



Red Hat Enterprise Linux

7

Administration du module complémentaire High Availability

Configurer et gérer le module complémentaire High Availability

Configurer et gérer le module complémentaire High Availability

Notice légale

Copyright © 2015 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

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

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

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

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

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

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

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

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

All other trademarks are the property of their respective owners.

Résumé

Configurer et gérer le module complémentaire High Availability décrit la configuration et la gestion du module complémentaire High Availability sur Red Hat Enterprise Linux 7.

Table des matières

Chapitre 1. Créer un cluster Red Hat High-Availability avec Pacemaker	2
1.1. Installation du logiciel de cluster	2
1.2. Création de cluster	3
1.3. Configuration d'une clôture	4
Chapitre 2. Un serveur NFS actif/passif dans un cluster Red Hat HA	6
2.1. Configurer un volume LVM dans un système de fichiers ext4	7
2.2. Configuration du serveur web	8
2.3. Activation exclusive d'un Groupe de volumes dans un cluster	8
2.4. Créer les ressources et les groupes de ressources par la commande pcs	10
2.5. Tester la configuration de ressource	12
Chapitre 3. Un serveur NFS actif/passif dans Red Hat High Availability Cluster	14
3.1. Créer le cluster NFS	14
3.2. Configurer un volume LVM dans un système de fichiers ext4	15
3.3. Mise en place de Partages NFS	16
3.4. Activation exclusive d'un Groupe de volumes dans un cluster	16
3.5. Configurer les ressources du cluster	18
3.6. Tester la configuration de ressource	21
Annexe A. Historique des versions	24

Chapitre 1. Créer un cluster Red Hat High-Availability avec Pacemaker

Ce chapitre décrit la procédure pour créer un cluster Red Hat High Availability en deux-nœuds par la commande **pcs**. Une fois que vous aurez créé un cluster, vous pourrez configurer les ressources et les groupes de ressources dont vous avez besoin.

La configuration du cluster qui vous est proposée dans ce chapitre exige que votre système inclue 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 pour la communication entre les nœuds du cluster et le reste du matériel du cluster, comme les commutateurs d'alimentation réseau et les interrupteurs Fibre Channel.
- Un périphérique de power fencing pour chaque nœud du cluster. Cet exemple utilise deux ports de commutateur d'alimentation APC avec un nom d'hôte **zapc.example.com**.

Ce chapitre se divise en trois sections.

- [Section 1.1, « Installation du logiciel de cluster »](#) fournit la procédure d'installation du logiciel du cluster.
- [Section 1.2, « Création de cluster »](#) fournit la procédure de configuration du cluster en deux-nœuds.
- [Section 1.3, « Configuration d'une clôture »](#) fournit la procédure de configuration des périphériques fence pour chaque nœud du cluster.

1.1. Installation du logiciel de cluster

La procédure d'installation et de configuration du cluster est la suivante.

1. Sur chaque nœud du cluster, installer les packages de logiciels du module Red Hat High Availability avec tous les agents de clôture du réseau HA.

```
# yum install pcs fence-agents-all
```

2. Si vous exécutez le démon **firewalld**, exécutez les commandes suivantes pour activer les ports requis par le module Red Hat High Availability (HA)



Note

Vous pourrez déterminer si le démon **firewalld** est installé sur votre système par la commande **rpm -q firewalld**. Si le démon **firewalld** est installé, vous pourrez vérifier s'il exécute grâce à la commande **firewall-cmd --state**.

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

3. Pour pouvoir utiliser **pcs** afin de configurer et communiquer entre les noeuds, vous devez définir un mot de passe sur chaque noeud pour l'ID utilisateur **hacluster**, qui corresponde au compte d'administration **pcs**. Il est conseillé d'utiliser le même mot de passe sur chaque noeud pour l'utilisateur **hacluster**.

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

4. Avant de configurer le cluster, le démon **pcsd** doit être démarré et activé pour démarrer au départ de chaque noeud. Ce démon fonctionne avec la commande **pcs** qui gère la configuration à travers tous les noeuds du cluster.

Exécuter la commande suivante sur chaque noeud du cluster pour démarrer le service **pcsd** et pour activer **pcsd** au démarrage du système.

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

5. Authentifier l'utilisateur **pcs hacluster** pour chaque noeud du cluster sur le noeud à partir duquel vous allez exécuter la commande **pcs**.

La commande suivante authentifie l'utilisateur **hacluster** sur **z1.example.com** pour les deux noeuds dans l'exemple de cluster en deux-noeuds, **z1.example.com** et **z2.example.com**.

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

1.2. Création de cluster

Cette procédure crée un module de cluster Red Hat HA composé de deux nœuds **z1.example.com** et **z2.example.com**.

1. Exécutez la commande suivante à partir de **z1.example.com** pour créer un cluster en deux-nœuds **my_cluster** composé des nœuds **z1.example.com** et **z2.example.com**. Cela aura pour effet de propager les fichiers de configuration du cluster aux deux nœuds du cluster. Cette commande inclut l'option **--start**, qui va démarrer les services du cluster sur les deux nœuds du cluster.

```
[root@z1 ~]# pcs cluster setup --start --name my_cluster \
z1.example.com z2.example.com
z1.example.com: Succeeded
z1.example.com: Starting Cluster...
z2.example.com: Succeeded
z2.example.com: Starting Cluster...
```

2. Activer les services du cluster pour qu'ils exécutent sur chaque noeud du cluster quand le noeud démarre.

**Note**

Pour votre environnement précis, vous pouvez décider de ne pas activer les services du cluster en ignorant cette étape. Cela vous permettra de vous assurer que, si un noeud venait à échouer, tous les problèmes avec votre cluster ou vos ressources seront résolus avant que le noeud ne rejoigne le cluster. Si vous désactivez les services du cluster, vous devrez démarrer les services manuellement au redémarrage d'un noeud, en exécutant la commande **pcs cluster start** sur ce noeud.

```
# pcs cluster enable --all
```

