Red Hat Enterprise Linux 9

Configurer et gérer des clusters à haute disponibilité

Utilisation de Red Hat High Availability Add-On pour créer et maintenir des clusters Pacemaker

Last Updated: 2024-02-28

Red Hat Enterprise Linux 9 Configurer et gérer des clusters à haute disponibilité

Utilisation de Red Hat High Availability Add-On pour créer et maintenir des clusters Pacemaker

Notice légale

Résumé

Le module complémentaire de haute disponibilité de Red Hat configure les grappes de haute disponibilité qui utilisent le gestionnaire de ressources de grappes Pacemaker. Ce titre fournit des procédures pour vous familiariser avec la configuration des clusters Pacemaker ainsi que des exemples de procédures pour configurer des clusters actifs/actifs et actifs/passifs.

Table des matières

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. APERÇU DU MODULE COMPLÉMENTAIRE DE HAUTE DISPONIBILITÉ

Le module complémentaire de haute disponibilité est un système en grappe qui assure la fiabilité, l'évolutivité et la disponibilité des services de production critiques.

Une grappe est constituée de deux ordinateurs ou plus (appelés *nodes* ou *members*) qui travaillent ensemble pour effectuer une tâche. Les grappes peuvent être utilisées pour fournir des services ou des ressources hautement disponibles. La redondance de plusieurs machines permet de se prémunir contre de nombreux types de défaillances.

Les grappes à haute disponibilité fournissent des services hautement disponibles en éliminant les points de défaillance uniques et en transférant les services d'un nœud à l'autre au cas où un nœud deviendrait inopérant. En règle générale, les services d'une grappe à haute disponibilité lisent et écrivent des données (au moyen de systèmes de fichiers montés en lecture-écriture). Par conséquent, une grappe à haute disponibilité doit maintenir l'intégrité des données lorsqu'un nœud de la grappe prend le contrôle d'un service d'un autre nœud de la grappe. Les défaillances de nœuds dans une grappe à haute disponibilité ne sont pas visibles pour les clients extérieurs à la grappe. (Les grappes à haute disponibilité sont parfois appelées grappes de basculement.) Le module complémentaire de haute disponibilité fournit une grappe à haute disponibilité par l'intermédiaire de son composant de gestion de services à haute disponibilité, **Pacemaker**.

1.1. COMPOSANTS COMPLÉMENTAIRES DE HAUTE DISPONIBILITÉ

Le module complémentaire de haute disponibilité de Red Hat est constitué de plusieurs composants qui fournissent le service de haute disponibilité.

Les principaux composants du module complémentaire de haute disponibilité sont les suivants :

- Infrastructure de grappe - Fournit les fonctions fondamentales permettant aux nœuds de travailler ensemble en tant que grappe : gestion des fichiers de configuration, gestion des membres, gestion des verrous et clôtures.
- Gestion des services à haute disponibilité - assure le basculement des services d'un nœud de la grappe à l'autre en cas de défaillance d'un nœud.
- Outils d'administration des clusters - Outils de configuration et de gestion permettant de mettre en place, de configurer et de gérer le module complémentaire de haute disponibilité. Ces outils sont destinés à être utilisés avec les composants de l'infrastructure du cluster, les composants de gestion de la haute disponibilité et des services, et le stockage.

Vous pouvez compléter le module complémentaire de haute disponibilité avec les composants suivants :

- Red Hat GFS2 (Global File System 2) - Faisant partie du module complémentaire de stockage résilient, il fournit un système de fichiers en cluster à utiliser avec le module complémentaire de haute disponibilité. GFS2 permet à plusieurs nœuds de partager le stockage au niveau des blocs comme si le stockage était connecté localement à chaque nœud du cluster. Le système de fichiers en cluster GFS2 nécessite une infrastructure en cluster.
- LVM Locking Daemon (**lvmlockd**) - Partie intégrante du module complémentaire Resilient Storage, il assure la gestion des volumes de stockage en grappe. La prise en charge de **lvmlockd** nécessite également une infrastructure en grappe.

- HAProxy - Logiciel de routage qui assure l'équilibrage de la charge à haute disponibilité et le basculement des services de la couche 4 (TCP) et de la couche 7 (HTTP, HTTPS).

1.2. CONCEPTS DU MODULE COMPLÉMENTAIRE DE HAUTE DISPONIBILITÉ

Certains des concepts clés d'un cluster Red Hat High Availability Add-On sont les suivants.

1.2.1. Clôture

Si la communication avec un seul nœud de la grappe échoue, les autres nœuds de la grappe doivent être en mesure de restreindre ou de libérer l'accès aux ressources auxquelles le nœud défaillant peut avoir accès. Il n'est pas possible de le faire en contactant le nœud de la grappe lui-même, car il risque de ne pas être réactif. Au lieu de cela, vous devez fournir une méthode externe, appelée clôture avec un agent de clôture. Un dispositif de clôture est un dispositif externe qui peut être utilisé par la grappe pour restreindre l'accès d'un nœud défaillant aux ressources partagées ou pour redémarrer brutalement le nœud de la grappe.

Sans dispositif de clôture configuré, vous n'avez aucun moyen de savoir que les ressources précédemment utilisées par le nœud de grappe déconnecté ont été libérées, ce qui pourrait empêcher les services de fonctionner sur l'un des autres nœuds de grappe. Inversement, le système peut supposer à tort que le nœud de la grappe a libéré ses ressources, ce qui peut entraîner la corruption et la perte de données. Sans dispositif de clôture configuré, l'intégrité des données ne peut être garantie et la configuration de la grappe ne sera pas prise en charge.

Lorsque la clôture est en cours, aucune autre opération de la grappe n'est autorisée. Le fonctionnement normal de la grappe ne peut pas reprendre avant la fin de la clôture ou avant que le nœud de la grappe ne rejoigne la grappe après avoir été redémarré.

Pour plus d'informations sur la clôture, voir Clôture dans un cluster Red Hat High Availability .

1.2.2. Quorum

Afin de maintenir l'intégrité et la disponibilité de la grappe, les systèmes de grappes utilisent un concept connu sous le nom de *quorum* pour empêcher la corruption et la perte de données. Une grappe a un quorum lorsque plus de la moitié des nœuds de la grappe sont en ligne. Pour réduire les risques de corruption des données en cas de défaillance, Pacemaker arrête par défaut toutes les ressources si la grappe n'a pas de quorum.

Le quorum est établi à l'aide d'un système de vote. Lorsqu'un nœud de la grappe ne fonctionne pas comme il le devrait ou perd la communication avec le reste de la grappe, la majorité des nœuds en activité peuvent voter pour isoler et, si nécessaire, clôturer le nœud pour qu'il soit réparé.

Par exemple, dans une grappe de 6 nœuds, le quorum est établi lorsqu'au moins 4 nœuds de la grappe fonctionnent. Si la majorité des nœuds se déconnectent ou deviennent indisponibles, la grappe n'a plus de quorum et Pacemaker arrête les services en grappe.

Les fonctions de quorum de Pacemaker permettent d'éviter ce que l'on appelle également *split-brain*, un phénomène dans lequel la grappe est séparée de la communication, mais où chaque partie continue à travailler comme des grappes séparées, en écrivant potentiellement sur les mêmes données et en provoquant éventuellement des corruptions ou des pertes. Pour plus d'informations sur ce que signifie être dans un état de cerveau divisé, et sur les concepts de quorum en général, voir Exploring Concepts of RHEL High Availability Clusters - Quorum (Explorer les concepts des clusters haute disponibilité RHEL - Quorum).

Un cluster Red Hat Enterprise Linux High Availability Add-On utilise le service **votequorum**, en conjonction avec la clôture, pour éviter les situations de cerveau divisé. Un nombre de votes est attribué à chaque système de la grappe, et les opérations de la grappe ne sont autorisées que lorsqu'une majorité de votes est présente.

1.2.3. Cluster resources

Un site *cluster resource* est une instance de programme, de données ou d'application qui doit être gérée par le service de cluster. Ces ressources sont abstraites par *agents* qui fournit une interface standard pour la gestion de la ressource dans un environnement de cluster.

Pour garantir que les ressources restent saines, vous pouvez ajouter une opération de contrôle à la définition d'une ressource. Si vous ne spécifiez pas d'opération de surveillance pour une ressource, une opération est ajoutée par défaut.

Vous pouvez déterminer le comportement d'une ressource dans un cluster en configurant *constraints*. Vous pouvez configurer les catégories de contraintes suivantes :

- contraintes d'emplacement - Une contrainte d'emplacement détermine les nœuds sur lesquels une ressource peut s'exécuter.
- contraintes d'ordre - Une contrainte d'ordre détermine l'ordre d'exécution des ressources.
- contraintes de colocalisation - Une contrainte de colocalisation détermine où les ressources seront placées par rapport à d'autres ressources.

L'un des éléments les plus courants d'un cluster est un ensemble de ressources qui doivent être situées ensemble, démarrer de manière séquentielle et s'arrêter dans l'ordre inverse. Pour simplifier cette configuration, Pacemaker prend en charge le concept de *groups*.

1.3. APERÇU DES STIMULATEURS CARDIAQUES

Pacemaker est un gestionnaire de ressources en grappe. Il assure une disponibilité maximale des services et des ressources de la grappe en utilisant les capacités de messagerie et d'adhésion de l'infrastructure de la grappe pour prévenir les défaillances au niveau des nœuds et des ressources et y remédier.

1.3.1. Composants de l'architecture du stimulateur cardiaque

Un cluster configuré avec Pacemaker comprend des démons de composants distincts qui surveillent l'appartenance au cluster, des scripts qui gèrent les services et des sous-systèmes de gestion des ressources qui surveillent les ressources disparates.

L'architecture de Pacemaker se compose des éléments suivants :

Base d'informations sur les clusters (CIB)

Le démon d'information Pacemaker, qui utilise XML en interne pour distribuer et synchroniser la configuration actuelle et les informations d'état du coordinateur désigné (DC) - un nœud assigné par Pacemaker pour stocker et distribuer l'état et les actions de la grappe au moyen de la CIB - à tous les autres nœuds de la grappe.

Démon de gestion des ressources du cluster (CRMd)

Les actions sur les ressources de la grappe Pacemaker sont acheminées par l'intermédiaire de ce démon. Les ressources gérées par CRMd peuvent être interrogées par les systèmes clients, déplacées, instanciées et modifiées si nécessaire.

Chaque nœud de la grappe comprend également un démon de gestion des ressources locales (LRMd) qui sert d'interface entre CRMd et les ressources. Le LRMd transmet les commandes de CRMd aux agents, comme le démarrage et l'arrêt et la transmission d'informations sur l'état des ressources.

Tirez dans la tête de l'autre nœud (STONITH)

STONITH est l'implémentation de la clôture de Pacemaker. Il agit comme une ressource de cluster dans Pacemaker qui traite les demandes de clôture, en arrêtant de force les nœuds et en les retirant du cluster pour garantir l'intégrité des données. STONITH est configuré dans la CIB et peut être surveillé comme une ressource de cluster normale.

corosync

corosync est le composant - et le démon du même nom - qui répond aux besoins essentiels en matière d'adhésion et de communication avec les membres pour les grappes à haute disponibilité. Il est nécessaire au fonctionnement du module complémentaire de haute disponibilité.
Outre ces fonctions d'adhésion et de messagerie, le site **corosync** offre également les services suivants :

- Gère les règles de quorum et la détermination de celui-ci.
- Fournit des capacités de messagerie pour les applications qui coordonnent ou opèrent à travers plusieurs membres du cluster et qui doivent donc communiquer des informations d'état ou autres entre les instances.
- Utilise la bibliothèque **kronosnet** comme transport réseau pour fournir des liens redondants multiples et un basculement automatique.

1.3.2. Outils de configuration et de gestion de Pacemaker

Le module complémentaire de haute disponibilité comprend deux outils de configuration pour le déploiement, la surveillance et la gestion des clusters.

pcs

L'interface de ligne de commande **pcs** contrôle et configure Pacemaker et le démon Heartbeat **corosync**. Programme basé sur la ligne de commande, **pcs** peut effectuer les tâches de gestion de cluster suivantes :

- Créer et configurer un cluster Pacemaker/Corosync
- Modifier la configuration de la grappe en cours d'exécution
- Configurer à distance Pacemaker et Corosync, ainsi que démarrer, arrêter et afficher des informations sur l'état de la grappe

pcsd Interface utilisateur Web

Une interface utilisateur graphique pour créer et configurer les clusters Pacemaker/Corosync.

1.3.3. Les fichiers de configuration du cluster et du pacemaker

Les fichiers de configuration pour le Red Hat High Availability Add-On sont **corosync.conf** et **cib.xml**.

Le fichier **corosync.conf** fournit les paramètres de cluster utilisés par **corosync**, le gestionnaire de cluster sur lequel Pacemaker est construit. En général, vous ne devez pas éditer directement le fichier **corosync.conf**, mais plutôt utiliser l'interface **pcs** ou **pcsd**.

Le fichier **cib.xml** est un fichier XML qui représente à la fois la configuration de la grappe et l'état actuel de toutes les ressources de la grappe. Ce fichier est utilisé par la base d'informations sur les clusters (CIB) de Pacemaker. Le contenu de la CIB est automatiquement synchronisé avec l'ensemble de la grappe. Ne modifiez pas directement le fichier **cib.xml**; utilisez plutôt l'interface **pcs** ou **pcsd**.

1.4. VOLUMES LOGIQUES LVM DANS UN CLUSTER DE HAUTE DISPONIBILITÉ RED HAT

Le module complémentaire de haute disponibilité de Red Hat prend en charge les volumes LVM dans deux configurations de cluster distinctes.

Les configurations de cluster que vous pouvez choisir sont les suivantes :

- Volumes LVM à haute disponibilité (HA-LVM) dans les configurations de basculement actif/passif dans lesquelles un seul nœud de la grappe accède au stockage à tout moment.
- Les volumes LVM qui utilisent le démon **lvmlockd** pour gérer les périphériques de stockage dans des configurations actives/actives dans lesquelles plus d'un nœud de la grappe a besoin d'accéder au stockage en même temps. Le démon **lvmlockd** fait partie du module complémentaire Resilient Storage.

1.4.1. Choisir HA-LVM ou des volumes partagés

L'utilisation de HA-LVM ou de volumes logiques partagés gérés par le démon **lvmlockd** doit se faire en fonction des besoins des applications ou des services déployés.

- Si plusieurs nœuds de la grappe ont besoin d'un accès simultané en lecture/écriture aux volumes LVM dans un système actif/actif, vous devez utiliser le démon **lvmlockd** et configurer vos volumes en tant que volumes partagés. Le démon **lvmlockd** fournit un système permettant de coordonner l'activation et les modifications des volumes LVM sur les nœuds d'une grappe simultanément. Le service de verrouillage du démon **lvmlockd** protège les métadonnées LVM lorsque les différents nœuds de la grappe interagissent avec les volumes et apportent des modifications à leur configuration. Cette protection dépend de la configuration de tout groupe de volumes qui sera activé simultanément sur plusieurs nœuds de la grappe en tant que volume partagé.
- Si le cluster de haute disponibilité est configuré pour gérer des ressources partagées de manière active/passive, avec un seul membre ayant besoin d'accéder à un volume LVM donné à la fois, vous pouvez utiliser HA-LVM sans le service de verrouillage **lvmlockd**.

La plupart des applications fonctionneront mieux dans une configuration active/passive, car elles ne sont pas conçues ou optimisées pour fonctionner simultanément avec d'autres instances. Le choix d'exécuter une application qui n'est pas compatible avec les clusters sur des volumes logiques partagés peut entraîner une dégradation des performances. En effet, dans ces instances, les volumes logiques font l'objet d'une surcharge de communication avec les clusters. Une application compatible avec les grappes doit être en mesure de réaliser des gains de performances supérieurs aux pertes de performances introduites par les systèmes de fichiers en grappes et les volumes logiques compatibles avec les grappes. Certaines applications et charges de travail y parviennent plus facilement que d'autres. Pour choisir entre les deux variantes de LVM, il faut déterminer quelles sont les exigences de la grappe et si l'effort supplémentaire d'optimisation pour une grappe active/active portera ses fruits. La plupart des utilisateurs obtiendront les meilleurs résultats en utilisant HA-LVM.

HA-LVM et les volumes logiques partagés utilisant **lvmlockd** sont similaires en ce sens qu'ils empêchent la corruption des métadonnées LVM et de ses volumes logiques, ce qui pourrait se produire si plusieurs machines sont autorisées à effectuer des modifications qui se chevauchent. HA-LVM impose la

restriction qu'un volume logique ne peut être activé que de manière exclusive, c'est-à-dire sur une seule machine à la fois. Cela signifie que seules les implémentations locales (non groupées) des pilotes de stockage sont utilisées. Le fait d'éviter ainsi les frais généraux liés à la coordination des clusters permet d'améliorer les performances. Un volume partagé utilisant **lvmlockd** n'impose pas ces restrictions et un utilisateur est libre d'activer un volume logique sur toutes les machines d'une grappe ; cela oblige à utiliser des pilotes de stockage adaptés à la grappe, ce qui permet de mettre en place des systèmes de fichiers et des applications adaptés à la grappe.

1.4.2. Configuration des volumes LVM dans un cluster

Les clusters sont gérés par Pacemaker. Les volumes logiques partagés et HA-LVM ne sont pris en charge qu'avec les clusters Pacemaker et doivent être configurés en tant que ressources de cluster.

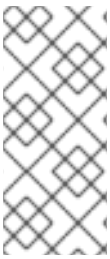
NOTE

Si un groupe de volumes LVM utilisé par un cluster Pacemaker contient un ou plusieurs volumes physiques résidant sur un stockage en bloc distant, tel qu'une cible iSCSI, Red Hat vous recommande de configurer une cible **systemd resource-agents-deps** et une unité de dépôt **systemd** pour la cible afin de garantir que le service démarre avant le démarrage de Pacemaker. Pour plus d'informations sur la configuration d'une cible **systemd resource-agents-deps**, reportez-vous à Configuration de l'ordre de démarrage pour les dépendances de ressources non gérées par Pacemaker.

- Pour des exemples de procédures de configuration d'un volume HA-LVM dans le cadre d'un cluster Pacemaker, voir Configuration d'un serveur HTTP Apache actif/passif dans un cluster Red Hat High Availability et Configuration d'un serveur NFS actif/passif dans un cluster Red Hat High Availability.
  Ces procédures comprennent les étapes suivantes :
  - S'assurer que seule la grappe est capable d'activer le groupe de volumes
  - Configuration d'un volume logique LVM
  - Configuration du volume LVM en tant que ressource de cluster
- Pour les procédures de configuration des volumes LVM partagés qui utilisent le démon **lvmlockd** pour gérer les périphériques de stockage dans des configurations actives/actives, voir Systèmes de fichiers GFS2 dans un cluster et Configuration d'un serveur Samba actif/actif dans un cluster Red Hat High Availability.

CHAPITRE 2. DÉMARRER AVEC PACEMAKER

Pour vous familiariser avec les outils et les processus utilisés pour créer une grappe Pacemaker, vous pouvez exécuter les procédures suivantes. Elles sont destinées aux utilisateurs qui souhaitent voir à quoi ressemble le logiciel de la grappe et comment il est administré, sans avoir besoin de configurer une grappe opérationnelle.



NOTE

Ces procédures ne créent pas un cluster Red Hat pris en charge, qui nécessite au moins deux nœuds et la configuration d'un dispositif de clôture. Pour obtenir des informations complètes sur les politiques d'assistance, les exigences et les limitations de Red Hat pour les clusters RHEL High Availability, voir Politiques d'assistance pour les clusters RHEL High Availability.

2.1. APPRENDRE À UTILISER UN STIMULATEUR CARDIAQUE

En suivant cette procédure, vous apprendrez à utiliser Pacemaker pour configurer une grappe, à afficher l'état de la grappe et à configurer un service de grappe. Cet exemple crée un serveur HTTP Apache en tant que ressource de grappe et montre comment la grappe réagit en cas d'échec de la ressource.

Dans cet exemple :

- Le nœud est **z1.example.com**.
- L'adresse IP flottante est 192.168.122.120.

Conditions préalables

- Un seul nœud fonctionnant sous RHEL 9
- Une adresse IP flottante qui réside sur le même réseau que l'une des adresses IP attribuées de manière statique au nœud
- Le nom du nœud sur lequel vous travaillez se trouve dans votre fichier **/etc/hosts**

Procédure

1. Installez les paquetages logiciels Red Hat High Availability Add-On à partir du canal High Availability, puis démarrez et activez le service **pcsd**.

```
# dnf install pcs pacemaker fence-agents-all
...
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

   Si vous exécutez le démon **firewalld**, activez les ports requis par le Red Hat High Availability Add-On.

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --reload
```

2. Définissez un mot de passe pour l'utilisateur **hacluster** sur chaque nœud du cluster et authentifiez l'utilisateur **hacluster** pour chaque nœud du cluster sur le nœud à partir duquel

vous exécuterez les commandes **pcs**. Cet exemple n'utilise qu'un seul nœud, le nœud à partir duquel vous exécutez les commandes, mais cette étape est incluse ici car il s'agit d'une étape nécessaire dans la configuration d'un cluster multi-nœuds pris en charge par Red Hat High Availability.

```
# passwd hacluster
...
# pcs host auth z1.example.com
```

3. Créez un cluster nommé **my_cluster** avec un membre et vérifiez l'état du cluster. Cette commande crée et démarre la grappe en une seule étape.

```
# pcs cluster setup my_cluster --start z1.example.com
...
# pcs cluster status
Cluster Status:
 Stack: corosync
 Current DC: z1.example.com (version 2.0.0-10.el8-b67d8d0de9) - partition with quorum
 Last updated: Thu Oct 11 16:11:18 2018
 Last change: Thu Oct 11 16:11:00 2018 by hacluster via crmd on z1.example.com
 1 node configured
 0 resources configured

PCSD Status:
  z1.example.com: Online
```

4. Un cluster Red Hat High Availability nécessite que vous configuriez la clôture pour le cluster. Les raisons de cette exigence sont décrites dans Fencing in a Red Hat High Availability Cluster (Clôture dans un cluster Red Hat High Availability). Cependant, pour cette introduction, qui est destinée à montrer uniquement comment utiliser les commandes de base de Pacemaker, désactivez la clôture en définissant l'option de cluster **stonith-enabled** sur **false**.

AVERTISSEMENT

L'utilisation de **stonith-enabled=false** est totalement inappropriée pour un cluster de production. Elle indique au cluster de simplement prétendre que les nœuds défaillants sont clôturés en toute sécurité.

```
# pcs property set stonith-enabled=false
```

5. Configurez un navigateur web sur votre système et créez une page web pour afficher un simple message texte. Si vous exécutez le démon **firewalld**, activez les ports requis par **httpd**.

NOTE

N'utilisez pas **systemctl enable** pour permettre aux services qui seront gérés par le cluster de démarrer au démarrage du système.

```
# dnf install -y httpd wget
...
# firewall-cmd --permanent --add-service=http
# firewall-cmd --reload

# cat <<-END >/var/www/html/index.html
<html>
<body>My Test Site - $(hostname)</body>
</html>
END
```

Pour que l'agent de ressources Apache puisse obtenir l'état d'Apache, ajoutez la configuration suivante à la configuration existante pour activer l'URL du serveur d'état.

```
# cat <<-END > /etc/httpd/conf.d/status.conf
<Location /server-status>
SetHandler server-status
Order deny,allow
Deny from all
Allow from 127.0.0.1
Allow from ::1
</Location>
END
```

6. Créez les ressources **IPaddr2** et **apache** pour le cluster à gérer. La ressource "IPaddr2" est une adresse IP flottante qui ne doit pas être déjà associée à un nœud physique. Si le périphérique NIC de la ressource "IPaddr2" n'est pas spécifié, l'adresse IP flottante doit résider sur le même réseau que l'adresse IP statiquement assignée utilisée par le nœud.
Vous pouvez afficher une liste de tous les types de ressources disponibles à l'aide de la commande **pcs resource list**. Vous pouvez utiliser la commande **pcs resource describe *resourcetype*** pour afficher les paramètres que vous pouvez définir pour le type de ressource spécifié. Par exemple, la commande suivante affiche les paramètres que vous pouvez définir pour une ressource de type **apache**:

```
# pcs resource describe apache
...
```

Dans cet exemple, la ressource adresse IP et la ressource apache sont toutes deux configurées comme faisant partie d'un groupe nommé **apachegroup**, ce qui garantit que les ressources sont conservées ensemble pour fonctionner sur le même nœud lorsque vous configurez un cluster multi-nœuds fonctionnel.

```
# pcs resource create ClusterIP ocf:heartbeat:IPaddr2 ip=192.168.122.120 --group
apachegroup

# pcs resource create WebSite ocf:heartbeat:apache
configfile=/etc/httpd/conf/httpd.conf statusurl="http://localhost/server-status" --group
apachegroup

# pcs status
Cluster name: my_cluster
Stack: corosync
Current DC: z1.example.com (version 2.0.0-10.el8-b67d8d0de9) - partition with quorum
Last updated: Fri Oct 12 09:54:33 2018
```

```
Last change: Fri Oct 12 09:54:30 2018 by root via cibadmin on z1.example.com

1 node configured
2 resources configured

Online: [ z1.example.com ]

Full list of resources:

Resource Group: apachegroup
   ClusterIP  (ocf::heartbeat:IPaddr2):       Started z1.example.com
   WebSite    (ocf::heartbeat:apache):        Started z1.example.com

PCSD Status:
  z1.example.com: Online
...
```

Après avoir configuré une ressource de cluster, vous pouvez utiliser la commande **pcs resource config** pour afficher les options configurées pour cette ressource.

```
# pcs resource config WebSite
Resource: WebSite (class=ocf provider=heartbeat type=apache)
 Attributes: configfile=/etc/httpd/conf/httpd.conf statusurl=http://localhost/server-status
 Operations: start interval=0s timeout=40s (WebSite-start-interval-0s)
             stop interval=0s timeout=60s (WebSite-stop-interval-0s)
             monitor interval=1min (WebSite-monitor-interval-1min)
```

7. Dirigez votre navigateur vers le site web que vous avez créé à l'aide de l'adresse IP flottante que vous avez configurée. Celui-ci devrait afficher le message texte que vous avez défini.

8. Arrêtez le service web Apache et vérifiez l'état de la grappe. L'utilisation de **killall -9** simule une panne au niveau de l'application.

```
# killall -9 httpd
```

Vérifiez l'état de la grappe. Vous devriez voir que l'arrêt du service web a provoqué un échec, mais que le logiciel du cluster a redémarré le service et que vous devriez toujours pouvoir accéder au site web.

```
# pcs status
Cluster name: my_cluster
...
Current DC: z1.example.com (version 1.1.13-10.el7-44eb2dd) - partition with quorum
1 node and 2 resources configured

Online: [ z1.example.com ]

Full list of resources:

Resource Group: apachegroup
   ClusterIP  (ocf::heartbeat:IPaddr2):       Started z1.example.com
   WebSite    (ocf::heartbeat:apache):        Started z1.example.com

Failed Resource Actions:
* WebSite_monitor_60000 on z1.example.com 'not running' (7): call=13, status=complete,
```

```
exitreason='none',
    last-rc-change='Thu Oct 11 23:45:50 2016', queued=0ms, exec=0ms

PCSD Status:
    z1.example.com: Online
```

Vous pouvez effacer l'état d'échec de la ressource qui a échoué une fois que le service est à nouveau opérationnel et que l'avis d'échec de l'action n'apparaît plus lorsque vous consultez l'état du cluster.

```
# pcs resource cleanup WebSite
```

9. Lorsque vous avez terminé d'examiner la grappe et son état, arrêtez les services de grappe sur le nœud. Même si vous n'avez démarré les services que sur un seul nœud pour cette introduction, le paramètre **--all** est inclus car il arrêterait les services de cluster sur tous les nœuds d'un cluster multi-nœuds réel.

```
# pcs cluster stop --all
```

2.2. APPRENDRE À CONFIGURER LE BASCULEMENT

La procédure suivante constitue une introduction à la création d'une grappe Pacemaker exécutant un service qui basculera d'un nœud à l'autre lorsque le nœud sur lequel le service s'exécute devient indisponible. En suivant cette procédure, vous apprendrez à créer un service dans une grappe à deux nœuds et vous pourrez ensuite observer ce qui arrive à ce service lorsqu'il tombe en panne sur le nœud sur lequel il s'exécute.

Cet exemple de procédure configure un cluster Pacemaker à deux nœuds exécutant un serveur HTTP Apache. Vous pouvez ensuite arrêter le service Apache sur un nœud pour voir comment le service reste disponible.

Dans cet exemple :

- Les nœuds sont **z1.example.com** et **z2.example.com**.
- L'adresse IP flottante est 192.168.122.120.

Conditions préalables

- Deux nœuds fonctionnant sous RHEL 9 et pouvant communiquer l'un avec l'autre
- Une adresse IP flottante qui réside sur le même réseau que l'une des adresses IP attribuées de manière statique au nœud
- Le nom du nœud sur lequel vous travaillez se trouve dans votre fichier **/etc/hosts**

Procédure

1. Sur les deux nœuds, installez les paquetages logiciels Red Hat High Availability Add-On à partir du canal High Availability, puis démarrez et activez le service **pcsd**.

```
# dnf install pcs pacemaker fence-agents-all
...
# systemctl start pcsd.service
```

```
# systemctl enable pcsd.service
```

Si vous exécutez le démon **firewalld**, activez sur les deux nœuds les ports requis par le Red Hat High Availability Add-On.

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --reload
```

2. Sur les deux nœuds du cluster, définissez un mot de passe pour l'utilisateur **hacluster**.

```
# passwd hacluster
```

3. Authentifiez l'utilisateur **hacluster** pour chaque nœud du cluster sur le nœud à partir duquel vous exécuterez les commandes **pcs**.

```
# pcs host auth z1.example.com z2.example.com
```

4. Créez un cluster nommé **my_cluster** avec les deux nœuds comme membres du cluster. Cette commande crée et démarre la grappe en une seule étape. Vous ne devez l'exécuter qu'à partir d'un seul nœud de la grappe, car les commandes de configuration de **pcs** s'appliquent à l'ensemble de la grappe.
Sur un nœud de la grappe, exécutez la commande suivante.

```
# pcs cluster setup my_cluster --start z1.example.com z2.example.com
```

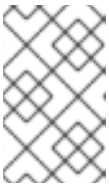
5. Un cluster Red Hat High Availability nécessite que vous configuriez la clôture pour le cluster. Les raisons de cette exigence sont décrites dans la section Clôture dans un cluster Red Hat High Availability. Pour cette introduction, cependant, afin de montrer uniquement comment le basculement fonctionne dans cette configuration, désactivez la clôture en définissant l'option **stonith-enabled** cluster sur **false**

AVERTISSEMENT

L'utilisation de **stonith-enabled=false** est totalement inappropriée pour un cluster de production. Elle indique au cluster de simplement prétendre que les nœuds défaillants sont clôturés en toute sécurité.

```
# pcs property set stonith-enabled=false
```

6. Après avoir créé une grappe et désactivé la clôture, vérifiez l'état de la grappe.



NOTE

Lorsque vous exécutez la commande **pcs cluster status**, il se peut que la sortie diffère légèrement des exemples au fur et à mesure que les composants du système démarrent.

```
# pcs cluster status
Cluster Status:
 Stack: corosync
 Current DC: z1.example.com (version 2.0.0-10.el8-b67d8d0de9) - partition with quorum
 Last updated: Thu Oct 11 16:11:18 2018
 Last change: Thu Oct 11 16:11:00 2018 by hacluster via crmd on z1.example.com
 2 nodes configured
 0 resources configured

PCSD Status:
  z1.example.com: Online
  z2.example.com: Online
```

7. Sur les deux nœuds, configurez un navigateur web et créez une page web pour afficher un simple message texte. Si vous exécutez le démon **firewalld**, activez les ports requis par **httpd**.

NOTE

N'utilisez pas **systemctl enable** pour permettre aux services qui seront gérés par le cluster de démarrer au démarrage du système.

```
# dnf install -y httpd wget
...
# firewall-cmd --permanent --add-service=http
# firewall-cmd --reload

# cat <<-END >/var/www/html/index.html
<html>
<body>My Test Site - $(hostname)</body>
</html>
END
```

Pour que l'agent de ressources Apache puisse obtenir l'état d'Apache, sur chaque nœud du cluster, créez l'ajout suivant à la configuration existante pour activer l'URL du serveur d'état.

```
# cat <<-END > /etc/httpd/conf.d/status.conf
<Location /server-status>
SetHandler server-status
Order deny,allow
Deny from all
Allow from 127.0.0.1
Allow from ::1
</Location>
END
```

8. Créez les ressources **IPaddr2** et **apache** pour le cluster à gérer. La ressource "IPaddr2" est une adresse IP flottante qui ne doit pas être déjà associée à un nœud physique. Si le périphérique NIC de la ressource "IPaddr2" n'est pas spécifié, l'adresse IP flottante doit résider sur le même réseau que l'adresse IP statiquement assignée utilisée par le nœud.
Vous pouvez afficher une liste de tous les types de ressources disponibles à l'aide de la commande **pcs resource list**. Vous pouvez utiliser la commande **pcs resource describe *resourcetype*** pour afficher les paramètres que vous pouvez définir pour le type de ressource spécifié. Par exemple, la commande suivante affiche les paramètres que vous pouvez définir pour une ressource de type **apache**:

```
# pcs resource describe apache
...
```

Dans cet exemple, la ressource adresse IP et la ressource apache sont toutes deux configurées comme faisant partie d'un groupe nommé **apachegroup**, ce qui garantit que les ressources sont conservées ensemble pour fonctionner sur le même nœud.

Exécutez les commandes suivantes à partir d'un nœud de la grappe :

```
# pcs resource create ClusterIP ocf:heartbeat:IPaddr2 ip=192.168.122.120 --group
apachegroup

# pcs resource create WebSite ocf:heartbeat:apache
configfile=/etc/httpd/conf/httpd.conf statusurl="http://localhost/server-status" --group
apachegroup

# pcs status
Cluster name: my_cluster
Stack: corosync
Current DC: z1.example.com (version 2.0.0-10.el8-b67d8d0de9) - partition with quorum
Last updated: Fri Oct 12 09:54:33 2018
Last change: Fri Oct 12 09:54:30 2018 by root via cibadmin on z1.example.com

2 nodes configured
2 resources configured

Online: [ z1.example.com z2.example.com ]

Full list of resources:

Resource Group: apachegroup
   ClusterIP  (ocf::heartbeat:IPaddr2):       Started z1.example.com
   WebSite    (ocf::heartbeat:apache):        Started z1.example.com

PCSD Status:
  z1.example.com: Online
  z2.example.com: Online
...
```

Notez que dans cet exemple, le service **apachegroup** est exécuté sur le nœud z1.example.com.

9. Accédez au site web que vous avez créé, arrêtez le service sur le nœud sur lequel il s'exécute et notez comment le service bascule sur le deuxième nœud.

   a. Dirigez un navigateur vers le site web que vous avez créé à l'aide de l'adresse IP flottante que vous avez configurée. Celui-ci doit afficher le message textuel que vous avez défini, en indiquant le nom du nœud sur lequel le site web est exécuté.

   b. Arrêtez le service web apache. L'utilisation de **killall -9** simule une panne au niveau de l'application.

      ```
      # killall -9 httpd
      ```

Vérifiez l'état de la grappe. Vous devriez constater que l'arrêt du service web a provoqué un échec, mais que le logiciel du cluster a redémarré le service sur le nœud sur lequel il s'exécutait et que vous devriez toujours pouvoir accéder au navigateur web.

```
# pcs status
Cluster name: my_cluster
Stack: corosync
Current DC: z1.example.com (version 2.0.0-10.el8-b67d8d0de9) - partition with quorum
Last updated: Fri Oct 12 09:54:33 2018
Last change: Fri Oct 12 09:54:30 2018 by root via cibadmin on z1.example.com

2 nodes configured
2 resources configured

Online: [ z1.example.com z2.example.com ]

Full list of resources:

Resource Group: apachegroup
   ClusterIP  (ocf::heartbeat:IPaddr2):       Started z1.example.com
   WebSite    (ocf::heartbeat:apache):        Started z1.example.com

Failed Resource Actions:
* WebSite_monitor_60000 on z1.example.com 'not running' (7): call=31,
status=complete, exitreason='none',
   last-rc-change='Fri Feb  5 21:01:41 2016', queued=0ms, exec=0ms
```

Effacez l'état d'échec une fois que le service est à nouveau opérationnel.

```
# pcs resource cleanup WebSite
```

c. Mettez le nœud sur lequel le service s'exécute en mode veille. Notez que, puisque nous avons désactivé la clôture, nous ne pouvons pas simuler efficacement une défaillance au niveau du nœud (comme le retrait d'un câble d'alimentation), car la clôture est nécessaire pour que le cluster se rétablisse dans de telles situations.

```
# pcs node standby z1.example.com
```

d. Vérifiez l'état de la grappe et notez où le service s'exécute maintenant.

```
# pcs status
Cluster name: my_cluster
Stack: corosync
Current DC: z1.example.com (version 2.0.0-10.el8-b67d8d0de9) - partition with quorum
Last updated: Fri Oct 12 09:54:33 2018
Last change: Fri Oct 12 09:54:30 2018 by root via cibadmin on z1.example.com

2 nodes configured
2 resources configured

Node z1.example.com: standby
Online: [ z2.example.com ]

Full list of resources:
```

```
Resource Group: apachegroup
    ClusterIP  (ocf::heartbeat:IPaddr2):       Started z2.example.com
    WebSite    (ocf::heartbeat:apache):        Started z2.example.com
```

e. Accédez au site web. Il ne devrait pas y avoir de perte de service, mais le message d'affichage devrait indiquer le nœud sur lequel le service est maintenant exécuté.

10. Pour restaurer les services de cluster sur le premier nœud, sortez le nœud du mode veille. Cela ne ramènera pas nécessairement le service sur ce nœud.

```
# pcs node unstandby z1.example.com
```

11. Pour le nettoyage final, arrêtez les services de cluster sur les deux nœuds.

```
# pcs cluster stop --all
```

CHAPITRE 3. L'INTERFACE DE LIGNE DE COMMANDE PCS

L'interface de ligne de commande **pcs** contrôle et configure les services de cluster tels que **corosync**, **pacemaker**,**booth** et **sbd** en facilitant l'accès à leurs fichiers de configuration.

Notez que vous ne devez pas modifier directement le fichier de configuration **cib.xml**. Dans la plupart des cas, Pacemaker rejettera un fichier **cib.xml** directement modifié.

3.1. AFFICHAGE DE L'AIDE PCS

L'option **-h** de **pcs** permet d'afficher les paramètres d'une commande **pcs** et une description de ces paramètres.

La commande suivante affiche les paramètres de la commande **pcs resource**.

```
# pcs resource -h
```

3.2. VISUALISATION DE LA CONFIGURATION BRUTE DU CLUSTER

Bien que vous ne deviez pas éditer directement le fichier de configuration de la grappe, vous pouvez visualiser la configuration brute de la grappe à l'aide de la commande **pcs cluster cib**.

Vous pouvez enregistrer la configuration brute du cluster dans un fichier spécifié avec la commande **pcs cluster cib *filename*** pour enregistrer la configuration brute du cluster dans un fichier spécifié. Si vous avez déjà configuré un cluster et qu'il y a déjà une CIB active, vous utilisez la commande suivante pour enregistrer le fichier xml brut.

```
pcs cluster cib filename
```

Par exemple, la commande suivante enregistre le xml brut de la CIB dans un fichier nommé **testfile**.

```
# pcs cluster cib testfile
```

3.3. ENREGISTREMENT D'UNE MODIFICATION DE CONFIGURATION DANS UN FICHIER DE TRAVAIL

Lors de la configuration d'un cluster, vous pouvez enregistrer les modifications de configuration dans un fichier spécifié sans affecter le CIB actif. Cela vous permet de spécifier des mises à jour de configuration sans mettre immédiatement à jour la configuration du cluster en cours d'exécution avec chaque mise à jour individuelle.

Pour plus d'informations sur l'enregistrement de la CIB dans un fichier, reportez-vous à la section Visualisation de la configuration brute du cluster . Une fois que vous avez créé ce fichier, vous pouvez enregistrer les modifications de configuration dans ce fichier plutôt que dans la CIB active en utilisant l'option **-f** de la commande **pcs**. Lorsque vous avez terminé les modifications et que vous êtes prêt à mettre à jour le fichier CIB actif, vous pouvez pousser les mises à jour du fichier avec la commande **pcs cluster cib-push**.

Procédure

La procédure suivante est recommandée pour apporter des modifications au fichier CIB. Cette procédure crée une copie du fichier CIB original sauvegardé et apporte des modifications à cette copie. Lors du transfert de ces modifications dans le fichier CIB actif, cette procédure spécifie l'option **diff-**

against de la commande **pcs cluster cib-push** afin que seules les modifications entre le fichier d'origine et le fichier mis à jour soient transférées dans le fichier CIB. Cela permet aux utilisateurs d'effectuer en parallèle des modifications qui ne s'écrasent pas les unes les autres et de réduire la charge de travail de Pacemaker, qui n'a pas besoin d'analyser l'intégralité du fichier de configuration.

1. Sauvegarder la CIB active dans un fichier. Cet exemple enregistre la CIB dans un fichier nommé **original.xml**.

```
# pcs cluster cib original.xml
```

2. Copiez le fichier enregistré dans le fichier de travail que vous utiliserez pour les mises à jour de la configuration.

```
# cp original.xml updated.xml
```

3. Mettez à jour votre configuration si nécessaire. La commande suivante crée une ressource dans le fichier **updated.xml** mais n'ajoute pas cette ressource à la configuration du cluster en cours d'exécution.

```
# pcs -f updated.xml resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.120
op monitor interval=30s
```

4. Transférer le fichier mis à jour vers la CIB active, en précisant que vous ne transférez que les modifications apportées au fichier d'origine.

```
# pcs cluster cib-push updated.xml diff-against=original.xml
```

Vous pouvez également afficher tout le contenu actuel d'un fichier CIB à l'aide de la commande suivante.

```
pcs cluster cib-push filename
```

Lors du transfert du fichier CIB complet, Pacemaker vérifie la version et ne vous permet pas de transférer un fichier CIB plus ancien que celui qui se trouve déjà dans un cluster. Si vous devez mettre à jour l'ensemble du fichier CIB avec une version plus ancienne que celle qui se trouve actuellement dans le cluster, vous pouvez utiliser l'option **--config** de la commande **pcs cluster cib-push**.

```
pcs cluster cib-push --config filename
```

3.4. AFFICHAGE DE L'ÉTAT DE LA GRAPPE

Plusieurs commandes permettent d'afficher l'état d'un cluster et de ses composants.

Vous pouvez afficher l'état de la grappe et de ses ressources à l'aide de la commande suivante.

```
# pcs status
```

Vous pouvez afficher l'état d'un composant particulier de la grappe avec le paramètre *commands* de la commande **pcs status**, en spécifiant **resources**, **cluster**, **nodes** ou **pcsd**.

```
état des pcs commands
```

Par exemple, la commande suivante affiche l'état des ressources de la grappe.

```
# pcs status resources
```

La commande suivante affiche l'état de la grappe, mais pas ses ressources.

```
# pcs cluster status
```

3.5. AFFICHAGE DE LA CONFIGURATION COMPLÈTE DU CLUSTER

La commande suivante permet d'afficher l'intégralité de la configuration actuelle du cluster.

```
# pcs config
```

3.6. MODIFIER LE FICHIER COROSYNC.CONF AVEC LA COMMANDE PCS

Vous pouvez utiliser la commande **pcs** pour modifier les paramètres du fichier **corosync.conf**.

La commande suivante modifie les paramètres du fichier **corosync.conf**.

```
pcs cluster config update [transport pass:quotes[transport options]] [compression pass:quotes[compression options]] [crypto pass:quotes[crypto options]] [totem pass:quotes[totem options]] [--corosync_conf pass:quotes[path]]
```

L'exemple de commande suivant met à jour la valeur de transport **knet_pmtud_interval** et les valeurs totem **token** et **join**.

```
# pcs cluster config update transport knet_pmtud_interval=35 totem token=10000 join=100
```

Ressources supplémentaires

- Pour plus d'informations sur l'ajout et la suppression de nœuds d'une grappe existante, voir Gestion des nœuds de la gra ppe.
- Pour plus d'informations sur l'ajout et la modification de liens dans un cluster existant, voir Ajout et modification de liens dans un cluster existant.
- Pour plus d'informations sur la modification des options de quorum et la gestion des paramètres des périphériques de quorum dans une grappe, voir Configuration du quorum de la grappe et Configuration des périphériques de quorum.

3.7. AFFICHAGE DU FICHIER COROSYNC.CONF AVEC LA COMMANDE PCS

La commande suivante affiche le contenu du fichier de configuration du cluster **corosync.conf**.

```
# pcs cluster corosync
```

Vous pouvez imprimer le contenu du fichier **corosync.conf** dans un format lisible par l'homme à l'aide de la commande **pcs cluster config**, comme dans l'exemple suivant.

La sortie de cette commande inclut l'UUID du cluster si celui-ci a été créé dans RHEL 9.1 ou une version ultérieure, ou si l'UUID a été ajouté manuellement comme décrit dans la section Identifier les clusters par UUID.

```
[root@r8-node-01 ~]# pcs cluster config
Cluster Name: HACluster
Cluster UUID: ad4ae07dcafe4066b01f1cc9391f54f5
Transport: knet
Nodes:
  r8-node-01:
    Link 0 address: r8-node-01
    Link 1 address: 192.168.122.121
    nodeid: 1
  r8-node-02:
    Link 0 address: r8-node-02
    Link 1 address: 192.168.122.122
    nodeid: 2
Links:
  Link 1:
    linknumber: 1
    ping_interval: 1000
    ping_timeout: 2000
    pong_count: 5
Compression Options:
  level: 9
  model: zlib
  threshold: 150
Crypto Options:
  cipher: aes256
  hash: sha256
Totem Options:
  downcheck: 2000
  join: 50
  token: 10000
Quorum Device: net
  Options:
    sync_timeout: 2000
    timeout: 3000
  Model Options:
    algorithm: lms
    host: r8-node-03
  Heuristics:
    exec_ping: ping -c 1 127.0.0.1
```

Vous pouvez exécuter la commande **pcs cluster config show** avec l'option **--output-format=cmd** pour afficher les commandes de configuration **pcs** qui peuvent être utilisées pour recréer le fichier **corosync.conf** existant, comme dans l'exemple suivant.

```
[root@r8-node-01 ~]# pcs cluster config show --output-format=cmd
pcs cluster setup HACluster \
  r8-node-01 addr=r8-node-01 addr=192.168.122.121 \
  r8-node-02 addr=r8-node-02 addr=192.168.122.122 \
  transport \
```

```
knet \
  link \
    linknumber=1 \
    ping_interval=1000 \
    ping_timeout=2000 \
    pong_count=5 \
  compression \
    level=9 \
    model=zlib \
    threshold=150 \
  crypto \
    cipher=aes256 \
    hash=sha256 \
totem \
  downcheck=2000 \
  join=50 \
  token=10000
```

CHAPITRE 4. CRÉATION D'UN CLUSTER RED HAT À HAUTE DISPONIBILITÉ AVEC PACEMAKER

Créez un cluster Red Hat High Availability à deux nœuds à l'aide de l'interface de ligne de commande **pcs** en suivant la procédure suivante.

La configuration du cluster dans cet exemple nécessite que votre système comprenne les composants suivants :

- 2 nœuds, qui seront utilisés pour créer le cluster. Dans cet exemple, les nœuds utilisés sont **z1.example.com** et **z2.example.com**.
- Commutateurs de réseau pour le réseau privé. Nous recommandons, mais n'exigeons pas, un réseau privé pour la communication entre les nœuds de la grappe et les autres équipements de la grappe, tels que les commutateurs d'alimentation réseau et les commutateurs Fibre Channel.
- Un dispositif de clôture pour chaque nœud du cluster. Cet exemple utilise deux ports du commutateur d'alimentation APC avec un nom d'hôte **zapc.example.com**.

4.1. INSTALLATION DU LOGICIEL DE LA GRAPPE

Installez le logiciel de cluster et configurez votre système pour la création de cluster en suivant la procédure suivante.

Procédure

1. Sur chaque nœud du cluster, activez le référentiel de haute disponibilité correspondant à l'architecture de votre système. Par exemple, pour activer le référentiel de haute disponibilité pour un système x86_64, vous pouvez entrer la commande suivante **subscription-manager**:

   ```
   # subscription-manager repos --enable=rhel-9-for-x86_64-highavailability-rpms
   ```

2. Sur chaque nœud du cluster, installez les paquetages logiciels Red Hat High Availability Add-On ainsi que tous les agents de clôture disponibles dans le canal High Availability.

   ```
   # dnf install pcs pacemaker fence-agents-all
   ```

   Vous pouvez également installer les paquetages logiciels Red Hat High Availability Add-On ainsi que l'agent de clôture dont vous avez besoin à l'aide de la commande suivante.

   ```
   # dnf install pcs pacemaker fence-agents-model
   ```

   La commande suivante affiche une liste des agents de clôture disponibles.

   ```
   # rpm -q -a | grep fence
   fence-agents-rhevm-4.0.2-3.el7.x86_64
   fence-agents-ilo-mp-4.0.2-3.el7.x86_64
   fence-agents-ipmilan-4.0.2-3.el7.x86_64
   ...
   ```

AVERTISSEMENT

Après avoir installé les paquetages de Red Hat High Availability Add-On, vous devez vous assurer que vos préférences de mise à jour logicielle sont définies de manière à ce que rien ne soit installé automatiquement. L'installation sur un cluster en cours d'exécution peut provoquer des comportements inattendus. Pour plus d'informations, voir Pratiques recommandées pour l'application de mises à jour logicielles à un cluster RHEL High Availability ou Resilient Storage.

3. Si vous exécutez le démon **firewalld**, exécutez les commandes suivantes pour activer les ports requis par le module complémentaire de haute disponibilité de Red Hat.

NOTE

Vous pouvez déterminer si le démon **firewalld** est installé sur votre système à l'aide de la commande **rpm -q firewalld**. S'il est installé, vous pouvez déterminer s'il fonctionne avec la commande **firewall-cmd --state**.

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
```

NOTE

La configuration idéale du pare-feu pour les composants de la grappe dépend de l'environnement local, où il peut être nécessaire de prendre en compte des considérations telles que l'existence d'interfaces réseau multiples pour les nœuds ou la présence d'un pare-feu hors hôte. L'exemple présenté ici, qui ouvre les ports généralement requis par un cluster Pacemaker, doit être modifié pour s'adapter aux conditions locales. L'activation des ports pour le module complémentaire de haute disponibilité montre les ports à activer pour le module complémentaire de haute disponibilité de Red Hat et fournit une explication sur l'utilisation de chaque port.

4. Afin d'utiliser **pcs** pour configurer le cluster et communiquer entre les nœuds, vous devez définir un mot de passe sur chaque nœud pour l'ID utilisateur **hacluster**, qui est le compte d'administration **pcs**. Il est recommandé que le mot de passe de l'utilisateur **hacluster** soit le même sur chaque nœud.

```
# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

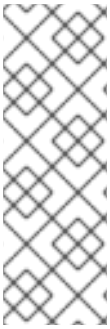
5. Avant de pouvoir configurer la grappe, le démon **pcsd** doit être lancé et autorisé à démarrer au démarrage sur chaque nœud. Ce démon travaille avec la commande **pcs** pour gérer la configuration des nœuds de la grappe.

Sur chaque nœud du cluster, exécutez les commandes suivantes pour démarrer le service **pcsd** et pour activer **pcsd** au démarrage du système.

```
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

4.2. INSTALLATION DU PAQUET PCP-ZEROCONF (RECOMMANDÉ)

Lorsque vous configurez votre cluster, il est recommandé d'installer le paquetage **pcp-zeroconf** pour l'outil Performance Co-Pilot (PCP). PCP est l'outil de surveillance des ressources recommandé par Red Hat pour les systèmes RHEL. L'installation du paquetage **pcp-zeroconf** vous permet d'exécuter PCP et de collecter des données de surveillance des performances dans le cadre d'enquêtes sur les clôtures, les défaillances de ressources et d'autres événements qui perturbent le cluster.



NOTE

Les déploiements de clusters où PCP est activé auront besoin de suffisamment d'espace pour les données capturées par PCP sur le système de fichiers qui contient **/var/log/pcp/**. L'utilisation typique de l'espace par PCP varie selon les déploiements, mais 10 Go suffisent généralement lorsque les paramètres par défaut de **pcp-zeroconf** sont utilisés, et certains environnements peuvent nécessiter moins d'espace. La surveillance de l'utilisation de ce répertoire sur une période de 14 jours d'activité typique peut fournir une estimation plus précise de l'utilisation.

Procédure

Pour installer le paquetage **pcp-zeroconf**, exécutez la commande suivante.

```
# dnf install pcp-zeroconf
```

Ce paquet permet d'activer **pmcd** et de configurer la capture de données à un intervalle de 10 secondes.

Pour plus d'informations sur l'examen des données PCP, voir Pourquoi un nœud de cluster RHEL High Availability a-t-il redémarré - et comment puis-je éviter que cela ne se reproduise ? sur le portail client de Red Hat.

4.3. CRÉATION D'UN CLUSTER DE HAUTE DISPONIBILITÉ

Créez un cluster Red Hat High Availability Add-On à l'aide de la procédure suivante. Cet exemple de procédure crée un cluster composé des nœuds **z1.example.com** et **z2.example.com**.

Procédure

1. Authentifiez l'utilisateur **pcs hacluster** pour chaque nœud du cluster sur le nœud à partir duquel vous exécuterez **pcs**.
   La commande suivante authentifie l'utilisateur **hacluster** sur **z1.example.com** pour les deux nœuds d'un cluster à deux nœuds qui comprendra **z1.example.com** et **z2.example.com**.

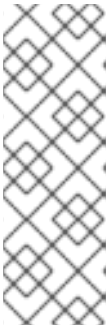
```
[root@z1 ~]# pcs host auth z1.example.com z2.example.com
Username: hacluster
Password:
```

```
z1.example.com: Authorized
z2.example.com: Authorized
```

2. Exécutez la commande suivante à partir de **z1.example.com** pour créer la grappe à deux nœuds **my_cluster** qui se compose des nœuds **z1.example.com** et **z2.example.com**. Les fichiers de configuration de la grappe seront propagés aux deux nœuds de la grappe. Cette commande inclut l'option **--start**, qui démarre les services de cluster sur les deux nœuds du cluster.

```
[root@z1 ~]# pcs cluster setup my_cluster --start z1.example.com z2.example.com
```

3. Activer les services de cluster pour qu'ils s'exécutent sur chaque nœud du cluster lorsque le nœud est démarré.



NOTE

Pour votre environnement particulier, vous pouvez choisir de laisser les services de la grappe désactivés en sautant cette étape. Cela vous permet de vous assurer qu'en cas de panne d'un nœud, tout problème lié à votre cluster ou à vos ressources est résolu avant que le nœud ne rejoigne le cluster. Si vous laissez les services de cluster désactivés, vous devrez les démarrer manuellement lors du redémarrage d'un nœud en exécutant la commande **pcs cluster start** sur ce nœud.

```
[root@z1 ~]# pcs cluster enable --all
```

Vous pouvez afficher l'état actuel de la grappe à l'aide de la commande **pcs cluster status**. Comme il peut y avoir un léger délai avant que la grappe soit opérationnelle lorsque vous démarrez les services de la grappe avec l'option **--start** de la commande **pcs cluster setup**, vous devez vous assurer que la grappe est opérationnelle avant d'effectuer toute action ultérieure sur la grappe et sa configuration.

```
[root@z1 ~]# pcs cluster status
Cluster Status:
 Stack: corosync
 Current DC: z2.example.com (version 2.0.0-10.el8-b67d8d0de9) - partition with quorum
 Last updated: Thu Oct 11 16:11:18 2018
 Last change: Thu Oct 11 16:11:00 2018 by hacluster via crmd on z2.example.com
 2 Nodes configured
 0 Resources configured

...
```

4.4. CRÉATION D'UN CLUSTER DE HAUTE DISPONIBILITÉ AVEC DES LIENS MULTIPLES

Vous pouvez utiliser la commande **pcs cluster setup** pour créer un cluster Red Hat High Availability avec des liens multiples en spécifiant tous les liens pour chaque nœud.

Le format de la commande de base pour créer un cluster à deux nœuds avec deux liens est le suivant.

```
pcs cluster setup pass:quotes[cluster_name] pass:quotes[node1_name]
addr=pass:quotes[node1_link0_address] addr=pass:quotes[node1_link1_address]
pass:quotes[node2_name] addr=pass:quotes[node2_link0_address]
```

```
addr=pass:quotes[node2_link1_address]
```

Pour la syntaxe complète de cette commande, voir la page de manuel **pcs**(8).

Lorsque vous créez une grappe avec plusieurs liens, vous devez tenir compte des éléments suivants.

- L'ordre des **addr=*address*** est important. La première adresse spécifiée après le nom d'un nœud est pour **link0**, la deuxième pour **link1**, et ainsi de suite.
- Par défaut, si **link_priority** n'est pas spécifié pour un lien, la priorité du lien est égale au numéro du lien. Les priorités des liens sont alors 0, 1, 2, 3, et ainsi de suite, selon l'ordre spécifié, 0 étant la priorité la plus élevée.
- Le mode de liaison par défaut est **passive**, ce qui signifie que la liaison active ayant la priorité de liaison la moins élevée est utilisée.
- Avec les valeurs par défaut de **link_mode** et **link_priority**, le premier lien spécifié sera utilisé comme lien de plus haute priorité, et si ce lien échoue, le lien suivant spécifié sera utilisé.
- Il est possible de spécifier jusqu'à huit liens en utilisant le protocole de transport **knet**, qui est le protocole de transport par défaut.
- Tous les nœuds doivent avoir le même nombre de paramètres **addr=**.
- Il est possible d'ajouter, de supprimer et de modifier des liens dans un cluster existant à l'aide des commandes **pcs cluster link add**, **pcs cluster link remove**, **pcs cluster link delete** et **pcs cluster link update**.
- Comme pour les clusters à lien unique, ne mélangez pas les adresses IPv4 et IPv6 dans un lien, bien qu'il soit possible d'avoir un lien fonctionnant sous IPv4 et l'autre sous IPv6.
- Comme pour les clusters à lien unique, vous pouvez spécifier des adresses sous forme d'adresses IP ou de noms tant que les noms se résolvent en adresses IPv4 ou IPv6 pour lesquelles les adresses IPv4 et IPv6 ne sont pas mélangées dans un lien.

L'exemple suivant crée un cluster à deux nœuds nommé **my_twolink_cluster** avec deux nœuds, **rh80-node1** et **rh80-node2**. **rh80-node1** possède deux interfaces, l'adresse IP 192.168.122.201 pour **link0** et 192.168.123.201 pour **link1**. **rh80-node2** possède deux interfaces, l'adresse IP 192.168.122.202 pour **link0** et 192.168.123.202 pour **link1**.

```
# pcs cluster setup my_twolink_cluster rh80-node1 addr=192.168.122.201
addr=192.168.123.201 rh80-node2 addr=192.168.122.202 addr=192.168.123.202
```

Pour définir une priorité de lien à une valeur différente de la valeur par défaut, qui est le numéro de lien, vous pouvez définir la priorité de lien avec l'option **link_priority** de la commande **pcs cluster setup**. Chacun des deux exemples de commande suivants crée un cluster à deux nœuds avec deux interfaces où le premier lien, le lien 0, a une priorité de lien de 1 et le second lien, le lien 1, a une priorité de lien de 0. Le lien 1 sera utilisé en premier et le lien 0 servira de lien de basculement. Le mode de liaison n'étant pas spécifié, il est par défaut passif.

Ces deux commandes sont équivalentes. Si vous ne spécifiez pas de numéro de lien après le mot-clé **link**, l'interface **pcs** ajoute automatiquement un numéro de lien, en commençant par le plus petit numéro de lien inutilisé.

```
# pcs cluster setup my_twolink_cluster rh80-node1 addr=192.168.122.201
addr=192.168.123.201 rh80-node2 addr=192.168.122.202 addr=192.168.123.202 transport knet
```

```
link link_priority=1 link link_priority=0

# pcs cluster setup my_twolink_cluster rh80-node1 addr=192.168.122.201
addr=192.168.123.201 rh80-node2 addr=192.168.122.202 addr=192.168.123.202 transport knet
link linknumber=1 link_priority=0 link link_priority=1
```

Vous pouvez définir le mode de liaison à une valeur différente de la valeur par défaut de **passive** avec l'option **link_mode** de la commande **pcs cluster setup**, comme dans l'exemple suivant.

```
# pcs cluster setup my_twolink_cluster rh80-node1 addr=192.168.122.201
addr=192.168.123.201 rh80-node2 addr=192.168.122.202 addr=192.168.123.202 transport knet
link_mode=active
```

L'exemple suivant définit à la fois le mode de liaison et la priorité de liaison.

```
# pcs cluster setup my_twolink_cluster rh80-node1 addr=192.168.122.201
addr=192.168.123.201 rh80-node2 addr=192.168.122.202 addr=192.168.123.202 transport knet
link_mode=active link link_priority=1 link link_priority=0
```

Pour plus d'informations sur l'ajout de nœuds à un cluster existant avec des liens multiples, voir Ajouter un nœud à un cluster avec des liens multiples.

Pour plus d'informations sur la modification des liens dans un cluster existant avec plusieurs liens, voir Ajouter et modifier des liens dans un cluster existant .

4.5. CONFIGURATION DES CLÔTURES

Vous devez configurer un dispositif de clôture pour chaque nœud du cluster. Pour plus d'informations sur les commandes et options de configuration de clôture, reportez-vous à Configuration de la clôture dans un cluster Red Hat High Availability.

Pour des informations générales sur la clôture et son importance dans un cluster Red Hat High Availability, voir Clôture dans un cluster Red Hat High Availability .

