

Author:

Raju Ingalagondi, Anand Gagrani

GSE, HPE

C7000 Blades –How to migrate to Hot Stand by Blade /Compute if dedicated OS Host fails.

Contents

[**Objective** 1](#_Toc15976905)

[**Process flow** 2](#_Toc15976906)

[**Prerequisites** 2](#_Toc15976907)

[Scripts and Paths 3](#_Toc15976908)

[Execution / input scripts: 3](#_Toc15976909)

[Action flow 4](#_Toc15976910)

[Results: 9](#_Toc15976911)

[Troubleshooting / Extras: 9](#_Toc15976912)

# **Objective**

To ensure that via HPE Monitor Scripts, If either of the Host ( in 4 Node Cluster of KVM), any of the Dedicated OS Host fails , Hot Stand-by Blade /Compute will take over as part of failover process.

# **Process flow**



# **Prerequisites**

1. 1 x C7000 enclosure with 4 BL460 g9/g10 blades.
2. 1 x Oneview virtual appliance that is monitoring c7000 enclosure and servers.
3. The Oneview profiles are created on 4 blades with boot disk from SAN and 3 nodes (that are being monitored) are installed with RHEL OS and are configured to be part of cluster.
4. 1 x Test Server/VM running Linux (CentOS 7)to excute migration scripts in same subnet as c7000 blades.
5. Yum update the above VM with latest patches and proceed with following steps on same VM:

* RUN **$Install Python 3.6, python-pip**
* RUN **$pip3 install > redfish, hponeview>=4.7.0**
* Download the HPE provided zip file for the scripts and place in dir **~/BSNL\_cluster\_failover\_POC/.**
* Directory structure should look like following:

**[root@appliance-vm BSNL\_cluster\_failover\_POC]# ll**

**total 44**

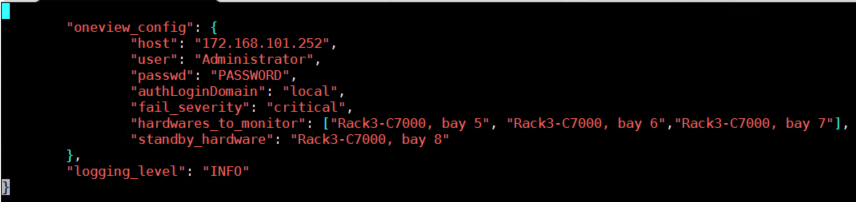
**-rw-r--r--. 1 root root 16107 Jul 19 00:26 failback\_node.py**

**-rw-r--r--. 1 root root 24330 Jul 25 03:04 failover\_node.py**

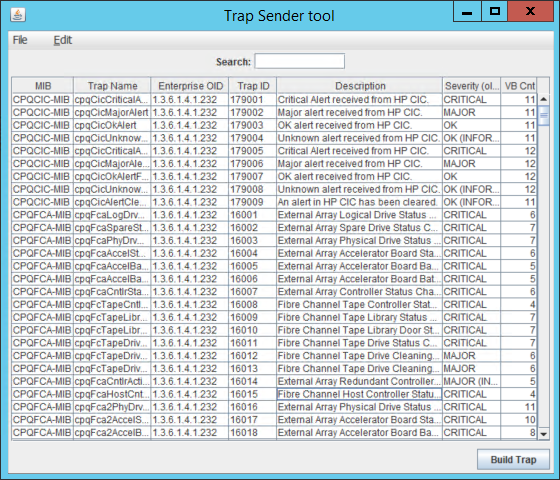
**drwxr-xr-x. 2 root root 60 Jul 23 03:17 inputs**

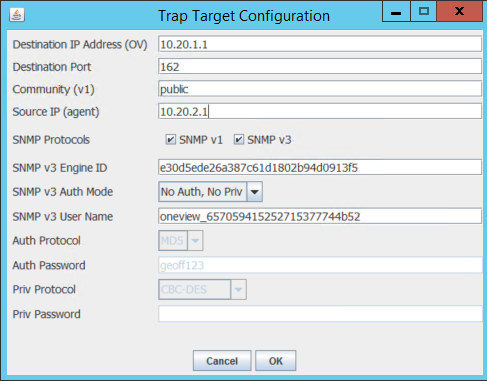
**-rw-r--r--. 1 root root 2472 Jul 23 03:37 README.md**

* Edit ./inpus/failover\_config.json and update it with Oneview credentials and hardware names of nodes to be monitored and the name of standby hardware.