Vous pourrez afficher le statut actuel du cluster par la commande **pcs cluster status**. Comme il peut y avoir un petit délai entre le moment où le cluster démarre et le démarrage des services du cluster avec l'option **--start** de la commande **pcs cluster setup**, vous devez vous assurer que le cluster exécute avant toute action à venir sur le cluster et sa configuration.

```
[root@z1 ~]# pcs cluster status
Cluster Status:
  Last updated: Thu Jul 25 13:01:26 2013
  Last change: Thu Jul 25 13:04:45 2013 via crmd on z2.example.com
  Stack: corosync
  Current DC: z2.example.com (2) - partition with quorum
  Version: 1.1.10-5.el7-9abe687
  2 Nodes configured
  0 Resources configured
```

1.3. Configuration d'une clôture

Vous devez configurer un périphérique de clôture pour chaque noeud du cluster. Pour obtenir des informations générales sur la configuration des périphériques de clôture, voir le guide *Red Hat Enterprise Linux 7 High Availability Add-On Reference*.

**Note**

Quand vous configurez un périphérique de clôture, vous devez veiller à ce que votre périphérique de clôture ne partage pas son alimentation avec le noeud qu'il contrôle.

Cette exemple utilise le commutateur d'alimentation APC avec le nom d'hôte **zapc.example.com** pour clôturer les nœuds, et il utilise l'agent de clôture **fence_apc_snmp**. Comme les deux nœuds seront clôturés par le même agent de clôture, vous pourrez configurer à la fois les deux périphériques de clôture en une seule ressource, en utilisant les options **pcmk_host_map** et **pcmk_host_list**.

Vous créez un périphérique de clôture en configurant le périphérique en tant que ressource **stonith** par 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** fait correspondre **z1.example.com** au port 1, et **z2.example.com** au port 2. La valeur pour la connexion et le mot de passe du périphérique APC sont tous les deux **apc**. Par défaut, ce périphérique utilisera un intervalle de contrôle de 60s par noeud.

Notez que vous pouvez utiliser une adresse IP quand vous indiquez le nom d'utilisateur hôte des noeuds.


```
[root@z1 ~]# pcs stonith create myapc fence_apc_snmp params \
ipaddr="zapc.example.com" pcmk_host_map="z1.example.com:1;z2.example.com:2"
\
pcmk_host_check="static-list" pcmk_host_list="z1.example.com,z2.example.com"
\
login="apc" passwd="apc"
```



Note

Quand vous créez un périphérique **fence_apc_snmp stonith**, vous apercevrez sans doute le message suivant que vous pouvez ignorer en toute sécurité :

```
Warning: missing required option(s): 'port, action' for resource type:
stonith:fence_apc_snmp
```

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

```
[root@rh7-1 ~]# pcs stonith show 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 pcmk_host_check=static-list
pcmk_host_list=z1.example.com,z2.example.com login=apc passwd=apc
Operations: monitor interval=60s (myapc-monitor-interval-60s)
```

Chapitre 2. Un serveur NFS actif/passif dans un cluster Red Hat HA

Ce chapitre décrit comment configurer un serveur web Apache actif/passif dans un cluster en deux-nœuds de module Red Hat Enterprise Linux HA par la commande **pcs** pour configurer les ressources de cluster. Dans ce cas d'utilisation, les clients accèdent au serveur web Apache via une adresse IP flottante. Le serveur web exécute sur un des deux nœuds du cluster. Si le nœud sur lequel le serveur web exécute n'opère plus, le serveur web fait une seconde tentative sur le second nœud du cluster avec une interruption de service minimale.

Figure 2.1, « Un serveur Web Apache actif/passif dans un cluster Red Hat HA à deux-nœuds » vous donne des informations détaillées sur le cluster. Le cluster est un cluster Red Hat High Availability à deux-nœuds qui est configuré avec un commutateur de réseau et un stockage partagé. Les nœuds du cluster sont connectés à un réseau public pour que le client puisse accéder à un serveur Web Apache via une adresse IP virtuelle. Le serveur Apache exécute sur le Nœud 1 ou le Nœud 2, ayant chacun accès au stockage où se trouvent les données d'Apache.

