



# Red Hat Enterprise Linux 9

## Configuration des réseaux InfiniBand et RDMA

Configuration et gestion des protocoles de réseau à haut débit et du matériel RDMA



# Red Hat Enterprise Linux 9 Configuration des réseaux InfiniBand et RDMA

---

Configuration et gestion des protocoles de réseau à haut débit et du matériel RDMA

## 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

Vous pouvez configurer et gérer les réseaux RDMA (Remote Directory Memory Access) et le matériel InfiniBand au niveau de l'entreprise en utilisant divers protocoles. Il s'agit notamment du protocole RDMA over Converged Ethernet (RoCE), de l'implémentation logicielle de RoCE (Soft-RoCE), du protocole de réseau IP tel que iWARP, de l'implémentation logicielle de iWARP (Soft-iWARP) et du protocole Network File System over RDMA (NFS over RDMA) en tant que support natif sur le matériel prenant en charge RDMA. Pour les connexions à faible latence et à haut débit, vous pouvez configurer IP over InfiniBand (IPoIB).

---

## Table des matières

<b>RENDRE L'OPEN SOURCE PLUS INCLUSIF</b> .....	<b>3</b>
<b>FOURNIR UN RETOUR D'INFORMATION SUR LA DOCUMENTATION DE RED HAT</b> .....	<b>4</b>
<b>CHAPITRE 1. COMPRENDRE INFINIBAND ET RDMA</b> .....	<b>5</b>
<b>CHAPITRE 2. CONFIGURATION DE SOFT-IWARP</b> .....	<b>6</b>
2.1. APERÇU DE L'IWARP ET DU SOFT-IWARP	6
2.2. CONFIGURATION DE SOFT-IWARP	6
<b>CHAPITRE 3. CONFIGURATION DE ROCE</b> .....	<b>8</b>
3.1. APERÇU DES VERSIONS DU PROTOCOLE ROCE	8
3.2. CHANGEMENT TEMPORAIRE DE LA VERSION PAR DÉFAUT DE ROCE	8
<b>CHAPITRE 4. CONFIGURATION DU SOUS-SYSTÈME RDMA PRINCIPAL</b> .....	<b>10</b>
4.1. RENOMMER LES PÉRIPHÉRIQUES IPOIB À L'AIDE DU FICHER LINK DE SYSTEMD	10
4.2. AUGMENTATION DE LA QUANTITÉ DE MÉMOIRE QUE LES UTILISATEURS SONT AUTORISÉS À UTILISER DANS LE SYSTÈME	11
4.3. ACTIVATION DE NFS SUR RDMA (NFSORDMA)	12
<b>CHAPITRE 5. CONFIGURATION D'UN GESTIONNAIRE DE SOUS-RÉSEAU INFINIBAND</b> .....	<b>13</b>
<b>CHAPITRE 6. CONFIGURATION D'IPOIB</b> .....	<b>14</b>
6.1. LES MODES DE COMMUNICATION IPOIB	14
6.2. COMPRENDRE LES ADRESSES MATÉRIELLES IPOIB	14
6.3. CONFIGURATION D'UNE CONNEXION IPOIB À L'AIDE DES COMMANDES NMCLI	15
6.4. CONFIGURATION D'UNE CONNEXION IPOIB À L'AIDE DU RÔLE DE RÉSEAU RHEL SYSTEM ROLE	16
6.5. CONFIGURATION D'UNE CONNEXION IPOIB À L'AIDE DE NM-CONNECTION-EDITOR	17
<b>CHAPITRE 7. TEST DES RÉSEAUX INFINIBAND</b> .....	<b>20</b>
7.1. TEST DES PREMIÈRES OPÉRATIONS RDMA INFINIBAND	20
7.2. TEST D'UN IPOIB À L'AIDE DE L'UTILITAIRE PING	22
7.3. TEST D'UN RÉSEAU RDMA AVEC IPERF3 APRÈS CONFIGURATION D'IPOIB	22



## 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. COMPRENDRE INFINIBAND ET RDMA

InfiniBand fait référence à deux choses distinctes :

- Protocole de la couche de liaison physique pour les réseaux InfiniBand
- L'API InfiniBand Verbs, une implémentation de la technologie RDMA (remote direct memory access)

RDMA permet l'accès entre la mémoire principale de deux ordinateurs sans impliquer de système d'exploitation, de cache ou de stockage. Grâce à RDMA, les données sont transférées avec un débit élevé, une faible latence et une faible utilisation de l'unité centrale.

Dans un transfert de données IP typique, lorsqu'une application sur une machine envoie des données à une application sur une autre machine, les actions suivantes se produisent au niveau de la réception :

1. Le noyau doit recevoir les données.
2. Le noyau doit déterminer que les données appartiennent à l'application.
3. Le noyau réveille l'application.
4. Le noyau attend que l'application effectue un appel système dans le noyau.
5. L'application copie les données de l'espace mémoire interne du noyau dans le tampon fourni par l'application.

Ce processus signifie que la majeure partie du trafic réseau est copiée à travers la mémoire principale du système si l'adaptateur hôte utilise l'accès direct à la mémoire (DMA) ou sinon au moins deux fois. En outre, l'ordinateur exécute certains changements de contexte pour passer du noyau à l'application. Ces changements de contexte peuvent entraîner une charge plus élevée du processeur avec des taux de trafic élevés tout en ralentissant les autres tâches.

Contrairement à la communication IP traditionnelle, la communication RDMA contourne l'intervention du noyau dans le processus de communication. Cela permet de réduire les frais généraux de l'unité centrale. Le protocole RDMA permet à l'adaptateur hôte de décider, après l'entrée d'un paquet dans le réseau, quelle application doit le recevoir et où le stocker dans l'espace mémoire de cette application. Au lieu d'envoyer le paquet pour traitement au noyau et de le copier dans la mémoire de l'application utilisateur, l'adaptateur hôte place directement le contenu du paquet dans la mémoire tampon de l'application. Ce processus nécessite une API distincte, l'API InfiniBand Verbs, et les applications doivent mettre en œuvre l'API InfiniBand Verbs pour utiliser RDMA.

Red Hat Enterprise Linux prend en charge à la fois le matériel InfiniBand et l'API InfiniBand Verbs. De plus, il prend en charge les technologies suivantes pour utiliser l'API InfiniBand Verbs sur du matériel non-InfiniBand :

- Internet Wide Area RDMA Protocol (iWARP) : protocole réseau qui met en œuvre RDMA sur les réseaux IP
- RDMA over Converged Ethernet (RoCE), également connu sous le nom d'InfiniBand over Ethernet (IBoE) : Protocole réseau qui met en œuvre le RDMA sur les réseaux Ethernet

## Ressources supplémentaires

- [Configuration de RoCE](#)

## CHAPITRE 2. CONFIGURATION DE SOFT-IWARP

L'accès direct à la mémoire à distance (RDMA) utilise plusieurs bibliothèques et protocoles sur Ethernet tels que iWARP, Soft-iWARP pour l'amélioration des performances et l'interface de programmation assistée.

### 2.1. APERÇU DE L'IWARP ET DU SOFT-IWARP

L'accès direct à la mémoire à distance (RDMA) utilise le protocole Internet Wide-area RDMA Protocol (iWARP) sur Ethernet pour une transmission de données convergente et à faible latence sur TCP. En utilisant des commutateurs Ethernet standard et la pile TCP/IP, iWARP achemine le trafic à travers les sous-réseaux IP. Cela permet d'utiliser efficacement l'infrastructure existante. Dans Red Hat Enterprise Linux, plusieurs fournisseurs implémentent l'iWARP dans leurs cartes d'interface réseau matérielles. Par exemple, **cxgb4**, **irdma**, **qedr** etc.

Soft-iWARP (siw) est un pilote de noyau iWARP basé sur un logiciel et une bibliothèque utilisateur pour Linux. Il s'agit d'un dispositif RDMA logiciel qui fournit une interface de programmation au matériel RDMA lorsqu'il est connecté à des cartes d'interface réseau. Il permet de tester et de valider facilement l'environnement RDMA.

### 2.2. CONFIGURATION DE SOFT-IWARP

Soft-iWARP (siw) met en œuvre le protocole Internet Wide-area RDMA Protocol (iWARP) Remote direct memory access (RDMA) transport over the Linux TCP/IP network stack. Il permet à un système doté d'un adaptateur Ethernet standard d'interopérer avec un adaptateur iWARP ou avec un autre système utilisant le pilote Soft-iWARP ou un hôte doté du matériel prenant en charge iWARP.



#### IMPORTANT

La fonctionnalité Soft-iWARP est fournie 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 Soft-iWARP, vous pouvez utiliser cette procédure dans un script à exécuter automatiquement au démarrage du système.

#### Conditions préalables

- Un adaptateur Ethernet est installé

#### Procédure

1. Installez les paquets **iproute**, **libibverbs**, **libibverbs-utils**, et **infiniband-diags**:

```
# dnf install iproute libibverbs libibverbs-utils infiniband-diags
```

- Afficher les liens RDMA :

```
# rdma link show
```

- Charger le module du noyau **siw**:

```
# modprobe siw
```

- Ajoutez un nouveau périphérique **siw** nommé **siw0** qui utilise l'interface **enp0s1**:

```
# rdma link add siw0 type siw netdev enp0s1
```

## Vérification

- Visualiser l'état de tous les liens RDMA :

```
# rdma link show
```

```
link siw0/1 state ACTIVE physical_state LINK_UP netdev enp0s1
```

- Répertorie les périphériques RDMA disponibles :

```
# ibv_devices
```

device	node GUID
-----	-----
siw0	0250b6fffea19d61

- Vous pouvez utiliser l'utilitaire **ibv\_devinfo** pour afficher un état détaillé :

```
# ibv_devinfo siw0
```

```
hca_id:      siw0
transport:   iWARP (1)
fw_ver:      0.0.0
node_guid:   0250:b6ff:fea1:9d61
sys_image_guid: 0250:b6ff:fea1:9d61
vendor_id:   0x626d74
vendor_part_id: 1
hw_ver:      0x0
phys_port_cnt: 1
port:        1
state:       PORT_ACTIVE (4)
max_mtu:     1024 (3)
active_mtu:  1024 (3)
sm_lid:      0
port_lid:    0
port_lmc:    0x00
link_layer:  Ethernet
```

## CHAPITRE 3. CONFIGURATION DE ROCE

L'accès direct à la mémoire à distance (RDMA) permet l'exécution à distance de l'accès direct à la mémoire (DMA). RDMA over Converged Ethernet (RoCE) est un protocole réseau qui utilise RDMA sur un réseau Ethernet. Pour la configuration, RoCE nécessite un matériel spécifique et certains des fournisseurs de matériel sont Mellanox, Broadcom et QLogic.

### 3.1. APERÇU DES VERSIONS DU PROTOCOLE ROCE

RoCE est un protocole réseau qui permet l'accès direct à la mémoire à distance (RDMA) sur Ethernet.

Les différentes versions de RoCE sont les suivantes :

#### RoCE v1

Le protocole RoCE version 1 est un protocole de couche de liaison Ethernet de type ether **0x8915** qui permet la communication entre deux hôtes dans le même domaine de diffusion Ethernet.

#### RoCE v2

Le protocole RoCE version 2 existe au dessus du protocole UDP over IPv4 ou UDP over IPv6. Pour RoCE v2, le numéro de port de destination UDP est **4791**.

Le RDMA\_CM établit une connexion fiable entre un client et un serveur pour le transfert de données. RDMA\_CM fournit une interface RDMA neutre en termes de transport pour établir des connexions. La communication utilise un dispositif RDMA spécifique et des transferts de données basés sur des messages.



#### IMPORTANT

L'utilisation de versions différentes, comme RoCE v2 sur le client et RoCE v1 sur le serveur, n'est pas prise en charge. Dans ce cas, configurez le serveur et le client pour qu'ils communiquent via RoCE v1.

RoCE v1 fonctionne au niveau de la couche de liaison de données (couche 2) et ne prend en charge que la communication de deux machines sur le même réseau. Par défaut, RoCE v2 est disponible. Il fonctionne au niveau de la couche réseau (couche 3). RoCE v2 prend en charge le routage des paquets, ce qui permet d'établir une connexion avec plusieurs réseaux Ethernet.

#### Ressources supplémentaires

- [Changement temporaire de la version par défaut de RoCE](#)

### 3.2. CHANGEMENT TEMPORAIRE DE LA VERSION PAR DÉFAUT DE ROCE

L'utilisation du protocole RoCE v2 sur le client et RoCE v1 sur le serveur n'est pas prise en charge. Si le matériel de votre serveur ne prend en charge que RoCE v1, configurez vos clients pour que RoCE v1 puisse communiquer avec le serveur. Par exemple, vous pouvez configurer un client qui utilise le pilote **mlx5\_0** pour le périphérique InfiniBand Mellanox ConnectX-5 qui ne prend en charge que RoCE v1.



#### NOTE

Les modifications décrites ici resteront effectives jusqu'à ce que vous redémarriez l'hôte.

## Conditions préalables

- Le client utilise un dispositif InfiniBand avec le protocole RoCE v2.
- Le serveur utilise un dispositif InfiniBand qui ne prend en charge que RoCE v1.

## Procédure

1. Créez le répertoire `/sys/kernel/config/rdma_cm/mlx5_0/` répertoire :

```
# mkdir /sys/kernel/config/rdma_cm/mlx5_0/
```

2. Affiche le mode RoCE par défaut :

```
# cat /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode
```

```
RoCE v2
```

3. Changez le mode RoCE par défaut en version 1 :

```
# echo "IB/RoCE v1" > /sys/kernel/config/rdma_cm/mlx5_0/ports/1/default_roce_mode
```

# CHAPITRE 4. CONFIGURATION DU SOUS-SYSTÈME RDMA PRINCIPAL

La configuration du service **rdma** gère les protocoles de réseau et les normes de communication telles que InfiniBand, iWARP et RoCE.

## 4.1. RENOMMER LES PÉRIPHÉRIQUES IPOIB À L'AIDE DU FICHIER LINK DE SYSTEMD

Par défaut, le noyau nomme les périphériques IPoIB (Internet Protocol over InfiniBand), par exemple **ib0**, **ib1**, etc. Pour éviter les conflits, créez un fichier de lien **systemd** pour créer des noms persistants et significatifs tels que **mlx4\_ib0**.

### Conditions préalables

- Vous avez installé un périphérique InfiniBand.

### Procédure

1. Affiche l'adresse matérielle de l'appareil **ib0**:

```
# ip addr show ib0

7: ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc fq_codel state UP
group default qlen 256
    link/infiniband 80:00:0a:28:fe:80:00:00:00:00:00:00:f4:52:14:03:00:7b:e1:b1 brd
00:ff:ff:ff:12:40:1b:ff:00:00:00:00:00:00:ff:ff:ff
    altname ibp7s0
    altname ibs2
    inet 172.31.0.181/24 brd 172.31.0.255 scope global dynamic noprefixroute ib0
        valid_lft 2899sec preferred_lft 2899sec
    inet6 fe80::f652:1403:7b:e1b1/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

2. Pour nommer l'interface avec l'adresse MAC **80:00:0a:28:fe:80:00:00:00:00:00:00:f4:52:14:03:00:7b:e1:b1** à **mlx4\_ib0**, créez le fichier **/etc/systemd/network/70-custom-ifnames.link** avec le contenu suivant :

```
[Match]
MACAddress=80:00:0a:28:fe:80:00:00:00:00:00:00:f4:52:14:03:00:7b:e1:b1

[Link]
Name=mlx4_ib0
```

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. Reboot the host:

```
# reboot
```

- Vérifiez que l'appareil portant l'adresse MAC que vous avez spécifiée dans le fichier de liaison a été attribué à **mlx4\_ib0**:

```
# ip addr show mlx4_ib0
```

```
7: mlx4_ib0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 65520 qdisc fq_codel state
UP group default qlen 256
    link/infiniband 80:00:0a:28:fe:80:00:00:00:00:00:00:00:00:f4:52:14:03:00:7b:e1:b1 brd
00:ff:ff:ff:12:40:1b:ff:ff:00:00:00:00:00:00:00:00:ff:ff:ff:ff
    altname ibp7s0
    altname ibs2
    inet 172.31.0.181/24 brd 172.31.0.255 scope global dynamic noprefixroute mlx4_ib0
        valid_lft 2899sec preferred_lft 2899sec
    inet6 fe80::f652:1403:7b:e1b1/64 scope link noprefixroute
        valid_lft forever preferred_lft forever
```

### Ressources supplémentaires

- **systemd.link(5)** page de manuel

## 4.2. AUGMENTATION DE LA QUANTITÉ DE MÉMOIRE QUE LES UTILISATEURS SONT AUTORISÉS À UTILISER DANS LE SYSTÈME

Les opérations d'accès direct à la mémoire à distance (RDMA) nécessitent l'épinglage de la mémoire physique. Par conséquent, le noyau n'est pas autorisé à écrire de la mémoire dans l'espace d'échange. Si un utilisateur épingle trop de mémoire, le système peut se retrouver à court de mémoire et le noyau met fin aux processus pour libérer plus de mémoire. Par conséquent, l'épinglage de la mémoire est une opération privilégiée.

Si des utilisateurs non root doivent exécuter de grandes applications RDMA, il est nécessaire d'augmenter la quantité de mémoire pour maintenir les pages en mémoire primaire épinglées en permanence.

### Procédure

- En tant qu'utilisateur **root**, créez le fichier **/etc/security/limits.conf** avec le contenu suivant :

```
@rdma soft memlock unlimited
@rdma hard memlock unlimited
```

### Vérification

- Connectez-vous en tant que membre du groupe **rdma** après avoir modifié le fichier **/etc/security/limits.conf**.  
Notez que Red Hat Enterprise Linux applique les paramètres mis à jour de **ulimit** lorsque l'utilisateur se connecte.
- Utilisez la commande **ulimit -l** pour afficher la limite :

```
$ ulimit -l
unlimited
```

Si la commande renvoie **unlimited**, l'utilisateur peut épingler une quantité illimitée de mémoire.

## Ressources supplémentaires

- [limits.conf\(5\)](#) page de manuel

## 4.3. ACTIVATION DE NFS SUR RDMA (NFSORDMA)

Dans Red Hat Enterprise Linux 9, le service d'accès direct à la mémoire à distance (RDMA) sur du matériel compatible RDMA fournit une prise en charge du protocole Network File System (NFS) pour le transfert de fichiers à grande vitesse sur le réseau.

### Procédure

1. Installez le paquetage **rdma-core**:

```
# dnf install rdma-core
```

2. Vérifiez que les lignes contenant **xprtrdma** et **svcrdma** ne sont pas commentées dans le fichier **/etc/rdma/modules/rdma.conf**:

```
# NFS over RDMA client support  
xprtrdma  
# NFS over RDMA server support  
svcrdma
```

3. Sur le serveur NFS, créez le répertoire **/mnt/nfsordma** et exportez-le vers **/etc/exports**:

```
# mkdir /mnt/nfsordma  
# echo "/mnt/nfsordma *(fsid=0,rw,async,insecure,no_root_squash)" >> /etc/exports
```

4. Sur le client NFS, montez le partage nfs avec l'adresse IP du serveur, par exemple, **172.31.0.186**:

```
# mount -o rdma,port=20049 172.31.0.186:/mnt/nfs-share /mnt/nfs
```

5. Redémarrez le service **nfs-server**:

```
# systemctl restart nfs-server
```

## Ressources supplémentaires

- [La norme RFC 5667](#)

## CHAPITRE 5. CONFIGURATION D'UN GESTIONNAIRE DE SOUS-RÉSEAU INFINIBAND

Tous les réseaux InfiniBand doivent avoir un gestionnaire de sous-réseau en cours d'exécution pour que le réseau fonctionne. Ceci est vrai même si deux machines sont connectées directement sans commutateur.

Il est possible d'avoir plus d'un gestionnaire de sous-réseau. Dans ce cas, l'un d'entre eux agit en tant que maître et un autre gestionnaire de sous-réseau agit en tant qu'esclave qui prendra le relais en cas de défaillance du gestionnaire de sous-réseau maître.

La plupart des commutateurs InfiniBand contiennent un gestionnaire de sous-réseau intégré. Cependant, si vous avez besoin d'un gestionnaire de sous-réseau plus récent ou si vous souhaitez avoir plus de contrôle, utilisez le gestionnaire de sous-réseau **OpenSM** fourni par Red Hat Enterprise Linux.

Pour plus de détails, voir [Installation du gestionnaire de sous-réseau OpenSM](#)

## CHAPITRE 6. CONFIGURATION D'IPOIB

Par défaut, InfiniBand n'utilise pas le protocole internet (IP) pour la communication. Cependant, IP over InfiniBand (IPoIB) fournit une couche d'émulation de réseau IP au-dessus des réseaux RDMA (remote direct memory access) InfiniBand. Cela permet aux applications existantes non modifiées de transmettre des données sur les réseaux InfiniBand, mais les performances sont moindres que si l'application utilisait RDMA de manière native.



### NOTE

Les équipements Mellanox, à partir de ConnectX-4, sur RHEL 8 et plus, utilisent par défaut le mode Enhanced IPoIB (datagramme uniquement). Le mode connecté n'est pas pris en charge sur ces équipements.

### 6.1. LES MODES DE COMMUNICATION IPOIB

Un appareil IPoIB peut être configuré en mode **Datagram** ou **Connected**. La différence réside dans le type de paire de files d'attente que la couche IPoIB tente d'ouvrir avec la machine à l'autre bout de la communication :

- Dans le mode **Datagram**, le système ouvre une paire de files d'attente non fiable et déconnectée.  
Ce mode ne prend pas en charge les paquets d'une taille supérieure à l'unité de transmission maximale (MTU) de la couche de liaison InfiniBand. Lors de la transmission des données, la couche IPoIB ajoute un en-tête IPoIB de 4 octets au paquet IP. Par conséquent, le MTU IPoIB est inférieur de 4 octets au MTU de la couche de liaison InfiniBand. Comme **2048** est un MTU commun de la couche de liaison InfiniBand, le MTU commun de l'appareil IPoIB en mode **Datagram** est **2044**.
- En mode **Connected**, le système ouvre une paire de files d'attente fiable et connectée.  
Ce mode permet d'envoyer des messages plus importants que le MTU de la couche de liaison InfiniBand. L'adaptateur hôte gère la segmentation et le réassemblage des paquets. Par conséquent, dans le mode **Connected**, les messages envoyés par les adaptateurs InfiniBand ne sont pas limités en taille. Cependant, les paquets IP sont limités en raison du champ **data** et du champ TCP/IP **header**. Pour cette raison, le MTU IPoIB dans le mode **Connected** est de **65520** octets.

Le mode **Connected** est plus performant mais consomme plus de mémoire du noyau.

Bien qu'un système soit configuré pour utiliser le mode **Connected**, il envoie toujours du trafic multicast en utilisant le mode **Datagram** car les commutateurs et la structure InfiniBand ne peuvent pas transmettre le trafic multicast en mode **Connected**. De même, lorsque l'hôte n'est pas configuré pour utiliser le mode **Connected**, le système revient au mode **Datagram**.

Lors de l'exécution d'une application qui envoie des données multicast jusqu'à MTU sur l'interface, configure l'interface en mode **Datagram** ou configure l'application pour limiter la taille d'envoi d'un paquet qui tiendra dans des paquets de la taille d'un datagramme.

### 6.2. COMPRENDRE LES ADRESSES MATÉRIELLES IPOIB

Les appareils IPoIB ont une adresse matérielle de **20** octets qui se compose des éléments suivants :

- Les 4 premiers octets sont des drapeaux et des numéros de paires de files d'attente
- Les 8 octets suivants sont le préfixe du sous-réseau

Le préfixe de sous-réseau par défaut est **0xfe:80:00:00:00:00:00:00**. Lorsque l'appareil se connecte au gestionnaire de sous-réseau, il modifie ce préfixe pour qu'il corresponde au gestionnaire de sous-réseau configuré.

- Les 8 derniers octets sont l'identifiant unique global (GUID) du port InfiniBand qui s'attache au dispositif IPoIB



#### NOTE

Comme les 12 premiers octets peuvent changer, ne les utilisez pas dans les règles du gestionnaire de périphérique **udev**.

## 6.3. CONFIGURATION D'UNE CONNEXION IPOIB À L'AIDE DES COMMANDES NMCLI

L'utilitaire de ligne de commande **nmcli** contrôle le NetworkManager et signale l'état du réseau à l'aide du CLI.

### Conditions préalables

- Un dispositif InfiniBand est installé sur le serveur
- Le module du noyau correspondant est chargé

### Procédure

1. Créez la connexion InfiniBand pour utiliser l'interface **mlx4\_ib0** dans le mode de transport **Connected** et le MTU maximum de **65520** octets :

```
# nmcli connection add type infiniband con-name mlx4_ib0 ifname mlx4_ib0 transport-mode Connected mtu 65520
```

2. Vous pouvez également définir **0x8002** comme interface **P\_Key** de la connexion **mlx4\_ib0**:

```
# nmcli connection modify mlx4_ib0 infiniband.p-key 0x8002
```

3. Pour configurer les paramètres IPv4, définissez une adresse IPv4 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **mlx4\_ib0**:

```
# nmcli connection modify mlx4_ib0 ipv4.addresses 192.0.2.1/24
# nmcli connection modify mlx4_ib0 ipv4.gateway 192.0.2.254
# nmcli connection modify mlx4_ib0 ipv4.dns 192.0.2.253
# nmcli connection modify mlx4_ib0 ipv4.method manual
```

4. Pour configurer les paramètres IPv6, définissez une adresse IPv6 statique, un masque de réseau, une passerelle par défaut et un serveur DNS pour la connexion **mlx4\_ib0**:

```
# nmcli connection modify mlx4_ib0 ipv6.addresses 2001:db8:1::1/32
# nmcli connection modify mlx4_ib0 ipv6.gateway 2001:db8:1::fffe
# nmcli connection modify mlx4_ib0 ipv6.dns 2001:db8:1::fffd
# nmcli connection modify mlx4_ib0 ipv6.method manual
```

5. Pour activer la connexion **mlx4\_ib0**:

```
# nmcli connection up mlx4_ib0
```

## 6.4. CONFIGURATION D'UNE CONNEXION IPOIB À L'AIDE DU RÔLE DE RÉSEAU RHEL SYSTEM ROLE

Vous pouvez utiliser le rôle système **network** RHEL pour créer à distance des profils de connexion NetworkManager pour les périphériques IP over InfiniBand (IPoIB). Par exemple, ajoutez à distance une connexion InfiniBand pour l'interface **mlx4\_ib0** avec les paramètres suivants en exécutant un playbook Ansible :

- Un dispositif IPoIB - **mlx4\_ib0.8002**
- Une clé de partition **p\_key** - **0x8002**
- Une adresse statique **IPv4** - **192.0.2.1** avec un masque de sous-réseau **/24**
- Une adresse statique **IPv6** - **2001:db8:1::1** avec un masque de sous-réseau **/64**

Effectuez cette procédure sur le nœud de contrôle Ansible.

### Conditions préalables

- [Vous avez préparé le nœud de contrôle et les nœuds gérés](#)
- Vous vous êtes connecté au nœud de contrôle en tant qu'utilisateur pouvant exécuter des sélections 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 InfiniBand nommé **mlx4\_ib0** est installé dans les nœuds gérés.
- Les nœuds gérés utilisent NetworkManager pour configurer le réseau.

### Procédure

1. Créez un fichier playbook, par exemple **~/IPoIB.yml** avec le contenu suivant :

```
---
- name: Configure the network
  hosts: managed-node-01.example.com
  tasks:
    - name: Configure IPoIB
      include_role:
        name: rhel-system-roles.network

  vars:
    network_connections:

    # InfiniBand connection mlx4_ib0
    - name: mlx4_ib0
      interface_name: mlx4_ib0
```

```

type: infiniband

# IPoIB device mlx4_ib0.8002 on top of mlx4_ib0
- name: mlx4_ib0.8002
  type: infiniband
  autoconnect: yes
  infiniband:
    p_key: 0x8002
    transport_mode: datagram
  parent: mlx4_ib0
  ip:
    address:
      - 192.0.2.1/24
      - 2001:db8:1::1/64
  state: up

```

Si vous définissez un paramètre **p\_key** comme dans cet exemple, ne définissez pas de paramètre **interface\_name** sur le périphérique IPoIB.

2. Exécutez le manuel de jeu :

```
# ansible-playbook ~/IPoIB.yml
```

### Vérification

1. Sur l'hôte **managed-node-01.example.com**, affichez les paramètres IP du périphérique **mlx4\_ib0.8002**:

```

# ip address show mlx4_ib0.8002
...
inet 192.0.2.1/24 brd 192.0.2.255 scope global noprefixroute ib0.8002
  valid_lft forever preferred_lft forever
inet6 2001:db8:1::1/64 scope link tentative noprefixroute
  valid_lft forever preferred_lft forever

```

2. Affichez la clé de partition (P\_Key) de l'appareil **mlx4\_ib0.8002**:

```
# cat /sys/class/net/mlx4_ib0.8002/pkey
0x8002
```

3. Affiche le mode de l'appareil **mlx4\_ib0.8002**:

```
# cat /sys/class/net/mlx4_ib0.8002/mode
datagram
```

### Ressources supplémentaires

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

## 6.5. CONFIGURATION D'UNE CONNEXION IPOIB À L'AIDE DE NM-CONNECTION-EDITOR

L'application **nmcli-connection-editor** configure et gère les connexions réseau stockées par NetworkManager à l'aide de la console de gestion.

### Conditions préalables

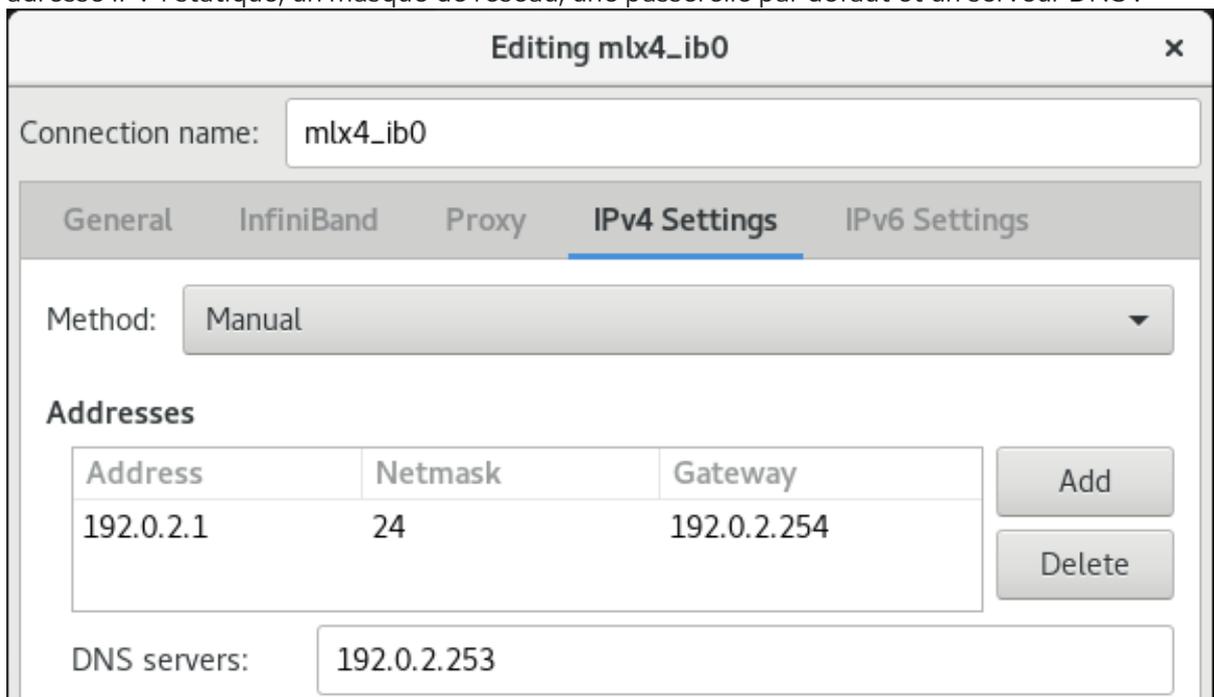
- Un dispositif InfiniBand est installé sur le serveur.
- Le module du noyau correspondant est chargé
- Le paquet **nm-connection-editor** est installé.

### Procédure

1. Entrez la commande :

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

2. Cliquez sur le bouton  pour ajouter une nouvelle connexion.
3. Sélectionnez le type de connexion **InfiniBand** et cliquez sur **Créer**.
4. Dans l'onglet **InfiniBand**:
  - a. Modifiez le nom de la connexion si vous le souhaitez.
  - b. Sélectionner le mode de transport.
  - c. Sélectionnez l'appareil.
  - d. Définissez un MTU si nécessaire.
5. 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 :



The screenshot shows the 'Editing mlx4\_ib0' window with the 'IPv4 Settings' tab selected. The 'Method' is set to 'Manual'. Under 'Addresses', there is a table with one entry: Address 192.0.2.1, Netmask 24, and Gateway 192.0.2.254. Below the table, the 'DNS servers' field is set to 192.0.2.253.

Address	Netmask	Gateway
192.0.2.1	24	192.0.2.254

6. 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 :

The screenshot shows a window titled "Editing mlx4\_ib0" with a close button (X) in the top right corner. The "Connection name" field contains "mlx4\_ib0". Below this is a tabbed interface with tabs for "General", "InfiniBand", "Proxy", "IPv4 Settings", and "IPv6 Settings". The "IPv6 Settings" tab is selected. Under "Method", a dropdown menu is set to "Manual". The "Addresses" section contains a table with the following data:

Address	Prefix	Gateway
2001:db8::1	32	2001:db8::fffe

To the right of the table are "Add" and "Delete" buttons. Below the table, the "DNS servers" field contains "2001:db8::fffd".

7. Cliquez sur **Enregistrer** pour sauvegarder la connexion de l'équipe.
8. Fermer **nm-connection-editor**.
9. Vous pouvez définir une interface **P\_Key**. Comme ce paramètre n'est pas disponible dans **nm-connection-editor**, vous devez le définir sur la ligne de commande. Par exemple, pour définir **0x8002** comme interface **P\_Key** de la connexion **mlx4\_ib0**:

```
# nmcli connection modify mlx4_ib0 infiniband.p-key 0x8002
```

# CHAPITRE 7. TEST DES RÉSEAUX INFINIBAND

## 7.1. TEST DES PREMIÈRES OPÉRATIONS RDMA INFINIBAND

InfiniBand offre une faible latence et des performances élevées pour l'accès direct à la mémoire à distance (RDMA).



### NOTE

Outre InfiniBand, si vous utilisez des dispositifs IP tels que Internet Wide-area Remote Protocol (iWARP) ou RDMA over Converged Ethernet (RoCE) ou InfiniBand over Ethernet (IBoE), reportez-vous à la section suivante :

- [Test d'un IPoIB à l'aide de l'utilitaire ping](#)
- [Test d'un réseau RDMA avec iperf3 après configuration d'IPoIB](#)

### Conditions préalables

- Vous avez configuré le service **rdma**.
- Vous avez installé les paquets **libibverbs-utils** et **infiniband-diags**.

### Procédure

1. Liste des périphériques InfiniBand disponibles :

```
# ibv_devices

device          node GUID
-----          -
mlx4_0          0002c903003178f0
mlx4_1          f4521403007bcba0
```

2. Affiche les informations relatives à l'appareil **mlx4\_1**:

```
# ibv_devinfo -d mlx4_1

hca_id: mlx4_1
transport:          InfiniBand (0)
fw_ver:             2.30.8000
node_guid:          f452:1403:007b:cba0
sys_image_guid:     f452:1403:007b:cba3
vendor_id:          0x02c9
vendor_part_id:     4099
hw_ver:             0x0
board_id:           MT_1090120019
phys_port_cnt:     2
  port: 1
    state:          PORT_ACTIVE (4)
    max_mtu:        4096 (5)
    active_mtu:     2048 (4)
    sm_lid:         2
    port_lid:       2
```

```

port_lmc:      0x01
link_layer:    InfiniBand

port: 2
state:         PORT_ACTIVE (4)
max_mtu:      4096 (5)
active_mtu:   4096 (5)
sm_lid:       0
port_lid:     0
port_lmc:     0x00
link_layer:   Ethernet

```

3. Affiche l'état de l'appareil **mlx4\_1**:

```

# ibstat mlx4_1

CA 'mlx4_1'
CA type: MT4099
Number of ports: 2
Firmware version: 2.30.8000
Hardware version: 0
Node GUID: 0xf4521403007bcba0
System image GUID: 0xf4521403007bcba3
Port 1:
  State: Active
  Physical state: LinkUp
  Rate: 56
  Base lid: 2
  LMC: 1
  SM lid: 2
  Capability mask: 0x0251486a
  Port GUID: 0xf4521403007bcba1
  Link layer: InfiniBand
Port 2:
  State: Active
  Physical state: LinkUp
  Rate: 40
  Base lid: 0
  LMC: 0
  SM lid: 0
  Capability mask: 0x04010000
  Port GUID: 0xf65214ffe7bcba2
  Link layer: Ethernet

```

4. L'utilitaire **ibping** interroge une adresse InfiniBand et fonctionne comme un client/serveur en configurant les paramètres.
- Démarrer le mode serveur **-S** sur le numéro de port **-P** avec le nom de l'autorité de certification (CA) InfiniBand **-C** sur l'hôte :

```
# ibping -S -C mlx4_1 -P 1
```

- Démarrez le mode client, envoyez quelques paquets **-c** sur le port numéro **-P** en utilisant le nom de l'autorité de certification (CA) InfiniBand **-C** avec l'identifiant local (LID) **-L** sur l'hôte :

```
# ibping -c 50 -C mlx4_0 -P 1 -L 2
```

### Ressources supplémentaires

- **ibping(8)** page de manuel

## 7.2. TEST D'UN IPOIB À L'AIDE DE L'UTILITAIRE PING

Après avoir configuré IP over InfiniBand (IPoIB), utilisez l'utilitaire **ping** pour envoyer des paquets ICMP afin de tester la connexion IPoIB.

### Conditions préalables

- Les deux hôtes RDMA sont connectés dans la même structure InfiniBand avec des ports RDMA
- Les interfaces IPoIB des deux hôtes sont configurées avec des adresses IP dans le même sous-réseau

### Procédure

- Utilisez l'utilitaire **ping** pour envoyer cinq paquets ICMP à la carte InfiniBand de l'hôte distant :

```
# ping -c5 192.0.2.1
```

## 7.3. TEST D'UN RÉSEAU RDMA AVEC IPERF3 APRÈS CONFIGURATION D'IPOIB

Dans l'exemple suivant, la grande taille du tampon est utilisée pour effectuer un test de 60 secondes afin de mesurer le débit maximal et d'utiliser pleinement la bande passante et la latence entre deux hôtes à l'aide de l'utilitaire **iperf3**.

### Conditions préalables

- Vous avez configuré IPoIB sur les deux hôtes.

### Procédure

1. Pour exécuter **iperf3** en tant que serveur sur un système, définissez un intervalle de temps pour fournir des mises à jour périodiques de la bande passante **-i** pour écouter en tant que serveur **-s** qui attend la réponse de la connexion du client :

```
# iperf3 -i 5 -s
```

2. Pour exécuter **iperf3** en tant que client sur un autre système, définissez un intervalle de temps pour fournir des mises à jour périodiques de la bande passante **-i** pour se connecter au serveur d'écoute **-c** de l'adresse IP **192.168.2.2** avec **-t** temps en secondes :

```
# iperf3 -i 5 -t 60 -c 192.168.2.2
```

3. Utilisez les commandes suivantes :
  - a. Afficher les résultats des tests sur le système qui fait office de serveur :

```
# iperf3 -i 10 -s
```

```
-----
Server listening on 5201
-----
```

```
Accepted connection from 192.168.2.3, port 22216
```

```
[5] local 192.168.2.2 port 5201 connected to 192.168.2.3 port 22218
```

```
[ID] Interval      Transfer    Bandwidth
[5]  0.00-10.00 sec  17.5 GBytes 15.0 Gbits/sec
[5]  10.00-20.00 sec  17.6 GBytes 15.2 Gbits/sec
[5]  20.00-30.00 sec  18.4 GBytes 15.8 Gbits/sec
[5]  30.00-40.00 sec  18.0 GBytes 15.5 Gbits/sec
[5]  40.00-50.00 sec  17.5 GBytes 15.1 Gbits/sec
[5]  50.00-60.00 sec  18.1 GBytes 15.5 Gbits/sec
[5]  60.00-60.04 sec  82.2 MBytes 17.3 Gbits/sec
```

```
-----
[ID] Interval      Transfer    Bandwidth
[5]  0.00-60.04 sec  0.00 Bytes  0.00 bits/sec sender
[5]  0.00-60.04 sec  107 GBytes 15.3 Gbits/sec receiver
```

b. Afficher les résultats des tests sur le système qui fait office de client :

```
# iperf3 -i 1 -t 60 -c 192.168.2.2
```

```
Connecting to host 192.168.2.2, port 5201
```

```
[4] local 192.168.2.3 port 22218 connected to 192.168.2.2 port 5201
```

```
[ID] Interval      Transfer    Bandwidth    Retr Cwnd
[4]  0.00-10.00 sec  17.6 GBytes 15.1 Gbits/sec  0 6.01 MBytes
[4]  10.00-20.00 sec  17.6 GBytes 15.1 Gbits/sec  0 6.01 MBytes
[4]  20.00-30.00 sec  18.4 GBytes 15.8 Gbits/sec  0 6.01 MBytes
[4]  30.00-40.00 sec  18.0 GBytes 15.5 Gbits/sec  0 6.01 MBytes
[4]  40.00-50.00 sec  17.5 GBytes 15.1 Gbits/sec  0 6.01 MBytes
[4]  50.00-60.00 sec  18.1 GBytes 15.5 Gbits/sec  0 6.01 MBytes
```

```
-----
[ID] Interval      Transfer    Bandwidth    Retr
[4]  0.00-60.00 sec  107 GBytes 15.4 Gbits/sec  0 sender
[4]  0.00-60.00 sec  107 GBytes 15.4 Gbits/sec  receiver
```

### Ressources supplémentaires

- **iperf3** page de manuel