le réseau.

Procédure

1. Pour ajouter un nouveau profil de connexion NetworkManager pour la connexion Ethernet et démarrer le mode interactif, entrez :

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

2. Définir l'interface réseau :

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

3. Optionnellement, changer le nom d'hôte que NetworkManager envoie au serveur DHCP lors de l'utilisation du profil **Example-Connection**:

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

4. Optionnellement, changer l'ID client que NetworkManager envoie à un serveur DHCP IPv4 lors de l'utilisation du profil **Example-Connection**:

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

Notez qu'il n'y a pas de paramètre **dhcp-client-id** pour IPv6. Pour créer un identifiant pour IPv6, configurez le service **dhclient**.

5. Sauvegarder et activer la connexion :

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

6. Quitter le mode interactif :

```
nmcli> quit
```

Vérification

1. Affiche l'état des appareils et des connexions :

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

- Utilisez l'utilitaire **ping** pour vérifier que cet hôte peut envoyer des paquets à d'autres hôtes :

```
# ping host_name_or_IP_address
```

Résolution de problèmes

- Vérifiez que le câble réseau est branché sur l'hôte et sur un commutateur.
- Vérifiez si la défaillance de la liaison existe uniquement sur cet hôte ou également sur d'autres hôtes connectés au même commutateur.
- Vérifiez que le câble réseau et l'interface réseau fonctionnent comme prévu. Effectuez les étapes de diagnostic du matériel et remplacez les câbles et les cartes d'interface réseau défectueux.
- Si la configuration du disque ne correspond pas à celle du périphérique, le démarrage ou le redémarrage de NetworkManager crée une connexion en mémoire qui reflète la configuration du périphérique. Pour plus de détails et pour savoir comment éviter ce problème, voir [NetworkManager duplique une connexion après le redémarrage du service NetworkManager](#) .

Ressources supplémentaires

- **nm-settings(5)** page de manuel
- **nmcli(1)** page de manuel
- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

2.9. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE À L'AIDE DE NMTUI

L'application **nmtui** fournit une interface utilisateur textuelle pour NetworkManager. Vous pouvez utiliser **nmtui** pour configurer une connexion Ethernet avec une adresse IP dynamique sur un hôte sans interface graphique.



NOTE

Sur **nmtui**:

- Naviguer à l'aide des touches du curseur.
- Appuyez sur un bouton en le sélectionnant et en appuyant sur **Entrée**.
- Sélectionnez et désélectionnez les cases à cocher en utilisant l'**espace**.

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Un serveur DHCP est disponible dans le réseau.

Procédure

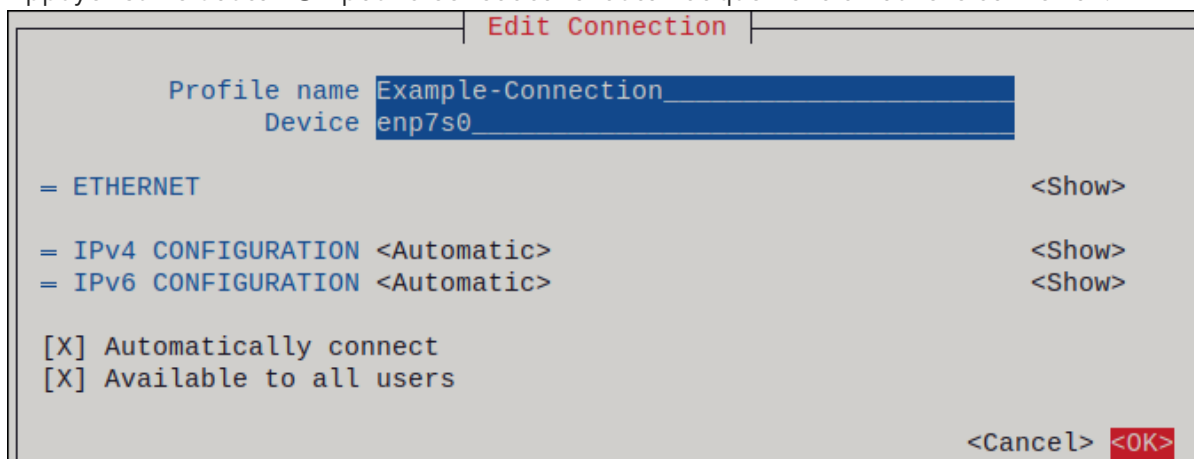
1. Si vous ne connaissez pas le nom du périphérique réseau que vous souhaitez utiliser pour la connexion, affichez les périphériques disponibles :

```
# nmcli device status
DEVICE  TYPE    STATE      CONNECTION
enp7s0  ethernet unavailable --
...
```

2. Démarrer **nmtui**:

```
# nmtui
```

3. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
4. Appuyez sur le bouton **Add**.
5. Sélectionnez **Ethernet** dans la liste des types de réseaux et appuyez sur **Entrée**.
6. Optionnel : Entrez un nom pour le profil NetworkManager à créer.
7. Saisissez le nom de l'appareil réseau dans le champ **Device**.
8. Appuyez sur le bouton **OK** pour créer et activer automatiquement la nouvelle connexion.



9. Appuyez sur le bouton **Back** pour revenir au menu principal.
10. Sélectionnez **Quit** et appuyez sur **Entrée** pour fermer l'application **nmtui**.

Vérification

1. Affiche l'état des appareils et des connexions :

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

2. Utilisez l'utilitaire **ping** pour vérifier que cet hôte peut envoyer des paquets à d'autres hôtes :

```
# ping host_name_or_IP_address
```

Résolution de problèmes

- Vérifiez que le câble réseau est branché sur l'hôte et sur un commutateur.
- Vérifiez si la défaillance de la liaison existe uniquement sur cet hôte ou également sur d'autres hôtes connectés au même commutateur.
- Vérifiez que le câble réseau et l'interface réseau fonctionnent comme prévu. Effectuez les étapes de diagnostic du matériel et remplacez les câbles et les cartes d'interface réseau défectueux.
- Si la configuration du disque ne correspond pas à celle du périphérique, le démarrage ou le redémarrage de NetworkManager crée une connexion en mémoire qui reflète la configuration du périphérique. Pour plus de détails et pour savoir comment éviter ce problème, voir [NetworkManager duplique une connexion après le redémarrage du service NetworkManager](#) .

Ressources supplémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

2.10. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE À L'AIDE DE NMSTATECTL

Pour configurer une connexion Ethernet à l'aide de l'API Nmstate, utilisez l'utilitaire **nmstatectl**. Pour les connexions avec des paramètres d'adresse IP dynamiques, NetworkManager demande les paramètres IP pour la connexion à partir d'un serveur DHCP.

L'utilitaire **nmstatectl** s'assure qu'après avoir défini la configuration, le résultat correspond au fichier de configuration. En cas d'échec, **nmstatectl** annule automatiquement les modifications pour éviter de laisser le système dans un état incorrect.

La procédure définit la configuration de l'interface au format YAML. Vous pouvez également spécifier la configuration au format JSON.

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Un serveur DHCP est disponible dans le réseau.
- Le paquet **nmstate** est installé.

Procédure

1. Créez un fichier YAML, par exemple `~/create-ethernet-profile.yml`, avec le contenu suivant :

```
---
interfaces:
- name: enp7s0
  type: ethernet
  state: up
  ipv4:
    enabled: true
    auto-dns: true
    auto-gateway: true
    auto-routes: true
```

```

dhcp: true
ipv6:
  enabled: true
  auto-dns: true
  auto-gateway: true
  auto-routes: true
  autoconf: true
  dhcp: true

```

2. Appliquer les paramètres au système :

```
# nmstatectl apply ~/create-ethernet-profile.yml
```

Vérification

1. Affiche l'état des appareils et des connexions :

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

2. Affiche tous les paramètres du profil de connexion :

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

3. Affiche les paramètres de connexion au format YAML :

```
# nmstatectl show enp7s0
```

Ressources supplémentaires

- [nmstatectl\(8\)](#) page de manuel
- [/usr/share/doc/nmstate/examples/](#) répertoire

2.11. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE EN UTILISANT LE RÔLE DE SYSTÈME RHEL AVEC UN NOM D'INTERFACE

Vous pouvez configurer à distance une connexion Ethernet à l'aide du rôle de système RHEL **network**. Pour les connexions avec des paramètres d'adresse IP dynamiques, NetworkManager demande les paramètres IP pour la connexion à partir d'un serveur DHCP.

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- Vous avez préparé le nœud de contrôle et les nœuds gérés
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Un serveur DHCP est disponible dans le réseau
- Les nœuds gérés utilisent NetworkManager pour configurer le réseau.

Procédure

1. Créez un fichier playbook, par exemple `~/ethernet-dynamic-IP.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
  - name: Configure an Ethernet connection with dynamic IP
    include_role:
      name: rhel-system-roles.network

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

2. Exécutez le manuel de jeu :

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

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

2.12. CONFIGURATION D'UNE CONNEXION ETHERNET AVEC UNE ADRESSE IP DYNAMIQUE EN UTILISANT LE RÔLE DE SYSTÈME RHEL AVEC UN CHEMIN D'ACCÈS DE PÉRIPHÉRIQUE

Vous pouvez configurer à distance une connexion Ethernet à l'aide du rôle de système RHEL **network**. Pour les connexions avec des paramètres d'adresse IP dynamiques, NetworkManager demande les paramètres IP pour la connexion à partir d'un serveur DHCP.

Vous pouvez identifier le chemin d'accès à l'appareil à l'aide de la commande suivante :

```
# udevadm info /sys/class/net/<device_name> | grep ID_PATH=
```

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- Vous avez préparé le nœud de contrôle et les nœuds gérés
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- Un serveur DHCP est disponible dans le réseau.
- Les hôtes gérés utilisent NetworkManager pour configurer le réseau.

Procédure

1. Créez un fichier playbook, par exemple `~/ethernet-dynamic-IP.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure an Ethernet connection with dynamic IP
      include_role:
        name: rhel-system-roles.network

  vars:
    network_connections:
      - name: example
        match:
          path:
            - pci-0000:00:0[1-3].0
            - &!pci-0000:00:02.0
          type: ethernet
          autoconnect: yes
          ip:
            dhcp4: yes
            auto6: yes
          state: up
```

Le paramètre **match** dans cet exemple définit qu'Ansible applique la pièce aux périphériques qui

correspondent à l'ID PCI **0000:00:0[1-3].0**, mais pas à **0000:00:02.0**. Pour plus de détails sur les modificateurs spéciaux et les jokers que vous pouvez utiliser, consultez la description du paramètre **match** dans le fichier **/usr/share/ansible/roles/rhel-system-roles.network/README.md**.

2. Exécutez le manuel de jeu :

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

Ressources supplémentaires

- **/usr/share/ansible/roles/rhel-system-roles.network/README.md** fichier

2.13. CONFIGURATION D'UNE CONNEXION ETHERNET À L'AIDE DU CENTRE DE CONTRÔLE


Les connexions Ethernet sont les types de connexions les plus fréquemment utilisés dans les machines physiques ou virtuelles. Si vous utilisez Red Hat Enterprise Linux avec une interface graphique, vous pouvez configurer ce type de connexion dans la fenêtre GNOME **control-center**.

Notez que **control-center** ne prend pas en charge autant d'options de configuration que l'application **nm-connection-editor** ou l'utilitaire **nmcli**.

Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- GNOME est installé.

Procédure

1. Appuyez sur la touche **Super**, entrez **Settings** et appuyez sur **Entrée**.
2. Sélectionnez **Network** dans le menu de gauche.
3. Cliquez sur le bouton  à côté de l'entrée **Wired** pour créer un nouveau profil.
4. Facultatif : Définissez un nom pour la connexion dans l'onglet **Identity**.
5. Dans l'onglet **IPv4**, configurez les paramètres IPv4. Par exemple, sélectionnez la méthode **Manual**, définissez une adresse IPv4 statique, un masque de réseau, une passerelle par défaut et un serveur DNS :

New Profile

Identity **IPv4** IPv6 Security

IPv4 Method

Automatic (DHCP) Link-Local Only

Manual Disable

Shared to other computers

Addresses

Address	Netmask	Gateway
192.0.2.1	24	192.0.2.254

DNS Automatic

192.0.2.1

Separate IP addresses with commas

6. Dans l'onglet **IPv6**, configurez les paramètres IPv6. Par exemple, sélectionnez la méthode **Manual**, définissez une adresse IPv6 statique, un masque de réseau, une passerelle par défaut et un serveur DNS :

New Profile

Identity IPv4 **IPv6** Security

IPv6 Method

Automatic Automatic, DHCP only

Link-Local Only **Manual**

Disable Shared to other computers

Addresses

Address	Prefix	Gateway
2001:db8:1::1	64	2001:db8:1::fff3

DNS Automatic

2001:db8:1::fffd

Separate IP addresses with commas

7. Cliquez sur le bouton **Ajouter** pour enregistrer la connexion. Le site GNOME **control-center** active automatiquement la connexion.

Vérification

1. Affiche l'état des appareils et des connexions :

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

2. Utilisez l'utilitaire **ping** pour vérifier que cet hôte peut envoyer des paquets à d'autres hôtes :

```
# ping host_name_or_IP_address
```

Étapes de dépannage

- Vérifiez que le câble réseau est branché sur l'hôte et sur un commutateur.
- Vérifiez si la défaillance de la liaison existe uniquement sur cet hôte ou également sur d'autres hôtes connectés au même commutateur.
- Vérifiez que le câble réseau et l'interface réseau fonctionnent comme prévu. Effectuez les étapes de diagnostic du matériel et remplacez les câbles et les cartes d'interface réseau défectueux.
- Si la configuration du disque ne correspond pas à celle du périphérique, le démarrage ou le redémarrage de NetworkManager crée une connexion en mémoire qui reflète la configuration du périphérique. Pour plus de détails et pour savoir comment éviter ce problème, voir [NetworkManager duplique une connexion après le redémarrage du service NetworkManager](#) .

Ressources complémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

2.14. CONFIGURATION D'UNE CONNEXION ETHERNET À L'AIDE DE NM-CONNECTION-EDITOR

Les connexions Ethernet sont les types de connexion les plus fréquemment utilisés dans les serveurs physiques ou virtuels. Si vous utilisez Red Hat Enterprise Linux avec une interface graphique, vous pouvez configurer ce type de connexion à l'aide de l'application **nm-connection-editor**.


Conditions préalables

- Un périphérique Ethernet physique ou virtuel existe dans la configuration du serveur.
- GNOME est installé.

Procédure

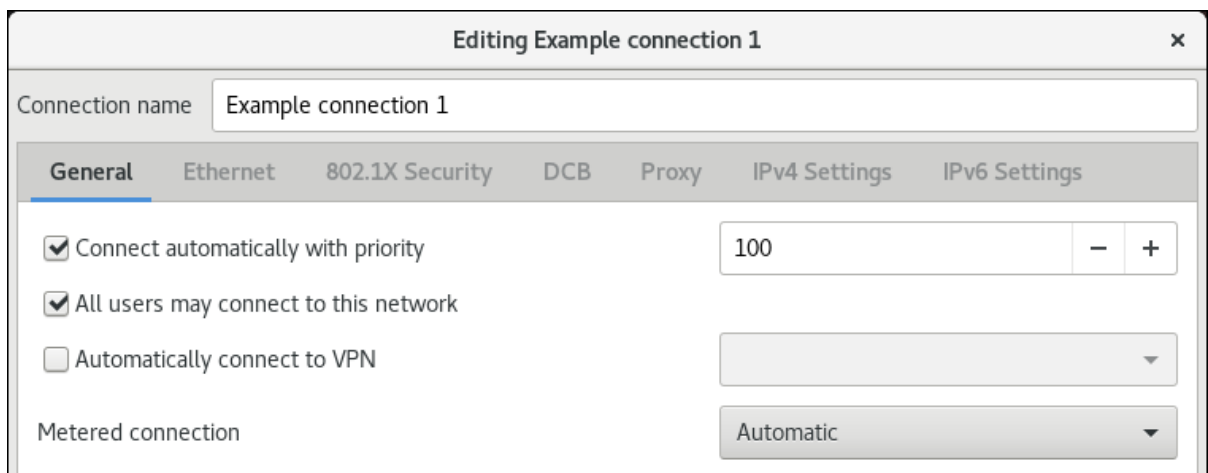
1. Ouvrez un terminal et entrez :

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

2. Cliquez sur le bouton  pour ajouter une nouvelle connexion.

3. Sélectionnez le type de connexion **Ethernet** et cliquez sur **Créer**.
4. Dans l'onglet **General**:
 - a. Pour activer automatiquement cette connexion lorsque le système démarre ou lorsque vous redémarrez le service **NetworkManager**:
 - i. Sélectionnez **Connect automatically with priority**.
 - ii. Optionnel : Modifiez la valeur de la priorité à côté de **Connect automatically with priority**.

Si plusieurs profils de connexion existent pour le même appareil, NetworkManager n'active qu'un seul profil. Par défaut, NetworkManager active le dernier profil utilisé dont l'auto-connexion est activée. Cependant, si vous définissez des valeurs de priorité dans les profils, NetworkManager active le profil ayant la priorité la plus élevée.
 - b. Décochez la case **All users may connect to this network** si le profil ne doit être accessible qu'à l'utilisateur qui a créé le profil de connexion.



5. Dans l'onglet **Ethernet**, sélectionnez un appareil et, éventuellement, d'autres paramètres liés à l'Ethernet.

Editing Ethernet connection 1 x

Connection name:

General **Ethernet** 802.1X Security DCB Proxy IPv4 Settings IPv6 Settings

Device:

Cloned MAC address:

MTU: bytes

Wake on LAN: Default Phy Unicast Multicast
 Ignore Broadcast Arp Magic

Wake on LAN password:

Link negotiation:

Speed:

Duplex:

6. Dans l'onglet **IPv4 Settings**, configurez les paramètres IPv4. Par exemple, définissez une adresse IPv4 statique, un masque de réseau, une passerelle par défaut et un serveur DNS :

Method:

Addresses

Address	Netmask	Gateway
192.0.2.1	24	192.0.2.254

DNS servers:

7. Dans l'onglet **IPv6 Settings**, configurez les paramètres IPv6. Par exemple, définissez une adresse IPv6 statique, un masque de réseau, une passerelle par défaut et un serveur DNS :

Method:

Addresses

Address	Prefix	Gateway
2001:db8:1::1	64	2001:db8:1::fff3

DNS servers:

8. Sauvegarder la connexion.

9. Fermer **nm-connection-editor**.

Vérification

1. Affiche l'état des appareils et des connexions :

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

2. Utilisez l'utilitaire **ping** pour vérifier que cet hôte peut envoyer des paquets à d'autres hôtes :

```
# ping host_name_or_IP_address
```

Ressources complémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

2.15. CHANGEMENT DU CLIENT DHCP DU NETWORKMANAGER

Par défaut, NetworkManager utilise son client DHCP interne. Cependant, si vous avez besoin d'un client DHCP avec des fonctionnalités que le client intégré ne fournit pas, vous pouvez configurer NetworkManager pour qu'il utilise **dhclient**.

Notez que RHEL ne fournit pas **dhcpcd** et que, par conséquent, NetworkManager ne peut pas utiliser ce client.

Procédure

1. Créez le fichier **/etc/NetworkManager/conf.d/dhcp-client.conf** avec le contenu suivant :

```
[main]
dhcp=dhclient
```

Vous pouvez régler le paramètre **dhcp** sur **internal** (valeur par défaut) ou **dhclient**.

2. Si vous attribuez la valeur **dhclient** au paramètre **dhcp**, installez le paquet **dhcp-client**:

```
# dnf install dhcp-client
```

3. Redémarrer NetworkManager :

```
# systemctl restart NetworkManager
```

Notez que le redémarrage interrompt temporairement toutes les connexions réseau.

Vérification

- Recherchez dans le fichier journal **/var/log/messages** une entrée similaire à la suivante :

```
Apr 26 09:54:19 server NetworkManager[27748]: <info> [1650959659.8483] dhcp-init: Using DHCP client 'dhclient'
```

Cette entrée de journal confirme que NetworkManager utilise **dhclient** comme client DHCP.

Ressources supplémentaires

- **NetworkManager.conf(5)** page de manuel

2.16. CONFIGURATION DU COMPORTEMENT DHCP D'UNE CONNEXION NETWORKMANAGER

Un client DHCP (Dynamic Host Configuration Protocol) demande une adresse IP dynamique et les informations de configuration correspondantes à un serveur DHCP chaque fois qu'un client se connecte au réseau.

Lorsque vous avez configuré une connexion pour récupérer une adresse IP à partir d'un serveur DHCP, le NetworkManager demande une adresse IP à un serveur DHCP. Par défaut, le client attend 45 secondes que cette demande soit terminée. Lorsqu'une connexion **DHCP** est lancée, un client dhcp demande une adresse IP à un serveur **DHCP**.

Conditions préalables

- Une connexion utilisant DHCP est configurée sur l'hôte.

Procédure

1. Définissez les propriétés **ipv4.dhcp-timeout** et **ipv6.dhcp-timeout**. Par exemple, pour définir les deux options sur **30** seconds, entrez :

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

Vous pouvez également définir les paramètres sur **infinity** pour configurer NetworkManager de manière à ce qu'il n'arrête pas d'essayer de demander et de renouveler une adresse IP jusqu'à ce qu'il y parvienne.

2. Optionnel : Configurer le comportement si NetworkManager ne reçoit pas d'adresse IPv4 avant l'expiration du délai :

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

Si l'option **ipv4.may-fail** est réglée sur :

- **yes** l'état de la connexion dépend de la configuration IPv6 :
 - Si la configuration IPv6 est activée et réussie, NetworkManager active la connexion IPv6 et n'essaie plus d'activer la connexion IPv4.
 - Si la configuration IPv6 est désactivée ou non configurée, la connexion échoue.
- **no** la connexion est désactivée. Dans ce cas, la connexion est désactivée :
 - Si la propriété **autoconnect** de la connexion est activée, NetworkManager tente d'activer la connexion autant de fois que défini dans la propriété **autoconnect-retries**. La valeur par défaut est **4**.

- Si la connexion ne peut toujours pas acquérir une adresse DHCP, l'auto-activation échoue. Notez qu'après 5 minutes, le processus d'auto-connexion recommence pour acquérir une adresse IP à partir du serveur DHCP.
3. Optionnel : Configurer le comportement si NetworkManager ne reçoit pas d'adresse IPv6 avant l'expiration du délai :

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

Ressources supplémentaires

- **nm-settings(5)** page de manuel

2.17. CONFIGURATION DE PLUSIEURS INTERFACES ETHERNET À L'AIDE D'UN SEUL PROFIL DE CONNEXION PAR NOM D'INTERFACE

Dans la plupart des cas, un profil de connexion contient les paramètres d'un seul périphérique réseau. Cependant, NetworkManager prend également en charge les caractères génériques lorsque vous définissez le nom de l'interface dans les profils de connexion. Si un hôte passe d'un réseau Ethernet à l'autre avec une attribution dynamique d'adresse IP, vous pouvez utiliser cette fonctionnalité pour créer un profil de connexion unique que vous pouvez utiliser pour plusieurs interfaces Ethernet.

Conditions préalables

- Plusieurs périphériques Ethernet physiques ou virtuels existent dans la configuration du serveur.
- Un serveur DHCP est disponible dans le réseau.
- Aucun profil de connexion n'existe sur l'hôte.

Procédure

1. Ajouter un profil de connexion qui s'applique à tous les noms d'interface commençant par **enp**:

```
# nmcli connection add con-name Example connection.multi-connect multiple
match.interface-name enp* type ethernet
```

Vérification

1. Affiche tous les paramètres du profil de connexion unique :

```
# nmcli connection show Example
connection.id:          Example
...
connection.multi-connect: 3 (multiple)
match.interface-name:   enp*
...
```

3 indique le nombre d'interfaces actives sur le profil de connexion au même moment, et non le nombre d'interfaces réseau dans le profil de connexion. Le profil de connexion utilise tous les appareils qui correspondent au modèle du paramètre **match.interface-name** et, par conséquent, les profils de connexion ont le même identifiant universel unique (UUID).

2. Affiche l'état des connexions :

```
# nmcli connection show
NAME                UUID                TYPE  DEVICE
...
Example 6f22402e-c0cc-49cf-b702-eaf0cd5ea7d1 ethernet enp7s0
Example 6f22402e-c0cc-49cf-b702-eaf0cd5ea7d1 ethernet enp8s0
Example 6f22402e-c0cc-49cf-b702-eaf0cd5ea7d1 ethernet enp9s0
```

Ressources supplémentaires

- **nmcli(1)** page de manuel
- **nm-settings(5)** page de manuel

2.18. CONFIGURATION D'UN PROFIL DE CONNEXION UNIQUE POUR PLUSIEURS INTERFACES ETHERNET À L'AIDE D'ID PCI

L'ID PCI est un identifiant unique des appareils connectés au système. Le profil de connexion ajoute plusieurs périphériques en faisant correspondre les interfaces sur la base d'une liste d'ID PCI. Vous pouvez utiliser cette procédure pour connecter plusieurs ID PCI de périphériques à un seul profil de connexion.

Conditions préalables

- Plusieurs périphériques Ethernet physiques ou virtuels existent dans la configuration du serveur.
- Un serveur DHCP est disponible dans le réseau.
- Aucun profil de connexion n'existe sur l'hôte.

Procédure

1. Identifiez le chemin d'accès au périphérique. Par exemple, pour afficher les chemins d'accès aux périphériques de toutes les interfaces commençant par **enp**, entrez :

```
# udevadm info /sys/class/net/enp* | grep ID_PATH=
...
E: ID_PATH=pci-0000:07:00.0
E: ID_PATH=pci-0000:08:00.0
```

2. Ajouter un profil de connexion qui s'applique à tous les ID PCI correspondant à l'expression **0000:00:0[7-8].0**:

```
# nmcli connection add type ethernet connection.multi-connect multiple match.path
"pci-0000:07:00.0 pci-0000:08:00.0" con-name Example
```

Vérification

1. Affiche l'état de la connexion :

```
# nmcli connection show
NAME  UUID  TYPE  DEVICE
```

```
Example 9cee0958-512f-4203-9d3d-b57af1d88466 ethernet enp7s0
Example 9cee0958-512f-4203-9d3d-b57af1d88466 ethernet enp8s0
...
```

2. Pour afficher tous les paramètres du profil de connexion :

```
# nmcli connection show Example
connection.id:      Example
...
connection.multi-connect: 3 (multiple)
match.path:         pci-0000:07:00.0,pci-0000:08:00.0
...
```

Ce profil de connexion utilise tous les appareils dont l'ID PCI correspond au modèle du paramètre **match.path** et, par conséquent, les profils de connexion ont le même identifiant universel unique (UUID).

Ressources supplémentaires

- **nmcli(1)** page de manuel
- **nm-settings(5)** page de manuel

CHAPITRE 3. CONFIGURATION DE LA LIAISON RÉSEAU

Une liaison réseau est une méthode permettant de combiner ou d'agrèger des interfaces réseau physiques et virtuelles afin de fournir une interface logique avec un débit plus élevé ou une redondance. Dans un lien, le noyau gère exclusivement toutes les opérations. Vous pouvez créer des liens sur différents types de périphériques, tels que des périphériques Ethernet ou des VLAN.

Red Hat Enterprise Linux offre aux administrateurs différentes options pour configurer les périphériques d'équipe. Par exemple :

- Utilisez **nmcli** pour configurer les connexions de liaison à l'aide de la ligne de commande.
- Utilisez la console web RHEL pour configurer les connexions bond à l'aide d'un navigateur web.
- Utilisez **nmtui** pour configurer les connexions de liaison dans une interface utilisateur textuelle.
- Utilisez l'application **nm-connection-editor** pour configurer les connexions de liaison dans une interface graphique.
- Utilisez **nmstatectl** pour configurer les connexions de liaison par l'intermédiaire de l'API Nmstate.
- Utilisez les rôles système RHEL pour automatiser la configuration des liens sur un ou plusieurs hôtes.

3.1. COMPRENDRE LA LIAISON RÉSEAU

La liaison réseau est une méthode permettant de combiner ou d'agrèger des interfaces réseau afin de fournir une interface logique avec un débit plus élevé ou une redondance.

Les modes **active-backup**, **balance-tilb** et **balance-alb** ne nécessitent aucune configuration spécifique du commutateur réseau. Cependant, d'autres modes de liaison nécessitent de configurer le commutateur pour agrèger les liens. Par exemple, les commutateurs Cisco nécessitent **EtherChannel** pour les modes 0, 2 et 3, mais pour le mode 4, le protocole LACP (Link Aggregation Control Protocol) et **EtherChannel** sont nécessaires. Pour plus de détails, consultez la documentation de votre commutateur.



IMPORTANT

Certaines fonctions de liaison réseau, telles que le mécanisme de basculement, ne prennent pas en charge les connexions directes par câble sans commutateur réseau. Pour plus de détails, voir la section [Le bonding est-il pris en charge avec une connexion directe utilisant des câbles croisés ?](#) Solution KCS.

3.2. COMPRENDRE LE COMPORTEMENT PAR DÉFAUT DES INTERFACES DES CONTRÔLEURS ET DES PORTS

Tenez compte du comportement par défaut suivant lorsque vous gérez ou dépannez des interfaces de port d'équipe ou de liaison à l'aide du service **NetworkManager**:

- Le démarrage de l'interface du contrôleur ne démarre pas automatiquement les interfaces des ports.
- Le démarrage d'une interface de port entraîne toujours le démarrage de l'interface du contrôleur.

- L'arrêt de l'interface du contrôleur entraîne également l'arrêt de l'interface du port.
- Un contrôleur sans ports peut démarrer des connexions IP statiques.
- Un contrôleur sans ports attend les ports lors du démarrage des connexions DHCP.
- Un contrôleur avec une connexion DHCP en attente de ports se termine lorsque vous ajoutez un port avec une porteuse.
- Un contrôleur avec une connexion DHCP en attente de ports continue d'attendre lorsque vous ajoutez un port sans support.

3.3. CONFIGURATION DU COMMUTATEUR EN AMONT EN FONCTION DES MODES DE LIAISON

Appliquez les paramètres suivants au commutateur en amont en fonction du mode de liaison :

Mode de liaison	Configuration sur le commutateur
0 - balance-rr	Nécessite un canal d'échange statique activé (non négocié LACP)
1 - active-backup	Nécessite des ports autonomes
2 - balance-xor	Nécessite un canal d'échange statique activé (non négocié LACP)
3 - broadcast	Nécessite un canal d'échange statique activé (non négocié LACP)
4 - 802.3ad	Nécessite l'activation du canal Ethernet négocié LACP
5 - balance-tlb	Nécessite des ports autonomes
6 - balance-alb	Nécessite des ports autonomes

Pour configurer ces paramètres sur votre commutateur, consultez la documentation du commutateur.

3.4. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE NMCLI

Pour configurer une liaison réseau sur la ligne de commande, utilisez l'utilitaire **nmcli**.

Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports de la liaison, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.

- Pour utiliser des périphériques team, bridge ou VLAN comme ports de la liaison, vous pouvez soit créer ces périphériques lors de la création de la liaison, soit les créer à l'avance comme décrit dans la section :
 - [Configuration d'une équipe réseau à l'aide de nmcli](#)
 - [Configuration d'un pont réseau à l'aide de nmcli](#)
 - [Configuration du marquage des VLAN à l'aide de nmcli](#)

Procédure

1. Créer une interface de liaison :

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

Cette commande crée un lien nommé **bond0** qui utilise le mode **active-backup**.

Pour définir en plus un intervalle de surveillance de l'interface indépendante de média (MI), ajoutez l'option **miimon=interval** à la propriété **bond.options**, par exemple :

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

2. Affichez les interfaces réseau et notez les noms des interfaces que vous prévoyez d'ajouter au lien :

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

Dans cet exemple :

- **enp7s0** et **enp8s0** ne sont pas configurés. Pour utiliser ces dispositifs comme ports, ajoutez des profils de connexion à l'étape suivante.
 - **bridge0** et **bridge1** ont des profils de connexion existants. Pour utiliser ces dispositifs comme ports, modifiez leurs profils à l'étape suivante.
3. Attribuer des interfaces à la liaison :
 - a. Si les interfaces que vous souhaitez attribuer à la liaison ne sont pas configurées, créez de nouveaux profils de connexion pour elles :

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

Ces commandes créent des profils pour **enp7s0** et **enp8s0** et les ajoutent à la connexion **bond0**.

b. Pour affecter un profil de connexion existant à la liaison :

i. Réglez le paramètre **master** de ces connexions sur **bond0**:

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

Ces commandes affectent les profils de connexion existants nommés **bridge0** et **bridge1** à la connexion **bond0**.

ii. Réactiver les connexions :

```
# nmcli connection up bridge0
# nmcli connection up bridge1
```

4. Configurez les paramètres IPv4 :

- Pour utiliser ce périphérique de liaison comme port d'autres périphériques, entrez :

```
# nmcli connection modify bond0 ipv4.method disabled
```

- Pour utiliser le DHCP, aucune action n'est nécessaire.

- Pour définir une adresse IPv4 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **bond0**, entrez :

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

5. Configurez les paramètres IPv6 :

- Pour utiliser ce périphérique de liaison comme port d'autres périphériques, entrez :

```
# nmcli connection modify bond0 ipv6.method disabled
```

- Pour utiliser le DHCP, aucune action n'est nécessaire.

- Pour définir une adresse IPv6 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **bond0**, entrez :

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

6. Facultatif : si vous souhaitez définir des paramètres sur les ports de liaison, utilisez la commande suivante :

```
# nmcli connection modify bond0-port1 bond-port.<parameter> <value>
```

7. Activer la connexion :

-

```
# nmcli connection up bond0
```

8. Vérifiez que les ports sont connectés et que la colonne **CONNECTION** affiche le nom de connexion du port :

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

Lorsque vous activez un port de la connexion, NetworkManager active également le lien, mais pas les autres ports. Vous pouvez configurer Red Hat Enterprise Linux pour qu'il active automatiquement tous les ports lorsque la liaison est activée :

- a. Active le paramètre **connection.autoconnect-slaves** de la connexion de la liaison :

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

- b. Réactiver le pont :

```
# nmcli connection up bond0
```

Vérification

1. Retirez temporairement le câble réseau de l'hôte.
Il convient de noter qu'il n'existe aucune méthode permettant de tester correctement les événements de défaillance de liaison à l'aide d'utilitaires logiciels. Les outils qui désactivent les connexions, tels que **nmcli**, ne montrent que la capacité du pilote de liaison à gérer les changements de configuration des ports et non les événements réels de défaillance de la liaison.
2. Affiche l'état de la liaison :

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

Ressources supplémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)
- [Documentation sur la liaison réseau](#)

3.5. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE LA CONSOLE WEB RHEL

Utilisez la console web RHEL pour configurer une liaison réseau si vous préférez gérer les paramètres réseau à l'aide d'une interface basée sur un navigateur web.

Conditions préalables

- Vous êtes connecté à la console web RHEL.
- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.

- Pour utiliser des périphériques Ethernet comme membres du lien, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.
- Pour utiliser des périphériques d'équipe, de pont ou de VLAN en tant que membres du lien, créez-les à l'avance comme décrit dans la section :
 - [Configuration d'une équipe réseau à l'aide de la console web RHEL](#)
 - [Configuration d'un pont réseau à l'aide de la console web RHEL](#)
 - [Configuration du marquage VLAN à l'aide de la console web RHEL](#)

Procédure

1. Sélectionnez l'onglet **Networking** dans le menu de navigation situé à gauche de l'écran.
2. Cliquez sur **Add bond** dans la section **Interfaces**.
3. Saisissez le nom du dispositif de liaison que vous souhaitez créer.
4. Sélectionnez les interfaces qui doivent être membres de la liaison.
5. Sélectionnez le mode de l'obligation.
Si vous sélectionnez **Active backup**, la console web affiche le champ supplémentaire **Primary** dans lequel vous pouvez sélectionner l'appareil actif préféré.
6. Définissez le mode de surveillance des liens. Par exemple, lorsque vous utilisez le mode **Adaptive load balancing**, réglez-le sur **ARP**.
7. En option : Ajustez les paramètres de l'intervalle de surveillance, du délai d'établissement de la liaison et du délai de rétablissement de la liaison. En général, vous ne modifiez les paramètres par défaut qu'à des fins de dépannage.

Bond settings

Name

Interfaces enp7s0
 enp8s0

MAC

Mode

Primary


Link monitoring

Monitoring interval

Link up delay

Link down delay

8. Cliquez sur **Appliquer**.
9. Par défaut, la liaison utilise une adresse IP dynamique. Si vous souhaitez définir une adresse IP statique :
 - a. Cliquez sur le nom de l'obligation dans la section **Interfaces**.
 - b. Cliquez sur **Edit** en regard du protocole que vous souhaitez configurer.
 - c. Sélectionnez **Manual** à côté de **Addresses**, et entrez l'adresse IP, le préfixe et la passerelle par défaut.
 - d. Dans la section **DNS**, cliquez sur le bouton et entrez l'adresse IP du serveur DNS. Répétez cette étape pour définir plusieurs serveurs DNS.

- e. Dans la section **DNS search domains**, cliquez sur le bouton  et entrez le domaine de recherche.
- f. Si l'interface nécessite des routes statiques, configurez-les dans la section **Routes**.

IPv4 settings ✕

Addresses Manual ▾ +

Address	Prefix length or netmask	Gateway	
<input type="text" value="192.0.2.1"/>	<input type="text" value="24"/>	<input type="text" value="192.0.2.254"/>	-

DNS Automatic +

Server -

DNS search domains Automatic +

Search domain -

Routes Automatic +

Apply Cancel

- g. Cliquez sur **Appliquer**

Vérification

1. Sélectionnez l'onglet **Networking** dans la navigation sur le côté gauche de l'écran, et vérifiez s'il y a du trafic entrant et sortant sur l'interface :

Interfaces Add bond Add team Add bridge Add VLAN 			
Name	IP address	Sending	Receiving
bond0	192.0.2.1/24	1.11 Mbps	61.2 Mbps

2. Retirez temporairement le câble réseau de l'hôte.
 Notez qu'il n'existe aucune méthode permettant de tester correctement les événements de défaillance de liaison à l'aide d'utilitaires logiciels. Les outils qui désactivent les connexions, comme la console web, ne montrent que la capacité du pilote de liaison à gérer les changements de configuration des membres et non les événements réels de défaillance de la liaison.
3. Affiche l'état de la liaison :

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

3.6. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE NMTUI

L'application **nmtui** fournit une interface utilisateur textuelle pour NetworkManager. Vous pouvez utiliser **nmtui** pour configurer une liaison réseau sur un hôte dépourvu d'interface graphique.



NOTE

Sur **nmtui**:

- Naviguer à l'aide des touches du curseur.
- Appuyez sur un bouton en le sélectionnant et en appuyant sur **Entrée**.
- Sélectionnez et désélectionnez les cases à cocher en utilisant l'**espace**.

Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports de la liaison, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.

Procédure

1. Si vous ne connaissez pas les noms des périphériques réseau sur lesquels vous souhaitez configurer une liaison réseau, affichez les périphériques disponibles :

```
# nmcli device status
DEVICE  TYPE    STATE      CONNECTION
enp7s0  ethernet unavailable --
enp8s0  ethernet unavailable --
...
```

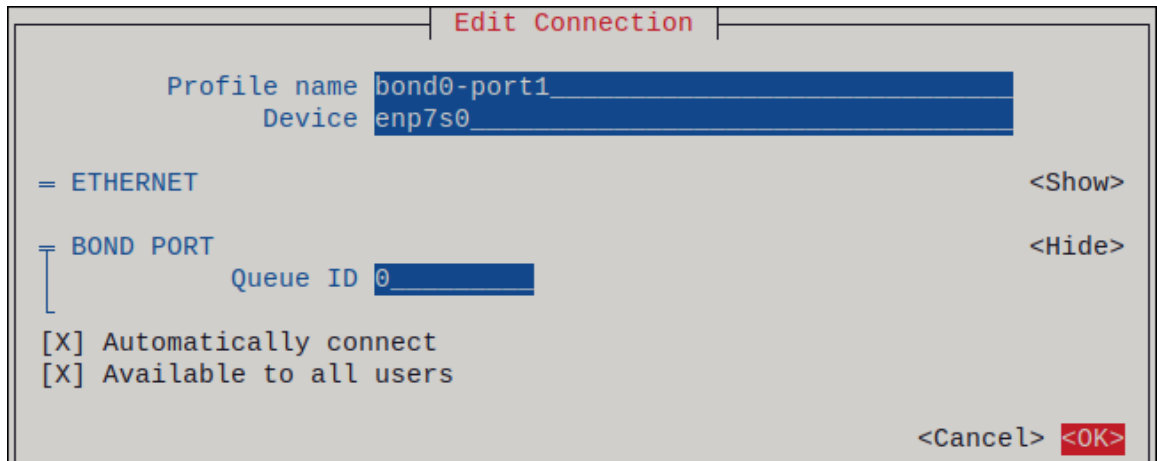
2. Démarrer **nmtui**:

```
# nmtui
```

3. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
4. Appuyez sur le bouton **Add**.
5. Sélectionnez **Bond** dans la liste des types de réseaux et appuyez sur **Entrée**.
6. Optionnel : Entrez un nom pour le profil NetworkManager à créer.
7. Saisissez le nom du dispositif de liaison à créer dans le champ **Device**.
8. Ajouter des ports à la liaison à créer :
 - a. Appuyez sur le bouton **Add** à côté de la liste **Slaves**.

- b. Sélectionnez le type d'interface que vous souhaitez ajouter en tant que port à la liaison, par exemple, **Ethernet**.
- c. Facultatif : Entrez un nom pour le profil NetworkManager à créer pour ce port de liaison.
- d. Saisissez le nom de l'appareil du port dans le champ **Device**.
- e. Appuyez sur le bouton **OK** pour revenir à la fenêtre des paramètres de liaison.

Figure 3.1. Ajout d'un périphérique Ethernet en tant que port à une liaison



- f. Répétez ces étapes pour ajouter d'autres ports à la liaison.
9. Définissez le mode de liaison. En fonction de la valeur que vous avez définie, **nmtui** affiche des champs supplémentaires pour les paramètres liés au mode sélectionné.
 10. En fonction de votre environnement, configurez les paramètres de l'adresse IP dans les zones **IPv4 configuration** et **IPv6 configuration**. Pour ce faire, appuyez sur la touche **Automatic** et sélectionnez :
 - **Disabled** si la liaison ne nécessite pas d'adresse IP.
 - **Automatic** si un serveur DHCP attribue dynamiquement une adresse IP à la liaison.
 - **Manual** si le réseau nécessite des paramètres d'adresse IP statiques. Dans ce cas, vous devez remplir d'autres champs :
 - i. Appuyez sur le bouton **Show** en regard du protocole que vous souhaitez configurer pour afficher des champs supplémentaires.
 - ii. Appuyez sur le bouton **Add** à côté de **Addresses**, et entrez l'adresse IP et le masque de sous-réseau au format CIDR (Classless Inter-Domain Routing).
Si vous ne spécifiez pas de masque de sous-réseau, NetworkManager définit un masque de sous-réseau **/32** pour les adresses IPv4 et **/64** pour les adresses IPv6.
 - iii. Saisissez l'adresse de la passerelle par défaut.
 - iv. Appuyez sur la touche **Add** à côté de **DNS servers**, et entrez l'adresse du serveur DNS.
 - v. Appuyez sur la touche **Add** à côté de **Search domains**, et entrez le domaine de recherche DNS.

Figure 3.2. Exemple d'une connexion de liaison avec des paramètres d'adresse IP statiques

Edit Connection

Profile name

Device

BOND <Hide>

Slaves

↑

↓

Mode

Primary

Link monitoring

Monitoring frequency ms

Link up delay ms

Link down delay ms

Cloned MAC address

IPv4 CONFIGURATION <Hide>

Addresses

Gateway

DNS servers

Search domains

Routing (No custom routes)

Never use this network for default route

Ignore automatically obtained routes

Ignore automatically obtained DNS parameters

Require IPv4 addressing for this connection

IPv6 CONFIGURATION <Hide>

Addresses

Gateway

DNS servers

Search domains

Routing (No custom routes)

Never use this network for default route

Ignore automatically obtained routes

Ignore automatically obtained DNS parameters

Require IPv6 addressing for this connection

Automatically connect

Available to all users

11. Appuyez sur le bouton **OK** pour créer et activer automatiquement la nouvelle connexion.

- Appuyez sur le bouton **Back** pour revenir au menu principal.
- Sélectionnez **Quit** et appuyez sur **Entrée** pour fermer l'application **nmtui**.

Vérification

- Retirez temporairement le câble réseau de l'hôte.
Il convient de noter qu'il n'existe aucune méthode permettant de tester correctement les événements de défaillance de liaison à l'aide d'utilitaires logiciels. Les outils qui désactivent les connexions, tels que **nmcli**, ne montrent que la capacité du pilote de liaison à gérer les changements de configuration des ports et non les événements réels de défaillance de la liaison.
- Affiche l'état de la liaison :

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

3.7. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE NM-CONNECTION-EDITOR

Si vous utilisez Red Hat Enterprise Linux avec une interface graphique, vous pouvez configurer les liaisons réseau à l'aide de l'application **nm-connection-editor**.

Notez que **nm-connection-editor** ne peut ajouter que de nouveaux ports à un lien. Pour utiliser un profil de connexion existant en tant que port, créez le lien à l'aide de l'utilitaire **nmcli**, comme décrit dans la section [Configuration d'un lien réseau à l'aide de nmcli](#) .


Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports de la liaison, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.
- Pour utiliser des périphériques team, bond ou VLAN comme ports du bond, assurez-vous que ces périphériques ne sont pas déjà configurés.

Procédure

- Ouvrez un terminal et entrez **nm-connection-editor**:

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

- Cliquez sur le bouton  pour ajouter une nouvelle connexion.
- Sélectionnez le type de connexion **Bond** et cliquez sur **Créer**.
- Dans l'onglet **Bond**:
 - Facultatif : Définissez le nom de l'interface de liaison dans le champ **Interface name**.
 - Cliquez sur le bouton **Ajouter** pour ajouter une interface réseau en tant que port à la liaison.
 - Sélectionnez le type de connexion de l'interface. Par exemple, sélectionnez **Ethernet** pour une connexion câblée.

- ii. Optionnel : Définir un nom de connexion pour le port.
 - iii. Si vous créez un profil de connexion pour un périphérique Ethernet, ouvrez l'onglet **Ethernet** et sélectionnez dans le champ **Device** l'interface réseau que vous souhaitez ajouter comme port à la liaison. Si vous avez sélectionné un autre type de périphérique, configurez-le en conséquence. Notez que vous ne pouvez utiliser dans un lien que des interfaces Ethernet non configurées.
 - iv. Cliquez sur **Enregistrer**.
- c. Répétez l'étape précédente pour chaque interface que vous souhaitez ajouter à la liaison :

Editing Bond connection 1

Connection name: Bond connection 1

General **Bond** Proxy IPv4 Settings IPv6 Settings

Interface name: bond0

Bonded connections

bond0-port1	Add
bond0-port2	

Edit

- d. Facultatif : Définissez d'autres options, telles que l'intervalle de surveillance de l'interface indépendante de média (MII).
5. Configurez les paramètres de l'adresse IP dans les onglets **IPv4 Settings** et **IPv6 Settings**:
- Pour utiliser ce dispositif de pont comme port d'autres dispositifs, réglez le champ **Method** sur **Disabled**.
 - Pour utiliser DHCP, laissez le champ **Method** à sa valeur par défaut, **Automatic (DHCP)**.
 - Pour utiliser des paramètres IP statiques, réglez le champ **Method** sur **Manual** et remplissez les champs en conséquence :

Editing Bond connection 1 (IPv4 Settings)

Connection name: Bond connection 1

General Bond Proxy **IPv4 Settings** IPv6 Settings

Method: Manual

Addresses

Address	Netmask	Gateway
192.0.2.1	24	192.0.2.254

DNS servers: 192.0.2.253
Search domains: example.com

Editing Bond connection 1 (IPv6 Settings)

Connection name: Bond connection 1

General Bond Proxy IPv4 Settings **IPv6 Settings**

Method: Manual

Addresses

Address	Prefix	Gateway
2001:db8:1::1	64	2001:db8:1::fff3

DNS servers: 2001:db8:1::ffff
Search domains: example.com

6. Cliquez sur **Enregistrer**.
7. Fermer **nm-connection-editor**.

Vérification

1. Retirez temporairement le câble réseau de l'hôte.
Il convient de noter qu'il n'existe aucune méthode permettant de tester correctement les événements de défaillance de liaison à l'aide d'utilitaires logiciels. Les outils qui désactivent les connexions, tels que **nmcli**, ne montrent que la capacité du pilote de liaison à gérer les changements de configuration des ports et non les événements réels de défaillance de la liaison.
2. Affiche l'état de la liaison :

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

Ressources supplémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)
- [Configuration d'une équipe réseau à l'aide de nm-connection-editor](#)
- [Configuration d'un pont réseau à l'aide de nm-connection-editor](#)
- [Configuration du marquage VLAN à l'aide de nm-connection-editor](#)

3.8. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DE NMSTATECTL

Pour configurer une liaison réseau à l'aide de l'API Nmstate, utilisez l'utilitaire **nmstatectl**.

Par exemple, la procédure ci-dessous crée un lien dans NetworkManager avec les paramètres suivants :

- Interfaces de réseau dans le lien : **enp1s0** et **enp7s0**
- Mode : **active-backup**
- Adresse IPv4 statique : **192.0.2.1** avec un masque de sous-réseau **/24**
- Adresse IPv6 statique : **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Passerelle par défaut IPv4 : **192.0.2.254**
- Passerelle par défaut IPv6 : **2001:db8:1::fffe**
- Serveur DNS IPv4 : **192.0.2.200**
- Serveur DNS IPv6 : **2001:db8:1::ffbb**
- Domaine de recherche DNS : **example.com**

Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports dans le lien, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.

- Pour utiliser des périphériques d'équipe, de pont ou de VLAN comme ports dans le lien, définissez le nom de l'interface dans la liste **port** et définissez les interfaces correspondantes.
- Le paquet **nmstate** est installé.

Procédure

1. Créez un fichier YAML, par exemple `~/create-bond.yml`, avec le contenu suivant :

```
---
interfaces:
- name: bond0
  type: bond
  state: up
  ipv4:
    enabled: true
    address:
      - ip: 192.0.2.1
        prefix-length: 24
    dhcp: false
  ipv6:
    enabled: true
    address:
      - ip: 2001:db8:1::1
        prefix-length: 64
    autoconf: false
    dhcp: false
  link-aggregation:
    mode: active-backup
    port:
      - enp1s0
      - enp7s0
- name: enp1s0
  type: ethernet
  state: up
- name: enp7s0
  type: ethernet
  state: up

routes:
  config:
    - destination: 0.0.0.0/0
      next-hop-address: 192.0.2.254
      next-hop-interface: bond0
    - destination: ::0
      next-hop-address: 2001:db8:1::fffe
      next-hop-interface: bond0

dns-resolver:
  config:
    search:
      - example.com
    server:
      - 192.0.2.200
      - 2001:db8:1::ffbb
```

2. Appliquer les paramètres au système :

```
# nmstatectl apply ~/create-bond.yml
```

Vérification

1. Affiche l'état des appareils et des connexions :

```
# nmcli device status
DEVICE  TYPE  STATE  CONNECTION
bond0   bond  connected bond0
```

2. Affiche tous les paramètres du profil de connexion :

```
# nmcli connection show bond0
connection.id:      bond0
connection.uuid:    79cbc3bd-302e-4b1f-ad89-f12533b818ee
connection.stable-id:  --
connection.type:    bond
connection.interface-name: bond0
...
```

3. Affiche les paramètres de connexion au format YAML :

```
# nmstatectl show bond0
```

Ressources supplémentaires

- [nmstatectl\(8\)](#) page de manuel
- [/usr/share/doc/nmstate/examples/](#) répertoire

3.9. CONFIGURATION D'UNE LIAISON RÉSEAU À L'AIDE DU RÔLE DE SYSTÈME RHEL RÉSEAU

Vous pouvez utiliser le site **network** RHEL System Roles pour configurer un lien Linux. Par exemple, vous pouvez l'utiliser pour configurer un lien réseau en mode de sauvegarde active qui utilise deux périphériques Ethernet et définit des adresses IPv4 et IPv6, des passerelles par défaut et une configuration DNS.



NOTE

Définir la configuration IP sur le lien et non sur les ports du lien Linux.

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.

- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.

Procédure

1. Créez un fichier playbook, par exemple `~/bond-ethernet.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure a network bond that uses two Ethernet ports
      include_role:
        name: rhel-system-roles.network

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

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

      # Add a second Ethernet profile to the bond
      - name: bond0-port2
        interface_name: enp8s0
        type: ethernet
        controller: bond0
        state: up
```

2. Exécutez le manuel de jeu :

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

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

3.10. CRÉATION D'UNE LIAISON RÉSEAU PERMETTANT DE PASSER D'UNE CONNEXION ETHERNET À UNE CONNEXION SANS FIL SANS INTERROMPRE LE VPN

Les utilisateurs de RHEL qui connectent leur poste de travail au réseau de leur entreprise utilisent généralement un VPN pour accéder à des ressources distantes. Cependant, si le poste de travail passe d'une connexion Ethernet à une connexion Wi-Fi, par exemple si vous libérez un ordinateur portable d'une station d'accueil dotée d'une connexion Ethernet, la connexion VPN est interrompue. Pour éviter ce problème, vous pouvez créer une liaison réseau qui utilise les connexions Ethernet et Wi-Fi en mode **active-backup**.

Conditions préalables

- L'hôte contient un dispositif Ethernet et un dispositif Wi-Fi.
- Un profil de connexion Ethernet et Wi-Fi NetworkManager a été créé et les deux connexions fonctionnent indépendamment.
Cette procédure utilise les profils de connexion suivants pour créer une liaison réseau nommée **bond0**:
 - **Docking_station** associé à l'appareil Ethernet **enp11s0u1**
 - **Wi-Fi** associé à l'appareil Wi-Fi **wlp1s0**

Procédure

1. Créer une interface de liaison en mode **active-backup**:

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

Cette commande nomme à la fois l'interface et le profil de connexion **bond0**.

2. Configurer les paramètres IPv4 de la liaison :

- Si un serveur DHCP dans votre réseau attribue des adresses IPv4 aux hôtes, aucune action n'est requise.
- Si votre réseau local nécessite des adresses IPv4 statiques, définissez l'adresse, le masque de réseau, la passerelle par défaut, le serveur DNS et le domaine de recherche DNS pour la connexion **bond0**:

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


3. Configurer les paramètres IPv6 de la liaison :

- Si votre routeur ou un serveur DHCP de votre réseau attribue des adresses IPv6 aux hôtes, aucune action n'est requise.
- Si votre réseau local nécessite des adresses IPv6 statiques, définissez l'adresse, le masque de réseau, la passerelle par défaut, le serveur DNS et le domaine de recherche DNS pour la connexion **bond0**:

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

4. Afficher les profils de connexion :

```
# nmcli connection show
NAME          UUID                                TYPE  DEVICE
Docking_station 256dd073-fecc-339d-91ae-9834a00407f9 ethernet enp11s0u1
Wi-Fi          1f1531c7-8737-4c60-91af-2d21164417e8 wifi   wlp1s0
...
```

Les noms des profils de connexion et le nom de l'appareil Ethernet sont nécessaires dans les étapes suivantes.

5. Attribuer le profil de connexion de la connexion Ethernet à la liaison :

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

6. Attribuer le profil de connexion de la connexion Wi-Fi à la liaison :

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

7. Si votre réseau Wi-Fi utilise le filtrage MAC pour n'autoriser que les adresses MAC d'une liste à accéder au réseau, configurez NetworkManager pour qu'il assigne dynamiquement l'adresse MAC du port actif à la liaison :

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

Avec ce paramètre, vous devez définir uniquement l'adresse MAC de l'appareil Wi-Fi dans la liste d'autorisation au lieu de l'adresse MAC de l'appareil Ethernet et de l'appareil Wi-Fi.

8. Définir l'appareil associé à la connexion Ethernet comme appareil primaire de la liaison :

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

Avec ce paramètre, la liaison utilise toujours la connexion Ethernet si elle est disponible.

9. Configurer NetworkManager pour qu'il active automatiquement les ports lorsque le périphérique **bond0** est activé :

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

10. Activez la connexion **bond0**:

```
# nmcli connection up bond0
```

Vérification

- Affiche l'appareil actuellement actif, l'état de la liaison et de ses ports :

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

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

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

Slave Interface: wlp1s0
MII Status: up
Speed: Unknown
Duplex: Unknown
Link Failure Count: 2
Permanent HW addr: 00:53:00:b3:22:ba
Slave queue ID: 0
```

Ressources supplémentaires

- [Configuration d'une connexion Ethernet](#)
- [Gestion des connexions Wi-Fi](#)
- [Configuration de la liaison réseau](#)

3.11. LES DIFFÉRENTS MODES DE CONNEXION AU RÉSEAU

Le pilote de liaison Linux permet l'agrégation de liens. Le bonding est le processus d'agrégation de plusieurs interfaces réseau en parallèle pour fournir une interface logique unique. Les actions d'une interface liée dépendent de la politique de liaison, également connue sous le nom de mode. Les différents modes fournissent soit des services d'équilibrage de charge, soit des services de secours à chaud.

Les modes suivants existent :

Balance-rr (Mode 0)

Balance-rr utilise l'algorithme round-robin qui transmet séquentiellement les paquets du premier port disponible au dernier. Ce mode assure l'équilibrage de la charge et la tolérance aux pannes. Ce mode nécessite la configuration d'un groupe d'agrégation de ports, également appelé EtherChannel ou groupement de ports similaire. Un EtherChannel est une technologie d'agrégation de liens portuaires permettant de regrouper plusieurs liens Ethernet physiques en un lien Ethernet logique.

L'inconvénient de ce mode est qu'il n'est pas adapté aux charges de travail lourdes et si le débit TCP ou la livraison ordonnée des paquets sont essentiels.

Sauvegarde active (Mode 1)

Active-backup utilise la politique qui détermine qu'un seul port est actif dans le lien. Ce mode offre une tolérance aux pannes et ne nécessite aucune configuration du commutateur.

Si le port actif tombe en panne, un autre port devient actif. La liaison envoie au réseau une réponse gratuite au protocole de résolution d'adresses (ARP). L'ARP gratuit oblige le récepteur de la trame ARP à mettre à jour sa table de transfert. Le mode **Active-backup** transmet un ARP gratuit pour annoncer le nouveau chemin afin de maintenir la connectivité pour l'hôte.

L'option **primary** définit le port préféré de l'interface de liaison.

Balance-xor (Mode 2)

Balance-xor utilise la politique de hachage de transmission sélectionnée pour envoyer les paquets. Ce mode permet l'équilibrage de la charge, la tolérance aux pannes et nécessite la configuration d'un commutateur pour mettre en place un canal Etherchannel ou un groupement de ports similaire. Pour modifier la transmission des paquets et équilibrer la transmission, ce mode utilise l'option **xmit_hash_policy**. En fonction de la source ou de la destination du trafic sur l'interface, l'interface nécessite une configuration supplémentaire d'équilibrage de la charge. Voir la description du [paramètre de liaison xmit_hash_policy](#).

Diffusion (Mode 3)

Broadcast utilise une politique qui transmet chaque paquet sur toutes les interfaces. Ce mode offre une tolérance aux pannes et nécessite la configuration d'un commutateur pour mettre en place un canal EtherChannel ou un groupement de ports similaire.

L'inconvénient de ce mode est qu'il n'est pas adapté aux charges de travail lourdes et si le débit TCP ou la livraison ordonnée des paquets sont essentiels.

802.3ad (Mode 4)

802.3ad utilise la politique d'agrégation dynamique de liens de la norme IEEE du même nom. Ce mode offre une tolérance aux pannes. Ce mode nécessite la configuration d'un commutateur pour établir un groupement de ports LACP (Link Aggregation Control Protocol).

Ce mode crée des groupes d'agrégation qui partagent les mêmes paramètres de vitesse et de duplex et utilise tous les ports de l'agrégateur actif. En fonction de la source ou de la destination du trafic sur l'interface, ce mode nécessite une configuration supplémentaire d'équilibrage de la charge.

Par défaut, la sélection des ports pour le trafic sortant dépend de la politique de hachage de transmission. Utilisez l'option **xmit_hash_policy** de la stratégie de hachage de transmission pour modifier la sélection des ports et équilibrer la transmission.

La différence entre **802.3ad** et **Balance-xor** est la conformité. La politique **802.3ad** négocie LACP entre les groupes d'agrégation de ports. Voir la description du [paramètre de liaison xmit_hash_policy](#)

Balance-tlb (Mode 5)

Balance-tlb utilise la politique d'équilibrage de la charge de transmission. Ce mode assure la tolérance aux pannes, l'équilibrage de la charge et établit un lien entre les canaux qui ne nécessitent pas de support de la part du commutateur.

Le port actif reçoit le trafic entrant. En cas de défaillance du port actif, un autre prend en charge l'adresse MAC du port défaillant. Pour décider quelle interface traite le trafic sortant, utilisez l'un des modes suivants :

- Valeur **0**: Utilise la politique de distribution de hachage pour distribuer le trafic sans équilibrer la charge
- Valeur **1**: Distribue le trafic à chaque port en utilisant l'équilibrage de charge. Avec l'option de liaison **tlb_dynamic_lb=0**, ce mode de liaison utilise l'option de liaison **xmit_hash_policy** pour équilibrer la transmission. L'option **primary** définit le port préféré de l'interface de liaison.

Voir la description du [paramètre de liaison xmit_hash_policy](#) .

Balance-alb (Mode 6)

Balance-alb utilise une politique d'équilibrage de charge adaptative. Ce mode assure la tolérance aux pannes, l'équilibrage de la charge et ne nécessite pas de support particulier de la part du commutateur.

Ce mode inclut l'équilibrage de la charge de transmission (**balance-tlb**) et l'équilibrage de la charge de réception pour le trafic IPv4 et IPv6. Le bonding intercepte les réponses ARP envoyées par le système local et écrase l'adresse matérielle source de l'un des ports du bonding. La négociation ARP gère l'équilibrage de la charge de réception. Par conséquent, différents ports utilisent différentes adresses matérielles pour le serveur.

L'option **primary** définit le port préféré de l'interface de liaison. Avec l'option de liaison **tlb_dynamic_lb=0**, ce mode de liaison utilise l'option de liaison **xmit_hash_policy** pour équilibrer la transmission. Voir la description du [paramètre de liaison xmit_hash_policy](#) .

Ressources supplémentaires

- [/usr/share/doc/kernel-doc-<version>/Documentation/networking/bonding.rst](#) fournie par le paquet **kernel-doc**
- [/usr/share/doc/kernel-doc-<version>/Documentation/networking/bonding.txt](#) fournie par le paquet **kernel-doc**
- [Quels sont les modes de liaison qui fonctionnent lorsqu'ils sont utilisés avec un pont auquel se connectent des invités ou des conteneurs de machines virtuelles ?](#)
- [Comment sont calculées les valeurs des différentes politiques dans le paramètre de liaison "xmit_hash_policy" ?](#)

3.12. LE PARAMÈTRE DE LIAISON XMIT_HASH_POLICY

Le paramètre d'équilibrage de charge **xmit_hash_policy** sélectionne la politique de hachage de transmission pour une sélection de nœuds dans les modes **balance-xor**, **802.3ad**, **balance-alb** et **balance-tlb**. Il ne s'applique qu'aux modes 5 et 6 si le mode **tlb_dynamic_lb parameter is 0**. Les valeurs possibles de ce paramètre sont **layer2**, **layer2 3**, **layer3 4**, **encap2 3**, **encap3 4**, et **vlan srcmac**.

Voir le tableau pour plus de détails :

Policy or Network layers	Layer2	Layer2 3	Layer3 4	encap2 3	encap3 4	VLAN srcmac
Uses	XOR des adresses MAC source et destination et du type de protocole Ethernet	XOR des adresses MAC et des adresses IP de la source et de la destination	XOR des ports et adresses IP source et destination	XOR des adresses MAC et IP source et destination à l'intérieur d'un tunnel pris en charge, par exemple, Virtual Extensible LAN (VXLAN). Ce mode s'appuie sur la fonction skb_flow_dissect() pour obtenir les champs de l'en-tête	XOR des ports et adresses IP source et destination à l'intérieur d'un tunnel pris en charge, par exemple VXLAN. Ce mode s'appuie sur la fonction skb_flow_dissect() pour obtenir les champs de l'en-tête	XOR de l'ID VLAN, du fournisseur MAC source et du dispositif MAC source
Placement of traffic	Tout le trafic à destination d'un pair de réseau particulier sur la même interface de réseau sous-jacente	Tout le trafic vers une adresse IP particulière sur la même interface réseau sous-jacente	Tout le trafic à destination d'une adresse IP et d'un port particuliers sur la même interface réseau sous-jacente			

Primary choice	Si le trafic réseau se fait entre ce système et plusieurs autres systèmes dans le même domaine de diffusion	Si le trafic réseau entre ce système et plusieurs autres systèmes passe par une passerelle par défaut	Si le trafic réseau entre ce système et un autre système utilise les mêmes adresses IP mais passe par plusieurs ports	Le trafic encapsulé est entre le système source et plusieurs autres systèmes utilisant plusieurs adresses IP	Le trafic encapsulé se fait entre le système source et d'autres systèmes en utilisant plusieurs numéros de port	Si le lien transporte du trafic réseau, provenant de plusieurs conteneurs ou machines virtuelles (VM), qui exposent leur adresse MAC directement au réseau externe tel que le réseau de pont, et que vous ne pouvez pas configurer un commutateur pour le mode 2 ou le mode 4, vous pouvez utiliser le mode 2 ou le mode 4
Secondary choice	Si le trafic réseau s'effectue principalement entre ce système et plusieurs autres systèmes derrière une passerelle par défaut	Si le trafic réseau se fait principalement entre ce système et un autre système				
Compliant	802.3ad	802.3ad	Pas 802.3ad			

Default policy	Il s'agit de la politique par défaut si aucune configuration n'est fournie	Pour le trafic non-IP, la formule est la même que pour la politique de transmission layer2	Pour le trafic non-IP, la formule est la même que pour la politique de transmission layer2			
-----------------------	--	---	---	--	--	--

CHAPITRE 4. CONFIGURATION DU TEAMING RÉSEAU

Une équipe de réseau est une méthode permettant de combiner ou d'agréger des interfaces de réseau physiques et virtuelles afin de fournir une interface logique avec un débit ou une redondance plus élevés. Dans une équipe réseau, un petit module du noyau et un service de l'espace utilisateur traitent les opérations. Vous pouvez créer des équipes réseau sur différents types de périphériques, tels que les périphériques Ethernet ou les VLAN.

Red Hat Enterprise Linux offre aux administrateurs différentes options pour configurer les périphériques d'équipe. Par exemple :

- Utilisez **nmcli** pour configurer les connexions d'équipes à l'aide de la ligne de commande.
- Utilisez la console web RHEL pour configurer les connexions d'équipe à l'aide d'un navigateur web.
- Utilisez l'application **nm-connection-editor** pour configurer les connexions d'équipe dans une interface graphique.

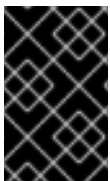


IMPORTANT

L'association de réseaux est obsolète dans Red Hat Enterprise Linux 9. Considérez l'utilisation du pilote de liaison réseau comme une alternative. Pour plus de détails, voir [Configuration de la liaison réseau](#).

4.1. MIGRATION D'UNE CONFIGURATION D'ÉQUIPE DE RÉSEAU VERS UN LIEN DE RÉSEAU

L'équipe réseau est obsolète dans Red Hat Enterprise Linux 9. Si vous avez déjà une équipe réseau fonctionnelle configurée, par exemple parce que vous avez effectué une mise à niveau à partir d'une version antérieure de RHEL, vous pouvez migrer la configuration vers un lien réseau qui est géré par NetworkManager.



IMPORTANT

L'utilitaire **team2bond** ne fait que convertir la configuration de l'équipe réseau en un lien. Ensuite, vous devez configurer manuellement les autres paramètres du lien, tels que les adresses IP et la configuration DNS.

Conditions préalables

- Le profil de connexion **team-team0** NetworkManager est configuré et gère le périphérique **team0**.
- Le paquet **teamd** est installé.

Procédure

1. Optionnel : Affichez la configuration IP de la connexion **team-team0** NetworkManager :

```
# nmcli connection show team-team0 | egrep "^ip"
...
ipv4.method:          manual
ipv4.dns:              192.0.2.253
```



```

ipv4.dns-search:          example.com
ipv4.addresses:          192.0.2.1/24
ipv4.gateway:            192.0.2.254
...
ipv6.method:             manual
ipv6.dns:                2001:db8:1::fffd
ipv6.dns-search:         example.com
ipv6.addresses:          2001:db8:1::1/64
ipv6.gateway:            2001:db8:1::fffe
...

```

2. Exporter la configuration de l'appareil **team0** dans un fichier JSON :

```
# teamdctl team0 config dump actual > /tmp/team0.json
```

3. Supprimez l'équipe réseau. Par exemple, si vous avez configuré l'équipe dans NetworkManager, supprimez le profil de connexion **team-team0** et les profils des ports associés :

```
# nmcli connection delete team-team0
# nmcli connection delete team-team0-port1
# nmcli connection delete team-team0-port2
```

4. Exécutez l'utilitaire **team2bond** en mode dry-run pour afficher les commandes **nmcli** qui établissent une liaison réseau avec des paramètres similaires à ceux de l'appareil de l'équipe :

```
# team2bond --config=/tmp/team0.json --rename=bond0
nmcli con add type bond ifname bond0 bond.options "mode=active-
backup,num_grat_arp=1,num_unsol_na=1,resent_igmp=1,miimon=100,miimon=100"
nmcli con add type ethernet ifname enp7s0 master bond0
nmcli con add type ethernet ifname enp8s0 master bond0
```

La première commande contient deux options **miimon** car le fichier de configuration de l'équipe contenait deux entrées **link_watch**. Notez que cela n'affecte pas la création du lien.

Si vous avez lié des services au nom de périphérique de l'équipe et que vous voulez éviter de mettre à jour ou d'interrompre ces services, omettez l'option **--rename=bond0**. Dans ce cas, **team2bond** utilise le même nom d'interface pour le lien que pour l'équipe.

5. Vérifiez que les options de l'obligation proposées par l'utilitaire **team2bond** sont correctes.
6. Créez le lien. Vous pouvez exécuter les commandes **nmcli** proposées ou réexécuter la commande **team2bond** avec l'option **--exec-cmd**:

```
# team2bond --config=/tmp/team0.json --rename=bond0 --exec-cmd
Connection 'bond-bond0' (0241a531-0c72-4202-80df-73eadfc126b5) successfully added.
Connection 'bond-slave-enp7s0' (38489729-b624-4606-a784-1ccf01e2f6d6) successfully
added.
Connection 'bond-slave-enp8s0' (de97ec06-7daa-4298-9a71-9d4c7909daa1) successfully
added.
```

Vous aurez besoin du nom du profil de connexion (**bond-bond0**) dans les étapes suivantes.

7. Définissez les paramètres IPv4 précédemment configurés sur **team-team0** pour la connexion **bond-bond0**:

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

- Définissez les paramètres IPv6 précédemment configurés sur **team-team0** pour la connexion **bond-bond0**:

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

- Activer la connexion :

```
# nmcli connection up bond-bond0
```

Vérification

- Affiche la configuration IP de la connexion **bond-bond0** NetworkManager :

```
# nmcli connection show bond-bond0 | egrep "^ip"
...
ipv4.method:                manual
ipv4.dns:                   192.0.2.253
ipv4.dns-search:           example.com
ipv4.addresses:            192.0.2.1/24
ipv4.gateway:              192.0.2.254
...
ipv6.method:                manual
ipv6.dns:                   2001:db8:1::fffd
ipv6.dns-search:           example.com
ipv6.addresses:            2001:db8:1::1/64
ipv6.gateway:              2001:db8:1::fffe
...
```

- Affiche l'état de la liaison :

```
# cat /proc/net/bonding/bond0
Ethernet Channel Bonding Driver: v5.13.0-0.rc7.51.el9.x86_64

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

Slave Interface: enp7s0
MII Status: up
```

```

Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 52:54:00:bf:b1:a9
Slave queue ID: 0