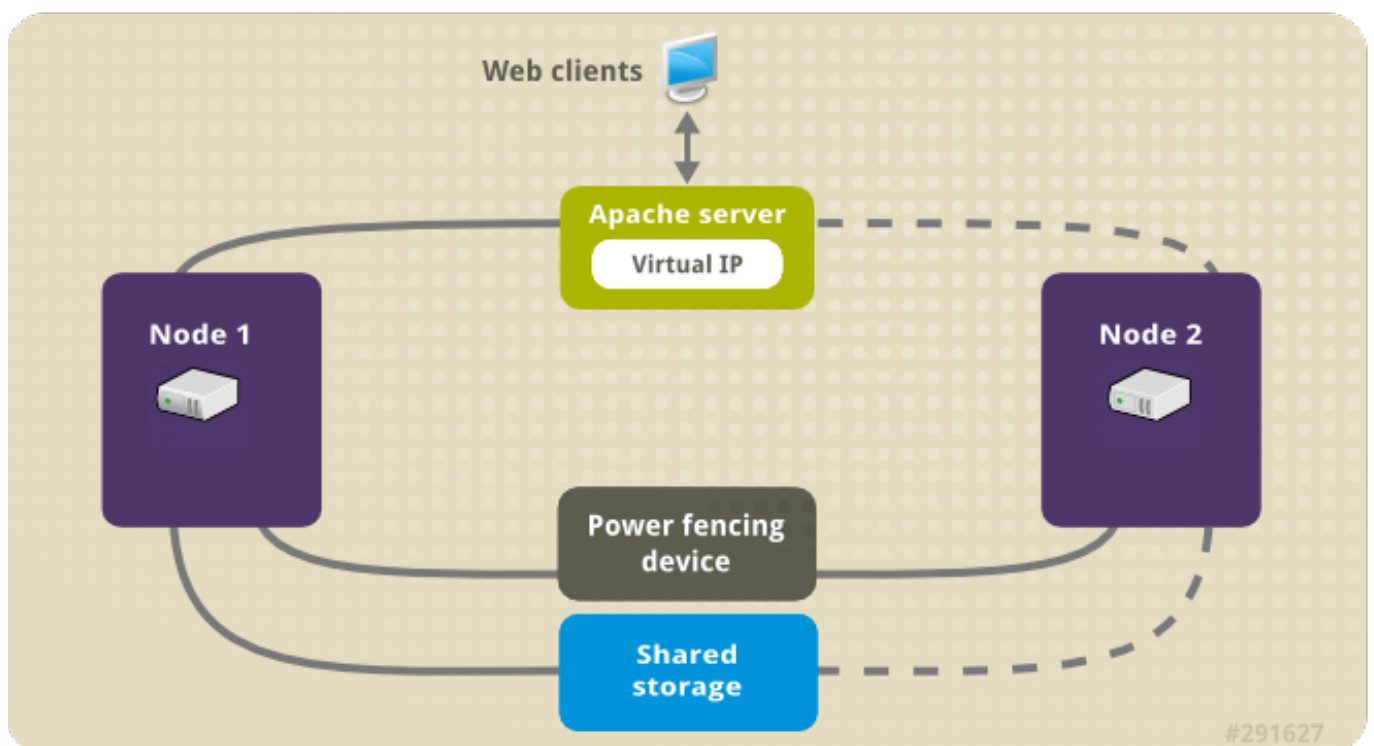


Figure 2.1. Un serveur Web Apache actif/passif dans un cluster Red Hat HA à deux-nœuds

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

- ✧ Un cluster Red Hat HA à deux-nœuds avec power fencing configuré pour chaque nœud. Cette procédure utilise l'exemple de cluster fourni dans [Chapitre 1, Créer un cluster Red Hat High-Availability avec Pacemaker](#).
- ✧ Une adresse IP virtuelle publique requise pour le serveur NFS.
- ✧ Stockage partagé pour les nœuds du cluster, utilisant iSCSI ou Fibre Channel.

Le cluster est configuré avec un groupe de ressources Apache, qui contient les composants de cluster dont le serveur web a besoin : une ressource LVM, une ressource système de fichiers, une ressource d'adresse IP, et une ressource de serveur web. Ce groupe de ressources peut échouer sur un nœud et passer à un autre nœud du cluster, permettant à chaque nœud d'exécuter sur le serveur web. Avant de créer un groupe de ressources pour ce cluster, procédez aux étapes suivantes :

1. Configurer un système de fichiers **ext4** monté sur le volume logique LVM **my_lv**, comme décrit dans [Section 3.2, « Configurer un volume LVM dans un système de fichiers ext4 »](#).
2. Configurer un serveur web, comme décrit dans [Section 2.2, « Configuration du serveur web »](#).
3. Veillez bien à ce que le cluster uniquement soit en mesure d'activer le groupe de volumes qui contient **my_lv**, et que le groupe de volumes ne soit pas activé en dehors du cluster au démarrage, comme décrit dans [Section 2.3, « Activation exclusive d'un Groupe de volumes dans un cluster »](#).

Une fois que vous aurez procédé, vous pourrez créer un groupe de ressources avec les ressources qu'il contient, comme décrit dans [Section 2.4, « Créer les ressources et les groupes de ressources par la commande pcs »](#).

2.1. Configurer un volume LVM dans un système de fichiers ext4

Ce cas d'utilisateur exige de créer un volume logique LVM dans un stockage partagé entre les nœuds du cluster.

La procédure suivante crée un volume logique LVM, puis crée un système de fichiers **ext4** sur ce volume. 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.



Note

Les volumes logiques, leurs partitions correspondantes et les périphériques utilisés par les nœuds du cluster doivent être connectés aux nœuds du cluster uniquement.

Comme la partition **/dev/sdb1** correspond à du stockage partagé, vous pourrez effectuer cette procédure sur un nœud uniquement.

1. Créez un volume physique LVM sur la partition **/dev/sdb1**.

```
# pvcreate /dev/sdb1
Physical volume "/dev/sdb1" successfully created
```

2. Créez le groupe de volume **my_vg** comprenant le volume physique **/dev/sdb1**.

```
# vgcreate my_vg /dev/sdb1
Volume group "my_vg" successfully created
```

3. Créer un volume logique en utilisant le groupe de volumes **my_vg**.

```
# 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.

```
# lvs
LV VG Attr LSize Pool Origin Data% Move Log Copy% Convert
my_lv my_vg -wi-a---- 452.00m
...
```

4. Créez un système de fichiers **ext4** sur le volume logique **my_lv**.

```
# mkfs.ext4 /dev/my_vg/my_lv
mke2fs 1.42.7 (21-Jan-2013)
Filesystem label=
OS type: Linux
...
```

2.2. Configuration du serveur web

La procédure suivante configure un serveur web Apache.

1. Veillez à ce que le serveur Apache HTTPD soit bien installé sur chaque noeud du cluster. Vous aurez également besoin de l'outil **wget** installé sur le cluster pour pouvoir vérifier le statut du serveur web Apache.

Sur chaque noeud, exécutez la commande suivante.

```
# yum install -y httpd wget
```

2. Pour que l'agent de ressource Apache obtienne le statut du serveur web Apache, veillez à ce que le texte suivant soit présent dans le fichier **/etc/httpd/conf/httpd.conf** sur chaque noeud du cluster, et qu'il ait été dé-commenté. Si le texte n'est pas déjà présent, ajouter le texte en fin du fichier.

