ERICSSON

Data Collection Guideline

Cloud Execution Environment

OPERATING INSTRUCTION

7/1543-AZE 102 01 Uen E

Contents

1 Introduction

This document lists the troubleshooting data to be collected and enclosed in a Customer Service Request (CSR). A CSR is made if a problem is experienced with Cloud Execution Environment (CEE).

This document also describes the procedure to collect the needed information.

1.1 Scope

This document is applicable for CEE configurations. It covers:

- Mandatory data collection, see Section 3 on page 5
- Additional (optional) data collection related to specific problems, see Section 4 on page 10

The process has been verified on the CEE certified configuration, as specified in *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 raising a CSR:

- Support organization personnel
- Customer O&M personnel

1.3 Prerequisites

This section provides information on the conditions that apply to the procedure.

1.3.1 User Access

The Operator must have access to the deployment-specific credentials:

☐ Username and password for CIC and Compute hosts with `sudo` privileges and OpenStack access

☐ Username and password for Fuel

☐ Username and password for Extreme switches

Note: Only needed for Extreme switches configured dynamically by the CEE.

☐ Username and password for EMC VNX

The user must belong to one or more of the following LDAP admin groups:

- ☐ storage_sanadmin
- ☐ storage_storageadmin
- ☐ storage_admin

☐ In case the Blade Server Platform (BSP) HW is used:

- ☐ BSP username and password for BSP (DMXC NBI) with System Administrator role
- ☐ DMXC NBI IP address (the IP address in network `bsp_om_sp`)

For information on Identity and Access Management, see *Security User Guide*.

1.3.2 Configuration Data

The address variables used in the site-specific *IP and VLAN plan* are used throughout this document.

The connectivity to Compute hosts is explained in Section 6.1 on page 27.

The IP addresses of CIC, Fuel, and Compute host are shown in Table 1.

Table 1 IP Addresses of CIC, Fuel, Compute Host

Designation	VLAN	Variable Name
cic-x	`cee_om_sp`	***<CIC Interface (dynamic)>***
compute-x-y		***<Compute Host (dynamic)>***(1).
Fuel	`fuel_ctrl_sp`	***<Fuel (static)>***

(1) See Section 6.1 on page 27, for how to obtain values

The Switch Management addresses are shown in Table 2. The VLAN is `cee_ctrl_sp`.

Table 2 Extreme Switch IP Addresses

Designation	IPv4 Address Variable
Traffic Switch A	***<Traffic_switch_A (static)>***
Traffic Switch B	***<Traffic_switch_B (static)>***
Storage Switch A	***<Storage_switch_A (static)>***

Designation	IPv4 Address Variable
Storage Switch B	*<Storage_switch_B (static)>*
Control Switch A	*<Control_switch_A (static)>*
Control Switch B	*<Control_switch_B (static)>*

Note: Only needed for Extreme switches configured dynamically by the CEE

The EMC VNX has two Storage Processors, these are accessible on the following addresses, as shown in Table 3. The VLAN is `cee_ctrl_sp`.

Table 3 EMC VNX Storage Processor IP Addresses

Designation	IPv4 Address Variable
Storage Processor A	*<EMC SP-A mgmt (static)>*
Storage Processor B	*<EMC SP-B mgmt (static)>*

2 Workflow

The workflow for collecting troubleshooting data is as follows:

1. Collecting mandatory data needed in connection with any problems experienced, see Section 3 on page 5.
2. Collecting additional, specific data based on the type of problem experienced, see Section 4 on page 10.

   For alarms and alerts, collect the data as specified in Table 4.
3. Finalize data collection, see Section 5 on page 24.

Note: If technical problems are experienced during the data collection, contact the next level of support.

3 Mandatory Data Collection

The following data is always collected, irrespective of the specific problem type.

3.1 Overview of Collected Data

The following section describes the steps to perform the data collection.

The following items must be added to a CSR, and the procedure to collect the data is described in this document.

- System version

  The system version is given in the `/etc/cee_version.txt` file.

  Example:

  ```
  [root@fuel ~]# cat /etc/cee_version.txt
  RELEASE=CEE CXC1737883_3-5636
  NAME=Kilo on Ubuntu 14.04
  VERSION=16-R1A16-084fef8-7.0
  ```

  In the example, the data for CSR are:

  - `CXC1737883_3` in line `RELEASE` is the product number: **CXC1737883/3**
  - The second cluster of characters in line `VERSION` are the revision: **R1A16**

  In this case the CEE system version in the CSR or TR is: **CXC1737883/3 R1A16**.

- Logs (Fuel, compute hosts, and CICs)
- Core dumps and kernel crash dumps
- Alarms and notifications
- System state

Only include logs and dumps created and updated within the relevant time period (one week before detecting the fault).

The following items can be added as contextual information to a CSR. The data must be acquired from other sources, and the procedures are not described in this document:

- Application name and version (for example: MSP 7.0 CP1)

- Hardware configuration
  - Compute hardware type (for example, HP c7000 BL460c gen9 blades)
  - Number of servers
  - Networking hardware (Extreme switch model)

    Note: Only needed for Extreme switches configured dynamically by the CEE
  - Storage hardware type
  - Border gateway and firewall
- Other site-specific configuration (for example: network diagram and solution description)

3.2 Log on to System

1. Check the options available to log on to the system. See Section 6.1 on page 27 if you need more information on how to connect.
2. Log on to the system.

3.3 Data Collection on Fuel

Step 1 , and Step 2 are executed on Fuel.

1. Create data collection directory on Fuel:

```
export DCG=datacollection-`fuel env |awk 'FNR == 3 ⇒
{print $5}'`-`date +%Y%m%d%H%M%S`

export DCGDIR=/root/$DCG

mkdir -p ${DCGDIR}
```

   Note: Make a note of the data collection folder name `DCG` created, it is used in finalizing data collection, see Section 3.6 on page 9.

2. Add subfolders for other components:

```
mkdir ${DCGDIR}/extreme ${DCGDIR}/emc ${DCGDIR}/Atlas
```

   Note: For further use of Atlas, see the *Atlas Troubleshooting Guideline*, Reference [2].

3. For mandatory data collection on Fuel, follow the below steps:

   a. Use the following commands:

```
fuel node > ${DCGDIR}/nodes.txt
fuel rel > ${DCGDIR}/rel.txt
fuel env > ${DCGDIR}/fuelenv.txt
```

```
cp /mnt/cee_config/* ${DCGDIR}
cp /etc/cee_version.txt ${DCGDIR}
cat /var/log/ansible.log > ${DCGDIR}/⇒
ansible_log.txt
```

b. Execute the following as `root` user on vFuel to collect data about all nodes:

```
for n in fuel $(fuel node | awk -F '|' '$7 ~ /controller|compute|cinder/ {print $3}'); ⇒
do echo ${n}; ssh -o LogLevel=quiet ${n} 'uname -a; ip a; netstat -alnp; df -a; ⇒
cat /proc/mounts; cat /proc/cpuinfo; cat /proc/meminfo; dmidecode; lspci; lsmod; dmesg;' ⇒
| gzip >${DCGDIR}/${n}_info.txt.gz; done
```

3.4 Data Collection on a CIC

Collect data on any one of the CICs.

1. Connect to the CIC:

   ```
   ssh <CEE administrator>@<CIC address>
   sudo -i
   source openrc
   ```

2. Create data collection directory on the CIC:

   ```
   export DCG=datacollection-`hostname`⇒
   -`date +%Y%m%d%H%M%S`

   export DCGDIR=/var/lib/glance/$DCG

   mkdir -p ${DCGDIR}
   ```

3. For System State, issue the commands:

   ```
   export OS_PASSWORD=<password for OpenStack user>

   watchmen-client active-alarm-list ⇒
   > ${DCGDIR}/alarm-list.txt
   watchmen-client alarm-history --from <previous day ⇒
   in yyyy-mm-dd format> > ${DCGDIR}/alarm-history.txt

   watchmen-client --os-username watchmen --os-password ⇒
   Watchmen Password --os-tenant-name services ⇒
   snmp-trap-config-list > ${DCGDIR}/trap-config-list.txt

   crm_mon -1 -rf > ${DCGDIR}/crm-status.txt

   cinder service-list > ${DCGDIR}/cinder-service-list.txt

   nova service-list > ${DCGDIR}/nova-service-list.txt
   ```

```
rabbitmqctl cluster_status > ${DCGDIR}/mq-status.txt
```

4. For high-level overview of Virtual Machines (VMs), networks and volumes:

   ```
   nova list --all-tenants > ${DCGDIR}/nova-list.txt

   cinder list > ${DCGDIR}/cinder-list.txt

   neutron net-list > ${DCGDIR}/neutron-service-list.txt

   neutron port-list > ${DCGDIR}/neutron-port-list.txt
   ```

5. For general data collection, use the following commands:
   sosreport --batch --tmp-dir ${DCGDIR}

   sosreport displays the following:

   ${DCGDIR}/sosreport-*<archive name>*.⇒
   <tr ref>-*<date>*.tar.xz

   /tmp/sosreport-*<archive name>*.⇒
   <tr ref>-*<date>*.tar.xz.md5

   If **sosreport** was unsuccessful, run the **top** command, then save a screenshot of the running processes for the next level of maintenance support.

6. For data collection related to IdAM:

   ```
   cee-idam user-list > ${DCGDIR}/cee-IdAM-list.txt
   ```

7. For data collection related to backup:

   ```
   su -c "cee-backup list" ceebackup > ⇒
   ${DCGDIR}/cee-backup-list.txt
   ```

8. For connectivity issues, collect all the files from the below library:

   /etc/ssl/certs/CEE

3.5 Data Collection from Compute Host

The following commands must be executed on all affected compute hosts.

1. Connect to Compute host:

   ```
   ssh <CEE administrator>@<Compute address>
   ```

```
sudo -i
source openrc
```

2. Create data collection directory on Compute host:
   ```
   export DCG=datacollection-`hostname`-⇒
   `date +%Y%m%d%H%M%S`
   export DCGDIR=/root/$DCG
   mkdir $DCGDIR
   ```

3. For general data collection, use the commands:
   ```
   sosreport --batch --tmp-dir ${DCGDIR}
   ```

   sosreport displays the following:

   ```
   ${DCGDIR}/sosreport-<archive name>.⇒
   <tr ref>-<date>.tar.xz

   /tmp/sosreport-<archive name>.⇒
   <tr ref>-<date>.tar.xz.md5
   ```

   If **sosreport** was unsuccessful, run the **top** command, then save a screenshot of the running processes for next level of maintenance support.

3.6 Finalization Steps

To finalize mandatory data collection, follow the steps in Section 5 on page 24.

4 Data Collected Based on Specific Problem Types

In addition to the mandatory data collection in Section 3 on page 5, collect data using one or more of the following subsections. Select subsections based on the type of the problem experienced, or according to the alert or alarm type, as listed in Table 4.

Table 4 Data Collection for Alarms and Alerts

Collect specified data after collecting mandatory data, as described in Section 3 on page 5.	
Alarm or Alert	**Data to Collect on Specific Problem Types**
Centralized Storage Alert	See Section 4.5.2 on page 20.
CIC Failed	See Section 4.2 on page 11.
CIC Restarted	See Section 4.2 on page 11.
Complete CIC Service Restarted	See Section 4.2 on page 11. For hardware-related information, refer to the documentation of the specific hardware.
Compute Host Failed	See Section 4.3 on page 13.
Compute Host Restarted	See Section 4.3 on page 13.
Ethernet Port Aggregator Fault	See Section 4.4 on page 14. For hardware-related information, refer to the documentation of the specific hardware.
Ethernet Port Fault	See Section 4.4 on page 14. For hardware-related information, refer to the documentation of the specific hardware.
Ethernet Switch Port Fault	See Section 4.4 on page 14. For hardware-related information, refer to the documentation of the specific hardware.
Fan Failure	See Section 4.7 on page 23. For hardware-related information, refer to the documentation of the specific hardware.
Fuel Failed	See Section 4.1 on page 11.

Fuel Restarted	See Section 4.1 on page 11.
High CPU Load	See Section 4.3 on page 13.
High Local Disk Utilization	See Section 4.3 on page 13.
High Memory Utilization	See Section 4.3 on page 13.
Power Supply Failure	See Section 4.7 on page 23. For hardware-related information, refer to the documentation of the specific hardware.
Service Stopped	See Section 4.2 on page 11 and Section 4.3 on page 13.
Service Permanently Stopped	See Section 4.2 on page 11 and Section 4.3 on page 13.
VM Evacuation Failed	See Section 4.6 on page 21.
VM Unavailable	See Section 4.6 on page 21.
VMs Restarted due to vSwitch Restart	See Section 4.6 on page 21.
NTP Authentication Failure	See Section 4.4.4 on page 17.
NTP Stratum Level Failure	See Section 4.4.4 on page 17.
NTP Upstream Server Failure	See Section 4.4.4 on page 17.

4.1 Fuel-Related Problems

For problems related to vFuel, including Fuel backup or deployment, issue the following commands:

```
fuel-utils check_all | grep ready | cut -d' ' -f1 >
${DCGDIR}/fuel-utils.txt

lsblk > ${DCGDIR}/lsblk.txt
```

Use the following command to collect the logs and crash dumps:

```
[root@fuel ~]#tar czvf ${DCGDIR}/logs.tgz /var/log
```

4.2 CIC-Related Problems

For problems related to CIC, execute the following commands on a CIC:

```
cd /var/lib/mysql

du -Sh
```

```
ls -laFRSh

mysql -e "show status like 'wsrep%';"

mysql -u root -e "SHOW STATUS LIKE '';"

mysql
```

Check the disk usage/occupancy of the tables in the MySQL database:

```
SELECT table_schema "Tables", Round(Sum(data_length + index_length) / 1024 / 1024, 1) ⇒
"DB Size in MB" FROM information_schema.tables GROUP BY table_schema;
```

Choose the database table that has the highest disk occupancy. For example, the Zabbix table:

```
use zabbix
SELECT CONCAT(table_schema, '.', table_name), CONCAT(ROUND(table_rows / 1000000, 2), 'M') rows, ⇒
CONCAT(ROUND(data_length / ( 1024 * 1024 * 1024 ), 2), 'G') DATA, CONCAT(ROUND(index_length / ⇒
( 1024 * 1024 * 1024 ), 2), 'G') idx, CONCAT(ROUND(( data_length + index_length ) / ⇒
( 1024 * 1024 * 1024 ), 2), 'G') total_size, ROUND(index_length / data_length, 2) idxfrac ⇒
FROM information_schema.TABLES ORDER BY data_length + index_length DESC LIMIT 10;
```

An example printout is:

```
+---------------------------------------+-------+-------+-------+------------+---------+
| CONCAT(table_schema, '.', table_name) | rows  | DATA  | idx   | total_size | idxfrac |
+---------------------------------------+-------+-------+-------+------------+---------+
| zabbix.history_uint                   | 1.64M | 0.08G | 0.07G | 0.15G      |    0.92 |
| zabbix.history                        | 0.37M | 0.02G | 0.02G | 0.03G      |    0.94 |
| zabbix.trends_uint                    | 0.02M | 0.00G | 0.00G | 0.00G      |    0.00 |
| zabbix.items                          | 0.00M | 0.00G | 0.00G | 0.00G      |    0.37 |
| zabbix.sessions                       | 0.01M | 0.00G | 0.00G | 0.00G      |    0.25 |
| zabbix.images                         | 0.00M | 0.00G | 0.00G | 0.00G      |    0.01 |
| zabbix.trends                         | 0.01M | 0.00G | 0.00G | 0.00G      |    0.00 |
| mysql.help_topic                      | 0.00M | 0.00G | 0.00G | 0.00G      |    0.04 |
| mysql.innodb_index_stats              | 0.00M | 0.00G | 0.00G | 0.00G      |    0.00 |
| nova.instances                        | 0.00M | 0.00G | 0.00G | 0.00G      |    0.64 |
+---------------------------------------+-------+-------+-------+------------+---------+
```

Example 1 Zabbix Table

Collect data from MongoDB. See the standard documentation.

Collect data for Pacemaker:
`cibadmin --query > ${DCGDIR}/pacemaker-configuration.txt`

Check status of the services:
`service --status-all > ${DCGDIR}/service-status.txt`

The RabbitMQ prints must be included with the following commands:

```
rabbitmqctl report > ${DCGDIR}/mq-report.txt

rabbitmqctl status > ${DCGDIR}/mq-status.txt

rabbitmqctl cluster_status > ${DCGDIR}/mq-cluster.txt

rabbitmqctl list_users > ${DCGDIR}/mq-users.txt
```

```
rabbitmqctl list_vhosts > ${DCGDIR}/mq-vhosts.txt

rabbitmqctl list_permissions > ${DCGDIR}/mq-permiss.txt

rabbitmqctl list_parameters > ${DCGDIR}/mq-params.txt

rabbitmqctl list_policies > ${DCGDIR}/mq-policy.txt

rabbitmqctl list_queues > ${DCGDIR}/mq-queues.txt

rabbitmqctl list_exchanges > ${DCGDIR}/mq-exchanges.txt

rabbitmqctl list_bindings > ${DCGDIR}/mq-binds.txt

rabbitmqctl list_connections > ${DCGDIR}/mq-connects.txt

rabbitmqctl list_channels > ${DCGDIR}/mq-channels.txt

rabbitmqctl list_consumers > ${DCGDIR}/mq-consums.txt
```

4.3 Compute-Related Problems

1. For data collection related to Nova (Compute), Image (Glance) and Identity (Keystone):

   ```
   nova list --all-tenants --fields name,status,⇒
   task_state,host,Networks,instance_name ⇒
   > ${DCGDIR}/nova-list-extended.txt

   nova hypervisor-list > ${DCGDIR}⇒
   /nova-hypervisor-list.txt

   nova availability-zone-list > ⇒
   ${DCGDIR}/nova-az-list.txt

   nova flavor-list > ${DCGDIR}/nova-flavor-list.txt

   glance image-list > ${DCGDIR}/glance-image-list.txt

   nova keypair-list > ${DCGDIR}/nova-keypair-list.txt

   nova hypervisor-stats > ⇒
   ${DCGDIR}/nova-hypervisor-stats.txt

   nova hypervisor-list |grep -v ID|grep -v + ⇒
   |awk '{print "nova hypervisor-show " $2 }'|bash > ⇒
   ${DCGDIR}/nova-hypervisor-show.txt

   nova usage-list > ${DCGDIR}/nova-usage-list.txt
   ```

```
nova absolute-limits > ⇒
${DCGDIR}/nova-absolute-limits-all.txt

openstack project list > ⇒
${DCGDIR}/openstack-projects.txt

nova quota-show > ${DCGDIR}/nova-quota-list.txt

openstack catalog list > ${DCGDIR}/keystone-catalog.txt

openstack user list --long > ⇒
${DCGDIR}/keystone-user-list.txt
```

4.4 Networking-Related Problems

If a networking problem is suspected, the following must be collected:

- ☐ OVS-bugtool output
- ☐ Neutron config and logs from CICs
- ☐ Neutron config and logs from all compute hosts
- ☐ Extreme switch configurations and logs

  Note: Only needed for Extreme switches configured dynamically by the CEE

- ☐ Control network switches configurations and logs
- ☐ Linux system logs and dumps

Use the following subsections, together with the procedure in Section 3 on page 5 for general data collection.

4.4.1 Neutron

1. For data collection related to networking, enter the following commands:

```
nova interface-list <VM name or UUID>

neutron agent-list > ${DCGDIR}/neutron-agent-list.txt

neutron ext-list > ${DCGDIR}/neutron-ext-list.txt

neutron host-list > ${DCGDIR}/neutron-host-list.txt

neutron staticroute-list > ${DCGDIR}/neutron-route.txt
```

```
neutron net-list > ${DCGDIR}/neutron-net-list.txt

neutron net-list |grep -v id|grep -v + |awk '{print ⇒
"neutron net-show " $2 }'|bash > ⇒
${DCGDIR}/neutron-net-show-all.txt

neutron subnet-list > ${DCGDIR}/neutron-subnet-list.txt

neutron subnet-list |grep -v id|grep -v + ⇒
|awk '{print "neutron subnet-show " $2 }'|bash > ⇒
${DCGDIR}/neutron-subnet-show-all.txt

neutron port-list > ${DCGDIR}/neutron-port-list.txt

neutron port-list |grep -v id|grep -v + |awk '{print ⇒
"neutron port-show " $2 }'|bash > ⇒
${DCGDIR}/neutron-port-show-all.txt

neutron router-list > ${DCGDIR}/neutron-router-list.txt

neutron router-list |grep -v id|grep -v + ⇒
|awk '{print "neutron router-show " $2 }'|bash > ⇒
${DCGDIR}/neutron-router-show-all.txt

neutron router-list |grep -v id|grep -v + ⇒
|awk '{print "neutron router-port-list " $2 }'⇒
|bash > ${DCGDIR}/neutron-port-list-all.txt

cp /etc/neutron/neutron.conf ${DCGDIR}

cp /etc/neutron/plugin.ini ${DCGDIR}
```

2. The following commands are only applicable for systems using Extreme switches, configured dynamically by the CEE:

```
neutron device-list > ${DCGDIR}/neutron-device-list.txt

neutron device-list |grep -v id|grep -v + ⇒
|awk '{print "neutron device-show " $2 }'|bash > ⇒
${DCGDIR}/neutron-device-show-all.txt

neutron deviceport-list > ⇒
${DCGDIR}/neutron-deviceport-list.txt

neutron deviceport-list |grep -v id|grep -v + ⇒
|awk '{print "neutron deviceport-show " $2 }'|bash > ⇒
${DCGDIR}/neutron-deviceport-show-all.txt
```

4.4.2 OVS/CSS Automatic Collection of Data

Issue the command on each CIC and compute host:
ovs-bugtool --yestoall

The last line of the output is the following:

`Writing tarball <bugtool filename> successful.`

Copy the created file to the data collection area:

cp *<bugtool filename>* $DCGDIR/

On each CIC and compute host, remove the temporary file:

rm -f *<bugtool filename>*

4.4.3 Host Networking

Enter the following commands:

```
dpdk_nic_bind.py --status > ${DCGDIR}/nicbinding.txt

ovs-appctl dpctl/show -s > ${DCGDIR}/dpctl.txt
cp /var/log/ndevalarm/log-NetDevAlarm*.log ${DCGDIR}/netdevalarm.txt
cp /var/log/ndevalarm/alarm_send.log ${DCGDIR}/netdevalarm_alarm_send.txt
```

4.4.3.1 Link Redundancy on BSP Traffic and Control Network

Execute the following commands on the compute hosts:

```
ovs-appctl cfm/show > ${DCGDIR}/ovs_cfm.txt

ovs-appctl bond/show > ${DCGDIR}/ovs_bond.txt

ovs-vsctl show > ${DCGDIR}/ovs_vsctl.txt

cp /var/log/arpmon/arpmon.log ${DCGDIR}

cp /etc/arpmon/arp_config.yaml ${DCGDIR}
```

Execute the following on the Fuel master:

```
cp /mnt/cee_config/config.yaml ${DCGDIR}
```

4.4.3.2 SR-IOV Networking

Execute the following commands on the compute hosts where SR-IOV feature is enabled:

```
cat /var/log/sriov.log > ${DCGDIR}/sriov_log.txt

/usr/sbin/nic_bind.sh -l | grep 'eth6\|eth7' > ⇒
${DCGDIR}/sriov_pf_driver.txt

/usr/sbin/nic_bind.sh -l | grep 'vfio-pci' > ⇒
${DCGDIR}/sriov_vf_driver.txt

grep 'intel_iommu=on\|iommu=pt' /proc/cmdline > ⇒
${DCGDIR}/sriov_kernel_parameters.txt
```

Note: The `eth6` and `eth7` interface names can differ on hardware platforms other than Dell R630.

4.4.4 NTP-Related Problems

For the NTP part on a vCIC:

```
ntpq -pn -c assoc

ps auxww|grep ntp > ${DCGDIR}/ntp_psaux.txt

which ntpd > ${DCGDIR}/ntp_which_ntpd.txt

cp /etc/ntp.conf ${DCGDIR}/ntp_conf.txt

ntpq -p > ${DCGDIR}/ntp_ntpq_p.txt

ntpq -c rv > ${DCGDIR}/ntp_ntpq_c_rv.txt

ntpq -c as > ${DCGDIR}/ntp_ntpq_c_as.txt
```

If authentication is enabled, enter:

```
cp /etc/ntp.keys ${DCGDIR}/ntp_keys.txt
```

For NTP on Extreme traffic switches (if present), enter:

```
ssh <Extreme Switch user>@$(host) show ntp association > ⇒
${DCGDIR}/extreme/switch-${host}-ntpassociations.txt

ssh <Extreme Switch user>@$(host) show ntp server > ⇒
${DCGDIR}/extreme/switch-${host}-ntpserver.txt

ssh <Extreme Switch user>@$(host) show ntp sys-info > ⇒
${DCGDIR}/extreme/switch-${host}-ntpsysinfo.txt
```

4.4.5 Extreme Switches

Perform the procedure described in the following sections for each switch in the system.

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

For each switch in the system, issue the following commands on the CIC:

```
export host=<switch IP address>

ssh <Extreme Switch user>@${host} 'show version' > ⇒
${DCGDIR}/extreme/switch-${host}-version.txt

ssh <Extreme Switch user>@${host} ⇒
'show log chronological' > ⇒
${DCGDIR}/extreme/switch-${host}.log

ssh <Extreme Switch user>@${host} ⇒
'show configuration' > ⇒
${DCGDIR}/extreme/switch-${host}.conf

show ports no-refresh

show vlan

show switch
```

Using SSH, log on to the management interfaces of all the Extreme switches, and issue the command:

```
ls internal-memory *.gz
```

For each core dump listed, issue the command:

```
scp2 vr "mgmtvrf" <core dump file> root@<Fuel (static)>:<datacollection-dir>/⇒
extreme/switch-<switch IP address>-<core dump file>
```

4.4.6 BSP

Perform the procedure described in the Data Collection Guideline for BSP, Reference [3].

4.4.7 HDS

If CEE is installed on the Ericsson Hyperscale Datacenter System (HDS), perform the steps described in the sections below:

4.4.7.1 CSS Configuration

To collect data about the Cloud SDN Switch (CSS), perform the following steps:

1. Use the following commands:

```
ovs-vsctl show > ${DCGDIR}/ovs_show.info

ovsdb-client dump > ${DCGDIR}/ovsdb.info
```

2. Use the following commands on a Compute blade:

```
ovs-ofctl dump-flows -O Openflow13 br-int > ⇒
${DCGDIR}/br_int_flow.info

ovs-ofctl dump-flows -O Openflow13 br-prv > ⇒
${DCGDIR}/br_prv_flow.info
```

4.4.7.2 System Routing Information

Use the following command to collect routing data:

```
ip route > ${DCGDIR}/routing.info
```

4.5 Storage

4.5.1 General

1. For data collection related to Cinder, execute the following commands on any CIC:

```
cinder service-list > ${DCGDIR}/cinder-service-list.txt


cinder snapshot-list > ⇒
${DCGDIR}/cinder-snapshot-list.txt


cinder type-list > ${DCGDIR}/cinder-type-list.txt


cinder extra-specs-list > ⇒
${DCGDIR}/cinder-extra-specs-list.txt


cinder availability-zone-list > ⇒
${DCGDIR}/cinder-az-list.txt


cinder-volume-usage-audit > ${DCGDIR}/cinder-usage.txt
```

```
cinder list --all-tenants > ${DCGDIR}/cinder-list.txt
```

2. For data collection related to Glance and Swift, execute the following commands on any CIC:

```
glance-control all status > ${DCGDIR}/glance-status.txt
```

```
ps -ef |grep swift > ${DCGDIR}/swift-process-status.txt
```

3. Enter the following commands on a CIC:

```
nova diagnostics <VMname/UUID> > ${DCGDIR}/ ⇒
nova-diagn.txt
nova console-log <VMname/UUID> > ${DCGDIR}/ ⇒
nova-consol.txt
```

4. Enter additional commands on CIC nodes and Compute hosts:

```
service open-iscsi status> ${DCGDIR}/iscsi-status.txt

iscsiadm -m session> ${DCGDIR}/iscsisessions.txt
ls -l /dev/disk/by-path> ${DCGDIR}/iscsidevices.txt
multipath -v3> ${DCGDIR}/multipathv3. txt
multipath –ll> ${DCGDIR}/multipathll.txt
```

4.5.2 Centralized Storage

This section only applies if the system has an EMC VNX storage device.

Note: The user executing the commands must have Navisec credentials set up.

1. Change to the data collection directory for EMC VNX:

```
cd ${DCGDIR}/emc
```

2. Issue the following commands for both Storage Processors:

```
/opt/Navisphere/bin/naviseccli -h <VNX SP IP>  ⇒
getagent ><VNX SP IP>-version.txt

/opt/Navisphere/bin/naviseccli -h <VNX SP IP> spcollect
```

3. To monitor the files on the devices, issue the command:

```
/opt/Navisphere/bin/naviseccli -h <VNX SP IP> ⇒
managefiles -list
```

4. After 20–30 minutes, a new file is displayed with the following format:

*<arrayserialnumber>*_SP*<A|B>*_*<date>*_*<time>*_*<spsignature>*_data.zip

To download the files, issue the command:
/opt/Navisphere/bin/naviseccli -h *<VNX SP IP>* manage files -retrieve -file *<VNX SP data file>*
/opt/Navisphere/bin/naviseccli -h *<VNX SP IP>* ⇒ managefiles -retrieve -file *<VNX SP data file>*

A prompt is displayed:

```
Files selected to be retrieved are
<VNX SP data file>
Do you want to continue (y/n)?
```

Press **y**, then **Enter** to perform the download.

4.5.3 Distributed Storage

This section only applies if EMC² ScaleIO disributed storage is used.

On each ScaleIO host, execute the following script:
/opt/emc/scaleio/*<scaleio_component>*/diag/get_info.sh
where *<scaleio_component>* has one of the following values:

- `mdm` for Meta Data Manager (MDM)
- `tb` for Tie-Breaker (TB)
- `sds` for ScaleIO Data Server (SDC)
- `sdc` for ScaleIO Data Client (SDC)

The script collects logs for all ScaleIO components running on the same host, therefore only execute the script once on each ScaleIO host.

4.6 Problems Related to Virtual Machines

4.6.1 Data Collection from Virtual Machines

The following commands apply to VMs running Linux OS deployed on CEE. When a command is not available on a specific Linux distribution, it can be omitted.

Use the following tools for a compute host:

- For SUSE Linux Enterprise Server (SLES), use the `supportconfig` tool.
- For Red Hat Enterprise Linux (RHEL), use the `sosreport` tool.

Syntax:
supportconfig

sosreport --batch [--name *<archive name>*] ⇒
[--ticket-number *<tr ref>*]

Issue the following commands in guest OS as `root` user and attach the output with the rest of the data. Commands from **uname -a** to **dmesg** can be omitted, if the `supportconfig` tool or the `sosreport` tool is used instead:

```
ip a

uname -a

df -a

fdisk -l

cat /proc/mounts

cat /proc/cpuinfo

cat /proc/meminfo

dmidecode

lspci

lsmod

dmesg
```

4.6.2 Data Collection from Compute Hosts Hosting Virtual Machines and CIC

Attach data from Console Log and diagnostics from VM. Collect the data from CIC:

```
nova console-log <instance name>|<UUID> > ${DCGDIR}/nova-consol.txt
nova diagnostics <instance name>|<UUID> > ${DCGDIR}/nova-diagn.txt
nova list --all-tenants --fields name,status,task_state,host,Networks,instance_name
```

Collect data from the compute host:

```
virsh list --all

virsh capabilities

virsh dumpxml <instance name or ID>
```

```
virsh domblklist <instance name or ID>
virsh domblkinfo <instance name or ID> <disk>
virsh domblkstat <instance name or ID> <disk>
virsh domiflist <instance name or ID>
virsh domifstat <instance name or ID> <tap interface>
virsh domif-getlink <instance name or ID> <tap interface>
virsh dominfo <instance name or ID>
virsh domstate <instance name or ID>
virsh dommemstat <instance name or ID>
virsh vcpuinfo <instance name or ID>
virsh vcpupin <instance name or ID>
virsh vcpucount <instance name or ID>
```

From a compute node, issue the command:

```
print free -m
```

To check hugepages use, print the following:

```
cat /sys/kernel/mm/hugepages/hugepages-2048kB/nr_hugepages
cat /sys/kernel/mm/hugepages/hugepages-2048kB/free_hugepages
cat /sys/kernel/mm/hugepages/hugepages-1048576kB/nr_hugepages
cat /sys/kernel/mm/hugepages/hugepages-1048576kB/free_hugepages
cat /sys/devices/system/node/node*/hugepages/hugepages-2048kB/nr_hugepages
cat /sys/devices/system/node/node*/hugepages/hugepages-2048kB/free_hugepages
cat /sys/devices/system/node/node*/hugepages/hugepages-1048576kB/nr_hugepages
cat /sys/devices/system/node/node*/hugepages/hugepages-1048576kB/free_hugepages
```

4.7 Hardware-Related Problems

Not applicable.

4.8 Finalizing Steps

Finalize specific data collection. Follow the steps in Section 5 on page 24.

5 Finalize Data Collection

5.1 Logs, Dumps and Configuration Files

Local logging

In case local logging is used for the Compute hosts, collect logs related to the following:

- CIC: See `cic-1` in the example below:

```
ssh <personal_user>@<cic_IP address>
sudo -i
ls -l /var/lib/glance/
cd /var/lib/glance/<directory created in Section 3.4 on page 7>
export DCGDIR=`pwd`
printenv DCGDIR
tar --exclude='/var/log/crash' -cvzf $DCGDIR/cic-1.tar.gz /var/log/
```

- Compute host: See `compute-0-2` in the example below:

```
ssh <personal_user>@<compute_IP address>
sudo -i
ls -l /var/lib/glance/
cd /var/lib/glance/<directory created in Section 3.5 on page 8>
export DCGDIR=`pwd`
printenv DCGDIR
tar --exclude='/var/log/crash' -cvzf $DCGDIR/compute-0-2.tar.gz ⇒
/var/log/
```

Remote logging

In case remote logging is used for Compute hosts, collect logs related to the following:

- CIC: See `cic-1` in the example below:

```
ssh <personal_user>@<cic_IP address>
sudo -i
ls -l /var/lib/glance/
cd /var/lib/glance/<directory created in Section 3.4 on page 7>
export DCGDIR=`pwd`
printenv DCGDIR
tar --exclude='/var/log/crash' --exclude='/var/log/remote' ⇒
-cvzf $DCGDIR/cic-1.tar.gz /var/log/
```

- Compute host: See `compute-0-2` in the example below:

Execute on the CIC:

```
tar -cvzf $DCGDIR/compute-0-2.tar.gz /var/log/remote/compute-0-2
```

5.2 Collect PM Report Files

Collect PM data on all vCICs:

```
ssh <personal_user>@<cic_IP address>

sudo -i

ls -l /var/lib/glance/

cd /var/lib/glance/<data collection directory on CIC>

export DCGDIR=$(pwd)

printenv DCGDIR

tar -cvzf $DCGDIR/$(hostname --short)_pm-xml.tgz
/var/cache/pmreports/
```

Where *<data collection directory on CIC>* is the directory created in Section 3.4 on page 7 in Step 2.

5.3 Finalize Mandatory Data Collection

1. After collecting all data, pack them by issuing the following commands:
   ```
   cd /

   tar -cvzf ${DCG}Printouts.tar.gz ${DCGDIR}
   ```

2. From Fuel, issue the following command:
   ```
   scp 'root@<hostname>:/var/lib/glance/DCG/*'/var/DCG/
   ```

   `DCG` is the folder name created for data collection, see Step 1, and *<hostname>* is the address of the CIC or compute node.

3. Repeat Step 2, for each CIC and compute node.

In the following printout from Fuel, `fuel node name` is *<hostname>*:

```
[root@fuel ~]# fuel node
id | status | name         | cluster | ip             |
---|--------|--------------|---------|----------------|-
5  | ready  | compute-0-2  | 1       | 192.168.0.24   |
4  | ready  | cic-5        | 1       | 192.168.0.23   |
3  | ready  | compute-0-3  | 1       | 192.168.0.22   |
6  | ready  | compute-0-7  | 1       | 192.168.0.25   |
1  | ready  | cic-4        | 1       | 192.168.0.20   |
2  | ready  | cic-6        | 1       | 192.168.0.21   |
```

Example 2 Fuel Printout for SCP Commands

A corresponding SCP command for CIC is:

```
scp 'root@cic-0-5:/var/lib/glance/DCG/*' /var/DCG/
```

Example 3 SCP Root Commands for cic-0-5

A corresponding SCP command for the compute host is:

```
scp 'root@compute-0-2:/var/DCG/*' /var/DCG/
```

Example 4 SCP Root Commands for compute-0-2

4. Transfer the resulting files from the system and provide it as part of the CSR, together with logs, dumps, and configuration files, as described in Section 5.1 on page 24.

5.4 Finalize Specific Data Collection

After collecting all data, pack them by issuing the commands:

```
cd /

tar -cvzf ${DC}.tar.gz ${DC}
```

Transfer the resulting file out of the system. Provide it as part of the CSR, together with logs, dumps, and configuration files, as described in Section 5.1 on page 24.

5.5 Remove Temporary Files

Delete the collected data from CIC and Compute hosts.

Example:

```
ssh <CEE administrator>@<CIC address>

rm -r $DCGDIR
```

6 Additional Information

6.1 How to Connect

CIC can be reached by using:

- CIC public IP addresses on VLAN `cee_om_sp`, and using IdAM username/password
- From Fuel. Use the command **`fuel node`**.

CEE Region has one CIC instance (cic-1) in case of Single Server and three CIC instances (cic-1, cic-2 and cic-3) in case of Multi-Server configurations with public IP addresses according to IP addresses allocated for CIC nodes in `cee_om_sp` network.

They have hostnames of the format `cic-<id>`, for example: `cic-2`

Compute hosts can be reached:

- From CIC
- From Fuel. Use the command **`fuel node`**.

They have hostnames of the format `compute-<shelf-id>-<blade id`, for example: `compute-0-3`.

More examples are provided inTable 5.

Table 5 Examples of Hostnames

Hostname	Description
`cic-2`	The CIC number (CIC-1, CIC-2, or CIC-3) is determined by config.yaml
`compute-0-5`	Compute host in shelf 0 (enclosure 0), device bay 5
`compute-1-10`	Compute host in shelf 1 (enclosure 1), device bay 10
…	Following the same pattern for further shelves
`compute-2-16`	Compute host in shelf 2 (enclosure 2), device bay 16

6.2 Description of Core and Kernel Crash Dump Data

The core dumps and Linux kernel crash dumps are binary, and represent a memory snapshot of the crashed process, VM, or Linux kernel.

Note: These files can contain sensitive information, and must not be distributed outside trusted parties without sanitizing possible password/key variables.

The alarm *Core Dump Generated* specifies the full path to the dump file.

The format of the core dump file name is `core.%h.%t.%e.%p`, which represents the following:

%h Hostname

%t Unix timestamp

%e Executable filename

%p PID of the process

The format of the crash dump file name is `vmcore-<timestamp>` with the timestamp generated in UTC.

6.3 Split Files before Adding to Trouble Report

Before adding the `tar.gz` file to the CSR as an enclosure, it must be split into pieces according to the appropriate enclosure limits.

```
split -d -b <piece>MB --verbose  ${DC}Printouts.tar.gz ${DC}Printouts.tar.gz.part
```

<piece> is less than the enclosure limit, for example 500 MB.

Pieces can be put together with `cat` command. Add this information to the CSR:

```
cat ${DC}Printouts.tar.gz.part.* ${DC}Printouts.tar.gz
```

Reference List

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

[2] *Atlas Troubleshooting Guideline*, 6/1553-CRA 119 1873/5

[3] *Data Collection Guideline for BSP*, 6/1543-APP 111 01