1. 1 x windows VM running the Traptool to generate the alerts on Oneview SCMB bus for simulating critical alerts for scripts to work like realtime.





# Scripts and Paths

Ensure that all the HPE scripts (as part of attachment) , be unzipped and placed in CentOS VM. Following is an example of the paths we used in our lab setup.

## Execution / input scripts:

1. ***~/BSNL\_cluster\_failover\_POC /failover\_node.py***  *-* This is master script that monitors alerts and migrates profile from failed to standby hardware
2. ***~/BSNL\_cluster\_failover\_POC /inputs/failover\_config.json*** *– This is the input file for master script failover\_node.py. It contains oneview credentials and hardware to monitor and standby hardware names.*
3. ***~/ BSNL\_cluster\_failover\_POC /internal/*service\_template** - This file acts as template file to create systemd service.
4. ***~/BSNL\_cluster\_failover\_POC /create\_failover\_service.sh*** – This shell script takes service name as command line parameter and create systemd service*. Later service keeps monitoring for hardware failures.*
5. ***~/BSNL\_cluster\_failover\_POC /logs/*BSNL\_failover\_<OneView IP>.log** - This log file gets generated when you run master scripts *failover\_node.py*

# Action flow

1. Set the desired hardware names to monitor in following file:

**$ vim *~/BSNL\_cluster\_failover\_POC /inputs/failover\_config.json***

**{**

**"oneview\_config": {**

**"host": "<OneView IP> ",**

**"user": "Administrator",**

**"passwd": "PASSWORD",**

**"authLoginDomain": "local",**

**"fail\_severity": "critical",**

**"hardwares\_to\_monitor": ["Rack3-C7000, bay 5", "Rack3-C7000, bay 6","Rack3-C7000, bay 7"],**

**"standby\_hardware": "Rack3-C7000, bay 8"**

**},**

**"logging\_level": "INFO"**

**}**

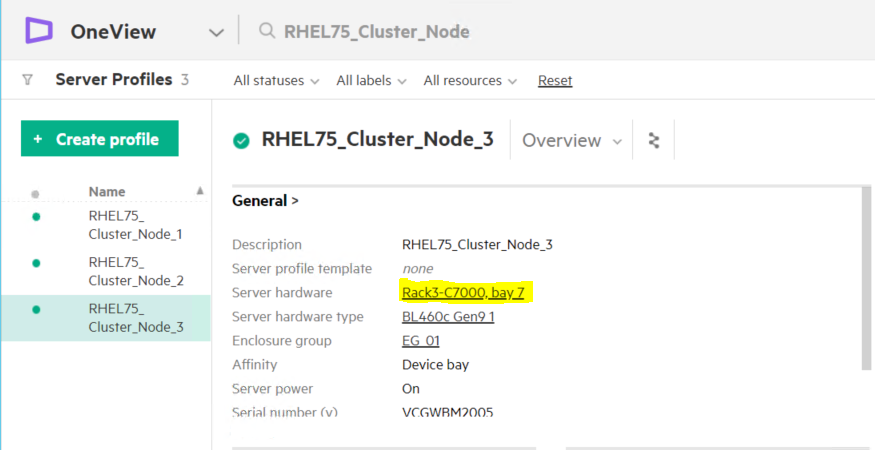
1. Execute the master script **failover\_node.py** from VM:

**$ python3 failover\_node.py**

This command would read the input file “*failover\_config.json*” and would first check the health of the hardware (primary and standby) and then starts watching the server hardwares via OneView SCMB bus for any critical alerts:



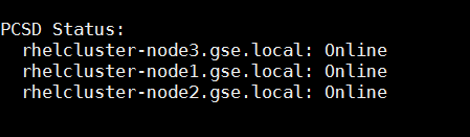
1. Check from Oneview GUI the current state of the hardware (**before failover migration**): The node3 profile is running on bay7 now. Our standby node is on bay8.



