

CEE Update and Rollback Guide

Cloud Execution Environment 6

Copyright

© Ericsson AB 2018. 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 Group	2
1.3	Prerequisites	2
1.4	Limitations	5
2	Overview	8
2.1	Component Update Descriptions	10
2.2	Procedure Durations	14
3	Update Orchestration Configuration File	15
4	Procedure	17
4.1	Mandatory Preparation Stage	17
4.2	Update Stage	19
4.3	Common Concluding Stage	27
5	Error Handling	29
5.1	Error Handling for Failed CSC Update	29
5.2	Error Handling for Failed Orchestrated Update	29
5.3	Error Handling for Failed Atlas Upgrade	40
6	Additional Operations	41
6.1	Checking Update State	41
6.2	Checking VM State	41
6.3	Listing Nodes	42
6.4	Identifying the Active and Cold Standby Fuel Hosts	42
6.5	Insert Forwarding Rule on vCICs	43
6.6	Generating New SSH Key for Compute Host Hosting vCIC or vFuel	43
Appendix		
7	update_groups.yaml Examples	45
8	NIC Firmware Version Check and Upgrade	49
Reference List		51





1 Introduction

The document describes the generic flow of the update procedure in the CEE region. This document is designed to be the starting point when performing update of the CEE software, or the additional components of the CEE region (such as Atlas or CSS), or both.

For the release-specific update procedure of the CEE region, refer to the CEE <release> SW Update and Rollback document for the specific release.

1.1 Scope

This document describes the following:

- The generic update flow for the update of the entire CEE region, or one or multiple additional components
- The order of the procedures that must be followed, regardless of which components are updated
- The available rollback options for the individual update procedures and their combinations

Note: Update of component or components of CEE is only supported to the component versions and on the update paths described in the Product Revision Information for Cloud Execution Environment document for the specific CEE release, Reference [4]

This document describes the procedures for the update and rollback of the following components:

- vCICs
- vFuel
- Compute hosts
- ScaleIO servers
- CSS
- HDS Agent

Although they are included in the flow description, the update and rollback procedures for the following components are described in separate documents:

- The following CSC components:
 - SDNc Fuel plugin



- L2GW Fuel plugin
- BGPVPN Fuel plugin

— Atlas

1.2 Target Group

This document is aimed at skilled professionals from the following groups:

- Customer O&M personnel
- Support organization personnel
- Ericsson personnel updating CEE at customer sites

1.3 Prerequisites

1.3.1 Tools and Equipment

This section describes the tools needed for some or all of the procedures described in this document.

1.3.1.1 User Access

root access to vFuel is required. The procedures below can only be executed as root.

1.3.1.2 Hardware and Software

The procedures in the document have the following hardware prerequisites:

- Sufficient disk space is required on an outside File Transfer Protocol Secure (FTPS) server for storing the backup of CEE components for rollback purposes. The amount of storage required depends on the size of the active vFuel Virtual Machine (VM) and the Virtual Cloud Infrastructure Controller (vCIC) VMs.

The size of the compressed vCIC and vFuel image backups is approximately the half of the image sizes. For example, with an average vCIC size of 500 GB, a minimum of 800 GB of free storage is required for the three vCIC and the vFuel image backups.

- Sufficient disk space is required in the /var/tmp directory on vFuel for the CEE software release tarball. If there is not enough free space in the /var/tmp directory, free up space before starting the procedure. The size of the CEE image is approximately 4.5 GB.



The amount of available space in the `/var/tmp` directory can be verified with the following command:

```
df -h /var/tmp/
```

The procedures in the document have the following firmware prerequisites:

- In HDS systems using servers with Intel x710 NICs assigned to DPDK, the firmware version of the X710 NICs must be 6.0.1 or above. To verify and if applicable, update firmware, see Section 8 on page 49.

Before starting the update, make sure that the following software are available:

- CEE software release tarball. For more information, refer to the [Configuration File Guide](#).
- Depending on the components to be updated, the new version of the following artifacts or packages must also be available:

Component	Package	Included in
vCIC	CEE software release tarball	CEE Container
vFuel	CEE software release tarball	CEE Container
Compute hosts	CEE software release tarball	CEE Container
Ericsson HDS Agent	ericsson_hds_agent-<version>.noarch.rpm	HDS Container
ScaleIO	scaleio-2<version>.noarch.rpm	CXC1740177_4_<release>.tar in the ScaleIO Container
CSS	ericsson_css-<version>.noarch.rpm	CSS Container

The required software can be downloaded from the SW Gateway. If encountering issues during the download procedure, contact the next level of support.

- A prepared `update_groups.yaml` file must be available before update. The template for `update_groups.yaml` is available in the CEE software release tarball. For more information, see Section 3 on page 15.
- The `update_orchestrator.sh` script, packaged with the CEE software release tarball.



1.3.1.3 Remote FTPS Server for Storing Backups

For rollback purposes, the vCIC and vFuel images and additional files must be backed up on a remote server. The remote server must fulfill the following requirements:

- Sufficient disk space for the backup files on external storage node, calculated on the basis of vCIC and vFuel VM sizes
- External connectivity through at least 1 Gbps Ethernet
- Installed and operational Ubuntu Linux 14.04 OS
- vsftpd FTPS server, with the following configuration:
 - /etc/vsftpd.conf file updated with the following values:


```
# Turn ON SSL
ssl_enable=YES
allow_anon_ssl=NO

# Use encryption for data
force_local_data_ssl=YES

# Use encryption for authentication
force_local_logins_ssl=YES

ssl_tlsv1=YES
ssl_sslv2=NO :
ssl_sslv3=NO
ssl_ciphers=HIGH

local_max_rate=41943040
```

1.3.2 Data

The following information must be available:

- Name and IP addresses of the compute hosts hosting the active and cold standby vFuel VM and the vCICs. To collect this data, perform the procedure described in Section 6.3 on page 41 and Section 6.4 on page 42.
- Management IP address of the compute hosts hosting the active vFuel VM and the vCICs. This information can be collected by executing the following command on the compute host:


```
ifconfig br-mgmt
```

 The management IP address of the host is listed as `inet addr:` in the printout.

An example of the printout is the following:

```
root@compute-0-1:~# ifconfig br-mgmt
br-mgmt    Link encap:Ethernet  HWaddr ca:59:75:45:4c:45
```




```
inet addr:192.168.2.23 Bcast:0.0.0.0 =>
Mask:255.255.255.128
```

- IP address and credentials to the external FTPS server described in Section 1.3.1.3 on page 3.

1.3.3

Conditions

The following conditions apply to all procedures described in this document:

- The environment must be healthy. Perform a health check as described in [Health Check Procedure](#).
- Limitations and workarounds for the update and rollback procedures must be known, refer to the Limitations and Workarounds for Cloud Execution Environment (CEE), Reference [2].

The following conditions apply to the different phases of the update procedures:

Phase	Conditions
Update	There must be no active alarms in the system when starting the update process.
Rollback	<p>The rollback procedure requires a backed up copy of the following:</p> <ul style="list-style-type: none"> • The vCIC VM images with the respective XML files and XML templates • The active vFuel VM image, and dump XML file generated from the vFuel VM image by the user <p>For more information, see Section 4.1 on page 17.</p>

The individual procedures can have additional conditions. See the relevant subsections of Section 2.1 on page 10 for any additional conditions of the individual procedures.

1.4

Limitations

Note: Update and rollback with Tightly Integrated SDN (SDN TI) is supported with limitations. Refer to the document Limitations and Workarounds for Cloud Execution Environment (CEE) <release>, Reference [2].

The following limitations apply to all procedures described in this document:



- During the procedure, `Fuel failed` and `CIC failed` alarms are expected to be issued. This is an intended behavior and the alarms cease after the affected nodes become operational again.
- No changes can be performed on the Virtualized Resources (VRs) (Compute, Networking and Storage) through the OpenStack REST APIs from the start of the backup procedure until successful update or rollback of all CEE components.

Table 1 shows the limitations that apply to the individual procedures:



Table 1 Limitations for Update and Rollback Procedures

Procedure	Limitation
Update	<ul style="list-style-type: none"> When the CEE software update is running, the OpenStack API service is unavailable for about one minute during the reboot of each vCIC. CM-HA fencing is automatically turned off during CEE software update to prevent unnecessary invocation. If fencing was initially enabled and serial update method is used, fencing is turned off each time a compute host is updated, and turned back on once the update of that host is successfully finished. If fencing was initially enabled and parallel update method is used, fencing is turned off when starting the parallel update, and turned back on once the update of all hosts is successfully finished. Reconfiguration of the CSS CPU reservation mode is not allowed during the update procedure. Changing the <code>css_mode</code> parameter in the <code>config.yaml</code> can cause severe system malfunction. For example, changing <code>normal-perf</code> to <code>high-perf</code> CSS mode can result in insufficient CPU resource reservation for the tenant VMs on the affected compute hosts, making them unable to boot up after reconfiguration. During the execution of the <code>update_orchestrator.sh</code> script, before the update of a compute host, the VMs located on the compute host are acted upon according to the defined High Availability (HA) policy. If multiple compute hosts are updated in one session, VMs can be migrated multiple times, and restarted multiple times during the procedure, depending on the defined migration policy. If compute hosts are updated one by one, the amount of multiple migration of VMs can be decreased. For unmanaged VMs, manual migration can be necessary. For more information on HA policies, refer to OpenStack Compute API in CEE. <p>If VMs are migrated during the update procedure, rollback is not possible.</p>
Rollback	<ul style="list-style-type: none"> Rollback is only possible if no VMs are migrated during update. For more information, see Section 5.2.1 on page 30. Due to the transfer of the backup files between the external server and the CEE region, bandwidth is affected, and transfer time must be taken into consideration. Rollback of the compute hosts hosting vCIC or vFuel is not possible. If update of any of these compute hosts fails, redeployment of CEE can be necessary. Rollback of compute hosts does not preserve data stored on the ephemeral disks of the VMs hosted on the rolled back compute host.



