

Remplacement PCRF du serveur de contrôleur UCS C240 M4

Contenu

[Introduction](#)

[Conditions préalables](#)

[Sauvegarde](#)

[Vérification de l'état préliminaire](#)

[Désactiver la clôture dans le cluster de contrôleurs](#)

[Installer le nouveau noeud de contrôleur](#)

[Remplacement de noeud de contrôleur dans le nuage](#)

[Préparation de la suppression du noeud de contrôleur défaillant](#)

[Préparation de l'ajout d'un nouveau noeud de contrôleur](#)

[Intervention manuelle](#)

[Vérifier les services Overcloud dans le contrôleur](#)

[Finaliser les routeurs d'agents de couche 3](#)

[Finaliser les services de calcul](#)

[Redémarrer la clôture sur les noeuds du contrôleur](#)

Introduction

Ce document décrit les étapes requises pour remplacer un serveur contrôleur défectueux dans une configuration Ultra-M qui héberge les fonctions de réseau virtuel (VNF) CPS.

Conditions préalables

Sauvegarde

En cas de récupération, Cisco recommande d'effectuer une sauvegarde de la base de données OSPD (DB) en procédant comme suit :

```
[root@director ~]# mysqldump --opt --all-databases > /root/undercloud-all-databases.sql
[root@director ~]# tar --xattrs -czf undercloud-backup-`date +%F`.tar.gz /root/undercloud-all-databases.sql
/etc/my.cnf.d/server.cnf /var/lib/glance/images /srv/node /home/stack
tar: Removing leading `/' from member names
```

Vérification de l'état préliminaire

Il est important de vérifier l'état actuel de l'environnement et des services OpenStack et de s'assurer qu'il est sain avant de procéder à la procédure de remplacement. Il peut aider à éviter des complications au moment du remplacement du contrôleur.

Étape 1. Vérifiez l'état d'OpenStack et la liste des noeuds :

```
[stack@director ~]$ source stackrc
[stack@director ~]$ openstack stack list --nested
[stack@director ~]$ ironic node-list
[stack@director ~]$ nova list
```

Étape 2. Vérifiez l'état de Pacemaker sur les contrôleurs.

Connectez-vous à l'un des contrôleurs actifs et vérifiez l'état du pacemaker. Tous les services doivent être exécutés sur les contrôleurs disponibles et arrêtés sur le contrôleur défaillant.

```
[stack@pod1-controller-0 ~]# pcs status

<snip>
Online: [ pod1-controller-0 pod1-controller-1 ]
OFFLINE: [ pod1-controller-2 ]
Full list of resources:
ip-11.120.0.109 (ocf::heartbeat:IPaddr2): Started pod1-controller-0
ip-172.25.22.109 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
ip-192.200.0.107 (ocf::heartbeat:IPaddr2): Started pod1-controller-0

Clone Set: haproxy-clone [haproxy]
Started: [ pod1-controller-0 pod1-controller-1 ]
Stopped: [ pod1-controller-2 ]

Master/Slave Set: galera-master [galera]
Masters: [ pod1-controller-0 pod1-controller-1 ]
Stopped: [ pod1-controller-2 ]
ip-11.120.0.110 (ocf::heartbeat:IPaddr2): Started pod1-controller-0
ip-11.119.0.110 (ocf::heartbeat:IPaddr2): Started pod1-controller-1

Clone Set: rabbitmq-clone [rabbitmq]
Started: [ pod1-controller-0 pod1-controller-1 ]
Stopped: [ pod1-controller-2 ]

Master/Slave Set: redis-master [redis]
Masters: [ pod1-controller-0 ]
Slaves: [ pod1-controller-1 ]
Stopped: [ pod1-controller-2 ]

ip-11.118.0.104 (ocf::heartbeat:IPaddr2): Started pod1-controller-1
openstack-cinder-volume (systemd:openstack-cinder-volume): Started pod1-controller-0

my-ipmilan-for-controller-6 (stonith:fence_ipmilan): Started pod1-controller-1
my-ipmilan-for-controller-4 (stonith:fence_ipmilan): Started pod1-controller-0
my-ipmilan-for-controller-7 (stonith:fence_ipmilan): Started pod1-controller-0

Failed Actions:
Daemon Status:

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

Dans cet exemple, Controller-2 est hors connexion. Il sera donc remplacé. Les contrôleurs 0 et 1 sont opérationnels et exécutent les services de cluster.

Étape 3. Vérifiez l'état MariaDB dans les contrôleurs actifs.

neutron-dhcp-agent.service	loaded active running	OpenStack Neutron DHCP Agent
neutron-openvswitch-agent.service Agent	loaded active running	OpenStack Neutron Open vSwitch
neutron-ovs-cleanup.service Cleanup Utility	loaded active exited	OpenStack Neutron Open vSwitch
neutron-server.service	loaded active running	OpenStack Neutron Server
openstack-aodh-evaluator.service service	loaded active running	OpenStack Alarm evaluator
openstack-aodh-listener.service service	loaded active running	OpenStack Alarm listener
openstack-aodh-notifier.service service	loaded active running	OpenStack Alarm notifier
openstack-ceilometer-central.service agent	loaded active running	OpenStack ceilometer central
openstack-ceilometer-collector.service service	loaded active running	OpenStack ceilometer collection
openstack-ceilometer-notification.service notification agent	loaded active running	OpenStack ceilometer
openstack-glance-api.service named Glance) API server	loaded active running	OpenStack Image Service (code-
openstack-glance-registry.service named Glance) Registry server	loaded active running	OpenStack Image Service (code-
openstack-heat-api-cfn.service API Service	loaded active running	Openstack Heat CFN-compatible
openstack-heat-api.service	loaded active running	OpenStack Heat API Service
openstack-heat-engine.service	loaded active running	Openstack Heat Engine Service
openstack-ironic-api.service	loaded active running	OpenStack Ironic API service
openstack-ironic-conductor.service service	loaded active running	OpenStack Ironic Conductor
openstack-ironic-inspector-dnsmasq.service Ironic Inspector	loaded active running	PXE boot dnsmasq service for
openstack-ironic-inspector.service for OpenStack Ironic	loaded active running	Hardware introspection service
openstack-mistral-api.service	loaded active running	Mistral API Server
openstack-mistral-engine.service	loaded active running	Mistral Engine Server
openstack-mistral-executor.service	loaded active running	Mistral Executor Server
openstack-nova-api.service	loaded active running	OpenStack Nova API Server
openstack-nova-cert.service	loaded active running	OpenStack Nova Cert Server
openstack-nova-compute.service	loaded active running	OpenStack Nova Compute Server
openstack-nova-conductor.service	loaded active running	OpenStack Nova Conductor Server
openstack-nova-scheduler.service	loaded active running	OpenStack Nova Scheduler Server
openstack-swift-account-reaper.service (swift) - Account Reaper	loaded active running	OpenStack Object Storage
openstack-swift-account.service (swift) - Account Server	loaded active running	OpenStack Object Storage
openstack-swift-container-updater.service (swift) - Container Updater	loaded active running	OpenStack Object Storage
openstack-swift-container.service (swift) - Container Server	loaded active running	OpenStack Object Storage
openstack-swift-object-updater.service (swift) - Object Updater	loaded active running	OpenStack Object Storage
openstack-swift-object.service (swift) - Object Server	loaded active running	OpenStack Object Storage
openstack-swift-proxy.service (swift) - Proxy Server	loaded active running	OpenStack Object Storage
openstack-zaqar.service Service (code-named Zaqar) Server	loaded active running	OpenStack Message Queuing
openstack-zaqar@1.service Service (code-named Zaqar) Server Instance 1	loaded active running	OpenStack Message Queuing
openvswitch.service	loaded active exited	Open vSwitch

LOAD = Reflects whether the unit definition was properly loaded.

ACTIVE = The high-level unit activation state, i.e. generalization of SUB.

SUB = The low-level unit activation state, values depend on unit type.

37 loaded units listed. Pass --all to see loaded but inactive units, too.
To show all installed unit files use 'systemctl list-unit-files'.

Désactiver la clôture dans le cluster de contrôleurs

```
[root@pod1-controller-0 ~]# sudo pcs property set stonith-enabled=false  
[root@pod1-controller-0 ~]# pcs property show
```

```
Cluster Properties:  
cluster-infrastructure: corosync  
cluster-name: tripleo_cluster  
dc-version: 1.1.15-11.e17_3.4-e174ec8  
have-watchdog: false  
last-lrm-refresh: 1510809585  
maintenance-mode: false  
redis_REPL_INFO: pod1-controller-0  
stonith-enabled: false
```