```
<Location /server-status>
  SetHandler server-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

3. Créez une page web pour le serveur Apache. Sur un noeud du cluster, montez le système de fichiers que vous avez créé dans [Section 2.1, « Configurer un volume LVM dans un système de fichiers ext4 »](#), créez le fichier **index.html** sur ce système de fichiers, et dé-montez le système de fichiers.

```
# 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
```

2.3. Activation exclusive d'un Groupe de volumes dans un cluster

La procédure suivante configure le groupe de volumes afin que seul le cluster puisse activer le groupe de volumes, et que le groupe de volumes ne puisse pas être activé en dehors du cluster au démarrage. Si le groupe de volumes est activé par un système en dehors du cluster, il y a un risque de corruption des métadonnées du groupe de volumes.

Cette procédure modifie l'entrée de **volume_list** dans le fichier de configuration **/etc/lvm/lvm.conf**. Les groupes de volume listés dans **volume_list** peuvent être automatiquement activés sur le noeud local en dehors du contrôle du gestionnaire de cluster. Les groupes de volumes associés aux répertoires d'accueil et root local doivent être inclus dans cette liste. Tous les groupes de volumes gérés par le gestionnaire de cluster doivent être exclus de l'entrée **volume_list**. Notez bien que cette procédure ne nécessite pas la commande **clvmd**.

Effectuer la même procédure pour chaque noeud du cluster.

1. Exécutez la commande suivante pour que **locking_type** soit sur 1 et **use_lvmetad** sur 0 dans le fichier **/etc/lvm/lvm.conf**. Cette commande désactive et stoppe également les processus **lvmetad** immédiatement.

```
# lvmconf --enable-halvm --services --startstopservices
```

2. Déterminez quels groupes de volumes sont actuellement configurés sur votre espace de stockage local par la commande suivante. Cela affichera une liste des groupes de volume actuellement configurés. Si vous avez de l'espace alloué dans les groupes de volumes distincts pour root et votre répertoire d'accueil sur ce nœud, vous verrez ces volumes dans la sortie, comme dans cet exemple.

```
# vgs --noheadings -o vg_name
my_vg
rhel_home
rhel_root
```

3. Ajoutez les groupes de volumes autres que **my_vg** (le groupe de volumes que vous venez de définir pour le cluster) comme entrées dans **volume_list** du fichier de configuration **/etc/lvm/lvm.conf**. Ainsi, si vous avez de l'espace alloué dans des groupes de volumes distincts pour root et pour votre répertoire d'accueil, vous pourrez dé-commenter la ligne **volume_list** du fichier **lvm.conf** et ajouter ces groupes de volumes en tant qu'entrées dans **volume_list** comme suit :

```
volume_list = [ "rhel_root", "rhel_home" ]
```



Note

Si aucun groupe de volume local n'est présent sur un noeud restant à activer en dehors du gestionnaire du cluster, vous devrez toujours initialiser l'entrée **volume_list** en tant que **volume_list = []**.

4. Reconstruire l'image de démarrage **initramfs** pour garantir que l'image de démarrage n'essaie pas d'activer un groupe de volumes contrôlée par le cluster, Mettez à jour le périphérique **initramfs** par la commande suivante. Cette commande met une minute à se compléter.

```
# dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

5. Redémarrez le nœud.

**Note**

Si vous avez installé un nouveau noyau Linux après avoir démarré le nœud sur lequel vous avez créé l'image de démarrage, la nouvelle image **initrd** sera pour le noyau qui était en cours lors de sa création et non pas pour le nouveau noyau qui exécute lorsque vous redémarrez le nœud. Vous pouvez vous assurer que le périphérique **initrd** qui convient est utilisée en exécutant la commande **uname - r** avant et après le redémarrage pour déterminer la version de noyau qui est en cours d'exécution. Si les versions ne sont pas les mêmes, mettez à jour le fichier **initrd** après le redémarrage avec le nouveau noyau, puis, redémarrez le nœud.

6. Lorsque le nœud a redémarré, vérifiez si les services de cluster ont démarré à nouveau sur ce nœud en exécutant la commande **pcs statut de cluster** sur ce nœud. Si vous obtenez le message **Error: cluster is not currently running on this node**, alors exécutez la commande suivante.

```
# pcs cluster start
```

Sinon, attendez d'avoir redémarré chaque nœud du cluster et les services de cluster sur tous les nœuds du cluster par la commande suivante.

```
# pcs cluster start --all
```

2.4. Créer les ressources et les groupes de ressources par la commande **pcs**

Vous devrez créer quatre ressources de cluster pour ce cas de ressource. Pour que ces ressources exécutent sur le même nœud, elles sont configurées comme faisant partie du groupe de ressources **apachegroup**. Les ressources à créer sont les suivantes, dans l'ordre par lesquelles elles commenceront.

1. Une ressource **LVM** nommée **my_lvm** utilisant le groupe de volumes LVM que vous avez créé dans [Section 2.1, « Configurer un volume LVM dans un système de fichiers ext4 »](#).
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 [Section 2.1, « Configurer un volume LVM dans un système de fichiers ext4 »](#).
3. Une ressource **IPaddr2**, qui est une adresse IP flottante pour le groupe de ressources **apachegroup**. L'adresse IP doit être déjà associée au nœud physique. Si le périphérique NIC de la ressource **IPaddr2** n'est pas spécifié, l'IP flottante doit se trouver sur le même réseau que les adresses IP utilisées par les nœuds du cluster et assignées statiquement, sinon le périphérique NIC qui doit assigner l'adresse IP ne pourra pas être détecté correctement.
4. Une ressource **apache** nommée **Website** qui utilise le fichier **index.html** et la configuration Apache que vous avez définie dans [Section 2.2, « Configuration du serveur web »](#).

La procédure suivante crée le groupe de ressources **apachegroup** et les ressources que ce groupe contient. Les ressources vont démarrer dans l'ordre dans lequel vous les avez ajoutées au groupe, et elles s'arrêteront dans l'ordre inverse dans lequel elles ont été ajoutées au groupe. Exécutez cette procédure à partir d'un nœud du cluster uniquement.

1. La commande suivante crée la ressource LVM **my_lvm**. Cette commande spécifie le paramètre **exclusive=true** pour s'assurer que le cluster uniquement soit capable d'activer le volume logique LVM. Comme le groupe de ressources **apachegroup** n'existe pas, cette commande crée un groupe de ressources.