NOTE

Lors de la configuration d'un dispositif de clôture, il convient de vérifier si ce dispositif partage l'alimentation avec d'autres nœuds ou dispositifs de la grappe. Si un nœud et son dispositif de clôture partagent l'alimentation, la grappe risque de ne pas pouvoir clôturer ce nœud en cas de perte d'alimentation du nœud et de son dispositif de clôture. Une telle grappe doit disposer soit d'alimentations redondantes pour les dispositifs de clôture et les nœuds, soit de dispositifs de clôture redondants qui ne partagent pas l'alimentation. D'autres méthodes de clôture, telles que les SBD ou les clôtures de stockage, peuvent également apporter une redondance en cas de perte d'alimentation isolée.

Procédure

Cet exemple utilise l'interrupteur d'alimentation APC dont le nom d'hôte est **zapc.example.com** pour clôturer les nœuds, ainsi que l'agent de clôture **fence_apc_snmp**. Étant donné que les deux nœuds seront clôturés par le même agent de clôture, vous pouvez configurer les deux dispositifs de clôture en tant que ressource unique, à l'aide de l'option **pcmk_host_map**.

Vous créez un dispositif de clôture en le configurant en tant que ressource **stonith** à l'aide de la

commande **pcs stonith create**. La commande suivante configure une ressource **stonith** nommée **myapc** qui utilise l'agent de clôture **fence_apc_snmp** pour les nœuds **z1.example.com** et **z2.example.com**. L'option **pcmk_host_map** associe **z1.example.com** au port 1 et **z2.example.com** au port 2. La valeur de connexion et le mot de passe du dispositif APC sont tous deux **apc**. Par défaut, ce dispositif utilise un intervalle de surveillance de soixante secondes pour chaque nœud.

Notez que vous pouvez utiliser une adresse IP lorsque vous spécifiez le nom d'hôte des nœuds.

```
[root@z1 ~]# pcs stonith create myapc fence_apc_snmp ipaddr="zapc.example.com"
pcmk_host_map="z1.example.com:1;z2.example.com:2" login="apc" passwd="apc"
```

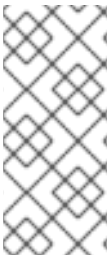
La commande suivante affiche les paramètres d'un dispositif STONITH existant.

```
[root@rh7-1 ~]# pcs stonith config myapc
 Resource: myapc (class=stonith type=fence_apc_snmp)
  Attributes: ipaddr=zapc.example.com pcmk_host_map=z1.example.com:1;z2.example.com:2
login=apc passwd=apc
  Operations: monitor interval=60s (myapc-monitor-interval-60s)
```

Après avoir configuré votre dispositif de clôture, vous devez le tester. Pour plus d'informations sur le test d'un dispositif de clôture, voir Test d'un dispositif de clôture .

NOTE

Ne testez pas votre dispositif de clôture en désactivant l'interface réseau, car cela ne permettra pas de tester correctement la clôture.



NOTE

Une fois que la clôture est configurée et qu'une grappe a été démarrée, un redémarrage du réseau déclenchera la clôture pour le nœud qui redémarre le réseau, même si le délai n'est pas dépassé. C'est pourquoi il ne faut pas redémarrer le service réseau lorsque le service de cluster est en cours d'exécution, car cela déclencherait un clôturage involontaire sur le nœud.

4.6. SAUVEGARDE ET RESTAURATION D'UNE CONFIGURATION DE CLUSTER

Les commandes suivantes sauvegardent la configuration d'un cluster dans une archive tar et restaurent les fichiers de configuration du cluster sur tous les nœuds à partir de la sauvegarde.

Procédure

Utilisez la commande suivante pour sauvegarder la configuration du cluster dans une archive tar. Si vous ne spécifiez pas de nom de fichier, la sortie standard sera utilisée.

```
pcs config backup filename
```

NOTE

La commande **pcs config backup** ne sauvegarde que la configuration du cluster telle qu'elle est configurée dans la CIB ; la configuration des démons de ressources n'entre pas dans le champ d'application de cette commande. Par exemple, si vous avez configuré une ressource Apache dans le cluster, les paramètres de la ressource (qui sont dans la CIB) seront sauvegardés, alors que les paramètres du démon Apache (tels qu'ils sont définis dans `/etc/httpd`) et les fichiers qu'il sert ne seront pas sauvegardés. De même, si une ressource de base de données est configurée dans le cluster, la base de données elle-même ne sera pas sauvegardée, mais la configuration de la ressource de base de données (CIB) le sera.

Utilisez la commande suivante pour restaurer les fichiers de configuration du cluster sur tous les nœuds du cluster à partir de la sauvegarde. L'option **--local** permet de restaurer les fichiers de configuration du cluster uniquement sur le nœud à partir duquel vous exécutez cette commande. Si vous ne spécifiez pas de nom de fichier, l'entrée standard sera utilisée.

```
pcs config restore [--local] [filename]
```

4.7. ACTIVATION DES PORTS POUR LE MODULE COMPLÉMENTAIRE DE HAUTE DISPONIBILITÉ

La configuration idéale du pare-feu pour les composants de la grappe dépend de l'environnement local, où il peut être nécessaire de prendre en compte des considérations telles que l'existence d'interfaces réseau multiples pour les nœuds ou la présence d'un pare-feu hors hôte.

Si vous exécutez le démon **firewalld**, exécutez les commandes suivantes pour activer les ports requis par le module complémentaire de haute disponibilité de Red Hat.

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
```

Il se peut que vous deviez modifier les ports ouverts en fonction des conditions locales.

NOTE

Vous pouvez déterminer si le démon **firewalld** est installé sur votre système à l'aide de la commande **rpm -q firewalld**. Si le démon **firewalld** est installé, vous pouvez déterminer s'il est en cours d'exécution à l'aide de la commande **firewall-cmd --state**.

Le tableau suivant indique les ports à activer pour le module complémentaire de haute disponibilité de Red Hat et fournit une explication de l'utilisation du port.

Tableau 4.1. Ports à activer pour le module complémentaire de haute disponibilité

Port	Quand cela est nécessaire

Port	Quand cela est nécessaire
TCP 2224	Port **pcsd** par défaut requis sur tous les nœuds (nécessaire pour l'interface Web pcsd et pour la communication entre nœuds). Vous pouvez configurer le port **pcsd** au moyen du paramètre **PCSD_PORT** dans le fichier **/etc/sysconfig/pcsd**. Il est essentiel d'ouvrir le port 2224 de manière à ce que **pcs** puisse communiquer avec tous les nœuds du cluster, y compris lui-même. Lorsque vous utilisez le gestionnaire de tickets du cluster Booth ou un dispositif de quorum, vous devez ouvrir le port 2224 sur tous les hôtes concernés, tels que les arbitres Booth ou l'hôte du dispositif de quorum.
TCP 3121	Requis sur tous les nœuds si la grappe comporte des nœuds Pacemaker Remote Le démon **pacemaker-based** de Pacemaker sur les nœuds de la grappe complète contactera le démon **pacemaker_remoted** sur les nœuds de Pacemaker Remote au port 3121. Si une interface distincte est utilisée pour la communication de la grappe, le port ne doit être ouvert que sur cette interface. Au minimum, le port doit être ouvert sur les nœuds distants de Pacemaker vers les nœuds de la grappe complète. Comme les utilisateurs peuvent convertir un hôte entre un nœud complet et un nœud distant, ou exécuter un nœud distant à l'intérieur d'un conteneur en utilisant le réseau de l'hôte, il peut être utile d'ouvrir le port à tous les nœuds. Il n'est pas nécessaire d'ouvrir le port à d'autres hôtes que les nœuds.
TCP 5403	Requis sur l'hôte du périphérique quorum lors de l'utilisation d'un périphérique quorum avec **corosync-qnetd**. La valeur par défaut peut être modifiée avec l'option **-p** de la commande **corosync-qnetd**.
UDP 5404-5412	Nécessaire sur les nœuds corosync pour faciliter la communication entre les nœuds. Il est essentiel d'ouvrir les ports 5404-5412 de manière à ce que **corosync**, quel que soit le nœud, puisse communiquer avec tous les nœuds de la grappe, y compris lui-même.
TCP 21064	Nécessaire sur tous les nœuds si le cluster contient des ressources nécessitant un DLM (telles que **GFS2**).

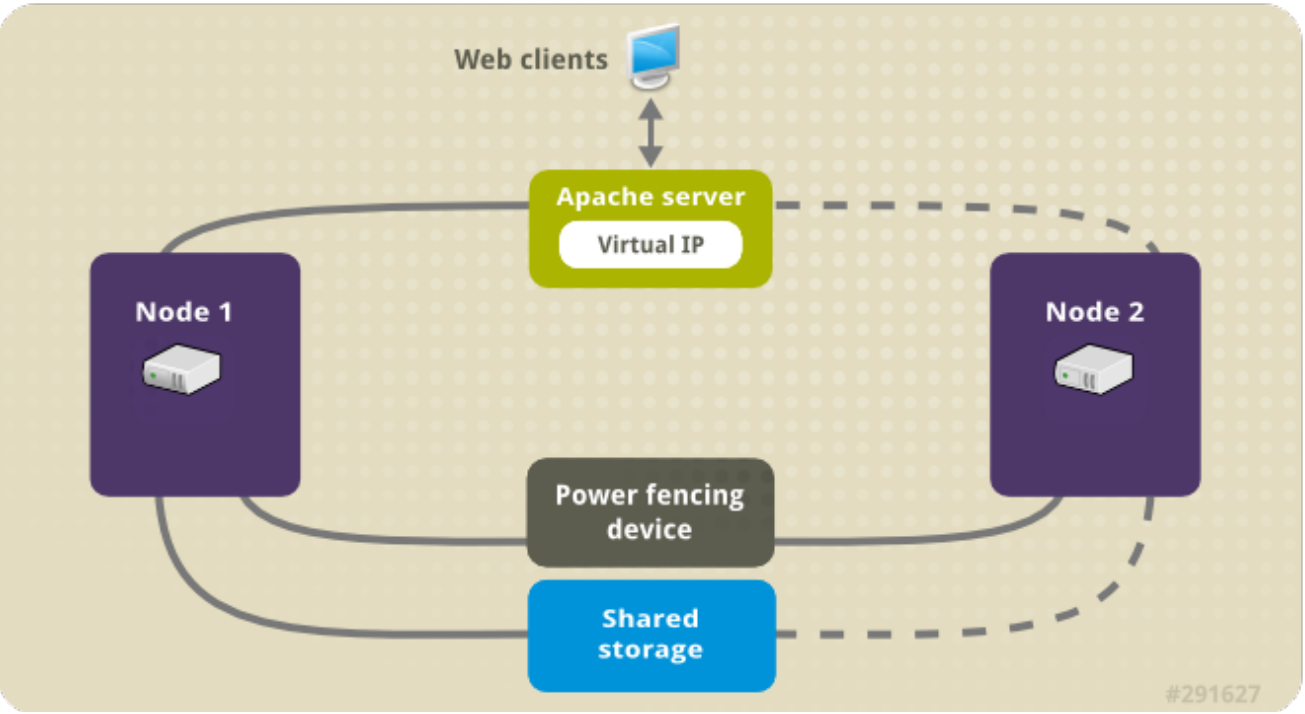
Port	Quand cela est nécessaire
TCP 9929, UDP 9929	Doit être ouvert sur tous les nœuds de la grappe et nœuds d'arbitrage pour les connexions à partir de n'importe lequel de ces nœuds lorsque le gestionnaire de tickets de cabine est utilisé pour établir une grappe multisite.

CHAPITRE 5. CONFIGURATION D'UN SERVEUR HTTP APACHE ACTIF/PASSIF DANS UN CLUSTER RED HAT HIGH AVAILABILITY

Configurez un serveur HTTP Apache actif/passif dans un cluster Red Hat Enterprise Linux High Availability Add-On à deux nœuds à l'aide de la procédure suivante. Dans ce cas d'utilisation, les clients accèdent au serveur HTTP Apache via une adresse IP flottante. Le serveur web s'exécute sur l'un des deux nœuds de la grappe. Si le nœud sur lequel le serveur web s'exécute devient inopérant, le serveur web redémarre sur le deuxième nœud du cluster avec une interruption de service minimale.

L'illustration suivante montre une vue d'ensemble du cluster dans lequel le cluster est un cluster Red Hat High Availability à deux nœuds qui est configuré avec un commutateur d'alimentation réseau et un stockage partagé. Les nœuds du cluster sont connectés à un réseau public, pour permettre aux clients d'accéder au serveur HTTP Apache par le biais d'une IP virtuelle. Le serveur Apache tourne sur le nœud 1 ou le nœud 2, chacun ayant accès au stockage sur lequel les données Apache sont conservées. Dans cette illustration, le serveur web tourne sur le nœud 1 tandis que le nœud 2 est disponible pour faire tourner le serveur si le nœud 1 devient inopérant.

Figure 5.1. Apache dans un cluster à deux nœuds Red Hat High Availability



Ce cas d'utilisation exige que votre système comprenne les composants suivants :

- Un cluster Red Hat High Availability à deux nœuds avec une clôture d'alimentation configurée pour chaque nœud. Nous recommandons l'utilisation d'un réseau privé, mais nous ne l'exigeons pas. Cette procédure utilise l'exemple de cluster fourni dans Création d'un cluster Red Hat à haute disponibilité avec Pacemaker.
- Une adresse IP virtuelle publique, requise pour Apache.
- Stockage partagé pour les nœuds de la grappe, à l'aide d'iSCSI, de Fibre Channel ou d'un autre périphérique de bloc réseau partagé.

La grappe est configurée avec un groupe de ressources Apache, qui contient les composants de la grappe dont le serveur web a besoin : une ressource LVM, une ressource de système de fichiers, une ressource d'adresse IP et une ressource de serveur web. Ce groupe de ressources peut basculer d'un

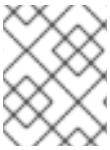
nœud à l'autre de la grappe, ce qui permet à l'un ou l'autre nœud d'exécuter le serveur web. Avant de créer le groupe de ressources pour cette grappe, vous devez effectuer les procédures suivantes :

1. Configurer un système de fichiers XFS sur le volume logique **my_lv**.
2. Configurer un serveur web.

Après avoir effectué ces étapes, vous créez le groupe de ressources et les ressources qu'il contient.

5.1. CONFIGURATION D'UN VOLUME LVM AVEC UN SYSTÈME DE FICHIERS XFS DANS UN CLUSTER PACEMAKER

Créez un volume logique LVM sur le stockage partagé entre les nœuds du cluster en suivant la procédure suivante.



NOTE

Les volumes LVM et les partitions et périphériques correspondants utilisés par les nœuds de cluster doivent être connectés aux nœuds de cluster uniquement.

La procédure suivante permet de créer un volume logique LVM, puis un système de fichiers XFS sur ce volume pour une utilisation dans un cluster Pacemaker. Dans cet exemple, la partition partagée **/dev/sdb1** est utilisée pour stocker le volume physique LVM à partir duquel le volume logique LVM sera créé.

Procédure

1. Sur les deux nœuds de la grappe, procédez comme suit pour définir la valeur de l'identifiant du système LVM sur la valeur de l'identifiant **uname** du système. L'ID système LVM sera utilisé pour s'assurer que seule la grappe est capable d'activer le groupe de volumes.
   a. Définissez l'option de configuration **system_id_source** dans le fichier de configuration **/etc/lvm/lvm.conf** sur **uname**.

   ```
   # Configuration option global/system_id_source.
   system_id_source = "uname"
   ```

   b. Vérifiez que l'ID du système LVM sur le nœud correspond à l'adresse **uname** du nœud.

   ```
   # lvm systemid
     system ID: z1.example.com
   # uname -n
     z1.example.com
   ```

2. Créez le volume LVM et créez un système de fichiers XFS sur ce volume. Étant donné que la partition **/dev/sdb1** est un espace de stockage partagé, cette partie de la procédure ne peut être exécutée que sur un seul nœud.

NOTE

Si votre groupe de volumes LVM contient un ou plusieurs volumes physiques résidant sur un stockage en bloc distant, tel qu'une cible iSCSI, Red Hat vous recommande de vous assurer que le service démarre avant le démarrage de Pacemaker. Pour plus d'informations sur la configuration de l'ordre de démarrage d'un volume physique distant utilisé par un cluster Pacemaker, reportez-vous à Configuration de l'ordre de démarrage pour les dépendances de ressources non gérées par Pacemaker.

a. Créer un volume physique LVM sur la partition **/dev/sdb1**.

```
[root@z1 ~]# pvcreate /dev/sdb1
  Physical volume "/dev/sdb1" successfully created
```

NOTE

Si votre groupe de volumes LVM contient un ou plusieurs volumes physiques résidant sur un stockage en bloc distant, tel qu'une cible iSCSI, Red Hat vous recommande de vous assurer que le service démarre avant le démarrage de Pacemaker. Pour plus d'informations sur la configuration de l'ordre de démarrage d'un volume physique distant utilisé par un cluster Pacemaker, reportez-vous à Configuration de l'ordre de démarrage pour les dépendances de ressources non gérées par Pacemaker.

b. Créez le groupe de volumes **my_vg** qui se compose du volume physique **/dev/sdb1**. Spécifiez l'indicateur **--setautoactivation n** pour vous assurer que les groupes de volumes gérés par Pacemaker dans un cluster ne seront pas automatiquement activés au démarrage. Si vous utilisez un groupe de volumes existant pour le volume LVM que vous créez, vous pouvez réinitialiser cet indicateur avec la commande **vgchange --setautoactivation n** pour le groupe de volumes.

```
[root@z1 ~]# vgcreate --setautoactivation n my_vg /dev/sdb1
  Volume group "my_vg" successfully created
```

c. Vérifiez que le nouveau groupe de volumes possède l'ID système du nœud sur lequel vous travaillez et à partir duquel vous avez créé le groupe de volumes.

```
[root@z1 ~]# vgs -o+systemid
  VG    #PV #LV #SN Attr   VSize  VFree  System ID
  my_vg   1   0   0 wz--n- <1.82t <1.82t z1.example.com
```

d. Créer un volume logique à l'aide du groupe de volumes **my_vg**.

```
[root@z1 ~]# lvcreate -L450 -n my_lv my_vg
  Rounding up size to full physical extent 452.00 MiB
  Logical volume "my_lv" created
```

Vous pouvez utiliser la commande **lvs** pour afficher le volume logique.

```
[root@z1 ~]# lvs
  LV      VG      Attr      LSize   Pool Origin Data%  Move Log Copy%  Convert
  my_lv   my_vg   -wi-a---- 452.00m
```

```
...
```

e. Créer un système de fichiers XFS sur le volume logique **my_lv**.

```
[root@z1 ~]# mkfs.xfs /dev/my_vg/my_lv
meta-data=/dev/my_vg/my_lv       isize=512    agcount=4, agsize=28928 blks
         =                       sectsz=512   attr=2, projid32bit=1
...
```

3. Ajoutez le périphérique partagé au fichier des périphériques LVM sur le deuxième nœud de la grappe.

```
[root@z2 ~]# lvmdevices --adddev /dev/sdb1
```

5.2. CONFIGURATION D'UN SERVEUR HTTP APACHE

Configurez un serveur HTTP Apache en suivant la procédure suivante.

Procédure

1. Assurez-vous que le serveur Apache HTTP est installé sur chaque nœud de la grappe. L'outil **wget** doit également être installé sur le cluster pour pouvoir vérifier l'état du serveur HTTP Apache.
   Sur chaque nœud, exécutez la commande suivante.

```
# dnf install -y httpd wget
```

   Si vous exécutez le démon **firewalld**, sur chaque nœud de la grappe, activez les ports requis par le module complémentaire de haute disponibilité de Red Hat et activez les ports dont vous aurez besoin pour exécuter **httpd**. Cet exemple active les ports **httpd** pour l'accès public, mais les ports spécifiques à activer pour **httpd** peuvent varier pour une utilisation en production.

```
# firewall-cmd --permanent --add-service=http
# firewall-cmd --permanent --zone=public --add-service=http
# firewall-cmd --reload
```

2. Pour que l'agent de ressources Apache puisse obtenir l'état d'Apache, sur chaque nœud du cluster, créez l'ajout suivant à la configuration existante pour activer l'URL du serveur d'état.

```
# cat <<-END > /etc/httpd/conf.d/status.conf
<Location /server-status>
   SetHandler server-status
   Require local
</Location>
END
```

3. Créez une page web pour qu'Apache la diffuse.
   Sur un nœud du cluster, assurez-vous que le volume logique que vous avez créé dans Configuration d'un volume LVM avec un système de fichiers XFS est activé, montez le système de fichiers que vous avez créé sur ce volume logique, créez le fichier **index.html** sur ce système de fichiers, puis démontez le système de fichiers.

```
# lvchange -ay my_vg/my_lv
# mount /dev/my_vg/my_lv /var/www/
# mkdir /var/www/html
# mkdir /var/www/cgi-bin
# mkdir /var/www/error
# restorecon -R /var/www
# cat <<-END >/var/www/html/index.html
<html>
<body>Hello</body>
</html>
END
# umount /var/www
```

5.3. CRÉATION DES RESSOURCES ET DES GROUPES DE RESSOURCES

Créez les ressources de votre cluster en suivant la procédure suivante. Pour s'assurer que ces ressources s'exécutent toutes sur le même nœud, elles sont configurées comme faisant partie du groupe de ressources **apachegroup**. Les ressources à créer sont les suivantes, énumérées dans l'ordre dans lequel elles démarreront.

1. Une ressource **LVM-activate** nommée **my_lvm** qui utilise le groupe de volumes LVM que vous avez créé dans Configuration d'un volume LVM avec un système de fichiers XFS .

2. Une ressource **Filesystem** nommée **my_fs**, qui utilise le périphérique de système de fichiers **/dev/my_vg/my_lv** que vous avez créé dans Configuration d'un volume LVM avec un système de fichiers XFS.

3. Une ressource **IPaddr2**, qui est une adresse IP flottante pour le groupe de ressources **apachegroup**. L'adresse IP ne doit pas être déjà associée à un nœud physique. Si le périphérique NIC de la ressource **IPaddr2** n'est pas spécifié, l'adresse IP flottante doit résider sur le même réseau que l'une des adresses IP attribuées de manière statique au nœud, faute de quoi le périphérique NIC qui attribue l'adresse IP flottante ne peut être correctement détecté.

4. Une ressource **apache** nommée **Website** qui utilise le fichier **index.html** et la configuration Apache que vous avez définie dans Configuration d'un serveur HTTP Apache .

La procédure suivante crée le groupe de ressources **apachegroup** et les ressources qu'il contient. Les ressources démarrent dans l'ordre dans lequel vous les ajoutez au groupe et s'arrêtent dans l'ordre inverse de celui dans lequel elles sont ajoutées au groupe. Exécutez cette procédure à partir d'un seul nœud de la grappe.

Procédure

1. La commande suivante crée la ressource **LVM-activate my_lvm** . Comme le groupe de ressources **apachegroup** n'existe pas encore, cette commande crée le groupe de ressources.

NOTE

Ne configurez pas plus d'une ressource **LVM-activate** utilisant le même groupe de volumes LVM dans une configuration HA active/passive, car cela pourrait entraîner une corruption des données. En outre, ne configurez pas une ressource **LVM-activate** en tant que ressource clone dans une configuration HA active/passive.

```
[root@z1 ~]# pcs resource create my_lvm ocf:heartbeat:LVM-activate vgname=my_vg
vg_access_mode=system_id --group apachegroup
```

Lorsque vous créez une ressource, celle-ci est démarrée automatiquement. Vous pouvez utiliser la commande suivante pour confirmer que la ressource a été créée et qu'elle a démarré.

```
# pcs resource status
 Resource Group: apachegroup
     my_lvm (ocf::heartbeat:LVM-activate): Started
```

Vous pouvez arrêter et démarrer manuellement une ressource individuelle à l'aide des commandes **pcs resource disable** et **pcs resource enable**.

2. Les commandes suivantes créent les ressources restantes pour la configuration, en les ajoutant au groupe de ressources existant **apachegroup**.

```
[root@z1 ~]# pcs resource create my_fs Filesystem device="/dev/my_vg/my_lv"
directory="/var/www" fstype="xfs" --group apachegroup

[root@z1 ~]# pcs resource create VirtualIP IPaddr2 ip=198.51.100.3 cidr_netmask=24 --
group apachegroup

[root@z1 ~]# pcs resource create Website apache
configfile="/etc/httpd/conf/httpd.conf" statusurl="http://127.0.0.1/server-status" --
group apachegroup
```

3. Après avoir créé les ressources et le groupe de ressources qui les contient, vous pouvez vérifier l'état du cluster. Notez que les quatre ressources s'exécutent sur le même nœud.

```
[root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Wed Jul 31 16:38:51 2013
Last change: Wed Jul 31 16:42:14 2013 via crm_attribute on z1.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
6 Resources configured

Online: [ z1.example.com z2.example.com ]

Full list of resources:
 myapc (stonith:fence_apc_snmp): Started z1.example.com
 Resource Group: apachegroup
     my_lvm (ocf::heartbeat:LVM-activate): Started z1.example.com
     my_fs (ocf::heartbeat:Filesystem): Started z1.example.com
     VirtualIP (ocf::heartbeat:IPaddr2): Started z1.example.com
     Website (ocf::heartbeat:apache): Started z1.example.com
```

Notez que si vous n'avez pas configuré de dispositif de clôture pour votre cluster, les ressources ne démarrent pas par défaut.

4. Une fois que le cluster est opérationnel, vous pouvez diriger un navigateur vers l'adresse IP que vous avez définie comme ressource **IPaddr2** pour voir l'exemple d'affichage, qui consiste en un simple mot "Hello".

```
Bonjour
```

Si vous constatez que les ressources que vous avez configurées ne fonctionnent pas, vous pouvez exécuter la commande **pcs resource debug-start *resource*** pour tester la configuration des ressources.

5. Lorsque vous utilisez l'agent de ressources **apache** pour gérer Apache, il n'utilise pas **systemd**. Pour cette raison, vous devez modifier le script **logrotate** fourni avec Apache afin qu'il n'utilise pas **systemctl** pour recharger Apache.
Supprimez la ligne suivante dans le fichier **/etc/logrotate.d/httpd** sur chaque nœud du cluster.

```
/bin/systemctl reload httpd.service > /dev/null 2>/dev/null || true
```

Remplacez la ligne que vous avez supprimée par les trois lignes suivantes, en spécifiant **/var/run/httpd-*website*.pid** comme chemin d'accès au fichier PID où *website* est le nom de la ressource Apache. Dans cet exemple, le nom de la ressource Apache est **Website**.

```
/usr/bin/test -f /var/run/httpd-Website.pid >/dev/null 2>/dev/null &&
/usr/bin/ps -q $(/usr/bin/cat /var/run/httpd-Website.pid) >/dev/null 2>/dev/null &&
/usr/sbin/httpd -f /etc/httpd/conf/httpd.conf -c "PidFile /var/run/httpd-Website.pid" -k graceful >
/dev/null 2>/dev/null || true
```

5.4. TEST DE LA CONFIGURATION DES RESSOURCES

Testez la configuration des ressources dans un cluster en suivant la procédure suivante.

Dans l'affichage de l'état du cluster présenté dans la section Création des ressources et des groupes de ressources, toutes les ressources sont exécutées sur le nœud **z1.example.com**. Vous pouvez tester si le groupe de ressources bascule sur le nœud **z2.example.com** en utilisant la procédure suivante pour mettre le premier nœud en mode **standby**, après quoi le nœud ne sera plus en mesure d'héberger des ressources.

Procédure

1. La commande suivante place le nœud **z1.example.com** en mode **standby**.

```
[root@z1 ~]# pcs node standby z1.example.com
```

2. Après avoir mis le nœud **z1** en mode **standby**, vérifiez l'état du cluster. Notez que les ressources devraient maintenant toutes être exécutées sur **z2**.

```
[root@z1 ~]# pcs status
Cluster name: my_cluster
Last updated: Wed Jul 31 17:16:17 2013
Last change: Wed Jul 31 17:18:34 2013 via crm_attribute on z1.example.com
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.10-5.el7-9abe687
2 Nodes configured
6 Resources configured

Node z1.example.com (1): standby
Online: [ z2.example.com ]
```

```
Full list of resources:

 myapc (stonith:fence_apc_snmp): Started z1.example.com
 Resource Group: apachegroup
     my_lvm (ocf::heartbeat:LVM-activate): Started z2.example.com
     my_fs (ocf::heartbeat:Filesystem): Started z2.example.com
     VirtualIP (ocf::heartbeat:IPaddr2): Started z2.example.com
     Website (ocf::heartbeat:apache): Started z2.example.com
```

Le site web à l'adresse IP définie doit s'afficher sans interruption.

3. Pour supprimer **z1** du mode **standby**, entrez la commande suivante.

```
[root@z1 ~]# pcs node unstandby z1.example.com
```

NOTE

Le retrait d'un nœud du mode **standby** n'entraîne pas en soi le basculement des ressources vers ce nœud. Cela dépend de la valeur de **resource-stickiness** pour les ressources. Pour plus d'informations sur le méta-attribut **resource-stickiness**, voir Configurer une ressource pour qu'elle préfère son nœud actuel .

CHAPITRE 6. CONFIGURATION D'UN SERVEUR NFS ACTIF/PASSIF DANS UN CLUSTER RED HAT HIGH AVAILABILITY

Le module complémentaire de haute disponibilité de Red Hat prend en charge l'exécution d'un serveur NFS actif/passif hautement disponible sur un cluster du module complémentaire de haute disponibilité de Red Hat Enterprise Linux à l'aide d'un stockage partagé. Dans l'exemple suivant, vous configurez un cluster à deux nœuds dans lequel les clients accèdent au système de fichiers NFS via une adresse IP flottante. Le serveur NFS s'exécute sur l'un des deux nœuds de la grappe. Si le nœud sur lequel le serveur NFS est exécuté devient inopérant, le serveur NFS redémarre sur le deuxième nœud de la grappe avec une interruption de service minimale.

Ce cas d'utilisation exige que votre système comprenne les composants suivants :

- Un cluster Red Hat High Availability à deux nœuds avec une clôture d'alimentation configurée pour chaque nœud. Nous recommandons l'utilisation d'un réseau privé, mais nous ne l'exigeons pas. Cette procédure utilise l'exemple de cluster fourni dans Création d'un cluster Red Hat à haute disponibilité avec Pacemaker.
- Une adresse IP virtuelle publique, requise pour le serveur NFS.
- Stockage partagé pour les nœuds de la grappe, à l'aide d'iSCSI, de Fibre Channel ou d'un autre périphérique de bloc réseau partagé.

La configuration d'un serveur NFS actif/passif hautement disponible sur un cluster Red Hat Enterprise Linux High Availability à deux nœuds existant requiert que vous effectuiez les étapes suivantes :

1. Configurez un système de fichiers sur un volume logique LVM sur le stockage partagé pour les nœuds du cluster.
2. Configurez un partage NFS sur le stockage partagé du volume logique LVM.
3. Créer les ressources du cluster.
4. Testez le serveur NFS que vous avez configuré.

6.1. CONFIGURATION D'UN VOLUME LVM AVEC UN SYSTÈME DE FICHIERS XFS DANS UN CLUSTER PACEMAKER

Créez un volume logique LVM sur le stockage partagé entre les nœuds du cluster en suivant la procédure suivante.

NOTE

Les volumes LVM et les partitions et périphériques correspondants utilisés par les nœuds de cluster doivent être connectés aux nœuds de cluster uniquement.

La procédure suivante permet de créer un volume logique LVM, puis un système de fichiers XFS sur ce volume pour une utilisation dans un cluster Pacemaker. Dans cet exemple, la partition partagée **/dev/sdb1** est utilisée pour stocker le volume physique LVM à partir duquel le volume logique LVM sera créé.

Procédure

1. Sur les deux nœuds de la grappe, procédez comme suit pour définir la valeur de l'identifiant du système LVM sur la valeur de l'identifiant **uname** du système. L'ID système LVM sera utilisé pour s'assurer que seule la grappe est capable d'activer le groupe de volumes.

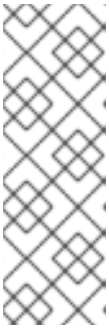
   a. Définissez l'option de configuration **system_id_source** dans le fichier de configuration **/etc/lvm/lvm.conf** sur **uname**.

   ```
   # Configuration option global/system_id_source.
   system_id_source = "uname"
   ```

   b. Vérifiez que l'ID du système LVM sur le nœud correspond à l'adresse **uname** du nœud.

   ```
   # lvm systemid
     system ID: z1.example.com
   # uname -n
     z1.example.com
   ```

2. Créez le volume LVM et créez un système de fichiers XFS sur ce volume. Étant donné que la partition **/dev/sdb1** est un espace de stockage partagé, cette partie de la procédure ne peut être exécutée que sur un seul nœud.



NOTE

Si votre groupe de volumes LVM contient un ou plusieurs volumes physiques résidant sur un stockage en bloc distant, tel qu'une cible iSCSI, Red Hat vous recommande de vous assurer que le service démarre avant le démarrage de Pacemaker. Pour plus d'informations sur la configuration de l'ordre de démarrage d'un volume physique distant utilisé par un cluster Pacemaker, reportez-vous à Configuration de l'ordre de démarrage pour les dépendances de ressources non gérées par Pacemaker.

   a. Créer un volume physique LVM sur la partition **/dev/sdb1**.

   ```
   [root@z1 ~]# pvcreate /dev/sdb1
     Physical volume "/dev/sdb1" successfully created
   ```

NOTE

Si votre groupe de volumes LVM contient un ou plusieurs volumes physiques résidant sur un stockage en bloc distant, tel qu'une cible iSCSI, Red Hat vous recommande de vous assurer que le service démarre avant le démarrage de Pacemaker. Pour plus d'informations sur la configuration de l'ordre de démarrage d'un volume physique distant utilisé par un cluster Pacemaker, reportez-vous à Configuration de l'ordre de démarrage pour les dépendances de ressources non gérées par Pacemaker.

   b. Créez le groupe de volumes **my_vg** qui se compose du volume physique **/dev/sdb1**. Spécifiez l'indicateur **--setautoactivation n** pour vous assurer que les groupes de volumes gérés par Pacemaker dans un cluster ne seront pas automatiquement activés au démarrage. Si vous utilisez un groupe de volumes existant pour le volume LVM que vous créez, vous pouvez réinitialiser cet indicateur avec la commande **vgchange --setautoactivation n** pour le groupe de volumes.

```
[root@z1 ~]# vgcreate --setautoactivation n my_vg /dev/sdb1
  Volume group "my_vg" successfully created
```

c. Vérifiez que le nouveau groupe de volumes possède l'ID système du nœud sur lequel vous travaillez et à partir duquel vous avez créé le groupe de volumes.

```
[root@z1 ~]# vgs -o+systemid
  VG    #PV #LV #SN Attr   VSize  VFree  System ID
  my_vg   1   0   0 wz--n- <1.82t <1.82t z1.example.com
```

d. Créer un volume logique à l'aide du groupe de volumes **my_vg**.

```
[root@z1 ~]# lvcreate -L450 -n my_lv my_vg
  Rounding up size to full physical extent 452.00 MiB
  Logical volume "my_lv" created
```

Vous pouvez utiliser la commande **lvs** pour afficher le volume logique.

```
[root@z1 ~]# lvs
  LV      VG      Attr      LSize   Pool Origin Data%  Move Log Copy%  Convert
  my_lv   my_vg   -wi-a---- 452.00m
  ...
```

e. Créer un système de fichiers XFS sur le volume logique **my_lv**.

```
[root@z1 ~]# mkfs.xfs /dev/my_vg/my_lv
meta-data=/dev/my_vg/my_lv       isize=512    agcount=4, agsize=28928 blks
         =                       sectsz=512   attr=2, projid32bit=1
...
```

3. Ajoutez le périphérique partagé au fichier des périphériques LVM sur le deuxième nœud de la grappe.

```
[root@z2 ~]# lvmdevices --adddev /dev/sdb1
```

6.2. CONFIGURATION D'UN PARTAGE NFS

Configurez un partage NFS pour un basculement de service NFS en suivant la procédure suivante.

Procédure

1. Sur les deux nœuds du cluster, créez le répertoire **/nfsshare**.

```
# mkdir /nfsshare
```

2. Sur un nœud de la grappe, effectuez la procédure suivante.

   a. Assurez-vous que le volume logique que vous avez créé dans Configuration d'un volume LVM avec un système de fichiers XFS est activé, puis montez le système de fichiers que vous avez créé sur le volume logique dans le répertoire **/nfsshare**.

```
[root@z1 ~]# lvchange -ay my_vg/my_lv
[root@z1 ~]# mount /dev/my_vg/my_lv /nfsshare
```

b. Créez une arborescence **exports** sur le répertoire **/nfsshare**.

```
[root@z1 ~]# mkdir -p /nfsshare/exports
[root@z1 ~]# mkdir -p /nfsshare/exports/export1
[root@z1 ~]# mkdir -p /nfsshare/exports/export2
```

c. Placez les fichiers dans le répertoire **exports** pour que les clients NFS puissent y accéder. Pour cet exemple, nous créons des fichiers de test nommés **clientdatafile1** et **clientdatafile2**.

```
[root@z1 ~]# touch /nfsshare/exports/export1/clientdatafile1
[root@z1 ~]# touch /nfsshare/exports/export2/clientdatafile2
```

d. Démontez le système de fichiers et désactivez le groupe de volumes LVM.

```
[root@z1 ~]# umount /dev/my_vg/my_lv
[root@z1 ~]# vgchange -an my_vg
```

6.3. CONFIGURATION DES RESSOURCES ET DU GROUPE DE RESSOURCES POUR UN SERVEUR NFS DANS UN CLUSTER

Configurez les ressources du cluster pour un serveur NFS dans un cluster en suivant la procédure suivante.

NOTE

Si vous n'avez pas configuré de dispositif de clôture pour votre cluster, les ressources ne démarrent pas par défaut.

Si vous constatez que les ressources que vous avez configurées ne fonctionnent pas, vous pouvez exécuter la commande **pcs resource debug-start *resource*** pour tester la configuration des ressources. Cela permet de démarrer le service en dehors du contrôle et de la connaissance du cluster. Lorsque les ressources configurées fonctionnent à nouveau, exécutez la commande **pcs resource cleanup *resource*** pour informer le cluster des mises à jour.

Procédure

La procédure suivante permet de configurer les ressources du système. Pour s'assurer que ces ressources s'exécutent toutes sur le même nœud, elles sont configurées dans le cadre du groupe de ressources **nfsgroup**. Les ressources démarrent dans l'ordre dans lequel vous les ajoutez au groupe et s'arrêtent dans l'ordre inverse de celui dans lequel elles sont ajoutées au groupe. Exécutez cette procédure à partir d'un seul nœud de la grappe.

1. Créez la ressource LVM-activate nommée **my_lvm**. Comme le groupe de ressources **nfsgroup** n'existe pas encore, cette commande crée le groupe de ressources.

AVERTISSEMENT

Ne configurez pas plus d'une ressource **LVM-activate** utilisant le même groupe de volumes LVM dans une configuration HA active/passive, car les données risquent d'être corrompues. En outre, ne configurez pas une ressource **LVM-activate** en tant que ressource clone dans une configuration HA active/passive.

```
[root@z1 ~]# pcs resource create my_lvm ocf:heartbeat:LVM-activate vgname=my_vg
vg_access_mode=system_id --group nfsgroup
```

2. Vérifiez l'état de la grappe pour vous assurer que la ressource est en cours d'exécution.

```
root@z1 ~]#  pcs status
Cluster name: my_cluster
Last updated: Thu Jan  8 11:13:17 2015
Last change: Thu Jan  8 11:13:08 2015
Stack: corosync
Current DC: z2.example.com (2) - partition with quorum
Version: 1.1.12-a14efad
2 Nodes configured
3 Resources configured

Online: [ z1.example.com z2.example.com ]

Full list of resources:
 myapc  (stonith:fence_apc_snmp):       Started z1.example.com
 Resource Group: nfsgroup
     my_lvm     (ocf::heartbeat:LVM-activate):   Started z1.example.com

PCSD Status:
  z1.example.com: Online
  z2.example.com: Online

Daemon Status:
  corosync: active/enabled
  pacemaker: active/enabled
  pcsd: active/enabled
```

3. Configurez une ressource **Filesystem** pour le cluster.
La commande suivante configure une ressource XFS **Filesystem** nommée **nfsshare** en tant que partie du groupe de ressources **nfsgroup**. Ce système de fichiers utilise le groupe de volumes LVM et le système de fichiers XFS que vous avez créés dans Configuration d'un volume LVM avec un système de fichiers XFS et sera monté sur le répertoire **/nfsshare** que vous avez créé dans Configuration d'un partage NFS .

```
[root@z1 ~]# pcs resource create nfsshare Filesystem device=/dev/my_vg/my_lv
directory=/nfsshare fstype=xfs --group nfsgroup
```

Vous pouvez spécifier des options de montage dans le cadre de la configuration d'une

ressource **Filesystem** à l'aide du paramètre **options=*options*** paramètre. Exécutez la commande **pcs resource describe Filesystem** pour obtenir toutes les options de configuration.

4. Vérifiez que les ressources **my_lvm** et **nfsshare** sont en cours d'exécution.

```
[root@z1 ~]# pcs status
...
Full list of resources:
 myapc  (stonith:fence_apc_snmp):       Started z1.example.com
 Resource Group: nfsgroup
     my_lvm     (ocf::heartbeat:LVM-activate):   Started z1.example.com
     nfsshare   (ocf::heartbeat:Filesystem):    Started z1.example.com
...
```

5. Créez la ressource **nfsserver** nommée **nfs-daemon** en tant que partie du groupe de ressources **nfsgroup**.

NOTE

La ressource **nfsserver** vous permet de spécifier un paramètre **nfs_shared_infodir**, qui est un répertoire que les serveurs NFS utilisent pour stocker les informations d'état liées à NFS.

Il est recommandé de définir cet attribut sur un sous-répertoire de l'une des ressources **Filesystem** que vous avez créées dans cette collection d'exportations. Cela permet de s'assurer que les serveurs NFS stockent leurs informations sur un périphérique qui sera disponible pour un autre nœud si ce groupe de ressources doit être déplacé. Dans cet exemple ;

- **/nfsshare** est le répertoire de stockage partagé géré par la ressource **Filesystem**
- **/nfsshare/exports/export1** et **/nfsshare/exports/export2** sont les répertoires d'exportation
- **/nfsshare/nfsinfo** est le répertoire d'informations partagées pour la ressource **nfsserver**

```
[root@z1 ~]# pcs resource create nfs-daemon nfsserver
nfs_shared_infodir=/nfsshare/nfsinfo nfs_no_notify=true --group nfsgroup

[root@z1 ~]# pcs status
...
```

6. Ajoutez les ressources **exportfs** pour exporter le répertoire **/nfsshare/exports**. Ces ressources font partie du groupe de ressources **nfsgroup**. Cela permet de créer un répertoire virtuel pour les clients NFSv4. Les clients NFSv3 peuvent également accéder à ces exportations.

NOTE

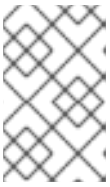
L'option **fsid=0** n'est nécessaire que si vous souhaitez créer un répertoire virtuel pour les clients NFSv4. Pour plus d'informations, voir Comment configurer l'option fsid dans le fichier /etc/exports d'un serveur NFS ?

```
[root@z1 ~]# pcs resource create nfs-root exportfs
clientspec=192.168.122.0/255.255.255.0 options=rw,sync,no_root_squash
directory=/nfsshare/exports fsid=0 --group nfsgroup

[root@z1 ~]# pcs resource create nfs-export1 exportfs
clientspec=192.168.122.0/255.255.255.0 options=rw,sync,no_root_squash
directory=/nfsshare/exports/export1 fsid=1 --group nfsgroup

[root@z1 ~]# pcs resource create nfs-export2 exportfs
clientspec=192.168.122.0/255.255.255.0 options=rw,sync,no_root_squash
directory=/nfsshare/exports/export2 fsid=2 --group nfsgroup
```

7. Ajoutez la ressource d'adresse IP flottante que les clients NFS utiliseront pour accéder au partage NFS. Cette ressource fait partie du groupe de ressources **nfsgroup**. Pour cet exemple de déploiement, nous utilisons 192.168.122.200 comme adresse IP flottante.

```
[root@z1 ~]# pcs resource create nfs_ip IPaddr2 ip=192.168.122.200 cidr_netmask=24 -
-group nfsgroup
```

8. Ajouter une ressource **nfsnotify** pour envoyer des notifications de redémarrage NFSv3 une fois que l'ensemble du déploiement NFS a été initialisé. Cette ressource fait partie du groupe de ressources **nfsgroup**.



NOTE

Pour que la notification NFS soit traitée correctement, l'adresse IP flottante doit être associée à un nom d'hôte cohérent à la fois sur les serveurs NFS et sur le client NFS.

```
[root@z1 ~]# pcs resource create nfs-notify nfsnotify source_host=192.168.122.200 --
group nfsgroup
```

9. Après avoir créé les ressources et les contraintes de ressources, vous pouvez vérifier l'état du cluster. Notez que toutes les ressources sont exécutées sur le même nœud.

```
[root@z1 ~]# pcs status
...
Full list of resources:
 myapc  (stonith:fence_apc_snmp):       Started z1.example.com
 Resource Group: nfsgroup
     my_lvm     (ocf::heartbeat:LVM-activate):   Started z1.example.com
     nfsshare   (ocf::heartbeat:Filesystem):    Started z1.example.com
     nfs-daemon (ocf::heartbeat:nfsserver):     Started z1.example.com
     nfs-root   (ocf::heartbeat:exportfs):      Started z1.example.com
     nfs-export1        (ocf::heartbeat:exportfs):      Started z1.example.com
     nfs-export2        (ocf::heartbeat:exportfs):      Started z1.example.com
     nfs_ip     (ocf::heartbeat:IPaddr2):       Started  z1.example.com
     nfs-notify (ocf::heartbeat:nfsnotify):     Started z1.example.com
...
```

6.4. TEST DE LA CONFIGURATION DES RESSOURCES NFS

Vous pouvez valider votre configuration de ressources NFS dans un cluster à haute disponibilité à l'aide des procédures suivantes. Vous devez être en mesure de monter le système de fichiers exporté avec NFSv3 ou NFSv4.

6.4.1. Test de l'exportation NFS

1. Si vous exécutez le démon **firewalld** sur les nœuds de votre cluster, assurez-vous que les ports dont votre système a besoin pour l'accès NFS sont activés sur tous les nœuds.

2. Sur un nœud extérieur au cluster, résidant sur le même réseau que le déploiement, vérifiez que le partage NFS peut être vu en montant le partage NFS. Pour cet exemple, nous utilisons le réseau 192.168.122.0/24.

```
# showmount -e 192.168.122.200
Export list for 192.168.122.200:
/nfsshare/exports/export1 192.168.122.0/255.255.255.0
/nfsshare/exports         192.168.122.0/255.255.255.0
/nfsshare/exports/export2 192.168.122.0/255.255.255.0
```

3. Pour vérifier que vous pouvez monter le partage NFS avec NFSv4, montez le partage NFS dans un répertoire sur le nœud client. Après le montage, vérifiez que le contenu des répertoires d'exportation est visible. Démontez le partage après le test.

```
# mkdir nfsshare
# mount -o "vers=4" 192.168.122.200:export1 nfsshare
# ls nfsshare
clientdatafile1
# umount nfsshare
```

4. Vérifiez que vous pouvez monter le partage NFS avec NFSv3. Après le montage, vérifiez que le fichier de test **clientdatafile1** est visible. Contrairement à NFSv4, NFSv3 n'utilise pas le système de fichiers virtuels, vous devez donc monter une exportation spécifique. Démontez le partage après le test.

```
# mkdir nfsshare
# mount -o "vers=3" 192.168.122.200:/nfsshare/exports/export2 nfsshare
# ls nfsshare
clientdatafile2
# umount nfsshare
```

6.4.2. Test de basculement

1. Sur un nœud extérieur au cluster, montez le partage NFS et vérifiez l'accès au fichier **clientdatafile1** que vous avez créé dans Configuration d'un partage NFS.

```
# mkdir nfsshare
# mount -o "vers=4" 192.168.122.200:export1 nfsshare
# ls nfsshare
clientdatafile1
```

2. Depuis un nœud du cluster, déterminez le nœud du cluster qui exécute **nfsgroup**. Dans cet exemple, **nfsgroup** est exécuté sur **z1.example.com**.

```
[root@z1 ~]# pcs status
```

```
...
Full list of resources:
 myapc  (stonith:fence_apc_snmp):       Started z1.example.com
 Resource Group: nfsgroup
     my_lvm     (ocf::heartbeat:LVM-activate):   Started z1.example.com
     nfsshare   (ocf::heartbeat:Filesystem):    Started z1.example.com
     nfs-daemon (ocf::heartbeat:nfsserver):     Started z1.example.com
     nfs-root   (ocf::heartbeat:exportfs):      Started z1.example.com
     nfs-export1        (ocf::heartbeat:exportfs):      Started z1.example.com
     nfs-export2        (ocf::heartbeat:exportfs):      Started z1.example.com
     nfs_ip     (ocf::heartbeat:IPaddr2):       Started  z1.example.com
     nfs-notify (ocf::heartbeat:nfsnotify):     Started z1.example.com
...
```

3. À partir d'un nœud du cluster, mettez le nœud qui exécute **nfsgroup** en mode veille.

```
[root@z1 ~]# pcs node standby z1.example.com
```

4. Vérifiez que **nfsgroup** démarre correctement sur l'autre nœud du cluster.

```
[root@z1 ~]# pcs status
...
Full list of resources:
 Resource Group: nfsgroup
     my_lvm     (ocf::heartbeat:LVM-activate):   Started z2.example.com
     nfsshare   (ocf::heartbeat:Filesystem):    Started z2.example.com
     nfs-daemon (ocf::heartbeat:nfsserver):     Started z2.example.com
     nfs-root   (ocf::heartbeat:exportfs):      Started z2.example.com
     nfs-export1        (ocf::heartbeat:exportfs):      Started z2.example.com
     nfs-export2        (ocf::heartbeat:exportfs):      Started z2.example.com
     nfs_ip     (ocf::heartbeat:IPaddr2):       Started  z2.example.com
     nfs-notify (ocf::heartbeat:nfsnotify):     Started z2.example.com
...
```

5. À partir du nœud extérieur au cluster sur lequel vous avez monté le partage NFS, vérifiez que ce nœud extérieur continue d'avoir accès au fichier de test dans le montage NFS.

```
# ls nfsshare
clientdatafile1
```

   Le service sera brièvement interrompu pour le client pendant le basculement, mais le client devrait le rétablir sans intervention de l'utilisateur. Par défaut, les clients utilisant NFSv4 peuvent prendre jusqu'à 90 secondes pour récupérer le montage ; ces 90 secondes représentent la période de grâce du bail de fichier NFSv4 observée par le serveur au démarrage. Les clients NFSv3 devraient récupérer l'accès au montage en quelques secondes.

6. À partir d'un nœud du cluster, retirez le nœud qui exécutait initialement **nfsgroup** du mode veille.

NOTE

Le retrait d'un nœud du mode **standby** n'entraîne pas en soi le basculement des ressources vers ce nœud. Cela dépend de la valeur de **resource-stickiness** pour les ressources. Pour plus d'informations sur le méta-attribut **resource-stickiness**, voir Configurer une ressource pour qu'elle préfère son nœud actuel .

```
[root@z1 ~]# pcs node unstandby z1.example.com
```

CHAPITRE 7. SYSTÈMES DE FICHIERS GFS2 DANS UN CLUSTER

Utilisez les procédures administratives suivantes pour configurer les systèmes de fichiers GFS2 dans un cluster de haute disponibilité Red Hat.

7.1. CONFIGURATION D'UN SYSTÈME DE FICHIERS GFS2 DANS UN CLUSTER

Vous pouvez configurer un cluster Pacemaker comprenant des systèmes de fichiers GFS2 à l'aide de la procédure suivante. Dans cet exemple, vous créez trois systèmes de fichiers GFS2 sur trois volumes logiques dans un cluster à deux nœuds.

Conditions préalables

- Installez et démarrez le logiciel de cluster sur les deux nœuds du cluster et créez un cluster de base à deux nœuds.
- Configurer la clôture pour le cluster.

Pour plus d'informations sur la création d'un cluster Pacemaker et la configuration de la clôture pour le cluster, voir Création d'un cluster Red Hat High-Availability avec Pacemaker .

Procédure

1. Sur les deux nœuds du cluster, activez le référentiel Resilient Storage correspondant à l'architecture de votre système. Par exemple, pour activer le référentiel Resilient Storage pour un système x86_64, vous pouvez entrer la commande **subscription-manager** suivante :

   ```
   # subscription-manager repos --enable=rhel-9-for-x86_64-resilientstorage-rpms
   ```

   Notez que le référentiel de stockage résilient est un surensemble du référentiel de haute disponibilité. Si vous activez le référentiel de stockage résilient, il n'est pas nécessaire d'activer également le référentiel de haute disponibilité.

2. Sur les deux nœuds du cluster, installez les paquets **lvm2-lockd**, **gfs2-utils**, et **dlm**. Pour prendre en charge ces paquets, vous devez être abonné au canal AppStream et au canal Resilient Storage.

   ```
   # dnf install lvm2-lockd gfs2-utils dlm
   ```

3. Sur les deux nœuds du cluster, définissez l'option de configuration **use_lvmlockd** dans le fichier **/etc/lvm/lvm.conf** sur **use_lvmlockd=1**.

   ```
   ...
   use_lvmlockd = 1
   ...
   ```

4. Réglez le paramètre global du stimulateur cardiaque **no-quorum-policy** sur **freeze**.

NOTE

Par défaut, la valeur de **no-quorum-policy** est fixée à **stop**, indiquant qu'une fois le quorum perdu, toutes les ressources sur la partition restante seront immédiatement arrêtées. En général, cette valeur par défaut est l'option la plus sûre et la plus optimale, mais contrairement à la plupart des ressources, GFS2 a besoin du quorum pour fonctionner. Lorsque le quorum est perdu, les applications utilisant les montages GFS2 et le montage GFS2 lui-même ne peuvent pas être arrêtés correctement. Toute tentative d'arrêt de ces ressources sans quorum échouera, ce qui aura pour conséquence de clôturer l'ensemble du cluster à chaque fois que le quorum est perdu.

Pour remédier à cette situation, définissez **no-quorum-policy** sur **freeze** lorsque GFS2 est utilisé. Cela signifie que lorsque le quorum est perdu, la partition restante ne fera rien jusqu'à ce que le quorum soit rétabli.

```
[root@z1 ~]# pcs property set no-quorum-policy=freeze
```

5. Configurer une ressource **dlm**. Il s'agit d'une dépendance nécessaire pour configurer un système de fichiers GFS2 dans un cluster. Cet exemple crée la ressource **dlm** dans le cadre d'un groupe de ressources nommé **locking**.

```
[root@z1 ~]# pcs resource create dlm --group locking ocf:pacemaker:controld op
monitor interval=30s on-fail=fence
```

6. Clonez le groupe de ressources **locking** afin que le groupe de ressources puisse être actif sur les deux nœuds du cluster.

```
[root@z1 ~]# pcs resource clone locking interleave=true
```

7. Créez une ressource **lvmlockd** dans le cadre du groupe de ressources **locking**.

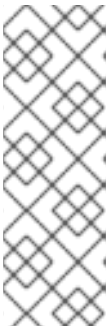
```
[root@z1 ~]# pcs resource create lvmlockd --group locking ocf:heartbeat:lvmlockd op
monitor interval=30s on-fail=fence
```

8. Vérifiez l'état du cluster pour vous assurer que le groupe de ressources **locking** a démarré sur les deux nœuds du cluster.

```
[root@z1 ~]# pcs status --full
Cluster name: my_cluster
[...]

Online: [ z1.example.com (1) z2.example.com (2) ]

Full list of resources:

 smoke-apc      (stonith:fence_apc):    Started z1.example.com
 Clone Set: locking-clone [locking]
     Resource Group: locking:0
         dlm    (ocf::pacemaker:controld):      Started z1.example.com
         lvmlockd       (ocf::heartbeat:lvmlockd):      Started z1.example.com
     Resource Group: locking:1
```

```
     dlm    (ocf::pacemaker:controld):      Started z2.example.com
     lvmlockd       (ocf::heartbeat:lvmlockd):      Started z2.example.com
 Started: [ z1.example.com z2.example.com ]
```

9. Sur un nœud de la grappe, créez deux groupes de volumes partagés. Un groupe de volumes contiendra deux systèmes de fichiers GFS2 et l'autre groupe de volumes contiendra un système de fichiers GFS2.



NOTE

Si votre groupe de volumes LVM contient un ou plusieurs volumes physiques résidant sur un stockage en bloc distant, tel qu'une cible iSCSI, Red Hat vous recommande de vous assurer que le service démarre avant le démarrage de Pacemaker. Pour plus d'informations sur la configuration de l'ordre de démarrage d'un volume physique distant utilisé par un cluster Pacemaker, reportez-vous à Configuration de l'ordre de démarrage pour les dépendances de ressources non gérées par Pacemaker.

La commande suivante crée le groupe de volumes partagés **shared_vg1** sur **/dev/vdb**.

```
[root@z1 ~]# vgcreate --shared shared_vg1 /dev/vdb
  Physical volume "/dev/vdb" successfully created.
  Volume group "shared_vg1" successfully created
  VG shared_vg1 starting dlm lockspace
  Starting locking.  Waiting until locks are ready...
```

La commande suivante crée le groupe de volumes partagés **shared_vg2** sur **/dev/vdc**.

```
[root@z1 ~]# vgcreate --shared shared_vg2 /dev/vdc
  Physical volume "/dev/vdc" successfully created.
  Volume group "shared_vg2" successfully created
  VG shared_vg2 starting dlm lockspace
  Starting locking.  Waiting until locks are ready...
```

10. Sur le deuxième nœud du cluster :

    a. Ajoutez les périphériques partagés au fichier des périphériques LVM.

    ```
    [root@z2 ~]# lvmdevices --adddev /dev/vdb
    [root@z2 ~]# lvmdevices --adddev /dev/vdc
    ```

    b. Lancez le gestionnaire de verrous pour chacun des groupes de volumes partagés.

    ```
    [root@z2 ~]# vgchange --lockstart shared_vg1
      VG shared_vg1 starting dlm lockspace
      Starting locking.  Waiting until locks are ready...
    [root@z2 ~]# vgchange --lockstart shared_vg2
      VG shared_vg2 starting dlm lockspace
      Starting locking.  Waiting until locks are ready...
    ```

11. Sur un nœud de la grappe, créez les volumes logiques partagés et formatez les volumes avec un système de fichiers GFS2. Un journal est nécessaire pour chaque nœud qui monte le système de fichiers. Veillez à créer suffisamment de journaux pour chacun des nœuds de votre grappe.

Le format du nom de la table de verrouillage est *ClusterName:FSName*, où *ClusterName* est le nom de la grappe pour laquelle le système de fichiers GFS2 est créé et *FSName* est le nom du système de fichiers, qui doit être unique pour tous les systèmes de fichiers **lock_dlm** de la grappe.

```
[root@z1 ~]# lvcreate --activate sy -L5G -n shared_lv1 shared_vg1
  Logical volume "shared_lv1" created.
[root@z1 ~]# lvcreate --activate sy -L5G -n shared_lv2 shared_vg1
  Logical volume "shared_lv2" created.
[root@z1 ~]# lvcreate --activate sy -L5G -n shared_lv1 shared_vg2
  Logical volume "shared_lv1" created.

[root@z1 ~]# mkfs.gfs2 -j2 -p lock_dlm -t my_cluster:gfs2-demo1
/dev/shared_vg1/shared_lv1
[root@z1 ~]# mkfs.gfs2 -j2 -p lock_dlm -t my_cluster:gfs2-demo2
/dev/shared_vg1/shared_lv2
[root@z1 ~]# mkfs.gfs2 -j2 -p lock_dlm -t my_cluster:gfs2-demo3
/dev/shared_vg2/shared_lv1
```

12. Créez une ressource **LVM-activate** pour chaque volume logique afin d'activer automatiquement ce volume logique sur tous les nœuds.

    a. Créez une ressource **LVM-activate** nommée **sharedlv1** pour le volume logique **shared_lv1** dans le groupe de volumes **shared_vg1**. Cette commande crée également le groupe de ressources **shared_vg1** qui inclut la ressource. Dans cet exemple, le groupe de ressources porte le même nom que le groupe de volumes partagés qui comprend le volume logique.

    ```
    [root@z1 ~]# pcs resource create sharedlv1 --group shared_vg1 ocf:heartbeat:LVM-
    activate lvname=shared_lv1 vgname=shared_vg1 activation_mode=shared
    vg_access_mode=lvmlockd
    ```

    b. Créez une ressource **LVM-activate** nommée **sharedlv2** pour le volume logique **shared_lv2** dans le groupe de volumes **shared_vg1**. Cette ressource fera également partie du groupe de ressources **shared_vg1**.

    ```
    [root@z1 ~]# pcs resource create sharedlv2 --group shared_vg1 ocf:heartbeat:LVM-
    activate lvname=shared_lv2 vgname=shared_vg1 activation_mode=shared
    vg_access_mode=lvmlockd
    ```

    c. Créez une ressource **LVM-activate** nommée **sharedlv3** pour le volume logique **shared_lv1** dans le groupe de volumes **shared_vg2**. Cette commande crée également le groupe de ressources **shared_vg2** qui inclut la ressource.

    ```
    [root@z1 ~]# pcs resource create sharedlv3 --group shared_vg2 ocf:heartbeat:LVM-
    activate lvname=shared_lv1 vgname=shared_vg2 activation_mode=shared
    vg_access_mode=lvmlockd
    ```

13. Clonez les deux nouveaux groupes de ressources.

```
[root@z1 ~]# pcs resource clone shared_vg1 interleave=true
[root@z1 ~]# pcs resource clone shared_vg2 interleave=true
```

14. Configurez les contraintes d'ordre pour vous assurer que le groupe de ressources **locking** qui inclut les ressources **dlm** et **lvmlockd** démarre en premier.