Slave Interface: enp8s0
MII Status: up
Speed: Unknown
Duplex: Unknown
Link Failure Count: 0
Permanent HW addr: 52:54:00:04:36:0f
Slave queue ID: 0

```

Dans cet exemple, les deux ports sont activés.

3. Pour vérifier que le basculement des liens fonctionne :
 - a. Retirez temporairement le câble réseau de l'hôte. Notez qu'il n'existe aucune méthode permettant de tester correctement les événements de défaillance de liaison à l'aide de la ligne de commande.
 - b. Affiche l'état de la liaison :

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

4.2. COMPRENDRE LE TRAVAIL EN ÉQUIPE EN RÉSEAU

L'équipe réseau est une fonction qui combine ou agrège des interfaces réseau pour fournir une interface logique avec un débit plus élevé ou une redondance.

Le teaming réseau utilise un pilote de noyau pour mettre en œuvre le traitement rapide des flux de paquets, ainsi que des bibliothèques et des services de l'espace utilisateur pour d'autres tâches. Ainsi, le teaming de réseau est une solution facilement extensible et évolutive pour les besoins d'équilibrage de charge et de redondance.



IMPORTANT

Certaines fonctions de teaming réseau, telles que le mécanisme de basculement, ne prennent pas en charge les connexions directes par câble sans commutateur réseau. Pour plus de détails, voir [Le bonding est-il pris en charge avec une connexion directe utilisant des câbles croisés ?](#)

4.3. COMPRENDRE LE COMPORTEMENT PAR DÉFAUT DES INTERFACES DES CONTRÔLEURS ET DES PORTS

Tenez compte du comportement par défaut suivant lorsque vous gérez ou dépannez des interfaces de port d'équipe ou de liaison à l'aide du service **NetworkManager**:

- Le démarrage de l'interface du contrôleur ne démarre pas automatiquement les interfaces des ports.
- Le démarrage d'une interface de port entraîne toujours le démarrage de l'interface du contrôleur.

- L'arrêt de l'interface du contrôleur entraîne également l'arrêt de l'interface du port.
- Un contrôleur sans ports peut démarrer des connexions IP statiques.
- Un contrôleur sans ports attend les ports lors du démarrage des connexions DHCP.
- Un contrôleur avec une connexion DHCP en attente de ports se termine lorsque vous ajoutez un port avec une porteuse.
- Un contrôleur avec une connexion DHCP en attente de ports continue d'attendre lorsque vous ajoutez un port sans support.

4.4. COMPRENDRE LE SERVICE TEAMD, LES RUNNERS ET LES LINK-WATCHERS

Le service d'équipe, **teamd**, contrôle une instance du pilote d'équipe. Cette instance du pilote ajoute des instances d'un pilote de périphérique matériel pour former une équipe d'interfaces réseau. Le pilote d'équipe présente une interface réseau, par exemple **team0**, au noyau.

Le service **teamd** met en œuvre la logique commune à toutes les méthodes de travail en équipe. Ces fonctions sont propres aux différentes méthodes de répartition de la charge et de sauvegarde, telles que le round-robin, et sont mises en œuvre par des unités de code distinctes appelées **runners**. Les administrateurs spécifient les exécutants au format JavaScript Object Notation (JSON), et le code JSON est compilé dans une instance de **teamd** lors de la création de l'instance. Par ailleurs, lorsque vous utilisez **NetworkManager**, vous pouvez définir l'unité d'exécution dans le paramètre **team.runner**, et **NetworkManager** crée automatiquement le code JSON correspondant.

Les coureurs suivants sont disponibles :

- **broadcast**: Transmet les données sur tous les ports.
- **roundrobin**: Transmet les données sur tous les ports à tour de rôle.
- **activebackup**: Transmet les données sur un port tandis que les autres sont conservés comme sauvegarde.
- **loadbalance**: Transmet les données sur tous les ports avec un équilibrage de charge Tx actif et des sélecteurs de port Tx basés sur le filtre de paquets Berkeley (BPF).
- **random**: Transmet des données sur un port sélectionné de manière aléatoire.
- **lACP**: Met en œuvre le protocole 802.3ad Link Aggregation Control Protocol (LACP).

Les services **teamd** utilisent un observateur de liens pour surveiller l'état des périphériques subordonnés. Les observateurs de liens suivants sont disponibles :

- **ethtool**: La bibliothèque **libteam** utilise l'utilitaire **ethtool** pour surveiller les changements d'état des liens. Il s'agit de l'observateur de liens par défaut.
- **arp_ping**: La bibliothèque **libteam** utilise l'utilitaire **arp_ping** pour surveiller la présence d'une adresse matérielle distante à l'aide du protocole de résolution d'adresses (ARP).
- **nsna_ping**: Sur les connexions IPv6, la bibliothèque **libteam** utilise les fonctions Neighbor Advertisement et Neighbor Solicitation du protocole IPv6 Neighbor Discovery pour surveiller la présence de l'interface d'un voisin.

Chaque exécutant peut utiliser n'importe quel observateur de liens, à l'exception de **lACP**. Cet exécutant ne peut utiliser que l'observateur de liens **ethtool**.

4.5. CONFIGURATION D'UNE ÉQUIPE RÉSEAU À L'AIDE DE NMCLI

Pour configurer une équipe réseau en ligne de commande, utilisez l'utilitaire **nmcli**.



IMPORTANT

L'association de réseaux est obsolète dans Red Hat Enterprise Linux 9. Considérez l'utilisation du pilote de liaison réseau comme une alternative. Pour plus de détails, voir [Configuration de la liaison réseau](#).

Conditions préalables

- Les paquets **teamd** et **NetworkManager-team** sont installés.
- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports de l'équipe, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur et connectés à un commutateur.
- Pour utiliser des dispositifs de liaison, de pont ou de VLAN comme ports de l'équipe, vous pouvez soit créer ces dispositifs lors de la création de l'équipe, soit les créer à l'avance comme décrit dans la section :
 - [Configuration d'une liaison réseau à l'aide de nmcli](#)
 - [Configuration d'un pont réseau à l'aide de nmcli](#)
 - [Configuration du marquage des VLAN à l'aide de nmcli](#)

Procédure

1. Créer une interface d'équipe :

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

Cette commande crée une équipe réseau nommée **team0** qui utilise le programme d'exécution **activebackup**.

2. Vous pouvez également définir un observateur de liens. Par exemple, pour définir l'observateur de liens **ethtool** dans le profil de connexion **team0**:

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

Les observateurs de liens prennent en charge différents paramètres. Pour définir les paramètres d'un observateur de liens, spécifiez-les en les séparant par des espaces dans la propriété **name**. Notez que la propriété **name** doit être entourée de guillemets. Par exemple, pour utiliser l'observateur de liens **ethtool** et définir son paramètre **delay-up** à **2500** millisecondes (2,5 secondes) :

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

Pour définir plusieurs observateurs de liens et chacun d'eux avec des paramètres spécifiques, les observateurs de liens doivent être séparés par une virgule. L'exemple suivant définit l'observateur de liens **ethtool** avec le paramètre **delay-up** et l'observateur de liens **arp_ping** avec les paramètres **source-host** et **target-host**:

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

3. Affichez les interfaces réseau et notez les noms des interfaces que vous souhaitez ajouter à l'équipe :

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

Dans cet exemple :

- **enp7s0** et **enp8s0** ne sont pas configurés. Pour utiliser ces périphériques comme ports, ajoutez des profils de connexion à l'étape suivante. Notez que vous ne pouvez utiliser dans une équipe que des interfaces Ethernet qui ne sont affectées à aucune connexion.
- **bond0** et **bond1** ont des profils de connexion existants. Pour utiliser ces dispositifs comme ports, modifiez leurs profils à l'étape suivante.

4. Attribuer les interfaces de port à l'équipe :

- a. Si les interfaces que vous souhaitez attribuer à l'équipe ne sont pas configurées, créez de nouveaux profils de connexion pour elles :

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

. Ces commandes créent des profils pour **enp7s0** et **enp8s0** et les ajoutent à la connexion **team0**.

- b. Pour affecter un profil de connexion existant à l'équipe :

- i. Réglez le paramètre **master** de ces connexions sur **team0**:

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

Ces commandes affectent les profils de connexion existants nommés **bond0** et **bond1** à la connexion **team0**.

- ii. Réactiver les connexions :

```
# nmcli connection up bond0
# nmcli connection up bond1
```

5. Configurez les paramètres IPv4 :

- Pour utiliser cet appareil d'équipe comme port d'autres appareils, entrez :

```
# nmcli connection modify team0 ipv4.method disabled
```

- Pour utiliser le DHCP, aucune action n'est nécessaire.
- Pour définir une adresse IPv4 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **team0**, entrez :

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

6. Configurez les paramètres IPv6 :

- Pour utiliser cet appareil d'équipe comme port d'autres appareils, entrez :

```
# nmcli connection modify team0 ipv6.method disabled
```

- Pour utiliser le DHCP, aucune action n'est nécessaire.
- Pour définir une adresse IPv6 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **team0**, entrez :

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

7. Activer la connexion :

```
# nmcli connection up team0
```

Vérification

- Afficher le statut de l'équipe :

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

```
link: up
down count: 0
runner:
active port: enp7s0
```

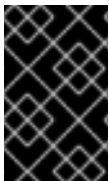
Dans cet exemple, les deux ports sont activés.

Ressources supplémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)
- [Comprendre le service teamd, les runners et les link-watchers](#)
- **nm-settings(5)** page de manuel
- **teamd.conf(5)** page de manuel

4.6. CONFIGURATION D'UNE ÉQUIPE RÉSEAU À L'AIDE DE LA CONSOLE WEB RHEL

Utilisez la console web RHEL pour configurer une équipe réseau si vous préférez gérer les paramètres réseau à l'aide d'une interface basée sur un navigateur web.



IMPORTANT

L'association de réseaux est obsolète dans Red Hat Enterprise Linux 9. Considérez l'utilisation du pilote de liaison réseau comme une alternative. Pour plus de détails, voir [Configuration de la liaison réseau](#).

Conditions préalables

- Les paquets **teamd** et **NetworkManager-team** sont installés.
- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports de l'équipe, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur et connectés à un commutateur.
- Pour utiliser des périphériques bond, bridge ou VLAN comme ports de l'équipe, créez-les à l'avance comme décrit dans la section :
 - [Configuration d'une liaison réseau à l'aide de la console web RHEL](#)
 - [Configuration d'un pont réseau à l'aide de la console web RHEL](#)
 - [Configuration du marquage VLAN à l'aide de la console web RHEL](#)

Procédure



1. Sélectionnez l'onglet **Networking** dans le menu de navigation situé à gauche de l'écran.
2. Cliquez sur **Ajouter une équipe** dans la section **Interfaces**.
3. Saisissez le nom de l'appareil d'équipe que vous souhaitez créer.

4. Sélectionnez les interfaces qui doivent être des ports de l'équipe.
5. Sélectionnez le coureur de l'équipe.
Si vous sélectionnez **Load balancing** ou **802.3ad LACP**, la console web affiche le champ supplémentaire **Balancer**.
6. Définir l'observateur de liens :
 - Si vous sélectionnez **Ethtool**, définissez en outre un délai d'établissement de la liaison et un délai de rétablissement de la liaison.
 - Si vous avez défini **ARP ping** ou **NSNA ping**, définissez également un intervalle de ping et une cible de ping.

Team settings ✕

Name	<input style="width: 80%;" type="text" value="team0"/>
Ports	<input checked="" type="checkbox"/> enp7s0 <input checked="" type="checkbox"/> enp8s0
Runner	<input style="border-bottom: 1px solid #ccc;" type="text" value="Active backup"/>
Link watch	<input style="border-bottom: 1px solid #ccc;" type="text" value="Ethtool"/>
Link up delay	<input style="width: 80%;" type="text" value="0"/>
Link down delay	<input style="width: 80%;" type="text" value="0"/>

7. Cliquez sur **Appliquer**.
8. Par défaut, l'équipe utilise une adresse IP dynamique. Si vous souhaitez définir une adresse IP statique :
 - a. Cliquez sur le nom de l'équipe dans la section **Interfaces**.
 - b. Cliquez sur **Edit** en regard du protocole que vous souhaitez configurer.

- c. Sélectionnez **Manual** à côté de **Addresses**, et entrez l'adresse IP, le préfixe et la passerelle par défaut.
- d. Dans la section **DNS**, cliquez sur le bouton  et entrez l'adresse IP du serveur DNS. Répétez cette étape pour définir plusieurs serveurs DNS.
- e. Dans la section **DNS search domains**, cliquez sur le bouton  et entrez le domaine de recherche.
- f. Si l'interface nécessite des routes statiques, configurez-les dans la section **Routes**.

IPv4 settings ✕

Addresses Manual ▼ +

Address	Prefix length or netmask	Gateway	
<input style="width: 150px;" type="text" value="192.0.2.1"/>	<input style="width: 100px;" type="text" value="24"/>	<input style="width: 150px;" type="text" value="192.0.2.254"/>	-

DNS Automatic +

Server -

DNS search domains Automatic +

Search domain -

Routes Automatic +

Apply Cancel

- g. Cliquez sur **Appliquer**

Vérification

1. Sélectionnez l'onglet **Networking** dans la navigation sur le côté gauche de l'écran, et vérifiez s'il y a du trafic entrant et sortant sur l'interface.

Interfaces Add bond Add team Add bridge Add VLAN 			
Name	IP address	Sending	Receiving
team0	192.0.2.1/24	1.11 Mbps	61.2 Mbps

2. Afficher le statut de l'équipe :

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

Dans cet exemple, les deux ports sont activés.

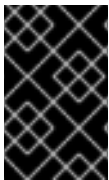
Ressources supplémentaires

- [Les coureurs de l'équipe du réseau](#)

4.7. CONFIGURATION D'UNE ÉQUIPE RÉSEAU À L'AIDE DE NM-CONNECTION-EDITOR

Si vous utilisez Red Hat Enterprise Linux avec une interface graphique, vous pouvez configurer les équipes réseau à l'aide de l'application **nm-connection-editor**.

Notez que **nm-connection-editor** ne peut ajouter que de nouveaux ports à une équipe. Pour utiliser un profil de connexion existant en tant que port, créez l'équipe à l'aide de l'utilitaire **nmcli**, comme décrit dans la section [Configuration d'une équipe réseau à l'aide de nmcli](#) .



IMPORTANT

L'association de réseaux est obsolète dans Red Hat Enterprise Linux 9. Considérez l'utilisation du pilote de liaison réseau comme une alternative. Pour plus de détails, voir [Configuration de la liaison réseau](#) .


Conditions préalables

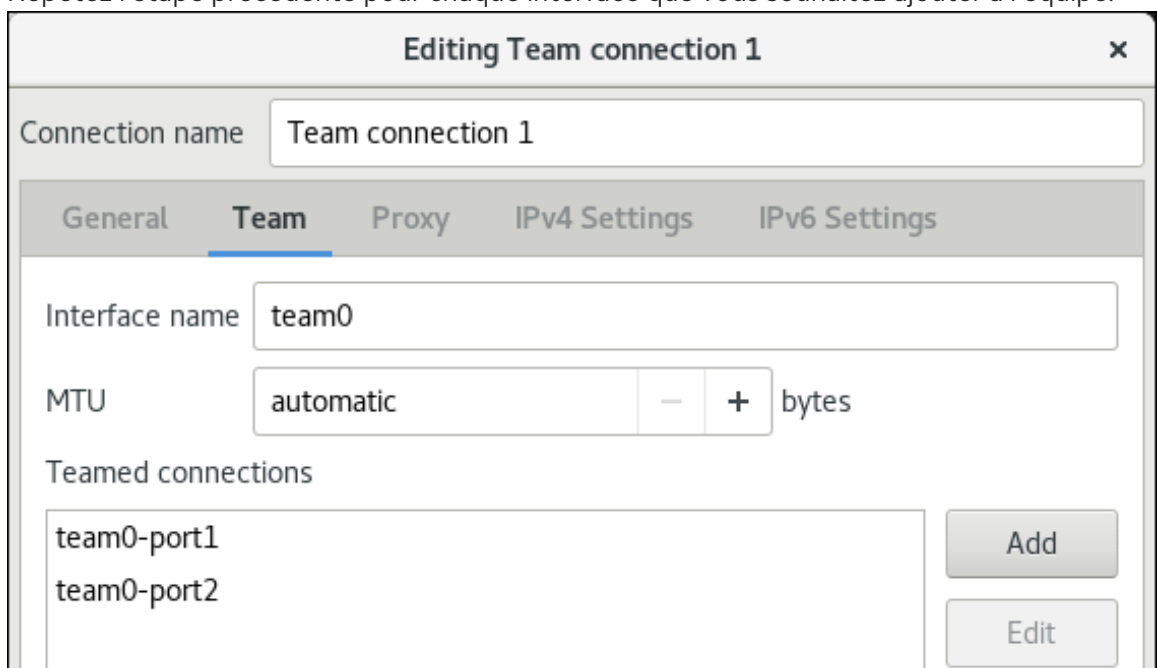
- Les paquets **teamd** et **NetworkManager-team** sont installés.
- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports de l'équipe, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.
- Pour utiliser des périphériques d'équipe, de lien ou de VLAN comme ports de l'équipe, assurez-vous que ces périphériques ne sont pas déjà configurés.

Procédure

1. Ouvrez un terminal et entrez **nm-connection-editor**:

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

2. Cliquez sur le bouton  pour ajouter une nouvelle connexion.
3. Sélectionnez le type de connexion **Team** et cliquez sur **Créer**.
4. Dans l'onglet **Team**:
 - a. Facultatif : Définissez le nom de l'interface de l'équipe dans le champ **Interface name**.
 - b. Cliquez sur le bouton **Ajouter** pour ajouter un nouveau profil de connexion pour une interface réseau et ajouter le profil en tant que port à l'équipe.
 - i. Sélectionnez le type de connexion de l'interface. Par exemple, sélectionnez **Ethernet** pour une connexion câblée.
 - ii. Optionnel : Définir un nom de connexion pour le port.
 - iii. Si vous créez un profil de connexion pour un périphérique Ethernet, ouvrez l'onglet **Ethernet** et sélectionnez dans le champ **Device** l'interface réseau que vous souhaitez ajouter comme port à l'équipe. Si vous avez sélectionné un autre type de périphérique, configurez-le en conséquence. Notez que vous ne pouvez utiliser dans une équipe que des interfaces Ethernet qui ne sont affectées à aucune connexion.
 - iv. Cliquez sur **Enregistrer**.
 - c. Répétez l'étape précédente pour chaque interface que vous souhaitez ajouter à l'équipe.



- d. Cliquez sur le bouton **Avancé** pour définir des options avancées pour la connexion d'équipe.
 - i. Dans l'onglet **Runner**, sélectionnez le coureur.
 - ii. Dans l'onglet **Link Watcher**, définissez l'observateur de liens et ses paramètres facultatifs.

iii. Cliquez sur **OK**.

5. Configurez les paramètres de l'adresse IP dans les onglets **IPv4 Settings** et **IPv6 Settings**:

- Pour utiliser ce dispositif de pont comme port d'autres dispositifs, réglez le champ **Method** sur **Disabled**.
- Pour utiliser DHCP, laissez le champ **Method** à sa valeur par défaut, **Automatic (DHCP)**.
- Pour utiliser des paramètres IP statiques, réglez le champ **Method** sur **Manual** et remplissez les champs en conséquence :

The image shows two side-by-side screenshots of the 'nm-connection-editor' window, both titled 'Editing Team connection 1'. The left window shows the 'IPv4 Settings' tab. The 'Method' is set to 'Manual'. The 'Addresses' table has one entry: Address '192.0.2.1', Netmask '24', and Gateway '192.0.2.254'. The 'DNS servers' field contains '192.0.2.253' and the 'Search domains' field contains 'example.com'. The right window shows the 'IPv6 Settings' tab. The 'Method' is also set to 'Manual'. The 'Addresses' table has one entry: Address '2001:db8:1::1', Prefix '64', and Gateway '2001:db8:1::fff3'. The 'DNS servers' field contains '2001:db8:1::fffd' and the 'Search domains' field contains 'example.com'.

6. Cliquez sur **Enregistrer**.

7. Fermer **nm-connection-editor**.

Vérification

- Afficher le statut de l'équipe :

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

Ressources supplémentaires

- [Configuration d'une liaison réseau à l'aide de nm-connection-editor](#)

- Configuration d'une équipe réseau à l'aide de nm-connection-editor
- Configuration du marquage VLAN à l'aide de nm-connection-editor
- Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut
- Comprendre le service teamd, les runners et les link-watchers
- NetworkManager duplique une connexion après le redémarrage du service NetworkManager

CHAPITRE 5. CONFIGURATION DU MARQUAGE VLAN

Un réseau local virtuel (VLAN) est un réseau logique au sein d'un réseau physique. L'interface VLAN marque les paquets avec l'ID VLAN lorsqu'ils passent par l'interface, et supprime les marques des paquets qui reviennent. Vous créez des interfaces VLAN au-dessus d'une autre interface, telle que des périphériques Ethernet, bond, team ou bridge. Ces interfaces sont appelées **parent interface**.

Red Hat Enterprise Linux offre aux administrateurs différentes options pour configurer les périphériques VLAN. Par exemple :

- Utilisez **nmcli** pour configurer le marquage VLAN à l'aide de la ligne de commande.
- Utilisez la console web RHEL pour configurer le marquage VLAN à l'aide d'un navigateur web.
- Utilisez **nmtui** pour configurer le marquage VLAN dans une interface utilisateur textuelle.
- Utilisez l'application **nm-connection-editor** pour configurer les connexions dans une interface graphique.
- Utilisez **nmstatectl** pour configurer les connexions via l'API Nmstate.
- Utilisez les rôles système RHEL pour automatiser la configuration du VLAN sur un ou plusieurs hôtes.

5.1. CONFIGURATION DU MARQUAGE DES VLAN À L'AIDE DE NMCLI

Vous pouvez configurer le marquage des réseaux locaux virtuels (VLAN) sur la ligne de commande à l'aide de l'utilitaire **nmcli**.

Conditions préalables

- L'interface que vous prévoyez d'utiliser comme parent de l'interface VLAN virtuelle prend en charge les balises VLAN.
- Si vous configurez le VLAN au-dessus d'une interface de liaison :
 - Les ports de la liaison sont en place.
 - La liaison n'est pas configurée avec l'option **fail_over_mac=follow**. Un périphérique virtuel VLAN ne peut pas modifier son adresse MAC pour qu'elle corresponde à la nouvelle adresse MAC du parent. Dans ce cas, le trafic serait toujours envoyé avec l'adresse MAC source incorrecte.
 - La liaison n'est généralement pas censée obtenir des adresses IP à partir d'un serveur DHCP ou d'une auto-configuration IPv6. Assurez-vous-en en définissant les options **ipv4.method=disable** et **ipv6.method=ignore** lors de la création du lien. Sinon, si le DHCP ou l'auto-configuration IPv6 échoue au bout d'un certain temps, l'interface risque d'être mise hors service.
- Le commutateur auquel l'hôte est connecté est configuré pour prendre en charge les balises VLAN. Pour plus de détails, consultez la documentation de votre commutateur.

Procédure

1. Affiche les interfaces réseau :

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

2. Créez l'interface VLAN. Par exemple, pour créer une interface VLAN nommée **vlan10** qui utilise **enp1s0** comme interface parent et qui marque les paquets avec l'ID VLAN **10**, entrez :

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

Notez que le VLAN doit être compris entre **0** et **4094**.

3. Par défaut, la connexion VLAN hérite de l'unité de transmission maximale (MTU) de l'interface mère. Il est possible de définir une valeur MTU différente :

```
# nmcli connection modify vlan10 ethernet.mtu 2000
```

4. Configurez les paramètres IPv4 :

- Pour utiliser ce périphérique VLAN comme port d'autres périphériques, entrez :

```
# nmcli connection modify vlan10 ipv4.method disabled
```

- Pour utiliser le DHCP, aucune action n'est nécessaire.
- Pour définir une adresse IPv4 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **vlan10**, entrez :

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

5. Configurez les paramètres IPv6 :

- Pour utiliser ce périphérique VLAN comme port d'autres périphériques, entrez :

```
# nmcli connection modify vlan10 ipv6.method disabled
```

- Pour utiliser le DHCP, aucune action n'est nécessaire.
- Pour définir une adresse IPv6 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **vlan10**, entrez :

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

6. Activer la connexion :

```
# nmcli connection up vlan10
```

Vérification

- Vérifiez les paramètres :

```
# ip -d addr show vlan10
4: vlan10@enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default qlen 1000
    link/ether 52:54:00:72:2f:6e brd ff:ff:ff:ff:ff:ff promiscuity 0
    vlan protocol 802.1Q id 10 <REORDER_HDR> numtxqueues 1 numrxqueues 1
gso_max_size 65536 gso_max_segs 65535
inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute vlan10
    valid_lft forever preferred_lft forever
inet6 2001:db8:1::1/32 scope global noprefixroute
    valid_lft forever preferred_lft forever
inet6 fe80::8dd7:9030:6f8e:89e6/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)
- [nm-settings\(5\)](#) page de manuel

5.2. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DE LA CONSOLE WEB RHEL

Utilisez la console web RHEL pour configurer le marquage VLAN si vous préférez gérer les paramètres du réseau à l'aide d'une interface basée sur un navigateur web.

Conditions préalables

- L'interface que vous prévoyez d'utiliser comme parent de l'interface VLAN virtuelle prend en charge les balises VLAN.
- Si vous configurez le VLAN au-dessus d'une interface de liaison :
 - Les ports de la liaison sont en place.
 - La liaison n'est pas configurée avec l'option **fail_over_mac=follow**. Un périphérique virtuel VLAN ne peut pas modifier son adresse MAC pour qu'elle corresponde à la nouvelle adresse MAC du parent. Dans ce cas, le trafic serait toujours envoyé avec l'adresse MAC source incorrecte.
 - Le lien n'est généralement pas censé obtenir des adresses IP à partir d'un serveur DHCP ou d'une auto-configuration IPv6. Assurez-vous que c'est le cas en désactivant les protocoles IPv4 et IPv6 qui créent le lien. Sinon, si la configuration automatique DHCP ou IPv6 échoue au bout d'un certain temps, l'interface risque d'être mise hors service.
- Le commutateur auquel l'hôte est connecté est configuré pour prendre en charge les balises VLAN. Pour plus de détails, consultez la documentation de votre commutateur.

Procédure

1. Sélectionnez l'onglet **Networking** dans le menu de navigation situé à gauche de l'écran.
2. Cliquez sur **Add VLAN** dans la section **Interfaces**.

- Sélectionnez l'appareil parent.
- Saisissez l'ID VLAN.
- Saisissez le nom du périphérique VLAN ou conservez le nom généré automatiquement.

VLAN settings ✕

Parent

VLAN ID

Name

- Cliquez sur **Appliquer**.
- Par défaut, le dispositif VLAN utilise une adresse IP dynamique. Si vous souhaitez définir une adresse IP statique :
 - Cliquez sur le nom du périphérique VLAN dans la section **Interfaces**.
 - Cliquez sur **Edit** en regard du protocole que vous souhaitez configurer.
 - Sélectionnez **Manual** à côté de **Addresses**, et entrez l'adresse IP, le préfixe et la passerelle par défaut.
 - Dans la section **DNS**, cliquez sur le bouton et entrez l'adresse IP du serveur DNS. Répétez cette étape pour définir plusieurs serveurs DNS.
 - Dans la section **DNS search domains**, cliquez sur le bouton et entrez le domaine de recherche.
 - Si l'interface nécessite des routes statiques, configurez-les dans la section **Routes**.

IPv4 settings ✕

Addresses Manual ▾ +

Address	Prefix length or netmask	Gateway	-
<input type="text" value="192.0.2.1"/>	<input type="text" value="24"/>	<input type="text" value="192.0.2.254"/>	-

DNS Automatic +

Server -

DNS search domains Automatic +

Search domain -

Routes Automatic +

Apply Cancel

g. Cliquez sur **Appliquer**

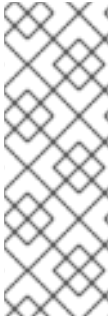
Vérification

- Sélectionnez l'onglet **Networking** dans la navigation sur le côté gauche de l'écran, et vérifiez s'il y a du trafic entrant et sortant sur l'interface :

Interfaces Add bond Add team Add bridge Add VLAN 				
Name	IP address	Sending	Receiving	
enp1s0.10	192.0.2.1/24	1.11 Mbps	61.2 Mbps	

5.3. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DE NMTUI

L'application **nmtui** fournit une interface utilisateur textuelle pour NetworkManager. Vous pouvez utiliser **nmtui** pour configurer le marquage VLAN sur un hôte sans interface graphique.



NOTE

Sur **nmtui**:

- Naviguer à l'aide des touches du curseur.
- Appuyez sur un bouton en le sélectionnant et en appuyant sur **Entrée**.
- Sélectionnez et désélectionnez les cases à cocher en utilisant l'**espace**.

Conditions préalables

- L'interface que vous prévoyez d'utiliser comme parent de l'interface VLAN virtuelle prend en charge les balises VLAN.
- Si vous configurez le VLAN au-dessus d'une interface de liaison :
 - Les ports de la liaison sont en place.
 - La liaison n'est pas configurée avec l'option **fail_over_mac=follow**. Un périphérique virtuel VLAN ne peut pas modifier son adresse MAC pour qu'elle corresponde à la nouvelle adresse MAC du parent. Dans ce cas, le trafic serait toujours envoyé avec l'adresse MAC source alors incorrecte.
 - La liaison n'est généralement pas censée obtenir des adresses IP à partir d'un serveur DHCP ou d'une auto-configuration IPv6. Assurez-vous-en en définissant les options **ipv4.method=disable** et **ipv6.method=ignore** lors de la création du lien. Sinon, si le DHCP ou l'auto-configuration IPv6 échoue au bout d'un certain temps, l'interface risque d'être mise hors service.
- Le commutateur auquel l'hôte est connecté est configuré pour prendre en charge les balises VLAN. Pour plus de détails, consultez la documentation de votre commutateur.

Procédure

1. Si vous ne connaissez pas le nom du périphérique réseau sur lequel vous souhaitez configurer le marquage VLAN, affichez les périphériques disponibles :

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

2. Démarrer **nmtui**:

```
# nmtui
```

3. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
4. Appuyez sur le bouton **Add**.
5. Sélectionnez **VLAN** dans la liste des types de réseaux et appuyez sur **Entrée**.
6. Optionnel : Entrez un nom pour le profil NetworkManager à créer.
7. Saisissez le nom du périphérique VLAN à créer dans le champ **Device**.

8. Saisissez le nom de l'appareil sur lequel vous souhaitez configurer le marquage VLAN dans le champ **Parent**.
9. Saisissez l'ID du VLAN. L'ID doit être compris entre **0** et **4094**.
10. En fonction de votre environnement, configurez les paramètres de l'adresse IP dans les zones **IPv4 configuration** et **IPv6 configuration**. Pour ce faire, appuyez sur la touche **Automatic** et sélectionnez :
 - **Disabled** si ce périphérique VLAN n'a pas besoin d'une adresse IP ou si vous voulez l'utiliser comme port d'autres périphériques.
 - **Automatic** si un serveur DHCP attribue dynamiquement une adresse IP à l'appareil du VLAN.
 - **Manual** si le réseau nécessite des paramètres d'adresse IP statiques. Dans ce cas, vous devez remplir d'autres champs :
 - i. Appuyez sur le bouton **Show** en regard du protocole que vous souhaitez configurer pour afficher des champs supplémentaires.
 - ii. Appuyez sur le bouton **Add** à côté de **Addresses**, et entrez l'adresse IP et le masque de sous-réseau au format CIDR (Classless Inter-Domain Routing).
Si vous ne spécifiez pas de masque de sous-réseau, NetworkManager définit un masque de sous-réseau **/32** pour les adresses IPv4 et **/64** pour les adresses IPv6.
 - iii. Saisissez l'adresse de la passerelle par défaut.
 - iv. Appuyez sur la touche **Add** à côté de **DNS servers**, et entrez l'adresse du serveur DNS.
 - v. Appuyez sur la touche **Add** à côté de **Search domains**, et entrez le domaine de recherche DNS.

Figure 5.1. Exemple d'une connexion VLAN avec des paramètres d'adresse IP statiques

The screenshot shows the 'Edit Connection' window for a connection named 'vlan10'. The configuration is as follows:

- Profile name:** vlan10
- Device:** vlan10
- VLAN:**
 - Parent: enp1s0
 - VLAN id: 10
 - Cloned MAC address: (empty)
 - MTU: (default)
- IPv4 CONFIGURATION <Manual>**
 - Addresses: 192.0.2.1/24 (with <Remove> and <Add...> buttons)
 - Gateway: 192.0.2.254
 - DNS servers: 192.0.2.253 (with <Remove> and <Add...> buttons)
 - Search domains: <Add...>
 - Routing (No custom routes) <Edit...>
 - Never use this network for default route
 - Ignore automatically obtained routes
 - Ignore automatically obtained DNS parameters
 - Require IPv4 addressing for this connection
- IPv6 CONFIGURATION <Manual>**
 - Addresses: 2001:db8:1::1/32 (with <Remove> and <Add...> buttons)
 - Gateway: 2001:db8:1::ffff
 - DNS servers: 2001:db8:1::fffd (with <Remove> and <Add...> buttons)
 - Search domains: <Add...>
 - Routing (No custom routes) <Edit...>
 - Never use this network for default route
 - Ignore automatically obtained routes
 - Ignore automatically obtained DNS parameters
 - Require IPv6 addressing for this connection
- Automatically connect
- Available to all users

Buttons at the bottom right: <Cancel> and <OK>

11. Appuyez sur le bouton **OK** pour créer et activer automatiquement la nouvelle connexion.
12. Appuyez sur le bouton **Back** pour revenir au menu principal.
13. Sélectionnez **Quit** et appuyez sur **Entrée** pour fermer l'application **nmtui**.

Vérification

- Vérifiez les paramètres :

```
# ip -d addr show vlan10
```

```
4: vlan10@enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue
state UP group default qlen 1000
    link/ether 52:54:00:72:2f:6e brd ff:ff:ff:ff:ff:ff promiscuity 0
    vlan protocol 802.1Q id 10 <REORDER_HDR> numtxqueues 1 numrxqueues 1
gso_max_size 65536 gso_max_segs 65535
    inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute vlan10
        valid_lft forever preferred_lft forever
    inet6 2001:db8:1::1/32 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::8dd7:9030:6f8e:89e6/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

5.4. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DE NM-CONNECTION-EDITOR

Vous pouvez configurer le marquage des réseaux locaux virtuels (VLAN) dans une interface graphique à l'aide de l'application **nm-connection-editor**.


Conditions préalables

- L'interface que vous prévoyez d'utiliser comme parent de l'interface VLAN virtuelle prend en charge les balises VLAN.
- Si vous configurez le VLAN au-dessus d'une interface de liaison :
 - Les ports de la liaison sont en place.
 - La liaison n'est pas configurée avec l'option **fail_over_mac=follow**. Un périphérique virtuel VLAN ne peut pas modifier son adresse MAC pour qu'elle corresponde à la nouvelle adresse MAC du parent. Dans ce cas, le trafic serait toujours envoyé avec l'adresse MAC source incorrecte.
- Le commutateur auquel l'hôte est connecté est configuré pour prendre en charge les balises VLAN. Pour plus de détails, consultez la documentation de votre commutateur.

Procédure

1. Ouvrez un terminal et entrez **nm-connection-editor**:

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

2. Cliquez sur le bouton  pour ajouter une nouvelle connexion.
3. Sélectionnez le type de connexion **VLAN** et cliquez sur **Créer**.
4. Dans l'onglet **VLAN**:
 - a. Sélectionnez l'interface parent.
 - b. Sélectionnez l'identifiant du VLAN. Notez que le VLAN doit être compris entre **0** et **4094**.
 - c. Par défaut, la connexion VLAN hérite de l'unité de transmission maximale (MTU) de l'interface parent. Il est possible de définir une valeur MTU différente.

- d. En option, définir le nom de l'interface VLAN et d'autres options spécifiques au VLAN.

Editing VLAN connection 1

Connection name:

General **VLAN** Proxy IPv4 Settings IPv6 Settings

Parent interface:

VLAN id: - +

VLAN interface name:

Cloned MAC address:

MTU: - + bytes

Flags: Reorder headers GVRP Loose binding MVRP

5. Configurez les paramètres de l'adresse IP dans les onglets **IPv4 Settings** et **IPv6 Settings**:

- Pour utiliser ce dispositif de pont comme port d'autres dispositifs, réglez le champ **Method** sur **Disabled**.
- Pour utiliser DHCP, laissez le champ **Method** à sa valeur par défaut, **Automatic (DHCP)**.
- Pour utiliser des paramètres IP statiques, réglez le champ **Method** sur **Manual** et remplissez les champs en conséquence :

Editing VLAN connection 1

Connection name:

General VLAN Proxy **IPv4 Settings** IPv6 Settings

Method:

Addresses

Address	Netmask	Gateway
192.0.2.1	24	192.0.2.254

DNS servers:

Editing VLAN connection 1

Connection name:

General VLAN Proxy IPv4 Settings **IPv6 Settings**

Method:

Addresses

Address	Prefix	Gateway
2001:db8:1::1	64	2001:db8:1::ffff3

DNS servers:

6. Cliquez sur **Enregistrer**.

7. Fermer **nm-connection-editor**.

Vérification

1. Vérifiez les paramètres :

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



```
valid_lft forever preferred_lft forever
inet6 fe80::8dd7:9030:6f8e:89e6/64 scope link noprefixroute
valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)

5.5. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DE NMSTATECTL

Vous pouvez utiliser l'utilitaire **nmstatectl** pour configurer le marquage des réseaux locaux virtuels (VLAN). Cet exemple configure un VLAN avec l'ID 10 qui utilise une connexion Ethernet. En tant que périphérique enfant, la connexion VLAN contient les configurations IP, passerelle par défaut et DNS.

En fonction de votre environnement, adaptez le fichier YAML en conséquence. Par exemple, pour utiliser un pont ou un dispositif de liaison dans le VLAN, adaptez les attributs **base-iface** et **type** des ports que vous utilisez dans le VLAN.

Conditions préalables

- Pour utiliser des périphériques Ethernet comme ports dans le VLAN, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.
- Le paquet **nmstate** est installé.

Procédure

1. Créez un fichier YAML, par exemple **~/create-vlan.yml**, avec le contenu suivant :

```
---
interfaces:
- name: vlan10
  type: vlan
  state: up
  ipv4:
    enabled: true
    address:
    - ip: 192.0.2.1
      prefix-length: 24
    dhcp: false
  ipv6:
    enabled: true
    address:
    - ip: 2001:db8:1::1
      prefix-length: 64
    autoconf: false
    dhcp: false
  vlan:
    base-iface: enp1s0
    id: 10
- name: enp1s0
  type: ethernet
```

```

state: up

routes:
  config:
    - destination: 0.0.0.0/0
      next-hop-address: 192.0.2.254
      next-hop-interface: vlan10
    - destination: ::0
      next-hop-address: 2001:db8:1::fffe
      next-hop-interface: vlan10

dns-resolver:
  config:
    search:
      - example.com
    server:
      - 192.0.2.200
      - 2001:db8:1::ffbb

```

2. Appliquer les paramètres au système :

```
# nmstatectl apply ~/create-vlan.yml
```

Vérification

1. Affiche l'état des appareils et des connexions :

```
# nmcli device status
DEVICE  TYPE  STATE  CONNECTION
vlan10  vlan  connected  vlan10
```

2. Affiche tous les paramètres du profil de connexion :

```
# nmcli connection show vlan10
connection.id:          vlan10
connection.uuid:        1722970f-788e-4f81-bd7d-a86bf21c9df5
connection.stable-id:   --
connection.type:        vlan
connection.interface-name:  vlan10
...
```

3. Affiche les paramètres de connexion au format YAML :

```
# nmstatectl show vlan0
```

Ressources supplémentaires

- [nmstatectl\(8\)](#) page de manuel
- [/usr/share/doc/nmstate/examples/](#) répertoire

5.6. CONFIGURATION DU MARQUAGE VLAN À L'AIDE DU RÔLE DE SYSTÈME RHEL DU RÉSEAU

Vous pouvez utiliser le rôle de système **network** RHEL pour configurer le marquage VLAN. Cet exemple ajoute une connexion Ethernet et un VLAN avec l'ID **10** au-dessus de cette connexion Ethernet. En tant que périphérique enfant, la connexion VLAN contient l'IP, la passerelle par défaut et les configurations DNS.

En fonction de votre environnement, ajustez le jeu en conséquence. Par exemple :

- Pour utiliser le VLAN comme port dans d'autres connexions, telles qu'un lien, omettre l'attribut **ip** et définir la configuration IP dans la configuration enfant.
- Pour utiliser des périphériques team, bridge ou bond dans le VLAN, adaptez les attributs **interface_name** et **type** des ports que vous utilisez dans le VLAN.

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.

Procédure

1. Créez un fichier playbook, par exemple `~/vlan-ethernet.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure a VLAN that uses an Ethernet connection
      include_role:
        name: rhel-system-roles.network

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

      # Define the VLAN profile
      - name: enp1s0.10
        type: vlan
        ip:
```

```
address:
  - "192.0.2.1/24"
  - "2001:db8:1::1/64"
gateway4: 192.0.2.254
gateway6: 2001:db8:1::ffe
dns:
  - 192.0.2.200
  - 2001:db8:1::ffbb
dns_search:
  - example.com
vlan_id: 10
parent: enp1s0
state: up
```

L'attribut **parent** du profil VLAN configure le VLAN pour qu'il fonctionne au-dessus du dispositif **enp1s0**.

2. Exécutez le manuel de jeu :

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

Ressources supplémentaires

- [/usr/share/ansible/roles/rhel-system-roles.network/README.md](#) fichier

5.7. RESSOURCES SUPPLÉMENTAIRES

- [VLANs pour les administrateurs système : Les bases](#)

CHAPITRE 6. CONFIGURATION D'UN PONT RÉSEAU

Un pont de réseau est un dispositif de couche de liaison qui transmet le trafic entre les réseaux sur la base d'une table d'adresses MAC. Le pont construit la table d'adresses MAC en écoutant le trafic réseau et en apprenant ainsi quels hôtes sont connectés à chaque réseau. Par exemple, vous pouvez utiliser un pont logiciel sur un hôte Red Hat Enterprise Linux pour émuler un pont matériel ou, dans des environnements de virtualisation, pour intégrer des machines virtuelles (VM) au même réseau que l'hôte.

Un pont nécessite un périphérique réseau dans chaque réseau qu'il doit connecter. Lorsque vous configurez un pont, celui-ci est appelé **controller** et les périphériques qu'il utilise **ports**.

Vous pouvez créer des passerelles sur différents types d'appareils, comme par exemple :

- Dispositifs Ethernet physiques et virtuels
- Obligations du réseau
- Équipes de réseau
- Dispositifs VLAN

En raison de la norme IEEE 802.11 qui spécifie l'utilisation de trames à trois adresses en Wi-Fi pour une utilisation efficace du temps d'antenne, vous ne pouvez pas configurer un pont sur des réseaux Wi-Fi fonctionnant en mode Ad-Hoc ou Infrastructure.

6.1. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE NMCLI

Pour configurer un pont réseau en ligne de commande, utilisez l'utilitaire **nmcli**.

Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports du pont, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.
- Pour utiliser des périphériques team, bond ou VLAN comme ports de la passerelle, vous pouvez soit créer ces périphériques lors de la création de la passerelle, soit les créer à l'avance comme décrit dans la section :
 - [Configuration d'une équipe réseau à l'aide de nmcli](#)
 - [Configuration d'une liaison réseau à l'aide de nmcli](#)
 - [Configuration du marquage des VLAN à l'aide de nmcli](#)

Procédure

1. Créer une interface de pont :

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

Cette commande crée un pont nommé **bridge0**, entrez :

2. Affichez les interfaces réseau et notez les noms des interfaces que vous souhaitez ajouter au pont :

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

Dans cet exemple :

- **enp7s0** et **enp8s0** ne sont pas configurés. Pour utiliser ces dispositifs comme ports, ajoutez des profils de connexion à l'étape suivante.
- **bond0** et **bond1** ont des profils de connexion existants. Pour utiliser ces dispositifs comme ports, modifiez leurs profils à l'étape suivante.

3. Attribuer les interfaces au pont.

- a. Si les interfaces que vous souhaitez affecter au pont ne sont pas configurées, créez de nouveaux profils de connexion pour elles :

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

Ces commandes créent des profils pour **enp7s0** et **enp8s0** et les ajoutent à la connexion **bridge0**.

- b. Si vous souhaitez affecter un profil de connexion existant à la passerelle :
 - i. Réglez le paramètre **master** de ces connexions sur **bridge0**:

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

Ces commandes affectent les profils de connexion existants nommés **bond0** et **bond1** à la connexion **bridge0**.

- ii. Réactiver les connexions :

```
# nmcli connection up bond0
# nmcli connection up bond1
```

4. Configurez les paramètres IPv4 :

- Pour utiliser ce dispositif de pont comme port d'autres dispositifs, entrez :

```
# nmcli connection modify bridge0 ipv4.method disabled
```

- Pour utiliser le DHCP, aucune action n'est nécessaire.
- Pour définir une adresse IPv4 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **bridge0**, entrez :

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

5. Configurez les paramètres IPv6 :

- Pour utiliser ce dispositif de pont comme port d'autres dispositifs, entrez :

```
# nmcli connection modify bridge0 ipv6.method disabled
```

- Pour utiliser le DHCP, aucune action n'est nécessaire.
- Pour définir une adresse IPv6 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **bridge0**, entrez :

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

6. En option : Configurez d'autres propriétés du pont. Par exemple, pour définir la priorité STP (Spanning Tree Protocol) de **bridge0** à **16384**, entrez :

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

Par défaut, le protocole STP est activé.

7. Activer la connexion :

```
# nmcli connection up bridge0
```

8. Vérifiez que les ports sont connectés et que la colonne **CONNECTION** affiche le nom de connexion du port :

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

Lorsque vous activez un port de la connexion, NetworkManager active également le pont, mais pas les autres ports. Vous pouvez configurer Red Hat Enterprise Linux pour qu'il active automatiquement tous les ports lorsque le pont est activé :

- Active le paramètre **connection.autoconnect-slaves** de la connexion de pont :

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

- Réactiver le pont :

```
# nmcli connection up bridge0
```

Vérification

- Utilisez l'utilitaire **ip** pour afficher l'état des liens des périphériques Ethernet qui sont des ports d'un pont spécifique :

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

- Utilisez l'utilitaire **bridge** pour afficher l'état des périphériques Ethernet qui sont des ports de n'importe quel périphérique de pont :

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

Pour afficher l'état d'un périphérique Ethernet spécifique, utilisez la commande **bridge link show dev ethernet_device_name** pour afficher l'état d'un périphérique Ethernet spécifique.

Ressources supplémentaires

- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)
- **nm-settings(5)** page de manuel
- **bridge(8)** page de manuel
- [NetworkManager duplique une connexion après le redémarrage du service NetworkManager](#)
- [Comment configurer un pont avec des informations sur les VLAN ?](#)

6.2. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE LA CONSOLE WEB RHEL

Utilisez la console web RHEL pour configurer un pont réseau si vous préférez gérer les paramètres réseau à l'aide d'une interface basée sur un navigateur web.

Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports du pont, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.

- Pour utiliser des périphériques team, bond ou VLAN comme ports de la passerelle, vous pouvez soit créer ces périphériques lors de la création de la passerelle, soit les créer à l'avance comme décrit dans la section :
 - [Configuration d'une équipe réseau à l'aide de la console web RHEL](#)
 - [Configuration d'une liaison réseau à l'aide de la console web RHEL](#)
 - [Configuration du marquage VLAN à l'aide de la console web RHEL](#)

Procédure

1. Sélectionnez l'onglet **Networking** dans le menu de navigation situé à gauche de l'écran.
2. Cliquez sur **Add bridge** dans la section **Interfaces**.
3. Saisissez le nom du dispositif de pont que vous souhaitez créer.
4. Sélectionnez les interfaces qui doivent être des ports du pont.
5. Facultatif : Activez la fonction **Spanning tree protocol (STP)** pour éviter les boucles de pont et les radiations de diffusion.

Bridge settings ✕

Name

Ports

- enp7s0
- enp8s0

Options

- Spanning tree protocol (STP)

6. Cliquez sur **Appliquer**.
7. Par défaut, le pont utilise une adresse IP dynamique. Si vous souhaitez définir une adresse IP statique :
 - a. Cliquez sur le nom du pont dans la section **Interfaces**.
 - b. Cliquez sur **Edit** en regard du protocole que vous souhaitez configurer.
 - c. Sélectionnez **Manual** à côté de **Addresses**, et entrez l'adresse IP, le préfixe et la passerelle par défaut.
 - d. Dans la section **DNS**, cliquez sur le bouton et entrez l'adresse IP du serveur DNS. Répétez cette étape pour définir plusieurs serveurs DNS.
 - e. Dans la section **DNS search domains**, cliquez sur le bouton et entrez le domaine de recherche.

- f. Si l'interface nécessite des routes statiques, configurez-les dans la section **Routes**.

IPv4 settings ✕

Addresses Manual ▾ +

Address	Prefix length or netmask	Gateway	
<input type="text" value="192.0.2.1"/>	<input type="text" value="24"/>	<input type="text" value="192.0.2.254"/>	-

DNS Automatic +

Server -

DNS search domains Automatic +

Search domain -

Routes Automatic +

Apply Cancel

- g. Cliquez sur **Appliquer**

Vérification

1. Sélectionnez l'onglet **Networking** dans la navigation sur le côté gauche de l'écran, et vérifiez s'il y a du trafic entrant et sortant sur l'interface :

Interfaces Add bond Add team Add bridge Add VLAN 			
Name	IP address	Sending	Receiving
bridge0	192.0.2.1/24	1.11 Mbps	61.2 Mbps

6.3. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE NMTUI

L'application **nmtui** fournit une interface utilisateur textuelle pour NetworkManager. Vous pouvez utiliser **nmtui** pour configurer un pont réseau sur un hôte sans interface graphique.



NOTE

Sur **nmtui**:

- Naviguer à l'aide des touches du curseur.
- Appuyez sur un bouton en le sélectionnant et en appuyant sur **Entrée**.
- Sélectionnez et désélectionnez les cases à cocher en utilisant l'**espace**.

Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports du pont, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.

Procédure

1. Si vous ne connaissez pas les noms des périphériques réseau sur lesquels vous souhaitez configurer un pont réseau, affichez les périphériques disponibles :

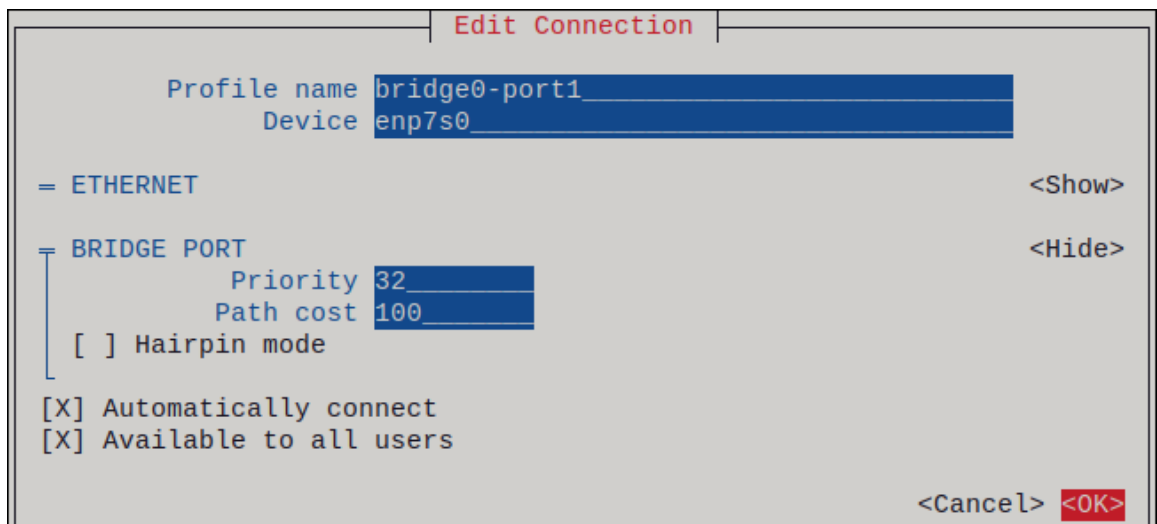
```
# nmcli device status
DEVICE  TYPE    STATE      CONNECTION
enp7s0  ethernet unavailable --
enp8s0  ethernet unavailable --
...
```

2. Démarrer **nmtui**:

```
# nmtui
```

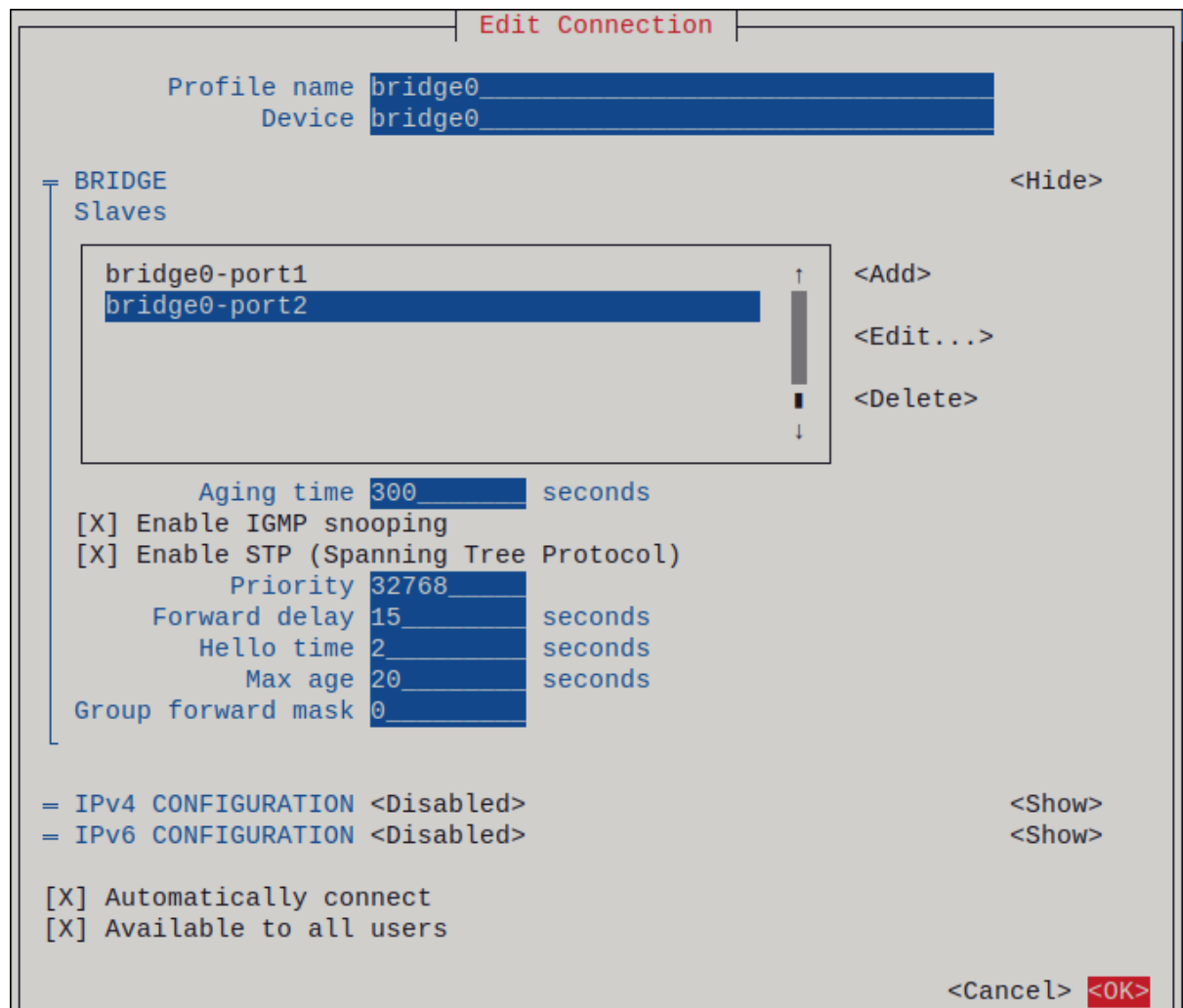
3. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
4. Appuyez sur le bouton **Add**.
5. Sélectionnez **Bridge** dans la liste des types de réseaux et appuyez sur **Entrée**.
6. Optionnel : Entrez un nom pour le profil NetworkManager à créer.
7. Saisissez le nom du dispositif de pont à créer dans le champ **Device**.
8. Ajouter des ports au pont à créer :
 - a. Appuyez sur le bouton **Add** à côté de la liste **Slaves**.
 - b. Sélectionnez le type d'interface que vous souhaitez ajouter en tant que port au pont, par exemple, **Ethernet**.
 - c. Facultatif : Entrez un nom pour le profil NetworkManager à créer pour ce port de pont.
 - d. Saisissez le nom de l'appareil du port dans le champ **Device**.
 - e. Appuyez sur le bouton **OK** pour revenir à la fenêtre des paramètres du pont.

Figure 6.1. Ajout d'un périphérique Ethernet en tant que port à un pont



- f. Répétez ces étapes pour ajouter d'autres ports à la passerelle.
9. En fonction de votre environnement, configurez les paramètres de l'adresse IP dans les zones **IPv4 configuration** et **IPv6 configuration**. Pour ce faire, appuyez sur la touche **Automatic** et sélectionnez :
- **Disabled** si le pont n'a pas besoin d'une adresse IP.
 - **Automatic** si un serveur DHCP attribue dynamiquement une adresse IP à la passerelle.
 - **Manual** si le réseau nécessite des paramètres d'adresse IP statiques. Dans ce cas, vous devez remplir d'autres champs :
 - i. Appuyez sur le bouton **Show** en regard du protocole que vous souhaitez configurer pour afficher des champs supplémentaires.
 - ii. Appuyez sur le bouton **Add** à côté de **Addresses**, et entrez l'adresse IP et le masque de sous-réseau au format CIDR (Classless Inter-Domain Routing).
Si vous ne spécifiez pas de masque de sous-réseau, NetworkManager définit un masque de sous-réseau **/32** pour les adresses IPv4 et **/64** pour les adresses IPv6.
 - iii. Saisissez l'adresse de la passerelle par défaut.
 - iv. Appuyez sur la touche **Add** à côté de **DNS servers**, et entrez l'adresse du serveur DNS.
 - v. Appuyez sur la touche **Add** à côté de **Search domains**, et entrez le domaine de recherche DNS.

Figure 6.2. Exemple de connexion par pont sans paramètres d'adresse IP



10. Appuyez sur le bouton **OK** pour créer et activer automatiquement la nouvelle connexion.
11. Appuyez sur le bouton **Back** pour revenir au menu principal.
12. Sélectionnez **Quit** et appuyez sur **Entrée** pour fermer l'application **nmtui**.

Vérification

1. Utilisez l'utilitaire **ip** pour afficher l'état des liens des périphériques Ethernet qui sont des ports d'un pont spécifique :

```
# ip link show master bridge0
```

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

2. Utilisez l'utilitaire **bridge** pour afficher l'état des périphériques Ethernet qui sont des ports de n'importe quel périphérique de pont :

```
# bridge link show
```

```
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge0 state
```

```
forwarding priority 32 cost 100
4: enp8s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 master bridge0 state
listening priority 32 cost 100
...
```

Pour afficher l'état d'un périphérique Ethernet spécifique, utilisez la commande **bridge link show dev ethernet_device_name** pour afficher l'état d'un périphérique Ethernet spécifique.

6.4. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE NM-CONNECTION-EDITOR

Si vous utilisez Red Hat Enterprise Linux avec une interface graphique, vous pouvez configurer les ponts réseau à l'aide de l'application **nm-connection-editor**.

Notez que **nm-connection-editor** ne peut ajouter que de nouveaux ports à une passerelle. Pour utiliser un profil de connexion existant comme port, créez la passerelle à l'aide de l'utilitaire **nmcli** comme décrit dans la section [Configuration d'une passerelle réseau à l'aide de nmcli](#).


Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports du pont, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.
- Pour utiliser des périphériques team, bond ou VLAN comme ports du pont, assurez-vous que ces périphériques ne sont pas déjà configurés.

Procédure

1. Ouvrez un terminal et entrez **nm-connection-editor**:

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

2. Cliquez sur le bouton  pour ajouter une nouvelle connexion.
3. Sélectionnez le type de connexion **Bridge** et cliquez sur **Créer**.
4. Dans l'onglet **Bridge**:
 - a. Facultatif : Définissez le nom de l'interface du pont dans le champ **Interface name**.
 - b. Cliquez sur le bouton **Ajouter** pour créer un nouveau profil de connexion pour une interface réseau et ajouter le profil en tant que port à la passerelle.
 - i. Sélectionnez le type de connexion de l'interface. Par exemple, sélectionnez **Ethernet** pour une connexion câblée.
 - ii. Il est possible de définir un nom de connexion pour le dispositif de port.
 - iii. Si vous créez un profil de connexion pour un périphérique Ethernet, ouvrez l'onglet **Ethernet** et sélectionnez dans le champ **Device** l'interface réseau que vous souhaitez ajouter comme port au pont. Si vous avez sélectionné un autre type de périphérique, configurez-le en conséquence.

- iv. Cliquez sur **Enregistrer**.
- c. Répétez l'étape précédente pour chaque interface que vous souhaitez ajouter au pont.

Editing Bridge connection 1

Connection name: Bridge connection 1

General **Bridge** Proxy IPv4 Settings IPv6 Settings

Interface name: bridge0

Bridged connections:

bridge0-port1	Add
bridge0-port2	

Edit

5. Facultatif : Configurez d'autres paramètres du pont, tels que les options du protocole STP (Spanning Tree Protocol).
6. Configurez les paramètres de l'adresse IP dans les onglets **IPv4 Settings** et **IPv6 Settings**:
- Pour utiliser ce dispositif de pont comme port d'autres dispositifs, réglez le champ **Method** sur **Disabled**.
 - Pour utiliser DHCP, laissez le champ **Method** à sa valeur par défaut, **Automatic (DHCP)**.
 - Pour utiliser des paramètres IP statiques, réglez le champ **Method** sur **Manual** et remplissez les champs en conséquence :

Editing Bridge connection 1

Connection name: Bridge connection 1

General Bridge Proxy **IPv4 Settings** IPv6 Settings

Method: Manual

Addresses:

Address	Netmask	Gateway
192.0.2.1	24	192.0.2.254

DNS servers: 192.0.2.1

Search domains: example.com

Editing Bridge connection 1

Connection name: Bridge connection 1

General Bridge Proxy IPv4 Settings **IPv6 Settings**

Method: Manual

Addresses:

Address	Prefix	Gateway
2001:db8:1::1	64	2001:db8:1::fff3

DNS servers: 2001:db8:1::ffff

Search domains: example.com

7. Cliquez sur **Enregistrer**.
8. Fermer **nm-connection-editor**.

Vérification

- Utilisez l'utilitaire **ip** pour afficher l'état des liens des périphériques Ethernet qui sont des ports d'un pont spécifique.

```
# ip link show master bridge0
```

```
3: enp7s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
```

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

- Utilisez l'utilitaire **bridge** pour afficher l'état des périphériques Ethernet qui sont des ports dans n'importe quel périphérique de pont :

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

Pour afficher l'état d'un périphérique Ethernet spécifique, utilisez la commande **bridge link show dev ethernet_device_name** pour afficher l'état d'un périphérique Ethernet spécifique.

Ressources supplémentaires

- [Configuration d'une liaison réseau à l'aide de nm-connection-editor](#)
- [Configuration d'une équipe réseau à l'aide de nm-connection-editor](#)
- [Configuration du marquage VLAN à l'aide de nm-connection-editor](#)
- [Configurer NetworkManager pour éviter d'utiliser un profil spécifique pour fournir une passerelle par défaut](#)
- [Comment configurer un pont avec des informations sur les VLAN ?](#)

6.5. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DE NMSTATECTL

Pour configurer un pont réseau à l'aide de l'API Nmstate, utilisez l'utilitaire **nmstatectl**.

Par exemple, la procédure ci-dessous crée un pont dans NetworkManager avec les paramètres suivants :

- Interfaces réseau dans le pont : **enp1s0** et **enp7s0**
- Protocole Spanning Tree (STP) : Activé
- Adresse IPv4 statique : **192.0.2.1** avec le masque de sous-réseau **/24**
- Adresse IPv6 statique : **2001:db8:1::1** avec le masque de sous-réseau **/64**
- Passerelle par défaut IPv4 : **192.0.2.254**
- Passerelle par défaut IPv6 : **2001:db8:1::ffe**
- Serveur DNS IPv4 : **192.0.2.200**

- Serveur DNS IPv6 : **2001:db8:1::ffbb**
- Domaine de recherche DNS : **example.com**

Conditions préalables

- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.
- Pour utiliser des périphériques Ethernet comme ports dans le pont, les périphériques Ethernet physiques ou virtuels doivent être installés sur le serveur.
- Pour utiliser des périphériques team, bond ou VLAN comme ports dans le pont, définissez le nom de l'interface dans la liste **port** et définissez les interfaces correspondantes.
- Le paquet **nmstate** est installé.

Procédure

1. Créez un fichier YAML, par exemple **~/create-bridge.yml**, avec le contenu suivant :

```
---
interfaces:
- name: bridge0
  type: linux-bridge
  state: up
  ipv4:
    enabled: true
    address:
      - ip: 192.0.2.1
        prefix-length: 24
    dhcp: false
  ipv6:
    enabled: true
    address:
      - ip: 2001:db8:1::1
        prefix-length: 64
    autoconf: false
    dhcp: false
  bridge:
    options:
      stp:
        enabled: true
  port:
    - name: enp1s0
    - name: enp7s0
- name: enp1s0
  type: ethernet
  state: up
- name: enp7s0
  type: ethernet
  state: up

routes:
  config:
    - destination: 0.0.0.0/0
      next-hop-address: 192.0.2.254
```

```

next-hop-interface: bridge0
- destination: ::0
next-hop-address: 2001:db8:1::fffe
next-hop-interface: bridge0
dns-resolver:
config:
search:
- example.com
server:
- 192.0.2.200
- 2001:db8:1::ffbb

```

2. Appliquer les paramètres au système :

```
# nmstatectl apply ~/create-bridge.yml
```

Vérification

1. Affiche l'état des appareils et des connexions :

```
# nmcli device status
DEVICE  TYPE  STATE  CONNECTION
bridge0 bridge connected bridge0
```

2. Affiche tous les paramètres du profil de connexion :

```
# nmcli connection show bridge0
connection.id:      bridge0
connection.uuid:    e2cc9206-75a2-4622-89cf-1252926060a9
connection.stable-id:  --
connection.type:    bridge
connection.interface-name: bridge0
...
```

3. Affiche les paramètres de connexion au format YAML :

```
# nmstatectl show bridge0
```

Ressources supplémentaires

- [nmstatectl\(8\)](#) page de manuel
- [/usr/share/doc/nmstate/examples/](#) répertoire
- [Comment configurer un pont avec des informations sur les vlans ?](#)

6.6. CONFIGURATION D'UN PONT RÉSEAU À L'AIDE DU RÔLE DE SYSTÈME RHEL DE RÉSEAU

Vous pouvez utiliser le rôle système **network** RHEL pour configurer un pont Linux. Par exemple, vous pouvez l'utiliser pour configurer un pont réseau qui utilise deux périphériques Ethernet et définit les adresses IPv4 et IPv6, les passerelles par défaut et la configuration DNS.



NOTE

Définir la configuration IP sur le pont et non sur les ports du pont Linux.

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- Vous avez préparé le nœud de contrôle et les nœuds gérés
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Deux ou plusieurs périphériques réseau physiques ou virtuels sont installés sur le serveur.

Procédure

1. Créez un fichier playbook, par exemple `~/bridge-ethernet.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure a network bridge that uses two Ethernet ports
      include_role:
        name: rhel-system-roles.network

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

      # Add an Ethernet profile to the bridge
      - name: bridge0-port1
        interface_name: enp7s0
        type: ethernet
```

```
controller: bridge0
port_type: bridge
state: up

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

2. Exécutez le manuel de jeu :

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

Ressources supplémentaires

- [/usr/share/ansible/roles/rhel-system-roles.network/README.md](#) fichier

CHAPITRE 7. CONFIGURATION D'UNE CONNEXION VPN

Un réseau privé virtuel (VPN) est un moyen de se connecter à un réseau local via Internet. **IPsec** fourni par **Libreswan** est la méthode préférée pour créer un VPN. **Libreswan** est une implémentation **IPsec** de l'espace utilisateur pour le VPN. Un VPN permet la communication entre votre réseau local et un autre réseau local distant en établissant un tunnel à travers un réseau intermédiaire tel que l'internet. Pour des raisons de sécurité, un tunnel VPN utilise toujours l'authentification et le cryptage. Pour les opérations cryptographiques, **Libreswan** utilise la bibliothèque **NSS**.

7.1. CONFIGURATION D'UNE CONNEXION VPN AVEC LE CENTRE DE CONTRÔLE

Si vous utilisez Red Hat Enterprise Linux avec une interface graphique, vous pouvez configurer une connexion VPN dans l'interface GNOME **control-center**.

Conditions préalables

- Le paquet **NetworkManager-libreswan-gnome** est installé.

Procédure

1. Appuyez sur la touche **Super**, tapez **Settings** et appuyez sur **Entrée** pour ouvrir l'application **control-center**.
2. Sélectionnez l'entrée **Network** sur la gauche.
3. Cliquez sur l'icône .
4. Sélectionnez **VPN**.
5. Sélectionnez l'entrée de menu **Identity** pour voir les options de configuration de base :
General

Gateway - Le nom ou l'adresse **IP** de la passerelle VPN distante.

Authentication

Type

- **IKEv2 (Certificate)**- le client est authentifié par un certificat. Il est plus sûr (par défaut).
- **IKEv1 (XAUTH)** - est authentifié par un nom d'utilisateur et un mot de passe, ou par une clé pré-partagée (PSK).

Les paramètres de configuration suivants sont disponibles dans la section **Advanced**:

Figure 7.1. Options avancées d'une connexion VPN

IPsec Advanced Options ✕

Identification

Domain:

Security

Phase1 Algorithms:

Phase2 Algorithms:

Disable PFS

Phase1 Lifetime:

Phase2 Lifetime:

Disable rekeying

Connectivity

Remote Network:

narrowing

Enable fragmentation ▼

Enable MOBIKE ▼



AVERTISSEMENT

Lors de la configuration d'une connexion VPN basée sur IPsec à l'aide de l'application **gnome-control-center**, la boîte de dialogue **Advanced** affiche la configuration, mais ne permet aucune modification. Par conséquent, les utilisateurs ne peuvent pas modifier les options IPsec avancées. Utilisez plutôt les outils **nm-connection-editor** ou **nmcli** pour configurer les propriétés avancées.

Identification

- **Domain** - Si nécessaire, saisissez le nom de domaine.

Security

- **Phase1 Algorithms** - correspond au paramètre **ike** Libreswan - entrez les algorithmes à utiliser pour l'authentification et la mise en place d'un canal crypté.
- **Phase2 Algorithms** - correspond au paramètre **esp** Libreswan - entrez les algorithmes à utiliser pour les négociations **IPsec**.
Cochez le champ **Disable PFS** pour désactiver Perfect Forward Secrecy (PFS) afin d'assurer la compatibilité avec les anciens serveurs qui ne prennent pas en charge PFS.
- **Phase1 Lifetime** - correspond au paramètre **ikelifetime** Libreswan - durée de validité de la clé utilisée pour chiffrer le trafic.
- **Phase2 Lifetime** - correspond au paramètre **salifetime** Libreswan - combien de temps une instance particulière d'une connexion doit durer avant d'expirer.
Notez que la clé de cryptage doit être modifiée de temps en temps pour des raisons de sécurité.
- **Remote network** - correspond au paramètre **rightsubnet** Libreswan - le réseau privé distant de destination qui doit être atteint par le VPN.
Cochez le champ **narrowing** pour activer le rétrécissement. Notez qu'il n'est efficace que dans la négociation IKEv2.
- **Enable fragmentation** - correspond au paramètre **fragmentation** Libreswan - autorise ou non la fragmentation IKE. Les valeurs valides sont **yes** (par défaut) ou **no**.
- **Enable Mobike** - correspond au paramètre **mobike** Libreswan - autorise ou non le protocole de mobilité et de multihébergement (MOBIKE, RFC 4555) pour permettre à une connexion de migrer son point d'extrémité sans avoir à redémarrer la connexion depuis le début. Ce protocole est utilisé sur les appareils mobiles qui passent d'une connexion câblée à une connexion sans fil ou à une connexion de données mobile. Les valeurs sont **no** (par défaut) ou **yes**.

6. Sélectionnez l'entrée de menu **IPv4**:

IPv4 Method

- **Automatic (DHCP)** - Choisissez cette option si le réseau auquel vous vous connectez utilise un serveur **DHCP** pour attribuer des adresses dynamiques **IP**.
- **Link-Local Only** - Choisissez cette option si le réseau auquel vous vous connectez ne

dispose pas d'un serveur **DHCP** et si vous ne souhaitez pas attribuer manuellement des adresses **IP**. Des adresses aléatoires seront attribuées conformément à [RFC 3927](#) avec le préfixe **169.254/16**.

- **Manual** - Choisissez cette option si vous souhaitez attribuer manuellement les adresses **IP**.
- **Disable** - **IPv4** est désactivé pour cette connexion.
DNS

Dans la section **DNS**, lorsque **Automatic** est **ON**, passez à **OFF** pour entrer l'adresse IP d'un serveur DNS que vous souhaitez utiliser en séparant les IP par une virgule.

Routes

Notez que dans la section **Routes**, lorsque **Automatic** est **ON**, les routes DHCP sont utilisées, mais vous pouvez également ajouter des routes statiques supplémentaires. Lorsque **OFF**, seules les routes statiques sont utilisées.

- **Address** - Saisissez l'adresse **IP** d'un réseau ou d'un hôte distant.
- **Netmask** - Le masque de réseau ou la longueur du préfixe de l'adresse **IP** saisie ci-dessus.
- **Gateway** - L'adresse **IP** de la passerelle menant au réseau ou à l'hôte distant saisi ci-dessus.
- **Metric** - Un coût de réseau, une valeur de préférence à donner à cet itinéraire. Les valeurs inférieures seront préférées aux valeurs supérieures.

Use this connection only for resources on its network

Cochez cette case pour éviter que la connexion ne devienne la route par défaut. En sélectionnant cette option, seul le trafic spécifiquement destiné aux itinéraires appris automatiquement sur la connexion ou saisis manuellement est acheminé sur la connexion.

7. Pour configurer les paramètres **IPv6** dans une connexion **VPN**, sélectionnez l'entrée de menu **IPv6**:

IPv6 Method

- **Automatic** - Choisissez cette option pour utiliser **IPv6** Stateless Address AutoConfiguration (SLAAC) afin de créer une configuration automatique et sans état basée sur l'adresse matérielle et les annonces de routeur (RA).
- **Automatic, DHCP only** - Choisissez cette option pour ne pas utiliser RA, mais demander directement des informations à **DHCPv6** pour créer une configuration avec état.
- **Link-Local Only** - Choisissez cette option si le réseau auquel vous vous connectez ne dispose pas d'un serveur **DHCP** et si vous ne souhaitez pas attribuer manuellement des adresses **IP**. Des adresses aléatoires seront attribuées conformément à [RFC 4862](#) avec le préfixe **FE80::0**.
- **Manual** - Choisissez cette option si vous souhaitez attribuer manuellement les adresses **IP**.
- **Disable** - **IPv6** est désactivé pour cette connexion.
Notez que **DNS**, **Routes**, **Use this connection only for resources on its network** sont communs aux paramètres de **IPv4**.

8. Une fois que vous avez fini de modifier la connexion **VPN**, cliquez sur le bouton **Ajouter** pour personnaliser la configuration ou sur le bouton **Appliquer** pour l'enregistrer dans la configuration existante.

9. Passez le profil à **ON** pour activer la connexion **VPN**.

Ressources supplémentaires

- **nm-settings-libreswan(5)**

7.2. CONFIGURATION D'UNE CONNEXION VPN À L'AIDE DE NM-CONNECTION-EDITOR

Si vous utilisez Red Hat Enterprise Linux avec une interface graphique, vous pouvez configurer une connexion VPN dans l'application **nm-connection-editor**.


Conditions préalables

- Le paquet **NetworkManager-libreswan-gnome** est installé.
- Si vous configurez une connexion IKEv2 (Internet Key Exchange version 2) :
 - Le certificat est importé dans la base de données des services de sécurité du réseau IPsec (NSS).
 - Le surnom du certificat dans la base de données du SSN est connu.

Procédure

1. Ouvrez un terminal et entrez :

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

2. Cliquez sur le bouton  pour ajouter une nouvelle connexion.
3. Sélectionnez le type de connexion **IPsec based VPN** et cliquez sur **Créer**.
4. Dans l'onglet **VPN**:
 - a. Saisissez le nom d'hôte ou l'adresse IP de la passerelle VPN dans le champ **Gateway** et sélectionnez un type d'authentification. En fonction du type d'authentification, vous devez saisir différentes informations supplémentaires :
 - **IKEv2 (Certifiate)** authentifie le client à l'aide d'un certificat, ce qui est plus sûr. Ce paramètre requiert le surnom du certificat dans la base de données IPsec NSS
 - **IKEv1 (XAUTH)** authentifie l'utilisateur à l'aide d'un nom d'utilisateur et d'un mot de passe (clé prépartagée). Ce paramètre exige que vous saisissiez les valeurs suivantes :
 - Nom de l'utilisateur
 - Mot de passe
 - Nom du groupe
 - Secret
 - b. Si le serveur distant spécifie un identifiant local pour l'échange IKE, saisissez la chaîne exacte dans le champ **Remote ID**. Si le serveur distant utilise Libreswan, cette valeur est définie dans le paramètre **leftid** du serveur.

Editing VPN connection 1 [X]

Connection name:

General | **VPN** | Proxy | IPv4 Settings

General


Gateway:

Authentication

Type:

Certificate name:

Remote ID:

 **Advanced...**

c. En option, vous pouvez configurer des paramètres supplémentaires en cliquant sur le bouton **Avancé**. Vous pouvez configurer les paramètres suivants :

- Identification
 - **Domain** - Si nécessaire, entrez le nom de domaine.
- Sécurité
 - **Phase1 Algorithms** correspond au paramètre **ike** Libreswan. Entrez les algorithmes à utiliser pour l'authentification et la mise en place d'un canal crypté.
 - **Phase2 Algorithms** correspond au paramètre **esp** Libreswan. Entrez les algorithmes à utiliser pour les négociations **IPsec**.
Cochez le champ **Disable PFS** pour désactiver Perfect Forward Secrecy (PFS) afin d'assurer la compatibilité avec les anciens serveurs qui ne prennent pas en charge PFS.
 - **Phase1 Lifetime** correspond au paramètre **ikelifetime** Libreswan. Ce paramètre définit la durée de validité de la clé utilisée pour chiffrer le trafic.
 - **Phase2 Lifetime** correspond au paramètre **salifetime** Libreswan. Ce paramètre définit la durée de validité d'une association de sécurité.
- Connectivité


- **Remote network** correspond au paramètre **rightsubnet** Libreswan et définit le réseau privé distant de destination qui doit être atteint par le VPN. Cochez le champ **narrowing** pour activer le rétrécissement. Notez qu'il n'est effectif que dans la négociation IKEv2.
 - **Enable fragmentation** correspond au paramètre **fragmentation** Libreswan et définit si la fragmentation IKE doit être autorisée ou non. Les valeurs valides sont **yes** (par défaut) ou **no**.
 - **Enable Mobike** correspond au paramètre **mobike** Libreswan. Ce paramètre définit s'il faut autoriser le protocole de mobilité et de multihébergement (MOBIKE) (RFC 4555) pour permettre à une connexion de migrer son point d'extrémité sans avoir à redémarrer la connexion à partir de zéro. Ce protocole est utilisé sur les appareils mobiles qui passent d'une connexion câblée à une connexion sans fil ou à une connexion de données mobile. Les valeurs sont **no** (par défaut) ou **yes**.
5. Dans l'onglet **IPv4 Settings**, sélectionnez la méthode d'attribution d'IP et, éventuellement, définissez des adresses statiques supplémentaires, des serveurs DNS, des domaines de recherche et des itinéraires.

The screenshot shows a window titled "Editing VPN connection 1" with a close button (X) in the top right corner. The window has a tabbed interface with four tabs: "General", "VPN", "Proxy", and "IPv4 Settings". The "IPv4 Settings" tab is selected and highlighted with a blue underline. Below the tabs, there is a "Method:" label followed by a dropdown menu showing "Automatic (VPN)". Underneath, there is a section titled "Additional static addresses" containing a table with three columns: "Address", "Netmask", and "Gateway". To the right of the table are two buttons: "Add" and "Delete". Below the table, there are two input fields: "Additional DNS servers:" and "Additional search domains:". At the bottom right of the window is a button labeled "Routes...".

6. Sauvegarder la connexion.
7. Fermer **nm-connection-editor**.



NOTE

Lorsque vous ajoutez une nouvelle connexion en cliquant sur le bouton , **NetworkManager** crée un nouveau fichier de configuration pour cette connexion et ouvre ensuite la même boîte de dialogue que celle utilisée pour modifier une connexion existante. La différence entre ces boîtes de dialogue est qu'un profil de connexion existant possède une entrée de menu **Details**.

- [nm-settings-libreswan\(5\)](#) page de manuel

7.3. CONFIGURATION DE LA DÉTECTION AUTOMATIQUE ET DE L'UTILISATION DE LA DÉCHARGE MATÉRIELLE ESP POUR ACCÉLÉRER UNE CONNEXION IPSEC

Le déchargement de l'Encapsulating Security Payload (ESP) vers le matériel accélère les connexions IPsec sur Ethernet. Par défaut, Libreswan détecte si le matériel prend en charge cette fonctionnalité et, par conséquent, active le délestage matériel de l'ESP. Dans le cas où la fonctionnalité a été désactivée ou explicitement activée, vous pouvez revenir à la détection automatique.

Conditions préalables

- La carte réseau prend en charge la décharge matérielle ESP.
- Le pilote de réseau prend en charge la décharge matérielle ESP.
- La connexion IPsec est configurée et fonctionne.

Procédure

1. Modifiez le fichier de configuration Libreswan dans le répertoire **/etc/ipsec.d/** de la connexion qui doit utiliser la détection automatique de la prise en charge du délestage matériel ESP.
2. Assurez-vous que le paramètre **nic-offload** n'est pas défini dans les paramètres de la connexion.
3. Si vous avez supprimé **nic-offload**, redémarrez le service **ipsec**:

```
# systemctl restart ipsec
```

Vérification

Si la carte réseau prend en charge la décharge matérielle ESP, procédez comme suit pour vérifier le résultat :

1. Affiche les compteurs **tx_ipsec** et **rx_ipsec** du périphérique Ethernet utilisé par la connexion IPsec :

```
# ethtool -S enp1s0 | egrep "_ipsec"  
tx_ipsec: 10  
rx_ipsec: 10
```

2. Envoyer du trafic à travers le tunnel IPsec. Par exemple, envoyer un ping à une adresse IP distante :

```
# ping -c 5 remote_ip_address
```

3. Affichez à nouveau les compteurs **tx_ipsec** et **rx_ipsec** de l'appareil Ethernet :

```
# ethtool -S enp1s0 | egrep "_ipsec"  
tx_ipsec: 15  
rx_ipsec: 15
```

Si les valeurs des compteurs ont augmenté, le délestage matériel ESP fonctionne.

Ressources supplémentaires

- [Configuration d'un VPN avec IPsec](#)

7.4. CONFIGURATION DE LA DÉCHARGE MATÉRIELLE ESP SUR UNE LIAISON POUR ACCÉLÉRER UNE CONNEXION IPSEC

Le déchargement de la charge utile d'encapsulation de sécurité (ESP) vers le matériel accélère les connexions IPsec. Si vous utilisez un lien réseau pour des raisons de basculement, les exigences et la procédure de configuration du déchargement matériel de l'ESP sont différentes de celles qui utilisent un périphérique Ethernet ordinaire. Par exemple, dans ce scénario, vous activez la prise en charge du délestage sur le lien et le noyau applique les paramètres aux ports du lien.

Conditions préalables

- Toutes les cartes réseau de la liaison prennent en charge la décharge matérielle ESP.
- Le pilote réseau prend en charge le délestage matériel ESP sur un périphérique de liaison. Dans RHEL, seul le pilote **ixgbe** prend en charge cette fonctionnalité.
- La liaison est configurée et fonctionne.
- La liaison utilise le mode **active-backup**. Le pilote de liaison ne prend pas en charge d'autres modes pour cette fonctionnalité.
- La connexion IPsec est configurée et fonctionne.

Procédure

1. Activer la prise en charge de la décharge matérielle ESP sur la liaison réseau :

```
# nmcli connection modify bond0 ethtool.feature-esp-hw-offload on
```

Cette commande active la prise en charge du délestage matériel ESP sur la connexion **bond0**.

2. Réactiver la connexion **bond0**:

```
# nmcli connection up bond0
```

3. Modifiez le fichier de configuration de Libreswan dans le répertoire **/etc/ipsec.d/** de la connexion qui doit utiliser le délestage matériel ESP, et ajoutez l'instruction **nic-offload=yes** à l'entrée de la connexion :

```
conn example
...
nic-offload=yes
```

4. Redémarrez le service **ipsec**:

```
# systemctl restart ipsec
```

Verification

1. Affiche le port actif de la liaison :

```
# grep "Currently Active Slave" /proc/net/bonding/bond0  
Currently Active Slave: enp1s0
```

2. Affiche les compteurs **tx_ipsec** et **rx_ipsec** du port actif :

```
# ethtool -S enp1s0 | egrep "_ipsec"  
tx_ipsec: 10  
rx_ipsec: 10
```

3. Envoyer du trafic à travers le tunnel IPsec. Par exemple, envoyer un ping à une adresse IP distante :

```
# ping -c 5 remote_ip_address
```

4. Affichez à nouveau les compteurs **tx_ipsec** et **rx_ipsec** du port actif :

```
# ethtool -S enp1s0 | egrep "_ipsec"  
tx_ipsec: 15  
rx_ipsec: 15
```

Si les valeurs des compteurs ont augmenté, le délestage matériel ESP fonctionne.

Ressources supplémentaires

- [Configuration de la liaison réseau](#)
- [Configuration d'un VPN avec IPsec](#) dans le document Sécurisation des réseaux

CHAPITRE 8. MISE EN PLACE D'UN VPN WIREGUARD

WireGuard est une solution VPN très performante qui fonctionne dans le noyau Linux. Elle utilise une cryptographie moderne et est plus facile à configurer que beaucoup d'autres solutions VPN. En outre, la petite base de code de WireGuard réduit la surface d'attaque et, par conséquent, améliore la sécurité. Pour l'authentification et le cryptage, WireGuard utilise des clés similaires à celles de SSH.



IMPORTANT

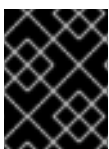
WireGuard est fourni en tant qu'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat, peuvent ne pas être complètes sur le plan fonctionnel et Red Hat ne recommande pas de les utiliser pour la production. Ces aperçus offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Consultez la section [Portée de l'assistance](#) pour les fonctionnalités de l'aperçu technologique sur le portail client de Red Hat pour obtenir des informations sur la portée de l'assistance pour les fonctionnalités de l'aperçu technologique.

Pour configurer un VPN WireGuard, vous devez effectuer les étapes suivantes. Vous pouvez effectuer la plupart des étapes en utilisant différentes options :

1. [Créez des clés publiques et privées pour chaque hôte du VPN](#) .
2. Configurez le serveur WireGuard en utilisant [nmcli](#), [nmtui](#), [nm-connection-editor](#), ou le service [wg-quick](#).
3. Configurez firewalld sur le serveur WireGuard en utilisant la [ligne de commande](#) ou l'[interface graphique](#).
4. Configurez le client WireGuard en utilisant [nmcli](#), [nm-connection-editor](#), ou le service [wg-quick](#).

WireGuard fonctionne sur la couche réseau (couche 3). Par conséquent, vous ne pouvez pas utiliser DHCP et devez attribuer des adresses IP statiques ou des adresses IPv6 link-local aux périphériques du tunnel, tant sur le serveur que sur les clients.



IMPORTANT

Vous ne pouvez utiliser WireGuard que si le mode Federal Information Processing Standard (FIPS) de RHEL est désactivé.

Notez que tous les hôtes qui participent à un VPN WireGuard sont des pairs. Cette documentation utilise les termes **client** pour décrire les hôtes qui établissent une connexion et **server** pour décrire l'hôte avec le nom d'hôte ou l'adresse IP fixe auquel les clients se connectent et acheminent éventuellement tout le trafic via ce serveur.

8.1. PROTOCOLES ET PRIMITIVES UTILISÉS PAR WIREGUARD

WireGuard utilise les protocoles et primitives suivants :

- ChaCha20 pour le chiffrement symétrique, authentifié avec Poly1305, en utilisant la construction AEAD (Authenticated Encryption with Associated Data) telle que décrite dans la [RFC7539](#)
- Curve25519 pour l'échange de clés Elliptic-curve Diffie-Hellman (ECDH)
- BLAKE2s pour le hachage et le hachage par clé, comme décrit dans [RFC7693](#)
- SipHash24 pour les clés des tables de hachage
- HKDF pour la dérivation de clé, comme décrit dans [RFC5869](#)

8.2. COMMENT WIREGUARD UTILISE LES ADRESSES IP DES TUNNELS, LES CLÉS PUBLIQUES ET LES POINTS DE TERMINAISON DISTANTS

Lorsque WireGuard envoie un paquet réseau à un pair :

1. WireGuard lit l'adresse IP de destination du paquet et la compare à la liste des adresses IP autorisées dans la configuration locale. Si le pair n'est pas trouvé, WireGuard laisse tomber le paquet.
2. Si le pair est valide, WireGuard crypte le paquet en utilisant la clé publique du pair.
3. L'hôte expéditeur recherche l'adresse IP Internet la plus récente de l'hôte et lui envoie le paquet crypté.

Lorsque WireGuard reçoit un paquet :

1. WireGuard décrypte le paquet en utilisant la clé privée de l'hôte distant.
2. WireGuard lit l'adresse source interne du paquet et vérifie si l'IP est configurée dans la liste des adresses IP autorisées dans les paramètres du pair sur l'hôte local. Si l'adresse IP source est sur la liste des adresses autorisées, WireGuard accepte le paquet. Si l'adresse IP n'est pas sur la liste, WireGuard laisse tomber le paquet.

L'association des clés publiques et des adresses IP autorisées est appelée **Cryptokey Routing Table**. Cela signifie que la liste des adresses IP se comporte comme une table de routage lors de l'envoi de paquets et comme une sorte de liste de contrôle d'accès lors de la réception de paquets.

8.3. UTILISATION D'UN CLIENT WIREGUARD DERRIÈRE UN NAT ET DES PARE-FEUX

WireGuard utilise le protocole UDP et transmet des données uniquement lorsqu'un pair envoie des paquets. Les pare-feu dynamiques et la traduction d'adresses réseau (NAT) sur les routeurs suivent les connexions pour permettre à un homologue situé derrière un NAT ou un pare-feu de recevoir des paquets.

Pour maintenir la connexion active, WireGuard supporte **persistent keepalives**. Cela signifie que vous pouvez définir un intervalle auquel WireGuard envoie des paquets keepalive. Par défaut, cette fonctionnalité est désactivée afin de réduire le trafic sur le réseau. Activez cette fonctionnalité sur le client si vous utilisez le client dans un réseau avec NAT ou si un pare-feu ferme la connexion après un certain temps d'inactivité.

8.4. CRÉATION DE CLÉS PRIVÉES ET PUBLIQUES À UTILISER DANS LES CONNEXIONS WIREGUARD

WireGuard utilise des clés privées et publiques encodées en base64 pour authentifier les hôtes entre eux. Par conséquent, vous devez créer les clés sur chaque hôte qui participe au VPN WireGuard.



IMPORTANT

Pour des connexions sécurisées, créez des clés différentes pour chaque hôte et assurez-vous que vous ne partagez la clé publique qu'avec l'hôte WireGuard distant. N'utilisez pas les clés d'exemple utilisées dans cette documentation.

Procédure

1. Installez le paquetage **wireguard-tools**:

```
# dnf install wireguard-tools
```

2. Créez une clé privée et une clé publique correspondante pour l'hôte :

```
# wg genkey | tee /etc/wireguard/$HOSTNAME.private.key | wg pubkey >
/etc/wireguard/$HOSTNAME.public.key
```

Vous aurez besoin du contenu des fichiers de clés, mais pas des fichiers eux-mêmes. Cependant, Red Hat recommande de conserver les fichiers au cas où vous auriez besoin de vous souvenir des clés à l'avenir.

3. Définir des autorisations sécurisées pour les fichiers clés :

```
# chmod 600 /etc/wireguard/$HOSTNAME.private.key
/etc/wireguard/$HOSTNAME.public.key
```

4. Afficher la clé privée :

```
# cat /etc/wireguard/$HOSTNAME.private.key
YFAnE0psgldiAF7XR4abxiwVRnlMfeltxu10s/c4JXg=
```

Vous aurez besoin de la clé privée pour configurer la connexion WireGuard sur l'hôte local. Ne partagez pas la clé privée.

5. Afficher la clé publique :

```
# cat /etc/wireguard/$HOSTNAME.public.key
UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
```

Vous aurez besoin de la clé publique pour configurer la connexion WireGuard sur l'hôte distant.

Ressources supplémentaires

- La page de manuel **wg(8)**

8.5. CONFIGURATION D'UN SERVEUR WIREGUARD À L'AIDE DE NMCLI

Vous pouvez configurer le serveur WireGuard en créant un profil de connexion dans NetworkManager. Utilisez cette méthode pour laisser NetworkManager gérer la connexion WireGuard.

Cette procédure suppose les paramètres suivants :

- Serveur :
 - Clé privée : **YFAnE0psgldiAF7XR4abxiwVRnIMfeltxu10s/c4JXg=**
 - Adresse IPv4 du tunnel : **192.0.2.1/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::1/32**
- Client :
 - Clé publique : **bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=**
 - Adresse IPv4 du tunnel : **192.0.2.2/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::2/32**

Conditions préalables

- Vous avez généré les clés publique et privée pour le serveur et le client.
- Vous connaissez les informations suivantes :
 - La clé privée du serveur
 - Les adresses IP du tunnel statique et les masques de sous-réseau du client
 - La clé publique du client
 - Les adresses IP du tunnel statique et les masques de sous-réseau du serveur

Procédure

1. Ajouter un profil de connexion NetworkManager WireGuard :

```
# nmcli connection add type wireguard con-name server-wg0 ifname wg0 autoconnect no
```

Cette commande crée un profil nommé **server-wg0** et lui affecte l'interface virtuelle **wg0**. Pour éviter que la connexion ne démarre automatiquement après l'avoir ajoutée sans finaliser la configuration, désactivez le paramètre **autoconnect**.

2. Définissez l'adresse IPv4 du tunnel et le masque de sous-réseau du serveur :

```
# nmcli connection modify server-wg0 ipv4.method manual ipv4.addresses 192.0.2.1/24
```

3. Définissez l'adresse IPv6 du tunnel et le masque de sous-réseau du serveur :

```
# nmcli connection modify server-wg0 ipv6.method manual ipv6.addresses
2001:db8:1::1/32
```

4. Ajouter la clé privée du serveur au profil de connexion :

```
# nmcli connection modify server-wg0 wireguard.private-key
"YFAnE0psgldiAF7XR4abxiwVRnlMfeltxu10s/c4JXg="
```

5. Définir le port pour les connexions WireGuard entrantes :

```
# nmcli connection modify server-wg0 wireguard.listen-port 51820
```

Définissez toujours un numéro de port fixe sur les hôtes qui reçoivent les connexions entrantes de WireGuard. Si vous ne définissez pas de port, WireGuard utilise un port libre aléatoire à chaque fois que vous activez l'interface **wg0**.

6. Ajoutez des configurations d'homologues pour chaque client que vous souhaitez autoriser à communiquer avec ce serveur. Vous devez ajouter ces paramètres manuellement, car l'utilitaire **nmcli** ne permet pas de définir les propriétés de connexion correspondantes.
 - a. Modifiez le fichier `/etc/NetworkManager/system-connections/server-wg0.nmconnection` et ajoutez :

```
[wireguard-peer.bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=]
allowed-ips=192.0.2.2;2001:db8:1::2;
```

- L'entrée `[wireguard-peer.<public_key_of_the_client>]` définit la section de l'homologue du client, et le nom de la section contient la clé publique du client.
- Le paramètre **allowed-ips** définit les adresses IP du tunnel du client qui sont autorisées à envoyer des données à ce serveur.
Ajouter une section pour chaque client.

- b. Recharger le profil de connexion **server-wg0**:

```
# nmcli connection load /etc/NetworkManager/system-connections/server-
wg0.nmconnection
```

7. Optionnel : Configurez la connexion pour qu'elle démarre automatiquement, entrez :

```
# nmcli connection modify server-wg0 autoconnect yes
```

8. Réactiver la connexion **server-wg0**:

```
# nmcli connection up server-wg0
```

Prochaines étapes

- [Configurez le service firewalld sur le serveur WireGuard](#) .

Vérification

1. Affichez la configuration de l'interface de l'appareil **wg0**:

```
# wg show wg0
interface: wg0
  public key: UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
  private key: (hidden)
  listening port: 51820

peer: bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
  allowed ips: 192.0.2.2/32, 2001:db8:1::2/128
```

Pour afficher la clé privée dans la sortie, utilisez la commande **WG_HIDE_KEYS=never wg show wg0** pour afficher la clé privée dans la sortie.

- Affichez la configuration IP de l'appareil **wg0**:

```
# ip address show wg0
20: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state
UNKNOWN group default qlen 1000
  link/none
  inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute wg0
    valid_lft forever preferred_lft forever
  inet6 2001:db8:1::1/32 scope global noprefixroute
    valid_lft forever preferred_lft forever
  inet6 fe80::3ef:8863:1ce2:844/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- La page de manuel **wg(8)**
- La section **WireGuard setting** dans la page de manuel **nm-settings(5)**

8.6. CONFIGURATION D'UN SERVEUR WIREGUARD À L'AIDE DE NMTUI

Vous pouvez configurer le serveur WireGuard en créant un profil de connexion dans NetworkManager. Utilisez cette méthode pour laisser NetworkManager gérer la connexion WireGuard.

Cette procédure suppose les paramètres suivants :

- Serveur :
 - Clé privée : **YFAnE0psgldiAF7XR4abxiwVRnlMfeltxu10s/c4JXg=**
 - Adresse IPv4 du tunnel : **192.0.2.1/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::1/32**
- Client :
 - Clé publique : **bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=**
 - Adresse IPv4 du tunnel : **192.0.2.2/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::2/32**

Conditions préalables

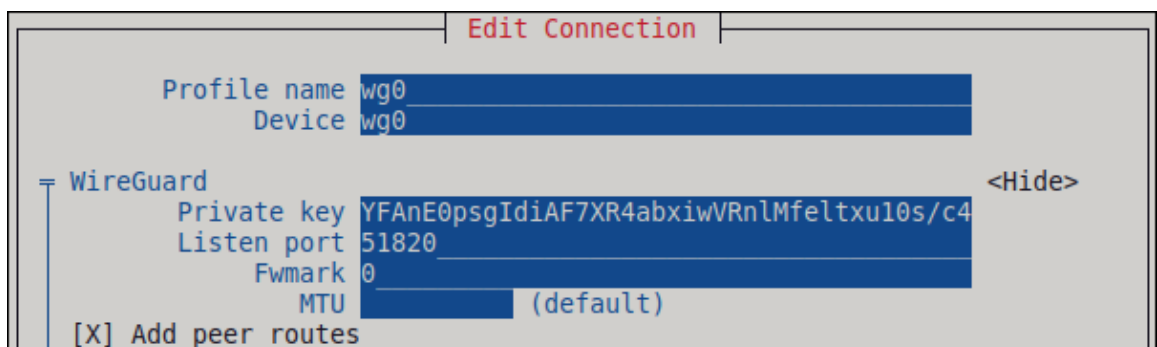
- Vous avez généré les clés publique et privée pour le serveur et le client.
- Vous connaissez les informations suivantes :
 - La clé privée du serveur
 - Les adresses IP du tunnel statique et les masques de sous-réseau du client
 - La clé publique du client
 - Les adresses IP du tunnel statique et les masques de sous-réseau du serveur
- Vous avez installé le paquetage **NetworkManager-tui**.

Procédure

1. Lancez l'application **nmtui**:

```
# nmtui
```

2. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
3. Sélectionnez **Ajouter** et appuyez sur **Entrée**.
4. Sélectionnez le type de connexion **WireGuard** dans la liste et appuyez sur **Entrée**.
5. Dans la fenêtre **Edit connection**:
 - a. Entrez le nom de la connexion et l'interface virtuelle, telle que **wg0**, que NetworkManager doit attribuer à la connexion.
 - b. Entrez la clé privée du serveur.
 - c. Définissez le numéro de port d'écoute, tel que **51820**, pour les connexions entrantes de WireGuard.
Définissez toujours un numéro de port fixe sur les hôtes qui reçoivent les connexions entrantes de WireGuard. Si vous ne définissez pas de port, WireGuard utilise un port libre aléatoire à chaque fois que vous activez l'interface.



- d. Cliquez sur **Ajouter à** côté du volet **Peers**:
 - i. Entrez la clé publique du client.
 - ii. Réglez le champ **Allowed IPs** sur les adresses IP du tunnel du client qui sont autorisées à envoyer des données à ce serveur.

- iii. Sélectionnez **OK** et appuyez sur **Entrée**.

```

Peer
Public key bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26n
Allowed IPs 192.0.2.2,2001:db8:1::2
Endpoint
Preshared key
Persistent keepalive seconds
<Cancel> <OK>
  
```

- e. Sélectionnez **Show** à côté de **IPv4 Configuration**, et appuyez sur **Enter**.
- Sélectionnez la méthode de configuration IPv4 **Manual**.
 - Saisissez l'adresse IPv4 du tunnel et le masque de sous-réseau. Laissez le champ **Gateway** vide.
- f. Sélectionnez **Show** à côté de **IPv6 Configuration**, et appuyez sur **Enter**.
- Sélectionnez la méthode de configuration IPv6 **Manual**.
 - Saisissez l'adresse IPv6 du tunnel et le masque de sous-réseau. Laissez le champ **Gateway** vide.
- g. Sélectionnez **OK** et appuyez sur **Enter**

```

IPv4 CONFIGURATION <Manual> <Hide>
Addresses 192.0.2.1/24 <Remove>
  <Add...>
Gateway
DNS servers <Add...>
Search domains <Add...>
Routing (No custom routes) <Edit...>
[ ] Never use this network for default route
[ ] Ignore automatically obtained routes
[ ] Ignore automatically obtained DNS parameters
[ ] Require IPv4 addressing for this connection

IPv6 CONFIGURATION <Manual> <Hide>
Addresses 2001:db8:1::1/32 <Remove>
  <Add...>
Gateway
DNS servers <Add...>
Search domains <Add...>
Routing (No custom routes) <Edit...>
[ ] Never use this network for default route
[ ] Ignore automatically obtained routes
[ ] Ignore automatically obtained DNS parameters
[ ] Require IPv6 addressing for this connection

[X] Automatically connect
[X] Available to all users
<Cancel> <OK>
  
```

6. Dans la fenêtre contenant la liste des connexions, sélectionnez **Retour** et appuyez sur **Entrée**.
7. Dans la fenêtre principale de **NetworkManager TUI**, sélectionnez **Quitter** et appuyez sur **Entrée**.

Prochaines étapes

- [Configurez le service firewalld sur le serveur WireGuard](#) .

Vérification

1. Affichez la configuration de l'interface de l'appareil **wg0**:

```
# wg show wg0
interface: wg0
public key: UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
private key: (hidden)
listening port: 51820

peer: bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
allowed ips: 192.0.2.2/32, 2001:db8:1::2/128
```

Pour afficher la clé privée dans la sortie, utilisez la commande **WG_HIDE_KEYS=never wg show wg0** pour afficher la clé privée dans la sortie.

2. Affichez la configuration IP de l'appareil **wg0**:

```
# ip address show wg0
20: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state
UNKNOWN group default qlen 1000
link/none
inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute wg0
    valid_lft forever preferred_lft forever
inet6 2001:db8:1::1/32 scope global noprefixroute
    valid_lft forever preferred_lft forever
inet6 fe80::3ef:8863:1ce2:844/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- La page de manuel **wg(8)**

8.7. CONFIGURATION D'UN SERVEUR WIREGUARD À L'AIDE DE NM-CONNECTION-EDITOR

Vous pouvez configurer le serveur WireGuard en créant un profil de connexion dans NetworkManager. Utilisez cette méthode pour laisser NetworkManager gérer la connexion WireGuard.

Conditions préalables

- Vous avez généré les clés publique et privée pour le serveur et le client.
- Vous connaissez les informations suivantes :

- La clé privée du serveur
- Les adresses IP du tunnel statique et les masques de sous-réseau du client
- La clé publique du client
- Les adresses IP du tunnel statique et les masques de sous-réseau du serveur

Procédure

1. Ouvrez un terminal et entrez :

```
# nm-connection-editor
```

2. Ajoutez une nouvelle connexion en cliquant sur le bouton .
3. Sélectionnez le type de connexion **WireGuard** et cliquez sur **Créer**.
4. Optionnel : Mettre à jour le nom de la connexion.
5. Dans l'onglet **General**, sélectionnez **Connect automatically with priority**. Définissez éventuellement une valeur de priorité.
6. Dans l'onglet **WireGuard**:
 - a. Entrez le nom de l'interface virtuelle, telle que **wg0**, que NetworkManager doit attribuer à la connexion.
 - b. Entrez la clé privée du serveur.
 - c. Définissez le numéro de port d'écoute, tel que **51820**, pour les connexions entrantes de WireGuard.
Définissez toujours un numéro de port fixe sur les hôtes qui reçoivent les connexions entrantes de WireGuard. Si vous ne définissez pas de port, WireGuard utilise un port libre aléatoire à chaque fois que vous activez l'interface.
 - d. Cliquez sur **Ajouter** pour ajouter des pairs :
 - i. Entrez la clé publique du client.
 - ii. Réglez le champ **Allowed IPs** sur les adresses IP du tunnel du client qui sont autorisées à envoyer des données à ce serveur.
 - iii. Cliquez sur **Appliquer**.
7. Dans l'onglet **IPv4 Settings**:
 - a. Sélectionnez **Manual** dans la liste **Method**.
 - b. Cliquez sur **Ajouter** pour entrer l'adresse IPv4 du tunnel et le masque de sous-réseau. Laissez le champ **Gateway** vide.
8. Dans l'onglet **IPv6 Settings**:
 - a. Sélectionnez **Manual** dans la liste **Method**.

- b. Cliquez sur **Ajouter** pour entrer l'adresse IPv6 du tunnel et le masque de sous-réseau. Laissez le champ **Gateway** vide.

9. Cliquez sur **Enregistrer** pour sauvegarder le profil de connexion.

Prochaines étapes

- [Configurez le service firewalld sur le serveur WireGuard](#) .

Vérification

1. Affichez la configuration de l'interface de l'appareil **wg0**:

```
# wg show wg0
interface: wg0
public key: UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
private key: (hidden)
listening port: 51820

peer: bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
allowed ips: 192.0.2.2/32, 2001:db8:1::2/128
```

Pour afficher la clé privée dans la sortie, utilisez la commande **WG_HIDE_KEYS=never wg show wg0** pour afficher la clé privée dans la sortie.

2. Affichez la configuration IP de l'appareil **wg0**:

```
# ip address show wg0
20: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state
UNKNOWN group default qlen 1000
link/none
inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute wg0
    valid_lft forever preferred_lft forever
inet6 2001:db8:1::1/32 scope global noprefixroute
    valid_lft forever preferred_lft forever
inet6 fe80::3ef:8863:1ce2:844/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- La page de manuel **wg(8)**

8.8. CONFIGURATION D'UN SERVEUR WIREGUARD À L'AIDE DU SERVICE WG-QUICK

Vous pouvez configurer le serveur WireGuard en créant un fichier de configuration dans le répertoire **/etc/wireguard/**. Cette méthode permet de configurer le service indépendamment de NetworkManager.

Cette procédure suppose les paramètres suivants :

- Serveur :
 - Clé privée : **YFAnE0psgldiAF7XR4abxiwVRnlMfeltxu10s/c4JXg=**
 - Adresse IPv4 du tunnel : **192.0.2.1/24**

- Adresse IPv6 du tunnel : **2001:db8:1::1/32**
- Client :
 - Clé publique : **bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=**
 - Adresse IPv4 du tunnel : **192.0.2.2/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::2/32**

Conditions préalables

- Vous avez généré les clés publique et privée pour le serveur et le client.
- Vous connaissez les informations suivantes :
 - La clé privée du serveur
 - Les adresses IP du tunnel statique et les masques de sous-réseau du client
 - La clé publique du client
 - Les adresses IP du tunnel statique et les masques de sous-réseau du serveur

Procédure

1. Installez le paquetage **wireguard-tools**:

```
# dnf install wireguard-tools
```

2. Créez le fichier **/etc/wireguard/wg0.conf** avec le contenu suivant :

```
[Interface]
Address = 192.0.2.1/24, 2001:db8:1::1/32
ListenPort = 51820
PrivateKey = YFAnE0psgldiAF7XR4abxiwVRnIMfeltxu10s/c4JXg=

[Peer]
PublicKey = bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
AllowedIPs = 192.0.2.2, 2001:db8:1::2
```

- La section **[Interface]** décrit les paramètres WireGuard de l'interface sur le serveur :
 - **Address**: Une liste d'adresses IP du tunnel du serveur, séparées par des virgules.
 - **PrivateKey**: La clé privée du serveur.
 - **ListenPort**: Le port sur lequel WireGuard écoute les connexions UDP entrantes. Définissez toujours un numéro de port fixe sur les hôtes qui reçoivent les connexions entrantes de WireGuard. Si vous ne définissez pas de port, WireGuard utilise un port libre aléatoire à chaque fois que vous activez l'interface **wg0**.
- Chaque section du site **[Peer]** décrit les paramètres d'un client :
 - **PublicKey**: La clé publique du client.

- **AllowedIPs**: Les adresses IP du tunnel du client qui sont autorisées à envoyer des données à ce serveur.

3. Activer et démarrer la connexion WireGuard :

```
# systemctl enable --now wg-quick@wg0
```

Le nom de l'instance systemd doit correspondre au nom du fichier de configuration dans le répertoire **/etc/wireguard/** sans le suffixe **.conf**. Le service utilise également ce nom pour l'interface réseau virtuelle.

Prochaines étapes

- [Configurez le service firewalld sur le serveur WireGuard](#) .

Vérification

1. Affichez la configuration de l'interface de l'appareil **wg0**:

```
# wg show wg0
interface: wg0
public key: UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
private key: (hidden)
listening port: 51820

peer: bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
allowed ips: 192.0.2.2/32, 2001:db8:1::2/128
```

Pour afficher la clé privée dans la sortie, utilisez la commande **WG_HIDE_KEYS=never wg show wg0** pour afficher la clé privée dans la sortie.

2. Affichez la configuration IP de l'appareil **wg0**:

```
# ip address show wg0
20: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state
UNKNOWN group default qlen 1000
link/none
inet 192.0.2.1/24 scope global wg0
valid_lft forever preferred_lft forever
inet6 2001:db8:1::1/32 scope global
valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- La page de manuel **wg(8)**
- La page de manuel **wg-quick(8)**

8.9. CONFIGURER FIREWALLD SUR UN SERVEUR WIREGUARD EN UTILISANT LA LIGNE DE COMMANDE

Vous devez configurer le service **firewalld** sur le serveur WireGuard pour autoriser les connexions entrantes des clients. De plus, si les clients doivent pouvoir utiliser le serveur WireGuard comme passerelle par défaut et acheminer tout le trafic à travers le tunnel, vous devez activer le masquage.

Procédure

1. Ouvrez le port WireGuard pour les connexions entrantes dans le service **firewalld**:

```
# firewall-cmd --permanent --add-port=51820/udp --zone=public
```

2. Si les clients doivent acheminer tout le trafic à travers le tunnel et utiliser le serveur WireGuard comme passerelle par défaut, activez le masquage pour la zone **public**:

```
# firewall-cmd --permanent --zone=public --add-masquerade
```

3. Recharger les règles de **firewalld**.

```
# firewall-cmd --reload
```

Vérification

- Affiche la configuration de la zone **public**:

```
# firewall-cmd --list-all
public (active)
...
ports: 51820/udp
masquerade: yes
...
```

Ressources supplémentaires

- La page de manuel **firewall-cmd(1)**

8.10. CONFIGURATION DE FIREWALLD SUR UN SERVEUR WIREGUARD À L'AIDE DE L'INTERFACE GRAPHIQUE

Vous devez configurer le service **firewalld** sur le serveur WireGuard pour autoriser les connexions entrantes des clients. De plus, si les clients doivent pouvoir utiliser le serveur WireGuard comme passerelle par défaut et acheminer tout le trafic à travers le tunnel, vous devez activer le masquage.

Procédure

1. Appuyez sur la touche **Super**, entrez **firewall** et sélectionnez l'application **Firewall** dans les résultats.
2. Sélectionnez **Permanent** dans la liste **Configuration**.
3. Sélectionnez la zone **public**.
4. Autoriser les connexions entrantes sur le port WireGuard :
 - a. Dans l'onglet **Ports**, cliquez sur **Ajouter**.
 - b. Entrez le numéro de port que vous avez défini pour les connexions entrantes de WireGuard :
 - c. Sélectionnez **udp** dans la liste **Protocol**.

- d. Cliquez sur **OK**.
5. Si les clients doivent acheminer tout le trafic à travers le tunnel et utiliser le serveur WireGuard comme passerelle par défaut :
 - a. Naviguez vers l'onglet **Masquerading** de la zone **public**.
 - b. Sélectionnez **Masquerade zone**.
6. Sélectionner **Options** → **Recharger Firewallld**.

Vérification

- Affiche la configuration de la zone **public**:

```
# firewall-cmd --list-all
public (active)
...
ports: 51820/udp
masquerade: yes
...
```

8.11. CONFIGURATION D'UN CLIENT WIREGUARD À L'AIDE DE NMCLI

Vous pouvez configurer un client WireGuard en créant un profil de connexion dans NetworkManager. Utilisez cette méthode pour laisser NetworkManager gérer la connexion WireGuard.

Cette procédure suppose les paramètres suivants :

- Client :
 - Clé privée : **aPUcp5vHz8yMLrzk8SsDyYnV33lhE/k20e52iKJFV0A=**
 - Adresse IPv4 du tunnel : **192.0.2.2/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::2/32**
- Serveur :
 - Clé publique : **UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=**
 - Adresse IPv4 du tunnel : **192.0.2.1/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::1/32**

Conditions préalables

- Vous avez généré les clés publique et privée pour le serveur et le client.
- Vous connaissez les informations suivantes :
 - La clé privée du client
 - Les adresses IP du tunnel statique et les masques de sous-réseau du client
 - La clé publique du serveur

- Les adresses IP du tunnel statique et les masques de sous-réseau du serveur

Procédure

1. Ajouter un profil de connexion NetworkManager WireGuard :

```
# nmcli connection add type wireguard con-name client-wg0 ifname wg0 autoconnect
no
```

Cette commande crée un profil nommé **client-wg0** et lui affecte l'interface virtuelle **wg0**. Pour éviter que la connexion ne démarre automatiquement après l'avoir ajoutée sans finaliser la configuration, désactivez le paramètre **autoconnect**.

2. Optionnel : Configurez NetworkManager pour qu'il ne démarre pas automatiquement la connexion **client-wg**:

```
# nmcli connection modify client-wg0 autoconnect no
```

3. Définissez l'adresse IPv4 du tunnel et le masque de sous-réseau du client :

```
# nmcli connection modify client-wg0 ipv4.method manual ipv4.addresses 192.0.2.2/24
```

4. Définissez l'adresse IPv6 du tunnel et le masque de sous-réseau du client :

```
# nmcli connection modify client-wg0 ipv6.method manual ipv6.addresses
2001:db8:1::2/32
```

5. Si vous souhaitez acheminer l'ensemble du trafic via le tunnel, définissez les adresses IP du tunnel du serveur comme passerelle par défaut :

```
# nmcli connection modify client-wg0 ipv4.gateway 192.0.2.1 ipv6.gateway
2001:db8:1::1
```

L'acheminement de tout le trafic à travers le tunnel nécessite que vous définissiez, dans une étape ultérieure, la valeur **allowed-ips** sur ce client à **0.0.0.0/::/0**.

Notez que l'acheminement de tout le trafic via le tunnel peut avoir un impact sur la connectivité avec d'autres hôtes en fonction du routage du serveur et de la configuration du pare-feu.

6. Ajouter la clé privée du serveur au profil de connexion :

```
# nmcli connection modify client-wg0 wireguard.private-key
"aPUcp5vHz8yMLrzk8SsDyYnV33lhE/k20e52iKJFV0A="
```

- a. Modifiez le fichier `/etc/NetworkManager/system-connections/client-wg0.nmconnection` et ajouter :

```
[wireguard-peer.UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=]
endpoint=server.example.com:51820
allowed-ips=192.0.2.1;2001:db8:1::1;
persistent-keepalive=20
```

- L'entrée `[wireguard-peer.<public_key_of_the_server>]` définit la section homologue du serveur, et le nom de la section contient la clé publique du serveur.

- Le paramètre **endpoint** définit le nom d'hôte ou l'adresse IP et le port du serveur. Le client utilise ces informations pour établir la connexion.
- Le paramètre **allowed-ips** définit une liste d'adresses IP qui peuvent envoyer des données à ce client. Par exemple, définissez le paramètre comme suit :
 - Les adresses IP du tunnel du serveur pour permettre uniquement au serveur de communiquer avec ce client. La valeur indiquée dans l'exemple ci-dessus configure ce scénario.
 - **0.0.0.0/0;:::0;** pour permettre à n'importe quelle adresse IPv4 et IPv6 distante de communiquer avec ce client. Utilisez ce paramètre pour acheminer tout le trafic à travers le tunnel et utiliser le serveur WireGuard comme passerelle par défaut.
- Le paramètre optionnel **persistent-keepalive** définit un intervalle en secondes dans lequel WireGuard envoie un paquet de maintien en vie au serveur. Définissez ce paramètre si vous utilisez le client dans un réseau avec traduction d'adresse réseau (NAT) ou si un pare-feu ferme la connexion UDP après un certain temps d'inactivité.

b. Recharger le profil de connexion **client-wg0**:

```
# nmcli connection load /etc/NetworkManager/system-connections/client-wg0.nmconnection
```

7. Réactiver la connexion **client-wg0**:

```
# nmcli connection up client-wg0
```

Vérification

1. Effectuer un sondage (ping) des adresses IP du serveur :

```
# ping 192.0.2.1
# ping6 2001:db8:1::1
```

2. Affichez la configuration de l'interface de l'appareil **wg0**:

```
# wg show wg0
interface: wg0
public key: bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
private key: (hidden)
listening port: 51820

peer: UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
endpoint: server.example.com:51820
allowed ips: 192.0.2.1/32, 2001:db8:1::1/128
latest handshake: 1 minute, 41 seconds ago
transfer: 824 B received, 1.01 KiB sent
persistent keepalive: every 20 seconds
```

Pour afficher la clé privée dans la sortie, utilisez la commande **WG_HIDE_KEYS=never wg show wg0** pour afficher la clé privée dans la sortie.

Notez que la sortie ne contient que les entrées **latest handshake** et **transfer** si vous avez déjà envoyé du trafic via le tunnel VPN.

- Affichez la configuration IP de l'appareil **wg0**:

```
# ip address show wg0
10: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/none
    inet 192.0.2.2/24 brd 192.0.2.255 scope global noprefixroute wg0
        valid_lft forever preferred_lft forever
    inet6 2001:db8:1::2/32 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::73d9:6f51:ea6f:863e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- La page de manuel **wg(8)**
- La section **WireGuard setting** dans la page de manuel **nm-settings(5)**

8.12. CONFIGURATION D'UN CLIENT WIREGUARD À L'AIDE DE NMTUI

Vous pouvez configurer un client WireGuard en créant un profil de connexion dans NetworkManager. Utilisez cette méthode pour laisser NetworkManager gérer la connexion WireGuard.

Cette procédure suppose les paramètres suivants :

- Client :
 - Clé privée : **aPUcp5vHz8yMLrzk8SsDyYnV33lhE/k20e52iKJFV0A=**
 - Adresse IPv4 du tunnel : **192.0.2.2/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::2/32**
- Serveur :
 - Clé publique : **UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=**
 - Adresse IPv4 du tunnel : **192.0.2.1/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::1/32**

Conditions préalables

- Vous avez généré les clés publique et privée pour le serveur et le client.
- Vous connaissez les informations suivantes :
 - La clé privée du client
 - Les adresses IP du tunnel statique et les masques de sous-réseau du client
 - La clé publique du serveur
 - Les adresses IP du tunnel statique et les masques de sous-réseau du serveur

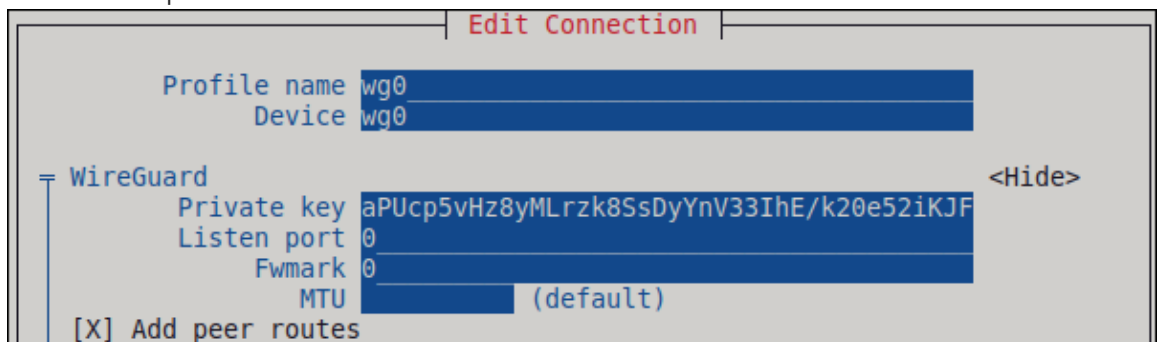
- Vous avez installé le paquetage **NetworkManager-tui**

Procédure

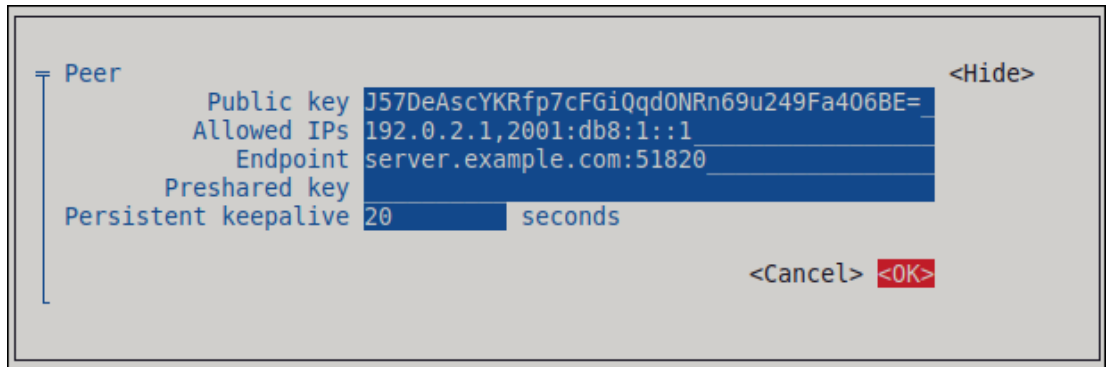
1. Lancez l'application **nmtui**:

```
# nmtui
```

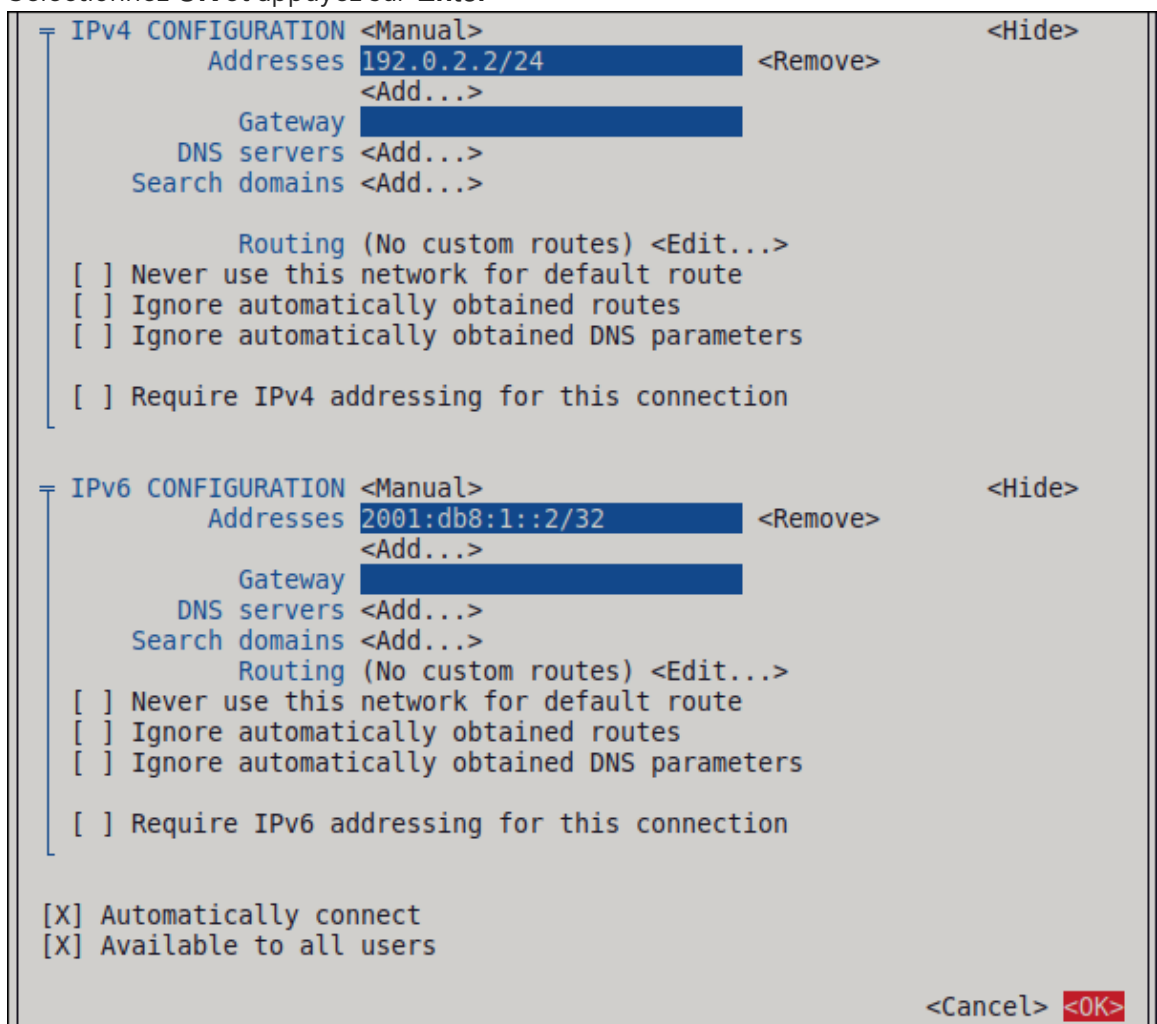
2. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
3. Sélectionnez **Ajouter** et appuyez sur **Entrée**.
4. Sélectionnez le type de connexion **WireGuard** dans la liste et appuyez sur **Entrée**.
5. Dans la fenêtre **Edit connection**:
 - a. Entrez le nom de la connexion et l'interface virtuelle, telle que **wg0**, que NetworkManager doit attribuer à la connexion.
 - b. Entrez la clé privée du client.



- c. Cliquez sur **Ajouter à** côté du volet **Peers**:
 - i. Entrez la clé publique du serveur.
 - ii. Définissez le champ **Allowed IPs**. Par exemple, définissez-le comme suit :
 - Les adresses IP du tunnel du serveur afin que seul le serveur puisse communiquer avec ce client.
 - **0.0.0.0/0,:::0** pour permettre à n'importe quelle adresse IPv4 et IPv6 distante de communiquer avec ce client. Utilisez ce paramètre pour acheminer tout le trafic à travers le tunnel et utiliser le serveur WireGuard comme passerelle par défaut.
 - iii. Entrez le nom d'hôte ou l'adresse IP et le port du serveur WireGuard dans le champ **Endpoint**. Utilisez le format suivant : **hostname_or_IP:port_number**
 - iv. Facultatif : si vous utilisez le client dans un réseau avec traduction d'adresse réseau (NAT) ou si un pare-feu ferme la connexion UDP après un certain temps d'inactivité, définissez un intervalle de maintien en vie persistant en secondes. Dans cet intervalle, le client envoie un paquet keepalive au serveur.
 - v. Sélectionnez **OK** et appuyez sur **Entrée**.



- d. Sélectionnez **Show** à côté de **IPv4 Configuration**, et appuyez sur **Enter**.
 - i. Sélectionnez la méthode de configuration IPv4 **Manual**.
 - ii. Saisissez l'adresse IPv4 du tunnel et le masque de sous-réseau. Laissez le champ **Gateway** vide.
- e. Sélectionnez **Show** à côté de **IPv6 Configuration**, et appuyez sur **Enter**.
 - i. Sélectionnez la méthode de configuration IPv6 **Manual**.
 - ii. Saisissez l'adresse IPv6 du tunnel et le masque de sous-réseau. Laissez le champ **Gateway** vide.
- f. En option : Sélectionnez **Automatically connect**.
- g. Sélectionnez **OK** et appuyez sur **Enter**



- Dans la fenêtre contenant la liste des connexions, sélectionnez **Retour** et appuyez sur **Entrée**.
- Dans la fenêtre principale de **NetworkManager TUI**, sélectionnez **Quitter** et appuyez sur **Entrée**.

Vérification

- Effectuer un sondage (ping) des adresses IP du serveur :

```
# ping 192.0.2.1
# ping6 2001:db8:1::1
```

- Affichez la configuration de l'interface de l'appareil **wg0**:

```
# wg show wg0
interface: wg0
  public key: bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
  private key: (hidden)
  listening port: 51820

peer: UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
  endpoint: server.example.com:51820
  allowed ips: 192.0.2.1/32, 2001:db8:1::1/128
  latest handshake: 1 minute, 41 seconds ago
  transfer: 824 B received, 1.01 KiB sent
  persistent keepalive: every 20 seconds
```

Pour afficher la clé privée dans la sortie, utilisez la commande **WG_HIDE_KEYS=never wg show wg0** pour afficher la clé privée dans la sortie.

Notez que la sortie ne contient que les entrées **latest handshake** et **transfer** si vous avez déjà envoyé du trafic via le tunnel VPN.

- Affichez la configuration IP de l'appareil **wg0**:

```
# ip address show wg0
10: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state
UNKNOWN group default qlen 1000
  link/none
  inet 192.0.2.2/24 brd 192.0.2.255 scope global noprefixroute wg0
    valid_lft forever preferred_lft forever
  inet6 2001:db8:1::2/32 scope global noprefixroute
    valid_lft forever preferred_lft forever
  inet6 fe80::73d9:6f51:ea6f:863e/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- La page de manuel **wg(8)**

8.13. CONFIGURATION D'UN CLIENT WIREGUARD À L'AIDE DE NM-CONNECTION-EDITOR

Vous pouvez configurer un client WireGuard en créant un profil de connexion dans NetworkManager. Utilisez cette méthode pour laisser NetworkManager gérer la connexion WireGuard.

Conditions préalables

- Vous avez généré les clés publique et privée pour le serveur et le client.
- Vous connaissez les informations suivantes :
 - La clé privée du client
 - Les adresses IP du tunnel statique et les masques de sous-réseau du client
 - La clé publique du serveur
 - Les adresses IP du tunnel statique et les masques de sous-réseau du serveur

Procédure

1. Ouvrez un terminal et entrez :

```
# nm-connection-editor
```

2. Ajoutez une nouvelle connexion en cliquant sur le bouton **+**.
3. Sélectionnez le type de connexion **WireGuard** et cliquez sur **Créer**.
4. Optionnel : Mettre à jour le nom de la connexion.
5. Optionnel : Dans l'onglet **General**, sélectionnez **Connect automatically with priority**.
6. Dans l'onglet **WireGuard**:
 - a. Entrez le nom de l'interface virtuelle, telle que **wg0**, que NetworkManager doit attribuer à la connexion.
 - b. Saisir la clé privée du client.
 - c. Cliquez sur **Ajouter** pour ajouter des pairs :
 - i. Entrez la clé publique du serveur.
 - ii. Définissez le champ **Allowed IPs**. Par exemple, définissez-le comme suit :
 - Les adresses IP du tunnel du serveur afin que seul le serveur puisse communiquer avec ce client.
 - **0.0.0.0/0:::0**; pour permettre à n'importe quelle adresse IPv4 et IPv6 distante de communiquer avec ce client. Utilisez ce paramètre pour acheminer tout le trafic à travers le tunnel et utiliser le serveur WireGuard comme passerelle par défaut. Notez que l'acheminement de tout le trafic via le tunnel peut avoir un impact sur la connectivité avec d'autres hôtes en fonction du routage du serveur et de la configuration du pare-feu.
 - iii. Entrez le nom d'hôte ou l'adresse IP et le port du serveur WireGuard dans le champ **Endpoint**. Utilisez le format suivant : **hostname_or_IP:port_number**

- iv. Facultatif : si vous utilisez le client dans un réseau avec traduction d'adresse réseau (NAT) ou si un pare-feu ferme la connexion UDP après un certain temps d'inactivité, définissez un intervalle de maintien en vie persistant en secondes. Dans cet intervalle, le client envoie un paquet de maintien en vie au serveur.
 - v. Cliquez sur **Appliquer**.
7. Dans l'onglet **IPv4 Settings**:
 - a. Sélectionnez **Manual** dans la liste **Method**.
 - b. Cliquez sur **Ajouter** pour entrer l'adresse IPv4 du tunnel et le masque de sous-réseau.
 - c. Si vous souhaitez acheminer tout le trafic via le tunnel, indiquez l'adresse IPv4 du serveur dans le champ **Gateway**. Sinon, laissez le champ vide.
Pour acheminer tout le trafic IPv4 via le tunnel, vous devez inclure **0.0.0.0/0** dans le champ **Allowed IPs** sur ce client.
 8. Dans l'onglet **IPv6 Settings**:
 - a. Sélectionnez **Manual** dans la liste **Method**.
 - b. Cliquez sur **Ajouter** pour entrer l'adresse IPv6 du tunnel et le masque de sous-réseau.
 - c. Si vous souhaitez acheminer tout le trafic via le tunnel, définissez l'adresse IPv6 du serveur dans le champ **Gateway**. Sinon, laissez le champ vide.
Pour acheminer tout le trafic IPv4 via le tunnel, vous devez inclure **::0** dans le champ **Allowed IPs** sur ce client.
 9. Cliquez sur **Enregistrer** pour sauvegarder le profil de connexion.

Vérification

1. Effectuer un sondage (ping) des adresses IP du serveur :

```
# ping 192.0.2.1
# ping6 2001:db8:1::1
```

2. Affichez la configuration de l'interface de l'appareil **wg0**:

```
# wg show wg0
interface: wg0
public key: bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
private key: (hidden)
listening port: 51820

peer: UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
endpoint: server.example.com:51820
allowed ips: 192.0.2.1/32, 2001:db8:1::1/128
latest handshake: 1 minute, 41 seconds ago
transfer: 824 B received, 1.01 KiB sent
persistent keepalive: every 20 seconds
```

Pour afficher la clé privée dans la sortie, utilisez la commande **WG_HIDE_KEYS=never wg show wg0** pour afficher la clé privée dans la sortie.

Notez que la sortie ne contient les entrées **latest handshake** et **transfer** que si vous avez déjà envoyé du trafic via le tunnel VPN.

- Affichez la configuration IP de l'appareil **wg0**:

```
# ip address show wg0
10: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state
UNKNOWN group default qlen 1000
    link/none
    inet 192.0.2.2/24 brd 192.0.2.255 scope global noprefixroute wg0
        valid_lft forever preferred_lft forever
    inet6 2001:db8:1::2/32 scope global noprefixroute
        valid_lft forever preferred_lft forever
    inet6 fe80::73d9:6f51:ea6f:863e/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- La page de manuel **wg(8)**

8.14. CONFIGURER UN CLIENT WIREGUARD EN UTILISANT LE SERVICE WG-QUICK

Vous pouvez configurer un client WireGuard en créant un fichier de configuration dans le répertoire **/etc/wireguard/**. Cette méthode permet de configurer le service indépendamment de NetworkManager.

Cette procédure suppose les paramètres suivants :

- Client :
 - Clé privée : **aPUcp5vHz8yMLrzk8SsDyYnV33lhE/k20e52iKJFV0A=**
 - Adresse IPv4 du tunnel : **192.0.2.2/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::2/32**
- Serveur :
 - Clé publique : **UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=**
 - Adresse IPv4 du tunnel : **192.0.2.1/24**
 - Adresse IPv6 du tunnel : **2001:db8:1::1/32**

Conditions préalables

- Vous avez généré les clés publique et privée pour le serveur et le client.
- Vous connaissez les informations suivantes :
 - La clé privée du client
 - Les adresses IP du tunnel statique et les masques de sous-réseau du client
 - La clé publique du serveur

- Les adresses IP du tunnel statique et les masques de sous-réseau du serveur

Procédure

1. Installez le paquetage **wireguard-tools**:

```
# dnf install wireguard-tools
```

2. Créez le fichier **/etc/wireguard/wg0.conf** avec le contenu suivant :

```
[Interface]
Address = 192.0.2.2/24, 2001:db8:1::2/32
PrivateKey = aPUcp5vHz8yMLrzk8SsDyYnV33lhE/k20e52iKJFV0A=
*
[Peer]
PublicKey = UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
AllowedIPs = 192.0.2.1, 2001:db8:1::1
Endpoint = server.example.com:51820
PersistentKeepalive = 20
```

- La section **[Interface]** décrit les paramètres WireGuard de l'interface sur le client :
 - **Address**: Une liste d'adresses IP du tunnel du client, séparées par des virgules.
 - **PrivateKey**: La clé privée du client.
- La section **[Peer]** décrit les paramètres du serveur :
 - **PublicKey**: La clé publique du serveur.
 - **AllowedIPs**: Les adresses IP autorisées à envoyer des données à ce client. Par exemple, le paramètre est réglé sur :
 - Les adresses IP du tunnel du serveur pour permettre uniquement au serveur de communiquer avec ce client. La valeur indiquée dans l'exemple ci-dessus configure ce scénario.
 - **0.0.0.0/0, ::/0** pour permettre à n'importe quelle adresse IPv4 et IPv6 distante de communiquer avec ce client. Utilisez ce paramètre pour acheminer tout le trafic à travers le tunnel et utiliser le serveur WireGuard comme passerelle par défaut.
 - **Endpoint**: Définit le nom d'hôte ou l'adresse IP et le port du serveur. Le client utilise ces informations pour établir la connexion.
 - Le paramètre optionnel **persistent-keepalive** définit un intervalle en secondes dans lequel WireGuard envoie un paquet keepalive au serveur. Définissez ce paramètre si vous utilisez le client dans un réseau avec traduction d'adresse réseau (NAT) ou si un pare-feu ferme la connexion UDP après un certain temps d'inactivité.
- 3. Activer et démarrer la connexion WireGuard :

```
# systemctl enable --now wg-quick@wg0
```

Le nom de l'instance systemd doit correspondre au nom du fichier de configuration dans le répertoire **/etc/wireguard/** sans le suffixe **.conf**. Le service utilise également ce nom pour l'interface réseau virtuelle.

Vérification

1. Effectuer un sondage (ping) des adresses IP du serveur :

```
# ping 192.0.2.1
# ping6 2001:db8:1::1
```

2. Affichez la configuration de l'interface de l'appareil **wg0**:

```
# wg show wg0
interface: wg0
  public key: bnwfQcC8/g2i4vvEqcRUM2e6Hi3Nskk6G9t4r26nFVM=
  private key: (hidden)
  listening port: 51820

peer: UtjqCJ57DeAscYKRfp7cFGiQqdONRn69u249Fa4O6BE=
  endpoint: server.example.com:51820
  allowed ips: 192.0.2.1/32, 2001:db8:1::1/128
  latest handshake: 1 minute, 41 seconds ago
  transfer: 824 B received, 1.01 KiB sent
  persistent keepalive: every 20 seconds
```

Pour afficher la clé privée dans la sortie, utilisez la commande **WG_HIDE_KEYS=never wg show wg0** pour afficher la clé privée dans la sortie.

Notez que la sortie ne contient que les entrées **latest handshake** et **transfer** si vous avez déjà envoyé du trafic via le tunnel VPN.

3. Affichez la configuration IP de l'appareil **wg0**:

```
# ip address show wg0
10: wg0: <POINTOPOINT,NOARP,UP,LOWER_UP> mtu 1420 qdisc noqueue state
UNKNOWN group default qlen 1000
  link/none
  inet 192.0.2.2/24 scope global wg0
    valid_lft forever preferred_lft forever
  inet6 2001:db8:1::2/32__ scope global
    valid_lft forever preferred_lft forever
```

Ressources supplémentaires

- La page de manuel **wg(8)**
- La page de manuel **wg-quick(8)**

CHAPITRE 9. CONFIGURATION DES TUNNELS IP

Semblable à un VPN, un tunnel IP relie directement deux réseaux par l'intermédiaire d'un troisième réseau, tel que l'internet. Cependant, tous les protocoles de tunnel ne prennent pas en charge le cryptage.

Les routeurs des deux réseaux qui établissent le tunnel ont besoin d'au moins deux interfaces :

- Une interface connectée au réseau local
- Une interface connectée au réseau par lequel le tunnel est établi.

Pour établir le tunnel, vous créez une interface virtuelle sur les deux routeurs avec une adresse IP du sous-réseau distant.

Le NetworkManager supporte les tunnels IP suivants :

- Encapsulation générique du routage (GRE)
- Encapsulation générique du routage sur IPv6 (IP6GRE)
- Point d'accès terminal d'encapsulation générique de routage (GRETAP)
- Point d'accès terminal d'encapsulation de routage générique sur IPv6 (IP6GRETAP)
- IPv4 sur IPv4 (IPIP)
- IPv4 sur IPv6 (IPIP6)
- IPv6 sur IPv6 (IP6IP6)
- Transition Internet simple (SIT)

Selon le type, ces tunnels agissent sur la couche 2 ou 3 du modèle d'interconnexion des systèmes ouverts (OSI).

9.1. CONFIGURATION D'UN TUNNEL IPIP À L'AIDE DE NMCLI POUR ENCAPSULER LE TRAFIC IPV4 DANS DES PAQUETS IPV4

Un tunnel IP sur IP (IPIP) fonctionne sur la couche 3 de l'OSI et encapsule le trafic IPv4 dans des paquets IPv4, comme décrit dans la [RFC 2003](#).

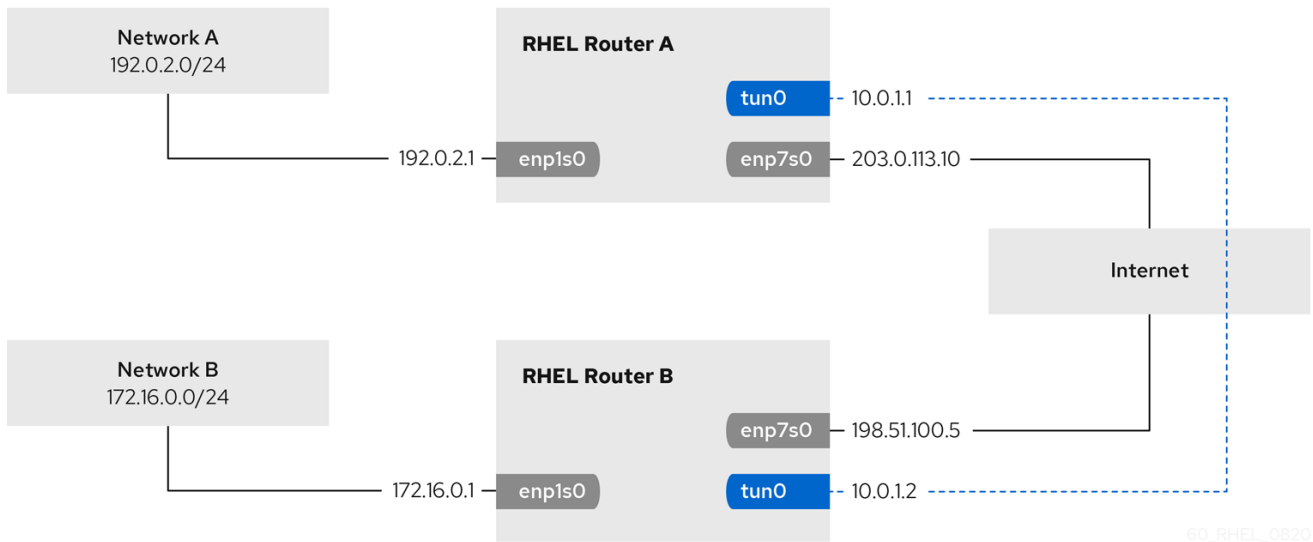


IMPORTANT

Les données envoyées par un tunnel IPIP ne sont pas cryptées. Pour des raisons de sécurité, n'utilisez le tunnel que pour des données déjà cryptées, par exemple par d'autres protocoles, tels que HTTPS.

Notez que les tunnels IPIP ne prennent en charge que les paquets unicast. Si vous avez besoin d'un tunnel IPv4 prenant en charge la multidiffusion, consultez la section [Configuration d'un tunnel GRE à l'aide de nmcli pour encapsuler le trafic de couche 3 dans des paquets IPv4](#).

Par exemple, vous pouvez créer un tunnel IPIP entre deux routeurs RHEL pour connecter deux sous-réseaux internes sur Internet, comme le montre le diagramme suivant :



60_RHEL_0820

Conditions préalables

- Chaque routeur RHEL dispose d'une interface réseau connectée à son sous-réseau local.
- Chaque routeur RHEL dispose d'une interface réseau connectée à Internet.
- Le trafic que vous souhaitez envoyer à travers le tunnel est un monodiffusion IPv4.

Procédure

1. Sur le routeur RHEL du réseau A :
 - a. Créer une interface tunnel IPIP nommée **tun0**:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name tun0 ifname
tun0 remote 198.51.100.5 local 203.0.113.10
```

Les paramètres **remote** et **local** définissent les adresses IP publiques des routeurs local et distant.

- b. Définissez l'adresse IPv4 de l'appareil **tun0**:

```
# nmcli connection modify tun0 ipv4.addresses '10.0.1.1/30'
```

Notez qu'un sous-réseau **/30** avec deux adresses IP utilisables est suffisant pour le tunnel.

- c. Configurer la connexion **tun0** pour utiliser une configuration IPv4 manuelle :

```
# nmcli connection modify tun0 ipv4.method manual
```

- d. Ajoutez une route statique qui achemine le trafic vers le réseau **172.16.0.0/24** vers l'IP du tunnel sur le routeur B :

```
# nmcli connection modify tun0 ipv4.routes "172.16.0.0/24 10.0.1.2"
```

- e. Activer la connexion **tun0**.

```
# nmcli connection up tun0
```

- f. Activer le transfert de paquets :

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

2. Sur le routeur RHEL du réseau B :

- a. Créer une interface tunnel IPIP nommée **tun0**:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode ipip con-name tun0 ifname
tun0 remote 203.0.113.10 local 198.51.100.5
```

Les paramètres **remote** et **local** définissent les adresses IP publiques des routeurs local et distant.

- b. Définissez l'adresse IPv4 de l'appareil **tun0**:

```
# nmcli connection modify tun0 ipv4.addresses '10.0.1.2/30'
```

- c. Configurer la connexion **tun0** pour utiliser une configuration IPv4 manuelle :

```
# nmcli connection modify tun0 ipv4.method manual
```

- d. Ajoutez une route statique qui achemine le trafic vers le réseau **192.0.2.0/24** vers l'IP du tunnel sur le routeur A :

```
# nmcli connection modify tun0 ipv4.routes "192.0.2.0/24 10.0.1.1"
```

- e. Activer la connexion **tun0**.

```
# nmcli connection up tun0
```

- f. Activer le transfert de paquets :

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

Vérification

- À partir de chaque routeur RHEL, envoyez un ping à l'adresse IP de l'interface interne de l'autre routeur :

- a. Sur le routeur A, ping **172.16.0.1**:

```
# ping 172.16.0.1
```

- b. Sur le routeur B, ping **192.0.2.1**:

```
# ping 192.0.2.1
```

Ressources supplémentaires

- **nmcli(1)** page de manuel
- **nm-settings(5)** page de manuel

9.2. CONFIGURATION D'UN TUNNEL GRE À L'AIDE DE NMCLI POUR ENCAPSULER LE TRAFIC DE COUCHE 3 DANS DES PAQUETS IPV4

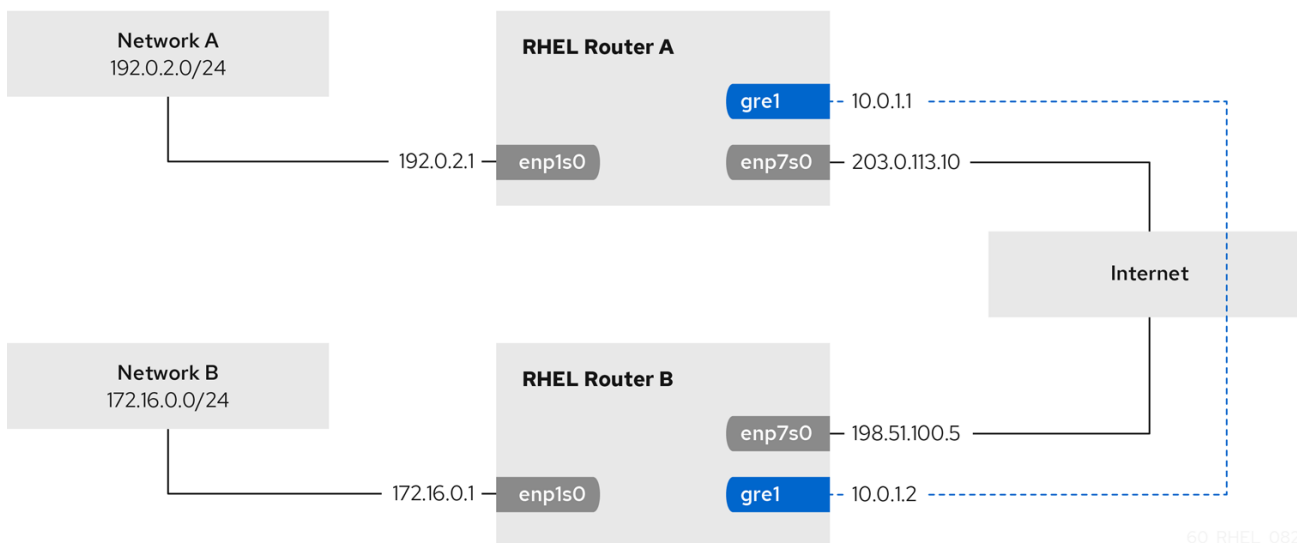
Un tunnel GRE (Generic Routing Encapsulation) encapsule le trafic de la couche 3 dans des paquets IPv4, comme décrit dans la [RFC 2784](#). Un tunnel GRE peut encapsuler n'importe quel protocole de couche 3 avec un type Ethernet valide.



IMPORTANT

Les données envoyées par un tunnel GRE ne sont pas cryptées. Pour des raisons de sécurité, n'utilisez le tunnel que pour des données déjà cryptées, par exemple par d'autres protocoles, tels que HTTPS.

Par exemple, vous pouvez créer un tunnel GRE entre deux routeurs RHEL pour connecter deux sous-réseaux internes sur Internet, comme le montre le diagramme suivant :



NOTE

Le nom de l'appareil **gre0** est réservé. Utilisez **gre1** ou un autre nom pour le dispositif.

Conditions préalables

- Chaque routeur RHEL dispose d'une interface réseau connectée à son sous-réseau local.
- Chaque routeur RHEL dispose d'une interface réseau connectée à Internet.

Procédure

1. Sur le routeur RHEL du réseau A :

- a. Créer une interface de tunnel GRE nommée **gre1**:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode gre con-name gre1 ifname
gre1 remote 198.51.100.5 local 203.0.113.10
```

Les paramètres **remote** et **local** définissent les adresses IP publiques des routeurs local et distant.

- b. Définissez l'adresse IPv4 de l'appareil **gre1**:

```
# nmcli connection modify gre1 ipv4.addresses '10.0.1.1/30'
```

Notez qu'un sous-réseau **/30** avec deux adresses IP utilisables est suffisant pour le tunnel.

- c. Configurer la connexion **gre1** pour utiliser une configuration IPv4 manuelle :

```
# nmcli connection modify gre1 ipv4.method manual
```

- d. Ajoutez une route statique qui achemine le trafic vers le réseau **172.16.0.0/24** vers l'IP du tunnel sur le routeur B :

```
# nmcli connection modify gre1 ipv4.routes "172.16.0.0/24 10.0.1.2"
```

- e. Activer la connexion **gre1**.

```
# nmcli connection up gre1
```

- f. Activer le transfert de paquets :

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

2. Sur le routeur RHEL du réseau B :

- a. Créer une interface de tunnel GRE nommée **gre1**:

```
# nmcli connection add type ip-tunnel ip-tunnel.mode gre con-name gre1 ifname
gre1 remote 203.0.113.10 local 198.51.100.5
```

Les paramètres **remote** et **local** définissent les adresses IP publiques des routeurs local et distant.

- b. Définissez l'adresse IPv4 de l'appareil **gre1**:

```
# nmcli connection modify gre1 ipv4.addresses '10.0.1.2/30'
```

- c. Configurer la connexion **gre1** pour utiliser une configuration IPv4 manuelle :

```
# nmcli connection modify gre1 ipv4.method manual
```

- d. Ajoutez une route statique qui achemine le trafic vers le réseau **192.0.2.0/24** vers l'IP du tunnel sur le routeur A :

```
# nmcli connection modify gre1 ipv4.routes "192.0.2.0/24 10.0.1.1"
```

e. Activer la connexion **gre1**.