```
[root@z1 ~]# pcs resource create my_lvm LVM volgrpname=my_vg \
exclusive=true --group apachegroup
```

Quand vous créez une ressource, la ressource démarre automatiquement. Vous pouvez utiliser la commande suivante pour confirmer que la ressource a été créée et a démarré.

```
# pcs resource show
Resource Group: apachegroup
my_lvm\t(ocf::heartbeat:LVM):\tStarted
```

Vous pouvez démarrer et arrêter une ressource individuelle manuellement par les commandes **pcs resource disable** et **pcs resource enable**.

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

```
[root@z1 ~]# pcs resource create my_fs Filesystem \
device="/dev/my_vg/my_lv" directory="/var/www" fstype="ext4" --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 pourrez vérifier le statut du cluster. Notez que les quatre ressources exécutent sur le même noeud.

```
[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\t(stonith:fence_apc_snmp):\tStarted z1.example.com
Resource Group: apachegroup
my_lvm\t(ocf::heartbeat:LVM):\tStarted z1.example.com
my_fs\t(ocf::heartbeat:Filesystem):\tStarted z1.example.com
VirtualIP\t(ocf::heartbeat:IPAddr2):\tStarted z1.example.com
```

```
Website\t(ocf::heartbeat:apache):\tStarted z1.example.com
```

Notez que si vous n'avez pas configuré de périphérique fence pour votre cluster, comme décrit dans [Section 1.3, « Configuration d'une clôture »](#), par défaut, les ressources ne démarreront pas.

- Une fois que le cluster est en place, vous pourrez faire pointer un navigateur sur l'adresse IP que vous avez définie quand la ressource d'échantillonnage **IPaddr2** s'affiche, avec le mot simple "Hello".

```
Hello
```

Si vous vous rendez compte que les ressources que vous configurez n'exécutent pas, vous pouvez exécuter la commande **pcs resource debug-start resource** pour tester la ressource de configuration. Pour obtenir des informations sur la commande **pcs resource debug-start**, consulter le manuel *High Availability Add-On Reference*.

2.5. Tester la configuration de ressource

Dans le statut de cluster qui apparaît dans [Section 2.4, « Créer les ressources et les groupes de ressources par la commande pcs »](#), toutes les ressources exécutent sur le noeud **z1.example.com**. Vous pouvez tester pour savoir si le groupe de ressources échoue sur le noeud **z2.example.com** en utilisant la procédure suivante pour mettre le premier noeud en mode **standby**, après quoi le noeud ne sera plus en mesure d'héberger des ressources.

- La commande suivante met le noeud **z1.example.com** en mode **standby**.