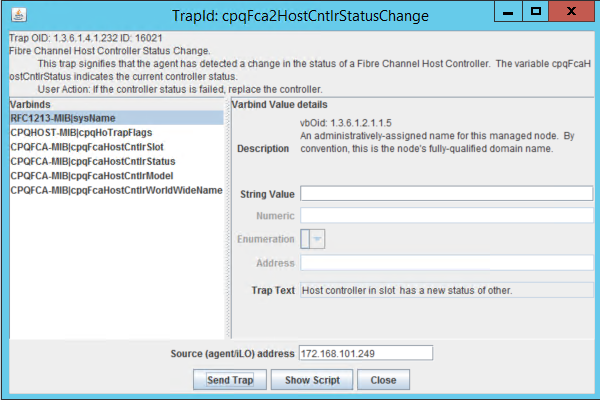
1. Run the following command on the RHEL cluster node to see the health of the cluster(before failure of one node).

It shows all nodes **ONLINE**.

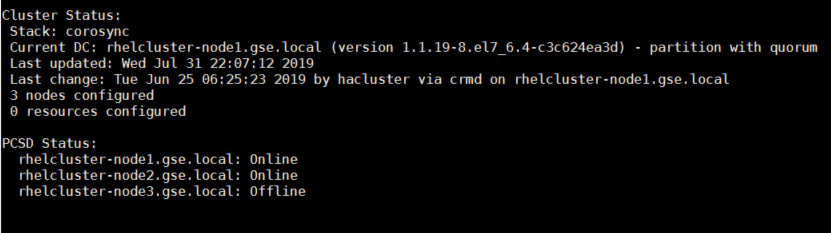
**$ pcs cluster status**



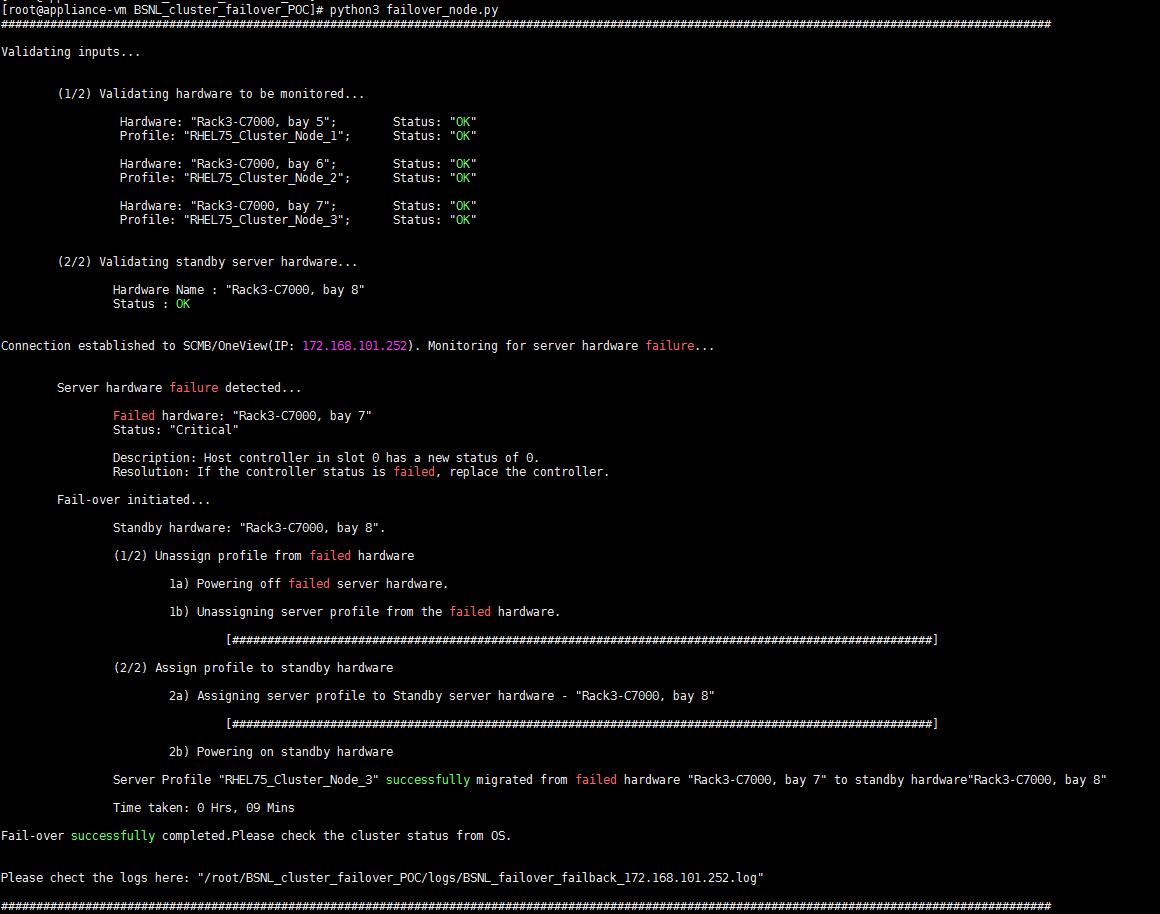
1. Now to **SIMULATE** a real hardware critical alert (eg: Processor failure) - Trigger the SNMP trap on OneView SCMB bus using the **traptool > Send Trap button**. This would cause the master script to proceed with profile migration.