```
# nmcli connection up gre1
```

f. Activer le transfert de paquets :

```
# echo "net.ipv4.ip_forward=1" > /etc/sysctl.d/95-IPv4-forwarding.conf  
# sysctl -p /etc/sysctl.d/95-IPv4-forwarding.conf
```

Vérification

1. À partir de chaque routeur RHEL, envoyez un ping à l'adresse IP de l'interface interne de l'autre routeur :
 - a. Sur le routeur A, ping **172.16.0.1**:

```
# ping 172.16.0.1
```

- b. Sur le routeur B, ping **192.0.2.1**:

```
# ping 192.0.2.1
```

Ressources supplémentaires

- **nmcli(1)** page de manuel
- **nm-settings(5)** page de manuel

9.3. CONFIGURATION D'UN TUNNEL GRE-TAP POUR TRANSFÉRER DES TRAMES ETHERNET SUR IPV4

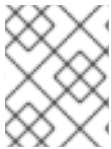
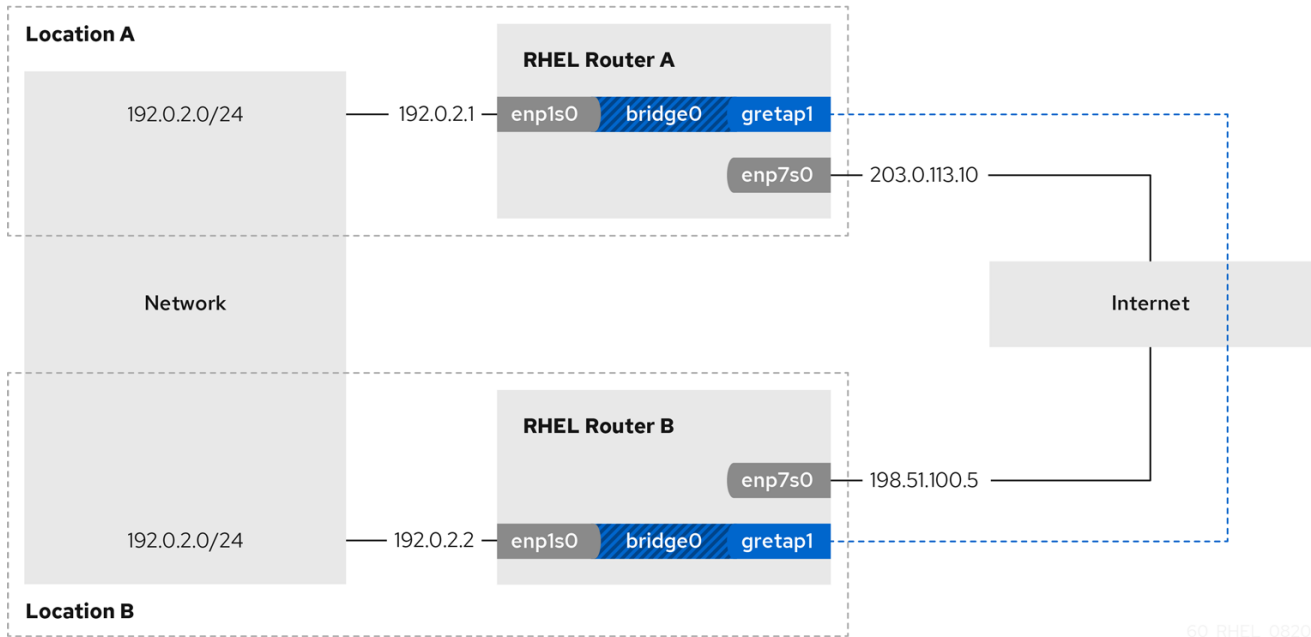
Un tunnel GRE-TAP (Generic Routing Encapsulation Terminal Access Point) fonctionne au niveau 2 de l'OSI et encapsule le trafic Ethernet dans des paquets IPv4, comme décrit dans la [RFC 2784](#).



IMPORTANT

Les données envoyées par un tunnel GRE-TAP ne sont pas cryptées. Pour des raisons de sécurité, établissez le tunnel via un VPN ou une autre connexion cryptée.

Par exemple, vous pouvez créer un tunnel GRE-TAP entre deux routeurs RHEL pour connecter deux réseaux à l'aide d'un pont, comme le montre le schéma suivant :



NOTE

Le nom de l'appareil **gretap0** est réservé. Utilisez **gretap1** ou un autre nom pour le dispositif.

Conditions préalables

- Chaque routeur RHEL dispose d'une interface réseau connectée à son réseau local et aucune configuration IP n'est attribuée à cette interface.
- Chaque routeur RHEL dispose d'une interface réseau connectée à Internet.

Procédure

1. Sur le routeur RHEL du réseau A :

a. Créer une interface de pont nommée **bridge0**:

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

b. Configurez les paramètres IP du pont :

```
# nmcli connection modify bridge0 ipv4.addresses '192.0.2.1/24'
# nmcli connection modify bridge0 ipv4.method manual
```

c. Ajoutez un nouveau profil de connexion pour l'interface qui est connectée au réseau local du pont :

```
# nmcli connection add type ethernet slave-type bridge con-name bridge0-port1
ifname enp1s0 master bridge0
```

d. Ajouter un nouveau profil de connexion pour l'interface du tunnel GRE-TAP au pont :

```
# nmcli connection add type ip-tunnel ip-tunnel.mode gretap slave-type bridge
con-name bridge0-port2 ifname gretap1 remote 198.51.100.5 local 203.0.113.10
master bridge0
```

Les paramètres **remote** et **local** définissent les adresses IP publiques des routeurs local et distant.

- e. Facultatif : Désactivez le protocole Spanning Tree (STP) si vous n'en avez pas besoin :

```
# nmcli connection modify bridge0 bridge.stp no
```

Par défaut, le protocole STP est activé et provoque un délai avant que vous ne puissiez utiliser la connexion.

- f. Configurer que l'activation de la connexion **bridge0** active automatiquement les ports du pont :

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

- g. Active la connexion **bridge0**:

```
# nmcli connection up bridge0
```

2. Sur le routeur RHEL du réseau B :

- a. Créer une interface de pont nommée **bridge0**:

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

- b. Configurez les paramètres IP du pont :

```
# nmcli connection modify bridge0 ipv4.addresses '192.0.2.2/24'
# nmcli connection modify bridge0 ipv4.method manual
```

- c. Ajoutez un nouveau profil de connexion pour l'interface qui est connectée au réseau local du pont :

```
# nmcli connection add type ethernet slave-type bridge con-name bridge0-port1
ifname enp1s0 master bridge0
```

- d. Ajouter un nouveau profil de connexion pour l'interface du tunnel GRETAP au pont :

```
# nmcli connection add type ip-tunnel ip-tunnel.mode gretap slave-type bridge
con-name bridge0-port2 ifname gretap1 remote 203.0.113.10 local 198.51.100.5
master bridge0
```

Les paramètres **remote** et **local** définissent les adresses IP publiques des routeurs local et distant.

- e. Facultatif : Désactivez le protocole Spanning Tree (STP) si vous n'en avez pas besoin :

```
# nmcli connection modify bridge0 bridge.stp no
```


- f. Configurer que l'activation de la connexion **bridge0** active automatiquement les ports du pont :

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

- g. Active la connexion **bridge0**:

```
# nmcli connection up bridge0
```

Vérification

1. Sur les deux routeurs, vérifiez que les connexions **enp1s0** et **gretap1** sont connectées et que la colonne **CONNECTION** affiche le nom de connexion du port :

```
# nmcli device
nmcli device
DEVICE TYPE STATE CONNECTION
...
bridge0 bridge connected bridge0
enp1s0 ethernet connected bridge0-port1
gretap1 iptunnel connected bridge0-port2
```

2. À partir de chaque routeur RHEL, envoyez un ping à l'adresse IP de l'interface interne de l'autre routeur :
- a. Sur le routeur A, ping **192.0.2.2**:

```
# ping 192.0.2.2
```

- b. Sur le routeur B, ping **192.0.2.1**:

```
# ping 192.0.2.1
```

Ressources supplémentaires

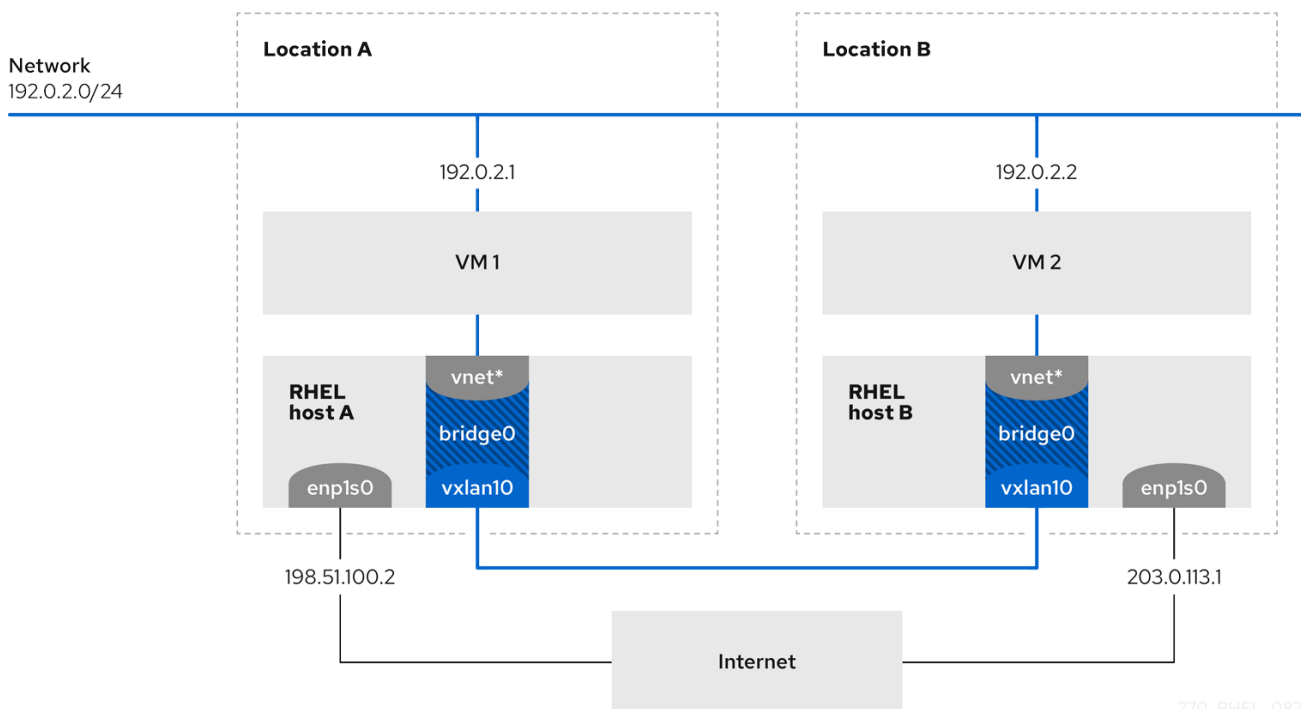
- **nmcli(1)** page de manuel
- **nm-settings(5)** page de manuel

9.4. RESSOURCES SUPPLÉMENTAIRES

- **ip-link(8)** page de manuel

CHAPITRE 10. UTILISATION D'UN VXLAN POUR CRÉER UN DOMAINE VIRTUEL DE COUCHE 2 POUR LES MACHINES VIRTUELLES

Un réseau local extensible virtuel (VXLAN) est un protocole de réseau qui tunnelise le trafic de couche 2 sur un réseau IP à l'aide du protocole UDP. Par exemple, certaines machines virtuelles (VM) fonctionnant sur différents hôtes peuvent communiquer par le biais d'un tunnel VXLAN. Les hôtes peuvent se trouver dans des sous-réseaux différents ou même dans des centres de données différents dans le monde entier. Du point de vue des machines virtuelles, les autres machines virtuelles du même VXLAN se trouvent dans le même domaine de couche 2 :



270_RHEL_0822

Dans cet exemple, les hôtes RHEL-A et RHEL-B utilisent un pont, **br0**, pour connecter le réseau virtuel d'une VM sur chaque hôte avec un VXLAN nommé **vxlan10**. Grâce à cette configuration, le VXLAN est invisible pour les machines virtuelles, qui n'ont besoin d'aucune configuration particulière. Si vous connectez ultérieurement d'autres VM au même réseau virtuel, les VM sont automatiquement membres du même domaine virtuel de couche 2.



IMPORTANT

Tout comme le trafic normal de couche 2, les données d'un VXLAN ne sont pas cryptées. Pour des raisons de sécurité, utilisez un VXLAN par le biais d'un VPN ou d'autres types de connexions cryptées.

10.1. AVANTAGES DES VXLAN

Un réseau local extensible virtuel (VXLAN) offre les principaux avantages suivants :

- Les VXLAN utilisent un ID de 24 bits. Vous pouvez donc créer jusqu'à 16 777 216 réseaux isolés. Par exemple, un réseau local virtuel (VLAN) ne prend en charge que 4 096 réseaux isolés.

- Les VXLAN utilisent le protocole IP. Cela vous permet d'acheminer le trafic et de faire fonctionner virtuellement des systèmes dans différents réseaux et emplacements au sein du même domaine de couche 2.
- Contrairement à la plupart des protocoles de tunnel, un VXLAN n'est pas seulement un réseau point à point. Un VXLAN peut apprendre les adresses IP des autres points d'extrémité soit dynamiquement, soit en utilisant des entrées de transfert configurées de manière statique.
- Certaines cartes réseau prennent en charge les fonctions de délestage liées au tunnel UDP.

Ressources supplémentaires

- `/usr/share/doc/kernel-doc-<kernel_version>/Documentation/networking/vxlan.rst` fournie par le paquet `kernel-doc`

10.2. CONFIGURATION DE L'INTERFACE ETHERNET SUR LES HÔTES

Pour connecter un hôte RHEL VM à l'Ethernet, créez un profil de connexion réseau, configurez les paramètres IP et activez le profil.

Exécutez cette procédure sur les deux hôtes RHEL et adaptez la configuration de l'adresse IP en conséquence.

Conditions préalables

- L'hôte est connecté à l'Ethernet.

Procédure

1. Ajouter un nouveau profil de connexion Ethernet au NetworkManager :

```
# nmcli connection add con-name Example ifname enp1s0 type ethernet
```

2. Configurez les paramètres IPv4 :

```
# nmcli connection modify Example ipv4.addresses 198.51.100.2/24 ipv4.method manual ipv4.gateway 198.51.100.254 ipv4.dns 198.51.100.200 ipv4.dns-search example.com
```

Sautez cette étape si le réseau utilise le protocole DHCP.

3. Activez la connexion **Example**:

```
# nmcli connection up Example
```

Vérification

1. Affiche l'état des appareils et des connexions :

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

2. Effectuer un sondage (ping) d'un hôte dans un réseau distant pour vérifier les paramètres IP :

```
# ping RHEL-host-B.example.com
```

Notez que vous ne pouvez pas envoyer de ping à l'autre hôte VM avant d'avoir configuré le réseau sur cet hôte également.

Ressources supplémentaires

- **nm-settings(5)** page de manuel

10.3. CRÉATION D'UN PONT RÉSEAU AVEC UN VXLAN ATTACHÉ

Pour rendre un réseau local extensible virtuel (VXLAN) invisible aux machines virtuelles (VM), créez un pont sur un hôte et attachez le VXLAN au pont. Utilisez NetworkManager pour créer le pont et le VXLAN. Vous n'ajoutez au pont aucun périphérique de point d'accès au trafic (TAP) des machines virtuelles, généralement nommé **vnet*** sur l'hôte. Le service **libvirtd** les ajoute dynamiquement au démarrage des VM.

Exécutez cette procédure sur les deux hôtes RHEL et ajustez les adresses IP en conséquence.

Procédure

1. Créer le pont **br0**:

```
# nmcli connection add type bridge con-name br0 ifname br0 ipv4.method disabled  
ipv6.method disabled
```

Cette commande ne définit aucune adresse IPv4 et IPv6 sur le périphérique de pont, car ce pont fonctionne sur la couche 2.

2. Créer l'interface VXLAN et l'attacher à **br0**:

```
# nmcli connection add type vxlan slave-type bridge con-name br0-vxlan10 ifname  
vxlan10 id 10 local 198.51.100.2 remote 203.0.113.1 master br0
```

Cette commande utilise les paramètres suivants :

- **id 10**: Définit l'identifiant VXLAN.
- **local 198.51.100.2**: Définit l'adresse IP source des paquets sortants.
- **remote 203.0.113.1**: Définit l'adresse IP unicast ou multicast à utiliser dans les paquets sortants lorsque l'adresse de la couche de liaison de destination n'est pas connue dans la base de données de transfert du dispositif VXLAN.
- **master br0**: Configure cette connexion VXLAN pour qu'elle soit créée en tant que port dans la connexion **br0**.
- **ipv4.method disabled** et **ipv6.method disabled**: Désactive IPv4 et IPv6 sur le pont.

Par défaut, NetworkManager utilise **8472** comme port de destination. Si le port de destination est différent, il faut ajouter l'option **destination-port <port_number>** à la commande.

3. Activer le profil de connexion **br0**:

```
# nmcli connection up br0
```

- Ouvrez le port **8472** pour les connexions UDP entrantes dans le pare-feu local :

```
# firewall-cmd --permanent --add-port=8472/udp
# firewall-cmd --reload
```

Vérification

- Affiche la table de transfert :

```
# bridge fdb show dev vxlan10
2a:53:bd:d5:b3:0a master br0 permanent
00:00:00:00:00:00 dst 203.0.113.1 self permanent
...
```

Ressources supplémentaires

- nm-settings(5)** page de manuel

10.4. CRÉER UN RÉSEAU VIRTUEL DANS LIBVIRT AVEC UN PONT EXISTANT

Pour permettre aux machines virtuelles (VM) d'utiliser le pont **br0** avec le réseau local extensible virtuel (VXLAN) attaché, ajoutez d'abord un réseau virtuel au service **libvirt** qui utilise ce pont.

Conditions préalables

- Vous avez installé le paquetage **libvirt**.
- Vous avez démarré et activé le service **libvirtd**.
- Vous avez configuré l'appareil **br0** avec le VXLAN sur RHEL.

Procédure

- Créez le fichier **~/vxlan10-bridge.xml** avec le contenu suivant :

```
<network>
  <name>vxlan10-bridge</name>
  <forward mode="bridge" />
  <bridge name="br0" />
</network>
```

- Utilisez le fichier **~/vxlan10-bridge.xml** pour créer un nouveau réseau virtuel dans **libvirt**:

```
# virsh net-define ~/vxlan10-bridge.xml
```

- Supprimez le fichier **~/vxlan10-bridge.xml**:

```
# rm ~/vxlan10-bridge.xml
```

- Démarrez le réseau virtuel **vxlan10-bridge**:

```
# virsh net-start vxlan10-bridge
```

- Configurez le réseau virtuel **vxlan10-bridge** pour qu'il démarre automatiquement lorsque le service **libvirtd** démarre :

```
# virsh net-autostart vxlan10-bridge
```

Vérification

- Affiche la liste des réseaux virtuels :

```
# virsh net-list
Name           State  Autostart  Persistent
-----
vxlan10-bridge active  yes        yes
...
```

Ressources supplémentaires

- virsh(1)** page de manuel

10.5. CONFIGURATION DES MACHINES VIRTUELLES POUR L'UTILISATION DE VXLAN

Pour configurer une VM afin qu'elle utilise un dispositif de pont avec un réseau local virtuel extensible (VXLAN) sur l'hôte, créez une nouvelle VM qui utilise le réseau virtuel **vxlan10-bridge** ou mettez à jour les paramètres des VM existantes afin qu'elles utilisent ce réseau.

Effectuez cette procédure sur les hôtes RHEL.

Conditions préalables

- Vous avez configuré le réseau virtuel **vxlan10-bridge** dans **libvirtd**.

Procédure

- Pour créer une nouvelle VM et la configurer pour utiliser le réseau **vxlan10-bridge**, passez l'option **--network network:vxlan10-bridge** à la commande **virt-install** lorsque vous créez la VM :

```
# virt-install ... --network network:vxlan10-bridge
```

- Pour modifier les paramètres réseau d'une VM existante :
 - Connectez l'interface réseau de la VM au réseau virtuel **vxlan10-bridge**:

```
# virt-xml VM_name --edit --network network=vxlan10-bridge
```

- Arrêtez la VM et redémarrez-la :

```
# virsh shutdown VM_name
# virsh start VM_name
```

Vérification

1. Affiche les interfaces réseau virtuelles de la VM sur l'hôte :

```
# virsh domiflist VM_name
Interface Type Source Model MAC
-----
vnet1 bridge vxlan10-bridge virtio 52:54:00:c5:98:1c
```

2. Affiche les interfaces attachées au pont **vxlan10-bridge**:

```
# ip link show master vxlan10-bridge
18: vxlan10: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master
br0 state UNKNOWN mode DEFAULT group default qlen 1000
    link/ether 2a:53:bd:d5:b3:0a brd ff:ff:ff:ff:ff:ff
19: vnet1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue master
br0 state UNKNOWN mode DEFAULT group default qlen 1000
    link/ether 52:54:00:c5:98:1c brd ff:ff:ff:ff:ff:ff
```

Notez que le service **libvirtd** met à jour dynamiquement la configuration de la passerelle. Lorsque vous démarrez une VM qui utilise le réseau **vxlan10-bridge**, le périphérique **vnet*** correspondant sur l'hôte apparaît comme un port du pont.

3. Utilisez les requêtes du protocole de résolution d'adresses (ARP) pour vérifier si les machines virtuelles se trouvent dans le même VXLAN :
 - a. Démarrer deux ou plusieurs VM dans le même VXLAN.
 - b. Envoyer une requête ARP d'une VM à l'autre :

```
# arping -c 1 192.0.2.2
ARPING 192.0.2.2 from 192.0.2.1 enp1s0
Unicast reply from 192.0.2.2 [52:54:00:c5:98:1c] 1.450ms
Sent 1 probe(s) (0 broadcast(s))
Received 1 response(s) (0 request(s), 0 broadcast(s))
```

Si la commande affiche une réponse, la VM se trouve dans le même domaine de couche 2 et, dans ce cas, dans le même VXLAN.

Installez le paquetage **iputils** pour utiliser l'utilitaire **arping**.

Ressources supplémentaires

- **virt-install(1)** page de manuel
- **virt-xml(1)** page de manuel
- **virsh(1)** page de manuel
- **arping(8)** page de manuel

CHAPITRE 11. GESTION DES CONNEXIONS WIFI

RHEL fournit de nombreux utilitaires et applications pour configurer et se connecter aux réseaux wifi, par exemple :

- Utilisez l'utilitaire **nmcli** pour configurer les connexions à l'aide de la ligne de commande.
- Utilisez l'application **nmtui** pour configurer les connexions dans une interface utilisateur textuelle.
- Utilisez le menu système GNOME pour vous connecter rapidement à des réseaux wifi qui ne nécessitent aucune configuration.
- Utilisez l'application **GNOME Settings** pour configurer les connexions à l'aide de l'application GNOME.
- Utilisez l'application **nm-connection-editor** pour configurer les connexions dans une interface utilisateur graphique.
- Utilisez le rôle système **network** RHEL pour automatiser la configuration des connexions sur un ou plusieurs hôtes.

11.1. TYPES DE SÉCURITÉ WIFI PRIS EN CHARGE

Selon le type de sécurité qu'un réseau wifi prend en charge, vous pouvez transmettre des données de manière plus ou moins sûre.



AVERTISSEMENT

Ne vous connectez pas à des réseaux wifi qui n'utilisent pas le cryptage ou qui ne prennent en charge que les normes WEP ou WPA non sécurisées.

Red Hat Enterprise Linux 9 prend en charge les types de sécurité wifi suivants :

- **None:** Le cryptage est désactivé et les données sont transférées en texte clair sur le réseau.
- **Enhanced Open:** Avec le cryptage sans fil opportuniste (OWE), les appareils négocient des clés maîtresses uniques par paire (PMK) pour crypter les connexions dans les réseaux sans fil sans authentification.
- **LEAP:** Le Lightweight Extensible Authentication Protocol, développé par Cisco, est une version propriétaire du protocole d'authentification extensible (EAP).
- **WPA & WPA2 Personal:** En mode personnel, les méthodes d'authentification Wi-Fi Protected Access (WPA) et Wi-Fi Protected Access 2 (WPA2) utilisent une clé prépartagée.
- **WPA & WPA2 Enterprise:** En mode entreprise, WPA et WPA2 utilisent le cadre EAP et authentifient les utilisateurs auprès d'un serveur RADIUS (Remote Authentication Dial-in User Service).

- **WPA3 Personal:** Wi-Fi Protected Access 3 (WPA3) Personal utilise l'authentification simultanée des égaux (SAE) au lieu des clés pré-partagées (PSK) pour empêcher les attaques par dictionnaire. Le WPA3 utilise le principe du "perfect forward secrecy" (PFS).

11.2. CONNEXION À UN RÉSEAU WIFI À L'AIDE DE NMCLI

Vous pouvez utiliser l'utilitaire **nmcli** pour vous connecter à un réseau wifi. Lorsque vous tentez de vous connecter à un réseau pour la première fois, l'utilitaire crée automatiquement un profil de connexion NetworkManager. Si le réseau nécessite des paramètres supplémentaires, tels que des adresses IP statiques, vous pouvez alors modifier le profil après sa création automatique.

Conditions préalables

- Un dispositif wifi est installé sur l'hôte.
- Le périphérique wifi est activé, s'il dispose d'un commutateur matériel.

Procédure

1. Si la radio wifi a été désactivée dans NetworkManager, activez cette fonction :

```
# nmcli radio wifi on
```

2. Facultatif : Affiche les réseaux wifi disponibles :

```
# nmcli device wifi list
IN-USE BSSID      SSID      MODE CHAN RATE  SIGNAL BARS SECURITY
      00:53:00:2F:3B:08 Office    Infra 44  270 Mbit/s 57  ████████ WPA2 WPA3
      00:53:00:15:03:BF --        Infra 1   130 Mbit/s 48  ████████ WPA2 WPA3
```

La colonne Service Set Identifier (**SSID**) contient les noms des réseaux. Si la colonne indique **--**, le point d'accès de ce réseau ne diffuse pas de SSID.

3. Se connecter au réseau wifi :

```
# nmcli device wifi connect Office --ask
Password: wifi-password
```

Si vous préférez définir le mot de passe dans la commande plutôt que de le saisir de manière interactive, utilisez l'option **password *wifi-password*** dans la commande au lieu de **--ask**:

```
# nmcli device wifi connect Office wifi-password
```

Notez que si le réseau nécessite des adresses IP statiques, NetworkManager n'active pas la connexion à ce stade. Vous pouvez configurer les adresses IP dans les étapes suivantes.

4. Si le réseau nécessite des adresses IP statiques :

- a. Configurez les paramètres de l'adresse IPv4, par exemple :

```
# nmcli connection modify Office ipv4.method manual ipv4.addresses 192.0.2.1/24
ipv4.gateway 192.0.2.254 ipv4.dns 192.0.2.200 ipv4.dns-search example.com
```

- b. Configurez les paramètres de l'adresse IPv6, par exemple :

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

5. Réactiver la connexion :

```
# nmcli connection up Office
```

Vérification

1. Affiche les connexions actives :

```
# nmcli connection show --active
NAME ID TYPE DEVICE
Office 2501eb7e-7b16-4dc6-97ef-7cc460139a58 wifi wlp0s20f3
```

Si la sortie indique la connexion wifi que vous avez créée, la connexion est active.

2. Effectuer une recherche ping sur un nom d'hôte ou une adresse IP :

```
# ping -c 3 example.com
```

Ressources supplémentaires

- [nm-settings-nmcli\(5\)](#) page de manuel

11.3. SE CONNECTER À UN RÉSEAU WIFI EN UTILISANT LE MENU SYSTÈME GNOME

Vous pouvez utiliser le menu système de GNOME pour vous connecter à un réseau wifi. Lorsque vous vous connectez à un réseau pour la première fois, GNOME crée un profil de connexion NetworkManager pour celui-ci. Si vous configurez le profil de connexion pour qu'il ne se connecte pas automatiquement, vous pouvez également utiliser le menu système de GNOME pour vous connecter manuellement à un réseau wifi avec un profil de connexion NetworkManager existant.



NOTE

L'utilisation du menu système GNOME pour établir une connexion à un réseau wifi pour la première fois présente certaines limites. Par exemple, vous ne pouvez pas configurer les paramètres de l'adresse IP. Dans ce cas, configurez d'abord les connexions :

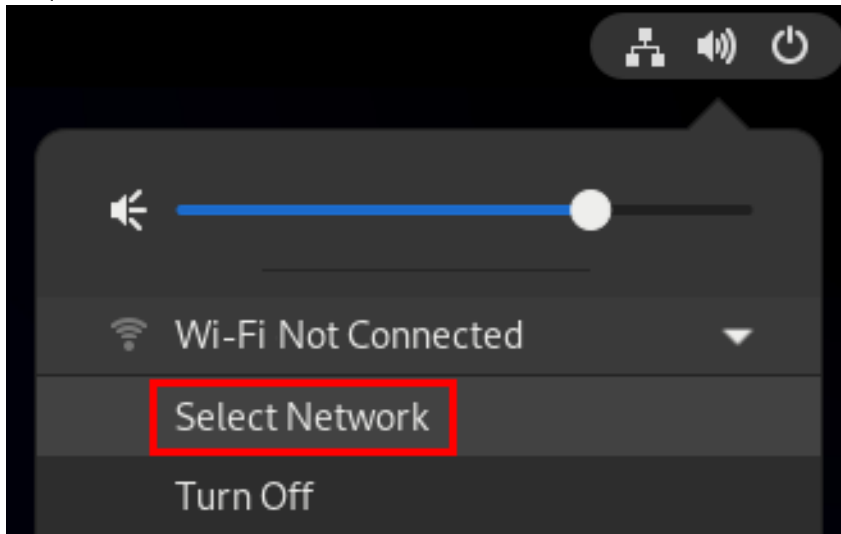
- Dans l'application [GNOME settings](#)
- Dans l'application [nm-connection-editor](#)
- Utilisation des commandes [nmcli](#)

Conditions préalables

- Un dispositif wifi est installé sur l'hôte.
- Le périphérique wifi est activé, s'il dispose d'un commutateur matériel.

Procédure

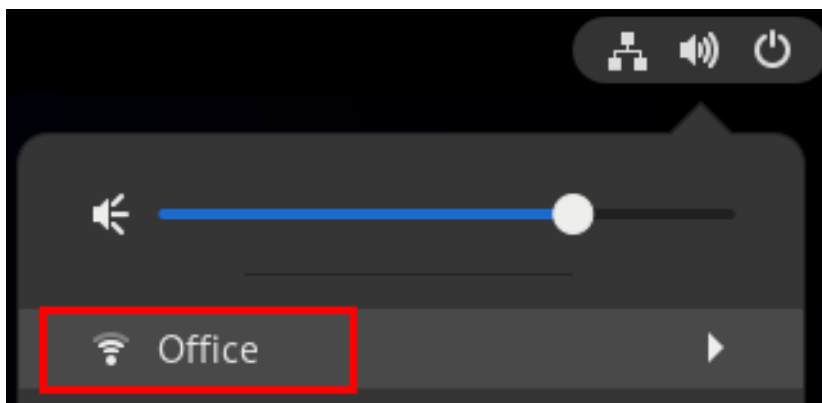
1. Ouvrez le menu système sur le côté droit de la barre supérieure.
2. Développez l'entrée **Wi-Fi Not Connected**.
3. Cliquez sur **Select Network**:



4. Sélectionnez le réseau wifi auquel vous souhaitez vous connecter.
5. Cliquez sur **Connect**.
6. Si c'est la première fois que vous vous connectez à ce réseau, entrez le mot de passe du réseau et cliquez sur **Connect**.

Vérification

1. Ouvrez le menu système sur le côté droit de la barre supérieure et vérifiez que le réseau wifi est connecté :



Si le réseau apparaît dans la liste, il est connecté.

2. Effectuer une recherche ping sur un nom d'hôte ou une adresse IP :

```
# ping -c 3 example.com
```

11.4. SE CONNECTER À UN RÉSEAU WIFI EN UTILISANT L'APPLICATION DE PARAMÉTRAGE GNOME

Vous pouvez utiliser l'application **GNOME settings**, également appelée **gnome-control-center**, pour vous connecter à un réseau wifi et configurer la connexion. Lorsque vous vous connectez au réseau pour la première fois, GNOME crée un profil de connexion NetworkManager.

Sur **GNOME settings**, vous pouvez configurer les connexions wifi pour tous les types de sécurité réseau wifi pris en charge par RHEL.

Conditions préalables

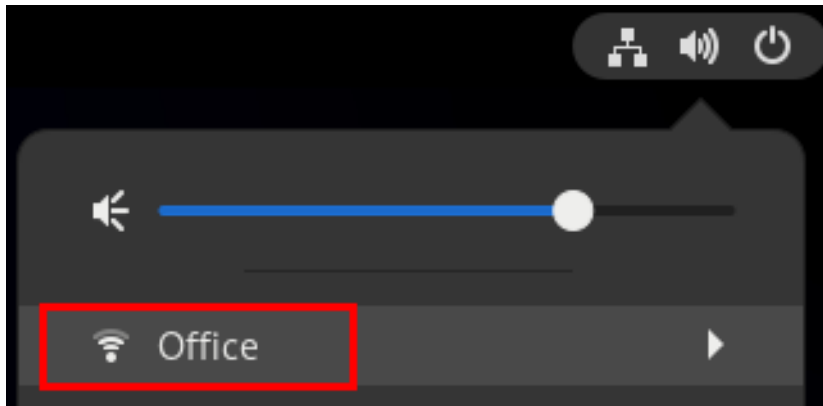
- Un dispositif wifi est installé sur l'hôte.
- Le périphérique wifi est activé, s'il dispose d'un commutateur matériel.

Procédure

1. Appuyez sur la touche **Super**, tapez **Wi-Fi** et appuyez sur **Entrée**.
2. Cliquez sur le nom du réseau wifi auquel vous souhaitez vous connecter.
3. Saisissez le mot de passe du réseau et cliquez sur **Connect**.
4. Si le réseau nécessite des paramètres supplémentaires, tels que des adresses IP statiques ou un type de sécurité autre que WPA2 Personal :
 - a. Cliquez sur l'icône en forme de roue dentée à côté du nom du réseau.
 - b. Optionnel : Configurez le profil réseau dans l'onglet **Details** pour qu'il ne se connecte pas automatiquement.
Si vous désactivez cette fonction, vous devez toujours vous connecter manuellement au réseau, par exemple en utilisant **GNOME settings** ou le menu système GNOME.
 - c. Configurez les paramètres IPv4 dans l'onglet **IPv4** et les paramètres IPv6 dans l'onglet **IPv6**.
 - d. Dans l'onglet **Security**, sélectionnez l'authentification du réseau, par exemple **WPA3 Personal**, et entrez le mot de passe.
En fonction de la sécurité sélectionnée, l'application affiche des champs supplémentaires. Remplissez-les en conséquence. Pour plus de détails, demandez à l'administrateur du réseau wifi.
 - e. Cliquez sur **Apply**.

Vérification

1. Ouvrez le menu système sur le côté droit de la barre supérieure et vérifiez que le réseau wifi est connecté :



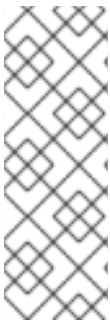
Si le réseau apparaît dans la liste, il est connecté.

2. Effectuer une recherche ping sur un nom d'hôte ou une adresse IP :

```
# ping -c 3 example.com
```

11.5. CONFIGURATION D'UNE CONNEXION WIFI À L'AIDE DE NMTUI

L'application **nmtui** fournit une interface utilisateur textuelle pour NetworkManager. Vous pouvez utiliser **nmtui** pour vous connecter à un réseau wifi.



NOTE

Sur **nmtui**:

- Naviguer à l'aide des touches du curseur.
- Appuyez sur un bouton en le sélectionnant et en appuyant sur **Entrée**.
- Sélectionnez et désélectionnez les cases à cocher en utilisant l'**espace**.

Procédure

1. Si vous ne connaissez pas le nom du périphérique réseau que vous souhaitez utiliser pour la connexion, affichez les périphériques disponibles :

```
# nmcli device status
DEVICE  TYPE  STATE      CONNECTION
wlp2s0  wifi  unavailable --
...
```

2. Démarrer **nmtui**:

```
# nmtui
```

3. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
4. Appuyez sur le bouton **Add**.
5. Sélectionnez **Wi-Fi** dans la liste des types de réseaux et appuyez sur **Entrée**.
6. Optionnel : Entrez un nom pour le profil NetworkManager à créer.

7. Saisissez le nom de l'appareil réseau dans le champ **Device**.
8. Saisissez le nom du réseau Wi-Fi, le Service Set Identifier (SSID), dans le champ **SSID**.
9. Laissez le champ **Mode** sur sa valeur par défaut, **Client**.
10. Sélectionnez le champ **Security**, appuyez sur **Enter** et définissez le type d'authentification du réseau dans la liste.
Selon le type d'authentification que vous avez sélectionné, **nmtui** affiche différents champs.
11. Remplissez les champs relatifs au type d'authentification.
12. Si le réseau Wi-Fi nécessite des adresses IP statiques :
 - a. Appuyez sur le bouton **Automatic** à côté du protocole et sélectionnez **Manual** dans la liste affichée.
 - b. Appuyez sur le bouton **Show** à côté du protocole que vous souhaitez configurer pour afficher des champs supplémentaires et les remplir.
13. Appuyez sur le bouton **OK** pour créer et activer automatiquement la nouvelle connexion.

The screenshot shows the 'Edit Connection' window in nmtui. The title bar reads 'Edit Connection'. The configuration is as follows:

- Profile name: Example-Connection
- Device: wlp2s0
- WI-FI section (collapsible):
 - SSID: Example-Wi-Fi
 - Mode: <Client>
 - Security: <WPA3 Personal>
 - Password: [REDACTED]
 - [] Show password
 - BSSID: [REDACTED]
 - Cloned MAC address: [REDACTED]
 - MTU: [REDACTED] (default)
- IPv4 CONFIGURATION: <Automatic> <Show>
- IPv6 CONFIGURATION: <Automatic> <Show>
- [X] Automatically connect
- [X] Available to all users

Buttons at the bottom right: <Cancel> and <OK>.

14. Appuyez sur le bouton **Back** pour revenir au menu principal.
15. Sélectionnez **Quit** et appuyez sur **Entrée** pour fermer l'application **nmtui**.

Vérification

1. Affiche les connexions actives :

```
# nmcli connection show --active
NAME ID TYPE DEVICE
Office 2501eb7e-7b16-4dc6-97ef-7cc460139a58 wifi wlp0s20f3
```

Si la sortie indique la connexion wifi que vous avez créée, la connexion est active.

2. Effectuer une recherche ping sur un nom d'hôte ou une adresse IP :

```
# ping -c 3 example.com
```

11.6. CONFIGURER UNE CONNEXION WIFI À L'AIDE DE NM-CONNECTION-EDITOR

Vous pouvez utiliser l'application **nm-connection-editor** pour créer un profil de connexion pour un réseau sans fil. Dans cette application, vous pouvez configurer tous les types d'authentification de réseau sans fil pris en charge par RHEL.

Par défaut, NetworkManager active la fonction de connexion automatique pour les profils de connexion et se connecte automatiquement à un réseau enregistré s'il est disponible.


Conditions préalables

- Un dispositif wifi est installé sur l'hôte.
- Le périphérique wifi est activé, s'il dispose d'un commutateur matériel.

Procédure

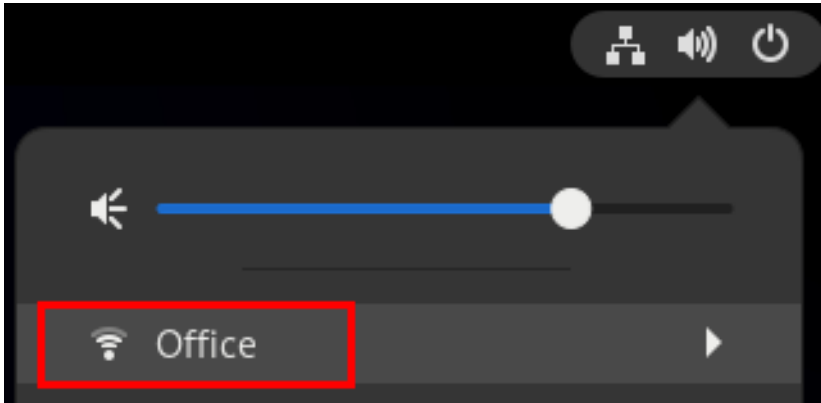
1. Ouvrez un terminal et entrez :

```
# nm-connection-editor
```

2. Cliquez sur le bouton  pour ajouter une nouvelle connexion.
3. Sélectionnez le type de connexion **Wi-Fi** et cliquez sur **Créer**.
4. Facultatif : Définissez un nom pour le profil de connexion.
5. Optionnel : Configurez le profil réseau dans l'onglet **General** pour qu'il ne se connecte pas automatiquement.
Si vous désactivez cette fonction, vous devez toujours vous connecter manuellement au réseau, par exemple en utilisant **GNOME settings** ou le menu système GNOME.
6. Dans l'onglet **Wi-Fi**, saisissez l'identifiant de l'ensemble de services (SSID) dans le champ **SSID**.
7. Dans l'onglet **Wi-Fi Security**, sélectionnez le type d'authentification pour le réseau, par exemple **WPA3 Personal**, et entrez le mot de passe.
En fonction de la sécurité sélectionnée, l'application affiche des champs supplémentaires. Remplissez-les en conséquence. Pour plus de détails, demandez à l'administrateur du réseau wifi.
8. Configurez les paramètres IPv4 dans l'onglet **IPv4** et les paramètres IPv6 dans l'onglet **IPv6**.
9. Cliquez sur **Save**.
10. Fermez la fenêtre **Network Connections**.

Vérification

1. Ouvrez le menu système sur le côté droit de la barre supérieure et vérifiez que le réseau wifi est connecté :



Si le réseau apparaît dans la liste, il est connecté.

2. Effectuer une recherche ping sur un nom d'hôte ou une adresse IP :

```
# ping -c 3 example.com
```

11.7. CONFIGURER UNE CONNEXION WIFI AVEC L'AUTHENTIFICATION RÉSEAU 802.1X EN UTILISANT LE RÔLE RÉSEAU RHEL SYSTEM ROLE

À l'aide des rôles système RHEL, vous pouvez automatiser la création d'une connexion wifi. Par exemple, vous pouvez ajouter à distance un profil de connexion sans fil pour l'interface **wlp1s0** à l'aide d'un playbook Ansible. Le profil créé utilise la norme 802.1X pour authentifier le client sur un réseau wifi. Le playbook configure le profil de connexion pour utiliser DHCP. Pour configurer des paramètres IP statiques, adaptez les paramètres du dictionnaire **ip** en conséquence.

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Le réseau prend en charge l'authentification réseau 802.1X.
- Vous avez installé le paquetage **wpa_supplicant** sur le nœud géré.
- DHCP est disponible dans le réseau du nœud géré.
- Les fichiers suivants, nécessaires à l'authentification TLS, existent sur le nœud de contrôle :
 - La clé du client est stockée dans le fichier **/srv/data/client.key**.
 - Le certificat du client est stocké dans le fichier **/srv/data/client.crt**.

- o Le certificat d'autorité de certification est stocké dans le fichier `/srv/data/ca.crt`.

Procédure

1. Créez un fichier playbook, par exemple `~/enable-802.1x.yml` avec le contenu suivant :

```
---
- name: Configure a wifi connection with 802.1X authentication
  hosts: managed-node-01.example.com
  tasks:
    - name: Copy client key for 802.1X authentication
      copy:
        src: "/srv/data/client.key"
        dest: "/etc/pki/tls/private/client.key"
        mode: 0400

    - name: Copy client certificate for 802.1X authentication
      copy:
        src: "/srv/data/client.crt"
        dest: "/etc/pki/tls/certs/client.crt"

    - name: Copy CA certificate for 802.1X authentication
      copy:
        src: "/srv/data/ca.crt"
        dest: "/etc/pki/ca-trust/source/anchors/ca.crt"

    - block:
      - import_role:
          name: linux-system-roles.network
        vars:
          network_connections:
            - name: Configure the Example-wifi profile
              interface_name: wlp1s0
              state: up
              type: wireless
              autoconnect: yes
              ip:
                dhcp4: true
                auto6: true
              wireless:
                ssid: "Example-wifi"
                key_mgmt: "wpa-eap"
              ieee802_1x:
                identity: "user_name"
                eap: tls
                private_key: "/etc/pki/tls/client.key"
                private_key_password: "password"
                private_key_password_flags: none
                client_cert: "/etc/pki/tls/client.pem"
                ca_cert: "/etc/pki/tls/cacert.pem"
                domain_suffix_match: "example.com"
```

2. Exécutez le manuel de jeu :

```
# ansible-playbook ~/enable-802.1x.yml
```

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

11.8. CONFIGURER UNE CONNEXION WIFI AVEC L'AUTHENTIFICATION RÉSEAU 802.1X DANS UN PROFIL EXISTANT EN UTILISANT NMCLI

À l'aide de l'utilitaire **nmcli**, vous pouvez configurer le client pour qu'il s'authentifie sur le réseau. Par exemple, vous pouvez configurer l'authentification PEAP (Protected Extensible Authentication Protocol) avec le Microsoft Challenge-Handshake Authentication Protocol version 2 (MSCHAPv2) dans un profil de connexion wifi NetworkManager existant nommé **wlp1s0**.

Conditions préalables

- Le réseau doit disposer d'une authentification réseau 802.1X.
- Le profil de connexion wifi existe dans NetworkManager et possède une configuration IP valide.
- Si le client doit vérifier le certificat de l'authentificateur, le certificat de l'autorité de certification (CA) doit être stocké dans le répertoire `/etc/pki/ca-trust/source/anchors/`.
- Le paquet **wpa_supplicant** est installé.

Procédure

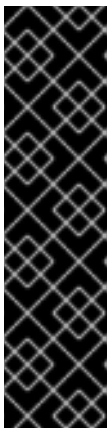
1. Réglez le mode de sécurité wifi sur **wpa-eap**, le protocole d'authentification extensible (EAP) sur **peap**, le protocole d'authentification interne sur **mschapv2**, et le nom d'utilisateur :

```
# nmcli connection modify wlp1s0 wireless-security.key-mgmt wpa-eap 802-1x.eap
peap 802-1x.phase2-auth mschapv2 802-1x.identity user_name
```

Notez que vous devez définir les paramètres **wireless-security.key-mgmt**, **802-1x.eap**, **802-1x.phase2-auth** et **802-1x.identity** en une seule commande.

2. Optionnellement, le mot de passe est stocké dans la configuration :

```
# nmcli connection modify wlp1s0 802-1x.password password
```



IMPORTANT

Par défaut, NetworkManager stocke le mot de passe en texte clair dans le fichier `/etc/sysconfig/network-scripts/keys-connection_name` qui n'est lisible que par l'utilisateur **root**. Cependant, les mots de passe en texte clair dans un fichier de configuration peuvent présenter un risque pour la sécurité.

Pour augmenter la sécurité, définissez le paramètre **802-1x.password-flags** sur **0x1**. Avec ce paramètre, sur les serveurs avec l'environnement de bureau GNOME ou **nm-applet** en cours d'exécution, NetworkManager récupère le mot de passe à partir de ces services. Dans les autres cas, NetworkManager demande le mot de passe.

3. Si le client doit vérifier le certificat de l'authentificateur, le paramètre **802-1x.ca-cert** du profil de connexion doit contenir le chemin d'accès au certificat de l'autorité de certification :

```
# nmcli connection modify wlp1s0 802-1x.ca-cert /etc/pki/ca-trust/source/anchors/ca.crt
```



NOTE

Pour des raisons de sécurité, Red Hat recommande le certificat de l'authentificateur pour permettre aux clients de valider l'identité de l'authentificateur.

4. Activer le profil de connexion :

```
# nmcli connection up wlp1s0
```

Vérification

- Accéder aux ressources du réseau qui nécessitent une authentification réseau.

Ressources supplémentaires

- [Gestion des connexions wifi](#)
- **nm-settings(5)** page de manuel
- **nmcli(1)** page de manuel

11.9. CONFIGURATION MANUELLE DU DOMAINE DE RÉGULATION SANS FIL

Sur RHEL, une règle **udev** exécute l'utilitaire **setregdomain** pour définir le domaine de régulation sans fil. L'utilitaire fournit ensuite ces informations au noyau.

Par défaut, **setregdomain** tente de déterminer automatiquement le code du pays. En cas d'échec, il se peut que le paramètre du domaine de régulation sans fil soit erroné. Pour contourner ce problème, vous pouvez définir manuellement le code du pays.



IMPORTANT

La configuration manuelle du domaine de réglementation désactive la détection automatique. Par conséquent, si vous utilisez ultérieurement l'ordinateur dans un autre pays, il se peut que le paramètre précédemment configuré ne soit plus correct. Dans ce cas, supprimez le fichier **/etc/sysconfig/regdomain** pour revenir à la détection automatique ou utilisez cette procédure pour mettre à jour manuellement le paramètre du domaine réglementaire.

Procédure

1. Optionnel : Affiche les paramètres actuels du domaine de réglementation :

```
# iw reg get
global
country US: DFS-FCC
...
```

2. Créez le fichier `/etc/sysconfig/regdomain` avec le contenu suivant :

```
COUNTRY=<country_code>
```

Attribuez à la variable **COUNTRY** un code de pays ISO 3166-1 alpha2, tel que **DE** pour l'Allemagne ou **US** pour les États-Unis d'Amérique.

3. Définir le domaine réglementaire :

```
# setregdomain
```

Vérification

- Affichez les paramètres du domaine de réglementation :

```
# iw reg get  
global  
country DE: DFS-ETSI  
...
```

Ressources supplémentaires

- **setregdomain(1)** page de manuel
- **iw(8)** page de manuel
- **regulatory.bin(5)** page de manuel
- [Codes de pays ISO 3166](#)

CHAPITRE 12. CONFIGURER RHEL COMME POINT D'ACCÈS WIFI

Sur un hôte doté d'un périphérique wifi, vous pouvez utiliser NetworkManager pour configurer cet hôte en tant que point d'accès. Les clients sans fil peuvent alors utiliser le point d'accès pour se connecter aux services sur l'hôte RHEL ou sur le réseau.

Lorsque vous configurez un point d'accès, le NetworkManager le fait automatiquement :

- Configure le service **dnsmasq** pour qu'il fournisse des services DHCP et DNS aux clients
- Active le transfert IP
- Ajoute les règles du pare-feu **nftables** pour masquer le trafic de l'appareil wifi et configure la redirection IP

12.1. IDENTIFIER SI UN APPAREIL WIFI SUPPORTE LE MODE POINT D'ACCÈS

Pour utiliser un périphérique wifi comme point d'accès, le périphérique doit prendre en charge cette fonctionnalité. Vous pouvez utiliser l'utilitaire **nmcli** pour déterminer si le matériel prend en charge le mode point d'accès.

Conditions préalables

- Un dispositif wifi est installé sur l'hôte.

Procédure

1. Dressez la liste des périphériques wifi pour identifier celui qui doit fournir le point d'accès :

```
# nmcli device status | grep wifi
wlp0s20f3 wifi disconnected --
```

2. Vérifiez que l'appareil prend en charge le mode point d'accès :

```
# nmcli -f WIFI-PROPERTIES.AP device show wlp0s20f3
WIFI-PROPERTIES.AP: yes
```

12.2. CONFIGURATION DE RHEL EN TANT QUE POINT D'ACCÈS PERSONNEL WPA2 OU WPA3

Wi-Fi Protected Access 2 (WPA2) et Wi-Fi Protected Access 3 (WPA3) Personal fournissent des méthodes d'authentification sécurisées dans les réseaux sans fil. Les utilisateurs peuvent se connecter au point d'accès à l'aide d'une clé pré-partagée (PSK).

Conditions préalables

- L'appareil wifi fonctionne en mode point d'accès.
- L'appareil wifi n'est pas utilisé.

- L'hôte dispose d'un accès à Internet.

Procédure

1. Installez les paquets **dnsmasq** et **NetworkManager-wifi**:

```
# dnf install dnsmasq NetworkManager-wifi
```

NetworkManager utilise le service **dnsmasq** pour fournir des services DHCP et DNS aux clients du point d'accès.

2. Créer la configuration initiale du point d'accès :

```
# nmcli device wifi hotspot ifname wlp0s20f3 con-name Example-Hotspot ssid
Example-Hotspot password "password"
```

Cette commande crée un profil de connexion pour un point d'accès sur le périphérique **wlp0s20f3** qui fournit une authentification personnelle WPA2 et WPA3. Le nom du réseau sans fil, le Service Set Identifier (SSID), est **Example-Hotspot** et utilise la clé prépartagée **password**.

3. En option : Configurez le point d'accès pour qu'il ne prenne en charge que le WPA3 :

```
# nmcli connection modify Example-Hotspot 802-11-wireless-security.key-mgmt sae
```

4. Par défaut, NetworkManager utilise l'adresse IP **10.42.0.1** pour le périphérique wifi et attribue les adresses IP du sous-réseau **10.42.0.0/24** restant aux clients. Pour configurer un sous-réseau et une adresse IP différents, entrez :

```
# nmcli connection modify Example-Hotspot ipv4.addresses 192.0.2.254/24
```

L'adresse IP que vous avez définie, dans ce cas **192.0.2.254**, est celle que NetworkManager attribue au périphérique wifi. Les clients utiliseront cette adresse IP comme passerelle par défaut et serveur DNS.

5. Activer le profil de connexion :

```
# nmcli connection up Example-Hotspot
```

Vérification

1. Sur le serveur :
 - a. Vérifiez que NetworkManager a démarré le service **dnsmasq** et que le service écoute sur les ports 67 (DHCP) et 53 (DNS) :

```
# ss -tulpn | egrep ":53|:67"
udp UNCONN 0 0 10.42.0.1:53 0.0.0.0:* users:(("dnsmasq",pid=55905,fd=6))
udp UNCONN 0 0 0.0.0.0:67 0.0.0.0:* users:(("dnsmasq",pid=55905,fd=4))
tcp LISTEN 0 32 10.42.0.1:53 0.0.0.0:* users:(("dnsmasq",pid=55905,fd=7))
```


- b. Affichez le jeu de règles **nftables** pour vous assurer que NetworkManager a activé le transfert et le masquage pour le trafic provenant du sous-réseau **10.42.0.0/24**:

```
# nft list ruleset
table ip nm-shared-wlp0s20f3 {
  chain nat_postrouting {
    type nat hook postrouting priority srcnat; policy accept;
    ip saddr 10.42.0.0/24 ip daddr != 10.42.0.0/24 masquerade
  }

  chain filter_forward {
    type filter hook forward priority filter; policy accept;
    ip daddr 10.42.0.0/24 oifname "wlp0s20f3" ct state { established, related } accept
    ip saddr 10.42.0.0/24 iifname "wlp0s20f3" accept
    iifname "wlp0s20f3" oifname "wlp0s20f3" accept
    iifname "wlp0s20f3" reject
    oifname "wlp0s20f3" reject
  }
}
```

2. Sur un client équipé d'un adaptateur wifi :

a. Affiche la liste des réseaux disponibles :

```
# nmcli device wifi
IN-USE BSSID          SSID          MODE  CHAN  RATE  SIGNAL  BARS
SECURITY
      00:53:00:88:29:04 Example-Hotspot Infra 11 130 Mbit/s 62  WPA3
...
```

b. Connectez-vous au réseau sans fil **Example-Hotspot**. Voir [Gestion des connexions Wi-Fi](#).

c. Effectuez un sondage Ping sur un hôte du réseau distant ou de l'internet pour vérifier que la connexion fonctionne :

```
# ping -c 3 www.redhat.com
```

Ressources supplémentaires

- [Identifier si un appareil wifi supporte le mode point d'accès](#)
- **nm-settings(5)** page de manuel

CHAPITRE 13. MODIFIER UN NOM D'HÔTE

Le nom d'hôte d'un système est le nom figurant sur le système lui-même. Vous pouvez définir ce nom lors de l'installation de RHEL, et vous pouvez le modifier par la suite.

13.1. MODIFIER UN NOM D'HÔTE À L'AIDE DE NMCLI

Vous pouvez utiliser l'utilitaire **nmcli** pour mettre à jour le nom d'hôte du système. Notez que d'autres utilitaires peuvent utiliser un terme différent, tel que nom d'hôte statique ou persistant.

Procédure

1. Facultatif : Affiche le paramètre actuel du nom d'hôte :

```
# nmcli general hostname  
old-hostname.example.com
```

2. Définir le nouveau nom d'hôte :

```
# nmcli general hostname new-hostname.example.com
```

3. NetworkManager redémarre automatiquement le site **systemd-hostnamed** pour activer le nouveau nom. Cependant, les actions manuelles suivantes peuvent être nécessaires si vous ne voulez pas redémarrer l'hôte :

- a. Redémarrer tous les services qui ne lisent le nom d'hôte qu'au démarrage :

```
# systemctl restart <service_name>
```

- b. Les utilisateurs de l'Active Shell doivent se reconnecter pour que les modifications soient prises en compte.

Vérification

- Afficher le nom d'hôte :

```
# nmcli general hostname  
new-hostname.example.com
```

13.2. MODIFIER UN NOM D'HÔTE À L'AIDE DE HOSTNAMECTL

Vous pouvez utiliser l'utilitaire **hostnamectl** pour mettre à jour le nom d'hôte. Par défaut, cet utilitaire définit les types de noms d'hôtes suivants :

- Nom d'hôte statique : enregistré dans le fichier **/etc/hostname**. En général, les services utilisent ce nom comme nom d'hôte.
- Joli nom d'hôte : un nom descriptif, tel que **Proxy server in data center A**.
- Nom d'hôte transitoire : une valeur de repli qui est généralement reçue de la configuration du réseau.

Procédure

1. Facultatif : Affiche le paramètre actuel du nom d'hôte :

```
# hostnamectl status --static  
old-hostname.example.com
```

2. Définir le nouveau nom d'hôte :

```
# hostnamectl set-hostname new-hostname.example.com
```

Cette commande définit le nom d'hôte statique, joli et transitoire avec la nouvelle valeur. Pour définir uniquement un type spécifique, passez l'option **--static**, **--pretty**, ou **--transient** à la commande.

3. L'utilitaire **hostnamectl** redémarre automatiquement l'hôte **systemd-hostnamed** pour activer le nouveau nom. Cependant, les actions manuelles suivantes peuvent être nécessaires si vous ne voulez pas redémarrer l'hôte :

- a. Redémarrer tous les services qui ne lisent le nom d'hôte qu'au démarrage :

```
# systemctl restart <service_name>
```

- b. Les utilisateurs de l'Active Shell doivent se reconnecter pour que les modifications soient prises en compte.

Vérification

- Afficher le nom d'hôte :

```
# hostnamectl status --static  
new-hostname.example.com
```

Ressources supplémentaires

- **hostnamectl(1)**
- **systemd-hostnamed.service(8)**

CHAPITRE 14. MISE EN MIROIR DES PORTS

Les administrateurs réseau peuvent utiliser la mise en miroir des ports pour répliquer le trafic réseau entrant et sortant communiqué d'un périphérique réseau à un autre. Les administrateurs utilisent la mise en miroir des ports pour surveiller le trafic réseau et collecter des données réseau pour :

- Débuguer les problèmes de réseau et régler le flux du réseau
- Inspecter et analyser le trafic réseau pour résoudre les problèmes de réseau
- Détecter une intrusion

14.1. MISE EN MIROIR D'UNE INTERFACE RÉSEAU À L'AIDE DE NMCLI

Vous pouvez configurer la mise en miroir des ports à l'aide de NetworkManager. La procédure suivante met en miroir le trafic réseau de **enp1s0** à **enp7s0** en ajoutant des règles et des filtres de contrôle du trafic (**tc**) à l'interface réseau **enp1s0**.

Conditions préalables

- Une interface réseau vers laquelle le trafic réseau doit être mis en miroir.

Procédure

1. Ajoutez un profil de connexion réseau à partir duquel vous souhaitez créer un miroir du trafic réseau :

```
# nmcli connection add type ethernet ifname enp1s0 con-name enp1s0 autoconnect no
```

2. Attachez un **prio qdisc** à **enp1s0** pour le trafic egress (sortant) avec la poignée **10::**

```
# nmcli connection modify enp1s0 tc.qdisc "root prio handle 10:"
```

Le site **prio qdisc** fixé sans les enfants permet d'attacher des filtres.

3. Ajoutez un **qdisc** pour le trafic entrant, avec la poignée **ffff::**

```
# nmcli connection modify enp1s0 tc.qdisc "ingress handle ffff:"
```

4. Ajoutez les filtres suivants pour faire correspondre les paquets à l'entrée et à la sortie de **qdiscs**, et pour les mettre en miroir sur **enp7s0**:

```
# nmcli connection modify enp1s0 +tc.tfilter "parent ffff: matchall action mirred egress mirror dev enp7s0"
```

```
# nmcli connection modify enp1s0 +tc.tfilter "parent 10: matchall action mirred egress mirror dev enp7s0"
```

Le filtre **matchall** correspond à tous les paquets et l'action **mirred** redirige les paquets vers la destination.

5. Activer la connexion :

```
# nmcli connection up enp1s0
```

Vérification

1. Installez l'utilitaire **tcpdump**:

```
# dnf install tcpdump
```

2. Afficher le trafic reflété sur l'appareil cible (**enp7s0**) :

```
# tcpdump -i enp7s0
```

Ressources supplémentaires

- [Comment capturer des paquets réseau en utilisant **tcpdump**](#)

CHAPITRE 15. CONFIGURER NETWORKMANAGER POUR QU'IL IGNORE CERTAINS PÉRIPHÉRIQUES

Par défaut, NetworkManager gère tous les appareils. Pour ignorer certains périphériques, vous pouvez configurer NetworkManager en définissant **unmanaged**.

15.1. CONFIGURATION DE L'INTERFACE DE BOUCLAGE À L'AIDE DE NMCLI

Par défaut, NetworkManager ne gère pas l'interface de bouclage (**lo**). Après avoir créé un profil de connexion pour l'interface **lo**, vous pouvez configurer ce périphérique à l'aide de NetworkManager. Voici quelques exemples :

- Attribuer des adresses IP supplémentaires à l'interface **lo**
- Définir les adresses DNS
- Modifier la taille de l'unité de transmission maximale (MTU) de l'interface **lo**

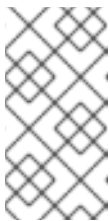
Procédure

1. Créer une nouvelle connexion de type **loopback**:

```
# nmcli connection add con-name example-loopback type loopback
```

2. Configurer des paramètres de connexion personnalisés, par exemple :
 - a. Pour attribuer une adresse IP supplémentaire à l'interface, entrez :

```
# nmcli connection modify example-loopback ipv4.addresses 192.0.2.1/24
```



NOTE

NetworkManager gère l'interface **lo** en assignant toujours les adresses IP **127.0.0.1** et **::1** qui sont persistantes à travers les redémarrages. Vous ne pouvez pas remplacer **127.0.0.1** et **::1**. Cependant, vous pouvez attribuer des adresses IP supplémentaires à l'interface.

- b. Pour définir une unité de transmission maximale (MTU) personnalisée, entrez :

```
# nmcli con mod example-loopback loopback.mtu 16384
```

- c. Pour définir une adresse IP pour votre serveur DNS, entrez :

```
# nmcli connection modify example-loopback ipv4.dns 192.0.2.0
```

Si vous définissez un serveur DNS dans le profil de connexion loopback, cette entrée est toujours disponible dans le fichier **/etc/resolv.conf**. L'entrée du serveur DNS reste indépendante de l'itinérance de l'hôte entre différents réseaux.

3. Activer la connexion :

```
# nmcli connection up example-loopback
```

Vérification

1. Affiche les paramètres de l'interface **lo**:

```
# ip address show lo

1: lo: <LOOPBACK,UP,LOWER_UP> mtu 16384 qdisc noqueue state UNKNOWN group
default qlen 1000

link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00 inet 127.0.0.1/8 scope host lo valid_lft
forever preferred_lft forever inet 192.0.2.1/24 brd 192.0.2.255 scope global lo valid_lft forever
preferred_lft forever

inet6 ::1/128 scope host
valid_lft forever preferred_lft forever
```

2. Vérifiez l'adresse DNS :

```
# cat /etc/resolv.conf

...
nameserver 192.0.2.0
...
```

15.2. CONFIGURATION PERMANENTE D'UN PÉRIPHÉRIQUE COMME NON GÉRÉ DANS NETWORKMANAGER

Vous pouvez configurer de façon permanente des appareils comme **unmanaged** en fonction de plusieurs critères, tels que le nom de l'interface, l'adresse MAC ou le type d'appareil.

Pour configurer temporairement des périphériques réseau comme **unmanaged**, voir [Configuration temporaire d'un périphérique comme non géré dans NetworkManager](#).

Procédure

1. Facultatif : Affichez la liste des appareils pour identifier l'appareil ou l'adresse MAC que vous souhaitez définir comme **unmanaged**:

```
# ip link show

...
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
mode DEFAULT group default qlen 1000
link/ether 52:54:00:74:79:56 brd ff:ff:ff:ff:ff:ff

...
```

2. Créez le fichier `/etc/NetworkManager/conf.d/99-unmanaged-devices.conf` avec le contenu suivant :

- Pour configurer une interface spécifique comme non gérée, ajoutez :

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

- Pour configurer un appareil avec une adresse MAC spécifique comme non géré, ajoutez :

```
[keyfile]
unmanaged-devices=mac:52:54:00:74:79:56
```

- Pour configurer tous les appareils d'un type spécifique comme non gérés, ajoutez :

```
[keyfile]
unmanaged-devices=type:ethernet
```

Pour définir plusieurs appareils comme non gérés, séparez les entrées du paramètre **unmanaged-devices** par des points-virgules :

3. Rechargez le service **NetworkManager**:

```
# systemctl reload NetworkManager
```

Vérification

- Affiche la liste des appareils :

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

L'état **unmanaged** à côté du périphérique **enp1s0** indique que NetworkManager ne gère pas ce périphérique.

Ressources supplémentaires

- **NetworkManager.conf(5)** page de manuel

15.3. CONFIGURER TEMPORAIREMENT UN PÉRIPHÉRIQUE COMME NON GÉRÉ DANS NETWORKMANAGER

Vous pouvez configurer temporairement des appareils comme **unmanaged**.

Utilisez cette méthode, par exemple, à des fins de test. Pour configurer de façon permanente des périphériques réseau comme **unmanaged**, voir [Configuration permanente d'un périphérique comme non géré dans NetworkManager](#).

Procédure

1. Facultatif : Affichez la liste des appareils pour identifier l'appareil que vous souhaitez définir comme **unmanaged**:

```
# nmcli device status
DEVICE TYPE   STATE   CONNECTION
```

```
enp1s0 ethernet disconnected --  
...
```

2. Mettre le dispositif **enp1s0** dans l'état **unmanaged**:

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

Vérification

- Affiche la liste des appareils :

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

L'état **unmanaged** à côté du périphérique **enp1s0** indique que NetworkManager ne gère pas ce périphérique.

Ressources supplémentaires

- **NetworkManager.conf(5)** page de manuel

CHAPITRE 16. CONFIGURER LES APPAREILS DU RÉSEAU POUR QU'ILS ACCEPTENT LE TRAFIC PROVENANT DE TOUTES LES ADRESSES MAC

Les périphériques réseau interceptent et lisent généralement les paquets que leur contrôleur est programmé pour recevoir. Vous pouvez configurer les équipements réseau pour qu'ils acceptent le trafic provenant de toutes les adresses MAC dans un commutateur virtuel ou au niveau du groupe de ports.

Vous pouvez utiliser ce mode réseau pour :

- Diagnostiquer les problèmes de connectivité du réseau
- Surveiller l'activité du réseau pour des raisons de sécurité
- Interception de données privées en transit ou intrusion dans le réseau

Vous pouvez activer ce mode pour tout type de périphérique réseau, à l'exception de **InfiniBand**.

16.1. CONFIGURER TEMPORAIREMENT UN APPAREIL POUR QU'IL ACCEPTE TOUT LE TRAFIC

Vous pouvez utiliser l'utilitaire **ip** pour configurer temporairement un périphérique réseau afin qu'il accepte tout le trafic, quelles que soient les adresses MAC.

Procédure

1. Facultatif : Affichez les interfaces réseau pour identifier celle pour laquelle vous souhaitez recevoir tout le trafic :

```
# ip address show
1: enp1s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state
DOWN group default qlen 1000
    link/ether 98:fa:9b:a4:34:09 brd ff:ff:ff:ff:ff:ff
    ...
```

2. Modifier le dispositif pour activer ou désactiver cette propriété :

- Pour activer le mode **accept-all-mac-addresses** pour **enp1s0**:

```
# ip link set enp1s0 promisc on
```

- Pour désactiver le mode **accept-all-mac-addresses** pour **enp1s0**:

```
# ip link set enp1s0 promisc off
```

Vérification

- Vérifiez que le mode **accept-all-mac-addresses** est activé :

```
# ip link show enp1s0
1: enp1s0: <NO-CARRIER,BROADCAST,MULTICAST,PROMISC,UP> mtu 1500 qdisc
fq_codel state DOWN mode DEFAULT group default qlen 1000
```



```
link/ether 98:fa:9b:a4:34:09 brd ff:ff:ff:ff:ff
```

Le drapeau **PROMISC** dans la description de l'appareil indique que le mode est activé.

16.2. CONFIGURATION PERMANENTE D'UN PÉRIPHÉRIQUE RÉSEAU POUR QU'IL ACCEPTE TOUT LE TRAFIC À L'AIDE DE NMCLI

Vous pouvez utiliser l'utilitaire **nmcli** pour configurer de façon permanente un périphérique réseau afin qu'il accepte tout le trafic, quelles que soient les adresses MAC.

Procédure

1. Facultatif : Affichez les interfaces réseau pour identifier celle pour laquelle vous souhaitez recevoir tout le trafic :

```
# ip address show
1: enp1s0: <NO-CARRIER,BROADCAST,MULTICAST,UP> mtu 1500 qdisc fq_codel state
DOWN group default qlen 1000
    link/ether 98:fa:9b:a4:34:09 brd ff:ff:ff:ff:ff
    ...
```

Vous pouvez créer une nouvelle connexion, si vous n'en avez pas.

2. Modifiez le périphérique réseau pour activer ou désactiver cette propriété.

- Pour activer le mode **ethernet.accept-all-mac-addresses** pour **enp1s0**:

```
# nmcli connection modify enp1s0 ethernet.accept-all-mac-addresses yes
```

- Pour désactiver le mode **accept-all-mac-addresses** pour **enp1s0**:

```
# nmcli connection modify enp1s0 ethernet.accept-all-mac-addresses no
```

3. Appliquez les modifications, réactivez la connexion :

```
# nmcli connection up enp1s0
```

Vérification

- Vérifiez que le mode **ethernet.accept-all-mac-addresses** est activé :

```
# nmcli connection show enp1s0
...
802-3-ethernet.accept-all-mac-addresses:1 (true)
```

Le site **802-3-ethernet.accept-all-mac-addresses: true** indique que le mode est activé.

16.3. CONFIGURATION PERMANENTE D'UN PÉRIPHÉRIQUE RÉSEAU POUR QU'IL ACCEPTE TOUT LE TRAFIC À L'AIDE DE NMSTATECTL

Vous pouvez utiliser l'utilitaire **nmstatectl** pour configurer de façon permanente un périphérique réseau afin qu'il accepte tout le trafic, quelles que soient les adresses MAC.

Conditions préalables

- Le paquet **nmstate** est installé.
- Le fichier **enp1s0.yml** que vous avez utilisé pour configurer l'appareil est disponible.

Procédure

1. Modifiez le fichier **enp1s0.yml** existant pour la connexion **enp1s0** et ajoutez-y le contenu suivant :

```
---  
interfaces:  
  - name: enp1s0  
    type: ethernet  
    state: up  
    accept-all-mac-address: true
```

2. Appliquer les paramètres du réseau :

```
# nmstatectl apply ~/enp1s0.yml
```

Vérification

- Vérifiez que le mode **802-3-ethernet.accept-all-mac-addresses** est activé :

```
# nmstatectl show enp1s0  
interfaces:  
  - name: enp1s0  
    type: ethernet  
    state: up  
    accept-all-mac-addresses: true  
...
```

Le site **802-3-ethernet.accept-all-mac-addresses: true** indique que le mode est activé.

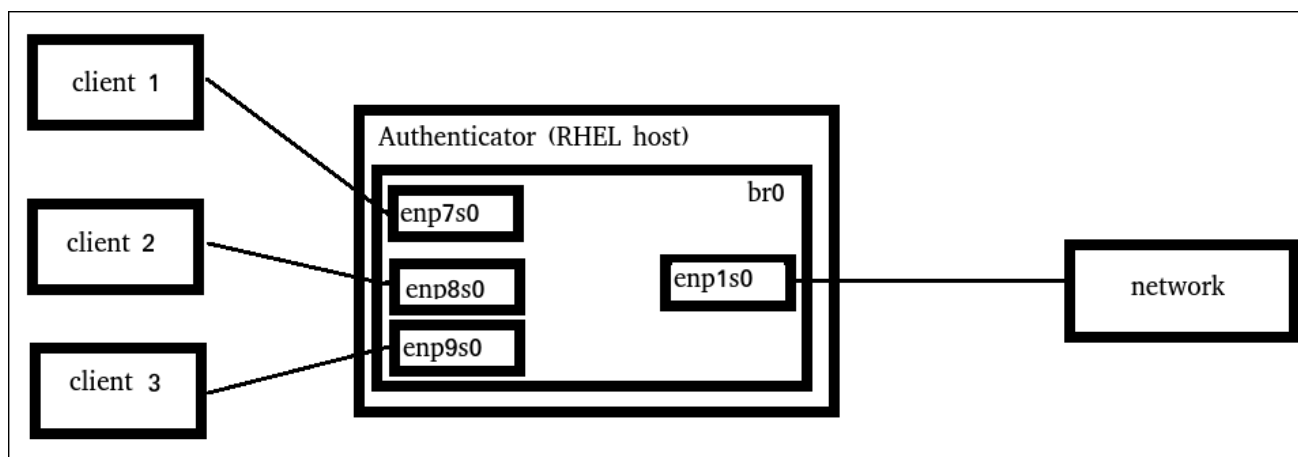
Ressources supplémentaires

- **nmstatectl(8)** page de manuel
- **/usr/share/doc/nmstate/examples/** répertoire

CHAPITRE 17. MISE EN PLACE D'UN SERVICE D'AUTHENTIFICATION RÉSEAU 802.1X POUR LES CLIENTS DU RÉSEAU LOCAL EN UTILISANT HOSTAPD AVEC FREERADIUS BACKEND

La norme IEEE 802.1X définit des méthodes d'authentification et d'autorisation sécurisées pour protéger les réseaux contre les clients non autorisés. En utilisant le service **hostapd** et FreeRADIUS, vous pouvez assurer le contrôle d'accès au réseau (NAC) dans votre réseau.

Dans cette documentation, l'hôte RHEL sert de pont pour connecter différents clients à un réseau existant. Cependant, l'hôte RHEL n'accorde l'accès au réseau qu'aux clients authentifiés.



17.1. CONDITIONS PRÉALABLES

- Une installation propre de FreeRADIUS.
Si le paquetage **freeradius** est déjà installé, supprimez le répertoire `/etc/raddb/`, désinstallez et réinstallez le paquetage. Ne réinstallez pas le paquetage à l'aide de la commande **dnf reinstall**, car les autorisations et les liens symboliques dans le répertoire `/etc/raddb/` sont alors différents.

17.2. MISE EN PLACE DE LA PASSERELLE SUR L'AUTHENTIFICATEUR

Un pont réseau est un dispositif de couche de liaison qui transfère le trafic entre les hôtes et les réseaux sur la base d'une table d'adresses MAC. Si vous configurez RHEL en tant qu'authentificateur 802.1X, ajoutez au pont les interfaces sur lesquelles l'authentification doit être effectuée et l'interface LAN.

Conditions préalables

- Le serveur possède plusieurs interfaces Ethernet.

Procédure

1. Créez l'interface de pont :