2 Overview

The CEE Update framework has the following use cases:

- Performing an update of the entire CEE region, in one or multiple phases or maintenance windows, affecting all hosts and components. Rollback is only possible with the manual Fuel and vCIC rollback procedures. The specific procedure for the current release is described in CEE <release> SW Update and Rollback.
- Performing an update of one or more components or additions of CEE to a newer, supported version without updating the entire CEE region. Some or all hosts can be affected, depending on the component to be updated. The affected hosts are listed in the description sections for the individual updates. For the description of the individual updates, see Section 2.1 on page 10.

Update of any component of CEE is only supported to the component versions described in the Product Revision Information document for the relevant CEE release, Reference [4]

The update is orchestrated using the `update_groups.yaml` file, and is executed using the `update_orchestrator.sh` script, unless described otherwise at the procedure description. All nodes are restarted during the procedure; however, it is possible to perform the update in multiple sessions, by preparing multiple versions of the `update_groups.yaml` and executing the update script multiple times. For more information on update orchestration configuration, see Section 3 on page 15.

The update procedure for any component or combination of components follows the flow shown in Figure 1:

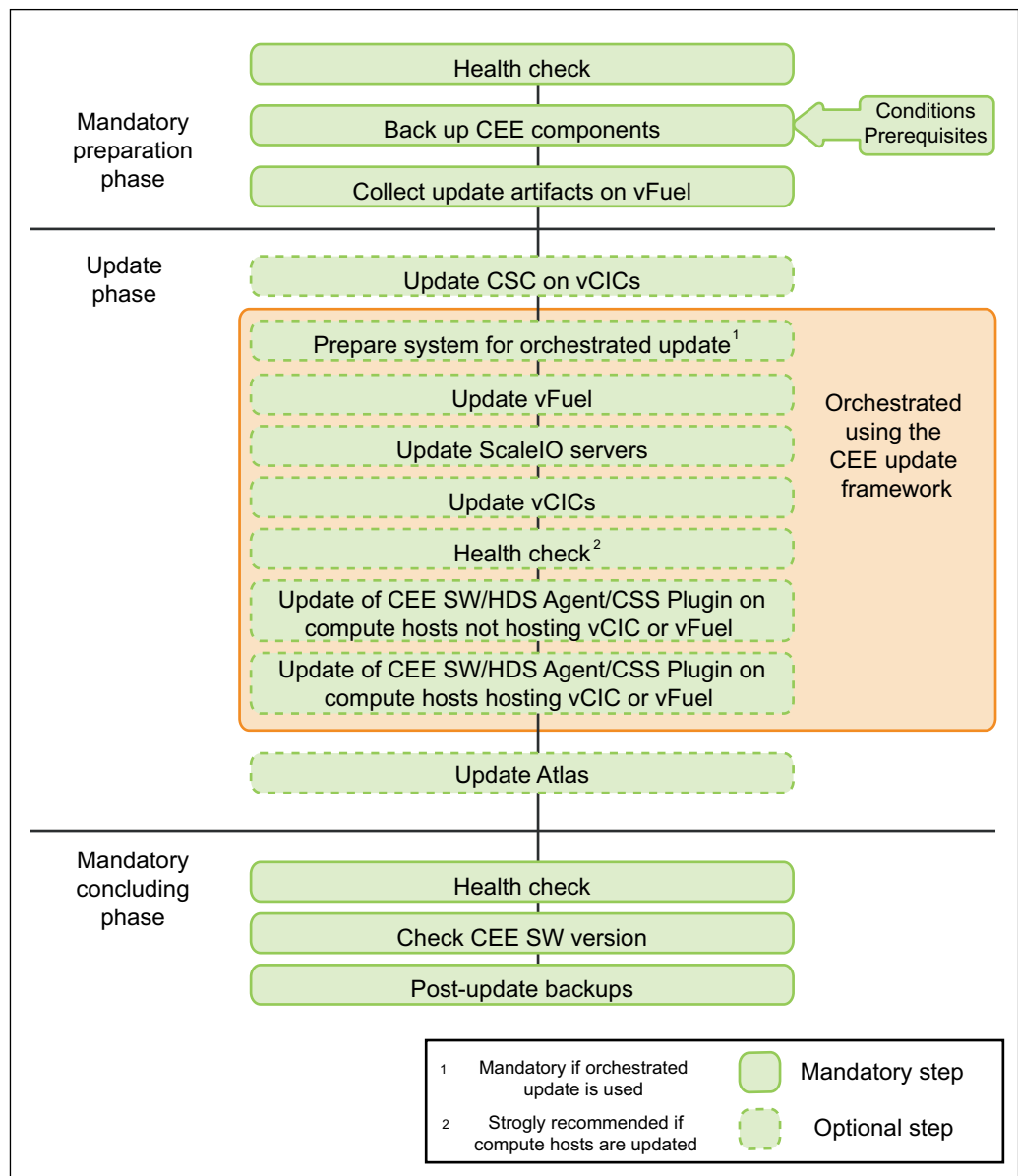


Figure 1 Component Update Order

Note: Also consider the conditions to the procedures, see Section 1.3.3 on page 5.

Update consists of the following phases:

Mandatory Preparation Stage

This phase consists of the following:

- Health check
- Creating backup of the CEE components



- Collecting artifacts required for the update on vFuel

Procedures are described in Section 4.1 on page 17.

Component Update Stage

This phase consists of the following:

- Update of external components (such as CSC or Atlas) using manual procedures
- Orchestrated update of CEE components using the CEE update framework

For more information on component update, see Section 2.1 on page 10.

Mandatory Concluding Stage

This phase consists of the following:

- Health check
- The CEE and component version is verified on all nodes.
- The required backup procedures are performed.

The procedures are described in Section 4.3 on page 27.

2.1 Component Update Descriptions

2.1.1 CSC Update

Note: This procedure is only applicable to CEE regions using SDN TI.

Update of CSC includes the following:

- SDNc Fuel plugin
- L2GW Fuel plugin
- BGPVPN Fuel plugin

The update of the CSC Fuel plugins is a manual procedure, not orchestrated by the update orchestrator script. For more information, refer to the CSC document Cloud SDN Upgrade and Rollback, Reference [1].

Affected Nodes

- vCICs



Limitations

For any limitations, refer to the CSC document Cloud SDN Upgrade and Rollback, Reference [1].

2.1.2 vFuel Update

vFuel is updated automatically by the update orchestrator script.

Affected Nodes

- vFuel

Orchestration Options

If the update is to be interrupted after updating vFuel, the `update_orcestrator.sh` must be run with the `--exit-after-fuel-update` switch.

The required steps for the update of the component are described in Section 4.2.4 on page 22.

2.1.3 ScaleIO Update

Note: This procedure is only applicable to CEE regions using managed ScaleIO.

The ScaleIO servers are updated automatically by the update orchestration script.

Affected Nodes

- ScaleIO servers
- vCICs
- Compute hosts, if ScaleIO Data Clients are updated

Orchestration Options

If only the Fuel plugins are updated, run the orchestrator script using the `--plugin-update` option. This option skips the vFuel update step.

For updating the ScaleIO nodes, `serial` update mode must be used.

The required steps for the update of the component are described in Section 4.2.5 on page 23.



2.1.4 vCIC Update

Affected Nodes

- vCICs

Orchestration Options

When updating vCICs, `serial` update mode must be used.

The required steps for the update of the component are described in Section 4.2.6 on page 23.

2.1.5 Compute Host Update

Compute hosts are updated automatically by the orchestrator script.

Compute host update includes the update of the integrated Cloud SDN Switch (CSS) component.

Compute host update includes the update of the integrated HDS Agent, if the system is based on HDS.

Affected Nodes

- Compute hosts
- Compute hosts hosting vFuel

Note: Rollback of compute hosts hosting vFuel is not possible. If update of these nodes fails, the CEE region must be redeployed.

- Compute hosts hosting vCICs

Note: Rollback of compute hosts hosting vCIC is not possible. If update of these nodes fails, the CEE region must be redeployed.

Orchestration Options

When updating the compute hosts hosting vCICs and the compute host hosting vFuel, `serial` update mode must be used.

The required steps for the update of the component are described in Section 4.2.8 on page 24.



2.1.6 CSS Update

Affected Nodes

- Compute hosts

Orchestration Options

If only Fuel plugins are updated, run the orchestrator script using the `--plugin-update` option. This option skips the vFuel update step.

The required steps for the update of the component are described in Section 4.2.9 on page 25.

2.1.7 HDS Agent Update

Note: This procedure is only applicable if the CEE region is using HDS.

Compute hosts are updated automatically by the orchestrator script, if multiple components are updated in one procedure.

If only the HDS Agent is updated, the update can also be performed manually, without compute host restart.