```
[root@z1 ~]# pcs constraint order start locking-clone then shared_vg1-clone
Adding locking-clone shared_vg1-clone (kind: Mandatory) (Options: first-action=start then-action=start)
[root@z1 ~]# pcs constraint order start locking-clone then shared_vg2-clone
Adding locking-clone shared_vg2-clone (kind: Mandatory) (Options: first-action=start then-action=start)
```

15. Configurez les contraintes de colocation pour vous assurer que les groupes de ressources **vg1** et **vg2** démarrent sur le même nœud que le groupe de ressources **locking**.

```
[root@z1 ~]# pcs constraint colocation add shared_vg1-clone with locking-clone
[root@z1 ~]# pcs constraint colocation add shared_vg2-clone with locking-clone
```

16. Sur les deux nœuds du cluster, vérifiez que les volumes logiques sont actifs. Il peut y avoir un délai de quelques secondes.

```
[root@z1 ~]# lvs
  LV         VG         Attr       LSize
  shared_lv1 shared_vg1 -wi-a----- 5.00g
  shared_lv2 shared_vg1 -wi-a----- 5.00g
  shared_lv1 shared_vg2 -wi-a----- 5.00g

[root@z2 ~]# lvs
  LV         VG         Attr       LSize
  shared_lv1 shared_vg1 -wi-a----- 5.00g
  shared_lv2 shared_vg1 -wi-a----- 5.00g
  shared_lv1 shared_vg2 -wi-a----- 5.00g
```

17. Créez une ressource de système de fichiers pour monter automatiquement chaque système de fichiers GFS2 sur tous les nœuds.
    Vous ne devez pas ajouter le système de fichiers au fichier **/etc/fstab** car il sera géré comme une ressource de cluster Pacemaker. Les options de montage peuvent être spécifiées dans le cadre de la configuration de la ressource à l'aide de la commande **options=*options***. Exécutez la commande **pcs resource describe Filesystem** pour afficher les options de configuration complètes.

    Les commandes suivantes créent les ressources du système de fichiers. Elles ajoutent chaque ressource au groupe de ressources qui comprend la ressource de volume logique pour ce système de fichiers.

```
[root@z1 ~]# pcs resource create sharedfs1 --group shared_vg1 ocf:heartbeat:Filesystem device="/dev/shared_vg1/shared_lv1" directory="/mnt/gfs1" fstype="gfs2" options=noatime op monitor interval=10s on-fail=fence
[root@z1 ~]# pcs resource create sharedfs2 --group shared_vg1 ocf:heartbeat:Filesystem device="/dev/shared_vg1/shared_lv2" directory="/mnt/gfs2" fstype="gfs2" options=noatime op monitor interval=10s on-fail=fence
[root@z1 ~]# pcs resource create sharedfs3 --group shared_vg2 ocf:heartbeat:Filesystem device="/dev/shared_vg2/shared_lv1" directory="/mnt/gfs3" fstype="gfs2" options=noatime op monitor interval=10s on-fail=fence
```

Verification steps

1. Vérifiez que les systèmes de fichiers GFS2 sont montés sur les deux nœuds du cluster.

```
[root@z1 ~]# mount | grep gfs2
/dev/mapper/shared_vg1-shared_lv1 on /mnt/gfs1 type gfs2 (rw,noatime,seclabel)
/dev/mapper/shared_vg1-shared_lv2 on /mnt/gfs2 type gfs2 (rw,noatime,seclabel)
/dev/mapper/shared_vg2-shared_lv1 on /mnt/gfs3 type gfs2 (rw,noatime,seclabel)

[root@z2 ~]# mount | grep gfs2
/dev/mapper/shared_vg1-shared_lv1 on /mnt/gfs1 type gfs2 (rw,noatime,seclabel)
/dev/mapper/shared_vg1-shared_lv2 on /mnt/gfs2 type gfs2 (rw,noatime,seclabel)
/dev/mapper/shared_vg2-shared_lv1 on /mnt/gfs3 type gfs2 (rw,noatime,seclabel)
```

2. Vérifier l'état de la grappe.

```
[root@z1 ~]# pcs status --full
Cluster name: my_cluster
[...]

Full list of resources:

 smoke-apc      (stonith:fence_apc):    Started z1.example.com
 Clone Set: locking-clone [locking]
     Resource Group: locking:0
         dlm    (ocf::pacemaker:controld):      Started z2.example.com
         lvmlockd       (ocf::heartbeat:lvmlockd):      Started z2.example.com
     Resource Group: locking:1
         dlm    (ocf::pacemaker:controld):      Started z1.example.com
         lvmlockd       (ocf::heartbeat:lvmlockd):      Started z1.example.com
     Started: [ z1.example.com z2.example.com ]
 Clone Set: shared_vg1-clone [shared_vg1]
     Resource Group: shared_vg1:0
         sharedlv1      (ocf::heartbeat:LVM-activate):  Started z2.example.com
         sharedlv2      (ocf::heartbeat:LVM-activate):  Started z2.example.com
         sharedfs1      (ocf::heartbeat:Filesystem):    Started z2.example.com
         sharedfs2      (ocf::heartbeat:Filesystem):    Started z2.example.com
     Resource Group: shared_vg1:1
         sharedlv1      (ocf::heartbeat:LVM-activate):  Started z1.example.com
         sharedlv2      (ocf::heartbeat:LVM-activate):  Started z1.example.com
         sharedfs1      (ocf::heartbeat:Filesystem):    Started z1.example.com
         sharedfs2      (ocf::heartbeat:Filesystem):    Started z1.example.com
     Started: [ z1.example.com z2.example.com ]
 Clone Set: shared_vg2-clone [shared_vg2]
     Resource Group: shared_vg2:0
         sharedlv3      (ocf::heartbeat:LVM-activate):  Started z2.example.com
         sharedfs3      (ocf::heartbeat:Filesystem):    Started z2.example.com
     Resource Group: shared_vg2:1
         sharedlv3      (ocf::heartbeat:LVM-activate):  Started z1.example.com
         sharedfs3      (ocf::heartbeat:Filesystem):    Started z1.example.com
     Started: [ z1.example.com z2.example.com ]

...
```

Ressources supplémentaires

- Configuration des systèmes de fichiers GFS2
- Configurer un cluster Red Hat High Availability sur Microsoft Azure

- Configuration d'un cluster Red Hat High Availability sur AWS
- Configuration de Red Hat High Availability Cluster sur Google Cloud Platform
- Configurer le stockage en bloc partagé pour un cluster Red Hat High Availability sur Alibaba Cloud

7.2. CONFIGURATION D'UN SYSTÈME DE FICHIERS GFS2 CRYPTÉ DANS UN CLUSTER

Vous pouvez créer un cluster Pacemaker qui inclut un système de fichiers GFS2 crypté LUKS à l'aide de la procédure suivante. Dans cet exemple, vous créez un système de fichiers GFS2 sur un volume logique et vous chiffrez le système de fichiers. Les systèmes de fichiers GFS2 chiffrés sont pris en charge par l'agent de ressources **crypt**, qui prend en charge le chiffrement LUKS.

Cette procédure comporte trois parties :

- Configuration d'un volume logique partagé dans un cluster Pacemaker
- Chiffrement du volume logique et création d'une ressource **crypt**
- Formatage du volume logique crypté avec un système de fichiers GFS2 et création d'une ressource de système de fichiers pour le cluster

7.2.1. Configurer un volume logique partagé dans un cluster Pacemaker

Conditions préalables

- Installez et démarrez le logiciel de cluster sur deux nœuds de cluster et créez un cluster de base à deux nœuds.
- Configurer la clôture pour le cluster.

Pour plus d'informations sur la création d'un cluster Pacemaker et la configuration de la clôture pour le cluster, voir Création d'un cluster Red Hat High-Availability avec Pacemaker .

Procédure

1. Sur les deux nœuds du cluster, activez le référentiel Resilient Storage correspondant à l'architecture de votre système. Par exemple, pour activer le référentiel Resilient Storage pour un système x86_64, vous pouvez entrer la commande **subscription-manager** suivante :

   ```
   # subscription-manager repos --enable=rhel-9-for-x86_64-resilientstorage-rpms
   ```

   Notez que le référentiel de stockage résilient est un surensemble du référentiel de haute disponibilité. Si vous activez le référentiel de stockage résilient, il n'est pas nécessaire d'activer également le référentiel de haute disponibilité.

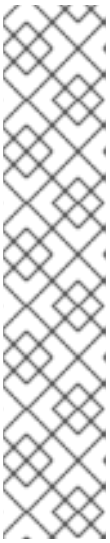
2. Sur les deux nœuds du cluster, installez les paquets **lvm2-lockd**, **gfs2-utils**, et **dlm**. Pour prendre en charge ces paquets, vous devez être abonné au canal AppStream et au canal Resilient Storage.

   ```
   # dnf install lvm2-lockd gfs2-utils dlm
   ```

3. Sur les deux nœuds du cluster, définissez l'option de configuration **use_lvmlockd** dans le fichier **/etc/lvm/lvm.conf** sur **use_lvmlockd=1**.

```
...
use_lvmlockd = 1
...
```

4. Réglez le paramètre global du stimulateur cardiaque **no-quorum-policy** sur **freeze**.



NOTE

Par défaut, la valeur de **no-quorum-policy** est fixée à **stop**, indiquant que lorsque le quorum est perdu, toutes les ressources sur la partition restante seront immédiatement arrêtées. En général, cette valeur par défaut est l'option la plus sûre et la plus optimale, mais contrairement à la plupart des ressources, GFS2 a besoin du quorum pour fonctionner. Lorsque le quorum est perdu, les applications utilisant les montages GFS2 et le montage GFS2 lui-même ne peuvent pas être arrêtés correctement. Toute tentative d'arrêt de ces ressources sans quorum échouera, ce qui aura pour conséquence de clôturer l'ensemble du cluster à chaque fois que le quorum est perdu.

Pour remédier à cette situation, définissez **no-quorum-policy** sur **freeze** lorsque GFS2 est utilisé. Cela signifie que lorsque le quorum est perdu, la partition restante ne fera rien jusqu'à ce que le quorum soit rétabli.

```
[root@z1 ~]# pcs property set no-quorum-policy=freeze
```

5. Configurer une ressource **dlm**. Il s'agit d'une dépendance nécessaire pour configurer un système de fichiers GFS2 dans un cluster. Cet exemple crée la ressource **dlm** dans le cadre d'un groupe de ressources nommé **locking**.

```
[root@z1 ~]# pcs resource create dlm --group locking ocf:pacemaker:controld op
monitor interval=30s on-fail=fence
```

6. Clonez le groupe de ressources **locking** afin que le groupe de ressources puisse être actif sur les deux nœuds du cluster.

```
[root@z1 ~]# pcs resource clone locking interleave=true
```

7. Créez une ressource **lvmlockd** dans le cadre du groupe **locking**.

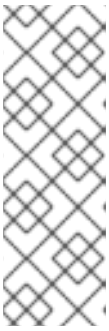
```
[root@z1 ~]# pcs resource create lvmlockd --group locking ocf:heartbeat:lvmlockd op
monitor interval=30s on-fail=fence
```

8. Vérifiez l'état du cluster pour vous assurer que le groupe de ressources **locking** a démarré sur les deux nœuds du cluster.

```
[root@z1 ~]# pcs status --full
Cluster name: my_cluster
[...]

Online: [ z1.example.com (1) z2.example.com (2) ]
```

```
Full list of resources:

 smoke-apc      (stonith:fence_apc):    Started z1.example.com
 Clone Set: locking-clone [locking]
     Resource Group: locking:0
         dlm    (ocf::pacemaker:controld):      Started z1.example.com
         lvmlockd       (ocf::heartbeat:lvmlockd):      Started z1.example.com
     Resource Group: locking:1
         dlm    (ocf::pacemaker:controld):      Started z2.example.com
         lvmlockd       (ocf::heartbeat:lvmlockd):      Started z2.example.com
     Started: [ z1.example.com z2.example.com ]
```

9. Sur un nœud de la grappe, créez un groupe de volumes partagés.



NOTE

Si votre groupe de volumes LVM contient un ou plusieurs volumes physiques résidant sur un stockage en bloc distant, tel qu'une cible iSCSI, Red Hat vous recommande de vous assurer que le service démarre avant le démarrage de Pacemaker. Pour obtenir des informations sur la configuration de l'ordre de démarrage d'un volume physique distant utilisé par un cluster Pacemaker, reportez-vous à Configuration de l'ordre de démarrage pour les dépendances de ressources non gérées par Pacemaker.

La commande suivante crée le groupe de volumes partagés **shared_vg1** sur **/dev/sda1**.

```
[root@z1 ~]# vgcreate --shared shared_vg1 /dev/sda1
  Physical volume "/dev/sda1" successfully created.
  Volume group "shared_vg1" successfully created
  VG shared_vg1 starting dlm lockspace
  Starting locking.  Waiting until locks are ready...
```

10. Sur le deuxième nœud du cluster :

    a. Ajoutez le périphérique partagé au fichier des périphériques LVM.

    ```
    [root@z2 ~]# lvmdevices --adddev /dev/sda1
    ```

    b. Démarrer le gestionnaire de verrouillage pour le groupe de volumes partagés.

    ```
    [root@z2 ~]# vgchange --lockstart shared_vg1
      VG shared_vg1 starting dlm lockspace
      Starting locking.  Waiting until locks are ready...
    ```

11. Sur un nœud de la grappe, créez le volume logique partagé.

```
[root@z1 ~]# lvcreate --activate sy -L5G -n shared_lv1 shared_vg1
  Logical volume "shared_lv1" created.
```

12. Créez une ressource **LVM-activate** pour le volume logique afin d'activer automatiquement le volume logique sur tous les nœuds.
La commande suivante crée une ressource **LVM-activate** nommée **sharedlv1** pour le volume logique **shared_lv1** dans le groupe de volumes **shared_vg1**. Cette commande crée également

le groupe de ressources **shared_vg1** qui inclut la ressource. Dans cet exemple, le groupe de ressources porte le même nom que le groupe de volumes partagés qui comprend le volume logique.

```
[root@z1 ~]# pcs resource create sharedlv1 --group shared_vg1 ocf:heartbeat:LVM-activate lvname=shared_lv1 vgname=shared_vg1 activation_mode=shared vg_access_mode=lvmlockd
```

13. Cloner le nouveau groupe de ressources.

```
[root@z1 ~]# pcs resource clone shared_vg1 interleave=true
```

14. Configurez une contrainte d'ordre pour garantir que le groupe de ressources **locking** qui inclut les ressources **dlm** et **lvmlockd** démarre en premier.

```
[root@z1 ~]# pcs constraint order start locking-clone then shared_vg1-clone
Adding locking-clone shared_vg1-clone (kind: Mandatory) (Options: first-action=start then-action=start)
```

15. Configurer les contraintes de colocation pour s'assurer que les groupes de ressources **vg1** et **vg2** démarrent sur le même nœud que le groupe de ressources **locking**.

```
[root@z1 ~]# pcs constraint colocation add shared_vg1-clone with locking-clone
```

Verification steps

Sur les deux nœuds du cluster, vérifiez que le volume logique est actif. Il peut y avoir un délai de quelques secondes.

```
[root@z1 ~]# lvs
  LV        VG         Attr      LSize
  shared_lv1 shared_vg1  -wi-a----- 5.00g

[root@z2 ~]# lvs
  LV        VG         Attr      LSize
  shared_lv1 shared_vg1  -wi-a----- 5.00g
```

7.2.2. Cryptage du volume logique et création d'une ressource cryptée

Conditions préalables

- Vous avez configuré un volume logique partagé dans un cluster Pacemaker.

Procédure

1. Sur un nœud du cluster, créez un nouveau fichier qui contiendra la clé cryptographique et définissez les autorisations sur le fichier de sorte qu'il ne soit lisible que par root.

```
[root@z1 ~]# touch /etc/crypt_keyfile
[root@z1 ~]# chmod 600 /etc/crypt_keyfile
```

2. Créer la clé cryptographique.

```
[root@z1 ~]# dd if=/dev/urandom bs=4K count=1 of=/etc/crypt_keyfile
1+0 records in
1+0 records out
4096 bytes (4.1 kB, 4.0 KiB) copied, 0.000306202 s, 13.4 MB/s
[root@z1 ~]# scp /etc/crypt_keyfile root@z2.example.com:/etc/
```

3. Distribuez le fichier clé cryptographique aux autres nœuds de la grappe, en utilisant le paramètre **-p** pour préserver les autorisations que vous avez définies.

```
[root@z1 ~]# scp -p /etc/crypt_keyfile root@z2.example.com:/etc/
```

4. Créez le périphérique crypté sur le volume LVM où vous configurerez le système de fichiers GFS2 crypté.

```
[root@z1 ~]# cryptsetup luksFormat /dev/shared_vg1/shared_lv1 --type luks2 --key-
file=/etc/crypt_keyfile
WARNING!
========
This will overwrite data on /dev/shared_vg1/shared_lv1 irrevocably.

Are you sure? (Type 'yes' in capital letters): YES
```

5. Créez la ressource cryptographique dans le cadre du groupe de volumes **shared_vg1**.

```
[root@z1 ~]# pcs resource create crypt --group shared_vg1 ocf:heartbeat:crypt
crypt_dev="luks_lv1" crypt_type=luks2 key_file=/etc/crypt_keyfile
encrypted_dev="/dev/shared_vg1/shared_lv1"
```

Verification steps

Assurez-vous que la ressource crypt a créé la clé de cryptage, qui dans cet exemple est **/dev/mapper/luks_lv1**.

```
[root@z1 ~]# ls -l /dev/mapper/
...
lrwxrwxrwx 1 root root 7 Mar 4 09:52 luks_lv1 -> ../dm-3
...
```

7.2.3. Formatez le volume logique crypté avec un système de fichiers GFS2 et créez une ressource de système de fichiers pour le cluster

Conditions préalables

- Vous avez chiffré le volume logique et créé une ressource cryptée.

Procédure

1. Sur un nœud du cluster, formatez le volume avec un système de fichiers GFS2. Un journal est nécessaire pour chaque nœud qui monte le système de fichiers. Veillez à créer suffisamment de journaux pour chacun des nœuds de votre grappe. Le format du nom de la table de verrouillage est *ClusterName:FSName*, où *ClusterName* est le nom de la grappe pour laquelle le système de fichiers GFS2 est créé et *FSName* est le nom du système de fichiers, qui doit être unique pour tous les systèmes de fichiers **lock_dlm** de la grappe.

```
[root@z1 ~]# mkfs.gfs2 -j3 -p lock_dlm -t my_cluster:gfs2-demo1 /dev/mapper/luks_lv1
/dev/mapper/luks_lv1 is a symbolic link to /dev/dm-3
This will destroy any data on /dev/dm-3
Are you sure you want to proceed? [y/n] y
Discarding device contents (may take a while on large devices): Done
Adding journals: Done
Building resource groups: Done
Creating quota file: Done
Writing superblock and syncing: Done
Device:                    /dev/mapper/luks_lv1
Block size:                4096
Device size:               4.98 GB (1306624 blocks)
Filesystem size:           4.98 GB (1306622 blocks)
Journals:                  3
Journal size:              16MB
Resource groups:           23
Locking protocol:          "lock_dlm"
Lock table:                "my_cluster:gfs2-demo1"
UUID:                      de263f7b-0f12-4d02-bbb2-56642fade293
```

2. Créez une ressource de système de fichiers pour monter automatiquement le système de fichiers GFS2 sur tous les nœuds.
   N'ajoutez pas le système de fichiers au fichier **/etc/fstab** car il sera géré comme une ressource de cluster Pacemaker. Les options de montage peuvent être spécifiées dans le cadre de la configuration de la ressource à l'aide de la commande **options=*options***. Exécutez la commande **pcs resource describe Filesystem** pour obtenir toutes les options de configuration.

   La commande suivante crée la ressource du système de fichiers. Cette commande ajoute la ressource au groupe de ressources qui comprend la ressource de volume logique pour ce système de fichiers.

```
[root@z1 ~]# pcs resource create sharedfs1 --group shared_vg1
ocf:heartbeat:Filesystem device="/dev/mapper/luks_lv1" directory="/mnt/gfs1"
fstype="gfs2" options=noatime op monitor interval=10s on-fail=fence
```

Verification steps

1. Vérifiez que le système de fichiers GFS2 est monté sur les deux nœuds du cluster.

```
[root@z1 ~]# mount | grep gfs2
/dev/mapper/luks_lv1 on /mnt/gfs1 type gfs2 (rw,noatime,seclabel)

[root@z2 ~]# mount | grep gfs2
/dev/mapper/luks_lv1 on /mnt/gfs1 type gfs2 (rw,noatime,seclabel)
```

2. Vérifier l'état de la grappe.

```
[root@z1 ~]# pcs status --full
Cluster name: my_cluster
[...]

Full list of resources:

  smoke-apc      (stonith:fence_apc):    Started z1.example.com
```

```
  Clone Set: locking-clone [locking]
      Resource Group: locking:0
          dlm    (ocf::pacemaker:controld):      Started z2.example.com
          lvmlockd       (ocf::heartbeat:lvmlockd):      Started z2.example.com
      Resource Group: locking:1
          dlm    (ocf::pacemaker:controld):      Started z1.example.com
          lvmlockd       (ocf::heartbeat:lvmlockd):      Started z1.example.com
    Started: [ z1.example.com z2.example.com ]
  Clone Set: shared_vg1-clone [shared_vg1]
    Resource Group: shared_vg1:0
          sharedlv1      (ocf::heartbeat:LVM-activate):  Started z2.example.com
          crypt       (ocf::heartbeat:crypt) Started z2.example.com
          sharedfs1      (ocf::heartbeat:Filesystem):    Started z2.example.com
   Resource Group: shared_vg1:1
          sharedlv1      (ocf::heartbeat:LVM-activate):  Started z1.example.com
          crypt      (ocf::heartbeat:crypt)  Started z1.example.com
          sharedfs1      (ocf::heartbeat:Filesystem):    Started z1.example.com
        Started:  [z1.example.com z2.example.com ]
...
```

Ressources supplémentaires

- Configuration des systèmes de fichiers GFS2

CHAPITRE 8. CONFIGURATION D'UN SERVEUR SAMBA ACTIF/ACTIF DANS UN CLUSTER RED HAT HIGH AVAILABILITY

Le module complémentaire de haute disponibilité de Red Hat prend en charge la configuration de Samba dans une configuration de grappe active/active. Dans l'exemple suivant, vous configurez un serveur Samba actif/actif sur un cluster RHEL à deux nœuds.

Pour plus d'informations sur les politiques d'assistance pour Samba, voir Politiques d'assistance pour RHEL High Availability - Politiques générales de ctdb et Politiques d'assistance pour RHEL Resilient Storage - Exportation de contenus gfs2 via d'autres protocoles sur le Portail client de Red Hat.

Pour configurer Samba dans un cluster actif/actif :

1. Configurer un système de fichiers GFS2 et ses ressources cluster associées.
2. Configurez Samba sur les nœuds du cluster.
3. Configurez les ressources du cluster Samba.
4. Testez le serveur Samba que vous avez configuré.

8.1. CONFIGURATION D'UN SYSTÈME DE FICHIERS GFS2 POUR UN SERVICE SAMBA DANS UN CLUSTER À HAUTE DISPONIBILITÉ

Avant de configurer un service Samba actif/actif dans un cluster Pacemaker, configurez un système de fichiers GFS2 pour le cluster.

Conditions préalables

- Un cluster Red Hat High Availability à deux nœuds avec des clôtures configurées pour chaque nœud
- Stockage partagé disponible pour chaque nœud du cluster
- Un abonnement au canal AppStream et au canal Resilient Storage pour chaque nœud du cluster

Pour plus d'informations sur la création d'un cluster Pacemaker et la configuration de la clôture pour le cluster, voir Création d'un cluster Red Hat High-Availability avec Pacemaker .

Procédure

1. Sur les deux nœuds de la grappe, effectuez les étapes de configuration initiale suivantes.
   a. Activez le référentiel Resilient Storage correspondant à l'architecture de votre système. Par exemple, pour activer le référentiel Resilient Storage pour un système x86_64, entrez la commande **subscription-manager** suivante :

      ```
      # subscription-manager repos --enable=rhel-9-for-x86_64-resilientstorage-rpms
      ```

      Le référentiel de stockage résilient est un surensemble du référentiel de haute disponibilité. Si vous activez le référentiel de stockage résilient, il n'est pas nécessaire d'activer également le référentiel de haute disponibilité.

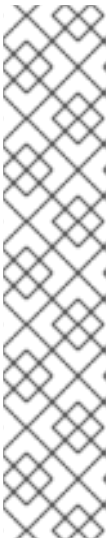
b. Installez les paquets **lvm2-lockd**, **gfs2-utils**, et **dlm**.

```
# yum install lvm2-lockd gfs2-utils dlm
```

c. Définissez l'option de configuration **use_lvmlockd** dans le fichier **/etc/lvm/lvm.conf** à **use_lvmlockd=1**.

```
...

use_lvmlockd = 1

...
```

2. Sur un nœud de la grappe, définissez le paramètre global de Pacemaker **no-quorum-policy** sur **freeze**.



NOTE

Par défaut, la valeur de **no-quorum-policy** est fixée à **stop**, indiquant qu'une fois le quorum perdu, toutes les ressources sur la partition restante seront immédiatement arrêtées. En général, cette valeur par défaut est l'option la plus sûre et la plus optimale, mais contrairement à la plupart des ressources, GFS2 a besoin du quorum pour fonctionner. Lorsque le quorum est perdu, les applications utilisant les montages GFS2 et le montage GFS2 lui-même ne peuvent pas être arrêtés correctement. Toute tentative d'arrêt de ces ressources sans quorum échouera, ce qui aura pour conséquence de clôturer l'ensemble du cluster à chaque fois que le quorum est perdu.

Pour remédier à cette situation, définissez **no-quorum-policy** sur **freeze** lorsque GFS2 est utilisé. Cela signifie que lorsque le quorum est perdu, la partition restante ne fera rien jusqu'à ce que le quorum soit rétabli.

```
[root@z1 ~]# pcs property set no-quorum-policy=freeze
```

3. Configurer une ressource **dlm**. Il s'agit d'une dépendance nécessaire pour configurer un système de fichiers GFS2 dans un cluster. Cet exemple crée la ressource **dlm** dans le cadre d'un groupe de ressources nommé **locking**. Si vous n'avez pas préalablement configuré la clôture pour le cluster, cette étape échoue et la commande **pcs status** affiche un message d'échec de la ressource.

```
[root@z1 ~]# pcs resource create dlm --group locking ocf:pacemaker:controld op
monitor interval=30s on-fail=fence
```

4. Clonez le groupe de ressources **locking** afin que le groupe de ressources puisse être actif sur les deux nœuds du cluster.

```
[root@z1 ~]# pcs resource clone locking interleave=true
```

5. Créez une ressource **lvmlockd** dans le cadre du groupe de ressources **locking**.

```
[root@z1 ~]# pcs resource create lvmlockd --group locking ocf:heartbeat:lvmlockd op
monitor interval=30s on-fail=fence
```

6. Créez un volume physique et un groupe de volumes partagés sur le périphérique partagé **/dev/vdb**. Cet exemple crée le groupe de volumes partagés **csmb_vg**.

```
[root@z1 ~]# pvcreate /dev/vdb
[root@z1 ~]# vgcreate -Ay --shared csmb_vg /dev/vdb
Volume group "csmb_vg" successfully created
VG csmb_vg starting dlm lockspace
Starting locking.  Waiting until locks are ready
```

7. Sur le deuxième nœud du cluster :

   a. Ajoutez le périphérique partagé au fichier des périphériques LVM.

   ```
   [root@z2 ~]# lvmdevices --adddev /dev/vdb
   ```

   b. Démarrer le gestionnaire de verrouillage pour le groupe de volumes partagés.

   ```
   [root@z2 ~]# vgchange --lockstart csmb_vg
     VG csmb_vg starting dlm lockspace
     Starting locking.  Waiting until locks are ready...
   ```

8. Sur un nœud du cluster, créez un volume logique et formatez le volume avec un système de fichiers GFS2 qui sera utilisé exclusivement par CTDB pour le verrouillage interne. Un seul système de fichiers de ce type est nécessaire dans un cluster, même si votre déploiement exporte plusieurs partages.
   Lorsque vous spécifiez le nom de la table de verrouillage avec l'option **-t** de la commande **mkfs.gfs2**, assurez-vous que le premier composant de *clustername:filesystemname* que vous spécifiez correspond au nom de votre cluster. Dans cet exemple, le nom du cluster est **my_cluster**.

```
[root@z1 ~]# lvcreate -L1G -n ctdb_lv csmb_vg
[root@z1 ~]# mkfs.gfs2 -j3 -p lock_dlm -t my_cluster:ctdb /dev/csmb_vg/ctdb_lv
```

9. Créez un volume logique pour chaque système de fichiers GFS2 qui sera partagé par Samba et formatez le volume avec le système de fichiers GFS2. Cet exemple crée un seul système de fichiers GFS2 et un seul partage Samba, mais vous pouvez créer plusieurs systèmes de fichiers et partages.

```
[root@z1 ~]# lvcreate -L50G -n csmb_lv1 csmb_vg
[root@z1 ~]# mkfs.gfs2 -j3 -p lock_dlm -t my_cluster:csmb1 /dev/csmb_vg/csmb_lv1
```

10. Configurez les ressources **LVM_Activate** pour vous assurer que les volumes partagés requis sont activés. Cet exemple crée les ressources **LVM_Activate** dans le cadre d'un groupe de ressources **shared_vg**, puis clone ce groupe de ressources afin qu'il s'exécute sur tous les nœuds du cluster.
    Créez les ressources comme étant désactivées afin qu'elles ne démarrent pas automatiquement avant que vous n'ayez configuré les contraintes d'ordre nécessaires.

```
[root@z1 ~]# pcs resource create --disabled --group shared_vg ctdb_lv
ocf:heartbeat:LVM-activate lvname=ctdb_lv vgname=csmb_vg
activation_mode=shared vg_access_mode=lvmlockd
[root@z1 ~]# pcs resource create --disabled --group shared_vg csmb_lv1
```

```
ocf:heartbeat:LVM-activate lvname=csmb_lv1 vgname=csmb_vg
activation_mode=shared vg_access_mode=lvmlockd
[root@z1 ~]# pcs resource clone shared_vg interleave=true
```

11. Configurez une contrainte d'ordre pour démarrer tous les membres du groupe de ressources **locking** avant les membres du groupe de ressources **shared_vg**.

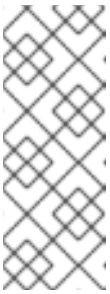
```
[root@z1 ~]# pcs constraint order start locking-clone then shared_vg-clone
Adding locking-clone shared_vg-clone (kind: Mandatory) (Options: first-action=start then-action=start)
```

12. Activer les ressources **LVM-activate**.

```
[root@z1 ~]# pcs resource enable ctdb_lv csmb_lv1
```

13. Sur un nœud du cluster, effectuez les étapes suivantes pour créer les ressources **Filesystem** dont vous avez besoin.

    a. Créez les ressources **Filesystem** en tant que ressources clonées, en utilisant les systèmes de fichiers GFS2 que vous avez précédemment configurés sur vos volumes LVM. Ceci configure Pacemaker pour monter et gérer les systèmes de fichiers.



NOTE

Vous ne devez pas ajouter le système de fichiers au fichier **/etc/fstab** car il sera géré comme une ressource de cluster Pacemaker. Vous pouvez spécifier des options de montage dans le cadre de la configuration de la ressource à l'aide de la commande **options=*options***. Exécutez la commande **pcs resource describe Filesystem** pour afficher les options de configuration complètes.

```
[root@z1 ~]# pcs resource create ctdb_fs Filesystem
device="/dev/csmb_vg/ctdb_lv" directory="/mnt/ctdb" fstype="gfs2" op monitor
interval=10s on-fail=fence clone interleave=true
[root@z1 ~]# pcs resource create csmb_fs1 Filesystem
device="/dev/csmb_vg/csmb_lv1" directory="/srv/samba/share1" fstype="gfs2" op
monitor interval=10s on-fail=fence clone interleave=true
```

    b. Configurez les contraintes d'ordre pour vous assurer que Pacemaker monte les systèmes de fichiers après le démarrage du groupe de volumes partagés **shared_vg**.

```
[root@z1 ~]# pcs constraint order start shared_vg-clone then ctdb_fs-clone
Adding shared_vg-clone ctdb_fs-clone (kind: Mandatory) (Options: first-action=start then-action=start)
[root@z1 ~]# pcs constraint order start shared_vg-clone then csmb_fs1-clone
Adding shared_vg-clone csmb_fs1-clone (kind: Mandatory) (Options: first-action=start then-action=start)
```

8.2. CONFIGURER SAMBA DANS UN CLUSTER À HAUTE DISPONIBILITÉ

Pour configurer un service Samba dans un cluster Pacemaker, configurez le service sur tous les nœuds du cluster.

Conditions préalables

- Un cluster Red Hat High Availability à deux nœuds configuré avec un système de fichiers GFS2, comme décrit dans Configuration d'un système de fichiers GFS2 pour un service Samba dans un cluster de haute disponibilité.
- Un répertoire public créé sur votre système de fichiers GFS2 à utiliser pour le partage Samba. Dans cet exemple, le répertoire est **/srv/samba/share1**.
- Adresses IP virtuelles publiques pouvant être utilisées pour accéder au partage Samba exporté par ce cluster.

Procédure

1. Sur les deux nœuds du cluster, configurez le service Samba et mettez en place une définition de partage :
   a. Installez les paquets Samba et CTDB.

   ```
   # dnf -y install samba ctdb cifs-utils samba-winbind
   ```

   b. Assurez-vous que les services **ctdb**, **smb**, **nmb** et **winbind** ne sont pas en cours d'exécution et ne démarrent pas au démarrage.

   ```
   # systemctl disable --now ctdb smb nmb winbind
   ```

   c. Dans le fichier **/etc/samba/smb.conf**, configurez le service Samba et mettez en place la définition de partage, comme dans l'exemple suivant pour un serveur autonome avec un seul partage.

   ```
   [global]
      netbios name = linuxserver
      workgroup = WORKGROUP
      security = user
      clustering = yes
   [share1]
      path = /srv/samba/share1
      read only = no
   ```

   d. Vérifiez le fichier **/etc/samba/smb.conf**.

   ```
   # testparm
   ```

2. Sur les deux nœuds du cluster, configurez CTDB :
   a. Créez le fichier **/etc/ctdb/nodes** et ajoutez les adresses IP des nœuds du cluster, comme dans cet exemple de fichier nodes.

   ```
   192.0.2.11
   192.0.2.12
   ```

b. Créez le fichier **/etc/ctdb/public_addresses** et ajoutez-y les adresses IP et les noms des périphériques réseau des interfaces publiques du cluster. Lorsque vous attribuez des adresses IP dans le fichier **public_addresses**, assurez-vous que ces adresses ne sont pas utilisées et qu'elles sont routables à partir du client visé. Le deuxième champ de chaque entrée du fichier **/etc/ctdb/public_addresses** est l'interface à utiliser sur les machines de la grappe pour l'adresse publique correspondante. Dans cet exemple de fichier **public_addresses**, l'interface **enp1s0** est utilisée pour toutes les adresses publiques.

```
192.0.2.201/24 enp1s0
192.0.2.202/24 enp1s0
```

Les interfaces publiques du cluster sont celles que les clients utilisent pour accéder à Samba depuis leur réseau. À des fins d'équilibrage de charge, ajoutez un enregistrement A pour chaque adresse IP publique de la grappe à votre zone DNS. Chacun de ces enregistrements doit résoudre le même nom d'hôte. Les clients utilisent le nom d'hôte pour accéder à Samba et le DNS distribue les clients aux différents nœuds du cluster.

c. Si vous exécutez le service **firewalld**, activez les ports requis par les services **ctdb** et **samba**.

```
# firewall-cmd --add-service=ctdb --add-service=samba --permanent
# firewall-cmd --reload
```

3. Sur un nœud de la grappe, mettez à jour les contextes SELinux :

a. Mettre à jour les contextes SELinux sur le partage GFS2.

```
[root@z1 ~]# semanage fcontext -at ctdbd_var_run_t -s system_u "/mnt/ctdb(/.)?"*
[root@z1 ~]# restorecon -Rv /mnt/ctdb
```

b. Mettre à jour le contexte SELinux sur le répertoire partagé dans Samba.

```
[root@z1 ~]# semanage fcontext -at samba_share_t -s system_u
"/srv/samba/share1(/.)?"*
[root@z1 ~]# restorecon -Rv /srv/samba/share1
```

Ressources supplémentaires

- Pour plus d'informations sur la configuration de Samba en tant que serveur autonome, comme dans cet exemple, voir le chapitre Utilisation de Samba en tant que serveur de la section Configuration et utilisation des services de fichiers en réseau .
- Mise en place d'une zone de transfert sur un serveur primaire BIND .

8.3. CONFIGURATION DES RESSOURCES DU CLUSTER SAMBA

Après avoir configuré un service Samba sur les deux nœuds d'un cluster de haute disponibilité à deux nœuds, configurez les ressources du cluster Samba pour le cluster.

Conditions préalables

- Un cluster Red Hat High Availability à deux nœuds configuré avec un système de fichiers GFS2, comme décrit dans Configuration d'un système de fichiers GFS2 pour un service Samba dans un cluster de haute disponibilité.

- Service Samba configuré sur les deux nœuds du cluster, comme décrit dans Configuration de Samba dans un cluster à haute disponibilité.

Procédure

1. Sur un nœud de la grappe, configurez les ressources de la grappe Samba :

   a. Créer la ressource CTDB, dans le groupe **samba-group**. L'agent de la ressource CTDB utilise les options **ctdb_*** spécifiées avec la commande **pcs** pour créer le fichier de configuration CTDB. Créez la ressource comme désactivée afin qu'elle ne démarre pas automatiquement avant que vous n'ayez configuré les contraintes d'ordre nécessaires.

   ```
   [root@z1 ~]# pcs resource create --disabled ctdb --group samba-group
   ocf:heartbeat:CTDB ctdb_recovery_lock=/mnt/ctdb/ctdb.lock
   ctdb_dbdir=/var/lib/ctdb ctdb_logfile=/var/log/ctdb.log op monitor interval=10
   timeout=30 op start timeout=90 op stop timeout=100
   ```

   b. Clonez le groupe de ressources **samba-group**.

   ```
   [root@z1 ~]# pcs resource clone samba-group
   ```

   c. Créer des contraintes d'ordre pour s'assurer que toutes les ressources de **Filesystem** sont exécutées avant les ressources de **samba-group**.

   ```
   [root@z1 ~]# pcs constraint order start ctdb_fs-clone then samba-group-clone
   [root@z1 ~]# pcs constraint order start csmb_fs1-clone then samba-group-clone
   ```

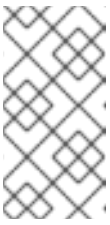
   d. Créez la ressource **samba** dans le groupe de ressources **samba-group**. Cela crée une contrainte d'ordre implicite entre CTDB et Samba, basée sur l'ordre dans lequel elles sont ajoutées.

   ```
   [root@z1 ~]# pcs resource create samba --group samba-group systemd:smb
   ```

   e. Activez les ressources **ctdb** et **samba**.

   ```
   [root@z1 ~]# pcs resource enable ctdb samba
   ```

   f. Vérifiez que tous les services ont bien démarré.

   NOTE

   Le démarrage de Samba, l'exportation des partages et la stabilisation de CTDB peuvent prendre quelques minutes. Si vous vérifiez l'état de la grappe avant la fin de ce processus, il se peut que les services **samba** ne soient pas encore en cours d'exécution.

   ```
   [root@z1 ~]# pcs status

   ...

   Full List of Resources:
     * fence-z1    (stonith:fence_xvm): Started z1.example.com
     * fence-z2    (stonith:fence_xvm): Started z2.example.com
   ```

```
 * Clone Set: locking-clone [locking]:
* Started: [ z1.example.com z2.example.com ]
 * Clone Set: shared_vg-clone [shared_vg]:
* Started: [ z1.example.com z2.example.com ]
 * Clone Set: ctdb_fs-clone [ctdb_fs]:
* Started: [ z1.example.com z2.example.com ]
 * Clone Set: csmb_fs1-clone [csmb_fs1]:
* Started: [ z1.example.com z2.example.com ]
  * Clone Set: samba-group-clone [samba-group]:
* Started: [ z1.example.com z2.example.com ]
```

2. Sur les deux nœuds du cluster, ajoutez un utilisateur local pour le répertoire de partage test.

   a. Ajouter l'utilisateur.

   ```
   # useradd -M -s /sbin/nologin example_user
   ```

   b. Définir un mot de passe pour l'utilisateur.

   ```
   # passwd example_user
   ```

   c. Définir un mot de passe SMB pour l'utilisateur.

   ```
   # smbpasswd -a example_user
   New SMB password:
   Retype new SMB password:
   Added user example_user
   ```

   d. Activer l'utilisateur dans la base de données Samba.

   ```
   # smbpasswd -e example_user
   ```

   e. Mettez à jour la propriété et les autorisations du fichier sur le partage GFS2 pour l'utilisateur Samba.

   ```
   # chown example_user:users /srv/samba/share1/
   # chmod 755 /srv/samba/share1/
   ```

8.4. VÉRIFICATION DE LA CONFIGURATION DE SAMBA EN CLUSTER

Si votre configuration de Samba en cluster a réussi, vous pouvez monter le partage Samba. Après avoir monté le partage, vous pouvez tester la récupération de Samba si le nœud de cluster qui exporte le partage Samba devient indisponible.

Procédure

1. Sur un système ayant accès à une ou plusieurs des adresses IP publiques configurées dans le fichier **/etc/ctdb/public_addresses** sur les nœuds du cluster, montez le partage Samba en utilisant l'une de ces adresses IP publiques.

   ```
   [root@testmount ~]# mkdir /mnt/sambashare
   [root@testmount ~]# mount -t cifs -o user=example_user //192.0.2.201/share1
   /mnt/sambashare
   ```

```
Password for example_user@//192.0.2.201/public: XXXXXXX
```

2. Vérifiez que le système de fichiers est monté.

```
[root@testmount ~]# mount | grep /mnt/sambashare
//192.0.2.201/public on /mnt/sambashare type cifs
(rw,relatime,vers=1.0,cache=strict,username=example_user,domain=LINUXSERVER,uid=0,nof
orceuid,gid=0,noforcegid,addr=192.0.2.201,unix,posixpaths,serverino,mapposix,acl,rsize=1048
76,wsize=65536,echo_interval=60,actimeo=1,user=example_user)
```

3. Vérifiez que vous pouvez créer un fichier sur le système de fichiers monté.

```
[root@testmount ~]# touch /mnt/sambashare/testfile1
[root@testmount ~]# ls /mnt/sambashare
testfile1
```

4. Déterminez le nœud de cluster qui exporte le partage Samba :

   a. Sur chaque nœud du cluster, affichez les adresses IP attribuées à l'interface spécifiée dans le fichier **public_addresses**. Les commandes suivantes affichent les adresses IPv4 attribuées à l'interface **enp1s0** sur chaque nœud.

```
[root@z1 ~]# ip -4 addr show enp1s0 | grep inet
    inet 192.0.2.11/24 brd 192.0.2.255 scope global dynamic noprefixroute enp1s0
    inet 192.0.2.201/24 brd 192.0.2.255 scope global secondary enp1s0

[root@z2 ~]# ip -4 addr show enp1s0 | grep inet
    inet 192.0.2.12/24 brd 192.0.2.255 scope global dynamic noprefixroute enp1s0
    inet 192.0.2.202/24 brd 192.0.2.255 scope global secondary enp1s0
```

   b. Dans la sortie de la commande **ip**, trouvez le nœud avec l'adresse IP que vous avez spécifiée avec la commande **mount** lorsque vous avez monté le partage.
   Dans cet exemple, l'adresse IP spécifiée dans la commande mount est 192.0.2.201. La sortie de la commande **ip** montre que l'adresse IP 192.0.2.201 est attribuée à **z1.example.com**.

5. Placez le nœud exportant le partage Samba en mode **standby**, ce qui aura pour effet d'empêcher le nœud d'héberger des ressources du cluster.

```
[root@z1 ~]# pcs node standby z1.example.com
```

6. À partir du système sur lequel vous avez monté le système de fichiers, vérifiez que vous pouvez toujours créer un fichier sur le système de fichiers.

```
[root@testmount ~]# touch /mnt/sambashare/testfile2
[root@testmount ~]# ls /mnt/sambashare
testfile1  testfile2
```

7. Supprimez les fichiers que vous avez créés pour vérifier que le système de fichiers a été monté avec succès. Si vous n'avez plus besoin que le système de fichiers soit monté, démontez-le à ce stade.

```
[root@testmount ~]# rm /mnt/sambashare/testfile1 /mnt/sambashare/testfile2
rm: remove regular empty file '/mnt/sambashare/testfile1'? y
rm: remove regular empty file '/mnt/sambashare/testfile1'? y
```

```
[root@testmount ~]# umount /mnt/sambashare
```

8. À partir de l'un des nœuds du cluster, restaurez les services du cluster sur le nœud que vous avez précédemment mis en mode veille. Cette opération ne ramènera pas nécessairement le service sur ce nœud.

```
[root@z1 ~]# pcs node unstandby z1.example.com
```

CHAPITRE 9. DÉMARRER AVEC L'INTERFACE WEB PCSD

L'interface Web **pcsd** est une interface utilisateur graphique qui permet de créer et de configurer les clusters Pacemaker/Corosync.

9.1. CONFIGURATION DE L'INTERFACE WEB PCSD

Configurez votre système afin d'utiliser l'interface Web **pcsd** pour configurer un cluster en suivant la procédure suivante.

Conditions préalables

- Les outils de configuration de Pacemaker sont installés.
- Votre système est prêt pour la configuration en grappe.

Pour obtenir des instructions sur l'installation du logiciel de cluster et la configuration de votre système pour la configuration du cluster, voir Installation du logiciel de cluster.

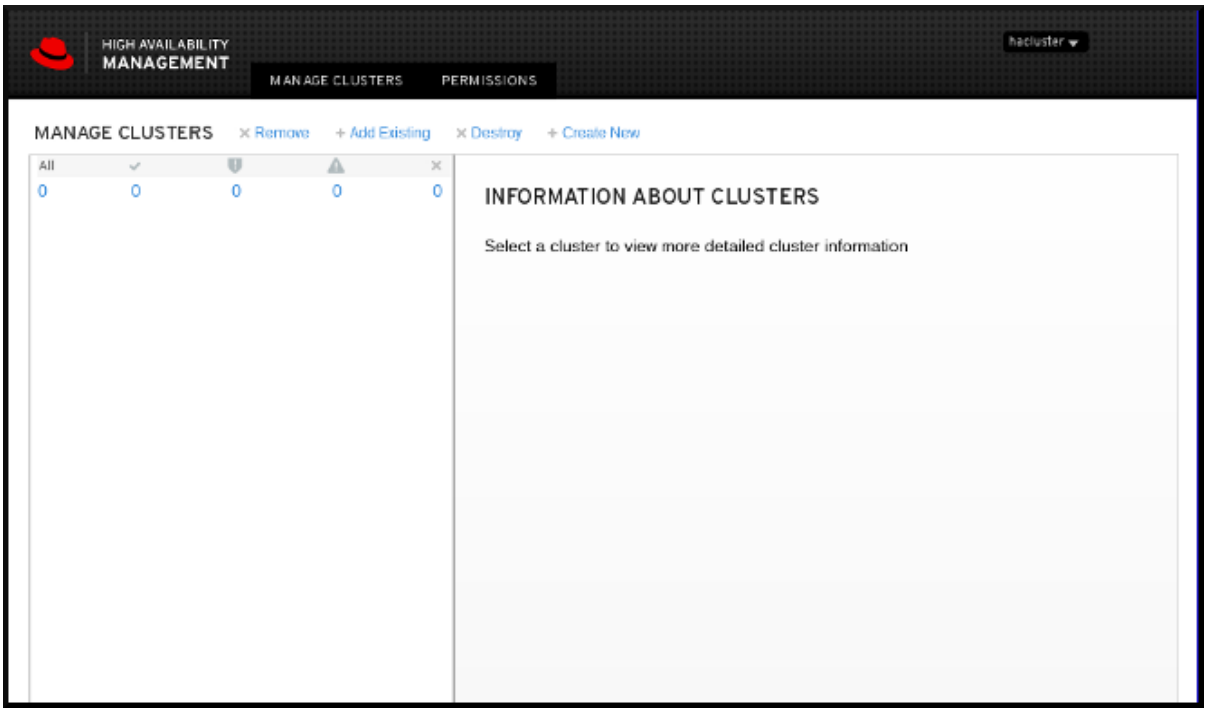
Procédure

1. Sur n'importe quel système, ouvrez un navigateur à l'URL suivante, en spécifiant l'un des nœuds du cluster (notez que cela utilise le protocole **https**). L'écran de connexion de l'interface Web **pcsd** s'affiche.

   ```
   https://nodename:2224
   ```

2. Connectez-vous en tant qu'utilisateur **hacluster**. La page **Manage Clusters** s'affiche alors, comme le montre la figure suivante.

   Figure 9.1. Page Gérer les clusters



9.2. CONFIGURATION D'UNE INTERFACE WEB PCSD À HAUTE DISPONIBILITÉ

Lorsque vous utilisez l'interface Web **pcsd**, vous vous connectez à l'un des nœuds de la grappe pour afficher les pages de gestion de la grappe. Si le nœud auquel vous vous connectez tombe en panne ou devient indisponible, vous pouvez vous reconnecter au cluster en ouvrant votre navigateur à une URL qui spécifie un autre nœud du cluster. Il est toutefois possible de configurer l'interface Web **pcsd** elle-même pour la haute disponibilité, auquel cas vous pouvez continuer à gérer la grappe sans avoir à saisir une nouvelle URL.

Procédure

Pour configurer l'interface Web **pcsd** pour la haute disponibilité, procédez comme suit.

1. Assurez-vous que les certificats de **pcsd** sont synchronisés entre les nœuds de la grappe en définissant **PCSD_SSL_CERT_SYNC_ENABLED** sur **true** dans le fichier de configuration **/etc/sysconfig/pcsd**. L'activation de la synchronisation des certificats permet à **pcsd** de synchroniser les certificats pour les commandes de configuration du cluster et d'ajout de nœuds. Dans RHEL 8, **PCSD_SSL_CERT_SYNC_ENABLED** est défini par défaut sur **false**.

2. Créez une ressource de cluster **IPaddr2**, qui est une adresse IP flottante que vous utiliserez pour vous connecter à l'interface Web **pcsd**. L'adresse IP ne doit pas être déjà associée à un nœud physique. Si le périphérique NIC de la ressource **IPaddr2** n'est pas spécifié, l'adresse IP flottante doit résider sur le même réseau que l'une des adresses IP attribuées de manière statique au nœud, sinon le périphérique NIC permettant d'attribuer l'adresse IP flottante ne peut pas être détecté correctement.

3. Créez des certificats SSL personnalisés à utiliser avec **pcsd** et assurez-vous qu'ils sont valides pour les adresses des nœuds utilisés pour se connecter à l'interface Web **pcsd**.

   a. Pour créer des certificats SSL personnalisés, vous pouvez utiliser des certificats génériques ou l'extension de certificat Subject Alternative Name. Pour plus d'informations sur le système de certificats Red Hat, consultez le Guide d'administration du système de certificats Red Hat.

   b. Installez les certificats personnalisés pour **pcsd** avec la commande **pcs pcsd certkey**.

   c. Synchronisez les certificats **pcsd** avec tous les nœuds du cluster à l'aide de la commande **pcs pcsd sync-certificates**.

4. Connectez-vous à l'interface Web **pcsd** à l'aide de l'adresse IP flottante que vous avez configurée en tant que ressource de cluster.

NOTE

Même si vous configurez l'interface Web **pcsd** pour la haute disponibilité, il vous sera demandé de vous connecter à nouveau lorsque le nœud auquel vous vous connectez sera hors service.

CHAPITRE 10. CONFIGURATION DE LA CLÔTURE DANS UN CLUSTER RED HAT HIGH AVAILABILITY

Un nœud qui ne répond pas peut encore accéder à des données. La seule façon d'être certain que vos données sont en sécurité est de clôturer le nœud à l'aide de STONITH. STONITH est un acronyme pour "Shoot The Other Node In The Head" (Tirez sur l'autre nœud dans la tête) et il protège vos données contre la corruption par des nœuds malveillants ou l'accès concurrent. Grâce à STONITH, vous pouvez être certain qu'un nœud est réellement hors ligne avant d'autoriser l'accès aux données à partir d'un autre nœud.

STONITH a également un rôle à jouer dans le cas où un service en cluster ne peut pas être arrêté. Dans ce cas, le cluster utilise STONITH pour forcer la mise hors ligne de l'ensemble du nœud, ce qui permet de démarrer le service ailleurs en toute sécurité.

Pour obtenir des informations générales plus complètes sur la clôture et son importance dans un cluster Red Hat High Availability, voir Clôture dans un cluster Red Hat High Availability .

Vous mettez en œuvre STONITH dans un cluster Pacemaker en configurant des périphériques de clôture pour les nœuds du cluster.

10.1. AFFICHAGE DES AGENTS DE CLÔTURE DISPONIBLES ET DE LEURS OPTIONS

Les commandes suivantes permettent d'afficher les agents de clôture disponibles et les options disponibles pour des agents de clôture spécifiques.

Cette commande dresse la liste de tous les agents de clôture disponibles. Lorsque vous spécifiez un filtre, cette commande n'affiche que les agents de clôtures qui correspondent au filtre.

```
pcs stonith list [filter]
```

Cette commande affiche les options de l'agent de clôture spécifié.

```
pcs stonith décrire [stonith_agent]
```

Par exemple, la commande suivante affiche les options de l'agent de clôture pour APC via telnet/SSH.

```
# pcs stonith describe fence_apc
Stonith options for: fence_apc
  ipaddr (required): IP Address or Hostname
  login (required): Login Name
  passwd: Login password or passphrase
  passwd_script: Script to retrieve password
  cmd_prompt: Force command prompt
  secure: SSH connection
  port (required): Physical plug number or name of virtual machine
  identity_file: Identity file for ssh
  switch: Physical switch number on device
  inet4_only: Forces agent to use IPv4 addresses only
  inet6_only: Forces agent to use IPv6 addresses only
  ipport: TCP port to use for connection with device
  action (required): Fencing Action
  verbose: Verbose mode
```

```
debug: Write debug information to given file
version: Display version information and exit
help: Display help and exit
separator: Separator for CSV created by operation list
power_timeout: Test X seconds for status change after ON/OFF
shell_timeout: Wait X seconds for cmd prompt after issuing command
login_timeout: Wait X seconds for cmd prompt after login
power_wait: Wait X seconds after issuing ON/OFF
delay: Wait X seconds before fencing is started
retry_on: Count of attempts to retry power on
```

AVERTISSEMENT

Pour les agents de clôture qui proposent l'option **method**, la valeur **cycle** n'est pas prise en charge et ne doit pas être spécifiée, car elle peut entraîner une corruption des données.

10.2. CRÉATION D'UN DISPOSITIF DE CLÔTURE

Le format de la commande de création d'un dispositif de clôture est le suivant. Pour obtenir la liste des options de création de dispositif de clôture disponibles, consultez l'écran **pcs stonith -h**.

```
pcs stonith create stonith_id stonith_device_type [stonith_device_options] [op  operation_action operation_options]
```

La commande suivante crée un seul dispositif de clôture pour un seul nœud.