```
# nmcli connection add type bridge con-name br0 ifname br0
```

2. Attribuer les interfaces Ethernet au pont :

```
# nmcli connection add type ethernet slave-type bridge con-name br0-port1 ifname
```

```

enp1s0 master br0
# nmcli connection add type ethernet slave-type bridge con-name br0-port2 ifname
enp7s0 master br0
# nmcli connection add type ethernet slave-type bridge con-name br0-port3 ifname
enp8s0 master br0
# nmcli connection add type ethernet slave-type bridge con-name br0-port4 ifname
enp9s0 master br0

```

3. Permet au pont de transmettre des paquets EAPOL (extensible authentication protocol over LAN) :

```
# nmcli connection modify br0 group-forward-mask 8
```

4. Configurer la connexion pour activer automatiquement les ports :

```
# nmcli connection modify br0 connection.autoconnect-slaves 1
```

5. Activer la connexion :

```
# nmcli connection up br0
```

Vérification

1. Affiche l'état des liaisons des périphériques Ethernet qui sont des ports d'un pont spécifique :

```

# ip link show master br0
3: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel master
br0 state UP mode DEFAULT group default qlen 1000
    link/ether 52:54:00:62:61:0e brd ff:ff:ff:ff:ff:ff
    ...

```

2. Vérifiez si le transfert des paquets EAPOL est activé sur le périphérique **br0**:

```
# cat /sys/class/net/br0/bridge/group_fwd_mask
0x8
```

Si la commande renvoie **0x8**, le transfert est activé.

Ressources supplémentaires

- [nm-settings\(5\)](#) page de manuel

17.3. EXIGENCES EN MATIÈRE DE CERTIFICAT PAR FREERADIUS

Pour un service FreeRADIUS sécurisé, vous avez besoin de certificats TLS pour différentes raisons :

- Un certificat de serveur TLS pour les connexions cryptées au serveur. Utilisez une autorité de certification (CA) de confiance pour émettre le certificat. Le certificat du serveur requiert que le champ "extended key usage" (EKU) soit défini sur **TLS Web Server Authentication**.

- Certificats clients émis par la même autorité de certification pour le protocole d'authentification étendu de sécurité de la couche transport (EAP-TLS). EAP-TLS fournit une authentification basée sur un certificat et est activé par défaut.

Les certificats des clients doivent avoir leur champ EKU réglé sur **TLS Web Client Authentication**.



AVERTISSEMENT

Pour sécuriser la connexion, utilisez l'autorité de certification de votre entreprise ou créez votre propre autorité de certification pour émettre des certificats pour FreeRADIUS. Si vous utilisez une autorité de certification publique, vous l'autorisez à authentifier les utilisateurs et à émettre des certificats clients pour EAP-TLS.

17.4. CRÉATION D'UN ENSEMBLE DE CERTIFICATS SUR UN SERVEUR FREERADIUS À DES FINS DE TEST

À des fins de test, le paquet **freeradius** installe des scripts et des fichiers de configuration dans le répertoire **/etc/raddb/certs/** afin de créer votre propre autorité de certification (AC) et d'émettre des certificats.



IMPORTANT

Si vous utilisez la configuration par défaut, les certificats générés par ces scripts expirent au bout de 60 jours et les clés utilisent un mot de passe peu sûr (quel qu'il soit). Vous pouvez toutefois personnaliser la configuration de l'autorité de certification, du serveur et du client.

Une fois la procédure effectuée, les fichiers suivants, dont vous aurez besoin dans la suite de cette documentation, sont créés :

- **/etc/raddb/certs/ca.pem**: Certificat CA
- **/etc/raddb/certs/server.key**: Clé privée du certificat du serveur
- **/etc/raddb/certs/server.pem**: Certificat du serveur
- **/etc/raddb/certs/client.key**: Clé privée du certificat du client
- **/etc/raddb/certs/client.pem**: Certificat du client

Conditions préalables

- Vous avez installé le paquetage **freeradius**.

Procédure

1. Allez dans le répertoire **/etc/raddb/certs/**:

```
# cd /etc/raddb/certs/
```

2. Facultatif : Personnalisez la configuration de l'autorité de certification dans le fichier **/etc/raddb/certs/ca.cnf**:

```
...
[ req ]
default_bits      = 2048
input_password    = ca_password
output_password   = ca_password
...
[certificate_authority]
countryName       = US
stateOrProvinceName = North Carolina
localityName      = Raleigh
organizationName  = Example Inc.
emailAddress      = admin@example.org
commonName        = "Example Certificate Authority"
...
```

3. Optionnel : Personnalisez la configuration du serveur dans le fichier **/etc/raddb/certs/server.cnf** :

```
...
[ CA_default ]
default_days      = 730
...
[ req ]
distinguished_name = server
default_bits      = 2048
input_password    = key_password
output_password   = key_password
...
[server]
countryName       = US
stateOrProvinceName = North Carolina
localityName      = Raleigh
organizationName  = Example Inc.
emailAddress      = admin@example.org
commonName        = "Example Server Certificate"
...
```

4. Facultatif : Personnaliser la configuration du client dans le fichier **/etc/raddb/certs/client.cnf** :

```
...
[ CA_default ]
default_days      = 365
...
[ req ]
distinguished_name = client
default_bits      = 2048
input_password     = password_on_private_key
output_password    = password_on_private_key
...
[client]
countryName       = US
stateOrProvinceName = North Carolina
```

```

localityName      = Raleigh
organizationName  = Example Inc.
emailAddress      = user@example.org
commonName        = user@example.org
...

```

5. Créer les certificats :

```
# make all
```

6. Modifier le groupe du fichier `/etc/raddb/certs/server.pem` en **radiusd**:

```
# chgrp radiusd /etc/raddb/certs/server.pem
```

Ressources supplémentaires

- `/etc/raddb/certs/README.md`

17.5. CONFIGURATION DE FREERADIUS POUR AUTHENTIFIER LES CLIENTS DU RÉSEAU EN TOUTE SÉCURITÉ À L'AIDE D'EAP

FreeRADIUS prend en charge différentes méthodes du protocole d'authentification extensible (EAP). Cependant, pour un réseau sécurisé, configurez FreeRADIUS pour qu'il prenne en charge uniquement les méthodes d'authentification EAP sécurisées suivantes :

- EAP-TLS (transport layer security) utilise une connexion TLS sécurisée pour authentifier les clients à l'aide de certificats. Pour utiliser EAP-TLS, vous avez besoin de certificats clients TLS pour chaque client du réseau et d'un certificat serveur pour le serveur. Notez que les certificats doivent être émis par la même autorité de certification (AC). Utilisez toujours votre propre autorité de certification pour créer des certificats, car tous les certificats clients émis par l'autorité de certification que vous utilisez peuvent s'authentifier auprès de votre serveur FreeRADIUS.
- EAP-TTLS (tunneled transport layer security) utilise une connexion TLS sécurisée et authentifie les clients à l'aide de mécanismes tels que le protocole d'authentification par mot de passe (PAP) ou le protocole d'authentification par challenge handshake (CHAP). Pour utiliser EAP-TTLS, vous avez besoin d'un certificat de serveur TLS.
- EAP-PEAP (protected extensible authentication protocol) utilise une connexion TLS sécurisée comme protocole d'authentification externe pour établir le tunnel. L'authentificateur authentifie le certificat du serveur RADIUS. Ensuite, le demandeur s'authentifie par le biais du tunnel crypté en utilisant le protocole d'authentification Microsoft challenge handshake version 2 (MS-CHAPv2) ou d'autres méthodes.



NOTE

Les fichiers de configuration par défaut de FreeRADIUS servent de documentation et décrivent tous les paramètres et directives. Si vous souhaitez désactiver certaines fonctionnalités, commentez-les au lieu de supprimer les parties correspondantes dans les fichiers de configuration. Cela vous permet de préserver la structure des fichiers de configuration et la documentation incluse.

Conditions préalables

- Vous avez installé le paquetage **freeradius**.
- Les fichiers de configuration du répertoire **/etc/raddb/** sont inchangés et fournis par le paquetage **freeradius**.
- Les fichiers suivants existent sur le serveur :
 - Clé privée TLS de l'hôte FreeRADIUS : **/etc/raddb/certs/server.key**
 - Certificat de serveur TLS de l'hôte FreeRADIUS : **/etc/raddb/certs/server.pem**
 - Certificat d'autorité de certification TLS : **/etc/raddb/certs/ca.pem**

Si vous stockez les fichiers à un autre endroit ou s'ils portent des noms différents, définissez les paramètres **private_key_file**, **certificate_file** et **ca_file** dans le fichier **/etc/raddb/mods-available/eap** en conséquence.

Procédure

1. Si le site **/etc/raddb/certs/dh** avec les paramètres Diffie-Hellman (DH) n'existe pas, créez-en un. Par exemple, pour créer un fichier DH avec une primauté de 2048 bits, entrez :

```
# openssl dhparam -out /etc/raddb/certs/dh 2048
```

Pour des raisons de sécurité, n'utilisez pas un fichier DH avec moins de 2048 bits. Selon le nombre de bits, la création du fichier peut prendre plusieurs minutes.

2. Définissez des autorisations sécurisées pour la clé privée TLS, le certificat du serveur, le certificat de l'autorité de certification et le fichier contenant les paramètres DH :

```
# chmod 640 /etc/raddb/certs/server.key /etc/raddb/certs/server.pem
/etc/raddb/certs/ca.pem /etc/raddb/certs/dh
# chown root:radiusd /etc/raddb/certs/server.key /etc/raddb/certs/server.pem
/etc/raddb/certs/ca.pem /etc/raddb/certs/dh
```

3. Modifiez le fichier **/etc/raddb/mods-available/eap**:

- a. Définissez le mot de passe de la clé privée dans le paramètre **private_key_password**:

```
eap {
  ...
  tls-config tls-common {
    ...
    private_key_password = key_password
    ...
  }
}
```

- b. En fonction de votre environnement, définissez le paramètre **default_eap_type** de la directive **eap** en fonction du type d'EAP primaire que vous utilisez :

```
eap {
  ...
  default_eap_type = tls
```



```

} ...
}

```

Pour un environnement sécurisé, utilisez uniquement **ttls**, **tls** ou **peap**.

- c. Commentez les directives **md5** pour désactiver la méthode d'authentification non sécurisée EAP-MD5 :

```

eap {
  ...
  # md5 {
  # }
  ...
}

```

Notez que, dans le fichier de configuration par défaut, les autres méthodes d'authentification EAP non sécurisées sont commentées par défaut.

4. Modifiez le fichier **/etc/raddb/sites-available/default** et mettez en commentaire toutes les méthodes d'authentification autres que **eap**:

```

authenticate {
  ...
  # Auth-Type PAP {
  #   pap
  # }

  # Auth-Type CHAP {
  #   chap
  # }

  # Auth-Type MS-CHAP {
  #   mschap
  # }

  # mschap

  # digest
  ...
}

```

Seule l'option EAP est activée et les méthodes d'authentification en texte clair sont désactivées.

5. Modifiez le fichier **/etc/raddb/clients.conf**:

- a. Définissez un mot de passe sécurisé dans les directives client **localhost** et **localhost_ipv6**:

```

client localhost {
  ipaddr = 127.0.0.1
  ...
  secret = client_password
  ...
}

client localhost_ipv6 {

```

```

    ipv6addr = ::1
    secret = client_password
}

```

- b. Si des clients RADIUS, tels que des authentificateurs de réseau, sur des hôtes distants doivent pouvoir accéder au service FreeRADIUS, ajoutez les directives client correspondantes pour eux :

```

client hostapd.example.org {
    ipaddr = 192.0.2.2/32
    secret = client_password
}

```

Le paramètre **ipaddr** accepte les adresses IPv4 et IPv6, et vous pouvez utiliser la notation facultative CIDR (classless inter-domain routing) pour spécifier des plages. Toutefois, vous ne pouvez définir qu'une seule valeur pour ce paramètre. Par exemple, pour accorder l'accès à une adresse IPv4 et IPv6, ajoutez deux directives client.

Utilisez un nom descriptif pour la directive client, tel qu'un nom d'hôte ou un mot décrivant l'endroit où la plage IP est utilisée.

6. Si vous souhaitez utiliser EAP-TTLS ou EAP-PEAP, ajoutez les utilisateurs au fichier **/etc/raddb/users**:

```

example_user    Cleartext-Password := "user_password"

```

Pour les utilisateurs qui doivent utiliser l'authentification par certificat (EAP-TLS), n'ajoutez aucune entrée.

7. Vérifier les fichiers de configuration :

```

# radiusd -XC
...
Configuration appears to be OK

```

8. Activez et démarrez le service **radiusd**:

```

# systemctl enable --now radiusd

```

Vérification

- [Test de l'authentification EAP-TTLS contre un serveur ou un authentificateur FreeRADIUS](#)
- [Test de l'authentification EAP-TLS contre un serveur ou un authentificateur FreeRADIUS](#)

Résolution de problèmes

1. Arrêtez le service **radiusd**:

```

# systemctl stop radiusd

```

2. Démarrer le service en mode débogage :

```
# radiusd -X
...
Ready to process requests
```

3. Effectuer les tests d'authentification sur l'hôte FreeRADIUS, comme indiqué dans la section **Verification**.

Prochaines étapes

- Désactivez les méthodes d'authentification qui ne sont plus nécessaires et les autres fonctions que vous n'utilisez pas.

17.6. CONFIGURER HOSTAPD COMME AUTHENTICATEUR DANS UN RÉSEAU CÂBLÉ

Le service host access point daemon (**hostapd**) peut servir d'authentificateur dans un réseau câblé pour fournir l'authentification 802.1X. Pour ce faire, le service **hostapd** a besoin d'un serveur RADIUS qui authentifie les clients.

Le service **hostapd** fournit un serveur RADIUS intégré. Cependant, n'utilisez le serveur RADIUS intégré qu'à des fins de test. Pour les environnements de production, utilisez le serveur FreeRADIUS, qui prend en charge des fonctions supplémentaires, telles que différentes méthodes d'authentification et de contrôle d'accès.



IMPORTANT

Le service **hostapd** n'interagit pas avec le plan de trafic. Il agit uniquement en tant qu'authentificateur. Par exemple, utilisez un script ou un service qui utilise l'interface de contrôle **hostapd** pour autoriser ou refuser le trafic en fonction du résultat des événements d'authentification.

Conditions préalables

- Vous avez installé le paquetage **hostapd**.
- Le serveur FreeRADIUS a été configuré et il est prêt à authentifier les clients.

Procédure

1. Créez le fichier **/etc/hostapd/hostapd.conf** avec le contenu suivant :

```
# General settings of hostapd
# =====

# Control interface settings
ctrl_interface=/var/run/hostapd
ctrl_interface_group=wheel

# Enable logging for all modules
logger_syslog=-1
logger_stdout=-1

# Log level
logger_syslog_level=2
```

```

logger_stdout_level=2

# Wired 802.1X authentication
# =====

# Driver interface type
driver=wired

# Enable IEEE 802.1X authorization
ieee8021x=1

# Use port access entry (PAE) group address
# (01:80:c2:00:00:03) when sending EAPOL frames
use_pae_group_addr=1

# Network interface for authentication requests
interface=br0

# RADIUS client configuration
# =====

# Local IP address used as NAS-IP-Address
own_ip_addr=192.0.2.2

# Unique NAS-Identifiant within scope of RADIUS server
nas_identifiant=hostapd.example.org

# RADIUS authentication server
auth_server_addr=192.0.2.1
auth_server_port=1812
auth_server_shared_secret=client_password

# RADIUS accounting server
acct_server_addr=192.0.2.1
acct_server_port=1813
acct_server_shared_secret=client_password

```

Pour plus de détails sur les paramètres utilisés dans cette configuration, voir leurs descriptions dans le fichier de configuration de l'exemple **/usr/share/doc/hostapd/hostapd.conf**.

2. Activez et démarrez le service **hostapd**:

```
# systemctl enable --now hostapd
```

Vérification

- Voir :
 - [Test de l'authentification EAP-TTLS contre un serveur ou un authentificateur FreeRADIUS](#)
 - [Test de l'authentification EAP-TLS contre un serveur ou un authentificateur FreeRADIUS](#)

Résolution de problèmes

1. Arrêtez le service **hostapd**:

```
# systemctl stop hostapd
```

2. Démarrer le service en mode débogage :

```
# hostapd -d /etc/hostapd/hostapd.conf
```

3. Effectuer les tests d'authentification sur l'hôte FreeRADIUS, comme indiqué dans la section **Verification**.

Ressources supplémentaires

- **hostapd.conf(5)** page de manuel
- **/usr/share/doc/hostapd/hostapd.conf** fichier

17.7. TEST DE L'AUTHENTIFICATION EAP-TTLS CONTRE UN SERVEUR OU UN AUTHENTICATEUR FREERADIUS

Pour vérifier si l'authentification à l'aide du protocole d'authentification extensible (EAP) sur la sécurité de la couche de transport tunnelisée (EAP-TTLS) fonctionne comme prévu, exécutez la procédure suivante :

- Après avoir configuré le serveur FreeRADIUS
- Après avoir configuré le service **hostapd** en tant qu'authentificateur pour l'authentification réseau 802.1X.

Les résultats des utilitaires de test utilisés dans cette procédure fournissent des informations supplémentaires sur la communication EAP et vous aident à résoudre les problèmes.

Conditions préalables

- Lorsque vous souhaitez vous authentifier auprès de :
 - Un serveur FreeRADIUS :
 - L'utilitaire **eapol_test**, fourni par le paquetage **hostapd**, est installé.
 - Le client sur lequel vous exécutez cette procédure a été autorisé dans les bases de données clients du serveur FreeRADIUS.
 - Un authentificateur, l'utilitaire **wpa_supplicant**, fourni par le paquet du même nom, est installé.
- Vous avez enregistré le certificat de l'autorité de certification (CA) dans le fichier **/etc/pki/tls/certs/ca.pem**.

Procédure

1. Créez le fichier **/etc/wpa_supplicant/wpa_supplicant-TTLS.conf** avec le contenu suivant :

```
ap_scan=0
```

```

network={
    eap=TTLS
    eapol_flags=0
    key_mgmt=IEEE8021X

    # Anonymous identity (sent in unencrypted phase 1)
    # Can be any string
    anonymous_identity="anonymous"

    # Inner authentication (sent in TLS-encrypted phase 2)
    phase2="auth=PAP"
    identity="example_user"
    password="user_password"

    # CA certificate to validate the RADIUS server's identity
    ca_cert="/etc/pki/tls/certs/ca.pem"
}

```

2. Pour s'authentifier auprès de :

- Un serveur FreeRADIUS, entrez :

```

# eapol_test -c /etc/wpa_supplicant/wpa_supplicant-TTLS.conf -a 192.0.2.1 -s
client_password
...
EAP: Status notification: remote certificate verification (param=success)
...
CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
...
SUCCESS

```

L'option **-a** définit l'adresse IP du serveur FreeRADIUS, et l'option **-s** spécifie le mot de passe de l'hôte sur lequel vous exécutez la commande dans la configuration client du serveur FreeRADIUS.

- Un authentificateur, entrez :

```

# wpa_supplicant -c /etc/wpa_supplicant/wpa_supplicant-TTLS.conf -D wired -i
enp0s31f6
...
enp0s31f6: CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
...

```

L'option **-i** spécifie le nom de l'interface réseau sur laquelle **wpa_supplicant** envoie des paquets EAPOL (extended authentication protocol over LAN).

Pour plus d'informations sur le débogage, ajoutez l'option **-d** à la commande.

Ressources supplémentaires

- `/usr/share/doc/wpa_supplicant/wpa_supplicant.conf` fichier

17.8. TEST DE L'AUTHENTIFICATION EAP-TLS CONTRE UN SERVEUR OU UN AUTHENTICATEUR FREERADIUS

Pour vérifier si l'authentification à l'aide du protocole d'authentification extensible (EAP) transport layer security (EAP-TLS) fonctionne comme prévu, exécutez la procédure suivante :

- Après avoir configuré le serveur FreeRADIUS
- Après avoir configuré le service **hostapd** en tant qu'authentificateur pour l'authentification réseau 802.1X.

Les résultats des utilitaires de test utilisés dans cette procédure fournissent des informations supplémentaires sur la communication EAP et vous aident à résoudre les problèmes.

Conditions préalables

- Lorsque vous souhaitez vous authentifier auprès de :
 - Un serveur FreeRADIUS :
 - L'utilitaire **eapol_test**, fourni par le paquetage **hostapd**, est installé.
 - Le client sur lequel vous exécutez cette procédure a été autorisé dans les bases de données clients du serveur FreeRADIUS.
 - Un authentificateur, l'utilitaire **wpa_supplicant**, fourni par le paquet du même nom, est installé.
- Vous avez enregistré le certificat de l'autorité de certification (CA) dans le fichier **/etc/pki/tls/certs/ca.pem**.
- L'autorité de certification qui a émis le certificat du client est la même que celle qui a émis le certificat du serveur FreeRADIUS.
- Vous avez stocké le certificat du client dans le fichier **/etc/pki/tls/certs/client.pem**.
- Vous avez stocké la clé privée du client dans le fichier **/etc/pki/tls/private/client.key**

Procédure

1. Créez le fichier **/etc/wpa_supplicant/wpa_supplicant-TLS.conf** avec le contenu suivant :

```
ap_scan=0

network={
    eap=TLS
    eapol_flags=0
    key_mgmt=IEEE8021X

    identity="user@example.org"
    client_cert="/etc/pki/tls/certs/client.pem"
    private_key="/etc/pki/tls/private/client.key"
    private_key_passwd="password_on_private_key"
```

```
# CA certificate to validate the RADIUS server's identity
ca_cert="/etc/pki/tls/certs/ca.pem"
}
```

2. Pour s'authentifier auprès de :

- Un serveur FreeRADIUS, entrez :

```
# eapol_test -c /etc/wpa_supplicant/wpa_supplicant-TLS.conf -a 192.0.2.1 -s
client_password
...
EAP: Status notification: remote certificate verification (param=success)
...
CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
...
SUCCESS
```

L'option **-a** définit l'adresse IP du serveur FreeRADIUS, et l'option **-s** spécifie le mot de passe de l'hôte sur lequel vous exécutez la commande dans la configuration client du serveur FreeRADIUS.

- Un authentificateur, entrez :

```
# wpa_supplicant -c /etc/wpa_supplicant/wpa_supplicant-TLS.conf -D wired -i
enp0s31f6
...
enp0s31f6: CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
...
```

L'option **-i** spécifie le nom de l'interface réseau sur laquelle **wpa_supplicant** envoie des paquets EAPOL (extended authentication protocol over LAN).

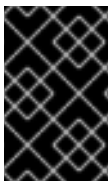
Pour plus d'informations sur le débogage, ajoutez l'option **-d** à la commande.

Ressources supplémentaires

- `/usr/share/doc/wpa_supplicant/wpa_supplicant.conf` fichier

17.9. BLOQUER ET AUTORISER LE TRAFIC EN FONCTION DES ÉVÉNEMENTS D'AUTHENTIFICATION HOSTAPD

Le service **hostapd** n'interagit pas avec le plan de trafic. Il agit uniquement en tant qu'authentificateur. Cependant, vous pouvez écrire un script pour autoriser ou refuser le trafic en fonction du résultat des événements d'authentification.



IMPORTANT

Cette procédure n'est pas prise en charge et ne constitue pas une solution prête pour l'entreprise. Elle montre seulement comment bloquer ou autoriser le trafic en évaluant les événements récupérés par **hostapd_cli**.

Lorsque le service systemd **802-1x-tr-mgmt** démarre, RHEL bloque tout le trafic sur le port d'écoute de **hostapd**, à l'exception des paquets EAPOL (extensible authentication protocol over LAN), et utilise l'utilitaire **hostapd_cli** pour se connecter à l'interface de contrôle **hostapd**. Le script `/usr/local/bin/802-`

1x-tr-mgmt évalue ensuite les événements. En fonction des différents événements reçus par **hostapd_cli**, le script autorise ou bloque le trafic pour les adresses MAC. Notez que lorsque le service **802-1x-tr-mgmt** s'arrête, tout le trafic est automatiquement autorisé à nouveau.

Effectuez cette procédure sur le serveur **hostapd**.

Conditions préalables

- Le service **hostapd** a été configuré et est prêt à authentifier les clients.

Procédure

1. Créez le fichier **/usr/local/bin/802-1x-tr-mgmt** avec le contenu suivant :

```
#!/bin/sh

if [ "x$1" == "xblock_all" ]
then

    nft delete table bridge tr-mgmt-br0 2>/dev/null || true
    nft -f - << EOF
table bridge tr-mgmt-br0 {
    set allowed_macs {
        type ether_addr
    }

    chain accesscontrol {
        ether saddr @allowed_macs accept
        ether daddr @allowed_macs accept
        drop
    }

    chain forward {
        type filter hook forward priority 0; policy accept;
        meta ibname "br0" jump accesscontrol
    }
}
EOF
    echo "802-1x-tr-mgmt Blocking all traffic through br0. Traffic for given host will be allowed
after 802.1x authentication"

elif [ "x$1" == "xallow_all" ]
then

    nft delete table bridge tr-mgmt-br0
    echo "802-1x-tr-mgmt Allowed all forwarding again"

fi

case ${2:-NOTANEVENT} in

    AP-STA-CONNECTED | CTRL-EVENT-EAP-SUCCESS | CTRL-EVENT-EAP-
SUCCESS2)
        nft add element bridge tr-mgmt-br0 allowed_macs { $3 }
        echo "$1: Allowed traffic from $3"
        ;;
```

```

AP-STA-DISCONNECTED | CTRL-EVENT-EAP-FAILURE)
  nft delete element bridge tr-mgmt-br0 allowed_macs { $3 }
  echo "802-1x-tr-mgmt $1: Denied traffic from $3"
  ;;

```

```
esac
```

2. Créez le fichier de service `/etc/systemd/system/802-1x-tr-mgmt@.service` systemd avec le contenu suivant :

```

[Unit]
Description=Example 802.1x traffic management for hostapd
After=hostapd.service
After=sys-devices-virtual-net-%i.device

[Service]
Type=simple
ExecStartPre=-/bin/sh -c '/usr/sbin/tc qdisc del dev %i ingress > /dev/null 2>&1'
ExecStartPre=-/bin/sh -c '/usr/sbin/tc qdisc del dev %i clsact > /dev/null 2>&1'
ExecStartPre=/usr/sbin/tc qdisc add dev %i clsact
ExecStartPre=/usr/sbin/tc filter add dev %i ingress pref 10000 protocol 0x888e matchall
action ok index 100
ExecStartPre=/usr/sbin/tc filter add dev %i ingress pref 10001 protocol all matchall action
drop index 101
ExecStart=/usr/sbin/hostapd_cli -i %i -a /usr/local/bin/802-1x-tr-mgmt
ExecStopPost=-/usr/sbin/tc qdisc del dev %i clsact

[Install]
WantedBy=multi-user.target

```

3. Recharger systemd :

```
# systemctl daemon-reload
```

4. Activez et démarrez le service `802-1x-tr-mgmt` avec le nom de l'interface sur laquelle `hostapd` écoute :

```
# systemctl enable --now 802-1x-tr-mgmt@br0.service
```

Vérification

- Authentifier un client sur le réseau. Voir :
 - [Test de l'authentification EAP-TTLS contre un serveur ou un authentificateur FreeRADIUS](#)
 - [Test de l'authentification EAP-TLS contre un serveur ou un authentificateur FreeRADIUS](#)

Ressources supplémentaires

- `systemd.service(5)` page de manuel

CHAPITRE 18. AUTHENTIFICATION D'UN CLIENT RHEL AU RÉSEAU À L'AIDE DE LA NORME 802.1X AVEC UN CERTIFICAT STOCKÉ SUR LE SYSTÈME DE FICHIERS

Les administrateurs utilisent fréquemment le contrôle d'accès au réseau (NAC) basé sur les ports et la norme IEEE 802.1X pour protéger un réseau contre les clients LAN et Wi-Fi non autorisés.

18.1. CONFIGURATION DE L'AUTHENTIFICATION RÉSEAU 802.1X SUR UNE CONNEXION ETHERNET EXISTANTE À L'AIDE DE NMCLI

À l'aide de l'utilitaire **nmcli**, vous pouvez configurer le client pour qu'il s'authentifie sur le réseau. Par exemple, configurez l'authentification TLS dans un profil de connexion Ethernet existant de NetworkManager nommé **enp1s0** pour s'authentifier sur le réseau.

Conditions préalables

- Le réseau prend en charge l'authentification réseau 802.1X.
- Le profil de connexion Ethernet existe dans NetworkManager et possède une configuration IP valide.
- Les fichiers suivants, nécessaires à l'authentification TLS, existent sur le client :
 - La clé client stockée se trouve dans le fichier **/etc/pki/tls/private/client.key**, qui appartient à l'utilisateur **root** et ne peut être lu que par lui.
 - Le certificat du client est stocké dans le fichier **/etc/pki/tls/certs/client.crt**.
 - Le certificat de l'autorité de certification (CA) est stocké dans le fichier **/etc/pki/tls/certs/ca.crt**.
- Le paquet **wpa_supplicant** est installé.

Procédure

1. Définissez le protocole d'authentification extensible (EAP) sur **tls** et les chemins d'accès au certificat du client et au fichier clé :

```
# nmcli connection modify enp1s0 802-1x.eap tls 802-1x.client-cert  
/etc/pki/tls/certs/client.crt 802-1x.private-key /etc/pki/tls/certs/certs/client.key
```

Notez que vous devez définir les paramètres **802-1x.eap**, **802-1x.client-cert** et **802-1x.private-key** en une seule commande.

2. Définissez le chemin d'accès au certificat de l'autorité de certification :

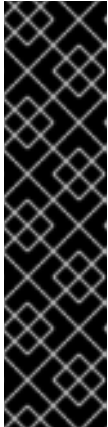
```
# nmcli connection modify enp1s0 802-1x.ca-cert /etc/pki/tls/certs/ca.crt
```

3. Définir l'identité de l'utilisateur utilisée dans le certificat :

```
# nmcli connection modify enp1s0 802-1x.identity user@example.com
```

4. Optionnellement, le mot de passe est stocké dans la configuration :

```
# nmcli connection modify enp1s0 802-1x.private-key-password password
```



IMPORTANT

Par défaut, NetworkManager stocke le mot de passe en clair dans le fichier `/etc/sysconfig/network-scripts/keys-connection_name` qui n'est lisible que par l'utilisateur **root**. Cependant, les mots de passe en texte clair dans un fichier de configuration peuvent présenter un risque pour la sécurité.

Pour augmenter la sécurité, définissez le paramètre **802-1x.password-flags** sur **0x1**. Avec ce paramètre, sur les serveurs avec l'environnement de bureau GNOME ou **nm-applet** en cours d'exécution, NetworkManager récupère le mot de passe à partir de ces services. Dans les autres cas, NetworkManager demande le mot de passe.

5. Activer le profil de connexion :

```
# nmcli connection up enp1s0
```

Vérification

- Accéder aux ressources du réseau qui nécessitent une authentification réseau.

Ressources supplémentaires

- [Configuration d'une connexion Ethernet](#)
- **nm-settings(5)** page de manuel
- **nmcli(1)** page de manuel

18.2. CONFIGURATION D'UNE CONNEXION ETHERNET STATIQUE AVEC AUTHENTIFICATION RÉSEAU 802.1X À L'AIDE DE NMSTATECTL

L'utilitaire **nmstate** permet de créer une connexion Ethernet qui utilise la norme 802.1X pour authentifier le client. Par exemple, ajoutez une connexion Ethernet pour l'interface **enp1s0** avec les paramètres suivants :

- Une adresse IPv4 statique - **192.0.2.1** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **192.0.2.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::fffe**
- Un serveur DNS IPv4 - **192.0.2.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**
- 802.1X authentification réseau utilisant le protocole d'authentification extensible (EAP) **TLS**



NOTE

La bibliothèque **nmstate** ne prend en charge que la méthode EAP **TLS**.

Conditions préalables

- Le réseau prend en charge l'authentification réseau 802.1X.
- Le nœud géré utilise NetworkManager.
- Les fichiers suivants, nécessaires à l'authentification TLS, existent sur le client :
 - La clé client stockée se trouve dans le fichier **/etc/pki/tls/private/client.key**, qui appartient à l'utilisateur **root** et ne peut être lu que par lui.
 - Le certificat du client est stocké dans le fichier **/etc/pki/tls/certs/client.crt**.
 - Le certificat de l'autorité de certification (CA) est stocké dans le fichier **/etc/pki/tls/certs/ca.crt**.

Procédure

1. Créez un fichier YAML, par exemple **~/create-ethernet-profile.yml**, avec le contenu suivant :

```
---
interfaces:
- name: enp1s0
  type: ethernet
  state: up
  ipv4:
    enabled: true
    address:
    - ip: 192.0.2.1
      prefix-length: 24
    dhcp: false
  ipv6:
    enabled: true
    address:
    - ip: 2001:db8:1::1
      prefix-length: 64
    autoconf: false
    dhcp: false
  802.1x:
    ca-cert: /etc/pki/tls/certs/ca.crt
    client-cert: /etc/pki/tls/certs/client.crt
    eap-methods:
    - tls
    identity: client.example.org
    private-key: /etc/pki/tls/private/client.key
    private-key-password: password
  routes:
  config:
  - destination: 0.0.0.0/0
    next-hop-address: 192.0.2.254
    next-hop-interface: enp1s0
  - destination: ::0
```

```
next-hop-address: 2001:db8:1::fffe
next-hop-interface: enp1s0
dns-resolver:
config:
search:
- example.com
server:
- 192.0.2.200
- 2001:db8:1::ffbb
```

2. Appliquer les paramètres au système :

```
# nmstatectl apply ~/create-ethernet-profile.yml
```

Vérification

- Accéder aux ressources du réseau qui nécessitent une authentification réseau.

18.3. CONFIGURATION D'UNE CONNEXION ETHERNET STATIQUE AVEC AUTHENTIFICATION RÉSEAU 802.1X À L'AIDE DU RÔLE DE SYSTÈME RHEL RÉSEAU

À l'aide du rôle système **network** RHEL, vous pouvez automatiser la création d'une connexion Ethernet qui utilise la norme 802.1X pour authentifier le client. Par exemple, ajoutez à distance une connexion Ethernet pour l'interface **enp1s0** avec les paramètres suivants en exécutant un script Ansible :

- Une adresse IPv4 statique - **192.0.2.1** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **192.0.2.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::fffe**
- Un serveur DNS IPv4 - **192.0.2.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**
- 802.1X authentification réseau utilisant le protocole d'authentification extensible (EAP) **TLS**

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.

- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible
- Le réseau prend en charge l'authentification réseau 802.1X.
- Les nœuds gérés utilisent NetworkManager.
- Les fichiers suivants, nécessaires à l'authentification TLS, existent sur le nœud de contrôle :
 - La clé du client est stockée dans le fichier **/srv/data/client.key**.
 - Le certificat du client est stocké dans le fichier **/srv/data/client.crt**.
 - Le certificat de l'autorité de certification (CA) est stocké dans le fichier **/srv/data/ca.crt**.

Procédure

1. Créez un fichier playbook, par exemple **~/enable-802.1x.yml** avec le contenu suivant :

```

---
- name: Configure an Ethernet connection with 802.1X authentication
  hosts: managed-node-01.example.com
  tasks:
    - name: Copy client key for 802.1X authentication
      copy:
        src: "/srv/data/client.key"
        dest: "/etc/pki/tls/private/client.key"
        mode: 0600

    - name: Copy client certificate for 802.1X authentication
      copy:
        src: "/srv/data/client.crt"
        dest: "/etc/pki/tls/certs/client.crt"

    - name: Copy CA certificate for 802.1X authentication
      copy:
        src: "/srv/data/ca.crt"
        dest: "/etc/pki/ca-trust/source/anchors/ca.crt"

    - include_role:
        name: rhel-system-roles.network

  vars:
    network_connections:
      - name: enp1s0
        type: ethernet
        autoconnect: yes
        ip:
          address:
            - 192.0.2.1/24
            - 2001:db8:1::1/64
          gateway4: 192.0.2.254
          gateway6: 2001:db8:1::ffe
        dns:
          - 192.0.2.200
          - 2001:db8:1::ffbb
    
```

```

dns_search:
  - example.com
ieee802_1x:
  identity: user_name
  eap: tls
  private_key: "/etc/pki/tls/private/client.key"
  private_key_password: "password"
  client_cert: "/etc/pki/tls/certs/client.crt"
  ca_cert: "/etc/pki/ca-trust/source/anchors/ca.crt"
  domain_suffix_match: example.com
  state: up

```

2. Exécutez le manuel de jeu :

```
# ansible-playbook ~/enable-802.1x.yml
```

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

18.4. CONFIGURER UNE CONNEXION WIFI AVEC L'AUTHENTIFICATION RÉSEAU 802.1X EN UTILISANT LE RÔLE RÉSEAU RHEL SYSTEM ROLE

À l'aide des rôles système RHEL, vous pouvez automatiser la création d'une connexion wifi. Par exemple, vous pouvez ajouter à distance un profil de connexion sans fil pour l'interface **wlp1s0** à l'aide d'un playbook Ansible. Le profil créé utilise la norme 802.1X pour authentifier le client sur un réseau wifi. Le playbook configure le profil de connexion pour utiliser DHCP. Pour configurer des paramètres IP statiques, adaptez les paramètres du dictionnaire **ip** en conséquence.

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Le réseau prend en charge l'authentification réseau 802.1X.
- Vous avez installé le paquetage **wpa_supplicant** sur le nœud géré.
- DHCP est disponible dans le réseau du nœud géré.
- Les fichiers suivants, nécessaires à l'authentification TLS, existent sur le nœud de contrôle :
 - La clé du client est stockée dans le fichier **/srv/data/client.key**.

- Le certificat du client est stocké dans le fichier **/srv/data/client.crt**.
- Le certificat d'autorité de certification est stocké dans le fichier **/srv/data/ca.crt**.

Procédure

1. Créez un fichier playbook, par exemple **~/enable-802.1x.yml** avec le contenu suivant :

```

---
- name: Configure a wifi connection with 802.1X authentication
  hosts: managed-node-01.example.com
  tasks:
    - name: Copy client key for 802.1X authentication
      copy:
        src: "/srv/data/client.key"
        dest: "/etc/pki/tls/private/client.key"
        mode: 0400

    - name: Copy client certificate for 802.1X authentication
      copy:
        src: "/srv/data/client.crt"
        dest: "/etc/pki/tls/certs/client.crt"

    - name: Copy CA certificate for 802.1X authentication
      copy:
        src: "/srv/data/ca.crt"
        dest: "/etc/pki/ca-trust/source/anchors/ca.crt"

    - block:
      - import_role:
          name: linux-system-roles.network
        vars:
          network_connections:
            - name: Configure the Example-wifi profile
              interface_name: wlp1s0
              state: up
              type: wireless
              autoconnect: yes
              ip:
                dhcp4: true
                auto6: true
              wireless:
                ssid: "Example-wifi"
                key_mgmt: "wpa-eap"
              ieee802_1x:
                identity: "user_name"
                eap: tls
                private_key: "/etc/pki/tls/client.key"
                private_key_password: "password"
                private_key_password_flags: none
                client_cert: "/etc/pki/tls/client.pem"
                ca_cert: "/etc/pki/tls/cacert.pem"
                domain_suffix_match: "example.com"
  
```

2. Exécutez le manuel de jeu :

■

■ **# ansible-playbook ~/enable-802.1x.yml**

Ressources supplémentaires

- [/usr/share/ansible/roles/rhel-system-roles.network/README.md](#) fichier

CHAPITRE 19. GESTION DE LA PASSERELLE PAR DÉFAUT

La passerelle par défaut est un routeur qui transmet les paquets du réseau lorsqu'aucune autre route ne correspond à la destination d'un paquet. Dans un réseau local, la passerelle par défaut est généralement l'hôte le plus proche de l'internet.

19.1. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DE NMCLI

Dans la plupart des cas, les administrateurs définissent la passerelle par défaut lorsqu'ils créent une connexion, comme expliqué, par exemple, dans [Configuration d'une connexion Ethernet avec une adresse IP statique à l'aide de nmcli](#).

Dans la plupart des cas, les administrateurs définissent la passerelle par défaut lorsqu'ils créent une connexion. Toutefois, vous pouvez également définir ou mettre à jour le paramètre de passerelle par défaut d'une connexion créée précédemment à l'aide de l'utilitaire **nmcli**.

Conditions préalables

- Au moins une adresse IP statique doit être configurée sur la connexion sur laquelle la passerelle par défaut sera définie.
- Si l'utilisateur est connecté à une console physique, les autorisations de l'utilisateur sont suffisantes. Dans le cas contraire, l'utilisateur doit disposer des autorisations **root**.

Procédure

1. Définissez l'adresse IP de la passerelle par défaut.

Par exemple, pour définir l'adresse IPv4 de la passerelle par défaut sur la connexion **example** à **192.0.2.1**:

```
# nmcli connection modify example ipv4.gateway "192.0.2.1"
```

Par exemple, pour définir l'adresse IPv6 de la passerelle par défaut sur la connexion **example** à **2001:db8:1::1**:

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

2. Redémarrez la connexion réseau pour que les modifications soient prises en compte. Par exemple, pour redémarrer la connexion **example** à l'aide de la ligne de commande :

```
# nmcli connection up example
```



AVERTISSEMENT

Toutes les connexions utilisant actuellement cette connexion réseau sont temporairement interrompues pendant le redémarrage.

- Optionnellement, vérifiez que l'itinéraire est actif.
Pour afficher la passerelle par défaut IPv4 :

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

Pour afficher la passerelle par défaut IPv6 :

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

Ressources supplémentaires

[Configuration d'une connexion Ethernet avec une adresse IP statique à l'aide de nmcli](#) .

19.2. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DU MODE INTERACTIF NMCLI

In most situations, administrators set the default gateway when they create a connection as explained in, for example, * [Configuring an Ethernet connection with a static IP address by using the nmcli interactive editor](#)

Dans la plupart des cas, les administrateurs définissent la passerelle par défaut lorsqu'ils créent une connexion. Toutefois, vous pouvez également définir ou mettre à jour le paramètre de passerelle par défaut d'une connexion créée précédemment en utilisant le mode interactif de l'utilitaire **nmcli**.

Conditions préalables

- Au moins une adresse IP statique doit être configurée sur la connexion sur laquelle la passerelle par défaut sera définie.
- Si l'utilisateur est connecté à une console physique, les autorisations de l'utilisateur suffisent. Dans le cas contraire, l'utilisateur doit disposer des autorisations **root**.

Procédure

- Ouvrez le mode interactif **nmcli** pour la connexion requise. Par exemple, pour ouvrir le mode interactif **nmcli** pour la connexion *example*:

```
# nmcli connection edit example
```

- Définir la passerelle par défaut.

Par exemple, pour définir l'adresse IPv4 de la passerelle par défaut sur la connexion **example** à **192.0.2.1**:

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

Par exemple, pour définir l'adresse IPv6 de la passerelle par défaut sur la connexion **example** à **2001:db8:1::1**:

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

3. En option, vérifiez que la passerelle par défaut a été définie correctement :

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

4. Sauvegarder la configuration :

```
nmcli> save persistent
```

5. Redémarrez la connexion réseau pour que les modifications soient prises en compte :

```
nmcli> activate example
```



AVERTISSEMENT

Toutes les connexions utilisant actuellement cette connexion réseau sont temporairement interrompues pendant le redémarrage.

6. Quittez le mode interactif de **nmcli**:

```
nmcli> quit
```

7. Optionnellement, vérifiez que l'itinéraire est actif.

Pour afficher la passerelle par défaut IPv4 :

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

Pour afficher la passerelle par défaut IPv6 :

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

Ressources supplémentaires

- [Configuration d'une connexion Ethernet avec une adresse IP statique à l'aide de l'éditeur interactif nmcli](#)

19.3. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DE NM-CONNECTION-EDITOR

Dans la plupart des cas, les administrateurs définissent la passerelle par défaut lorsqu'ils créent une connexion. Toutefois, vous pouvez également définir ou mettre à jour le paramètre de passerelle par défaut d'une connexion créée précédemment à l'aide de l'application **nm-connection-editor**.

Conditions préalables

- Au moins une adresse IP statique doit être configurée sur la connexion sur laquelle la passerelle par défaut sera définie.

Procédure

1. Ouvrez un terminal et entrez **nm-connection-editor**:

```
# nm-connection-editor
```

2. Sélectionnez la connexion à modifier et cliquez sur l'icône de la roue dentée pour modifier la connexion existante.
3. Définir la passerelle par défaut IPv4. Par exemple, pour définir l'adresse IPv4 de la passerelle par défaut sur la connexion à **192.0.2.1**:
 - a. Ouvrez l'onglet **IPv4 Settings**.
 - b. Saisissez l'adresse dans le champ **gateway** à côté de la plage IP dans laquelle se trouve l'adresse de la passerelle :

Addresses		
Address	Netmask	Gateway
192.0.2.123	24	192.0.2.1

4. Définir la passerelle par défaut IPv6. Par exemple, pour définir l'adresse IPv6 de la passerelle par défaut sur la connexion à **2001:db8:1::1**:
 - a. Ouvrez l'onglet **IPv6**.
 - b. Saisissez l'adresse dans le champ **gateway** à côté de la plage IP dans laquelle se trouve l'adresse de la passerelle :

Addresses		
Address	Prefix	Gateway
2001:db8:1::5	64	2001:db8:1::1

5. Cliquez sur **OK**.
6. Cliquez sur **Enregistrer**.
7. Redémarrez la connexion réseau pour que les modifications soient prises en compte. Par exemple, pour redémarrer la connexion **example** à l'aide de la ligne de commande :

```
# nmcli connection up example
```



AVERTISSEMENT

Toutes les connexions utilisant actuellement cette connexion réseau sont temporairement interrompues pendant le redémarrage.

8. Optionnellement, vérifiez que l'itinéraire est actif.
Pour afficher la passerelle par défaut IPv4 :

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

Pour afficher la passerelle par défaut IPv6 :

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

Ressources supplémentaires

- [Configuration d'une connexion Ethernet à l'aide de nm-connection-editor](#)

19.4. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DU CENTRE DE CONTRÔLE

Dans la plupart des cas, les administrateurs définissent la passerelle par défaut lorsqu'ils créent une connexion. Toutefois, vous pouvez également définir ou mettre à jour le paramètre de passerelle par défaut d'une connexion créée précédemment à l'aide de l'application **control-center**.

Conditions préalables

- Au moins une adresse IP statique doit être configurée sur la connexion sur laquelle la passerelle par défaut sera définie.
- La configuration réseau de la connexion est ouverte dans l'application **control-center**.

Procédure

1. Définir la passerelle par défaut IPv4. Par exemple, pour définir l'adresse IPv4 de la passerelle par défaut sur la connexion à **192.0.2.1**:
 - a. Ouvrez l'onglet **IPv4**.
 - b. Saisissez l'adresse dans le champ **gateway** à côté de la plage IP dans laquelle se trouve l'adresse de la passerelle :

Addresses		
Address	Netmask	Gateway
192.0.2.123	255.255.255.0	192.0.2.1

2. Définir la passerelle par défaut IPv6. Par exemple, pour définir l'adresse IPv6 de la passerelle par défaut sur la connexion à **2001:db8:1::1**:
 - a. Ouvrez l'onglet **IPv6**.
 - b. Saisissez l'adresse dans le champ **gateway** à côté de la plage IP dans laquelle se trouve l'adresse de la passerelle :

Addresses		
Address	Prefix	Gateway
2001:db8:1::5	64	2001:db8:1::1

3. Cliquez sur **Appliquer**.
4. De retour dans la fenêtre **Network**, désactivez et réactivez la connexion en mettant le bouton de la connexion sur **Off** et en le remettant sur **On** pour que les changements prennent effet.



AVERTISSEMENT

Toutes les connexions utilisant actuellement cette connexion réseau sont temporairement interrompues pendant le redémarrage.

5. Optionnellement, vérifiez que l'itinéraire est actif.
Pour afficher la passerelle par défaut IPv4 :

```
$ ip -4 route
```

```
default via 192.0.2.1 dev example proto static metric 100
```

Pour afficher la passerelle par défaut IPv6 :

```
$ ip -6 route
```

```
default via 2001:db8:1::1 dev example proto static metric 100 pref medium
```

Ressources supplémentaires

- [Configuration d'une connexion Ethernet à l'aide du centre de contrôle](#)

19.5. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DE NMSTATECTL

Dans la plupart des cas, les administrateurs définissent la passerelle par défaut lorsqu'ils créent une connexion. Toutefois, vous pouvez également définir ou mettre à jour le paramètre de passerelle par défaut d'une connexion réseau à l'aide de l'utilitaire **nmstatectl**.

Conditions préalables

- Au moins une adresse IP statique doit être configurée sur la connexion sur laquelle la passerelle par défaut sera définie.
- L'interface **enp1s0** est configurée et l'adresse IP de la passerelle par défaut se trouve dans le sous-réseau de la configuration IP de cette interface.
- Le paquet **nmstate** est installé.

Procédure

1. Créez un fichier YAML, par exemple `~/set-default-gateway.yml`, avec le contenu suivant :

```
---
routes:
  config:
    - destination: 0.0.0.0/0
      next-hop-address: 192.0.2.1
      next-hop-interface: enp1s0
```

2. Appliquer les paramètres au système :

```
# nmstatectl apply ~/set-default-gateway.yml
```

Ressources supplémentaires

- **nmstatectl(8)** page de manuel
- `/usr/share/doc/nmstate/examples/` répertoire

19.6. DÉFINITION DE LA PASSERELLE PAR DÉFAUT SUR UNE CONNEXION EXISTANTE À L'AIDE DU RÔLE DE RÉSEAU RHEL SYSTEM ROLE

Vous pouvez utiliser le rôle de système **network** RHEL pour définir la passerelle par défaut.



IMPORTANT

Lorsque vous exécutez une séquence qui utilise le rôle système **network** RHEL, le rôle système remplace un profil de connexion existant portant le même nom si la valeur des paramètres ne correspond pas à ceux spécifiés dans la séquence. Par conséquent, indiquez toujours la configuration complète du profil de connexion réseau dans la pièce, même si, par exemple, la configuration IP existe déjà. Dans le cas contraire, le rôle rétablit les valeurs par défaut.

Selon qu'il existe déjà ou non, la procédure crée ou met à jour le profil de connexion **enp1s0** avec les paramètres suivants :

- Une adresse IPv4 statique - **198.51.100.20** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **198.51.100.254**

- Une passerelle par défaut IPv6 - **2001:db8:1::fffe**
- Un serveur DNS IPv4 - **198.51.100.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.

Procédure

1. Créez un fichier playbook, par exemple `~/ethernet-connection.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
  - name: Configure an Ethernet connection with static IP and default gateway
    include_role:
      name: rhel-system-roles.network

  vars:
    network_connections:
    - name: enp1s0
      type: ethernet
      autoconnect: yes
      ip:
        address:
        - 198.51.100.20/24
        - 2001:db8:1::1/64
        gateway4: 198.51.100.254
        gateway6: 2001:db8:1::fffe
        dns:
        - 198.51.100.200
        - 2001:db8:1::ffbb
        dns_search:
        - example.com
      state: up
```

2. Exécutez le manuel de jeu :

```
# ansible-playbook ~/ethernet-connection.yml
```

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

19.7. COMMENT NETWORKMANAGER GÈRE PLUSIEURS PASSERELLES PAR DÉFAUT

Dans certaines situations, par exemple pour des raisons de repli, vous définissez plusieurs passerelles par défaut sur un hôte. Toutefois, pour éviter les problèmes de routage asynchrone, chaque passerelle par défaut du même protocole nécessite une valeur métrique distincte. Notez que RHEL utilise uniquement la connexion à la passerelle par défaut dont la valeur métrique est la plus faible.

Vous pouvez définir la métrique pour la passerelle IPv4 et IPv6 d'une connexion à l'aide de la commande suivante :

```
# nmcli connection modify connection-name ipv4.route-metric value ipv6.route-metric value
```



IMPORTANT

Ne définissez pas la même valeur métrique pour le même protocole dans plusieurs profils de connexion afin d'éviter les problèmes de routage.

Si vous définissez une passerelle par défaut sans valeur métrique, NetworkManager définit automatiquement la valeur métrique en fonction du type d'interface. Pour cela, NetworkManager assigne la valeur par défaut de ce type de réseau à la première connexion activée, et définit une valeur incrémentée à chaque autre connexion du même type dans l'ordre où elles sont activées. Par exemple, s'il existe deux connexions Ethernet avec une passerelle par défaut, NetworkManager définit une métrique de **100** sur la route vers la passerelle par défaut de la connexion que vous activez en premier. Pour la deuxième connexion, NetworkManager définit **101**.

Voici un aperçu des types de réseaux fréquemment utilisés et de leurs paramètres par défaut :

Type de connexion	Valeur métrique par défaut
VPN	50
Ethernet	100
MACsec	125
InfiniBand	150
Obligation	300
L'équipe	350
VLAN	400

Type de connexion	Valeur métrique par défaut
Pont	425
TUN	450
Wi-Fi	600
Tunnel IP	675

Ressources supplémentaires

- [Configuration du routage basé sur des règles pour définir des itinéraires alternatifs](#)
- [Premiers pas avec TCP Multipath](#)

19.8. CONFIGURER NETWORKMANAGER POUR ÉVITER D'UTILISER UN PROFIL SPÉCIFIQUE POUR FOURNIR UNE PASSERELLE PAR DÉFAUT

Vous pouvez configurer NetworkManager pour qu'il n'utilise jamais un profil spécifique pour fournir la passerelle par défaut. Suivez cette procédure pour les profils de connexion qui ne sont pas connectés à la passerelle par défaut.

Conditions préalables

- Le profil de connexion NetworkManager pour la connexion qui n'est pas connectée à la passerelle par défaut existe.

Procédure

1. Si la connexion utilise une configuration IP dynamique, configurer que NetworkManager n'utilise pas la connexion comme route par défaut pour les connexions IPv4 et IPv6 :

```
# nmcli connection modify connection_name ipv4.never-default yes ipv6.never-default yes
```

Notez que le fait de définir **ipv4.never-default** et **ipv6.never-default** sur **yes**, supprime automatiquement du profil de connexion l'adresse IP de la passerelle par défaut pour le protocole correspondant.

2. Activer la connexion :

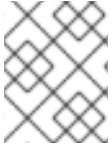
```
# nmcli connection up connection_name
```

Vérification

- Utilisez les commandes **ip -4 route** et **ip -6 route** pour vérifier que RHEL n'utilise pas l'interface réseau pour la route par défaut des protocoles IPv4 et IPv6.

19.9. CORRECTION D'UN COMPORTEMENT DE ROUTAGE INATTENDU DÙ À LA PRÉSENCE DE PLUSIEURS PASSERELLES PAR DÉFAUT

Il n'y a que quelques scénarios, tels que l'utilisation de TCP multipath, dans lesquels vous avez besoin de plusieurs passerelles par défaut sur un hôte. Dans la plupart des cas, vous ne configurez qu'une seule passerelle par défaut afin d'éviter un comportement de routage inattendu ou des problèmes de routage asynchrone.



NOTE

Pour acheminer le trafic vers différents fournisseurs d'accès à Internet, utilisez le routage basé sur des règles plutôt que plusieurs passerelles par défaut.

Conditions préalables

- L'hôte utilise NetworkManager pour gérer les connexions réseau, ce qui est la valeur par défaut.
- L'hôte possède plusieurs interfaces réseau.
- Plusieurs passerelles par défaut sont configurées sur l'hôte.

Procédure

1. Afficher la table de routage :

- Pour IPv4, entrez :

```
# ip -4 route
default via 192.0.2.1 dev enp1s0 proto static metric 101
default via 198.51.100.1 dev enp7s0 proto static metric 102
...
```

- Pour IPv6, entrez :

```
# ip -6 route
default via 2001:db8:1::1 dev enp1s0 proto static metric 101 pref medium
default via 2001:db8:2::1 dev enp7s0 proto static metric 102 pref medium
...
```

Les entrées commençant par **default** indiquent une route par défaut. Notez les noms d'interface de ces entrées affichés à côté de **dev**.

2. Utilisez les commandes suivantes pour afficher les connexions NetworkManager qui utilisent les interfaces identifiées à l'étape précédente :

```
# nmcli -f GENERAL.CONNECTION,IP4.GATEWAY,IP6.GATEWAY device show enp1s0
GENERAL.CONNECTION: Corporate-LAN
IP4.GATEWAY: 192.168.122.1
IP6.GATEWAY: 2001:db8:1::1
```

```
# nmcli -f GENERAL.CONNECTION,IP4.GATEWAY,IP6.GATEWAY device show enp7s0
GENERAL.CONNECTION: Internet-Provider
IP4.GATEWAY: 198.51.100.1
IP6.GATEWAY: 2001:db8:2::1
```

Dans ces exemples, les profils nommés **Corporate-LAN** et **Internet-Provider** ont les passerelles par défaut définies. Étant donné que, dans un réseau local, la passerelle par défaut est généralement l'hôte le plus proche d'Internet, le reste de cette procédure suppose que les passerelles par défaut de l'adresse **Corporate-LAN** sont incorrectes.

3. Configurez NetworkManager pour qu'il n'utilise pas la connexion **Corporate-LAN** comme route par défaut pour les connexions IPv4 et IPv6 :

```
# nmcli connection modify Corporate-LAN ipv4.never-default yes ipv6.never-default yes
```

Notez que le fait de définir **ipv4.never-default** et **ipv6.never-default** sur **yes**, supprime automatiquement du profil de connexion l'adresse IP de la passerelle par défaut pour le protocole correspondant.

4. Activez la connexion **Corporate-LAN**:

```
# nmcli connection up Corporate-LAN
```

Vérification

- Affichez les tables de routage IPv4 et IPv6 et vérifiez qu'une seule passerelle par défaut est disponible pour chaque protocole :
 - Pour IPv4, entrez :

```
# ip -4 route
default via 192.0.2.1 dev enp1s0 proto static metric 101
...
```

- Pour IPv6, entrez :

```
# ip -6 route
default via 2001:db8:1::1 dev enp1s0 proto static metric 101 pref medium
...
```

Ressources supplémentaires

- [Configuration du routage basé sur des règles pour définir des itinéraires alternatifs](#)
- [Premiers pas avec TCP Multipath](#)

CHAPITRE 20. CONFIGURATION DES ROUTES STATIQUES

Le routage permet d'envoyer et de recevoir du trafic entre des réseaux connectés entre eux. Dans les grands environnements, les administrateurs configurent généralement les services de manière à ce que les routeurs puissent connaître dynamiquement les autres routeurs. Dans les environnements plus petits, les administrateurs configurent souvent des routes statiques pour s'assurer que le trafic peut être acheminé d'un réseau à l'autre.

Vous avez besoin d'itinéraires statiques pour assurer une communication efficace entre plusieurs réseaux si toutes les conditions suivantes sont réunies :

- Le trafic doit passer par plusieurs réseaux.
- Le flux de trafic exclusif passant par les passerelles par défaut n'est pas suffisant.

Section 20.1, « Exemple de réseau nécessitant des routes statiques » décrit des scénarios et la manière dont le trafic circule entre différents réseaux lorsque vous ne configurez pas d'itinéraires statiques.

20.1. EXEMPLE DE RÉSEAU NÉCESSITANT DES ROUTES STATIQUES

Dans cet exemple, vous avez besoin d'itinéraires statiques car tous les réseaux IP ne sont pas directement connectés par un seul routeur. Sans routes statiques, certains réseaux ne peuvent pas communiquer entre eux. En outre, le trafic de certains réseaux ne circule que dans une seule direction.



NOTE

La topologie du réseau dans cet exemple est artificielle et n'est utilisée que pour expliquer le concept de routage statique. Elle n'est pas recommandée dans les environnements de production.

Pour que la communication fonctionne entre tous les réseaux de cet exemple, configurez une route statique vers Raleigh (**198.51.100.0/24**) avec le prochain saut Router 2 (**203.0.113.10**). L'adresse IP du saut suivant est celle du routeur 2 du réseau du centre de données (**203.0.113.0/24**).

Vous pouvez configurer la route statique comme suit :

- Pour une configuration simplifiée, définissez cette route statique uniquement sur le routeur 1. Cependant, cela augmente le trafic sur le routeur 1 car les hôtes du centre de données (**203.0.113.0/24**) envoient le trafic vers Raleigh (**198.51.100.0/24**) toujours via le routeur 1 vers le routeur 2.
- Pour une configuration plus complexe, configurez cette route statique sur tous les hôtes du centre de données (**203.0.113.0/24**). Tous les hôtes de ce sous-réseau envoient alors le trafic directement au routeur 2 (**203.0.113.10**) qui est plus proche de Raleigh (**198.51.100.0/24**).

Pour plus de détails sur les réseaux entre lesquels le trafic circule ou non, voir les explications sous le diagramme.

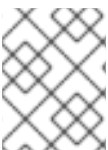


270_RHEL_0822

In case that the required static routes are not configured, les situations dans lesquelles la communication fonctionne et celles dans lesquelles elle ne fonctionne pas sont indiquées ci-dessous :

- Hôtes du réseau de Berlin (**192.0.2.0/24**) :
 - Ils peuvent communiquer avec d'autres hôtes du même sous-réseau parce qu'ils sont directement connectés.
 - Il peut communiquer avec l'internet parce que le routeur 1 fait partie du réseau berlinois (**192.0.2.0/24**) et possède une passerelle par défaut qui mène à l'internet.
 - Il peut communiquer avec le réseau du centre de données (**203.0.113.0/24**) parce que le routeur 1 a des interfaces à la fois dans le réseau de Berlin (**192.0.2.0/24**) et dans le réseau du centre de données (**203.0.113.0/24**).
 - Impossible de communiquer avec le réseau Raleigh (**198.51.100.0/24**) car le routeur 1 n'a pas d'interface dans ce réseau. Par conséquent, le routeur 1 envoie le trafic à sa propre passerelle par défaut (Internet).
- Hôtes dans le réseau du centre de données (**203.0.113.0/24**) :
 - Ils peuvent communiquer avec d'autres hôtes du même sous-réseau parce qu'ils sont directement connectés.
 - Ils peuvent communiquer avec l'internet parce que leur passerelle par défaut est réglée sur le routeur 1 et que le routeur 1 a des interfaces dans les deux réseaux, le centre de données (**203.0.113.0/24**) et l'internet.

- Ils peuvent communiquer avec le réseau de Berlin (**192.0.2.0/24**) parce que leur passerelle par défaut est réglée sur le routeur 1 et que ce dernier possède des interfaces à la fois dans le centre de données (**203.0.113.0/24**) et dans le réseau de Berlin (**192.0.2.0/24**).
- Impossible de communiquer avec le réseau Raleigh (**198.51.100.0/24**) car le réseau du centre de données n'a pas d'interface dans ce réseau. Par conséquent, les hôtes du centre de données (**203.0.113.0/24**) envoient du trafic à leur passerelle par défaut (routeur 1). Le routeur 1 n'a pas non plus d'interface dans le réseau de Raleigh (**198.51.100.0/24**) et, par conséquent, le routeur 1 envoie ce trafic à sa propre passerelle par défaut (Internet).
- Hôtes du réseau Raleigh (**198.51.100.0/24**) :
 - Ils peuvent communiquer avec d'autres hôtes du même sous-réseau parce qu'ils sont directement connectés.
 - Impossible de communiquer avec des hôtes sur Internet. Le routeur 2 envoie le trafic au routeur 1 en raison des paramètres de la passerelle par défaut. Le comportement réel du routeur 1 dépend de la configuration du filtre de chemin inverse (**rp_filter**) et du contrôle du système (**sysctl**). Par défaut sur RHEL, le routeur 1 laisse tomber le trafic sortant au lieu de l'acheminer vers l'internet. Toutefois, quel que soit le comportement configuré, la communication n'est pas possible sans la route statique.
 - Impossible de communiquer avec le réseau du centre de données (**203.0.113.0/24**). Le trafic sortant atteint la destination via le routeur 2 en raison du paramètre de passerelle par défaut. Toutefois, les réponses aux paquets ne parviennent pas à l'expéditeur car les hôtes du réseau du centre de données (**203.0.113.0/24**) envoient les réponses à leur passerelle par défaut (routeur 1). Le routeur 1 envoie alors le trafic vers l'internet.
 - Impossible de communiquer avec le réseau de Berlin (**192.0.2.0/24**). Le routeur 2 envoie le trafic au routeur 1 en raison des paramètres de la passerelle par défaut. Le comportement réel du routeur 1 dépend du paramètre **rp_filter sysctl**. Par défaut sur RHEL, le routeur 1 laisse tomber le trafic sortant au lieu de l'envoyer au réseau de Berlin (**192.0.2.0/24**). Toutefois, quel que soit le comportement configuré, la communication n'est pas possible sans la route statique.



NOTE

Outre la configuration des routes statiques, vous devez activer le transfert IP sur les deux routeurs.

Ressources supplémentaires

- [Pourquoi ne peut-on pas envoyer de ping à un serveur si net.ipv4.conf.all.rp_filter est défini sur le serveur ?](#)
- [Activation de la redirection IP](#)

20.2. COMMENT UTILISER LA COMMANDE NMCLI POUR CONFIGURER UNE ROUTE STATIQUE ?

Pour configurer une route statique, utilisez l'utilitaire **nmcli** avec la syntaxe suivante :

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

La commande prend en charge les attributs de route suivants :

- **cwnd=*n***: Définit la taille de la fenêtre de congestion (CWND), définie en nombre de paquets.
- **lock-cwnd=true|false**: Définit si le noyau peut ou non mettre à jour la valeur CWND.
- **lock-mtu=true|false**: Définit si le noyau peut ou non mettre à jour le MTU pour la découverte du MTU du chemin.
- **lock-window=true|false**: Définit si le noyau peut ou non mettre à jour la taille maximale de la fenêtre pour les paquets TCP.
- **mtu=*n***: Définit l'unité de transfert maximale (MTU) à utiliser le long du chemin vers la destination.
- **onlink=true|false**: Définit si le prochain saut est directement attaché à ce lien même s'il ne correspond à aucun préfixe d'interface.
- **scope=*n***: Pour un itinéraire IPv4, cet attribut définit l'étendue des destinations couvertes par le préfixe de l'itinéraire. La valeur est un nombre entier (0-255).
- **src=*address***: Définit l'adresse source à privilégier lors de l'envoi de trafic vers les destinations couvertes par le préfixe de l'itinéraire.
- **table=*table_id***: Définit l'ID de la table à laquelle la route doit être ajoutée. Si ce paramètre est omis, NetworkManager utilise la table **main**.
- **tos=*n***: Définit la clé de type de service (TOS). Définissez la valeur sous forme d'un nombre entier (0-255).
- **type=*value***: Définit le type d'itinéraire. NetworkManager supporte les types d'itinéraires **unicast**, **local**, **blackhole**, **unreachable**, **prohibit**, et **throw**. La valeur par défaut est **unicast**.
- **window=*n***: Définit la taille maximale de la fenêtre que TCP doit annoncer à ces destinations, mesurée en octets.

Si vous utilisez la sous-commande **ipv4.routes, nmcli** remplace tous les réglages actuels de ce paramètre.

Pour ajouter un itinéraire :

```
$ nmcli connection modify connection_name ipv4.routes "<route>"
```

De même, pour supprimer un itinéraire spécifique :

```
$ nmcli connection modify connection_name -ipv4.routes "<route>"
```

20.3. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DE NMCLI

Vous pouvez ajouter une route statique à un profil de connexion NetworkManager existant à l'aide de la commande **nmcli connection modify**.

La procédure ci-dessous permet de configurer les routes suivantes :

- Une route IPv4 vers le réseau distant **198.51.100.0/24**. La passerelle correspondante avec l'adresse IP **192.0.2.10** est accessible via la connexion **example**.

- Une route IPv6 vers le réseau distant **2001:db8:2::/64**. La passerelle correspondante avec l'adresse IP **2001:db8:1::10** est accessible via la connexion **example**.

Conditions préalables

- Le profil de connexion **example** existe et il configure cet hôte pour qu'il soit dans le même sous-réseau IP que les passerelles.

Procédure

1. Ajoutez la route IPv4 statique au profil de connexion **example**:

```
# nmcli connection modify example ipv4.routes "198.51.100.0/24 192.0.2.10"
```

Pour définir plusieurs itinéraires en une seule étape, passez les itinéraires individuels séparés par des virgules à la commande. Par exemple, pour ajouter un itinéraire vers les réseaux **198.51.100.0/24** et **203.0.113.0/24**, tous deux acheminés par la passerelle **192.0.2.10**, entrez :

```
# nmcli connection modify example ipv4.routes "198.51.100.0/24 192.0.2.10, 203.0.113.0/24 192.0.2.10"
```

2. Ajoutez la route IPv6 statique au profil de connexion **example**:

```
# nmcli connection modify example ipv6.routes "2001:db8:2::/64 2001:db8:1::10"
```

3. Réactiver la connexion :

```
# nmcli connection up example
```

Vérification

1. Afficher les itinéraires IPv4 :

```
# ip -4 route
...
198.51.100.0/24 via 192.0.2.10 dev enp1s0
```

2. Afficher les itinéraires IPv6 :

```
# ip -6 route
...
2001:db8:2::/64 via 2001:db8:1::10 dev enp1s0 metric 1024 pref medium
```

Ressources supplémentaires

- **nmcli(1)** page de manuel
- **nm-settings-nmcli(5)** page de manuel

20.4. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DE NMTUI

L'application **nmtui** fournit une interface utilisateur textuelle pour NetworkManager. Vous pouvez utiliser **nmtui** pour configurer des routes statiques sur un hôte sans interface graphique.

Par exemple, la procédure ci-dessous ajoute une route vers le réseau **192.0.2.0/24** qui utilise la passerelle fonctionnant sur **198.51.100.1**, qui est accessible par le biais d'un profil de connexion existant.



NOTE

Sur **nmtui**:

- Naviguer à l'aide des touches du curseur.
- Appuyez sur un bouton en le sélectionnant et en appuyant sur **Entrée**.
- Sélectionnez et désélectionnez les cases à cocher en utilisant l'**espace**.

Conditions préalables

- Le réseau est configuré.
- La passerelle de la route statique doit être directement accessible sur l'interface.
- Si l'utilisateur est connecté à une console physique, les droits d'utilisateur sont suffisants. Dans le cas contraire, la commande nécessite les autorisations de l'administrateur.

Procédure

1. Démarrer **nmtui**:

```
# nmtui
```

2. Sélectionnez **Edit a connection** et appuyez sur **Enter**.
3. Sélectionnez le profil de connexion par lequel vous pouvez atteindre le saut suivant vers le réseau de destination, et appuyez sur **Entrée**.
4. Selon qu'il s'agit d'une route IPv4 ou IPv6, appuyez sur le bouton **Show** à côté de la zone de configuration du protocole.
5. Appuyez sur le bouton **Edit** à côté de **Routing**, ce qui ouvre une nouvelle fenêtre dans laquelle vous pouvez configurer les routes statiques :
 - a. Appuyez sur le bouton **Add** et remplissez le formulaire :
 - Le réseau de destination, y compris le préfixe au format CIDR (Classless Inter-Domain Routing)
 - L'adresse IP du prochain saut
 - Une valeur métrique, si vous ajoutez plusieurs itinéraires vers le même réseau et que vous souhaitez donner la priorité aux itinéraires en fonction de leur efficacité
 - b. Répétez l'étape précédente pour chaque route que vous souhaitez ajouter et qui est accessible via ce profil de connexion.
 - c. Appuyez sur le bouton **OK** pour revenir à la fenêtre des paramètres de connexion.

Figure 20.1. Exemple de route statique sans métrique

Destination/Prefix	Next Hop	Metric	
192.0.2.0/24	198.51.100.1		<Remove>
<Add...>			
			<Cancel> <OK>

- Appuyez sur le bouton **OK** pour revenir au menu principal **nmtui**.
- Sélectionnez **Activate a connection** et appuyez sur **Entrée**.
- Sélectionnez le profil de connexion que vous avez modifié et appuyez deux fois sur **Entrée** pour le désactiver et le réactiver.



IMPORTANT

Sautez cette étape si vous exécutez **nmtui** via une connexion distante, telle que SSH, qui utilise le profil de connexion que vous souhaitez réactiver. Dans ce cas, si vous le désactivez dans **nmtui**, la connexion est interrompue et, par conséquent, vous ne pouvez pas l'activer à nouveau. Pour éviter ce problème, utilisez la commande **nmcli connection connection_profile_name up** pour réactiver la connexion dans le scénario mentionné.

- Appuyez sur le bouton **Back** pour revenir au menu principal.
- Sélectionnez **Quit** et appuyez sur **Entrée** pour fermer l'application **nmtui**.

Vérification

- Vérifiez que la route est active :

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

20.5. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DU CENTRE DE CONTRÔLE

Vous pouvez utiliser **control-center** dans GNOME pour ajouter une route statique à la configuration d'une connexion réseau.

La procédure ci-dessous permet de configurer les routes suivantes :

- Une route IPv4 vers le réseau distant **198.51.100.0/24**. La passerelle correspondante a l'adresse IP **192.0.2.10**.
- Une route IPv6 vers le réseau distant **2001:db8:2::/64**. La passerelle correspondante a l'adresse IP **2001:db8:1::10**.

Conditions préalables

- Le réseau est configuré.
- Cet hôte se trouve dans le même sous-réseau IP que les passerelles.
- La configuration réseau de la connexion est ouverte dans l'application **control-center**. Voir [Configuration d'une connexion Ethernet à l'aide de nm-connection-editor](#).

Procédure

1. Dans l'onglet **IPv4**:
 - a. Facultatif : Désactivez les itinéraires automatiques en cliquant sur le bouton **On** dans la section **Routes** de l'onglet **IPv4** pour n'utiliser que des itinéraires statiques. Si les itinéraires automatiques sont activés, Red Hat Enterprise Linux utilise les itinéraires statiques et les itinéraires reçus d'un serveur DHCP.
 - b. Saisissez l'adresse, le masque de réseau, la passerelle et, éventuellement, la valeur métrique de la route IPv4 :

Routes				Automatic <input checked="" type="checkbox"/>
Address	Netmask	Gateway	Metric	
198.51.100.0	24	192.0.2.10		<input checked="" type="checkbox"/>

2. Dans l'onglet **IPv6**:
 - a. Facultatif : Désactivez les routes automatiques en cliquant sur le bouton **On** dans la section **Routes** de l'onglet **IPv4** pour n'utiliser que des routes statiques.
 - b. Saisissez l'adresse, le masque de réseau, la passerelle et, éventuellement, la valeur métrique de la route IPv6 :

Routes				Automatic <input checked="" type="checkbox"/>
Address	Prefix	Gateway	Metric	
2001:db8:2::	64	2001:db8:1::10		<input checked="" type="checkbox"/>

3. Cliquez sur **Appliquer**.
4. De retour dans la fenêtre **Network**, désactivez et réactivez la connexion en mettant le bouton de la connexion sur **Off** et en le remettant sur **On** pour que les changements prennent effet.



AVERTISSEMENT

Le redémarrage de la connexion interrompt brièvement la connectivité sur cette interface.

Vérification

1. Afficher les itinéraires IPv4 :

```
# ip -4 route
...
198.51.100.0/24 via 192.0.2.10 dev enp1s0
```

2. Afficher les itinéraires IPv6 :

```
# ip -6 route
...
2001:db8:2::/64 via 2001:db8:1::10 dev enp1s0 metric 1024 pref medium
```

20.6. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DE NM-CONNECTION-EDITOR

Vous pouvez utiliser l'application **nm-connection-editor** pour ajouter une route statique à la configuration d'une connexion réseau.

La procédure ci-dessous permet de configurer les routes suivantes :

- Une route IPv4 vers le réseau distant **198.51.100.0/24**. La passerelle correspondante avec l'adresse IP **192.0.2.10** est accessible via la connexion **example**.
- Une route IPv6 vers le réseau distant **2001:db8:2::/64**. La passerelle correspondante avec l'adresse IP **2001:db8:1::10** est accessible via la connexion **example**.

Conditions préalables

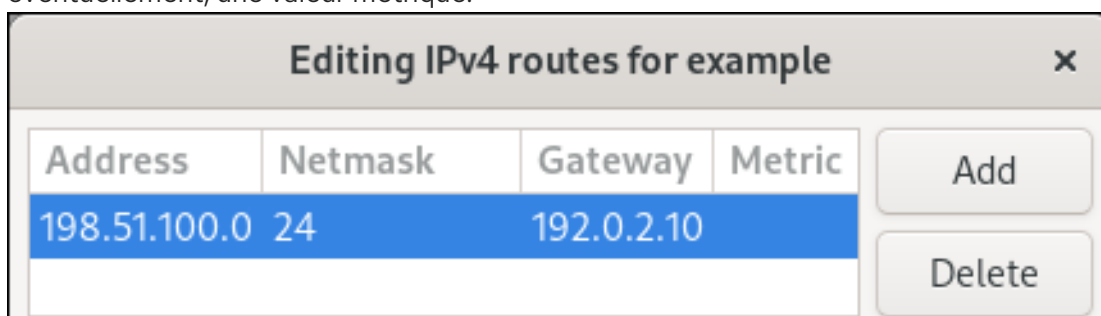
- Le réseau est configuré.
- Cet hôte se trouve dans le même sous-réseau IP que les passerelles.

Procédure

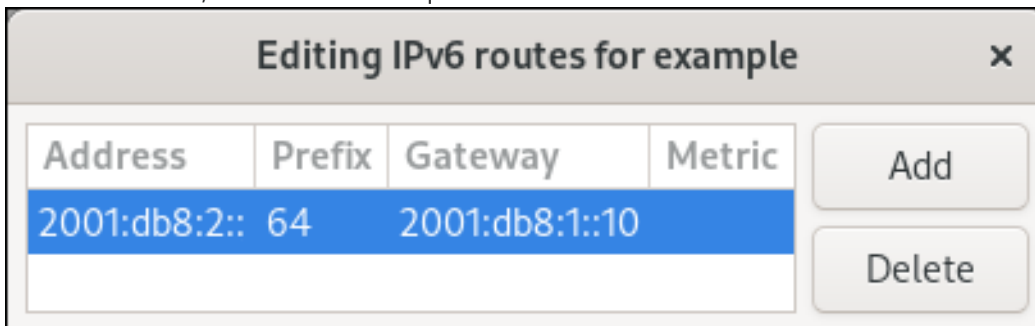
1. Ouvrez un terminal et entrez **nm-connection-editor**:

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

2. Sélectionnez le profil de connexion **example** et cliquez sur l'icône de la roue dentée pour modifier la connexion existante.
3. Dans l'onglet **IPv4 Settings**:
- Cliquez sur le bouton **Routes**.
 - Cliquez sur le bouton **Ajouter** et entrez l'adresse, le masque de réseau, la passerelle et, éventuellement, une valeur métrique.



- c. Cliquez sur **OK**.
4. Dans l'onglet **IPv6 Settings**:
 - a. Cliquez sur le bouton **Routes**.
 - b. Cliquez sur le bouton **Ajouter** et entrez l'adresse, le masque de réseau, la passerelle et, éventuellement, une valeur métrique.



- c. Cliquez sur **OK**.
5. Cliquez sur **Enregistrer**.
6. Redémarrez la connexion réseau pour que les modifications soient prises en compte. Par exemple, pour redémarrer la connexion **example** à l'aide de la ligne de commande :

```
# nmcli connection up example
```

Vérification

1. Afficher les itinéraires IPv4 :

```
# ip -4 route
...
198.51.100.0/24 via 192.0.2.10 dev enp1s0
```

2. Afficher les itinéraires IPv6 :

```
# ip -6 route
...
2001:db8:2::/64 via 2001:db8:1::10 dev enp1s0 metric 1024 pref medium
```

20.7. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DU MODE INTERACTIF NMCLI

Vous pouvez utiliser le mode interactif de l'utilitaire **nmcli** pour ajouter une route statique à la configuration d'une connexion réseau.

La procédure ci-dessous permet de configurer les routes suivantes :

- Une route IPv4 vers le réseau distant **198.51.100.0/24**. La passerelle correspondante avec l'adresse IP **192.0.2.10** est accessible via la connexion **example**.
- Une route IPv6 vers le réseau distant **2001:db8:2::/64**. La passerelle correspondante avec l'adresse IP **2001:db8:1::10** est accessible via la connexion **example**.

Conditions préalables

- Le profil de connexion **example** existe et il configure cet hôte pour qu'il soit dans le même sous-réseau IP que les passerelles.

Procédure

1. Ouvrez le mode interactif **nmcli** pour la connexion **example**:

```
# nmcli connection edit example
```

2. Ajouter la route IPv4 statique :

```
nmcli> set ipv4.routes 198.51.100.0/24 192.0.2.10
```

3. Ajouter la route IPv6 statique :

```
nmcli> set ipv6.routes 2001:db8:2::/64 2001:db8:1::10
```

4. En option, vérifiez que les routes ont été ajoutées correctement à la configuration :

```
nmcli> print
...
ipv4.routes: { ip = 198.51.100.0/24, nh = 192.0.2.10 }
...
ipv6.routes: { ip = 2001:db8:2::/64, nh = 2001:db8:1::10 }
...
```

L'attribut **ip** indique le réseau à acheminer et l'attribut **nh** la passerelle (next hop).

5. Sauvegarder la configuration :

```
nmcli> save persistent
```

6. Redémarrer la connexion réseau :

```
nmcli> activate example
```

7. Quittez le mode interactif de **nmcli**:

```
nmcli> quit
```

Vérification

1. Afficher les itinéraires IPv4 :

```
# ip -4 route
...
198.51.100.0/24 via 192.0.2.10 dev enp1s0
```

2. Afficher les itinéraires IPv6 :

```
# ip -6 route
```

```
...
2001:db8:2::/64 via 2001:db8:1::10 dev enp1s0 metric 1024 pref medium
```

Ressources supplémentaires

- **nmcli(1)** page de manuel
- **nm-settings-nmcli(5)** page de manuel

20.8. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DE NMSTATECTL

Vous pouvez ajouter une route statique à la configuration d'une connexion réseau à l'aide de l'utilitaire **nmstatectl**.

La procédure ci-dessous permet de configurer les routes suivantes :

- Une route IPv4 vers le réseau distant **198.51.100.0/24**. La passerelle correspondante avec l'adresse IP **192.0.2.10** est accessible via l'interface **enp1s0**.
- Une route IPv6 vers le réseau distant **2001:db8:2::/64**. La passerelle correspondante avec l'adresse IP **2001:db8:1::10** est accessible via l'interface **enp1s0**.

Conditions préalables

- L'interface réseau **enp1s0** est configurée et se trouve dans le même sous-réseau IP que les passerelles.
- Le paquet **nmstate** est installé.

Procédure

1. Créez un fichier YAML, par exemple **~/add-static-route-to-enp1s0.yml**, avec le contenu suivant :

```
---
routes:
  config:
    - destination: 198.51.100.0/24
      next-hop-address: 192.0.2.10
      next-hop-interface: enp1s0
    - destination: 2001:db8:2::/64
      next-hop-address: 2001:db8:1::10
      next-hop-interface: enp1s0
```

2. Appliquer les paramètres au système :

```
# nmstatectl apply ~/add-static-route-to-enp1s0.yml
```

Vérification

1. Afficher les itinéraires IPv4 :

```
# ip -4 route
...
198.51.100.0/24 via 192.0.2.10 dev enp1s0
```

2. Afficher les itinéraires IPv6 :

```
# ip -6 route
...
2001:db8:2::/64 via 2001:db8:1::10 dev enp1s0 metric 1024 pref medium
```

Ressources supplémentaires

- **nmstatectl(8)** page de manuel
- `/usr/share/doc/nmstate/examples/` répertoire

20.9. CONFIGURATION D'UNE ROUTE STATIQUE À L'AIDE DU RÔLE RÉSEAU RHEL SYSTEM ROLE

Vous pouvez utiliser le rôle de système **network** RHEL pour configurer des itinéraires statiques.



IMPORTANT

Lorsque vous exécutez une séquence qui utilise le rôle système **network** RHEL, le rôle système remplace un profil de connexion existant portant le même nom si la valeur des paramètres ne correspond pas à ceux spécifiés dans la séquence. Par conséquent, indiquez toujours la configuration complète du profil de connexion réseau dans la pièce, même si, par exemple, la configuration IP existe déjà. Dans le cas contraire, le rôle rétablit les valeurs par défaut.

Selon qu'il existe déjà ou non, la procédure crée ou met à jour le profil de connexion **enp7s0** avec les paramètres suivants :

- Une adresse IPv4 statique - **192.0.2.1** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **192.0.2.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::fffe**
- Un serveur DNS IPv4 - **192.0.2.200**
- Un serveur DNS IPv6 - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**
- Routes statiques :
 - **198.51.100.0/24** avec passerelle **192.0.2.10**
 - **2001:db8:2::/64** avec passerelle **2001:db8:1::10**

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- Vous avez préparé le nœud de contrôle et les nœuds gérés
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.

Procédure

1. Créez un fichier playbook, par exemple `~/add-static-routes.yml` avec le contenu suivant :

```

---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure an Ethernet connection with static IP and additional routes
      include_role:
        name: rhel-system-roles.network

  vars:
    network_connections:
      - name: enp7s0
        type: ethernet
        autoconnect: yes
        ip:
          address:
            - 192.0.2.1/24
            - 2001:db8:1::1/64
          gateway4: 192.0.2.254
          gateway6: 2001:db8:1::fffe
        dns:
          - 192.0.2.200
          - 2001:db8:1::ffbb
        dns_search:
          - example.com
        route:
          - network: 198.51.100.0
            prefix: 24
            gateway: 192.0.2.10
          - network: 2001:db8:2::
            prefix: 64
            gateway: 2001:db8:1::10
        state: up

```

2. Exécutez le manuel de jeu :

```
# ansible-playbook ~/add-static-routes.yml
```

Vérification

1. Sur les nœuds gérés :

a. Afficher les itinéraires IPv4 :

```
# ip -4 route  
...  
198.51.100.0/24 via 192.0.2.10 dev enp7s0
```

b. Afficher les itinéraires IPv6 :

```
# ip -6 route  
...  
2001:db8:2::/64 via 2001:db8:1::10 dev enp7s0 metric 1024 pref medium
```

Ressources supplémentaires

- [/usr/share/ansible/roles/rhel-system-roles.network/README.md](#) fichier

CHAPITRE 21. CONFIGURATION DU ROUTAGE BASÉ SUR DES RÈGLES POUR DÉFINIR DES ITINÉRAIRES ALTERNATIFS

Par défaut, le noyau de RHEL décide où transmettre les paquets réseau en fonction de l'adresse de destination à l'aide d'une table de routage. Le routage basé sur des règles vous permet de configurer des scénarios de routage complexes. Par exemple, vous pouvez acheminer les paquets en fonction de divers critères, tels que l'adresse source, les métadonnées du paquet ou le protocole.



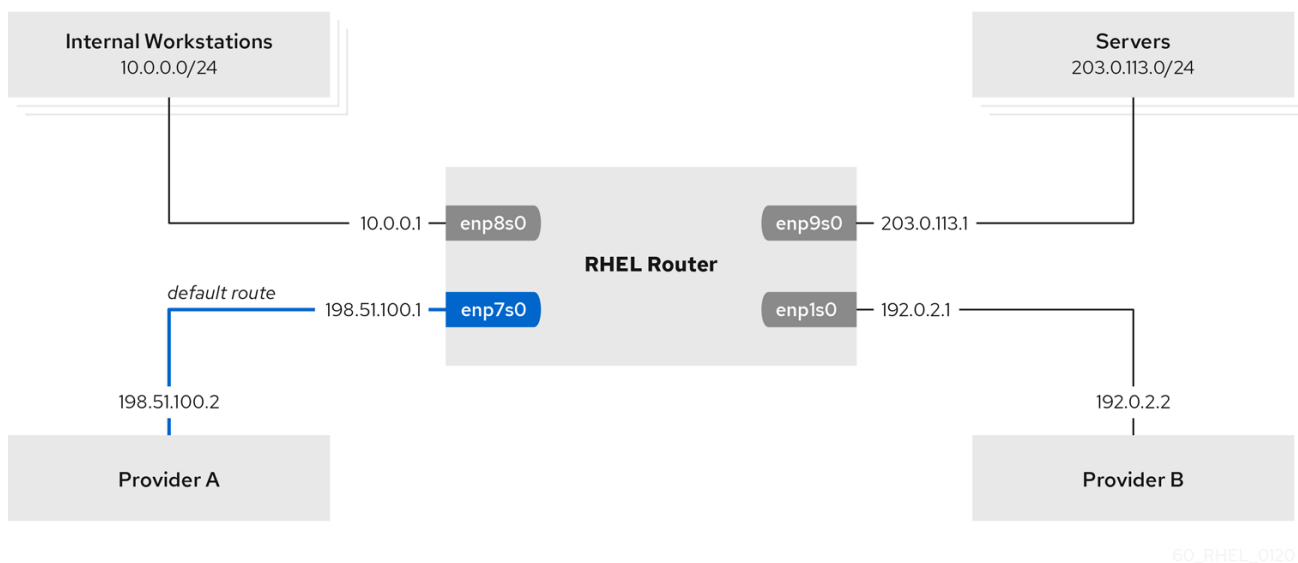
NOTE

Sur les systèmes qui utilisent NetworkManager, seul l'utilitaire **nmcli** permet de définir des règles de routage et d'affecter des itinéraires à des tables spécifiques.

21.1. ROUTAGE DU TRAFIC D'UN SOUS-RÉSEAU SPÉCIFIQUE VERS UNE PASSERELLE PAR DÉFAUT DIFFÉRENTE À L'AIDE DE NMCLI

Vous pouvez utiliser le routage basé sur des règles pour configurer une passerelle par défaut différente pour le trafic provenant de certains sous-réseaux. Par exemple, vous pouvez configurer RHEL comme un routeur qui, par défaut, achemine tout le trafic vers le fournisseur d'accès Internet A à l'aide de la route par défaut. Toutefois, le trafic reçu du sous-réseau des postes de travail internes est acheminé vers le fournisseur B.

La procédure suppose la topologie de réseau suivante :



Conditions préalables

- Le système utilise **NetworkManager** pour configurer le réseau, ce qui est la valeur par défaut.
- Le routeur RHEL que vous souhaitez configurer dans la procédure possède quatre interfaces réseau :
 - L'interface **enp7s0** est connectée au réseau du fournisseur A. L'IP de la passerelle dans le réseau du fournisseur est **198.51.100.2**, et le réseau utilise un masque de réseau **/30**.
 - L'interface **enp1s0** est connectée au réseau du fournisseur B. L'IP de la passerelle dans le réseau du fournisseur est **192.0.2.2**, et le réseau utilise un masque de réseau **/30**.

- L'interface **enp8s0** est connectée au sous-réseau **10.0.0.0/24** avec des postes de travail internes.
- L'interface **enp9s0** est connectée au sous-réseau **203.0.113.0/24** où se trouvent les serveurs de l'entreprise.
- Les hôtes du sous-réseau des postes de travail internes utilisent **10.0.0.1** comme passerelle par défaut. Dans la procédure, vous attribuez cette adresse IP à l'interface réseau **enp8s0** du routeur.
- Les hôtes du sous-réseau du serveur utilisent **203.0.113.1** comme passerelle par défaut. Dans la procédure, vous attribuez cette adresse IP à l'interface réseau **enp9s0** du routeur.
- Le service **firewalld** est activé et actif.

Procédure

1. Configurez l'interface réseau vers le fournisseur A :

```
# nmcli connection add type ethernet con-name Provider-A ifname enp7s0
ipv4.method manual ipv4.addresses 198.51.100.1/30 ipv4.gateway 198.51.100.2
ipv4.dns 198.51.100.200 connection.zone external
```

La commande **nmcli connection add** crée un profil de connexion NetworkManager. La commande utilise les options suivantes :

- **type ethernet**: Définit que le type de connexion est Ethernet.
 - **con-nameconnection_name**: Définit le nom du profil. Utilisez un nom significatif pour éviter toute confusion.
 - **ifnamenetwork_device**: Définit l'interface réseau.
 - **ipv4.method manual** permet de configurer une adresse IP statique.
 - **ipv4.addressesIP_address/subnet_mask**: Définit les adresses IPv4 et le masque de sous-réseau.
 - **ipv4.gatewayIP_address**: Définit l'adresse de la passerelle par défaut.
 - **ipv4.dnsIP_of_DNS_server**: Définit l'adresse IPv4 du serveur DNS.
 - **connection.zonefirewalld_zone**: Affecte l'interface réseau à la zone définie **firewalld**. Notez que **firewalld** active automatiquement le masquage pour les interfaces assignées à la zone **external**.
2. Configurez l'interface réseau vers le fournisseur B :

```
# nmcli connection add type ethernet con-name Provider-B ifname enp1s0
ipv4.method manual ipv4.addresses 192.0.2.1/30 ipv4.routes "0.0.0.0/0 192.0.2.2
table=5000" connection.zone external
```

Cette commande utilise le paramètre **ipv4.routes** au lieu de **ipv4.gateway** pour définir la passerelle par défaut. Ceci est nécessaire pour assigner la passerelle par défaut de cette connexion à une table de routage (**5000**) différente de la table par défaut. NetworkManager crée automatiquement cette nouvelle table de routage lorsque la connexion est activée.

- Configurez l'interface réseau vers le sous-réseau des postes de travail internes :

```
# nmcli connection add type ethernet con-name Internal-Workstations ifname enp8s0
ipv4.method manual ipv4.addresses 10.0.0.1/24 ipv4.routes "10.0.0.0/24 table=5000"
ipv4.routing-rules "priority 5 from 10.0.0.0/24 table 5000" connection.zone trusted
```

Cette commande utilise le paramètre **ipv4.routes** pour ajouter une route statique à la table de routage avec l'ID **5000**. Cette route statique pour le sous-réseau **10.0.0.0/24** utilise l'IP de l'interface réseau locale vers le fournisseur B (**192.0.2.1**) comme prochain saut.

En outre, la commande utilise le paramètre **ipv4.routing-rules** pour ajouter une règle de routage avec la priorité **5** qui achemine le trafic du sous-réseau **10.0.0.0/24** vers la table **5000**. Les valeurs faibles ont une priorité élevée.

Notez que la syntaxe du paramètre **ipv4.routing-rules** est la même que celle d'une commande **ip rule add**, sauf que **ipv4.routing-rules** exige toujours la spécification d'une priorité.

- Configurez l'interface réseau vers le sous-réseau du serveur :

```
# nmcli connection add type ethernet con-name Servers ifname enp9s0 ipv4.method
manual ipv4.addresses 203.0.113.1/24 connection.zone trusted
```

Vérification

- Sur un hôte RHEL dans le sous-réseau des stations de travail internes :

- Installez le paquetage **traceroute**:

```
# dnf install traceroute
```

- Utilisez l'utilitaire **traceroute** pour afficher l'itinéraire vers un hôte sur Internet :

```
# traceroute redhat.com
traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
 1 10.0.0.1 (10.0.0.1)  0.337 ms 0.260 ms 0.223 ms
 2 192.0.2.1 (192.0.2.1) 0.884 ms 1.066 ms 1.248 ms
 ...
```

La sortie de la commande indique que le routeur envoie des paquets sur **192.0.2.1**, qui est le réseau du fournisseur B.

- Sur un hôte RHEL dans le sous-réseau du serveur :

- Installez le paquetage **traceroute**:

```
# dnf install traceroute
```

- Utilisez l'utilitaire **traceroute** pour afficher l'itinéraire vers un hôte sur Internet :

```
# traceroute redhat.com
traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
 1 203.0.113.1 (203.0.113.1) 2.179 ms 2.073 ms 1.944 ms
 2 198.51.100.2 (198.51.100.2) 1.868 ms 1.798 ms 1.549 ms
 ...
```


La sortie de la commande indique que le routeur envoie des paquets sur **198.51.100.2**, qui est le réseau du fournisseur A.

Étapes de dépannage

Sur le routeur RHEL :

1. Affichez la liste des règles :

```
# ip rule list
0: from all lookup local
5: from 10.0.0.0/24 lookup 5000
32766: from all lookup main
32767: from all lookup default
```

Par défaut, RHEL contient des règles pour les tables **local**, **main**, et **default**.

2. Afficher les itinéraires dans la table **5000**:

```
# ip route list table 5000
0.0.0.0/0 via 192.0.2.2 dev enp1s0 proto static metric 100
10.0.0.0/24 dev enp8s0 proto static scope link src 192.0.2.1 metric 102
```

3. Affichez les interfaces et les zones de pare-feu :

```
# firewall-cmd --get-active-zones
external
  interfaces: enp1s0 enp7s0
trusted
  interfaces: enp8s0 enp9s0
```

4. Vérifiez que le masquage est activé dans la zone **external**:

```
# firewall-cmd --info-zone=external
external (active)
target: default
icmp-block-inversion: no
interfaces: enp1s0 enp7s0
sources:
services: ssh
ports:
protocols:
masquerade: yes
...
```

Ressources supplémentaires

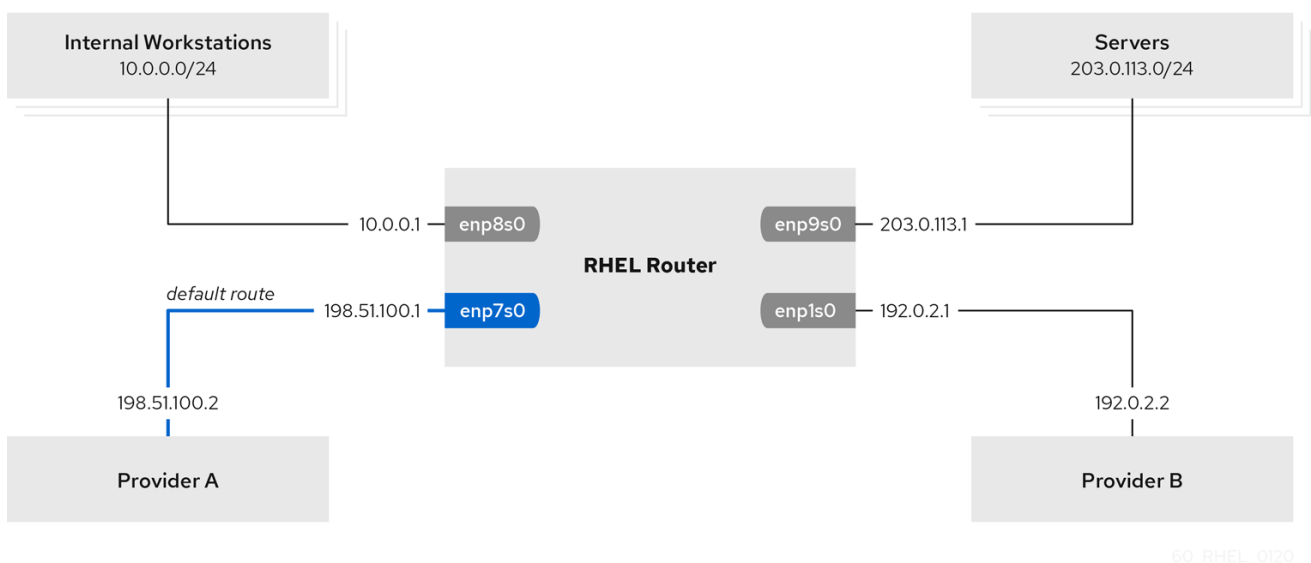
- **nm-settings(5)** page de manuel
- **nmcli(1)** page de manuel
- [Est-il possible de mettre en place un routage basé sur des règles avec NetworkManager dans RHEL ?](#)

21.2. ROUTAGE DU TRAFIC D'UN SOUS-RÉSEAU SPÉCIFIQUE VERS UNE PASSERELLE PAR DÉFAUT DIFFÉRENTE EN UTILISANT LE RÔLE DE SYSTÈME RHEL DU RÉSEAU

Vous pouvez utiliser le routage basé sur des règles pour configurer une passerelle par défaut différente pour le trafic provenant de certains sous-réseaux. Par exemple, vous pouvez configurer RHEL comme un routeur qui, par défaut, achemine tout le trafic vers le fournisseur d'accès Internet A à l'aide de la route par défaut. Toutefois, le trafic reçu du sous-réseau des postes de travail internes est acheminé vers le fournisseur B.

Pour configurer le routage basé sur des stratégies à distance et sur plusieurs nœuds, vous pouvez utiliser le rôle système RHEL **network**. Effectuez cette procédure sur le nœud de contrôle Ansible.

Cette procédure suppose la topologie de réseau suivante :



Conditions préalables

- Vous avez préparé le nœud de contrôle et les nœuds gérés
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo** sur ces nœuds.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.
- Les nœuds gérés utilisent les services **NetworkManager** et **firewalld**.
- Les nœuds gérés que vous souhaitez configurer possèdent quatre interfaces réseau :
 - L'interface **enp7s0** est connectée au réseau du fournisseur A. L'IP de la passerelle dans le réseau du fournisseur est **198.51.100.2**, et le réseau utilise un masque de réseau **/30**.
 - L'interface **enp1s0** est connectée au réseau du fournisseur B. L'IP de la passerelle dans le réseau du fournisseur est **192.0.2.2**, et le réseau utilise un masque de réseau **/30**.

- L'interface **enp8s0** est connectée au sous-réseau **10.0.0.0/24** avec des postes de travail internes.
- L'interface **enp9s0** est connectée au sous-réseau **203.0.113.0/24** où se trouvent les serveurs de l'entreprise.
- Les hôtes du sous-réseau des postes de travail internes utilisent **10.0.0.1** comme passerelle par défaut. Dans la procédure, vous attribuez cette adresse IP à l'interface réseau **enp8s0** du routeur.
- Les hôtes du sous-réseau du serveur utilisent **203.0.113.1** comme passerelle par défaut. Dans la procédure, vous attribuez cette adresse IP à l'interface réseau **enp9s0** du routeur.

Procédure

1. Créez un fichier playbook, par exemple `~/pbr.yml`, avec le contenu suivant :

```

---
- name: Configuring policy-based routing
  hosts: managed-node-01.example.com
  tasks:
    - name: Routing traffic from a specific subnet to a different default gateway
      include_role:
        name: rhel-system-roles.network

  vars:
    network_connections:
      - name: Provider-A
        interface_name: enp7s0
        type: ethernet
        autoconnect: True
        ip:
          address:
            - 198.51.100.1/30
          gateway4: 198.51.100.2
          dns:
            - 198.51.100.200
        state: up
        zone: external

      - name: Provider-B
        interface_name: enp1s0
        type: ethernet
        autoconnect: True
        ip:
          address:
            - 192.0.2.1/30
        route:
          - network: 0.0.0.0
            prefix: 0
            gateway: 192.0.2.2
            table: 5000
        state: up
        zone: external

    - name: Internal-Workstations
  
```

```

interface_name: enp8s0
type: ethernet
autoconnect: True
ip:
  address:
    - 10.0.0.1/24
  route:
    - network: 10.0.0.0
      prefix: 24
      table: 5000
  routing_rule:
    - priority: 5
      from: 10.0.0.0/24
      table: 5000
state: up
zone: trusted

- name: Servers
  interface_name: enp9s0
  type: ethernet
  autoconnect: True
  ip:
    address:
      - 203.0.113.1/24
  state: up
  zone: trusted

```

2. Exécutez le manuel de jeu :

```
# ansible-playbook ~/pbr.yml
```

Vérification

1. Sur un hôte RHEL dans le sous-réseau des stations de travail internes :
 - a. Installez le paquetage **traceroute**:

```
# dnf install traceroute
```

- b. Utilisez l'utilitaire **traceroute** pour afficher l'itinéraire vers un hôte sur Internet :

```

# traceroute redhat.com
traceroute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
 1 10.0.0.1 (10.0.0.1)  0.337 ms  0.260 ms  0.223 ms
 2 192.0.2.1 (192.0.2.1) 0.884 ms 1.066 ms 1.248 ms
 ...

```

La sortie de la commande indique que le routeur envoie des paquets sur **192.0.2.1**, qui est le réseau du fournisseur B.

2. Sur un hôte RHEL dans le sous-réseau du serveur :
 - a. Installez le paquetage **traceroute**:

```
# dnf install traceroute
```

- b. Utilisez l'utilitaire **tracertoute** pour afficher l'itinéraire vers un hôte sur Internet :

```
# tracertoute redhat.com
tracertoute to redhat.com (209.132.183.105), 30 hops max, 60 byte packets
 1 203.0.113.1 (203.0.113.1)  2.179 ms  2.073 ms  1.944 ms
 2 198.51.100.2 (198.51.100.2) 1.868 ms  1.798 ms  1.549 ms
 ...
```

La sortie de la commande indique que le routeur envoie des paquets sur **198.51.100.2**, qui est le réseau du fournisseur A.

3. Sur le routeur RHEL que vous avez configuré à l'aide du rôle de système RHEL :

- a. Affichez la liste des règles :

```
# ip rule list
0:   from all lookup local
5:   from 10.0.0.0/24 lookup 5000
32766: from all lookup main
32767: from all lookup default
```

Par défaut, RHEL contient des règles pour les tables **local**, **main**, et **default**.

- b. Affichez les itinéraires dans la table **5000**:

```
# ip route list table 5000
0.0.0.0/0 via 192.0.2.2 dev enp1s0 proto static metric 100
10.0.0.0/24 dev enp8s0 proto static scope link src 192.0.2.1 metric 102
```

- c. Affichez les interfaces et les zones de pare-feu :

```
# firewall-cmd --get-active-zones
external
  interfaces: enp1s0 enp7s0
trusted
  interfaces: enp8s0 enp9s0
```

- d. Vérifiez que le masquage est activé dans la zone **external**:

```
# firewall-cmd --info-zone=external
external (active)
  target: default
  icmp-block-inversion: no
  interfaces: enp1s0 enp7s0
  sources:
  services: ssh
  ports:
  protocols:
masquerade: yes
 ...
```

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

CHAPITRE 22. CRÉATION D'UNE INTERFACE FICTIVE

En tant qu'utilisateur de Red Hat Enterprise Linux, vous pouvez créer et utiliser des interfaces réseau fictives à des fins de débogage et de test. Une interface fictive permet à un périphérique d'acheminer des paquets sans les transmettre. Elle vous permet de créer des périphériques supplémentaires de type loopback gérés par NetworkManager et de faire en sorte qu'une adresse SLIP (Serial Line Internet Protocol) inactive ressemble à une adresse réelle pour les programmes locaux.

22.1. CRÉATION D'UNE INTERFACE FICTIVE AVEC UNE ADRESSE IPV4 ET IPV6 À L'AIDE DE NMCLI

Vous pouvez créer une interface fictive avec différents paramètres, tels que les adresses IPv4 et IPv6. Après avoir créé l'interface, NetworkManager l'assigne automatiquement à la zone par défaut **public firewalld**.

Procédure

- Créez une interface fictive nommée **dummy0** avec des adresses IPv4 et IPv6 statiques :

```
# nmcli connection add type dummy ifname dummy0 ipv4.method manual
  ipv4.addresses 192.0.2.1/24 ipv6.method manual ipv6.addresses 2001:db8:2::1/64
```



NOTE

Pour configurer une interface fictive sans adresses IPv4 et IPv6, définissez les paramètres **ipv4.method** et **ipv6.method** sur **disabled**. Sinon, l'auto-configuration IP échoue et NetworkManager désactive la connexion et supprime le périphérique.

Vérification

- Liste des profils de connexion :

```
# nmcli connection show
NAME          UUID                                TYPE  DEVICE
dummy-dummy0  aaf6eb56-73e5-4746-9037-eed42caa8a65  dummy  dummy0
```

Ressources supplémentaires

- **nm-settings(5)** page de manuel

CHAPITRE 23. UTILISATION DE NMSTATE-AUTOCONF POUR CONFIGURER AUTOMATIQUEMENT L'ÉTAT DU RÉSEAU À L'AIDE DE LLDP

Les périphériques réseau peuvent utiliser le protocole Link Layer Discovery Protocol (LLDP) pour annoncer leur identité, leurs capacités et leurs voisins au sein d'un réseau local. L'utilitaire **nmstate-autoconf** peut utiliser ces informations pour configurer automatiquement les interfaces du réseau local.



IMPORTANT

L'utilitaire **nmstate-autoconf** est fourni en tant qu'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat, peuvent ne pas être complètes sur le plan fonctionnel et Red Hat ne recommande pas de les utiliser pour la production. Ces aperçus offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Consultez la section [Portée de l'assistance](#) pour les fonctionnalités de l'aperçu technologique sur le portail client de Red Hat pour obtenir des informations sur la portée de l'assistance pour les fonctionnalités de l'aperçu technologique.

23.1. UTILISATION DE NMSTATE-AUTOCONF POUR CONFIGURER AUTOMATIQUEMENT LES INTERFACES RÉSEAU

L'utilitaire **nmstate-autoconf** utilise LLDP pour identifier les paramètres VLAN des interfaces connectées à un commutateur afin de configurer les périphériques locaux.

Cette procédure suppose le scénario suivant et que le commutateur diffuse les paramètres VLAN à l'aide de LLDP :

- Les interfaces **enp1s0** et **enp2s0** du serveur RHEL sont connectées à des ports de commutation configurés avec l'ID VLAN **100** et le nom VLAN **prod-net**.
- L'interface **enp3s0** du serveur RHEL est connectée à un port de commutateur configuré avec l'ID VLAN **200** et le nom VLAN **mgmt-net**.

L'utilitaire **nmstate-autoconf** utilise ensuite ces informations pour créer les interfaces suivantes sur le serveur :

- **bond100** - Une interface de liaison avec **enp1s0** et **enp2s0** comme ports.
- **prod-net** - Une interface VLAN au sommet de **bond100** avec l'ID VLAN **100**.
- **mgmt-net** - Une interface VLAN sur **enp3s0** avec ID VLAN **200**

Si vous connectez plusieurs interfaces réseau à différents ports de commutateur pour lesquels LLDP diffuse le même ID VLAN, **nmstate-autoconf** crée un lien avec ces interfaces et configure en outre l'ID VLAN commun.

Conditions préalables

- Le paquet **nmstate** est installé.

- LLDP est activé sur le commutateur réseau.
- Les interfaces Ethernet sont en service.

Procédure

1. Activer LLDP sur les interfaces Ethernet :

- a. Créez un fichier YAML, par exemple `~/enable-lldp.yml`, avec le contenu suivant :

```
interfaces:
- name: enp1s0
  type: ethernet
  lldp:
    enabled: true
- name: enp2s0
  type: ethernet
  lldp:
    enabled: true
- name: enp3s0
  type: ethernet
  lldp:
    enabled: true
```

- b. Appliquer les paramètres au système :

```
# nmstatectl apply ~/enable-lldp.yml
```

2. Configurez les interfaces réseau à l'aide de LLDP :

- a. En option, lancer une simulation pour afficher et vérifier la configuration YAML générée par **nmstate-autoconf**:

```
# nmstate-autoconf -d enp1s0,enp2s0,enp3s0
---
interfaces:
- name: prod-net
  type: vlan
  state: up
  vlan:
    base-iface: bond100
    id: 100
- name: mgmt-net
  type: vlan
  state: up
  vlan:
    base-iface: enp3s0
    id: 200
- name: bond100
  type: bond
  state: up
  link-aggregation:
    mode: balance-rr
```



```
port:  
- enp1s0  
- enp2s0
```

- b. Utilisez **nmstate-autoconf** pour générer la configuration en fonction des informations reçues de LLDP et appliquer les paramètres au système :

```
# nmstate-autoconf enp1s0,enp2s0,enp3s0
```

Prochaines étapes

- S'il n'y a pas de serveur DHCP dans votre réseau qui fournit les paramètres IP aux interfaces, configurez-les manuellement. Pour plus de détails, voir :
 - [Configuration d'une connexion Ethernet](#)
 - [Configuration de la liaison réseau](#)

Vérification

1. Affiche les paramètres des différentes interfaces :

```
# nmstatectl show <interface_name>
```

Ressources supplémentaires

- **nmstate-autoconf(8)** page de manuel

CHAPITRE 24. UTILISATION DE LLDP POUR DÉBOGUER LES PROBLÈMES DE CONFIGURATION DU RÉSEAU

Vous pouvez utiliser le protocole Link Layer Discovery Protocol (LLDP) pour déboguer les problèmes de configuration du réseau dans la topologie. Cela signifie que LLDP peut signaler des incohérences de configuration avec d'autres hôtes ou routeurs et commutateurs.

24.1. DÉBOGAGE D'UNE CONFIGURATION VLAN INCORRECTE À L'AIDE D'INFORMATIONS LLDP

Si vous avez configuré un port de commutateur pour utiliser un certain VLAN et qu'un hôte ne reçoit pas ces paquets VLAN, vous pouvez utiliser le protocole Link Layer Discovery Protocol (LLDP) pour déboguer le problème. Effectuez cette procédure sur l'hôte qui ne reçoit pas les paquets.

Conditions préalables

- Le paquet **nmstate** est installé.
- Le commutateur prend en charge le protocole LLDP.
- LLDP est activé sur les appareils voisins.

Procédure

1. Créez le fichier **~/enable-LLDP-enp1s0.yml** avec le contenu suivant :

```
interfaces:  
  - name: enp1s0  
    type: ethernet  
    lldp:  
      enabled: true
```

2. Utilisez le fichier **~/enable-LLDP-enp1s0.yml** pour activer LLDP sur l'interface **enp1s0**:

```
# nmstatectl apply ~/enable-LLDP-enp1s0.yml
```

3. Affiche les informations LLDP :

```
# nmstatectl show enp1s0  
- name: enp1s0  
  type: ethernet  
  state: up  
  ipv4:  
    enabled: false  
    dhcp: false  
  ipv6:  
    enabled: false  
    autoconf: false  
    dhcp: false  
  lldp:  
    enabled: true  
    neighbors:  
      - type: 5
```

```

system-name: Summit300-48
- type: 6
system-description: Summit300-48 - Version 7.4e.1 (Build 5)
  05/27/05 04:53:11
- type: 7
system-capabilities:
- MAC Bridge component
- Router
- type: 1
  _description: MAC address
  chassis-id: 00:01:30:F9:AD:A0
  chassis-id-type: 4
- type: 2
  _description: Interface name
  port-id: 1/1
  port-id-type: 5
- type: 127
ieee-802-1-vlans:
- name: v2-0488-03-0505
  vid: 488
  oui: 00:80:c2
  subtype: 3
- type: 127
ieee-802-3-mac-phy-conf:
  autoneg: true
  operational-mau-type: 16
  pmd-autoneg-cap: 27648
  oui: 00:12:0f
  subtype: 1
- type: 127
ieee-802-1-ppvids:
- 0
  oui: 00:80:c2
  subtype: 2
- type: 8
management-addresses:
- address: 00:01:30:F9:AD:A0
  address-subtype: MAC
  interface-number: 1001
  interface-number-subtype: 2
- type: 127
ieee-802-3-max-frame-size: 1522
  oui: 00:12:0f
  subtype: 4
mac-address: 82:75:BE:6F:8C:7A
mtu: 1500

```

4. Vérifiez la sortie pour vous assurer que les paramètres correspondent à la configuration attendue. Par exemple, les informations LLDP de l'interface connectée au commutateur montrent que le port du commutateur auquel cet hôte est connecté utilise l'ID VLAN **448**:

```

- type: 127
ieee-802-1-vlans:
- name: v2-0488-03-0505
  vid: 488

```

Si la configuration réseau de l'interface **enp1s0** utilise un ID VLAN différent, modifiez-le en conséquence.

Ressources supplémentaires

[Configuration du marquage VLAN](#)

CHAPITRE 25. CRÉATION MANUELLE DE PROFILS NETWORKMANAGER AU FORMAT KEYFILE

Par défaut, NetworkManager stocke les profils au format keyfile. Par exemple, l'utilitaire **nmcli**, le rôle de système RHEL **network** ou l'API **nmstate** pour gérer les profils utilisent ce format. Cependant, NetworkManager prend toujours en charge les profils au format **ifcfg**, qui est obsolète.

25.1. FORMAT DU FICHIER CLÉ DES PROFILS NETWORKMANAGER

NetworkManager utilise le format INI pour stocker les profils de connexion sur le disque.

Exemple de profil de connexion Ethernet au format fichier clé :

```
[connection]
id=example_connection
uuid=82c6272d-1ff7-4d56-9c7c-0eb27c300029
type=ethernet
autoconnect=true

[ipv4]
method=auto

[ipv6]
method=auto

[ethernet]
mac-address=00:53:00:8f:fa:66
```

Chaque section correspond à un nom de paramètre de NetworkManager tel que décrit dans les pages de manuel **nm-settings(5)** et **nm-settings-keyfile(5)**. Chaque paire clé-valeur d'une section correspond à l'une des propriétés énumérées dans la spécification des paramètres de la page de manuel.

La plupart des variables des fichiers clés de NetworkManager ont une correspondance biunivoque. Cela signifie qu'une propriété du NetworkManager est stockée dans le fichier clé sous la forme d'une variable portant le même nom et ayant le même format. Il existe cependant des exceptions, principalement pour faciliter la lecture de la syntaxe du fichier clé. Pour une liste de ces exceptions, voir la page de manuel **nm-settings-keyfile(5)**.



IMPORTANT

Pour des raisons de sécurité, étant donné que les profils de connexion peuvent contenir des informations sensibles, telles que des clés privées et des phrases de passe, NetworkManager n'utilise que des fichiers de configuration appartenant à **root** et qui ne peuvent être lus et écrits que par **root**.

En fonction de l'objectif du profil de connexion, enregistrez-le dans l'un des répertoires suivants :

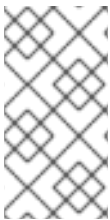
- **/etc/NetworkManager/system-connections/**: Emplacement des profils persistants. Si vous modifiez un profil persistant à l'aide de l'API NetworkManager, NetworkManager écrit et écrase les fichiers dans ce répertoire.
- **/run/NetworkManager/system-connections/**: Pour les profils temporaires qui sont automatiquement supprimés lorsque vous redémarrez le système.

- **/usr/lib/NetworkManager/system-connections/**: Pour les profils immuables pré-déployés. Lorsque vous modifiez un tel profil à l'aide de l'API de NetworkManager, NetworkManager copie ce profil dans le stockage persistant ou temporaire.

NetworkManager ne recharge pas automatiquement les profils à partir du disque. Lorsque vous créez ou mettez à jour un profil de connexion au format keyfile, utilisez la commande **nmcli connection reload** pour informer NetworkManager des changements.

25.2. CRÉATION D'UN PROFIL NETWORKMANAGER AU FORMAT KEYFILE

Vous pouvez créer manuellement un profil de connexion NetworkManager au format keyfile.



NOTE

La création ou la mise à jour manuelle des fichiers de configuration peut entraîner une configuration réseau inattendue ou non fonctionnelle. Red Hat vous recommande d'utiliser les utilitaires NetworkManager, tels que **nmcli**, le rôle de système RHEL **network** ou l'API **nmstate** pour gérer les connexions NetworkManager.

Procédure

1. Si vous créez un profil pour une interface matérielle, telle qu'Ethernet, affichez l'adresse MAC de cette interface :

```
# ip address show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
link/ether 00:53:00:8f:fa:66 brd ff:ff:ff:ff:ff:ff
```

2. Créez un profil de connexion. Par exemple, pour un profil de connexion d'un périphérique Ethernet qui utilise DHCP, créez le fichier **/etc/NetworkManager/system-connections/example.nmconnection** avec le contenu suivant :

```
[connection]
id=example_connection
type=ethernet
autoconnect=true

[ipv4]
method=auto

[ipv6]
method=auto

[ethernet]
mac-address=00:53:00:8f:fa:66
```



NOTE

Vous pouvez utiliser n'importe quel nom de fichier avec un suffixe **.nmconnection**. Toutefois, lorsque vous utiliserez ultérieurement les commandes **nmcli** pour gérer la connexion, vous devrez utiliser le nom de la connexion défini dans la variable **id** lorsque vous ferez référence à cette connexion. Si vous omettez la variable **id**, utilisez le nom de fichier sans **.nmconnection** pour faire référence à cette connexion.

3. Définissez les autorisations sur le fichier de configuration de sorte que seul l'utilisateur **root** puisse le lire et le mettre à jour :

```
# chown root:root /etc/NetworkManager/system-connections/example.nmconnection
# chmod 600 /etc/NetworkManager/system-connections/example.nmconnection
```

4. Recharger les profils de connexion :

```
# nmcli connection reload
```

5. Vérifiez que NetworkManager a bien lu le profil dans le fichier de configuration :

```
# nmcli -f NAME,UUID,FILENAME connection
NAME          UUID          FILENAME
example-connection 86da2486-068d-4d05-9ac7-957ec118afba
/etc/NetworkManager/system-connections/example.nmconnection
...
```

Si la commande n'affiche pas la connexion nouvellement ajoutée, vérifiez que les autorisations du fichier et la syntaxe utilisée dans le fichier sont correctes.

6. Facultatif : si vous définissez la variable **autoconnect** dans le profil sur **false**, activez la connexion :

```
# nmcli connection up example_connection
```

Vérification

1. Afficher le profil de connexion :

```
# nmcli connection show example_connection
```

2. Affiche les paramètres IP de l'interface :

```
# ip address show enp1s0
```

Ressources supplémentaires

- [nm-settings-keyfile \(5\)](#)

25.3. MIGRATION DES PROFILS NETWORKMANAGER DU FORMAT IFCFG AU FORMAT KEYFILE

Vous pouvez utiliser la commande **nmcli connection migrate** pour migrer vos profils de connexion **ifcfg** existants vers le format keyfile. De cette manière, tous vos profils de connexion se trouveront à un seul endroit et dans le format préféré.

Conditions préalables

- Vous disposez de profils de connexion au format **ifcfg** dans le répertoire **/etc/sysconfig/network-scripts/**.

Procédure

- Migrer les profils de connexion :

```
# nmcli connection migrate
Connection 'enp1s0' (43ed18ab-f0c4-4934-af3d-2b3333948e45) successfully migrated.
Connection 'enp2s0' (883333e8-1b87-4947-8ceb-1f8812a80a9b) successfully migrated.
...
```

Vérification

- En option, vous pouvez vérifier que vous avez migré avec succès tous vos profils de connexion :

```
# nmcli -f TYPE,FILENAME,NAME connection
TYPE  FILENAME                                     NAME
ethernet /etc/NetworkManager/system-connections/enp1s0.nmconnection  enp1s0
ethernet /etc/NetworkManager/system-connections/enp2s0.nmconnection  enp2s0
...
```

Ressources supplémentaires

- [nm-settings-keyfile\(5\)](#)
- [nm-settings-ifcfg-rh\(5\)](#)
- [nmcli\(1\)](#)

25.4. UTILISATION DE NMCLI POUR CRÉER DES PROFILS DE CONNEXION DE FICHIERS CLÉS EN MODE DÉCONNECTÉ

Red Hat recommande d'utiliser les utilitaires NetworkManager, tels que **nmcli**, le rôle de système RHEL **network** ou l'API **nmstate** pour gérer les connexions NetworkManager, afin de créer et de mettre à jour les fichiers de configuration. Cependant, vous pouvez également créer divers profils de connexion au format keyfile en mode hors ligne à l'aide de la commande **nmcli --offline connection add**.

Le mode hors ligne permet à **nmcli** de fonctionner sans le service **NetworkManager** pour produire des profils de connexion de fichiers clés par le biais de la sortie standard. Cette fonction peut s'avérer utile dans les cas suivants

- Vous souhaitez créer vos profils de connexion qui doivent être pré-déployés quelque part. Par exemple dans une image de conteneur, ou en tant que paquetage RPM.
- Vous souhaitez créer vos profils de connexion dans un environnement où le service **NetworkManager** n'est pas disponible. Par exemple, lorsque vous souhaitez utiliser l'utilitaire **chroot**. Ou encore, lorsque vous souhaitez créer ou modifier la configuration réseau du système

RHEL à installer via le script Kickstart **%post**.

Vous pouvez créer les types de profils de connexion suivants :

- connexion Ethernet statique
- connexion Ethernet dynamique
- lien de réseau
- pont de réseau
- VLAN ou tout autre type de connexion prise en charge



AVERTISSEMENT

La création ou la mise à jour manuelle des fichiers de configuration peut entraîner une configuration inattendue ou non fonctionnelle du réseau.

Conditions préalables

- Le service **NetworkManager** est arrêté.

Procédure

1. Créez un nouveau profil de connexion au format keyfile. Par exemple, pour un profil de connexion d'un périphérique Ethernet qui n'utilise pas le protocole DHCP, exécutez une commande similaire à l'adresse **nmcli**:

```
# nmcli --offline connection add type ethernet con-name Example-Connection
  ipv4.addresses 192.0.2.1/24 ipv4.dns 192.0.2.200 ipv4.method manual >
  /etc/NetworkManager/system-connections/output.nmconnection
```



NOTE

Le nom de la connexion que vous avez spécifié avec la clé **con-name** est enregistré dans la variable **id** du profil généré. Lorsque vous utiliserez ultérieurement la commande **nmcli** pour gérer cette connexion, spécifiez-la comme suit :

- Lorsque la variable **id** n'est pas omise, utilisez le nom de la connexion, par exemple **Example-Connection**.
- Lorsque la variable **id** est omise, utilisez le nom du fichier sans le suffixe **.nmconnection**, par exemple **output**.

2. Définissez les autorisations pour le fichier de configuration afin que seul l'utilisateur **root** puisse le lire et le mettre à jour :

```
# chmod 600 /etc/NetworkManager/system-connections/output.nmconnection
# chown root:root /etc/NetworkManager/system-connections/output.nmconnection
```

- Démarrez le service **NetworkManager**:

```
# systemctl start NetworkManager.service
```

- Facultatif : si vous définissez la variable **autoconnect** dans le profil sur **false**, activez la connexion :

```
# nmcli connection up Example-Connection
```

Vérification

- Vérifiez que le service **NetworkManager** est en cours d'exécution :

```
# systemctl status NetworkManager.service
● NetworkManager.service - Network Manager
  Loaded: loaded (/usr/lib/systemd/system/NetworkManager.service; enabled; vendor preset:
  enabled)
  Active: active (running) since Wed 2022-08-03 13:08:32 CEST; 1min 40s ago
    Docs: man:NetworkManager(8)
  Main PID: 7138 (NetworkManager)
    Tasks: 3 (limit: 22901)
  Memory: 4.4M
  CGroup: /system.slice/NetworkManager.service
          └─7138 /usr/sbin/NetworkManager --no-daemon

Aug 03 13:08:33 example.com NetworkManager[7138]: <info> [1659524913.3600] device
(vlan20): state change: secondaries -> activated (reason 'none', sys-iface-state: 'assume')
Aug 03 13:08:33 example.com NetworkManager[7138]: <info> [1659524913.3607] device
(vlan20): Activation: successful, device activated.
...
```

- Vérifiez que NetworkManager peut lire le profil à partir du fichier de configuration :

```
# nmcli -f TYPE,FILENAME,NAME connection
TYPE      FILENAME                                     NAME
ethernet  /etc/NetworkManager/system-connections/output.nmconnection Example-
Connection
ethernet  /etc/sysconfig/network-scripts/ifcfg-enp1s0  enp1s0
...
```

Si la sortie n'indique pas la connexion nouvellement créée, vérifiez que les autorisations du fichier clé et la syntaxe utilisée sont correctes.

- Afficher le profil de connexion :

```
# nmcli connection show Example-Connection
connection.id:          Example-Connection
connection.uuid:        232290ce-5225-422a-9228-cb83b22056b4
connection.stable-id:   --
connection.type:        802-3-ethernet
```

```
connection.interface-name:  --  
connection.autoconnect:    yes  
...
```

Ressources supplémentaires

- **nmcli(1)**
- **nm-settings-keyfile(5)**
- [Format du fichier clé des profils NetworkManager](#)
- [Configuration d'une connexion Ethernet avec une adresse IP statique à l'aide de nmcli](#)
- [Configuration d'une connexion Ethernet avec une adresse IP dynamique à l'aide de nmcli](#)
- [Configuration du marquage des VLAN à l'aide de nmcli](#)
- [Configuration d'un pont réseau à l'aide de nmcli](#)
- [Configuration d'une liaison réseau à l'aide de nmcli](#)

CHAPITRE 26. CIBLES ET SERVICES RÉSEAU SYSTEMD

NetworkManager configure le réseau pendant le processus de démarrage du système. Cependant, lors du démarrage avec une racine distante (/), par exemple si le répertoire racine est stocké sur un périphérique iSCSI, les paramètres réseau sont appliqués dans le disque RAM initial (**initrd**) avant que RHEL ne soit démarré. Par exemple, si la configuration réseau est spécifiée sur la ligne de commande du noyau à l'aide de **rd.neednet=1** ou si une configuration est spécifiée pour monter des systèmes de fichiers distants, les paramètres réseau sont appliqués sur **initrd**.

RHEL utilise les cibles **network** et **network-online** et le service **NetworkManager-wait-online** lors de l'application des paramètres réseau. Vous pouvez également configurer les services **systemd** pour qu'ils démarrent une fois que le réseau est entièrement disponible si ces services ne peuvent pas se recharger dynamiquement.

26.1. DIFFÉRENCES ENTRE LA CIBLE SYSTEMD RÉSEAU ET LA CIBLE SYSTEMD RÉSEAU EN LIGNE

Systemd gère les unités cibles **network** et **network-online**. Les unités spéciales telles que **NetworkManager-wait-online.service**, ont les paramètres **WantedBy=network-online.target** et **Before=network-online.target**. Si elles sont activées, ces unités démarrent avec **network-online.target** et retardent la cible à atteindre jusqu'à ce qu'une certaine forme de connectivité réseau soit établie. Elles retardent la cible **network-online** jusqu'à ce que le réseau soit connecté.

La cible **network-online** démarre un service, ce qui retarde considérablement la suite de l'exécution. Systemd ajoute automatiquement des dépendances avec les paramètres **Wants** et **After** pour cette unité cible à toutes les unités de service du script **init** de System V (SysV) avec un en-tête Linux Standard Base (LSB) faisant référence à l'installation **\$network**. L'en-tête LSB est une métadonnée pour les scripts **init**. Vous pouvez l'utiliser pour spécifier des dépendances. Cette méthode est similaire à la cible **systemd**.

La cible **network** ne retarde pas de manière significative l'exécution du processus de démarrage. Atteindre la cible **network** signifie que le service responsable de la configuration du réseau a démarré. Toutefois, cela ne signifie pas qu'un périphérique réseau a été configuré. Cette cible est importante lors de l'arrêt du système. Par exemple, si un service a été commandé après la cible **network** lors du démarrage, cette dépendance est inversée lors de l'arrêt du système. Le réseau n'est pas déconnecté tant que le service n'a pas été arrêté. Toutes les unités de montage pour les systèmes de fichiers réseau distants démarrent automatiquement l'unité cible **network-online** et se placent après elle.



NOTE

L'unité cible **network-online** n'est utile que pendant le démarrage du système. Une fois le démarrage du système terminé, cette cible ne suit pas l'état en ligne du réseau. Par conséquent, vous ne pouvez pas utiliser **network-online** pour surveiller la connexion au réseau. Cette cible fournit un concept de démarrage unique du système.

26.2. APERÇU DE NETWORKMANAGER-WAIT-ONLINE

Le service **NetworkManager-wait-online** attend, avec un délai, que le réseau soit configuré. Cette configuration du réseau implique le branchement d'un périphérique Ethernet, la recherche d'un périphérique Wi-Fi, etc. NetworkManager active automatiquement les profils appropriés qui sont configurés pour démarrer automatiquement. L'échec du processus d'activation automatique en raison d'un dépassement de délai DHCP ou d'un événement similaire peut occuper NetworkManager pendant une période prolongée. En fonction de la configuration, NetworkManager réessaie d'activer le même profil ou un profil différent.

Lorsque le démarrage est terminé, soit tous les profils sont déconnectés, soit ils sont activés avec succès. Vous pouvez configurer les profils pour qu'ils se connectent automatiquement. Voici quelques exemples de paramètres qui fixent des délais d'attente ou définissent le moment où la connexion est considérée comme active :

- **connection.wait-device-timeout** - définit le délai d'attente pour que le pilote détecte le périphérique
- **ipv4.may-fail** et **ipv6.may-fail** - définit l'activation avec une famille d'adresses IP prête, ou si une famille d'adresses particulière doit avoir terminé sa configuration.
- **ipv4.gateway-ping-timeout** - retarde l'activation.

Ressources supplémentaires

- **nm-settings(5)** page de manuel

26.3. CONFIGURER UN SERVICE SYSTEMD POUR QU'IL DÉMARRE APRÈS LE DÉMARRAGE DU RÉSEAU

Red Hat Enterprise Linux installe les fichiers de service **systemd** dans le répertoire `/usr/lib/systemd/system/`. Cette procédure crée un extrait d'insertion pour un fichier de service dans `/etc/systemd/system/service_name.service.d/` qui est utilisé avec le fichier de service dans `/usr/lib/systemd/system/` pour démarrer une version particulière de `service` une fois que le réseau est en ligne. La priorité est plus élevée si les paramètres de l'extrait d'insertion se chevauchent avec ceux du fichier de service de `/usr/lib/systemd/system/`.

Procédure

1. Pour ouvrir le fichier de service dans l'éditeur, entrez :

```
# systemctl edit service_name
```

2. Saisissez les données suivantes et enregistrez les modifications :

```
[Unit]
After=network-online.target
```

3. Rechargez le service **systemd**.

```
# systemctl daemon-reload
```

CHAPITRE 27. CONTRÔLE DU TRAFIC SOUS LINUX

Linux offre des outils pour gérer et manipuler la transmission des paquets. Le sous-système Linux de contrôle du trafic (TC) permet de contrôler, de classer, de mettre en forme et de planifier le trafic du réseau. Le sous-système TC manipule également le contenu des paquets lors de la classification en utilisant des filtres et des actions. Le sous-système TC y parvient en utilisant des disciplines de mise en file d'attente (**qdisc**), un élément fondamental de l'architecture TC.

Le mécanisme d'ordonnancement organise ou réorganise les paquets avant qu'ils n'entrent ou ne sortent des différentes files d'attente. L'ordonnanceur le plus courant est l'ordonnanceur FIFO (First-In-First-Out). Vous pouvez effectuer les opérations **qdiscs** temporairement à l'aide de l'utilitaire **tc** ou de façon permanente à l'aide de NetworkManager.

Dans Red Hat Enterprise Linux, vous pouvez configurer les disciplines de mise en file d'attente par défaut de différentes manières afin de gérer le trafic sur une interface réseau.

27.1. VUE D'ENSEMBLE DES DISCIPLINES DE LA FILE D'ATTENTE

Les disciplines de mise en file d'attente (**qdiscs**) aident à la mise en file d'attente et, plus tard, à l'ordonnancement de la transmission du trafic par une interface de réseau. Un site **qdisc** effectue deux opérations ;

- les demandes de mise en file d'attente afin qu'un paquet puisse être mis en file d'attente en vue d'une transmission ultérieure et
- les demandes de mise en file d'attente afin que l'un des paquets mis en file d'attente puisse être choisi pour une transmission immédiate.

Chaque site **qdisc** possède un numéro d'identification hexadécimal de 16 bits, appelé **handle**, auquel sont joints deux points, comme **1:** ou **abcd:**. Ce numéro est appelé numéro majeur **qdisc**. Si un site **qdisc** a des classes, les identificateurs sont formés d'une paire de deux numéros, le numéro majeur précédant le numéro mineur, **<major>:<minor>**, par exemple **abcd:1**. Le schéma de numérotation des numéros mineurs dépend du type de **qdisc**. Parfois, la numérotation est systématique, la première classe ayant l'ID **<major>:1**, la deuxième **<major>:2**, et ainsi de suite. Certains sites **qdiscs** permettent à l'utilisateur de définir arbitrairement les numéros mineurs des classes lors de leur création.

Classe **qdiscs**

Il existe différents types de **qdiscs** qui permettent de transférer des paquets vers et depuis une interface réseau. Vous pouvez configurer **qdiscs** avec des classes racines, parents ou enfants. Les points auxquels les enfants peuvent être attachés sont appelés classes. Les classes de **qdisc** sont flexibles et peuvent toujours contenir plusieurs classes enfants ou un seul enfant, **qdisc**. Il n'y a aucune interdiction à ce qu'une classe contienne elle-même une classe **qdisc**, ce qui facilite les scénarios complexes de contrôle du trafic.

Classful **qdiscs** ne stocke aucun paquet lui-même. Au lieu de cela, ils mettent les demandes en file d'attente et les retirent à l'un de leurs enfants en fonction de critères spécifiques à **qdisc**.

Finalement, ce passage récursif de paquets aboutit à l'endroit où les paquets sont stockés (ou récupérés dans le cas d'une mise en file d'attente).

Sans classe **qdiscs**

Certains sites **qdiscs** ne contiennent aucune classe enfantine et sont appelés **qdiscs** sans classe. Les **qdiscs** sans classe nécessitent moins de personnalisation que les **qdiscs** avec classe. Il suffit généralement de les rattacher à une interface.

Ressources supplémentaires

- **tc(8)** page de manuel
- **tc-actions(8)** page de manuel

27.2. INTRODUCTION AU SUIVI DES CONNEXIONS

Le cadre **Netfilter** filtre les paquets provenant d'un réseau externe au niveau d'un pare-feu. À l'arrivée d'un paquet, **Netfilter** attribue une entrée de suivi de connexion. Cette entrée stocke une marque **Netfilter** et suit les informations relatives à l'état de la connexion dans la table de mémoire. Un nouveau tuple de paquet correspond à une entrée existante. Si le tuple de paquet ne correspond pas à une entrée existante, le paquet ajoute une nouvelle entrée de suivi de connexion qui regroupe les paquets de la même connexion. Par exemple, pour s'assurer que tous les paquets d'une connexion FTP fonctionnent de la même manière, affectez une entrée de suivi de connexion à la connexion FTP.

La commande **tc** est un utilitaire de contrôle du trafic qui permet de configurer un planificateur de paquets en utilisant le **qdisc**, une discipline de mise en file d'attente configurée par le noyau pour capturer tout le trafic avant qu'un périphérique réseau ne le transmette. La commande **tc qdisc** limite le taux de bande passante des paquets appartenant à la même connexion.

L'utilitaire **tc** possède le module d'information de suivi de connexion (**ctinfo**) qui récupère les données des marques de suivi de connexion dans divers champs, avec la fonctionnalité **connmark**. Pour stocker les informations relatives à la marque du paquet, le module **ctinfo** copie la marque **Netfilter** et les informations relatives à l'état de la connexion dans le champ de métadonnées **skb**, qui correspond à la marque du tampon de la socket (**skb**).

En cas de transmission sur un support physique, un paquet perd ses métadonnées. Avant qu'un paquet ne perde ses métadonnées, le module **ctinfo** met en correspondance et copie la valeur de la marque **Netfilter** avec une valeur spécifique du point de code Diffserv (DSCP) dans le champ **IP** du paquet.

Ressources supplémentaires

- **tc(8)** et **tc-ctinfo(8)** pages de manuel

27.3. INSPECTION DES QDISCS D'UNE INTERFACE RÉSEAU À L'AIDE DE L'UTILITAIRE TC

Par défaut, les systèmes Red Hat Enterprise Linux utilisent **fq_codel qdisc**. Vous pouvez inspecter les compteurs **qdisc** à l'aide de l'utilitaire **tc**.

Procédure

1. Optionnel : Consultez votre site actuel **qdisc**:

```
# tc qdisc show dev enp0s1
```

2. Inspecter les compteurs actuels de **qdisc**:

```
# tc -s qdisc show dev enp0s1
qdisc fq_codel 0: root refcnt 2 limit 10240p flows 1024 quantum 1514 target 5.0ms interval
100.0ms memory_limit 32Mb ecn
Sent 1008193 bytes 5559 pkt (dropped 233, overlimits 55 requeues 77)
backlog 0b 0p requeues 0
```

- **dropped** - le nombre de fois qu'un paquet est abandonné parce que toutes les files d'attente sont pleines
- **overlimits** - le nombre de fois où la capacité de liaison configurée est remplie
- **sent** - le nombre de mises en file d'attente

27.4. MISE À JOUR DU QDISC PAR DÉFAUT

Si vous observez des pertes de paquets sur le réseau avec l'adresse **qdisc** actuelle, vous pouvez modifier l'adresse **qdisc** en fonction de vos besoins en matière de réseau.

Procédure

1. Visualiser la valeur par défaut actuelle de **qdisc**:

```
# sysctl -a | grep qdisc
net.core.default_qdisc = fq_codel
```

2. Visualiser le site **qdisc** de la connexion Ethernet actuelle :

```
# tc -s qdisc show dev enp0s1
qdisc fq_codel 0: root refcnt 2 limit 10240p flows 1024 quantum 1514 target 5.0ms interval
100.0ms memory_limit 32Mb ecn
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
backlog 0b 0p requeues 0
maxpacket 0 drop_overlimit 0 new_flow_count 0 ecn_mark 0
new_flows_len 0 old_flows_len 0
```

3. Mettre à jour le site **qdisc**:

```
# sysctl -w net.core.default_qdisc=pfifo_fast
```

4. Pour appliquer les modifications, rechargez le pilote de réseau :

```
# rmmod NETWORKDRIVERNAME
# modprobe NETWORKDRIVERNAME
```

5. Démarrer l'interface réseau :

```
# ip link set enp0s1 up
```

Vérification

- Visualisez le site **qdisc** de la connexion Ethernet :

```
# tc -s qdisc show dev enp0s1
qdisc pfifo_fast 0: root refcnt 2 bands 3 priomap 1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
Sent 373186 bytes 5333 pkt (dropped 0, overlimits 0 requeues 0)
backlog 0b 0p requeues 0
....
```


Ressources supplémentaires

- [Comment définir les variables de `sysctl` sur Red Hat Enterprise Linux](#)

27.5. DÉFINITION TEMPORAIRE DU QDISK ACTUEL D'UNE INTERFACE RÉSEAU À L'AIDE DE L'UTILITAIRE TC

Vous pouvez mettre à jour le site **qdisc** actuel sans modifier le site par défaut.

Procédure

1. Optionnel : Visualiser le site actuel **qdisc**:

```
# tc -s qdisc show dev enp0s1
```

2. Mettre à jour le site actuel **qdisc**:

```
# tc qdisc replace dev enp0s1 root htb
```

Vérification

- Consultez la version actualisée du site **qdisc**:

```
# tc -s qdisc show dev enp0s1
qdisc htb 8001: root refcnt 2 r2q 10 default 0 direct_packets_stat 0 direct_qlen 1000
Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
backlog 0b 0p requeues 0
```

27.6. FIXER DE FAÇON PERMANENTE LE QDISK COURANT D'UNE INTERFACE RÉSEAU EN UTILISANT NETWORKMANAGER

Vous pouvez mettre à jour la valeur actuelle de **qdisc** d'une connexion NetworkManager.

Procédure

1. Optionnel : Visualiser le site actuel **qdisc**:

```
# tc qdisc show dev enp0s1
qdisc fq_codel 0: root refcnt 2
```

2. Mettre à jour le site actuel **qdisc**:

```
# nmcli connection modify enp0s1 tc.qdiscs 'root pfifo_fast'
```

3. Optionnel : Pour ajouter une autre adresse **qdisc** à l'adresse **qdisc** existante, utilisez l'option **tc.qdisc**:

```
# nmcli connection modify enp0s1 tc.qdisc 'ingress handle ffff'
```

4. Activer les modifications :

```
# nmcli connection up enp0s1
```

Vérification

- Visualiser le site **qdisc** de l'interface réseau :

```
# tc qdisc show dev enp0s1
qdisc pfifo_fast 8001: root refcnt 2 bands 3 priomap  1 2 2 2 1 2 0 0 1 1 1 1 1 1 1 1
qdisc ingress ffff: parent ffff:fff1 -----
```

Ressources supplémentaires

- **nm-settings(5)** page de manuel

27.7. CONFIGURATION DE LA LIMITATION DU DÉBIT DES PAQUETS À L'AIDE DE L'UTILITAIRE TC-CTINFO

L'entrée de suivi de connexion stocke la marque **Netfilter** et les informations de connexion. Lorsqu'un routeur transmet un paquet provenant du pare-feu, il supprime ou modifie l'entrée de suivi de connexion du paquet. Le module d'informations de suivi de connexion (**ctinfo**) récupère les données des marques de suivi de connexion dans différents champs. Ce module du noyau préserve la marque **Netfilter** en la copiant dans le champ de métadonnées de la marque de la mémoire tampon de la socket (**skb**). Les étapes suivantes permettent de limiter le débit des paquets sur les systèmes serveurs à l'aide du module **ctinfo**.

Conditions préalables

- L'utilitaire **iperf3** est installé sur un serveur et un client.

Procédure

1. Effectuez les étapes suivantes sur le serveur :

- a. Ajouter un lien virtuel à l'interface réseau :

```
# ip link add name ifb4eth0 numtxqueues 48 numrxqueues 48 type ifb
```

Cette commande a les paramètres suivants :

- **name ifb4eth0** nom de l'appareil virtuel : Définit le nouveau nom de l'appareil virtuel
- **numtxqueues 48** nombre de files d'attente : Définit le nombre de files d'attente d'émission
- **numrxqueues 48** nombre de files d'attente de réception : Définit le nombre de files d'attente de réception
- **type ifb** type d'appareil : définit le type du nouvel appareil

- b. Modifier l'état de l'interface **ifb4eth0**:

```
# ip link set dev ifb4eth0 up
```

- c. Ajouter le site **qdisc** sur l'interface réseau et l'appliquer au trafic entrant :

```
# tc qdisc add dev enp1s0 handle ffff: ingress
```

Dans l'option **handle ffff:**, le paramètre **handle** attribue le numéro majeur comme valeur par défaut **ffff:** à une classe **qdisc**. Le paramètre **qdisc** est un paramètre de discipline de mise en file d'attente permettant d'analyser le contrôle du trafic.

- d. Ajouter un filtre sur l'interface **enp1s0** du protocole **ip** pour classifier les paquets :

```
# tc filter add dev enp1s0 parent ffff: protocol ip u32 match u32 0 0 action ctinfo cpmark 100 action mirrored egress redirect dev ifb4eth0
```

Cette commande possède les attributs suivants :

- **parent ffff:** Définit le numéro majeur **ffff:** pour le parent **qdisc**.
- **u32 match u32 0 0** le filtre **u32** est défini pour **match** les en-têtes IP du modèle **u32**. Le premier **0** représente le deuxième octet de l'en-tête IP et le second **0** correspond au masque qui indique au filtre quels bits doivent être pris en compte.
- **action ctinfo** action : Définit l'action pour récupérer les données du marqueur de suivi de connexion dans les différents champs.
- **cpmark 100** copie la marque de suivi de connexion (connmark) **100** dans le champ d'en-tête IP du paquet.
- **action mirrored egress redirect dev ifb4eth0** le message suivant s'affiche : **action mirrored** redirige les paquets reçus vers l'interface de destination **ifb4eth0**.

- e. Ajouter un classful **qdisc** à l'interface **ifb4eth0**:

```
# tc qdisc add dev ifb4eth0 root handle 1: htb default 1000
```

Cette commande définit le numéro majeur **1** à la racine **qdisc** et utilise le jeton de hiérarchie bucket **htb** classful **qdisc** of minor-id **1000**.

- f. Limiter le trafic sur **ifb4eth0** à 1 Mbit/s avec une limite supérieure de 2 Mbit/s :

```
# tc class add dev ifb4eth0 parent 1:1 classid 1:100 htb ceil 2mbit rate 1mbit prio 100
```

Cette commande a les paramètres suivants :

- **parent 1:1**: Définit **parent** avec **classid** en tant que **1** et **root** en tant que **1**.
- **classid 1:100** le site **classid** est nommé **1:100** où **1** est le numéro du parent **qdisc** et **100** est le numéro des classes du parent **qdisc**.
- **htb ceil 2mbit** la classe **htb qdisc** autorise la limite supérieure de la bande passante de **2 Mbit/s** en tant que limite de débit de **ceil**.

- g. Appliquer le Stochastic Fairness Queuing (**sfq**) de l'interface sans classe **qdisc** à l'interface **ifb4eth0** avec un intervalle de temps de **60** secondes pour réduire la perturbation de l'algorithme de la file d'attente :

```
# tc qdisc add dev ifb4eth0 parent 1:100 sfq perturb 60
```

-
- h. Ajouter le filtre Firewall mark (**fw**) à l'appareil **ifb4eth0**:

```
# tc filter add dev ifb4eth0 parent 1:0 protocol ip prio 100 handle 100 fw classid 1:100
```

- i. Restaurer la méta-marque de paquet à partir de la marque de connexion (**CONNMARK**) :

```
# nft add rule ip mangle PREROUTING counter meta mark set ct mark
```

Dans cette commande, l'utilitaire **nft** dispose de la table **mangle** avec la spécification de la règle de chaîne **PREROUTING** qui modifie les paquets entrants avant le routage pour remplacer la marque du paquet par **CONNMARK**.

- j. Pour créer une table et ajouter une règle de chaîne s'il n'existe pas de table et de chaîne **nft**:

```
# nft add table ip mangle
# nft add chain ip mangle PREROUTING {type filter hook prerouting priority mangle \;}
```

- k. Définir la marque méta sur les paquets **tcp** qui sont reçus à l'adresse de destination spécifiée **192.0.2.3**:

```
# nft add rule ip mangle PREROUTING ip daddr 192.0.2.3 counter meta mark set 0x64
```

- l. Enregistrer la marque de paquet dans la marque de connexion :

```
# nft add rule ip mangle PREROUTING counter ct mark set mark
```

- m. Exécutez **iperf3** en tant que serveur sur un système et écoutez en tant que serveur **-s** qui attend la réponse de la connexion du client :

```
# iperf3 -s
```

2. Exécutez **iperf3** sur un autre système en tant que client et connectez-vous au serveur qui écoute l'adresse IP **192.0.2.3** pour obtenir l'horodatage des requêtes et des réponses HTTP périodiques :

```
# iperf3 -c 192.0.2.3 -t TCP_STREAM | tee rate
```

3. Sur le serveur, appuyez sur **Ctrl+C**.
4. L'utilitaire **iperf3** du serveur affiche les résultats :

```
Accepted connection from 192.0.2.4, port 52128
[5] local 192.0.2.3 port 5201 connected to 192.0.2.4 port 52130
[ID] Interval      Transfer Bitrate
[5] 0.00-1.00 sec 119 KBytes 973 Kbits/sec
[5] 1.00-2.00 sec 116 KBytes 950 Kbits/sec
....
-----
[ID] Interval      Transfer Bitrate
[5] 0.00-14.81 sec 1.51 MBytes 853 Kbits/sec receiver

iperf3: interrupt - the server has terminated
```

5. Appuyer sur **Ctrl+C** pour mettre fin à **iperf3** sur le serveur :

```

Connecting to host 192.0.2.3, port 5201
[5] local 192.0.2.4 port 52130 connected to 192.0.2.3 port 5201
[ID] Interval      Transfer Bitrate   Retr Cwnd
[5] 0.00-1.00 sec  481 KBytes 3.94 Mb/s 0 76.4 KBytes
[5] 1.00-2.00 sec  223 KBytes 1.83 Mb/s 0 82.0 KBytes
....
-----
[ID] Interval      Transfer Bitrate   Retr
[5] 0.00-14.00 sec  3.92 MBytes 2.35 Mb/s 32 sender
[5] 0.00-14.00 sec  0.00 Bytes 0.00 b/s receiver

iperf3: error - the server has terminated

```

Vérification

1. Affichez les statistiques sur le nombre de paquets des classes **htb** et **sfq** sur l'interface **ifb4eth0**:

```

# tc -s qdisc show dev ifb4eth0

qdisc htb 1: root
....
Sent 26611455 bytes 3054 pkt (dropped 76, overlimits 4887 requeues 0)
....
qdisc sfq 8001: parent
....
Sent 26535030 bytes 2296 pkt (dropped 76, overlimits 0 requeues 0)
....