```
Node Attributes:  
pod1-controller-0: rmq-node-attr-last-known-rabbitmq=rabbit@pod1-controller-0  
pod1-controller-1: rmq-node-attr-last-known-rabbitmq=rabbit@pod1-controller-1  
pod1-controller-2: rmq-node-attr-last-known-rabbitmq=rabbit@pod1-controller-2
```

Installer le nouveau noeud de contrôleur

Étape 1. Les étapes permettant d'installer un nouveau serveur UCS C240 M4 et les étapes de configuration initiale peuvent être référencées dans le [Guide d'installation et de maintenance du serveur Cisco UCS C240 M4](#).

Étape 2. Connectez-vous au serveur à l'aide de l'adresse IP CIMC.

Étape 3. Effectuez la mise à niveau du BIOS si le micrologiciel n'est pas conforme à la version recommandée précédemment. Les étapes de mise à niveau du BIOS sont indiquées ici :

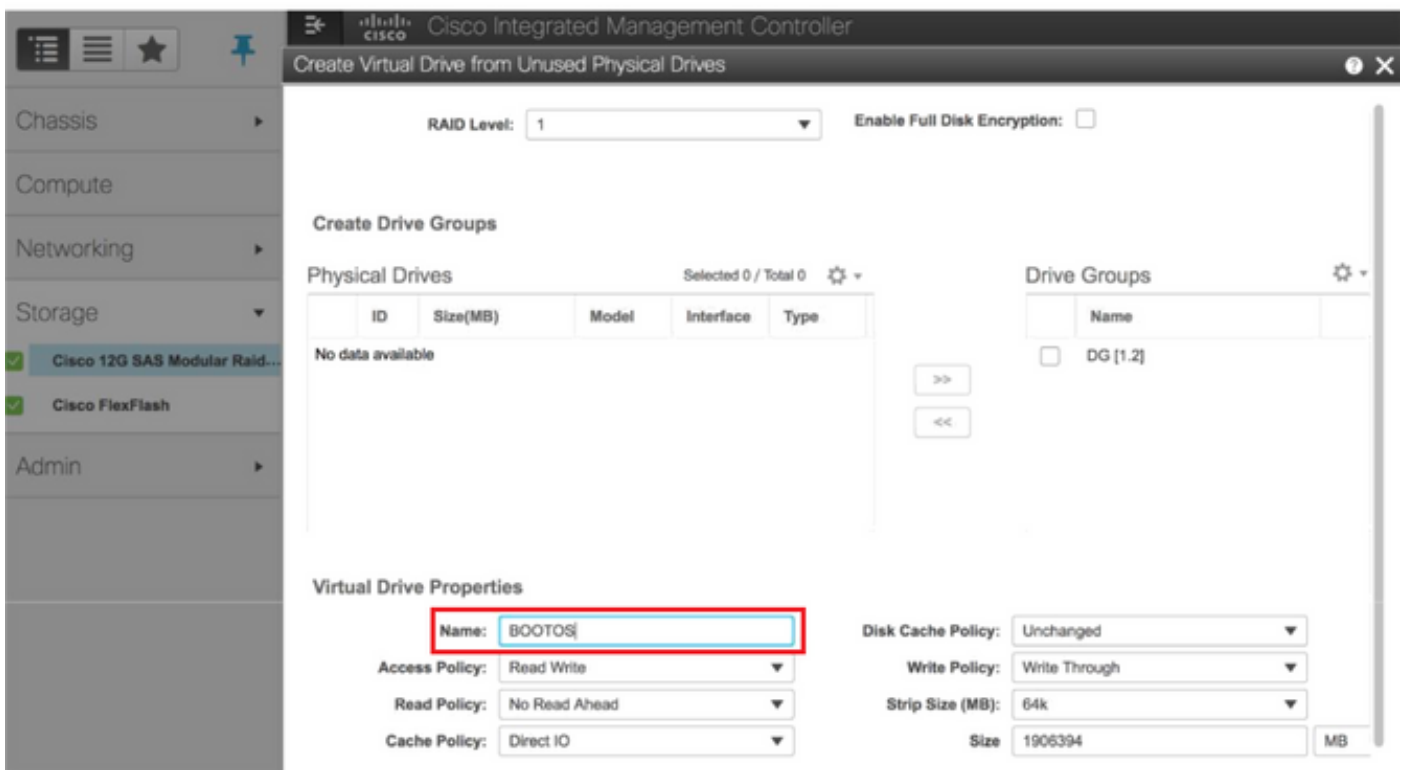
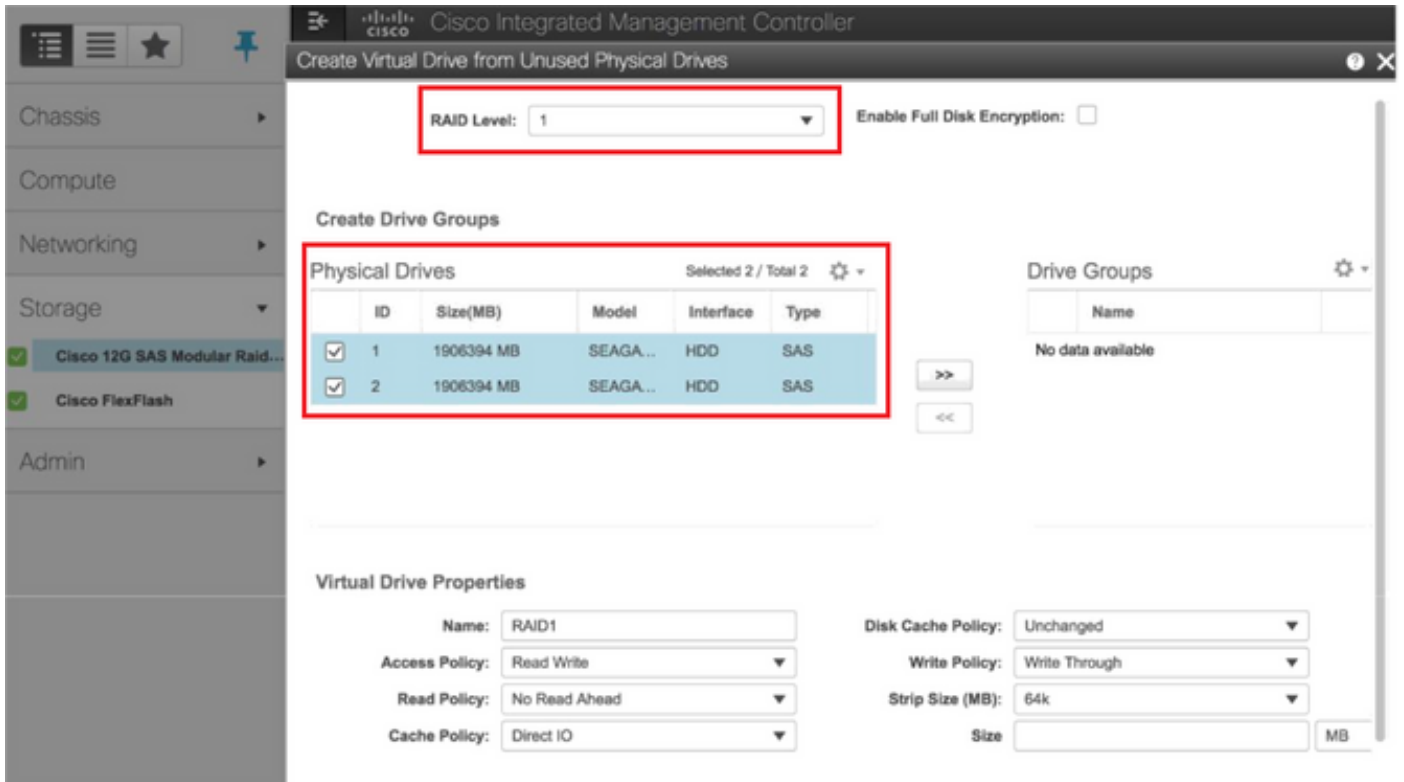
[Guide de mise à niveau du BIOS du serveur rack Cisco UCS série C](#)

Étape 4. Vérifiez l'état des lecteurs physiques. Il doit être **Non configuré**. Accédez à **Storage > Cisco 12G SAS Modular Raid Controller (SLOT-HBA) > Physical Drive Info**.

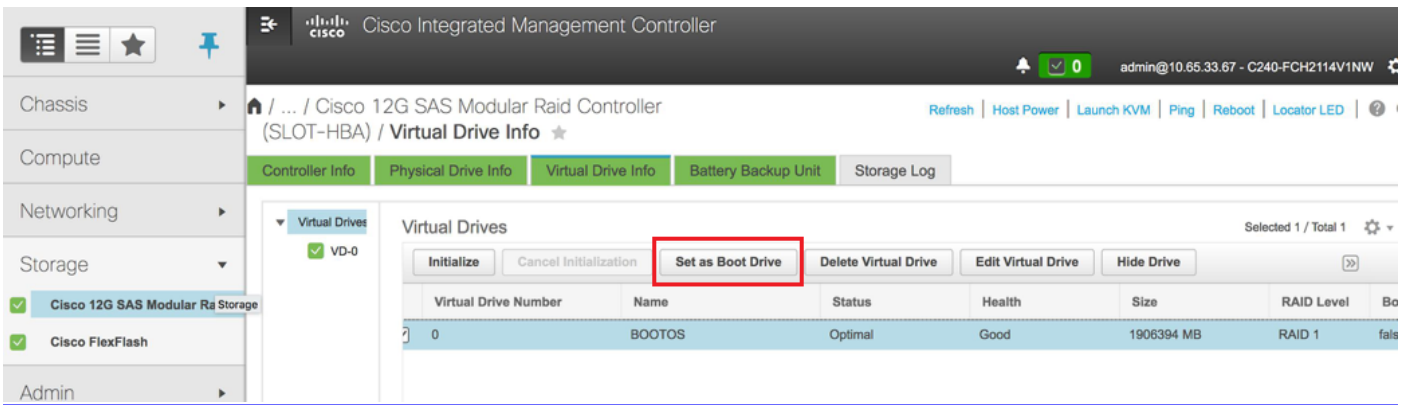
Controller	Physical Drive Number	Status	Health	Boot Drive	Drive Firmware
<input type="checkbox"/> SLOT-HBA	1	Unconfigured Good	Good	false	N003
<input type="checkbox"/> SLOT-HBA	2	Unconfigured Good	Good	false	N003

Étape 5. Afin de créer un lecteur virtuel à partir des lecteurs physiques avec RAID Niveau 1 :

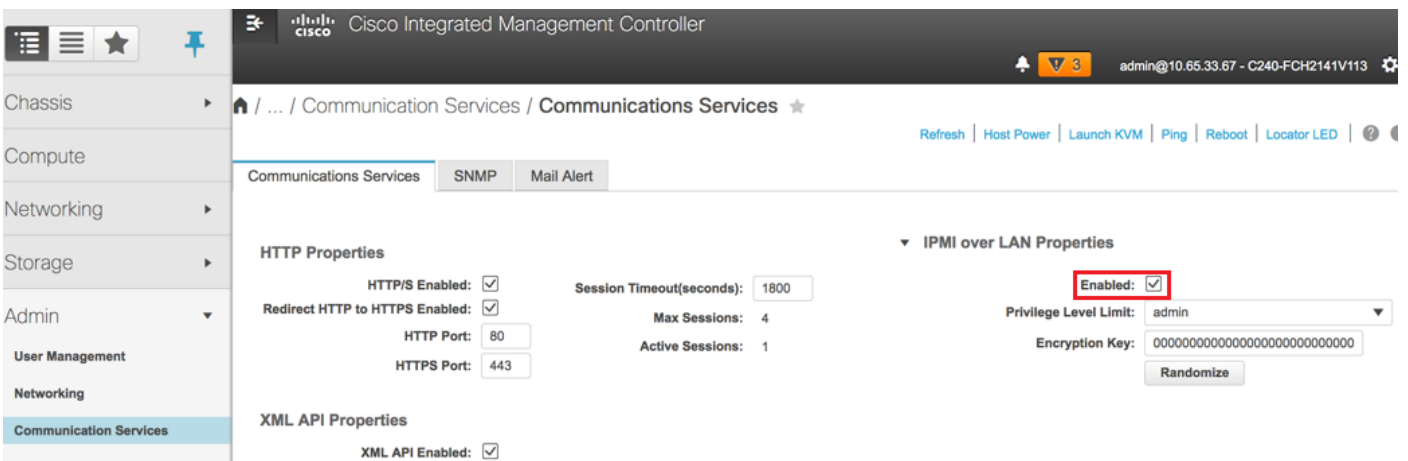
accédez à **Storage > Cisco 12G SAS Modular Raid Controller (SLOT-HBA) > Controller Info > Create Virtual Drive from Unused Physical Drives**, comme illustré dans l'image.



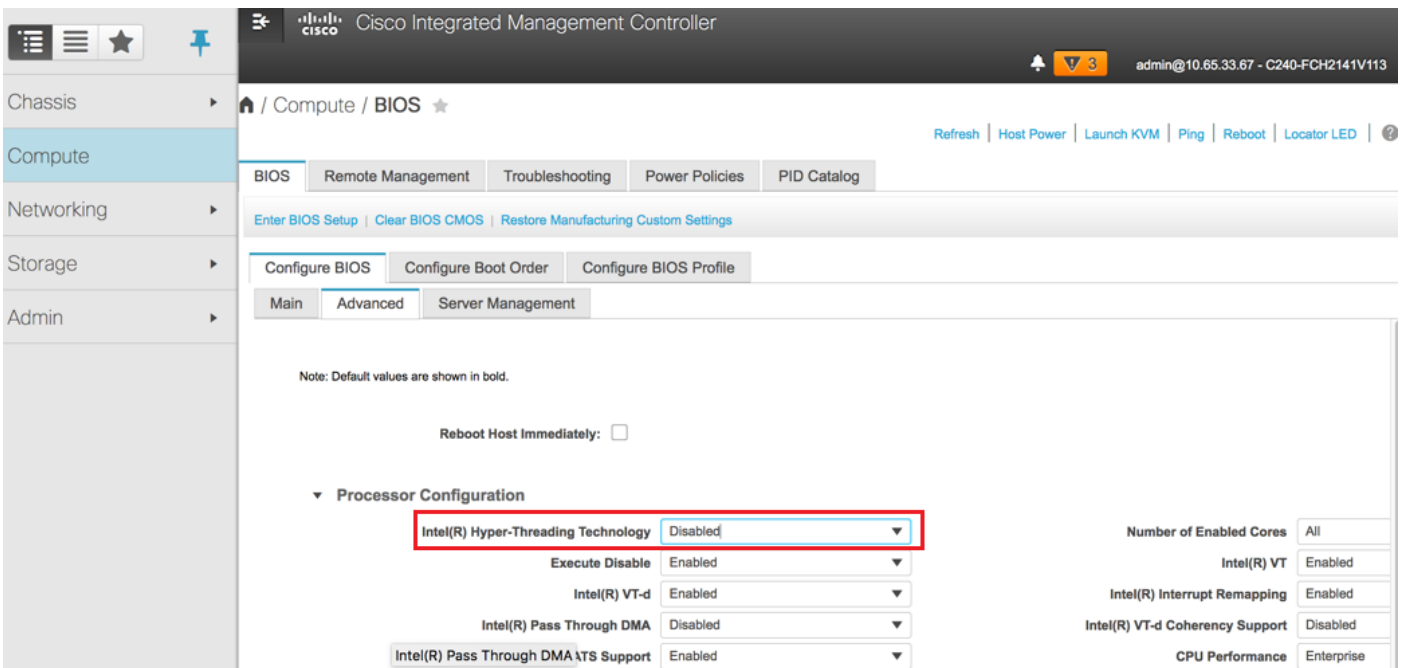
- Sélectionnez la VD et configurez **Set as Boot Drive** :



Étape 6. Afin d'activer IPMI sur LAN, accédez à **Admin > Communications Services > Communication Services**.



Étape 7. Afin de désactiver l'hyperthreading, naviguez jusqu'à **Compute > BIOS > Configure BIOS > Advanced > Processor Configuration**, comme illustré dans l'image.



Note: L'image est montrée ici et les étapes de configuration mentionnées dans cette section se rapportent à la version 3.0(3e) du firmware et il peut y avoir de légères variations si vous travaillez sur d'autres versions.

Remplacement de noeud de contrôleur dans le nuage

Cette section décrit les étapes requises pour remplacer le contrôleur défectueux par le nouveau dans le nuage de dessus. Pour cela, le script **Deployment.sh** qui a été utilisé pour activer la pile sera réutilisé. Au moment du déploiement, dans la phase **ControllerNodesPostDeployment**, la mise à jour échouerait en raison de certaines limitations dans les modules Puppet. Une intervention manuelle est requise avant de redémarrer le script de déploiement.

Préparation de la suppression du noeud de contrôleur défaillant

Étape 1. Identifiez l'index du contrôleur défaillant. L'index est le suffixe numérique sur le nom du contrôleur dans la sortie de la liste des serveurs OpenStack. Dans cet exemple, l'index est 2 :

