

Swift Store on VNX Activation Cloud Execution Environment

OPERATING INSTRUCTIONS

Copyright

© Ericsson AB 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	Description	1
1.2	Prerequisites	1
1.3	Procedure Overview	3
2	Preparation	3
3	Check Available Capacity on the VNX	5
4	Check OpenStack Quotas	6
5	Activate Storage on the VNX for Swift	9
6	Conclude Activation	10
Appendix		
7	Additional Information	13
7.1	List CIC and Compute Nodes	13
7.2	Check Write Cache Status	13
7.3	Configurable Storage Connector Parameters	15
Reference List		17





1 Introduction

This Operational Instruction (OPI) describes how to activate the feature Swift store on VNX and move the location of the Swift store from the local disks to the centralized storage (EMC VNX).

1.1 Description

By default, the Swift store is located on the local disks of the CIC hosts. This storage has capacity limitations due to the limited number and capacity of the local disks.

Part of the storage pool used for Cinder located on the VNX is therefore used for Swift store. Each Cloud Infrastructure Controller (CIC) will get its own Logical Unit Number (LUN) in the storage pool used by Cinder. This OPI describes the procedure for moving the existing Swift storage from the local disks to the VNX without downtime.

1.2 Prerequisites

1.2.1 Documents

Ensure that the following documents have been read:

- *IP and VLAN Plan*, updated with customer and site-specific values.
Note: All examples in this document use the default values from the document IP and VLAN plan, Reference [1]. The actual customer-defined addresses must be used when performing the steps in this document.
- *CEE Connectivity User Guide*

1.2.2 Tools and Equipment

A computer is required, that can be used to connect to the Cloud Execution Environment (CEE) using Secure Shell (SSH) protocol.



1.2.3 Conditions

Before starting this procedure, ensure that the following conditions are met:

- There are no errors reported for the VNX (for example check the EMC Unisphere Graphical User Interface).
- There are no alarms reported with `Critical` or `Major` severity in the CEE.
- The user of this OPI must be familiar with how to log in to all three CIC nodes and to the Fuel node from a remote location. For more information about these procedures, see the *CEE Connectivity User Guide* and Section 7.1 on page 13 in the Appendix.
- The IdAM credentials for remote CIC login are available. See the *CEE Connectivity User Guide*.
- The credentials for logging in as `ceedm` user with `sudo` privileges are available.

Note: All commands in this OPI (except logging in to the CIC node from a remote location) must be executed as user `ceedm`.

- The write cache must be enabled for each Storage Processor (SP) on the VNX. For more information see Section 7.2 on page 13.
- Ensure that no other maintenance activities are taking place at the same time.

1.2.4 Installation Data

Before starting this procedure, make sure that the following data is available:

Variable	Value
<code><sp_ip></code>	192.168.2.12 for Storage Processor A (SPA) ⁽¹⁾ 192.168.2.13 for Storage Processor B (SPB) ⁽¹⁾
<code><STORAGE.POOL.NAME></code>	This is the name of the storage pool used for Cinder as defined during VNX5400 SW installation. Use the same name that was noted down for the <i>Configuration File Guide</i> .
<code><size></code>	This is the planned initial size of the Swift Store on the VNX in GiB. The maximum initial size of the Swift Store on the VNX is 6000 GiB. Larger Swift Store sizes can be created later, as described in the document <i>Swift Store on VNX Expansion</i> .

(1) These values are valid if a certified configuration is used with default setup. Otherwise, the corresponding customer and site-specific values must be used according to the local IP and VLAN Plan.

1.3 Procedure Overview

Figure 1 gives an overview of the procedures covered by this OPI.

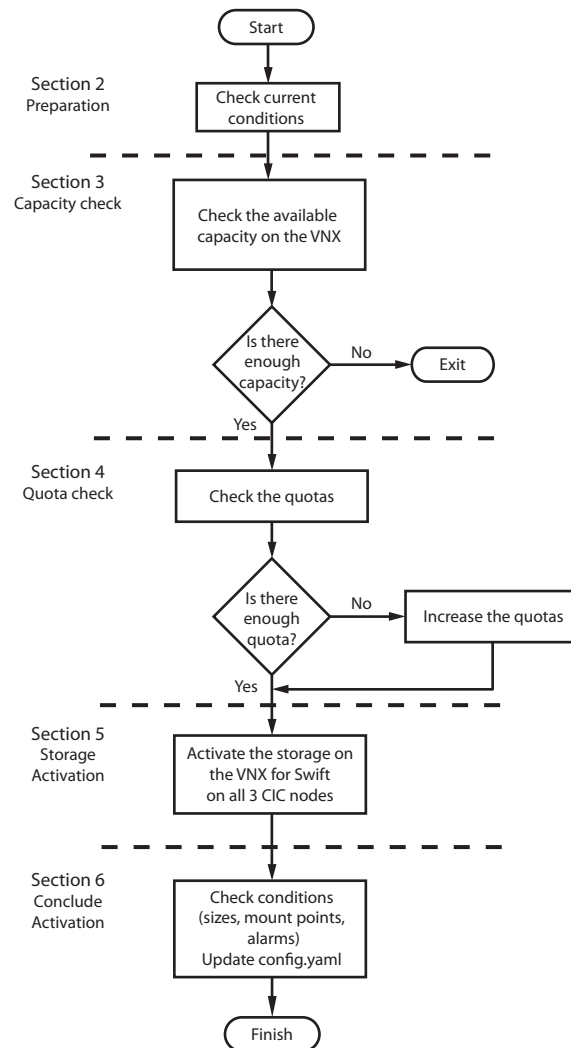


Figure 1 Procedure Overview

2 Preparation

This section describes how to prepare for creating the LUNs on the VNX and moving the Swift store.



Perform the following steps:

1. If the VNX activation is performed from a remote location, log onto one of the CICs by using IdAM credentials, then change to user `ceedm`, and log in to the Fuel node as described in the *CEE Connectivity User Guide*. Else, start with the next step below.
2. From the Fuel node, log onto one of the CIC nodes as user `ceedm`.
3. Print the properties of the logical volume for the Swift store by issuing the following command:

```
sudo lvdisplay image
```

Note: The path is located in the line starting with “LV Path”, the size of the Swift store is located in the line starting with “LV Size”. Note down these values for later use.

Example 1 shows a printout of the `sudo lvdisplay image` command:

```
ceedm@cic-1:~# sudo lvdisplay image
--- Logical volume ---
LV Path                /dev/image/glance
LV Name                 glance
VG Name                 image
LV UUID                 crQyfr-r8y9-99qE-Iydo-7LPw-rhso-LDTeeC
LV Write Access         read/write
LV Creation host, time ,
LV Status                available
# open                  1
LV Size                 649.91 GiB
Current LE              20797
Segments                3
Allocation               inherit
Read ahead sectors      auto
- currently set to      256
Block device            252:4
```

Example 1 *Printout of the `sudo lvdisplay image` Command*

4. Check that the file `/etc/backend_storage_connector.conf` exists, and that it contains the default values as described in Section 7.3 on page 15.
5. Repeat Step 2, Step 3, and Step 4 in Section 2 on page 3 for the other two CIC nodes.

Note: The displayed volume sizes must be the same on the three CIC nodes. If they are not equal, note down the highest value and keep it for later use.

6. Continue with the procedures in Section 3 on page 4.



3 Check Available Capacity on the VNX

This section describes how to verify that there is enough capacity available on the VNX to be able to move the Swift store there.

1. Ensure that you are logged onto one of the CIC nodes as `ceedadm`.
2. Check the “Available Capacity” of the storage pool used by the `cinder` service on the VNX by issuing the following command:

```
/opt/Navisphere/bin/naviseccli -h <sp_ip>⇒
storagepool -list -name <STORAGE.POOL.NAME>
```

Note: For the `<sp_ip>` variable use the IP address of any of the two storage processors.

Example 2 shows a partial printout with the relevant “Available Capacity” information:

```
ceedadm@cic-1:~# /opt/Navisphere/bin/naviseccli -h 192.168.2.12⇒
storagepool -list
Pool Name: cinderpool
[...]
Raw Capacity (GBs): 19506.114
[...]
User Capacity (GBs): 15296.818
[...]
Consumed Capacity (GBs): 4639.293
[...]
Available Capacity (GBs): 10657.525
Percent Full: 30.328
[...]
Total Subscribed Capacity (GBs): 4639.293
Percent Subscribed: 30.328
Oversubscribed by (Blocks): 0
Oversubscribed by (GBs): 0.000
[...]
```

Example 2 Available Capacity on the VNX

Note: Although in the printout the available capacity is displayed in “GBs”, in fact the correct unit of measure for the displayed capacity is “GiBs”.

3. Check if the “Available Capacity” is sufficient, and proceed according to the result as follows:
 - If the “Available Capacity” is at least three times as large as the planned new size of the Swift storage for one CIC on the VNX, continue with the procedures in Section 4 on page 6. This triple size for the Swift



storage must be available on the VNX because each CIC must be extended by the same amount of storage capacity.

- If the “Available Capacity” is less than three times the planned new size of the Swift storage for one CIC on the VNX, then the available capacity on the VNX must be increased first. This is not in the scope of this document. Consult the next level of maintenance support.

4. Continue with the procedure in Section 4 on page 6.

4 Check OpenStack Quotas

This section describes how to check the OpenStack quotas for the “admin” project and how to expand them if needed.

1. Ensure that you are logged onto one of the CIC nodes as `ceeadm` user.
2. Print the project list by issuing the following command:

```
openstack project list
```

```
ceeadm@cic-2:/root$ openstack project list
+-----+-----+
| ID                               | Name      |
+-----+-----+
| 3052cafca2e14f85b8f02263025d2a8f | admin     |
| c16e42d79bfb406fb4730fa8bdbb6d8f | services  |
+-----+-----+
```

Example 3 Openstack Project List

Note: Identify the project with the name “admin” and note down its ID (`<project_id>`) for later use.

3. Print the quota usage by issuing the following command:

```
cinder quota-usage <project_id>
```

Example 4 is a printout of the following `cinder quota-usage` command:
cinder quota-usage 3052cafca2e14f85b8f02263025d2a8f



```
ceeadm@cic-1:~# cinder quota-usage =>
3052cafca2e14f85b8f02263025d2a8f
```

Type	In_use	Reserved	Limit
[...]	[...]	[...]	[...]
gigabytes	3100	0	10000
[...]	[...]	[...]	[...]
volumes	4	0	100
[...]	[...]	[...]	[...]

Example 4 *quota-usage Command Printout*

Note: In the `cinder` command printouts, the capacity values are given in “GBs”, but in fact the correct unit of measure for the same values are “GiBs”.

4. Verify the quota types “gigabytes” and “volumes” as follows:
 - For the type “volumes”, the difference between “Limit” and “In_use” must be at least 3.
 - For the type “gigabytes”, the difference between “Limit” and “In_use” must be at least three times as big as the intended Swift store size for a single CIC. For example if the intended Swift store size for one CIC is 1000 GiB, the difference between “Limit” and “In_use” must be at least 3000 GiB.

The example printout in Step 3 in Section 4 on page 6 shows a disposition of 96 volumes and 6900 GiB of capacity.

After calculating the quotas, perform the relevant one of the following actions:

- If enough quotas are available for the activation, then continue with the procedures in Section 5 on page 8.
 - If the “gigabytes” quota must be increased continue with Step 5 in Section 4 on page 6.
 - If the “volumes” quota must be increased continue with Step 6 in Section 4 on page 6.
5. If the quota “gigabytes” needs to be increased, issue the following command:

```
cinder quota-update <project_id> --gigabytes
<new_g_limit>
```

where



- `<new_g_limit>` must be equal to or greater than:

$[(3 * \text{<size>}) + \text{<current_use>}]$.

Here `<size>` is the planned new size of the Swift storage and `current_use` is the “in_use” value of “gigabytes” from Step 3 in Section 4 on page 6.

- `<project_id>` is the project ID of the project admin.

Example 5 shows how to increase the gigabytes quota by 99, from 10000 GiB to 10099 GiB for the project admin, and an example printout:

```
ceeadm@cic-1:~# cinder quota-update =>
3052cafca2e14f85b8f02263025d2a8f --gigabytes 10099
```

Property	Value
backup_gigabytes	1000
backups	10
gigabytes	10099
per_volume_gigabytes	-1
snapshots	10
volumes	100

Example 5 cinder quota-update— 'gigabytes'

6. If the quota “volumes” needs to be increased, issue the following command:

```
cinder quota-update <project_id> --volumes <new_vol_
limit>
```

where

- `<new_vol_limit>` is at least three volumes more than the current value,
- `<project_id>` is the project ID of the project admin.

Example 6 shows how to increase the “volumes” quota by 3, from 100 to 103, for the project admin:

```
ceeadm@cic-1:~# cinder quota-update 3052cafca2e14f8
5b8f02263025d2a8f --volumes 103
```

Example 6 cinder quota-update — volumes

7. Continue with the procedure in Section 5 on page 8.



5 Activate Storage on the VNX for Swift

This section describes how to activate the storage on the VNX for Swift.

1. Ensure that you are logged onto one of the CIC nodes as `ceeadm`.
2. Start a “screen” session with the following command:

```
sudo -E screen
```

3. Press **Space** or **Return** after you have read the instruction on the screen.

Note: The `screen` command starts a session, that is independent from the current terminal window. Even if the terminal window crashes or is ended by other means, it is possible to reconnect to the session by using the command `sudo -E screen -x` from another terminal on the same host.

4. To expand and activate the Swift storage on the VNX, issue the following command:

```
backend-storage-connector activate <path>=>
<size>
```

The arguments are explained below:

Variable	Comment
<path>	This is the path of the logical volume (“LV Path”) noted down in Step 3 in Section 2 on page 3 .
<size>	<p>This is the planned initial Swift store size on the VNX for each CIC in GiB. The maximum initial size of the Swift Store on the VNX is 6000 GiB. Larger Swift Store sizes can be created later, as described in the document <i>Swift Store on VNX Expansion</i>.</p> <p>Provide a size in GiBs, that is equal to or larger, than the largest value noted down during the preparation in Section 2 on page 3. The activation does not work if a smaller size is specified. ⁽¹⁾</p>

(1) All CIC nodes must have the same storage capacity for Swift. Use the same value of <size> for each CIC.

Note: The execution time is approximately 90 minutes for transferring 600GiB.

The successful job is indicated by the final status information message “Success.”, as shown in Example 7 below:



```
root@cic-1:~# backend-storage-connector activate=>
/dev/image/glance 650
[...]
<ts> INFO: Updated fstab with option "_netdev,..."
<ts> INFO: Extending volume group: image with...
<ts> INFO: Move device /dev/sda7. This may take a while.
<ts> INFO: Move device /dev/sdb2. This may take a while.
<ts> INFO: Successfully disconnected all local...
<ts> INFO: Extending logical volume /dev/imag...
<ts> INFO: Growing xfs filesystem of logical...
<ts> INFO: Success.
root@cic-1:~#
```

Example 7 Expand and Activate the Swift Store on the VNX

5. If the execution fails, the file `/var/backend-storage-connector.fail` is created. Before another attempt at activation, remove the `.fail` file using the following command:
`rm /var/backend-storage-connector.fail`
6. End the screen session by pressing **CTRL+D**.
7. Repeat all the steps in Section 5 on page 8 for the other two CIC nodes and then continue with the procedures in Section 6 on page 10.

Note: A log file is generated under the following path on each CIC:

`/var/log/backend-storage-connector.log`

6 Conclude Activation

This section describes how to confirm that the activation has taken place.

1. Ensure that you are logged onto one of the CIC nodes as `ceeadm` user.
2. Check the logical volume size for the Swift store by issuing the following command on one of the CIC nodes:

`sudo lvdisplay image`

“LV Size” displays the size of the Swift store in GiB. The size must match the chosen value for `<size>`.

Example printout:



```

ceeadm@cic-1:~# sudo lvdisplay image
--- Logical volume ---
LV Path                /dev/image/glance
LV Name                 glance
VG Name                 image
LV UUID                 crQyfr-r8y9-99qE-Iydo-7LPw-rhso-LDTeEC
LV Write Access         read/write
LV Creation host, time  ,
LV Status                available
# open                  1
LV Size                 649.97 GiB
Current LE              22396
Segments                4
Allocation              inherit
Read ahead sectors      auto
- currently set to     256
Block device            252:4

```

Note: In the example the “LV Size” of “649.97 GiB” does not match the defined 650GiB exactly, but the difference is only caused by rounding inaccuracy.

3. Repeat Step 1 and Step 2 in Section 6 on page 10 for the other two CIC nodes and then continue with Step 4.
4. Issue the following command on one of the CIC nodes:

cinder list

```

ceeadm@cic-1:~# cinder list
+-----+-----+-----+-----+-----+-----+
| ID      | Status | Display Name          | Size | [...] |
+-----+-----+-----+-----+-----+-----+
| 49[...]6 | in-use | CEE+cic-1+/dev/image/glance | 650 | [...] |
| 8d[...]1 | in-use | CEE+cic-2+/dev/image/glance | 650 | [...] |
| ac[...]0 | in-use | CEE+cic-3+/dev/image/glance | 650 | [...] |
+-----+-----+-----+-----+-----+-----+
ceeadm@cic-1:~#

```

Example 8 cinder list Command Printout

For each CIC one volume must exist with a “Display Name” starting with “CEE+”, followed by the CIC name and the logical volume path. The size must match the value *<size>*, which has been provided in Section 5 on page 8.

5. From the CIC node log in to the Fuel node by using SSH as user `ceeadm`.
6. On the Fuel node, edit the `config.yaml` file under `/mnt/cee_config` with root privileges (for example with the `sudo vi /mnt/cee_config/config.yaml` command) as follows:



- a. Under “Swift:” > “swift_on_backend_storage:” > “activation_mode” change the value from “manual” to “automatic”.
- b. Under “Swift:” > “swift_on_backend_storage:” > “lun_size” set the value to the *<size>* (as it has been set in Step 4 in Section 5 on page 8).

Note: These modifications are necessary, so that in case of a rollback and CIC repair, the Swift store is set up again in the VNX automatically with the correct size. Once the Swift storage is moved to the VNX, it cannot be reversed back to the local disk.

Example 9 shows the relevant `config.yaml` hierarchy.

```
ericsson:
  swift:
    swift_on_backend_storage:
      type: centralized
      activation_mode: automatic
      lun_size: 650GiB
```

Example 9 Editing config.yaml

7. Save the changes in the `config.yaml` file.



Appendix

7 Additional Information

This section describes the following:

- How to list the hostnames and addresses of the CIC nodes
- How to check write cache status of the SPs on the VNX
- Configurable storage connector parameters

7.1 List CIC and Compute Nodes

To display the hostnames and IP addresses of the CIC nodes issue the following command, while being logged in to the Fuel node:

```
sudo fuel node
```

Note: From a remote location only the CIC servers can be reached. For more information see the *CEE Connectivity User Guide*.

Example 10 is a partial printout that shows only the relevant CIC nodes:

```
[ceeadm@fuel ~]# sudo fuel node
```

id	status	name	...	ip	...
---	-----	-----	...	-----	...
7	ready	compute-0-3	...	192.168.0.20	...
8	ready	cic-1	...	192.168.0.21	...
9	ready	cic-2	...	192.168.0.22	...
12	ready	cic-3	...	192.168.0.25	...
10	ready	compute-0-10	...	192.168.0.23	...
11	ready	compute-0-2	...	192.168.0.24	...

Example 10 CIC and Compute Node Printout

7.2 Check Write Cache Status

To check if the write cache is enabled for the SPs on the VNX and enable it in case it is not, follow these steps:



1. Log in to a CIC node with `personal-user` by using SSH and change to user `ceeadm` with the `su - ceeadm` command.
2. To check if the write cache is enabled for the SPs issue the following command:

```
/opt/Navisphere/bin/naviseccli -h <sp_ip> getcache
```

Note: For the `<sp_ip>` variable use the IP address of any of the two storage processors.

Example 11 shows a scenario where the write cache is disabled for the SPs.

```
ceeadm@cic-1:~# /opt/Navisphere/bin/naviseccli -h⇒
192.168.2.12 getcache
SP Read Cache State: Enabled
SP Write Cache State: Disabled
Cache Page size (KB): 8
Write Cache Mirrored: YES
Low Watermark: 60
High Watermark: 80
SPA Cache Pages: 250303
SPB Cache Pages: 250304
Unassigned Cache Pages: 0
Read Hit Ratio: N/A
Write Hit Ratio: N/A
Prct Dirty Cache Pages = 0
Prct Cache Pages Owned = 0
SPA Read Cache State: Enabled
SPB Read Cache State: Enabled
SPA Write Cache State: Disabled
SPB Write Cache State: Disabled
System Buffer (spA): 7550 MB
System Buffer (spB): 7550 MB
SPS Test Day: Sunday
SPS Test Time: 03:00
SPA Physical Memory Size (MB) = 12288
SPB Physical Memory Size (MB) = 12288
```

Example 11 SPA Information Printout — SP Write Cache Disabled

3. Check the given value for the “SP Write Cache State”.
 - a. If the value is `Enabled`, the write cache is enabled for both storage processors, and the procedure is complete.
 - b. If the value is `Disabled`, the write cache is disabled for both of the SPs. Continue with Step 4.
4. To enable the write cache option on both storage processors, issue the following command:



```
/opt/Navisphere/bin/naviseccli -h <sp_ip> setcache -wc
1
```

Note: This command has no printout.

5. To verify that the write cache has been enabled, issue the following command:

```
/opt/Navisphere/bin/naviseccli -h <sp_ip> getcache
```

Example 12 shows a scenario where the write cache is enabled for the SPs.

```
ceeadm@cic-1:~# /opt/Navisphere/bin/naviseccli -h=>
192.168.2.12 getcache
SP Read Cache State: Enabled
SP Write Cache State: Enabled
Cache Page size (KB): 8
Write Cache Mirrored: YES
Low Watermark: 60
High Watermark: 80
SPA Cache Pages: 250303
SPB Cache Pages: 250304
Unassigned Cache Pages: 0
Read Hit Ratio: N/A
Write Hit Ratio: N/A
Prct Dirty Cache Pages = 0
Prct Cache Pages Owned = 0
SPA Read Cache State: Enabled
SPB Read Cache State: Enabled
SPA Write Cache State: Enabled
SPB Write Cache State: Enabled
System Buffer (spA): 7550 MB
System Buffer (spB): 7550 MB
SPS Test Day: Sunday
SPS Test Time: 03:00
SPA Physical Memory Size (MB) = 12288
SPB Physical Memory Size (MB) = 12288
```

Example 12 SPA Information Printout — SP Write Cache Enabled

7.3 Configurable Storage Connector Parameters

Some storage connector parameters can be configured according to local requirements in the file `/etc/backend_storage_connector.conf`. The configurable parameters and the default values of the parameters are listed in Table 1.

**Table 1** Storage Connector Parameters

Parameter	Default value	Description
<volume_wait_timeout>	1800	Maximum time to wait in seconds until a created cinder volume has status <code>available</code>
<volume_wait_time>	10	Scan interval in seconds checking for volume status
<volume_deletion_timeout>	600	Maximum time to wait in seconds until the deleted cinder volume or volumes are not found anymore using Openstack commands ⁽¹⁾
<volume_deletion_time>	10	Scan interval in seconds checking for cinder volume deletion
<retry_vol_create_on_error>	True	If value <code>True</code> is set, volumes in error status are deleted and re-created ⁽¹⁾⁽²⁾
<log_path>	/var/log/backend-storage-connector.log	Path to log file
<astute>	/etc/swift_backend_astute.yaml	Path to astute file
<use_multipath>	True	
<supported_storage_types>	centralized	
<external_path_list>	/dev/mapper	
<swift_logical_volume_path>	/dev/image/glance	
<supported_logical_volume_path>	/dev/image/glance	

(1) Used only in repair mode

(2) In repair mode volumes are deleted and re-created. Due to the latency of the backend storage system, when deleting volumes, volume creation can fail if the backend storage system has limited capacity. In this case, the created volumes end up in error status. The number of retries and the time interval are automatically calculated depending on the size of the volume.



Reference List

- [1] *IP and VLAN plan, 2/102 62-CRA 119 1862/5 Uen*