```

2. Affichez les statistiques du nombre de paquets pour les actions **mirred** et **ctinfo**:

```

# tc -s filter show dev enp1s0 ingress

filter parent ffff: protocol ip pref 49152 u32 chain 0
filter parent ffff: protocol ip pref 49152 u32 chain 0 fh 800: ht divisor 1
filter parent ffff: protocol ip pref 49152 u32 chain 0 fh 800::800 order 2048 key ht 800 bkt 0
terminal flowid not_in_hw (rule hit 8075 success 8075)
  match 00000000/00000000 at 0 (success 8075 )
  action order 1: ctinfo zone 0 pipe
    index 1 ref 1 bind 1 cpmark 0x00000064 installed 3105 sec firstused 3105 sec DSCP set
0 error 0
  CPMARK set 7712
  Action statistics:
  Sent 25891504 bytes 3137 pkt (dropped 0, overlimits 0 requeues 0)
  backlog 0b 0p requeues 0

  action order 2: mirred (Egress Redirect to device ifb4eth0) stolen
    index 1 ref 1 bind 1 installed 3105 sec firstused 3105 sec
  Action statistics:
  Sent 25891504 bytes 3137 pkt (dropped 0, overlimits 61 requeues 0)
  backlog 0b 0p requeues 0

```

3. Affiche les statistiques de **htb** rate-limiter et sa configuration :

■

```
# tc -s class show dev ifb4eth0
```

```
class htb 1:100 root leaf 8001: prio 7 rate 1Mbit ceil 2Mbit burst 1600b cburst 1600b
Sent 26541716 bytes 2373 pkt (dropped 61, overlimits 4887 requeues 0)
backlog 0b 0p requeues 0
lended: 7248 borrowed: 0 giants: 0
tokens: 187250 ctokens: 93625
```

Ressources supplémentaires

- **tc(8)** page de manuel
- **tc-ctinfo(8)** page de manuel
- **nft(8)** page de manuel

27.8. DISQUES DURS DISPONIBLES DANS RHEL

Chaque site **qdisc** aborde des problèmes uniques liés à la mise en réseau. Voici la liste des **qdiscs** disponibles dans RHEL. Vous pouvez utiliser n'importe lequel des **qdisc** suivants pour modeler le trafic réseau en fonction de vos besoins en matière de réseau.

Tableau 27.1. Ordonnanceurs disponibles dans RHEL

qdisc nom	Inclus dans	Prise en charge du délestage
Mode de transfert asynchrone (ATM)	kernel-modules-extra	
Mise en file d'attente par classe	kernel-modules-extra	
Formateur basé sur le crédit	kernel-modules-extra	Oui
CHOOSE and Keep pour les flux réactifs, CHOOSE and Kill pour les flux non réactifs (CHOKe)	kernel-modules-extra	
Délai contrôlé (CoDel)	kernel-core	
Tour de table des déficits (DRR)	kernel-modules-extra	
Marqueur de services différenciés (DSMARK)	kernel-modules-extra	
Sélection de transmission améliorée (ETS)	kernel-modules-extra	Oui
File d'attente équitable (FQ)	kernel-core	
Délai contrôlé de mise en file d'attente équitable (FQ_CODEL)	kernel-core	

qdisc nom	Inclus dans	Prise en charge du délestage
Détection précoce aléatoire généralisée (GRED)	kernel-modules-extra	
Courbe hiérarchique de service équitable (CSFE)	kernel-core	
Filtre pour les gros frappeurs (HHF)	kernel-core	
Panier de jetons de hiérarchie (HTB)	kernel-core	
INGRESS	kernel-core	Oui
Priorité aux files d'attente multiples (MQPRIO)	kernel-modules-extra	Oui
File d'attente multiple (MULTIQ)	kernel-modules-extra	Oui
Emulateur de réseau (NETEM)	kernel-modules-extra	
Contrôleur proportionnel intégral amélioré (PIE)	kernel-core	
PLUG	kernel-core	
File d'attente rapide et équitable (QFQ)	kernel-modules-extra	
Détection précoce aléatoire (RED)	kernel-modules-extra	Oui
Stochastique Fair Blue (SFB)	kernel-modules-extra	
File d'attente stochastique et équitable (SFQ)	kernel-core	
Filtre à jetons (TBF)	kernel-core	Oui
Égaliseur de liens trivial (TEQL)	kernel-modules-extra	



IMPORTANT

Le délestage de **qdisc** nécessite la prise en charge du matériel et du pilote par la carte d'interface réseau.

Ressources supplémentaires

- **tc(8)** page de manuel

CHAPITRE 28. PREMIERS PAS AVEC TCP MULTIPATH

Le protocole de contrôle de transmission (TCP) assure une livraison fiable des données via l'internet et ajuste automatiquement sa bande passante en fonction de la charge du réseau. Le protocole TCP à chemins multiples (MPTCP) est une extension du protocole TCP original (à chemin unique). MPTCP permet à une connexion de transport de fonctionner simultanément sur plusieurs chemins, et apporte la redondance de la connexion réseau aux terminaux des utilisateurs.

28.1. COMPRENDRE MPTCP

Le protocole TCP Multipath (MPTCP) permet l'utilisation simultanée de plusieurs chemins entre les points d'extrémité de la connexion. La conception du protocole améliore la stabilité de la connexion et apporte également d'autres avantages par rapport au TCP à chemin unique.



NOTE

Dans la terminologie MPTCP, les liens sont considérés comme des chemins.

Voici quelques-uns des avantages de l'utilisation de MPTCP :

- Il permet à une connexion d'utiliser simultanément plusieurs interfaces réseau.
- Si une connexion est liée à la vitesse d'un lien, l'utilisation de plusieurs liens peut augmenter le débit de la connexion. Il est à noter que si la connexion est liée à une unité centrale, l'utilisation de plusieurs liens entraîne un ralentissement de la connexion.
- Il augmente la résistance aux défaillances des liaisons.

Pour plus de détails sur MPTCP, nous vous recommandons vivement de consulter le site *Additional resources*.

Ressources supplémentaires

- [Comprendre le TCP à trajets multiples : haute disponibilité pour les points d'extrémité et l'autoroute du réseau de l'avenir](#)
- [RFC8684 : Extensions TCP pour l'exploitation de chemins multiples avec des adresses multiples](#)

28.2. PRÉPARATION DE RHEL À LA PRISE EN CHARGE DE MPTCP

Par défaut, la prise en charge de MPTCP est désactivée dans RHEL. Activez MPTCP pour que les applications qui prennent en charge cette fonctionnalité puissent l'utiliser. En outre, vous devez configurer les applications de l'espace utilisateur pour forcer l'utilisation des sockets MPTCP si ces applications ont des sockets TCP par défaut.

Conditions préalables

Les paquets suivants sont installés :

- **iperf3**
- **mptcpd**
- **systemtap**

Procédure

1. Activer les sockets MPTCP dans le noyau :

```
# echo "net.mptcp.enabled=1" > /etc/sysctl.d/90-enable-MPTCP.conf
# sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf
```

2. Démarrez le serveur **iperf3** et forcez-le à créer des sockets MPTCP au lieu de sockets TCP :

```
# mptcpize run iperf3 -s

Server listening on 5201
```

3. Connecter le client au serveur et le forcer à créer des sockets MPTCP au lieu de sockets TCP :

```
# mptcpize iperf3 -c 127.0.0.1 -t 3
```

4. Une fois la connexion établie, vérifiez la sortie **ss** pour voir l'état spécifique du sous-flux :

```
# ss -nti '( dport :5201 )'

State Recv-Q Send-Q Local Address:Port Peer Address:Port Process
ESTAB 0 0 127.0.0.1:41842 127.0.0.1:5201
cubic wscale:7,7 rto:205 rtt:4.455/8.878 ato:40 mss:21888 pmtu:65535 rcvmss:536
advms:65483 cwnd:10 bytes_sent:141 bytes_acked:142 bytes_received:4 segs_out:8
segs_in:7 data_segs_out:3 data_segs_in:3 send 393050505bps lastsnd:2813 lastrcv:2772
lastack:2772 pacing_rate 785946640bps delivery_rate 10944000000bps delivered:4
busy:41ms rcv_space:43690 rcv_ssthresh:43690 minrtt:0.008 tcp-ulp-mptcp flags:Mmec
token:0000(id:0)/2ff053ec(id:0) seq:3e2cbea12d7673d4 sfseq:3 ssnoff:ad3d00f4 maplen:2
```

5. Vérifier les compteurs MPTCP :

```
# nstat MPTcp*

#kernel
MPTcpExtMPCapableSYNRX      2          0.0
MPTcpExtMPCapableSYNTAX    2          0.0
MPTcpExtMPCapableSYNACKRX   2          0.0
MPTcpExtMPCapableACKRX      2          0.0
```

Ressources supplémentaires

- **tcp(7)** page de manuel
- **mptcpize(8)** page de manuel

28.3. UTILISATION DE IPROUTE2 POUR CONFIGURER ET ACTIVER TEMPORAIREMENT PLUSIEURS CHEMINS POUR LES APPLICATIONS MPTCP

Chaque connexion MPTCP utilise un seul sous-flux similaire au TCP ordinaire. Pour bénéficier des avantages de MPTCP, spécifiez une limite plus élevée pour le nombre maximum de sous-flux pour chaque connexion MPTCP. Configurez ensuite des points d'extrémité supplémentaires pour créer ces

sous-flux.



IMPORTANT

La configuration de cette procédure ne sera pas maintenue après le redémarrage de votre machine.

Notez que MPTCP ne supporte pas encore les terminaux mixtes IPv6 et IPv4 pour la même socket. Utilisez des terminaux appartenant à la même famille d'adresses.

Conditions préalables

- Le paquet **mptcpd** est installé
- Le paquet **iperf3** est installé
- Paramètres de l'interface réseau du serveur :
 - enp4s0 : **192.0.2.1/24**
 - enp1s0 : **198.51.100.1/24**
- Paramètres de l'interface réseau du client :
 - enp4s0f0 : **192.0.2.2/24**
 - enp4s0f1 : **198.51.100.2/24**

Procédure

1. Configurer le client pour qu'il accepte jusqu'à une adresse distante supplémentaire, fournie par le serveur :

```
# ip mptcp limits set add_addr_accepted 1
```

2. Ajouter l'adresse IP **198.51.100.1** comme nouveau point d'extrémité MPTCP sur le serveur :

```
# ip mptcp endpoint add 198.51.100.1 dev enp1s0 signal
```

L'option **signal** garantit que le paquet **ADD_ADDR** est envoyé après la poignée de main à trois voies.

3. Démarrez le serveur **iperf3** et forcez-le à créer des sockets MPTCP au lieu de sockets TCP :

```
# mptcpize run iperf3 -s
```

```
Server listening on 5201
```

4. Connecter le client au serveur et le forcer à créer des sockets MPTCP au lieu de sockets TCP :

```
# mptcpize iperf3 -c 192.0.2.1 -t 3
```

Vérification

1. Vérifiez que la connexion est établie :

```
# ss -nti '( sport :5201 )'
```

2. Vérifiez la connexion et la limite de l'adresse IP :

```
# ip mptcp limit show
```

3. Vérifier le nouveau point d'extrémité ajouté :

```
# ip mptcp endpoint show
```

4. Vérifiez les compteurs MPTCP en utilisant la commande **nstat MPTcp*** sur un serveur :

```
# nstat MPTcp*

#kernel
MPTcpExtMPCapableSYNRX      2          0.0
MPTcpExtMPCapableACKRX      2          0.0
MPTcpExtMPJoinSynRx         2          0.0
MPTcpExtMPJoinAckRx         2          0.0
MPTcpExtEchoAdd             2          0.0
```

Ressources supplémentaires

- **ip-mptcp(8)** page de manuel
- **mptcpize(8)** page de manuel

28.4. CONFIGURATION PERMANENTE DE CHEMINS MULTIPLES POUR LES APPLICATIONS MPTCP

Vous pouvez configurer MultiPath TCP (MPTCP) à l'aide de la commande **nmcli** pour établir en permanence plusieurs sous-flux entre un système source et un système de destination. Les sous-flux peuvent utiliser différentes ressources, différents itinéraires vers la destination et même différents réseaux. Il peut s'agir d'un réseau Ethernet, d'un réseau cellulaire, d'un réseau wifi, etc. Vous obtenez ainsi des connexions combinées qui augmentent la résilience et le débit du réseau.

Dans notre exemple, le serveur utilise les interfaces réseau suivantes :

- enp4s0 : **192.0.2.1/24**
- enp1s0 : **198.51.100.1/24**
- enp7s0 : **192.0.2.3/24**

Dans notre exemple, le client utilise les interfaces réseau suivantes :

- enp4s0f0 : **192.0.2.2/24**
- enp4s0f1 : **198.51.100.2/24**
- enp6s0 : **192.0.2.5/24**

Conditions préalables

- Vous avez configuré la passerelle par défaut sur les interfaces concernées.

Procédure

1. Activer les sockets MPTCP dans le noyau :

```
# echo "net.mptcp.enabled=1" > /etc/sysctl.d/90-enable-MPTCP.conf
# sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf
```

2. Facultatif : La valeur par défaut du noyau RHEL pour la limite de sous-débit est de 2. Si vous avez besoin de plus :

- a. Créez le fichier `/etc/systemd/system/set_mptcp_limit.service` avec le contenu suivant :

```
[Unit]
Description=Set MPTCP subflow limit to 3
After=network.target

[Service]
ExecStart=ip mptcp limits set subflows 3
Type=oneshot

[Install]
WantedBy=multi-user.target
```

L'unité **oneshot** exécute la commande **ip mptcp limits set subflows 3** une fois que votre réseau (**network.target**) est opérationnel au cours de chaque processus de démarrage.

La commande **ip mptcp limits set subflows 3** définit le nombre maximum de sous-flux *additional* pour chaque connexion, soit 4 au total. Il est possible d'ajouter au maximum 3 sous-flux supplémentaires.

- b. Activer le service **set_mptcp_limit**:

```
# systemctl enable --now set_mptcp_limit
```

3. Activez MPTCP sur tous les profils de connexion que vous souhaitez utiliser pour l'agrégation de connexions :

```
# nmcli connection modify <profile_name> connection.mptcp-flags
signal,subflow,also-without-default-route
```

Le paramètre **connection.mptcp-flags** configure les points d'extrémité MPTCP et les drapeaux d'adresse IP. Si MPTCP est activé dans un profil de connexion NetworkManager, le paramètre configure les adresses IP de l'interface réseau concernée en tant que points d'extrémité MPTCP.

Par défaut, NetworkManager n'ajoute pas de drapeaux MPTCP aux adresses IP s'il n'y a pas de passerelle par défaut. Si vous souhaitez contourner cette vérification, vous devez également utiliser le drapeau **also-without-default-route**.

Vérification

1. Vérifiez que vous avez activé le paramètre noyau MPTCP :

```
# sysctl net.mptcp.enabled
net.mptcp.enabled = 1
```

2. Vérifiez que vous avez correctement défini la limite de sous-débit, au cas où la valeur par défaut n'était pas suffisante :

```
# ip mptcp limit show
add_addr_accepted 2 subflows 3
```

3. Vérifiez que vous avez configuré correctement le paramètre MPTCP par adresse :

```
# ip mptcp endpoint show
192.0.2.1 id 1 subflow dev enp4s0
198.51.100.1 id 2 subflow dev enp1s0
192.0.2.3 id 3 subflow dev enp7s0
192.0.2.4 id 4 subflow dev enp3s0
...
```

Ressources supplémentaires

- [nm-settings-nmcli\(5\)](#)
- [ip-mptcp\(8\)](#)
- [Section 28.1, « Comprendre MPTCP »](#)
- [Comprendre le TCP à trajets multiples : haute disponibilité pour les points d'extrémité et l'autoroute du réseau de l'avenir](#)
- [RFC8684 : Extensions TCP pour l'exploitation de chemins multiples avec des adresses multiples](#)
- [Utiliser le protocole TCP Multipath pour mieux survivre aux pannes et augmenter la bande passante](#)

28.5. SURVEILLANCE DES SOUS-FLUX MPTCP

Le cycle de vie d'une socket TCP multipath (MPTCP) peut être complexe : la socket MPTCP principale est créée, le chemin MPTCP est validé, un ou plusieurs sous-flux sont créés et éventuellement supprimés. Enfin, la socket MPTCP est terminée.

Le protocole MPTCP permet de surveiller les événements spécifiques au MPTCP liés à la création et à la suppression de sockets et de sous-flux, à l'aide de l'utilitaire **ip** fourni par le paquetage **iproute**. Cet utilitaire utilise l'interface **netlink** pour surveiller les événements MPTCP.

Cette procédure montre comment surveiller les événements MPTCP. Pour cela, elle simule une application serveur MPTCP, et un client se connecte à ce service. Les clients impliqués dans cet exemple utilisent les interfaces et les adresses IP suivantes :

- Serveur : **192.0.2.1**
- Client (connexion Ethernet) : **192.0.2.2**
- Client (connexion WiFi) : **192.0.2.3**

Pour simplifier cet exemple, toutes les interfaces se trouvent dans le même sous-réseau. Ce n'est pas une obligation. Cependant, il est important que le routage ait été configuré correctement et que le client puisse atteindre le serveur via les deux interfaces.

Conditions préalables

- Un client RHEL avec deux interfaces réseau, tel qu'un ordinateur portable avec Ethernet et WiFi
- Le client peut se connecter au serveur via les deux interfaces
- Un serveur RHEL
- Le client et le serveur fonctionnent sous RHEL 9.0 ou une version ultérieure
- Vous avez installé le paquetage **mptcpd** sur le client et le serveur

Procédure

1. Fixez les limites de sous-flux supplémentaires par connexion à **1**, tant pour le client que pour le serveur :

```
# ip mptcp limits set add_addr_accepted 0 subflows 1
```

2. Sur le serveur, pour simuler une application serveur MPTCP, démarrez **netcat (nc)** en mode écoute avec des sockets MPTCP appliqués au lieu de sockets TCP :

```
# mptcpize run nc -l -k -p 12345
```

L'option **-k** fait en sorte que **nc** ne ferme pas l'écouteur après la première connexion acceptée. Ceci est nécessaire pour démontrer la surveillance des sous-flux.

3. Sur le client :
 - a. Identifier l'interface avec la métrique la plus basse :

```
# ip -4 route
192.0.2.0/24 dev enp1s0 proto kernel scope link src 192.0.2.2 metric 100
192.0.2.0/24 dev wlp1s0 proto kernel scope link src 192.0.2.3 metric 600
```

L'interface **enp1s0** a une métrique inférieure à celle de **wlp1s0**. Par conséquent, RHEL utilise **enp1s0** par défaut.

- b. Sur le premier terminal, démarrez la surveillance :

```
# ip mptcp monitor
```

- c. Sur le second terminal, démarrez une connexion MPTCP avec le serveur :

```
# mptcpize run nc 192.0.2.1 12345
```

RHEL utilise l'interface **enp1s0** et son adresse IP associée comme source pour cette connexion.

Sur le terminal de surveillance, la commande **ip mptcp monitor** est maintenant enregistrée :

■

```
[ CREATED] token=63c070d2 remid=0 locid=0 saddr4=192.0.2.2 daddr4=192.0.2.1
sport=36444 dport=12345
```

Le jeton identifie le socket MPTCP sous la forme d'un identifiant unique et permet ensuite de corréliser les événements MPTCP sur le même socket.

- d. Sur le terminal avec la connexion **nc** en cours au serveur, appuyez sur **Entrée**. Ce premier paquet de données établit complètement la connexion. Notez que tant qu'aucune donnée n'a été envoyée, la connexion n'est pas établie.

Sur le terminal de surveillance, **ip mptcp monitor** est maintenant enregistré :

```
[ ESTABLISHED] token=63c070d2 remid=0 locid=0 saddr4=192.0.2.2
daddr4=192.0.2.1 sport=36444 dport=12345
```

- e. Optionnel : Affichez les connexions au port **12345** du serveur :

```
# ss -taunp | grep ":12345"
tcp ESTAB 0 0      192.0.2.2:36444 192.0.2.1:12345
```

À ce stade, une seule connexion au serveur a été établie.

- f. Sur un troisième terminal, créez un autre point de terminaison :

```
# ip mptcp endpoint add dev wlp1s0 192.0.2.3 subflow
```

Cette commande définit le nom et l'adresse IP de l'interface WiFi du client dans cette commande.

Sur le terminal de surveillance, **ip mptcp monitor** est maintenant enregistré :

```
[SF_ESTABLISHED] token=63c070d2 remid=0 locid=2 saddr4=192.0.2.3
daddr4=192.0.2.1 sport=53345 dport=12345 backup=0 ifindex=3
```

Le champ **locid** affiche l'ID de l'adresse locale du nouveau sous-flux et identifie ce sous-flux même si la connexion utilise la traduction d'adresse réseau (NAT). Le champ **saddr4** correspond à l'adresse IP de l'extrémité de la commande **ip mptcp endpoint add**.

- g. Optionnel : Affichez les connexions au port **12345** du serveur :

```
# ss -taunp | grep ":12345"
tcp ESTAB 0 0      192.0.2.2:36444 192.0.2.1:12345
tcp ESTAB 0 0 192.0.2.3%wlp1s0:53345 192.0.2.1:12345
```

La commande affiche maintenant deux connexions :

- La connexion avec l'adresse source **192.0.2.2** correspond au premier sous-flux MPTCP que vous avez établi précédemment.
- La connexion du sous-flux sur l'interface **wlp1s0** avec l'adresse source **192.0.2.3**.

- h. Sur le troisième terminal, supprimez le point d'arrivée :

```
# ip mptcp endpoint delete id 2
```


Utilisez l'ID du champ **locid** de la sortie **ip mptcp monitor**, ou récupérez l'ID du point de terminaison à l'aide de la commande **ip mptcp endpoint show**.

Sur le terminal de surveillance, **ip mptcp monitor** est maintenant enregistré :

```
[ SF_CLOSED] token=63c070d2 remid=0 locid=2 saddr4=192.0.2.3 daddr4=192.0.2.1
sport=53345 dport=12345 backup=0 ifindex=3
```

- i. Sur le premier terminal avec le client **nc**, appuyez sur **Ctrl+C** pour mettre fin à la session. Sur le terminal de surveillance, **ip mptcp monitor** est maintenant enregistré :

```
[ CLOSED] token=63c070d2
```

Ressources supplémentaires

- **ip-mptcp(1)** page de manuel
- [Comment NetworkManager gère plusieurs passerelles par défaut](#)

28.6. DÉSACTIVATION DE TCP MULTIPATH DANS LE NOYAU

Vous pouvez désactiver explicitement l'option MPTCP dans le noyau.

Procédure

- Désactiver l'option **mptcp.enabled**.

```
# echo "net.mptcp.enabled=0" > /etc/sysctl.d/90-enable-MPTCP.conf
# sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf
```

Vérification

- Vérifiez si **mptcp.enabled** est désactivé dans le noyau.

```
# sysctl -a | grep mptcp.enabled
net.mptcp.enabled = 0
```

CHAPITRE 29. GESTION DU SERVICE MPTCPD

Cette section décrit la gestion de base du service **mptcpd**. Le paquetage **mptcpd** fournit l'outil **mptcpize**, qui active le protocole **mptcp** dans l'environnement **TCP**.

29.1. CONFIGURATION DE MPTCPD

Le service **mptcpd** est un composant du protocole **mptcp** qui fournit un instrument permettant de configurer les points de terminaison **mptcp**. Le service **mptcpd** crée par défaut un point d'extrémité de sous-flux pour chaque adresse. La liste des points de terminaison est mise à jour dynamiquement en fonction de la modification des adresses IP sur l'hôte en cours d'exécution. Le service **mptcpd** crée automatiquement la liste des points d'extrémité. Il permet d'activer des chemins multiples au lieu d'utiliser l'utilitaire **ip**.

Conditions préalables

- Le paquet **mptcpd** est installé

Procédure

1. Activez l'option **mptcp.enabled** dans le noyau avec la commande suivante :

```
# echo "net.mptcp.enabled=1" > /etc/sysctl.d/90-enable-MPTCP.conf
# sysctl -p /etc/sysctl.d/90-enable-MPTCP.conf
```

2. Démarrez le service **mptcpd**:
systemctl start mptcp.service
3. Vérifier la création du point d'accès :
ip mptcp endpoint
4. Pour arrêter le service **mptcpd**, utilisez la commande suivante :
systemctl stop mptcp.service
5. Pour configurer manuellement le service **mptcpd**, modifiez le fichier de configuration de **/etc/mptcpd/mptcpd.conf**.

Notez que le point de terminaison créé par le service **mptcpd** dure jusqu'à l'arrêt de l'hôte.

Ressources supplémentaires

- **mptcpd(8)** page de manuel.

29.2. GESTION DES APPLICATIONS AVEC L'OUTIL MPTCPIZE

Utiliser l'outil **mptcpize** pour gérer les applications et les services.

L'instruction ci-dessous montre comment utiliser l'outil **mptcpize** pour gérer les applications dans l'environnement **TCP**.

En supposant que vous ayez besoin d'exécuter l'utilitaire **iperf3** avec le socket **MPTCP** activé. Vous pouvez atteindre cet objectif en suivant la procédure ci-dessous.

Conditions préalables

- Le paquet **mptcpd** est installé
- Le paquet **iperf3** est installé

Procédure

- Démarrer l'utilitaire **iperf3** avec les sockets **MPTCP** activés :
mptcpize run iperf3 -s &

29.3. ACTIVATION DES SOCKETS MPTCP POUR UN SERVICE À L'AIDE DE L'UTILITAIRE MPTCPIZE

Les commandes suivantes vous indiquent comment gérer les services à l'aide de l'outil **mptcpize**. Vous pouvez activer ou désactiver la prise **mptcp** pour un service.

Vous devez gérer la socket **mptcp** pour le service **nginx**. Vous pouvez atteindre cet objectif en suivant la procédure ci-dessous.

Conditions préalables

- Le paquet **mptcpd** est installé
- Le paquet **nginx** est installé

Procédure

1. Activer les sockets **MPTCP** pour un service :

```
# mptcpize enable nginx
```

2. Désactiver les sockets **MPTCP** pour un service :

```
# mptcpize disable nginx
```

3. Redémarrez le service pour que les modifications soient prises en compte :

```
# systemctl restart nginx
```

CHAPITRE 30. CONFIGURATION DE L'ORDRE DES SERVEURS DNS

La plupart des applications utilisent la fonction `getaddrinfo()` de la bibliothèque `glibc` pour résoudre les requêtes DNS. Par défaut, `glibc` envoie toutes les requêtes DNS au premier serveur DNS spécifié dans le fichier `/etc/resolv.conf`. Si ce serveur ne répond pas, RHEL utilise le serveur suivant dans ce fichier. NetworkManager vous permet d'influencer l'ordre des serveurs DNS dans `etc/resolv.conf`.

30.1. COMMENT NETWORKMANAGER ORDONNE LES SERVEURS DNS DANS /ETC/RESOLV.CONF

NetworkManager ordonne les serveurs DNS dans le fichier `/etc/resolv.conf` en fonction des règles suivantes :

- S'il n'existe qu'un seul profil de connexion, NetworkManager utilise l'ordre des serveurs DNS IPv4 et IPv6 spécifiés dans cette connexion.
- Si plusieurs profils de connexion sont activés, NetworkManager ordonne les serveurs DNS en fonction d'une valeur de priorité DNS. Si vous définissez des priorités DNS, le comportement de NetworkManager dépend de la valeur définie dans le paramètre `dns`. Vous pouvez définir ce paramètre dans la section `[main]` du fichier `/etc/NetworkManager/NetworkManager.conf`:

- `dns=default` ou si le paramètre `dns` n'est pas défini :
NetworkManager commande les serveurs DNS de différentes connexions en fonction des paramètres `ipv4.dns-priority` et `ipv6.dns-priority` de chaque connexion.

Si vous ne définissez aucune valeur ou si vous définissez `ipv4.dns-priority` et `ipv6.dns-priority` sur `0`, NetworkManager utilise la valeur globale par défaut. Voir [Valeurs par défaut des paramètres de priorité DNS](#).

- `dns=dnsmasq` ou `dns=systemd-resolved`:
Lorsque vous utilisez l'un de ces paramètres, NetworkManager définit soit `127.0.0.1` pour `dnsmasq`, soit `127.0.0.53` pour `nameserver` dans le fichier `/etc/resolv.conf`.

Les services `dnsmasq` et `systemd-resolved` transmettent les requêtes relatives au domaine de recherche défini dans une connexion NetworkManager au serveur DNS spécifié dans cette connexion, et transmettent les requêtes relatives à d'autres domaines à la connexion avec la route par défaut. Lorsque plusieurs connexions ont le même domaine de recherche, `dnsmasq` et `systemd-resolved` transmettent les requêtes pour ce domaine au serveur DNS défini dans la connexion avec la valeur de priorité la plus basse.

Valeurs par défaut des paramètres de priorité DNS

NetworkManager utilise les valeurs par défaut suivantes pour les connexions :

- **50** pour les connexions VPN
- **100** pour les autres connexions

Valeurs de priorité DNS valables :

Vous pouvez définir les paramètres par défaut globaux et les paramètres spécifiques à la connexion `ipv4.dns-priority` et `ipv6.dns-priority` à une valeur comprise entre `-2147483647` et `2147483647`.

- Une valeur inférieure a une priorité plus élevée.
- Les valeurs négatives ont pour effet particulier d'exclure d'autres configurations ayant une

valeur supérieure. Par exemple, s'il existe au moins une connexion avec une valeur de priorité négative, NetworkManager utilise uniquement les serveurs DNS spécifiés dans le profil de connexion avec la priorité la plus basse.

- Si plusieurs connexions ont la même priorité DNS, NetworkManager donne la priorité aux DNS dans l'ordre suivant :
 - a. Connexions VPN
 - b. Connexion avec une route par défaut active. La route par défaut active est la route par défaut dont le métrique est le plus bas.

Ressources supplémentaires

- **nm-settings(5)** page de manuel
- [Utiliser des serveurs DNS différents pour des domaines différents](#)

30.2. DÉFINITION D'UNE VALEUR DE PRIORITÉ DU SERVEUR DNS PAR DÉFAUT POUR L'ENSEMBLE DU NETWORKMANAGER

NetworkManager utilise les valeurs par défaut de priorité DNS suivantes pour les connexions :

- **50** pour les connexions VPN
- **100** pour les autres connexions

Vous pouvez remplacer ces valeurs par défaut à l'échelle du système par une valeur par défaut personnalisée pour les connexions IPv4 et IPv6.

Procédure

1. Modifiez le fichier `/etc/NetworkManager/NetworkManager.conf`:

- a. Ajouter la section **[connection]**, si elle n'existe pas :

```
[connection]
```

- b. Ajoutez les valeurs par défaut personnalisées à la section **[connection]**. Par exemple, pour définir la nouvelle valeur par défaut pour IPv4 et IPv6 à **200**, ajoutez :

```
ipv4.dns-priority=200
ipv6.dns-priority=200
```

Vous pouvez définir les paramètres sur une valeur comprise entre **-2147483647** et **2147483647**. Notez que la valeur **0** active les valeurs par défaut intégrées (**50** pour les connexions VPN et **100** pour les autres connexions).

2. Rechargez le service **NetworkManager**:

```
# systemctl reload NetworkManager
```

Ressources supplémentaires

- **NetworkManager.conf(5)** page de manuel

30.3. DÉFINITION DE LA PRIORITÉ DNS D'UNE CONNEXION NETWORKMANAGER

Si vous souhaitez un ordre spécifique des serveurs DNS, vous pouvez définir des valeurs de priorité dans les profils de connexion. NetworkManager utilise ces valeurs pour ordonner les serveurs lorsque le service crée ou met à jour le fichier **/etc/resolv.conf**.

Notez que la définition des priorités DNS n'a de sens que si vous avez plusieurs connexions avec différents serveurs DNS configurés. Si vous n'avez qu'une seule connexion avec plusieurs serveurs DNS configurés, définissez manuellement les serveurs DNS dans l'ordre préféré dans le profil de connexion.

Conditions préalables

- Plusieurs connexions NetworkManager sont configurées sur le système.
- Le système n'a pas de paramètre **dns** défini dans le fichier **/etc/NetworkManager/NetworkManager.conf** ou le paramètre est défini sur **default**.

Procédure

1. En option, afficher les connexions disponibles :

```
# nmcli connection show
NAME          UUID                                TYPE  DEVICE
Example_con_1 d17ee488-4665-4de2-b28a-48befab0cd43 ethernet enp1s0
Example_con_2 916e4f67-7145-3ffa-9f7b-e7cada8f6bf7 ethernet enp7s0
...
```

2. Définissez les paramètres **ipv4.dns-priority** et **ipv6.dns-priority**. Par exemple, pour définir les deux paramètres sur **10** pour la connexion **Example_con_1**:

```
# nmcli connection modify Example_con_1 ipv4.dns-priority 10 ipv6.dns-priority 10
```

3. Il est possible de répéter l'étape précédente pour d'autres connexions.
4. Réactivez la connexion que vous avez mise à jour :

```
# nmcli connection up Example_con_1
```

Vérification

- Affichez le contenu du fichier **/etc/resolv.conf** pour vérifier que l'ordre des serveurs DNS est correct :

```
# cat /etc/resolv.conf
```

CHAPITRE 31. UTILISATION DE NETWORKMANAGER POUR DÉSACTIVER IPV6 POUR UNE CONNEXION SPÉCIFIQUE

Sur un système qui utilise NetworkManager pour gérer les interfaces réseau, vous pouvez désactiver le protocole IPv6 si le réseau n'utilise qu'IPv4. Si vous désactivez **IPv6**, NetworkManager définit automatiquement les valeurs **sysctl** correspondantes dans le noyau.



NOTE

Si vous désactivez IPv6 à l'aide des tunables du noyau ou des paramètres d'amorçage du noyau, vous devez tenir compte de la configuration du système. Pour plus d'informations, voir l'article [Comment désactiver ou activer le protocole IPv6 dans RHEL ?](#)

31.1. DÉSACTIVATION D'IPV6 SUR UNE CONNEXION À L'AIDE DE NMCLI

Vous pouvez utiliser l'utilitaire **nmcli** pour désactiver le protocole **IPv6** sur la ligne de commande.

Conditions préalables

- Le système utilise NetworkManager pour gérer les interfaces réseau.

Procédure

1. Optionnellement, afficher la liste des connexions réseau :

```
# nmcli connection show
NAME UUID TYPE DEVICE
Example 7a7e0151-9c18-4e6f-89ee-65bb2d64d365 ethernet enp1s0
...
```

2. Réglez le paramètre **ipv6.method** de la connexion sur **disabled**:

```
# nmcli connection modify Example ipv6.method "disabled"
```

3. Redémarrer la connexion réseau :

```
# nmcli connection up Example
```

Vérification

1. Affiche les paramètres IP de l'appareil :

```
# ip address show enp1s0
2: enp1s0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP
group default qlen 1000
link/ether 52:54:00:6b:74:be brd ff:ff:ff:ff:ff:ff
inet 192.0.2.1/24 brd 192.10.2.255 scope global noprefixroute enp1s0
valid_lft forever preferred_lft forever
```

Si aucune entrée **inet6** n'est affichée, **IPv6** est désactivé sur l'appareil.

2. Vérifiez que le fichier `/proc/sys/net/ipv6/conf/enp1s0/disable_ipv6` contient désormais la valeur **1**:

```
# cat /proc/sys/net/ipv6/conf/enp1s0/disable_ipv6
1
```

La valeur **1** signifie que **IPv6** est désactivé pour l'appareil.

CHAPITRE 32. AUGMENTATION DES TAMPONS DE L'ANNEAU POUR RÉDUIRE UN TAUX ÉLEVÉ DE PERTE DE PAQUETS

Les tampons d'anneau de réception sont partagés entre le pilote du périphérique et le contrôleur d'interface réseau (NIC). La carte attribue un tampon circulaire de transmission (TX) et de réception (RX). Comme son nom l'indique, le ring buffer est un tampon circulaire dans lequel un débordement écrase les données existantes. Il existe deux façons de transférer des données de la carte d'interface réseau au noyau : les interruptions matérielles et les interruptions logicielles, également appelées SoftIRQ.

Le noyau utilise le tampon de l'anneau RX pour stocker les paquets entrants jusqu'à ce qu'ils puissent être traités par le pilote de périphérique. Le pilote de périphérique vide l'anneau RX, généralement à l'aide de SoftIRQ, ce qui place les paquets entrants dans une structure de données du noyau appelée **sk_buff** ou **skb** pour commencer son voyage à travers le noyau et jusqu'à l'application qui possède la prise correspondante.

Le noyau utilise le tampon circulaire TX pour contenir les paquets sortants destinés au réseau. Ces tampons en anneau se trouvent au bas de la pile et constituent un point crucial où des paquets peuvent être abandonnés, ce qui affecte négativement les performances du réseau.

Augmentez la taille des tampons en anneau d'un périphérique Ethernet si le taux de perte de paquets entraîne des pertes de données, des dépassements de délai ou d'autres problèmes pour les applications.

Procédure

1. Affiche les statistiques de chute de paquets de l'interface :

```
# ethtool -S enp1s0
...
rx_queue_0_drops: 97326
rx_queue_1_drops: 63783
...
```

Notez que la sortie de la commande dépend de la carte réseau et du pilote.

Des valeurs élevées dans les compteurs **discard** ou **drop** indiquent que le tampon disponible se remplit plus rapidement que le noyau ne peut traiter les paquets. L'augmentation des tampons de l'anneau peut aider à éviter de telles pertes.

2. Affiche la taille maximale des tampons de l'anneau :

```
# ethtool -g enp1s0
Ring parameters for enp1s0:
Pre-set maximums:
RX:          4096
RX Mini:     0
RX Jumbo:    16320
TX:          4096
Current hardware settings:
RX:          255
RX Mini:     0
RX Jumbo:    0
TX:          255
```

Si les valeurs de la section **Pre-set maximums** sont plus élevées que celles de la section **Current hardware settings**, vous pouvez modifier les paramètres dans les étapes suivantes.

3. Identifier le profil de connexion NetworkManager qui utilise l'interface :

```
# nmcli connection show
NAME                UUID                                TYPE    DEVICE
Example-Connection a5eb6490-cc20-3668-81f8-0314a27f3f75 ethernet enp1s0
```

4. Mettez à jour le profil de connexion et augmentez les tampons de l'anneau :

- Pour augmenter la mémoire tampon de l'anneau RX, entrez :

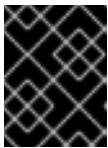
```
# nmcli connection modify Example-Connection ethtool.ring-rx 4096
```

- Pour augmenter la mémoire tampon de l'anneau TX, entrez :

```
# nmcli connection modify Example-Connection ethtool.ring-tx 4096
```

5. Recharger la connexion au NetworkManager :

```
# nmcli connection up Example-Connection
```



IMPORTANT

Selon le pilote utilisé par votre carte d'interface réseau, un changement dans la mémoire tampon de l'anneau peut interrompre brièvement la connexion réseau.

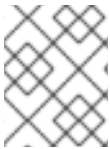
Ressources supplémentaires

- [les commandes ifconfig et ip signalent les chutes de paquets](#)
- [Dois-je m'inquiéter d'un taux d'abandon de paquets de 0,05 % ?](#)
- [ethtool\(8\)](#) page de manuel

CHAPITRE 33. CONFIGURATION DES PARAMÈTRES DE LIAISON 802.3

33.1. COMPRENDRE L'AUTO-NÉGOCIATION

L'auto-négociation est une fonction du protocole Fast Ethernet IEEE 802.3u. Elle cible les ports de l'appareil afin de fournir une performance optimale en termes de vitesse, de mode duplex et de contrôle de flux pour l'échange d'informations sur une liaison. En utilisant le protocole d'auto-négociation, vous obtenez des performances optimales de transfert de données sur l'Ethernet.



NOTE

Pour tirer le meilleur parti de l'auto-négociation, utilisez la même configuration des deux côtés d'une liaison.

33.2. CONFIGURATION DES PARAMÈTRES DE LIAISON 802.3 À L'AIDE DE L'UTILITAIRE NMCLI

Pour configurer les paramètres de liaison 802.3 d'une connexion Ethernet, modifiez les paramètres de configuration suivants :

- **802-3-ethernet.auto-negotiate**
- **802-3-ethernet.speed**
- **802-3-ethernet.duplex**

Procédure

1. Affiche les paramètres actuels de la connexion :

```
# nmcli connection show Example-connection
...
802-3-ethernet.speed: 0
802-3-ethernet.duplex: --
802-3-ethernet.auto-negotiate: no
...
```

Vous pouvez utiliser ces valeurs si vous devez réinitialiser les paramètres en cas de problème.

2. Définir les paramètres de vitesse et de liaison duplex :

```
# nmcli connection modify Example-connection 802-3-ethernet.auto-negotiate yes 802-3-ethernet.speed 10000 802-3-ethernet.duplex full
```

Cette commande active l'auto-négociation et règle la vitesse de la connexion sur **10000** Mbit full duplex.

3. Réactiver la connexion :

```
# nmcli connection up Example-connection
```

Vérification

- Utilisez l'utilitaire **ethtool** pour vérifier les valeurs de l'interface Ethernet **enp1s0**:

```
# ethtool enp1s0  
  
Settings for enp1s0:  
...  
Speed: 10000 Mb/s  
Duplex: Full  
Auto-negotiation: on  
...  
Link detected: yes
```

Ressources supplémentaires

- **nm-settings(5)** page de manuel

CHAPITRE 34. CONFIGURATION DES FONCTIONS DE DÉLESTAGE D'ETHTOOL

Les cartes d'interface réseau peuvent utiliser le moteur de déchargement TCP (TOE) pour décharger le traitement de certaines opérations sur le contrôleur réseau afin d'améliorer le débit du réseau.

34.1. FONCTIONNALITÉS D'OFFLOAD SUPPORTÉES PAR NETWORKMANAGER

Vous pouvez configurer les fonctions de délestage **ethtool** suivantes à l'aide de NetworkManager :

- **ethtool.feature-esp-hw-offload**
- **ethtool.feature-esp-tx-csum-hw-offload**
- **ethtool.feature-fcoe-mtu**
- **ethtool.feature-gro**
- **ethtool.feature-gso**
- **ethtool.feature-highdma**
- **ethtool.feature-hw-tc-offload**
- **ethtool.feature-l2-fwd-offload**
- **ethtool.feature-loopback**
- **ethtool.feature-lro**
- **ethtool.feature-macsec-hw-offload**
- **ethtool.feature-ntuple**
- **ethtool.feature-rx**
- **ethtool.feature-rx-all**
- **ethtool.feature-rx-fcs**
- **ethtool.feature-rx-gro-hw**
- **ethtool.feature-rx-gro-list**
- **ethtool.feature-rx-udp_tunnel-port-offload**
- **ethtool.feature-rx-udp-gro-forwarding**
- **ethtool.feature-rx-vlan-filter**
- **ethtool.feature-rx-vlan-stag-filter**
- **ethtool.feature-rx-vlan-stag-hw-parse**
- **ethtool.feature-rxhash**

- **ethtool.feature-rxvlan**
- **ethtool.feature-sg**
- **ethtool.feature-tls-hw-record**
- **ethtool.feature-tls-hw-rx-offload**
- **ethtool.feature-tls-hw-tx-offload**
- **ethtool.feature-tso**
- **ethtool.feature-tx**
- **ethtool.feature-tx-checksum-fcoe-crc**
- **ethtool.feature-tx-checksum-ip-generic**
- **ethtool.feature-tx-checksum-ipv4**
- **ethtool.feature-tx-checksum-ipv6**
- **ethtool.feature-tx-checksum-sctp**
- **ethtool.feature-tx-esp-segmentation**
- **ethtool.feature-tx-fcoe-segmentation**
- **ethtool.feature-tx-gre-csum-segmentation**
- **ethtool.feature-tx-gre-segmentation**
- **ethtool.feature-tx-gso-list**
- **ethtool.feature-tx-gso-partial**
- **ethtool.feature-tx-gso-robust**
- **ethtool.feature-tx-ipxip4-segmentation**
- **ethtool.feature-tx-ipxip6-segmentation**
- **ethtool.feature-tx-nocache-copy**
- **ethtool.feature-tx-scatter-gather**
- **ethtool.feature-tx-scatter-gather-fraglist**
- **ethtool.feature-tx-sctp-segmentation**
- **ethtool.feature-tx-tcp-ecn-segmentation**
- **ethtool.feature-tx-tcp-mangleid-segmentation**
- **ethtool.feature-tx-tcp-segmentation**
- **ethtool.feature-tx-tcp6-segmentation**

- **ethtool.feature-tx-tunnel-remcsum-segmentation**
- **ethtool.feature-tx-udp-segmentation**
- **ethtool.feature-tx-udp_tnl-csum-segmentation**
- **ethtool.feature-tx-udp_tnl-segmentation**
- **ethtool.feature-tx-vlan-stag-hw-insert**
- **ethtool.feature-txvlan**

Pour plus de détails sur les différentes fonctions de délestage, voir la documentation de l'utilitaire **ethtool** et la documentation du noyau.

34.2. CONFIGURATION D'UNE FONCTION DE DÉLESTAGE ETHTOOL À L'AIDE DE NMCLI

Vous pouvez utiliser NetworkManager pour activer et désactiver les fonctions de délestage de **ethtool** dans un profil de connexion.

Procédure

1. Par exemple, pour activer la fonction de délestage RX et désactiver le délestage TX dans le profil de connexion **enp1s0**, entrez :

```
# nmcli con modify enp1s0 ethtool.feature-rx on ethtool.feature-tx off
```

Cette commande active explicitement la décharge RX et désactive la décharge TX.

2. Pour supprimer la configuration d'une fonction de délestage que vous avez précédemment activée ou désactivée, réglez le paramètre de la fonction sur **ignore**. Par exemple, pour supprimer la configuration de la décharge TX, entrez :

```
# nmcli con modify enp1s0 ethtool.feature-tx ignore
```

3. Réactiver le profil réseau :

```
# nmcli connection up enp1s0
```

Vérification

- Utilisez la commande **ethtool -k** pour afficher les caractéristiques de décharge actuelles d'un périphérique réseau :

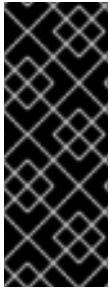
```
# ethtool -k network_device
```

Ressources supplémentaires

- [Fonctionnalités d'offload supportées par NetworkManager](#)

34.3. CONFIGURATION D'UNE FONCTION DE DÉLESTAGE ETHTOOL À L'AIDE DU RÔLE DE SYSTÈME RHEL DE RÉSEAU

Vous pouvez utiliser le rôle de système **network** RHEL pour configurer les fonctionnalités **ethtool** d'une connexion NetworkManager.



IMPORTANT

Lorsque vous exécutez une séquence qui utilise le rôle système **network** RHEL, le rôle système remplace un profil de connexion existant portant le même nom si la valeur des paramètres ne correspond pas à ceux spécifiés dans la séquence. Par conséquent, indiquez toujours la configuration complète du profil de connexion réseau dans la pièce, même si, par exemple, la configuration IP existe déjà. Dans le cas contraire, le rôle rétablit les valeurs par défaut.

Selon qu'il existe déjà ou non, la procédure crée ou met à jour le profil de connexion **enp1s0** avec les paramètres suivants :

- Une adresse statique **IPv4** - **198.51.100.20** avec un masque de sous-réseau **/24**
- Une adresse statique **IPv6** - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut **IPv4** - **198.51.100.254**
- Une passerelle par défaut **IPv6** - **2001:db8:1::fffe**
- Un serveur DNS **IPv4** - **198.51.100.200**
- Un serveur DNS **IPv6** - **2001:db8:1::ffbb**
- Un domaine de recherche DNS - **example.com**
- **ethtool** caractéristiques :
 - Déchargement générique de la réception (GRO) : désactivé
 - Délestage générique de segmentation (GSO) : activé
 - Segmentation du protocole de transmission de contrôle de flux TX (SCTP) : désactivée

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.

Procédure

1. Créez un fichier playbook, par exemple `~/configure-ethernet-device-with-ethtool-features.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
  - name: Configure an Ethernet connection with ethtool features
    include_role:
      name: rhel-system-roles.network

  vars:
    network_connections:
      - name: enp1s0
        type: ethernet
        autoconnect: yes
        ip:
          address:
            - 198.51.100.20/24
            - 2001:db8:1::1/64
          gateway4: 198.51.100.254
          gateway6: 2001:db8:1::ffe
          dns:
            - 198.51.100.200
            - 2001:db8:1::ffbb
          dns_search:
            - example.com
        ethtool:
          features:
            gro: "no"
            gso: "yes"
            tx_sctp_segmentation: "no"
          state: up
```

2. Exécutez le manuel de jeu :

```
# ansible-playbook ~/configure-ethernet-device-with-ethtool-features.yml
```

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

CHAPITRE 35. CONFIGURATION DES PARAMÈTRES ETHTOOL COALESCE

Grâce à la coalescence des interruptions, le système collecte les paquets réseau et génère une seule interruption pour plusieurs paquets. Cela permet d'augmenter la quantité de données envoyées au noyau avec une seule interruption matérielle, ce qui réduit la charge d'interruption et maximise le débit.

35.1. PARAMÈTRES DE COALESCENCE SUPPORTÉS PAR NETWORKMANAGER

Vous pouvez définir les paramètres de coalescence **ethtool** suivants à l'aide de NetworkManager :

- **coalesce-adaptive-rx**
- **coalesce-adaptive-tx**
- **coalesce-pkt-rate-high**
- **coalesce-pkt-rate-low**
- **coalesce-rx-frames**
- **coalesce-rx-frames-high**
- **coalesce-rx-frames-irq**
- **coalesce-rx-frames-low**
- **coalesce-rx-usecs**
- **coalesce-rx-usecs-high**
- **coalesce-rx-usecs-irq**
- **coalesce-rx-usecs-low**
- **coalesce-sample-interval**
- **coalesce-stats-block-usecs**
- **coalesce-tx-frames**
- **coalesce-tx-frames-high**
- **coalesce-tx-frames-irq**
- **coalesce-tx-frames-low**
- **coalesce-tx-usecs**
- **coalesce-tx-usecs-high**
- **coalesce-tx-usecs-irq**
- **coalesce-tx-usecs-low**

35.2. CONFIGURATION DES PARAMÈTRES ETHTOOL COALESCE À L'AIDE DE NMCLI

Vous pouvez utiliser NetworkManager pour définir les paramètres de coalescence de **ethtool** dans les profils de connexion.

Procédure

1. Par exemple, pour définir le nombre maximum de paquets reçus à retarder à **128** dans le profil de connexion **enp1s0**, entrez :

```
# nmcli connection modify enp1s0 ethtool.coalesce-rx-frames 128
```

2. Pour supprimer un paramètre de coalescence, définissez le paramètre sur **ignore**. Par exemple, pour supprimer le paramètre **ethtool.coalesce-rx-frames**, entrez :

```
# nmcli connection modify enp1s0 ethtool.coalesce-rx-frames ignore
```

3. Pour réactiver le profil réseau :

```
# nmcli connection up enp1s0
```

Vérification

1. Utilisez la commande **ethtool -c** pour afficher les caractéristiques de décharge actuelles d'un périphérique réseau :

```
# ethtool -c network_device
```

Ressources supplémentaires

- [Paramètres de coalescence supportés par NetworkManager](#)

35.3. CONFIGURATION DES PARAMÈTRES DE COALESCE D'UN OUTIL ETHTOOL À L'AIDE DU RÔLE DE SYSTÈME RHEL DE RÉSEAU

Vous pouvez utiliser le rôle système **network** RHEL pour configurer les paramètres **ethtool** coalesce d'une connexion NetworkManager.



IMPORTANT

Lorsque vous exécutez une séquence qui utilise le rôle système **network** RHEL, le rôle système remplace un profil de connexion existant portant le même nom si la valeur des paramètres ne correspond pas à ceux spécifiés dans la séquence. Par conséquent, indiquez toujours la configuration complète du profil de connexion réseau dans la pièce, même si, par exemple, la configuration IP existe déjà. Dans le cas contraire, le rôle rétablit les valeurs par défaut.

Selon qu'il existe déjà ou non, la procédure crée ou met à jour le profil de connexion **enp1s0** avec les paramètres suivants :

- Une adresse IPv4 statique - **198.51.100.20** avec un masque de sous-réseau **/24**
- Une adresse IPv6 statique - **2001:db8:1::1** avec un masque de sous-réseau **/64**
- Une passerelle par défaut IPv4 - **198.51.100.254**
- Une passerelle par défaut IPv6 - **2001:db8:1::ffe**
- Un serveur DNS IPv4 - **198.51.100.200**
- Un serveur DNS IPv6 - **2001:db8:1::fbb**
- Un domaine de recherche DNS - **example.com**
- **ethtool** coalescer les paramètres :
 - Trames RX : **128**
 - Trames TX : **128**

Effectuez cette procédure sur le nœud de contrôle Ansible.

Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des séquences sur les nœuds gérés.
- Le compte que vous utilisez pour vous connecter aux nœuds gérés dispose des autorisations **sudo**.
- Les nœuds gérés ou les groupes de nœuds gérés sur lesquels vous souhaitez exécuter cette séquence sont répertoriés dans le fichier d'inventaire Ansible.

Procédure

1. Créez un fichier playbook, par exemple `~/configure-ethernet-device-with-ethtoolcoalesce-settings.yml` avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure an Ethernet connection with ethtool coalesce settings
      include_role:
        name: rhel-system-roles.network

  vars:
    network_connections:
      - name: enp1s0
        type: ethernet
        autoconnect: yes
        ip:
          address:
            - 198.51.100.20/24
            - 2001:db8:1::1/64
```

```
gateway4: 198.51.100.254
gateway6: 2001:db8:1::ffe
dns:
  - 198.51.100.200
  - 2001:db8:1::ffbb
dns_search:
  - example.com
ethtool:
  coalesce:
    rx_frames: 128
    tx_frames: 128
state: up
```

2. Exécutez le manuel de jeu :

```
# ansible-playbook ~/configure-ethernet-device-with-ethtoolcoalesce-settings.yml
```

Ressources supplémentaires

- `/usr/share/ansible/roles/rhel-system-roles.network/README.md` fichier

CHAPITRE 36. UTILISATION DE MACSEC POUR CRYPTER LE TRAFIC DE COUCHE 2 DANS LE MÊME RÉSEAU PHYSIQUE

Vous pouvez utiliser MACsec pour sécuriser la communication entre deux appareils (point à point). Par exemple, si votre succursale est reliée au bureau central par une connexion Metro-Ethernet, vous pouvez configurer MACsec sur les deux hôtes qui relient les bureaux afin d'accroître la sécurité.

Media Access Control Security (MACsec) est un protocole de couche 2 qui sécurise différents types de trafic sur les liaisons Ethernet, notamment :

- protocole de configuration dynamique de l'hôte (DHCP)
- le protocole de résolution d'adresses (ARP)
- Protocole Internet version 4 / 6 (**IPv4 / IPv6**) et
- tout trafic sur IP tel que TCP ou UDP

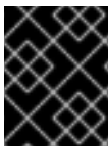
MACsec chiffre et authentifie tout le trafic dans les réseaux locaux, par défaut avec l'algorithme GCM-AES-128, et utilise une clé pré-partagée pour établir la connexion entre les hôtes participants. Si vous souhaitez modifier la clé pré-partagée, vous devez mettre à jour la configuration NM sur tous les hôtes du réseau qui utilisent MACsec.

Une connexion MACsec utilise un périphérique Ethernet, tel qu'une carte réseau Ethernet, un VLAN ou un périphérique tunnel, comme parent. Vous pouvez soit définir une configuration IP uniquement sur le périphérique MACsec pour communiquer avec d'autres hôtes uniquement à l'aide de la connexion cryptée, soit définir également une configuration IP sur le périphérique parent. Dans ce dernier cas, vous pouvez utiliser le dispositif parent pour communiquer avec d'autres hôtes en utilisant une connexion non chiffrée et le dispositif MACsec pour les connexions chiffrées.

MACsec ne nécessite pas de matériel particulier. Par exemple, vous pouvez utiliser n'importe quel commutateur, sauf si vous souhaitez crypter le trafic uniquement entre un hôte et un commutateur. Dans ce cas, le commutateur doit également prendre en charge MACsec.

En d'autres termes, il existe deux méthodes courantes pour configurer MACsec ;

- d'hôte à hôte et
- hôte à basculer puis basculer vers d'autres hôtes



IMPORTANT

Vous ne pouvez utiliser MACsec qu'entre des hôtes qui se trouvent dans le même réseau local (physique ou virtuel).

36.1. CONFIGURATION D'UNE CONNEXION MACSEC À L'AIDE DE NMCLI

Vous pouvez configurer les interfaces Ethernet pour utiliser MACsec à l'aide de l'utilitaire **nmcli**. Par exemple, vous pouvez créer une connexion MACsec entre deux hôtes connectés par Ethernet.

Procédure

1. Sur le premier hôte sur lequel vous configurez MACsec :

- Créez la clé d'association de connectivité (CAK) et le nom de la clé d'association de connectivité (CKN) pour la clé pré-partagée :
 - a. Créer un CAK hexadécimal de 16 octets :

```
# dd if=/dev/urandom count=16 bs=1 2> /dev/null | hexdump -e '1/2 "%04x"'
50b71a8ef0bd5751ea76de6d6c98c03a
```

- b. Créer un CKN hexadécimal de 32 octets :

```
# dd if=/dev/urandom count=32 bs=1 2> /dev/null | hexdump -e '1/2 "%04x"'
f2b4297d39da7330910a74abc0449feb45b5c0b9fc23df1430e1898fcf1c4550
```

2. Sur les deux hôtes, vous souhaitez établir une connexion MACsec :

3. Créer la connexion MACsec :

```
# nmcli connection add type macsec con-name macsec0 ifname macsec0
connection.autoconnect yes macsec.parent enp1s0 macsec.mode psk macsec.mka-
cak 50b71a8ef0bd5751ea76de6d6c98c03a macsec.mka-ckn
f2b4297d39da7330910a74abc0449feb45b5c0b9fc23df1430e1898fcf1c4550
```

Utilisez les CAK et CKN générés à l'étape précédente dans les paramètres **macsec.mka-cak** et **macsec.mka-ckn**. Les valeurs doivent être identiques sur chaque hôte du réseau protégé par MACsec.

4. Configurez les paramètres IP de la connexion MACsec.

- a. Configurez les paramètres **IPv4**. Par exemple, pour définir une adresse **IPv4** statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **macsec0**, entrez :

```
# nmcli connection modify macsec0 ipv4.method manual ipv4.addresses
'192.0.2.1/24' ipv4.gateway '192.0.2.254' ipv4.dns '192.0.2.253'
```

- b. Configurez les paramètres **IPv6**. Par exemple, pour définir une adresse **IPv6** statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **macsec0**, entrez :

```
# nmcli connection modify macsec0 ipv6.method manual ipv6.addresses
'2001:db8:1::1/32' ipv6.gateway '2001:db8:1::fffe' ipv6.dns '2001:db8:1::fffd'
```

5. Activer la connexion :

```
# nmcli connection up macsec0
```

Vérification

1. Vérifiez que le trafic est crypté :

```
# tcpdump -nn -i enp1s0
```

2. Facultatif : Affichez le trafic non crypté :

■

```
# tcpdump -nn -i macsec0
```

3. Affiche les statistiques MACsec :

```
# ip macsec show
```

4. Afficher les compteurs individuels pour chaque type de protection : intégrité seule (cryptage désactivé) et cryptage (cryptage activé)

```
# ip -s macsec show
```

36.2. RESSOURCES SUPPLÉMENTAIRES

- [MACsec : une solution différente pour crypter le trafic réseau](#) blog.

CHAPITRE 37. UTILISER DES SERVEURS DNS DIFFÉRENTS POUR DES DOMAINES DIFFÉRENTS

Par défaut, Red Hat Enterprise Linux (RHEL) envoie toutes les requêtes DNS au premier serveur DNS spécifié dans le fichier `/etc/resolv.conf`. Si ce serveur ne répond pas, RHEL utilise le serveur suivant dans ce fichier.

Dans les environnements où un serveur DNS ne peut pas résoudre tous les domaines, les administrateurs peuvent configurer RHEL pour qu'il envoie les requêtes DNS pour un domaine spécifique à un serveur DNS sélectionné. Par exemple, vous pouvez configurer un serveur DNS pour résoudre les requêtes pour **example.com** et un autre serveur DNS pour résoudre les requêtes pour **example.net**. Pour toutes les autres requêtes DNS, RHEL utilise le serveur DNS configuré dans la connexion avec la passerelle par défaut.



IMPORTANT

Le service **systemd-resolved** est fourni en tant qu'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat, peuvent ne pas être complètes sur le plan fonctionnel et Red Hat ne recommande pas de les utiliser pour la production. Ces aperçus offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Consultez la section [Portée de l'assistance](#) pour les fonctionnalités de l'aperçu technologique sur le portail client de Red Hat pour obtenir des informations sur la portée de l'assistance pour les fonctionnalités de l'aperçu technologique.

37.1. ENVOI DE REQUÊTES DNS POUR UN DOMAINE SPÉCIFIQUE À UN SERVEUR DNS SÉLECTIONNÉ

Vous pouvez configurer le service **systemd-resolved** et NetworkManager pour qu'ils envoient des requêtes DNS pour un domaine spécifique à un serveur DNS sélectionné.

Si vous suivez la procédure, RHEL utilise le service DNS fourni par **systemd-resolved** dans le fichier `/etc/resolv.conf`. Le service **systemd-resolved** démarre un service DNS qui écoute sur le port **53** l'adresse IP **127.0.0.53**. Le service achemine dynamiquement les requêtes DNS vers les serveurs DNS correspondants spécifiés dans NetworkManager.



NOTE

L'adresse **127.0.0.53** n'est accessible qu'à partir du système local et non du réseau.

Conditions préalables

- Plusieurs connexions NetworkManager sont configurées sur le système.
- Un serveur DNS et un domaine de recherche sont configurés dans les connexions du NetworkManager qui sont responsables de la résolution d'un domaine spécifique
Par exemple, si le serveur DNS spécifié dans une connexion VPN doit résoudre les requêtes pour le domaine **example.com**, le profil de connexion VPN doit avoir :
 - Configuration d'un serveur DNS capable de résoudre **example.com**

- Configuré le domaine de recherche à **example.com** dans les paramètres **ipv4.dns-search** et **ipv6.dns-search**

Procédure

1. Démarrez et activez le service **systemd-resolved**:

```
# systemctl --now enable systemd-resolved
```

2. Modifiez le fichier **/etc/NetworkManager/NetworkManager.conf** et définissez l'entrée suivante dans la section **[main]**:

```
dns=systemd-resolved
```

3. Rechargez le service **NetworkManager**:

```
# systemctl reload NetworkManager
```

Vérification

1. Vérifiez que l'entrée **nameserver** dans le fichier **/etc/resolv.conf** fait référence à **127.0.0.53**:

```
# cat /etc/resolv.conf
nameserver 127.0.0.53
```

2. Vérifiez que le service **systemd-resolved** écoute sur le port **53** à l'adresse IP locale **127.0.0.53**:

```
# ss -tulpn | grep "127.0.0.53"
udp UNCONN 0 0 127.0.0.53%lo:53 0.0.0.0:* users:(("systemd-
resolve",pid=1050,fd=12))
tcp LISTEN 0 4096 127.0.0.53%lo:53 0.0.0.0:* users:(("systemd-
resolve",pid=1050,fd=13))
```

Ressources supplémentaires

- **NetworkManager.conf(5)** page de manuel

CHAPITRE 38. PREMIERS PAS AVEC IPVLAN

IPVLAN est un pilote pour un dispositif de réseau virtuel qui peut être utilisé dans un environnement de conteneur pour accéder au réseau hôte. L'IPVLAN expose une seule adresse MAC au réseau externe, quel que soit le nombre de dispositifs IPVLAN créés dans le réseau hôte. Cela signifie qu'un utilisateur peut avoir plusieurs dispositifs IPVLAN dans plusieurs conteneurs et que le commutateur correspondant lit une seule adresse MAC. Le pilote IPVLAN est utile lorsque le commutateur local impose des contraintes sur le nombre total d'adresses MAC qu'il peut gérer.

38.1. MODES IPVLAN

Les modes suivants sont disponibles pour IPVLAN :

- L2 mode**
 Dans le réseau IPVLAN **L2 mode**, les dispositifs virtuels reçoivent des requêtes ARP (protocole de résolution d'adresses) et y répondent. Le cadre **netfilter** s'exécute uniquement à l'intérieur du conteneur qui possède le dispositif virtuel. Aucune chaîne **netfilter** n'est exécutée dans l'espace de noms par défaut sur le trafic conteneurisé. L'utilisation de **L2 mode** offre de bonnes performances, mais moins de contrôle sur le trafic réseau.
- L3 mode**
 Dans **L3 mode**, les dispositifs virtuels ne traitent que le trafic **L3** et supérieur. Les dispositifs virtuels ne répondent pas aux requêtes ARP et les utilisateurs doivent configurer manuellement les entrées de voisinage pour les adresses IPVLAN sur les homologues concernés. Le trafic de sortie d'un conteneur donné est atterri sur les chaînes **netfilter** POSTROUTING et OUTPUT dans l'espace de noms par défaut, tandis que le trafic d'entrée est traité de la même manière que **L2 mode**. L'utilisation de **L3 mode** permet un bon contrôle mais diminue les performances du trafic réseau.
- L3S mode**
 Dans **L3S mode**, les dispositifs virtuels fonctionnent de la même manière que dans **L3 mode**, sauf que les trafics de sortie et d'entrée d'un conteneur donné sont atterris sur la chaîne **netfilter** dans l'espace de noms par défaut. **L3S mode** se comporte de la même manière que **L3 mode** mais offre un plus grand contrôle sur le réseau.



NOTE

L'appareil virtuel IPVLAN ne reçoit pas de trafic de diffusion et de multidiffusion dans le cas des modes **L3** et **L3S**.

38.2. COMPARAISON ENTRE IPVLAN ET MACVLAN

Le tableau suivant présente les principales différences entre MACVLAN et IPVLAN.

MACVLAN	IPVLAN
Utilise l'adresse MAC pour chaque appareil MACVLAN. Le dépassement du nombre d'adresses MAC dans la table MAC du commutateur peut entraîner une perte de connectivité.	Utilise une seule adresse MAC qui ne limite pas le nombre d'appareils IPVLAN.

MACVLAN	IPVLAN
Les règles Netfilter pour l'espace de noms global ne peuvent pas affecter le trafic à destination ou en provenance d'un périphérique MACVLAN dans un espace de noms enfant.	Il est possible de contrôler le trafic en provenance ou à destination d'un périphérique IPVLAN sur les sites L3 mode et L3S mode .

Notez que les réseaux IPVLAN et MACVLAN ne nécessitent aucun niveau d'encapsulation.

38.3. CRÉATION ET CONFIGURATION DU DISPOSITIF IPVLAN À L'AIDE DE IPRROUTE2

Cette procédure montre comment configurer le dispositif IPVLAN à l'aide de **iproute2**.

Procédure

1. Pour créer un périphérique IPVLAN, entrez la commande suivante :

```
# ip link add link real_NIC_device name IPVLAN_device type ipvlan mode I2
```

Notez que le contrôleur d'interface réseau (NIC) est un composant matériel qui connecte un ordinateur à un réseau.

Exemple 38.1. Création d'un dispositif IPVLAN

```
# ip link add link enp0s31f6 name my_ipvlan type ipvlan mode I2
# 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. Pour attribuer une adresse **IPv4** ou **IPv6** à l'interface, entrez la commande suivante :

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

3. En cas de configuration d'un dispositif IPVLAN sur **L3 mode** ou **L3S mode**, procédez aux réglages suivants :

- a. Configurer la configuration du voisin pour le pair distant sur l'hôte distant :

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

où *MAC_address* est l'adresse MAC du NIC réel sur lequel un dispositif IPVLAN est basé.

- b. Configurez un périphérique IPVLAN pour **L3 mode** à l'aide de la commande suivante :

```
# ip route add dev <real_NIC_device> <peer_IP_address/32>
```

Pour L3S mode:

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

où l'adresse IP représente l'adresse de l'homologue distant.

4. Pour activer un périphérique IPVLAN, entrez la commande suivante :

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

5. Pour vérifier si le dispositif IPVLAN est actif, exécutez la commande suivante sur l'hôte distant :

```
# ping IP_address
```

où le site *IP_address* utilise l'adresse IP de l'appareil IPVLAN.

CHAPITRE 39. RÉUTILISATION DE LA MÊME ADRESSE IP SUR DIFFÉRENTES INTERFACES

Avec le routage et le transfert virtuels (VRF), les administrateurs peuvent utiliser simultanément plusieurs tables de routage sur le même hôte. Pour ce faire, la VRF partitionne un réseau au niveau de la couche 3. Cela permet à l'administrateur d'isoler le trafic en utilisant des tables de routage séparées et indépendantes par domaine VRF. Cette technique est similaire à celle des réseaux locaux virtuels (VLAN), qui partitionne un réseau au niveau de la couche 2, où le système d'exploitation utilise différentes étiquettes VLAN pour isoler le trafic partageant le même support physique.

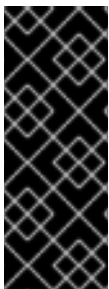
L'un des avantages de la VRF par rapport au partitionnement sur la couche 2 est que le routage s'adapte mieux au nombre de pairs concernés.

Red Hat Enterprise Linux utilise un périphérique virtuel **vrt** pour chaque domaine VRF et ajoute des routes à un domaine VRF en ajoutant des périphériques réseau existants à un périphérique VRF. Les adresses et les itinéraires précédemment attachés au périphérique d'origine seront déplacés à l'intérieur du domaine VRF.

Notez que chaque domaine VRF est isolé l'un de l'autre.

39.1. RÉUTILISATION PERMANENTE DE LA MÊME ADRESSE IP SUR DIFFÉRENTES INTERFACES

Vous pouvez utiliser la fonction de routage et de transfert virtuels (VRF) pour utiliser en permanence la même adresse IP sur différentes interfaces d'un serveur.



IMPORTANT

Pour permettre aux homologues distants de contacter les deux interfaces VRF tout en réutilisant la même adresse IP, les interfaces réseau doivent appartenir à des domaines de diffusion différents. Un domaine de diffusion dans un réseau est un ensemble de nœuds qui reçoivent le trafic de diffusion envoyé par n'importe lequel d'entre eux. Dans la plupart des configurations, tous les nœuds connectés au même commutateur appartiennent au même domaine de diffusion.

Conditions préalables

- Vous êtes connecté en tant qu'utilisateur **root**.
- Les interfaces réseau ne sont pas configurées.

Procédure

1. Créez et configurez le premier dispositif VRF :
 - a. Créez une connexion pour le périphérique VRF et affectez-la à une table de routage. Par exemple, pour créer un périphérique VRF nommé **vrf0** qui est affecté à la table de routage **1001**:

```
# nmcli connection add type vrf ifname vrf0 con-name vrf0 table 1001 ipv4.method disabled ipv6.method disabled
```

- b. Activez le dispositif **vrf0**:

■

```
# nmcli connection up vrf0
```

- c. Attribuez un périphérique réseau au VRF qui vient d'être créé. Par exemple, pour ajouter le périphérique Ethernet **enp1s0** au périphérique VRF **vrf0** et attribuer une adresse IP et un masque de sous-réseau à **enp1s0**, entrez :

```
# nmcli connection add type ethernet con-name vrf.enp1s0 ifname enp1s0 master vrf0 ipv4.method manual ipv4.address 192.0.2.1/24
```

- d. Activez la connexion **vrf.enp1s0**:

```
# nmcli connection up vrf.enp1s0
```

2. Créez et configurez le périphérique VRF suivant :

- a. Créez le périphérique VRF et affectez-le à une table de routage. Par exemple, pour créer un périphérique VRF nommé **vrf1** qui est assigné à la table de routage **1002**, entrez :

```
# nmcli connection add type vrf ifname vrf1 con-name vrf1 table 1002 ipv4.method disabled ipv6.method disabled
```

- b. Activez le dispositif **vrf1**:

```
# nmcli connection up vrf1
```

- c. Attribuez un périphérique réseau au VRF qui vient d'être créé. Par exemple, pour ajouter le périphérique Ethernet **enp7s0** au périphérique VRF **vrf1** et attribuer une adresse IP et un masque de sous-réseau à **enp7s0**, entrez :

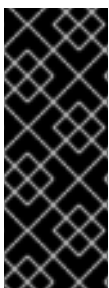
```
# nmcli connection add type ethernet con-name vrf.enp7s0 ifname enp7s0 master vrf1 ipv4.method manual ipv4.address 192.0.2.1/24
```

- d. Activez le dispositif **vrf.enp7s0**:

```
# nmcli connection up vrf.enp7s0
```

39.2. RÉUTILISATION TEMPORAIRE DE LA MÊME ADRESSE IP SUR DIFFÉRENTES INTERFACES

Vous pouvez utiliser la fonction de routage et de transfert virtuels (VRF) pour utiliser temporairement la même adresse IP sur différentes interfaces d'un serveur. N'utilisez cette procédure qu'à des fins de test, car la configuration est temporaire et perdue après le redémarrage du système.



IMPORTANT

Pour permettre aux homologues distants de contacter les deux interfaces VRF tout en réutilisant la même adresse IP, les interfaces réseau doivent appartenir à des domaines de diffusion différents. Un domaine de diffusion dans un réseau est un ensemble de nœuds qui reçoivent le trafic de diffusion envoyé par n'importe lequel d'entre eux. Dans la plupart des configurations, tous les nœuds connectés au même commutateur appartiennent au même domaine de diffusion.

Conditions préalables

- Vous êtes connecté en tant qu'utilisateur **root**.
- Les interfaces réseau ne sont pas configurées.

Procédure

1. Créez et configurez le premier dispositif VRF :

- a. Créez le périphérique VRF et assignez-le à une table de routage. Par exemple, pour créer un périphérique VRF nommé **blue** qui est assigné à la table de routage **1001**:

```
# ip link add dev blue type vrf table 1001
```

- b. Activez le dispositif **blue**:

```
# ip link set dev blue up
```

- c. Attribuez un périphérique réseau au périphérique VRF. Par exemple, pour ajouter le périphérique Ethernet **enp1s0** au périphérique VRF **blue**:

```
# ip link set dev enp1s0 master blue
```

- d. Activez le dispositif **enp1s0**:

```
# ip link set dev enp1s0 up
```

- e. Attribuez une adresse IP et un masque de sous-réseau à l'appareil **enp1s0**. Par exemple, pour le configurer sur **192.0.2.1/24**:

```
# ip addr add dev enp1s0 192.0.2.1/24
```

2. Créez et configurez le périphérique VRF suivant :

- a. Créez le périphérique VRF et assignez-le à une table de routage. Par exemple, pour créer un périphérique VRF nommé **red** qui est assigné à la table de routage **1002**:

```
# ip link add dev red type vrf table 1002
```

- b. Activez le dispositif **red**:

```
# ip link set dev red up
```

- c. Attribuez un périphérique réseau au périphérique VRF. Par exemple, pour ajouter le périphérique Ethernet **enp7s0** au périphérique VRF **red**:

```
# ip link set dev enp7s0 master red
```

- d. Activez le dispositif **enp7s0**:

```
# ip link set dev enp7s0 up
```


- e. Attribuez au périphérique **enp7s0** la même adresse IP et le même masque de sous-réseau que ceux utilisés pour **enp1s0** dans le domaine VRF **blue**:

```
# ip addr add dev enp7s0 192.0.2.1/24
```

3. Il est possible de créer d'autres dispositifs VRF comme décrit ci-dessus.

39.3. RESSOURCES SUPPLÉMENTAIRES

- `/usr/share/doc/kernel-doc-<kernel_version>/Documentation/networking/vrf.txt` du paquet `kernel-doc`

CHAPITRE 40. DÉMARRAGE D'UN SERVICE DANS UN RÉSEAU VRF ISOLÉ

Le routage et la transmission virtuels (VRF) permettent de créer des réseaux isolés avec une table de routage différente de la table de routage principale du système d'exploitation. Vous pouvez ensuite lancer des services et des applications de manière à ce qu'ils n'aient accès qu'au réseau défini dans cette table de routage.

40.1. CONFIGURATION D'UN DISPOSITIF VRF

Pour utiliser le routage et le transfert virtuels (VRF), vous devez créer un périphérique VRF et lui attacher une interface réseau physique ou virtuelle ainsi que des informations de routage.



AVERTISSEMENT

Pour éviter de vous bloquer vous-même à distance, effectuez cette procédure sur la console locale ou à distance via une interface réseau que vous ne souhaitez pas affecter au périphérique VRF.

Conditions préalables

- Vous êtes connecté localement ou vous utilisez une interface réseau différente de celle que vous souhaitez attribuer au périphérique VRF.

Procédure

1. Créez la connexion **vrf0** avec un dispositif virtuel du même nom et attachez-la à la table de routage **1000**:

```
# nmcli connection add type vrf ifname vrf0 con-name vrf0 table 1000 ipv4.method disabled ipv6.method disabled
```

2. Ajoutez l'appareil **enp1s0** à la connexion **vrf0** et configurez les paramètres IP :

```
# nmcli connection add type ethernet con-name enp1s0 ifname enp1s0 master vrf0 ipv4.method manual ipv4.address 192.0.2.1/24 ipv4.gateway 192.0.2.254
```

Cette commande crée la connexion **enp1s0** en tant que port de la connexion **vrf0**. Grâce à cette configuration, les informations de routage sont automatiquement attribuées à la table de routage **1000** associée au périphérique **vrf0**.

3. Si vous avez besoin de routes statiques dans le réseau isolé :
 - a. Ajouter les routes statiques :

```
# nmcli connection modify enp1s0 ipv4.routes "198.51.100.0/24 192.0.2.2"
```

Ceci ajoute une route vers le réseau **198.51.100.0/24** qui utilise **192.0.2.2** comme routeur.

b. Activer la connexion :

```
# nmcli connection up enp1s0
```

Vérification

1. Affichez les paramètres IP de l'appareil associé à **vrf0**:

```
# ip -br addr show vrf vrf0
enp1s0 UP 192.0.2.1/24
```

2. Affichez les périphériques VRF et leur table de routage associée :

```
# ip vrf show
Name          Table
-----
vrf0         1000
```

3. Affiche la table de routage principale :

```
# ip route show
default via 203.0.113.0/24 dev enp7s0 proto static metric 100
```

La table de routage principale ne mentionne aucune route associée au périphérique **enp1s0** ou au sous-réseau **192.0.2.1/24**.

4. Afficher la table de routage **1000**:

```
# ip route show table 1000
default via 192.0.2.254 dev enp1s0 proto static metric 101
broadcast 192.0.2.0 dev enp1s0 proto kernel scope link src 192.0.2.1
192.0.2.0/24 dev enp1s0 proto kernel scope link src 192.0.2.1 metric 101
local 192.0.2.1 dev enp1s0 proto kernel scope host src 192.0.2.1
broadcast 192.0.2.255 dev enp1s0 proto kernel scope link src 192.0.2.1
198.51.100.0/24 via 192.0.2.2 dev enp1s0 proto static metric 101
```

L'entrée **default** indique que les services qui utilisent cette table de routage utilisent **192.0.2.254** comme passerelle par défaut et non la passerelle par défaut de la table de routage principale.

5. Exécutez l'utilitaire **traceroute** dans le réseau associé à **vrf0** pour vérifier que l'utilitaire utilise la route de la table **1000**:

```
# ip vrf exec vrf0 traceroute 203.0.113.1
traceroute to 203.0.113.1 (203.0.113.1), 30 hops max, 60 byte packets
 1 192.0.2.254 (192.0.2.254) 0.516 ms 0.459 ms 0.430 ms
 ...
```

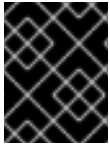
Le premier saut est la passerelle par défaut attribuée à la table de routage **1000** et non la passerelle par défaut de la table de routage principale du système.

Ressources supplémentaires

- **ip-vrf(8)** page de manuel

40.2. DÉMARRAGE D'UN SERVICE DANS UN RÉSEAU VRF ISOLÉ

Vous pouvez configurer un service, tel que le serveur HTTP Apache, pour qu'il démarre au sein d'un réseau VRF (virtual routing and forwarding) isolé.



IMPORTANT

Les services ne peuvent se lier qu'à des adresses IP locales situées dans le même réseau VRF.

Conditions préalables

- Vous avez configuré le dispositif **vrf0**.
- Vous avez configuré Apache HTTP Server pour qu'il écoute uniquement l'adresse IP attribuée à l'interface associée au périphérique **vrf0**.

Procédure

1. Afficher le contenu du service **httpd** systemd :

```
# systemctl cat httpd
...
[Service]
ExecStart=/usr/sbin/httpd $OPTIONS -DFOREGROUND
...
```

Vous aurez besoin du contenu du paramètre **ExecStart** dans une étape ultérieure pour exécuter la même commande dans le réseau VRF isolé.

2. Créez le répertoire **/etc/systemd/system/httpd.service.d/**:

```
# mkdir /etc/systemd/system/httpd.service.d/
```

3. Créez le fichier **/etc/systemd/system/httpd.service.d/override.conf** avec le contenu suivant :

```
[Service]
ExecStart=
ExecStart=/usr/sbin/ip vrf exec vrf0 /usr/sbin/httpd $OPTIONS -DFOREGROUND
```

Pour remplacer le paramètre **ExecStart**, vous devez d'abord le désactiver, puis lui attribuer une nouvelle valeur, comme indiqué.

4. Recharger systemd.