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

- Une fois que vous aurez mis le noeud **z1** en mode d'attente, vérifier le statut du cluster. Notez que les ressources n'exécutent pas toutes 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\t(stonith:fence_apc_snmp):\tStarted z1.example.com
Resource Group: apachegroup
my_lvm\t(ocf::heartbeat:LVM):\tStarted z2.example.com
my_fs\t(ocf::heartbeat:Filesystem):\tStarted z2.example.com
VirtualIP\t(ocf::heartbeat:IPaddr2):\tStarted z2.example.com
Website\t(ocf::heartbeat:apache):\tStarted z2.example.com
```


Le site web de l'adresse IP définie doit toujours s'afficher, sans interruption.

3. Pour supprimer **z1** du mode **en attente**, exécutez la commande suivante.

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



Note

Supprimez un noeud du mode **en attente** n'est pas ce qui amène les ressources à se retrouver sur ce noeud. Pour obtenir des informations sur la façon de contrôler sur quel noeud les ressources doivent exécuter, voir le chapitre sur la façon de configurer les ressources de cluster dans *Red Hat High Availability Add-On Reference*.

Chapitre 3. Un serveur NFS actif/passif dans Red Hat High Availability Cluster

Ce chapitre décrit comment configurer un cluster de module complémentaire de Red Hat Enterprise Linux High Availability en utilisant le stockage partagé. La procédure utilise **pcs** pour configurer les ressources de cluster de Pacemaker. Dans ce cas d'utilisation, les clients accèdent au système de fichiers NFS par une adresse IP flottante. Le serveur NFS exécute sur un des deux nœuds du cluster. Si le nœud sur lequel le serveur NFS exécute n'opère plus, le serveur NFS démarrera à nouveau sur le second nœud du cluster, occasionnant ainsi une interruption de service minimale.

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

- ✦ Deux nœuds qui seront utilisés pour créer le cluster exécutant le serveur web Apache. Dans cet exemple, les nœuds utilisés sont **z1.example.com** et **z2.example.com**.
- ✦ Un périphérique de power fencing pour chaque nœud du cluster webfarm. Cet exemple utilise deux ports de commutateur d'alimentation APC avec un nom d'hôte **zapc.example.com**.
- ✦ Une adresse IP virtuelle publique requise pour le serveur NFS.
- ✦ Stockage partagé pour les nœuds du cluster, utilisant iSCSI ou Fibre Channel.

Configurer un serveur actif/passif NFS HA sur Red Hat Enterprise Linux High requiert les deux étapes suivantes :

1. Créer le cluster qui exécutera le serveur NFS et configurer le clôturage pour chaque nœud du cluster, comme décrit dans [Section 3.1, « Créer le cluster NFS »](#).
2. Configurer un système de fichiers **ext4** monté sur le volume logique LVM **my_lv** dans le stockage partagé des nœuds du cluster, comme décrit dans [Section 3.2, « Configurer un volume LVM dans un système de fichiers ext4 »](#).
3. Configurer un partage NFS dans le stockage partagé sur le volume logique, comme décrit dans [Section 3.3, « Mise en place de Partages NFS »](#).
4. Veillez bien à ce que le cluster uniquement soit en mesure d'activer le groupe de volumes LVM qui contient le volume logique **my_lv**, et que le groupe de volumes ne soit pas activé en dehors du cluster au démarrage, comme décrit dans [Section 3.4, « Activation exclusive d'un Groupe de volumes dans un cluster »](#).
5. Créer les ressources du cluster, comme décrit dans [Section 3.5, « Configurer les ressources du cluster »](#).
6. Tester le serveur NFS que vous avez configuré, comme décrit dans [Section 3.6, « Tester la configuration de ressource »](#).

3.1. Créer le cluster NFS

Utiliser la procédure suivante pour installer et créer le cluster NFS.

1. Installer le cluster software sur les nœuds **z1.example.com** et **z2.example.com**, en suivant les étapes décrites dans [Section 1.1, « Installation du logiciel de cluster »](#).
2. Créer le cluster webfarm en deux nœuds, composé de **z1.example.com** et **z2.example.com**, en suivant la procédure donnée dans [Section 1.2, « Création de cluster »](#). Comme dans cet exemple de procédure, on utilise les noms du cas, comme le cluster **my_cluster**.

3. Configurer les périphériques de fencing pour chaque nœud de la webfarm, en suivant les étapes décrites dans [Section 1.3, « Configuration d'une clôture »](#). Cet exemple configure le clôturage en utilisant deux ports de commutateur d'alimentation APC avec le nom d'hôte **zapc.example.com**.

3.2. Configurer un volume LVM dans un système de fichiers ext4

Ce cas d'utilisateur exige de créer un volume logique LVM dans un stockage partagé entre les nœuds du cluster.

La procédure suivante crée un volume logique LVM, puis crée un système de fichiers **ext4** sur ce volume. 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.



Note

Les volumes logiques, leurs partitions correspondantes et les périphériques utilisés par le cluster doivent être connectés au nœuds du cluster uniquement.

Comme la partition **/dev/sdb1** correspond à du stockage partagé, vous pourrez effectuer cette procédure sur un nœud uniquement.

1. Créez un volume physique LVM sur la partition **/dev/sdb1**.

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

2. Créez le groupe de volumes **my_vg** qui comprend le volume physique **/dev/sdb1**.

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

3. Créez un volume logique en utilisant le 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
...
```

4. Créer un système de fichiers **ext4** sur le volume logique **my_lv**.

```
[root@z1 ~]# mkfs.ext4 /dev/my_vg/my_lv
mke2fs 1.42.7 (21-Jan-2013)
Filesystem label=
OS type: Linux
...
```

3.3. Mise en place de Partages NFS

La procédure suivante configure le partage NFS d'échec de démon. Vous devez effectuer cette procédure sur un noeud du cluster uniquement.

1. Créez le répertoire **/nfsshare**.

```
[root@z1 ~]# mkdir /nfsshare
```

2. Montez le système de fichiers **ext4** que vous avez créé dans [Section 3.2, « Configurer un volume LVM dans un système de fichiers ext4 »](#) sur le répertoire **/nfsshare**.

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

3. Créez une arborescence de répertoires **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
```

4. Mettez les fichiers dans le répertoire **exports** pour que les clients NFS puissent y accéder. Dans cet exemple, nous avons créé des fichiers test nommés **clientdatafile1** et **clientdatafile2**.

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

5. Dé-montez le système de fichiers ext4 et désactivez le groupe de volumes LVM.

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

3.4. Activation exclusive d'un Groupe de volumes dans un cluster

La procédure suivante configure le groupe de volumes afin que seul le cluster puisse activer le groupe de volumes, et que le groupe de volumes ne puisse pas être activé en dehors du cluster au démarrage. Si le groupe de volumes est activé par un système en dehors du cluster, il y a un risque de corruption des métadonnées du groupe de volumes.

Cette procédure modifie l'entrée de **volume_list** dans le fichier de configuration **/etc/lvm/lvm.conf**. Les groupes de volume listés dans **volume_list** peuvent être automatiquement activés sur le noeud local en dehors du contrôle du gestionnaire de cluster. Les groupes de volumes associés au répertoires d'accueil et root local doivent être inclus dans cette liste. Tous les groupes de volumes gérés par le gestionnaire de cluster doivent être exclus de l'entrée **volume_list**. Notez bien que cette procédure ne nécessite pas la commande **clvmd**.

Effectuer la même procédure pour chaque noeud du cluster.

1. Exécutez la commande suivante pour que **locking_type** soit sur 1 et **use_lvmetad** sur 0 dans le fichier **/etc/lvm/lvm.conf**. Cette commande désactive et stoppe également les processus **lvmetad** immédiatement.

```
# lvmconf --enable-halvm --services --startstopservices
```

- Déterminez quels groupes de volumes sont actuellement configurés sur votre espace de stockage local par la commande suivante. Cela affichera une liste des groupes de volume actuellement configurés. Si vous avez de l'espace alloué dans les groupes de volumes distincts pour root et votre répertoire d'accueil sur ce nœud, vous verrez ces volumes dans la sortie, comme dans cet exemple.