```
[stack@director ~]$ nova list | grep controller
| 5813a47e-af27-4fb9-8560-75decd3347b4 | pod1-controller-0 | ACTIVE | - | Running
| ctlplane=192.200.0.152 |
| 457f023f-d077-45c9-bbea-dd32017d9708 | pod1-controller-1 | ACTIVE | - | Running
| ctlplane=192.200.0.154 |
| d13bb207-473a-4e42-a1e7-05316935ed65 | pod1-controller-2 | ACTIVE | - | Running
| ctlplane=192.200.0.151 |
```

Étape 2. Créez un fichier Yaml **~templates/remove-controller.yaml** qui définirait le noeud à supprimer. Utilisez l'index trouvé à l'étape précédente pour l'entrée de la liste des ressources.

```
[stack@director ~]$ cat templates/remove-controller.yaml
```

```
parameters:
  ControllerRemovalPolicies:
    [{'resource_list': ['2']}]
```

```
parameter_defaults:
  CorosyncSettleTries: 5
```

Étape 3. Faites une copie du script de déploiement qui est utilisé afin d'installer le surcloud et insérez une ligne afin d'inclure le fichier **remove-controller.yaml** créé précédemment.

```
[stack@director ~]$ cp deploy.sh deploy-removeController.sh
[stack@director ~]$ cat deploy-removeController.sh
time openstack overcloud deploy --templates \
-r ~/custom-templates/custom-roles.yaml \
-e /home/stack/templates/remove-controller.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/puppet-pacemaker.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml \
-e /usr/share/openstack-tripleo-heat-templates/environments/neutron-sriov.yaml \
-e ~/custom-templates/network.yaml \
-e ~/custom-templates/ceph.yaml \
-e ~/custom-templates/compute.yaml \
-e ~/custom-templates/layout-removeController.yaml \
-e ~/custom-templates/rabbitmq.yaml \
--stack pod1 \
--debug \
--log-file overcloudDeploy_$(date +%m_%d_%y__%H_%M_%S).log \
--neutron-flat-networks phys_pcie1_0,phys_pcie1_1,phys_pcie4_0,phys_pcie4_1 \
--neutron-network-vlan-ranges datacentre:101:200 \
--neutron-disable-tunneling \
```



```
--verbose --timeout 180
```

Étape 4. Identifiez l'ID du contrôleur à remplacer, en utilisant les commandes mentionnées ici et passez en mode maintenance.

```
[stack@director ~]$ nova list | grep controller
```

```
| 5813a47e-af27-4fb9-8560-75decd3347b4 | pod1-controller-0 | ACTIVE | - | Running  
| ctlplane=192.200.0.152 |  
  
| 457f023f-d077-45c9-bbea-dd32017d9708 | pod1-controller-1 | ACTIVE | - | Running  
| ctlplane=192.200.0.154 |  
  
| d13bb207-473a-4e42-a1e7-05316935ed65 | pod1-controller-2 | ACTIVE | - | Running  
| ctlplane=192.200.0.151 |
```

```
[stack@director ~]$ openstack baremetal node list | grep d13bb207-473a-4e42-a1e7-05316935ed65
```

```
| e7c32170-c7d1-4023-b356-e98564a9b85b | None | d13bb207-473a-4e42-a1e7-05316935ed65 | power  
off | active | False |
```

```
[stack@b10-ospd ~]$ openstack baremetal node maintenance set e7c32170-c7d1-4023-b356-e98564a9b85b
```

```
[stack@director ~]$ openstack baremetal node list | grep True
```

```
| e7c32170-c7d1-4023-b356-e98564a9b85b | None | d13bb207-473a-4e42-a1e7-05316935ed65 | power  
off | active | True |
```

Étape 5. Afin de s'assurer que la base de données s'exécute au moment de la procédure de remplacement, supprimez Galera du contrôle pacemaker et exécutez cette commande sur l'un des contrôleurs actifs.

```
[root@pod1-controller-0 ~]# sudo pcs resource unmanage galera  
[root@pod1-controller-0 ~]# sudo pcs status
```

```
Cluster name: tripleo_cluster  
Stack: corosync  
Current DC: pod1-controller-0 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum  
Last updated: Thu Nov 16 16:51:18 2017 Last change: Thu Nov 16 16:51:12 2017  
by root via crm_resource on pod1-controller-0  
3 nodes and 22 resources configured  
Online: [ pod1-controller-0 pod1-controller-1 ]  
OFFLINE: [ pod1-controller-2 ]
```

```
Full list of resources:
```

```
ip-11.120.0.109 (ocf::heartbeat:IPaddr2): Started pod1-controller-0  
ip-172.25.22.109 (ocf::heartbeat:IPaddr2): Started pod1-controller-1  
ip-192.200.0.107 (ocf::heartbeat:IPaddr2): Started pod1-controller-0
```

```
Clone Set: haproxy-clone [haproxy]  
Started: [ pod1-controller-0 pod1-controller-1 ]  
Stopped: [ pod1-controller-2 ]
```

```
Master/Slave Set: galera-master [galera] (unmanaged)
```

```
galera (ocf::heartbeat:galera): Master pod1-controller-0 (unmanaged)  
galera (ocf::heartbeat:galera): Master pod1-controller-1 (unmanaged)
```

```
Stopped: [ pod1-controller-2 ]
ip-11.120.0.110      (ocf::heartbeat:IPAddr2):      Started pod1-controller-0
ip-11.119.0.110     (ocf::heartbeat:IPAddr2):      Started pod1-controller-1
```

<snip>

Préparation de l'ajout d'un nouveau noeud de contrôleur

Étape 1. Créez un fichier **controllerRMA.json** avec uniquement les détails du nouveau contrôleur. Vérifiez que le numéro d'index du nouveau contrôleur n'a pas été utilisé auparavant. Généralement, passez au numéro de contrôleur le plus élevé suivant.

Exemple : Le plus haut précédent était Controller-2, donc créez Controller-3.

Note: Tenez compte du format json.

```
[stack@director ~]$ cat controllerRMA.json
{
  "nodes": [
    {
      "mac": [
        <MAC_ADDRESS>
      ],
      "capabilities": "node:controller-3,boot_option:local",
      "cpu": "24",
      "memory": "256000",
      "disk": "3000",
      "arch": "x86_64",
      "pm_type": "pxe_ipmitool",
      "pm_user": "admin",
      "pm_password": "<PASSWORD>",
      "pm_addr": "<CIMC_IP>"
    }
  ]
}
```

Étape 2. Importez le nouveau noeud à l'aide du fichier json créé à l'étape précédente.

```
[stack@director ~]$ openstack baremetal import --json controllerRMA.json
```

```
Started Mistral Workflow. Execution ID: 67989c8b-1225-48fe-ba52-3a45f366e7a0
```

```
Successfully registered node UUID 048ccb59-89df-4f40-82f5-3d90d37ac7dd
```

```
Started Mistral Workflow. Execution ID: c6711b5f-fa97-4c86-8de5-b6bc7013b398
```

```
Successfully set all nodes to available.
```

```
[stack@director ~]$ openstack baremetal node list | grep available
```

```
| 048ccb59-89df-4f40-82f5-3d90d37ac7dd | None | None | power
off | available | False
```

Étape 3. Définissez le noeud à gérer.

```
[stack@director ~]$ openstack baremetal node manage 048ccb59-89df-4f40-82f5-3d90d37ac7dd
```

```
[stack@director ~]$ openstack baremetal node list | grep off
```

```
| 048ccb59-89df-4f40-82f5-3d90d37ac7dd | None | None | power off | manageable | False |
```

Étape 4. Exécuter l'introspection.

```
[stack@director ~]$ openstack overcloud node introspect 048ccb59-89df-4f40-82f5-3d90d37ac7dd --
provide
Started Mistral Workflow. Execution ID: f73fb275-c90e-45cc-952b-bfc25b9b5727
Waiting for introspection to finish...
Successfully introspected all nodes.
Introspection completed.
Started Mistral Workflow. Execution ID: a892b456-eb15-4c06-b37e-5bc3f6c37c65
Successfully set all nodes to available
```

```
[stack@director ~]$ openstack baremetal node list | grep available
| 048ccb59-89df-4f40-82f5-3d90d37ac7dd | None | None | power
off | available | False |
```

Étape 5. Marquer le noeud disponible avec les nouvelles propriétés du contrôleur. Assurez-vous d'utiliser l'ID de contrôleur désigné pour le nouveau contrôleur, comme utilisé dans le fichier **controllerRMA.json**.

```
[stack@director ~]$ openstack baremetal node set --property capabilities='node:controller-
3,profile:control,boot_option:local' 048ccb59-89df-4f40-82f5-3d90d37ac7dd
```

Étape 6. Dans le script de déploiement, il y a un modèle personnalisé appelé **layout.yaml** qui, entre autres, spécifie quelles adresses IP sont attribuées aux contrôleurs pour les différentes interfaces. Sur une nouvelle pile, trois adresses sont définies pour Controller-0, Controller-1 et Controller-2. Lorsque vous ajoutez un nouveau contrôleur, assurez-vous d'ajouter une adresse IP suivante dans l'ordre pour chaque sous-réseau.

```
ControllerIPs:
internal_api:
- 11.120.0.10
- 11.120.0.11
- 11.120.0.12
- 11.120.0.13
tenant:
- 11.117.0.10
- 11.117.0.11
- 11.117.0.12
- 11.117.0.13
storage:
- 11.118.0.10
- 11.118.0.11
- 11.118.0.12
- 11.118.0.13
storage_mgmt:
- 11.119.0.10
- 11.119.0.11
- 11.119.0.12
- 11.119.0.13
```

Étape 7. Exécutez maintenant le fichier **Deployment-removecontroller.sh** précédemment créé, afin de supprimer l'ancien noeud et d'ajouter le nouveau noeud.

Note: Cette étape devrait échouer dans ControllerNodesDeployment_Step1. À ce stade, une intervention manuelle est nécessaire.

```
[stack@b10-ospd ~]$ ./deploy-addController.sh
```

```
START with options: [u'overcloud', u'deploy', u'--templates', u'-r', u'/home/stack/custom-templates/custom-roles.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-templates/environments/puppet-pacemaker.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-templates/environments/neutron-sriov.yaml', u'-e', u'/home/stack/custom-templates/network.yaml', u'-e', u'/home/stack/custom-templates/ceph.yaml', u'-e', u'/home/stack/custom-templates/compute.yaml', u'-e', u'/home/stack/custom-templates/layout-removeController.yaml', u'-e', u'/home/stack/custom-templates/rabbitmq.yaml', u'--stack', u'newtonoc', u'--debug', u'--log-file', u'overcloudDeploy_11_15_17__07_46_35.log', u'--neutron-flat-networks', u'phys_pcie1_0,phys_pcie1_1,phys_pcie4_0,phys_pcie4_1', u'--neutron-network-vlan-ranges', u'datacentre:101:200', u'--neutron-disable-tunneling', u'--verbose', u'--timeout', u'180']
```

DeploymentError: Heat Stack update failed
END return value: 1

```
real    42m1.525s
user    0m3.043s
sys     0m0.614s
```

La progression/l'état du déploiement peut être surveillée à l'aide des commandes suivantes :

```
[stack@director~]$ openstack stack list --nested | grep -iv complete
```

```

+-----+-----+-----+-----+-----+-----+
| ID                                     | Stack                                     |
Name                                     |                                     |
| ID                                     | Stack                                     |
Time          | Updated Time          | Parent                                     | Stack Status          | Creation
+-----+-----+-----+-----+-----+-----+
| c1e338f2-877e-4817-93b4-9a3f0c0b3d37 | pod1-AllNodesDeploySteps-5psegydpwxij-  | UPDATE_FAILED        |
ComputeDeployment_Step1-swnuzjixac43     | e90f00ef-2499-4ec3-90b4-d7def6e97c47 |
| 1db4fef4-45d3-4125-bd96-2cc3297a69ff | pod1-AllNodesDeploySteps-5psegydpwxij-  | UPDATE_FAILED        |
ControllerDeployment_Step1-             | e90f00ef-2499-4ec3-90b4-                |
hmn3hpruubcn                            | d7def6e97c47 |
| e90f00ef-2499-4ec3-90b4-d7def6e97c47 | pod1-AllNodesDeploySteps-               | UPDATE_FAILED        | 2017-10-08T13:59:25Z | 2017-11-
16T18:09:25Z | 6c4b604a-55a4-4a19-9141-28c844816c0d |
| 6c4b604a-55a4-4a19-9141-28c844816c0d | pod1                                     | UPDATE_FAILED        | 2017-10-
08T12:37:11Z | 2017-11-16T17:35:35Z | None
+-----+-----+-----+-----+-----+-----+

```

Intervention manuelle

Étape 1. Sur le serveur OSP-D, exécutez la commande `OpenStack server list` afin de répertorier les contrôleurs disponibles. Le contrôleur nouvellement ajouté doit apparaître dans la liste.

```
[stack@director ~]$ openstack server list | grep controller
| 3e6c3db8-ba24-48d9-b0e8-1e8a2eb8b5ff | pod1-controller-3 | ACTIVE | ctlplane=192.200.0.103 |
overcloud-full |
| 457f023f-d077-45c9-bbea-dd32017d9708 | pod1-controller-1 | ACTIVE | ctlplane=192.200.0.154 |
overcloud-full |
| 5813a47e-af27-4fb9-8560-75decd3347b4 | pod1-controller-0 | ACTIVE | ctlplane=192.200.0.152 |
overcloud-full |
```

Étape 2. Connectez-vous à l'un des contrôleurs actifs (et non au contrôleur récemment ajouté) et consultez le fichier `/etc/corosync/corosync.conf`. Recherchez la `nodelist` qui attribue un `noeud` à chaque contrôleur. Recherchez l'entrée du noeud ayant échoué et notez son `noeud` :

```
[root@pod1-controller-0 ~]# cat /etc/corosync/corosync.conf
totem {
    version: 2
    secauth: off
    cluster_name: tripleo_cluster
    transport: udpu
    token: 10000
}

nodelist {
    node {
        ring0_addr: pod1-controller-0
        nodeid: 5
    }
    node {
        ring0_addr: pod1-controller-1
        nodeid: 7
    }
    node {
        ring0_addr: pod1-controller-2
        nodeid: 8
    }
}
```

Étape 3. Connectez-vous à chacun des contrôleurs actifs. Supprimez le noeud défaillant et redémarrez le service. Dans ce cas, supprimez `pod1-controller-2`. N'effectuez pas cette action sur le contrôleur nouvellement ajouté.

```
[root@pod1-controller-0 ~]# sudo pcs cluster localnode remove pod1-controller-2
pod1-controller-2: successfully removed!
[root@pod1-controller-0 ~]# sudo pcs cluster reload corosync
Corosync reloaded
```

```
[root@pod1-controller-1 ~]# sudo pcs cluster localnode remove pod1-controller-2
pod1-controller-2: successfully removed!
[root@pod1-controller-1 ~]# sudo pcs cluster reload corosync
Corosync reloaded
```

Étape 4. Exécutez cette commande à partir de l'un des contrôleurs actifs afin de supprimer le noeud défaillant du cluster.

```
[root@pod1-controller-0 ~]# sudo crm_node -R pod1-controller-2 --force
```

Étape 5. Exécutez cette commande à partir de l'un des contrôleurs actifs afin de supprimer le noeud défaillant du cluster **rabbitmq**.

```
[root@pod1-controller-0 ~]# sudo rabbitmqctl forget_cluster_node rabbit@pod1-controller-2  
Removing node 'rabbit@newtonoc-controller-2' from cluster ...
```

Étape 6. Supprimer le noeud ayant échoué de la base de données MongoDB. Pour ce faire, vous devez trouver le noeud Mongo actif. Utilisez **netstat** pour trouver l'adresse IP de l'hôte.

```
[root@pod1-controller-0 ~]# sudo netstat -tulnp | grep 27017  
tcp        0      0 11.120.0.10:27017    0.0.0.0:*           LISTEN  
219577/mongod
```

Étape 7. Connectez-vous au noeud et vérifiez s'il s'agit du maître à l'aide de l'adresse IP et du numéro de port de la commande précédente.

```
[heat-admin@pod1-controller-0 ~]$ echo "db.isMaster()" | mongo --host 11.120.0.10:27017  
MongoDB shell version: 2.6.11  
connecting to: 11.120.0.10:27017/test  
{  
  "setName" : "tripleo",  
  "setVersion" : 9,  
  "ismaster" : true,  
  "secondary" : false,  
  "hosts" : [  
    "11.120.0.10:27017",  
    "11.120.0.12:27017",  
    "11.120.0.11:27017"  
  ],  
  "primary" : "11.120.0.10:27017",  
  "me" : "11.120.0.10:27017",  
  "electionId" : ObjectId("5a0d2661218cb0238b582fb1"),  
  "maxBsonObjectSize" : 16777216,  
  "maxMessageSizeBytes" : 48000000,  
  "maxWriteBatchSize" : 1000,  
  "localTime" : ISODate("2017-11-16T18:36:34.473Z"),  
  "maxWireVersion" : 2,  
  "minWireVersion" : 0,  
  "ok" : 1  
}
```

Si le noeud n'est pas le maître, connectez-vous à l'autre contrôleur actif et exécutez la même étape.

Étape 8. À partir du maître, listez les noeuds disponibles à l'aide de la commande **rs.status()**. Recherchez l'ancien noeud ou le noeud qui ne répond pas et identifiez le nom du noeud mongo.

```
[root@pod1-controller-0 ~]# mongo --host 11.120.0.10  
MongoDB shell version: 2.6.11  
connecting to: 11.120.0.10:27017/test  
<snip>  
tripleo:PRIMARY> rs.status()  
{  
  "set" : "tripleo",
```

```

"date" : ISODate("2017-11-14T13:27:14Z"),
"myState" : 1,
"members" : [
  {
    "_id" : 0,
    "name" : "11.120.0.10:27017",
    "health" : 1,
    "state" : 1,
    "stateStr" : "PRIMARY",
    "uptime" : 418347,
    "optime" : Timestamp(1510666033, 1),
    "optimeDate" : ISODate("2017-11-14T13:27:13Z"),
    "electionTime" : Timestamp(1510247693, 1),
    "electionDate" : ISODate("2017-11-09T17:14:53Z"),
    "self" : true
  },
  {
    "_id" : 2,
    "name" : "11.120.0.12:27017",
    "health" : 1,
    "state" : 2,
    "stateStr" : "SECONDARY",
    "uptime" : 418347,
    "optime" : Timestamp(1510666033, 1),
    "optimeDate" : ISODate("2017-11-14T13:27:13Z"),
    "lastHeartbeat" : ISODate("2017-11-14T13:27:13Z"),
    "lastHeartbeatRecv" : ISODate("2017-11-14T13:27:13Z"),
    "pingMs" : 0,
    "syncingTo" : "11.120.0.10:27017"
  },
  {
    "_id" : 3,
    "name" : "11.120.0.11:27017",
    "health" : 0,
    "state" : 8,
    "stateStr" : "(not reachable/healthy)",
    "uptime" : 0,
    "optime" : Timestamp(1510610580, 1),
    "optimeDate" : ISODate("2017-11-13T22:03:00Z"),
    "lastHeartbeat" : ISODate("2017-11-14T13:27:10Z"),
    "lastHeartbeatRecv" : ISODate("2017-11-13T22:03:01Z"),
    "pingMs" : 0,
    "syncingTo" : "11.120.0.10:27017"
  }
],
"ok" : 1
}

```

Étape 9. À partir du maître, supprimez le noeud ayant échoué à l'aide de la commande **rs.remove**. Certaines erreurs apparaissent lorsque vous exécutez cette commande, mais vérifiez à nouveau l'état pour constater que le noeud a été supprimé :

```

[root@pod1-controller-0 ~]$ mongo --host 11.120.0.10
<snip>
tripleo:PRIMARY> rs.remove('11.120.0.12:27017')
2017-11-16T18:41:04.999+0000 DBClientCursor::init call() failed
2017-11-16T18:41:05.000+0000 Error: error doing query: failed at src/mongo/shell/query.js:81
2017-11-16T18:41:05.001+0000 trying reconnect to 11.120.0.10:27017 (11.120.0.10) failed
2017-11-16T18:41:05.003+0000 reconnect 11.120.0.10:27017 (11.120.0.10) ok

tripleo:PRIMARY> rs.status()
{
  "set" : "tripleo",

```

```

    "date" : ISODate("2017-11-16T18:44:11Z"),
    "myState" : 1,
    "members" : [
      {
        "_id" : 3,
        "name" : "11.120.0.11:27017",
        "health" : 1,
        "state" : 2,
        "stateStr" : "SECONDARY",
        "uptime" : 187,
        "optime" : Timestamp(1510857848, 3),
        "optimeDate" : ISODate("2017-11-16T18:44:08Z"),
        "lastHeartbeat" : ISODate("2017-11-16T18:44:11Z"),
        "lastHeartbeatRecv" : ISODate("2017-11-16T18:44:09Z"),
        "pingMs" : 0,
        "syncingTo" : "11.120.0.10:27017"
      },
      {
        "_id" : 4,
        "name" : "11.120.0.10:27017",
        "health" : 1,
        "state" : 1,
        "stateStr" : "PRIMARY",
        "uptime" : 89820,
        "optime" : Timestamp(1510857848, 3),
        "optimeDate" : ISODate("2017-11-16T18:44:08Z"),
        "electionTime" : Timestamp(1510811232, 1),
        "electionDate" : ISODate("2017-11-16T05:47:12Z"),
        "self" : true
      }
    ],
    "ok" : 1
  }
}
tripleo:PRIMARY> exit
bye

```

Étape 10. Exécutez cette commande afin de mettre à jour la liste des noeuds de contrôleur actifs. Inclure le nouveau noeud contrôleur dans cette liste.

```
[root@pod1-controller-0 ~]# sudo pcs resource update galera wsrep_cluster_address=gcomm://pod1-controller-0,pod1-controller-1,pod1-controller-2
```

Étape 11. Copiez ces fichiers à partir d'un contrôleur qui existe déjà sur le nouveau contrôleur :

/etc/sysconfig/clustercheck

/root/.my.cnf

On existing controller:

```
[root@pod1-controller-0 ~]# scp /etc/sysconfig/clustercheck stack@192.200.0.1:/tmp/.
[root@pod1-controller-0 ~]# scp /root/.my.cnf stack@192.200.0.1:/tmp/my.cnf
```

On new controller:

```
[root@pod1-controller-3 ~]# cd /etc/sysconfig
```

```
[root@pod1-controller-3 sysconfig]# scp stack@192.200.0.1:/tmp/clustercheck .
```

```
[root@pod1-controller-3 sysconfig]# cd /root
```



```
[root@pod1-controller-3 ~]# scp stack@192.200.0.1:/tmp/my.cnf .my.cnf
```

Étape 12. Exécutez la commande **cluster node add** à partir de l'un des contrôleurs qui existe déjà.

```
[root@pod1-controller-1 ~]# sudo pcs cluster node add pod1-controller-3
```

```
Disabling SBD service...
```

```
pod1-controller-3: sbd disabled
```

```
pod1-controller-0: Corosync updated
```

```
pod1-controller-1: Corosync updated
```

```
Setting up corosync...
```

```
pod1-controller-3: Succeeded
```

```
Synchronizing pcsd certificates on nodes pod1-controller-3...
```

```
pod1-controller-3: Success
```

```
Restarting pcsd on the nodes in order to reload the certificates...
```

```
pod1-controller-3: Success
```

Étape 13. Connectez-vous à chaque contrôleur et affichez le fichier

/etc/corosync/corosync.conf. Assurez-vous que le nouveau contrôleur est répertorié et que le **noeud** attribué à ce contrôleur est le numéro suivant dans la séquence qui n'a pas été précédemment utilisé. Assurez-vous que cette modification est effectuée sur les 3 contrôleurs.

```
[root@pod1-controller-1 ~]# cat /etc/corosync/corosync.conf
```

```
totem {
    version: 2
    secauth: off
    cluster_name: tripleo_cluster
    transport: udpu
    token: 10000
}
nodelist {
    node {
        ring0_addr: pod1-controller-0
        nodeid: 5
    }
    node {
        ring0_addr: pod1-controller-1
        nodeid: 7
    }
    node {
        ring0_addr: pod1-controller-3
        nodeid: 6
    }
}
quorum {
    provider: corosync_votequorum
}
logging {
    to_logfile: yes
    logfile: /var/log/cluster/corosync.log
    to_syslog: yes
}
```

Par exemple **/etc/corosync/corosync.conf** après modification :

```
totem {
version: 2
secauth: off
```

```

cluster_name: tripleo_cluster
transport: udpu
token: 10000
}
nodelist {
  node {
    ring0_addr: pod1-controller-0
    nodeid: 5
  }
  node {
    ring0_addr: pod1-controller-1
    nodeid: 7
  }
  node {
    ring0_addr: pod1-controller-3
    nodeid: 9
  }
}
quorum {
  provider: corosync_votequorum
}
logging {
  to_logfile: yes
  logfile: /var/log/cluster/corosync.log
  to_syslog: yes
}

```

Étape 14. Redémarrez **corosync** sur les contrôleurs actifs. Ne démarrez pas **corosync** sur le nouveau contrôleur.

```

[root@pod1-controller-0 ~]# sudo pcs cluster reload corosync
[root@pod1-controller-1 ~]# sudo pcs cluster reload corosync

```

Étape 15. Démarrez le nouveau noeud contrôleur à partir de l'un des contrôleurs actifs.

```

[root@pod1-controller-1 ~]# sudo pcs cluster start pod1-controller-3

```

Étape 16. Redémarrez Galera à partir de l'un des contrôleurs actifs.

```

[root@pod1-controller-1 ~]# sudo pcs cluster start pod1-controller-3

```

```

pod1-controller-0: Starting Cluster...

```

```

[root@pod1-controller-1 ~]# sudo pcs resource cleanup galera
Cleaning up galera:0 on pod1-controller-0, removing fail-count-galera
Cleaning up galera:0 on pod1-controller-1, removing fail-count-galera
Cleaning up galera:0 on pod1-controller-3, removing fail-count-galera
* The configuration prevents the cluster from stopping or starting 'galera-master' (unmanaged)

```

```

Waiting for 3 replies from the CRMD... OK

```

```

[root@pod1-controller-1 ~]#
[root@pod1-controller-1 ~]# sudo pcs resource manage galera

```

Étape 17. Le cluster est en mode maintenance. Désactivez le mode de maintenance afin d'obtenir le démarrage des services.

```

[root@pod1-controller-2 ~]# sudo pcs property set maintenance-mode=false --wait

```

Étape 18. Vérifiez l'état des PC pour Galera jusqu'à ce que les 3 contrôleurs soient répertoriés en tant que maîtres dans Galera.

Note: Pour les configurations de grande taille, la synchronisation des bases de données peut prendre un certain temps.

```
[root@pod1-controller-1 ~]# sudo pcs status | grep galera -A1
```

```
Master/Slave Set: galera-master [galera]
```

```
Masters: [ pod1-controller-0 pod1-controller-1 pod1-controller-3 ]
```

Étape 19. Passez en mode maintenance.

```
[root@pod1-controller-1~]# sudo pcs property set maintenance-mode=true --wait
```

```
[root@pod1-controller-1 ~]# pcs cluster status
```

```
Cluster Status:
```

```
Stack: corosync
```

```
Current DC: pod1-controller-0 (version 1.1.15-11.e17_3.4-e174ec8) - partition with quorum
```

```
Last updated: Thu Nov 16 19:17:01 2017
```

```
Last change: Thu Nov 16 19:16:48 2017
```

```
by root via cibadmin on pod1-controller-1
```

```
*** Resource management is DISABLED ***
```

```
The cluster will not attempt to start, stop or recover services
```

```
PCSD Status:
```

```
pod1-controller-3: Online
```

```
pod1-controller-0: Online
```

```
pod1-controller-1: Online
```

Étape 20. Exécutez à nouveau le script de déploiement que vous avez exécuté précédemment. Cette fois-ci, elle devrait réussir.

```
[stack@director ~]$ ./deploy-addController.sh
```

```
START with options: [u'overcloud', u'deploy', u'--templates', u'-r', u'/home/stack/custom-templates/custom-roles.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-templates/environments/puppet-pacemaker.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-templates/environments/network-isolation.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-templates/environments/storage-environment.yaml', u'-e', u'/usr/share/openstack-tripleo-heat-templates/environments/neutron-sriov.yaml', u'-e', u'/home/stack/custom-templates/network.yaml', u'-e', u'/home/stack/custom-templates/ceph.yaml', u'-e', u'/home/stack/custom-templates/compute.yaml', u'-e', u'/home/stack/custom-templates/layout-removeController.yaml', u'--stack', u'newtonoc', u'--debug', u'--log-file', u'overcloudDeploy_11_14_17__13_53_12.log', u'--neutron-flat-networks', u'phys_pcie1_0,phys_pcie1_1,phys_pcie4_0,phys_pcie4_1', u'--neutron-network-vlan-ranges', u'datacentre:101:200', u'--neutron-disable-tunneling', u'--verbose', u'--timeout', u'180']
options: Namespace(access_key='', access_secret='****', access_token='****',
access_token_endpoint='', access_token_type='', aodh_endpoint='', auth_type='',
auth_url='https://192.200.0.2:13000/v2.0', authorization_code='', cacert=None, cert='',
client_id='', client_secret='****', cloud='', consumer_key='', consumer_secret='****', debug=True,
default_domain='default', default_domain_id='', default_domain_name='', deferred_help=False,
discovery_endpoint='', domain_id='', domain_name='', endpoint='', identity_provider='',
identity_provider_url='', insecure=None, inspector_api_version='1', inspector_url=None,
interface='', key='', log_file=u'overcloudDeploy_11_14_17__13_53_12.log', murano_url='',
old_profile=None, openid_scope='', os_alarming_api_version='2',
os_application_catalog_api_version='1', os_baremetal_api_version='1.15', os_beta_command=False,
os_compute_api_version='', os_container_infra_api_version='1',
os_data_processing_api_version='1.1', os_data_processing_url='', os_dns_api_version='2',
os_identity_api_version='', os_image_api_version='1', os_key_manager_api_version='1',
os_metrics_api_version='1', os_network_api_version='', os_object_api_version='',
os_orchestration_api_version='1', os_project_id=None, os_project_name=None,
os_queues_api_version='2', os_tripleoclient_api_version='1', os_volume_api_version='',
os_workflow_api_version='2', passcode='', password='****', profile=None, project_domain_id='',
```

```
project_domain_name='', project_id='', project_name='admin', protocol='', redirect_uri='',
region_name='', roles='', timing=False, token='***', trust_id='', url='', user='',
user_domain_id='', user_domain_name='', user_id='', username='admin', verbose_level=3,
verify=None)
Auth plugin password selected
```

```
Starting new HTTPS connection (1): 192.200.0.2
"POST /v2/action_executions HTTP/1.1" 201 1696
HTTP POST https://192.200.0.2:13989/v2/action_executions 201
Overcloud Endpoint: http://172.25.22.109:5000/v2.0
Overcloud Deployed
clean_up DeployOvercloud:
END return value: 0
```

```
real    54m17.197s
user    0m3.421s
sys     0m0.670s
```

Vérifier les services Overcloud dans le contrôleur

Assurez-vous que tous les services gérés s'exécutent correctement sur les noeuds de contrôleur.

```
[heat-admin@pod1-controller-2 ~]$ sudo pcs status
```

Finaliser les routeurs d'agents de couche 3

Vérifiez les routeurs afin de vous assurer que les agents de couche 3 sont correctement hébergés. Assurez-vous de trouver le fichier de nuage lorsque vous effectuez cette vérification.

Étape 1. Recherchez le nom du routeur.

```
[stack@director~]$ source corerc
[stack@director ~]$ neutron router-list
```

```
+-----+-----+-----+
-----+-----+-----+
| id | name | distributed | ha |
external_gateway_info |
+-----+-----+-----+
-----+-----+-----+
| d814dc9d-2b2f-496f-8c25-24911e464d02 | main | {"network_id": "18c4250c-e402-428c-87d6-
a955157d50b5", | False | True |
```

Dans cet exemple, le nom du routeur est main.

Étape 2. Répertoriez tous les agents L3 afin de trouver l'UUID du noeud ayant échoué et du nouveau noeud.

```
[stack@director ~]$ neutron agent-list | grep "neutron-l3-agent"

| 70242f5c-43ab-4355-abd6-9277f92e4ce6 | L3 agent | pod1-controller-0.localdomain |
nova | :- ) | True | neutron-l3-agent |
| 8d2ffbc9-b6ff-42cd-b5b8-da31d8da8a40 | L3 agent | pod1-controller-2.localdomain |
nova | xxx | True | neutron-l3-agent |
```

```

| a410a491-e271-4938-8a43-458084ffe15d | L3 agent          | pod1-controller-3.localdomain |
nova          | :-) | True          | neutron-l3-agent          |
| cb4bc1ad-ac50-42e9-ae69-8a256d375136 | L3 agent          | pod1-controller-1.localdomain |
nova          | :-) | True          | neutron-l3-agent          |

```

Étape 3. Dans cet exemple, l'agent L3 qui correspond à **pod1-controller-2.localdomain** doit être supprimé du routeur et celui qui correspond à **pod1-controller-3.localdomain** doit être ajouté au routeur.