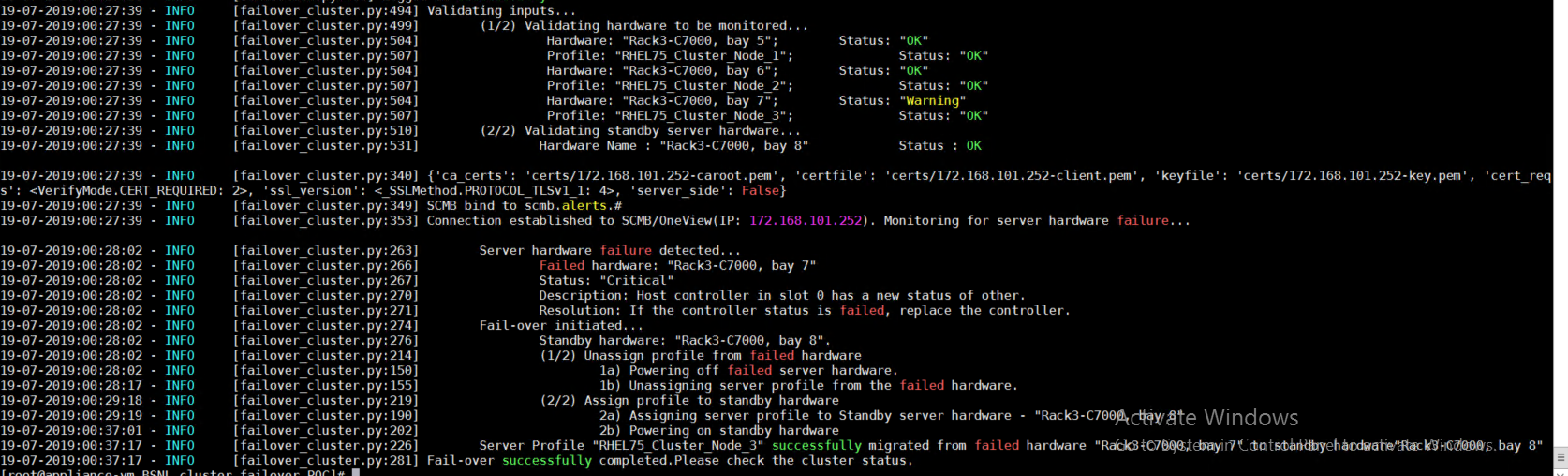
1. Run the following command on the RHEL cluster node to see the health of the cluster(when due to critical alert the node is powered down by master script as part of migration in step shown below ). It shows 2 nodes **ONLINE** and 1 **OFFLINE**:



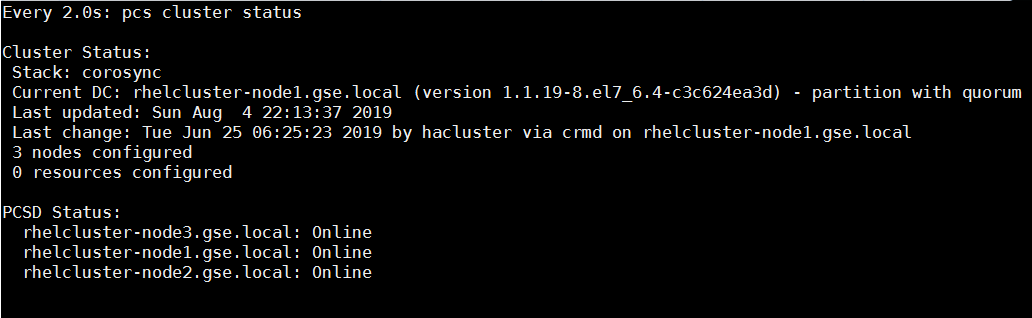
1. As soon as the critical alert is received on the server hardware, the master script would initiate failover process from failed hardware to standby hardware. The script also shows the **migration time taken**.



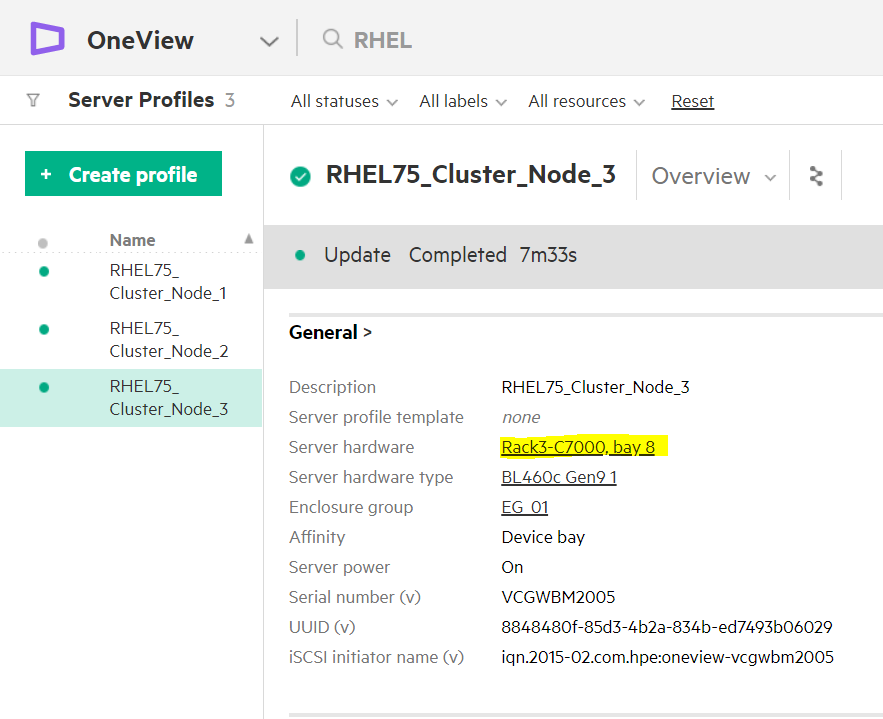
1. Once the master script completes the migration / failover as shown in step above, please check the log file for steps performed and errors.



1. Now run the following command on the RHEL cluster node to check the cluster health. It shows all 3 nodes **ONLINE** again.



1. Check from OneView GUI the profile is migrated to new standby hardware (**after migration is complete**): The node3 profile which was running on bay7 earlier is now running on bay8.



# Results:

Using the master failover script, we were able to migrate host profile (SAN storage as well) from a failed node to a standby node successfully and RHEL cluster rebuilds itself with new standby node.In this case it took approximately 9 minutes for migration.

Once the failover is successfully completed the script still continues to run and keep monitoring for critical alerts.

Now failed server acts as standby server but no failover occurs until it is fixed.

If any active hardware fails failover occurs if standby server is healthy else the error will be logged saying that “Aborted failover as the standby server was unhealthy”

Script can also be run as linux service.

# Troubleshooting / Extras: