PCRF Replacement of Controller Server UCS C240 M4

Contents

Introduction

This document describes the steps required to replace a faulty controller server in an Ultra-M setup that hosts CPS Virtual Network Functions (VNFs).

Prerequisites

Backup

In case of recovery, Cisco recommends to take a backup of the OSPD database (DB) with the use of these steps:

```
[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
```

Preliminary Status Check

It is important to check the current status of the OpenStack environment and services and ensure that it is healthy before you proceed with the replacement procedure. It can help avoid complications at the time of the Controller replacement process.

Step 1. Check the status of OpenStack and the node list:

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

Step 2. Check the Pacemaker status on controllers.

Login to one of the active controllers and check the pacemaker status. All services should be running on the available controllers and stopped on the failed controller.

```
[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
```

In this example, Controller-2 is offline. It will, therefore, be replaced. Controller-0 and Controller-1 are operational and are running the cluster services.

Step 3. Check MariaDB status in the active controllers.

```
[stack@director] nova list | grep control
| 4361358a-922f-49b5-89d4-247a50722f6d | pod1-controller-0 | ACTIVE | - | Running |
ctlplane=192.200.0.102 |
| d0f57f27-93a8-414f-b4d8-957de0d785fc | pod1-controller-1 | ACTIVE | - | Running |
ctlplane=192.200.0.110 |

[stack@director ~]$ for i in 192.200.0.102 192.200.0.110 ; do echo "*** $i ***" ; ssh heat-
admin@$i "sudo mysql --exec=\"SHOW STATUS LIKE 'wsrep_local_state_comment'\" ; sudo mysql --
exec=\"SHOW STATUS LIKE 'wsrep_cluster_size'\""; done
*** 192.200.0.152 ***
Variable_name     Value
wsrep_local_state_comment  Synced
Variable_name     Value
wsrep_cluster_size         2
*** 192.200.0.154 ***
Variable_name     Value
wsrep_local_state_comment  Synced
Variable_name     Value
wsrep_cluster_size         2
```

Verify that these lines are present for each active controller:

wsrep_local_state_comment: Synced

wsrep_cluster_size: 2

Step 4. Check Rabbitmq status in the active controllers. The failed controller should not appear in the list of the nodes that run.

```
[heat-admin@pod1-controller-0 ~] sudo rabbitmqctl cluster_status
Cluster status of node 'rabbit@pod1-controller-0' ...
[{nodes,[{disc,['rabbit@pod1-controller-0','rabbit@pod1-controller-1',
               'rabbit@pod1-controller-2']}]},
{running_nodes,['rabbit@pod1-controller-1',
                'rabbit@pod1-controller-0']},
 {cluster_name,<<"rabbit@pod1-controller-2.localdomain">>},
 {partitions,[]},
 {alarms,[{'rabbit@pod1-controller-1',[]},
         {'rabbit@pod1-controller-0',[]}]}]
```

```
[heat-admin@pod1-controller-1 ~] sudo rabbitmqctl cluster_status
Cluster status of node 'rabbit@pod1-controller-1' ...
[{nodes,[{disc,['rabbit@pod1-controller-0','rabbit@pod1-controller-1',
               'rabbit@pod1-controller-2']}]},
 {running_nodes,['rabbit@pod1-controller-0',
                 'rabbit@pod1-controller-1']},
 {cluster_name,<<"rabbit@pod1-controller-2.localdomain">>},
 {partitions,[]},
 {alarms,[{'rabbit@pod1-controller-0',[]},
         {'rabbit@pod1-controller-1',[]}]}]
```

Step 5. Check if all the undercloud services are in loaded, active and running status from the OSP-D node.

```
[heat-admin@pod1-controller-0 ~] sudo rabbitmqctl cluster_status
Cluster status of node 'rabbit@pod1-controller-0' ...
[{nodes,[{disc,['rabbit@pod1-controller-0','rabbit@pod1-controller-1',
               'rabbit@pod1-controller-2']}]},
```

```
{running_nodes,['rabbit@pod1-controller-1',
                'rabbit@pod1-controller-0']},
 {cluster_name,<<"rabbit@pod1-controller-2.localdomain">>},
 {partitions,[]},
 {alarms,[{'rabbit@pod1-controller-1',[]},
          {'rabbit@pod1-controller-0',[]}]}]


[heat-admin@pod1-controller-1 ~] sudo rabbitmqctl cluster_status
Cluster status of node 'rabbit@pod1-controller-1' ...
[{nodes,[{disc,['rabbit@pod1-controller-0','rabbit@pod1-controller-1',
                'rabbit@pod1-controller-2']}]},
 {running_nodes,['rabbit@pod1-controller-0',
                 'rabbit@pod1-controller-1']},
 {cluster_name,<<"rabbit@pod1-controller-2.localdomain">>},
 {partitions,[]},
 {alarms,[{'rabbit@pod1-controller-0',[]},
          {'rabbit@pod1-controller-1',[]}]}]
```

Disable Fencing in the Controller Cluster

```
[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.el7_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
```

Install the New Controller Node

Step 1. The steps in order to install a new UCS C240 M4 server and the initial setup steps can be referred from Cisco UCS C240 M4 Server Installation and Service Guide

Step 2. Log in to the server with the use of the CIMC IP.

Step 3.Perform BIOS upgrade if the firmware is not as per the recommended version used previously. Steps for BIOS upgrade are given here:

Cisco UCS C-Series Rack-Mount Server BIOS Upgrade Guide

Step 4.Verify the status of Physical Drives. It must be **Unconfigured Good**. Navigate to **Storage > Cisco 12G SAS Modular Raid Controller (SLOT-HBA) > Physical Drive Info.**

Step 5. In order to create a virtual drive from the physical drives with RAID Level 1: navigate

to **Storage > Cisco 12G SAS Modular Raid Controller (SLOT-HBA) > Controller Info > Create Virtual Drive from Unused Physical Drives**, as shown in the image.

- Select the VD and configure **Set as Boot Drive:**

Step 6. In order to enable IPMI over LAN, navigate to **Admin > Communication Services > Communication Services**.

Step 7. In order to disable hyperthreading, navigate to **Compute > BIOS > Configure BIOS > Advanced > Processor Configuration**, as shown in the image.

> **Note**: The image is shown here and the configuration steps mentioned in this section are with reference to the firmware version 3.0(3e) and there might be slight variations if you work on other versions.

Controller Node Replacement in Overcloud

This section covers the steps that are required in order to replace the faulty controller with the new one in the overcloud. For this, the **deploy.sh** script that was used to bring up the stack would be re-used. At the time of the deployment, in the ControllerNodesPostDeployment phase, the update would fail due to some limitations in the Puppet modules. Manual intervention is required before you restart the deployment script.

Prepare to Remove Failed Controller Node

Step 1. Identify the index of the failed controller. The index is the numeric suffix on the controller name in the OpenStack server list output. In this example, the index is 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 |
```

Step 2. Create a Yaml file **~templates/remove-controller.yaml** that would define the node to delete. Use the index found in the previous step for the entry in the resource list.

```
[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 |
```

Step 3. Make a copy of the deploy script that is used in order to install the overcloud and insert a line in order to include the **remove-controller.yaml** file created previously.

```
[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 |
```

Step 4. Identify the ID of the controller to be replaced, with the use of the commands mentioned here and move it to the maintenance mode.

```
[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        |
```

Step 5. In order to ensure that the DB runs at the time of the replacement procedure, remove Galera from pacemaker control and run this command on one of the active controllers.

```
[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.el7_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>
```

Prepare to Add New Controller Node

Step 1. Create a **controllerRMA.json** file with only the new Controller details. Ensure that the index number on the new Controller has not been used before. Typically, increment to the next highest controller number.

Example: Highest prior was Controller-2, so create Controller-3.

> **Note**: Be mindful of the json format.

```
[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.el7_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>
```

Step 2. Import the new node with the use of the json file created in the previous step.

```
[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
```

Step 3. Set the node to manage state.

```
[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 |
```

Step 4. Run 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       |
```

Step 5. Mark the available node with the new controller properties. Ensure to use the controller ID designated for the new controller, as used in the **controllerRMA.json** file.

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

Step 6. In the deploy script, there is a custom-template called **layout.yaml** which among other things, specifies what IP addresses are assigned to the controllers for the various interfaces. On a new stack, there are 3 addresses defined for Controller-0, Controller-1, and Controller-2. When you add a new controller, ensure that you add a next IP address in sequence for each subnet.

```
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
```

Step 7. Now run the **deploy-removecontroller.sh** that was previously created, in order to remove the old node and add the new node.

> **Note**: This step is expected to fail in ControllerNodesDeployment_Step1. At that point, manual intervention is required.

```
[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
```

The progress/status of the deployment can be monitored with these commands:


```
[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
```


Manual Intervention

Step 1. On the OSP-D server, run the OpenStack server list command in order to list the available controllers. The newly added controller should appear in the list.

```
[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 |
```

Step 2. Connect to one of the active controllers (not the newly added controller) and look at the file **/etc/corosync/corosycn.conf**. Find the **nodelist** which assigns a **nodeid** to each controller. Find the entry for the failed node and note its **nodeid:**

```
[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
   }
}
```

Step 3. Log in to each of the active controllers. Remove the failed node and restart the service. In this case, remove **pod1-controller-2**. Do not perform this action on the newly added controller.

```
[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
   }
}
```

Step 4. Run this command from one of the active controllers in order to delete the failed node from the cluster.

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

Step 5. Run this command from one of the active controllers in order to delete the failed node from the **rabbitmq** cluster.

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

Step 6. Delete the failed node from the MongoDB. In order to do this, you need to find the active Mongo node. Use **netstat** to find the IP address of the host.

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

Step 7. Login to the node and check in order to see if it is the master with the use of the IP address and port number from the previous command.

```
[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
}
```

If the node is not the master, log in to the other active controller and perform the same step.

Step 8. From the master, list the available nodes with the use of the **rs.status()** command. Find the old/unresponsive node and identify the mongo node name.

```
[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
}
```

Step 9. From the master, delete the failed node with the use of the **rs.remove** command. Some errors are seen when you run this command, but check the status once more to find that the node has been removed:

```
[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
```

Step 10. Run this command in order to update the list of active controller nodes. Include the new controller node in this list.

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

Step 11. Copy these files from a controller that already exists to the new controller:

/etc/sysconfig/clustercheck

/root/.my.cnf

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

Step 12. Run the **cluster node add** command from one of the controllers that already exists.

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

Step 13. Log in to each controller and view the file **/etc/corosync/corosync.conf**. Ensure that the

new controller is listed and that the **node id**assigned to that controller is the next number in the sequence that has not been previously used. Ensure that this change is done on all 3 controllers.

```
[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
}
```

For example **/etc/corosync/corosync.conf** after 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
}
```

Step 14. Restart **corosync** on the active controllers. Do not start **corosync** on the new controller.

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

Step 15. Start the new controller node from one of the acting controllers.

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

Step 16. Restart Galera from one of the acting controllers.

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

Step 17. The cluster is in maintenance mode. Disable the maintenance mode in order to get the services to start.

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

Step 18. Check the PCs status for Galera until all 3 controllers are listed as masters in Galera.

> **Note**: For large setups, it can take some time to sync DBs.

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

Step 19. Switch the cluster to maintenance mode.

```
[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.el7_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
```

Step 20. Re-run the deploy script that you ran previously. This time it should succeed.

```
[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
```

Verify Overcloud Services in the Controller

Ensure that all managed services run properly on the controller nodes.

```
[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
```

Finalize the L3 Agent Routers

Check the routers in order to ensure that L3 agents are properly hosted. Ensure to source the overcloudrc file when you perform this check.

Step 1. Find the router name.

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

+--------------------------------------+------+-----------------------------------------------
-----------------+-------------+------+

| id                                   | name |
external_gateway_info                                          | distributed | ha   |

+--------------------------------------+------+-----------------------------------------------
-----------------+-------------+------+

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

In this example, the name of the router is main.

Step 2. List all L3 agents in order to find UUID of failed node and the new node.

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

+--------------------------------------+------+-----------------------------------------------
-----------------+-------------+------+

| id                                   | name |
external_gateway_info                                          | distributed | ha   |

+--------------------------------------+------+-----------------------------------------------
-----------------+-------------+------+

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

Step 3. In this example, L3 agent that corresponds to **pod1-controller-2.localdomain** should be removed from the router and the one that corresponds to **pod1-controller-3.localdomain** should be added to the router.

```
[stack@director ~]$ neutron l3-agent-router-remove 8d2ffbcb-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
```

Step 4. Check updated list of L3-agents.

```
[stack@director ~]$ neutron l3-agent-router-remove 8d2ffbcb-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
```

Step 5. List any services that run from the removed controller node and remove them.

```
[stack@director ~]$ neutron l3-agent-router-remove 8d2ffbcb-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
```

Finalize Compute Services

Step 1. Check **nova service-list** items left from the removed node and delete them.

```
[stack@director ~]$ neutron l3-agent-router-remove 8d2ffbcb-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
```

Step 2. Ensure that the **consoleauth** process runs on all the controllers or restart it with the use of this command: **pcs resource restart openstack-nova-consoleauth:**

```
[stack@director ~]$ neutron l3-agent-router-remove 8d2ffbcb-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
```

Restart Fencing on the Controller Nodes

Step 1. Check all controllers for IP route to the undercloud 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
```

Step 2. Check the current **stonith** configuration. Remove any reference to the old controller node.

```
[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=Csco@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
```

Step 3. Add **stonith** configuration for new controller.

```
[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=Csco@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
```

Step 4. Restart fencing from any controller and verify the status.

```
[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=Csco@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
```