Affected Nodes

- ScaleIO servers
- Compute hosts

Orchestration Options

If only Fuel plugins are updated, run the orchestrator script using the `--plugin-update` option. This option skips the vFuel update step.

The required steps for the update of the component are described in Section 2.1.7 on page 13.

2.1.8 Atlas Update

Affected Nodes

- Atlas

The update and rollback procedures for Atlas are described in the respective Operating Instructions.



2.2 Procedure Durations

The complete time required for the update and rollback procedures can be estimated using the following approximate durations:

Update

Note: Procedure times for compute hosts includes the update time of the included CSS or HDS Agent components.

- Fuel update: 15 minutes
- Update of the vCIC cluster: 110 minutes
- Update of each vCIC, on systems running SDN TI: 60 to 80 minutes
- Update of compute host hosting the active vFuel VM: 50 minutes
- Serial update of compute hosts hosting vCIC: 120 minutes
- Compute host update for each compute using **serial** method: 30 minutes
- Update of a subset of compute hosts using **parallel** method: 30 minutes
- Update of each ScaleIO server using **serial** method: 20 minutes, depending on the data stored on the server
- Update of CSS, for each compute host: 30 minutes
- Update of HDS Agent for each compute host (using orchestrated update): 30 minutes
- Update of CSC: refer to Cloud SDN Upgrade and Rollback, Reference [1]

Rollback

- Transfer time for each vCIC VM: 20 minutes
- Transfer time for vFuel VM: 7 to 8 minutes



3 Update Orchestration Configuration File

In the update procedure, the `update_groups.yaml` specifies the nodes to be updated, the update order, and the update method (serial or parallel) to be used.

This section describes the preparation of the `update_groups.yaml` before the update procedure.

The `update_groups.yaml` follows the YAML Specification, Reference [5].

The `update_groups.yaml` can be used to perform update procedures on all nodes of the region, a subset of nodes, or individual nodes as well. If update is performed in multiple sessions, the overall update order must strictly follow the update order described in Figure 1. If update is performed in multiple sessions, the `update_groups.yaml` must be changed before each execution of the `update_orchestrator.sh`, to only contain the nodes that are involved in the particular session.

Note: If no `update_groups.yaml` file is present in the `/mnt/cee_config` directory, all nodes in the CEE region are updated in serial mode.

The CEE software tarball contains the `CEE_RELEASE/update_groups.yaml.template` file, which can be used as a template when creating the `/mnt/cee_config/update_groups.yaml` update configuration file. The template file contains predefined sections for the node types to be updated. The template file also contains commented instructions on preparing the `update_groups.yaml` file.

The `update_groups.yaml` consists of sections. Each section defines an update phase, that is, a subset of nodes to be updated together. A section must be defined, even for a single-node update phase or session.

Note: If the update stops and needs to be restarted, the already updated nodes must be removed from the `update_groups.yaml` file. To check the update progress, see Section 6.1 on page 41.

Each section must have the following structure:

```
- type: <mode>
  nodes:
    - <node_1_name>
    - <node_2_name>
    - <node_3_name>
  ...
```

type:

The `type` key defines the update mode. `<mode>` can have the following values:



- `serial` if nodes are to be updated one by one, consecutively
- `parallel` if multiple nodes of the same type are to be updated concurrently

If `parallel` update mode is used for compute hosts, the number of hosts that can be updated at the same time must be defined. Depending on HA policies, some or all running VMs must be migrated from the nodes updated concurrently. Therefore, the size of the group is determined by the size of the region and the available free resources on the remaining compute hosts.

For example, if the free capacity is enough to host all VMs currently located on two compute hosts, the maximum size for parallel update is two.

nodes:

The `nodes` list contains the list of the nodes to be updated in each phase. For the value for the `<node_name>` variable, see name column in the printout of the `fuel node` command.

Examples

For examples for configuration files for different update procedures, see Section 7 on page 45.

Editing YAML Files in Windows

If the configuration file is edited in Windows, it is likely that the file contains CRLF characters. To remove CR characters (Linux only uses LF), run the following command after transferring the file to vFuel:

```
$> sed -i.bak -e 's/\r//g' <FILE.NAME>
```

A backup of the original file with the name `<FILE.NAME>.bak` is also created.



4 Procedure

The update procedures have the same preparation and concluding steps if one, or multiple, or all components are updated. Do the following:

1. Perform the steps described in Section 4.1 on page 17.
2. Perform the required procedures from Section 4.2 on page 19.
3. Perform the steps described in Section 4.3 on page 27.

4.1 Mandatory Preparation Stage

Before performing any of the update procedures, do the following:

1. Perform CEE health check as described in the document [Health Check Procedure](#).
2. Synchronize the active and the cold standby Fuel VMs as described in [Fuel Synchronization](#).
3. Create the following backups and store them in a persistent storage outside of the CEE region:
 - CIC domain data backup as described in [CIC Domain Data Backup](#)
 - Atlas backup as described in [Atlas Backup](#), if Atlas is used in the system
 - Manual backup of the three CIC VM images and the Fuel VM image and configuration files, as described in [Backup and Restore Overview](#)

Note: The CIC and Fuel VM backup is a prerequisite of rollback, and must be executed before update.

4. If only the CSC Fuel plugins are updated, continue with Section 4.2.1 on page 19. Else, continue with this procedure.
5. Log on to vFuel as root using SSH. For more information, refer to the [CEE Connectivity User Guide](#).

Note: Connectivity to the vCICs will be lost during the update.

6. Copy all relevant plugins to the `/var/www/nailgun/ericsson/fuel-plugins/` directory on vFuel:
 - a. Move any old plugin files to a backup directory using the following commands:

```
mkdir -p /var/www/nailgun/ericsson/fuel-plugins/backup
mv /var/www/nailgun/ericsson/fuel-plugins/<plugin_file>⇒
/var/www/nailgun/ericsson/fuel-plugins/backup
```



where `<plugin_file>` corresponds to the following values:

Component	Value
CSS Fuel Plugin	<code>ericsson_css*</code>
ScaleIO Fuel Plugin	<code>scaleio-2*rpm</code>
HDS Agent Fuel Plugin	<code>ericsson_hds_agent-*rpm</code>

- b. Transfer the new plugin files to `/var/www/nailgun/ericsson/fuel-plugins/` on vFuel.

- c. If the plugin file is packaged in a `.tar` file, unpack the file:

```
tar -xvf <plugin_file_name>.tar
```

- d. If applicable, validate the integrity of the plugin `.rpm` file by comparing the outputs of the following commands with the contents of the respective `.md5` or `.sha1` file:

```
md5sum <plugin>.rpm
```

```
sha1sum <plugin>.rpm
```

If the checksums do not match, contact the next level of maintenance support.

- e. Make sure that the plugin `rpm` file used for the update is the last item listed in the printout of the `ls <plugin_name>` command:

```
ls /var/www/nailgun/ericsson/fuel-plugins/<plugin_name> |tail -1
```

where `<plugin_name>` corresponds to the following:

Component	Value
CSS Fuel Plugin	<code>ericsson_css-*</code>
ScaleIO Fuel Plugin	<code>scaleio-*</code>
HDS Agent Fuel Plugin	<code>ericsson_hds_agent-*</code>

7. Transfer the CEE tarball to the `/var/tmp` directory on vFuel.

8. Extract the tarball:

```
tar -xvf <tarball_name>
```

9. Copy the `update_orchestrator.sh` file from the CEE software tarball to the `/root` directory on vFuel:

```
cp /<update_orchestrator_path>/update_orchestrator.sh⇒
```



/root

10. To ensure that the update process is not interrupted, start a screen session and run the commands in it:

```
cd ~; screen -r update -R -L
```

Later during the update, if a node is rebooted and the connection is lost towards vFuel, log back to vFuel with the steps above, and reattach the screen session with the command:

```
screen -r update
```

Note: The screen session can only be reattached after the node rebooted and is back online.

After exiting the screen session, the screen log file is available in `~/screenlog.0`.

11. Copy the prepared `update_groups.yaml` in the `/mnt/cee_config` directory on vFuel:

```
cp /<update_groups_path>/update_groups.yaml => /mnt/cee_config
```

4.2 Update Stage

The relevant procedures described in this section must be executed in the order they are presented. Only perform the procedures that correspond to the relevant flow in section Section 2 on page 8

Depending on the combination of the components to be updated, skip individual procedures as applicable. For example, if compute hosts are updated, CSS and HDS Agent are updated automatically, and the respective procedures are not required.

4.2.1 Update CSC

Note: This procedure is only applicable to CEE regions using SDN TI.

The update of the CSC Fuel plugins is a manual procedure not orchestrated by the CEE update orchestrator script. For the update and rollback procedures of the CSC Fuel plugins, refer to the SDN document Cloud SDN Upgrade and Rollback, Reference [1].

If the update of the component fails, continue with Section 5.1 on page 29.

4.2.2 Preparation for the Orchestrated Update

Perform the steps of this procedure if any or multiple of the following are updated:



- vFuel
- ScaleIO Fuel plugin
- vCICs
- HDS Agent in combination with other components

Note: If only the HDS Agent Fuel plugin is updated, skip this procedure and continue with Section 4.2.10 on page 25

- Compute hosts not hosting vCIC or vFuel
- Compute hosts hosting vCIC or vFuel

Do the following:

1. Create backups of the configuration YAML files:

```
mkdir -p /mnt/cee_config/backup-<date>
cp /mnt/cee_config/*.yaml /mnt/cee_config/backup-<date>
```
2. Remove all servers in State: `discover` from the Fuel database. Do the following:
 - a. Check the state of all servers according to the procedure described in Section 6.3 on page 41.
 - b. Lock the servers in State: `discover`:

```
setadminstate <shelf-id> locked --blade <blade-id>
```

An example of the command is the following:

```
setadminstate 0 locked --blade 2
```

- c. Remove the servers from the Fuel database:

```
fuel node --node-id <node-id> --delete-from-db --force
```

An example of the command is the following:

```
fuel node --node-id 8 --delete-from-db --force
```

- d. In `config.yaml`, comment out the definitions related to the servers. The following is an example:

```
#      -
#      id: 2
#      nic_assignment: *BSP_GEP5_nic_assignment
#      reservedHugepages: *BSP_GEP5_reservedHugepages
#      reservedCPUs: *auto_reservedCPUs
```




- e. Remove the entries related to the servers from `/mnt/cee_config/update_groups.yaml`.
3. If there is no active screen session, start a screen session and run the commands in it:


```
cd ~; screen -r update -R -L
```

Later during the update, if a node is rebooted and the connection is lost towards vFuel, log back to vFuel with the steps above, and reattach the screen session with the command:

```
screen -r update
```

Note: The screen session can only be reattached after the node rebooted and is back online.

After exiting the screen session, the screen log file is available in `~/screenlog.0`.
4. Check if there are any changes in the `config.yaml` between the CEE releases. If necessary, update the `config.yaml` using the new templates bundled with the ISO image.

Note: Only update the `config.yaml` according to configuration changes between the CEE releases. Reconfiguration of the system during update (for example, reallocation of vCPUs) is not possible in CEE.

Reconfiguration of CSS CPU reservation mode by changing the `css_mode` parameter in the `config.yaml` is not allowed during the update procedure. For more information, see Table 1.

If the NeLS server connection and certificates are not configured on the system before the update, licensing must be configured only after update, with the respective post-installation step.

If the NeLS server connection settings and certificates are configured already before the update, the configuration in the `config.yaml` must correspond to the actual configuration, and the certificate files must be in place. If the configuration and the values in `config.yaml` are not correct, the update fails. For more information, refer to the [Configuration File Guide](#).

Verify that the configuration of mandatory Fuel plugins corresponds to the [Fuel Plugin Configuration Guide](#).
5. Make sure that the `/mnt/cee_config/update_groups.yaml` file is available and specifies the nodes to be updated, **and strictly follows the correct update order**, see Figure 1.

Note: If `/mnt/cee_config/update_groups.yaml` does not exist, CEE update will be executed on all hosts of the CEE region, in serial mode.



4.2.3 Execute the Orchestrated Update Script

1. Start the update script by executing the following commands:

```
/<path_to_update_orchestrator>/update_orchestrator.sh =>  
<path_to_cee_iso>
```

An example of the command with the locations described in this procedure is the following:

```
/root/update_orchestrator.sh /var/tmp/<cee_iso>
```

Note: If the update process is required to stop after vFuel update, execute the script using the `--exit-after-fuel-update` option.

If update is performed in multiple sessions, the `update_groups.yaml` must be changed before each execution of the `update_orchestrator.sh` to only contain the nodes that are updated in the particular session. The update order described in this section must be strictly followed also if update is performed in multiple sessions.

If only Fuel plugins are updated, execute the command using the `--plugin-update` option. This option skips vFuel update; however, vFuel backup is still a prerequisite.

2. If the update orchestrator script stops, see Section 5 on page 29 for the error handling procedures.

If all nodes have been updated, continue with Section 4.3 on page 27.

4.2.4 Update vFuel

The CEE update process initiated by the orchestrator script always starts with the update of vFuel. The update of the vFuel node is not defined in `update_groups.yaml`. If the vFuel software version already corresponds to the vFuel software version included in the CEE release, the update orchestrator skips the update of the vFuel node.

If the system is using SDN TI, the `--exit-after-fuel-update` switch must be used. After vFuel update, but before updating any further components, do the following:

1. Open the `/usr/share/ericsson-orchestration/playbooks/update-fuel-deployment.vars.yml` using nano or similar.
2. By prepending `#`, comment out the following line:

```
- odl_neutron_config
```

An example of the commented line is the following:

```
#- odl_neutron_config
```



3. Save the changes and exit the editor.
4. Continue with the orchestrated update.

Start the update script by executing the following commands:

```
/<path_to_update_orchestrator>/update_orchestrator.sh =>
<path_to_cee_iso>
```

An example of the command with the locations described in this procedure is the following:

```
/root/update_orchestrator.sh /var/tmp/<cee_iso>
```

Note: The update orchestrator script can be run multiple times with updated `update_group.yaml` to update the nodes of the CEE region in subsets. For more information, see Section 3 on page 15.

If the update process is required to stop after vFuel update, execute the script using the `--exit-after-fuel-update` command-line switch.

If the update orchestrator script stops, see Section 5 on page 29 for the error handling procedures.

4.2.5 Update ScaleIO

The update process of the ScaleIO plugin and the ScaleIO servers is automatically performed by the `update_orchestrator.sh` script if the nodes are specified for update in the `update_groups.yaml`.

Start the update script by executing the following commands:

If only Fuel plugins are updated, execute the command using the `--plugin-update` option

```
/<path_to_update_orchestrator>/update_orchestrator.sh =>
<path_to_cee_iso>
```

An example of the command with the locations described in this procedure is the following:

```
/root/update_orchestrator.sh /var/tmp/<cee_iso>
```

Note: The update orchestrator script can be run multiple times with updated `update_group.yaml` to update the nodes of the CEE region in subsets. For more information, see Section 3 on page 15.

If the update orchestrator script stops, see Section 5 on page 29 for the error handling procedures.



4.2.6 Update vCIC

The update process of the vCICs is automatically performed by the `update_orchestrator.sh` script if the nodes are specified for update in the `update_groups.yaml`.

Start the update script by executing the following commands:

```
/<path_to_update_orchestrator>/update_orchestrator.sh⇒  
  <path_to_cee_iso>
```

An example of the command with the locations described in this procedure is the following:

```
/root/update_orchestrator.sh /var/tmp/<cee_iso>
```

Note: The update orchestrator script can be run multiple times with updated `update_group.yaml` to update the nodes of the CEE region in subsets. For more information, see Section 3 on page 15.

If the update orchestrator script stops, see Section 5 on page 29 for the error handling procedures.

4.2.7 Health Check

It is strongly recommended to check that the system is healthy before performing compute host update. If the update procedure fails after compute host update is started, the options for rollback and recovery of the system are limited. Perform health check as described in the [Health Check Procedure](#).

4.2.8 Compute Host Update

The update process of the compute hosts is automatically performed by the `update_orchestrator.sh` script if the nodes are specified for update in the `update_groups.yaml`.

Start the update script by executing the following commands:

```
/<path_to_update_orchestrator>/update_orchestrator.sh⇒  
  <path_to_cee_iso>
```

An example of the command with the locations described in this procedure is the following:

```
/root/update_orchestrator.sh /var/tmp/<cee_iso>
```

Note: The update orchestrator script can be run multiple times with updated `update_groups.yaml` to update the nodes of the CEE region in subsets. For more information, see Section 3 on page 15.



If the update orchestrator script stops, see Section 5 on page 29 for the error handling procedures.

4.2.9 CSS Update

The update process of CSS is automatically performed by the `update_orchestrator.sh` script if the plugin version in `/var/www/nailgun/ericsson/fuel-plugins/` is different from the version already installed in the system.

Start the update script by executing the following commands:

If only Fuel plugins are updated, execute the command using the `--plugin-update` option

```
/<path_to_update_orchestrator>/update_orchestrator.sh⇒
<path_to_cee_iso>
```

An example of the command with the locations described in this procedure is the following:

```
/root/update_orchestrator.sh /var/tmp/<cee_iso>
```

Note: The update orchestrator script can be run multiple times with updated `update_groups.yaml` to update the nodes of the CEE region in subsets. For more information, see Section 3 on page 15.

If the update orchestrator script stops, see Section 5 on page 29 for the error handling procedures.

4.2.10 HDS Agent Update

Note: This procedure is only applicable if the CEE region is using HDS.

The HDS Agent can be updated using the following methods:

- Manual procedure

This is recommended if **only** the HDS Agent is updated, as the manual procedure can be performed without compute host reboot.

- Automatic update orchestrated by the CEE update orchestrator script

This is recommended if the HDS Agent is updated in combination with other components, as the hosts are rebooted regardless, due to the other component updates.

Manual Procedure

1. Update the `ericsson_hds_agent` plugin in Fuel:

```
fuel plugins --update⇒
```



```
/var/www/nailgun/ericsson/fuel-plugins/ericsson_hds_agent-<version>.noarch.rpm
```

2. Make sure the version of the `ericsson_hds_agent` plugin installed in Fuel is according to the Product Revision Information for the CEE release , Reference [4]:

```
fuel plugins list
```

3. Enable the `ericsson_hds_agent` plugin:

```
apply_settings /var/lib/ericsson/pre_deploy/⇒  
/mnt/cee_config/config.yaml 1 update⇒  
/etc/cee/eri_deployment_tasks.yaml⇒  
/etc/cee/repos.yaml
```

4. Update the deployment files with the plugin information:

```
pre_deploy /var/lib/ericsson/pre_deploy/⇒  
/mnt/cee_config/config.yaml 1 /etc/cee/openstack_config/
```

5. Synchronize the plugin tasks:

```
fuel plugins --sync
```

6. Execute the plugin tasks to update the plugin information in the slave nodes:

```
fuel node --node <node_ids> --tasks ⇒  
upload_configuration plugins_rsync ⇒  
plugins_setup_repositories setup_repositories ⇒  
eri_hds_agent_install --force
```

Where `<node_ids>` is a comma-separated list of all the node IDs in the `fuel node` command. For more information, see Section 6.3 on page 41.

Example command for a system with 9 nodes:

```
fuel node --node 1,2,3,4,5,6,7,8,9 --tasks upload_configuration plugins_rsync plugins_setup_r  
setup_repositories eri_hds_agent_install --force
```

7. Verify that all nodes are operational by logging on to vFuel and executing the `fuel node` command. For more information, see Section 6.3 on page 41.
8. Verify that the environment is operational by logging on to vFuel and executing the `fuel env` command. The environment must be in `Operational` state.

Orchestrated Procedure

The update process of the HDS Agent is automatically performed by the `update_orchestrator.sh` script if the plugin version in `/var/www/nailgun/ericsson/fuel-plugins/` is different from the version already installed in the system.



Start the update script by executing the following commands:

If only Fuel plugins are updated, execute the command using the `--plugin-update` option

```
/<path_to_update_orchestrator>/update_orchestrator.sh⇒  
<path_to_cee_iso>
```

An example of the command with the locations described in this procedure is the following:

```
/root/update_orchestrator.sh /var/tmp/<cee_iso>
```

Note: The update orchestrator script can be run multiple times with updated `update_groups.yaml` to update the nodes of the CEE region in subsets. For more information, see Section 3 on page 15.

If the update orchestrator script stops, see Section 5 on page 29 for the error handling procedures.

This is the last stage of the orchestrated update procedure.

4.2.11

Atlas Update

Note: This procedure is only applicable to CEE regions using Atlas.

The update of Atlas is a manual procedure not orchestrated by the CEE update orchestrator script. For the update and rollback procedures of Atlas, refer to the [Atlas SW Upgrade](#) document.

4.3

Common Concluding Stage

1. Verify that the update is performed successfully. Perform health check according to the [Health Check Procedure](#).
2. Verify the version of CEE by executing the following command on the vFuel master node:

```
cat /etc/cee_version.txt
```

The output has the following format:

```
RELEASE=CEE CXC1737883_4-<build_number>  
NAME=Mitaka on Ubuntu 14.04  
VERSION=R6-<r-state>-<specific_build_number>-9.0
```

Verify the CEE version by comparing the `<build_number>` and the `<r-state>` to the Product Revision Information for Cloud Execution Environment (CEE), Reference [4].



An example output is:

```
[root@fuel ~]# cat /etc/cee_version.txt
RELEASE=CEE CXC1737883_4-1918
NAME=Mitaka on Ubuntu 14.04
VERSION=R6-R7B06-5384594593-9.0
```

If verification fails, see Section 5 on page 29.

3. Verify the version of CEE on all vCICs and compute hosts by executing the following command on the vFuel master node:

```
for n in $(fuel node | awk -F '|' '{print $7 ~ =>
/controller|compute/ {print $3}'); do echo ${n}; ssh =>
-o LogLevel=error ${n} 'cat /etc/cee_version.txt'; done
```

Verify the CEE version by comparing the <build_number> and the <r-state> to the Product Revision Information for Cloud Execution Environment (CEE), Reference [4].

4. Synchronize the active and the cold standby vFuel VM as described in the document [Fuel Synchronization](#).
5. After update, there can be an active NeLS Server Communication Problem alarm, because the NeLS server is not configured and not available.

To configure the connection to the NeLS server, follow the instructions in the [Runtime Configuration Guide](#). If the alarm does not clear, follow the instructions in the [NeLS Server Communication Problem alarm OPI](#).

6. If applicable, exit the screen session:

```
exit
```

7. Verify, and if applicable, update the OpenStack administrator password in Keystone on vFuel and the vCICs, as described in the relevant sections of the document [Security User Guide](#), as manual changes since deployment are overwritten at update.
8. For disaster recovery purposes, the installation media used for the update must be backed up, outside the CEE region. For more information, refer to the document [Disaster Recovery](#).
9. Verify that each node is updated, see Section 6.1 on page 41.



5 Error Handling

Note: Rollback should only be performed if the update procedure cannot be recovered using the procedures described in section Section 5 on page 29. The failure of the rollback procedure can result in a state that can only be recovered by redeployment. Notify the next level of customer support before attempting the rollback procedures.

5.1 Error Handling for Failed CSC Update

Note: Before attempting rollback of the CSC Fuel plugins, contact next level of support.

This procedure is only applicable to CEE regions using tightly integrated SDN (SDN TI).

Do the following:

1. Attempt the downgrade of the CSC Fuel plugins using the manual procedure described in the SDN document Cloud SDN Upgrade and Rollback, Reference [1].
2. If the downgrade procedure for the CSC Fuel plugins fails, the vCICs can be restored to the state before the update using the backed up vCIC images and configuration files. For more information, see Section 5.2.2.3 on page 34.

5.2 Error Handling for Failed Orchestrated Update

In case any error occurs during the update procedures orchestrated by CEE, follow these steps to repair:

1. Check the following logs:
 - a. `/var/log/ansible.log`
 - b. `/var/log/puppet-error.log` and `/var/log/puppet.log` of the failed systems according to `ansible.log`
 - c. The logs of the failed systems according to `ansible.log`
 - d. Update execution log, located at `/var/log/update_orchestrator.log`

The `update_orchestrator.log` can contain very long lines, that can cause editors to crash. To reformat the log to a readable format, execute the following command:

```
/<path_to_update_orchestrator>/update_orchestrator.sh ⇒
```



`--pretty-log <filename>`, where `<filename>` is the filename for the reformatted log. If no filename is specified, the reformatted log file is stored under the filename `update_orchestrator.pretty.log`.

The reformatted log file is stored in the `/var/log/` folder.

- e. Update procedure progress, stored at `/var/tmp/update_orchestrator.state`
2. Perform data collection according to the [Data Collection Guideline](#).
3. Fix the possible problems and rerun the update towards the failing node.
4. Contact the next level of support.
5. If applicable, attempt rollback using the procedures described in Section 5.2.1 on page 30, depending on the updated components.

5.2.1 Rollback

The rollback procedure is used to restore the system to the software versions used before the update, if the update procedure fails. The rollback procedure includes the rollback of all of the updated nodes.

Rollback of ScaleIO servers is not supported by Dell EMC.

vCIC rollback is achieved through restoring the vCIC VM from a backed up image, databases are restored to the state at the time of the update as well. Databases include information on the location of each VM, that is, which compute host is hosting which VM. Rollback is only possible if the actual VM locations match the databases, therefore rollback is only possible if VMs are not migrated during the update.

Compute host rollback is achieved, using the server replacement procedure, as described in the document [Server Replacement](#). Hardware replacement is not required. After the repair procedure, the compute host will be running the CEE version corresponding to the version of the vFuel node used for the repair.

Note: Contact next level of support before attempting compute rollback.

Compute host rollback is only possible if VMs were not migrated during update.

Compute hosts hosting vCIC or vFuel cannot be rolled back using server replacement. If the update of a vFuel or a vCIC host fails, redeployment of the CEE region is required.

Component or components

vFuel

Procedures

vFuel rollback, see Section 5.2.2.2 on page 31



Component or components	Procedures
vCIC	vCIC rollback, see Section 5.2.2.3 on page 34
CSS	Rollback of CSS, see Section 5.2.2.4 on page 39
Compute hosts not hosting vCIC or vFuel ⁽¹⁾	Compute host rollback, see Section 5.2.2.5 on page 39
HDS Agent	Rollback of HDS Agent, see Section 5.2.2.6 on page 39

(1) Rollback of compute hosts hosting vCIC or vFuel is not possible in this CEE release.

5.2.2 Rollback Procedures

5.2.2.1 ScaleIO Rollback

Rollback of ScaleIO servers is not supported by Dell EMC.

5.2.2.2 vFuel Rollback

Do the following:

1. If not performed earlier in the rollback procedure, insert forwarding rule on all three vCICs, as described in Section 6.5 on page 42.
2. Log on to the compute host hosting the active vFuel VM as root using SSH and the data collected in Section 6.4 on page 42. For more information, refer to CEE Connectivity User Guide.
3. Shut down the active vFuel VM by executing the following command:
virsh shutdown fuel_master

The expected printout is the following:
Domain fuel_master is being shutdown.

4. Verify that the active vFuel VM is shut down by executing the command **virsh list --all**. For more information, see Section 6.2 on page 41.
5. Undefine the active vFuel VM by executing the following command:
virsh undefine fuel_master

The expected printout is the following:
Domain fuel_master has been undefined

6. Verify that the active vFuel VM has been undefined by executing the command **virsh list --all**. For more information, see Section 6.2 on page 41.

If the vFuel VM has been undefined, it is not listed in the printout.



7. Remove the active vFuel VM by executing the following command:
`rm /var/lib/nova/<fuel_vm_image_file>`

An example of the command is the following:
`rm /var/lib/nova/fuel_master.qcow2`

8. Add a route between the host hosting vFuel and the external FTPS server by executing the following command:

```
route add <ftps_server_ip> gw <vcic_ip>
```

The variables are the following:

- <ftps_server_ip> is the IP address of the external FTPS server.
- <vcic_ip> is the IP address of a vCIC that is operational or in maintenance mode.

9. Copy and transfer the dump XML file from the external FTPS server described in Section 1.3.1.3 on page 3 to /var/lib/nova by executing the following command:

```
curl -k --ftp-ssl ftp://<username>:<password>@<ftps_server_ip>//<source_path>/<file_name> > =>
/var/lib/nova/<file_name>
```

The variables are the following:

- <file_name> is the name of the dump XML file.
- <username> and <password> are the credentials to the FTPS server.
- <ftps_server_ip> is the IP address of the external FTPS server used for storing the CEE component backups.
- <source_path> is the path on the FTPS server to the directory for storing the CEE component backup files.

An example of the command is the following:

```
root@compute-0-6:/var/lib/nova# curl -k --ftp-ssl =>
ftp://admin:admin@10.0.0.1//rollback/>=>
fuel_master_compute6_running.xml > =>
/var/lib/nova/fuel_master_compute6_running.xml
```

10. Copy, transfer, and decompress the vFuel VM image from /var/lib/nova from the external FTPS server described in Section 1.3.1.3 on page 3 to /var/lib/nova by executing the following command:

```
curl -k --ftp-ssl ftp://<username>:<password>@<ftps_server_ip> // <source_path>/<compressed_file_name>=>
| pigz --stdout --decompress --processes=>
$(xmlstarlet sel -t -v /domain/vcpu < ./<<fuel_xml_name>>)>=>
```



```
> /var/lib/nova/<vfuel-img_file_name>
```

The variables are the following:

- <username> and <password> are the credentials to the FTPS server.
- <ftps_server_ip> is the IP address of the external FTPS server used for storing the CEE component backups.
- <source_path> is the path on the FTPS server to the directory for storing the CEE component backup files.
- <compressed_file_name> is the file name for the compressed vCIC image set at rollback. If the recommended values are used, the value is <vcic-img_file_name>.gz.
- <vfuel-img_file_name> is The vFuel VM image file name.
- <fuel_xml_name> is the corresponding configuration XML file.

An example of the command is the following:

```
root@compute-0-6:/var/lib/nova# curl -k --ftp-ssl⇒
ftp://admin:admin@10.0.0.1//rollback/fuel_master.qcow2.gz⇒
| pigz --stdout --decompress --processes⇒
$(xmlstarlet sel -t -v /domain/vcpu <⇒
./fuel_master_compute6_running.xml)⇒
> /var/lib/nova/fuel_master.qcow2
```

11. Define the vFuel VM using the backed up XML dump by executing the following command:
virsh define <dump_file_name>.xml

An example of the command and the printout is the following:

```
root@compute-0-6:~# ⇒
virsh define fuel_master_compute6_running.xml
Domain fuel_master defined from ⇒
fuel_master_compute6_running.xml
```

12. Verify that the active vFuel VM has been defined by executing the command **virsh list --all**. For more information, see Section 6.2 on page 41.

If the active vFuel VM has been defined, it is listed in the printout with State: shut off.

13. Start the active vFuel VM by executing the following command:
virsh start fuel_master

The expected printout is the following:

```
Domain fuel_master started
```



14. Verify that the active vFuel VM is running by executing the command `virsh list --all`. For more information, see Section 6.2 on page 41.
15. Verify that all nodes are operational by logging on to vFuel and executing the `fuel node` command. For more information, see Section 6.3 on page 41.
16. Restore the `/root/openrc` files on all vCICs. These files were temporarily changed during the update on the vCICs. Execute the following command on Fuel:

`/opt/ecs-fuel-utils/restore_openrc.sh`
17. Check the system for active alarms. If the `Fuel failed` alarm did not cease after the active vFuel VM is rolled back and is operational again, generate new SSH key as described in Section 6.6 on page 43. For more information on listing active alarms using CLI, refer to the document [CEE CLI Guide](#).
18. Synchronize the active and cold standby vFuel VMs using the procedure described in [Fuel Synchronization](#).

5.2.2.3

vCIC Rollback

vCIC rollback is achieved through restoring the vCIC VM from a backed up image, databases are restored to the state at the time of the update as well. Databases include information on the location of each VM, that is, which compute host is hosting which VM. Rollback is only possible if the actual VM locations match the databases, therefore vCIC rollback is only possible if VMs are not migrated during the update procedures.

Note: Perform this procedure only if the updated vFuel VM has already been rolled back and synchronized.

In case of vCIC rollback, all updated vCICs must be rolled back.

The rollback procedure must be performed in the reverse order of the VM image backup, that is, the vCIC that was backed up **last** must be rolled back **first**.

In this section, the three vCICs are referred to as vCIC1, vCIC2 and vCIC3. The assignment of numbers is the following:

- vCIC1 is the vCIC backed up first and rolled back last.
- vCIC2 is the vCIC backed up second and rolled back second.
- vCIC3 is the vCIC backed up third and rolled back first.

The procedure is described for rolling back vCIC3. The procedure must be repeated on the remaining vCICs with different values for the variables, respectively.

Do the following:

1. Verify that the forwarding rule to the FTPS server is established by doing the following on **each vCIC**:



- a. Log on to the vCIC using SSH. For more information, refer to the [CEE Connectivity User Guide](#).
 - b. Enter maintenance mode by executing the following command:


```
sudo umm on
```
 - c. Verify that the forwarding rule is established by executing the following command:

```
iptables -t nat -C POSTROUTING -j MASQUERADE
```

Note: If the printout indicates failure, append the rule by executing the following command:

```
iptables -t nat -A POSTROUTING -j MASQUERADE
```
 - d. Log out of the vCIC:

```
exit
```
 - e. Repeat the procedure on **all vCICs**.
2. Log on to the compute host hosting vCIC3, using SSH. For more information, refer to the [CEE Connectivity User Guide](#).
3. Shut down the vCIC VM by executing the following command:

```
virsh shutdown <vcic_vm_name>
```

The expected printout is the following:
Domain <cic_vm_name> is being shutdown
4. Verify that the vCIC VM is shut down by executing the command `virsh list --all`. For more information, see Section 6.2 on page 41.
5. Undefine the vCIC by executing the following command:

```
virsh undefine <cic_vm_name>
```

An example of the command and the printout is the following:

```
root@compute-0-1:# virsh undefine cic-3_vm
```

Domain cic-3_vm has been undefined
6. Verify that the vCIC VM has been undefined by executing the command `virsh list --all`. For more information, see Section 6.2 on page 41.

 If the vCIC VM has been undefined, it is not listed in the printout.
7. Remove the vCIC VM image file, `<cic_name>_vm.xml` configuration XML file and `template_<cic_name>_vm.xml` template file by doing the following:
 - a. Navigate to `/var/lib/nova`:

```
cd /var/lib/nova
```
 - b. Remove the files by executing the following command:



```
rm <vm_image_file_name> <cic_vm_xml_name>⇒
<xml_template_file_name>
```

An example of the command is the following:

```
root@compute-0-1:/var/lib/nova# rm cic-3_vm.img
cic-3_vm.xml template_cic-3_vm.xml
```

8. Add a route between the host hosting the vCIC and the external FTPS server by executing the following command:

```
route add <ftps_server_ip> gw <vcic_ip>
```

The variables are the following:

- <ftps_server_ip> is the IP address of the external FTPS server.
- <vcic_ip> is the IP address of an operational vCIC on the fuel_ctrl_sp network, that is, if vCIC3 image is transferred, the IP address of vCIC1 or vCIC2 on the fuel_ctrl_sp network.

9. Copy and transfer the XML configuration file and XML template **one by one** from the external FTPS server described in Section 1.3.1.3 on page 3 to /var/lib/nova by executing the following command:

```
curl -k --ftp-ssl ftp://<username>:<password>@⇒
<ftps_server_ip>//<source_path>/<file_name> >⇒
/var/lib/nova/<file_name>
```

The variables are the following:

- <file_name> is the filename of one of the following:
 - The corresponding <cic_name>_vm.xml configuration XML file
 - The corresponding template_<cic_name>_vm.xml template file
- <username> and <password> are the credentials to the FTPS server.
- <ftps_server_ip> is the IP address of the external FTPS server used for storing the CEE component backups.
- <source_path> is the path on the FTPS server to the directory for storing the CEE component backup files.

An example of the command is the following:

```
root@compute-0-6:/var/lib/nova# curl -k --ftp-ssl ⇒
ftp://admin:admin@10.0.0.1//rollback/cic-1_vm.xml > ⇒
/var/lib/nova/cic-1_vm.xml
root@compute-0-6:/var/lib/nova# curl -k --ftp-ssl ⇒
ftp://admin:admin@10.0.0.1//rollback/template_cic-1_vm.xml >⇒
/var/lib/nova/template_cic-1_vm.xml
```




10. Copy, transfer, and decompress the vCIC VM image from /var/lib/nova from the external FTPS server described in Section 1.3.1.3 on page 3 to /var/lib/nova by executing the following command:

```
curl -k --ftp-ssl ftp://<username>:<password>@=>
<ftps_server_ip> // <source_path>/<compressed_file_name>=>
| pigz --stdout --decompress --processes=>
$(xmlstarlet sel -t -v /domain/vcpu < ./<cic_name>_vm.xml>)>=>
> /var/lib/nova/<vcic-img_file_name>
```

The variables are the following:

- <username> and <password> are the credentials to the FTPS server.
- <ftps_server_ip> is the IP address of the external FTPS server used for storing the CEE component backups.
- <source_path> is the path on the FTPS server to the directory for storing the CEE component backup files.
- <compressed_file_name> is the file name for the compressed vCIC image set at rollback. If the recommended values are used, the value is <vcic-img_file_name>.gz.
- <vcic-img_file_name> is The vCIC VM image file name.
- <cic_name>_vm.xml is the corresponding configuration XML file.

An example of the command is the following:

```
root@compute-0-6:/var/lib/nova# curl -k --ftp-ssl=>
ftp://admin:admin@10.0.0.1//rollback/cic-1_vm.img.gz=>
| pigz --stdout --decompress --processes=>
$(xmlstarlet sel -t -v /domain/vcpu < ./cic-1_vm.xml)>=>
> /var/lib/nova/cic-1_vm.img
```

11. Verify that user and group permissions for the VM image are nova:nova by executing the following command and checking user and group permissions in the respective columns of the printout:

```
ls -l <vm_image_file_name>
```

If the user or group permissions changed, update them by executing the following command:

```
chown nova:nova <vm_image_file_name>
```

12. Define the vCIC VM using the respective <cic_name>_vm.xml configuration XML file by executing the following command:

```
virsh define <xml_file_name>
```

13. Verify that the vCIC VM has been defined by executing the command **virsh list --all**. For more information, see Section 6.2 on page 41.

If the vCIC VM has been defined, it is listed in the printout with State:shut off.

14. Set the vCIC VM to autostart:

```
virsh autostart <cic_vm_name>
```

The expected printout is the following:

```
Domain <cic_vm_name> marked as autostarted
```

15. Start the vCIC VM by executing the following command:
virsh start <cic_vm_name>

The expected printout is the following:

```
Domain <cic_vm_name> started
```

16. Wait until the vCIC VM is operational. The time required for the vCIC VM to start is approximately four minutes. Do not perform any operations until the vCIC VM is operational.

17. Verify that the vCIC VM is operational by executing the command **virsh list --all**. For more information, see Section 6.2 on page 41.

18. If not all vCICs have been rolled back, repeat the procedure on one of the remaining vCICs starting from Step 1.

- For **vCIC2**, the route must be added and removed on **vCIC1** and **vCIC3**.
- For **vCIC1**, the route must be added and removed on **vCIC2** and **vCIC3**.

19. If all vCICs have been rolled back, do the following:

- a. Verify that all vCICs are operational by executing the **sudo umm status** command on all vCICs. Do one of the following:
 - If any of the vCICs is in maintenance mode, exit maintenance mode by executing the **sudo umm off** command on the affected vCIC.
 - If any of the vCICs fails to start after 15 minutes, contact next level of support and exit this procedure.
 - If all vCICs are operational, continue with the procedure.
- b. When all three vCICs are in active state, wait until the databases are synchronized. Database synchronization takes less than 10 minutes.
- c. Perform vCIC health check according to the procedures described in the [Health Check Procedure](#), including the following:



- Check Pacemaker - vCIC state and cluster resource state
 - Check Galera cluster status
 - Check vCIC Maintenance Mode
- d. Check the system for active alarms. If the CIC failed alarm did not cease after a vCIC is rolled back and is operational again, generate new SSH key as described in Section 6.6 on page 43. For more information on listing active alarms using CLI, refer to the document CEE CLI Guide.
 - e. If applicable, continue with Section 5.2.2.5 on page 39.

5.2.2.4 CSS Rollback

If the update procedure for the CSS fails, and only the CSS is updated, refer to the manual CSS rollback procedure described in Cloud SDN Upgrade and Rollback, Reference [1].

5.2.2.5 Compute Host Rollback

Note: Contact next level of support before attempting compute rollback.

Compute host rollback is achieved, using the server replacement procedure, as described in the document [Server Replacement](#). Hardware replacement is not required. After the repair procedure, the compute host will be running the CEE version corresponding to the version of the vFuel node used for the repair.

Compute repair can only be attempted if the following conditions are fulfilled:

- a No VMs were migrated during update.
- b vFuel is rolled back.
- c The active vFuel VM and the cold standby vFuel VM are synchronized.
- d vCICs are rolled back.

If the update of a compute host not hosting vFuel or vCIC fails, the compute host must be removed from the CEE region and repaired with the procedure described in [Server Replacement](#). Hardware replacement is not required. After the repair procedure, the compute host is running the CEE version corresponding to the version of vFuel, that is, the rolled back version of CEE.

Repair of compute hosts hosting vCIC, vFuel or both is not possible. If the update of such compute hosts fails, redeployment of the CEE region is required.

5.2.2.6 HDS Agent Rollback

Note: This section is only applicable if the CEE region is using HDS.



The HDS agent rollback is performed together with the rollback of the compute hosts. Limitations of compute host rollback apply to HDS agent rollback as well.

5.3 Error Handling for Failed Atlas Upgrade

Note: This procedure is only applicable if the CEE region is using Atlas.

Perform Atlas rollback as described in the relevant section of *Atlas SW Upgrade*.



6 Additional Operations

This section describes operations required by multiple procedures in this document.

6.1 Checking Update State

After vFuel has been updated, the state of the update can be checked anytime during the update process from vFuel. Run the command optionally with node names:

update_state [node_name]

This gives a short state report of the nodes. The following is an example of the update state report:

```
[root@fuel ~]# update_state
```

Node	State	Current	Target
compute-0-1	finished	R6-R7B06-5384594593-9.0	None
compute-0-3	finished	R6-R7B06-5384594593-9.0	None
compute-0-4	finished	R6-R7B06-5384594593-9.0	None
compute-0-5	finished	R6-R7B06-5384594593-9.0	None
cic-1	finished	R6-R7B06-5384594593-9.0	None
cic-2	finished	R6-R7B06-5384594593-9.0	None
cic-3	finished	R6-R7B06-5384594593-9.0	None

Example 1 Update State Report

Note: The version under the Current and Target columns must match the versions of the update path.

6.2 Checking VM State

To check if a VM is operational or shut down, execute the following command on the compute host hosting the VM: **virsh list --all**

An example of the printout is the following:

```
root@compute-0-1:~$ virsh list --all
```

Id	Name	State
2	cic-1_vm	running
-	fuel_master	shut off



6.3 Listing Nodes

To verify that all nodes are operational, or to list node names and IP addresses, execute the following command on vFuel:

fuel node

An example of the printout is the following:

id	status	name	cluster	ip	mac	roles	
7	ready	cic-1	1	192.168.0.32	6a:df:69:05:25:4d	controller, mongo	⇒
8	ready	cic-3	1	192.168.0.31	8e:f0:49:45:6a:43	controller, mongo	⇒
1	ready	compute-0-5	1	192.168.0.24	90:55:ae:3a:05:f6	compute	⇒
2	ready	compute-0-4	1	192.168.0.22	90:55:ae:3a:e5:76	compute	⇒
5	ready	compute-0-1	1	192.168.0.23	90:55:ae:39:f7:26	compute, virt	⇒
4	ready	compute-0-2	1	192.168.0.21	90:55:ae:3a:e3:ae	compute, virt	⇒
6	ready	cic-2	1	192.168.0.30	92:f9:49:4c:d4:4f	controller, mongo	⇒
3	ready	compute-0-3	1	192.168.0.25	90:55:ae:3a:e3:96	compute, virt	⇒
9	ready	compute-0-6	1	192.168.0.26	56:bd:11:f2:cd:42	compute	⇒
10	ready	compute-0-7	1	192.168.0.27	fa:30:2d:96:16:40	compute	⇒

pending_roles	online	group_id
	True	1
	True	1
	True	1
	True	1
	True	1
	True	1
	True	1
	True	1
	True	1
	True	1
	True	1

6.4 Identifying the Active and Cold Standby Fuel Hosts

Identify the compute hosts hosting the active vFuel VM and the cold standby vFuel VM by executing the following script:

```
[root@fuel ~]# for node in primary secondary
do
ip=$(get_vfuel_info --ip --$node);
name=$(ssh $ip hostname -s 2>&1 | grep compute);
stat=$(ssh $ip sudo virsh list --all 2>&1 | grep fuel);
stat=$(echo $stat | awk '{print $3 " " " $4}');
printf "%-10s | %s | %s\n" "$name" "$ip" "$stat";
done
```

An example of the printout is the following:

```
compute-0-6 | 192.168.0.23 | running
compute-0-1 | 192.168.0.20 | shut off
```

In the printout, **running** identifies the compute host hosting the active vFuel VM, and **shut off** identifies the compute host hosting the cold standby vFuel VM. Record the IP addresses of the hosts hosting the vFuel VMs from the printouts as this data is required in the rollback procedure.



6.5 Insert Forwarding Rule on vCICs

Do the following:

1. Log on to one of the vCICs using SSH. For more information, refer to the [CEE Connectivity User Guide](#).
2. Insert a forwarding rule for the routes to the external FTPS server by executing the following command:
iptables -t nat -A POSTROUTING -j MASQUERADE
3. Log out of the vCIC:
exit
4. Repeat the procedure on **all vCICs**.

6.6 Generating New SSH Key for Compute Host Hosting vCIC or vFuel

If the `Fuel failed` or `CIC failed` alarms issued during the rollback of the vFuel VM or any of the vCIC VMs did not cease after successful rollback and start of the node, a new SSH key must be generated for the host hosting the node. Do the following:

1. Log on to vFuel using SSH. For more information, refer to the document [CEE Connectivity User Guide](#).
2. Identify the ID of the host running the vFuel or vCIC VM by executing the following command:
fuel node

The ID of the host is listed under the `id` column of the printout. The host can be identified based on the name of the host listed in the `name` column of the printout.

Save this data as it will be used in a later step of the procedure.

For more information, see Section 6.3 on page 41.

3. Generate new SSH key for the node by executing the following command:

```
fuel node --node <node_id> ⇒  
--tasks eri_idam_distribute_fuel_creds --force
```

where `<node_id>` corresponds to the ID of the host identified in Section 6.3 on page 41.

An example of the command is the following:

```
fuel node --node 5 --tasks ⇒  
eri_idam_distribute_fuel_creds --force
```



4. Check the system for active alarms to see if the alarm has ceased. For more information on listing active alarms using CLI, refer to the [CEE CLI Guide](#).



Appendix

7 update_groups.yaml Examples

In Example 2, the `update_groups.yaml` file is configured for a 16-node CEE region, with a 5-node managed ScaleIO cluster. Update is done in one session. All nodes are updated in `serial` mode. In the last phase, the compute hosts hosting vFuel and the vCICs are updated. `compute-0-3` is hosting vFuel and one of the vCICs.

```
- type: serial
  nodes:
    - scaleio-0-4
    - scaleio-0-5
    - scaleio-0-6
    - scaleio-0-7
    - scaleio-0-8
- type: serial
  nodes:
    - cic-1
    - cic-2
    - cic-3
- type: serial
  nodes:
    - compute-0-9
    - compute-0-10
    - compute-0-11
    - compute-0-12
    - compute-0-13
    - compute-0-14
    - compute-0-15
    - compute-0-16
- type: serial
  nodes:
    - compute-0-3
    - compute-0-1
    - compute-0-2
```

Example 2 `update_groups.yaml` for 16-node CEE with ScaleIO, single session

In Example 3, the `update_groups.yaml` file is configured for a 24-node CEE region, with a 5-node managed ScaleIO cluster. Update is done in one session. Compute hosts are updated in `parallel` mode in multiple phases, in subsets



of four. compute-0-9 is hosting vFuel. compute-0-1, compute-0-2 and compute-0-3 are hosting the vCICs.

```
- type: serial
  nodes:
    - scaleio-0-4
    - scaleio-0-5
    - scaleio-0-6
    - scaleio-0-7
    - scaleio-0-8
- type: serial
  nodes:
    - cic-1
    - cic-2
    - cic-3
- type: parallel
  nodes:
    - compute-0-10
    - compute-0-11
    - compute-0-12
    - compute-0-13
- type: parallel
  nodes:
    - compute-0-14
    - compute-0-15
    - compute-0-16
    - compute-1-1
- type: parallel
  nodes:
    - compute-1-2
    - compute-1-3
    - compute-1-4
    - compute-1-5
- type: parallel
  nodes:
    - compute-1-6
    - compute-1-7
    - compute-1-8
- type: serial
  nodes:
    - compute-0-9
- type: serial
  nodes:
    - compute-0-1
    - compute-0-2
    - compute-0-3
```

Example 3 update_groups.yaml for 24-node CEE with ScaleIO, single session



In the following example, the `update_groups.yaml` file is configured for a 12-node CEE region. In this example, the update is accomplished in multiple sessions, with updated `update_groups.yaml` for each session:

- 1 vFuel and vCICs, see Example 4
- 2 Six compute hosts in parallel mode, in groups of two, see Example 5
- 3 The remaining three compute hosts in a single phase, in serial mode, see Example 6
- 4 The vFuel and vCIC hosts, in serial mode. `compute-0-1` is hosting vFuel and one of the vCICs, see Example 7.

Compute hosts are updated in parallel mode in subsets of four, in one session.

```
- type: serial
  nodes:
    - cic-1
    - cic-2
    - cic-3
```

Example 4 `update_groups.yaml` for 12-node CEE, session 1 - vFuel, vCICs

```
- type: parallel
  nodes:
    - compute-0-4
    - compute-0-5
- type: parallel
  nodes:
    - compute-0-6
    - compute-0-7
- type: parallel
  nodes:
    - compute-0-8
    - compute-0-9
```

Example 5 `update_groups.yaml` for 12-node CEE, session 2 - Compute hosts

```
- type: serial
  nodes:
    - compute-0-10
    - compute-0-11
    - compute-0-12
```

Example 6 `update_groups.yaml` for 12-node CEE, session 3 - Compute hosts



```
- type: serial
  nodes:
    - compute-0-1
    - compute-0-2
    - compute-0-3
```

Example 7 update_groups.yaml for 12-node CEE, session 4 - vFuel and vCIC hosts



8 NIC Firmware Version Check and Upgrade

To check the firmware version of any X710 NICs assigned to DPDK, do the following on **each compute host**:

1. Log on to the compute host as root using SSH. For more information, refer to the [CEE Connectivity User Guide](#).
2. Check NIC driver binding and record the PCI address and device name of any X710 NIC assigned to DPDK using the following command:

```
dpdk-devbind.py -s
```

An example of the printout is the following:

```
root@compute-0-3:~# dpdk-devbind.py -s
Network devices using DPDK-compatible driver
=====
0000:83:00.0 'Ethernet Controller X710 for 10GbE SFP+' drv=vfio-pci unused=
0000:83:00.3 'Ethernet Controller X710 for 10GbE SFP+' drv=vfio-pci unused=

Network devices using kernel driver
=====
0000:01:00.0 'I350 Gigabit Network Connection' if=eth0 drv=igb unused=vfio-pci
0000:01:00.1 'I350 Gigabit Network Connection' if=eth1 drv=igb unused=vfio-pci
0000:03:00.0 '82599ES 10-Gigabit SFI/SFP+ Network Connection' if=eth2 drv=ixgbe unused=vfio-pci
0000:03:00.1 '82599ES 10-Gigabit SFI/SFP+ Network Connection' if=eth3 drv=ixgbe unused=vfio-pci
0000:83:00.1 'Ethernet Controller X710 for 10GbE SFP+' if=eth5 drv=i40e unused=vfio-pci
0000:83:00.2 'Ethernet Controller X710 for 10GbE SFP+' if=eth6 drv=i40e unused=vfio-pci

Other network devices
=====
<none>

Crypto devices using DPDK-compatible driver
=====
<none>

Crypto devices using kernel driver
=====
<none>

Other crypto devices
=====
```

3. Check the firmware version of the NICs using one of the following options:

- a. Query the device information for the NIC using the following command:

```
ethtools -i <device_name>
```

where <device_name> is the device name of the NIC recorded earlier in the procedure.

An example of the command is the following:



```
root@compute-0-3:~# ethtools -i eth5
driver: i40e
version: 2.2.4
firmware-version: 4.53 0x80001fad 0.0.0
bus-info: 0000:83:00.1
supports-statistics: yes
supports-test: yes
supports-EEPROM-access: yes
supports-register-dump: yes
supports-priv-flags: yes
root@compute-0-3:~#
```

- b. If only DPDK interfaces are used, execute the following command:

```
egrep "<PCI_address> fw [0-9]\.[1-9][0-9]\.[0-9]{5}" /var/log/dmesg
```

where <PCI_address> is the PCI address of the NIC recorded earlier in the procedure.

An example of the printout is the following:

```
root@compute-0-3:~# egrep "0000:83:00.3: fw
[0-9]\.[1-9][0-9]\.[0-9]{5}" /var/log/dmesg
[ 15.122395] i40e 0000:83:00.3: fw 5.50.47059 api
1.5 nvm 5.51 0x80002bca 1.1568.0
```

4. If the NIC firmware version is lower than 6.0.1, update the firmware version according to the procedure described by the NIC manufacturer. Refer to Reference [3].

Note: Before firmware update, VMs hosted in the affected compute host must be migrated.

Note: In the procedure provided by the NIC manufacturer, the following step must be changed:

Instead of the `chmod 755 nvupdate.cfg` command, `chmod 755 nvupdate64e` must be used.

5. After firmware update, restart the server to activate the NIC firmware by executing the following command:

```
shutdown -r
```



Reference List

- [1] Cloud SDN Upgrade and Rollback, 1/1543-HSD 101 048/2-3
- [2] Limitations and Workarounds for Cloud Execution Environment (CEE) <release>, 5/109 21-AZE 102 01/5-n, where "n" is different for each release
- [3] Non-Volatile Memory (NVM) Update Utility for Intel® Ethernet Adapters—Linux, <https://downloadcenter.intel.com/download/25791/Ethernet-Non-Volatile-Memory-NVM-Update-Utility-for-Intel-Ethernet-Adapters-Linux-?product=82947>
- [4] Product Revision Information for Cloud Execution Environment (CEE) <release>, 109 21-AZE 102 01/5-n, where "n" is different for each release
- [5] YAML Specification, <http://www.yaml.org/spec/1.2/spec.html>