```
# vgs --noheadings -o vg_name
my_vg
rhel_home
rhel_root
```

- Ajouter les groupes de volumes autres que **my_vg** (le groupe de volumes que vous venez de définir pour le cluster) comme entrées dans **volume_list** dans le fichier de configuration **/etc/lvm/lvm.conf**. Ainsi, si vous avez de l'espace alloué dans des groupes de volumes distincts pour root et pour votre répertoire d'accueil, vous pourrez dé-commenter la ligne **volume_list** du fichier **lvm.conf** et ajouter ces groupes de volumes en tant qu'entrées dans **volume_list** comme suit :

```
volume_list = [ "rhel_root", "rhel_home" ]
```



Note

Si aucun groupe de volume local n'est présent sur un nœud restant à activer en dehors du gestionnaire du cluster, vous devrez toujours initialiser l'entrée **volume_list** en tant que **volume_list = []**.

- Reconstruire l'image de démarrage **initramfs** pour garantir que l'image de démarrage n'essaiera pas d'activer un groupe de volumes contrôlée par le cluster, Mettez à jour le périphérique **initramfs** par la commande suivante. Cette commande met une minute à se compléter.

```
# dracut -H -f /boot/initramfs-$(uname -r).img $(uname -r)
```

- Redémarrez le nœud.



Note

Si vous avez installé un nouveau noyau Linux après avoir démarré le nœud sur lequel vous avez créé l'image de démarrage, la nouvelle image **initrd** sera pour le noyau qui était en cours lors de sa création et non pas pour le nouveau noyau qui exécute lorsque vous redémarrez le nœud. Vous pouvez vous assurer que le périphérique **initrd** qui convient est utilisée en exécutant la commande **uname -r** avant et après le redémarrage pour déterminer la version de noyau qui est en cours d'exécution. Si les versions ne sont pas les mêmes, mettez à jour le fichier **initrd** après le redémarrage avec le nouveau noyau, puis, redémarrez le nœud.

- Lorsque le nœud a redémarré, vérifiez si les services de cluster ont démarré à nouveau sur ce nœud en exécutant la commande **pcs statut de cluster** sur ce nœud. Si vous obtenez le message **Error: cluster is not currently running on this node**, alors exécutez la commande suivante.

```
# pcs cluster start
```

Sinon, attendez d'avoir redémarré chaque nœud du cluster et les services de cluster sur tous les nœuds du cluster par la commande suivante.

```
# pcs cluster start --all
```

3.5. Configurer les ressources du cluster

Cette section fournit la procédure de configuration des ressources du cluster pour ce cas d'utilisation.



Note

Il est conseillé, quand on crée une ressource de cluster avec **pcs resource create**, d'exécuter la commande **pcs status** immédiatement après avoir vérifié que la ressource exécute. Notez que si vous n'avez pas configuré de périphérique fence pour votre cluster, comme décrit dans [Section 1.3, « Configuration d'une clôture »](#), par défaut, les ressources ne redémarreront pas.

Si vous vous rendez compte que les ressources que vous configurez n'exécutent pas, vous pouvez exécuter la commande **pcs resource debug-start resource** pour tester la ressource de configuration. Cela va démarrer le service en dehors du contrôle du cluster et sans qu'il soit au courant. Quand les ressources configurées exécutent à nouveau, exécutez **pcs cluster cleanup resource** pour mettre le cluster au courant des mises à jour. Pour obtenir des informations sur la commande **pcs resource debug-start**, consulter le manuel *High Availability Add-On Reference*.

La procédure suivante configure les ressources système. Pour s'assurer que ces ressources fonctionnent toutes sur le même nœud, elles sont configurées dans le groupe de ressources **nfsgroup**. Les ressources vont démarrer dans l'ordre dans lequel vous les avez ajoutées au groupe, et elles s'arrêteront dans l'ordre inverse dans lequel elles ont été ajoutées au groupe. Exécutez cette procédure à partir d'un nœud du cluster uniquement.

1. La commande suivante crée la ressource LVM nommée **my_lvm**. Cette commande spécifie le paramètre **exclusive=true** pour s'assurer que le cluster uniquement soit capable d'activer le volume logique LVM. Comme le groupe de ressources **nfsgroup** n'existe pas, cette commande crée un groupe de ressources.

```
[root@z1 ~]# pcs resource create my_lvm LVM volgrpname=my_vg \
exclusive=true --group nfsgroup
```

Vérifier le statut du cluster pour vérifier 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): 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
```

2. Configurer une ressource **Filesystem** pour le cluster.



Note

Vous pouvez spécifier des options de montage dans le cadre de la configuration de ressource pour une ressource **Filesystem** par l'intermédiaire du paramètre **options=options**. Exécutez la commande **pcs resource describe Filesystem** pour obtenir des informations sur toutes les options de configuration.

La commande suivante configure une ressource **Filesystem** ext4 nommée **nfsshare** du groupe de ressource **nfsgroup**. Ce système de fichiers utilise le groupe de volume LVM et le système de fichiers ext4 que vous avez créés dans [Section 3.2, « Configurer un volume LVM dans un système de fichiers ext4 »](#) et qui sera monté sur le répertoire **/nfsshare** que vous avez créé dans [Section 3.3, « Mise en place de Partages NFS »](#).

```
[root@z1 ~]# pcs resource create nfsshare Filesystem \
device=/dev/my_vg/my_lv directory=/nfsshare \
fstype=ext4 --group nfsgroup
```

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): Started z1.example.com
  nfsshare (ocf::heartbeat:Filesystem): Started z1.example.com
...
```

3. Créer la ressource **nfsserver** nommée **nfs-daemon** faisant partie du groupe de ressources **nfsgroup**.

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

```
--group nfsgroup
[root@z1 ~]# pcs status
...
```

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

```
[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
```

5. Ajouter la ressource d'adresse IP flottante que les clients nfs utiliseront pour accéder au partage nfs. L'adresse IP flottante que vous spécifiez doit avoir une recherche DNS inversée **/etc/hosts** sur tous les nœuds du cluster. 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
```

6. Ajouter une ressource **nfsnotify** pour envoyer les notifications de redémarrage NFSv3 une fois que tout le déploiement NFS a été initialisé.



Note

Pour que la notification NFS soit traitée correctement, l'adresse IP flottante doit avoir un nom d'hôte associé, qui soit consistant avec les serveurs nfs et le client nfs à la fois.

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

Vous pourrez vérifier le statut du cluster après avoir créé les ressources et les ressources de contrainte. Notez que toutes les ressources exécutent 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): 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
...

```

3.6. Tester la configuration de ressource

Vous pouvez valider votre configuration système par la procédure suivante. Vous devez être en mesure de monter le système de fichiers exporté avec NFSv3 ou NFSv4.

1. Vérifier sur un noeud en dehors du cluster se trouvant dans le même réseau que le déploiement, que le partage NFS puisse être aperçu en montant un partage NFS. Dans 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

```

2. Pour vérifier que vous pouvez monter le partage NFS avec NFSv4, monter le partage NFS sur un répertoire du noeud du client. Après l'avoir monté, vérifiez que le contenu des répertoires d'exportation soient visibles. Dé-monter le partage après le test.

```

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

```

3. Vérifiez de pouvoir monter le partage NFS avec NFSv3. Après l'avoir monté, vérifiez que le fichier **clientdatafile1** soit bien visible. À la différence de NFSv4, comme NFSv3 n'utilise pas le système de fichiers virtuels, vous devrez monter une exportation spécifique. Dé-monter le partage après le test.

```

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

```

4. Pour tester les basculements, procédez aux étapes suivantes.
 - a. Montez le partage nfs sur un noeud qui se trouve en dehors du cluster et vérifiez l'accès au **clientdatafile1** que nous avons créé dans [Section 3.3, « Mise en place de Partages NFS »](#).

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

- b. À partir d'un noeud du cluster, déterminer quel noeud du cluster exécute le **nfsgroup**. Dans cet exemple, **nfsgroup** exécute 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): 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
  ...
```

- c. À partir d'un noeud du cluster, mettez le noeud qui exécute **nfsgroup** en mode de veille.

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

- d. Vérifiez que **nfsgroup** démarre bien sur l'autre noeud de cluster.

```
[root@z1 ~]# pcs status
...
Full list of resources:
  Resource Group: nfsgroup
  my_lvm (ocf::heartbeat:LVM): 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
  ...
```

- e. À partir du noeud qui se trouve en dehors du cluster et sur lequel vous avez monté le partage nfs, vérifiez que ce noeud extérieur continue d'avoir accès au fichier test dans le montage NFS.

```
# ls nfsshare
clientdatafile1
```

Le service disparaîtra durant un bref moment lors du basculement, mais le client devrait se restaurer sans intervention de la part de l'utilisateur. Par défaut, les clients utilisant NFSv4 peuvent prendre jusqu'à 90 secondes pour se restaurer à la suite d'un montage ; ces 90 secondes représentent la période de grâce du bail du fichier NFSv4 observée par le serveur

au démarrage. Les clients NFSv3 devraient pouvoir accéder au montage à nouveau en quelques secondes.

- f. À partir d'un noeud du cluster, supprimer le noeud qui exécutait **nfs**group en mode d'attente. Cela ne déplacera pas en soi les ressources du cluster vers ce noeud.

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

Annexe A. Historique des versions

Version 1.2-4	Tue Oct 25 2016	Corina Galicher Roe
Fichiers de traduction synchronisés avec les sources XML 1.2-3		
Version 1.2-3.2	Tue Oct 11 2016	Terry Chuang
Fichiers de traduction synchronisés avec les sources XML 1.2-3		
Version 1.2-3.1	Thu Sep 8 2016	Terry Chuang
Fichiers de traduction synchronisés avec les sources XML 1.2-3		
Version 1.2-3	Mon Nov 9 2015	Steven Levine
Préparer un document pour une publication 7.2 GA.		
Version 1.2-2	Tue Aug 18 2015	Steven Levine
Préparer un document pour une publication 7.2 Beta		
Version 1.1-9	Mon Feb 16 2015	Steven Levine
Version pour la distribution 7.1 GA		
Version 1.1-8	Thu Dec 11 2014	Steven Levine
Version pour la distribution 7.1 Bêta		
Version 1.1-7	Tue Dec 9 2014	Steven Levine
Ajout de la procédure de configuration de cluster		
Version 1.1-6	Mon Dec 8 2014	Steven Levine
Mise à jour de la procédure de cluster d'équilibrage des charges		
Version 1.1-5.1	Fri Dec 5 2014	Steven Levine
Mise à jour pour créer un nouvel ordonnancement sur la page splash de Red Hat Enterprise Linux.		
Version 1.1-5	Thu Dec 04 2014	Steven Levine
Version pour la distribution 7.1 Bêta		
Version 0.1-3.3	Mon Jun 2 2014	Steven Levine
Version pour la distribution GA 7.0		
Version 0.1-3.1	Wed May 21 2014	Steven Levine
Résoud : #886235 Utilisation du document volume_list		
Version 0.1-3	Tue May 20 2014	Steven Levine
Reconstruire pour intégrer les changements de style ou les mises à jour d'ébauches		
Version 0.1-2	Wed Apr 9 2014	Steven Levine
Mise à jour de l'ébauche Beta		
Version 0.1-1	Fri Dec 6 2013	Steven Levine
Ébauche Beta		

Version 0.0-1**Wed Jan 16 2013****Steven Levine**

Première version de Red Hat Enterprise Linux 7