

Health Check Procedure

Cloud Execution Environment

OPERATING INSTRUCTIONS

Copyright

© Ericsson 2016. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Scope	1
1.2	Target Groups	1
1.3	Prerequisites	1
2	Overview	2
2.1	Execution of Commands for Several Hosts	2
2.2	Execution of Commands for vCICs	3
2.3	Execution of Commands for Compute Hosts	3
3	Daily Health Check Procedure	4
3.1	Check Alarms And Alarm History	4
3.2	Check Presence of Crash Dumps	4
3.3	Verify Date and Time	5
3.4	Check Pacemaker - CIC State/Cluster Resource State	6
3.5	Check CIC Maintenance Mode	6
3.6	Check Neutron Agents	7
3.7	Check Nova Services	7
3.8	Verify Disk Space Utilization	8
3.9	Verify RAM Utilization	8
3.10	Check iSCSI Multipath Connection to VNX	9
3.11	Check ScaleIO Cluster Status	10
4	Pre- and Post-Activity Health Check Procedure	13
4.1	Check OpenStack Components	13
4.2	State of Extreme Switches from Perspective of Neutron	14
4.3	Check the State of Ethernet Interfaces	15
4.4	Check Service Status	17
4.5	Check Ethernet Statistics	18
4.6	Check RabbitMQ Cluster Status	19
4.7	Check Zombie Processes	20
4.8	Check Fuel Status	20
4.9	Check Fuel Services	21
4.10	Check Swift Store on VNX / ScaleIO	21



5	Report Problems	23
	Reference List	24



1 Introduction

This document is to help support engineers check that the Cloud Execution Environment (CEE) operates in a fault-free state, and to detect issues that can affect normal operation.

1.1 Scope

This document has been verified on the CEE certified configuration, as specified in the *BOM for Certified HW Configurations*, Reference [1]. The process is applicable to other CEE configurations.

1.2 Target Groups

This document is intended for both internal and external customers monitoring system health:

- Support organization personnel
- Customer O&M personnel

1.3 Prerequisites

This section describes the prerequisites for performing the health check procedure.

1.3.1 Documents

Before starting the procedure, ensure that the following documents are available:

- ☐ *Data Collection Guideline*

1.3.2 Conditions

Before performing a health check, ensure that the following conditions are met:

- Users of this document must be familiar with commands and tools within CEE and OpenStack.
- Access to deployment-specific credentials must be available.
- There are no ongoing CEE maintenance activities.



2 Overview

This document covers the procedures for checking the health of CEE and detecting issues before they become threats to the system.

“Health” in the context of this document means that CEE is running, provides the required functionality, and is available for the users.

Health condition is evaluated by executing several checks. These checks are based on the information collected from printouts. If problems are encountered during any of the checks, the user is provided with a recommendation.

The time needed to execute health checks depends on factors such as the complexity of the checks or the system performance. If a check takes a long time, it is possible that CEE is not functioning correctly. The checks in this document are classified as:

- Daily Health Check: Checks for CEE to be executed on a daily basis
- Pre- and Post-Activity Health Check: Checks to be executed before and after an update of CEE or other maintenance activities are performed. It is recommended to execute these checks on a monthly basis.

Note: CEE collects a large amount of ISP and Fault Management (FM) data. Alarms must be available in the management system (Atlas).

2.1 Execution of Commands for Several Hosts

It is possible to execute commands for several hosts by using the below syntax, replacing `<command>` with the specific command to be executed. The following examples show the execution of commands from Fuel, as root user.

- Example syntax for execution of commands for Virtual Cloud Infrastructure Controller (vCIC) nodes:

```
for n in $(fuel node | awk -F '|' '{ $7 ~ =>
/controller/ {print $3}'); do echo ${n}; ssh -o =>
LogLevel=quiet ${n} '<command>'; done
for example
for n in $(fuel node | awk -F '|' '{ $7 ~ =>
/controller/ {print $3}'); do echo ${n}; ssh -o =>
LogLevel=quiet ${n} 'date'; done
```
- Example syntax for execution of commands for Compute Hosts:

```
for n in $(fuel node | awk -F '|' '{ $7 ~ =>
/compute/ {print $3}'); do echo ${n}; ssh -o =>
LogLevel=quiet ${n} '<command>'; done
for example
for n in $(fuel node | awk -F '|' '{ $7 ~ =>
```



```
/compute/ {print $3}'); do echo ${n}; ssh -o ⇒
LogLevel=quiet ${n} 'date'; done
```

- Example syntax for execution of commands for Fuel, vCIC nodes and Compute Hosts:

```
for n in fuel $(fuel node | awk -F '|' '{ $7 ~ ⇒
/controller|compute/ {print $3}'); do echo ${n}; ⇒
ssh -o LogLevel=quiet ${n} '<command>'; done
```

2.2 Execution of Commands for vCICs

In order to populate the environment variables needed for execution of OpenStack commands, and to execute following commands as root, the following commands are recommended:

```
ssh <CEE_Administrator>@<CIC_Hostname/IP_Address>
sudo -i
source openrc
```

2.3 Execution of Commands for Compute Hosts

In order to execute commands as root—which is needed to get command output—it is recommended to use the `sudo` command:

```
ssh <CEE_Administrator>@<Compute_Host_IP_Address>
sudo -i
```



3 Daily Health Check Procedure

This section describes the procedures for checking the health of the CEE system on a daily basis.

3.1 Check Alarms And Alarm History

Performance and Fault Management alarms are reported by Watchmen to Atlas or the Ericsson Cloud Manager (ECM). Check the active alarms and act according to the relevant Operating Instructions (OPIs) in each case.

Required tools

Atlas or ECM Graphical User Interface (GUI)

Conditions

There are no conditions.

Procedure

To check the alarms and alarm history using Atlas, do the following:

1. Log on to Atlas through the GUI.
2. Check for alarms.

Expected result

One of the following results is expected:

- No alarms are found.
- Alarms are found. The cause of the alarms is known. In case of any active alarms, refer to the corresponding OPI and act accordingly.

3.2 Check Presence of Crash Dumps

Required tools

Command-Line Interface (CLI)

**Conditions**

There are no conditions.

Procedure

To check for the presence of crash dumps, do the following:

1. Execute the below commands on all Cloud Infrastructure Controller (CIC) hosts and compute hosts:

```
ls -al /var/log/crash/cores  
ls -al /var/log/crash/kernelcrashes/
```

Expected result

No crash dumps are present.

3.3 Verify Date and Time

Note: Execute the command on all CIC hosts and compute hosts.

Required tools

CLI

Conditions

There are no conditions.

Procedure

To verify the date and time, do the following:

1. Execute the below command from Fuel:

```
for i in `fuel node |grep 'cic-'|awk '{print $5}'`;=>  
do ssh $i date; done
```

Expected result

Time is correct and identical in all CIC hosts.

Note: After the command is executed for the first time, a difference of one second can occur across outputs. Accept the new hostkey insertion with **yes** per CIC, then execute the command a second time and check the new output for the expected result.



3.4 Check Pacemaker - CIC State/Cluster Resource State

Pacemaker is a cluster resource manager.

Required tools

- CLI

Conditions

Installation of CEE has concluded successfully.

Procedure

To check CIC state and Cluster Resource state, do the following:

1. Execute commands on any of the CIC hosts:

```
crm_mon -l -rf | grep FAILED
crm_mon -l -rf | grep -i STOPPED
```

Expected result

A printout. If any resources are in `FAILED` state, then they must be acted upon.

If there are any resources that are in `STOPPED` state, that can be because of dependencies on resources in a `FAILED` state.

3.5 Check CIC Maintenance Mode

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check CIC maintenance mode, do the following:

1. Execute the below command on all CIC hosts:

```
umm status
```

**Expected results**

The command output must indicate that system is in `runlevel 2` (Multiuser mode):

```
runlevel N 2
```

Note: When a CIC is in maintenance mode, the command output contains `umm`. In this case, all OpenStack commands fail.

3.6 Check Neutron Agents

Neutron agents mentioned are monitored by CEE ISP.

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check Neutron agents, do the following:

1. Execute the following command on any CIC:
`neutron agent-list`

Expected result

The agents are alive.

Note: The Neutron DHCP agent has to be active on one CIC host, but it can be present on other CIC hosts with `down` status (displayed as `xxx`).

3.7 Check Nova Services

Required tools

CLI

Conditions

There are no conditions.



Procedure

To check Nova services, do the following:

1. Execute the following command on any CIC:
`nova service-list`

Expected result

A printout where all services per CIC are in `enabled` status.

It is expected that `nova-scheduler`, `nova-conductor` and `nova-consoleauth` services are present and enabled on each CIC. In addition, the `nova-compute` service is enabled on each compute host.

3.8 Verify Disk Space Utilization

Required tools

CLI

Conditions

There are no conditions.

Procedure

To verify disk space utilization, do the following:

1. Execute the below command on all CIC hosts, compute hosts and Fuel:
`df -h`

Expected result

The use per partition is less than 80%.

3.9 Verify RAM Utilization

Note: CPU, RAM, and local disk usage is monitored by Fault Management. Check for alarms.

Required tools

CLI

**Conditions**

There are no conditions.

Procedure

To verify RAM utilization, do the following:

1. Execute the below command on all CIC hosts and compute hosts (output is shown in kB):
`/etc/zabbix/scripts/check_free_memory.sh`

Check the use of RAM.

Expected result

There must be at least 20% of RAM free on the CIC.

Note: In Compute hosts, the use of `ReservedHugePages` for VMs can result in close to 100% RAM usage.

3.10 Check iSCSI Multipath Connection to VNX

The storage on VNX is connected by multi-path connection. Check that these multi-path connections are working.

Required tools

CLI

Conditions

VNX is used for Storage.

Procedure

To check the iSCSI multipath connection to VNX, do the following:

1. Execute the following command from each compute host:
`multipath -ll`
2. Execute the following command from each controller node:
`multipath -ll`

Expected result

- In case of a compute host, verify the following:



- One multipath device corresponds to the Cinder volume.
- The quantity of iSCSI paths corresponds to the number of iSCSI ports used to connect the specific CEE region to VNX.
- In case of a controller node:

If the Swift store on VNX is activated, verify the following:
 - A multipath device exists for each Cinder volume used by the controller for Swift store on VNX.
 - The quantity of iSCSI paths corresponds to the number of iSCSI ports used to connect the specific CEE region to VNX.
- All paths must be in `active ready running state`.

3.11 Check ScaleIO Cluster Status

Check that the ScaleIO cluster status is `Normal` and the ScaleIO components are `Connected`.

Required tools

CLI

Conditions

ScaleIO is used for Storage.

Procedure

To check the status of the ScaleIO cluster, do the following:

1. Execute the below command from Fuel:

```
for node in $(fuel node | grep 'scaleio' | awk '{print $5}' ⇒  
| sort); do echo "Checking $node ... "; ssh -q ⇒  
-oStrictHostKeyChecking=no -oUserKnownHostsFile⇒  
=/dev/null root@${node} scli --query_cluster; done
```

Result:

The command is successfully executed only on the Master MDM ScaleIO host. The ScaleIO cluster status has to be `Normal` and all three MDMs and the two Tie-Breakers has to have the status `Connected`.



```
[root@fuel ~]# for node in $(fuel node | grep 'scaleio' | awk '{print $5}' | sort); do echo "Checking =>
$node ... "; ssh -q -oStrictHostKeyChecking=no -oUserKnownHostsFile=/dev/null root@${node} scl_i --=>
query_cluster; done
Checking scaleio-0-4 ...
Cluster:
  Mode: 5_node, State: Normal, Active: 5/5, Replicas: 3/3
  Virtual IPs: N/A
Master MDM:
  Name: scaleio-0-4, ID: 0x08e9d36a61220180
  IPs: 192.168.11.20, 192.168.12.20, Management IPs: 192.168.2.20, Port: 9011, Virtual IP =>
interfaces: N/A
  Version: 2.0.10000
Slave MDMs:
  Name: scaleio-0-5, ID: 0x3822465b211557c2
  IPs: 192.168.11.26, 192.168.12.26, Management IPs: 192.168.2.26, Port: 9011, Virtual IP =>
interfaces: N/A
  Status: Normal, Version: 2.0.10000
  Name: scaleio-0-6, ID: 0x0392987232ca31f1
  IPs: 192.168.11.25, 192.168.12.25, Management IPs: 192.168.2.25, Port: 9011, Virtual IP =>
interfaces: N/A
  Status: Normal, Version: 2.0.10000
Tie-Breakers:
  Name: scaleio-0-7, ID: 0x3c10d0385bc5f9f3
  IPs: 192.168.11.24, 192.168.12.24, Port: 9011
  Status: Normal, Version: 2.0.10000
  Name: scaleio-0-8, ID: 0x3c88927e3a479294
  IPs: 192.168.11.27, 192.168.12.27, Port: 9011
  Status: Normal, Version: 2.0.10000
Checking scaleio-0-5 ...
Error: MDM failed command. Status: This command is not supported on the Slave MDM. Please use the =>
Master MDM IP to access the cluster
Checking scaleio-0-6 ...
Error: MDM failed command. Status: This command is not supported on the Slave MDM. Please use the =>
Master MDM IP to access the cluster
Checking scaleio-0-7 ...
```

Example 1 Example Output for ScaleIO Cluster Query in Fuel

2. Log into the ScaleIO management system from the MDM master host, which is identified in the previous command:
`scli --login --username <NAME> [--password <PASSWORD>]`
3. Execute the following command from the Master MDM host:
`scli --query_all_sdc`



```

MDM restricted SDC mode: Disabled
Query all SDC returned 6 SDC nodes.
SDC ID: 3990177300000000 Name: cic-1 IP: 192.168.11.30 State: Connected GUID: =>
E558396B-B111-4906-A1BC-E9E1360880C2 OS Type: LINUX Loaded Version: 2.0.11000 Installed Version: 2.0.11000
  Read bandwidth: 86 IOPS 9.6 MB (9780 KB) per-second
  Write bandwidth: 26 IOPS 104.0 KB (106496 Bytes) per-second
SDC ID: 3990177400000001 Name: compute-0-1 IP: 192.168.12.24 State: Connected GUID: =>
95F93469-786A-44BD-BFFE-675EB4FCDF79 OS Type: LINUX Loaded Version: 2.0.11000 Installed Version: 2.0.11000
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 0 IOPS 0 Bytes per-second
SDC ID: 3990177500000002 Name: cic-3 IP: 192.168.12.28 State: Connected GUID: =>
BDB410CA-414C-47D2-84A7-EBFD3105B638 OS Type: LINUX Loaded Version: 2.0.11000 Installed Version: 2.0.11000
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 21 IOPS 81.0 KB (82944 Bytes) per-second
SDC ID: 3990177600000003 Name: compute-0-3 IP: 192.168.11.26 State: Connected GUID: =>
F8D92569-3DA7-40A7-85C1-8FDB2A613A72 OS Type: LINUX Loaded Version: 2.0.11000 Installed Version: 2.0.11000
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 0 IOPS 0 Bytes per-second
SDC ID: 3990177700000004 Name: cic-2 IP: 192.168.11.29 State: Connected GUID: =>
6B238228-A97B-4026-A463-0496DF12BC31 OS Type: LINUX Loaded Version: 2.0.11000 Installed Version: 2.0.11000
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 0 IOPS 0 Bytes per-second
SDC ID: 3990177800000005 Name: compute-0-2 IP: 192.168.12.20 State: Connected GUID: =>
54EBEF9E-6164-442F-B4FF-AA8F9F5E92C0 OS Type: LINUX Loaded Version: 2.0.11000 Installed Version: 2.0.11000
  Read bandwidth: 0 IOPS 0 Bytes per-second
  Write bandwidth: 0 IOPS 0 Bytes per-second

```

Example 2 Example Output for SDC Query

4. Execute the following command from the Master MDM host:

```
scli --query_all_sds
```

```

Protection Domain b0ac10ad00000000 Name: protection_domain1
SDS ID: 1036a03500000004 Name: scaleio-0-6 State: Connected, Joined IP: 192.168.11.25,192.168.12.25 =>
Port: 7072 Version: 2.0.10000
SDS ID: 1036a03400000003 Name: scaleio-0-7 State: Connected, Joined IP: 192.168.11.22,192.168.12.22 =>
Port: 7072 Version: 2.0.10000
SDS ID: 1036a03300000002 Name: scaleio-0-8 State: Connected, Joined IP: 192.168.11.21,192.168.12.21 =>
Port: 7072 Version: 2.0.10000
SDS ID: 1036a03200000001 Name: scaleio-0-5 State: Connected, Joined IP: 192.168.11.27,192.168.12.27 =>
Port: 7072 Version: 2.0.10000
SDS ID: 1036a03100000000 Name: scaleio-0-4 State: Connected, Joined IP: 192.168.11.23,192.168.12.23 =>
Port: 7072 Version: 2.0.10000

```

Example 3 Example Output for SDS Query

Expected results

- The ScaleIO cluster status is Normal
- All three MDMs and the two Tie-Breakers have the status Connected
- All SDCs are in Connected status.
- All SDSs are in Connected status.



4 Pre- and Post-Activity Health Check Procedure

This section describes the checks to be executed before and after an update of CEE or other maintenance activities are performed. It is recommended to execute these checks on a monthly basis.

4.1 Check OpenStack Components

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check the OpenStack components, do the following:

1. Execute the commands below on any CIC host, or in case of an extensive health check, on all CIC hosts:

```
nova list
nova hypervisor-list
glance image-list
cinder service-list
ceilometer meter-list
openstack project list
openstack service list
neutron net-list
```

Note: In certain fault scenarios, the commands above fail on certain CIC hosts and succeed on other CIC hosts.

In case of a Single Server environment, ceilometer is not available, so the command `ceilometer meter-list` is not needed.

Expected result

- A successful output. This is used as a first indication that the corresponding OpenStack component is running correctly.
- Glance: at least one image is returned in the list.



- Cinder: schedulers are enabled and up on all the controller nodes.
- `cinder-volume` is present as one of the services when EMC VNX is used.
Note: `cinder-volume` must only be up in one of the controller nodes.
- Ceilometer: meters are listed.

4.2 State of Extreme Switches from Perspective of Neutron

Note: This section is only applicable for systems using Extreme switches configured dynamically by CEE.

Extreme switches are monitored by Fault Management. Check for alarms.

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check the state of the Extreme switches, do the following:

1. Execute the command below on any CIC:
`neutron deviceport-list`

Expected result

All configured traffic ports are listed as expected in the output.



id	name	port_type
1f958a74-6ba0-43e3-8222-8f3cc18db6b7	DC196_SWB_X670V_port_7	SERVER
2e11ce9f-d07c-45cd-92ef-a205c0ae64b1	DC196_SWA_X670V_port_15	SERVER
55a9d6cb-dae1-4491-baf8-a3e72a18ce8d	DC196_SWB_X670V_port_13	SERVER
5faf41f2-dab4-4a6c-8abe-5880d9e3712b	DC196_SWB_X670V_port_49	GATEWAY
82fa89fe-820f-4d52-8225-635a33ca3454	DC196_SWB_X670V_port_5	SERVER
8a287b82-c1d4-4e7f-a216-c887bf7b1890		DISCONNECTED
alf5474d-1651-4561-879e-176b135c83d0	DC196_SWA_X670V_port_57	ISC
b419a91d-4f6e-4b26-b803-e620552d7f86	DC196_SWB_X670V_port_57	ISC
c2bffca4-4cb1-4c47-b9dc-dd4e6695c4e5	DC196_SWA_X670V_port_13	SERVER
c3b24f91-b491-4cfe-b354-d9b5657d8cbf	DC196_SWA_X670V_port_7	SERVER
cbd48a5b-7d91-4d08-bd2d-7d0ff2213a74	DC196_SWB_X670V_port_15	SERVER
d80d4cf6-cbcf-4682-ad33-5437f656d512		DISCONNECTED
edbd40aa-e688-4533-a79b-049cb696527d	DC196_SWA_X670V_port_49	GATEWAY
efee50b7-49c7-4cd7-8615-12a84f889c9c	DC196_SWA_X670V_port_5	SERVER

Example 4 Neutron deviceport-list Command Output Example

4.3 Check the State of Ethernet Interfaces

Required tools

CLI

Conditions

Installation of CEE has concluded successfully.

Procedure

To check the state of the Ethernet interfaces, do the following:

1. Execute the command below on all CIC hosts and compute hosts:
`ip a`

2. Execute the command below on all compute hosts:

`ovs-appctl bond/show`

Expected result

Note: For Dell Single Server platform no output is expected, as this platform does not support redundancy.

- In general, the state of all Ethernet interfaces must be `up`. However, if some interfaces are not in use, they can be in `down` state.
- OVS bonding must show that Ethernet ports are `enabled`.

Example outputs are shown below.



```
root@compute-0-1:~# ovs-appctl bond/show
---- bond-fw-admin ----
bond_mode: active-backup
bond may use recirculation: no, Recirc-ID : -1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
lacp_status: off
active slave mac: 01:23:45:67:89:ab(eth0)

slave eth0: enabled
    active slave
    may_enable: true

slave eth1: enabled
    may_enable: true

---- bond-prv ----
bond_mode: balance-slb
bond may use recirculation: no, Recirc-ID : -1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
next rebalance: 725 ms
lacp_status: negotiated
active slave mac: 23:45:67:89:ab:cd(dpdk0)

slave dpdk0: enabled
    active slave
    may_enable: true

slave dpdk1: enabled
    may_enable: true
    hash 247: 13 kB load
```

Example 5 HP Multi-Server platform, compute host Ethernet interface



```

root@compute-0-1:~# ovs-appctl bond/show
---- bond-fw-admin ----
bond_mode: active-backup
bond may use recirculation: no, Recirc-ID : -1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
lacp_status: off
active slave mac: 45:67:89:ab:cd:ef(eth4)

slave eth4: enabled
    active slave
    may_enable: true

slave eth5: enabled
    may_enable: true

---- bond-prv ----
bond_mode: active-backup
bond may use recirculation: no, Recirc-ID : -1
bond-hash-basis: 0
updelay: 0 ms
downdelay: 0 ms
lacp_status: off
active slave mac: 67:89:ab:cd:ef:01(dpdk0)

slave dpdk0: enabled
    active slave
    may_enable: true

slave dpdk1: enabled
    may_enable: true

```

Example 6 BSP platform, compute host Ethernet interface

4.4 Check Service Status

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check the service status, do the following:



1. Execute the below command on Fuel:

```
for i in `fuel node |grep 'cic-'|awk '{print $5}'`; =>  
do ssh $i service --status-all; done
```

Expected results

The printout must be complete, and must not fail or hang. The analysis of the service state is outside the scope of this document.

4.5 Check Ethernet Statistics

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check the Ethernet statistics, do the following:

1. Execute the command below on **all** compute hosts:

```
ovs-dpctl show -s
```
2. Execute the below command on all CIC hosts and compute hosts:

```
netstat -i
```
3. Check that only a few or no Receiver Mode (RX) or Transmitter Mode (TX) errors are indicated.

To verify if the system is considered healthy, use the following guidelines:

- In the output of the `ovs-appctl dpctl/show -s <datapath type>` command the threshold for acceptable “lost” and “error” frames is 0.002%.
- In the output of the `netstat -i` command the threshold for acceptable “dropped” and “errors” frames (RX-DROP, TX-DROP, RX-ERR, TX-ERR) is 0.01%.



4.6 Check RabbitMQ Cluster Status

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check the RabbitMQ cluster status, do the following:

1. Execute the command below on any CIC:
`rabbitmqctl cluster_status`
2. Execute the command below on any CIC:
`rabbitmqctl list_queues`

Expected result

A printout showing the correct status of the node.

The following printout example is showing the correct status of the node `rabbit@cic-0-3`:

```
[{nodes, [{disc, ['rabbit@cic-0-1', 'rabbit@cic-0-2', =>
'rabbit@cic-0-3']}]},
{running_nodes, ['rabbit@cic-0-2', 'rabbit@cic-0-1', =>
'rabbit@cic-0-3']},
{cluster_name, <<"rabbit@cic-0-1.domain.tld">>},
{partitions, []}]
...done.
```

Example 7 RabbitMQ Cluster Status

There must be close to “0” messages in each queue.

- The printout must be complete, must not fail or hang in between, and must terminate with the word: `done`.
- There must be no warnings or errors in the beginning of the printouts.
- Perform further checks if you suspect that the queue disappears when, for example, OpenStack components are not running correctly. For example, one Ceilometer agent queue must be present for each compute host, and one Nova compute queue for each host.



4.7 Check Zombie Processes

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check for zombie processes, do the following:

1. Execute the below command on all CIC hosts and compute hosts:
`ps -efa | grep defunct`

Expected result

A printout listing all defunct processes.

4.8 Check Fuel Status

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check the Fuel status, do the following:

1. By CLI (from Fuel master):

```
fuel node
```

Expected result

The command must return a printout with a list of CIC and compute hosts. Each node must display the status column as `ready`, and the online column as `True`.



4.9 Check Fuel Services

Required tools

CLI

Conditions

There are no conditions.

Procedure

To check the Fuel services, do the following:

1. By CLI from Fuel:

```
fuel-utils check_all | grep ready | cut -d' ' -f1  
fuel-utils check_all
```

The `fuel-utils check_all` command checks the following services: astute, cobbler, keystone, mcollective, nailgun, nginx, ostf, postgres, rabbitmq, rsync, rsyslog.

Expected result

The command returns a list of all working Fuel services. Any service missing from the list is not functional.

4.10 Check Swift Store on VNX / ScaleIO

Required tools

CLI

Conditions

Swift store on VNX or ScaleIO needs to be activated.

Procedure

To check the Swift store on either VNX or ScaleIO, do the following:

1. Use the `cinder list` command in one of the CIC hosts.
 - a. Check the number of cinder volumes displayed for each CIC in the `Name` column of the command output. Each volume name includes



the name of the relevant CIC as shown in the example of volume name below:

```
CEE+cic-2.domain.tld+/dev/image/glance+1
```

These structured volume names consist of the following elements in the below order:

- CEE+
- CIC name, for example, `cic-2.domain.tld+`
- Logical volume path, for example, `/dev/image/glance`
- Optional: an integer number, for example, `+1`

- b. Summarize the values displayed in the `Size` column for each CIC.
2. Use the `pvs` command in each CIC to verify that there are as many physical volumes connected to the volume group `image` as the number of cinder volumes displayed for the specific CIC in Step 1.
 3. For each CIC, use the `lvdisplay image` command and verify that the logical volume size of the Swift store displayed in the `LV Size` row of the command output equals the summarized size of the cinder volumes for the specific CIC checked in Step 1.

Expected result

- The number of physicals volumes connected to the volume group `image` at each CIC equals the number of cinder volumes for the specific CIC.
- The logical volume size of the Swift store in each CIC equals the summarized size of the cinder volumes for the specific CIC.



5 Report Problems

Collect data related to the occurring problems, see the *Data Collection Guideline*.

For persistent problems, contact the next level of support.



Reference List

- [1] *BOM for Certified HW Configurations*, 1/006 51-CSA 113 125/5 Uen