```
# pcs stonith create MyStonith fence_virt pcmk_host_list=f1 op monitor interval=30s
```

Certains dispositifs de clôture ne peuvent clôturer qu'un seul nœud, tandis que d'autres peuvent clôturer plusieurs nœuds. Les paramètres que vous spécifiez lorsque vous créez un dispositif de clôture dépendent de ce que votre dispositif de clôture prend en charge et exige.

- Certains dispositifs de clôture peuvent déterminer automatiquement les nœuds qu'ils peuvent clôturer.
- Vous pouvez utiliser le paramètre **pcmk_host_list** lors de la création d'un dispositif de clôture pour spécifier toutes les machines qui sont contrôlées par ce dispositif de clôture.
- Certains dispositifs de clôture exigent une correspondance entre les noms d'hôtes et les spécifications qu'ils comprennent. Vous pouvez mapper les noms d'hôtes avec le paramètre **pcmk_host_map** lors de la création d'un dispositif de clôture.

Pour plus d'informations sur les paramètres **pcmk_host_list** et **pcmk_host_map**, voir Propriétés générales des dispositifs de clôture.

Après avoir configuré un dispositif de clôture, il est impératif de le tester pour s'assurer qu'il fonctionne correctement. Pour plus d'informations sur le test d'un dispositif de clôture, voir Test d'un dispositif de clôture.

10.3. PROPRIÉTÉS GÉNÉRALES DES DISPOSITIFS DE CLÔTURE

Il existe de nombreuses propriétés générales que vous pouvez définir pour les dispositifs de clôture, ainsi que diverses propriétés de cluster qui déterminent le comportement des clôtures.

Tout nœud de cluster peut clôturer n'importe quel autre nœud de cluster avec n'importe quel dispositif de clôture, que la ressource de clôture soit démarrée ou arrêtée. La question de savoir si la ressource est démarrée ne concerne que le moniteur récurrent du dispositif, et non son utilisation, à l'exception des cas suivants :

- Vous pouvez désactiver un dispositif de clôture en exécutant la commande **pcs stonith disable *stonith_id*** en exécutant la commande Cela empêchera tout nœud d'utiliser ce dispositif.
- Pour empêcher un nœud spécifique d'utiliser un dispositif de clôture, vous pouvez configurer des contraintes d'emplacement pour la ressource de clôture à l'aide de la commande **pcs constraint location … avoids**.
- La configuration de **stonith-enabled=false** désactivera complètement la clôture. Notez cependant que Red Hat ne prend pas en charge les clusters lorsque la clôture est désactivée, car elle n'est pas adaptée à un environnement de production.

Le tableau suivant décrit les propriétés générales que vous pouvez définir pour les dispositifs de clôture.

Tableau 10.1. Propriétés générales des dispositifs de clôture

Field	Type	Défaut	Description
pcmk_host_map	chaîne de caractères		Une correspondance entre les noms d'hôtes et les numéros de ports pour les dispositifs qui ne prennent pas en charge les noms d'hôtes. Par exemple : **node1:1;node2:2,3** indique au cluster d'utiliser le port 1 pour le nœud 1 et les ports 2 et 3 pour le nœud 2. La propriété **pcmk_host_map** prend en charge les caractères spéciaux à l'intérieur des valeurs **pcmk_host_map** en utilisant une barre oblique inverse devant la valeur. Par exemple, vous pouvez spécifier **pcmk_host_map="node3:plug\ 1"** pour inclure un espace dans l'alias d'hôte.
pcmk_host_list	chaîne de caractères		Liste des machines contrôlées par ce dispositif (facultatif sauf si **pcmk_host_check=static-list**).

Field	Type	Défaut	Description
pcmk_host_check	chaîne de caractères	* **static-list** si **pcmk_host_list** ou **pcmk_host_map** est activé * Sinon, **dynamic-list** si le dispositif de clôture prend en charge l'action **list** * Sinon, **status** si le dispositif de clôture prend en charge l'action **status** *Sinon, **none**.	Comment déterminer quelles machines sont contrôlées par le dispositif. Valeurs autorisées : **dynamic-list** (interroger l'appareil), **static-list** (vérifier l'attribut **pcmk_host_list**), none (supposer que chaque appareil peut clôturer chaque machine)

Le tableau suivant résume les propriétés supplémentaires que vous pouvez définir pour les dispositifs de clôture. Notez que ces propriétés ne sont destinées qu'à un usage avancé.

Tableau 10.2. Propriétés avancées des dispositifs de clôture

Field	Type	Défaut	Description
pcmk_host_argument	chaîne de caractères	port	Paramètre alternatif à fournir à la place de port. Certains appareils ne prennent pas en charge le paramètre port standard ou peuvent en fournir d'autres. Utilisez ce paramètre pour spécifier un autre paramètre, spécifique au périphérique, qui doit indiquer la machine à clôturer. Une valeur de **none** peut être utilisée pour indiquer au cluster de ne pas fournir de paramètres supplémentaires.
pcmk_reboot_action	chaîne de caractères	redémarrage	Une commande alternative à exécuter à la place de **reboot**. Certains périphériques ne prennent pas en charge les commandes standard ou peuvent en fournir d'autres. Utilisez cette option pour spécifier une commande alternative, spécifique au périphérique, qui met en œuvre l'action de redémarrage.

Field	Type	Défaut	Description
pcmk_reboot_timeout	temps	60s	Spécifiez un délai alternatif à utiliser pour les actions de redémarrage au lieu de **stonith-timeout**. Certains périphériques ont besoin de beaucoup plus/moins de temps pour se terminer que la normale. Cette option permet de spécifier un délai alternatif, spécifique au périphérique, pour les actions de redémarrage.
pcmk_reboot_retries	entier	2	Nombre maximal de tentatives de la commande **reboot** dans le délai imparti. Certains appareils ne supportent pas les connexions multiples. Les opérations peuvent échouer si l'appareil est occupé par une autre tâche, de sorte que Pacemaker retente automatiquement l'opération, s'il reste du temps. Utilisez cette option pour modifier le nombre de tentatives de redémarrage de Pacemaker avant d'abandonner.
pcmk_off_action	chaîne de caractères	éteint	Une commande alternative à exécuter à la place de **off**. Certains appareils ne prennent pas en charge les commandes standard ou peuvent en fournir d'autres. Utilisez cette option pour spécifier une commande alternative, spécifique à l'appareil, qui met en œuvre l'action off.
pcmk_off_timeout	temps	60s	Spécifiez un délai alternatif à utiliser pour les actions d'arrêt au lieu de **stonith-timeout**. Certains périphériques ont besoin de beaucoup plus ou de beaucoup moins de temps que la normale pour se terminer. Cette option permet de spécifier un délai alternatif, spécifique au périphérique, pour les actions d'arrêt.
pcmk_off_retries	entier	2	Nombre maximal de tentatives de commande de désactivation dans le délai imparti. Certains appareils ne supportent pas les connexions multiples. Les opérations peuvent échouer si l'appareil est occupé par une autre tâche, de sorte que Pacemaker retente automatiquement l'opération, s'il reste du temps. Utilisez cette option pour modifier le nombre de fois que Pacemaker retente les actions off avant d'abandonner.

Field	Type	Défaut	Description
pcmk_list_action	chaîne de caractères	liste	Une commande alternative à exécuter à la place de **list**. Certains appareils ne prennent pas en charge les commandes standard ou peuvent en fournir d'autres. Utilisez cette option pour spécifier une commande alternative, spécifique à l'appareil, qui met en œuvre l'action de liste.
pcmk_list_timeout	temps	60s	Spécifiez un autre délai à utiliser pour les actions de la liste. Certains périphériques ont besoin de beaucoup plus ou de beaucoup moins de temps que la normale pour terminer une action. Cette option permet de spécifier un délai alternatif, spécifique au périphérique, pour les actions de liste.
pcmk_list_retries	entier	2	Nombre maximal de tentatives de la commande **list** dans le délai imparti. Certains appareils ne supportent pas les connexions multiples. Les opérations peuvent échouer si l'appareil est occupé par une autre tâche, de sorte que Pacemaker retente automatiquement l'opération, s'il reste du temps. Utilisez cette option pour modifier le nombre de fois que Pacemaker retente les actions de la liste avant d'abandonner.
pcmk_monitor_action	chaîne de caractères	moniteur	Une commande alternative à exécuter à la place de **monitor**. Certains appareils ne prennent pas en charge les commandes standard ou peuvent en fournir d'autres. Utilisez cette option pour spécifier une commande alternative, spécifique à l'appareil, qui met en œuvre l'action du moniteur.
pcmk_monitor_timeout	temps	60s	Spécifiez un délai alternatif à utiliser pour les actions de surveillance au lieu de **stonith-timeout**. Certains périphériques ont besoin de beaucoup plus ou de beaucoup moins de temps que la normale pour se terminer. Cette option permet de spécifier un délai alternatif, spécifique au périphérique, pour les actions de surveillance.

Field	Type	Défaut	Description
pcmk_monitor_retries	entier	2	Nombre maximal de tentatives de la commande **monitor** dans le délai imparti. Certains appareils ne supportent pas les connexions multiples. Les opérations peuvent échouer si l'appareil est occupé par une autre tâche, de sorte que Pacemaker retente automatiquement l'opération, s'il reste du temps. Utilisez cette option pour modifier le nombre de fois que Pacemaker retente les actions de surveillance avant d'abandonner.
pcmk_status_action	chaîne de caractères	status	Une commande alternative à exécuter à la place de **status**. Certains appareils ne prennent pas en charge les commandes standard ou peuvent en fournir d'autres. Utilisez cette option pour spécifier une commande alternative, spécifique au périphérique, qui met en œuvre l'action d'état.
pcmk_status_timeout	temps	60s	Spécifiez un délai alternatif à utiliser pour les actions d'état au lieu de **stonith-timeout**. Certains périphériques ont besoin de beaucoup plus ou de beaucoup moins de temps que la normale pour se terminer. Cette option permet de spécifier un délai alternatif, spécifique au périphérique, pour les actions d'état.
pcmk_status_retries	entier	2	Nombre maximal de tentatives de commande d'état dans le délai imparti. Certains appareils ne supportent pas les connexions multiples. Les opérations peuvent échouer si l'appareil est occupé par une autre tâche, de sorte que Pacemaker retente automatiquement l'opération, s'il reste du temps. Utilisez cette option pour modifier le nombre de fois que Pacemaker retente les actions d'état avant d'abandonner.

Field	Type	Défaut	Description
pcmk_delay_base	chaîne de caractères	0s	Activez un délai de base pour les actions stonith et spécifiez une valeur de délai de base. Dans un cluster avec un nombre pair de nœuds, la configuration d'un délai peut permettre d'éviter que les nœuds se clôturent les uns les autres en même temps et de manière égale. Un délai aléatoire peut être utile lorsque le même dispositif de clôture est utilisé pour tous les nœuds, et des délais statiques différents peuvent être utiles sur chaque dispositif de clôture lorsqu'un dispositif distinct est utilisé pour chaque nœud. Le délai global est dérivé d'une valeur de délai aléatoire à laquelle s'ajoute ce délai statique, de sorte que la somme reste inférieure au délai maximal. Si vous définissez **pcmk_delay_base** mais pas **pcmk_delay_max**, le délai ne comporte pas de composante aléatoire et sera égal à la valeur de **pcmk_delay_base**. Vous pouvez spécifier des valeurs différentes pour différents nœuds avec le paramètre **pcmk_delay_base**. Cela permet d'utiliser un seul dispositif de clôture dans un cluster à deux nœuds, avec un délai différent pour chaque nœud. Cela permet d'éviter que chaque nœud tente de clôturer l'autre nœud en même temps. Pour spécifier des valeurs différentes pour différents nœuds, vous associez les noms d'hôte à la valeur du délai pour ce nœud en utilisant une syntaxe similaire à celle de **pcmk_host_map**. Par exemple, **node1:0;node2:10s** n'utilise aucun délai pour clôturer **node1** et un délai de 10 secondes pour clôturer **node2**. Certains agents de clôture individuels mettent en œuvre un paramètre "delay", qui est indépendant des délais configurés avec la propriété **pcmk_delay_***. Si ces deux délais sont configurés, ils sont additionnés et ne doivent donc généralement pas être utilisés conjointement.

Field	Type	Défaut	Description
pcmk_delay_max	temps	0s	Activez un délai aléatoire pour les actions stonith et spécifiez le délai aléatoire maximum. Dans un cluster avec un nombre pair de nœuds, la configuration d'un délai peut permettre d'éviter que les nœuds se clôturent les uns les autres en même temps et de manière égale. Un délai aléatoire peut être utile lorsque le même dispositif de clôture est utilisé pour tous les nœuds, et des délais statiques différents peuvent être utiles sur chaque dispositif de clôture lorsqu'un dispositif distinct est utilisé pour chaque nœud. Le délai global est dérivé de cette valeur de délai aléatoire, à laquelle on ajoute un délai statique de sorte que la somme reste inférieure au délai maximal. Si vous définissez **pcmk_delay_max** mais pas **pcmk_delay_base**, il n'y a pas de composante statique dans le délai. Certains agents de clôture individuels mettent en œuvre un paramètre "delay", qui est indépendant des délais configurés avec la propriété **pcmk_delay_*** . Si ces deux délais sont configurés, ils sont additionnés et ne doivent donc généralement pas être utilisés conjointement.
pcmk_action_limit	entier	1	Le nombre maximum d'actions qui peuvent être effectuées en parallèle sur cet appareil. La propriété de cluster **concurrent-fencing=true** doit d'abord être configurée (c'est la valeur par défaut). La valeur -1 est illimitée.
pcmk_on_action	chaîne de caractères	sur	Pour une utilisation avancée uniquement : Une commande alternative à exécuter à la place de **on**. Certains appareils ne prennent pas en charge les commandes standard ou peuvent en fournir d'autres. Utilisez cette option pour spécifier une commande alternative, spécifique à l'appareil, qui met en œuvre l'action **on**.

Field	Type	Défaut	Description
pcmk_on_timeout	temps	60s	Pour une utilisation avancée uniquement : Spécifiez un délai alternatif à utiliser pour les actions **on** au lieu de **stonith-timeout**. Certains dispositifs ont besoin de beaucoup plus ou de beaucoup moins de temps que la normale pour s'exécuter. Cette option permet de spécifier un délai alternatif, spécifique au périphérique, pour les actions **on**.
pcmk_on_retries	entier	2	Pour une utilisation avancée uniquement : Nombre maximal de tentatives de commande **on** dans le délai imparti. Certains appareils ne prennent pas en charge les connexions multiples. Les opérations peuvent **fail** si l'appareil est occupé par une autre tâche, de sorte que Pacemaker retente automatiquement l'opération, s'il reste du temps. Utilisez cette option pour modifier le nombre de fois que Pacemaker retente les actions **on** avant d'abandonner.

Outre les propriétés que vous pouvez définir pour les dispositifs de clôture individuels, vous pouvez également définir des propriétés de cluster qui déterminent le comportement des clôtures, comme décrit dans le tableau suivant.

Tableau 10.3. Propriétés des grappes qui déterminent le comportement des clôtures

Option	Défaut	Description
stonith-enabled	true	Indique que les nœuds défaillants et les nœuds dont les ressources ne peuvent être arrêtées doivent être clôturés. Pour protéger vos données, vous devez définir ce paramètre à l'adresse **true**. Si **true**, ou non défini, le cluster refusera de démarrer les ressources à moins qu'une ou plusieurs ressources STONITH n'aient été configurées également. Red Hat ne prend en charge que les clusters dont la valeur est définie sur **true**.
stonith-action	redémarrage	Action à envoyer au dispositif STONITH. Valeurs autorisées : **reboot**, **off**. La valeur **poweroff** est également autorisée, mais elle n'est utilisée que pour les anciens dispositifs.

Option	Défaut	Description
stonith-timeout	60s	Durée d'attente pour la réalisation d'une action STONITH.
stonith-max-attempts	10	Combien de fois la clôture peut-elle échouer pour une cible avant que le cluster n'essaie plus immédiatement de la réessayer.
stonith-watchdog-timeout		Temps d'attente maximal avant qu'un nœud puisse être considéré comme tué par le chien de garde matériel. Il est recommandé de fixer cette valeur à deux fois la valeur du délai d'attente du chien de garde matériel. Cette option n'est nécessaire que si la configuration SBD watchdog-only est utilisée pour la clôture.
concurrent-fencing	true	Permet d'effectuer des opérations de clôture en parallèle.
fence-reaction	arrêter	Détermine la manière dont un nœud de grappe doit réagir s'il est informé de l'existence de sa propre clôture. Un nœud de la grappe peut recevoir une notification de sa propre clôture si celle-ci est mal configurée ou si une clôture de tissu est utilisée et ne coupe pas la communication de la grappe. Les valeurs autorisées sont **stop** pour tenter d'arrêter immédiatement Pacemaker et de le maintenir à l'arrêt, ou **panic** pour tenter de redémarrer immédiatement le nœud local, en revenant à l'arrêt en cas d'échec. Bien que la valeur par défaut de cette propriété soit **stop**, le choix le plus sûr est **panic**, qui tente de redémarrer immédiatement le nœud local. Si vous préférez le comportement d'arrêt, comme c'est probablement le cas avec la clôture de tissu, il est recommandé de définir explicitement cette propriété.

Pour plus d'informations sur la définition des propriétés de la grappe, voir Définition et suppression des propriétés de la grappe.

10.4. TEST D'UN DISPOSITIF DE CLÔTURE

La clôture est une partie fondamentale de l'infrastructure Red Hat Cluster et il est important de valider ou de tester le bon fonctionnement de la clôture.

Procédure

Utilisez la procédure suivante pour tester un dispositif de clôture.

1. Utilisez ssh, telnet, HTTP ou tout autre protocole à distance utilisé pour vous connecter au dispositif afin de vous connecter manuellement et de tester le dispositif de clôture ou de voir les résultats obtenus. Par exemple, si vous allez configurer la clôture pour un périphérique IPMI, essayez de vous connecter à distance à l'aide de **ipmitool**. Prenez note des options utilisées lors de la connexion manuelle, car ces options peuvent être nécessaires lors de l'utilisation de l'agent de clôture.
   Si vous ne parvenez pas à vous connecter au dispositif de clôture, vérifiez que le dispositif est pingable, qu'aucune configuration de pare-feu, par exemple, n'empêche l'accès au dispositif de clôture, que l'accès à distance est activé sur le dispositif de clôture et que les informations d'identification sont correctes.

2. Exécutez l'agent de clôture manuellement, à l'aide du script de l'agent de clôture. Il n'est pas nécessaire que les services du cluster soient en cours d'exécution, vous pouvez donc effectuer cette étape avant que le périphérique ne soit configuré dans le cluster. Cela permet de s'assurer que le dispositif de clôture répond correctement avant de poursuivre.

   NOTE

   Ces exemples utilisent le script de l'agent de clôture **fence_ipmilan** pour un périphérique iLO. L'agent de clôture que vous utiliserez et la commande qui appelle cet agent dépendent du matériel de votre serveur. Vous devez consulter la page de manuel de l'agent de clôture que vous utilisez pour déterminer les options à spécifier. Vous devrez généralement connaître le nom d'utilisateur et le mot de passe du périphérique de clôture, ainsi que d'autres informations relatives au périphérique de clôture.

   L'exemple suivant montre le format à utiliser pour exécuter le script de l'agent de clôture **fence_ipmilan** avec le paramètre **-o status** afin de vérifier l'état de l'interface du dispositif de clôture sur un autre nœud sans le clôturer. Cela vous permet de tester le dispositif et de le faire fonctionner avant de tenter de redémarrer le nœud. Lors de l'exécution de cette commande, vous devez spécifier le nom et le mot de passe d'un utilisateur iLO disposant d'autorisations de mise sous tension et hors tension pour le dispositif iLO.

   ```
   # fence_ipmilan -a ipaddress -l username -p password -o status
   ```

   L'exemple suivant présente le format à utiliser pour exécuter le script de l'agent de clôture **fence_ipmilan** avec le paramètre **-o reboot**. L'exécution de cette commande sur un nœud redémarre le nœud géré par ce dispositif iLO.

   ```
   # fence_ipmilan -a ipaddress -l username -p password -o reboot
   ```

   Si l'agent de clôture n'a pas réussi à effectuer correctement une action d'état, d'arrêt, de mise en marche ou de redémarrage, vous devez vérifier le matériel, la configuration du dispositif de clôture et la syntaxe de vos commandes. En outre, vous pouvez exécuter le script de l'agent de clôture avec la sortie de débogage activée. La sortie de débogage est utile pour certains agents de clôture afin de voir où, dans la séquence des événements, le script de l'agent de clôture échoue lors de la connexion au dispositif de clôture.

   ```
   # fence_ipmilan -a ipaddress -l username -p password -o status -D /tmp/$(hostname)-fence_agent.debug
   ```

Lorsque vous diagnostiquez une défaillance, vous devez vous assurer que les options que vous avez spécifiées lors de la connexion manuelle au dispositif de clôture sont identiques à celles que vous avez transmises à l'agent de clôture à l'aide du script de l'agent de clôture.

Pour les agents de clôture qui prennent en charge une connexion cryptée, vous pouvez voir une erreur due à l'échec de la validation du certificat, ce qui nécessite que vous fassiez confiance à l'hôte ou que vous utilisiez le paramètre **ssl-insecure** de l'agent de clôture. De même, si SSL/TLS est désactivé sur le périphérique cible, vous devrez peut-être en tenir compte lors de la définition des paramètres SSL pour l'agent de clôture.

NOTE

Si l'agent de clôture testé est **fence_drac**, **fence_ilo** ou un autre agent de clôture pour un dispositif de gestion des systèmes qui continue d'échouer, essayez de nouveau **fence_ipmilan**. La plupart des cartes de gestion des systèmes prennent en charge la connexion à distance IPMI et le seul agent de clôture pris en charge est **fence_ipmilan**.

3. Une fois que le dispositif de clôture a été configuré dans le cluster avec les mêmes options que celles qui ont fonctionné manuellement et que le cluster a été démarré, testez la clôture avec la commande **pcs stonith fence** à partir de n'importe quel nœud (ou même plusieurs fois à partir de différents nœuds), comme dans l'exemple suivant. La commande **pcs stonith fence** lit la configuration du cluster à partir de la CIB et appelle l'agent fence tel qu'il est configuré pour exécuter l'action de clôture. Cela permet de vérifier que la configuration de la grappe est correcte.

```
# pcs stonith fence node_name
```

Si la commande **pcs stonith fence** fonctionne correctement, cela signifie que la configuration de clôture de la grappe devrait fonctionner lorsqu'un événement de clôture se produit. Si la commande échoue, cela signifie que la gestion de la grappe ne peut pas invoquer le dispositif de clôture par le biais de la configuration qu'elle a récupérée. Vérifiez les points suivants et mettez à jour la configuration de votre cluster si nécessaire.

- Vérifiez la configuration de votre clôture. Par exemple, si vous avez utilisé une carte d'hôte, vous devez vous assurer que le système peut trouver le nœud à l'aide du nom d'hôte que vous avez fourni.

- Vérifiez si le mot de passe et le nom d'utilisateur du périphérique contiennent des caractères spéciaux susceptibles d'être mal interprétés par l'interpréteur de commandes bash. Veiller à ce que les mots de passe et les noms d'utilisateur soient entourés de guillemets peut résoudre ce problème.

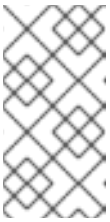
- Vérifiez que vous pouvez vous connecter à l'appareil en utilisant l'adresse IP ou le nom d'hôte exact que vous avez spécifié dans la commande **pcs stonith**. Par exemple, si vous indiquez le nom d'hôte dans la commande stonith mais que vous testez en utilisant l'adresse IP, il ne s'agit pas d'un test valide.

- Si le protocole utilisé par le dispositif de clôture vous est accessible, utilisez ce protocole pour essayer de vous connecter au dispositif. Par exemple, de nombreux agents utilisent ssh ou telnet. Vous devez essayer de vous connecter au dispositif avec les informations d'identification que vous avez fournies lors de la configuration du dispositif, pour voir si vous obtenez une invite valide et si vous pouvez vous connecter au dispositif.
  Si vous déterminez que tous vos paramètres sont appropriés mais que vous avez toujours des difficultés à vous connecter à votre dispositif de clôture, vous pouvez vérifier la

journalisation sur le dispositif de clôture lui-même, si le dispositif le permet, qui montrera si l'utilisateur s'est connecté et quelle commande l'utilisateur a émise. Vous pouvez également rechercher dans le fichier **/var/log/messages** les occurrences de stonith et d'error, qui peuvent donner une idée de ce qui se passe, mais certains agents peuvent fournir des informations supplémentaires.

4. Une fois que les tests du dispositif de clôture fonctionnent et que la grappe est opérationnelle, testez une défaillance réelle. Pour ce faire, effectuez une action dans le cluster qui devrait déclencher une perte de jeton.

   - Démonter un réseau. La manière dont vous démontez un réseau dépend de votre configuration spécifique. Dans de nombreux cas, vous pouvez retirer physiquement les câbles réseau ou d'alimentation de l'hôte. Pour plus d'informations sur la simulation d'une panne de réseau, voir Quelle est la bonne façon de simuler une panne de réseau sur un cluster RHEL ?

     NOTE

     Il n'est pas recommandé de désactiver l'interface réseau de l'hôte local plutôt que de déconnecter physiquement le réseau ou les câbles d'alimentation pour tester les clôtures, car cela ne simule pas avec précision une défaillance typique dans le monde réel.

   - Bloquer le trafic corosync entrant et sortant à l'aide du pare-feu local.
     L'exemple suivant bloque corosync, en supposant que le port par défaut de corosync est utilisé, que **firewalld** est utilisé comme pare-feu local et que l'interface réseau utilisée par corosync se trouve dans la zone de pare-feu par défaut :

     ```
     # firewall-cmd --direct --add-rule ipv4 filter OUTPUT 2 -p udp --dport=5405 -j DROP
     # firewall-cmd --add-rich-rule='rule family="ipv4" port port="5405" protocol="udp" drop
     ```

   - Simulez un crash et faites paniquer votre machine avec **sysrq-trigger**. Notez toutefois que le déclenchement d'une panique du noyau peut entraîner une perte de données ; il est recommandé de désactiver d'abord les ressources de votre cluster.

     ```
     # echo c > /proc/sysrq-trigger
     ```

10.5. CONFIGURATION DES NIVEAUX DE CLÔTURE

Pacemaker prend en charge les nœuds de clôture avec plusieurs dispositifs grâce à une fonction appelée topologies de clôture. Pour mettre en œuvre des topologies, créez les dispositifs individuels comme vous le feriez normalement, puis définissez un ou plusieurs niveaux de clôture dans la section topologie de clôture de la configuration.

Pacemaker traite les niveaux de clôture comme suit :

- Chaque niveau est tenté par ordre numérique croissant, en commençant par le niveau 1.
- En cas d'échec d'un dispositif, le traitement s'arrête au niveau actuel. Aucun autre dispositif de ce niveau n'est utilisé et le niveau suivant est tenté à la place.
- Si tous les dispositifs sont clôturés avec succès, ce niveau est réussi et aucun autre niveau n'est tenté.

- L'opération est terminée lorsqu'un niveau a été franchi (succès) ou que tous les niveaux ont été tentés (échec).

Utilisez la commande suivante pour ajouter un niveau de clôture à un nœud. Les dispositifs sont donnés sous la forme d'une liste d'identifiants de stonith séparés par des virgules, qui sont tentés pour le nœud à ce niveau.

```
pcs stonith level add level node devices
```

La commande suivante répertorie tous les niveaux de clôture actuellement configurés.

```
pcs stonith level
```

Dans l'exemple suivant, deux dispositifs de clôture sont configurés pour le nœud **rh7-2**: un dispositif de clôture ilo appelé **my_ilo** et un dispositif de clôture apc appelé **my_apc**. Ces commandes configurent les niveaux de clôture de sorte que si le dispositif **my_ilo** tombe en panne et ne peut clôturer le nœud, Pacemaker tentera d'utiliser le dispositif **my_apc**. Cet exemple montre également la sortie de la commande **pcs stonith level** après la configuration des niveaux.

```
# pcs stonith level add 1 rh7-2 my_ilo
# pcs stonith level add 2 rh7-2 my_apc
# pcs stonith level
 Node: rh7-2
  Level 1 - my_ilo
  Level 2 - my_apc
```

La commande suivante supprime le niveau de clôture pour le nœud et les dispositifs spécifiés. Si aucun nœud ou dispositif n'est spécifié, le niveau de clôture que vous indiquez est supprimé pour tous les nœuds.

```
pcs stonith level remove level [node_id] [stonith_id] ... [stonith_id]
```

La commande suivante permet d'effacer les niveaux de clôture du nœud ou de l'identifiant du stonith spécifié. Si vous ne spécifiez pas de nœud ou d'identifiant de stonith, tous les niveaux de clôture sont effacés.

```
pcs stonith level clear [node]|stonith_id(s)]
```

Si vous spécifiez plus d'un identifiant de stonith, ils doivent être séparés par une virgule et sans espace, comme dans l'exemple suivant.

```
# pcs stonith level clear dev_a,dev_b
```

La commande suivante vérifie que tous les dispositifs et nœuds de clôture spécifiés dans les niveaux de clôture existent.

```
pcs stonith level verify
```

Vous pouvez spécifier des nœuds dans la topologie de clôture par une expression régulière appliquée à un nom de nœud et par un attribut de nœud et sa valeur. Par exemple, les commandes suivantes configurent les nœuds **node1**, **node2** et **node3** pour qu'ils utilisent les dispositifs de clôture **apc1** et **apc2**, et les nœuds **node4**, **node5** et **node6** pour qu'ils utilisent les dispositifs de clôture **apc3** et **apc4**.

```
# pcs stonith level add 1 "regexp%node[1-3]" apc1,apc2
# pcs stonith level add 1 "regexp%node[4-6]" apc3,apc4
```

Les commandes suivantes permettent d'obtenir les mêmes résultats en utilisant la correspondance des attributs des nœuds.

```
# pcs node attribute node1 rack=1
# pcs node attribute node2 rack=1
# pcs node attribute node3 rack=1
# pcs node attribute node4 rack=2
# pcs node attribute node5 rack=2
# pcs node attribute node6 rack=2
# pcs stonith level add 1 attrib%rack=1 apc1,apc2
# pcs stonith level add 1 attrib%rack=2 apc3,apc4
```

10.6. CONFIGURATION DES CLÔTURES POUR LES ALIMENTATIONS REDONDANTES

Lors de la configuration de la clôture pour les alimentations redondantes, le cluster doit s'assurer que lors de la tentative de redémarrage d'un hôte, les deux alimentations sont éteintes avant que l'une d'entre elles ne soit rallumée.

Si le nœud ne perd jamais complètement son alimentation, il peut ne pas libérer ses ressources. Il est donc possible que des nœuds accèdent simultanément à ces ressources et les corrompent.

Vous devez définir chaque dispositif une seule fois et spécifier que les deux sont nécessaires pour clôturer le nœud, comme dans l'exemple suivant.

```
# pcs stonith create apc1 fence_apc_snmp ipaddr=apc1.example.com login=user
passwd='7a4D#1j!pz864' pcmk_host_map="node1.example.com:1;node2.example.com:2"

# pcs stonith create apc2 fence_apc_snmp ipaddr=apc2.example.com login=user
passwd='7a4D#1j!pz864' pcmk_host_map="node1.example.com:1;node2.example.com:2"

# pcs stonith level add 1 node1.example.com apc1,apc2
# pcs stonith level add 1 node2.example.com apc1,apc2
```

10.7. AFFICHAGE DES DISPOSITIFS DE CLÔTURE CONFIGURÉS

La commande suivante affiche tous les périphériques de clôture actuellement configurés. Si l'option *stonith_id* est spécifiée, la commande affiche uniquement les options de ce dispositif stonith configuré. Si l'option **--full** est spécifiée, toutes les options stonith configurées sont affichées.

```
pcs stonith config [stonith_id] [--full]
```

10.8. EXPORTATION DES DISPOSITIFS DE CLÔTURE SOUS FORME DE COMMANDES PCS

Depuis Red Hat Enterprise Linux 9.1, vous pouvez afficher les commandes **pcs** qui peuvent être utilisées pour recréer les périphériques de clôture configurés sur un système différent en utilisant l'option **--output-format=cmd** de la commande **pcs stonith config**.

Les commandes suivantes créent un périphérique de clôture **fence_apc_snmp** et affichent la commande **pcs** que vous pouvez utiliser pour recréer le périphérique.

```
# pcs stonith create myapc fence_apc_snmp ip="zapc.example.com"
pcmk_host_map="z1.example.com:1;z2.example.com:2" username="apc" password="apc"
# pcs stonith config --output-format=cmd
Warning: Only 'text' output format is supported for stonith levels
pcs stonith create --no-default-ops --force -- myapc fence_apc_snmp \
  ip=zapc.example.com password=apc 'pcmk_host_map=z1.example.com:1;z2.example.com:2'
username=apc \
  op \
    monitor interval=60s id=myapc-monitor-interval-60s
```

10.9. MODIFICATION ET SUPPRESSION DES DISPOSITIFS DE CLÔTURE

Modifiez ou ajoutez des options à un dispositif de clôture actuellement configuré à l'aide de la commande suivante.

```
pcs stonith update stonith_id [stonith_device_options]
```

La mise à jour d'un périphérique de clôture SCSI à l'aide de la commande **pcs stonith update** entraîne le redémarrage de toutes les ressources s'exécutant sur le même nœud que la ressource stonith. Vous pouvez utiliser l'une ou l'autre version de la commande suivante pour mettre à jour les périphériques SCSI sans provoquer le redémarrage des autres ressources du cluster. À partir de RHEL 9.1, les périphériques de clôture SCSI peuvent être configurés en tant que périphériques à chemins multiples.

```
pcs stonith update-scsi-devices stonith_id set device-path1 device-path2
pcs stonith update-scsi-devices stonith_id add device-path1 remove device-path2
```

La commande suivante permet de supprimer un dispositif de clôture de la configuration actuelle.

```
pcs stonith delete stonith_id
```

10.10. CLÔTURE MANUELLE D'UN NŒUD DE CLUSTER

Vous pouvez clôturer un nœud manuellement avec la commande suivante. Si vous spécifiez **--off**, vous utiliserez l'appel de l'API **off** à stonith, qui éteindra le nœud au lieu de le redémarrer.

```
pcs stonith fence node [--off]
```

Si aucun dispositif de clôture n'est en mesure de clôturer un nœud même s'il n'est plus actif, il se peut que le cluster ne puisse pas récupérer les ressources sur le nœud. Dans ce cas, après vous être assuré manuellement que le nœud est hors tension, vous pouvez entrer la commande suivante pour confirmer au cluster que le nœud est hors tension et libérer ses ressources pour la récupération.

AVERTISSEMENT

Si le nœud spécifié n'est pas réellement éteint, mais qu'il exécute le logiciel ou les services normalement contrôlés par la grappe, une corruption des données ou une défaillance de la grappe se produira.

```
pcs stonith confirmer node
```

10.11. DÉSACTIVATION D'UN DISPOSITIF DE CLÔTURE

Pour désactiver un dispositif/une ressource de clôture, exécutez la commande **pcs stonith disable**.

La commande suivante désactive le périphérique de clôture **myapc**.

```
# pcs stonith disable myapc
```

10.12. EMPÊCHER UN NŒUD D'UTILISER UN DISPOSITIF DE CLÔTURE

Pour empêcher un nœud spécifique d'utiliser un dispositif de clôture, vous pouvez configurer des contraintes d'emplacement pour la ressource de clôture.

L'exemple suivant empêche le dispositif de clôture **node1-ipmi** de fonctionner sur **node1**.

```
# pcs constraint location node1-ipmi avoids node1
```

10.13. CONFIGURATION DE L'ACPI POUR UNE UTILISATION AVEC DES PÉRIPHÉRIQUES DE CLÔTURE INTÉGRÉS

Si votre cluster utilise des dispositifs de clôture intégrés, vous devez configurer l'ACPI (Advanced Configuration and Power Interface) pour garantir une clôture immédiate et complète.

Si un nœud de cluster est configuré pour être clôturé par un dispositif de clôture intégré, désactivez ACPI Soft-Off pour ce nœud. La désactivation de l'arrêt progressif de l'ACPI permet à un dispositif de clôture intégré d'éteindre un nœud immédiatement et complètement plutôt que de tenter un arrêt propre (par exemple, **shutdown -h now**). Dans le cas contraire, si l'option ACPI Soft-Off est activée, un dispositif de clôture intégré peut prendre quatre secondes ou plus pour éteindre un nœud (voir la note qui suit). En outre, si l'option ACPI Soft-Off est activée et qu'un nœud panique ou se fige pendant l'arrêt, un dispositif de clôture intégré peut ne pas être en mesure d'éteindre le nœud. Dans ces circonstances, la clôture est retardée ou échoue. Par conséquent, lorsqu'un nœud est clôturé à l'aide d'un dispositif de clôture intégré et que l'option ACPI Soft-Off est activée, la grappe se rétablit lentement ou nécessite une intervention administrative pour se rétablir.

NOTE

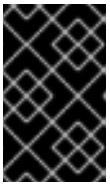
Le temps nécessaire pour clôturer un nœud dépend du dispositif de clôture intégré utilisé. Certains dispositifs de clôture intégrés effectuent l'équivalent d'une pression sur le bouton d'alimentation et le maintiennent enfoncé ; par conséquent, le dispositif de clôture éteint le nœud en quatre à cinq secondes. D'autres dispositifs de clôture intégrés effectuent l'équivalent d'une pression momentanée sur le bouton d'alimentation, en se fiant au système d'exploitation pour éteindre le nœud ; par conséquent, le dispositif de clôture éteint le nœud dans un laps de temps beaucoup plus long que quatre à cinq secondes.

- La meilleure façon de désactiver l'arrêt progressif de l'ACPI est de modifier le réglage du BIOS en " arrêt instantané " ou un réglage équivalent qui éteint le nœud sans délai, comme décrit dans la section " Désactivation de l'arrêt progressif de l'ACPI à l'aide du BIOS " ci-dessous.

La désactivation de l'ACPI Soft-Off avec le BIOS peut ne pas être possible avec certains systèmes. Si la désactivation de l'ACPI Soft-Off avec le BIOS n'est pas satisfaisante pour votre cluster, vous pouvez désactiver l'ACPI Soft-Off avec l'une des méthodes alternatives suivantes :

- Définir **HandlePowerKey=ignore** dans le fichier **/etc/systemd/logind.conf** et vérifier que le nœud s'éteint immédiatement lorsqu'il est clôturé, comme décrit dans " Désactivation de l'arrêt progressif de l'ACPI dans le fichier logind.conf ", ci-dessous. Il s'agit de la première méthode alternative pour désactiver le Soft-Off de l'ACPI.

- En ajoutant **acpi=off** à la ligne de commande de démarrage du noyau, comme décrit dans "Désactiver complètement l'ACPI dans le fichier GRUB 2", ci-dessous. Il s'agit de la deuxième méthode alternative pour désactiver l'ACPI Soft-Off, si la méthode préférée ou la première méthode alternative n'est pas disponible.

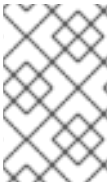


IMPORTANT

Cette méthode désactive complètement l'ACPI ; certains ordinateurs ne démarrent pas correctement si l'ACPI est complètement désactivé. Utilisez cette méthode *only* si les autres méthodes ne sont pas efficaces pour votre cluster.

10.13.1. Désactivation de l'arrêt progressif de l'ACPI par le BIOS

Vous pouvez désactiver l'arrêt progressif de l'ACPI en configurant le BIOS de chaque nœud de cluster à l'aide de la procédure suivante.



NOTE

La procédure de désactivation de l'ACPI Soft-Off par le BIOS peut varier d'un système de serveur à l'autre. Vous devez vérifier cette procédure à l'aide de la documentation de votre matériel.

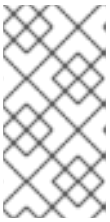
Procédure

1. Redémarrez le nœud et lancez le programme **BIOS CMOS Setup Utility**.

2. Naviguez jusqu'au menu Alimentation (ou un menu équivalent de gestion de l'alimentation).

3. Dans le menu d'alimentation, réglez la fonction **Soft-Off by PWR-BTTN** (ou un paramètre équivalent) sur **Instant-Off** (ou un paramètre équivalent qui éteint le nœud en appuyant sur le bouton d'alimentation sans délai). L'exemple **BIOS CMOS Setup Utiliy** ci-dessous montre un

menu d'alimentation dans lequel **ACPI Function** est réglé sur **Enabled** et **Soft-Off by PWR-BTTN** sur **Instant-Off**.



NOTE

Les équivalents de **ACPI Function**, **Soft-Off by PWR-BTTN**, et **Instant-Off** peuvent varier d'un ordinateur à l'autre. Toutefois, l'objectif de cette procédure est de configurer le BIOS de manière à ce que l'ordinateur soit mis hors tension sans délai à l'aide du bouton d'alimentation.

4. Quitter le programme **BIOS CMOS Setup Utility**, en sauvegardant la configuration du BIOS.

5. Vérifiez que le nœud s'éteint immédiatement lorsqu'il est clôturé. Pour plus d'informations sur le test d'un dispositif de clôture, voir Test d'un dispositif de clôture .

BIOS CMOS Setup Utility:

```
`Soft-Off by PWR-BTTN` set to
`Instant-Off`
```

```
+---------------------------------------------|-------------------+
|   ACPI Function             [Enabled]       |    Item Help      |
|   ACPI Suspend Type         [S1(POS)]       |-------------------|
| x Run VGABIOS if S3 Resume   Auto           |   Menu Level   *  |
|   Suspend Mode              [Disabled]      |                   |
|   HDD Power Down            [Disabled]      |                   |
|   Soft-Off by PWR-BTTN      [Instant-Off    |                   |
|   CPU THRM-Throttling       [50.0%]         |                   |
|   Wake-Up by PCI card       [Enabled]       |                   |
|   Power On by Ring          [Enabled]       |                   |
|   Wake Up On LAN            [Enabled]       |                   |
| x USB KB Wake-Up From S3     Disabled       |                   |
|   Resume by Alarm           [Disabled]      |                   |
| x  Date(of Month) Alarm       0             |                   |
| x  Time(hh:mm:ss) Alarm       0 :  0 :      |                   |
|   POWER ON Function         [BUTTON ONLY    |                   |
| x KB Power ON Password       Enter          |                   |
| x Hot Key Power ON           Ctrl-F1        |                   |
|                                             |                   |
|                                             |                   |
+---------------------------------------------|-------------------+
```

Cet exemple montre que **ACPI Function** est réglé sur **Enabled**, et **Soft-Off by PWR-BTTN** sur **Instant-Off**.

10.13.2. Désactivation de l'arrêt progressif de l'ACPI dans le fichier logind.conf

Pour désactiver la gestion des touches de fonction dans le fichier **/etc/systemd/logind.conf**, procédez comme suit.

Procédure

1. Définissez la configuration suivante dans le fichier **/etc/systemd/logind.conf**:

```
HandlePowerKey=ignore
```

2. Redémarrez le service **systemd-logind**:

```
# systemctl restart systemd-logind.service
```

3. Vérifiez que le nœud s'éteint immédiatement lorsqu'il est clôturé. Pour plus d'informations sur le test d'un dispositif de clôture, voir Test d'un dispositif de clôture .

10.13.3. Désactivation complète de l'ACPI dans le fichier GRUB 2

Vous pouvez désactiver l'arrêt progressif de l'ACPI en ajoutant **acpi=off** à l'entrée du menu GRUB pour un noyau.

IMPORTANT

Cette méthode désactive complètement l'ACPI ; certains ordinateurs ne démarrent pas correctement si l'ACPI est complètement désactivé. Utilisez cette méthode *only* si les autres méthodes ne sont pas efficaces pour votre cluster.

Procédure

Utilisez la procédure suivante pour désactiver l'ACPI dans le fichier GRUB 2 :

1. Utilisez l'option **--args** en combinaison avec l'option **--update-kernel** de l'outil **grubby** pour modifier le fichier **grub.cfg** de chaque nœud de cluster comme suit :

```
# grubby --args=acpi=off --update-kernel=ALL
```

2. Redémarrez le nœud.

3. Vérifiez que le nœud s'éteint immédiatement lorsqu'il est clôturé. Pour plus d'informations sur le test d'un dispositif de clôture, voir Test d'un dispositif de clôture .

CHAPITRE 11. CONFIGURATION DES RESSOURCES DU CLUSTER

Créez et supprimez des ressources de cluster à l'aide des commandes suivantes.

Le format de la commande de création d'une ressource de cluster est le suivant :

> pcs resource create *resource_id* [*standard*:[*provider*:]]*type* [*resource_options*] [op *operation_action operation_options* [*operation_action operation options*]...] [meta *meta_options*...] [clone [*clone_id*] [*clone_options*] | promotable [*clone_id*] [*clone_options*] [--wait[=*n*]]]

Les principales options de création de ressources de cluster sont les suivantes :

- Les options **--before** et **--after** précisent la position de la ressource ajoutée par rapport à une ressource qui existe déjà dans un groupe de ressources.
- La spécification de l'option **--disabled** indique que la ressource n'est pas démarrée automatiquement.

Il n'y a pas de limite au nombre de ressources que vous pouvez créer dans un cluster.

Vous pouvez déterminer le comportement d'une ressource dans un cluster en configurant des contraintes pour cette ressource.

Exemples de création de ressources

La commande suivante crée une ressource portant le nom **VirtualIP**, de norme **ocf**, de fournisseur **heartbeat** et de type **IPaddr2**. L'adresse flottante de cette ressource est 192.168.0.120, et le système vérifie si la ressource fonctionne toutes les 30 secondes.

> **# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.120 cidr_netmask=24 op monitor interval=30s**

Vous pouvez également omettre les champs *standard* et *provider* et utiliser la commande suivante. Par défaut, la norme sera **ocf** et le fournisseur **heartbeat**.

> **# pcs resource create VirtualIP IPaddr2 ip=192.168.0.120 cidr_netmask=24 op monitor interval=30s**

Suppression d'une ressource configurée

Supprimez une ressource configurée à l'aide de la commande suivante.

> suppression de ressources pcs *resource_id*

Par exemple, la commande suivante supprime une ressource existante dont l'identifiant est **VirtualIP**.

> **# pcs resource delete VirtualIP**

11.1. IDENTIFIANTS DES AGENTS DE RESSOURCES

Les identifiants que vous définissez pour une ressource indiquent au cluster quel agent utiliser pour la ressource, où trouver cet agent et à quelles normes il se conforme.

Le tableau suivant décrit les propriétés d'un agent de ressource.

Tableau 11.1. Identifiants des agents de ressources

Field	Description
standard	La norme à laquelle l'agent se conforme. Valeurs autorisées et leur signification : * **ocf** - L'adresse *type* spécifiée est le nom d'un fichier exécutable conforme à l'API de l'agent de ressources de l'Open Cluster Framework et situé en dessous de **/usr/lib/ocf/resource.d/*provider*** * **lsb** - L'adresse *type* spécifiée est le nom d'un fichier exécutable conforme aux actions de script d'initialisation de base standard de Linux. Si le type ne spécifie pas de chemin complet, le système le cherchera dans le répertoire **/etc/init.d**. * **systemd** - L'adresse *type* spécifiée est le nom d'une unité **systemd** installée * **service** - Pacemaker recherchera l'adresse *type* spécifiée, d'abord en tant qu'agent **lsb**, puis en tant qu'agent **systemd** * **nagios** - L'adresse *type* spécifiée est le nom d'un fichier exécutable conforme à l'API des plugins de Nagios et situé dans le répertoire **/usr/libexec/nagios/plugins**, avec des métadonnées de style OCF stockées séparément dans le répertoire **/usr/share/nagios/plugins-metadata** (disponibles dans le paquetage **nagios-agents-metadata** pour certains plugins communs).
type	Le nom de l'agent de ressource que vous souhaitez utiliser, par exemple **IPaddr** ou **Filesystem**
fournisseur	La spécification OCF permet à plusieurs fournisseurs de proposer le même agent de ressources. La plupart des agents fournis par Red Hat utilisent **heartbeat** comme fournisseur.

Le tableau suivant résume les commandes qui affichent les propriétés des ressources disponibles.

Tableau 11.2. Commandes pour afficher les propriétés des ressources

pcs Display Command	Sortie
pcs resource list	Affiche une liste de toutes les ressources disponibles.
pcs resource standards	Affiche une liste des normes disponibles pour les agents de ressources.
pcs resource providers	Affiche une liste des fournisseurs d'agents de ressources disponibles.

pcs Display Command	Sortie
pcs resource list *string*	Affiche une liste de ressources disponibles filtrées par la chaîne spécifiée. Vous pouvez utiliser cette commande pour afficher les ressources filtrées par le nom d'une norme, d'un fournisseur ou d'un type.

11.2. AFFICHAGE DES PARAMÈTRES SPÉCIFIQUES AUX RESSOURCES

Pour toute ressource individuelle, vous pouvez utiliser la commande suivante pour afficher une description de la ressource, les paramètres que vous pouvez définir pour cette ressource et les valeurs par défaut définies pour la ressource.

```
pcs resource describe [standard:[provider:]]type
```

Par exemple, la commande suivante affiche des informations sur une ressource de type **apache**.

```
# pcs resource describe ocf:heartbeat:apache
This is the resource agent for the Apache Web server.
This resource agent operates both version 1.x and version 2.x Apache
servers.

...
```

11.3. CONFIGURATION DES MÉTA-OPTIONS DES RESSOURCES

Outre les paramètres spécifiques aux ressources, vous pouvez configurer des options supplémentaires pour chaque ressource. Ces options sont utilisées par le cluster pour décider du comportement de votre ressource.

Le tableau suivant décrit les options de méta ressources.

Tableau 11.3. Options méta des ressources

Field	Défaut	Description
priority	**0**	Si toutes les ressources ne peuvent pas être actives, le cluster arrêtera les ressources moins prioritaires afin de maintenir actives les ressources plus prioritaires.

Field	Défaut	Description
target-role	**Started**	Indique l'état dans lequel le cluster doit tenter de maintenir cette ressource. Valeurs autorisées : * **Stopped** - Forcer l'arrêt de la ressource * **Started** - Autoriser le démarrage de la ressource (et, dans le cas de clones promouvables, sa promotion le cas échéant) * **Promoted** - Permettre le démarrage de la ressource et, le cas échéant, sa promotion * **Unpromoted** - Autoriser le démarrage de la ressource, mais uniquement en mode non promu si la ressource est promouvable
is-managed	**true**	Indique si le cluster est autorisé à démarrer et à arrêter la ressource. Valeurs autorisées : **true**, **false**
resource-stickiness	1	Valeur indiquant dans quelle mesure la ressource préfère rester là où elle se trouve.

Field	Défaut	Description
requires	Calculé	Indique dans quelles conditions la ressource peut être démarrée. La valeur par défaut est **fencing**, sauf dans les conditions indiquées ci-dessous. Valeurs possibles : * **nothing** - Le cluster peut toujours démarrer la ressource. * **quorum** - Le cluster ne peut démarrer cette ressource que si la majorité des nœuds configurés sont actifs. Il s'agit de la valeur par défaut si **stonith-enabled** est **false** ou si **standard** est **stonith**. * **fencing** - Le cluster ne peut démarrer cette ressource que si la majorité des nœuds configurés sont actifs *and*. Les nœuds défaillants ou inconnus ont été clôturés. * **unfencing** - Le cluster ne peut démarrer cette ressource que si la majorité des nœuds configurés sont actifs. *and* Tous les nœuds défaillants ou inconnus ont été clôturés. *and* Uniquement sur les nœuds qui ont été clôturés. *unfenced*. Il s'agit de la valeur par défaut si l'option **provides=unfencing stonith** meta a été définie pour un dispositif de clôture.
migration-threshold	**INFINITY**	Nombre d'échecs pouvant survenir pour cette ressource sur un nœud avant que ce nœud ne soit marqué comme inéligible pour héberger cette ressource. Une valeur de 0 indique que cette fonctionnalité est désactivée (le nœud ne sera jamais marqué comme inéligible) ; par contre, le cluster traite **INFINITY** (la valeur par défaut) comme un nombre très grand mais fini. Cette option n'a d'effet que si l'opération qui a échoué a pour valeur **on-fail=restart** (par défaut), et en outre pour les opérations de démarrage qui ont échoué si la propriété du cluster **start-failure-is-fatal** est **false**.

Field	Défaut	Description
failure-timeout	**0** (désactivé)	Utilisé avec l'option **migration-threshold**, indique le nombre de secondes à attendre avant de faire comme si l'échec ne s'était pas produit et d'autoriser éventuellement le retour de la ressource au nœud sur lequel elle a échoué.
multiple-active	**stop_start**	Indique ce que le cluster doit faire s'il constate que la ressource est active sur plus d'un nœud. Valeurs autorisées : * **block** - marquer la ressource comme non gérée * **stop_only** - arrêter toutes les instances actives et les laisser ainsi * **stop_start** - arrêter toutes les instances actives et démarrer la ressource à un seul endroit * **stop_unexpected** - (RHEL 9.1 et versions ultérieures) n'arrêtent que les instances inattendues de la ressource, sans nécessiter un redémarrage complet. Il est de la responsabilité de l'utilisateur de vérifier que le service et son agent de ressource peuvent fonctionner avec des instances actives supplémentaires sans nécessiter un redémarrage complet.
critical	**true**	Définit la valeur par défaut de l'option **influence** pour toutes les contraintes de colocalisation impliquant la ressource en tant que ressource dépendante (*target_resource*), y compris les contraintes de colocalisation implicites créées lorsque la ressource fait partie d'un groupe de ressources. L'option de contrainte de colocation **influence** détermine si le cluster déplacera les ressources primaires et dépendantes vers un autre nœud lorsque la ressource dépendante atteint son seuil de migration en cas de défaillance, ou si le cluster laissera la ressource dépendante hors ligne sans provoquer de changement de service. L'option **critical** resource meta peut avoir une valeur de **true** ou **false**, avec une valeur par défaut de **true**.

Field	Défaut	Description
allow-unhealthy-nodes	**false**	(RHEL 9.1 et versions ultérieures) Lorsque cet attribut est défini sur **true**, la ressource n'est pas forcée de quitter un nœud en raison de la dégradation de son état de santé. Lorsque cet attribut est défini pour les ressources de santé, la grappe peut automatiquement détecter si l'état de santé du nœud se rétablit et déplacer à nouveau les ressources vers ce nœud. L'état de santé d'un nœud est déterminé par la combinaison des attributs de santé définis par les agents des ressources de santé en fonction des conditions locales et des options stratégiques qui déterminent la manière dont la grappe réagit à ces conditions.

11.3.1. Modifier la valeur par défaut d'une option de ressource

Vous pouvez modifier la valeur par défaut d'une option de ressource pour toutes les ressources à l'aide de la commande **pcs resource defaults update**. La commande suivante réinitialise la valeur par défaut de **resource-stickiness** à 100.

```
# pcs resource defaults update resource-stickiness=100
```

La commande **pcs resource defaults *name=value*** qui définissait les valeurs par défaut pour toutes les ressources dans les versions précédentes, reste prise en charge, à moins que plusieurs valeurs par défaut ne soient configurées. Cependant, **pcs resource defaults update** est désormais la version préférée de la commande.

11.3.2. Modification de la valeur par défaut d'une option de ressource pour des ensembles de ressources

Vous pouvez créer plusieurs ensembles de ressources par défaut à l'aide de la commande **pcs resource defaults set create**, qui vous permet de spécifier une règle contenant des expressions **resource**. Seules les expressions **resource** et **date**, y compris **and**, **or** et les parenthèses, sont autorisées dans les règles que vous spécifiez avec cette commande.

La commande **pcs resource defaults set create** permet de configurer une valeur de ressource par défaut pour toutes les ressources d'un type particulier. Si, par exemple, vous exécutez des bases de données qui mettent longtemps à s'arrêter, vous pouvez augmenter la valeur par défaut de **resource-stickiness** pour toutes les ressources du type base de données afin d'éviter que ces ressources ne soient déplacées vers d'autres nœuds plus souvent que vous ne le souhaitez.

La commande suivante définit la valeur par défaut de **resource-stickiness** à 100 pour toutes les ressources de type **pqsql**.

- L'option **id**, qui désigne l'ensemble des ressources par défaut, n'est pas obligatoire. Si vous ne définissez pas cette option, **pcs** génère automatiquement un identifiant. La définition de cette valeur vous permet de fournir un nom plus descriptif.

- Dans cet exemple, **::pgsql** désigne une ressource de n'importe quelle classe, de n'importe quel fournisseur, de type **pgsql**.
  - La spécification de **ocf:heartbeat:pgsql** indique la classe **ocf**, le fournisseur **heartbeat**, le type **pgsql**,
  - En spécifiant **ocf:pacemaker:**, vous indiquez toutes les ressources de la classe **ocf**, du fournisseur **pacemaker**, quel que soit leur type.

```
# pcs resource defaults set create id=pgsql-stickiness meta resource-stickiness=100 rule resource ::pgsql
```

Pour modifier les valeurs par défaut d'un ensemble existant, utilisez la commande **pcs resource defaults set update**.

11.3.3. Affichage des valeurs par défaut des ressources actuellement configurées

La commande **pcs resource defaults** affiche une liste des valeurs par défaut actuellement configurées pour les options de ressources, y compris les règles que vous avez spécifiées.

L'exemple suivant montre la sortie de cette commande après avoir réinitialisé la valeur par défaut de **resource-stickiness** à 100.

```
# pcs resource defaults
Meta Attrs: rsc_defaults-meta_attributes
  resource-stickiness=100
```

L'exemple suivant montre la sortie de cette commande après avoir réinitialisé la valeur par défaut de **resource-stickiness** à 100 pour toutes les ressources de type **pqsql** et défini l'option **id** à **id=pgsql-stickiness**.

```
# pcs resource defaults
Meta Attrs: pgsql-stickiness
  resource-stickiness=100
  Rule: boolean-op=and score=INFINITY
    Expression: resource ::pgsql
```

11.3.4. Définition des méta-options lors de la création d'une ressource

Que vous ayez ou non réinitialisé la valeur par défaut d'une méta-option de ressource, vous pouvez définir une option de ressource pour une ressource particulière à une valeur différente de la valeur par défaut lorsque vous créez la ressource. Voici le format de la commande **pcs resource create** que vous utilisez pour spécifier une valeur pour une méta-option de ressource.

```
pcs resource create resource_id [standard:[provider:]]type [resource options] [meta meta_options...]
```

Par exemple, la commande suivante crée une ressource avec une valeur **resource-stickiness** de 50.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.120 meta resource-stickiness=50
```

Vous pouvez également définir la valeur d'une méta-option de ressource pour une ressource existante, un groupe ou une ressource clonée à l'aide de la commande suivante.

```
pcs resource meta resource_id | group_id | clone_id meta_options
```

Dans l'exemple suivant, il existe une ressource nommée **dummy_resource**. Cette commande définit l'option **failure-timeout** meta à 20 secondes, de sorte que la ressource puisse tenter de redémarrer sur le même nœud dans 20 secondes.

```
# pcs resource meta dummy_resource failure-timeout=20s
```

Après avoir exécuté cette commande, vous pouvez afficher les valeurs de la ressource pour vérifier que **failure-timeout=20s** est défini.

```
# pcs resource config dummy_resource
 Resource: dummy_resource (class=ocf provider=heartbeat type=Dummy)
  Meta Attrs: failure-timeout=20s
  ...
```

11.4. CONFIGURATION DES GROUPES DE RESSOURCES

L'un des éléments les plus courants d'un cluster est un ensemble de ressources qui doivent être situées ensemble, démarrer de manière séquentielle et s'arrêter dans l'ordre inverse. Pour simplifier cette configuration, Pacemaker prend en charge le concept de groupes de ressources.

11.4.1. Création d'un groupe de ressources

Vous créez un groupe de ressources avec la commande suivante, en spécifiant les ressources à inclure dans le groupe. Si le groupe n'existe pas, cette commande le crée. Si le groupe existe, cette commande ajoute des ressources supplémentaires au groupe. Les ressources démarrent dans l'ordre indiqué par cette commande et s'arrêtent dans l'ordre inverse de leur ordre de départ.

```
pcs resource group add group_name resource_id [resource_id] ... [resource_id] [--avant resource_id | --après resource_id]
```

Vous pouvez utiliser les options **--before** et **--after** de cette commande pour spécifier la position des ressources ajoutées par rapport à une ressource qui existe déjà dans le groupe.

Vous pouvez également ajouter une nouvelle ressource à un groupe existant lorsque vous créez la ressource, en utilisant la commande suivante. La ressource que vous créez est ajoutée au groupe nommé *group_name*. Si le groupe *group_name* n'existe pas, il sera créé.

```
pcs resource create resource_id [standard:[provider:]]type [resource_options] [op operation_action operation_options ] --group group_name
```

Il n'y a pas de limite au nombre de ressources qu'un groupe peut contenir. Les propriétés fondamentales d'un groupe sont les suivantes.

- Les ressources sont regroupées au sein d'un groupe.
- Les ressources sont lancées dans l'ordre dans lequel vous les avez spécifiées. Si une ressource du groupe ne peut s'exécuter nulle part, aucune ressource spécifiée après cette ressource n'est autorisée à s'exécuter.
- Les ressources sont arrêtées dans l'ordre inverse de celui dans lequel vous les avez spécifiées.

L'exemple suivant crée un groupe de ressources nommé **shortcut** qui contient les ressources existantes **IPaddr** et **Email**.

```
# pcs resource group add shortcut IPaddr Email
```

Dans cet exemple :

- Le site **IPaddr** est démarré en premier, puis **Email**.
- La ressource **Email** est arrêtée en premier, puis **IPAddr**.
- Si **IPaddr** ne peut courir nulle part, **Email** ne le peut pas non plus.
- Si **Email** ne peut courir nulle part, cela n'affecte en rien **IPaddr**.

11.4.2. Suppression d'un groupe de ressources

La commande suivante permet de supprimer une ressource d'un groupe. S'il ne reste plus de ressources dans le groupe, cette commande supprime le groupe lui-même.

```
pcs resource group remove group_name resource_id ...
```

11.4.3. Affichage des groupes de ressources

La commande suivante répertorie tous les groupes de ressources actuellement configurés.

```
liste des groupes de ressources pcs
```

11.4.4. Options du groupe

Vous pouvez définir les options suivantes pour un groupe de ressources. Elles ont la même signification que lorsqu'elles sont définies pour une ressource unique : **priority**, **target-role**, **is-managed**. Pour plus d'informations sur les méta-options des ressources, voir Configuration des méta-options des ressources.

11.4.5. Adhésion au groupe

L'adhérence, la mesure de la volonté d'une ressource de rester là où elle est, est additive dans les groupes. Chaque ressource active du groupe apportera sa valeur d'adhérence au total du groupe. Ainsi, si la valeur par défaut de **resource-stickiness** est de 100 et qu'un groupe compte sept membres, dont cinq sont actifs, le groupe dans son ensemble préférera son emplacement actuel avec un score de 500.

11.5. DÉTERMINER LE COMPORTEMENT DES RESSOURCES

Vous pouvez déterminer le comportement d'une ressource dans un cluster en configurant des contraintes pour cette ressource. Vous pouvez configurer les catégories de contraintes suivantes :

- **location** contraintes - Une contrainte d'emplacement détermine les nœuds sur lesquels une ressource peut s'exécuter. Pour plus d'informations sur la configuration des contraintes d'emplacement, voir Détermination des nœuds sur lesquels une ressource peut s'exécuter .

- **order** contraintes - Une contrainte d'ordre détermine l'ordre d'exécution des ressources. Pour plus d'informations sur la configuration des contraintes d'ordre, voir Détermination de l'ordre d'exécution des ressources de la grappe.

- **colocation** contraintes - Une contrainte de colocalisation détermine l'emplacement des ressources par rapport à d'autres ressources. Pour plus d'informations sur les contraintes de colocalisation, voir Colocalisation des ressources du cluster.

En guise d'abréviation pour configurer un ensemble de contraintes qui localiseront un ensemble de ressources ensemble et garantiront que les ressources démarrent de manière séquentielle et s'arrêtent dans l'ordre inverse, Pacemaker prend en charge le concept de groupes de ressources. Après avoir créé un groupe de ressources, vous pouvez configurer des contraintes sur le groupe lui-même de la même manière que vous configurez des contraintes pour des ressources individuelles.

CHAPITRE 12. DÉTERMINER LES NŒUDS SUR LESQUELS UNE RESSOURCE PEUT S'EXÉCUTER

Les contraintes d'emplacement déterminent les nœuds sur lesquels une ressource peut s'exécuter. Vous pouvez configurer les contraintes d'emplacement pour déterminer si une ressource préférera ou évitera un nœud spécifique.

Outre les contraintes d'emplacement, le nœud sur lequel une ressource s'exécute est influencé par la valeur **resource-stickiness** de cette ressource, qui détermine dans quelle mesure une ressource préfère rester sur le nœud sur lequel elle s'exécute actuellement. Pour plus d'informations sur la définition de la valeur **resource-stickiness**, voir Configurer une ressource pour qu'elle préfère son nœud actuel.

12.1. CONFIGURATION DES CONTRAINTES DE LOCALISATION

Vous pouvez configurer une contrainte de localisation de base pour spécifier si une ressource préfère ou évite un nœud, avec une valeur facultative **score** pour indiquer le degré relatif de préférence pour la contrainte.

La commande suivante crée une contrainte d'emplacement pour une ressource afin de privilégier le ou les nœuds spécifiés. Notez qu'il est possible de créer des contraintes sur une ressource particulière pour plus d'un nœud avec une seule commande.

```
pcs constraint location rsc prefers node[=score] [node[=score]] ...
```

La commande suivante crée une contrainte de localisation pour une ressource afin d'éviter le ou les nœuds spécifiés.

```
pcs constraint location rsc évite node[=score] [node[=score]] ...
```

Le tableau suivant résume la signification des options de base pour la configuration des contraintes de localisation.

Tableau 12.1. Options de contrainte de localisation

Field	Description
rsc	Un nom de ressource
node	Nom d'un nœud

Field	Description
score	Valeur entière positive indiquant le degré de préférence selon lequel la ressource donnée doit préférer ou éviter le nœud donné. **INFINITY** est la valeur par défaut de **score** pour une contrainte de localisation de la ressource. Une valeur de **INFINITY** pour **score** dans une commande **pcs constraint location *rsc* prefers** indique que la ressource préférera ce nœud s'il est disponible, mais n'empêche pas la ressource de fonctionner sur un autre nœud si le nœud spécifié n'est pas disponible. Une valeur de **INFINITY** pour **score** dans une commande indique que la ressource ne sera jamais exécutée sur ce nœud, même si aucun autre nœud n'est disponible **pcs constraint location *rsc* avoids** indique que la ressource ne sera jamais exécutée sur ce nœud, même si aucun autre nœud n'est disponible. Cela équivaut à définir une commande **pcs constraint location add** avec un score de**-INFINITY**. Un score numérique (c'est-à-dire non **INFINITY**) signifie que la contrainte est facultative et qu'elle sera respectée à moins qu'un autre facteur ne l'emporte. Par exemple, si la ressource est déjà placée sur un autre nœud et que son score **resource-stickiness** est supérieur au score d'une contrainte de localisation **prefers**, la ressource sera laissée là où elle est.

La commande suivante crée une contrainte de localisation pour spécifier que la ressource **Webserver** préfère le nœud **node1**.

```
# pcs constraint location Webserver prefers node1
```

pcs prend en charge les expressions régulières dans les contraintes d'emplacement sur la ligne de commande. Ces contraintes s'appliquent à plusieurs ressources en fonction de l'expression régulière correspondant au nom de la ressource. Cela vous permet de configurer plusieurs contraintes d'emplacement à l'aide d'une seule ligne de commande.

La commande suivante crée une contrainte de localisation pour spécifier que les ressources **dummy0** à **dummy9** préfèrent **node1**.

```
# pcs constraint location 'regexp\rmmy[0-9]' prefers node1
```

Comme Pacemaker utilise des expressions régulières étendues POSIX telles que documentées à l'adresse suivante http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap09.html#tag_09_04vous pouvez spécifier la même contrainte avec la commande suivante.

```
# pcs constraint location 'regexp\rmmy[[:digit:]]' prefers node1
```

12.2. LIMITER LA DÉCOUVERTE DES RESSOURCES À UN SOUS-ENSEMBLE DE NŒUDS

Avant que Pacemaker ne démarre une ressource n'importe où, il exécute d'abord une opération de surveillance unique (souvent appelée "sonde") sur chaque nœud, pour savoir si la ressource est déjà en cours d'exécution. Ce processus de découverte des ressources peut entraîner des erreurs sur les nœuds qui ne sont pas en mesure d'exécuter le moniteur.

Lors de la configuration d'une contrainte d'emplacement sur un nœud, vous pouvez utiliser l'option **resource-discovery** de la commande **pcs constraint location** pour indiquer si Pacemaker doit effectuer la recherche de ressources sur ce nœud pour la ressource spécifiée. Limiter la découverte des ressources à un sous-ensemble de nœuds sur lesquels la ressource est physiquement capable de fonctionner peut améliorer considérablement les performances lorsqu'un grand nombre de nœuds est présent. Lorsque **pacemaker_remote** est utilisé pour étendre le nombre de nœuds à plusieurs centaines, cette option doit être envisagée.

La commande suivante montre le format pour spécifier l'option **resource-discovery** de la commande **pcs constraint location**. Dans cette commande, une valeur positive pour *score* correspond à une contrainte d'emplacement de base qui configure une ressource pour qu'elle préfère un nœud, tandis qu'une valeur négative pour *score* correspond à une contrainte d'emplacement de base qui configure une ressource pour qu'elle évite un nœud. Comme pour les contraintes d'emplacement de base, vous pouvez également utiliser des expressions régulières pour les ressources soumises à ces contraintes.

```
pcs constraint location add id rsc node score [resource-discovery=option]
```

Le tableau suivant résume la signification des paramètres de base permettant de configurer les contraintes pour la découverte des ressources.

Tableau 12.2. Paramètres de contrainte de découverte de ressources

Field	Description
id	Nom choisi par l'utilisateur pour la contrainte elle-même.
rsc	Un nom de ressource
node	Nom d'un nœud

score	Valeur entière indiquant le degré de préférence de la ressource donnée pour le nœud donné ou son évitement. Une valeur positive pour le score correspond à une contrainte de localisation de base qui configure une ressource pour qu'elle préfère un nœud, tandis qu'une valeur négative pour le score correspond à une contrainte de localisation de base qui configure une ressource pour qu'elle évite un nœud. Une valeur de **INFINITY** pour **score** indique que la ressource préférera ce nœud si le nœud est disponible, mais n'empêche pas la ressource de s'exécuter sur un autre nœud si le nœud spécifié n'est pas disponible. Une valeur de **-INFINITY** pour **score** indique que la ressource ne s'exécutera jamais sur ce nœud, même si aucun autre nœud n'est disponible. Un score numérique (c'est-à-dire pas **INFINITY** ou **-INFINITY**) signifie que la contrainte est facultative et qu'elle sera respectée à moins qu'un autre facteur ne l'emporte. Par exemple, si la ressource est déjà placée sur un autre nœud et que son score **resource-stickiness** est supérieur au score d'une contrainte de localisation **prefers**, la ressource sera laissée là où elle est.
resource-discovery options	* **always** - Toujours effectuer la découverte de la ressource spécifiée sur ce nœud. Il s'agit de la valeur par défaut de **resource-discovery** pour une contrainte d'emplacement de ressource. * **never** - Ne jamais effectuer de recherche de ressources pour la ressource spécifiée sur ce nœud. * **exclusive** - Effectuer la recherche de la ressource spécifiée uniquement sur ce nœud (et sur d'autres nœuds marqués de la même manière comme **exclusive**). Les contraintes d'emplacement multiples utilisant la découverte **exclusive** pour la même ressource sur différents nœuds créent un sous-ensemble de nœuds auquel **resource-discovery** est exclusif. Si une ressource est marquée pour la recherche **exclusive** sur un ou plusieurs nœuds, cette ressource ne peut être placée que dans ce sous-ensemble de nœuds.

AVERTISSEMENT

En définissant **resource-discovery** comme **never** ou **exclusive**, Pacemaker n'est plus en mesure de détecter et d'arrêter les instances indésirables d'un service qui s'exécute là où il n'est pas censé être. C'est à l'administrateur du système de s'assurer que le service ne peut jamais être actif sur des nœuds sans découverte de ressources (par exemple en laissant le logiciel concerné désinstallé).

12.3. CONFIGURATION D'UNE STRATÉGIE DE CONTRAINTE DE LOCALISATION

Lorsque vous utilisez des contraintes d'emplacement, vous pouvez configurer une stratégie générale pour spécifier les nœuds sur lesquels une ressource peut s'exécuter :

- Clusters opt-in - Configurez un cluster dans lequel, par défaut, aucune ressource ne peut s'exécuter nulle part, puis activez sélectivement les nœuds autorisés pour des ressources spécifiques.
- Clusters opt-out - Configurez un cluster dans lequel, par défaut, toutes les ressources peuvent s'exécuter n'importe où, puis créez des contraintes d'emplacement pour les ressources qui ne sont pas autorisées à s'exécuter sur des nœuds spécifiques.

Le choix d'une configuration opt-in ou opt-out dépend à la fois de vos préférences personnelles et de la composition de votre cluster. Si la plupart de vos ressources peuvent être exécutées sur la plupart des nœuds, une configuration opt-out sera probablement plus simple. En revanche, si la plupart des ressources ne peuvent être exécutées que sur un petit sous-ensemble de nœuds, une configuration opt-in peut s'avérer plus simple.

12.3.1. Configuration d'un cluster "opt-in

Pour créer un cluster opt-in, définissez la propriété du cluster **symmetric-cluster** sur **false** afin d'empêcher les ressources de s'exécuter n'importe où par défaut.

```
# pcs property set symmetric-cluster=false
```

Activer les nœuds pour les ressources individuelles. Les commandes suivantes configurent les contraintes d'emplacement de sorte que la ressource **Webserver** préfère le nœud **example-1**, la ressource **Database** préfère le nœud **example-2** et les deux ressources peuvent basculer vers le nœud **example-3** en cas d'échec de leur nœud préféré. Lors de la configuration des contraintes d'emplacement pour un cluster opt-in, la définition d'un score de zéro permet à une ressource de s'exécuter sur un nœud sans indiquer de préférence pour ce nœud ou pour l'éviter.

```
# pcs constraint location Webserver prefers example-1=200
# pcs constraint location Webserver prefers example-3=0
# pcs constraint location Database prefers example-2=200
# pcs constraint location Database prefers example-3=0
```

12.3.2. Configuration d'un cluster "Opt-Out

Pour créer un cluster opt-out, définissez la propriété du cluster **symmetric-cluster** sur **true** pour permettre aux ressources de s'exécuter partout par défaut. Il s'agit de la configuration par défaut si **symmetric-cluster** n'est pas défini explicitement.

```
# pcs property set symmetric-cluster=true
```

Les commandes suivantes permettent d'obtenir une configuration équivalente à l'exemple présenté dans la section " Configuration d'un cluster Opt-In ". Les deux ressources peuvent basculer vers le nœud **example-3** si leur nœud préféré tombe en panne, puisque chaque nœud a un score implicite de 0.

```
# pcs constraint location Webserver prefers example-1=200
# pcs constraint location Webserver avoids example-2=INFINITY
```

```
# pcs constraint location Database avoids example-1=INFINITY
# pcs constraint location Database prefers example-2=200
```

Notez qu'il n'est pas nécessaire de spécifier un score INFINITE dans ces commandes, puisque c'est la valeur par défaut du score.

12.4. CONFIGURER UNE RESSOURCE POUR QU'ELLE PRÉFÈRE SON NŒUD ACTUEL

Les ressources ont une valeur **resource-stickiness** que vous pouvez définir en tant qu'attribut méta lorsque vous créez la ressource, comme décrit dans Configuration des options méta des ressources. La valeur **resource-stickiness** détermine dans quelle mesure une ressource souhaite rester sur le nœud où elle s'exécute actuellement. Pacemaker prend en compte la valeur **resource-stickiness** en conjonction avec d'autres paramètres (par exemple, les valeurs de score des contraintes d'emplacement) pour déterminer s'il convient de déplacer une ressource vers un autre nœud ou de la laisser en place.

Avec une valeur **resource-stickiness** de 0, une grappe peut déplacer des ressources si nécessaire pour équilibrer les ressources entre les nœuds. Il peut en résulter un déplacement des ressources lorsque des ressources non apparentées démarrent ou s'arrêtent. Avec une valeur positive, les ressources préfèrent rester là où elles sont et ne se déplacent que si d'autres circonstances l'emportent. Cela peut avoir pour conséquence que les nœuds nouvellement ajoutés ne se voient pas attribuer de ressources sans l'intervention de l'administrateur.

Les grappes nouvellement créées dans RHEL 9 définissent la valeur par défaut de **resource-stickiness** sur 1. Cette petite valeur peut facilement être remplacée par d'autres contraintes que vous créez, mais elle est suffisante pour empêcher Pacemaker de déplacer inutilement des ressources saines au sein de la grappe. Si vous préférez le comportement du cluster résultant d'une valeur **resource-stickiness** de 0, vous pouvez modifier la valeur par défaut de **resource-stickiness** en 0 à l'aide de la commande suivante :

```
# pcs resource defaults update resource-stickiness=0
```

Si vous mettez à niveau un cluster existant vers RHEL 9 et que vous n'avez pas explicitement défini une valeur par défaut pour **resource-stickiness**, la valeur **resource-stickiness** reste 0 et la commande **pcs resource defaults** n'indiquera rien pour l'adhérence.

Si la valeur de **resource-stickiness** est positive, aucune ressource ne sera déplacée vers un nœud nouvellement ajouté. Si vous souhaitez équilibrer les ressources à ce moment-là, vous pouvez temporairement fixer la valeur **resource-stickiness** à 0.

Il convient de noter que si le score d'une contrainte de localisation est supérieur à la valeur **resource-stickiness**, le cluster peut toujours déplacer une ressource saine vers le nœud où pointe la contrainte de localisation.

Pour plus d'informations sur la manière dont Pacemaker détermine où placer une ressource, voir Configuration d'une stratégie de placement de nœuds .

CHAPITRE 13. DÉTERMINER L'ORDRE D'EXÉCUTION DES RESSOURCES DE LA GRAPPE

Pour déterminer l'ordre d'exécution des ressources, vous devez configurer une contrainte d'ordre.

Le tableau suivant présente le format de la commande permettant de configurer une contrainte d'ordre.

```
pcs constraint order [action] resource_id then [action] resource_id [options]
```

Le tableau suivant résume les propriétés et les options permettant de configurer les contraintes d'ordre.

Tableau 13.1. Propriétés d'une contrainte de commande

Field	Description
id_ressource	Le nom d'une ressource sur laquelle une action est effectuée.
action	L'action à ordonner sur la ressource. Les valeurs possibles de la propriété *action* sont les suivantes : * **start** - Ordre des actions de démarrage de la ressource. * **stop** - Ordonner l'arrêt des actions de la ressource. * **promote** - Promouvoir la ressource d'une ressource non promue à une ressource promue. * **demote** - Rétrograder la ressource d'une ressource promue à une ressource non promue. Si aucune action n'est spécifiée, l'action par défaut est **start**.
kind option	Comment appliquer la contrainte. Les valeurs possibles de l'option **kind** sont les suivantes : * **Optional** - Ne s'applique que si les deux ressources exécutent l'action spécifiée. Pour plus d'informations sur l'ordonnancement facultatif, voir Configuration de l'ordonnancement consultatif. * **Mandatory** - Toujours appliquer la contrainte (valeur par défaut). Si la première ressource spécifiée s'arrête ou ne peut pas être démarrée, la deuxième ressource spécifiée doit être arrêtée. Pour plus d'informations sur l'ordonnancement obligatoire, voir Configuration de l'ordonnancement obligatoire. * **Serialize** - Veillez à ce qu'il n'y ait pas deux actions d'arrêt/démarrage simultanées pour les ressources que vous spécifiez. La première et la deuxième ressource que vous spécifiez peuvent démarrer dans n'importe quel ordre, mais l'une d'entre elles doit avoir terminé son démarrage avant que l'autre ne puisse être démarrée. Un cas d'utilisation typique est celui où le démarrage d'une ressource impose une charge élevée à l'hôte.

Field	Description
symmetrical option	Si elle est vraie, l'inverse de la contrainte s'applique à l'action opposée (par exemple, si B commence après A, alors B s'arrête avant A). Les contraintes d'ordre pour lesquelles **kind** est **Serialize** ne peuvent pas être symétriques. La valeur par défaut est **true** pour les types **Mandatory** et **Optional**, **false** pour **Serialize**.

Utilisez la commande suivante pour supprimer des ressources de toute contrainte de classement.

```
pcs constraint order remove resource1 [resourceN]...
```

13.1. CONFIGURATION DE LA COMMANDE OBLIGATOIRE

Une contrainte d'ordre obligatoire indique que la deuxième action ne doit pas être lancée pour la deuxième ressource tant que la première action ne s'est pas achevée avec succès pour la première ressource. Les actions qui peuvent être ordonnées sont **stop**, **start** et, en outre, pour les clones promouvables, **demote** et **promote**. Par exemple, \N "A puis B\N" (qui est équivalent à \N "démarrer A puis démarrer B\N") signifie que B ne sera pas démarré à moins et jusqu'à ce que A démarre avec succès. Une contrainte d'ordre est obligatoire si l'option **kind** de la contrainte est définie sur **Mandatory** ou laissée par défaut.

Si l'option **symmetrical** est définie sur **true** ou laissée par défaut, les actions opposées seront ordonnées en sens inverse. Les actions **start** et **stop** sont opposées, et **demote** et **promote** sont opposées. Par exemple, un ordre symétrique "promouvoir A puis démarrer B" implique "arrêter B puis rétrograder A", ce qui signifie que A ne peut pas être rétrogradé tant que B n'a pas réussi à s'arrêter. Un ordre symétrique signifie que les changements dans l'état de A peuvent entraîner la programmation d'actions pour B. Par exemple, si A redémarre en raison d'une défaillance, B sera arrêté en premier, puis A sera arrêté, puis A sera démarré, puis B sera démarré.

Notez que le cluster réagit à chaque changement d'état. Si la première ressource est redémarrée et se trouve à nouveau dans un état de démarrage avant que la seconde ressource ne lance une opération d'arrêt, la seconde ressource n'aura pas besoin d'être redémarrée.

13.2. CONFIGURATION DE LA COMMANDE CONSULTATIVE

Lorsque l'option **kind=Optional** est spécifiée pour une contrainte d'ordre, la contrainte est considérée comme facultative et ne s'applique que si les deux ressources exécutent les actions spécifiées. Tout changement d'état de la première ressource spécifiée n'aura aucun effet sur la deuxième ressource spécifiée.

La commande suivante configure une contrainte d'ordre consultatif pour les ressources nommées **VirtualIP** et **dummy_resource**.

```
# pcs constraint order VirtualIP then dummy_resource kind=Optional
```

13.3. CONFIGURATION DES ENSEMBLES DE RESSOURCES ORDONNÉS

Il est fréquent qu'un administrateur crée une chaîne de ressources ordonnées, où, par exemple, la ressource A démarre avant la ressource B, qui démarre avant la ressource C. Si votre configuration exige que vous créiez un ensemble de ressources colocalisées et démarrées dans l'ordre, vous pouvez configurer un groupe de ressources qui contient ces ressources.

Toutefois, dans certaines situations, il n'est pas approprié de configurer les ressources qui doivent démarrer dans un ordre précis en tant que groupe de ressources :

- Il se peut que vous deviez configurer les ressources pour qu'elles démarrent dans l'ordre et qu'elles ne soient pas nécessairement colocalisées.
- Vous pouvez avoir une ressource C qui doit démarrer après que la ressource A ou B a démarré, mais il n'y a pas de relation entre A et B.
- Vous pouvez avoir des ressources C et D qui doivent démarrer après que les ressources A et B ont démarré, mais il n'y a pas de relation entre A et B ou entre C et D.

Dans ce cas, vous pouvez créer une contrainte d'ordre sur un ou plusieurs ensembles de ressources à l'aide de la commande **pcs constraint order set**.

Vous pouvez définir les options suivantes pour un ensemble de ressources à l'aide de la commande **pcs constraint order set**.

- **sequential**qui peut prendre la valeur **true** ou **false** pour indiquer si l'ensemble des ressources doit être ordonné les unes par rapport aux autres. La valeur par défaut est **true**.
  La définition de **sequential** à **false** permet à un ensemble d'être ordonné par rapport à d'autres ensembles dans la contrainte d'ordonnancement, sans que ses membres ne soient ordonnés les uns par rapport aux autres. Par conséquent, cette option n'a de sens que si plusieurs ensembles sont énumérés dans la contrainte ; dans le cas contraire, la contrainte n'a aucun effet.
- **require-all**qui peut être défini à **true** ou **false** pour indiquer si toutes les ressources de l'ensemble doivent être actives avant de continuer. La définition de **require-all** à **false** signifie qu'une seule ressource de l'ensemble doit être démarrée avant de passer à l'ensemble suivant. La définition de **require-all** à **false** n'a aucun effet, sauf si elle est utilisée avec des ensembles non ordonnés, c'est-à-dire des ensembles pour lesquels **sequential** est défini sur **false**. La valeur par défaut est **true**.
- **action**qui peut être défini sur **start**, **promote**, **demote** ou **stop**, comme décrit dans le tableau "Propriétés d'une contrainte d'ordre" de la section Détermination de l'ordre d'exécution des ressources d'un cluster.
- **role**qui peut être réglé sur **Stopped**, **Started**, **Promoted**, ou **Unpromoted**.

Vous pouvez définir les options de contrainte suivantes pour un ensemble de ressources en utilisant le paramètre **setoptions** de la commande **pcs constraint order set**.

- **id**pour donner un nom à la contrainte que vous êtes en train de définir.
- **kind**qui indique comment appliquer la contrainte, comme décrit dans le tableau "Propriétés d'une contrainte d'ordre" de la section Détermination de l'ordre d'exécution des ressources d'un cluster.
- **symmetrical**pour définir si l'inverse de la contrainte s'applique à l'action opposée, comme décrit dans le tableau "Propriétés d'une contrainte d'ordre" de la section Détermination de l'ordre d'exécution des ressources d'un cluster.

```
pcs constraint order set resource1 resource2 [resourceN]... [options] [set resourceX resourceY ... [options]] [setoptions [constraint_options]]
```

Si vous avez trois ressources nommées **D1**, **D2**, et **D3**, la commande suivante les configure en tant qu'ensemble de ressources ordonné.

```
# pcs constraint order set D1 D2 D3
```

Si vous avez six ressources nommées **A**, **B**, **C**, **D**, **E**, et **F**, cet exemple configure une contrainte d'ordre pour l'ensemble des ressources qui commenceront comme suit :

- **A** et **B** démarrent indépendamment l'un de l'autre
- **C** démarre une fois que **A** ou **B** a démarré
- **D** démarre une fois que **C** a démarré
- **E** et **F** démarrent indépendamment l'un de l'autre une fois que **D** a démarré

L'arrêt des ressources n'est pas influencé par cette contrainte puisque **symmetrical=false** est défini.

```
# pcs constraint order set A B sequential=false require-all=false set C D set E F sequential=false setoptions symmetrical=false
```

13.4. CONFIGURATION DE L'ORDRE DE DÉMARRAGE POUR LES DÉPENDANCES NON GÉRÉES PAR PACEMAKER

Il est possible qu'un cluster comprenne des ressources avec des dépendances qui ne sont pas elles-mêmes gérées par le cluster. Dans ce cas, vous devez vous assurer que ces dépendances sont démarrées avant le démarrage de Pacemaker et arrêtées après l'arrêt de Pacemaker.

Vous pouvez configurer votre ordre de démarrage pour tenir compte de cette situation au moyen de la cible **systemd resource-agents-deps** . Vous pouvez créer une unité d'insertion **systemd** pour cette cible et Pacemaker s'ordonnera de manière appropriée par rapport à cette cible.

Par exemple, si un cluster inclut une ressource qui dépend du service externe **foo** qui n'est pas géré par le cluster, effectuez la procédure suivante.

1. Créez l'unité de dépôt **/etc/systemd/system/resource-agents-deps.target.d/foo.conf** qui contient les éléments suivants :

   ```
   [Unit]
   Requires=foo.service
   After=foo.service
   ```

2. Exécutez la commande **systemctl daemon-reload**.

Une dépendance de cluster spécifiée de cette manière peut être autre chose qu'un service. Par exemple, vous pouvez avoir une dépendance sur le montage d'un système de fichiers à l'adresse **/srv**, auquel cas vous devez exécuter la procédure suivante :

1. Assurez-vous que **/srv** figure dans le fichier **/etc/fstab**. Celui-ci sera automatiquement converti en fichier **systemd srv.mount** au démarrage lorsque la configuration du gestionnaire de système sera rechargée. Pour plus d'informations, voir les pages de manuel **systemd.mount**(5)

et **systemd-fstab-generator**(8).

2. Pour s'assurer que Pacemaker démarre après le montage du disque, créez l'unité de dépôt **/etc/systemd/system/resource-agents-deps.target.d/srv.conf** qui contient les éléments suivants.

```
[Unit]
Requires=srv.mount
After=srv.mount
```

3. Exécutez la commande **systemctl daemon-reload**.

Si un groupe de volumes LVM utilisé par un cluster Pacemaker contient un ou plusieurs volumes physiques résidant sur un stockage en bloc distant, tel qu'une cible iSCSI, vous pouvez configurer une cible **systemd resource-agents-deps** et une unité de dépôt **systemd** pour la cible afin de garantir que le service démarre avant le démarrage de Pacemaker.

La procédure suivante permet de configurer **blk-availability.service** en tant que dépendance. Le service **blk-availability.service** est un wrapper qui inclut **iscsi.service**, entre autres services. Si votre déploiement l'exige, vous pouvez configurer **iscsi.service** (pour iSCSI uniquement) ou **remote-fs.target** comme dépendance au lieu de **blk-availability**.

1. Créez l'unité de dépôt **/etc/systemd/system/resource-agents-deps.target.d/blk-availability.conf** qui contient les éléments suivants :

```
[Unit]
Requires=blk-availability.service
After=blk-availability.service
```

2. Exécutez la commande **systemctl daemon-reload**.

CHAPITRE 14. COLOCALISATION DES RESSOURCES DE LA GRAPPE

Pour spécifier que l'emplacement d'une ressource dépend de l'emplacement d'une autre ressource, vous configurez une contrainte de colocalisation.

La création d'une contrainte de colocalisation entre deux ressources a un effet secondaire important : elle affecte l'ordre dans lequel les ressources sont affectées à un nœud. En effet, vous ne pouvez pas placer la ressource A par rapport à la ressource B si vous ne savez pas où se trouve la ressource B. Par conséquent, lorsque vous créez des contraintes de colocalisation, il est important de déterminer si vous devez placer la ressource A avec la ressource B ou la ressource B avec la ressource A.

Une autre chose à garder à l'esprit lors de la création de contraintes de colocalisation est que, en supposant que la ressource A soit colocalisée avec la ressource B, le cluster prendra également en compte les préférences de la ressource A lorsqu'il décidera du nœud à choisir pour la ressource B.

La commande suivante crée une contrainte de colocation.

```
pcs constraint colocation add [promoted|unpromoted] source_resource with [promoted|unpromoted]
target_resource [score] [options]
```

Le tableau suivant résume les propriétés et les options permettant de configurer les contraintes de colocation.

Tableau 14.1. Paramètres d'une contrainte de colocalisation

Paramètres	Description
source_resource	La source de colocation. Si la contrainte ne peut être satisfaite, le cluster peut décider de ne pas autoriser la ressource à fonctionner.
ressource_cible	La cible de colocation. Le cluster décidera d'abord de l'emplacement de cette ressource, puis de l'emplacement de la ressource source.
score	Les valeurs positives indiquent que la ressource doit être exécutée sur le même nœud. Les valeurs négatives indiquent que les ressources ne doivent pas être exécutées sur le même nœud. Une valeur de **INFINITY**, la valeur par défaut, indique que la ressource *source_resource* doit être exécutée sur le même nœud que la ressource *target_resource*. Une valeur de -**INFINITY** indique que la ressource *source_resource* ne doit pas être exécutée sur le même nœud que la ressource *target_resource*.

Paramètres	Description
influence option	Détermine si le cluster déplacera la ressource primaire (*source_resource*) et les ressources dépendantes (*target_resource)*) vers un autre nœud lorsque la ressource dépendante atteint son seuil de migration en cas de défaillance, ou si le cluster laissera la ressource dépendante hors ligne sans provoquer de changement de service. L'option de contrainte de colocation **influence** peut avoir une valeur de **true** ou **false**. La valeur par défaut de cette option est déterminée par la valeur de la méta-option de ressource **critical** de la ressource dépendante, qui a une valeur par défaut de **true**. Lorsque cette option a une valeur de **true**, Pacemaker tentera de maintenir la ressource principale et la ressource dépendante actives. Si la ressource dépendante atteint son seuil de migration pour les défaillances, les deux ressources seront déplacées vers un autre nœud si possible. Lorsque cette option a une valeur de **false**, Pacemaker évitera de déplacer la ressource primaire en fonction de l'état de la ressource dépendante. Dans ce cas, si la ressource dépendante atteint son seuil de migration pour les défaillances, elle s'arrêtera si la ressource primaire est active et peut rester sur son nœud actuel.

14.1. SPÉCIFIER LE PLACEMENT OBLIGATOIRE DES RESSOURCES

Le placement obligatoire a lieu chaque fois que le score de la contrainte est **INFINITY** ou **-INFINITY**. Dans ce cas, si la contrainte ne peut pas être satisfaite, *source_resource* n'est pas autorisé à s'exécuter. Pour **score=INFINITY**, cela inclut les cas où *target_resource* n'est pas actif.

Si vous souhaitez que **myresource1** s'exécute toujours sur la même machine que **myresource2**, vous devez ajouter la contrainte suivante :

```
# pcs constraint colocation add myresource1 with myresource2 score=INFINITY
```

Étant donné que **INFINITY** a été utilisé, si **myresource2** ne peut pas être exécuté sur l'un des nœuds du cluster (pour quelque raison que ce soit), **myresource1** ne sera pas autorisé à s'exécuter.

Vous pouvez aussi vouloir configurer l'inverse, c'est-à-dire un cluster dans lequel **myresource1** ne peut pas s'exécuter sur la même machine que **myresource2**. Dans ce cas, utilisez **score=-INFINITY**

```
# pcs constraint colocation add myresource1 with myresource2 score=-INFINITY
```

Là encore, en spécifiant **-INFINITY**, la contrainte est contraignante. Ainsi, si le seul endroit où il est possible de courir est celui où se trouve déjà **myresource2**, alors **myresource1** ne peut courir nulle part.

14.2. SPÉCIFIER L'EMPLACEMENT CONSULTATIF DES RESSOURCES

Le placement consultatif des ressources indique que le placement des ressources est une préférence, mais qu'il n'est pas obligatoire. Pour les contraintes dont le score est supérieur à **-INFINITY** et inférieur à **INFINITY**, le cluster essaiera de tenir compte de vos souhaits, mais pourra les ignorer si l'alternative est d'arrêter certaines ressources du cluster.

14.3. COLOCALISATION D'ENSEMBLES DE RESSOURCES

Si votre configuration exige que vous créiez un ensemble de ressources colocalisées et démarrées dans l'ordre, vous pouvez configurer un groupe de ressources contenant ces ressources. Toutefois, dans certaines situations, il n'est pas approprié de configurer les ressources qui doivent être colocalisées sous la forme d'un groupe de ressources :

- Vous pouvez avoir besoin de colocaliser un ensemble de ressources, mais les ressources ne doivent pas nécessairement démarrer dans l'ordre.
- Vous pouvez avoir une ressource C qui doit être colocalisée avec une ressource A ou B, mais il n'y a pas de relation entre A et B.
- Vous pouvez avoir des ressources C et D qui doivent être colocalisées avec les ressources A et B, mais il n'y a pas de relation entre A et B ou entre C et D.

Dans ce cas, vous pouvez créer une contrainte de colocation sur un ou plusieurs ensembles de ressources à l'aide de la commande **pcs constraint colocation set**.

Vous pouvez définir les options suivantes pour un ensemble de ressources à l'aide de la commande **pcs constraint colocation set**.

- **sequential**qui peut être fixé à **true** ou **false** pour indiquer si les membres de l'ensemble doivent être colocalisés les uns avec les autres.
  La définition de **sequential** à **false** permet aux membres de cet ensemble d'être colocalisés avec un autre ensemble répertorié plus loin dans la contrainte, quels que soient les membres de cet ensemble qui sont actifs. Par conséquent, cette option n'a de sens que si un autre ensemble figure après celui-ci dans la contrainte ; dans le cas contraire, la contrainte n'a aucun effet.
- **role**qui peut être réglé sur **Stopped**, **Started**, **Promoted**, ou **Unpromoted**.

Vous pouvez définir l'option de contrainte suivante pour un ensemble de ressources en utilisant le paramètre **setoptions** de la commande **pcs constraint colocation set**.

- **id**pour donner un nom à la contrainte que vous êtes en train de définir.
- **score**pour indiquer le degré de préférence pour cette contrainte. Pour plus d'informations sur cette option, voir le tableau "Options de contrainte d'emplacement" dans la section Configuration des contraintes d'emplacement

Lors de l'énumération des membres d'un ensemble, chaque membre est associé à celui qui le précède. Par exemple, "ensemble A B" signifie "B est colocalisé avec A". Toutefois, lorsque l'on énumère plusieurs ensembles, chaque ensemble est associé à celui qui le suit. Par exemple, "set C D sequential=false set A B" signifie que "set C D (où C et D n'ont aucune relation entre eux) est colocalisé avec set A B (où B est colocalisé avec A)".

La commande suivante crée une contrainte de colocation sur un ou plusieurs ensembles de ressources.

```
pcs constraint colocation set resource1 resource2 ] [resourceN]... [options] [set resourceX resourceY ] ... [options]] [setoptions [constraint_options]]
```

Utilisez la commande suivante pour supprimer les contraintes de colocation avec *source_resource*.

```
pcs constraint colocation remove source_resource target_resource
```

CHAPITRE 15. AFFICHAGE DES CONTRAINTES ET DES DÉPENDANCES DES RESSOURCES

Plusieurs commandes permettent d'afficher les contraintes configurées. Vous pouvez afficher toutes les contraintes de ressources configurées ou limiter l'affichage des contraintes de ressources à des types spécifiques de contraintes de ressources. En outre, vous pouvez afficher les dépendances de ressources configurées.

Affichage de toutes les contraintes configurées

La commande suivante répertorie toutes les contraintes actuelles en matière d'emplacement, d'ordre et de colocation. Si l'option **--full** est spécifiée, elle affiche les identifiants internes des contraintes.

```
pcs constraint [list|show] [--full]
```

Par défaut, la liste des contraintes de ressources n'affiche pas les contraintes expirées. Pour inclure les contraintes expirées dans la liste, utilisez l'option **--all** de la commande **pcs constraint**. Les contraintes expirées seront alors listées, et les contraintes et les règles qui leur sont associées seront notées **(expired)** dans l'affichage.

Affichage des contraintes de localisation

La commande suivante répertorie toutes les contraintes de localisation actuelles.

- Si **resources** est spécifié, les contraintes de localisation sont affichées par ressource. Il s'agit du comportement par défaut.
- Si **nodes** est spécifié, les contraintes de localisation sont affichées par nœud.
- Si des ressources ou des nœuds spécifiques sont spécifiés, seules les informations relatives à ces ressources ou nœuds sont affichées.

```
pcs constraint location [show [resources [resource...]] | [nœuds [node...]]] [--full]
```

Affichage des contraintes d'ordre

La commande suivante dresse la liste de toutes les contraintes d'ordre actuelles.

```
pcs constraint order [show]
```

Affichage des contraintes de colocation

La commande suivante répertorie toutes les contraintes de colocation actuelles.

```
pcs constraint colocation [show]
```

Affichage des contraintes spécifiques aux ressources

La commande suivante répertorie les contraintes qui font référence à des ressources spécifiques.

```
pcs contrainte ref resource...
```

Affichage des dépendances des ressources

La commande suivante affiche les relations entre les ressources du cluster dans une structure arborescente.

```
pcs resource relations resource [--full]
```

Si l'option **--full** est utilisée, la commande affiche des informations supplémentaires, notamment les identifiants des contraintes et les types de ressources.

Dans l'exemple suivant, il y a 3 ressources configurées : C, D et E.

```
# pcs constraint order start C then start D
Adding C D (kind: Mandatory) (Options: first-action=start then-action=start)
# pcs constraint order start D then start E
Adding D E (kind: Mandatory) (Options: first-action=start then-action=start)

# pcs resource relations C
C
`- order
  |  start C then start D
  `- D
    `- order
      |  start D then start E
      `- E
# pcs resource relations D
D
|- order
|  |  start C then start D
|  `- C
`- order
  |  start D then start E
  `- E
# pcs resource relations E
E
`- order
  |  start D then start E
  `- D
    `- order
      |  start C then start D
      `- C
```

Dans l'exemple suivant, il y a deux ressources configurées, A et B. Les ressources A et B font partie du groupe de ressources G : A et B. Les ressources A et B font partie du groupe de ressources G.

```
# pcs resource relations A
A
`- outer resource
  `- G
    `- inner resource(s)
      |  members: A B
      `- B
# pcs resource relations B
B
`- outer resource
  `- G
    `- inner resource(s)
```

```
        |  members: A B
        `- A
# pcs resource relations G
G
`- inner resource(s)
   |  members: A B
   |- A
   `- B
```

CHAPITRE 16. DÉTERMINER L'EMPLACEMENT DES RESSOURCES À L'AIDE DE RÈGLES

Pour des contraintes de localisation plus complexes, vous pouvez utiliser les règles de Pacemaker pour déterminer la localisation d'une ressource.

16.1. RÈGLES RELATIVES AUX STIMULATEURS CARDIAQUES

Les règles de Pacemaker peuvent être utilisées pour rendre votre configuration plus dynamique. Elles peuvent notamment servir à affecter des machines à différents groupes de traitement (à l'aide d'un attribut de nœud) en fonction de l'heure, puis à utiliser cet attribut lors de la création de contraintes d'emplacement.

Chaque règle peut contenir un certain nombre d'expressions, d'expressions de date et même d'autres règles. Les résultats des expressions sont combinés sur la base du champ **boolean-op** de la règle afin de déterminer si la règle est finalement évaluée à **true** ou **false**. La suite dépend du contexte dans lequel la règle est utilisée.

Tableau 16.1. Propriétés d'une règle

Field	Description
role	Limite l'application de la règle au seul cas où la ressource est dans ce rôle. Valeurs autorisées : **Started**, **Unpromoted,** et **Promoted**. REMARQUE : Une règle avec **role="Promoted"** ne peut pas déterminer l'emplacement initial d'une instance clone. Elle n'affecte que l'instance active qui sera promue.
score	Le score à appliquer si la règle est évaluée à **true**. Limité à l'utilisation dans les règles qui font partie des contraintes de localisation.
score-attribute	Attribut du nœud à rechercher et à utiliser comme score si la règle est évaluée à **true**. Limité à l'utilisation dans les règles qui font partie des contraintes de localisation.
boolean-op	Comment combiner le résultat de plusieurs objets d'expression. Valeurs autorisées : **and** et **or**. La valeur par défaut est **and**.

16.1.1. Expressions d'attributs de nœuds

Les expressions d'attributs de nœuds sont utilisées pour contrôler une ressource sur la base des attributs définis par un ou plusieurs nœuds.

Tableau 16.2. Propriétés d'une expression

Field	Description
attribute	L'attribut du nœud à tester
type	Détermine la manière dont la ou les valeurs doivent être testées. Valeurs autorisées : **string**, **integer**, **number**, **version**. La valeur par défaut est **string**.
operation	La comparaison à effectuer. Valeurs autorisées : * **lt** - Vrai si la valeur de l'attribut du nœud est inférieure à **value** * **gt** - Vrai si la valeur de l'attribut du nœud est supérieure à **value** * **lte** - Vrai si la valeur de l'attribut du nœud est inférieure ou égale à **value** * **gte** - Vrai si la valeur de l'attribut du nœud est supérieure ou égale à **value** * **eq** - Vrai si la valeur de l'attribut du nœud est égale à **value** * **ne** - Vrai si la valeur de l'attribut du nœud n'est pas égale à **value** * **defined** - Vrai si le nœud possède l'attribut nommé * **not_defined** - Vrai si le nœud n'a pas l'attribut nommé
value	Valeur fournie par l'utilisateur pour la comparaison (obligatoire sauf si **operation** est **defined** ou **not_defined**)

Outre les attributs ajoutés par l'administrateur, le cluster définit pour chaque nœud des attributs spéciaux intégrés qui peuvent également être utilisés, comme décrit dans le tableau suivant.

Tableau 16.3. Attributs de nœuds intégrés

Nom	Description
#uname	Nom du nœud
#id	ID du nœud

Nom	Description
#kind	Type de nœud. Les valeurs possibles sont **cluster**, **remote** et **container**. La valeur de**kind** est **remote** pour les nœuds Pacemaker Remote créés avec la ressource **ocf:pacemaker:remote**, et **container** pour les nœuds invités et les nœuds de regroupement Pacemaker Remote.
#is_dc	**true** si ce nœud est un contrôleur désigné (DC), **false** sinon
#cluster_name	La valeur de la propriété **cluster-name** cluster, si elle est définie
#site_name	La valeur de l'attribut du nœud **site-name**, s'il est défini, sinon identique à **#cluster-name**
#role	Le rôle du clone promouvable concerné sur ce nœud. Valable uniquement dans le cadre d'une règle relative à une contrainte d'emplacement pour un clone promouvable.

16.1.2. Expressions basées sur l'heure ou la date

Les expressions de date sont utilisées pour contrôler une ressource ou une option de cluster en fonction de la date et de l'heure actuelles. Elles peuvent contenir une spécification de date facultative.

Tableau 16.4. Propriétés d'une expression de date

Field	Description
start	Une date/heure conforme à la spécification ISO8601.
end	Une date/heure conforme à la spécification ISO8601.

Field	Description
operation	Compare la date et l'heure actuelles avec la date de début ou la date de fin ou les deux, selon le contexte. Valeurs autorisées : * **gt** - Vrai si la date/heure actuelle est postérieure à **start** * **lt** - Vrai si la date/heure actuelle est antérieure à **end** * **in_range** - Vrai si la date/heure actuelle est postérieure à **start** et antérieure à**end** * **date-spec** - effectue une comparaison avec la date et l'heure actuelles, à la manière d'un cron

16.1.3. Date des spécifications

Les spécifications de date sont utilisées pour créer des expressions de type cron relatives au temps. Chaque champ peut contenir un seul nombre ou une seule plage. Au lieu de prendre la valeur zéro par défaut, tout champ non fourni est ignoré.

Par exemple, **monthdays="1"** correspond au premier jour de chaque mois et **hours="09-17"** correspond aux heures comprises entre 9 heures et 17 heures (incluses). Cependant, vous ne pouvez pas spécifier **weekdays="1,2"** ou **weekdays="1-2,5-6"** car ils contiennent plusieurs plages.

Tableau 16.5. Propriétés d'une spécification de date

Field	Description
id	Un nom unique pour la date
hours	Valeurs autorisées : 0-23
monthdays	Valeurs autorisées : 0-31 (en fonction du mois et de l'année)
weekdays	Valeurs autorisées : 1-7 (1=lundi, 7=dimanche)
yeardays	Valeurs autorisées : 1-366 (en fonction de l'année)
months	Valeurs autorisées : 1-12
weeks	Valeurs autorisées : 1-53 (en fonction de **weekyear**)
years	Année selon le calendrier grégorien

Field	Description
weekyears	Peut différer des années grégoriennes ; par exemple, **2005-001 Ordinal** est aussi **2005-01-01 Gregorian** est aussi **2004-W53-6 Weekly**
moon	Valeurs autorisées : 0-7 (0 est la nouvelle lune, 4 est la pleine lune).

16.2. CONFIGURATION D'UNE CONTRAINTE DE LOCALISATION DU STIMULATEUR CARDIAQUE À L'AIDE DE RÈGLES

Utilisez la commande suivante pour configurer une contrainte Pacemaker qui utilise des règles. Si **score** est omis, la valeur par défaut est INFINITY. Si **resource-discovery** est omis, la valeur par défaut est **always**.

Pour plus d'informations sur l'option **resource-discovery**, voir Limitation de la découverte des ressources à un sous-ensemble de nœuds.

Comme pour les contraintes de localisation de base, vous pouvez utiliser des expressions régulières pour les ressources avec ces contraintes.

Lors de l'utilisation de règles pour configurer les contraintes de localisation, la valeur de **score** peut être positive ou négative, une valeur positive indiquant "préfère" et une valeur négative indiquant "évite".

```
pcs constraint location rsc rule [resource-discovery=option] [role=promoted|unpromoted]
[score=score | score-attribute=attribute] expression
```

L'option *expression* peut être l'une des suivantes, où *duration_options* et *date_spec_options* sont : heures, jours du mois, jours de la semaine, jours de l'année, mois, semaines, années, années de la semaine et lune, comme décrit dans le tableau "Properties of a Date Specification" (Propriétés d'une spécification de date) dans Date specifications (spécifications de date).

- **defined|not_defined *attribute***
- ***attribute* lt|gt|lte|gte|eq|ne [string|integer|number|version] *value***
- **date gt|lt *date***
- **date in_range *date* to *date***
- **date in_range *date* to duration *duration_options* …**
- **date-spec *date_spec_options***
- ***expression* and|or *expression***
- **(*expression*)**

Notez que les durées sont un moyen alternatif de spécifier une fin pour les opérations **in_range** au moyen de calculs. Par exemple, vous pouvez spécifier une durée de 19 mois.

La contrainte de lieu suivante configure une expression qui est vraie si l'on se trouve à n'importe quel moment de l'année 2018.

```
# pcs constraint location Webserver rule score=INFINITY date-spec years=2018
```

La commande suivante configure une expression qui est vraie de 9 heures à 17 heures, du lundi au vendredi. Notez que la valeur de 16 heures correspond à 16:59:59, car la valeur numérique (heure) correspond toujours.

```
# pcs constraint location Webserver rule score=INFINITY date-spec hours="9-16"
weekdays="1-5"
```

La commande suivante configure une expression qui est vraie lorsqu'il y a une pleine lune le vendredi 13.

```
# pcs constraint location Webserver rule date-spec weekdays=5 monthdays=13 moon=4
```

Pour supprimer une règle, utilisez la commande suivante. Si la règle que vous supprimez est la dernière règle de sa contrainte, la contrainte sera supprimée.

```
pcs contrainte règle supprimer rule_id
```

CHAPITRE 17. GESTION DES RESSOURCES DE LA GRAPPE

Il existe plusieurs commandes permettant d'afficher, de modifier et d'administrer les ressources d'un cluster.

17.1. AFFICHAGE DES RESSOURCES CONFIGURÉES

Pour afficher une liste de toutes les ressources configurées, utilisez la commande suivante.

```
état des ressources pcs
```

Par exemple, si votre système est configuré avec une ressource nommée **VirtualIP** et une ressource nommée **WebSite**, la commande **pcs resource status** produit le résultat suivant.

```
# pcs resource status
 VirtualIP (ocf::heartbeat:IPaddr2): Started
 WebSite (ocf::heartbeat:apache): Started
```

Pour afficher les paramètres configurés pour une ressource, utilisez la commande suivante.

```
configuration des ressources pcs resource_id
```

Par exemple, la commande suivante affiche les paramètres actuellement configurés pour la ressource **VirtualIP**.

```
# pcs resource config VirtualIP
 Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
  Attributes: ip=192.168.0.120 cidr_netmask=24
  Operations: monitor interval=30s
```

Pour afficher l'état d'une ressource individuelle, utilisez la commande suivante.

```
état des ressources pcs resource_id
```

Par exemple, si votre système est configuré avec une ressource nommée **VirtualIP**, la commande **pcs resource status VirtualIP** produit le résultat suivant.

```
# pcs resource status VirtualIP
 VirtualIP      (ocf::heartbeat:IPaddr2):       Started
```

Pour afficher l'état des ressources exécutées sur un nœud spécifique, utilisez la commande suivante. Vous pouvez utiliser cette commande pour afficher l'état des ressources sur les nœuds de cluster et les nœuds distants.

```
état des ressources pcs node=node_id
```

Par exemple, si **node-01** exécute des ressources nommées **VirtualIP** et **WebSite**, la commande **pcs resource status node=node-01** peut produire le résultat suivant.

```
# pcs resource status node=node-01
 VirtualIP      (ocf::heartbeat:IPaddr2):       Started
 WebSite        (ocf::heartbeat:apache):        Started
```

17.2. EXPORTER LES RESSOURCES D'UN CLUSTER SOUS FORME DE COMMANDES PCS

Depuis Red Hat Enterprise Linux 9.1, vous pouvez afficher les commandes **pcs** qui peuvent être utilisées pour recréer les ressources configurées du cluster sur un système différent en utilisant l'option **--output-format=cmd** de la commande **pcs resource config**.

Les commandes suivantes créent quatre ressources pour un serveur HTTP Apache actif/passif dans un cluster de haute disponibilité Red Hat : une ressource **LVM-activate**, une ressource **Filesystem**, une ressource **IPaddr2** et une ressource **Apache**.

```
# pcs resource create my_lvm ocf:heartbeat:LVM-activate vgname=my_vg
vg_access_mode=system_id --group apachegroup
# pcs resource create my_fs Filesystem device="/dev/my_vg/my_lv" directory="/var/www"
fstype="xfs" --group apachegroup
# pcs resource create VirtualIP IPaddr2 ip=198.51.100.3 cidr_netmask=24 --group apachegroup
# pcs resource create Website apache configfile="/etc/httpd/conf/httpd.conf"
statusurl="http://127.0.0.1/server-status" --group apachegroup
```

Après avoir créé les ressources, la commande suivante affiche les commandes **pcs** que vous pouvez utiliser pour recréer ces ressources sur un autre système.

```
# pcs resource config --output-format=cmd
pcs resource create --no-default-ops --force -- my_lvm ocf:heartbeat:LVM-activate \
  vg_access_mode=system_id vgname=my_vg \
  op \
    monitor interval=30s id=my_lvm-monitor-interval-30s timeout=90s \
    start interval=0s id=my_lvm-start-interval-0s timeout=90s \
    stop interval=0s id=my_lvm-stop-interval-0s timeout=90s;
pcs resource create --no-default-ops --force -- my_fs ocf:heartbeat:Filesystem \
  device=/dev/my_vg/my_lv directory=/var/www fstype=xfs \
  op \
    monitor interval=20s id=my_fs-monitor-interval-20s timeout=40s \
    start interval=0s id=my_fs-start-interval-0s timeout=60s \
    stop interval=0s id=my_fs-stop-interval-0s timeout=60s;
pcs resource create --no-default-ops --force -- VirtualIP ocf:heartbeat:IPaddr2 \
  cidr_netmask=24 ip=198.51.100.3 \
  op \
    monitor interval=10s id=VirtualIP-monitor-interval-10s timeout=20s \
    start interval=0s id=VirtualIP-start-interval-0s timeout=20s \
    stop interval=0s id=VirtualIP-stop-interval-0s timeout=20s;
pcs resource create --no-default-ops --force -- Website ocf:heartbeat:apache \
  configfile=/etc/httpd/conf/httpd.conf statusurl=http://127.0.0.1/server-status \
  op \
    monitor interval=10s id=Website-monitor-interval-10s timeout=20s \
    start interval=0s id=Website-start-interval-0s timeout=40s \
    stop interval=0s id=Website-stop-interval-0s timeout=60s;
pcs resource group add apachegroup \
  my_lvm my_fs VirtualIP Website
```

Pour afficher la commande **pcs** ou les commandes que vous pouvez utiliser pour recréer une seule ressource configurée, indiquez l'ID de cette ressource.

```
# pcs resource config VirtualIP --output-format=cmd
```

```
pcs resource create --no-default-ops --force -- VirtualIP ocf:heartbeat:IPaddr2 \
  cidr_netmask=24 ip=198.51.100.3 \
  op \
    monitor interval=10s id=VirtualIP-monitor-interval-10s timeout=20s \
    start interval=0s id=VirtualIP-start-interval-0s timeout=20s \
    stop interval=0s id=VirtualIP-stop-interval-0s timeout=20s
```

17.3. MODIFICATION DES PARAMÈTRES DES RESSOURCES

Pour modifier les paramètres d'une ressource configurée, utilisez la commande suivante.

```
mise à jour des ressources pcs resource_id [resource_options]
```

La séquence de commandes suivante montre les valeurs initiales des paramètres configurés pour la ressource **VirtualIP**, la commande de modification de la valeur du paramètre **ip** et les valeurs qui suivent la commande de mise à jour.

```
# pcs resource config VirtualIP
 Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
  Attributes: ip=192.168.0.120 cidr_netmask=24
  Operations: monitor interval=30s
# pcs resource update VirtualIP ip=192.169.0.120
# pcs resource config VirtualIP
 Resource: VirtualIP (type=IPaddr2 class=ocf provider=heartbeat)
  Attributes: ip=192.169.0.120 cidr_netmask=24
  Operations: monitor interval=30s
```

NOTE

Lorsque vous mettez à jour le fonctionnement d'une ressource à l'aide de la commande **pcs resource update**, toutes les options que vous ne mentionnez pas expressément sont ramenées à leur valeur par défaut.

17.4. EFFACEMENT DE L'ÉTAT D'ÉCHEC DES RESSOURCES DE LA GRAPPE

Si une ressource a échoué, un message d'échec apparaît lorsque vous affichez l'état de la grappe à l'aide de la commande **pcs status**. Après avoir tenté de résoudre la cause de l'échec, vous pouvez vérifier l'état actualisé de la ressource en exécutant à nouveau la commande **pcs status**, et vous pouvez vérifier le nombre d'échecs pour les ressources du cluster avec la commande **pcs resource failcount show --full**.

Vous pouvez effacer l'état d'échec d'une ressource à l'aide de la commande **pcs resource cleanup**. La commande **pcs resource cleanup** réinitialise l'état de la ressource et la valeur **failcount** pour la ressource. Cette commande supprime également l'historique des opérations de la ressource et redétermine son état actuel.

La commande suivante réinitialise l'état de la ressource et la valeur **failcount** pour la ressource spécifiée par *resource_id*.

```
nettoyage des ressources de la PCS resource_id
```

Si vous ne spécifiez pas *resource_id*, la commande **pcs resource cleanup** réinitialise l'état de la ressource et la valeur **failcount** pour toutes les ressources avec un nombre d'échecs.

Outre la commande **pcs resource cleanup *resource_id*** vous pouvez également réinitialiser l'état de la ressource et effacer l'historique des opérations d'une ressource à l'aide de la commande **pcs resource refresh *resource_id*** pour réinitialiser l'état d'une ressource et effacer l'historique des opérations d'une ressource. Comme pour la commande **pcs resource cleanup**, vous pouvez exécuter la commande **pcs resource refresh** sans spécifier d'options pour réinitialiser l'état de la ressource et la valeur **failcount** pour toutes les ressources.

Les commandes **pcs resource cleanup** et **pcs resource refresh** effacent l'historique des opérations pour une ressource et redétectent l'état actuel de la ressource. La commande **pcs resource cleanup** n'agit que sur les ressources dont les actions ont échoué, comme le montre l'état de la grappe, tandis que la commande **pcs resource refresh** agit sur les ressources quel que soit leur état actuel.

17.5. DÉPLACER DES RESSOURCES DANS UN CLUSTER

Pacemaker propose divers mécanismes pour configurer une ressource afin qu'elle soit déplacée d'un nœud à l'autre et pour déplacer manuellement une ressource en cas de besoin.

Vous pouvez déplacer manuellement des ressources dans une grappe à l'aide des commandes **pcs resource move** et **pcs resource relocate**, comme décrit dans Déplacement manuel des ressources de la grappe. Outre ces commandes, vous pouvez également contrôler le comportement des ressources de la grappe en activant, en désactivant et en interdisant des ressources, comme décrit dans la section Désactivation, activation et interdiction des ressources de la gra ppe.

Vous pouvez configurer une ressource de manière à ce qu'elle soit déplacée vers un nouveau nœud après un nombre défini d'échecs, et vous pouvez configurer un cluster pour qu'il déplace des ressources en cas de perte de connectivité externe.

17.5.1. Déplacement des ressources en cas de défaillance

Lorsque vous créez une ressource, vous pouvez la configurer de manière à ce qu'elle passe à un nouveau nœud après un nombre défini d'échecs en définissant l'option **migration-threshold** pour cette ressource. Une fois le seuil atteint, ce nœud ne sera plus autorisé à faire fonctionner la ressource défaillante jusqu'à ce qu'elle atteigne ce seuil :

- La valeur **failure-timeout** de la ressource est atteinte.
- L'administrateur réinitialise manuellement le nombre d'échecs de la ressource à l'aide de la commande **pcs resource cleanup**.

La valeur de **migration-threshold** est fixée par défaut à **INFINITY**. **INFINITY** est défini en interne comme un nombre très grand mais fini. La valeur 0 désactive la fonction **migration-threshold**.

NOTE

La définition de **migration-threshold** pour une ressource n'est pas la même chose que la configuration d'une ressource pour la migration, dans laquelle la ressource est déplacée vers un autre emplacement sans perte d'état.

L'exemple suivant ajoute un seuil de migration de 10 à la ressource nommée **dummy_resource**, ce qui indique que la ressource sera déplacée vers un nouveau nœud après 10 échecs.

```
# pcs resource meta dummy_resource migration-threshold=10
```

Vous pouvez ajouter un seuil de migration aux valeurs par défaut pour l'ensemble du cluster à l'aide de la commande suivante.

```
# pcs resource defaults update migration-threshold=10
```

Pour déterminer l'état de défaillance actuel de la ressource et ses limites, utilisez la commande **pcs resource failcount show**.

Il existe deux exceptions au concept de seuil de migration ; elles se produisent lorsqu'une ressource ne démarre pas ou ne s'arrête pas. Si la propriété de cluster **start-failure-is-fatal** est définie sur **true** (ce qui est le cas par défaut), les échecs de démarrage entraînent la définition de **failcount** sur **INFINITY** et provoquent toujours le déplacement immédiat de la ressource.

Les échecs d'arrêt sont légèrement différents et cruciaux. Si une ressource ne s'arrête pas et que l'option STONITH est activée, le cluster clôturera le nœud afin de pouvoir démarrer la ressource ailleurs. Si STONITH n'est pas activé, le cluster n'a aucun moyen de continuer et n'essaiera pas de démarrer la ressource ailleurs, mais essaiera de l'arrêter à nouveau après le délai d'échec.

17.5.2. Déplacement des ressources en raison de changements de connectivité

La configuration du cluster pour déplacer les ressources en cas de perte de connectivité externe se fait en deux étapes.

1. Ajoutez une ressource **ping** au cluster. La ressource **ping** utilise l'utilitaire système du même nom pour tester si une liste de machines (spécifiée par le nom d'hôte DNS ou l'adresse IPv4/IPv6) est accessible et utilise les résultats pour maintenir un attribut de nœud appelé **pingd**.

2. Configurez une contrainte d'emplacement pour la ressource qui déplacera la ressource vers un autre nœud en cas de perte de connectivité.

Le tableau suivant décrit les propriétés que vous pouvez définir pour une ressource **ping**.

Tableau 17.1. Propriétés d'une ressource ping

Field	Description
dampen	Le temps d'attente (amortissement) pour que d'autres changements se produisent. Cela permet d'éviter qu'une ressource ne rebondisse dans la grappe lorsque les nœuds de la grappe remarquent la perte de connectivité à des moments légèrement différents.
multiplier	Le nombre de nœuds ping connectés est multiplié par cette valeur pour obtenir un score. Utile lorsque plusieurs nœuds ping sont configurés.
host_list	Les machines à contacter pour déterminer l'état actuel de la connectivité. Les valeurs autorisées comprennent les noms d'hôtes DNS résolvables, les adresses IPv4 et IPv6. Les entrées de la liste d'hôtes sont séparées par des espaces.

L'exemple de commande suivant crée une ressource **ping** qui vérifie la connectivité avec **gateway.example.com**. En pratique, vous vérifiez la connectivité avec la passerelle/le routeur de votre réseau. Vous configurez la ressource **ping** comme un clone afin qu'elle s'exécute sur tous les nœuds du cluster.

```
# pcs resource create ping ocf:pacemaker:ping dampen=5s multiplier=1000
host_list=gateway.example.com clone
```

L'exemple suivant configure une règle de contrainte d'emplacement pour la ressource existante nommée **Webserver**. Ainsi, la ressource **Webserver** sera déplacée vers un hôte capable d'envoyer un ping à **gateway.example.com** si l'hôte sur lequel elle est actuellement exécutée ne peut pas envoyer de ping à **gateway.example.com**.

```
# pcs constraint location Webserver rule score=-INFINITY pingd lt 1 or not_defined pingd
```

17.6. DÉSACTIVATION D'UNE OPÉRATION DE SURVEILLANCE

La façon la plus simple d'arrêter un moniteur récurrent est de le supprimer. Cependant, il peut arriver que vous ne souhaitiez le désactiver que temporairement. Dans ce cas, ajoutez **enabled="false"** à la définition de l'opération. Lorsque vous souhaitez rétablir l'opération de surveillance, ajoutez **enabled="true"** à la définition de l'opération.

Lorsque vous mettez à jour l'opération d'une ressource à l'aide de la commande **pcs resource update**, toutes les options que vous ne mentionnez pas spécifiquement sont réinitialisées à leur valeur par défaut. Par exemple, si vous avez configuré une opération de surveillance avec un délai d'attente personnalisé de 600, l'exécution des commandes suivantes réinitialisera le délai d'attente à la valeur par défaut de 20 (ou à la valeur par défaut que vous avez définie à l'aide de la commande **pcs resource op defaults**).

```
# pcs resource update resourceXZY op monitor enabled=false
# pcs resource update resourceXZY op monitor enabled=true
```

Afin de conserver la valeur originale de 600 pour cette option, lorsque vous rétablissez l'opération de surveillance, vous devez spécifier cette valeur, comme dans l'exemple suivant.

```
# pcs resource update resourceXZY op monitor timeout=600 enabled=true
```

17.7. CONFIGURATION ET GESTION DES BALISES DE RESSOURCES DE LA GRAPPE

Vous pouvez utiliser la commande **pcs** pour étiqueter les ressources d'un cluster. Cela vous permet d'activer, de désactiver, de gérer ou d'annuler la gestion d'un ensemble de ressources spécifié à l'aide d'une seule commande.

17.7.1. Marquage des ressources des clusters pour l'administration par catégorie

La procédure suivante permet de baliser deux ressources avec une balise de ressource et de désactiver les ressources balisées. Dans cet exemple, les ressources existantes à baliser sont nommées **d-01** et **d-02**.

Procédure

1. Créer une balise nommée **special-resources** pour les ressources **d-01** et **d-02**.

```
[root@node-01]# pcs tag create special-resources d-01 d-02
```

2. Affiche la configuration de la balise de ressource.

```
[root@node-01]# pcs tag config
special-resources
  d-01
  d-02
```

3. Désactiver toutes les ressources étiquetées avec la balise **special-resources**.

```
[root@node-01]# pcs resource disable special-resources
```

4. Affichez l'état des ressources pour confirmer que les ressources **d-01** et **d-02** sont désactivées.

```
[root@node-01]# pcs resource
  * d-01        (ocf::pacemaker:Dummy): Stopped (disabled)
  * d-02        (ocf::pacemaker:Dummy): Stopped (disabled)
```

Outre la commande **pcs resource disable**, les commandes **pcs resource enable**, **pcs resource manage** et **pcs resource unmanage** permettent d'administrer les ressources balisées.

Après avoir créé une balise de ressource :

- Vous pouvez supprimer une balise de ressource à l'aide de la commande **pcs tag delete**.
- Vous pouvez modifier la configuration d'une balise de ressource existante à l'aide de la commande **pcs tag update**.

17.7.2. Suppression d'une ressource cluster étiquetée

Vous ne pouvez pas supprimer une ressource cluster balisée avec la commande **pcs**. Pour supprimer une ressource balisée, suivez la procédure suivante.

Procédure

1. Supprimer la balise ressource.

   a. La commande suivante supprime la balise de ressource **special-resources** de toutes les ressources portant cette balise,

```
[root@node-01]# pcs tag remove special-resources
[root@node-01]# pcs tag
 No tags defined
```

   b. La commande suivante supprime le tag de ressource **special-resources** de la ressource **d-01** uniquement.

```
[root@node-01]# pcs tag update special-resources remove d-01
```

2. Supprimer la ressource.

```
[root@node-01]# pcs resource delete d-01
Attempting to stop: d-01... Stopped
```

CHAPITRE 18. CRÉATION DE RESSOURCES DE CLUSTER ACTIVES SUR PLUSIEURS NŒUDS (RESSOURCES CLONÉES)

Vous pouvez cloner une ressource de cluster afin qu'elle soit active sur plusieurs nœuds. Par exemple, vous pouvez utiliser des ressources clonées pour configurer plusieurs instances d'une ressource IP à distribuer dans un cluster pour l'équilibrage des nœuds. Vous pouvez cloner n'importe quelle ressource à condition que l'agent de ressources la prenne en charge. Un clone se compose d'une ressource ou d'un groupe de ressources.

NOTE

Seules les ressources qui peuvent être actives sur plusieurs nœuds en même temps conviennent au clonage. Par exemple, une ressource **Filesystem** qui monte un système de fichiers non groupé tel que **ext4** à partir d'un périphérique à mémoire partagée ne doit pas être clonée. Étant donné que la partition **ext4** ne tient pas compte des clusters, ce système de fichiers n'est pas adapté aux opérations de lecture/écriture effectuées simultanément sur plusieurs nœuds.

18.1. CRÉATION ET SUPPRESSION D'UNE RESSOURCE CLONÉE

Vous pouvez créer une ressource et un clone de cette ressource en même temps.

Pour créer une ressource et un clone de la ressource avec la commande unique suivante.

```
pcs resource create resource_id [standard:[provider:]]type [resource options] [meta resource meta options] clone [clone_id] [clone options]
```

```
pcs resource create resource_id [standard:[provider:]]type [resource options] [meta resource meta options] clone [clone options]
```

Par défaut, le nom du clone sera ***resource_id*-clone**. Vous pouvez définir un nom personnalisé pour le clone en spécifiant une valeur pour l'option *clone_id*.

Vous ne pouvez pas créer un groupe de ressources et un clone de ce groupe de ressources en une seule commande.

Vous pouvez également créer un clone d'une ressource ou d'un groupe de ressources créé précédemment à l'aide de la commande suivante.

```
pcs resource clone resource_id | group_id [clone_id][clone options]...
```

```
pcs resource clone resource_id | group_id [clone options]...
```

Par défaut, le nom du clone sera ***resource_id*-clone** ou ***group_name*-clone**. Vous pouvez définir un nom personnalisé pour le clone en spécifiant une valeur pour l'option *clone_id*.

NOTE

Vous devez modifier la configuration des ressources sur un seul nœud.

NOTE

Lors de la configuration des contraintes, utilisez toujours le nom du groupe ou du clone.

Lorsque vous créez un clone d'une ressource, le clone prend par défaut le nom de la ressource avec **-clone** ajouté au nom. La commande suivante crée une ressource de type **apache** nommée **webfarm** et un clone de cette ressource nommé **webfarm-clone**.

```
# pcs resource create webfarm apache clone
```

NOTE

Lorsque vous créez un clone de ressource ou de groupe de ressources qui sera ordonné après un autre clone, vous devez presque toujours définir l'option **interleave=true**. Cela garantit que les copies du clone dépendant peuvent s'arrêter ou démarrer lorsque le clone dont il dépend s'est arrêté ou a démarré sur le même nœud. Si vous ne définissez pas cette option, si une ressource clonée B dépend d'une ressource clonée A et qu'un nœud quitte le cluster, lorsque le nœud revient dans le cluster et que la ressource A démarre sur ce nœud, toutes les copies de la ressource B sur tous les nœuds redémarreront. En effet, lorsqu'une ressource clonée dépendante n'a pas l'option **interleave**, toutes les instances de cette ressource dépendent d'une instance en cours d'exécution de la ressource dont elle dépend.

La commande suivante permet de supprimer un clone d'une ressource ou d'un groupe de ressources. Cette opération ne supprime pas la ressource ou le groupe de ressources lui-même.

```
pcs resource unclone resource_id | clone_id | group_name
```

Le tableau suivant décrit les options que vous pouvez spécifier pour une ressource clonée.

Tableau 18.1. Options de clonage des ressources

Field	Description
priority, target-role, is-managed	Options héritées de la ressource clonée, comme décrit dans le tableau "Resource Meta Options" de la section Configuration des méta-options des ressources.
clone-max	Nombre de copies de la ressource à démarrer. La valeur par défaut est le nombre de nœuds dans le cluster.
clone-node-max	Combien de copies de la ressource peuvent être démarrées sur un seul nœud ; la valeur par défaut est **1**.
notify	Lors de l'arrêt ou du démarrage d'une copie du clone, il convient d'informer au préalable toutes les autres copies et de leur indiquer quand l'action a réussi. Valeurs autorisées : **false**, **true**. La valeur par défaut est **false**.

Field	Description
globally-unique	Chaque copie du clone remplit-elle une fonction différente ? Valeurs autorisées : **false**, **true** Si la valeur de cette option est **false**, ces ressources se comportent de manière identique partout où elles fonctionnent et il ne peut donc y avoir qu'une seule copie du clone actif par machine. Si la valeur de cette option est **true**, une copie du clone s'exécutant sur une machine n'est pas équivalente à une autre instance, que celle-ci s'exécute sur un autre nœud ou sur le même nœud. La valeur par défaut est **true** si la valeur de**clone-node-max** est supérieure à un ; sinon, la valeur par défaut est **false**.
ordered	Les copies doivent-elles être lancées en série (au lieu d'être lancées en parallèle). Valeurs autorisées : **false**, **true**. La valeur par défaut est**false**.
interleave	Modifie le comportement des contraintes d'ordre (entre clones) de sorte que les copies du premier clone puissent démarrer ou s'arrêter dès que la copie sur le même nœud du second clone a démarré ou s'est arrêtée (au lieu d'attendre que chaque instance du second clone ait démarré ou se soit arrêtée). Valeurs autorisées : **false**, **true**. La valeur par défaut est **false**.
clone-min	Si une valeur est spécifiée, les clones ordonnés après ce clone ne pourront pas démarrer avant que le nombre spécifié d'instances du clone d'origine ne soit en cours d'exécution, même si l'option **interleave** est définie sur **true**.

Pour obtenir un modèle d'allocation stable, les clones sont légèrement collants par défaut, ce qui indique qu'ils ont une légère préférence pour rester sur le nœud où ils s'exécutent. Si aucune valeur n'est fournie pour **resource-stickiness**, le clone utilisera une valeur de 1. Comme il s'agit d'une petite valeur, elle perturbe peu les calculs de score des autres ressources, mais elle est suffisante pour empêcher Pacemaker de déplacer inutilement des copies dans le cluster. Pour plus d'informations sur la définition de la méta-option de ressource **resource-stickiness**, voir Configuration des méta-options de ressource.

18.2. CONFIGURATION DES CONTRAINTES DE RESSOURCES DES CLONES

Dans la plupart des cas, un clone aura une seule copie sur chaque nœud actif de la grappe. Vous pouvez toutefois définir **clone-max** pour le clone de ressources à une valeur inférieure au nombre total de nœuds dans le cluster. Dans ce cas, vous pouvez indiquer les nœuds auxquels le cluster doit

préférentiellement affecter des copies à l'aide de contraintes d'emplacement des ressources. Ces contraintes sont écrites de la même manière que pour les ressources ordinaires, à l'exception de l'utilisation de l'identifiant du clone.

La commande suivante crée une contrainte d'emplacement pour le cluster afin d'affecter de préférence le clone de ressources **webfarm-clone** à **node1**.

```
# pcs constraint location webfarm-clone prefers node1
```

Les contraintes d'ordre se comportent légèrement différemment pour les clones. Dans l'exemple ci-dessous, comme l'option de clonage **interleave** est laissée par défaut en tant que **false**, aucune instance de **webfarm-stats** ne démarrera avant que toutes les instances de **webfarm-clone** qui doivent être démarrées ne l'aient été. Ce n'est que si aucune copie de **webfarm-clone** ne peut être démarrée que **webfarm-stats** sera empêché d'être actif. En outre, **webfarm-clone** attendra que **webfarm-stats** soit arrêté avant de s'arrêter lui-même.

```
# pcs constraint order start webfarm-clone then webfarm-stats
```

La colocation d'une ressource régulière (ou d'un groupe) avec un clone signifie que la ressource peut s'exécuter sur n'importe quelle machine ayant une copie active du clone. Le cluster choisira une copie en fonction de l'endroit où le clone s'exécute et des préférences de localisation de la ressource.

La colocalisation entre clones est également possible. Dans ce cas, l'ensemble des emplacements autorisés pour le clone est limité aux nœuds sur lesquels le clone est (ou sera) actif. L'allocation s'effectue alors normalement.

La commande suivante crée une contrainte de colocalisation pour garantir que la ressource **webfarm-stats** s'exécute sur le même nœud qu'une copie active de **webfarm-clone**.

```
# pcs constraint colocation add webfarm-stats with webfarm-clone
```

18.3. RESSOURCES CLONALES PROMOUVABLES

Les ressources clones promouvables sont des ressources clones dont le méta-attribut **promotable** est défini sur **true**. Elles permettent aux instances d'être dans l'un des deux modes de fonctionnement, appelés **promoted** et **unpromoted**. Les noms des modes n'ont pas de signification particulière, à l'exception de la limitation selon laquelle, lorsqu'une instance est démarrée, elle doit se trouver dans l'état **Unpromoted**. Remarque : les noms des rôles Promoted et Unpromoted sont l'équivalent fonctionnel des rôles Master et Slave Pacemaker dans les versions précédentes de RHEL.

18.3.1. Créer une ressource clonale promouvable

Vous pouvez créer une ressource en tant que clone promouvable à l'aide de la commande suivante.

```
pcs resource create resource_id [standard:[provider:]]type [resource options] promotable [clone_id]
[clone options]
```

Par défaut, le nom du clone promouvable sera ***resource_id*-clone**.

Vous pouvez définir un nom personnalisé pour le clone en spécifiant une valeur pour l'option *clone_id*.

Vous pouvez également créer une ressource promouvable à partir d'une ressource ou d'un groupe de ressources créé précédemment, à l'aide de la commande suivante.

```
pcs resource promotable resource_id [clone_id] [clone options]
```

Par défaut, le nom du clone promouvable sera ***resource_id*-clone** ou ***group_name*-clone**.

Vous pouvez définir un nom personnalisé pour le clone en spécifiant une valeur pour l'option *clone_id*.

Le tableau suivant décrit les options de clonage supplémentaires que vous pouvez spécifier pour une ressource promouvable.

Tableau 18.2. Options de clonage supplémentaires disponibles pour les clones promouvables

Field	Description
promoted-max	Combien de copies de la ressource peuvent être promues ; par défaut 1.
promoted-node-max	Nombre de copies de la ressource pouvant être promues sur un seul nœud ; par défaut 1.

18.3.2. Configurer les contraintes de ressources promouvables

Dans la plupart des cas, une ressource promouvable aura une seule copie sur chaque nœud actif du cluster. Si ce n'est pas le cas, vous pouvez indiquer à quels nœuds le cluster doit préférentiellement assigner des copies avec des contraintes de localisation de la ressource. Ces contraintes sont écrites de la même manière que celles qui s'appliquent aux ressources ordinaires.

Vous pouvez créer une contrainte de colocalisation qui spécifie si les ressources fonctionnent dans un rôle promu ou non promu. La commande suivante crée une contrainte de colocalisation des ressources.

```
pcs constraint colocation add [promoted|unpromoted] source_resource with [promoted|unpromoted]
target_resource [score] [options]
```

Pour plus d'informations sur les contraintes de colocation, voir Colocalisation des ressources du cluster.

Lorsque vous configurez une contrainte de classement qui inclut des ressources promouvables, l'une des actions que vous pouvez spécifier pour les ressources est **promote**, indiquant que la ressource est promue d'un rôle non promu à un rôle promu. En outre, vous pouvez spécifier l'action **demote**, qui indique que la ressource doit être rétrogradée d'un rôle promu à un rôle non promu.

La commande permettant de configurer une contrainte de commande est la suivante.

```
pcs constraint order [action] resource_id then [action] resource_id [options]
```

Pour plus d'informations sur les contraintes relatives à l'ordre des ressources, voir Détermination de l'ordre d'exécution des ressources de la grappe.

18.4. RÉTROGRADATION D'UNE RESSOURCE PROMUE EN CAS D'ÉCHEC

Vous pouvez configurer une ressource promouvable de telle sorte que lorsqu'une action **promote** ou **monitor** échoue pour cette ressource, ou que la partition dans laquelle la ressource s'exécute perd le quorum, la ressource sera rétrogradée mais ne sera pas complètement arrêtée. Cela permet d'éviter une intervention manuelle dans des situations où l'arrêt complet de la ressource le nécessiterait.

- Pour configurer une ressource promouvable afin qu'elle soit rétrogradée en cas d'échec de l'action **promote**, définissez la méta-option de l'opération **on-fail** sur **demote**, comme dans l'exemple suivant.

```
# pcs resource op add my-rsc promote on-fail="demote"
```

- Pour configurer une ressource promouvable afin qu'elle soit rétrogradée en cas d'échec de l'action **monitor**, attribuez à **interval** une valeur non nulle, attribuez à la méta-option **on-fail** la valeur **demote** et attribuez à **role** la valeur **Promoted**, comme dans l'exemple suivant.

```
# pcs resource op add my-rsc monitor interval="10s" on-fail="demote"
role="Promoted"
```

- Pour configurer un cluster de manière à ce que, lorsqu'une partition du cluster perd le quorum, toutes les ressources promues soient rétrogradées mais laissées en fonctionnement et que toutes les autres ressources soient arrêtées, définissez la propriété du cluster **no-quorum-policy** sur **demote**

La définition du méta-attribut **on-fail** à **demote** pour une opération n'affecte pas la manière dont la promotion d'une ressource est déterminée. Si le nœud concerné a toujours le score de promotion le plus élevé, il sera sélectionné pour être promu à nouveau.

CHAPITRE 19. GESTION DES NŒUDS DE LA GRAPPE

Il existe une variété de commandes **pcs** que vous pouvez utiliser pour gérer les nœuds de cluster, y compris des commandes pour démarrer et arrêter les services de cluster et pour ajouter et supprimer des nœuds de cluster.

19.1. ARRÊT DES SERVICES DE CLUSTER

La commande suivante arrête les services de cluster sur le ou les nœuds spécifiés. Comme pour la commande **pcs cluster start**, l'option **--all** arrête les services de cluster sur tous les nœuds et si vous ne spécifiez aucun nœud, les services de cluster sont arrêtés sur le nœud local uniquement.

```
pcs cluster stop [--all | node] [...]
```

Vous pouvez forcer l'arrêt des services de cluster sur le nœud local à l'aide de la commande suivante, qui exécute une commande **kill -9**.

```
pcs cluster kill
```

19.2. ACTIVATION ET DÉSACTIVATION DES SERVICES DE CLUSTER

Activez les services de cluster à l'aide de la commande suivante. Cette commande configure les services de cluster pour qu'ils s'exécutent au démarrage sur le ou les nœuds spécifiés.

L'activation permet aux nœuds de rejoindre automatiquement la grappe après avoir été clôturés, ce qui réduit la durée pendant laquelle la grappe n'est pas au maximum de ses capacités. Si les services de cluster ne sont pas activés, un administrateur peut rechercher manuellement ce qui n'a pas fonctionné avant de démarrer manuellement les services de cluster, de sorte que, par exemple, un nœud présentant des problèmes matériels ne soit pas autorisé à revenir dans le cluster alors qu'il est susceptible de tomber à nouveau en panne.

- Si vous spécifiez l'option **--all**, la commande active les services de cluster sur tous les nœuds.
- Si vous ne spécifiez aucun nœud, les services de cluster sont activés sur le nœud local uniquement.

```
pcs cluster enable [--all | node] [...]
```

Utilisez la commande suivante pour configurer les services de cluster afin qu'ils ne s'exécutent pas au démarrage sur le ou les nœuds spécifiés.

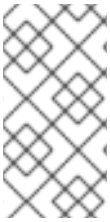
- Si vous spécifiez l'option **--all**, la commande désactive les services de cluster sur tous les nœuds.
- Si vous ne spécifiez aucun nœud, les services de cluster sont désactivés sur le nœud local uniquement.

```
pcs cluster disable [--all | node] [...]
```

19.3. AJOUT DE NŒUDS DE CLUSTER

Ajoutez un nouveau nœud à un cluster existant en suivant la procédure suivante.

Cette procédure permet d'ajouter des nœuds de grappes standard exécutant **corosync**. Pour plus d'informations sur l'intégration de nœuds non corosync dans une grappe, voir Intégration de nœuds non corosync dans une grappe : le service pacemaker_remote.



NOTE

Il est recommandé d'ajouter des nœuds aux clusters existants uniquement pendant une fenêtre de maintenance de la production. Cela vous permet d'effectuer les tests de ressources et de déploiement appropriés pour le nouveau nœud et sa configuration de clôture.

Dans cet exemple, les nœuds de cluster existants sont **clusternode-01.example.com**, **clusternode-02.example.com** et **clusternode-03.example.com**. Le nouveau nœud est **newnode.example.com**.

Procédure

Sur le nouveau nœud à ajouter au cluster, effectuez les tâches suivantes.

1. Installez les paquets de la grappe. Si le cluster utilise SBD, le gestionnaire de tickets Booth ou un dispositif de quorum, vous devez également installer manuellement les paquets correspondants (**sbd**, **booth-site**, **corosync-qdevice**) sur le nouveau nœud.

```
[root@newnode ~]# dnf install -y pcs fence-agents-all
```

   En plus des paquets de cluster, vous devrez également installer et configurer tous les services que vous exécutez dans le cluster et que vous avez installés sur les nœuds de cluster existants. Par exemple, si vous exécutez un serveur HTTP Apache dans un cluster Red Hat à haute disponibilité, vous devrez installer le serveur sur le nœud que vous ajoutez, ainsi que l'outil **wget** qui vérifie l'état du serveur.

2. Si vous exécutez le démon **firewalld**, exécutez les commandes suivantes pour activer les ports requis par le module complémentaire de haute disponibilité de Red Hat.

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
```

3. Définissez un mot de passe pour l'ID utilisateur **hacluster**. Il est recommandé d'utiliser le même mot de passe pour chaque nœud du cluster.

```
[root@newnode ~]# passwd hacluster
Changing password for user hacluster.
New password:
Retype new password:
passwd: all authentication tokens updated successfully.
```

4. Exécutez les commandes suivantes pour démarrer le service **pcsd** et pour activer **pcsd** au démarrage du système.

```
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

Sur un nœud du cluster existant, effectuez les tâches suivantes.

1. Authentifier l'utilisateur **hacluster** sur le nouveau nœud de cluster.

```
[root@clusternode-01 ~]# pcs host auth newnode.example.com
Username: hacluster
Password:
newnode.example.com: Authorized
```

2. Ajoutez le nouveau nœud au cluster existant. Cette commande synchronise également le fichier de configuration du cluster **corosync.conf** avec tous les nœuds du cluster, y compris le nouveau nœud que vous ajoutez.

```
[root@clusternode-01 ~]# pcs cluster node add newnode.example.com
```

Sur le nouveau nœud à ajouter au cluster, effectuez les tâches suivantes.

1. Démarrez et activez les services de cluster sur le nouveau nœud.

```
[root@newnode ~]# pcs cluster start
Starting Cluster...
[root@newnode ~]# pcs cluster enable
```

2. Veillez à configurer et à tester un dispositif de clôture pour le nouveau nœud de cluster.

19.4. SUPPRESSION DE NŒUDS DE CLUSTER

La commande suivante arrête le nœud spécifié et le supprime du fichier de configuration de la grappe, **corosync.conf**, sur tous les autres nœuds de la grappe.

```
suppression d'un nœud de cluster pcs node
```

19.5. AJOUT D'UN NŒUD À UNE GRAPPE AVEC PLUSIEURS LIENS

Lors de l'ajout d'un nœud à un cluster comportant plusieurs liens, vous devez spécifier des adresses pour tous les liens.

L'exemple suivant ajoute le nœud **rh80-node3** à un cluster, en spécifiant l'adresse IP 192.168.122.203 pour le premier lien et 192.168.123.203 pour le second lien.

```
# pcs cluster node add rh80-node3 addr=192.168.122.203 addr=192.168.123.203
```

19.6. AJOUT ET MODIFICATION DE LIENS DANS UN CLUSTER EXISTANT

Dans la plupart des cas, vous pouvez ajouter ou modifier les liens dans un cluster existant sans redémarrer le cluster.

19.6.1. Ajout et suppression de liens dans un cluster existant

Pour ajouter un nouveau lien à un cluster en cours d'exécution, utilisez la commande **pcs cluster link add**.

- Lors de l'ajout d'un lien, vous devez spécifier une adresse pour chaque nœud.

- L'ajout et la suppression d'un lien ne sont possibles que si vous utilisez le protocole de transport **knet**.
- Au moins un lien dans le cluster doit être défini à tout moment.
- Le nombre maximum de liens dans un groupe est de 8, numérotés de 0 à 7. Les liens définis n'ont pas d'importance, vous pouvez donc, par exemple, définir uniquement les liens 3, 6 et 7.
- Lorsque vous ajoutez un lien sans spécifier son numéro, **pcs** utilise le lien le plus bas disponible.
- Les numéros des liens actuellement configurés sont contenus dans le fichier **corosync.conf**. Pour afficher le fichier **corosync.conf**, exécutez la commande **pcs cluster corosync** ou **pcs cluster config show**.

La commande suivante ajoute le lien numéro 5 à une grappe de trois nœuds.

```
[root@node1 ~] # pcs cluster link add node1=10.0.5.11 node2=10.0.5.12 node3=10.0.5.31
options linknumber=5
```

Pour supprimer un lien existant, utilisez la commande **pcs cluster link delete** ou **pcs cluster link remove**. L'une ou l'autre des commandes suivantes supprimera le lien numéro 5 du cluster.

```
[root@node1 ~] # pcs cluster link delete 5

[root@node1 ~] # pcs cluster link remove 5
```

19.6.2. Modification d'un lien dans une grappe à liens multiples

S'il existe plusieurs liens dans le cluster et que vous souhaitez modifier l'un d'entre eux, suivez la procédure suivante.

Procédure

1. Supprimez le lien que vous souhaitez modifier.

```
[root@node1 ~] # pcs cluster link remove 2
```

2. Ajoutez le lien vers le cluster avec les adresses et options mises à jour.

```
[root@node1 ~] # pcs cluster link add node1=10.0.5.11 node2=10.0.5.12
node3=10.0.5.31 options linknumber=2
```

19.6.3. Modifier les adresses des liens dans un cluster avec un seul lien

Si votre cluster n'utilise qu'un seul lien et que vous souhaitez modifier ce lien pour utiliser des adresses différentes, suivez la procédure suivante. Dans cet exemple, le lien d'origine est le lien 1.

1. Ajouter un nouveau lien avec les nouvelles adresses et options.

```
[root@node1 ~] # pcs cluster link add node1=10.0.5.11 node2=10.0.5.12
node3=10.0.5.31 options linknumber=2
```

2. Supprimer le lien original.

```
[root@node1 ~] # pcs cluster link remove 1
```

Notez que vous ne pouvez pas spécifier des adresses en cours d'utilisation lorsque vous ajoutez des liens à une grappe. Cela signifie, par exemple, que si vous avez un cluster à deux nœuds avec un lien et que vous souhaitez modifier l'adresse d'un seul nœud, vous ne pouvez pas utiliser la procédure ci-dessus pour ajouter un nouveau lien qui spécifie une nouvelle adresse et une adresse existante. Au lieu de cela, vous pouvez ajouter un lien temporaire avant de supprimer le lien existant et de le rajouter avec l'adresse mise à jour, comme dans l'exemple suivant.

Dans cet exemple :

- La liaison pour la grappe existante est la liaison 1, qui utilise l'adresse 10.0.5.11 pour le nœud 1 et l'adresse 10.0.5.12 pour le nœud 2.
- Vous souhaitez modifier l'adresse du nœud 2 en 10.0.5.31.

Procédure

Pour mettre à jour une seule des adresses d'un cluster à deux nœuds avec un seul lien, utilisez la procédure suivante.

1. Ajouter un nouveau lien temporaire au cluster existant, en utilisant des adresses qui ne sont pas actuellement utilisées.

```
[root@node1 ~] # pcs cluster link add node1=10.0.5.13 node2=10.0.5.14 options linknumber=2
```

2. Supprimer le lien original.

```
[root@node1 ~] # pcs cluster link remove 1
```

3. Ajouter le nouveau lien modifié.

```
[root@node1 ~] # pcs cluster link add node1=10.0.5.11 node2=10.0.5.31 options linknumber=1
```

4. Supprimez le lien temporaire que vous avez créé

```
[root@node1 ~] # pcs cluster link remove 2
```

19.6.4. Modifier les options d'un lien dans un cluster avec un seul lien

Si votre cluster n'utilise qu'un seul lien et que vous souhaitez modifier les options de ce lien sans changer l'adresse à utiliser, vous pouvez ajouter un lien temporaire avant de supprimer et de mettre à jour le lien à modifier.

Dans cet exemple :

- La liaison pour la grappe existante est la liaison 1, qui utilise l'adresse 10.0.5.11 pour le nœud 1 et l'adresse 10.0.5.12 pour le nœud 2.
- Vous souhaitez modifier l'option de lien **link_priority** en 11.

Procédure

Modifiez l'option de lien dans un cluster avec un seul lien en suivant la procédure suivante.

1. Ajouter un nouveau lien temporaire au cluster existant, en utilisant des adresses qui ne sont pas actuellement utilisées.

```
[root@node1 ~] # pcs cluster link add node1=10.0.5.13 node2=10.0.5.14 options linknumber=2
```

2. Supprimer le lien original.

```
[root@node1 ~] # pcs cluster link remove 1
```

3. Ajoutez le lien original avec les options mises à jour.

```
[root@node1 ~] # pcs cluster link add node1=10.0.5.11 node2=10.0.5.12 options linknumber=1 link_priority=11
```

4. Supprimer le lien temporaire.

```
[root@node1 ~] # pcs cluster link remove 2
```

19.6.5. La modification d'un lien lors de l'ajout d'un nouveau lien n'est pas possible

Si, pour une raison quelconque, l'ajout d'un nouveau lien n'est pas possible dans votre configuration et que votre seule option consiste à modifier un seul lien existant, vous pouvez utiliser la procédure suivante, qui nécessite l'arrêt de votre cluster.

Procédure

L'exemple de procédure suivant met à jour le lien numéro 1 dans le cluster et définit l'option **link_priority** pour le lien à 11.

1. Arrêtez les services de cluster pour le cluster.

```
[root@node1 ~] # pcs cluster stop --all
```

2. Mettre à jour les adresses et les options des liens.
La commande **pcs cluster link update** n'exige pas que vous spécifiiez toutes les adresses et options des nœuds. Au lieu de cela, vous pouvez spécifier uniquement les adresses à modifier. Cet exemple modifie les adresses de **node1** et **node3** et l'option **link_priority** uniquement.

```
[root@node1 ~] # pcs cluster link update 1 node1=10.0.5.11 node3=10.0.5.31 options link_priority=11
```

Pour supprimer une option, vous pouvez lui attribuer une valeur nulle à l'aide de la commande ***option*=** format.

3. Redémarrer le cluster

```
[root@node1 ~] # pcs cluster start --all
```

19.7. CONFIGURATION D'UNE STRATÉGIE DE SANTÉ DES NŒUDS

Un nœud peut fonctionner suffisamment bien pour conserver son statut de membre d'une grappe, mais être en mauvaise santé à certains égards, ce qui en fait un emplacement indésirable pour les ressources. Par exemple, une unité de disque peut signaler des erreurs SMART ou le processeur peut être très chargé. Depuis RHEL 9.1, vous pouvez utiliser une stratégie de santé des nœuds dans Pacemaker pour déplacer automatiquement les ressources hors des nœuds malsains.

Vous pouvez contrôler l'état de santé d'un nœud à l'aide des agents de ressources de nœuds de santé suivants, qui définissent les attributs du nœud en fonction de l'état de l'unité centrale et du disque :

- **ocf:pacemaker:HealthCPU**qui surveille le fonctionnement au ralenti de l'unité centrale
- **ocf:pacemaker:HealthIOWait**qui surveille l'attente E/S de l'unité centrale
- **ocf:pacemaker:HealthSMART**qui surveille l'état SMART d'un lecteur de disque
- **ocf:pacemaker:SysInfo**qui définit une série d'attributs de nœuds à l'aide d'informations sur le système local et fonctionne également comme un agent de santé qui surveille l'utilisation de l'espace disque

En outre, tout agent de ressource peut fournir des attributs de nœud qui peuvent être utilisés pour définir une stratégie de nœud de santé.

Procédure

La procédure suivante permet de configurer une stratégie de santé des nœuds pour une grappe, afin de retirer des ressources à tout nœud dont le temps d'attente E/S du processeur dépasse 15 %.

1. La propriété **health-node-strategy** cluster permet de définir la manière dont Pacemaker réagit aux modifications de l'état des nœuds.

   ```
   # pcs property set node-health-strategy=migrate-on-red
   ```

2. Créez une ressource cluster clonée qui utilise un agent de ressource de nœud de santé, en définissant l'option méta de ressource **allow-unhealthy-nodes** pour définir si le cluster détectera si la santé du nœud se rétablit et déplacera à nouveau les ressources vers le nœud. Configurez cette ressource avec une action de surveillance récurrente, afin de vérifier en permanence l'état de santé de tous les nœuds.
   Cet exemple crée un agent de ressources **HealthIOWait** pour surveiller l'attente E/S de l'unité centrale, en fixant à 15 % la limite rouge pour le déplacement des ressources hors d'un nœud. Cette commande définit l'option de méta ressource **allow-unhealthy-nodes** sur **true** et configure un intervalle de surveillance récurrent de 10 secondes.

   ```
   # pcs resource create io-monitor ocf:pacemaker:HealthIOWait red_limit=15 op monitor interval=10s meta allow-unhealthy-nodes=true clone
   ```

19.8. CONFIGURATION D'UN GRAND CLUSTER AVEC DE NOMBREUSES RESSOURCES

Si le cluster que vous déployez comprend un grand nombre de nœuds et de ressources, vous devrez peut-être modifier les valeurs par défaut des paramètres suivants pour votre cluster.

La propriété de la grappe cluster-ipc-limit

La propriété de cluster **cluster-ipc-limit** est l'accumulation maximale de messages IPC avant qu'un démon de cluster n'en déconnecte un autre. Lorsqu'un grand nombre de ressources sont nettoyées ou modifiées simultanément dans un grand cluster, un grand nombre de mises à jour CIB arrivent en

même temps. Cela peut entraîner l'expulsion des clients les plus lents si le service Pacemaker n'a pas le temps de traiter toutes les mises à jour de configuration avant que le seuil de la file d'attente d'événements CIB ne soit atteint.
La valeur recommandée de **cluster-ipc-limit** pour les grandes grappes est le nombre de ressources de la grappe multiplié par le nombre de nœuds. Cette valeur peut être augmentée si vous voyez dans les journaux des messages "Evicting client" pour les PID des démons de la grappe.

Vous pouvez augmenter la valeur de **cluster-ipc-limit** par rapport à sa valeur par défaut de 500 à l'aide de la commande **pcs property set**. Par exemple, pour un cluster de dix nœuds avec 200 ressources, vous pouvez définir la valeur de **cluster-ipc-limit** à 2000 avec la commande suivante.

```
# pcs property set cluster-ipc-limit=2000
```

Le paramètre PCMK_ipc_buffer Pacemaker

Dans le cas de déploiements très importants, les messages internes de Pacemaker peuvent dépasser la taille de la mémoire tampon. Lorsque cela se produit, vous verrez un message dans les journaux du système au format suivant :

```
Le message compressé dépasse X% de la limite IPC configurée (X bytes) ; envisager de régler PCMK_ipc_buffer sur X ou plus
```

Lorsque ce message apparaît, vous pouvez augmenter la valeur de **PCMK_ipc_buffer** dans le fichier de configuration **/etc/sysconfig/pacemaker** sur chaque nœud. Par exemple, pour augmenter la valeur de **PCMK_ipc_buffer** de sa valeur par défaut à 13396332 octets, modifiez le champ non commenté **PCMK_ipc_buffer** dans le fichier **/etc/sysconfig/pacemaker** sur chaque nœud du cluster comme suit.

```
PCMK_ipc_buffer=13396332
```

Pour appliquer cette modification, exécutez la commande suivante.

```
# systemctl restart pacemaker
```

CHAPITRE 20. DÉFINITION DES AUTORISATIONS DES UTILISATEURS POUR UN CLUSTER PACEMAKER

Vous pouvez autoriser des utilisateurs spécifiques autres que l'utilisateur **hacluster** à gérer un cluster Pacemaker. Il existe deux types d'autorisations que vous pouvez accorder à des utilisateurs individuels :

- Permissions permettant aux utilisateurs individuels de gérer la grappe via l'interface Web et d'exécuter les commandes **pcs** qui se connectent aux nœuds sur un réseau. Les commandes qui se connectent aux nœuds sur un réseau comprennent les commandes permettant de configurer une grappe ou d'ajouter ou de supprimer des nœuds d'une grappe.

- Permissions pour les utilisateurs locaux afin d'autoriser l'accès en lecture seule ou en lecture-écriture à la configuration de la grappe. Les commandes qui ne nécessitent pas de connexion réseau comprennent les commandes qui modifient la configuration de la grappe, telles que celles qui créent des ressources et configurent des contraintes.

Dans les cas où les deux ensembles d'autorisations ont été attribués, les autorisations pour les commandes qui se connectent sur un réseau sont appliquées en premier, puis les autorisations pour la modification de la configuration de la grappe sur le nœud local sont appliquées. La plupart des commandes **pcs** ne nécessitent pas d'accès au réseau et, dans ce cas, les autorisations réseau ne s'appliquent pas.

20.1. DÉFINITION DES AUTORISATIONS D'ACCÈS AUX NŒUDS SUR UN RÉSEAU

Pour autoriser des utilisateurs spécifiques à gérer la grappe via l'interface Web et à exécuter des commandes **pcs** qui se connectent aux nœuds via un réseau, ajoutez ces utilisateurs au groupe **haclient**. Cette opération doit être effectuée sur chaque nœud de la grappe.

20.2. DÉFINITION DES AUTORISATIONS LOCALES À L'AIDE D'ACL

Vous pouvez utiliser la commande **pcs acl** pour définir des autorisations pour les utilisateurs locaux afin de permettre un accès en lecture seule ou en lecture-écriture à la configuration du cluster en utilisant des listes de contrôle d'accès (ACL).

Par défaut, les ACLS ne sont pas activés. Lorsque les ACLS ne sont pas activées, tout utilisateur membre du groupe **haclient** sur tous les nœuds dispose d'un accès local complet en lecture/écriture à la configuration de la grappe, tandis que les utilisateurs qui ne sont pas membres du groupe **haclient** n'y ont pas accès. Cependant, lorsque les ACL sont activées, même les utilisateurs membres du groupe **haclient** n'ont accès qu'à ce qui leur a été accordé par les ACL. Les comptes d'utilisateur root et **hacluster** ont toujours un accès complet à la configuration de la grappe, même lorsque les listes de contrôle d'accès sont activées.

La définition des autorisations pour les utilisateurs locaux se fait en deux étapes :

1. Exécutez la commande **pcs acl role create...** pour créer un *role* qui définit les autorisations pour ce rôle.

2. Attribuez le rôle que vous avez créé à un utilisateur à l'aide de la commande **pcs acl user create**. Si vous attribuez plusieurs rôles au même utilisateur, l'autorisation **deny** est prioritaire, puis **write**, puis **read**.

Procédure

L'exemple de procédure suivant permet à un utilisateur local nommé **rouser** d'accéder en lecture seule à la configuration d'un cluster. Notez qu'il est également possible de restreindre l'accès à certaines parties de la configuration.

AVERTISSEMENT

Il est important d'effectuer cette procédure en tant que root ou de sauvegarder toutes les mises à jour de configuration dans un fichier de travail que vous pourrez ensuite transférer dans le CIB actif lorsque vous aurez terminé. Dans le cas contraire, vous risquez de ne pas pouvoir effectuer d'autres modifications. Pour plus d'informations sur l'enregistrement des mises à jour de configuration dans un fichier de travail, voir Enregistrement d'une modification de configuration dans un fichier de travail.

1. Cette procédure nécessite que l'utilisateur **rouser** existe sur le système local et que l'utilisateur **rouser** soit membre du groupe **haclient**.

```
# adduser rouser
# usermod -a -G haclient rouser
```

2. Activez les ACL de Pacemaker avec la commande **pcs acl enable**.

```
# pcs acl enable
```

3. Créez un rôle nommé **read-only** avec des permissions de lecture seule pour le cib.

```
# pcs acl role create read-only description="Read access to cluster" read xpath /cib
```

4. Créez l'utilisateur **rouser** dans le système pcs ACL et attribuez-lui le rôle **read-only**.

```
# pcs acl user create rouser read-only
```

5. Visualiser les ACL en cours.

```
# pcs acl
User: rouser
  Roles: read-only
Role: read-only
  Description: Read access to cluster
  Permission: read xpath /cib (read-only-read)
```

6. Sur chaque nœud où **rouser** exécutera les commandes **pcs**, connectez-vous en tant que **rouser** et authentifiez-vous auprès du service local **pcsd**. Cela est nécessaire pour exécuter certaines commandes **pcs**, telles que **pcs status**, en tant qu'utilisateur ACL.

```
[rouser ~]$ pcs client local-auth
```

CHAPITRE 21. OPÉRATIONS DE SURVEILLANCE DES RESSOURCES

Pour vous assurer que les ressources restent saines, vous pouvez ajouter une opération de surveillance à la définition d'une ressource. Si vous ne spécifiez pas d'opération de surveillance pour une ressource, la commande **pcs** créera par défaut une opération de surveillance, avec un intervalle déterminé par l'agent de ressources. Si l'agent de ressources ne fournit pas d'intervalle de surveillance par défaut, la commande pcs créera une opération de surveillance avec un intervalle de 60 secondes.

Le tableau suivant résume les propriétés d'une opération de contrôle des ressources.

Tableau 21.1. Propriétés d'une opération

Field	Description
id	Nom unique de l'action. Le système l'attribue lorsque vous configurez une opération.
name	L'action à effectuer. Valeurs courantes : **monitor**, **start**, **stop**
interval	Si la valeur est différente de zéro, une opération récurrente est créée et se répète à cette fréquence, en secondes. Une valeur non nulle n'a de sens que lorsque l'action **name** est définie sur **monitor**. Une action de contrôle récurrente est exécutée immédiatement après le démarrage d'une ressource, et les actions de contrôle suivantes sont programmées à partir de l'heure à laquelle l'action de contrôle précédente s'est achevée. Par exemple, si une action de contrôle avec **interval=20s** est exécutée à 01:00:00, l'action de contrôle suivante ne se produit pas à 01:00:20, mais 20 secondes après la fin de la première action de contrôle. S'il est fixé à zéro, ce qui est la valeur par défaut, ce paramètre vous permet de fournir des valeurs à utiliser pour les opérations créées par le cluster. Par exemple, si **interval** est défini à zéro, que **name** de l'opération est défini à **start** et que la valeur **timeout** est définie à 40, Pacemaker utilisera un délai d'attente de 40 secondes lors du démarrage de cette ressource. Une opération **monitor** avec un intervalle de zéro vous permet de définir les valeurs **timeout**/**on-fail**/**enabled** pour les sondes que Pacemaker effectue au démarrage afin d'obtenir l'état actuel de toutes les ressources lorsque les valeurs par défaut ne sont pas souhaitables.
timeout	Si l'opération ne se termine pas dans le délai fixé par ce paramètre, elle est interrompue et considérée comme ayant échoué. La valeur par défaut est la valeur de **timeout** si elle est définie avec la commande **pcs resource op defaults**, ou 20 secondes si elle n'est pas définie. Si vous constatez que votre système comprend une ressource qui nécessite plus de temps que le système ne le permet pour effectuer une opération (telle que **start**, **stop**, ou **monitor**), recherchez-en la cause et si le temps d'exécution prolongé est prévu, vous pouvez augmenter la valeur de ce paramètre. La valeur **timeout** n'est pas un délai de quelque nature que ce soit, et le cluster n'attend pas la totalité de la période de temporisation si l'opération revient avant la fin de la période de temporisation.

Field	Description
on-fail	L'action à entreprendre si cette action échoue. Valeurs autorisées : * **ignore** - Faire comme si la ressource n'avait pas échoué * **block** - Ne pas effectuer d'autres opérations sur la ressource * **stop** - Arrêter la ressource et ne pas la lancer ailleurs * **restart** - Arrêter la ressource et la relancer (éventuellement sur un autre nœud) * **fence** - STONITH le nœud sur lequel la ressource a échoué * **standby** - Déplacer les ressources*all* loin du nœud sur lequel la ressource a échoué * **migrate** - Migrer la ressource vers un autre nœud, si possible. Cela équivaut à donner la valeur 1 à l'option **migration-threshold** resource meta. * **demote** - Lorsqu'une action **promote** échoue pour la ressource, celle-ci est rétrogradée mais n'est pas complètement arrêtée. Lorsqu'une action **monitor** échoue pour une ressource, si **interval** a une valeur non nulle et que**role** a une valeur égale à **Promoted**, la ressource est rétrogradée mais n'est pas totalement arrêtée. La valeur par défaut de l'opération **stop** est **fence** lorsque l'option STONITH est activée et **block** dans le cas contraire. Pour toutes les autres opérations, la valeur par défaut est **restart**.
enabled	Si **false**, l'opération est traitée comme si elle n'existait pas. Valeurs autorisées : **true**, **false**

21.1. CONFIGURATION DES OPÉRATIONS DE SURVEILLANCE DES RESSOURCES

Vous pouvez configurer les opérations de surveillance lorsque vous créez une ressource à l'aide de la commande suivante.

```
pcs resource create resource_id standard:provider:type|type [resource_options] [op operation_action operation_options [operation_type operation_options ]...]
```

Par exemple, la commande suivante crée une ressource **IPaddr2** avec une opération de surveillance. La nouvelle ressource s'appelle **VirtualIP** et possède une adresse IP de 192.168.0.99 et un masque de réseau de 24 sur **eth2**. Une opération de surveillance sera effectuée toutes les 30 secondes.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.99 cidr_netmask=24 nic=eth2 op monitor interval=30s
```

Vous pouvez également ajouter une opération de surveillance à une ressource existante à l'aide de la commande suivante.

```
pcs resource op add resource_id operation_action [operation_properties]
```

La commande suivante permet de supprimer une opération de ressource configurée.

```
pcs resource op remove resource_id operation_name operation_properties
```

NOTE

Vous devez spécifier les propriétés exactes de l'opération pour supprimer correctement une opération existante.

Pour modifier les valeurs d'une option de surveillance, vous pouvez mettre à jour la ressource. Par exemple, vous pouvez créer un site **VirtualIP** à l'aide de la commande suivante.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.99 cidr_netmask=24
nic=eth2
```

Par défaut, cette commande crée ces opérations.

```
Operations: start interval=0s timeout=20s (VirtualIP-start-timeout-20s)
            stop interval=0s timeout=20s (VirtualIP-stop-timeout-20s)
            monitor interval=10s timeout=20s (VirtualIP-monitor-interval-10s)
```

Pour modifier le délai d'arrêt, exécutez la commande suivante.

```
# pcs resource update VirtualIP op stop interval=0s timeout=40s

# pcs resource config VirtualIP
 Resource: VirtualIP (class=ocf provider=heartbeat type=IPaddr2)
  Attributes: ip=192.168.0.99 cidr_netmask=24 nic=eth2
  Operations: start interval=0s timeout=20s (VirtualIP-start-timeout-20s)
              monitor interval=10s timeout=20s (VirtualIP-monitor-interval-10s)
              stop interval=0s timeout=40s (VirtualIP-name-stop-interval-0s-timeout-40s)
```

21.2. CONFIGURATION DES VALEURS PAR DÉFAUT DES OPÉRATIONS SUR LES RESSOURCES GLOBALES

Vous pouvez modifier la valeur par défaut d'une opération sur une ressource pour toutes les ressources à l'aide de la commande **pcs resource op defaults update**.

La commande suivante définit une valeur globale par défaut de **timeout** de 240 secondes pour toutes les opérations de surveillance.

```
# pcs resource op defaults update timeout=240s
```

La commande **pcs resource op defaults *name=value*** qui définissait les valeurs par défaut des opérations sur les ressources pour toutes les ressources dans les versions précédentes, reste prise en charge, à moins que plusieurs valeurs par défaut ne soient configurées. Cependant, **pcs resource op defaults update** est désormais la version préférée de la commande.

21.2.1. Remplacer les valeurs des opérations spécifiques aux ressources

Notez qu'une ressource de cluster n'utilisera la valeur globale par défaut que si l'option n'est pas spécifiée dans la définition de la ressource de cluster. Par défaut, les agents de ressources définissent

l'option **timeout** pour toutes les opérations. Pour que la valeur du délai global de l'opération soit respectée, vous devez créer la ressource de cluster sans l'option **timeout** explicitement ou vous devez supprimer l'option **timeout** en mettant à jour la ressource de cluster, comme dans la commande suivante.

```
# pcs resource update VirtualIP op monitor interval=10s
```

Par exemple, après avoir défini une valeur globale par défaut de 240 secondes pour **timeout** pour toutes les opérations de surveillance et mis à jour la ressource de cluster **VirtualIP** pour supprimer la valeur de temporisation pour l'opération **monitor**, la ressource **VirtualIP** aura alors des valeurs de temporisation pour les opérations **start**, **stop**, et **monitor** de 20s, 40s et 240s, respectivement. La valeur globale par défaut pour les opérations de délai d'attente n'est appliquée ici qu'à l'opération **monitor**, où l'option par défaut **timeout** a été supprimée par la commande précédente.

```
# pcs resource config VirtualIP
 Resource: VirtualIP (class=ocf provider=heartbeat type=IPaddr2)
  Attributes: ip=192.168.0.99 cidr_netmask=24 nic=eth2
  Operations: start interval=0s timeout=20s (VirtualIP-start-timeout-20s)
              monitor interval=10s (VirtualIP-monitor-interval-10s)
              stop interval=0s timeout=40s (VirtualIP-name-stop-interval-0s-timeout-40s)
```

21.2.2. Modification de la valeur par défaut d'une opération sur les ressources pour les ensembles de ressources

Vous pouvez créer plusieurs ensembles de valeurs par défaut d'opérations sur les ressources à l'aide de la commande **pcs resource op defaults set create**, qui vous permet de spécifier une règle contenant **resource** et des expressions d'opérations. Toutes les expressions de règles prises en charge par Pacemaker sont autorisées.

Cette commande permet de configurer une valeur d'opération de ressource par défaut pour toutes les ressources d'un type particulier. Par exemple, il est désormais possible de configurer les ressources implicites **podman** créées par Pacemaker lorsque des bundles sont utilisés.

La commande suivante définit un délai d'attente par défaut de 90s pour toutes les opérations de toutes les ressources **podman**. Dans cet exemple, **::podman** désigne une ressource de n'importe quelle classe, de n'importe quel fournisseur, de type **podman**.

L'option **id**, qui désigne l'ensemble des valeurs par défaut des opérations sur les ressources, n'est pas obligatoire. Si vous ne définissez pas cette option, **pcs** génère automatiquement un identifiant. La définition de cette valeur vous permet de fournir un nom plus descriptif.

```
# pcs resource op defaults set create id=podman-timeout meta timeout=90s rule resource ::podman
```

La commande suivante définit un délai d'attente par défaut de 120 secondes pour l'opération **stop** pour toutes les ressources.

```
# pcs resource op defaults set create id=stop-timeout meta timeout=120s rule op stop
```

Il est possible de définir la valeur par défaut du délai d'attente pour une opération spécifique pour toutes les ressources d'un type particulier. L'exemple suivant définit un délai d'attente par défaut de 120 secondes pour l'opération **stop** pour toutes les ressources **podman**.

```
# pcs resource op defaults set create id=podman-stop-timeout meta timeout=120s rule
resource ::podman and op stop
```

21.2.3. Affichage des valeurs par défaut des opérations sur les ressources actuellement configurées

La commande **pcs resource op defaults** affiche une liste des valeurs par défaut actuellement configurées pour les opérations sur les ressources, y compris les règles que vous avez spécifiées.

La commande suivante affiche les valeurs d'opération par défaut pour un cluster qui a été configuré avec un délai d'attente par défaut de 90s pour toutes les opérations de toutes les ressources **podman**, et pour lequel un ID pour l'ensemble des valeurs d'opération par défaut des ressources a été défini comme **podman-timeout**.

```
# pcs resource op defaults
Meta Attrs: podman-timeout
  timeout=90s
  Rule: boolean-op=and score=INFINITY
    Expression: resource ::podman
```

La commande suivante affiche les valeurs d'opération par défaut pour un cluster qui a été configuré avec une valeur de délai par défaut de 120 secondes pour l'opération **stop** pour toutes les ressources **podman**, et pour lequel un identifiant pour l'ensemble des valeurs d'opération par défaut des ressources a été défini comme **podman-stop-timeout**.

```
# pcs resource op defaults]
Meta Attrs: podman-stop-timeout
  timeout=120s
  Rule: boolean-op=and score=INFINITY
    Expression: resource ::podman
    Expression: op stop
```

21.3. CONFIGURATION DE PLUSIEURS OPÉRATIONS DE SURVEILLANCE

Vous pouvez configurer une ressource unique avec autant d'opérations de surveillance qu'un agent de ressource en prend en charge. Vous pouvez ainsi effectuer un contrôle de santé superficiel toutes les minutes et des contrôles de plus en plus intensifs à des intervalles plus élevés.

NOTE

Lorsque vous configurez des opérations de surveillance multiples, vous devez vous assurer que deux opérations ne sont pas effectuées au même intervalle.

Pour configurer des opérations de contrôle supplémentaires pour une ressource qui prend en charge des contrôles plus approfondis à différents niveaux, vous ajoutez une option **OCF_CHECK_LEVEL=*n*** option.

Par exemple, si vous configurez la ressource **IPaddr2** suivante, celle-ci crée par défaut une opération de surveillance avec un intervalle de 10 secondes et un délai d'attente de 20 secondes.

```
# pcs resource create VirtualIP ocf:heartbeat:IPaddr2 ip=192.168.0.99 cidr_netmask=24
nic=eth2
```

Si l'IP virtuel prend en charge un contrôle différent avec une profondeur de 10, la commande suivante fait en sorte que Pacemaker effectue le contrôle de surveillance plus avancé toutes les 60 secondes en plus du contrôle normal de l'IP virtuel toutes les 10 secondes. (Comme indiqué, vous ne devez pas configurer l'opération de surveillance supplémentaire avec un intervalle de 10 secondes également)

```
# pcs resource op add VirtualIP monitor interval=60s OCF_CHECK_LEVEL=10
```

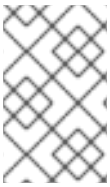
CHAPITRE 22. PROPRIÉTÉS DE LA GRAPPE DE STIMULATEURS CARDIAQUES

Les propriétés de la grappe contrôlent le comportement de la grappe lorsqu'elle est confrontée à des situations susceptibles de se produire au cours de son fonctionnement.

22.1. RÉSUMÉ DES PROPRIÉTÉS ET DES OPTIONS DE LA GRAPPE

Le tableau suivant résume les propriétés de la grappe Pacemaker, en indiquant les valeurs par défaut des propriétés et les valeurs possibles que vous pouvez définir pour ces propriétés.

D'autres propriétés des clusters déterminent le comportement des clôtures. Pour plus d'informations sur ces propriétés, voir le tableau des propriétés des clusters qui déterminent le comportement des clôtures dans Propriétés générales des dispositifs de clôture .



NOTE

Outre les propriétés décrites dans ce tableau, d'autres propriétés de la grappe sont exposées par le logiciel de la grappe. Il est recommandé de ne pas modifier les valeurs par défaut de ces propriétés.

Tableau 22.1. Propriétés de la grappe

Option	Défaut	Description
batch-limit	0	Le nombre d'actions sur les ressources que la grappe est autorisée à exécuter en parallèle. La valeur de "correct" dépend de la vitesse et de la charge de votre réseau et des nœuds de la grappe. La valeur par défaut de 0 signifie que la grappe imposera dynamiquement une limite lorsqu'un nœud a une charge de CPU élevée.
migration-limit	-1 (illimité)	Nombre de tâches de migration que le cluster est autorisé à exécuter en parallèle sur un nœud.

Option	Défaut	Description
no-quorum-policy	arrêter	Que faire lorsque le cluster n'a pas de quorum. Valeurs autorisées : * ignorer - poursuivre la gestion des ressources * freeze - continuer la gestion des ressources, mais ne pas récupérer les ressources des nœuds qui ne sont pas dans la partition affectée * stop - arrête toutes les ressources dans la partition de cluster concernée * suicide - clôturer tous les nœuds de la partition de cluster affectée * demote - si une partition de cluster perd le quorum, rétrograder les ressources promues et arrêter toutes les autres ressources
symmetric-cluster	true	Indique si les ressources peuvent être exécutées sur n'importe quel nœud par défaut.
cluster-delay	60s	Délai aller-retour sur le réseau (hors exécution de l'action). La valeur de "correct" dépendra de la vitesse et de la charge de votre réseau et de vos nœuds de cluster.
dc-deadtime	20s	Durée d'attente d'une réponse des autres nœuds lors du démarrage. La valeur de "correct" dépend de la vitesse et de la charge de votre réseau, ainsi que du type de commutateurs utilisés.
stop-orphan-resources	true	Indique si les ressources supprimées doivent être arrêtées.
stop-orphan-actions	true	Indique si les actions supprimées doivent être annulées.

Option	Défaut	Description
start-failure-is-fatal	true	Indique si l'échec du démarrage d'une ressource sur un nœud particulier empêche d'autres tentatives de démarrage sur ce nœud. Lorsque l'option est définie sur **false**, le cluster décide s'il faut réessayer de démarrer sur le même nœud en fonction du nombre d'échecs et du seuil de migration de la ressource. Pour plus d'informations sur la définition de l'option **migration-threshold** pour une ressource, voir Configuration des méta-options des ressources. En définissant **start-failure-is-fatal** sur **false**, on court le risque qu'un nœud défectueux, incapable de démarrer une ressource, bloque toutes les actions dépendantes. C'est pourquoi la valeur par défaut de **start-failure-is-fatal** est vraie. Le risque lié à la définition de **start-failure-is-fatal=false** peut être atténué en définissant un seuil de migration bas, de sorte que d'autres actions puissent se poursuivre après ce nombre d'échecs.
pe-error-series-max	-1 (tous)	Nombre d'entrées de l'ordonnanceur entraînant des ERREURS à sauvegarder. Utilisé pour signaler des problèmes.
pe-warn-series-max	-1 (tous)	Nombre d'entrées de l'ordonnanceur entraînant des AVERTISSEMENTS à sauvegarder. Utilisé pour signaler des problèmes.
pe-input-series-max	-1 (tous)	Le nombre d'entrées de l'ordonnanceur "normal" à sauvegarder. Utilisé pour signaler des problèmes.
cluster-infrastructure		Pile de messagerie sur laquelle Pacemaker fonctionne actuellement. Utilisée à des fins d'information et de diagnostic ; non configurable par l'utilisateur.
dc-version		Version de Pacemaker sur le contrôleur désigné (DC) de la grappe. Utilisée à des fins de diagnostic ; non configurable par l'utilisateur.

Option	Défaut	Description
cluster-recheck-interval	15 minutes	Pacemaker est principalement piloté par les événements et sait à l'avance quand il doit revérifier le cluster pour les délais d'échec et la plupart des règles basées sur le temps. Pacemaker revérifie également la grappe après la durée d'inactivité spécifiée par cette propriété. Cette revérification du cluster a deux objectifs : les règles avec **date-spec** sont garanties d'être vérifiées aussi souvent, et cela sert de sécurité pour certains types de bogues de l'ordonnanceur. La valeur 0 désactive cette interrogation ; les valeurs positives indiquent un intervalle de temps.
maintenance-mode	false	Le mode maintenance indique à la grappe de passer en mode "mains libres" et de ne pas démarrer ou arrêter de services jusqu'à ce qu'on lui dise le contraire. Lorsque le mode de maintenance est terminé, la grappe effectue une vérification de l'état actuel de tous les services, puis arrête ou démarre ceux qui en ont besoin.
shutdown-escalation	20 minutes	Délai au terme duquel il convient d'abandonner toute tentative d'arrêt gracieux et de se contenter de quitter le système. Utilisation avancée uniquement.
stop-all-resources	false	Le cluster doit-il arrêter toutes les ressources ?
enable-acl	false	Indique si le cluster peut utiliser des listes de contrôle d'accès, telles que définies par la commande **pcs acl**.
placement-strategy	par défaut	Indique si et comment le cluster prendra en compte les attributs d'utilisation lors de la détermination de l'emplacement des ressources sur les nœuds du cluster.
priority-fencing-delay	0 (désactivé)	Permet de configurer un cluster à deux nœuds de manière à ce que, dans une situation de cerveau divisé, le nœud ayant le moins de ressources en cours d'exécution soit celui qui est clôturé. La propriété **priority-fencing-delay** peut être définie comme une durée. La valeur par défaut de cette propriété est 0 (désactivé). Si cette propriété est définie sur une valeur

Option	Défaut	Description
		non nulle et que le méta-attribut **priority** est configuré pour au moins une ressource, le nœud ayant la priorité combinée la plus élevée de toutes les ressources fonctionnant sur lui aura plus de chances de survivre dans une situation où le cerveau est divisé. Par exemple, si vous définissez **pcs resource defaults priority=1** et **pcs property set priority-fencing-delay=15s** et qu'aucune autre priorité n'est définie, le nœud exploitant le plus de ressources aura plus de chances de survivre car l'autre nœud attendra 15 secondes avant d'initier la clôture. Si une ressource particulière est plus importante que les autres, vous pouvez lui donner une priorité plus élevée. Le nœud jouant le rôle promu d'un clone promouvable obtient un point supplémentaire si une priorité a été configurée pour ce clone. Tout délai défini à l'aide de la propriété **priority-fencing-delay** sera ajouté à tout délai défini à l'aide des propriétés de périphérique de clôture **pcmk_delay_base** et **pcmk_delay_max**. Ce comportement autorise un certain délai lorsque les deux nœuds ont la même priorité ou que les deux nœuds doivent être clôturés pour une raison autre que la perte d'un nœud (par exemple, si **on-fail=fencing** est défini pour une opération de surveillance des ressources). En cas d'utilisation combinée, il est recommandé de définir la propriété **priority-fencing-delay** à une valeur nettement supérieure au délai maximal des propriétés **pcmk_delay_base** et **pcmk_delay_max**, afin de s'assurer que le nœud prioritaire est privilégié (une valeur deux fois plus élevée serait tout à fait sûre). Seules les clôtures programmées par Pacemaker lui-même respecteront **priority-fencing-delay**. Les clôtures programmées par un code externe tel que **dlm_controld** ne fourniront pas les informations nécessaires au dispositif de clôture.

Option	Défaut	Description
node-health-strategy	aucun	Lorsqu'il est utilisé avec un agent de ressources de santé, contrôle la manière dont Pacemaker réagit aux changements dans la santé du nœud. Valeurs autorisées : * **none** - Ne pas suivre l'évolution de la santé des nœuds. * **migrate-on-red** - Les ressources sont retirées de tout nœud pour lequel un agent de santé a déterminé que l'état du nœud est **red**, sur la base des conditions locales surveillées par l'agent. * **only-green** - Les ressources sont retirées de tout nœud pour lequel un agent de santé a déterminé que l'état du nœud est **yellow** ou **red**, sur la base des conditions locales surveillées par l'agent. * **progressive custom** - Stratégies avancées de santé des nœuds offrant un contrôle plus fin de la réponse de la grappe aux conditions de santé en fonction des valeurs numériques internes des attributs de santé.

22.2. DÉFINITION ET SUPPRESSION DES PROPRIÉTÉS D'UN CLUSTER

Pour définir la valeur d'une propriété de cluster, utilisez la commande suivante **pcs** suivante.

```
pcs property set property=value
```

Par exemple, pour définir la valeur de **symmetric-cluster** à **false**, utilisez la commande suivante.

```
# pcs property set symmetric-cluster=false
```

Vous pouvez supprimer une propriété de cluster de la configuration à l'aide de la commande suivante.

```
pcs property unset property
```

Vous pouvez également supprimer une propriété de cluster d'une configuration en laissant le champ value de la commande **pcs property set** vide. La valeur par défaut de cette propriété est alors rétablie. Par exemple, si vous avez précédemment défini la propriété **symmetric-cluster** sur **false**, la commande suivante supprime la valeur que vous avez définie de la configuration et restaure la valeur de **symmetric-cluster** sur **true**, qui est sa valeur par défaut.

```
# pcs property set symmetic-cluster=
```

22.3. INTERROGER LES PARAMÈTRES DES PROPRIÉTÉS DES CLUSTERS

Dans la plupart des cas, lorsque vous utilisez la commande **pcs** pour afficher les valeurs des différents composants de la grappe, vous pouvez utiliser indifféremment **pcs list** ou **pcs show**. Dans les exemples suivants, **pcs list** est le format utilisé pour afficher une liste complète de tous les paramètres de plusieurs propriétés, tandis que **pcs show** est le format utilisé pour afficher les valeurs d'une propriété spécifique.

Pour afficher les valeurs des paramètres de propriété qui ont été définis pour le cluster, utilisez la commande suivante **pcs** suivante.

```
liste des propriétés des pcs
```

Pour afficher toutes les valeurs des paramètres de propriété du cluster, y compris les valeurs par défaut des paramètres de propriété qui n'ont pas été explicitement définis, utilisez la commande suivante.

```
pcs property list --all
```

Pour afficher la valeur actuelle d'une propriété de cluster spécifique, utilisez la commande suivante.

```
salon de l'immobilier pcs property
```

Par exemple, pour afficher la valeur actuelle de la propriété **cluster-infrastructure**, exécutez la commande suivante :

```
# pcs property show cluster-infrastructure
Cluster Properties:
 cluster-infrastructure: cman
```

A titre d'information, vous pouvez afficher une liste de toutes les valeurs par défaut des propriétés, qu'elles aient été définies à une valeur autre que la valeur par défaut ou non, en utilisant la commande suivante.

```
pcs property [list|show] --defaults
```

CHAPITRE 23. CONFIGURATION DES RESSOURCES POUR QU'ELLES RESTENT ARRÊTÉES LORS DE L'ARRÊT DU NŒUD PROPRE

Lorsqu'un nœud de cluster s'arrête, la réponse par défaut de Pacemaker est d'arrêter toutes les ressources en cours d'exécution sur ce nœud et de les récupérer ailleurs, même s'il s'agit d'un arrêt propre. Vous pouvez configurer Pacemaker de sorte que lorsqu'un nœud s'arrête proprement, les ressources attachées au nœud soient bloquées sur le nœud et incapables de démarrer ailleurs jusqu'à ce qu'elles redémarrent lorsque le nœud qui s'est arrêté réintègre le cluster. Cela vous permet de mettre hors tension les nœuds pendant les fenêtres de maintenance, lorsque les interruptions de service sont acceptables, sans que les ressources de ce nœud ne basculent sur d'autres nœuds de la grappe.

23.1. PROPRIÉTÉS DU CLUSTER POUR CONFIGURER LES RESSOURCES QUI DOIVENT RESTER ARRÊTÉES LORS DE L'ARRÊT D'UN NŒUD PROPRE

La possibilité d'empêcher les ressources de basculer lors de l'arrêt d'un nœud propre est mise en œuvre au moyen des propriétés suivantes du cluster.

shutdown-lock

Lorsque cette propriété est définie sur la valeur par défaut **false**, le cluster récupère les ressources actives sur les nœuds en cours d'arrêt propre. Lorsque cette propriété a pour valeur **true**, les ressources actives sur les nœuds faisant l'objet d'un arrêt propre sont incapables de démarrer ailleurs jusqu'à ce qu'elles démarrent à nouveau sur le nœud après qu'il a rejoint le cluster.
La propriété **shutdown-lock** fonctionne pour les nœuds de cluster ou les nœuds distants, mais pas pour les nœuds invités.

Si **shutdown-lock** est défini sur **true**, vous pouvez supprimer le verrou d'une ressource de cluster lorsqu'un nœud est en panne afin que la ressource puisse démarrer ailleurs en effectuant un rafraîchissement manuel sur le nœud à l'aide de la commande suivante.

pcs resource refresh *resource* node=*nodename*

Notez qu'une fois les ressources déverrouillées, le cluster est libre de les déplacer ailleurs. Vous pouvez contrôler la probabilité que cela se produise en utilisant des valeurs d'adhérence ou des préférences de localisation pour la ressource.

NOTE

Une actualisation manuelle ne fonctionnera avec les nœuds distants que si vous exécutez d'abord les commandes suivantes :

1. Exécutez la commande **systemctl stop pacemaker_remote** sur le nœud distant pour arrêter le nœud.
2. Exécutez la **pcs resource disable *remote-connection-resource*** commande.

Vous pouvez ensuite procéder à une actualisation manuelle sur le nœud distant.

shutdown-lock-limit

Lorsque cette propriété de cluster est définie sur une durée différente de la valeur par défaut de 0, les ressources seront disponibles pour la récupération sur d'autres nœuds si le nœud n'est pas rétabli dans le délai spécifié depuis le début de l'arrêt.

NOTE

La propriété **shutdown-lock-limit** ne fonctionnera avec les nœuds distants que si vous exécutez d'abord les commandes suivantes :

1. Exécutez la commande **systemctl stop pacemaker_remote** sur le nœud distant pour arrêter le nœud.
2. Exécutez la **pcs resource disable *remote-connection-resource*** commande.

Après l'exécution de ces commandes, les ressources qui étaient en cours d'exécution sur le nœud distant seront disponibles pour la récupération sur d'autres nœuds lorsque le délai spécifié à l'adresse **shutdown-lock-limit** se sera écoulé.

23.2. DÉFINITION DE LA PROPRIÉTÉ "SHUTDOWN-LOCK" DU CLUSTER

L'exemple suivant définit la propriété de cluster **shutdown-lock** sur **true** dans un exemple de cluster et montre l'effet que cela a lorsque le nœud est arrêté et redémarré. Cet exemple de cluster se compose de trois nœuds : **z1.example.com z2.example.com** , et **z3.example.com**.

Procédure

1. Définissez la propriété **shutdown-lock** à **true** et vérifiez sa valeur. Dans cet exemple, la propriété **shutdown-lock-limit** conserve sa valeur par défaut de 0.

```
[root@z3 ~]# pcs property set shutdown-lock=true
[root@z3 ~]# pcs property list --all | grep shutdown-lock
 shutdown-lock: true
 shutdown-lock-limit: 0
```

2. Vérifiez l'état de la grappe. Dans cet exemple, les ressources **third** et **fifth** sont exécutées sur **z1.example.com**.

```
[root@z3 ~]# pcs status
...
Full List of Resources:
...
 * first (ocf::pacemaker:Dummy): Started z3.example.com
 * second (ocf::pacemaker:Dummy): Started z2.example.com
 * third (ocf::pacemaker:Dummy): Started z1.example.com
 * fourth (ocf::pacemaker:Dummy): Started z2.example.com
 * fifth (ocf::pacemaker:Dummy): Started z1.example.com
...
```

3. Arrêtez **z1.example.com**, ce qui aura pour effet d'arrêter les ressources en cours d'exécution sur ce nœud.

```
[root@z3 ~] # pcs cluster stop z1.example.com
Stopping Cluster (pacemaker)...
Stopping Cluster (corosync)...
```

4. L'exécution de la commande **pcs status** montre que le nœud **z1.example.com** est hors ligne et que les ressources qui s'exécutaient sur **z1.example.com** sont **LOCKED** alors que le nœud est hors ligne.

```
[root@z3 ~]# pcs status
...

Node List:
 * Online: [ z2.example.com z3.example.com ]
 * OFFLINE: [ z1.example.com ]

Full List of Resources:
...
 * first (ocf::pacemaker:Dummy): Started z3.example.com
 * second (ocf::pacemaker:Dummy): Started z2.example.com
 * third (ocf::pacemaker:Dummy): Stopped z1.example.com (LOCKED)
 * fourth (ocf::pacemaker:Dummy): Started z3.example.com
 * fifth (ocf::pacemaker:Dummy): Stopped z1.example.com (LOCKED)

...
```

5. Redémarrez les services de cluster sur **z1.example.com** pour qu'il rejoigne le cluster. Les ressources verrouillées devraient démarrer sur ce nœud, bien qu'une fois démarrées, elles ne resteront pas nécessairement sur le même nœud.

```
[root@z3 ~]# pcs cluster start z1.example.com
Starting Cluster...
```

6. Dans cet exemple, les ressources **third** et **fifth** sont récupérées sur le nœud **z1.example.com**.

```
[root@z3 ~]# pcs status
...

Node List:
 * Online: [ z1.example.com z2.example.com z3.example.com ]

Full List of Resources:
..
 * first (ocf::pacemaker:Dummy): Started z3.example.com
 * second (ocf::pacemaker:Dummy): Started z2.example.com
 * third (ocf::pacemaker:Dummy): Started z1.example.com
 * fourth (ocf::pacemaker:Dummy): Started z3.example.com
 * fifth (ocf::pacemaker:Dummy): Started z1.example.com

...
```

CHAPITRE 24. CONFIGURATION D'UNE STRATÉGIE DE PLACEMENT DE NŒUDS

Pacemaker décide de l'emplacement d'une ressource en fonction des scores d'allocation des ressources sur chaque nœud. La ressource sera allouée au nœud où elle a le score le plus élevé. Ce score d'allocation est dérivé d'une combinaison de facteurs, y compris les contraintes de ressources, les paramètres **resource-stickiness**, l'historique des défaillances d'une ressource sur chaque nœud et l'utilisation de chaque nœud.

Si les scores d'allocation des ressources sur tous les nœuds sont égaux, Pacemaker choisira, par la stratégie de placement par défaut, un nœud avec le plus petit nombre de ressources allouées pour équilibrer la charge. Si le nombre de ressources sur chaque nœud est égal, le premier nœud éligible listé dans la CIB sera choisi pour exécuter la ressource.

Souvent, cependant, les différentes ressources utilisent des proportions sensiblement différentes des capacités d'un nœud (telles que la mémoire ou les E/S). Il n'est pas toujours possible d'équilibrer idéalement la charge en tenant compte uniquement du nombre de ressources allouées à un nœud. En outre, si les ressources sont placées de telle sorte que leurs besoins combinés dépassent la capacité fournie, elles peuvent ne pas démarrer complètement ou fonctionner avec des performances dégradées. Pour tenir compte de ces facteurs, Pacemaker vous permet de configurer les composants suivants :

- la capacité d'un nœud particulier
- la capacité requise par une ressource particulière
- une stratégie globale de placement des ressources

24.1. CARACTÉRISTIQUES D'UTILISATION ET STRATÉGIE DE PLACEMENT

Pour configurer la capacité qu'un nœud fournit ou qu'une ressource requiert, vous pouvez utiliser *utilization attributes* pour les nœuds et les ressources. Pour ce faire, vous devez définir une variable d'utilisation pour une ressource et lui attribuer une valeur pour indiquer ce dont la ressource a besoin, puis définir cette même variable d'utilisation pour un nœud et lui attribuer une valeur pour indiquer ce que ce nœud fournit.

Vous pouvez nommer les attributs d'utilisation selon vos préférences et définir autant de paires de noms et de valeurs que votre configuration l'exige. Les valeurs des attributs d'utilisation doivent être des nombres entiers.

24.1.1. Configuration de la capacité des nœuds et des ressources

L'exemple suivant configure un attribut d'utilisation de la capacité de l'unité centrale pour deux nœuds, en définissant cet attribut comme la variable **cpu**. Il configure également un attribut d'utilisation de la capacité de mémoire vive, en définissant cet attribut comme la variable **memory**. Dans cet exemple :

- Le nœud 1 est défini comme ayant une capacité d'unité centrale de deux et une capacité de mémoire vive de 2048
- Le nœud 2 est défini comme ayant une capacité d'unité centrale de quatre et une capacité de mémoire vive de 2048

```
# pcs node utilization node1 cpu=2 memory=2048
# pcs node utilization node2 cpu=4 memory=2048
```

L'exemple suivant spécifie les mêmes attributs d'utilisation pour trois ressources différentes. Dans cet exemple :

- la ressource **dummy-small** nécessite une capacité de CPU de 1 et une capacité de RAM de 1024
- la ressource **dummy-medium** nécessite une capacité de CPU de 2 et une capacité de RAM de 2048
- la ressource **dummy-large** nécessite une capacité de CPU de 1 et une capacité de RAM de 3072

```
# pcs resource utilization dummy-small cpu=1 memory=1024
# pcs resource utilization dummy-medium cpu=2 memory=2048
# pcs resource utilization dummy-large cpu=3 memory=3072
```

Un nœud est considéré comme éligible pour une ressource s'il dispose d'une capacité libre suffisante pour satisfaire aux exigences de la ressource, telles que définies par les attributs d'utilisation.

24.1.2. Configuration de la stratégie de placement

Une fois que vous avez configuré les capacités de vos nœuds et les capacités requises par vos ressources, vous devez définir la propriété de cluster **placement-strategy**, sinon les configurations de capacité n'auront aucun effet.

Quatre valeurs sont disponibles pour la propriété **placement-strategy** cluster :

- **default** - Les valeurs d'utilisation ne sont pas du tout prises en compte. Les ressources sont allouées en fonction des scores d'allocation. Si les scores sont égaux, les ressources sont réparties de manière égale entre les nœuds.
- **utilization** - Les valeurs d'utilisation ne sont prises en compte que lorsqu'il s'agit de décider si un nœud est considéré comme éligible (c'est-à-dire s'il dispose d'une capacité libre suffisante pour satisfaire aux exigences de la ressource). L'équilibrage de la charge se fait toujours sur la base du nombre de ressources allouées à un nœud.
- **balanced** - Les valeurs d'utilisation sont prises en compte lorsqu'il s'agit de décider si un nœud est éligible pour servir une ressource et lors de l'équilibrage de la charge, de sorte que l'on s'efforce de répartir les ressources de manière à optimiser leurs performances.
- **minimal** - Les valeurs d'utilisation ne sont prises en compte que lorsqu'il s'agit de décider si un nœud est éligible pour servir une ressource. Pour l'équilibrage de la charge, on essaie de concentrer les ressources sur le plus petit nombre possible de nœuds, ce qui permet d'économiser de l'énergie sur les nœuds restants.

L'exemple de commande suivant définit la valeur de **placement-strategy** à **balanced**. Après avoir exécuté cette commande, Pacemaker veillera à ce que la charge de vos ressources soit répartie uniformément dans l'ensemble du cluster, sans qu'il soit nécessaire d'appliquer des contraintes de colocalisation complexes.

```
# pcs property set placement-strategy=balanced
```

24.2. ALLOCATION DES RESSOURCES POUR LES STIMULATEURS CARDIAQUES

Pacemaker alloue les ressources en fonction de la préférence des nœuds, de la capacité des nœuds et de la préférence d'allocation des ressources.

24.2.1. Préférence pour les nœuds

Pacemaker détermine quel nœud est privilégié lors de l'allocation des ressources selon la stratégie suivante.

- Le nœud ayant le poids le plus élevé est consommé en premier. Le poids du nœud est un score maintenu par le cluster pour représenter la santé du nœud.
- Si plusieurs nœuds ont le même poids :
  - Si la propriété du cluster **placement-strategy** est **default** ou **utilization**:
    - Le nœud qui a le moins de ressources allouées est consommé en premier.
    - Si le nombre de ressources allouées est égal, le premier nœud éligible répertorié dans le CIB est consommé en premier.
  - Si la propriété de la grappe **placement-strategy** est **balanced**:
    - Le nœud qui a la plus grande capacité libre est consommé en premier.
    - Si les capacités libres des nœuds sont égales, le nœud qui a le moins de ressources allouées est consommé en premier.
    - Si les capacités libres des nœuds sont égales et que le nombre de ressources allouées est égal, le premier nœud éligible répertorié dans la CIB est consommé en premier.
  - Si la propriété du cluster **placement-strategy** est **minimal**, le premier nœud éligible listé dans le CIB est consommé en premier.

24.2.2. Capacité des nœuds

Pacemaker détermine quel nœud a la plus grande capacité libre selon la stratégie suivante.

- Si un seul type d'attribut d'utilisation a été défini, la capacité libre est une simple comparaison numérique.
- Si plusieurs types d'attributs d'utilisation ont été définis, le nœud dont la valeur numérique est la plus élevée dans le plus grand nombre de types d'attributs dispose de la plus grande capacité libre. Par exemple :
  - Si le nœud A dispose de plus de CPU libres et le nœud B de plus de mémoire libre, leurs capacités libres sont égales.
  - Si le nœud A dispose de plus d'unités centrales libres, tandis que le nœud B dispose de plus de mémoire et de stockage libres, alors le nœud B dispose d'une plus grande capacité libre.

24.2.3. Préférence en matière d'allocation des ressources

Pacemaker détermine quelle ressource est allouée en premier selon la stratégie suivante.

- La ressource qui a la priorité la plus élevée est allouée en premier. Vous pouvez définir la priorité d'une ressource lorsque vous la créez.

- Si les priorités des ressources sont égales, la ressource qui a le score le plus élevé sur le nœud où elle s'exécute est allouée en premier, afin d'éviter le brassage des ressources.

- Si les scores des ressources sur les nœuds où les ressources sont en cours d'exécution sont égaux ou si les ressources ne sont pas en cours d'exécution, la ressource qui a le score le plus élevé sur le nœud préféré est allouée en premier. Si les scores des ressources sur le nœud préféré sont égaux dans ce cas, la première ressource en cours d'exécution répertoriée dans la CIB est allouée en premier.

24.3. LIGNES DIRECTRICES RELATIVES À LA STRATÉGIE DE PLACEMENT DES RESSOURCES

Pour que la stratégie de placement des ressources de Pacemaker soit la plus efficace possible, vous devez tenir compte des considérations suivantes lors de la configuration de votre système.

- Assurez-vous que vous disposez d'une capacité physique suffisante.
  Si la capacité physique de vos nœuds est utilisée au maximum dans des conditions normales, des problèmes peuvent survenir lors du basculement. Même sans la fonction d'utilisation, vous pouvez commencer à subir des dépassements de délai et des défaillances secondaires.

- Ajoutez un tampon dans les capacités que vous configurez pour les nœuds.
  Annoncez un peu plus de ressources de nœuds que vous n'en avez physiquement, en partant du principe qu'une ressource Pacemaker n'utilisera pas en permanence 100 % de la quantité configurée de CPU, de mémoire, etc. Cette pratique est parfois appelée overcommit.

- Spécifier les priorités en matière de ressources.
  Si le cluster doit sacrifier des services, ce doit être ceux dont vous vous souciez le moins. Assurez-vous que les priorités des ressources sont correctement définies afin que les ressources les plus importantes soient planifiées en premier.

24.4. L'AGENT DE RESSOURCES NODEUTILIZATION

L'agent de la ressource **NodeUtilization** peut détecter les paramètres système de l'unité centrale disponible, de la disponibilité de la mémoire hôte et de la disponibilité de la mémoire de l'hyperviseur et ajouter ces paramètres dans la CIB. Vous pouvez exécuter l'agent en tant que ressource clone pour qu'il remplisse automatiquement ces paramètres sur chaque nœud.

Pour obtenir des informations sur l'agent de ressources **NodeUtilization** et les options de ressources de cet agent, exécutez la commande **pcs resource describe NodeUtilization**.

CHAPITRE 25. CONFIGURER UN DOMAINE VIRTUEL EN TANT QUE RESSOURCE

Vous pouvez configurer un domaine virtuel géré par le cadre de virtualisation **libvirt** en tant que ressource de cluster avec la commande **pcs resource create**, en spécifiant **VirtualDomain** comme type de ressource.

Lors de la configuration d'un domaine virtuel en tant que ressource, tenez compte des considérations suivantes :

- Un domaine virtuel doit être arrêté avant d'être configuré comme ressource de cluster.
- Une fois qu'un domaine virtuel est une ressource de cluster, il ne doit être démarré, arrêté ou migré qu'à l'aide des outils de cluster.
- Ne configurez pas un domaine virtuel que vous avez configuré en tant que ressource de cluster pour qu'il démarre lorsque son hôte démarre.
- Tous les nœuds autorisés à exécuter un domaine virtuel doivent avoir accès aux fichiers de configuration et aux périphériques de stockage nécessaires pour ce domaine virtuel.

Si vous souhaitez que le cluster gère les services au sein du domaine virtuel lui-même, vous pouvez configurer le domaine virtuel en tant que nœud invité.

25.1. OPTIONS DE RESSOURCES DU DOMAINE VIRTUEL

Le tableau suivant décrit les options de ressources que vous pouvez configurer pour une ressource **VirtualDomain**.

Tableau 25.1. Options de ressources pour les ressources de domaines virtuels

Field	Défaut	Description
config		(obligatoire) Chemin absolu vers le fichier de configuration **libvirt** pour ce domaine virtuel.
hypervisor	En fonction du système	URI de l'hyperviseur auquel se connecter. Vous pouvez déterminer l'URI par défaut du système en exécutant la commande **virsh --quiet uri**.
force_stop	**0**	Toujours arrêter de force (détruire) le domaine à l'arrêt. Le comportement par défaut est de recourir à un arrêt forcé uniquement après l'échec d'une tentative d'arrêt gracieux. Vous ne devez définir ce paramètre sur **true** que si votre domaine virtuel (ou votre back-end de virtualisation) ne prend pas en charge l'arrêt progressif.

Field	Défaut	Description
migration_transport	En fonction du système	Transport utilisé pour se connecter à l'hyperviseur distant lors de la migration. Si ce paramètre est omis, la ressource utilisera le transport par défaut de **libvirt** pour se connecter à l'hyperviseur distant.
migration_network_suffix		Utiliser un réseau de migration dédié. L'URI de migration est composé en ajoutant la valeur de ce paramètre à la fin du nom du nœud. Si le nom du nœud est un nom de domaine entièrement qualifié (FQDN), insérez le suffixe immédiatement avant le premier point (.) du FQDN. Assurez-vous que ce nom d'hôte composé peut être résolu localement et que l'adresse IP associée est accessible via le réseau favorisé.
monitor_scripts		Pour surveiller en plus les services dans le domaine virtuel, ajoutez ce paramètre avec une liste de scripts à surveiller. *Note*: Lorsque des scripts de surveillance sont utilisés, les opérations **start** et **migrate_from** ne se termineront que lorsque tous les scripts de surveillance auront été exécutés avec succès. Veillez à définir le délai d'attente de ces opérations pour tenir compte de ce retard
autoset_utilization_cpu	**true**	S'il est défini sur **true**, l'agent détectera le nombre de **domainU**'s **vCPU**s de **virsh**, et l'intégrera dans l'utilisation de l'unité centrale de la ressource lors de l'exécution du moniteur.
autoset_utilization_hv_memory	**true**	S'il est défini comme vrai, l'agent détectera le nombre de **Max memory** à partir de **virsh** et l'ajoutera à l'utilisation de **hv_memory** de la source lors de l'exécution du moniteur.
migrateport	aléatoire highport	Ce port sera utilisé dans l'URI de migration de **qemu**. S'il n'est pas défini, le port sera un port élevé aléatoire.

Field	Défaut	Description
snapshot		Chemin d'accès au répertoire d'instantanés où l'image de la machine virtuelle sera stockée. Lorsque ce paramètre est défini, l'état de la mémoire vive de la machine virtuelle sera sauvegardé dans un fichier dans le répertoire d'instantanés lorsqu'elle est arrêtée. Si, au démarrage, un fichier d'état est présent pour le domaine, le domaine sera restauré dans l'état dans lequel il se trouvait juste avant son dernier arrêt. Cette option est incompatible avec l'option **force_stop**.

Outre les options de ressource **VirtualDomain**, vous pouvez configurer l'option de métadonnées **allow-migrate** pour permettre la migration en direct de la ressource vers un autre nœud. Lorsque cette option est définie sur **true**, la ressource peut être migrée sans perte d'état. Lorsque cette option est définie sur **false**, qui est l'état par défaut, le domaine virtuel est arrêté sur le premier nœud, puis redémarré sur le deuxième nœud lorsqu'il est déplacé d'un nœud à l'autre.

25.2. CRÉATION DE LA RESSOURCE DU DOMAINE VIRTUEL

La procédure suivante crée une ressource **VirtualDomain** dans un cluster pour une machine virtuelle que vous avez précédemment créée.

Procédure

1. Pour créer l'agent de ressources **VirtualDomain** pour la gestion de la machine virtuelle, Pacemaker a besoin que le fichier de configuration **xml** de la machine virtuelle soit déchargé dans un fichier sur le disque. Par exemple, si vous avez créé une machine virtuelle nommée **guest1**, transférez le fichier **xml** dans un fichier situé sur l'un des nœuds du cluster qui sera autorisé à exécuter l'invité. Vous pouvez utiliser un nom de fichier de votre choix ; cet exemple utilise **/etc/pacemaker/guest1.xml**.

   ```
   # virsh dumpxml guest1 > /etc/pacemaker/guest1.xml
   ```

2. Copiez le fichier de configuration **xml** de la machine virtuelle sur tous les autres nœuds du cluster qui seront autorisés à exécuter l'invité, au même endroit sur chaque nœud.

3. Assurez-vous que tous les nœuds autorisés à exécuter le domaine virtuel ont accès aux périphériques de stockage nécessaires pour ce domaine virtuel.

4. Testez séparément que le domaine virtuel peut démarrer et s'arrêter sur chaque nœud qui exécutera le domaine virtuel.

5. S'il est en cours d'exécution, arrêtez le nœud invité. Pacemaker démarrera le nœud lorsqu'il sera configuré dans le cluster. La machine virtuelle ne doit pas être configurée pour démarrer automatiquement lorsque l'hôte démarre.

6. Configurez la ressource **VirtualDomain** avec la commande **pcs resource create**. Par exemple,

la commande suivante configure une ressource **VirtualDomain** nommée **VM**. Étant donné que l'option **allow-migrate** est définie sur **true**, une commande **pcs resource move VM *nodeX*** serait effectuée comme une migration en direct.
Dans cet exemple, **migration_transport** est remplacé par **ssh**. Notez que pour que la migration SSH fonctionne correctement, la journalisation sans clé doit fonctionner entre les nœuds.

```
# pcs resource create VM VirtualDomain config=/etc/pacemaker/guest1.xml
migration_transport=ssh meta allow-migrate=true
```

CHAPITRE 26. CONFIGURATION DU QUORUM DU CLUSTER

Un cluster Red Hat Enterprise Linux High Availability Add-On utilise le service **votequorum**, en conjonction avec la clôture, pour éviter les situations de cerveau divisé. Un nombre de votes est attribué à chaque système de la grappe, et les opérations de la grappe ne sont autorisées que lorsqu'une majorité de votes est présente. Le service doit être chargé sur tous les nœuds ou sur aucun ; s'il est chargé sur un sous-ensemble de nœuds de la grappe, les résultats seront imprévisibles. Pour plus d'informations sur la configuration et le fonctionnement du service **votequorum**, voir la page de manuel **votequorum**(5).

26.1. CONFIGURATION DES OPTIONS DE QUORUM

Vous pouvez définir certaines caractéristiques spéciales de la configuration du quorum lorsque vous créez un cluster à l'aide de la commande **pcs cluster setup**. Le tableau suivant résume ces options.

Tableau 26.1. Options de quorum

Option	Description
auto_tie_breaker	Lorsqu'elle est activée, la grappe peut subir une défaillance simultanée de 50 % des nœuds, de manière déterministe. La partition de la grappe, ou l'ensemble des nœuds qui sont toujours en contact avec le site **nodeid** configuré dans **auto_tie_breaker_node** (ou le site **nodeid** le plus bas s'il n'est pas configuré), restera en nombre suffisant. Les autres nœuds ne seront pas saturés. L'option **auto_tie_breaker** est principalement utilisée pour les clusters avec un nombre pair de nœuds, car elle permet au cluster de continuer à fonctionner avec une répartition égale. Pour des défaillances plus complexes, telles que des divisions multiples et inégales, il est recommandé d'utiliser un périphérique quorum L'option **auto_tie_breaker** est incompatible avec les dispositifs de quorum.
wait_for_all	Lorsqu'elle est activée, la grappe ne sera quorate pour la première fois que lorsque tous les nœuds auront été visibles au moins une fois en même temps. L'option **wait_for_all** est principalement utilisée pour les grappes à deux nœuds et pour les grappes à nœuds pairs utilisant le dispositif de quorum **lms** (algorithme du dernier homme debout). L'option **wait_for_all** est automatiquement activée lorsqu'un cluster comporte deux nœuds, n'utilise pas de périphérique quorum et que **auto_tie_breaker** est désactivé. Vous pouvez remplacer cette option en définissant explicitement **wait_for_all** à 0.

Option	Description
last_man_standing	Lorsque cette option est activée, le cluster peut recalculer dynamiquement **expected_votes** et le quorum dans des circonstances spécifiques. Vous devez activer **wait_for_all** lorsque vous activez cette option. L'option **last_man_standing** est incompatible avec les périphériques de quorum.
last_man_standing_window	Le temps, en millisecondes, à attendre avant de recalculer **expected_votes** et le quorum après qu'un cluster ait perdu des nœuds.

Pour plus d'informations sur la configuration et l'utilisation de ces options, voir la page de manuel **votequorum**(5).

26.2. MODIFIER LES OPTIONS DE QUORUM

Vous pouvez modifier les options générales de quorum pour votre cluster avec la commande **pcs quorum update**. Vous pouvez modifier les options **quorum.two_node** et **quorum.expected_votes** sur un système en cours d'exécution. Pour toutes les autres options de quorum, l'exécution de cette commande nécessite l'arrêt du cluster. Pour plus d'informations sur les options de quorum, consultez la page de manuel **votequorum**(5).

Le format de la commande **pcs quorum update** est le suivant.

```
mise à jour du quorum [auto_tie_breaker=[0|1]] [last_man_standing=[0|1]] [last_man_standing=[0|1]]
[last_man_standing_window=[time-in-ms] [wait_for_all=[0|1]]
```

La série de commandes suivante modifie l'option **wait_for_all** quorum et affiche l'état actualisé de l'option. Notez que le système ne vous permet pas d'exécuter cette commande lorsque le cluster est en cours d'exécution.

```
[root@node1:~]# pcs quorum update wait_for_all=1
Checking corosync is not running on nodes...
Error: node1: corosync is running
Error: node2: corosync is running

[root@node1:~]# pcs cluster stop --all
node2: Stopping Cluster (pacemaker)...
node1: Stopping Cluster (pacemaker)...
node1: Stopping Cluster (corosync)...
node2: Stopping Cluster (corosync)...

[root@node1:~]# pcs quorum update wait_for_all=1
Checking corosync is not running on nodes...
node2: corosync is not running
node1: corosync is not running
Sending updated corosync.conf to nodes...
node1: Succeeded
node2: Succeeded
```

```
[root@node1:~]# pcs quorum config
Options:
  wait_for_all: 1
```

26.3. AFFICHAGE DE LA CONFIGURATION ET DE L'ÉTAT DU QUORUM

Lorsqu'un cluster est en cours d'exécution, vous pouvez entrer les commandes cluster quorum suivantes pour afficher la configuration et l'état du quorum.

La commande suivante montre la configuration du quorum.

```
pcs quorum [config]
```

La commande suivante indique l'état de l'exécution du quorum.

```
état du quorum de la pcs
```

26.4. EXÉCUTION DE GRAPPES D'INDICES

Si vous retirez des nœuds d'une grappe pendant une longue période et que la perte de ces nœuds entraînerait une perte de quorum, vous pouvez modifier la valeur du paramètre **expected_votes** pour la grappe active à l'aide de la commande **pcs quorum expected-votes**. Cela permet à la grappe de continuer à fonctionner lorsqu'elle n'a pas de quorum.

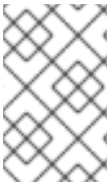
AVERTISSEMENT

La modification des votes attendus dans une grappe en activité doit être effectuée avec une extrême prudence. Si moins de 50 % de la grappe fonctionne parce que vous avez modifié manuellement les votes attendus, les autres nœuds de la grappe pourraient être démarrés séparément et exécuter les services de la grappe, ce qui entraînerait une corruption des données et d'autres résultats inattendus. Si vous modifiez cette valeur, vous devez vous assurer que le paramètre **wait_for_all** est activé.

La commande suivante définit les votes attendus dans le cluster live à la valeur spécifiée. Cette commande n'affecte que la grappe en direct et ne modifie pas le fichier de configuration ; la valeur de **expected_votes** est réinitialisée à la valeur du fichier de configuration en cas de rechargement.

```
pcs quorum votes attendus votes
```

Dans une situation où vous savez que le cluster est inquorate mais que vous voulez que le cluster procède à la gestion des ressources, vous pouvez utiliser la commande **pcs quorum unblock** pour empêcher le cluster d'attendre tous les nœuds lors de l'établissement du quorum.



NOTE

Cette commande doit être utilisée avec une extrême prudence. Avant de lancer cette commande, il est impératif de s'assurer que les nœuds qui ne font pas partie du cluster sont éteints et n'ont pas accès aux ressources partagées.

```
# pcs quorum unblock
```

CHAPITRE 27. CONFIGURATION DES DISPOSITIFS DE QUORUM

Vous pouvez permettre à une grappe de supporter plus de défaillances de nœuds que ne le permettent les règles de quorum standard en configurant un dispositif de quorum séparé qui agit comme un dispositif d'arbitrage tiers pour la grappe. Il est recommandé d'utiliser un dispositif de quorum pour les grappes comportant un nombre pair de nœuds. Dans le cas des grappes à deux nœuds, l'utilisation d'un dispositif de quorum permet de mieux déterminer le nœud qui survivra en cas de défaillance d'un nœud.

Vous devez tenir compte des éléments suivants lors de la configuration d'un dispositif de quorum.

- Il est recommandé de faire fonctionner un dispositif quorum sur un réseau physique différent sur le même site que le cluster qui utilise le dispositif quorum. Idéalement, l'hôte du dispositif quorum devrait se trouver dans un rack séparé de celui de la grappe principale, ou au moins sur une unité d'alimentation séparée, et non sur le même segment de réseau que l'anneau ou les anneaux corosync.
- Vous ne pouvez pas utiliser simultanément plus d'un dispositif quorum dans un cluster.
- Bien qu'il ne soit pas possible d'utiliser simultanément plus d'un dispositif de quorum dans une grappe, un seul dispositif de quorum peut être utilisé par plusieurs grappes en même temps. Chaque grappe utilisant ce dispositif de quorum peut utiliser des algorithmes et des options de quorum différents, car ceux-ci sont stockés sur les nœuds de la grappe eux-mêmes. Par exemple, un seul dispositif de quorum peut être utilisé par une grappe avec un algorithme **ffsplit** (fifty/fifty split) et par une deuxième grappe avec un algorithme **lms** (last man standing).
- Un dispositif de quorum ne doit pas être exécuté sur un nœud de cluster existant.

27.1. INSTALLATION DES PAQUETS DE PÉRIPHÉRIQUES QUORUM

La configuration d'un dispositif de quorum pour un cluster nécessite l'installation des paquets suivants :

- Installez **corosync-qdevice** sur les nœuds d'un cluster existant.

```
[root@node1:~]# dnf install corosync-qdevice
[root@node2:~]# dnf install corosync-qdevice
```

- Installez **pcs** et **corosync-qnetd** sur l'hôte du périphérique quorum.

```
[root@qdevice:~]# dnf install pcs corosync-qnetd
```

- Démarrez le service **pcsd** et activez **pcsd** au démarrage du système sur l'hôte du périphérique quorum.

```
[root@qdevice:~]# systemctl start pcsd.service
[root@qdevice:~]# systemctl enable pcsd.service
```

27.2. CONFIGURATION D'UN DISPOSITIF DE QUORUM

Configurez un dispositif quorum et ajoutez-le au cluster en suivant la procédure suivante.

Dans cet exemple :

- Le nœud utilisé pour le dispositif de quorum est **qdevice**.
- Le modèle de périphérique quorum est **net**, qui est actuellement le seul modèle pris en charge. Le modèle **net** prend en charge les algorithmes suivants :
  - **ffsplit**la répartition des votes est la suivante : cinquante-cinquante. Cela donne exactement une voix à la partition ayant le plus grand nombre de nœuds actifs.
  - **lms**: dernier survivant. Si le nœud est le seul de la grappe à pouvoir voir le serveur **qnetd**, il renvoie un vote.

AVERTISSEMENT

L'algorithme LMS permet à la grappe de rester quorate même s'il ne reste qu'un seul nœud, mais cela signifie également que le pouvoir de vote du dispositif de quorum est important puisqu'il est égal au nombre_de_nœuds - 1. Perdre la connexion avec le dispositif de quorum signifie perdre le nombre_de_nœuds - 1 votes, ce qui signifie que seule une grappe dont tous les nœuds sont actifs peut rester quorate (en sur-votant le dispositif de quorum) ; toute autre grappe devient inquorate.

Pour des informations plus détaillées sur l'implémentation de ces algorithmes, voir la page de manuel **corosync-qdevice**(8).

- Les nœuds de la grappe sont **node1** et **node2**.

Procédure

1. Sur le nœud que vous utiliserez pour héberger votre dispositif quorum, configurez le dispositif quorum à l'aide de la commande suivante. Cette commande configure et démarre le modèle de périphérique quorum **net** et configure le périphérique pour qu'il démarre au démarrage.

```
[root@qdevice:~]# pcs qdevice setup model net --enable --start
Quorum device 'net' initialized
quorum device enabled
Starting quorum device...
quorum device started
```

Après avoir configuré le périphérique quorum, vous pouvez vérifier son état. Cela devrait montrer que le démon **corosync-qnetd** est en cours d'exécution et qu'à ce stade, aucun client n'y est connecté. L'option de commande **--full** fournit une sortie détaillée.

```
[root@qdevice:~]# pcs qdevice status net --full
QNetd address:                  *:5403
TLS:                            Supported (client certificate required)
Connected clients:              0
Connected clusters:             0
Maximum send/receive size:      32768/32768 bytes
```

2. Activez les ports du pare-feu nécessaires au démon **pcsd** et au dispositif de quorum **net** en activant le service **high-availability** sur **firewalld** à l'aide des commandes suivantes.

```
[root@qdevice:~]# firewall-cmd --permanent --add-service=high-availability
[root@qdevice:~]# firewall-cmd --add-service=high-availability
```

3. Depuis l'un des nœuds du cluster existant, authentifiez l'utilisateur **hacluster** sur le nœud qui héberge le dispositif quorum. Cela permet à **pcs** sur le cluster de se connecter à **pcs** sur l'hôte **qdevice**, mais ne permet pas à **pcs** sur l'hôte **qdevice** de se connecter à **pcs** sur le cluster.

```
[root@node1:~] # pcs host auth qdevice
Username: hacluster
Password:
qdevice: Authorized
```

4. Ajoutez le dispositif quorum au cluster.
Avant d'ajouter le dispositif de quorum, vous pouvez vérifier la configuration et l'état actuels du dispositif de quorum à des fins de comparaison ultérieure. La sortie de ces commandes indique que le cluster n'utilise pas encore de dispositif de quorum et que l'état d'appartenance à **Qdevice** de chaque nœud est **NR** (Not Registered).

```
[root@node1:~]# pcs quorum config
Options:
```

```
[root@node1:~]# pcs quorum status
Quorum information
------------------
Date:             Wed Jun 29 13:15:36 2016
Quorum provider:  corosync_votequorum
Nodes:            2
Node ID:          1
Ring ID:          1/8272
Quorate:          Yes

Votequorum information
----------------------
Expected votes:   2
Highest expected: 2
Total votes:      2
Quorum:           1
Flags:            2Node Quorate

Membership information
----------------------
    Nodeid      Votes    Qdevice Name
         1          1         NR node1 (local)
         2          1         NR node2
```

La commande suivante permet d'ajouter à la grappe le dispositif quorum que vous avez créé précédemment. Vous ne pouvez pas utiliser simultanément plus d'un dispositif quorum dans une grappe. Cependant, un dispositif quorum peut être utilisé par plusieurs clusters en même temps. Cet exemple de commande configure le périphérique quorum pour qu'il utilise l'algorithme **ffsplit**. Pour plus d'informations sur les options de configuration du dispositif de quorum, consultez la page de manuel **corosync-qdevice**(8).

```
[root@node1:~]# pcs quorum device add model net host=qdevice algorithm=ffsplit
Setting up qdevice certificates on nodes...
node2: Succeeded
node1: Succeeded
Enabling corosync-qdevice...
node1: corosync-qdevice enabled
node2: corosync-qdevice enabled
Sending updated corosync.conf to nodes...
node1: Succeeded
node2: Succeeded
Corosync configuration reloaded
Starting corosync-qdevice...
node1: corosync-qdevice started
node2: corosync-qdevice started
```

5. Vérifier l'état de la configuration du dispositif quorum.
   Du côté du cluster, vous pouvez exécuter les commandes suivantes pour voir comment la configuration a changé.

   Le site **pcs quorum config** indique le périphérique quorum qui a été configuré.

```
[root@node1:~]# pcs quorum config
Options:
Device:
  Model: net
    algorithm: ffsplit
    host: qdevice
```

   La commande **pcs quorum status** affiche l'état d'exécution du quorum, indiquant que le dispositif de quorum est en cours d'utilisation. La signification des valeurs de l'état des informations sur l'appartenance à **Qdevice** pour chaque nœud de cluster est la suivante :

   - **A**/**NA** - Le dispositif quorum est vivant ou non, indiquant s'il y a un battement de cœur entre **qdevice** et **corosync**. Cela devrait toujours indiquer que le dispositif quorum est en vie.
   - **V**/**NV** - **V** est défini lorsque le dispositif de quorum a accordé un vote à un nœud. Dans cet exemple, les deux nœuds sont définis sur **V** puisqu'ils peuvent communiquer entre eux. Si la grappe devait se diviser en deux grappes à nœud unique, l'un des nœuds serait réglé sur **V** et l'autre sur **NV**.
   - **MW**/**NMW** - Le drapeau du dispositif de quorum interne est activé (**MW**) ou désactivé (**NMW**). Par défaut, l'indicateur n'est pas activé et la valeur est **NMW**.

```
[root@node1:~]# pcs quorum status
Quorum information
------------------
Date:             Wed Jun 29 13:17:02 2016
Quorum provider:  corosync_votequorum
Nodes:            2
Node ID:          1
Ring ID:          1/8272
Quorate:          Yes

Votequorum information
----------------------
```

```
Expected votes:   3
Highest expected: 3
Total votes:      3
Quorum:           2
Flags:            Quorate Qdevice

Membership information
----------------------
    Nodeid      Votes    Qdevice Name
         1          1    A,V,NMW node1 (local)
         2          1    A,V,NMW node2
         0          1            Qdevice
```

Le site **pcs quorum device status** indique l'état d'exécution du dispositif quorum.

```
[root@node1:~]# pcs quorum device status
Qdevice information
-------------------
Model:                  Net
Node ID:                1
Configured node list:
    0   Node ID = 1
    1   Node ID = 2
Membership node list:   1, 2

Qdevice-net information
----------------------
Cluster name:           mycluster
QNetd host:             qdevice:5403
Algorithm:              ffsplit
Tie-breaker:            Node with lowest node ID
State:                  Connected
```

Du côté du périphérique quorum, vous pouvez exécuter la commande d'état suivante, qui indique l'état du démon **corosync-qnetd**.

```
[root@qdevice:~]# pcs qdevice status net --full
QNetd address:                  *:5403
TLS:                            Supported (client certificate required)
Connected clients:              2
Connected clusters:             1
Maximum send/receive size:      32768/32768 bytes
Cluster "mycluster":
    Algorithm:          ffsplit
    Tie-breaker:        Node with lowest node ID
    Node ID 2:
        Client address:         ::ffff:192.168.122.122:50028
        HB interval:            8000ms
        Configured node list:   1, 2
        Ring ID:                1.2050
        Membership node list:   1, 2
        TLS active:             Yes (client certificate verified)
        Vote:                   ACK (ACK)
    Node ID 1:
        Client address:         ::ffff:192.168.122.121:48786
```

```
HB interval:          8000ms
Configured node list:   1, 2
Ring ID:              1.2050
Membership node list:   1, 2
TLS active:           Yes (client certificate verified)
Vote:                 ACK (ACK)
```

27.3. GÉRER LE SERVICE DE DISPOSITIF DE QUORUM

PCS permet de gérer le service de périphérique quorum sur l'hôte local (**corosync-qnetd**), comme le montrent les exemples de commandes suivants. Notez que ces commandes n'affectent que le service **corosync-qnetd**.

```
[root@qdevice:~]# pcs qdevice start net
[root@qdevice:~]# pcs qdevice stop net
[root@qdevice:~]# pcs qdevice enable net
[root@qdevice:~]# pcs qdevice disable net
[root@qdevice:~]# pcs qdevice kill net
```

27.4. GESTION D'UN DISPOSITIF QUORUM DANS UN CLUSTER

Il existe une variété de commandes **pcs** que vous pouvez utiliser pour modifier les paramètres des périphériques quorum dans un cluster, désactiver un périphérique quorum et supprimer un périphérique quorum.

27.4.1. Modification des paramètres du dispositif de quorum

Vous pouvez modifier la configuration d'un périphérique quorum à l'aide de la commande **pcs quorum device update**.

AVERTISSEMENT

Pour modifier l'option **host** du modèle de périphérique quorum **net**, utilisez les commandes **pcs quorum device remove** et **pcs quorum device add** pour définir la configuration correctement, sauf si l'ancien et le nouvel hôte sont la même machine.

La commande suivante modifie l'algorithme du dispositif de quorum en **lms**.

```
[root@node1:~]# pcs quorum device update model algorithm=lms
Sending updated corosync.conf to nodes...
node1: Succeeded
node2: Succeeded
Corosync configuration reloaded
Reloading qdevice configuration on nodes...
node1: corosync-qdevice stopped
```

```
node2: corosync-qdevice stopped
node1: corosync-qdevice started
node2: corosync-qdevice started
```

27.4.2. Suppression d'un dispositif quorum

La commande suivante supprime un dispositif de quorum configuré sur un nœud de cluster.

```
[root@node1:~]# pcs quorum device remove
Sending updated corosync.conf to nodes...
node1: Succeeded
node2: Succeeded
Corosync configuration reloaded
Disabling corosync-qdevice...
node1: corosync-qdevice disabled
node2: corosync-qdevice disabled
Stopping corosync-qdevice...
node1: corosync-qdevice stopped
node2: corosync-qdevice stopped
Removing qdevice certificates from nodes...
node1: Succeeded
node2: Succeeded
```

Après avoir supprimé un périphérique quorum, vous devriez voir le message d'erreur suivant lors de l'affichage de l'état du périphérique quorum.

```
[root@node1:~]# pcs quorum device status
Error: Unable to get quorum status: corosync-qdevice-tool: Can't connect to QDevice socket (is
QDevice running?): No such file or directory
```

27.4.3. Destruction d'un dispositif quorum

La commande suivante désactive et arrête un périphérique quorum sur l'hôte du périphérique quorum et supprime tous ses fichiers de configuration.

```
[root@qdevice:~]# pcs qdevice destroy net
Stopping quorum device...
quorum device stopped
quorum device disabled
Quorum device 'net' configuration files removed
```

CHAPITRE 28. DÉCLENCHEMENT DE SCRIPTS POUR LES ÉVÉNEMENTS DE LA GRAPPE

Une grappe Pacemaker est un système piloté par les événements, où un événement peut être une défaillance d'une ressource ou d'un nœud, un changement de configuration ou le démarrage ou l'arrêt d'une ressource. Vous pouvez configurer les alertes de la grappe Pacemaker pour qu'elles prennent une mesure externe lorsqu'un événement de la grappe se produit au moyen d'agents d'alerte, qui sont des programmes externes que la grappe appelle de la même manière que la grappe appelle des agents de ressources pour gérer la configuration et l'exploitation des ressources.

Le cluster transmet les informations relatives à l'événement à l'agent au moyen de variables d'environnement. Les agents peuvent faire ce qu'ils veulent avec ces informations, par exemple envoyer un message électronique, enregistrer dans un fichier ou mettre à jour un système de surveillance.

- Pacemaker fournit plusieurs exemples d'agents d'alerte, qui sont installés par défaut sur **/usr/share/pacemaker/alerts**. Ces exemples de scripts peuvent être copiés et utilisés tels quels, ou servir de modèles à modifier en fonction de vos besoins. Reportez-vous au code source des exemples d'agents pour connaître l'ensemble des attributs qu'ils prennent en charge.

- Si les exemples d'agents d'alerte ne répondent pas à vos besoins, vous pouvez écrire vos propres agents d'alerte pour appeler une alerte Pacemaker.

28.1. INSTALLATION ET CONFIGURATION D'EXEMPLES D'AGENTS D'ALERTE

Lorsque vous utilisez l'un des exemples d'agents d'alerte, vous devez examiner le script pour vous assurer qu'il répond à vos besoins. Ces exemples d'agents sont fournis comme point de départ pour des scripts personnalisés destinés à des environnements de clusters spécifiques. Notez que si Red Hat prend en charge les interfaces que les scripts d'agents d'alerte utilisent pour communiquer avec Pacemaker, Red Hat ne prend pas en charge les agents personnalisés eux-mêmes.

Pour utiliser l'un des exemples d'agents d'alerte, vous devez installer l'agent sur chaque nœud du cluster. Par exemple, la commande suivante installe le script **alert_file.sh.sample** en tant que **alert_file.sh**.

```
# install --mode=0755 /usr/share/pacemaker/alerts/alert_file.sh.sample
/var/lib/pacemaker/alert_file.sh
```

Après avoir installé le script, vous pouvez créer une alerte qui utilise le script.

L'exemple suivant configure une alerte qui utilise l'agent d'alerte **alert_file.sh** installé pour enregistrer les événements dans un fichier. Les agents d'alerte s'exécutent sous l'utilisateur **hacluster**, qui dispose d'un ensemble minimal d'autorisations.

Cet exemple crée le fichier journal **pcmk_alert_file.log** qui sera utilisé pour enregistrer les événements. Il crée ensuite l'agent d'alerte et ajoute le chemin d'accès au fichier journal comme destinataire.

```
# touch /var/log/pcmk_alert_file.log
# chown hacluster:haclient /var/log/pcmk_alert_file.log
# chmod 600 /var/log/pcmk_alert_file.log
# pcs alert create id=alert_file description="Log events to a file."
path=/var/lib/pacemaker/alert_file.sh
# pcs alert recipient add alert_file id=my-alert_logfile value=/var/log/pcmk_alert_file.log
```

L'exemple suivant installe le script **alert_snmp.sh.sample** en tant que **alert_snmp.sh** et configure une

alerte qui utilise l'agent d'alerte **alert_snmp.sh** installé pour envoyer les événements de la grappe sous forme de trappes SNMP. Par défaut, le script envoie au serveur SNMP tous les événements à l'exception des appels de moniteur réussis. Cet exemple configure le format d'horodatage en tant qu'option méta. Après avoir configuré l'alerte, cet exemple configure un destinataire pour l'alerte et affiche la configuration de l'alerte.

```
# install --mode=0755 /usr/share/pacemaker/alerts/alert_snmp.sh.sample
/var/lib/pacemaker/alert_snmp.sh
# pcs alert create id=snmp_alert path=/var/lib/pacemaker/alert_snmp.sh meta timestamp-
format="%Y-%m-%d,%H:%M:%S.%01N"
# pcs alert recipient add snmp_alert value=192.168.1.2
# pcs alert
Alerts:
 Alert: snmp_alert (path=/var/lib/pacemaker/alert_snmp.sh)
  Meta options: timestamp-format=%Y-%m-%d,%H:%M:%S.%01N.
  Recipients:
   Recipient: snmp_alert-recipient (value=192.168.1.2)
```

L'exemple suivant installe l'agent **alert_smtp.sh**, puis configure une alerte qui utilise l'agent d'alerte installé pour envoyer des événements de cluster sous forme de messages électroniques. Après avoir configuré l'alerte, cet exemple configure un destinataire et affiche la configuration de l'alerte.

```
# install --mode=0755 /usr/share/pacemaker/alerts/alert_smtp.sh.sample
/var/lib/pacemaker/alert_smtp.sh
# pcs alert create id=smtp_alert path=/var/lib/pacemaker/alert_smtp.sh options
email_sender=donotreply@example.com
# pcs alert recipient add smtp_alert value=admin@example.com
# pcs alert
Alerts:
 Alert: smtp_alert (path=/var/lib/pacemaker/alert_smtp.sh)
  Options: email_sender=donotreply@example.com
  Recipients:
   Recipient: smtp_alert-recipient (value=admin@example.com)
```

28.2. CRÉATION D'UNE ALERTE DE CLUSTER

La commande suivante crée une alerte de cluster. Les options que vous configurez sont des valeurs de configuration spécifiques à l'agent qui sont transmises au script de l'agent d'alerte au chemin d'accès que vous spécifiez en tant que variables d'environnement supplémentaires. Si vous ne spécifiez pas de valeur pour **id**, une valeur sera générée.

```
pcs alert create path=path [id=alert-id] [description=description] [options [option=value]...] [meta
[meta-option=value]...]
```

Plusieurs agents d'alerte peuvent être configurés ; la grappe les appellera tous pour chaque événement. Les agents d'alerte ne seront appelés que sur les nœuds de la grappe. Ils seront appelés pour les événements impliquant des nœuds distants Pacemaker, mais ils ne seront jamais appelés sur ces nœuds.

L'exemple suivant crée une alerte simple qui appellera **myscript.sh** pour chaque événement.

```
# pcs alert create id=my_alert path=/path/to/myscript.sh
```

28.3. AFFICHAGE, MODIFICATION ET SUPPRESSION DES ALERTES DE CLUSTER

Il existe une variété de commandes **pcs** que vous pouvez utiliser pour afficher, modifier et supprimer les alertes de cluster.

La commande suivante affiche toutes les alertes configurées ainsi que les valeurs des options configurées.

```
pcs alert [config|show]
```

La commande suivante met à jour une alerte existante avec la valeur *alert-id* spécifiée.

```
pcs alert update alert-id [path=path] [description=description] [options [option=value]...] [meta [meta-option=value]...]
```

La commande suivante supprime une alerte avec la valeur *alert-id* spécifiée.

```
pcs alert remove alert-id
```

Vous pouvez également exécuter la commande **pcs alert delete**, qui est identique à la commande **pcs alert remove**. Les commandes **pcs alert delete** et **pcs alert remove** vous permettent de spécifier plusieurs signalements à supprimer.

28.4. CONFIGURATION DES DESTINATAIRES DES ALERTES DE CLUSTER

En général, les alertes sont dirigées vers un destinataire. Chaque alerte peut donc être configurée avec un ou plusieurs destinataires. Le cluster appellera l'agent séparément pour chaque destinataire.

Le destinataire peut être tout ce que l'agent d'alerte peut reconnaître : une adresse IP, une adresse électronique, un nom de fichier ou tout autre élément pris en charge par l'agent en question.

La commande suivante ajoute un nouveau destinataire à l'alerte spécifiée.

```
pcs alert recipient add alert-id value=recipient-value [id=recipient-id] [description=description] [options [option=value]...] [meta [meta-option=value]...]
```

La commande suivante met à jour un destinataire d'alerte existant.

```
pcs alert recipient update recipient-id [value=recipient-value] [description=description] [options [option=value]...] [meta [meta-option=value]...]
```

La commande suivante supprime le destinataire de l'alerte spécifié.

```
pcs alerte destinataire supprimer recipient-id
```

Vous pouvez également exécuter la commande **pcs alert recipient delete**, qui est identique à la commande **pcs alert recipient remove**. Les commandes **pcs alert recipient remove** et **pcs alert recipient delete** vous permettent de supprimer plusieurs destinataires d'alerte.

L'exemple de commande suivant ajoute le destinataire de l'alerte **my-alert-recipient** avec un ID de destinataire de **my-recipient-id** à l'alerte **my-alert**. Le cluster sera ainsi configuré pour appeler le script

d'alerte configuré pour **my-alert** à chaque événement, en transmettant le destinataire **some-address** en tant que variable d'environnement.

```
# pcs alert recipient add my-alert value=my-alert-recipient id=my-recipient-id options
value=some-address
```

28.5. OPTIONS MÉTA D'ALERTE

Comme pour les agents de ressources, des méta-options peuvent être configurées pour les agents d'alerte afin d'affecter la manière dont Pacemaker les appelle. Le tableau suivant décrit les méta-options d'alerte. Les méta-options peuvent être configurées par agent d'alerte ainsi que par destinataire.

Tableau 28.1. Alert Meta Options

Méta-Attribut	Défaut	Description
timestamp-format	%H:%M:%S.N	Format que le cluster utilisera pour envoyer l'horodatage de l'événement à l'agent. Il s'agit d'une chaîne de caractères utilisée avec la commande **date**(1).
timeout	30s	Si l'agent d'alerte ne se termine pas dans ce délai, il sera interrompu.

L'exemple suivant configure une alerte qui appelle le script **myscript.sh** et ajoute ensuite deux destinataires à l'alerte. Le premier destinataire a pour identifiant **my-alert-recipient1** et le second **my-alert-recipient2**. Le script sera appelé deux fois pour chaque événement, chaque appel étant assorti d'un délai d'attente de 15 secondes. Un appel sera transmis au destinataire **someuser@example.com** avec un horodatage au format %H:%M, tandis que l'autre appel sera transmis au destinataire **otheruser@example.com** avec un horodatage au format

```
# pcs alert create id=my-alert path=/path/to/myscript.sh meta timeout=15s
# pcs alert recipient add my-alert value=someuser@example.com id=my-alert-recipient1 meta
timestamp-format="%D %H:%M"
# pcs alert recipient add my-alert value=otheruser@example.com id=my-alert-recipient2 meta
timestamp-format="%c"
```

28.6. EXEMPLES DE COMMANDES DE CONFIGURATION DES ALERTES DE CLUSTER

Les exemples suivants présentent quelques commandes de base de configuration des alertes afin de montrer le format à utiliser pour créer des alertes, ajouter des destinataires et afficher les alertes configurées.

Notez que si vous devez installer les agents d'alerte eux-mêmes sur chaque nœud d'un cluster, vous ne devez exécuter les commandes **pcs** qu'une seule fois.

Les commandes suivantes créent une alerte simple, ajoutent deux destinataires à l'alerte et affichent les valeurs configurées.

- Comme aucune valeur d'ID d'alerte n'est spécifiée, le système crée une valeur d'ID d'alerte de **alert**.
- La première commande de création de destinataire spécifie le destinataire **rec_value**. Comme cette commande ne spécifie pas d'ID de destinataire, la valeur de **alert-recipient** est utilisée comme ID de destinataire.
- La deuxième commande de création de destinataire spécifie le destinataire **rec_value2**. Cette commande spécifie l'ID de destinataire **my-recipient** pour le destinataire.

```
# pcs alert create path=/my/path
# pcs alert recipient add alert value=rec_value
# pcs alert recipient add alert value=rec_value2 id=my-recipient
# pcs alert config
Alerts:
 Alert: alert (path=/my/path)
  Recipients:
   Recipient: alert-recipient (value=rec_value)
   Recipient: my-recipient (value=rec_value2)
```

Les commandes suivantes ajoutent une deuxième alerte et un destinataire pour cette alerte. L'identifiant de la deuxième alerte est **my-alert** et la valeur du destinataire est **my-other-recipient**. Comme aucun ID de destinataire n'est spécifié, le système fournit un ID de destinataire de **my-alert-recipient**.

```
# pcs alert create id=my-alert path=/path/to/script description=alert_description options
option1=value1 opt=val meta timeout=50s timestamp-format="%H%B%S"
# pcs alert recipient add my-alert value=my-other-recipient
# pcs alert
Alerts:
 Alert: alert (path=/my/path)
  Recipients:
   Recipient: alert-recipient (value=rec_value)
   Recipient: my-recipient (value=rec_value2)
 Alert: my-alert (path=/path/to/script)
  Description: alert_description
  Options: opt=val option1=value1
  Meta options: timestamp-format=%H%B%S timeout=50s
  Recipients:
   Recipient: my-alert-recipient (value=my-other-recipient)
```

Les commandes suivantes modifient les valeurs de l'alerte **my-alert** et du destinataire **my-alert-recipient**.

```
# pcs alert update my-alert options option1=newvalue1 meta timestamp-format="%H%M%S"
# pcs alert recipient update my-alert-recipient options option1=new meta timeout=60s
# pcs alert
Alerts:
 Alert: alert (path=/my/path)
  Recipients:
   Recipient: alert-recipient (value=rec_value)
   Recipient: my-recipient (value=rec_value2)
```

```
 Alert: my-alert (path=/path/to/script)
  Description: alert_description
  Options: opt=val option1=newvalue1
  Meta options: timestamp-format=%H%M%S timeout=50s
  Recipients:
   Recipient: my-alert-recipient (value=my-other-recipient)
    Options: option1=new
    Meta options: timeout=60s
```

La commande suivante supprime le destinataire **my-alert-recipient** de **alert**.

```
# pcs alert recipient remove my-recipient
# pcs alert
Alerts:
 Alert: alert (path=/my/path)
  Recipients:
   Recipient: alert-recipient (value=rec_value)
 Alert: my-alert (path=/path/to/script)
  Description: alert_description
  Options: opt=val option1=newvalue1
  Meta options: timestamp-format="%M%B%S" timeout=50s
  Recipients:
   Recipient: my-alert-recipient (value=my-other-recipient)
    Options: option1=new
    Meta options: timeout=60s
```

La commande suivante supprime **myalert** de la configuration.

```
# pcs alert remove myalert
# pcs alert
Alerts:
 Alert: alert (path=/my/path)
  Recipients:
   Recipient: alert-recipient (value=rec_value)
```

28.7. ÉCRITURE D'UN AGENT D'ALERTE POUR LES CLUSTERS

Il existe trois types d'alertes de grappes Pacemaker : les alertes de nœuds, les alertes de clôtures et les alertes de ressources. Les variables d'environnement transmises aux agents d'alerte peuvent varier en fonction du type d'alerte. Le tableau suivant décrit les variables d'environnement transmises aux agents d'alerte et précise quand la variable d'environnement est associée à un type d'alerte spécifique.

Tableau 28.2. Variables d'environnement transmises aux agents d'alerte

Variable d'environnement	Description
CRM_alert_kind	Le type d'alerte (nœud, clôture ou ressource)
CRM_alert_version	La version de Pacemaker qui envoie l'alerte
CRM_alert_recipient	Le destinataire configuré

Variable d'environnement	Description
CRM_alert_node_sequence	Un numéro de séquence augmente chaque fois qu'une alerte est émise sur le nœud local, ce qui peut être utilisé pour référencer l'ordre dans lequel les alertes ont été émises par Pacemaker. Une alerte pour un événement qui s'est produit plus tard dans le temps a, de manière fiable, un numéro de séquence plus élevé que les alertes pour des événements antérieurs. Sachez que ce numéro n'a aucune signification à l'échelle de la grappe.
CRM_alert_timestamp	Un horodatage créé avant l'exécution de l'agent, dans le format spécifié par la méta-option **timestamp-format**. Cela permet à l'agent de disposer d'une date fiable et précise de l'événement, indépendamment du moment où l'agent lui-même a été invoqué (ce qui pourrait être retardé en raison de la charge du système ou d'autres circonstances).
CRM_alert_node	Nom du nœud affecté
CRM_alert_desc	Détail de l'événement. Pour les alertes de nœuds, il s'agit de l'état actuel du nœud (membre ou perdu). Pour les alertes de clôture, il s'agit d'un résumé de l'opération de clôture demandée, y compris l'origine, la cible et le code d'erreur de l'opération de clôture, le cas échéant. Pour les alertes relatives aux ressources, il s'agit d'une chaîne lisible équivalente à **CRM_alert_status**.
CRM_alert_nodeid	ID du nœud dont l'état a changé (fourni avec les alertes de nœuds uniquement)
CRM_alert_task	L'opération de clôture ou de ressource demandée (fournie uniquement avec les alertes de clôture et de ressource)
CRM_alert_rc	Le code de retour numérique de l'opération de clôture ou de ressource (fourni uniquement avec les alertes de clôture et de ressource)
CRM_alert_rsc	Le nom de la ressource affectée (alertes sur les ressources uniquement)
CRM_alert_interval	Intervalle de l'opération sur la ressource (alertes sur les ressources uniquement)
CRM_alert_target_rc	Code de retour numérique attendu de l'opération (alertes sur les ressources uniquement)
CRM_alert_status	Code numérique utilisé par Pacemaker pour représenter le résultat de l'opération (alertes sur les ressources uniquement)

Lors de la rédaction d'un agent d'alerte, vous devez tenir compte des éléments suivants.

- Les agents d'alerte peuvent être appelés sans destinataire (si aucun n'est configuré) ; l'agent doit donc être capable de gérer cette situation, même s'il ne fait que sortir dans ce cas. Les utilisateurs peuvent modifier la configuration par étapes et ajouter un destinataire ultérieurement.

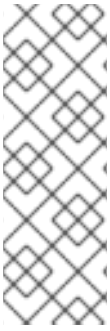
- Si plusieurs destinataires sont configurés pour une alerte, l'agent d'alerte sera appelé une fois par destinataire. Si un agent n'est pas en mesure de fonctionner simultanément, il doit être configuré avec un seul destinataire. L'agent est toutefois libre d'interpréter le destinataire comme une liste.

- Lorsqu'un événement se produit dans un cluster, toutes les alertes sont déclenchées en même temps en tant que processus distincts. En fonction du nombre d'alertes et de destinataires configurés et de ce qui est fait au sein des agents d'alerte, il peut se produire une augmentation significative de la charge. L'agent pourrait être écrit de manière à prendre cela en considération, par exemple en mettant en file d'attente les actions gourmandes en ressources dans une autre instance, au lieu de les exécuter directement.

- Les agents d'alerte sont exécutés sous l'utilisateur **hacluster**, qui dispose d'un ensemble minimal d'autorisations. Si un agent a besoin de privilèges supplémentaires, il est recommandé de configurer **sudo** pour permettre à l'agent d'exécuter les commandes nécessaires en tant qu'autre utilisateur disposant des privilèges appropriés.

- Prenez soin de valider et d'assainir les paramètres configurés par l'utilisateur, tels que **CRM_alert_timestamp** (dont le contenu est spécifié par le paramètre **timestamp-format** configuré par l'utilisateur), **CRM_alert_recipient**, et toutes les options d'alerte. Cela est nécessaire pour se prémunir contre les erreurs de configuration. En outre, si un utilisateur peut modifier la CIB sans disposer d'un accès de niveau **hacluster** aux nœuds du cluster, il s'agit également d'un problème de sécurité potentiel et vous devez éviter la possibilité d'une injection de code.

- Si un cluster contient des ressources avec des opérations pour lesquelles le paramètre **on-fail** est défini sur **fence**, il y aura plusieurs notifications de clôture en cas d'échec, une pour chaque ressource pour laquelle ce paramètre est défini plus une notification supplémentaire. Les notifications seront envoyées à la fois par **pacemaker-fenced** et **pacemaker-controld**. Dans ce cas, Pacemaker n'effectue qu'une seule opération de clôture, quel que soit le nombre de notifications envoyées.



NOTE

L'interface des alertes est conçue pour être rétro-compatible avec l'interface des scripts externes utilisée par la ressource **ocf:pacemaker:ClusterMon**. Pour préserver cette compatibilité, les variables d'environnement transmises aux agents d'alerte sont disponibles avec les préfixes **CRM_notify_** et **CRM_alert_**. Une rupture de compatibilité est due au fait que la ressource **ClusterMon** exécutait les scripts externes en tant qu'utilisateur root, alors que les agents d'alerte sont exécutés en tant qu'utilisateur **hacluster**.

CHAPITRE 29. GRAPPES DE STIMULATEURS CARDIAQUES MULTISITES

Lorsqu'un cluster s'étend sur plus d'un site, les problèmes de connectivité réseau entre les sites peuvent conduire à des situations de "split brain" (cerveau divisé). Lorsque la connectivité est interrompue, il n'y a aucun moyen pour un nœud sur un site de déterminer si un nœud sur un autre site est tombé en panne ou s'il fonctionne toujours avec un lien inter-sites défectueux. En outre, il peut être problématique de fournir des services de haute disponibilité sur deux sites qui sont trop éloignés pour rester synchronisés. Pour résoudre ces problèmes, Pacemaker offre une prise en charge complète de la capacité à configurer des grappes de haute disponibilité qui couvrent plusieurs sites grâce à l'utilisation d'un gestionnaire de tickets de grappe Booth.

29.1. VUE D'ENSEMBLE DU GESTIONNAIRE DE BILLETS EN GRAPPE BOOTH

Le site Booth *ticket manager* est un service distribué destiné à être exécuté sur un réseau physique différent des réseaux qui relient les nœuds de la grappe sur des sites particuliers. Il produit une autre grappe souple, *Booth formation*, qui se superpose aux grappes régulières des sites. Cette couche de communication agrégée facilite les processus de décision basés sur le consensus pour les tickets individuels de Booth.

Un Booth *ticket* est un singleton dans la formation Booth et représente une unité d'autorisation mobile et sensible au temps. Les ressources peuvent être configurées de manière à nécessiter un certain ticket pour être exécutées. Cela permet de s'assurer que les ressources ne sont exécutées que sur un seul site à la fois, pour lequel un ou plusieurs tickets ont été accordés.

Vous pouvez considérer une formation Booth comme un cluster superposé composé de clusters fonctionnant sur différents sites, où tous les clusters d'origine sont indépendants les uns des autres. C'est le service Booth qui indique aux grappes si elles ont reçu un ticket, et c'est Pacemaker qui détermine si des ressources doivent être exécutées dans une grappe sur la base d'une contrainte de ticket Pacemaker. Cela signifie qu'en utilisant le gestionnaire de tickets, chaque groupe peut exécuter ses propres ressources ainsi que des ressources partagées. Par exemple, les ressources A, B et C peuvent être exécutées uniquement dans une grappe, les ressources D, E et F uniquement dans l'autre grappe, et les ressources G et H dans l'une ou l'autre des deux grappes en fonction d'un ticket. Il est également possible d'avoir une ressource supplémentaire J qui peut s'exécuter dans l'un ou l'autre des deux clusters, comme déterminé par un ticket séparé.

29.2. CONFIGURATION DE CLUSTERS MULTI-SITES AVEC PACEMAKER

Vous pouvez configurer une configuration multi-site qui utilise le gestionnaire de tickets Booth en suivant la procédure suivante.

Ces exemples de commandes utilisent la disposition suivante :

- Le groupe 1 est constitué des nœuds **cluster1-node1** et **cluster1-node2**
- L'adresse IP flottante de la grappe 1 est 192.168.11.100
- Le groupe 2 se compose de **cluster2-node1** et **cluster2-node2**
- L'adresse IP flottante de la grappe 2 est 192.168.22.100
- Le nœud d'arbitrage est **arbitrator-node** avec une adresse IP de 192.168.99.100

- Le nom du ticket Booth utilisé dans cette configuration est le suivant **apacheticket**

Ces exemples de commandes supposent que les ressources du cluster pour un service Apache ont été configurées dans le cadre du groupe de ressources **apachegroup** pour chaque cluster. Il n'est pas nécessaire que les ressources et les groupes de ressources soient les mêmes sur chaque cluster pour configurer une contrainte de ticket pour ces ressources, puisque l'instance Pacemaker de chaque cluster est indépendante, mais il s'agit d'un scénario de basculement courant.

Notez qu'à tout moment de la procédure de configuration, vous pouvez entrer la commande **pcs booth config** pour afficher la configuration de la cabine pour le nœud ou la grappe en cours ou la commande **pcs booth status** pour afficher l'état actuel de la cabine sur le nœud local.

Procédure

1. Installez le paquetage **booth-site** Booth ticket manager sur chaque nœud des deux clusters.

```
[root@cluster1-node1 ~]# dnf install -y booth-site
[root@cluster1-node2 ~]# dnf install -y booth-site
[root@cluster2-node1 ~]# dnf install -y booth-site
[root@cluster2-node2 ~]# dnf install -y booth-site
```

2. Installez les paquets **pcs**, **booth-core**, et **booth-arbitrator** sur le nœud de l'arbitre.

```
[root@arbitrator-node ~]# dnf install -y pcs booth-core booth-arbitrator
```

3. Si vous exécutez le démon **firewalld**, exécutez les commandes suivantes sur tous les nœuds des deux clusters ainsi que sur le nœud de l'arbitre afin d'activer les ports requis par Red Hat High Availability Add-On.

```
# firewall-cmd --permanent --add-service=high-availability
# firewall-cmd --add-service=high-availability
```

   Il se peut que vous deviez modifier les ports ouverts en fonction des conditions locales. Pour plus d'informations sur les ports requis par le module complémentaire de haute disponibilité de Red Hat, voir Activation des ports pour le module complémentaire de haute disponibilité .

4. Créez une configuration Booth sur un nœud d'une grappe. Les adresses que vous spécifiez pour chaque grappe et pour l'arbitre doivent être des adresses IP. Pour chaque grappe, vous devez spécifier une adresse IP flottante.

```
[cluster1-node1 ~] # pcs booth setup sites 192.168.11.100 192.168.22.100 arbitrators
192.168.99.100
```

   Cette commande crée les fichiers de configuration **/etc/booth/booth.conf** et **/etc/booth/booth.key** sur le nœud à partir duquel elle est exécutée.

5. Créez un ticket pour la configuration Booth. C'est le ticket que vous utiliserez pour définir la contrainte de ressources qui permettra aux ressources de s'exécuter uniquement lorsque ce ticket a été accordé au cluster.
   Cette procédure de configuration de base du basculement n'utilise qu'un seul ticket, mais vous pouvez en créer d'autres pour des scénarios plus complexes dans lesquels chaque ticket est associé à une ou plusieurs ressources différentes.

```
[cluster1-node1 ~] # pcs booth ticket add apacheticket
```

6. Synchroniser la configuration de Booth avec tous les nœuds de la grappe actuelle.

```
[cluster1-node1 ~] # pcs booth sync
```

7. À partir du nœud de l'arbitre, tirez la configuration de Booth vers l'arbitre. Si vous ne l'avez pas encore fait, vous devez d'abord vous authentifier à l'adresse **pcs** sur le nœud d'où vous tirez la configuration.

```
[arbitrator-node ~] # pcs host auth cluster1-node1
[arbitrator-node ~] # pcs booth pull cluster1-node1
```

8. Transférez la configuration de Booth vers l'autre cluster et synchronisez tous les nœuds de ce cluster. Comme pour le nœud d'arbitrage, si vous ne l'avez pas fait auparavant, vous devez d'abord authentifier **pcs** auprès du nœud à partir duquel vous récupérez la configuration.

```
[cluster2-node1 ~] # pcs host auth cluster1-node1
[cluster2-node1 ~] # pcs booth pull cluster1-node1
[cluster2-node1 ~] # pcs booth sync
```

9. Démarrer et activer Booth sur l'arbitre.

NOTE

Vous ne devez pas démarrer ou activer manuellement Booth sur l'un des nœuds des clusters, car Booth fonctionne en tant que ressource Pacemaker dans ces clusters.

```
[arbitrator-node ~] # pcs booth start
[arbitrator-node ~] # pcs booth enable
```

10. Configurez Booth pour qu'il s'exécute en tant que ressource de cluster sur les deux sites du cluster. Cela crée un groupe de ressources dont les membres sont **booth-ip** et **booth-service**.

```
[cluster1-node1 ~] # pcs booth create ip 192.168.11.100
[cluster2-node1 ~] # pcs booth create ip 192.168.22.100
```

11. Ajoutez une contrainte de ticket au groupe de ressources que vous avez défini pour chaque cluster.

```
[cluster1-node1 ~] # pcs constraint ticket add apacheticket apachegroup
[cluster2-node1 ~] # pcs constraint ticket add apacheticket apachegroup
```

Vous pouvez entrer la commande suivante pour afficher les contraintes de ticket actuellement configurées.

```
pcs constraint ticket [show]
```

12. Accordez le ticket que vous avez créé pour cette configuration au premier cluster.
Notez qu'il n'est pas nécessaire d'avoir défini des contraintes de ticket avant d'accorder un ticket. Une fois que vous avez accordé un ticket à un cluster, Booth prend en charge la gestion des tickets, à moins que vous ne l'annuliez manuellement avec la commande **pcs booth ticket**

revoke. Pour plus d'informations sur les commandes d'administration **pcs booth**, consultez l'écran d'aide PCS pour la commande **pcs booth**.

```
[cluster1-node1 ~] # pcs booth ticket grant apacheticket
```

Il est possible d'ajouter ou de supprimer des tickets à tout moment, même après avoir terminé cette procédure. Cependant, après avoir ajouté ou supprimé un ticket, vous devez synchroniser les fichiers de configuration avec les autres nœuds et clusters, ainsi qu'avec l'arbitre, et accorder le ticket comme indiqué dans cette procédure.

Pour plus d'informations sur les autres commandes d'administration du kiosque que vous pouvez utiliser pour nettoyer et supprimer les fichiers de configuration, les tickets et les ressources du kiosque, consultez l'écran d'aide du PCS pour la commande **pcs booth**.

CHAPITRE 30. INTÉGRATION DE NŒUDS NON COROSYNCHRONES DANS UN CLUSTER : LE SERVICE PACEMAKER_REMOTE

Le service **pacemaker_remote** permet aux nœuds n'exécutant pas **corosync** de s'intégrer dans le cluster et de faire en sorte que le cluster gère leurs ressources comme s'il s'agissait de véritables nœuds de cluster.

Le service **pacemaker_remote** offre notamment les possibilités suivantes :

- Le service **pacemaker_remote** vous permet de dépasser la limite de 32 nœuds fixée par Red Hat.
- Le service **pacemaker_remote** vous permet de gérer un environnement virtuel en tant que ressource de cluster et de gérer des services individuels au sein de l'environnement virtuel en tant que ressources de cluster.

Les termes suivants sont utilisés pour décrire le service **pacemaker_remote**.

- *cluster node* - Un nœud exécutant les services de haute disponibilité (**pacemaker** et **corosync**).
- *remote node* - Un nœud exécutant **pacemaker_remote** pour s'intégrer à distance dans le cluster sans qu'il soit nécessaire d'être membre du cluster **corosync**. Un nœud distant est configuré comme une ressource de cluster qui utilise l'agent de ressource **ocf:pacemaker:remote**.
- *guest node* - Un nœud invité virtuel exécutant le service **pacemaker_remote**. La ressource virtuelle invitée est gérée par le cluster ; elle est à la fois démarrée par le cluster et intégrée dans le cluster en tant que nœud distant.
- *pacemaker_remote* - Un démon de service capable de gérer des applications à distance sur des nœuds distants et des nœuds invités KVM dans un environnement de cluster Pacemaker. Ce service est une version améliorée du démon exécuteur local de Pacemaker (**pacemaker-execd**) qui est capable de gérer les ressources à distance sur un nœud qui n'exécute pas corosync.

Un cluster Pacemaker exécutant le service **pacemaker_remote** présente les caractéristiques suivantes.

- Les nœuds distants et les nœuds invités exécutent le service **pacemaker_remote** (avec très peu de configuration requise du côté de la machine virtuelle).
- La pile du cluster (**pacemaker** et **corosync**), fonctionnant sur les nœuds du cluster, se connecte au service **pacemaker_remote** sur les nœuds distants, ce qui leur permet de s'intégrer dans le cluster.
- La pile de cluster (**pacemaker** et **corosync**), fonctionnant sur les nœuds de cluster, lance les nœuds invités et se connecte immédiatement au service **pacemaker_remote** sur les nœuds invités, ce qui leur permet de s'intégrer dans le cluster.

La principale différence entre les nœuds de cluster et les nœuds distants et invités gérés par les nœuds de cluster est que les nœuds distants et invités n'exécutent pas la pile de cluster. Cela signifie que les nœuds distants et invités ont les limitations suivantes :

- elles n'ont pas lieu au quorum
- ils n'exécutent pas les actions du dispositif de clôture

- ils ne sont pas éligibles au poste de contrôleur désigné (CD) du cluster
- ils n'exécutent pas eux-mêmes toute la gamme des commandes **pcs**

D'autre part, les nœuds distants et les nœuds invités ne sont pas soumis aux limites d'évolutivité associées à la pile du cluster.

Hormis ces limitations, les nœuds distants et invités se comportent comme les nœuds de la grappe en ce qui concerne la gestion des ressources, et les nœuds distants et invités peuvent eux-mêmes être clôturés. Le cluster est tout à fait capable de gérer et de surveiller les ressources sur chaque nœud distant et invité : Vous pouvez leur imposer des contraintes, les mettre en veille ou effectuer toute autre action sur les nœuds de la grappe à l'aide des commandes **pcs**. Les nœuds distants et invités apparaissent dans la sortie d'état de la grappe au même titre que les nœuds de la grappe.

30.1. AUTHENTIFICATION DE L'HÔTE ET DE L'INVITÉ POUR LES NŒUDS PACEMAKER_REMOTE

La connexion entre les nœuds de la grappe et pacemaker_remote est sécurisée à l'aide de Transport Layer Security (TLS) avec chiffrement et authentification par clé pré-partagée (PSK) sur TCP (en utilisant le port 3121 par défaut). Cela signifie que le nœud de cluster et le nœud exécutant **pacemaker_remote** doivent partager la même clé privée. Par défaut, cette clé doit être placée à l'adresse **/etc/pacemaker/authkey** sur les nœuds du cluster et les nœuds distants.

La commande **pcs cluster node add-guest** configure **authkey** pour les nœuds invités et la commande **pcs cluster node add-remote** configure **authkey** pour les nœuds distants.

30.2. CONFIGURATION DES NŒUDS INVITÉS KVM

Un nœud invité Pacemaker est un nœud invité virtuel qui exécute le service **pacemaker_remote**. Le nœud invité virtuel est géré par le cluster.

30.2.1. Options de ressources du nœud invité

Lorsque vous configurez une machine virtuelle pour qu'elle agisse en tant que noeud invité, vous créez une ressource **VirtualDomain** qui gère la machine virtuelle. Pour obtenir une description des options que vous pouvez définir pour une ressource **VirtualDomain**, consultez le tableau "Options de ressources pour les ressources du domaine virtuel" dans Options de ressources du domaine virtuel.

Outre les options de la ressource **VirtualDomain**, les options de métadonnées définissent la ressource en tant que nœud invité et définissent les paramètres de connexion. Vous définissez ces options de ressource avec la commande **pcs cluster node add-guest**. Le tableau suivant décrit ces options de métadonnées.

Tableau 30.1. Options de métadonnées pour la configuration des ressources KVM en tant que nœuds distants

Field	Défaut	Description

Field	Défaut	Description
remote-node	<none>	Le nom du nœud invité que cette ressource définit. Cela permet à la fois d'activer la ressource en tant que nœud invité et de définir le nom unique utilisé pour identifier le nœud invité. *WARNING*: Cette valeur ne peut pas se chevaucher avec des identifiants de ressources ou de nœuds.
remote-port	3121	Configure un port personnalisé à utiliser pour la connexion de l'invité à **pacemaker_remote**
remote-addr	L'adresse fournie dans la commande **pcs host auth**	L'adresse IP ou le nom d'hôte à laquelle se connecter
remote-connect-timeout	60s	Délai d'attente avant qu'une connexion d'invité en attente ne soit interrompue

30.2.2. Intégration d'une machine virtuelle en tant que nœud invité

La procédure suivante est un résumé de haut niveau des étapes à suivre pour que Pacemaker lance une machine virtuelle et intègre cette machine en tant que nœud invité, en utilisant **libvirt** et les invités virtuels KVM.

Procédure

1. Configurer les ressources **VirtualDomain**.
2. Entrez les commandes suivantes sur chaque machine virtuelle pour installer les paquets **pacemaker_remote**, démarrer le service **pcsd** et lui permettre de s'exécuter au démarrage, et autoriser le port TCP 3121 à travers le pare-feu.

```
# dnf install pacemaker-remote resource-agents pcs
# systemctl start pcsd.service
# systemctl enable pcsd.service
# firewall-cmd --add-port 3121/tcp --permanent
# firewall-cmd --add-port 2224/tcp --permanent
# firewall-cmd --reload
```

3. Attribuez à chaque machine virtuelle une adresse réseau statique et un nom d'hôte unique, qui doivent être connus de tous les nœuds.
4. Si vous ne l'avez pas encore fait, authentifiez-vous sur **pcs** auprès du nœud que vous allez intégrer en tant que nœud de quête.

```
# pcs host auth nodename
```

5. Utilisez la commande suivante pour convertir une ressource **VirtualDomain** existante en un nœud invité. Cette commande doit être exécutée sur un nœud de cluster et non sur le nœud invité qui est ajouté. Outre la conversion de la ressource, cette commande copie le site **/etc/pacemaker/authkey** sur le nœud invité et démarre et active le démon **pacemaker_remote** sur le nœud invité. Le nom du nœud invité, que vous pouvez définir arbitrairement, peut être différent du nom d'hôte du nœud.

```
# pcs cluster node add-guest nodename resource_id [options]
```

6. Après avoir créé la ressource **VirtualDomain**, vous pouvez traiter le nœud invité comme n'importe quel autre nœud du cluster. Par exemple, vous pouvez créer une ressource et placer une contrainte de ressource sur la ressource à exécuter sur le nœud invité, comme dans les commandes suivantes, qui sont exécutées à partir d'un nœud de cluster. Vous pouvez inclure des nœuds invités dans des groupes, ce qui vous permet de regrouper un périphérique de stockage, un système de fichiers et une VM.

```
# pcs resource create webserver apache configfile=/etc/httpd/conf/httpd.conf op
monitor interval=30s
# pcs constraint location webserver prefers nodename
```

30.3. CONFIGURATION DES NŒUDS DISTANTS PACEMAKER

Un nœud distant est défini comme une ressource de cluster avec **ocf:pacemaker:remote** comme agent de ressource. Vous créez cette ressource avec la commande **pcs cluster node add-remote**.

30.3.1. Options de ressources du nœud distant

Le tableau suivant décrit les options de ressources que vous pouvez configurer pour une ressource **remote**.

Tableau 30.2. Options de ressources pour les nœuds distants

Field	Défaut	Description
reconnect_interval	0	Temps d'attente en secondes avant de tenter de se reconnecter à un nœud distant après la rupture d'une connexion active avec le nœud distant. Cette attente est récurrente. Si la reconnexion échoue après la période d'attente, une nouvelle tentative de reconnexion sera effectuée après avoir respecté le temps d'attente. Lorsque cette option est utilisée, Pacemaker continuera à essayer de se connecter au nœud distant indéfiniment après chaque intervalle d'attente.
server	Adresse spécifiée par la commande **pcs host auth**	Serveur auquel se connecter. Il peut s'agir d'une adresse IP ou d'un nom d'hôte.

Field	Défaut	Description
port		Port TCP auquel se connecter.

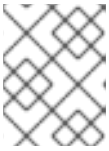
30.3.2. Aperçu de la configuration du nœud distant

La procédure suivante fournit un résumé de haut niveau des étapes à suivre pour configurer un nœud Pacemaker Remote et pour intégrer ce nœud dans un environnement de cluster Pacemaker existant.

Procédure

1. Sur le nœud que vous allez configurer comme nœud distant, autorisez les services liés au cluster à travers le pare-feu local.

```
# firewall-cmd --permanent --add-service=high-availability
success
# firewall-cmd --reload
success
```



NOTE

Si vous utilisez directement **iptables**, ou une autre solution de pare-feu que **firewalld**, ouvrez simplement les ports suivants : Ports TCP 2224 et 3121.

2. Installer le démon **pacemaker_remote** sur le nœud distant.

```
# dnf install -y pacemaker-remote resource-agents pcs
```

3. Démarrer et activer **pcsd** sur le nœud distant.

```
# systemctl start pcsd.service
# systemctl enable pcsd.service
```

4. Si vous ne l'avez pas encore fait, authentifiez **pcs** auprès du nœud que vous allez ajouter en tant que nœud distant.

```
# pcs host auth remote1
```

5. Ajoutez la ressource du nœud distant au cluster à l'aide de la commande suivante. Cette commande synchronise également tous les fichiers de configuration pertinents sur le nouveau nœud, démarre le nœud et le configure pour qu'il démarre **pacemaker_remote** au démarrage. Cette commande doit être exécutée sur un nœud du cluster et non sur le nœud distant qui est ajouté.

```
# pcs cluster node add-remote remote1
```

6. Après avoir ajouté la ressource **remote** au cluster, vous pouvez traiter le nœud distant comme n'importe quel autre nœud du cluster. Par exemple, vous pouvez créer une ressource et placer une contrainte de ressource sur la ressource à exécuter sur le nœud distant, comme dans les commandes suivantes, qui sont exécutées à partir d'un nœud de cluster.

```
# pcs resource create webserver apache configfile=/etc/httpd/conf/httpd.conf op
monitor interval=30s
# pcs constraint location webserver prefers remote1
```

AVERTISSEMENT

Ne jamais impliquer une ressource de connexion de nœud distant dans un groupe de ressources, une contrainte de colocation ou une contrainte d'ordre.

7. Configurez les ressources de clôture pour le nœud distant. Les nœuds distants sont clôturés de la même manière que les nœuds de grappe. Configurez les ressources de clôture à utiliser avec les nœuds distants de la même manière qu'avec les nœuds de grappe. Notez toutefois que les nœuds distants ne peuvent jamais initier une action de clôture. Seuls les nœuds de grappe sont capables d'exécuter une opération de clôture contre un autre nœud.

30.4. MODIFIER L'EMPLACEMENT DU PORT PAR DÉFAUT

Si vous devez modifier l'emplacement du port par défaut de Pacemaker ou de **pacemaker_remote**, vous pouvez définir la variable d'environnement **PCMK_remote_port** qui affecte ces deux démons. Cette variable d'environnement peut être activée en la plaçant dans le fichier **/etc/sysconfig/pacemaker** comme suit.

```
\#==#==# Pacemaker Remote
...
#
# Specify a custom port for Pacemaker Remote connections
PCMK_remote_port=3121
```

Pour modifier le port par défaut utilisé par un nœud invité ou un nœud distant particulier, la variable **PCMK_remote_port** doit être définie dans le fichier **/etc/sysconfig/pacemaker** de ce nœud, et la ressource cluster créant la connexion au nœud invité ou au nœud distant doit également être configurée avec le même numéro de port (à l'aide de l'option de métadonnées **remote-port** pour les nœuds invités, ou de l'option **port** pour les nœuds distants).

30.5. MISE À NIVEAU DES SYSTÈMES AVEC DES NŒUDS PACEMAKER_REMOTE

Si le service **pacemaker_remote** est arrêté sur un nœud Pacemaker Remote actif, le cluster migre les ressources hors du nœud avant d'arrêter le nœud. Cela vous permet d'effectuer des mises à jour logicielles et d'autres procédures de maintenance de routine sans retirer le nœud de la grappe. Cependant, une fois que **pacemaker_remote** est arrêté, le cluster tente immédiatement de se reconnecter. Si **pacemaker_remote** n'est pas redémarré dans le délai de surveillance de la ressource, le cluster considérera que l'opération de surveillance a échoué.

Si vous souhaitez éviter les pannes de moniteur lorsque le service **pacemaker_remote** est arrêté sur un nœud Pacemaker Remote actif, vous pouvez utiliser la procédure suivante pour retirer le nœud du cluster avant d'effectuer toute administration système susceptible d'arrêter **pacemaker_remote**.

Procédure

1. Arrêtez la ressource de connexion du nœud avec la commande **pcs resource disable *resourcename*** ce qui aura pour effet de déplacer tous les services hors du nœud. La ressource de connexion est la ressource **ocf:pacemaker:remote** pour un nœud distant ou, plus couramment, la ressource **ocf:heartbeat:VirtualDomain** pour un nœud invité. Pour les nœuds invités, cette commande arrête également la VM, qui doit donc être démarrée en dehors du cluster (par exemple, à l'aide de **virsh**) pour effectuer toute opération de maintenance.

   ```
   désactivation des ressources pcs resourcename
   ```

2. Effectuer les opérations de maintenance nécessaires.

3. Lorsque le nœud est prêt à être réintégré dans le cluster, réactivez la ressource à l'aide de la commande **pcs resource enable**.

   ```
   activation des ressources pcs resourcename
   ```

CHAPITRE 31. MAINTENANCE DE LA GRAPPE

Pour effectuer la maintenance des nœuds de votre grappe, il se peut que vous deviez arrêter ou déplacer les ressources et les services fonctionnant sur cette grappe. Il se peut également que vous deviez arrêter le logiciel de la grappe tout en laissant les services intacts. Pacemaker propose plusieurs méthodes pour effectuer la maintenance du système.

- Si vous devez arrêter un nœud d'une grappe tout en continuant à fournir les services de cette grappe sur un autre nœud, vous pouvez mettre le nœud de la grappe en mode veille. Un nœud en mode veille n'est plus en mesure d'héberger des ressources. Toute ressource actuellement active sur le nœud sera déplacée vers un autre nœud ou arrêtée si aucun autre nœud n'est en mesure d'exécuter la ressource. Pour plus d'informations sur le mode veille, voir Mise en veille d'un nœud.

- Si vous devez déplacer une ressource individuelle hors du nœud sur lequel elle s'exécute actuellement sans l'arrêter, vous pouvez utiliser la commande **pcs resource move** pour déplacer la ressource vers un nœud différent.
  Lorsque vous exécutez la commande **pcs resource move**, vous ajoutez une contrainte à la ressource pour l'empêcher de s'exécuter sur le nœud sur lequel elle s'exécute actuellement. Lorsque vous êtes prêt à déplacer la ressource, vous pouvez exécuter la commande **pcs resource clear** ou **pcs constraint delete** pour supprimer la contrainte. Cependant, cela ne ramène pas nécessairement les ressources sur le nœud d'origine, car l'endroit où les ressources peuvent s'exécuter à ce moment-là dépend de la manière dont vous avez configuré vos ressources au départ. Vous pouvez relocaliser une ressource vers son nœud préféré à l'aide de la commande **pcs resource relocate run**.

- Si vous devez arrêter complètement une ressource en cours d'exécution et empêcher le cluster de la redémarrer, vous pouvez utiliser la commande **pcs resource disable**. Pour plus d'informations sur la commande **pcs resource disable**, reportez-vous à la section Désactivation, activation et interdiction des ressources de cluster .

- Si vous souhaitez empêcher Pacemaker de prendre des mesures pour une ressource (par exemple, si vous souhaitez désactiver les actions de récupération lors de la maintenance de la ressource ou si vous devez recharger les paramètres **/etc/sysconfig/pacemaker**), utilisez la commande **pcs resource unmanage**, comme décrit dans la section Définition d'une ressource en mode non géré. Les ressources de connexion à distance de Pacemaker ne doivent jamais être non gérées.

- Si vous avez besoin de mettre le cluster dans un état où aucun service ne sera démarré ou arrêté, vous pouvez définir la propriété **maintenance-mode** cluster. Le passage du cluster en mode maintenance entraîne automatiquement la suppression de la gestion de toutes les ressources. Pour plus d'informations sur la mise en mode maintenance de la grappe, voir Mise en mode maintenance d'une grappe.

- Si vous devez mettre à jour les paquets qui composent les modules complémentaires RHEL High Availability et Resilient Storage, vous pouvez le faire sur un nœud à la fois ou sur l'ensemble de la grappe, comme indiqué dans la section Mise à jour d'une grappe RHEL haute disponibilité .

- Si vous devez effectuer des opérations de maintenance sur un nœud distant Pacemaker, vous pouvez retirer ce nœud du cluster en désactivant la ressource nœud distant, comme décrit dans la section Mise à niveau des nœuds distants et des nœuds invités .

- Si vous devez migrer une VM dans un cluster RHEL, vous devrez d'abord arrêter les services de cluster sur la VM pour supprimer le nœud du cluster, puis redémarrer le cluster après avoir effectué la migration, comme décrit dans la section Migration de VM dans un cluster RHEL .

31.1. MISE EN VEILLE D'UN NŒUD

Lorsqu'un nœud de cluster est en mode veille, il n'est plus en mesure d'héberger des ressources. Toutes les ressources actuellement actives sur le nœud seront déplacées vers un autre nœud.

La commande suivante met le nœud spécifié en mode veille. Si vous spécifiez l'adresse **--all**, cette commande met tous les nœuds en mode veille.

Vous pouvez utiliser cette commande pour mettre à jour les paquets d'une ressource. Vous pouvez également utiliser cette commande pour tester une configuration, afin de simuler une reprise sans arrêter réellement un nœud.

```
pcs node standby node | --all
```

La commande suivante supprime le nœud spécifié du mode veille. Après l'exécution de cette commande, le nœud spécifié est en mesure d'héberger des ressources. Si vous spécifiez l'adresse **--all**, cette commande supprime tous les nœuds du mode veille.

```
pcs node unstandby node | --all
```

Notez que l'exécution de la commande **pcs node standby** empêche l'exécution des ressources sur le nœud indiqué. Lorsque vous exécutez la commande **pcs node unstandby**, cela permet aux ressources de s'exécuter sur le nœud indiqué. Cela ne ramène pas nécessairement les ressources sur le nœud indiqué ; l'endroit où les ressources peuvent s'exécuter à ce moment-là dépend de la manière dont vous avez configuré vos ressources au départ.

31.2. DÉPLACEMENT MANUEL DES RESSOURCES DE LA GRAPPE

Vous pouvez passer outre le cluster et forcer les ressources à quitter leur emplacement actuel. Il y a deux cas où vous voudrez faire cela :

- Lorsqu'un nœud fait l'objet d'une maintenance et que vous devez déplacer toutes les ressources fonctionnant sur ce nœud vers un autre nœud
- Lorsque des ressources spécifiées individuellement doivent être déplacées

Pour déplacer toutes les ressources en cours d'exécution sur un nœud vers un autre nœud, vous mettez le nœud en mode veille.

Vous pouvez déplacer des ressources spécifiées individuellement de l'une ou l'autre des manières suivantes.

- Vous pouvez utiliser la commande **pcs resource move** pour déplacer une ressource hors d'un nœud sur lequel elle est en cours d'exécution.
- Vous pouvez utiliser la commande **pcs resource relocate run** pour déplacer une ressource vers son nœud préféré, en fonction de l'état actuel du cluster, des contraintes, de l'emplacement des ressources et d'autres paramètres.

31.2.1. Déplacement d'une ressource à partir de son nœud actuel

Pour déplacer une ressource hors du nœud sur lequel elle est actuellement exécutée, utilisez la commande suivante, en spécifiant l'adresse *resource_id* de la ressource telle qu'elle est définie. Spécifiez le **destination_node** si vous souhaitez indiquer sur quel nœud exécuter la ressource que vous déplacez.

```
pcs resource move resource_id [destination_node] [--promu] [--strict] [--attendu[=n]]]
```

Lorsque vous exécutez la commande **pcs resource move**, celle-ci ajoute une contrainte à la ressource pour l'empêcher de s'exécuter sur le nœud sur lequel elle s'exécute actuellement. Par défaut, la contrainte d'emplacement créée par la commande est automatiquement supprimée une fois que la ressource a été déplacée. Si la suppression de la contrainte entraîne le retour de la ressource sur le nœud d'origine, ce qui pourrait se produire si la valeur **resource-stickiness** de la ressource est égale à 0, la commande **pcs resource move** échoue. Si vous souhaitez déplacer une ressource et laisser la contrainte en place, utilisez la commande **pcs resource move-with-constraint**.

Si vous spécifiez le paramètre **--promoted** de la commande **pcs resource move**, la contrainte ne s'applique qu'aux instances promues de la ressource.

Si vous spécifiez le paramètre **--strict** de la commande **pcs resource move**, la commande échouera si d'autres ressources que celles spécifiées dans la commande sont affectées.

Vous pouvez éventuellement configurer un paramètre **--wait[=*n*]** pour la commande **pcs resource move** afin d'indiquer le nombre de secondes à attendre pour que la ressource démarre sur le nœud de destination avant de renvoyer 0 si la ressource est démarrée ou 1 si la ressource n'a pas encore démarré. Si vous ne spécifiez pas n, la valeur par défaut est de 60 minutes.

31.2.2. Déplacement d'une ressource vers son nœud préféré

Une fois qu'une ressource a été déplacée, soit en raison d'un basculement, soit parce qu'un administrateur a déplacé manuellement le nœud, elle ne retournera pas nécessairement à son nœud d'origine, même si les circonstances qui ont provoqué le basculement ont été corrigées. Pour déplacer les ressources vers leur nœud préféré, utilisez la commande suivante. Le nœud préféré est déterminé par l'état actuel du cluster, les contraintes, l'emplacement des ressources et d'autres paramètres, et peut changer au fil du temps.

```
pcs resource relocate run [resource1] [resource2] ...
```

Si vous ne spécifiez aucune ressource, toutes les ressources sont déplacées vers leurs nœuds préférés.

Cette commande calcule le nœud préféré de chaque ressource tout en ignorant l'adhérence des ressources. Après avoir calculé le nœud préféré, elle crée des contraintes de localisation qui amèneront les ressources à se déplacer vers leurs nœuds préférés. Une fois les ressources déplacées, les contraintes sont automatiquement supprimées. Pour supprimer toutes les contraintes créées par la commande **pcs resource relocate run**, vous pouvez entrer la commande **pcs resource relocate clear**. Pour afficher l'état actuel des ressources et leur nœud optimal en ignorant l'adhérence des ressources, entrez la commande **pcs resource relocate show**.

31.3. DÉSACTIVATION, ACTIVATION ET INTERDICTION DES RESSOURCES DE LA GRAPPE

Outre les commandes **pcs resource move** et **pcs resource relocate**, il existe un certain nombre d'autres commandes que vous pouvez utiliser pour contrôler le comportement des ressources de la grappe.

Désactivation d'une ressource de cluster

Vous pouvez arrêter manuellement une ressource en cours d'exécution et empêcher le cluster de la redémarrer avec la commande suivante. En fonction du reste de la configuration (contraintes, options, échecs, etc.), la ressource peut rester démarrée. Si vous spécifiez l'option **--wait**, **pcs** attendra jusqu'à 'n'

secondes que la ressource s'arrête et renverra 0 si la ressource est arrêtée ou 1 si la ressource ne s'est pas arrêtée. Si "n" n'est pas spécifié, la valeur par défaut est de 60 minutes.

```
pcs resource disable resource_id [--wait[=n]]]
```

Vous pouvez spécifier qu'une ressource ne doit être désactivée que si cette désactivation n'a pas d'effet sur d'autres ressources. Il peut être impossible de s'en assurer à la main lorsque des relations complexes sont établies entre les ressources.

- La commande **pcs resource disable --simulate** montre les effets de la désactivation d'une ressource sans modifier la configuration du cluster.
- La commande **pcs resource disable --safe** ne désactive une ressource que si aucune autre ressource n'est affectée de quelque manière que ce soit, par exemple en cas de migration d'un nœud à un autre. La commande **pcs resource safe-disable** est un alias de la commande **pcs resource disable --safe**.
- La commande **pcs resource disable --safe --no-strict** ne désactive une ressource que si aucune autre ressource n'est arrêtée ou rétrogradée

Vous pouvez spécifier l'option **--brief** pour que la commande **pcs resource disable --safe** n'imprime que les erreurs. Le rapport d'erreur généré par la commande **pcs resource disable --safe** en cas d'échec de l'opération de désactivation sécurisée contient les identifiants des ressources concernées. Si vous souhaitez connaître uniquement les identifiants des ressources qui seraient affectées par la désactivation d'une ressource, utilisez l'option **--brief**, qui ne fournit pas le résultat complet de la simulation.

Activation d'une ressource de cluster

Utilisez la commande suivante pour permettre au cluster de démarrer une ressource. En fonction du reste de la configuration, la ressource peut rester arrêtée. Si vous spécifiez l'option **--wait**, **pcs** attendra jusqu'à 'n' secondes pour que la ressource démarre et renverra 0 si la ressource est démarrée ou 1 si la ressource n'a pas démarré. Si 'n' n'est pas spécifié, la valeur par défaut est de 60 minutes.

```
pcs resource enable resource_id [--wait[=n]]]
```

Empêcher l'exécution d'une ressource sur un nœud particulier

La commande suivante permet d'empêcher l'exécution d'une ressource sur un nœud spécifié ou sur le nœud actuel si aucun nœud n'est spécifié.

```
pcs resource ban resource_id [node] [--promu] [lifetime=lifetime] [--attendu[=n]]]
```

Notez que lorsque vous exécutez la commande **pcs resource ban**, celle-ci ajoute une contrainte d'emplacement -INFINITY à la ressource afin d'empêcher son exécution sur le nœud indiqué. Vous pouvez exécuter la commande **pcs resource clear** ou **pcs constraint delete** pour supprimer cette contrainte. Cela ne ramène pas nécessairement les ressources sur le nœud indiqué ; l'endroit où les ressources peuvent s'exécuter à ce moment-là dépend de la façon dont vous avez configuré vos ressources au départ.

Si vous spécifiez le paramètre **--promoted** de la commande **pcs resource ban**, la portée de la contrainte est limitée au rôle promu et vous devez spécifier *promotable_id* au lieu de *resource_id*.

Vous pouvez éventuellement configurer un paramètre **lifetime** pour la commande **pcs resource ban** afin d'indiquer une période de temps pendant laquelle la contrainte doit être maintenue.

Vous pouvez éventuellement configurer un paramètre **--wait[=*n*]** pour la commande **pcs resource ban**

afin d'indiquer le nombre de secondes à attendre pour que la ressource démarre sur le nœud de destination avant de renvoyer 0 si la ressource est démarrée ou 1 si la ressource n'a pas encore démarré. Si vous ne spécifiez pas n, le délai d'attente par défaut de la ressource sera utilisé.

Forcer le démarrage d'une ressource sur le nœud actuel

Utilisez le paramètre **debug-start** de la commande **pcs resource** pour forcer le démarrage d'une ressource spécifiée sur le nœud actuel, en ignorant les recommandations de la grappe et en imprimant la sortie du démarrage de la ressource. Cette commande est principalement utilisée pour le débogage des ressources ; le démarrage des ressources sur un cluster est (presque) toujours effectué par Pacemaker et non directement avec la commande **pcs**. Si votre ressource ne démarre pas, cela est généralement dû à une mauvaise configuration de la ressource (que vous pouvez déboguer dans le journal du système), à des contraintes qui empêchent la ressource de démarrer, ou à la désactivation de la ressource. Vous pouvez utiliser cette commande pour tester la configuration d'une ressource, mais elle ne doit normalement pas être utilisée pour démarrer des ressources dans un cluster.

Le format de la commande **debug-start** est le suivant.

```
pcs resource debug-start resource_id
```

31.4. PASSAGE D'UNE RESSOURCE EN MODE NON GÉRÉ

Lorsqu'une ressource est en mode **unmanaged**, elle est toujours dans la configuration mais Pacemaker ne la gère pas.

La commande suivante définit les ressources indiquées en mode **unmanaged**.

```
pcs resource unmanage resource1 [resource2] ...
```

La commande suivante définit les ressources en mode **managed**, qui est l'état par défaut.

```
pcs resource manage resource1 [resource2] ...
```

Vous pouvez spécifier le nom d'un groupe de ressources avec la commande **pcs resource manage** ou **pcs resource unmanage**. La commande agit sur toutes les ressources du groupe, de sorte que vous pouvez mettre toutes les ressources d'un groupe en mode **managed** ou **unmanaged** à l'aide d'une seule commande, puis gérer individuellement les ressources qu'il contient.

31.5. MISE EN MODE MAINTENANCE D'UN CLUSTER

Lorsqu'un cluster est en mode maintenance, il ne démarre ni n'arrête aucun service jusqu'à ce qu'on lui dise le contraire. Lorsque le mode maintenance est terminé, la grappe vérifie l'état actuel de tous les services, puis arrête ou démarre ceux qui en ont besoin.

Pour mettre un cluster en mode maintenance, utilisez la commande suivante pour définir la propriété du cluster **maintenance-mode** sur **true**.

```
# pcs property set maintenance-mode=true
```

Pour retirer un cluster du mode maintenance, utilisez la commande suivante pour définir la propriété du cluster **maintenance-mode** sur **false**.

```
# pcs property set maintenance-mode=false
```

Vous pouvez supprimer une propriété de cluster de la configuration à l'aide de la commande suivante.

```
pcs property unset property
```

Vous pouvez également supprimer une propriété de cluster d'une configuration en laissant le champ value de la commande **pcs property set** vide. La valeur par défaut de cette propriété est alors rétablie. Par exemple, si vous avez précédemment défini la propriété **symmetric-cluster** sur **false**, la commande suivante supprime la valeur que vous avez définie de la configuration et restaure la valeur de **symmetric-cluster** sur **true**, qui est sa valeur par défaut.

```
# pcs property set symmetric-cluster=
```

31.6. MISE À JOUR D'UN CLUSTER RHEL À HAUTE DISPONIBILITÉ

La mise à jour des paquets qui composent les modules complémentaires RHEL High Availability et Resilient Storage, individuellement ou dans leur ensemble, peut s'effectuer de deux manières :

- *Rolling Updates*: Retirer un nœud à la fois du service, mettre à jour son logiciel, puis le réintégrer dans la grappe. Cela permet à la grappe de continuer à fournir des services et à gérer les ressources pendant que chaque nœud est mis à jour.
- *Entire Cluster Update*: Arrêter l'ensemble de la grappe, appliquer les mises à jour à tous les nœuds, puis redémarrer la grappe.

AVERTISSEMENT

Lors de l'exécution des procédures de mise à jour logicielle pour les clusters Red Hat Enterprise LInux High Availability et Resilient Storage, il est essentiel de s'assurer que tout nœud devant subir des mises à jour n'est pas un membre actif du cluster avant que ces mises à jour ne soient lancées.

Pour une description complète de chacune de ces méthodes et des procédures à suivre pour les mises à jour, voir Pratiques recommandées pour l'application de mises à jour logicielles à un cluster RHEL High Availability ou Resilient Storage.

31.7. MISE À NIVEAU DES NŒUDS DISTANTS ET DES NŒUDS INVITÉS

Si le service **pacemaker_remote** est arrêté sur un nœud distant actif ou un nœud invité, le cluster migre les ressources hors du nœud avant d'arrêter le nœud. Cela vous permet d'effectuer des mises à jour logicielles et d'autres procédures de maintenance de routine sans retirer le nœud du cluster. Cependant, une fois que **pacemaker_remote** est arrêté, le cluster tente immédiatement de se reconnecter. Si **pacemaker_remote** n'est pas redémarré dans le délai de surveillance de la ressource, le cluster considérera que l'opération de surveillance a échoué.

Si vous souhaitez éviter les pannes de moniteur lorsque le service **pacemaker_remote** est arrêté sur un nœud Pacemaker Remote actif, vous pouvez utiliser la procédure suivante pour retirer le nœud du cluster avant d'effectuer toute administration système susceptible d'arrêter **pacemaker_remote**.

Procédure

1. Arrêtez la ressource de connexion du nœud avec la commande **pcs resource disable *resourcename*** ce qui aura pour effet de déplacer tous les services hors du nœud. La ressource de connexion est la ressource **ocf:pacemaker:remote** pour un nœud distant ou, plus couramment, la ressource **ocf:heartbeat:VirtualDomain** pour un nœud invité. Pour les nœuds invités, cette commande arrête également la VM, qui doit donc être démarrée en dehors du cluster (par exemple, à l'aide de **virsh**) pour effectuer toute opération de maintenance.

   ```
   désactivation des ressources pcs resourcename
   ```

2. Effectuer les opérations de maintenance nécessaires.

3. Lorsque le nœud est prêt à être réintégré dans le cluster, réactivez la ressource à l'aide de la commande **pcs resource enable**.

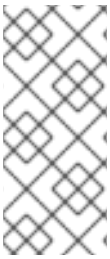
   ```
   activation des ressources pcs resourcename
   ```

31.8. MIGRATION DES MACHINES VIRTUELLES DANS UN CLUSTER RHEL

Red Hat ne prend pas en charge la migration en direct des nœuds de grappe actifs entre les hyperviseurs ou les hôtes, comme indiqué dans les Politiques de support pour les grappes de haute disponibilité RHEL - Conditions générales avec les membres de grappe virtualisés. Si vous devez effectuer une migration en direct, vous devrez d'abord arrêter les services de cluster sur la VM pour supprimer le nœud du cluster, puis redémarrer le cluster après avoir effectué la migration. Les étapes suivantes décrivent la procédure de retrait d'une VM d'un cluster, de migration de la VM et de restauration de la VM dans le cluster.

Les étapes suivantes décrivent la procédure à suivre pour supprimer une VM d'un cluster, migrer la VM et restaurer la VM dans le cluster.

Cette procédure s'applique aux machines virtuelles utilisées comme nœuds de cluster complets, et non aux machines virtuelles gérées comme ressources de cluster (y compris les machines virtuelles utilisées comme nœuds invités) qui peuvent être migrées en direct sans précautions particulières. Pour des informations générales sur la procédure complète requise pour la mise à jour des paquets qui composent les modules complémentaires RHEL High Availability et Resilient Storage, individuellement ou dans leur ensemble, voir Pratiques recommandées pour l'application de mises à jour logicielles à un cluster RHEL High Availability ou Resilient Storage.



NOTE

Avant d'exécuter cette procédure, tenez compte de l'effet de la suppression d'un nœud sur le quorum de la grappe. Par exemple, si vous avez une grappe à trois nœuds et que vous en supprimez un, votre grappe ne peut supporter qu'une seule défaillance de nœud supplémentaire. Si un nœud d'une grappe à trois nœuds est déjà hors service, la suppression d'un deuxième nœud entraînera la perte du quorum.

Procédure

1. Si des préparatifs doivent être effectués avant d'arrêter ou de déplacer les ressources ou les logiciels s'exécutant sur la VM à migrer, effectuez ces étapes.

2. Exécutez la commande suivante sur la VM pour arrêter le logiciel de cluster sur la VM.

```
# pcs cluster stop
```

3. Effectuer la migration en direct de la VM.
4. Démarrer les services de cluster sur la VM.

```
# pcs cluster start
```

31.9. IDENTIFIER LES CLUSTERS PAR UUID

Depuis Red Hat Enterprise Linux 9.1, lorsque vous créez une grappe, celle-ci est associée à un UUID. Étant donné que le nom d'un cluster n'est pas un identifiant de cluster unique, un outil tiers, tel qu'une base de données de gestion de configuration qui gère plusieurs clusters portant le même nom, peut identifier un cluster de manière unique au moyen de son UUID. Vous pouvez afficher l'UUID actuel du cluster avec la commande **pcs cluster config [show]**, qui inclut l'UUID du cluster dans sa sortie.

Pour ajouter un UUID à un cluster existant, exécutez la commande suivante.

```
# pcs cluster config uuid generate
```

Pour régénérer un UUID pour un cluster avec un UUID existant, exécutez la commande suivante.

```
# pcs cluster config uuid generate --force
```

CHAPITRE 32. CONFIGURATION DES GRAPPES DE REPRISE APRÈS SINISTRE

L'une des méthodes permettant d'assurer la reprise après sinistre d'un cluster à haute disponibilité consiste à configurer deux clusters. Vous pouvez alors configurer un cluster en tant que cluster de site primaire et le second cluster en tant que cluster de reprise après sinistre.

Dans des circonstances normales, le cluster primaire exécute des ressources en mode production. Le cluster de reprise après sinistre dispose également de toutes les ressources configurées et les exécute en mode dégradé ou pas du tout. Par exemple, une base de données peut être exécutée dans le cluster primaire en mode promu et dans le cluster de reprise après sinistre en mode rétrogradé. Dans ce cas, la base de données est configurée de manière à ce que les données soient synchronisées entre le site principal et le site de reprise après sinistre. Cette opération s'effectue par le biais de la configuration de la base de données elle-même plutôt que par l'interface de commande **pcs**.

Lorsque le cluster primaire tombe en panne, les utilisateurs peuvent utiliser l'interface de commande **pcs** pour basculer manuellement les ressources vers le site de reprise après sinistre. Ils peuvent ensuite se connecter au site de secours et promouvoir et démarrer les ressources sur ce site. Une fois le cluster primaire rétabli, les utilisateurs peuvent utiliser l'interface de commande **pcs** pour déplacer manuellement les ressources vers le site primaire.

Vous pouvez utiliser la commande **pcs** pour afficher l'état du cluster du site principal et du cluster du site de reprise après sinistre à partir d'un seul nœud sur l'un ou l'autre site.

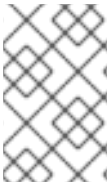
32.1. CONSIDÉRATIONS RELATIVES AUX GRAPPES DE REPRISE APRÈS SINISTRE

Lors de la planification et de la configuration d'un site de reprise après sinistre que vous gérerez et surveillerez à l'aide de l'interface de commande **pcs**, tenez compte des considérations suivantes.

- Le site de reprise après sinistre doit être un cluster. Cela permet de le configurer avec les mêmes outils et des procédures similaires à celles du site primaire.
- Les clusters primaires et de reprise après sinistre sont créés par des commandes indépendantes à l'adresse **pcs cluster setup**.
- Les clusters et leurs ressources doivent être configurés de manière à ce que les données soient synchronisées et que le basculement soit possible.
- Les nœuds de cluster du site de récupération ne peuvent pas avoir les mêmes noms que les nœuds du site primaire.
- L'utilisateur pcs **hacluster** doit être authentifié pour chaque nœud des deux clusters sur le nœud à partir duquel vous exécuterez les commandes **pcs**.

32.2. AFFICHAGE DE L'ÉTAT DES GRAPPES DE RÉCUPÉRATION

Pour configurer un cluster primaire et un cluster de reprise après sinistre afin de pouvoir afficher l'état des deux clusters, procédez comme suit.



NOTE

La configuration d'un cluster de reprise après sinistre ne permet pas de configurer automatiquement les ressources ou de répliquer les données. Ces éléments doivent être configurés manuellement par l'utilisateur.

Dans cet exemple :

- La grappe primaire sera nommée **PrimarySite** et sera composée des nœuds **z1.example.com**. et **z2.example.com**.
- Le cluster du site de reprise après sinistre sera nommé **DRsite** et comprendra les nœuds **z3.example.com** et **z4.example.com**.

Cet exemple met en place un cluster de base sans ressources ni clôtures configurées.

Procédure

1. Authentifiez tous les nœuds qui seront utilisés pour les deux clusters.

```
[root@z1 ~]# pcs host auth z1.example.com z2.example.com z3.example.com
z4.example.com -u hacluster -p password
z1.example.com: Authorized
z2.example.com: Authorized
z3.example.com: Authorized
z4.example.com: Authorized
```

2. Créez le cluster qui sera utilisé comme cluster primaire et démarrez les services de cluster pour le cluster.

```
[root@z1 ~]# pcs cluster setup PrimarySite z1.example.com z2.example.com --start
{...}
Cluster has been successfully set up.
Starting cluster on hosts: 'z1.example.com', 'z2.example.com'...
```

3. Créez le cluster qui sera utilisé comme cluster de reprise après sinistre et démarrez les services de cluster pour le cluster.

```
[root@z1 ~]# pcs cluster setup DRSite z3.example.com z4.example.com --start
{...}
Cluster has been successfully set up.
Starting cluster on hosts: 'z3.example.com', 'z4.example.com'...
```

4. À partir d'un nœud du cluster primaire, configurez le second cluster en tant que site de récupération. Le site de reprise est défini par le nom de l'un de ses nœuds.

```
[root@z1 ~]# pcs dr set-recovery-site z3.example.com
Sending 'disaster-recovery config' to 'z3.example.com', 'z4.example.com'
z3.example.com: successful distribution of the file 'disaster-recovery config'
z4.example.com: successful distribution of the file 'disaster-recovery config'
Sending 'disaster-recovery config' to 'z1.example.com', 'z2.example.com'
z1.example.com: successful distribution of the file 'disaster-recovery config'
z2.example.com: successful distribution of the file 'disaster-recovery config'
```

5. Vérifier la configuration de la reprise après sinistre.

```
[root@z1 ~]# pcs dr config
Local site:
  Role: Primary
Remote site:
  Role: Recovery
  Nodes:
    z1.example.com
    z2.example.com
```

6. Vérifiez l'état du cluster primaire et du cluster de reprise après sinistre à partir d'un nœud du cluster primaire.

```
[root@z1 ~]# pcs dr status
--- Local cluster - Primary site ---
Cluster name: PrimarySite

WARNINGS:
No stonith devices and stonith-enabled is not false

Cluster Summary:
  * Stack: corosync
  * Current DC: z2.example.com (version 2.0.3-2.el8-2c9cea563e) - partition with quorum
  * Last updated: Mon Dec  9 04:10:31 2019
  * Last change:  Mon Dec  9 04:06:10 2019 by hacluster via crmd on z2.example.com
  * 2 nodes configured
  * 0 resource instances configured

Node List:
  * Online: [ z1.example.com z2.example.com ]

Full List of Resources:
  * No resources

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled


--- Remote cluster - Recovery site ---
Cluster name: DRSite

WARNINGS:
No stonith devices and stonith-enabled is not false

Cluster Summary:
  * Stack: corosync
  * Current DC: z4.example.com (version 2.0.3-2.el8-2c9cea563e) - partition with quorum
  * Last updated: Mon Dec  9 04:10:34 2019
  * Last change:  Mon Dec  9 04:09:55 2019 by hacluster via crmd on z4.example.com
  * 2 nodes configured
  * 0 resource instances configured

Node List:
```

```
  * Online: [ z3.example.com z4.example.com ]

Full List of Resources:
  * No resources

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

Pour obtenir des options d'affichage supplémentaires pour une configuration de reprise après sinistre, consultez l'écran d'aide de la commande **pcs dr**.

CHAPITRE 33. INTERPRÉTATION DES CODES DE RETOUR OCF DES AGENTS DE RESSOURCES

Les agents de ressources Pacemaker sont conformes à l'API de l'agent de ressources de l'Open Cluster Framework (OCF). Les tableaux suivants décrivent les codes de retour OCF et leur interprétation par Pacemaker.

La première chose que fait le cluster lorsqu'un agent renvoie un code est de vérifier le code de retour par rapport au résultat attendu. Si le résultat ne correspond pas à la valeur attendue, l'opération est considérée comme ayant échoué et une action de récupération est lancée.

Pour toute invocation, les agents de ressources doivent sortir avec un code de retour défini qui informe l'appelant du résultat de l'action invoquée.

Il existe trois types de reprise sur panne, décrits dans le tableau suivant.

Tableau 33.1. Types de récupération effectués par le cluster

Type	Description	Mesures prises par le groupe
doux	Une erreur transitoire s'est produite.	Redémarrer la ressource ou la déplacer vers un nouvel emplacement .
dur	Une erreur non transitoire qui peut être spécifique au nœud actuel s'est produite.	Déplacer la ressource ailleurs et empêcher qu'elle soit réessayée sur le nœud actuel.
mortel	Une erreur non transitoire commune à tous les nœuds de la grappe s'est produite (par exemple, une mauvaise configuration a été spécifiée).	Arrêter la ressource et empêcher qu'elle soit démarrée sur n'importe quel nœud du cluster.

Le tableau suivant présente les codes de retour OCF et le type de récupération que le cluster entreprend lorsqu'un code d'échec est reçu. Notez que même les actions qui renvoient 0 (alias OCF **OCF_SUCCESS**) peuvent être considérées comme ayant échoué si 0 n'était pas la valeur de retour attendue.

Tableau 33.2. Codes de retour de l'OCF

Code de retour	Label OCF	Description
0	**OCF_SUCCESS**	* L'action s'est terminée avec succès. Il s'agit du code de retour attendu pour toute commande de démarrage, d'arrêt, de promotion et de rétrogradation réussie. * Type si inattendu : doux

Code de retour	Label OCF	Description
1	**OCF_ERR_GENERIC**	* L'action a renvoyé une erreur générique. * Type : doux * Le gestionnaire de ressources tentera de récupérer la ressource ou de la déplacer vers un nouvel emplacement.
2	**OCF_ERR_ARGS**	* La configuration de la ressource n'est pas valide sur cette machine. Par exemple, elle fait référence à un emplacement introuvable sur le nœud. * Type : dur * Le gestionnaire de ressources déplacera la ressource ailleurs et l'empêchera d'être réessayée sur le nœud actuel
3	**OCF_ERR_UNIMPLEMENTED**	* L'action demandée n'est pas mise en œuvre. * Type : dur
4	**OCF_ERR_PERM**	* L'agent de ressource n'a pas les privilèges suffisants pour accomplir la tâche. Cela peut être dû, par exemple, au fait que l'agent n'est pas en mesure d'ouvrir un certain fichier, d'écouter sur un socket spécifique ou d'écrire dans un répertoire. * Type : dur * Sauf configuration spécifique contraire, le gestionnaire de ressources tentera de récupérer une ressource qui a échoué avec cette erreur en redémarrant la ressource sur un nœud différent (où le problème de permission peut ne pas exister).
5	**OCF_ERR_INSTALLED**	* Un composant requis est manquant sur le nœud où l'action a été exécutée. Cela peut être dû au fait qu'un binaire requis n'est pas exécutable ou qu'un fichier de configuration essentiel est illisible. * Type : dur * Sauf configuration spécifique contraire, le gestionnaire de ressources tentera de récupérer une ressource qui a échoué avec cette erreur en redémarrant la ressource sur un nœud différent (où les fichiers ou les binaires requis peuvent être présents).

Code de retour	Label OCF	Description
6	**OCF_ERR_CONFIGURED**	* La configuration de la ressource sur le nœud local n'est pas valide. * Type : fatal * Lorsque ce code est renvoyé, Pacemaker empêchera l'exécution de la ressource sur tout nœud du cluster, même si la configuration du service est valide sur un autre nœud.
7	**OCF_NOT_RUNNING**	* La ressource est arrêtée en toute sécurité. Cela signifie que la ressource s'est arrêtée de manière élégante ou qu'elle n'a jamais été démarrée. * Type si inattendu : doux * Le cluster n'essaiera pas d'arrêter une ressource qui renvoie ce message pour quelque action que ce soit.
8	**OCF_RUNNING_PROMOTED**	* La ressource est exécutée dans un rôle promu. * Type si inattendu : doux
9	**OCF_FAILED_PROMOTED**	* La ressource est (ou pourrait être) dans un rôle promu mais a échoué. * Type : doux * La ressource sera rétrogradée, arrêtée, puis reprise (et éventuellement promue).
190		* Il s'avère que le service est correctement actif, mais dans un état tel que de futures défaillances sont plus probables.
191		* L'agent de ressources prend en charge les rôles et le service est considéré comme correctement actif dans le rôle promu, mais dans un état tel que les défaillances futures sont plus probables.
autres	N/A	Code d'erreur personnalisé.

CHAPITRE 34. CONFIGURATION D'UN CLUSTER RED HAT HIGH AVAILABILITY AVEC DES INSTANCES IBM Z/VM EN TANT QUE MEMBRES DU CLUSTER

Red Hat propose plusieurs articles qui peuvent être utiles lors de la conception, de la configuration et de l'administration d'un cluster Red Hat High Availability fonctionnant sur des machines virtuelles z/VM.

- Conseils de conception pour les grappes de haute disponibilité RHEL - Instances IBM z/VM en tant que membres de la grappe
- Procédures administratives pour les clusters haute disponibilité RHEL - Configuration de la clôture SMAPI z/VM avec fence_zvmip pour les membres de clusters IBM z Systems RHEL 7 ou 8
- Les nœuds de cluster RHEL High Availability sur IBM z Systems subissent des dépassements de délai de l'appareil STONITH vers minuit tous les soirs
- Administrative Procedures for RHEL High Availability Clusters - Préparation d'un périphérique de stockage dasd en vue de son utilisation par une grappe de membres IBM z Systems

Les articles suivants peuvent également vous être utiles lors de la conception d'un cluster Red Hat High Availability en général.

- Politiques de support pour les clusters de haute disponibilité RHEL
- Explorer les concepts des clusters de haute disponibilité RHEL - Fencing/STONITH