```
# systemctl daemon-reload
```

5. Redémarrez le service **httpd**.

```
# systemctl restart httpd
```

Vérification

1. Affiche les identifiants de processus (PID) des processus **httpd**:

```
# pidof -c httpd
1904 ...
```

2. Affichez l'association VRF pour les PID, par exemple :

```
# ip vrf identify 1904
vrf0
```

3. Affiche tous les PID associés à l'appareil **vrf0**:

```
# ip vrf pids vrf0
1904 httpd
...
```

Ressources supplémentaires

- **ip-vrf(8)** page de manuel

CHAPITRE 41. EXÉCUTION DES CROCHETS DE SORTIE DE DHCLIENT À L'AIDE D'UN SCRIPT DE DISTRIBUTION DE NETWORKMANAGER

Vous pouvez utiliser un script de distribution de NetworkManager pour exécuter les crochets de sortie de **dhclient**.

41.1. LE CONCEPT DE SCRIPTS DE DISTRIBUTION DU NETWORKMANAGER

Le service **NetworkManager-dispatcher** exécute les scripts fournis par l'utilisateur dans l'ordre alphabétique lorsque des événements réseau se produisent. Ces scripts sont généralement des scripts shell, mais il peut s'agir de n'importe quel script exécutable ou de n'importe quelle application. Vous pouvez utiliser les scripts du répartiteur, par exemple, pour ajuster les paramètres liés au réseau que vous ne pouvez pas gérer avec NetworkManager.

Vous pouvez stocker les scripts du répartiteur dans les répertoires suivants :

- **/etc/NetworkManager/dispatcher.d/**: L'emplacement général des scripts du répartiteur que l'utilisateur de **root** peut modifier.
- **/usr/lib/NetworkManager/dispatcher.d/**: Pour les scripts de distribution immuables pré-déployés.

Pour des raisons de sécurité, le service **NetworkManager-dispatcher** n'exécute les scripts que si les conditions suivantes sont remplies :

- Le script appartient à l'utilisateur **root**.
- Le script ne peut être lu et écrit que par **root**.
- Le bit **setuid** n'est pas activé dans le script.

Le service **NetworkManager-dispatcher** exécute chaque script avec deux arguments :

1. Le nom de l'interface de l'appareil sur lequel l'opération s'est produite.
2. L'action, telle que **up**, lorsque l'interface a été activée.

La section **Dispatcher scripts** de la page de manuel **NetworkManager(8)** donne un aperçu des actions et des variables d'environnement que vous pouvez utiliser dans les scripts.

Le service **NetworkManager-dispatcher** exécute un script à la fois, mais de manière asynchrone par rapport au processus principal de NetworkManager. Notez que si un script est mis en file d'attente, le service l'exécutera toujours, même si un événement ultérieur le rend obsolète. Cependant, le service **NetworkManager-dispatcher** exécute immédiatement les scripts qui sont des liens symboliques renvoyant à des fichiers dans **/etc/NetworkManager/dispatcher.d/no-wait.d/**, sans attendre la fin des scripts précédents, et en parallèle.

Ressources supplémentaires

- **NetworkManager(8)** page de manuel

41.2. CRÉATION D'UN SCRIPT DE DISTRIBUTION DE NETWORKMANAGER QUI EXÉCUTE LES CROCHETS DE SORTIE DE DHCLIENT

Lorsqu'un serveur DHCP attribue ou met à jour une adresse IPv4, NetworkManager peut exécuter un script de distribution stocké dans le répertoire `/etc/dhcp/dhclient-exit-hooks.d/`. Ce script peut alors, par exemple, exécuter les crochets de sortie **dhclient**.

Conditions préalables

- Les crochets de sortie de **dhclient** sont stockés dans le répertoire `/etc/dhcp/dhclient-exit-hooks.d/`.

Procédure

1. Créez le fichier `/etc/NetworkManager/dispatcher.d/12-dhclient-down` avec le contenu suivant :

```
#!/bin/bash
# Run dhclient.exit-hooks.d scripts

if [ -n "$DHCP4_DHCP_LEASE_TIME" ]; then
  if [ "$2" = "dhcp4-change" ] || [ "$2" = "up" ]; then
    if [ -d /etc/dhcp/dhclient-exit-hooks.d ]; then
      for f in /etc/dhcp/dhclient-exit-hooks.d/*.sh ; do
        if [ -x "$f" ]; then
          . "$f"
        fi
      done
    fi
  fi
fi
```

2. Définir l'utilisateur **root** comme propriétaire du fichier :

```
# chown root:root /etc/NetworkManager/dispatcher.d/12-dhclient-down
```

3. Définissez les autorisations de manière à ce que seul l'utilisateur root puisse l'exécuter :

```
# chmod 0700 /etc/NetworkManager/dispatcher.d/12-dhclient-down
```

4. Rétablir le contexte SELinux :

```
# restorecon /etc/NetworkManager/dispatcher.d/12-dhclient-down
```

Ressources supplémentaires

- **NetworkManager(8)** page de manuel

CHAPITRE 42. INTRODUCTION AU DÉBOGAGE DE NETWORKMANAGER

L'augmentation des niveaux de journalisation pour tous les domaines ou pour certains d'entre eux permet d'enregistrer plus de détails sur les opérations effectuées par NetworkManager. Vous pouvez utiliser ces informations pour résoudre les problèmes. NetworkManager fournit différents niveaux et domaines pour produire des informations de journalisation. Le fichier `/etc/NetworkManager/NetworkManager.conf` est le fichier de configuration principal de NetworkManager. Les journaux sont stockés dans le journal.

42.1. INTRODUCTION À LA MÉTHODE DE RÉAPPLICATION DU NETWORKMANAGER

Le service **NetworkManager** utilise un profil pour gérer les paramètres de connexion d'un appareil. L'API Desktop Bus (D-Bus) peut créer, modifier et supprimer ces paramètres de connexion. Pour toute modification d'un profil, l'API D-Bus clone les paramètres existants dans les paramètres modifiés d'une connexion. Malgré le clonage, les changements ne s'appliquent pas aux paramètres modifiés. Pour que le clonage soit effectif, il faut réactiver les paramètres existants d'une connexion ou utiliser la méthode **reapply()**.

La méthode **reapply()** présente les caractéristiques suivantes :

1. Mise à jour des paramètres de connexion modifiés sans désactivation ou redémarrage d'une interface réseau.
2. Suppression des changements en attente dans les paramètres de connexion modifiés. Comme **NetworkManager** n'annule pas les modifications manuelles, vous pouvez reconfigurer l'appareil et annuler les paramètres externes ou manuels.
3. Création de paramètres de connexion modifiés différents des paramètres de connexion existants.

En outre, la méthode **reapply()** prend en charge les attributs suivants :

- **bridge.ageing-time**
- **bridge.forward-delay**
- **bridge.group-address**
- **bridge.group-forward-mask**
- **bridge.hello-time**
- **bridge.max-age**
- **bridge.multicast-hash-max**
- **bridge.multicast-last-member-count**
- **bridge.multicast-last-member-interval**
- **bridge.multicast-membership-interval**
- **bridge.multicast-querier**

- **bridge.multicast-querier-interval**
- **bridge.multicast-query-interval**
- **bridge.multicast-query-response-interval**
- **bridge.multicast-query-use-ifaddr**
- **bridge.multicast-router**
- **bridge.multicast-snooping**
- **bridge.multicast-startup-query-count**
- **bridge.multicast-startup-query-interval**
- **bridge.priority**
- **bridge.stp**
- **bridge.VLAN-filtering**
- **bridge.VLAN-protocol**
- **bridge.VLANs**
- **802-3-ethernet.accept-all-mac-addresses**
- **802-3-ethernet.cloned-mac-address**
- **IPv4.addresses**
- **IPv4.dhcp-client-id**
- **IPv4.dhcp-iaid**
- **IPv4.dhcp-timeout**
- **IPv4.DNS**
- **IPv4.DNS-priority**
- **IPv4.DNS-search**
- **IPv4.gateway**
- **IPv4.ignore-auto-DNS**
- **IPv4.ignore-auto-routes**
- **IPv4.may-fail**
- **IPv4.method**
- **IPv4.never-default**
- **IPv4.route-table**

- IPv4.routes
- IPv4.routing-rules
- IPv6.addr-gen-mode
- IPv6.addresses
- IPv6.dhcp-duid
- IPv6.dhcp-iaid
- IPv6.dhcp-timeout
- IPv6.DNS
- IPv6.DNS-priority
- IPv6.DNS-search
- IPv6.gateway
- IPv6.ignore-auto-DNS
- IPv6.may-fail
- IPv6.method
- IPv6.never-default
- IPv6.ra-timeout
- IPv6.route-metric
- IPv6.route-table
- IPv6.routes
- IPv6.routing-rules

Ressources supplémentaires

- [nm-settings-nmcli\(5\)](#) page de manuel

42.2. CONFIGURATION DU NIVEAU DE JOURNALISATION DE NETWORKMANAGER

Par défaut, tous les domaines de journalisation sont configurés pour enregistrer le niveau de journalisation **INFO**. Désactivez la limitation du débit avant de collecter les journaux de débogage. Avec la limitation du débit, **systemd-journald** laisse tomber les messages s'ils sont trop nombreux en peu de temps. Cela peut se produire lorsque le niveau de journalisation est **TRACE**.

Cette procédure permet de désactiver la limitation de débit et d'activer l'enregistrement des journaux de débogage pour tous les domaines (ALL).

Procédure

1. Pour désactiver la limitation de vitesse, modifiez le fichier `/etc/systemd/journald.conf`, décommentez le paramètre **RateLimitBurst** dans la section **[Journal]** et définissez sa valeur comme **0**:

```
RateLimitBurst=0
```

2. Redémarrez le service **systemd-journald**.

```
# systemctl restart systemd-journald
```

3. Créez le fichier `/etc/NetworkManager/conf.d/95-nm-debug.conf` avec le contenu suivant :

```
[logging]
domains=ALL:TRACE
```

Le paramètre **domains** peut contenir plusieurs paires **domain:level** séparées par des virgules.

4. Redémarrez le service NetworkManager.

```
# systemctl restart NetworkManager
```

Vérification

- Interroger le journal **systemd** pour afficher les entrées du journal de l'unité **NetworkManager**:

```
# journalctl -u NetworkManager
```

```
...
Jun 30 15:24:32 server NetworkManager[164187]: <debug> [1656595472.4939] active-
connection[0x5565143c80a0]: update activation type from assume to managed
Jun 30 15:24:32 server NetworkManager[164187]: <trace> [1656595472.4939]
device[55b33c3bdb72840c] (enp1s0): sys-iface-state: assume -> managed
Jun 30 15:24:32 server NetworkManager[164187]: <trace> [1656595472.4939]
l3cfg[4281fdf43e356454,ifindex=3]: commit type register (type "update", source "device",
existing a369f23014b9ede3) -> a369f23014b9ede3
Jun 30 15:24:32 server NetworkManager[164187]: <info> [1656595472.4940] manager:
NetworkManager state is now CONNECTED_SITE
...
```

42.3. RÉGLAGE TEMPORAIRE DES NIVEAUX DE JOURNALISATION AU MOMENT DE L'EXÉCUTION EN UTILISANT NMCLI

Vous pouvez modifier le niveau de journalisation au moment de l'exécution en utilisant **nmcli**. Cependant, Red Hat recommande d'activer le débogage à l'aide des fichiers de configuration et de redémarrer NetworkManager. La mise à jour du débogage **levels** et **domains** à l'aide du fichier **.conf** permet de déboguer les problèmes de démarrage et de capturer tous les journaux à partir de l'état initial.

Procédure

1. Facultatif : Affiche les paramètres de journalisation actuels :

```
# nmcli general logging
LEVEL DOMAINS
```

INFO

```
PLATFORM,RFKILL,ETHER,WIFI,BT,MB,DHCP4,DHCP6,PPP,WIFI_SCAN,IP4,IP6,A
UTOIP4,DNS,VPN,SHARING,SUPPLICANT,AGENTS,SETTINGS,SUSPEND,CORE,DEVIC
E,OLPC,
WIMAX,INFINIBAND,FIREWALL,ADSL,BOND,VLAN,BRIDGE,DBUS_PROPS,TEAM,CONC
HECK,DC
B,DISPATCH
```

2. Pour modifier le niveau de journalisation et les domaines, utilisez les options suivantes :

- Pour définir le niveau de journalisation de tous les domaines sur le même **LEVEL**, entrez :

```
# nmcli general logging level LEVEL domains ALL
```

- Pour modifier le niveau pour des domaines spécifiques, entrez :

```
# nmcli general logging level LEVEL domains DOMAINS
```

Notez que la mise à jour du niveau de journalisation à l'aide de cette commande désactive la journalisation pour tous les autres domaines.

- Pour modifier le niveau de certains domaines et préserver le niveau de tous les autres domaines, entrez :

```
# nmcli general logging level KEEP domains DOMAIN:LEVEL,DOMAIN:LEVEL
```

42.4. VISUALISATION DES JOURNAUX DE NETWORKMANAGER

Vous pouvez consulter les journaux de NetworkManager pour le dépannage.

Procédure

- Pour consulter les journaux, entrez :

```
# journalctl -u NetworkManager -b
```

Ressources supplémentaires

- **NetworkManager.conf(5)** page de manuel
- **journalctl(1)** page de manuel

42.5. NIVEAUX ET DOMAINES DE DÉBOGAGE

Vous pouvez utiliser les paramètres **levels** et **domains** pour gérer le débogage de NetworkManager. Le niveau définit le niveau de verbosité, tandis que les domaines définissent la catégorie des messages à enregistrer dans les journaux avec une gravité donnée (**level**).

Niveaux de journalisation	Description
OFF	N'enregistre aucun message concernant NetworkManager
ERR	N'enregistre que les erreurs critiques
WARN	Consigne les avertissements qui peuvent refléter l'opération
INFO	Enregistre divers messages d'information utiles pour le suivi de l'état et des opérations
DEBUG	Active la journalisation verbeuse à des fins de débogage
TRACE	Permet une journalisation plus verbeuse que le niveau DEBUG

Notez que les niveaux suivants enregistrent tous les messages des niveaux précédents. Par exemple, en définissant le niveau de journalisation à **INFO**, les messages contenus dans les niveaux de journalisation **ERR** et **WARN** sont également journalisés.

Ressources supplémentaires

- **NetworkManager.conf(5)** page de manuel

CHAPITRE 43. INTRODUCTION À NMSTATE

Nmstate est une API déclarative de gestion de réseau. Le paquetage **nmstate** fournit la bibliothèque Python **libnmstate** et un utilitaire en ligne de commande, **nmstatectl**, pour gérer NetworkManager sur RHEL. Lorsque vous utilisez Nmstate, vous décrivez l'état attendu du réseau à l'aide d'instructions formatées YAML ou JSON.

Le Nmstate présente de nombreux avantages. Par exemple, il :

- Fournit une interface stable et extensible pour gérer les capacités du réseau RHEL
- Prise en charge des opérations atomiques et transactionnelles au niveau de l'hôte et du cluster
- Permet la modification partielle de la plupart des propriétés et préserve les paramètres existants qui ne sont pas spécifiés dans les instructions
- La prise en charge des plug-ins permet aux administrateurs d'utiliser leurs propres plug-ins

43.1. UTILISATION DE LA BIBLIOTHÈQUE LIBNMSTATE DANS UNE APPLICATION PYTHON

La bibliothèque Python **libnmstate** permet aux développeurs d'utiliser le Nmstate dans leurs propres applications

Pour utiliser la bibliothèque, importez-la dans votre code source :

```
import libnmstate
```

Notez que vous devez installer le paquetage **nmstate** pour utiliser cette bibliothèque.

Exemple 43.1. Interroger l'état du réseau à l'aide de la bibliothèque libnmstate

Le code Python suivant importe la bibliothèque **libnmstate** et affiche les interfaces réseau disponibles et leur état :

```
import json
import libnmstate
from libnmstate.schema import Interface

net_state = libnmstate.show()
for iface_state in net_state[Interface.KEY]:
    print(iface_state[Interface.NAME] + ": "
          + iface_state[Interface.STATE])
```

43.2. MISE À JOUR DE LA CONFIGURATION ACTUELLE DU RÉSEAU À L'AIDE DE NMSTATECTL

Vous pouvez utiliser l'utilitaire **nmstatectl** pour enregistrer la configuration réseau actuelle d'une ou de toutes les interfaces dans un fichier. Vous pouvez ensuite utiliser ce fichier pour :

- Modifier la configuration et l'appliquer au même système.

- Copiez le fichier sur un autre hôte et configurez l'hôte avec les mêmes paramètres ou des paramètres modifiés.

Par exemple, vous pouvez exporter les paramètres de l'interface **enp1s0** vers un fichier, modifier la configuration et appliquer les paramètres à l'hôte.

Conditions préalables

- Le paquet **nmstate** est installé.

Procédure

1. Exporter les paramètres de l'interface **enp1s0** vers le fichier `~/network-config.yml`:

```
# nmstatectl show enp1s0 > ~/network-config.yml
```

Cette commande enregistre la configuration de **enp1s0** au format YAML. Pour stocker la sortie au format JSON, passez l'option **--json** à la commande.

Si vous ne spécifiez pas de nom d'interface, **nmstatectl** exporte la configuration de toutes les interfaces.

2. Modifiez le fichier `~/network-config.yml` à l'aide d'un éditeur de texte pour mettre à jour la configuration.
3. Appliquez les paramètres du fichier `~/network-config.yml`:

```
# nmstatectl apply ~/network-config.yml
```

Si vous avez exporté les paramètres au format JSON, passez l'option **--json** à la commande.

43.3. ÉTATS DU RÉSEAU POUR LE RÔLE DE SYSTÈME RHEL

Le rôle système **network** RHEL prend en charge les configurations d'état dans les playbooks pour configurer les périphériques. Pour ce faire, utilisez la variable **network_state** suivie des configurations d'état.

Avantages de l'utilisation de la variable **network_state** dans un playbook :

- En utilisant la méthode déclarative avec les configurations d'état, vous pouvez configurer des interfaces et le NetworkManager crée un profil pour ces interfaces en arrière-plan.
- Avec la variable **network_state**, vous pouvez spécifier les options que vous souhaitez modifier, et toutes les autres options resteront inchangées. En revanche, avec la variable **network_connections**, vous devez spécifier tous les paramètres pour modifier le profil de connexion au réseau.

Par exemple, pour créer une connexion Ethernet avec des paramètres d'adresse IP dynamiques, utilisez le bloc **vars** suivant dans votre playbook :

Playbook avec configurations d'état	Manuel de jeu régulier
-------------------------------------	------------------------

```
vars:
  network_state:
  interfaces:
  - name: enp7s0
    type: ethernet
    state: up
  ipv4:
    enabled: true
    auto-dns: true
    auto-gateway: true
    auto-routes: true
    dhcp: true
  ipv6:
    enabled: true
    auto-dns: true
    auto-gateway: true
    auto-routes: true
    autoconf: true
    dhcp: true
```

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

Par exemple, pour modifier uniquement l'état de connexion des paramètres d'adresse IP dynamique que vous avez créés comme indiqué ci-dessus, utilisez le bloc **vars** suivant dans votre playbook :

Playbook avec configurations d'état	Manuel de jeu régulier
<pre>vars: network_state: interfaces: - name: enp7s0 type: ethernet state: down</pre>	<pre>vars: network_connections: - name: enp7s0 interface_name: enp7s0 type: ethernet autoconnect: yes ip: dhcp4: yes auto6: yes state: down</pre>

Ressources supplémentaires

- [/usr/share/ansible/roles/rhel-system-roles.network/README.md](#) fichier
- [Introduction à Nmstate](#)

43.4. RESSOURCES SUPPLÉMENTAIRES

- [/usr/share/doc/nmstate/README.md](#)
- [/usr/share/doc/nmstate/examples/](#)

CHAPITRE 44. CAPTURER DES PAQUETS RÉSEAU

Pour déboguer les problèmes de réseau et les communications, vous pouvez capturer des paquets réseau. Les sections suivantes fournissent des instructions et des informations supplémentaires sur la capture de paquets réseau.

44.1. UTILISATION DE XDPDUMP POUR CAPTURER DES PAQUETS RÉSEAU, Y COMPRIS LES PAQUETS ABANDONNÉS PAR LES PROGRAMMES XDP

L'utilitaire **xpdump** capture les paquets du réseau. Contrairement à l'utilitaire **tcpdump**, **xpdump** utilise un programme Berkeley Packet Filter (eBPF) étendu pour cette tâche. Cela permet à **xpdump** de capturer également les paquets abandonnés par les programmes Express Data Path (XDP). Les utilitaires de l'espace utilisateur, tels que **tcpdump**, ne sont pas en mesure de capturer ces paquets abandonnés, ni les paquets originaux modifiés par un programme XDP.

Vous pouvez utiliser **xpdump** pour déboguer les programmes XDP qui sont déjà attachés à une interface. Par conséquent, l'utilitaire peut capturer des paquets avant le démarrage d'un programme XDP et après sa fin. Dans ce dernier cas, **xpdump** capture également l'action XDP. Par défaut, **xpdump** capture les paquets entrants à l'entrée du programme XDP.

IMPORTANT

Sur les architectures autres que AMD et Intel 64-bit, l'utilitaire **xpdump** est fourni en tant qu'aperçu technologique uniquement. Les fonctionnalités de l'aperçu technologique ne sont pas prises en charge par les accords de niveau de service (SLA) de production de Red Hat, peuvent ne pas être complètes sur le plan fonctionnel et Red Hat ne recommande pas de les utiliser pour la production. Ces aperçus offrent un accès anticipé aux fonctionnalités des produits à venir, ce qui permet aux clients de tester les fonctionnalités et de fournir un retour d'information pendant le processus de développement.

Consultez la section [Portée de l'assistance](#) pour les fonctionnalités de l'aperçu technologique sur le portail client de Red Hat pour obtenir des informations sur la portée de l'assistance pour les fonctionnalités de l'aperçu technologique.

Notez que **xpdump** n'a aucune capacité de filtrage ou de décodage de paquets. Cependant, vous pouvez l'utiliser en combinaison avec **tcpdump** pour le décodage de paquets.

Conditions préalables

- Pilote de réseau qui prend en charge les programmes XDP.
- Un programme XDP est chargé sur l'interface **enp1s0**. Si aucun programme n'est chargé, **xpdump** capture les paquets de la même manière que **tcpdump**, pour des raisons de compatibilité ascendante.

Procédure

1. Pour capturer des paquets sur l'interface **enp1s0** et les écrire dans le fichier **/root/capture.pcap**, entrez :

```
# xpdump -i enp1s0 -w /root/capture.pcap
```

2. Pour arrêter la capture de paquets, appuyez sur **Ctrl+C**.

Ressources supplémentaires

- **xdpdump(8)** page de manuel
- Si vous êtes un développeur et que vous êtes intéressé par le code source de **xdpdump**, téléchargez et installez le RPM source correspondant (SRPM) à partir du portail client de Red Hat.

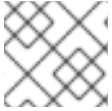
44.2. RESSOURCES SUPPLÉMENTAIRES

- [Comment capturer des paquets réseau avec tcpdump ?](#)

CHAPITRE 45. DÉMARRER AVEC DPDK

Le kit de développement du plan de données (DPDK) fournit des bibliothèques et des pilotes de réseau pour accélérer le traitement des paquets dans l'espace utilisateur.

Les administrateurs utilisent DPDK, par exemple, dans les machines virtuelles pour utiliser la virtualisation d'E/S à racine unique (SR-IOV) afin de réduire les latences et d'augmenter le débit d'E/S.



NOTE

Red Hat ne prend pas en charge les API DPDK expérimentales.

45.1. INSTALLATION DU PAQUET DPDK

Pour utiliser DPDK, installez le paquetage **dpdk**.

Procédure

- Utilisez l'utilitaire **dnf** pour installer le paquet **dpdk**:

```
# dnf install dpdk
```

45.2. RESSOURCES SUPPLÉMENTAIRES

- [Matrice de prise en charge des caractéristiques des adaptateurs réseau à chemin de données rapide](#)

CHAPITRE 46. COMPRENDRE LES FONCTIONNALITÉS DE MISE EN RÉSEAU EBPF DANS RHEL 9

Le filtre de paquets Berkeley étendu (eBPF) est une machine virtuelle intégrée au noyau qui permet l'exécution de code dans l'espace du noyau. Ce code s'exécute dans un environnement restreint (sandbox) qui n'a accès qu'à un ensemble limité de fonctions.

Dans le domaine des réseaux, vous pouvez utiliser eBPF pour compléter ou remplacer le traitement des paquets du noyau. Selon le crochet que vous utilisez, les programmes eBPF ont, par exemple, les caractéristiques suivantes

- Accès en lecture et en écriture aux données et métadonnées des paquets
- Peut consulter les sockets et les routes
- Peut définir les options de la prise
- Peut rediriger les paquets

46.1. VUE D'ENSEMBLE DES FONCTIONNALITÉS DE MISE EN RÉSEAU EBPF DANS RHEL 9

Vous pouvez attacher des programmes de mise en réseau Berkeley Packet Filter (eBPF) étendus aux crochets suivants dans RHEL :

- **eXpress Data Path (XDP)** : Fournit un accès anticipé aux paquets reçus avant que la pile réseau du noyau ne les traite.
- **tc** classificateur eBPF avec drapeau d'action directe : Fournit un traitement puissant des paquets à l'entrée et à la sortie.
- **Groupes de contrôle version 2 (cgroup v2)** : Permet de filtrer et d'outrepasser les opérations basées sur les sockets effectuées par les programmes d'un groupe de contrôle.
- **Filtrage des sockets** : Permet de filtrer les paquets reçus des sockets. Cette fonction était également disponible dans le filtre de paquets Berkeley classique (cBPF), mais elle a été étendue pour prendre en charge les programmes eBPF.
- **Analyseur de flux** : Permet de diviser les flux en messages individuels, de les filtrer et de les rediriger vers des sockets.
- **SO_REUSEPORT** sélection de la prise : Permet la sélection programmable d'une prise de réception à partir d'un groupe de prises **reuseport**.
- **Dissecteur de flux** : Permet d'outrepasser la manière dont le noyau analyse les en-têtes de paquets dans certaines situations.
- **Rappels de contrôle de congestion TCP** : Permet de mettre en œuvre un algorithme de contrôle de congestion TCP personnalisé.
- **Routes avec encapsulation** : Permet de créer une encapsulation de tunnel personnalisée.

XDP

Vous pouvez attacher des programmes du type **BPF_PROG_TYPE_XDP** à une interface réseau. Le noyau exécute alors le programme sur les paquets reçus avant que la pile réseau du noyau ne commence

à les traiter. Cela permet une transmission rapide des paquets dans certaines situations, telles que l'abandon rapide de paquets pour prévenir les attaques par déni de service distribué (DDoS) et les redirections rapides de paquets pour les scénarios d'équilibrage de la charge.

Vous pouvez également utiliser XDP pour différentes formes de surveillance et d'échantillonnage des paquets. Le noyau permet aux programmes XDP de modifier les paquets et de les transmettre pour traitement ultérieur à la pile réseau du noyau.

Les modes XDP suivants sont disponibles :

- **XDP natif (pilote) :** Le noyau exécute le programme à partir du point le plus précoce possible pendant la réception du paquet. À ce moment, le noyau n'a pas analysé le paquet et, par conséquent, aucune métadonnée fournie par le noyau n'est disponible. Ce mode nécessite que le pilote de l'interface réseau prenne en charge XDP, mais tous les pilotes ne prennent pas en charge ce mode natif.
- **XDP générique :** La pile réseau du noyau exécute le programme XDP au début du traitement. À ce moment-là, les structures de données du noyau ont été allouées et le paquet a été prétraité. Si un paquet doit être abandonné ou redirigé, cela nécessite un surcoût important par rapport au mode natif. Cependant, le mode générique ne nécessite pas de pilote d'interface réseau et fonctionne avec toutes les interfaces réseau.
- **XDP déchargé :** Le noyau exécute le programme XDP sur l'interface réseau et non sur l'unité centrale de l'hôte. Notez que cela nécessite un matériel spécifique et que seules certaines fonctionnalités de l'eBPF sont disponibles dans ce mode.

Sur RHEL, chargez tous les programmes XDP à l'aide de la bibliothèque **libxdp**. Cette bibliothèque permet de contrôler l'utilisation de XDP par le système.



NOTE

Actuellement, il existe certaines limitations de configuration du système pour les programmes XDP. Par exemple, vous devez désactiver certaines fonctions de décharge matérielle sur l'interface de réception. En outre, toutes les fonctionnalités ne sont pas disponibles avec tous les pilotes qui prennent en charge le mode natif.

Dans RHEL 9, Red Hat prend en charge les fonctionnalités XDP uniquement si vous utilisez la bibliothèque **libxdp** pour charger le programme dans le noyau.

AF_XDP

En utilisant un programme XDP qui filtre et redirige les paquets vers une socket **AF_XDP** donnée, vous pouvez utiliser une ou plusieurs sockets de la famille de protocoles **AF_XDP** pour copier rapidement des paquets du noyau vers l'espace utilisateur.

Contrôle du trafic

Le sous-système "Contrôle du trafic" (**tc**) propose les types de programmes eBPF suivants :

- **BPF_PROG_TYPE_SCHED_CLS**
- **BPF_PROG_TYPE_SCHED_ACT**

Ces types vous permettent d'écrire des classificateurs **tc** et des actions **tc** personnalisés dans eBPF. Avec les éléments de l'écosystème **tc**, ils permettent un traitement puissant des paquets et sont au cœur de plusieurs solutions d'orchestration de réseaux de conteneurs.

Dans la plupart des cas, seul le classificateur est utilisé, car avec l'indicateur d'action directe, le classificateur eBPF peut exécuter des actions directement à partir du même programme eBPF. La discipline de mise en file d'attente **clsact (qdisc)** a été conçue pour permettre cela du côté de l'entrée.

Il convient de noter que l'utilisation d'un programme eBPF de dissecteur de flux peut influencer le fonctionnement de certains autres classificateurs **qdiscs** et **tc**, tels que **flower**.

Filtre à douille

Plusieurs utilitaires utilisent ou ont utilisé le filtre de paquets classique de Berkeley (cBPF) pour filtrer les paquets reçus sur une socket. Par exemple, l'utilitaire **tcpdump** permet à l'utilisateur de spécifier des expressions, que **tcpdump** traduit ensuite en code cBPF.

Comme alternative au cBPF, le noyau autorise les programmes eBPF du type **BPF_PROG_TYPE_SOCKET_FILTER** dans le même but.

Groupes de contrôle

Dans RHEL, vous pouvez utiliser plusieurs types de programmes eBPF que vous pouvez attacher à un cgroup. Le noyau exécute ces programmes lorsqu'un programme du cgroup donné effectue une opération. Notez que vous ne pouvez utiliser que les cgroups version 2.

Les programmes cgroup eBPF suivants, liés à la mise en réseau, sont disponibles dans RHEL :

- **BPF_PROG_TYPE SOCK_OPS**: Le noyau appelle ce programme lors de divers événements TCP. Le programme peut ajuster le comportement de la pile TCP du noyau, y compris les options d'en-tête TCP personnalisées, etc.
- **BPF_PROG_TYPE CGROUP SOCK_ADDR**: Le noyau appelle ce programme pendant les opérations **connect**, **bind**, **sendto**, **recvmsg**, **getpeername**, et **getsockname**. Ce programme permet de modifier les adresses IP et les ports. Ceci est utile lorsque vous implémentez la traduction d'adresse réseau (NAT) basée sur les sockets dans l'eBPF.
- **BPF_PROG_TYPE CGROUP SOCKOPT**: Le noyau appelle ce programme pendant les opérations **setsockopt** et **getsockopt** et permet de modifier les options.
- **BPF_PROG_TYPE CGROUP SOCK**: Le noyau appelle ce programme lors de la création d'une socket, de la libération d'une socket et de la liaison à des adresses. Vous pouvez utiliser ces programmes pour autoriser ou refuser l'opération, ou seulement pour inspecter la création de la socket pour les statistiques.
- **BPF_PROG_TYPE CGROUP SKB**: Ce programme filtre les paquets individuels à l'entrée et à la sortie, et peut accepter ou rejeter des paquets.
- **BPF_PROG_TYPE CGROUP SYSCTL**: Ce programme permet de filtrer l'accès aux contrôles du système (**sysctl**).

Analyseur de flux

Un analyseur de flux fonctionne sur un groupe de sockets qui sont ajoutés à une carte eBPF spéciale. Le programme eBPF traite ensuite les paquets que le noyau reçoit ou envoie sur ces sockets.

Les programmes eBPF d'analyseur de flux suivants sont disponibles dans RHEL :

- **BPF_PROG_TYPE SK_SKB**: Un programme eBPF analyse les paquets reçus de la socket en messages individuels et demande au noyau d'abandonner ces messages ou de les envoyer à une autre socket du groupe.
- **BPF_PROG_TYPE SK_MSG**: Ce programme filtre les messages de sortie. Un programme eBPF analyse les paquets en messages individuels et les approuve ou les rejette.

Sélection du socket `SO_REUSEPORT`

Cette option permet de lier plusieurs sockets à la même adresse IP et au même port. Sans eBPF, le noyau sélectionne la socket de réception sur la base d'un hachage de connexion. Avec le programme `BPF_PROG_TYPE_SK_REUSEPORT`, la sélection de la socket de réception est entièrement programmable.

Dissecteur de flux

Lorsque le noyau doit traiter les en-têtes de paquets sans passer par le décodage complet du protocole, ils sont **dissected**. Cela se produit par exemple dans le sous-système `tc`, dans le routage par trajets multiples, dans la liaison ou lors du calcul du hachage d'un paquet. Dans cette situation, le noyau analyse les en-têtes de paquets et remplit les structures internes avec les informations des en-têtes de paquets. Vous pouvez remplacer cette analyse interne à l'aide du programme `BPF_PROG_TYPE_FLOW_DISSECTOR`. Notez que vous ne pouvez disséquer que TCP et UDP sur IPv4 et IPv6 dans eBPF sous RHEL.

Contrôle de congestion TCP

Vous pouvez écrire un algorithme de contrôle de congestion TCP personnalisé en utilisant un groupe de programmes `BPF_PROG_TYPE_STRUCT_OPS` qui implémentent les rappels `struct tcp_congestion_ops`. Un algorithme implémenté de cette manière est disponible pour le système en même temps que les algorithmes intégrés du noyau.

Routes avec encapsulation

Vous pouvez attacher l'un des types de programme eBPF suivants aux itinéraires de la table de routage en tant qu'attribut d'encapsulation du tunnel :

- `BPF_PROG_TYPE_LWT_IN`
- `BPF_PROG_TYPE_LWT_OUT`
- `BPF_PROG_TYPE_LWT_XMIT`

La fonctionnalité d'un tel programme eBPF est limitée à des configurations de tunnel spécifiques et ne permet pas de créer une solution générique d'encapsulation ou de décapsulation.

Recherche de sockets

Pour contourner les limitations de l'appel système `bind`, utilisez un programme eBPF de type `BPF_PROG_TYPE_SK_LOOKUP`. Ces programmes peuvent sélectionner un socket d'écoute pour les nouvelles connexions TCP entrantes ou un socket non connecté pour les paquets UDP.

46.2. VUE D'ENSEMBLE DES FONCTIONNALITÉS XDP DANS RHEL 9 PAR CARTES RÉSEAU

Voici un aperçu des cartes réseau compatibles XDP et des fonctions XDP que vous pouvez utiliser avec elles :

Carte réseau	Conducteur	De base	Redirection	Cible	Déchargement du matériel	Copie zéro	Grand MTU
Adaptateur de réseau élastique Amazon	ena	yes	yes	oui [a]	non	non	non

Carte réseau	Conducteur	De base	Redirection	Cible	Déchargement du matériel	Copie zéro	Grand MTU
Broadcom NetXtreme-C/E 10/25/40/50 gigabit Ethernet	bnxt_en	yes	yes	oui [a]	non	non	yes
Fonction virtuelle Cavium Thunder	nicvf	yes	non	non	non	non	non
Intel® 10GbE PCI Express Virtual Function Ethernet	ixgbev	yes	non	non	non	non	non
Adaptateurs Intel® 10GbE PCI Express	ixgbe	yes	yes	oui [a]	non	yes	non
Connexion Ethernet Intel® série E800	ice	yes	yes	oui [a]	non	yes	non
Famille Intel® Ethernet Controller I225-LM/I225-V	igc	yes	yes	yes	non	yes	non
Adaptateurs Intel® PCI Express Gigabit	igb	yes	yes	oui [a]	non	non	non
Contrôleur Ethernet Intel® de la famille XL710	i40e	yes	yes	oui [a] [b]	non	yes	non
Marvell OcteonTX2	rvu_nicpf	yes	yes	oui [a] [b]	non	non	non
Adaptateurs réseau Mellanox de 5ème génération (série ConnectX)	mlx5_core	yes	yes	oui [b]	non	yes	yes
Mellanox Technologies 1/10/40Gbit Ethernet	mlx4_en	yes	yes	non	non	non	non
Adaptateur réseau Microsoft Azure	mana	yes	yes	yes	non	non	non
Réseau virtuel Microsoft Hyper-V	hv_netvsc	yes	yes	yes	non	non	non
Carte réseau Netronome® NFP4000/NFP6000 [c]	nfp	yes	non	non	yes	yes	non
Réseau QEMU Virtio	virtio_net	yes	yes	oui [a]	non	non	non

Carte réseau	Conducteur	De base	Redirection	Cible	Déchargement du matériel	Copie zéro	Grand MTU
Carte réseau Ethernet QLogic QED 25/40/100Gb	qede	yes	yes	yes	non	non	non
STMicroelectronics Multi-Gigabit Ethernet	stmmac	yes	yes	yes	non	yes	non
Solarflare SFC9000/SFC9100/EF100-family	sfc	yes	yes	oui [b]	non	non	non
Dispositif universel TUN/TAP	tun	yes	yes	yes	non	non	non
Dispositif de paires virtuelles Ethernet	veth	yes	yes	yes	non	non	yes
Dispositif de réseau paravirtuel Xen	xen-netfront	yes	yes	yes	non	non	non

[a] Uniquement si un programme XDP est chargé sur l'interface.

[b] Nécessite l'allocation de plusieurs files d'attente XDP TX d'une taille supérieure ou égale à l'index le plus important du CPU.

[c] Certaines des fonctions énumérées ne sont pas disponibles pour la carte d'interface réseau Netronome® NFP3800.

Légende :

- Basic : prend en charge les codes de retour de base : **DROP, PASS, ABORTED**, et **TX**.
- Redirect : prend en charge le code de retour **XDP_REDIRECT**.
- Cible : Peut être la cible d'un code de retour **XDP_REDIRECT**.
- HW offload : Prend en charge la décharge matérielle XDP.
- Zero-copy : Prend en charge le mode zéro-copie pour la famille de protocoles **AF_XDP**.
- Large MTU : Prend en charge les paquets plus grands que la taille de la page. Notez que sur Red Hat Enterprise Linux 9.2, vous ne pouvez pas utiliser de gros paquets avec le code de retour **XDP_REDIRECT**.

CHAPITRE 47. TRAÇAGE DE RÉSEAUX À L'AIDE DE LA COLLECTION DE COMPILATEURS BPF

BPF Compiler Collection (BCC) est une bibliothèque qui facilite la création de programmes Berkeley Packet Filter étendus (eBPF). La principale utilité des programmes eBPF est d'analyser les performances du système d'exploitation et du réseau sans subir de surcharge ou de problèmes de sécurité.

BCC évite aux utilisateurs de devoir connaître les détails techniques de l'eBPF et fournit de nombreux points de départ prêts à l'emploi, tels que le paquet **bcc-tools** avec des programmes eBPF pré-crés.



NOTE

Les programmes eBPF sont déclenchés par des événements tels que des entrées/sorties de disque, des connexions TCP et des créations de processus. Il est peu probable que les programmes provoquent un crash du noyau, une boucle ou une absence de réponse, car ils s'exécutent dans une machine virtuelle sûre au sein du noyau.

47.1. INSTALLATION DU PAQUETAGE BCC-TOOLS

Installez le paquetage **bcc-tools**, qui installe également la bibliothèque BPF Compiler Collection (BCC) en tant que dépendance.

Procédure

1. Installer **bcc-tools**.

```
# dnf install bcc-tools
```

Les outils BCC sont installés dans le répertoire **/usr/share/bcc/tools/**.

2. Optionnellement, inspecter les outils :

```
# ll /usr/share/bcc/tools/
...
-rwxr-xr-x. 1 root root 4198 Dec 14 17:53 dcsnoop
-rwxr-xr-x. 1 root root 3931 Dec 14 17:53 dcstat
-rwxr-xr-x. 1 root root 20040 Dec 14 17:53 deadlock_detector
-rw-r--r--. 1 root root 7105 Dec 14 17:53 deadlock_detector.c
drwxr-xr-x. 3 root root 8192 Mar 11 10:28 doc
-rwxr-xr-x. 1 root root 7588 Dec 14 17:53 execsnoop
-rwxr-xr-x. 1 root root 6373 Dec 14 17:53 ext4dist
-rwxr-xr-x. 1 root root 10401 Dec 14 17:53 ext4slower
...
```

Le répertoire **doc** de la liste ci-dessus contient la documentation de chaque outil.

47.2. AFFICHAGE DES CONNEXIONS TCP AJOUTÉES À LA FILE D'ATTENTE D'ACCEPTATION DU NOYAU

Après avoir reçu le paquet **ACK** dans le cadre d'une poignée de main TCP à trois voies, le noyau déplace la connexion de la file d'attente **SYN** vers la file d'attente **accept** après que l'état de la connexion est passé à **ESTABLISHED**. Par conséquent, seules les connexions TCP réussies sont visibles dans cette

file d'attente.

L'utilitaire **tcpaccept** utilise les fonctionnalités de l'eBPF pour afficher toutes les connexions que le noyau ajoute à la file d'attente **accept**. L'utilitaire est léger car il retrace la fonction **accept()** du noyau au lieu de capturer des paquets et de les filtrer. Par exemple, utilisez **tcpaccept** pour le dépannage général afin d'afficher les nouvelles connexions acceptées par le serveur.

Procédure

1. Saisissez la commande suivante pour lancer l'analyse de la file d'attente du noyau **accept**:

```
# /usr/share/bcc/tools/tcpaccept
PID COMM IP RADDR RPORT LADDR LPORT
843 sshd 4 192.0.2.17 50598 192.0.2.1 22
1107 ns-slapd 4 198.51.100.6 38772 192.0.2.1 389
1107 ns-slapd 4 203.0.113.85 38774 192.0.2.1 389
...
```

Chaque fois que le noyau accepte une connexion, **tcpaccept** affiche les détails des connexions.

2. Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcpaccept(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcpaccept_example.txt` fichier

47.3. TRAÇAGE DES TENTATIVES DE CONNEXION TCP SORTANTES

L'utilitaire **tcpconnect** utilise les fonctionnalités de l'eBPF pour tracer les tentatives de connexion TCP sortantes. La sortie de l'utilitaire comprend également les connexions qui ont échoué.

L'utilitaire **tcpconnect** est léger car il trace, par exemple, la fonction **connect()** du noyau au lieu de capturer des paquets et de les filtrer.

Procédure

1. Entrez la commande suivante pour lancer le processus de traçage qui affiche toutes les connexions sortantes :

```
# /usr/share/bcc/tools/tcpconnect
PID COMM IP SADDR DADDR DPORT
31346 curl 4 192.0.2.1 198.51.100.16 80
31348 telnet 4 192.0.2.1 203.0.113.231 23
31361 isc-worker00 4 192.0.2.1 192.0.2.254 53
...
```

Chaque fois que le noyau traite une connexion sortante, **tcpconnect** affiche les détails des connexions.

2. Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcpconnect(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcpconnect_example.txt` fichier

47.4. MESURE DE LA LATENCE DES CONNEXIONS TCP SORTANTES

La latence d'une connexion TCP est le temps nécessaire pour établir une connexion. Il s'agit généralement du temps de traitement TCP/IP du noyau et du temps d'aller-retour sur le réseau, et non du temps d'exécution de l'application.

L'utilitaire **tcpconlat** utilise les fonctions eBPF pour mesurer le temps écoulé entre l'envoi d'un paquet **SYN** et la réception du paquet de réponse.

Procédure

1. Commencez à mesurer la latence des connexions sortantes :

```
# /usr/share/bcc/tools/tcpconlat
PID COMM      IP SADDR  DADDR      DPORT LAT(ms)
32151 isc-worker00 4 192.0.2.1 192.0.2.254 53 0.60
32155 ssh        4 192.0.2.1 203.0.113.190 22 26.34
32319 curl      4 192.0.2.1 198.51.100.59 443 188.96
...
```

Chaque fois que le noyau traite une connexion sortante, **tcpconlat** affiche les détails de la connexion après que le noyau a reçu le paquet de réponses.

2. Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcpconlat(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcpconlat_example.txt` fichier

47.5. AFFICHAGE DES DÉTAILS CONCERNANT LES PAQUETS ET LES SEGMENTS TCP QUI ONT ÉTÉ ABANDONNÉS PAR LE NOYAU

L'utilitaire **tcpdrop** permet aux administrateurs d'afficher des détails sur les paquets et les segments TCP qui ont été abandonnés par le noyau. Utilisez cet utilitaire pour déboguer les taux élevés de paquets abandonnés qui peuvent amener le système distant à envoyer des retransmissions temporisées. Des taux élevés de paquets et de segments abandonnés peuvent avoir un impact sur les performances d'un serveur.

Au lieu de capturer et de filtrer les paquets, ce qui demande beaucoup de ressources, l'utilitaire **tcpdrop** utilise les fonctions eBPF pour récupérer les informations directement dans le noyau.

Procédure

1. Entrez la commande suivante pour commencer à afficher les détails des paquets et des segments TCP abandonnés :

```
# /usr/share/bcc/tools/tcpdrop
TIME  PID  IP SADDR:SPORT  > DADDR:DPORT  STATE (FLAGS)
```

```

13:28:39 32253 4 192.0.2.85:51616 > 192.0.2.1:22 CLOSE_WAIT (FIN|ACK)
b'tcp_drop+0x1'
b'tcp_data_queue+0x2b9'
...

13:28:39 1 4 192.0.2.85:51616 > 192.0.2.1:22 CLOSE (ACK)
b'tcp_drop+0x1'
b'tcp_rcv_state_process+0xe2'
...

```

Chaque fois que le noyau abandonne des paquets et des segments TCP, **tcpdrop** affiche les détails de la connexion, y compris la trace de la pile du noyau qui a conduit à l'abandon du paquet.

2. Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcpdrop(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcpdrop_example.txt` fichier

47.6. TRAÇAGE DES SESSIONS TCP

L'utilitaire **tcplife** utilise eBPF pour suivre les sessions TCP qui s'ouvrent et se ferment, et imprime une ligne de sortie pour résumer chacune d'entre elles. Les administrateurs peuvent utiliser **tcplife** pour identifier les connexions et le volume de trafic transféré.

Par exemple, vous pouvez afficher les connexions au port **22** (SSH) pour obtenir les informations suivantes :

- L'ID du processus local (PID)
- Le nom du processus local
- L'adresse IP locale et le numéro de port
- L'adresse IP et le numéro de port distants
- Quantité de trafic reçue et transmise en Ko.
- Durée en millisecondes pendant laquelle la connexion a été active

Procédure

1. Entrez la commande suivante pour lancer le suivi des connexions au port local **22**:

```

/usr/share/bcc/tools/tcplife -L 22
PID COMM  LADDR  LPORT RADDR  RPORT TX_KB RX_KB  MS
19392 sshd  192.0.2.1 22 192.0.2.17 43892 53 52 6681.95
19431 sshd  192.0.2.1 22 192.0.2.245 43902 81 249381 7585.09
19487 sshd  192.0.2.1 22 192.0.2.121 43970 6998 7 16740.35
...

```

Chaque fois qu'une connexion est fermée, **tcplife** affiche les détails des connexions.

- Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcplife(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcplife_example.txt` fichier

47.7. SUIVI DES RETRANSMISSIONS TCP

L'utilitaire **tcpretrans** affiche des détails sur les retransmissions TCP, tels que l'adresse IP locale et distante et le numéro de port, ainsi que l'état TCP au moment des retransmissions.

L'utilitaire utilise les fonctionnalités de l'eBPF et, par conséquent, son coût est très faible.

Procédure

- Utilisez la commande suivante pour commencer à afficher les détails de la retransmission TCP :

```
# /usr/share/bcc/tools/tcpretrans
TIME  PID IP LADDR:LPORT  T> RADDR:RPORT  STATE
00:23:02 0  4 192.0.2.1:22  R> 198.51.100.0:26788 ESTABLISHED
00:23:02 0  4 192.0.2.1:22  R> 198.51.100.0:26788 ESTABLISHED
00:45:43 0  4 192.0.2.1:22  R> 198.51.100.0:17634 ESTABLISHED
...
```

Chaque fois que le noyau appelle la fonction de retransmission TCP, **tcpretrans** affiche les détails de la connexion.

- Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcpretrans(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcpretrans_example.txt` fichier

47.8. AFFICHAGE DES INFORMATIONS SUR LES CHANGEMENTS D'ÉTAT DE TCP

Au cours d'une session TCP, l'état TCP change. L'utilitaire **tcpstates** utilise les fonctions eBPF pour suivre ces changements d'état et affiche les détails, y compris la durée de chaque état. Par exemple, utilisez **tcpstates** pour identifier si les connexions passent trop de temps dans l'état d'initialisation.

Procédure

- Utilisez la commande suivante pour commencer à suivre les changements d'état de TCP :

```
# /usr/share/bcc/tools/tcpstates
SKADDR      C-PID C-COMM  LADDR  LPORT RADDR  RPORT OLDSTATE  ->
NEWSTATE  MS
fff9cd377b3af80 0  swapper/1 0.0.0.0  22  0.0.0.0  0  LISTEN  -> SYN_RECV
0.000
fff9cd377b3af80 0  swapper/1 192.0.2.1 22  192.0.2.45 53152 SYN_RECV  ->
```

```

ESTABLISHED 0.067
ffff9cd377b3af80 818  sssd_nss  192.0.2.1 22  192.0.2.45 53152 ESTABLISHED ->
CLOSE_WAIT 65636.773
ffff9cd377b3af80 1432 sshd    192.0.2.1 22  192.0.2.45 53152 CLOSE_WAIT ->
LAST_ACK 24.409
ffff9cd377b3af80 1267 pulseaudio 192.0.2.1 22  192.0.2.45 53152 LAST_ACK ->
CLOSE 0.376
...

```

Chaque fois qu'une connexion change d'état, **tcpstates** affiche une nouvelle ligne avec les détails actualisés de la connexion.

Si plusieurs connexions changent d'état en même temps, utilisez l'adresse du socket dans la première colonne (**SKADDR**) pour déterminer quelles entrées appartiennent à la même connexion.

2. Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcpstates(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcpstates_example.txt` fichier

47.9. RÉSUMER ET AGRÉGER LE TRAFIC TCP ENVOYÉ À DES SOUS-RÉSEAUX SPÉCIFIQUES

L'utilitaire **tcpsubnet** résume et regroupe le trafic TCP IPv4 que l'hôte local envoie aux sous-réseaux et affiche les résultats à un intervalle fixe. L'utilitaire utilise les fonctions eBPF pour collecter et résumer les données afin de réduire la charge de travail.

Par défaut, **tcpsubnet** résume le trafic pour les sous-réseaux suivants :

- **127.0.0.1/32**
- **10.0.0.0/8**
- **172.16.0.0/12**
- **192.0.2.0/24/16**
- **0.0.0.0/0**

Notez que le dernier sous-réseau (**0.0.0.0/0**) est une option fourre-tout. L'utilitaire **tcpsubnet** comptabilise tout le trafic pour les sous-réseaux différents des quatre premiers dans cette entrée fourre-tout.

Suivez la procédure pour compter le trafic pour les sous-réseaux **192.0.2.0/24** et **198.51.100.0/24**. Le trafic vers d'autres sous-réseaux sera suivi dans l'entrée du sous-réseau fourre-tout **0.0.0.0/0**.

Procédure

1. Commencez à surveiller le volume de trafic envoyé aux sous-réseaux **192.0.2.0/24**, **198.51.100.0/24**, et autres :

```
# /usr/share/bcc/tools/tcpsubnet 192.0.2.0/24,198.51.100.0/24,0.0.0.0/0
Tracing... Output every 1 secs. Hit Ctrl-C to end
[02/21/20 10:04:50]
192.0.2.0/24      856
198.51.100.0/24  7467
[02/21/20 10:04:51]
192.0.2.0/24      1200
198.51.100.0/24  8763
0.0.0.0/0        673
...
```

Cette commande affiche le trafic en octets pour les sous-réseaux spécifiés, une fois par seconde.

- Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcpsubnet(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcpsubnet.txt` fichier

47.10. AFFICHAGE DU DÉBIT DU RÉSEAU PAR ADRESSE IP ET PAR PORT

L'utilitaire **tcptop** affiche le trafic TCP que l'hôte envoie et reçoit en kilo-octets. Le rapport est automatiquement actualisé et ne contient que les connexions TCP actives. L'utilitaire utilise les fonctionnalités de l'eBPF et n'a donc qu'une très faible surcharge.

Procédure

- Pour surveiller le trafic envoyé et reçu, entrez :

```
# /usr/share/bcc/tools/tcptop
13:46:29 loadavg: 0.10 0.03 0.01 1/215 3875

PID  COMM      LADDR      RADDR      RX_KB  TX_KB
3853 3853      192.0.2.1:22 192.0.2.165:41838 32    102626
1285 sshd      192.0.2.1:22 192.0.2.45:39240 0      0
...
```

La sortie de la commande ne comprend que les connexions TCP actives. Si le système local ou distant ferme une connexion, celle-ci n'est plus visible dans la sortie.

- Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcptop(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcptop.txt` fichier

47.11. TRAÇAGE DES CONNEXIONS TCP ÉTABLIES

L'utilitaire **tcptracer** trace les fonctions du noyau qui connectent, acceptent et ferment les connexions TCP. L'utilitaire utilise les fonctionnalités de l'eBPF et, par conséquent, a un très faible surcoût.

Procédure

1. Utilisez la commande suivante pour lancer le processus de traçage :

```
# /usr/share/bcc/tools/tcptracer
Tracing TCP established connections. Ctrl-C to end.
T PID  COMM   IP SADDR  DADDR  SPORT DPORT
A 1088 ns-slapd 4 192.0.2.153 192.0.2.1 0 65535
A 845  sshd   4 192.0.2.1 192.0.2.67 22 42302
X 4502 sshd   4 192.0.2.1 192.0.2.67 22 42302
...
```

Chaque fois que le noyau se connecte, accepte ou ferme une connexion, **tcptracer** affiche les détails des connexions.

2. Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **tcptracer(8)** page de manuel
- `/usr/share/bcc/tools/doc/tcptracer_example.txt` fichier

47.12. TRAÇAGE DES TENTATIVES D'ÉCOUTE IPV4 ET IPV6

L'utilitaire **solisten** retrace toutes les tentatives d'écoute IPv4 et IPv6. Il trace les tentatives d'écoute, y compris celles qui échouent ou le programme d'écoute qui n'accepte pas la connexion. L'utilitaire trace la fonction que le noyau appelle lorsqu'un programme veut écouter les connexions TCP.

Procédure

1. Entrez la commande suivante pour lancer le processus de traçage qui affiche toutes les tentatives d'écoute TCP :

```
# /usr/share/bcc/tools/solisten
PID  COMM      PROTO  BACKLOG  PORT  ADDR
3643 nc        TCPv4   1        4242  0.0.0.0
3659 nc        TCPv6   1        4242  2001:db8:1::1
4221 redis-server TCPv6   128     6379  ::
4221 redis-server TCPv4   128     6379  0.0.0.0
....
```

2. Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **solisten(9)** page de manuel
- `/usr/share/bcc/tools/doc/solisten_example.txt` fichier

47.13. RÉSUMÉ DU TEMPS DE SERVICE DES INTERRUPTIONS DOUCES

L'utilitaire **softirqs** résume le temps passé à gérer les interruptions logicielles (soft IRQ) et affiche ce temps sous forme de totaux ou d'histogrammes. Cet utilitaire utilise les tracepoints du noyau **irq:softirq_enter** et **irq:softirq_exit**, qui est un mécanisme de traçage stable.

Procédure

1. Entrez la commande suivante pour lancer le traçage **soft irq** event time :

```
# /usr/share/bcc/tools/softirqs
Tracing soft irq event time... Hit Ctrl-C to end.
^C
SOFTIRQ      TOTAL_usec
tasklet      166
block        9152
net_rx       12829
rcu          53140
sched        182360
timer        306256
```

2. Appuyez sur **Ctrl C** pour arrêter le processus de traçage.

Ressources supplémentaires

- **softirqs(8)** page de manuel
- **/usr/share/bcc/tools/doc/softirqs_example.txt** fichier
- **mpstat(1)** page de manuel

47.14. RÉSUMÉ DE LA TAILLE ET DU NOMBRE DE PAQUETS SUR UNE INTERFACE RÉSEAU

L'utilitaire **netqtop** affiche des statistiques sur les attributs des paquets reçus (RX) et transmis (TX) sur chaque file d'attente réseau d'une interface réseau particulière. Les statistiques comprennent :

- Octets par seconde (BPS)
- Paquets par seconde (PPS)
- La taille moyenne des paquets
- Nombre total de paquets

Pour générer ces statistiques, **netqtop** trace les fonctions du noyau qui réalisent les événements des paquets transmis **net_dev_start_xmit** et des paquets reçus **netif_receive_skb**.

Procédure

1. Affiche le nombre de paquets dans la plage de taille d'octets de l'intervalle de temps de **2** secondes :

```
# /usr/share/bcc/tools/netqtop -n enp1s0 -i 2
```

```

Fri Jan 31 18:08:55 2023
TX
QueueID avg_size [0, 64) [64, 512) [512, 2K) [2K, 16K) [16K, 64K)
0 0 0 0 0 0 0
Total 0 0 0 0 0 0

RX
QueueID avg_size [0, 64) [64, 512) [512, 2K) [2K, 16K) [16K, 64K)
0 38.0 1 0 0 0 0
Total 38.0 1 0 0 0 0
-----

Fri Jan 31 18:08:57 2023
TX
QueueID avg_size [0, 64) [64, 512) [512, 2K) [2K, 16K) [16K, 64K)
0 0 0 0 0 0 0
Total 0 0 0 0 0 0

RX
QueueID avg_size [0, 64) [64, 512) [512, 2K) [2K, 16K) [16K, 64K)
0 38.0 1 0 0 0 0
Total 38.0 1 0 0 0 0
-----

```

- Appuyer sur **Ctrl+C** pour arrêter **netqtop**.

Ressources supplémentaires

- **netqtop(8)** page de manuel
- `/usr/share/bcc/tools/doc/netqtop_example.txt`

47.15. RESSOURCES SUPPLÉMENTAIRES

- `/usr/share/doc/bcc/README.md`

CHAPITRE 48. DÉMARRER AVEC LE TIPC

Transparent Inter-process Communication (TIPC), également connu sous le nom de **Cluster Domain Sockets**, est un service de communication interprocessus (IPC) pour les opérations à l'échelle d'une grappe.

Les applications qui s'exécutent dans un environnement de grappe dynamique et hautement disponible ont des besoins particuliers. Le nombre de nœuds dans une grappe peut varier, les routeurs peuvent tomber en panne et, pour des raisons d'équilibrage de la charge, les fonctionnalités peuvent être déplacées vers différents nœuds de la grappe. Le TIPC minimise l'effort des développeurs d'applications pour faire face à de telles situations et maximise les chances qu'elles soient traitées de manière correcte et optimale. En outre, le TIPC fournit une communication plus efficace et tolérante aux pannes que les protocoles généraux, tels que le TCP.

48.1. L'ARCHITECTURE DU TIPC

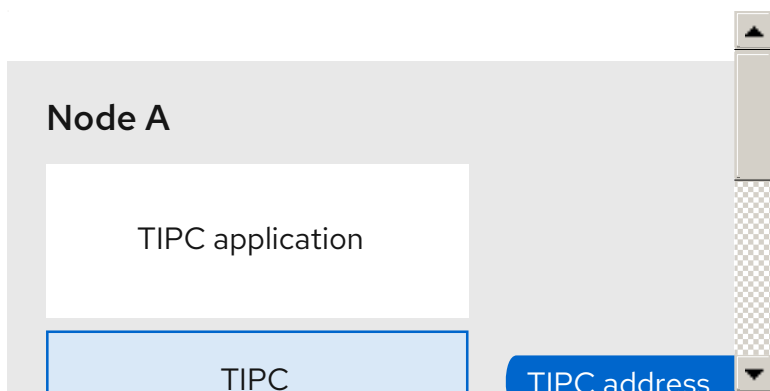
Le TIPC est une couche entre les applications utilisant le TIPC et un service de transport de paquets (**bearer**), et couvre le niveau des couches de transport, de réseau et de liaison de signalisation. Toutefois, le TIPC peut utiliser un protocole de transport différent comme support, de sorte que, par exemple, une connexion TCP peut servir de support à une liaison de signalisation TIPC.

Le TIPC prend en charge les supports suivants :

- Ethernet
- InfiniBand
- Protocole UDP

Le TIPC assure un transfert fiable des messages entre les ports du TIPC, qui sont les points d'arrivée de toutes les communications du TIPC.

Voici un schéma de l'architecture du TIPC :



48.2. CHARGEMENT DU MODULE TIPC AU DÉMARRAGE DU SYSTÈME

Avant de pouvoir utiliser le protocole TIPC, vous devez charger le module du noyau **tipc**. Vous pouvez configurer Red Hat Enterprise Linux pour qu'il charge automatiquement ce module de noyau lorsque le système démarre.

Procédure

1. Créez le fichier `/etc/modules-load.d/tipc.conf` avec le contenu suivant :

```
tipc
```

2. Redémarrez le service **systemd-modules-load** pour charger le module sans redémarrer le système :

```
# systemctl start systemd-modules-load
```

Vérification

1. Utilisez la commande suivante pour vérifier que RHEL a chargé le module **tipc**:

```
# lsmod | grep tipc
tipc 311296 0
```

Si la commande n'indique aucune entrée pour le module **tipc**, cela signifie que RHEL n'a pas réussi à le charger.

Ressources supplémentaires

- **modules-load.d(5)** page de manuel

48.3. CRÉER UN RÉSEAU TIPC

Pour créer un réseau TIPC, effectuez cette procédure sur chaque hôte qui doit rejoindre le réseau TIPC.



IMPORTANT

Les commandes ne configurent le réseau TIPC que temporairement. Pour configurer le TIPC de manière permanente sur un nœud, utilisez les commandes de cette procédure dans un script et configurez RHEL pour qu'il exécute ce script lorsque le système démarre.

Conditions préalables

- Le module **tipc** a été chargé. Pour plus de détails, voir [Chargement du module tipc au démarrage du système](#)

Procédure

1. Facultatif : Définissez une identité de nœud unique, telle qu'un UUID ou le nom d'hôte du nœud :

```
# tipc node set identity host_name
```

L'identité peut être une chaîne unique composée d'un maximum de 16 lettres et chiffres.

Vous ne pouvez pas définir ou modifier une identité après cette étape.

2. Ajouter un support. Par exemple, pour utiliser Ethernet comme support et le périphérique **enp0s1** comme périphérique de support physique, entrez :

```
# tipc bearer enable media eth device enp1s0
```

3. En option : Pour une redondance et de meilleures performances, attachez d'autres supports en utilisant la commande de l'étape précédente. Vous pouvez configurer jusqu'à trois supports, mais pas plus de deux sur le même média.
4. Répétez toutes les étapes précédentes sur chaque nœud qui doit rejoindre le réseau TIPC.

Vérification

1. Affiche l'état de la liaison pour les membres du cluster :

```
# tipc link list
broadcast-link: up
5254006b74be:enp1s0-525400df55d1:enp1s0: up
```

Cette sortie indique que le lien entre le support **enp1s0** sur le nœud **5254006b74be** et le support **enp1s0** sur le nœud **525400df55d1** est **up**.

2. Afficher la table de publication du TIPC :

```
# tipc nametable show
Type   Lower   Upper   Scope  Port   Node
0      1795222054 1795222054 cluster 0      5254006b74be
0      3741353223 3741353223 cluster 0      525400df55d1
1      1         1       node   2399405586 5254006b74be
2      3741353223 3741353223 node   0       5254006b74be
```

- Les deux entrées avec le type de service **0** indiquent que deux nœuds sont membres de cette grappe.
- L'entrée avec le type de service **1** représente le service de suivi du service de topologie intégré.
- L'entrée avec le type de service **2** affiche le lien tel qu'il est vu par le nœud émetteur. La limite de l'intervalle **3741353223** représente l'adresse du point d'extrémité du pair (une valeur de hachage unique de 32 bits basée sur l'identité du nœud) au format décimal.

Ressources supplémentaires

- **tipc-bearer(8)** page de manuel
- **tipc-namespace(8)** page de manuel

48.4. RESSOURCES SUPPLÉMENTAIRES

- Red Hat recommande d'utiliser d'autres protocoles de niveau support pour crypter la communication entre les nœuds en fonction du support de transport. Par exemple :
 - MACSec : Voir [Utilisation de MACsec pour crypter le trafic de la couche 2](#)
 - IPsec : Voir [Configuration d'un VPN avec IPsec](#)
- Pour obtenir des exemples d'utilisation du TIPC, clonez le dépôt GIT en amont à l'aide de la commande **git clone git://git.code.sf.net/p/tipc/tipcutils**. Ce dépôt contient le code source des démonstrations et des programmes de test qui utilisent les fonctionnalités du TIPC. Notez que ce dépôt n'est pas fourni par Red Hat.

- `/usr/share/doc/kernel-doc-<kernel_version>/Documentation/output/networking/tipc.html` fournie par le paquet **kernel-doc**.

CHAPITRE 49. CONFIGURATION AUTOMATIQUE DES INTERFACES RÉSEAU DANS LES NUAGES PUBLICS À L'AIDE DE NM-CLOUD-SETUP

Normalement, une machine virtuelle (VM) n'a qu'une seule interface configurable par DHCP. Cependant, certaines VM peuvent avoir plusieurs interfaces réseau, adresses IP et sous-réseaux IP sur une interface qui n'est pas configurable par DHCP. De plus, les administrateurs peuvent reconfigurer le réseau pendant que la machine est en cours d'exécution. L'utilitaire **nm-cloud-setup** récupère automatiquement les informations de configuration à partir du serveur de métadonnées du fournisseur de services en nuage et met à jour les configurations réseau des VM dans les nuages publics.

49.1. CONFIGURATION ET PRÉ-DÉPLOIEMENT DE NM-CLOUD-SETUP

Pour activer et configurer les interfaces réseau dans les clouds publics, exécutez **nm-cloud-setup** en tant que timer et service.



NOTE

Sur les images Red Hat Enterprise Linux On Demand et AWS golden, **nm-cloud-setup** est déjà activé et aucune action n'est requise.

Prérequis

- Une connexion réseau existe.
- La connexion utilise DHCP.
Par défaut, NetworkManager crée un profil de connexion qui utilise DHCP. Si aucun profil n'a été créé parce que vous avez défini le paramètre **no-auto-default** dans `/etc/NetworkManager/NetworkManager.conf`, créez cette connexion initiale manuellement.

Procédure

1. Installez le paquetage **nm-cloud-setup**:

```
# dnf install NetworkManager-cloud-setup
```

2. Créez et exécutez le fichier snap-in pour le service **nm-cloud-setup**:

- a. Utilisez la commande suivante pour commencer à éditer le fichier du snap-in :

```
# systemctl edit nm-cloud-setup.service
```

Il est important de démarrer explicitement le service ou de redémarrer le système pour que les paramètres de configuration soient pris en compte.

- b. Utilisez le fichier snap-in **systemd** pour configurer le fournisseur de cloud dans **nm-cloud-setup**. Par exemple, pour utiliser Amazon EC2, entrez :

```
[Service]
Environment=NM_CLOUD_SETUP_EC2=yes
```

Vous pouvez définir les variables d'environnement suivantes pour activer les services en nuage que vous utilisez :

- **NM_CLOUD_SETUP_AZURE** pour Microsoft Azure
- **NM_CLOUD_SETUP_EC2** pour Amazon EC2 (AWS)
- **NM_CLOUD_SETUP_GCP** pour Google Cloud Platform (GCP)
- **NM_CLOUD_SETUP_ALIYUN** pour Alibaba Cloud (Aliyun)

c. Enregistrez le fichier et quittez l'éditeur.

3. Recharger la configuration de **systemd**:

```
# systemctl daemon-reload
```

4. Activez et démarrez le service **nm-cloud-setup**:

```
# systemctl enable --now nm-cloud-setup.service
```

5. Active et démarre la minuterie **nm-cloud-setup**:

```
# systemctl enable --now nm-cloud-setup.timer
```

Ressources supplémentaires

- **nm-cloud-setup(8)** page de manuel
- [Configuration d'une connexion Ethernet](#)