```
[stack@director ~]$ neutron l3-agent-router-remove 8d2ffbc-b6ff-42cd-b5b8-da31d8da8a40 main
```

Removed router main from L3 agent

```
[stack@director ~]$ neutron l3-agent-router-add a410a491-e271-4938-8a43-458084ffe15d main
```

Added router main to L3 agent

Étape 4. Vérifiez la liste mise à jour des agents L3.

```
[stack@director ~]$ neutron l3-agent-list-hosting-router main
```

```

+-----+-----+-----+-----+
----+-----+
| id          | host          | admin_state_up |
alive | ha_state |
+-----+-----+-----+-----+
----+-----+
| 70242f5c-43ab-4355-abd6-9277f92e4ce6 | pod1-controller-0.localdomain | True          | :-)
| standby |
| a410a491-e271-4938-8a43-458084ffe15d | pod1-controller-3.localdomain | True          | :-)
| standby |
| cb4bc1ad-ac50-42e9-ae69-8a256d375136 | pod1-controller-1.localdomain | True          | :-)
| active |
+-----+-----+-----+-----+
----+-----+

```

Étape 5. Répertoriez tous les services qui s'exécutent à partir du noeud contrôleur supprimé et supprimez-les.

```
[stack@director ~]$ neutron agent-list | grep controller-2
```

```

| 877314c2-3c8d-4666-a6ec-69513e83042d | Metadata agent    | pod1-controller-2.localdomain | |
|          | xxx | True          | neutron-metadata-agent    |
| 8d2ffbc-b6ff-42cd-b5b8-da31d8da8a40 | L3 agent          | pod1-controller-2.localdomain |
nova          | xxx | True          | neutron-l3-agent          |
| 911c43a5-df3a-49ec-99ed-1d722821ec20 | DHCP agent        | pod1-controller-2.localdomain |
nova          | xxx | True          | neutron-dhcp-agent        |
| a58a3dd3-4cdc-48d4-ab34-612a6cd72768 | Open vSwitch agent | pod1-controller-2.localdomain |
|          | xxx | True          | neutron-openvswitch-agent |

```

```
[stack@director ~]$ neutron agent-delete 877314c2-3c8d-4666-a6ec-69513e83042d
```

Deleted agent(s): 877314c2-3c8d-4666-a6ec-69513e83042d

```
[stack@director ~]$ neutron agent-delete 8d2ffbc-b6ff-42cd-b5b8-da31d8da8a40
```

Deleted agent(s): 8d2ffbc-b6ff-42cd-b5b8-da31d8da8a40

```
[stack@director ~]$ neutron agent-delete 911c43a5-df3a-49ec-99ed-1d722821ec20
```

Deleted agent(s): 911c43a5-df3a-49ec-99ed-1d722821ec20

```
[stack@director ~]$ neutron agent-delete a58a3dd3-4cdc-48d4-ab34-612a6cd72768
```

Deleted agent(s): a58a3dd3-4cdc-48d4-ab34-612a6cd72768

```
[stack@director ~]$ neutron agent-list | grep controller-2
```

```
[stack@director ~]$
```

Finaliser les services de calcul

Étape 1. Vérifiez les éléments nova **service-list** laissés du noeud supprimé et supprimez-les.

```
[stack@director ~]$ nova service-list | grep controller-2
```

```
| 615 | nova-consoleauth | pod1-controller-2.localdomain | internal | enabled | down
| 2017-11-16T16:08:14.000000 | - |
| 618 | nova-scheduler | pod1-controller-2.localdomain | internal | enabled | down
| 2017-11-16T16:08:13.000000 | - |
| 621 | nova-conductor | pod1-controller-2.localdomain | internal | enabled | down
| 2017-11-16T16:08:14.000000 | - |
```

```
[stack@director ~]$ nova service-delete 615
```

```
[stack@director ~]$ nova service-delete 618
```

```
[stack@director ~]$ nova service-delete 621
```

```
stack@director ~]$ nova service-list | grep controller-2
```

Étape 2. Assurez-vous que le processus **de console** s'exécute sur tous les contrôleurs ou redémarrez-le à l'aide de cette commande : **pcs resource restart openstack-nova-consoleauth** :

```
[stack@director ~]$ nova service-list | grep consoleauth
```

```
| 601 | nova-consoleauth | pod1-controller-0.localdomain | internal | enabled | up
| 2017-11-16T21:00:10.000000 | - |
| 608 | nova-consoleauth | pod1-controller-1.localdomain | internal | enabled | up
| 2017-11-16T21:00:13.000000 | - |
| 622 | nova-consoleauth | pod1-controller-3.localdomain | internal | enabled | up
| 2017-11-16T21:00:13.000000 | - |
```

Redémarrer la clôture sur les noeuds du contrôleur

Étape 1. Vérifiez tous les contrôleurs pour la route IP vers le nuage de sous-réseau 192.0.0.0/8

```
[root@pod1-controller-3 ~]# ip route
```

```
default via 172.25.22.1 dev vlan101
```

```
11.117.0.0/24 dev vlan17 proto kernel scope link src 11.117.0.12
```

```
11.118.0.0/24 dev vlan18 proto kernel scope link src 11.118.0.12
```

```
11.119.0.0/24 dev vlan19 proto kernel scope link src 11.119.0.12
```

```
11.120.0.0/24 dev vlan20 proto kernel scope link src 11.120.0.12
```

```
169.254.169.254 via 192.200.0.1 dev eno1
```

```
172.25.22.0/24 dev vlan101 proto kernel scope link src 172.25.22.102
```

```
192.0.0.0/8 dev eno1 proto kernel scope link src 192.200.0.103
```

Étape 2. Vérifiez la configuration actuelle du stonith. Supprimez toute référence à l'ancien noeud contrôleur.

```
[root@pod1-controller-3 ~]# sudo pcs stonith show --full
```

```
Resource: my-ipmilan-for-controller-6 (class=stonith type=fence_ipmilan)
```

```
Attributes: pcmk_host_list=pod1-controller-1 ipaddr=192.100.0.1 login=admin
```

```
passwd=Cscoc@123Starent lanplus=1
```

```
Operations: monitor interval=60s (my-ipmilan-for-controller-6-monitor-interval-60s)
```

```
Resource: my-ipmilan-for-controller-4 (class=stonith type=fence_ipmilan)
```

```
Attributes: pcmk_host_list=pod1-controller-0 ipaddr=192.100.0.14 login=admin
```

```
passwd=Csco@123Starent lanplus=1
Operations: monitor interval=60s (my-ipmilan-for-controller-4-monitor-interval-60s)
Resource: my-ipmilan-for-controller-7 (class=stonith type=fence_ipmilan)
Attributes: pcmk_host_list=pod1-controller-2 ipaddr=192.100.0.15 login=admin
passwd=Csco@123Starent lanplus=1
Operations: monitor interval=60s (my-ipmilan-for-controller-7-monitor-interval-60s)
```

```
[root@pod1-controller-3 ~]# pcs stonith delete my-ipmilan-for-controller-7
Attempting to stop: my-ipmilan-for-controller-7...Stopped
```

Étape 3. Ajoutez la configuration de **stonith** pour le nouveau contrôleur.

```
[root@pod1-controller-3 ~]# sudo pcs stonith create my-ipmilan-for-controller-8 fence_ipmilan
pcmk_host_list=pod1-controller-3 ipaddr=<CIMC_IP> login=admin passwd=<PASSWORD> lanplus=1 op
monitor interval=60s
```

Étape 4. Redémarrez la clôture à partir de n'importe quel contrôleur et vérifiez l'état.

```
[root@pod1-controller-1 ~]# sudo pcs property set stonith-enabled=true
[root@pod1-controller-3 ~]# pcs status
```

<snip>

```
my-ipmilan-for-controller-1 (stonith:fence_ipmilan): Started pod1-controller-3
my-ipmilan-for-controller-0 (stonith:fence_ipmilan): Started pod1-controller-3
my-ipmilan-for-controller-3 (stonith:fence_ipmilan): Started pod1-controller-3
```