

Configuration File Guide

Cloud Execution Environment

USER GUIDE

Copyright

© Ericsson AB 2016–2018. 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
1.1.1	Hardware Terminology	2
1.2	Target Groups	2
1.3	Prerequisites	2
1.3.1	CEE Software Release Tarball	3
1.4	Generated and Prefilled Passwords	4
1.5	Editing the YAML Files	5
1.5.1	Using Anchors and Aliases	5
1.5.2	Editing config.yaml in Windows	6
2	Basic Parameter Settings	6
2.1	Region Name	6
2.2	Neutron Configuration	7
2.3	Hardware Switches	9
2.3.1	Configuring Extreme Switches	10
2.3.2	Configuring CMX Switches on BSP	13
2.3.3	Configuring Unmanaged Switch	14
2.4	Cloud Management	14
2.4.1	General Configuration	14
2.4.2	Additional Configuration for SDN Deployments	16
2.5	Server Configuration	17
2.5.1	Shelf and Blade Management	17
2.5.2	Compute Hosts	23
2.5.3	Virtual Cloud Infrastructure Controller (vCIC)	24
2.5.4	Virtual Fuel	25
2.5.5	NIC Assignment	25
2.5.6	Memory Allocation	26
2.5.7	CPU Allocation	28
2.5.8	Disk Reservation	29
2.5.9	Host Networking	30
2.6	Networks	32
2.6.1	Hardware-dependent Network Configuration	33
2.6.2	ScaleIO Network Configuration	36
2.6.3	SDN Network Configuration	37
2.6.4	Glance on Storage Network Configuration	39
2.7	Configure NTP	40
2.7.1	NTP Authentication	40
2.8	Legal Text Presented at Logon	42
2.9	Storage	42



2.9.1	ScaleIO Configuration	42
2.9.2	Local Storage (SSD)	45
2.9.3	Software RAID Configuration	46
2.10	Local Disk Partition Sizes	47
2.11	IdAM	48
2.12	LDAP Users	49
2.13	Glance Image Service	49
2.14	GRUB Configuration	50
2.15	Swift Configuration Options	50
2.16	SDN Integration on HDS	51
2.17	CM-HA	52
2.18	Fuel Plugins	52
2.19	Change of Linux I/O Scheduler	53
2.20	License Management Configuration	54
3	Advanced Parameter Settings	55
3.1	Advanced CPU Allocation	55
3.1.1	Automatic CPU Allocation Rules	56
3.1.2	Allocating CPUs for CSS (OVS)	57
3.1.3	Resource Allocation for vCIC	61
3.1.4	CPU Allocation for Single Server Deployment	62
3.2	NUMA Balancing	63
3.3	DPDK Physical Interface Driver	64
3.4	Kernel Parameter Settings	64
3.5	Increasing Virtio Queue Size	66
3.6	NIC Information	66
3.7	SR-IOV	67
3.7.1	SR-IOV Blade Configuration	67
3.7.2	SR-IOV Global Configuration	69
3.7.3	SR-IOV Cabling Scheme Configuration	70
3.8	PCI Passthrough	71
3.8.1	PCI Passthrough Alias Configuration	72
3.8.2	PCI Passthrough Blade Configuration	72
3.8.3	PCI Passthrough Cabling Scheme Configuration	73
3.9	Bandwidth-Based Scheduling	74
3.9.1	Nominal Bandwidth of Neutron Physical Networks	74
3.9.2	vSwitch Capacity	75
3.10	Neutron Configuration Options	76
3.11	Nova Configuration Options	78
3.12	Hardware Switch Configuration Options	79
3.13	Multiple Data Center Gateways	80



3.14	Change of Data Center Gateway Settings	82
3.15	Time Zone	83
3.16	Secure NBI API Endpoints	83
3.17	Fuel Administration Network	86
3.18	Location of Logs	87
3.19	Link Monitoring for CEE on BSP	88
3.20	Reduced Footprint Monitoring Data Collection	89
3.21	Zabbix CEE User	90
3.22	Excluding Disk At Deployment	91
3.23	Deployment Debugging Information	91
	Reference List	93





1 Introduction

This document describes how to prepare the site-specific configuration used when installing Cloud Execution Environment (CEE).

Before installation, the CEE configuration files must be edited. The configuration file templates are included in the installation tarball. The files provide configuration for the following areas:

- Compute hardware
- Cloud Management Networks
- Networking and NTP
- Storage
- Identity and Access Management (IdAM)

1.1 Scope

Configurations are described in both this document and the System Dimensioning Guides. This has been verified against the Cloud Execution Environment reference configurations. Refer to the following documents:

- Multi-Server System Dimensioning Guide, CEE 6
- Single Server System Dimensioning Guide, CEE 6

This document describes the following parameter types:

Mandatory parameters

These parameters **must** be configured before deployment.

Optional parameters

These parameters **can** be changed for some configurations, as described in Section 3 on page 55. Optional parameters not described in Section 3 on page 55 are outside the scope of this document.

Storage is measured in gibibyte (GiB), tebibyte (TiB), and mebibyte (MiB) in this document. 1 GiB is equal to 2^{30} bytes.

This guide describes the initial configurations needed before CEE deployment. For post-deployment configuration options, refer to the [Runtime Configuration Guide](#).



1.1.1 Hardware Terminology

In the configuration file templates for all hardware configurations, a universal CEE terminology is used, which maps to the following hardware-specific terms:

Table 1 Hardware Terminology

CEE Term	Definition	Dell Rack Servers Term	BSP (EBS) Term	HDS Term
CEE region	Several blade servers or rack-mounted servers	N/A	N/A	A vPOD with a set of ComputerSystems
Shelf	A collection of servers that shares a control switch	Servers connecting to the same X440 Extreme switches	Subrack	N/A ⁽¹⁾
Server	A basic compute engine to host cloud applications in CEE Region	Server	Blade	ComputerSystem
Blade	Logical/physical location of the server	Logical definition that typically corresponds to the iDRAC IP addressing plan	Defined by slot of Generic Ericsson Processor (GEP) blade in subrack	Logical definition that corresponds to a ComputerSystem identified with a UUID

(1) Only one shelf is to be configured for HDS, listing all blades per vPOD.

1.2 Target Groups

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

- Ericsson field personnel
- Ericsson designers and test personnel
- Support organization personnel

1.3 Prerequisites

This section describes the site-specific configuration parameters that are to be collected and defined before the installation. The overall workflow is described in [CEE Installation](#). Ensure that the following is available:

- A site-specific IP and VLAN plan, based on Reference [1].



- If applicable, the network entity that is managed by the OpenStack Neutron module of the CEE region to be installed.
- System dimensioning details, based on the hardware-relevant [System Dimensioning Guide](#).
- Additional site-specific system information, required to update the configuration file, including:
 - The number of shelves in the system to be installed
 - The number of blades installed in each shelf
- Configuration file templates in the CEE_RELEASE folder of the tarball:
 - `config.yaml.<hardware_platform>`. In case of HDS deployment, both an SDN and a non-SDN template are available.
 - Switch configuration templates (`<switch_model>_switch.yaml`) (for deployment with managed Extreme or CMX/CAX switches).
 - Host network templates (`host_nw.<hardware_platform>.yaml`).

Additional host network templates are available for features that require them. See Section 1.3.1 on page 3 and Section 2.5.9 on page 30.

In case of HDS deployment, both SDN and non-SDN templates are available.
 - Cabling scheme templates (for deployment with managed Extreme switches).
 - Neutron configuration templates (`neutron_ericsson.<neutron_template>.yaml`).
- EMC storage username and password (for deployment with EMC ScaleIO).
- IP addresses for the external NTP servers.

For more information on the required tarball contents, see Section 1.3.1 on page 3.

Use the `config.yaml` template that matches the hardware, that is, Dell multi-server, Dell single server, HDS, or BSP. For more information on the YAML files used for the reference configurations, see the following sections. Compare the examples in this document to the relevant part of the files. A file editor that can render indentations well (for example, UltraEdit) is recommended.

1.3.1 CEE Software Release Tarball

The CEE software release tarball is required for CEE installation and can be downloaded from the SW Gateway. The tarball consists of the following files:

- `CXC1737883_4-<release>.tar`



- CXC1737883_4-<release>.tar.md5
- CXC1737883_4-<release>.tar.sha1

Below is an example of the release tarball contents:

Note: The contents of the release tarball may vary according to the release.

```

cee-CXC1737883_4-<release>.iso
CEE_RELEASE/config.yaml.bsp
CEE_RELEASE/config.yaml.dell
CEE_RELEASE/config.yaml.dell-single_server
CEE_RELEASE/config.yaml.hds-with-sdn
CEE_RELEASE/config.yaml.hds-without-sdn
CEE_RELEASE/config.yaml.hp
CEE_RELEASE/update_groups.yaml.template
CEE_RELEASE/cabling_scheme/2_x670v_hp.yaml
CEE_RELEASE/cabling_scheme/2_x670v_dell.yaml
CEE_RELEASE/cabling_scheme/4_x670v_dell.yaml
CEE_RELEASE/cabling_scheme/4_x670v_hp.yaml
CEE_RELEASE/cabling_scheme/4_x770_hp.yaml
CEE_RELEASE/cabling_scheme/cax_bsp.yaml
CEE_RELEASE/host_net_templates/host_nw_bsp.yaml
CEE_RELEASE/host_net_templates/host_nw_bsp-bm-glance-on-storage-nics.yaml
CEE_RELEASE/host_net_templates/host_nw_bsp-glance-on-storage-nics.yaml
CEE_RELEASE/host_net_templates/host_nw_bsp_without_compute_storage.yaml
CEE_RELEASE/host_net_templates/host_nw_dell-single-server.yaml
CEE_RELEASE/host_net_templates/host_nw_dell.yaml
CEE_RELEASE/host_net_templates/host_nw_dell-bm-glance-on-storage-nics.yaml
CEE_RELEASE/host_net_templates/host_nw_dell-glance-on-storage-nics.yaml
CEE_RELEASE/host_net_templates/host_nw_dell_without_compute_storage.yaml
CEE_RELEASE/host_net_templates/host_nw_hds-with-sdn.yaml
CEE_RELEASE/host_net_templates/host_nw_hds-with-sdn-glance-on-storage-nics.yaml
CEE_RELEASE/host_net_templates/host_nw_hds-with-sdn_without_compute_storage.yaml
CEE_RELEASE/host_net_templates/host_nw_hds-without-sdn.yaml
CEE_RELEASE/host_net_templates/host_nw_hds-without-sdn-glance-on-storage-nics.yaml
CEE_RELEASE/host_net_templates/host_nw_hds-without-sdn_without_compute_storage.yaml
CEE_RELEASE/host_net_templates/host_nw_hp.yaml
CEE_RELEASE/host_net_templates/host_nw_hp-bm-glance-on-storage-nics.yaml
CEE_RELEASE/host_net_templates/host_nw_hp-glance-on-storage-nics.yaml
CEE_RELEASE/host_net_templates/host_nw_hp_without_compute_storage.yaml
CEE_RELEASE/neutron/neutron_ericsson_cmx.yaml
CEE_RELEASE/neutron/neutron_ericsson_extreme.yaml
CEE_RELEASE/neutron/neutron_ericsson_sdn_tight.yaml
CEE_RELEASE/neutron/neutron_ericsson_user_spec.yaml
CEE_RELEASE/scripts/install_vfuel.sh
CEE_RELEASE/scripts/migrate_fuel.sh
CEE_RELEASE/scripts/parseyaml.rb
CEE_RELEASE/scripts/update_orchestrator.sh
CEE_RELEASE/switch_config/2_x670v_switch.yaml
CEE_RELEASE/switch_config/4_x670v_switch.yaml
CEE_RELEASE/switch_config/4_x770_switch.yaml
CEE_RELEASE/switch_config/cmx_switch.yaml

```

1.4 Generated and Prefilled Passwords

Some generated passwords are stored in a separate file, `/etc/openstack_deploy/user_secrets.yml` on the Fuel node. The `user_secrets.yml` file contains passwords queried from Fuel, for example **rabbitmq_password**, **galera_root_password**.

As shown in Example 1, other passwords used by Ericsson components must be prefilled in `/etc/openstack_deploy/user_secrets.yml` on the Fuel node,



or generated by `/usr/share/ericsson-orchestration/scripts/pw-token-gen.py`:

```
rabbitmq_password: "{{ fuel_generated.rabbit.password }}"
galera_root_password: "{{ fuel_generated.mysql.root_password }}"
keystone_admin_password: "{{ fuel_settings.editable.access.password.value }}"
keystone_admin_token: "{{ fuel_generated.keystone.admin_token }}"
neutron_galera_password: "{{ fuel_generated.quantum_settings.database.passwd }}"
neutron_service_password: "{{ fuel_generated.quantum_settings.keystone.admin_password }}"
nova_service_password: '{{ fuel_generated.nova.user_password }}'
nova_metadata_proxy_secret: '{{ fuel_generated.quantum_settings.metadata.metadata_proxy_shared_secret }}'
cinder_password: "{{ fuel_generated.cinder.user_password }}"

cmha_galera_password:
cmha_service_password:
watchmen_galera_password:
watchmen_service_password:
idam_ldap_root_password:
idam_ldap_anonymous_bind_password:
idam_ldap_manager_bind_password:
idam_ldap_sync_bind_password:
idam_user_vnxlf_vnx_key:
idam_user_vnxlf_galera_password:
zabbix_cee_user_password:
```

Example 1 user_secrets.yml

1.5 Editing the YAML Files

1.5.1 Using Anchors and Aliases

The CEE configuration file is a YAML file. The YAML standard defines anchors and aliases, see section Anchors and Aliases in the YAML specification, Reference [4]. An anchor is used to attach a label to a section of the data structure, so the section can be referred by an alias. An anchor consists of a label prefixed with an `&` character: `&label1`. An alias consists of a label prefixed with a `*` character: `*label1`.

```
...
presets:
  - &label1
    key1: value11
    key2: value12
  - &label2
    key1: value21
    key2: value22
...
actual_use: *label2
...
```

Example 2 YAML Anchors and Aliases

An alias can be used to reference the data structure defined by the anchor.

The CEE configuration templates define the recommended settings for several parts of the configuration. Each of these settings is marked with an anchor that



can be used to reference them with an alias at the place where they are used. For example, settings such as the Network Interface Controller (NIC) assignment are expected to be identical for each server. With the use of aliases, the same NIC assignment is referenced by each server definition.

1.5.2 Editing config.yaml 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 the Fuel master node:

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

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

2 Basic Parameter Settings

This section describes how to update config.yaml with site-specific parameters. The updated configuration is used as input to the automated installation. The placeholder <variables> in the config.yaml must be replaced with valid values based on the information in this section.

Note: The indentation in the template files must be kept. Use the **SPACE** key (blanks) to make the indentation. **TAB** must not be used.

2.1 Region Name

The region_name parameter refers to the CEE region name. The parameter is included in config.yaml as follows:

```
ericsson:
  ...
  region_name: <CEE Region Name>
  ...
```

Example 3 CEE Region Name

<CEE Region Name> must be replaced by an OpenStack region name, with a maximum length of 14 characters. The CEE region name must not contain the **underscore** “_” character.



Note: The OpenStack endpoints are deployed by Fuel where the region name cannot be specified, so all the OpenStack endpoints will be configured with the default region name RegionOne. This difference does not affect the operation of CEE. Contact the next level of maintenance support to change the region name manually after installation if it is required.

The CEE region name and an **underscore** () are prepended to the system names of the hardware switches, if the switches are configured. For hardware switch configuration, see Section 2.3 on page 9, and Section 3.12 on page 78.

Note: Each switch name is a maximum of 17 characters long, and the resulting total string length must not exceed 32 characters.

2.2 Neutron Configuration

The neutron section in the configuration file templates enables configuration of the OpenStack networking module (Neutron) in CEE. The section is included in `config.yaml` as follows:

```
ericsson:
  ...
  neutron:
    mgmt_vip: 192.168.2.15
    mgmt_subnetmask: 24
    l2_vlan_start: <L2.VLAN.START>
    l2_vlan_end: <L2.VLAN.END>
    neutron_config_yaml_file: <neutron_CONFIG.yaml>
  ...
```

Example 4 Neutron Non-SDN Configuration

```
ericsson:
  ...
  neutron:
    mgmt_vip: 192.168.2.15
    mgmt_subnetmask: 24
    tunnel_id_start: <VXLAN.VNI.RANGE.START>
    tunnel_id_end: <VXLAN.VNI.RANGE.END>
    neutron_config_yaml_file: neutron_sdn_tight.yaml
  ...
```

Example 5 Neutron SDN Configuration

Change the configuration template values to reflect the site specific values:

mgmt_vip

`mgmt_vip` is the common IP for the Neutron server process. This value must be a valid IP in `cee_ctrl_sp` network static sub range.



mgmt_subnetmask

mgmt_subnetmask is consistent with the subnet mask size of the cee_ctrl_sp network.

l2_vlan_start and l2_vlan_end

The L2 VLAN range only applies to non-SDN deployments.

l2_vlan_start is the first element of the range for tenant VLAN IDs used for Neutron network segments.

l2_vlan_end is the last element of the range for tenant VLAN IDs used for Neutron network segments.

Note: This VLAN range is only for CEE tenant traffic, it must not include any CEE service VLANs (for example, management and storage networks).

tunnel_id_start and tunnel_id_end

Tunnel ID only applies to SDN deployments.

tunnel_id_start is the first element of the VXLAN VNI range. The value must be an integer in the range of 1 – 16777216.

tunnel_id_end is the last element of the VXLAN VNI range. The value must be an integer in the range of 1 – 16777216 and larger than tunnel_id_start.

neutron_config_yaml_file

neutron_config_yaml_file is the name of the file containing the Neutron configuration parameters. The following templates can be used for Neutron:

- neutron_ericsson_user_spec.yaml: Used for all configurations excluding those with SDN or managed CMX, CAX or Extreme switches.
- neutron_ericsson_extreme.yaml: Used for Extreme traffic switches.
- neutron_ericsson_cmx.yaml: Used for BSP deployment of CEE.
- neutron_ericsson_sdn_tight.yaml: Used for deployment with SDN.

The following example is a Neutron BSP configuration:



```
ericsson:
...
neutron:
  mgmt_vip: 192.168.2.15
  mgmt_subnetmask: 24
  l2_vlan_start: 130
  l2_vlan_end: 3999
  neutron_config_yaml_file: neutron_ericsson_cmx.yaml
...
```

Example 6 Neutron Configuration for BSP

For more information on the Neutron configuration file configuration, see Section 3.10 on page 76.

2.3 Hardware Switches

The `hw_switches` section describes the configuration to be deployed to the hardware switches during CEE region installation. The section can contain the following parameters:

```
ericsson:
...
hw_switches:
  initial_setup: <INITIAL_SETUP>
  switching_scheme_yaml_file: <SWITCH.CONFIGURATION_FILE.yaml>
  cabling_scheme_yaml_file: <CABLING.SCHEME_FILE.yaml>
```

Example 7 Hardware Switch Configuration

initial_setup

`initial_setup` specifies the initial setup of the hardware switches. The following values are valid:

- **extreme**, if using Extreme switch setup (Dell multi-server).
- **cmx**, if using CMX/CAX switches setup (BSP). If `initial_setup` is set to **cmx**, `cabling_scheme_yaml_file` is not applicable.
- **None**, if setting up unmanaged CEE switching (HDS without SDN, single server) or HDS with SDN configuration. If `initial_setup` is set to **None**, `switching_scheme_yaml_file` and `cabling_scheme_yaml_file` are not applicable.

Note: Semi-managed switch configuration is not possible in CEE. The `neutron_ericsson_user_spec.yaml` Neutron configuration file can only be used if `initial_setup` is set to **None**.

switching_scheme_yaml_file

Note: This keyword is only applicable if `initial_setup` is **cmx** or **extreme**.



The value of `<SWITCH.CONFIGURATION_FILE.yaml>` specifies the switch setup used:

- For Extreme switches, the following values are valid:
 - `2_x670v_switch.yaml`
 - `4_x670v_switch.yaml`
 - `4_x770_switch.yaml`
- For CMX/CAX switches on BSP, the following value is valid:
 - `cmx_switch.yaml`

The switch configuration file has to be aligned with the hardware setup. For more information, refer to the relevant hardware installation instructions: [Dell PowerEdge R630 HW Installation](#). For information on how to configure the selected `<switch_model>_switch.yaml` file, see Section 2.3.1 on page 10 (Extreme) or Section 2.3.2 on page 13 (CMX).

cabling_scheme_yaml_file

Note: This keyword is only applicable if `initial_setup` is **extreme**.

This parameter specifies the cabling schema used. The following values are valid:

- For Dell multi-server deployment:
 - `2_x670v_dell.yaml`
 - `4_x670v_dell.yaml`

The cabling configuration file has to be aligned with the hardware setup. For more information, refer to the relevant hardware installation instructions: [Dell PowerEdge R630 HW Installation](#). For advanced hardware switch configuration options, see Section 3.12 on page 78.

2.3.1 Configuring Extreme Switches

To configure the Extreme switches in multi-server deployment, the following networks need to be configured in the `<switch_model>_switch.yaml` file with site-specific values for both traffic switch A and B:

- `subrack_ctrl_sp` network for subrack management.
- `cee_om_sp` network for vCIC northbound communication.
- `subrack_om_sp` link network between the traffic switches and the Data Center Gateway (DC-GW), referred to as Border Gateway (BGW) in `config.yaml`.

For more information on these networks, see Section 2.6 on page 32.



The structure of the configuration options is as follows:

Switch A

```
switching:
-
...
  mgmt_vrrp_config:
  -
    vip: 10.0.3.1/24
    is_master: true
    vlan: subrack_ctrl_sp
  -
    vip: VIP.OF.THE.VRRP/PREFIXSIZE
    is_master: true
    vlan: subrack_om_sp
  ...
  mgmt_config:
    vlans:
    -
      name: cee_ctrl_sp
      tagged: true
    -
      name: subrack_ctrl_sp
      ip: 10.0.3.2/24
      tagged: true
  bgw_config:
  -
    id: 1
    vlans:
    -
      name: cee_om_sp
      tagged: true
      ip: <IP.OF.SWITCH-A/PREFIXSIZE>
    -
      name: subrack_om_sp
      tagged: true
      ip: IP.OF.THE.SWITCH/PREFIXSIZE
  ...
```

Switch B

```
switching:
-
...
  mgmt_vrrp_config:
  -
    vip: 10.0.3.1/24
    is_master: false
    vlan: subrack_ctrl_sp
```



```

-
  vip: VIP.OF.THE.VRRP/PREFIXSIZE
  is_master: false
  vlan: subrack_om_sp
...
mgmt_config:
  vlans:
    -
      name: cee_ctrl_sp
      tagged: true
    -
      name: subrack_ctrl_sp
      ip: 10.0.3.3/24
      tagged: true
bgw_config:
-
  id: 2
  vlans:
    -
      name: cee_om_sp
      tagged: true
      ip: <IP.OF.SWITCH-B/PREFIXSIZE>
    -
      name: subrack_om_sp
      tagged: true
      ip: IP.OF.THE.SWITCH/PREFIXSIZE
...

```

Example 8 Update Switch IP Addresses

mgmt_vrrp_config

The following parameters are needed:

- vip for subrack_ctrl_sp is the VRRP IP address of the traffic switches on subnet subrack_ctrl_sp.
- vip for subrack_om_sp is the VRRP IP address of the traffic switches on subnet subrack_om_sp (<VIP.OF.THE.VRRP/PREFIXSIZE>).

mgmt_config

The following parameters are needed:

- ip for subrack_ctrl_sp is the IP address of the traffic switch on subnet subrack_ctrl_sp, used as source address when the switch is synchronizing towards external NTP servers.

bgw_config

The following parameters are needed:



- ip for `cee_om_sp` is the IP address of the traffic switch on subnet `cee_om_sp`, used as source address when the switch is synchronizing towards external NTP servers (`<IP.OF.SWITCH-A/PREFIXSIZE> / <IP.OF.SWITCH-B/PREFIXSIZE>`).
- ip for `subrack_om_sp` is the IP address of the traffic switch on subnet `subrack_om_sp`, used as source address when the switch is synchronizing towards external NTP servers (`<IP.OF.THE.SWITCH/PREFIXSIZE>`).

2.3.2 Configuring CMX Switches on BSP

Note: The same configuration and switch configuration templates are applicable for CMX and CAX switches.

This section describes CEE on BSP hardware. The configuration file template contains the following section:

```
ericsson:
  ...
  hw_switches:
    initial_setup: cmx
    switching_scheme_yaml_file: cmx_switch.yaml
    ...
```

Example 9 Hardware Switches

The switch configuration template `cmx_switch.yaml` contains the following section:

```
switching:
-
  model: cmx
  provider_vlan_start: <PROVIDER.VLAN.START>
  provider_vlan_end: <PROVIDER.VLAN.END>
  provider_name_prefix: provider_
switch_config:
  initial_backup: postCEETenant
  migrated_backup: postMigrateCEETenant
```

Example 10 Switch Configuration

- `provider_vlan_start`: the first element of the range of VLAN tags to be used as segmentation IDs for Neutron network with provider extension.
- `provider_vlan_end`: the last element of the range of VLAN tags to be used as segmentation IDs for Neutron network with provider extension.
- `provider_name_prefix`: prefix of the VLAN names corresponding to the specified range. The VLANs are created in CMX switches during CEE installation.



- `initial_backup`: is the name of the BSP configuration backup saved by CEE installation script.
- `migrated_backup`: is the name of the BSP configuration backup that is created after vFuel is migrated inside the blade system.

2.3.3 Configuring Unmanaged Switch

This section describes unmanaged CEE switching. The configuration file template contains the following section:

```
ericsson:  
  ...  
  hw_switches:  
    initial_setup: None  
  ...
```

Example 11 Unmanaged Hardware Switch

2.4 Cloud Management

2.4.1 General Configuration

The Atlas northbound and southbound networks need to be configured in the `cloud_mgmt` section of `config.yaml`. Both the Atlas Northbound Interface (NBI) and the Atlas Southbound Interface (SBI) have to be configured with site-specific information. See the structure of the `nbi` and `sbi` subsections:



```
ericsson:
...
cloud_mgmt:
  nbi:
    name: <NETWORK.NAME>
    tag: <VLAN.TAG> / <VXLAN.TAG>
    cidr: <NETWORK.IP/PREFIXSIZE>
    start: <FIRST.IP.TO.USE>
    end: <LAST.IP.TO.USE>
    gateway: <IP.OF.THE.GW>
    ip: <NBI.IP.OF.ATLAS>
  sbi:
    name: <NETWORK.NAME>
    tag: <VLAN.TAG> / <VXLAN.TAG>
    cidr: <NETWORK.IP/PREFIXSIZE>
    start: <FIRST.IP.TO.USE>
    end: <LAST.IP.TO.USE>
    gateway: <IP.OF.THE.GW>
    ip: <SBI.IP.OF.ATLAS>
....
```

Example 12 Cloud Management Configuration

The following parameters are to be configured with unique values for NBI and SBI, respectively:

- name is the name of the Atlas northbound or southbound network.
- tag is the unique VLAN or VXLAN tag of the Atlas northbound or southbound network.
- cidr is the subnet of the NBI or SBI. All CIDR must be non-overlapping.
- start is the first northbound or southbound IP assigned to the Atlas Cloud Manager.
- end is the last northbound or southbound IP assigned to the Atlas Cloud Manager.
- gateway is the northbound or southbound IP on the DC-GW.
- ip is the northbound or southbound IP address of the Atlas Cloud Manager. This IP must be within range of the <FIRST.IP.TO.USE> and <LAST.IP.TO.USE> of the interface.

During the Atlas installation, there are two options available on how IP addresses are assigned to Atlas:

- The default option is to use the IP addresses defined in <NBI.IP.OF.ATLAS> and <SBI.IP.OF.ATLAS> to run an instance of Atlas.



- The other option is to assign IP addresses from the start--end ranges. This can be used if the intention is to run several instances of Atlas.

Note: `<NETWORK.NAME>`, `<FIRST.IP.TO.USE>`, `<LAST.IP.TO.USE>`, `<NBI.IP.OF.ATLAS>`, and `<SBI.IP.OF.ATLAS>` must always be defined, regardless of how Atlas is installed.

2.4.2 Additional Configuration for SDN Deployments

In SDN deployment, `cloud_mgmt` can be configured with either L3 VPN or L2 GW connectivity.

2.4.2.1 SDN Deployment with L3 VPN Connectivity

For SDN deployment with L3 VPN connectivity, configure the following extra `cloud_mgmt` parameters are needed:

```
cloud_mgmt:
  network_type: <NETWORK.TYPE>
  vpn:
    name: <VPN.NAME>
    rd: <ROUTE.DISTINGUISHER>
    export_rt: <EXPORT.ROUTE.TARGET>
    import_rt: <IMPORT.ROUTE.TARGET>
```

Example 13 Cloud Management SDN Configuration with VPN

network_type

The mechanism through which the Atlas northbound and southbound networks are implemented. The allowed value is `vxlan`.

vpn

A VPN has to be configured to provide connectivity for the Atlas VM. The following parameters are needed to create the VPN entity in the Cloud SDN Controller (CSC):

- `name` is a unique string value defining the name of the VPN. The length of the VPN name must not exceed the CSC REST API buffer size.
- `rd` is the Route Distinguisher of the VPN.
- `export_rt` is the export Route Target of the VPN. The value needs to be within quotation marks: `"value"`
- `import_rt` is the import Route Target of the VPN. The value needs to be within quotation marks: `"value"`



2.4.2.2 SDN Deployment with L2 GW Connectivity

For SDN deployment with L2 GW connectivity, configure the following extra cloud_mgmt parameters are needed:

```
cloud_mgmt:
  network_type: <NETWORK.TYPE>
  nbi:
    ...
    l2gw: <NBI.L2GW.FOR.ATLAS>
    l2gw_vlan: <VLAN.TAG.FOR.L2GW>
  sbi:
    ...
    l2gw: <SBI.L2GW.FOR.ATLAS>
    l2gw_vlan: <VLAN.TAG.FOR.L2GW>
```

Example 14 Cloud Management SDN Configuration with L2 GW

network_type

The mechanism through which the Atlas northbound and southbound networks are implemented. The allowed value is vxlan.

l2gw

The name of the Neutron L2 GW connected to the DC-GW. The L2 connectivity must be defined according to the section L2 Connectivity to DC-GW in the [OpenStack Networking API in CEE with SDN](#). To use the same L2 GW for both southbound and northbound interfaces, provide the same l2gw name under both nbi and sbi. Otherwise, provide a different L2 GW name for each.

l2gw_vlan

The unique VLAN ID used to connect the Atlas NBI or SBI networks with L2 GW.

2.5 Server Configuration

2.5.1 Shelf and Blade Management

The shelf or blade management sections must be edited with site-specific information. Remove the unused shelves (including definitions for the unused shelf) from config.yaml, or add further shelves if needed.

Some shelves are excluded from the following example for readability. The exact structure of the shelf or blade management information is hardware dependent. The hardware-specific details are shown in the sections below the example.

Note: The passwd and username parameters configured in this section are also used by fencing for out-of-band management access.



BSP

The shelf configuration options for BSP deployment are as follows:

```
ericsson:
  ...
  shelf:
    -
      id: 0
      shelf_mgmt:
        ip: <IP.SHELF.MANAGER>
        name: cee_ctrl_sp
        lct_ip: 10.0.10.2
        bsp_tenant: CEE
        passwd: <PASSWORD-SHELF.MANAGER>
        username: <USERNAME.SHELF.MANAGER>
        bsp_tenant: CEE
      blade:
        -
          id: 1
```

Example 15 BSP Shelf Configuration

Note: A shelf with `id: 0` must be present in `config.yaml` and contain `shelf_mgmt`, even if no blades are allocated to it.

The following shelf manager information has to be updated:

- `id` is the subrack ID.
- `ip` is the IP address of the DMXC (BGCI) interface.
- `passwd` is the administrator password of the single shelf manager.
- `username` is the administrator username of the single shelf manager.
- `bsp_tenant` is the name of the BSP tenant used by the CEE installation. During the installation, CEE checks if the BSP tenant already exists on BSP. If the BSP tenant already exists, the installation script removes the existing configuration of the BSP tenant and reconfigures it according to CEE requirement.

The blade ID is the position of the blade, $\text{blade ID} = (\text{slot ID} + 1) / 2$. For example:

- Blade ID 1 is the blade in slot 1.
- Blade ID 2 is the blade in slot 3.
- Blade ID 5 is the blade in slot 9.

Dell

The shelf configuration options for Dell deployment are as follows:



```
ericsson:
...
shelf:
-
  id: 0
  blade:
  -
    id: 1
    blade_mgmt:
      ip: <IP.FIRST.SERVER>
      name: subrack_ctrl_sp
      passwd: <PASSWORD.FIRST.SERVER>
      username: <USERNAME.FIRST.SERVER>
    ...
  -
    id: 2
    blade_mgmt:
      ip: <IP.SECOND.SERVER>
      name: subrack_ctrl_sp
      passwd: <PASSWORD.SECOND.SERVER>
      username: <USERNAME.SECOND.SERVER>
    ...
```

Example 16 Dell Multi-Server Blade Configuration

The following blade manager information has to be updated:

- ip is the IP address of iDRAC in each server.
- passwd is the administrator password.
- username is the administrator username.

The blade ID is the position of iDRAC. For example:

- Blade ID 1 is the first iDRAC.
- Blade ID 2 is the second iDRAC.

HDS

The following shelf configuration options are required for HDS:



```
ericsson:
...
shelf:
-
  id: 0
  shelf_mgmt:
    type: HDS
    api_url: https://<IP.TO.CCM.API>/<CUSTOMER.UUID>/<VPOD.UUID>
    username: <USERNAME.DCC.VPOD>
    passwd: <PASSWORD.DCC.VPOD>
  blade:
    -
      id: 1
      hw_uuid: <SERVER.UUID>
...

```

Example 17 HDS Shelf Configuration

- **<IP.TO.CCM.API>** is the IP address used for customer access to the HDS CCM API.
- **<CUSTOMER.UUID>** is the customer UUID of the vPOD owner.
- **<VPOD.UUID>** is the UUID of the vPOD.
- **<USERNAME.DCC.VPOD>** is the username for the vPOD where CEE is being deployed.
- **<PASSWORD.DCC.VPOD>** is the password for the vPOD where CEE is being deployed.
- **<SERVER.UUID>** is the UUID of the server used for all requests towards the CCM.

SDN Blade Configuration

Additional blade parameters are needed for SDN deployments on HDS versions earlier than 2.7 to configure HW-VTEP reachability on the compute hosts:

```
ericsson:
...
shelf:
-
  id: 0
  blade:
    -
      id: 1
      hw_vtep_subnet: <HW_VTEP_SUBNET>
      hw_vtep_gw: <HW_VTEP_GW>

```

Example 18 SDN HW-VTEP Configuration



Note: The `hw_vtep_subnet` and `hw_vtep_gw` values are mandatory on HDS versions earlier than 2.7, and can be retrieved from the switch. Contact the Data Center (DC) owner. On HDS 2.7 and later, the keys are ignored and the values are fetched from the Command Center Manager (CCM) API.

- `hw_vtep_subnet` is the HW-VTEP subnet address.
- `hw_vtep_gw` is the HW-VTEP gateway IP address, retrieved from the switch.

Unmanaged Server

Note: Unmanaged CEE deployment is not fully supported with documented procedures.

The blade configuration options for unmanaged CEE are as follows:

```
shelf:
-
  id: 0
  cee_managed: false
  blade:
  -
    id: 1
    hw_uuid: <SERVER.UUID>
    mac_assignment:
      control0: <LEFT.MAC.ADDRESS>
      control1: <RIGHT.MAC.ADDRESS>
```

Example 19 Unmanaged Server Configuration

Unmanaged server mode requires the servers to be discovered and preconfigured before running the installation. The `cee_managed` shelf configuration option needs to be set to `false`. The following blade configuration options have to be updated:

- `hw_uuid` is the UUID of the server to be included in applicable alarms. The `hw_uuid` parameter is optional, but if used, it must be defined for all servers. See Section 2.5.2 on page 23.

Note: This parameter only applies to HDS deployment.

- `control0` is the MAC address of the primary boot NIC of the server.
- `control1` is the MAC address of the alternate boot NIC (optional).

Managed ScaleIO Blade Configuration

To dedicate a blade to be part of the ScaleIO cluster in a managed ScaleIO solution, the role of the blade in the cluster has to be defined in the `scaleio` subsection of blade configuration. This section is optional and only has effect if the global `scaleio` parameters are defined in the `storage` section, see Section 2.9.1 on page 42. The following `blade.scaleio` parameters are available:



```
ericsson:
  shelf:
    blade
      - id: ..
        scaleio:
          mode: dedicated
          roles:
            mdm:
            gw:
            tb:
            sds:
              - protection_domain: domain1
                faultset: fs1
                devices:
                  - name: /dev/sdb
                    pool: pool1
                  - name: /dev/disk/by-partlabel/scaleio_sds_1
                    pool: pool2
```

Example 20 ScaleIO Blade Configuration Example

mode has the following available values:

- **dedicated**
- **shared**

Currently only the **dedicated** value is supported.

roles has the following available parameters:

- **mdm** is used to configure the blade as Meta Data Manager (MDM).
- **tb** is used to configure the blade as Tie-Breaker (TB) MDM.
- **gw** is used to configure the blade as ScaleIO Gateway (GW).
- **sds** is used to configure the blade as ScaleIO Data Server (SDS).

Multiple roles can be configured for a ScaleIO dedicated host, however, MDM and TB roles are mutually exclusive. The blade is configured with the combination of roles present in **roles**. For more information on each role, refer to the [EMC ScaleIO Version 2.0.x User Guide](#).

Note: The default configuration is a 5-node MDM cluster, consisting of three MDM and two TB servers. If there are more than three MDM or two TB servers defined, the extra nodes will be configured as standby MDM or standby TB, respectively.

For **sds** role, the following parameters need to be specified:



- `protection_domain` is the protection domain, as defined in the `scaleio` storage section.
- `devices` is the list of devices to be used for storage. Each device has to be listed in the form of a dictionary containing the following parameters:
 - `name` is the persistent device path. It is advised to use the `/dev/disk/by-path` of the device, as this path depends on the physical connection between the disk and the storage controller, and so it can be collected after Fuel installation on the kickstart server, with one blade powered on. Fuel boots a bootstrap image on the blade, where the `/dev/disk/by-path` IDs are the same as the post-deployment IDs. Issue the following command to retrieve the path for each device:

```
find /dev/disk/ |grep sas|grep -v part |sort | \
xargs sudo /sbin/hdparm -I | \
grep 'Serial Number\|Model Number\|path'
```
 - `pool` is the name of the pool where the device belongs. The storage pools are defined in the `scaleio` storage section.

Note: Make sure that the disk devices configured for ScaleIO are removed from the RAID configuration.

- `faultset` is an optional parameter used to configure Fault Sets.

Note: There are specific conditions to be met when configuring Fault Sets for SDSs. For more information on Fault Sets, refer to the Fault Sets section in the *EMC ScaleIO Version 2.0.x User Guide*.

In configurations using managed Extreme switches (Dell multi-server deployments), the cabling scheme YAML file must also be updated according to the allocation of ScaleIO ports in the traffic switch. The ScaleIO ports have the value `usage: storage`.

```
cabling_scheme:
  shelves:
    - blades:
        - blade_id: 1
          network interfaces:
            - {nic_id: 1, switch_id: 1, switch_port: 3, usage: storage}
            - {nic_id: 2, switch_id: 2, switch_port: 3, usage: storage}
            - {nic_id: 3, switch_id: 1, switch_port: 4, usage: storage}
            - {nic_id: 4, switch_id: 2, switch_port: 4, usage: storage}
```

Example 21 ScaleIO Cabling Scheme Configuration for Extreme Switches

2.5.2 Compute Hosts

Compute hosts are defined as a list of blades within each shelf. Each compute host must have an ID that corresponds to the physical/logical location of the server, see Section 1.1.1 on page 1. Assignment of physical NIC devices to different CEE



networks must be defined for each server. Memory (hugepages), CPU, and storage must also be allocated to different resource owners. The allocation of these resources is controlled through the configuration file.

A compute host can contain vCIC or vFuel VMs, or both. The allocation of resources depends on whether the host contains these infrastructure VMs or not.

hw_uuid

Note: For HDS deployment, `hw_uuid` is a mandatory blade parameter to be configured for CCM communication, see Page 19.

Certain CEE alarms contain the UUID of the compute host associated with the alarm. The UUID can be used to correlate CEE alarms with alarms generated by the lower-layer server management tools. By default, the UUID of each compute blade is obtained automatically during CEE install by running the following command:

```
dmidecode --string system-uuid
```

`hw_uuid` is an optional parameter for servers. If it is defined in `config.yaml`, the applicable alarms show the specified UUID instead of the default, automatically obtained one. If `hw_uuid` is used, it must be defined for all servers. If `hw_uuid` is not defined in `config.yaml`, the default `auto` value is assumed. The UUID can be assigned as follows:

```
ericsson:
  ...
  blade:
    -
      id: 2
      hw_uuid: <SERVER.UUID>
      ...
    -
      id: 3
      hw_uuid: <SERVER.UUID>
      ...
```

Example 22 Hardware System UUID Assignment

The value of the `hw_uuid` parameter must be either a valid UUID or `auto`.

2.5.3 Virtual Cloud Infrastructure Controller (vCIC)

In a multi-server deployment, three compute hosts must be configured to host a vCIC. In a single server deployment, the single compute host must be configured to host a vCIC.

To contain a vCIC, the corresponding blade must contain the `virt` key with a data structure that defines a vCIC as shown in the examples in the previous



subsections. In addition, the blade definition must contain resource reservations for hugepages, CPU, and disk storage suitable for vCIC.

2.5.4 Virtual Fuel

In a multi-server deployment, two compute hosts must be configured to be able to host a vFuel. In a single server deployment, while the vFuel VM is associated with the compute host, it remains on the kickstart server and does not migrate to the compute host.

To be able to contain a vFuel, the blade definition must contain resource reservations for hugepages, CPU, and disk storage suitable for vFuel. In addition, one of the two blade definitions must contain the `vfuel` key with an empty string as its value (see the previous sections for examples).

2.5.5 NIC Assignment

Each blade must have a `nic_assignment` section to define which physical NIC to use for control, storage, and data traffic. Each NIC is defined by its PCI address. The actual mapping depends on the cabling of the server.

The configuration template contains a list of predefined NIC assignments for supported hardware assuming the Ericsson-recommended cabling scheme. Predefined NIC assignments are listed in `nic_assignments` and each setting is labeled by an anchor. Normally, it is sufficient to refer to the appropriate predefined NIC assignment by using an alias in the blade definition.

```
ericsson:
  ...
  blade:
    -
      id: 2
      ...
      nic_assignment: *DELL_630_OEM_nic_assignment
      ....
```

Example 23 NIC Assignment of a Dell Server

```
shelf:
  cee_managed : false
  blade:
    id: 1
    hw_uuid: 4c4c4544-0030-3310-8039-b8c04f423232
    nic_assignment: *DELL_620_nic_assignment
    mac_assignment:
      control0: ec:f4:bb:c1:27:d4
      control1: ec:f4:bb:c1:27:d5
```

Example 24 NIC Assignment of an Unmanaged Server



Note: If `cee_managed` is **true**, the CEE is managed and the `mac_assignment` fields are ignored.

If the alias for a blade used in `nic_assignment` is not appropriate for the hardware, change the alias to refer to the relevant setting. The predefined values are found in the `nic_assignments` section of the configuration template. Refer to one of the available predefined settings by using its alias. If the hardware is not listed in `nic_assignments`, add a new NIC mapping to the list. See Section 3.6 on page 66 for information on how to define a new NIC mapping.

2.5.6 Memory Allocation

The physical memory of a server is partitioned into memory pages. Pages can have three different sizes: 4 KiB, 2 MiB, and 1 GiB. The 2 MiB and 1 GiB pages are called hugepages. In CEE, the memory is used as follows:

- 4 KiB pages are used by the host OS (hypervisor).
- 2 MiB pages are used by Open Virtual Switch (OVS) as buffers for Data Plane Development Kit (DPDK).
- 1 GiB pages are used for the memory of tenant and infrastructure VMs (vCIC and vFuel).

The memory to be allocated in hugepages is defined in `config.yaml` for each blade. The remainder of the physical memory that is not allocated in hugepages is accessible in 4 KiB pages. If using the keyword **auto** for the VM hugepage allocation, the installer calculates the hugepage count and ensures that they are evenly split between multiple NUMA nodes. If the number of hugepages is specified for VMs, the installer will reserve them evenly between NUMA nodes if an even number is configured. In case an odd number is configured, NUMA node 0 will be configured with one more hugepage than NUMA 1.

To reserve hugepages on a compute host, the `reservedHugepages` section of the corresponding blade must be filled in.



```
ericsson:
...
blade:
-
  id: 1
  nic_assignment: *BSP_GEP7_nic_assignment
  reservedHugepages: *reservedHugepages_with_vcic_and_vfuel
  reservedCPUs: *reservedCPUs_with_vcic_and_vfuel
  reservedDisk: *reservedDisk_for_vcic_and_vfuel
  virt:
    cic:
      id: 1
-
  id: 2
  nic_assignment: *BSP_GEP5_nic_assignment
  reservedHugepages: *reservedHugepages_with_vcic_and_vfuel
  reservedCPUs: *reservedCPUs_with_vcic_and_vfuel
  reservedDisk: *reservedDisk_for_vcic_and_vfuel
  virt:
    cic:
      id: 2
-
  id: 3
  nic_assignment: *BSP_GEP5_nic_assignment
  reservedHugepages: *reservedHugepages_with_vcic
  reservedCPUs: *reservedCPUs_with_vcic
  reservedDisk: *reservedDisk_for_vcic
  virt:
    cic:
      id: 3
-
  id: 4
  nic_assignment: *BSP_GEP5_nic_assignment
  reservedHugepages: *reservedHugepages
  reservedCPUs: *reservedCPUs
...
```

Example 25 Hugepage Reservations for Three Different Blades in BSP

The amount of memory to reserve in hugepages depends on whether the compute host contains a vCIC or vFuel, or both. The examples above illustrate different alternatives. For the recommended hugepage reservations, refer to the relevant System Dimensioning Guide:

- Multi-Server System Dimensioning Guide, CEE 6
- Single Server System Dimensioning Guide, CEE 6

Predefined hugepage reservations are listed in the `reservedHugepages` section. Refer to one of the available predefined settings by using an alias as shown in the examples.



The installer calculates the amount of memory to be allocated for each component and verifies its consistency with the total nominal memory of the physical server. The managed memory reservation can be configured using the `compute_os_reserved_mem` parameter, with the default value of 6 GiB. 6 GiB of managed memory is the minimum requirement. For the recommended allocation, refer to the [Multi-Server System Dimensioning Guide, CEE 6](#).

Note: The amount of unmanaged memory depends on the nominal physical memory of the server. The installer reads this value directly from the compute host.

The `compute_os_reserved_mem` configuration parameter is a dictionary with the nominal physical memory amount of the server as key, and the amount of memory to reserve as host OS managed memory as value. The values are in GiB. This parameter is defined globally for all blades in the CEE region. The configuration templates contain the following values:

```
compute_os_reserved_mem: {64: 6, 128: 6, 192: 6, 256: 6, 320: 6, 384: 6, 448: 6, 512: 6}
```

The example above shows the minimum required values. These parameters are defined globally and are valid for all blades.

The default vCIC swap space is 512 MiB. The swap space can be changed by setting the `vcic_swap_size` optional parameter. The value is expressed in MiB. For example, the following setting increases the swap space to 5 GiB:

```
vcic_swap_size: 5120
```

2.5.7

CPU Allocation

The CPUs of a compute host can be reserved for different purposes, such as OVS, tenant VMs, and infrastructure VMs (vCIC and vFuel). Reservation of the CPUs means that the reserved CPUs are used exclusively by the owner of the reservation. Reserved CPUs are isolated, which means that the kernel scheduler does not schedule processes to run on these CPUs by itself. The CPUs not reserved for these owners remain non-isolated and regular processes of the host OS are scheduled on these CPUs.

To reserve CPUs on a compute host, the `reservedCPUs` section of the corresponding blade must be filled in.



```
ericsson:
...
-
  id: 3
  blade_mgmt:
    ...
    nic_assignment: *DELL_630_OEM_nic_assignment
    reservedHugepages: *reservedHugepages_with_vcic
    reservedCPUs: *reservedCPUs_with_vcic
    reservedDisk: *reservedDisk_for_vcic
  virt:
    cic:
      id: 3
-
  id: 4
  blade_mgmt:
    ...
    nic_assignment: *DELL_630_OEM_nic_assignment
    reservedHugepages: *reservedHugepages
    reservedCPUs: *reservedCPUs
-
  id: 5
  blade_mgmt:
    ...
    nic_assignment: *DELL_630_OEM_nic_assignment
    reservedHugepages: *reservedHugepages_with_vfuel
    reservedCPUs: *reservedCPUs_with_vfuel
    reservedDisk: *reservedDisk_for_vfuel
    vfuel: ""
...

```

Example 26 CPU Allocation

If the compute host contains a vCIC or vFuel, or both, CPUs must be reserved for those as well. The examples above illustrate different alternatives. For the recommended CPU reservations, refer to the relevant System Dimensioning Guide. The CPU reservation anchors are predefined in the `reservedCPUs` section of the configuration template. Refer to one of the available predefined settings by using an alias, as shown in the examples.

Note: The default OVS and VM CPU reservation uses automatic CPU allocation in multi-server configurations but uses manual CPU allocation for single server deployments. See Section 3.1 on page 55 for details on the supported CPU allocation methods.

2.5.8 Disk Reservation

Storage space must be reserved on compute hosts that contain vCIC or vFuel. Use the `reservedDisk` section for reserving disk storage on a blade. Such disk reservation is not needed on compute hosts that do not contain infrastructure VMs.



```
ericsson:
  ...
  -
    id: 1
    ...
    nic_assignment: *DELL_630_OEM_nic_assignment
    reservedHugepages: *reservedHugepages_with_vcic
    reservedCPUs: *reservedCPUs_with_vcic
    reservedDisk: *reservedDisk_for_vcic
    virt:
      cic:
        id: 2
  -
    id: 2
    ...
    nic_assignment: *DELL_630_OEM_nic_assignment
    reservedHugepages: *reservedHugepages_with_vfuel
    reservedCPUs: *reservedCPUs_with_vfuel
    reservedDisk: *reservedDisk_for_vfuel
    vfuel: ""
    ...
```

Example 27 Disk Reservation for vCIC and vFuel

For the correct vFuel disk size value, refer to the relevant System Dimensioning Guide. The disk size value has to be given in GiB, with “G” appended after the value.

```
ericsson:
  ...
  - &reservedDisk_for_vfuel
    - owner: vfuel
      size: 80G
    ...
```

Or...

```
- &reservedDisk_for_vcic_and_vfuel
  - owner: vfuel
    size: 80G
  ...
```

Example 28 vFuel Disk Size 80 GiB

2.5.9 Host Networking

The used host networking YAML file must correspond to the used hardware configuration.



```
host_networking:
  template_yaml_file: host_nw_dell.yaml
```

Example 29 Host Networking in Dell Multi-Server Configuration

In case of HDS deployment, the host networking template must be selected based on whether CEE is deployed with or without SDN TI.

The following host networking templates are available:

Standard networking templates

- host_nw_bsp.yaml
- host_nw_dell.yaml
- host_nw_dell-single-server.yaml
- host_nw_hds-with-sdn.yaml
- host_nw_hds-without-sdn.yaml
- host_nw_hp.yaml

Networking templates for deployment without SAN connectivity

To deploy compute servers (not hosting a vCIC) in the whole CEE region without SAN connectivity, the following templates are available for the multi-server platforms:

- host_nw_bsp_without_compute_storage.yaml
- host_nw_dell_without_compute_storage.yaml
- host_nw_hds-with-sdn_without_compute_storage.yaml
- host_nw_hds-without-sdn_without_compute_storage.yaml
- host_nw_hp_without_compute_storage.yaml

The compute servers that are not hosting a vCIC will not have storage interfaces (storage switching domain) configured. The surplus interfaces can be used for other purposes, such as PCI passthrough (see Section 3.8 on page 71), SR-IOV Physical Function passthrough (see Section 3.8 on page 71), or SR-IOV (see Section 3.7 on page 67). Use the PCI addresses of the unused storage interfaces for SR-IOV, SR-IOV Physical Function passthrough, or PCI passthrough at interface configuration.

Networking templates for deployment with Glance on storage network

To deploy CEE with Glance on the storage network, the following templates are available:



- `host_nw_bsp-glance-on-storage-nics.yaml`
- `host_nw_dell-glance-on-storage-nics.yaml`
- `host_nw_hds-with-sdn-glance-on-storage-nics.yaml`
- `host_nw_hds-without-sdn-glance-on-storage-nics.yaml`
- `host_nw_hp-glance-on-storage-nics.yaml`

Note: The templates `host_nw_bsp-bm-glance-on-storage-nics.yaml`, `host_nw_dell-bm-glance-on-storage-nics.yaml`, and `host_nw_hp-bm-glance-on-storage-nics.yaml` are not supported for commercial use.

For more information on configuring Glance on the storage network, see Section 2.6.4 on page 39.

2.6 Networks

The `networks` section must be edited with site-specific information. See the following example for network configurations:

```
ericsson:
  ...
  networks:
    -
      name: cee_om_sp
      mos_name: public
      tag: <VLAN.TAG>
      enable_ntp: true
      cidr: <NETWORK.IP/PREFIXSIZE>
      start: <FIRST.IP.TO.USE>
      end: <LAST.IP.TO.USE>
      gateway: <IP.OF.THE.GW>
```

Example 30 Network Configuration

The network `fuel_ctrl_sp` is used for PXE boot of compute hosts (host OS) and vCIC nodes. For information on how to change this network, see Section 3.17 on page 86.

The following networks have to be configured based on hardware deployment:

Table 2 Hardware-dependent Network Configuration

Network name	HDS	Dell multi-server	BSP	Single server
<code>subrack_ctrl_sp</code>	N/A	to be configured	N/A	to be configured
<code>subrack_om_sp</code>	N/A	to be configured	N/A	N/A



cee_om_sp	to be configured	to be configured	to be configured	to be configured
cee_ctrl_sp	to be configured	preconfigured	to be configured	preconfigured
iscsi_san_pda	to be configured	preconfigured	to be configured	N/A
iscsi_san_pdb	to be configured	preconfigured	to be configured	N/A
swift_san_sp	to be configured	preconfigured	to be configured	N/A
migration_san_sp	to be configured	preconfigured	to be configured	N/A
hds_agent	to be configured	N/A	N/A	N/A

Note: In case of HDS and BSP deployment, the VLAN tag has to be configured for all relevant networks above.

For more information on configuring these networks, see Section 2.6.1 on page 33 and the site-specific IP and VLAN plan.

If using ScaleIO, additional networks must be configured, see Section 2.6.2 on page 36.

If Glance is configured on the storage network, the network `glance_san_sp` must be configured, see Section 2.6.4 on page 39.

Additionally, the following networks have to be configured for deployments with SDN:

- `sdnc_internal_sp`
- `sdnc_sbi_sp`
- `sdnc_sig_sp`

For more information on configuring the SDN networks, see Section 2.6.3 on page 37.

2.6.1 Hardware-dependent Network Configuration

`subrack_ctrl_sp`

Note: The `subrack_ctrl_sp` network is used in case of non-Ericsson hardware configurations.

The network `subrack_ctrl_sp` is used for subrack management. The VLAN is used to monitor and manage the server blades. This is done regardless of whether the blade is powered on, or if an operating system is installed or functional. Update `tag` with the correct VLAN tag, and `cidr` with the IP address in the `subrack_ctrl_sp` network used by Fuel.

The `start`, `end`, and `mos_name` parameters have to be configured for the CM-HA fencing feature, see Section 2.17 on page 52.

```
ericsson:
  ...
  networks:
    ...
    -
      name: subrack_ctrl_sp
      mos_name: subrack_ctrl_sp
      tag: <VLAN.TAG>
      vr: subrack_om
      ipforwarding: true
      cidr: <IP.IN.FUEL/PREFIXSIZE>
      start: <FIRST.IP.TO.USE>
      end: <LAST.IP.TO.USE>
    ...
```

Example 31 Subrack Management

- `tag` is the VLAN tag of subrack management. The VLAN is used to monitor and manage the server blades. This is done regardless of whether the blade is powered on, or if an operating system is installed or functional.
- `cidr` is the IP address used by Fuel in the `subrack_ctrl_sp` network.
- `start` is the first IP address in range for the subnet offered to the vCIC for the CM-HA fencing feature.
- `end` is the last IP address in range for the subnet offered to the vCIC for the CM-HA fencing feature.

Additional parameters must be set for `subrack_ctrl_sp` in the specific `<switch_model>_switch.yaml` file, see Section 2.3.1 on page 10.

subrack_om_sp

Note: The `subrack_om_sp` is used in case of non-Ericsson multi-server configurations.

The network `subrack_om_sp` is used as a link network between the traffic switches and the BGWs (DC-GWs). Subrack management traffic is routed over this network.



```
ericsson:
  ...
  networks:
    ...
    -
      name: subrack_om_sp
      tag: <VLAN.TAG>
      vr: subrack_om
      cidr: <NETWORK.IP/PREFIXSIZE>
      gateway: <IP.OF.THE.GW>
      ipforwarding: true
    ...
```

Example 32 Network subrack_om_sp

- tag is the VLAN tag of subrack management.
- cidr is the subnet of subrack management.
- gateway is the IP of the gateway.

Additional parameters must be set for subrack_om_sp in the specific <switch_model>_switch.yaml file, see Section 2.3.1 on page 10.

cee_om_sp

The network cee_om_sp is used for vCIC northbound communication:

```
ericsson:
  ...
  networks:
    ...
    -
      name: cee_om_sp
      mos_name: public
      tag: <VLAN.TAG>
      enable_ntp: true
      cidr: <NETWORK.IP/PREFIXSIZE>
      start: <FIRST.IP.TO.USE>
      end: <LAST.IP.TO.USE>
      gateway: <IP.OF.THE.GW>
    ...
```

Example 33 vCIC Northbound Communication

The number of dynamically assigned IP addresses inside cee_om_sp in the range **FIRST.IP.TO.USE** to **LAST.IP.TO.USE** must be at least seven, including the endpoints:

- The first, second, and third IP addresses in the range are individual addresses of the three vCICs.



- The fourth IP address in the range is always the vRouter address.
- The fifth address in the range is the CIC HA proxy.
- The last two IP addresses are reserved for vCIC internal use.

Note: For HDS configuration, the kickstart server IP also has to be provided, using the `kickstart` key.

hds_agent

The network used for HDS In-band Metrics Collection. The following parameters need to be configured:

- `tag` is the VLAN for the `hds_agent` network.
- `cidr` is the IP range of the `hds_agent` network.
- `start` is the first management IP address.
- `end` is the last management IP address.

2.6.2 ScaleIO Network Configuration

If using ScaleIO, the ScaleIO frontend networks must be configured. A pair of back-end networks can optionally be configured for managed ScaleIO. For more information on ScaleIO configuration, see Section 2.9.1 on page 42.

Note: The host networking configuration has to be aligned with the ScaleIO cluster:

- In case of CEE-managed ScaleIO, the `mos_name` of the configured networks must be added to the global `scaleio` configuration.
- In case of deploying CEE with unmanaged ScaleIO, the `mos_name` for each configured ScaleIO network must be manually configured in the external ScaleIO cluster.

Establishing connectivity between an unmanaged ScaleIO cluster and CEE is outside the scope of the customer documentation. Contact next level of support. The following connectivity must be configured:

- L2 connectivity between the SDCs and the SDSs
- L3 connectivity between the vCICs and the ScaleIO gateway
- L3 connectivity between the compute hosts and the ScaleIO gateway
- L2 and L3 connectivity between the compute hosts and the MDM

The configuration templates contain configuration details for both ScaleIO frontend and back-end networks, however, a VLAN tag has to be added to each configured ScaleIO network.



The following ScaleIO frontend networks must be configured:

sio_fe_san_pda

The left ScaleIO frontend network used for storage traffic.

— `mos_name` is **scaleio-frontend-left**

sio_fe_san_pdb

The right ScaleIO frontend network used for storage traffic.

— `mos_name` is **scaleio-frontend-right**

Optionally, the following ScaleIO back-end networks can be configured:

Note: Backend network configuration only applies to managed ScaleIO.

sio_be_san_pda

The left ScaleIO back-end network for SDC to SDS, MDM to MDM, MDM to SDS, and MDM to SDC traffic.

— `mos_name` is **scaleio-backend-left**

sio_be_san_pdb

The right ScaleIO back-end network for SDC to SDS, MDM to MDM, MDM to SDS, and MDM to SDC traffic.

— `mos_name` is **scaleio-backend-right**

For more information about ScaleIO network configuration, refer to the [ScaleIO Architecture Description](#).

2.6.3 SDN Network Configuration

The following additional networks have to be configured for SDN:

sdnc_internal_sp

`sdnc_internal_sp` is the network used for CSC internal communication. The CSCs running on the three vCICs use this network to communicate with each other and form a cluster.



```
name: sdnc_internal_sp
mos_name: mos-sdnc-int
tag: <VLAN.TAG>
cidr: <NETWORK.IP/PREFIXSIZE>
start: <START.IP>
end: <END.IP>
mip: <MOVABLE.IP.QUAGGA>
mip2: <MOVABLE.IP.BGP_MASTER>
```

Example 34 sdnc_internal_sp Configuration

mip and mip2 are the movable IPs of the sdnc_internal_sp network, used as single access IPs for all three vCICs.

mip is attached to the Quagga service and is used for OpenDaylight (ODL) to Quagga communication.

mip2 is attached to the BGP master resource and is used for Quagga to ODL communication. The ODL service runs in active-active-active mode on the three vCICs, but only one ODL instance has BGP as master. The ODL service will open port 6644 in the vCIC where BGP is master within the ODL service.

Both values must be:

- inside the cidr range
- outside the start – end range

sdnc_sbi_sp

sdnc_sbi_sp is the management network used for communication between the CSC and the Cloud SDN Switches (CSSs). sdnc_sbi_sp carries the OVSDB and Openflow traffic.

```
name: sdnc_sbi_sp
mos_name: mos-sdnc-sbi
tag: <VLAN.TAG>
cidr: <NETWORK.IP/PREFIXSIZE>
start: <START.IP>
end: <END.IP>
```

Example 35 sdnc_sbi_sp Configuration

sdnc_sig_sp

sdnc_sig_sp is the network used for Border Gateway Protocol (BGP) traffic between the CSCs and the DC-GW.



```

name: sdnc_sig_sp
mos_name: mos-sdnc-sig
tag: <VLAN.TAG>
cidr: <NETWORK.IP/PREFIXSIZE>
start: <START.IP>
end: <END.IP>
mip: <MOVABLE.IP.EXTRENAL.CONNECTIVITY>
gateway: <IP.OF.THE.GATEWAY>
bgp_gateway: [<BGP.GATEWAY.1.IP>, <BGP.GATEWAY.2.IP>]
bgp_neighbour: [<BGP.NEIGHBOUR.1.IP>, <BGP.NEIGHBOUR.2.IP>]

```

Example 36 sdnc_sig_sp Configuration

- mip is the movable IP of the sdnc_sig_sp network, used as a single access IP for all three vCICs. The mip is attached to the Quagga service and is used for communicating with the DC-GW for external connectivity. The value of this parameter must be:
 - Inside the cidr range
 - Outside the start – end range
- bgp_gateway is a list of the DC-GW A and DC-GW B interface addresses in the site-specific IP and VLAN plan.
- bgp_neighbour is a list of the BGP neighbor addresses. The length of the list must be equal to that of bgp_gateway.

2.6.4 Glance on Storage Network Configuration

Glance can be optionally configured on storage network, by configuring the glance_san_sp network.

Note: If glance_san_sp is configured, the appropriate host networking template must be selected, see Section 2.5.9 on page 30.

glance_san_sp is used for Glance image transfer traffic from the vCICs to the compute hosts. The Glance API on each vCIC use this network to communicate with each other and form a cluster. The Swift internal and admin endpoints use a virtual IP address on this network.

```

name: glance_san_sp
mos_name: glance
tag: <VLAN.TAG>
cidr: <NETWORK.IP/PREFIXSIZE>
start: <START.IP>
end: <END.IP>

```

Example 37 glance_san_sp Configuration

The following parameters have to be configured, based on the site-specific IP and VLAN plan, Reference [1]:



- `tag` is the VLAN for the `glance_san_sp` network.
- `cidr` is the subnet of the `glance_san_sp` network.
- `start` is the first IP address in range for the subnet offered to the vCIC and compute nodes for Glance on storage network.
- `end` is the last IP address in range for the subnet offered to the vCIC and compute nodes for Glance on storage network.

Refer to the *Multi-Server System Dimensioning Guide*, CEE 6.

2.7 Configure NTP

The `ntp_config` section contains NTP servers accessible by the vCIC. The section has to be updated with site-specific information.

```
ericsson:
...
ntp_config:
  servers: [<NTP.SERVER.1.IP>, <NTP.SERVER.2.IP>, <NTP.SERVER.3.IP>, <NTP.SERVER.4.IP>]
  orphan_mode_stratum: <ORPHAN.MODE.STRATUM>
...
```

Example 38 NTP Configuration

Define up to four external NTP server IP addresses. A minimum of one NTP server IP address is required for deployment, but four are recommended for CEE operation. For example, `servers: [IP-ADDRESS.FOR.NTP.SERVER-1, IP-ADDRESS.FOR.NTP.SERVER-2], IP-ADDRESS.FOR.NTP.SERVER-3]`.

When the NTP server running on one of the compute nodes enters NTP Orphan mode, the value of the parameter `orphan_mode_stratum` is used as stratum by that NTP server. To configure a correct value, the following criteria must be fulfilled:

- The value is in the range of 2 through 14.
- The value exceeds $N+2$ or be equal to $N+2$, where N is the maximum stratum of the upstream NTP servers. The upstream NTP servers are the NTP external sources for the CEE region.

2.7.1 NTP Authentication

The `ntp_config` section contains NTP authentication setup. The authentication can be set up between the compute host hosting the vCIC and the CIC upstream NTP servers.



```
ericsson:
...
ntp_config:
...
authentication: None
...
```

Example 39 NTP Authentication

To enable authentication towards the NTP servers upstream of the compute host that hosts the vCIC, set the md5 to the supported authentication method. Currently, the only available encryption method is md5.

```
ericsson:
...
ntp_config:
...
authentication: md5
...
```

Example 40 NTP Authentication, MD5

To enable authentication between the controllers in CEE and the upstream NTP servers, configure the following:

- group key
- group password

Configure for the upstream NTP servers, and for the controllers in CEE. The group key is a decimal number from 1 to 65534, inclusive. The group password consists of a printable ASCII string, less than or equal to 16 characters.

```
ericsson:
...
ntp_config:
...
authentication_upstream_group_key: 1
authentication_upstream_group_password: upstream_group_password
authentication_ericsson_group_key: 10
authentication_ericsson_group_password: ericsson_password
...
```

Example 41 Group Key and Password

Note: Enabling the NTP authentication requires the support and configuration of the upstream NTP servers.



2.8 Legal Text Presented at Logon

There are predefined messages in the `config.yaml` template that are shown at logon. These can be changed if needed.

If some specific legal text is to be displayed before logon attempts, update the predefined text in the section `legaltext`:

```
ericsson:
  ...
  legaltext:
    local: "Attention! Unauthorized local access is strictly prohibited!\n"
    remote: "\nAttention! Unauthorized remote access is strictly =>
prohibited!\n\n"
  ...
```

Example 42 Predefined Legal Text

Legal text has two items: `local` and `remote`:

- `local` is used for local logons, for example: serial console.
- `remote` is used for remote logons, for example: SSH.

Text can be formatted using normal C-style (`\t` for tab, `\n` for new line for example).

2.9 Storage

The storage section must be edited with site-specific information.

2.9.1 ScaleIO Configuration

ScaleIO block storage is configured in the `scaleio` storage subsection in `config.yaml`. ScaleIO can be configured to function in managed or unmanaged mode. In managed mode, a ScaleIO cluster is configured in `config.yaml` to be deployed during installation. In case of unmanaged mode, CEE uses an already deployed external ScaleIO cluster, and does not manage it. For more information on the ScaleIO feature and the configurable parameters, refer to the [EMC ScaleIO Version 2.0.x User Guide](#). ScaleIO can be configured using the `scaleio` section. To enable ScaleIO, the `cee_managed` key must be set to `true` (managed ScaleIO) or `client` (unmanaged ScaleIO).

Note: In case of unmanaged ScaleIO, the parameters must be aligned to the configuration of the already deployed ScaleIO cluster to be used. Additionally, the ScaleIO networks defined in `config.yaml` must be defined in the external ScaleIO cluster, see Section 2.6.2 on page 36.



```
ericsson:
  storage:
    scaleio:
      cee_managed: true
      protection_domains:
        - name: <PROTECTION_DOMAIN_NAME_1>
          pools:
            - name: <STORAGE_POOL_NAME_1>
              zeropadding: <enabled|disabled>
              types:
                - name: <VOLUME_TYPE>
                  provisioning_type: <thick|thin>
      frontend_networks: ['<FRONTEND_NETWORK_1>', '<FRONTEND_NETWORK_2>']
      backend_networks: ['<BACKEND_NETWORK_1>', '<BACKEND_NETWORK_2>']
      password: <SCALEIO_PASSWORD>
      gateway_admin_password: '<SCALEIO_GATEWAY_ADMIN_PASSWORD>'
      gateway_port: 4443
      gateway_user: '<cinder_username>'
      users:
        - name: '<cinder_username>'
          pwd: '<cinder_password>'
          role: 'FrontendConfigure'
        - name: '<username_N>'
          pwd: '<password_N>'
          role: '<Administrator|Security|BackEndConfigure|FrontEndConfigure|⇒
Monitor|Configure>'
```

Example 43 Managed ScaleIO Configuration Parameters

```
ericsson:
  storage:
    scaleio:
      cee_managed: client
      protection_domains:
        - name: <PROTECTION_DOMAIN_NAME_1>
          pools:
            - name: <STORAGE_POOL_NAME_1>
              zeropadding: <enabled|disabled>
              types:
                - name: <VOLUME_TYPE>
                  provisioning_type: <thick|thin>
      gateway_ip: '<SCALEIO_GATEWAY_IP_ADDRESS>'
      gateway_port: <SCALEIO_GATEWAY_PORT>
      gateway_user: '<cinder_username>'
      users:
        - name: '<cinder_username>'
          pwd: '<cinder_password>'
          role: 'FrontendConfigure'
```

Example 44 Unmanaged ScaleIO Configuration Parameters



The following parameters can be configured for ScaleIO:

cee_managed

- `true`: enable ScaleIO in managed mode to configure and deploy a ScaleIO cluster in CEE as storage back end.
- `client`: enable ScaleIO in unmanaged mode to connect CEE to an already deployed external ScaleIO cluster.
- `false`: disable ScaleIO.

protection_domains

This is a list of protection domains, where each listed element is a dictionary containing the `name` and `pools` parameters:

`name` is the name of the protection domain. `pools` is a list of the storage pools in the protection domain, in the form of dictionaries containing the following parameters:

- `name` is the name of the storage pool.
- `zeropadding` enables or disables the zero padding policy. The accepted parameters are **enabled** and **disabled**.
- `types` is a list of dictionaries containing the volume types defined for this pool. The dictionaries must consist of the following parameters:
 - `name` is the volume type name. Unique volume type names must be defined for each protection domain.
 - `provisioning_type` is the provisioning type of the volume type. The supported values are **thick** and **thin**.

frontend_networks

List of the `mos_name` of the ScaleIO frontend networks. This parameter only applies to managed ScaleIO. For more information, see Section 2.6 on page 32.

backend_networks

List of the `mos_name` of the ScaleIO back-end networks. This parameter only applies to managed ScaleIO. For more information, see Section 2.6 on page 32.

password

`password` is the password for the MDM admin user. This parameter only applies to managed ScaleIO.

**gateway_admin_password**

gateway_admin_password is the password for the gateway admin user, in quotes. This parameter only applies to managed ScaleIO.

gateway_ip

The gateway IP address of the ScaleIO cluster. This parameter only applies to unmanaged ScaleIO.

gateway_port

The TCP port that the gateway exposes to the REST API interface. For managed ScaleIO, the value must be 4443.

gateway_user

The username to the gateway REST API interface. The user must be defined under the users key with FrontEndConfigure role.

users

The ScaleIO user accounts, in the form of dictionaries containing the following parameters:

- name is the username.
- pwd is the password of the user.
- role is the user role. The following user roles can be defined: Administrator, Security, BackEndConfigure, FrontEndConfigure, Monitor, Configure.

Note: In unmanaged mode, only one user is to be defined, with FrontEndConfigure role, which must be the same user that was created in the ScaleIO cluster for this particular CEE domain.

To dedicate a blade to be part of the ScaleIO cluster in managed ScaleIO mode, the role of the blade in the cluster has to be defined in the blade section, see Page 21.

2.9.2

Local Storage (SSD)

fstrim is used on mounted file systems to discard blocks which are not in use by the file system. fstrim can be configured in the following ways:

- Configured using the default Canonical settings
- Configured using low priority settings
- Disabled



For more information on `fstrim`, see the SSD `fstrim` Configuration section in the CEE Architecture Description, Reference [6].

The structure of the `fstrim` section in `config.yaml` is as follows:

```
blade:
...
  fstrim:
    disabled: <true | false>
    type: <default | ericsson>
...
```

Example 45 Ericsson Modified `fstrim` Configuration

disabled

To completely disable the automatic `fstrim`, set to **true**. If set to **false**, `fstrim` runs automatically once a month.

type

Note: If `disabled` is set to **true**, this parameter does not apply.

The following parameters are available:

- **ericsson**: to run `fstrim` with lower scheduling class and priority (`ionice -c 3 nice -n 19`), set to **ericsson**. In this case, the TRIM is applied in 5 GB chunks in five second intervals.
- **default**: for the default Canonical best-effort settings (`ionice -c 2 -n 2`), set to **default**.

This is an optional parameter. If the `type` parameter is omitted, the **default** configuration will apply.

2.9.3

Software RAID Configuration

The `swraid` Boolean key is used to enable software RAID configuration on a per compute blade basis.

Note: Only RAID 1 is supported, and the compute blades must have two physical disks to use this feature. Provisioning can take more time than usual, as RAID 1 is created during this phase.

The write I/O operations on the installed system will be impacted due to the RAID 1 configuration. The disk capacity of the compute blade will be reduced by half, as a result of RAID 1.

The following values are valid:

- **True**: RAID configuration is enabled.



— **False:** RAID configuration is disabled.

`swraid` is an optional parameter. If undefined, RAID is not configured on the compute blade.

```
ericsson:
  ...
  shelf:
    -
      id: 0
      blade:
        -
          id: 1
          blade_mgmt:
            ...
            swraid: True
            ...
```

Example 46 Software RAID Configuration

2.10 Local Disk Partition Sizes

The section `localdisks` has local disk partition sizes for local disk (for compute hosts) and partition sizes of virtual disk (for vCIC). The sizes are in **mebibyte (MiB)**. The sizes are applied to all partitioning for vCIC (guest OS) and Host OS. The default settings in `config.yaml` are the minimum requirement and cannot be decreased. On compute nodes, the remaining disk space is allocated to `/var/lib/nova` (for data that includes local storage and ephemeral disks).

On vCIC nodes, the remaining disk space is allocated to `/var/lib/glance` and is used for Glance, Swift, and CIC Domain Data Backup.

The partition sizes must be dimensioned according to the site-specific needs. The `mysql_size` value is not filled in the template, must be added based on the System Dimensioning Guides:

- Multi-Server System Dimensioning Guide, CEE 6
- Single Server System Dimensioning Guide, CEE 6

Note: Consider the size of virtual disk available to vCIC.



```
ericsson:
...
storage:
  localdisks:
    cic:
      os_size: 51200
      logs_size: 40960
      mysql_size: <MYSQL_SIZE_IN_MiB>
      mongo_size: 40960
    compute:
      os_size: 51200
      logs_size: 40960
...
```

Example 47 Local Disks

Note: Do not define `mongo_size` in a single server deployment.

2.11 IdAM

IdAM is used for managing the system administrator accounts. The `idam:` section includes credentials for the IdAM component. It allows setting initial passwords for the predefined accounts needed for CEE to operate.

The `ldap` section includes credentials of LDAP entities used exclusively by infrastructure applications. The recommendation is to leave these credentials blank to enable the system to use generated passwords.

It is possible to set the initial password of the `ceeadm` and `ceebackup` in the `users` section:

```
ericsson:
...
  users:
    ceebackup:
      idam_tag:
        - admin
      passwd: ' <IDAM.CEEBACKUP.PASSWD> '
    ceeadm:
      idam_tag:
        - none
      passwd: ' <IDAM.CEEADM.PASSWD> '
...
```

Example 48 Password for the ceeadm Predefined User

Passwords must be quoted by using single-quotes and must be compliant with CEE password policy, otherwise the deployment fails. The minimal accepted password length is 12 characters. There is no factory default password.



Note: cebackup is not applicable to single server deployment.

2.12 LDAP Users

The LDAP section includes credentials used by infrastructure applications, if these credentials are left blank in the `config.yaml`, the entries from the `idam_ldap_*_password` section in `/etc/openstack_deploy/user_secrets.yaml` are used.

```
ericsson:
  idam:
    ldap:
      basedn: <IDAM.LDAP.BASE>
      rootdn: <IDAM.LDAP.ROOTRDN>
      rootpw: ''
      anonymous_binddn: <IDAM.LDAP.ANONRDN>
      anonymous_bindpwd: ''
      manager_binddn: <IDAM.LDAP.MNGRRDN>
      manager_bindpwd: ''
      sync_binddn: <IDAM.LDAP.SYNCRDN>
```

Example 49 LDAP

2.13 Glance Image Service

The Glance API server can be configured to have an optional local image cache. A local image cache stores a copy of image files, essentially enabling multiple API servers to serve the same image file. This increases scalability due to an increased number of endpoints serving an image file.

The Glance image cache is disabled in the configuration file templates.

Enable the local image cache for CEE deployments where there is sufficient disk space available for Glance image cache on local disks. For CEE deployments where the local disk space is limited it is more favorable to disable the function or to reduce the disk space used for the local image cache on the `/var/lib/glance` disk partition.

Two parameters are available in the storage section of `config.yaml` to manipulate the Glance image cache function:

- Set `enable_glance_image_cache` to **true** to activate the Glance image cache.
- Use `image_cache_max_size` to define the maximum size of the disk space in bytes used for Glance image cache. The value set in the `config.yaml` templates for `image_cache_max_size` is 5368709120 bytes.

Glance can optionally be configured on the storage network. For more information, see Section 2.6.4 on page 39.



2.14 GRUB Configuration

Credentials for accessing Grub menu entries must be configured in the grub section. The GRUB username and password are mandatory parameters.

```
ericsson:
  grub:
    grub_username: <SUPER_USERNAME>
    grub_password: ''
```

Example 50 GRUB Configuration

For successful deployment, the grub_password value must be in single quotes and must be compliant with the CEE password policy. The minimum accepted password length is 12 characters.

Note: There is no default password. The parameter must not be left empty.

2.15 Swift Configuration Options

The swift section in the configuration file template allows to configure Swift to use the back-end storage system, and by this to move the location of the Swift store from the local disks to the back-end storage. Currently, the supported storage back end for Swift is ScaleIO.

Note: The prerequisite to configure Swift on back-end storage is as follows:

- To use ScaleIO as storage back end: a properly configured ScaleIO block storage, see Section 2.9.1 on page 42.

If the prerequisite is not fulfilled, the swift_on_backend_storage section is not applicable and will be ignored.

The following configuration options are available:

```
ericsson:
  ...
  swift:
    swift_on_backend_storage:
      type: <SWIFT.SWIFT_ON_BACKEND_STORAGE.TYPE>
      activation_mode: <SWIFT.SWIFT_ON_BACKEND_STORAGE.MODE>
      lun_size: <SWIFT.SWIFT_ON_BACKEND_STORAGE.SIZE>
```

Example 51 Swift Configuration Options

type

The type of the back-end storage system. To configure Swift to use ScaleIO as storage back end, the type must be set to **scaleio**.



activation_mode

The supported values are **manual** and **automatic**. The value set in the `config.yaml` templates is **manual**.

To deploy Swift on ScaleIO manually after installation, refer to the [Swift Store on ScaleIO Activation](#) operating instructions.

The **automatic** activation mode can be used to deploy Swift on back-end storage during installation.

lun_size

`lun_size` specifies the LUN size of Swift on the storage back end. The value must be given as an integer value followed by the unit (GiB or TiB). The value set in the `config.yaml` templates is **0GiB** in combination with `activation_mode` set to **manual**.

If `activation_mode` is set to **automatic**, a value different from **0GiB** must be used. The unit has to be given in GiB or TiB. The minimum value is **1GiB**.

Note: ScaleIO only supports volumes with a granularity of 8 GiBs. As a result, the physical size of the LUN will always be rounded up to the nearest multiple of 8 GiBs, while Cinder is not aware of this rounding and uses the given size. Therefore, it is recommended to set a `lun_size` which is a multiple of 8 GiBs.

2.16 SDN Integration on HDS

To enable the SDN integration feature of HDS, the `sdn` section has to be defined in `config.yaml`. This feature enables the user to deploy CEE with the CSC integrated, without any other extension package. The CSCs running inside CEE vCICs will be used to manage the tenant network and create tunnels on compute blades for tenants network isolation. Software VTEPs are created automatically during CEE deployment. The parameters needed for SDN configuration are as follows:

```
sdn:
  sdnc_admin_username: <SDNC.ADMIN.USERNAME>
  sdnc_admin_password: <SDNC.ADMIN.PASSWORD>
  remote_gre_term: <REMOTE.GRE.TERM.IP>
```

Example 52 SDN Configuration

sdnc_admin_username

Username for CSC authentication

**sdnc_admin_password**

Password for CSC authentication

remote_gre_term

The DC-GW loopback IP(s). If there are multiple IP addresses, they are to be added as a list, for example: ['10.33.215.30', '10.33.215.31']

2.17 CM-HA

The `cmha` section of `config.yaml` is used to configure the Continuous Monitoring High Availability (CM-HA) service, and includes the following parameters:

Note: The parameters below are optional. If these parameters are not included in `config.yaml`, the default values apply.

fence_compute_before_evacuation

If `fence_compute_before_evacuation` is **true**, CM-HA is to fence down the compute host before evacuating the VMs. This function is to prevent VM duplication in case of a partial compute failure. Default value: **true**

try_to_recover_compute_after_evacuation

If `try_to_recover_compute_after_evacuation` is **true**, CM-HA attempts to power on the compute host after finishing the evacuation of the VMs. This function can help recover the failed compute. Default value: **true**

Note: In the HDS with SDN configuration template, the default value for `fence_compute_before_evacuation` and `try_to_recover_compute_after_evacuation` is **false**.

In case of non-Ericsson hardware, the `subrack_ctrl_sp` network has to be configured with `start`, `end`, and `mos_name` parameters for vCIC access, see Section 2.6 on page 32.

2.18 Fuel Plugins

The `fuel-plugins` section of `config.yaml` is used for the installation and configuration of Fuel plugins. For more information on Fuel plugins, including the plugin name, configuration attributes, and the list of mandatory and optional Fuel plugins, refer to the [Fuel Plugin Configuration Guide](#).

The `fuel-plugins` section includes the following parameters:



```
ericsson:
  ...
  fuel-plugins:
    -
      name: <PLUGIN-NAME>
      config_attributes:
        <ATTRIBUTE1>: <VALUE1>
        <ATTRIBUTE2>: <VALUE2>
    -
      name: <PLUGIN-NAME>
  ...
```

Example 53 Fuel Plugin Configuration

name

The name of the Fuel plugin to be installed.

Note: The variable `<PLUGIN-NAME>` must match the name mentioned in the `metadata.yaml` file of the Fuel plugin.

config_attributes (optional)

Any configuration attributes needed for the plugin are added in the `config_attributes` section using the below structure:

```
...
config_attributes:
  <ATTRIBUTE1>: <VALUE1>
  <ATTRIBUTE2>: <VALUE2>
...
```

If no configuration attributes are specified, the default Fuel plugin values are automatically set.

2.19 Change of Linux I/O Scheduler

The user must select the I/O scheduler for the compute hosts. This changes the strategy used for scheduling I/O requests. The parameter is global and applies to all compute hosts.

```
ericsson:
  ...
  timezone: Etc/UTC
  compute_io_scheduler: <IO.Scheduler>
  ...
  neutron:
  ...
```

Example 54 Selecting IO Scheduler Options



The following values can be configured for `compute_io_scheduler`:

- `deadline` starts the Deadline scheduler that caps the maximum latency per request, and maintains high disk throughput.
- `cfq` starts the Completely Fair Queuing scheduler that is used for maintaining system-wide fairness of I/O bandwidth.
- `noop` starts the simple NOOP scheduler. NOOP is used for memory-backed block devices such as RAM disks, and non-rotational media such as flash.

2.20 License Management Configuration

Note: The `ericsson_sheriff` Fuel plugin must be enabled in the `config.yaml` for license management. Refer to the [Fuel Plugin Configuration Guide](#).

The Sheriff license management can be configured using the `license` key in the `config.yaml`. The following configuration options are available:

```
ericsson:
  ...
  license:
    nels_server_host: <NELS.SERVER.IP>
    nels_server_port: <NELS.SERVER.PORT>
    nels_server_trusted_hostname: <NELS.TRUSTED.HOSTNAME>
    tls_trusted_ca_certificate: <TRUSTED.CA.CERT>
    tls_nels_client_private_key: <NELS.CLIENT.PRIVATE.KEY>
    tls_nels_client_certificate: <NELS.CLIENT.CERT>
    openssl:
      ciphersuites: <CIPHERSUITES>
      protocols: <SSL.PROTOCOL>
```

Example 55 Example License Management Configuration

NeLS Server Configuration

The NeLS server can be configured using the following parameters:

- `nels_server_host`: The hostname or IP of the NeLS server.
- `nels_server_port`: The port to be used on the NeLS server.
- `nels_server_trusted_hostname`: The trusted hostname of the NeLS server.

Certificate Configuration

Note: The files needed for certificate configuration must be copied to `/mnt/cee_config/` before CEE deployment.

The certificates can be configured using the following keys:



- `tls_trusted_ca_certificate`: The certificate for authentication of the TLS connection.
- `tls_nels_client_private_key`: The file containing the private key for the TLS connection.
- `tls_nels_client_certificate`: The client certificate for the NeLS authentication.

SSL Configuration

The SSL encryption can be configured using the `openssl` key, which has the following subkeys:

- `ciphersuites`: The string containing the list of allowed OpenSSL cipher suites.
- `protocols`: The protocols to be used for TLS.

Note: In the current release, only the TLSv1.2 protocol is possible to be used.

3 Advanced Parameter Settings

This section contains the advanced configuration options of `config.yaml` and explains how to change the default values used in the templates.

Note: Not all possible combinations have been formally verified by CEE Integration and Verification.

3.1 Advanced CPU Allocation

To reserve CPUs on a specific compute host, the `reservedCPUs` section of the corresponding blade must be filled in with information on the reservation. Reservation for a server consists of a list of CPU reservations. Each item in the list represents the CPU reservation for a specific system component (owner). The reservation for a system component is defined by a mapping (also called hash or dictionary in programming languages) containing the keys `owner`, `count`, and `cpus`.

The `owner` key is mandatory. It specifies the component for whom the reservation is intended. Supported owners are `vm`, `ovs`, `vcic`, and `vfuel`.

Note: It is possible to reserve CPUs for not supported owners. The CPUs reserved for such owners remain idle since they are added to the list of isolated CPUs and none of the supported system components use them.



Either the `count` or the `cpus` key must be defined, but not both. Deviations from this rule are explained later.

The value of the `count` key is either the string `auto` or an integer. When the value `auto` is used, the CEE installer determines how many CPUs to reserve for the given owner and which ones. If the value is an integer, it defines the number of CPU cores to reserve. Due to hyperthreading, one core corresponds to two CPUs. When a CPU core is reserved for an owner, both CPUs on that core are dedicated to that owner. The System Dimensioning Guides provide information about the relation between CPUs and cores for supported hardware models.

Refer to the following documents:

- Multi-Server System Dimensioning Guide, CEE 6
- Single Server System Dimensioning Guide, CEE 6

The value of the `cpus` key is a list of CPU IDs or ID ranges. The ranges are inclusive. By using the `cpus` key, the cloud administrator can directly control the allocation of CPUs to owners. Although it is possible to reserve CPUs by mixing allocations that use the `count` and the `cpus` keys for different CPU owners (`ovs`, `vm`, `vfuel`, `vcic`), this practice is not recommended, because the configuration validation before deployment may fail.

Note: Reserving CPU 0 for any owner is not recommended. The CPUs reserved for any owner in `config.yaml` are added to the kernel list of isolated CPUs, ensuring that the kernel does not schedule common host OS processes to run on the isolated CPUs. The kernel uses CPU 0 during boot, and does not enable any other CPUs until late in the boot process. The `isolcpus` boot parameter of the kernel does not move already started processes (such as driver or worker threads) off CPU 0, only ensures that processes started subsequently will not use the isolated CPUs by default. Therefore, it is possible that processes keep using CPU 0 even if it is listed in the `isolcpus` kernel parameter.

3.1.1 Automatic CPU Allocation Rules

The CEE installer allocates CPU resources using the rules described here when only the `count` key is used in CPU reservations.

Note: These rules are not applicable when the CPUs are reserved manually using the `cpus` key.

Two CPU cores are assigned to the host OS (that is, left as non-isolated) when the reservation `count: auto` is used for the owner: `vm`. If the number of CPU cores to be reserved for VMs is explicitly set, or no CPU reservation is made for VMs, then all remaining CPU cores are used by the host OS. It is a configuration error if there are no CPU cores left for the host OS.

Six CPU cores are reserved for vCIC, and one CPU core is reserved for vFuel, if `count: auto` is used. The `auto` configuration is only applicable to small regions, for proper dimensioning, refer to the [Multi-Server System Dimensioning Guide](#),



CEE 6. The installer attempts to allocate all cores from the same NUMA node, starting with NUMA 0. If unsuccessful, an attempt is made to split the cores evenly across all NUMA nodes. If the cores cannot be evenly split across the NUMA nodes, the installer will allocate as many cores as possible from NUMA 0, then from NUMA 1. If this is not possible, it will follow the same logic starting from NUMA 1. If the installer cannot find enough available cores to satisfy the request, CEE installation fails.

CEE deployment fails if the CPU allocation succeeds, but no sufficient memory is available in the NUMA node. In this case, the `cpus` key must be used to manually distribute the cores over the available NUMA nodes, as memory is allocated proportionally to the amount of reserved CPU per NUMA node. For example: in case of more than one NUMA nodes and 64 GiB RAM: if the 30 GiB is reserved for the vCIC, and all cores fit in a single NUMA node, an insufficient amount of memory is available in that NUMA node for the host OS. As a result, CEE deployment fails. In this case, the user can manually split the required number of cores equally across the two NUMA nodes, so 15×1 GiB hugepages will be allocated from each node for the vCIC.

The first CPU core on each NUMA node is assigned to the host OS (left non-isolated) if the number of required host OS cores are at least equal to the number of NUMA nodes. Then one CPU core is reserved for OVS PMD threads on each NUMA node. On NUMA node 0 (and NUMA node 1, if it exists), the first one or two available cores are reserved for OVS PMD threads, depending on the `css_mode` configuration (see Section 3.1.2 on page 57). After that, the CPUs are reserved for vCIC, vFuel, and tenant VMs, in this order. For more information on CSS auto configuration options, see Section 3.1.2.1 on page 58.

The first CPU used by the host OS is also assigned to the OVS control process unless it is explicitly defined. See Section 3.1.2 on page 57 for details.

3.1.2 Allocating CPUs for CSS (OVS)

Note: Open vSwitch (OVS) is also referred to as Ericsson Cloud SDN Switch (CSS).

Note: After CEE deployment, the CSS CPU allocation can only be changed during server replacement. For details, refer to the [Runtime Configuration Guide](#).

OVS requires CPUs for different purposes. Some of the CPUs are used for poll-mode driver (PMD) threads. The PMD threads continuously poll the physical and virtual NICs to check if there is new data. It is crucial that no other tasks are scheduled on the CPUs assigned to OVS PMD threads for performance reasons. In addition, one CPU must be specified for the OVS control process. This process does not have special requirements on its CPU. It can share one of the non-isolated CPUs with the other host OS processes. CPUs reserved for the owner `ovs` in `config.yaml` are used for PMD threads

Note: At least one CPU must be reserved for PMD threads on each NUMA node. Failure to fulfill this constraint leads to system malfunction.



Refer to the [System Dimensioning Guide](#) for the details on CPU allocation for CSS.

There are two main options for allocating CPUs for CSS (OVS), using the mandatory key `css_mode`:

- Automatic configuration, using one of the following modes: `normal-perf`, `high-perf`, `ultra-perf`. For more information about using the automatic configuration, see Section 3.1.2.1 on page 58.
- Custom configuration, using the `custom` mode. If this mode is selected, the `count` or the `cpus` key can be used to allocate a number of cores, or specific cores, respectively. For more information about using the custom configuration, see Section 3.1.2.2 on page 59.

`css_mode` is a mandatory key if CPUs are allocated using `count` or `cpus` (custom configuration).

The appropriate CPU reservation has to be set for each blade in the `blade` section, depending on whether it hosts vCIC or vFuel. An example is:

```
ericsson:
  ...
  shelf:
    ...
    blade:
      -
        id: 1
        blade_mgmt:
          ...
          nic_assignment: *DELL_630_nic_assignment
          reservedHugepages: *DELL_630_reservedHugepages
          reservedCPUs: *OVS_single_thread_DELL_⇒
630_reservedCPUs
```

3.1.2.1 Automatic CPU Allocation for CSS

Note: CSS version R10A (Fuel plugin version 1.0-1.0.13-1) or later is required for `normal-perf`, `high-perf`, and `ultra-perf` configuration. For automatic CPU allocation using an earlier CSS version, use the `count: auto` configuration without defining `css_mode`.

Cores can be automatically reserved for CSS by setting the `css_mode` key to either of the following values:

- `normal-perf`: One core is allocated for CSS per NUMA node (1 physical CPU + 1 hyper-thread CPU).
- `high-perf`: Two cores are allocated for CSS per NUMA node (2 physical CPUs).



- **ultra-perf:** Two cores are allocated for CSS per NUMA node (2 physical CPUs + 1 hyper-thread CPU).

Note: The **ultra-perf** configuration is an experimental feature with limited verification. Refer to the CSS User Guide, Reference [9].

For details about the exact CPU (physical and hyper-thread sibling) reservation methods used for **normal**, **high** and **ultra** performance allocations and for the performance implications, refer to the CSS User Guide, Reference [9].

Compute servers with only one NUMA node (for example, BSP deployment) provide decreased CSS performance with the automatic allocation method compared to platforms with two (for example: Dell, HDS deployment) or more NUMA nodes.

Note: If the `css_mode` key is not defined in `config.yaml`, `normal-perf` is configured by default.

In case of automatic CPU allocation, `count: auto` must be used.

```
ericsson:
  ...
  reservedCPUs:
  - &CSS_auto_reservedCPUs
    - owner: vm
      count: auto
    - owner: ovs
      count: auto
      css_mode: high-perf
```

Example 56 Example Auto Core Allocation for CSS

3.1.2.2 Custom CPU Allocation for CSS

In case of custom CPU allocation, the `count` or the `cpus` key must be used to allocate cores. If the `count` key is used, the value must be an integer, not `auto`. The `css_mode` key must be defined with the value `custom`.

Note: Using custom CPU allocation when `css_mode` is not defined as `custom` results in installation failure.

If the `count` key is used to specify the allocated cores for reserving the OVS PMD CPUs, at least one CPU core per NUMA node must be allocated to OVS PMD threads. For example, at least two cores are required on Dell and HDS compute servers, while only one is required on BSP. In this case, both the physical CPUs and the hyper-threading siblings of the respective cores are allocated for the OVS PMD threads, and added to the `isolcpus` and `nohz_full` boot parameter lists of the kernel. Any additional cores are allocated sequentially from the available cores, starting with NUMA node 0 and then continuing with NUMA node 1, if necessary.



```
ericsson:
...
  reservedCPUs:
  - &CSS_custom_3_reservedCPUs
    - owner: vm
      count: auto
    - owner: ovs
      css_mode: custom
      count: 3
```

Example 57 Example Custom CPU Allocation for CSS

In Example 57, three cores are allocated for CSS. An even more precise allocation option is to specify the exact CPUs to be allocated using the `cpus` key, with the value being a list of CPU IDs or ID ranges.

To allocate a single CPU per core for OVS, new definitions have to be created under `reservedCPUs` for each compute host and deployment scenario (with or without vCIC and vFuel). An example is:

```
ericsson:
...
  reservedCPUs:
  - &CSS_single_thread_DELL_630_reservedCPUs
    - owner: vm
      cpus: 1,3,5-22,25,27,29-46
    - owner: ovs
      css_mode: custom
      cpus: 2,23
      cpus_nonpmd: 0
    - owner: idle
      cpus: 26,47
      notick: true
```

For more information on the CSS CPU reservation methods, refer to the CSS User Guide, Reference [9].

Of the two CPUs in a core, the lower numbered must be reserved for OVS.

The hyper-thread siblings of OVS PMD CPUs must be allocated to the owner **idle** and the `notick` allocation parameter is set to the value **true**.

This results in the following:

- These CPUs are listed in the kernel `isolcpus` boot parameter list, ensuring that the process scheduler does not assign any processes to these CPUs apart from the ones allocated for OVS OMD threads (`owner: ovs`).
- These CPUs are listed in the kernel `nohz_full` boot parameter list, ensuring that the kernel does not generate scheduling clock interrupts on these CPUs apart from the ones allocated for OVS OMD threads.



By default, the OVS control process uses the first CPU allocated for the host OS. This can be overruled by defining the `cpus_nonpmd` key. The recommended value is a single CPU ID. Only non-isolated CPUs are allowed. That is, CPUs reserved for other purposes (including OVS PMD threads) are not allowed. Make sure that the constraints listed above are fulfilled.

Note: Although it is possible to define a set of CPUs for the OVS control process, it is currently not recommended to assign more than one CPU.

OVS is configured to use DPDK only if CPUs and huge pages are both reserved for the owner `ovs`. OVS is configured to run without DPDK if there are no reservations. It is a configuration error if only one of CPUs or huge pages are reserved. VMs must also use hugepages to be able to communicate over OVS/DPDK. Therefore, it is mandatory to reserve 1 GiB huge pages for VMs if OVS/DPDK is used.

Note: It is not possible to configure a CEE region without DPDK acceleration for traffic network (provided by the corresponding OVS bridges).

3.1.3

Resource Allocation for vCIC

The CPU allocation for vCIC (owner: `vcic`) supports the boolean key `isolated` to control if dedicated CPUs have to be allocated for the vCIC VM. Its value is either the default `true` or `false`.

If `isolated: true` is configured or the key is omitted, the vCIC uses dedicated physical CPUs, and the virtual CPUs of the vCIC VM are mapped one-to-one to the reserved physical CPUs. In this case, the hugepages are taken from the same NUMA nodes to which the physical CPUs belong. In case cores are allocated to the vCIC from two NUMA nodes, the huge pages are allocated in proportion to the number of CPUs on each NUMA node.

If `isolated: false` is configured, no physical CPUs are reserved for the vCIC. Instead, it shares the non-isolated physical CPUs, or a subset of them, with the host OS processes. If the `cpus` key is defined, it must be a subset of the non-isolated CPUs and the vCIC only uses these physical CPUs. If `cpus` is not defined, the vCIC uses all non-isolated CPUs. If the `count` key is defined, it is used to calculate the number of vCPUs in the vCIC. The number of vCPUs to use is calculated as $\text{count} \times \text{HTs} / \text{core}$. The resulting number of vCPUs cannot be higher than the number of physical CPUs on which the vCIC is allowed to run.

```
ericsson:
...
  reservedCPUs:
    - owner: vcic
      count: 2
      isolated: false
...
```

Example 58 Use non-isolated CPUs for a vCIC with 4 vCPUs

Note: Using the `isolated: false` CPU allocation is not recommended on multi-server systems.



In case `isolated: false` is used for CPU configuration, then in the memory allocation for the `vcic` owner, the `memnode` key can be used to specify the NUMA node ID on the physical host. If specified, then the memory of the VM is allocated from the given NUMA node. If the `memnode` key is not defined, the VM memory is allocated from NUMA node 0.

```
ericsson:
  ...
  reservedCPUs:
    - owner: vcic
      count: 2
      isolated: false
  ...
  reservedHugepages:
    - owner: vcic
      size: 1GB
      count: 30
      memnode: 1
  ...
```

Example 59 Reserve 30 GiB RAM for a vCIC using physical memory on NUMA node 1

If `isolated: false` is not used for CPU reservation, the memory will be allocated from the NUMA nodes proportionally to the number of CPUs in the NUMA node.

3.1.4 CPU Allocation for Single Server Deployment

The following example shows the recommended CPU allocation for a Single Server deployment:

```
ericsson:
  ...
  reservedCPUs:
    - owner: vm
      cpus: 6,30,8,32,10,34,7,31,9,33,11,35,13,37,15,39,17,41,19,43,21,45
    - owner: ovs
      count: auto
      cpus_nonpmd: 1
    - owner: vcic
      count: 2
      isolated: false
  ...
```

Example 60 Allocating vCPUs Reserved for VMs

This example illustrates some of the concepts discussed before:

- CPUs are assigned manually using the `cpus` key to reserve specific CPUs.



- As a consequence, this allocation is suitable for a specific server and processor model because the NUMA topology and the numbering of CPUs vary among systems. This example assumes a Dell R630 server equipped with 2 Intel Xeon E5-2680 v3 processors.
- No dedicated CPUs are reserved for the vCIC, it uses any of the non-isolated CPUs.

For more information on single server CPU allocation, refer to [Single Server System Dimensioning Guide, CEE 6](#).

3.2 NUMA Balancing

The automatic NUMA balancing kernel feature can be disabled on the compute blades, both globally and on a per blade basis:

```
ericsson:
  global_numa_balancing: false
  shelf:
    -
      id: 0
      ...
      blade:
        ...
        -
          id: 2
          numa_balancing: true
```

Example 61 Example NUMA Balancing Configuration

global_numa_balancing

The global configuration key for NUMA balancing is `global_numa_balancing`, which takes a boolean value. If set to **true**, the automatic NUMA balancing is enabled in the whole region. If set to **false**, NUMA balancing is disabled in the region. The default value is **true**.

This configuration can be overridden on individual blades using the `numa_balancing` blade key.

numa_balancing

The `shelf/blade/numa_balancing` key can be used to override the `global_numa_balancing` or the default configuration on a per blade basis. If set to **true**, the NUMA balancing kernel feature is enabled on the respective compute blade. If set to **false**, NUMA balancing is disabled on the blade.

If neither `global_numa_balancing` nor `numa_balancing` is configured, all compute blades will be installed with the NUMA balancing kernel feature enabled.



3.3 DPDK Physical Interface Driver

The DPDK physical interface driver can be configured on each compute blade as follows:

```
ericsson:
  global_ovs_dpdn_io_driver: vfio-pci
  shelf:
    -
      id: 0
      ...
      blade:
        ...
        -
          id: 2
          ovs_dpdn_io_driver: uio_pci_generic
```

Example 62 Example DPDK Physical Interface Driver Configuration

The supported DPDK physical interface driver on each respective platform is as follows:

- Dell multi-server: `vfio-pci`
- BSP: `vfio-pci`
- HDS: `vfio-pci`
- Single server: `vfio-pci`

Note: The OVS DPDK physical interface must be defined, either globally or for each individual compute blade.

global_ovs_dpdn_io_driver

The `global_ovs_dpdn_io_driver` key is used to set the DPDK physical interface driver in the whole region. The supported values are `vfio-pci` and `uio_pci_generic`. The global configuration can be overridden on a per blade basis using the `ovs_dpdn_io_driver` key.

ovs_dpdn_io_driver

The `shelf/blade/ovs_dpdn_io_driver` key is used to set the DPDK physical interface driver of a specific blade. The supported values are `vfio-pci` and `uio_pci_generic`. This key overrides the parameter set in `global_ovs_dpdn_io_driver` for the blade.

3.4 Kernel Parameter Settings

In order to fine-tune the system performance, the following parameters can be configured:



vm_stat_interval / global_vm_stat_interval

The interval of the memory statistic updates can be configured in the whole region using the `global_vm_stat_interval` key. The value is to be given in seconds.

The `vm_stat_interval` blade configuration key can be used to configure the interval on a per blade basis. This key overrides the global setting.

Both keys are optional. If neither the global nor the blade key is defined, the default value (120 seconds) becomes active for each blade.

```
ericsson:
  global_vm_stat_interval: 110
  shelf:
    -
      id: 0
      blade:
        -
          id: 2
          vm_stat_interval: 115
```

Example 63 Example vm_stat_interval Configuration

kernel_nmi_watchdog / kernel_nmi_watchdog

Non Maskable Interrupts (NMI) can be enabled or disabled in the whole region using the `global_kernel_nmi_watchdog` key. Correct values are 0 (disabled) and 1 (enabled).

The `kernel_nmi_watchdog` blade configuration key can be used to configure the NMI on a per blade basis. This key overrides the global setting.

Both keys are optional. If neither the global nor the blade key is defined, the default value (0) becomes active for each blade.

```
ericsson:
  global_kernel_nmi_watchdog: 1
  shelf:
    -
      id: 0
      blade:
        -
          id: 2
          kernel_nmi_watchdog: 0
```

Example 64 Example kernel_nmi_watchdog Configuration



3.5 Increasing Virtio Queue Size

Note: This feature requires the `ericsson_performance` Fuel plugin to be configured. For more information, refer to the [Fuel Plugin Configuration Guide](#).

An updated QEMU package with `virtio_queue_size` increased to 1024 descriptors can be optionally enabled on a per blade basis using the `increased_virtio_queue` key:

```
ericsson:
  shelf:
    -
      id: 0
      blade:
        -
          id: 2
          increased_virtio_queue: True
```

Example 65 Example increased_virtio_queue Configuration

This configuration is only applicable to blades with the role `compute`.

If the `increased_virtio_queue` key is set to **True**, the QEMU package will be updated on the blade. If undefined or set to **False**, the standard QEMU package will be installed.

3.6 NIC Information

The section `nic_assignments` defines role-based PCI addresses for all NICs in the system, for the control, traffic and storage networks. If the relevant hardware is not listed in `nic_assignments` in the relevant template, add or modify the relevant element in the list.

The PCI addresses must be defined in the following format: `XXXX:YY:ZZ.W` where:
XXXX = domain, must always be 0000
YY = bus
ZZ = slot
W = function

Anchors are used to label each assignment in `nic_assignments`. See Section 1.5.1 on page 5 for the use of anchors and aliases.

Note: Every network role (`control`, `data`, `storage`) must be configured for a NIC assignment. Failure to do so results in deployment failure.



```
ericsson:
  nic_assignments:
    - &DELL_630_nic_assignment
      control0: "0000:01:00.0"
      control1: "0000:01:00.1"
      data0:    "0000:81:00.0"
      data1:    "0000:04:00.0"
      storage0: "0000:81:00.1"
      storage1: "0000:04:00.1"
    - &DELL_620_nic_assignment
      control0: "0000:01:00.0"
      control1: "0000:01:00.1"
      data0:    "0000:42:00.0"
      data1:    "0000:04:00.0"
      storage0: "0000:42:00.1"
      storage1: "0000:04:00.1"
```

Example 66 Example NIC Assignment

3.7 SR-IOV

Note: The SR-IOV feature is configurable on Dell multi-server hardware platform, and on HDS platform with additional NICs available for SR-IOV.

To enable SR-IOV traffic network support, the feature has to be configured in `config.yaml` in the following places:

- On a per blade basis, using the `sriov` key, see Section 3.7.1 on page 67.
- In the global configuration, using the `sriov_segmentation_type` key, see Section 3.7.2 on page 69
- In case of CEE-managed switch configuration (Extreme), in the switch cabling scheme configuration template, see Section 3.7.3 on page 70

3.7.1 SR-IOV Blade Configuration

Note: SR-IOV can only be enabled on compute nodes.

The `sriov` key is to be configured individually for each blade that will have the SR-IOV feature enabled. The key is not to be configured for blades not using SR-IOV. To define SR-IOV on a specific blade, include the `sriov` key with the number of virtual functions and devices properties in the blade section. You can reference the devices with aliases that point to pre-defined anchors in `sriov_configs` at the beginning of the configuration file.

Additionally, the configuration of an arbitrary provider physical network for each SR-IOV Physical Function (PF) is supported. This makes the PF resource management significantly more dynamic.



More anchors can be added in `sriov_configs` if the used hardware configuration is different from the provided ones. See also Section 1.5.1 on page 5 about YAML syntax.

See the configuration example below:

```
ericsson:
  ...
  sriov_configs:
    - &DELL_620_sriov_info
      pci_address: "0000:41:00.0"
      bandwidth: 10000000
      physical_network: "physnet1"
    - pci_address: "0000:41:00.1"
      bandwidth: 10000000
      physical_network: "physnet1"
    - &DELL_630_OEM_sriov_info
      pci_address: "0000:83:00.0"
      bandwidth: 10000000
      physical_network: "physnet1"
    - pci_address: "0000:83:00.1"
      bandwidth: 10000000
      physical_network: "physnet1"
    - &DELL_630_sriov_info
      pci_address: "0000:84:00.0"
      bandwidth: 10000000
      physical_network: "physnet1"
    - pci_address: "0000:84:00.1"
      bandwidth: 10000000
      physical_network: "physnet1"
  ...
  shelf
  -
    blade:
      -
        id: 2
        sriov:
          devices: *DELL_630_sriov_info
          vf: 8
  ...
```

Example 67 8 VFs with 2 SR-IOV Devices in Blade 2, All PFs Connected to "physnet1"

ericsson/sriov_configs

The dictionaries containing the PCI addresses of SR-IOV NICs (PFs) have to be defined in `sriov_configs`. The N-th SR-IOV PF on a blade is the N-th listed item in the dictionary for the blade. Each listed item must contain the `pci_address` and `physical_network` parameters.



- `pci_address` identifies the network interfaces that must be used for SR-IOV purpose on the compute node. Enter in format "DDDD:BB:SS.F" key-value pairs.
D=domain (always 0000), B=Bus, S=Slot, F=Function. The double quoting is mandatory for the value of `pci_address`. This is a mandatory parameter if `sriov` is defined.
- `bandwidth` is the bandwidth of the traffic Ethernet interfaces as specified by the hardware NIC vendor, in kilobit/s. This parameter is not mandatory.
- `physical_network` is the name of the physical network assigned to the respective device. This is a mandatory parameter if `sriov` is defined. The name of the physical network can contain alphanumeric characters, underscores ("_"), and hyphens ("-"). The same `physical_network` name can be used for different SR-IOV PFs.

ericsson/shelf/blade/sriov

- `devices` is a list of dictionaries containing the PCI addresses of SR-IOV NICs (PFs). The list of devices cannot contain more than two devices.
- `vf` specifies the number of Virtual Functions (VFs) on compute nodes assigned to each PF (NIC). The maximum value is 16. This is a mandatory parameter if `sriov` is defined.

Note: It is highly recommended to define the same number of VFs for each blade on which SR-IOV is configured. Asymmetric `vf` configuration can result in SR-IOV VMs not being migrated or evacuated to other compute hosts.

3.7.2 SR-IOV Global Configuration

The global `sriov_segmentation_type` key has to be set if the SR-IOV blade configuration is enabled on one or more blades, see Section 3.7.1 on page 67. This key is used to enable type(s) of network segmentation in a CEE region. The following network segmentation options are available:

- **vlan**: enables VLAN segmentation in the CEE region, see Example 68.
- **flat**: enables flat networking in the CEE region, see Example 69.
- both **vlan** and **flat**: enables both VLAN segmentation and flat networking in the CEE region. see Example 70.



```
ericsson:
  ...
  sriov_segmentation_type:
    - vlan
  neutron:
    ...
    neutron_config_yaml_file: neutron_ericsson_extreme.yaml
  ...
  hw_switches:
    initial_setup: extreme
    switching_scheme_yaml_file: 2_x670v_switch.yaml
    cabling_scheme_yaml_file: 2_x670v_dell.yaml
```

Example 68 SR-IOV with VLAN Segmentation Enabled in the Region

```
ericsson:
  ...
  sriov_segmentation_type:
    - flat
  neutron:
    ...
    neutron_config_yaml_file: neutron_ericsson_user_spec.yaml
  ...
  hw_switches:
    initial_setup: None
```

Example 69 SR-IOV with Flat Networking Enabled in the Region

```
ericsson:
  ...
  sriov_segmentation_type:
    - flat
    - vlan
  neutron:
    ...
    neutron_config_yaml_file: neutron_ericsson_user_spec.yaml
  ...
  hw_switches:
    initial_setup: None
```

Example 70 SR-IOV with Both Flat Networking and VLAN Segmentation Enabled in the Region

3.7.3 SR-IOV Cabling Scheme Configuration

In case of CEE-managed switch configuration, the cabling scheme YAML file must be updated according to the allocation of SR-IOV ports in the traffic switch. The SR-IOV ports have the value **usage: sriov**.



```

cabling_scheme:
  shelves:
    - blades:
        - blade_id: 1
          network interfaces:
            - {nic_id: 1, switch_id: 1, switch_port: 1, usage: data}
            - {nic_id: 2, switch_id: 2, switch_port: 1, usage: data}
            - {nic_id: 3, switch_id: 1, switch_port: 2, usage: storage}
            - {nic_id: 4, switch_id: 2, switch_port: 2, usage: storage}
            - {nic_id: 5, switch_id: 1, switch_port: 40, usage: sriov}
            - {nic_id: 6, switch_id: 2, switch_port: 40, usage: sriov}

```

Example 71 SR-IOV Cabling Scheme Configuration

Extreme traffic switch ports are **not** configured when a VM is booted in a Neutron network used for SR-IOV. The physical ports used for SR-IOV tenant traffic must be manually configured with the respective Neutron VLAN. Refer to the *SW Installation in Multi-Server Deployment* document for more information.

3.8 PCI Passthrough

The PCI passthrough feature is configurable on Dell multi-server hardware platform, BSP, and HDS platform, if free NICs are available for PCI passthrough.

PCI passthrough can only be configured using compatible NICs on which the SR-IOV virtualization mode can be disabled. As a result, PCI passthrough is not possible on HDS and Dell servers using Intel Fortville NICs (XXV710-AM1, XXV710-AM2, XL710-AM1, XL710-AM2, XL710-BM1, XL710-BM2, X710-AM2, X710-BM2, XL710-QDA1, XL710-QDA2, X710-DA2, and X710-DA4). If the SR-IOV virtualization mode cannot be disabled, the OpenStack functionality SR-IOV Physical Function passthrough (PCI passthrough based on type-PF) can be configured and used. In this case, the device type of the interfaces used by Nova is not **type-PCI**, but **type-PF**.

To enable PCI passthrough traffic network support, the feature has to be configured in `config.yaml` in the following places:

- The `ericsson_pci_passthrough` Fuel plugin must be enabled, refer to the *Fuel Plugin Configuration Guide*.
- On a per blade basis, using the `pci_passthrough` key, see Section 3.8.2 on page 72.
- The PCI aliases optionally on a global level, see Section 3.8.1 on page 72.
- If using CEE-managed switch configuration (Extreme), in the switch cabling scheme configuration template, see Section 3.8.3 on page 73.

SR-IOV and PCI passthrough cannot be configured on the same compute host.



Note: The PCI passthrough feature can only be configured when performing the initial deployment of CEE. It cannot be configured during the CEE software update procedure.

Note: To provide High Availability for the VMs using PCI passthrough interfaces, one or more compute servers must be left with free PCI passthrough interfaces for VM evacuation purpose in case of a compute server failure. If each PCI passthrough interface is used in the CEE system, Nova scheduler will fail in case of a compute server failure because of lack of needed resources on other compute servers.

3.8.1 PCI Passthrough Alias Configuration

Arbitrary alias names can be configured for the network devices used for PCI passthrough or SR-IOV Physical Function passthrough functionality. This allows users to specify the alias with the PCI property requirements in the `extra_specs` of the Nova flavor.

The aliases can be defined under the `pci_passthrough_alias` key, using the following parameters in a combination:

- `device_type` is the type of the used PCI device. The value must be **type-PCI** or **type-PF**.
- `product_id` is the product number of the used network device in hexadecimal. This parameter is optional.
- `vendor_id` is the vendor number of the manufacturer of the network device in hexadecimal. This parameter is optional.

`pci_passthrough_alias` is an optional parameter. If not configured, the default PCI alias devices `PCI_devs` and `PF_devs` are used in the whole CEE region:

- `PCI_devs` devices can be used by VMs for PCI passthrough (SR-IOV virtualization mode is set to off on the NIC), in which the `device_type` is **type-PCI**
- `PF_devs` devices can be used by VMs for SR-IOV Physical Function passthrough (SR-IOV virtualization mode is set to on on the NIC), in which the `device_type` is **type-PF**

```
ericsson:
  ..
  pci_passthrough_alias:
    PCI_Intel_X520: {'device_type': 'type-PCI', 'vendor_id': '8086', 'product_id': '154d'}
    PCI_Intel_device: {'device_type': 'type-PCI', 'vendor_id': '8086'}
    PCI: {'device_type': 'type-PCI'}
```

Example 72 Example PCI Passthrough Alias Configuration

3.8.2 PCI Passthrough Blade Configuration

Note: PCI passthrough can only be enabled on compute hosts.



To define PCI passthrough or SR-IOV Physical Function passthrough on a specific blade, configure the `pci_passthrough` key with devices properties in the blade section. The `pci_passthrough` key must be configured for each blade where the PCI passthrough or SR-IOV Physical Function passthrough feature is to be enabled.

It is advised to refer to the devices with aliases that point to pre-defined anchors at the beginning of the configuration file. For information on using aliases, see Section 1.5.1 on page 5.

```
ericsson:
  ...
  pci_passthrough_configs:
    - &DELL_620_pci_pt_info
      - pci_address: "0000:41:00.0"
      - pci_address: "0000:41:00.1"
    ...
  shelf:
    -
      blade:
        -
          id: 6
          pci_passthrough:
            devices: *DELL_620_pci_pt_info
```

Example 73 PCI Passthrough Blade Configuration with Alias

```
pci_passthrough:
  devices:
    - pci_address: "0000:42:00.0"
    - pci_address: "0000:42:00.1"
```

Example 74 PCI Passthrough Blade Configuration without Alias

`devices` is a list of dictionaries containing the unique PCI address of each PCI passthrough NIC (Physical Function), where the key of each PCI address is `pci_address`, and the value is the standard PCI address in double quotes. A minimum of one PCI address must be configured.

3.8.3 PCI Passthrough Cabling Scheme Configuration

In case of CEE-managed switch configuration on Dell platforms, the cabling scheme YAML file must be updated according to the allocation of PCI passthrough ports in the traffic switch. The PCI passthrough ports have the value usage: `pci_passthrough`. This value is valid for both PCI passthrough or SR-IOV Physical Function passthrough use cases.



```

cabling_scheme:
  shelves:
    - blades:
        - blade_id: 6
          network_interfaces:
            - {nic_id: 1, switch_id: 1, switch_port: 1, usage: data}
            - {nic_id: 2, switch_id: 2, switch_port: 1, usage: data}
            - {nic_id: 3, switch_id: 1, switch_port: 2, usage: storage}
            - {nic_id: 4, switch_id: 2, switch_port: 2, usage: storage}
            - {nic_id: 5, switch_id: 1, switch_port: 40, usage: pci_passthrough}
            - {nic_id: 6, switch_id: 2, switch_port: 40, usage: pci_passthrough}

```

Example 75 Example PCI Passthrough Cabling Scheme Configuration

The physical ports of the traffic switch used for PCI passthrough tenant traffic must be manually configured with the respective VLAN. For more information, refer to:

- Multi-Server System Dimensioning Guide, CEE 6
- In case of BSP deployment: Manage VLAN, Reference [10]
- In case of HDS deployment: Connect an L2 Network to an Ethernet Interface with VLAN Tagging Using REST API in the HDS documentation, Reference [11]

3.9 Bandwidth-Based Scheduling

3.9.1 Nominal Bandwidth of Neutron Physical Networks

The `neutron_networks` section describes the characteristics of the Neutron physical networks used to identify the physical ports dedicated to CSS physical interfaces. It defines the bond interfaces, Neutron name, and bandwidth capacity.

Format:

```

neutron_networks:
  <neutron-physical-network-name>:
    devices:
      - <device>
      ...
    bandwidth: <capacity-of-network>

```

`devices` lists the interfaces used to bond the network. `bandwidth` defines the capacity of the network in kbit per second.

Note: In the current release, only the default Neutron physical network is supported, and the `devices` key is not used.



```
ericsson:
...
neutron_networks:
- &neutron_networks_std_limit
  control:
    devices:
    - control0
    - control1
    bandwidth: 1000000
  default:
    devices:
    - data0
    - data1
    bandwidth: 10000000
...
shelf:
-
...
blade:
...
-
  id: 3
  nic_assignment: *DELL_630_nic_assignment
  reservedHugepages:
  ...
  reservedCPUs:
  ...
  vswitch_capacity: <vswitch capacity>
  neutron_networks: *neutron_networks_std_limit
```

Example 76 neutron_networks Configuration

If SR-IOV is used, the bandwidth of the Neutron physical network is specified in the `sriov_configs` section. See Section 3.7.1 on page 67.

3.9.2

vSwitch Capacity

The `vswitch_capacity` attribute defines the capacity of the virtual switch on each host. The capacity is shown in kilo packet per second. It is used for bandwidth-based scheduling.



```
ericsson:
  ...
  shelf:
    -
      ...
      blade:
        ...
        -
          id: 3
          nic_assignment: *DELL_630_nic_assignment
          reservedHugepages:
            ...
          reservedCPUs:
            ...
          vswitch_capacity: <vswitch capacity>
```

Example 77 vswitch_capacity

Note: For more information about the recommended way to determine the values for vswitch_capacity, refer to the [Multi-Server System Dimensioning Guide, CEE 6](#).

3.10 Neutron Configuration Options

The Neutron configuration file template (selected by using information provided in Section 2.2 on page 7) can be modified compared to the Ericsson default parameters and values. The change must be part of a system integration activity that includes CEE verification.

Note: CEE was verified only with unchanged Neutron configuration files.



```
# Without further configuration ericsson_user_spec is equivalent to
# ericsson_basic, but ericsson_user_spec is not locked down.
conf_type: ericsson_user_spec
# The .deb files must be included into the Fuel .iso.
# Will be installed on the given target groups.
# Multiple groups are allowed in the target list with comma separation.
# The followin groups are usually enough to install the additional package:
# all - this means the package will be installed on all compute and
# controller nodes
# compute - this means that the package will be installed on all compute nodes
# controller - this means that the package will be instaleed on all controllers
additional_packages:
  - name: "<DEBIAN.PACKAGE>"
    target: ["<TARGET.NODE.GROUP>"]
  - name: "<DEBIAN.PACKAGE>"
    target: ["<TARGET.NODE.GROUP>"]
neutron_configuration_files:
  -
    name: neutron.conf
    option:
      # default if next line is not present: no service plugins
      DEFAULT/service_plugins: <COMMA.SEPARATED.LIST.OF.SERVICE.PLUGINS>
  -
    # It is possible to list multiple .ini files here and they will get
    # merged into a single plugin.ini.
    name: ml2_conf.ini
    option:
      # default if next line is not present: openvswitch
      ml2/mechanism_drivers: <COMMA.SEPARATED.LIST.OF.MECHANISM.DRIVERS>
```

Example 78 neutron_ericsson_user_spec.yaml Template File

```
conf_type: ericsson_user_spec
additional_packages:
  - name: "neutron-plugin-bsp"
    target: ["controller"]
neutron_configuration_files:
  -
    name: ml2_conf.ini
    option:
      ml2/mechanism_drivers: openvswitch,bsp
  -
    name: ml2_conf_bsp.ini
    option:
      ml2_bsp/management_ip: 192.168.2.2
      ml2_bsp/audit_interval: 10
      ml2_bsp/timeout: 10
      ml2_bsp/port_list_restorable: True
```

Example 79 neutron_ericsson_cmx.yaml Template File



Note: Do not modify the initial indentation when editing the configuration files.

- `additional_packages`: list Debian packages included in the .iso Fuel build to be deployed to the controllers during CEE installation. For example, drivers that are not included in the OpenStack distribution, and add functionality inside Neutron. Put each package name in quotation marks. Specify at least one package, otherwise the installation fails.
- `neutron_configuration_files`: list Neutron configuration files to change or create, refer to OpenStack for valid names.
- `name`: the name of the section in the corresponding Neutron configuration file. The parameter value is a string that is deployed exactly as it is written.
- `option`: lists the parameters that the user wants to write in the corresponding Neutron configuration file. Use the following format: `<section name>/<key name>: <value>`

3.11 Nova Configuration Options

The default values in the nova section is used for a regular deployment scenario. The parameters below are optional. If these parameters are not included in `config.yaml`, the default values apply. Do not modify the `config.yaml` file unless the listed use cases are required.

- `disk_cachemodes`: The cache modes to use for different disk types.
- `enable_nova_quotas`: The default value is **True** to enable quota support, **False** means no Nova quota support.
- `force_config_drive`: The default value is **True**, to force injection to take place on a config drive. **False** means not forcing the injection. The user can still specify the config drive use on boot.
- `vms_use_raw_images`: The default value is **False**, that is, cow images are used by the VMs. If it is set to **True**, then raw images are used instead.

ericsson:

```
...
nova:
  disk_cachemodes: file=directsync,block=none
  enable_nova_quotas: true
  force_config_drive: true
  vms_use_raw_images: false
...
```

Example 80 Nova Configuration Options



3.12 Hardware Switch Configuration Options

The `hw_switches` section in the `config.yaml` template provides the `initial_setup` parameter. According to the settings in this section, the CEE installation deploys the initial configuration relevant to the used switch type, that is, Extreme traffic and storage switches in Dell multi-server deployments, and CMXB in BSP hardware. No initial hardware switch configuration is used for Single Server CEE and for user-specific Neutron options.

The setting of `initial_setup` in the `hw_switches` section must be aligned with the setting of the `neutron_config_yaml_file` in the `neutron` section. Table 3 shows the settings that can be used together.

Table 3 Neutron Configuration File and Initial Setup Values

Hardware Deployment	neutron: neutron_config_yaml_file:	hw_switches: initial_setup:
BSP	<code>neutron_ericsson_cmx.yaml</code>	<code>cmx</code>
Dell multi-server ⁽¹⁾	<code>neutron_ericsson_extreme.yaml</code>	<code>extreme</code>
Single Server HDS without SDN Other, user specific	<code>neutron_ericsson_user_spec.yaml</code>	<code>None</code>
HDS with SDN	<code>neutron_ericsson_sdn_tight.yaml</code>	<code>None</code>

(1) In case of Dell multi-server configuration with SR-IOV, see Section 3.7.2 on page 69 for the correct Neutron configuration.

See the following sections for more information:

- Section 2.3 on page 9 for hardware switch configuration
- Section 2.2 on page 7 for Neutron configuration files
- Section 3.10 on page 76 for Neutron options

```
ericsson:
...
neutron:
...
  neutron_config_yaml_file: neutron_ericsson_cmx.yaml
...
hw_switches:
  initial_setup: cmx
  switching_scheme_yaml_file: cmx_switch.yaml
...
```

Example 81 Neutron Configuration File and Hardware Switch Settings for CMX, BSP



```
ericsson:
  ...
  neutron:
    ...
    neutron_config_yaml_file: neutron_ericsson_user_spec.yaml
    ...
    hw_switches:
      initial_setup: None
    ...
```

Example 82 No Hardware Switch Configuration, Single Server

3.13 Multiple Data Center Gateways

Note: Border Gateway (BGW) is a deprecated term for Data Center Gateway (DC-GW). The term BGW is used in this section.

Two BGWs are specified in the cabling schema, `4_x670v_dell.yaml` or `2_x670v_dell.yaml`, used for configuring managed Extreme switches:

```
external_components:
  border_gateways:
    - id: 1
      name: BGW-1
      switch_id: 1
      ports: [49, 50, 51, 52]
      master: 49
      partition: 4x10G
    - id: 2
      name: BGW-2
      switch_id: 2
      ports: [49, 50, 51, 52]
      master: 49
      partition: 4x10G      ...
```

Example 83 Two BGWs in Cabling Schema

The BGWs are also specified in the switch configuration file:



```
switching:
-
  name: TRAFFIC_SWA_X770
  device_id: 1
  ...
  bgw_config:
  -
    id: 1
    vlans:
    -
      name: cee_om_sp
      tagged: true
      ip: <IP.OF.SWITCH-A/PREFIXSIZE>
    -
      name: subrack_om_sp
      tagged: true
      ip: IP.OF.THE.SWITCH/PREFIXSIZE
  ...
-
  name: TRAFFIC_SWB_X770
  device_id: 2
  ...
  bgw_config:
  -
    id: 2
    vlans:
    -
      name: cee_om_sp
      tagged: true
      ip: <IP.OF.SWITCH-B/PREFIXSIZE>
    -
      name: subrack_om_sp
      tagged: true
      ip: IP.OF.THE.SWITCH/PREFIXSIZE
  ...
```

Example 84 BGWs in Switch Configuration File

Adding further BGWs requires the following:

- Entities must be added to the cabling schema and to the switch configuration file, with increasing IDs.
- The ports used by the additional gateways must be specified.



```

external_components:
  border_gateways:
    - id: x
      name: BGW-x
      switch_id: <1 or 2>
      ports: [y, y+1, y+2, y+3, ... , y+n]
      master: y
      partition: 4x10G

```

Example 85 Additional BGW in Cabling Schema

Note: A partition of 1x40G is also possible, if 40G connections are used towards the BGW.

Switch ID is 1 or 2, depending on the switch to which the BGW is connected.

The switch configuration file can be updated as shown in the following example:

```

switching:
-
  name: TRAFFIC_SWA_X770
  device_id: <1 or 2>
  ...
  bgw_config:
  -
    id: z
    vlans:
    -
      name: cee_om_sp
      tagged: true
      ip: <IP.OF.SWITCH-A/PREFIXSIZE> =>
--- (same as the other BGW, in the same switch)
    -
      name: subrack_om_sp
      tagged: true
      ip: <IP.OF.THE.SWITCH/PREFIXSIZE> =>
--- (same as the other BGW, in the same switch)

```

Example 86 Switch Configuration File Update for Multiple BGWs

3.14 Change of Data Center Gateway Settings

To configure the DC-GW with settings different from the default in CEE, the startup configuration of managed Extreme switches must be handled differently for traffic and storage.

Global process:

1. If the system is already deployed, change the configuration version in the following file: /mnt/cee_config/<switch_model>_switch.yaml



- ```
switch_config:
 restore_conf_version: 15B_R4
15B_R4 must be replaced with higher number, for example 15B_R5
```
2. Change the default configuration of the switches in the following file:  
/opt/ecs-fuel-utils/python\_libdir/extreme\_conf/sw\_conf\_XXX.xsf
  3. If the hardware configuration contains dedicated storage switches, make sure that the storage-specific default configuration file is named as follows: sw\_conf\_XXX\_storage.xsf. Tip: Make a copy of the original traffic-specific file and add \_storage to the name of the new file.
  4. Modify the default traffic and storage switch configuration in the files:  
/opt/ecs-fuel-utils/python\_libdir/extreme\_conf/sw\_conf\_XXX.xsf  
and  
/opt/ecs-fuel-utils/python\_libdir/extreme\_conf/sw\_conf\_XXX\_storage.xsf

Finally, continue with the normal installation process.

## 3.15 Time Zone

The time zone to be used on Fuel and the deployed nodes in the CEE region is stated in the config.yaml template. The time zone configured in the templates is UTC (Etc/UTC).

```
ericsson:
 ...
 timezone: Etc/UTC
 ...
```

### Example 87 Time Zone

For a list of available time zone settings, execute the following command on a Linux system, for example on the Kickstart Server:

```
ls -R /usr/share/zoneinfo
```

## 3.16 Secure NBI API Endpoints

API NBI endpoints are exposed over SSL/TLS on HTTP. To set the needed trust on the client side, a set of CA certificates must exist.

The options are in the Ericsson namespace, so each option is prefixed with “ericsson”.



```

ericsson:
 security:
 openssl:
 ciphersuites: 'ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256⇒
-GCM-SHA384:ECDHE-EDSA-AES128-GCM-SHA256:
 ECDHE-RSA-AES128-GCM-SHA256'
 protocols: TLSv1.2
 gnutls:
 priority: 'NONE:+SUITEB128:+SUITEB192:+VERS-TLS-1.2'
 haproxy:
 sslprotocols: no-sslv3 no-tls10 no-tls11
 sslrate: 100
 sslconns: 40
 nbi:
 atlas:
 hostname: <ATLAS_HOSTNAME>
 certfilename: <ATLAS_CERTFILENAME>
 cafilename: <ATLAS_CAFilename>
 controller:
 hostname: <CONTROLLER_HOSTNAME>
 certfilename: <CONTROLLER_CERTFILENAME>
 cafilename: <CONTROLLER_CAFilename>
 ...

```

#### Example 88 Secure NBI API Options

##### **security.openssl.ciphersuites**

The string containing the list of allowed OpenSSL cipher suites. Validate the support of cipher suites with the external hosts that use the REST API and change the specified values if needed.

Examples on external hosts using the REST API:

- Network Function Virtualization Orchestrator (NFVO), for example, Ericsson Cloud Manager (ECM)
- Virtualized Network Function Manager (VNFM)

**Note:** The order of the listed cipher suites signifies the precedence honored by CEE components.

Recommended setting:

ECDHE-RSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-RSA-AES128-GCM-SHA256

##### **security.openssl.protocols**

The string containing the list of allowed SSL/TLS protocols.

Recommended setting: TLSv1.2

**security.gnutls.priority**

The string containing gnuTLS protocol/ciphersuite settings.

Recommended setting:

NONE : +SUITEB128 : +SUITEB192 : +VERS-TLS-1.2

**security.haproxy.sslprotocols**

The string containing accepted/disabled SSL/TLS protocols to offer.

Recommended setting:

no-ssl3 no-tls10 no-tls11

**security.haproxy.sslrate**

The number of SSL sessions allowed to be established per second.

Recommended setting: 100

Recommended setting for Dell Single server: 20

**security.haproxy.sslconns**

The number of SSL connections to allow per listener.

Recommended setting: 40

**Note:** One session requires two connections.

**security.nbi.controller.hostname**

The string containing the hostname by which the CIC is referenced through the NBI.

**security.nbi.controller.certfilename**

The string containing the filename (relative to /mnt/cee\_config) of the CIC certificate.

**security.nbi.controller.cafilename**

The string containing the filename (relative to /mnt/cee\_config) of the CA certificate signing the CIC certificate.

**security.nbi.atlas.hostname**

The string containing the hostname of the Atlas VM that is reachable from the vCICs. Only needed if Atlas is installed.

**security.nbi.atlas.certfilename**

The string containing the filename (relative to `/mnt/cee_config`) of the Atlas certificate when Atlas is installed.

**security.nbi.atlas.cafilename**

The string containing the filename (relative to `/mnt/cee_config`) of the CA certificate signing the Atlas certificate if Atlas is installed.

The hostnames for the vCICs and Atlas are user-supplied. End users must make sure that their hostnames are unique and the hostnames used for cloud endpoints are resolving to the proper IP addresses. The hostnames must conform to RFC 952, Reference [7] and RFC 1123, Reference [8].

**Note:** The certification files are either acquired from a third party or generated by own CA authority, and out of scope of this document. Refer to the [SW Installation in Single Server Deployment](#) and [SW Installation in Multi-server Deployment](#) for more information on the certificate for the Northbound Interface (NBI) required for secure HTTPS access to CEE.

## 3.17 Fuel Administration Network

`fuel_ctrl_sp` is used for PXE boot of compute hosts (host OS) and vCIC nodes.

The network can be configured under the `networks` key in the `config.yaml` template:

```
ericsson:
 ...
 networks:
 ...
 -
 name: fuel_ctrl_sp
 mos_name: fuelweb_admin
 cidr: 192.168.0.11/24
 dhcp_pool_start: 192.168.0.20
 dhcp_pool_end: 192.168.0.253
 gateway: 192.168.0.254
 dns: 10.51.40.100
 ...
```

### Example 89 Fuel Administration Network

If the IP address of the Fuel administration network is different from the network specified in `config.yaml` and the IP and VLAN plan Reference [1], update `fuel_ctrl_sp` before running `CEE_RELEASE/scripts/install_vfuel.sh`.

**Note:** Make sure that the Fuel IP is **not** included in the dynamic IP range of `dhcp_pool_start` and `dhcp_pool_end`.



## 3.18 Location of Logs

The location of core and crash dump logs can be changed using the following parameters:

```
ericsson:
 ...
 logging:
 crashes: local
 forward_to_fuel: false
 forward_to_controller: true
 forward_to_external: false
 external_server_ip:
 external_server_port:
 local_on_controller: true
 local_on_compute: false
 ...
```

### Example 90 Crashes Stored Locally

```
ericsson:
 ...
 logging:
 forward_to_fuel: false
 forward_to_controller: true
 forward_to_external: true
 external_server_ip: 1.2.3.4
 external_server_port: 5678
 local_on_controller: true
 local_on_compute: false
 ...
```

### Example 91 Setting Server IP and Port for External Server

**Note:** The boolean parameters of logging must be included in the `config.yaml`. The default values mentioned below refer to the values originally set in the template.

#### crashes

The destination of crashes (core and kernel crash dumps). `crashes` can have the value `local` or `cics`. Dell and single server store crashes locally in each blade or server by default. BSP stores most crashes in the vCICs, BSP still saves crashes locally where possible, since disk space is scarce in most BSP installations. If `cics` is selected for single server, local crashes are still used when needed, for example at compute host kernel crash or core dump of QEMU.

#### forward\_to\_fuel

Indicates whether to forward logs from both vCICs and compute hosts to Fuel or not. Boolean parameter, default value: **false**

**forward\_to\_controller**

Indicates whether to forward logs from compute hosts to vCIC or not. Boolean parameter, default value: **false** on Dell and Single Server, **true** on BSP

**forward\_to\_external**

Indicates whether to forward logs from both vCIC and compute host to an external log server or not. Boolean parameter, default value: **false**

**external\_server\_ip**

The IP address of an external log server. Mandatory if **forward\_to\_external** is set to **true**

**external\_server\_port**

The port of an external log server. Mandatory if **forward\_to\_external** is set to **true**

**local\_on\_controller**

Enable or disable local logging on vCIC. Boolean parameter, default value: **true**

**local\_on\_compute**

Enable or disable local logging on compute hosts. Boolean parameter, default value: **true** on Dell and Single Server, **false** on BSP.

**auditlog\_local**

If enabled, up to three audit logs will be stored on the vCICs in `/var/log/audittrail`. The size of an audit log is 50 MiB.

This Boolean parameter is optional. If omitted, the default value **false** is configured.

**auditlog\_remote**

Enable or disable audit log forwarding to the SBI, if the SBI address is configured.

This Boolean parameter is optional. If omitted, the default value **true** is configured.

## 3.19

### Link Monitoring for CEE on BSP

**Note:** Port state tracking is only applicable to BSP R9.1 and later releases.



Link monitoring for CEE on BSP is implemented using port state tracking. To enable port state tracking, configure the following settings:

```
ericsson:
 ...
 port_state_tracking:
 enabled: <TRUE/FALSE>
```

#### Example 92 Port State Tracking Configuration

Port state tracking is implemented for the control and traffic networks with the following mandatory parameter:

- **enabled:** If this parameter is set to **True**, the system automatically configures port state tracking in CEE. The parameter is set to **True** in the `config.yaml` template.

```
ericsson:
 ...
 port_state_tracking:
 enabled: True
```

#### Example 93 Port State Tracking Example Setup

If **enabled** is set to **True**, the bond settings must be configured in `host_nw_bsp.yaml`. The bond setting is pre-configured in the template. The following parameters are needed:

```
- action: add-bond
 bond_properties:
 mode: active-backup
 use_carrier: 0
 downdelay: 0
 miimon: 100
 bridge: br-fw-admin
 interfaces:
 - <% if1 %>
 - <% if2 %>
 name: bond-fw-admin
 provider: ovs
```

#### Example 94 Host Network Template Configuration for Port State Tracking

## 3.20 Reduced Footprint Monitoring Data Collection

**Note:** The `reduced_footprint` key is deprecated and will be removed in the next major release. For more information, refer to the [Preconfigured Key Performance Indicators](#).



`reduced_footprint` enables reduced KPI data collection in Zabbix to save storage, computing, and network resources. It uses a set of alternative KPI/metric lists for Zabbix that gather and store less measurement data.

`monitoring_data_collection` default value is **false**.

Both keys are optional, if omitted, the default value applies.

```
ericsson:
...
 reduced_footprint:
 monitoring_data_collection: false
...
```

Example 95 Reduced Footprint

## 3.21 Zabbix CEE User

The `zabbix_cee_user` section contains configuration options to configure the user group, username, and password of the read-only user in Zabbix. All keys are optional. If the keys are not present, the following default values are used for the user group, user, and password, respectively:

- **CEEUserGroup**
- **ceeuser**
- An automatically generated secure password

```
ericsson:
...
 zabbix_cee_user:
 zabbix_user_group: <ZABBIX_USER_GROUP>
 zabbix_user: <ZABBIX_USER>
 zabbix_password: <ZABBIX_PASSWORD>
...
```

Example 96 Zabbix CEE User

`zabbix_user_group`: String, the name of the user group. The value `admin` cannot be used, because the global access administrator user “admin” exists by default and cannot be overwritten.

`zabbix_user`: String, the name of the user.

`zabbix_password`: String, the password of the user.

Use single quote marks as shown in the example below:





```
ericsson:
 ...
 zabbix_cee_user:
 zabbix_user_group: 'CEEUserGroup'
 zabbix_user: 'ceeuser'
 zabbix_password: 'examplepassword'
 ...
```

Example 97 Zabbix CEE User Example

## 3.22 Excluding Disk At Deployment

Any disk attached to a compute blade is mounted and erased as part of the CEE deployment.

One or more disks can be configured to be excluded from the CEE deployment by defining the `exclude_disks` key for the blade to which the disk is attached.

```
ericsson:
 ...
 shelf:
 -
 id: 0
 blade:
 -
 id: 1
 exclude_disks: disk/by-id/<DISK_ID>
 ...
 -
 id: 2
 exclude_disks:
 - disk/by-id/<DISK_ID>
 - disk/by-id/<DISK_ID>
```

Example 98 Excluding Disks from CEE Deployment

Multiple disks to be excluded must be submitted in list format.

The disks are specified using the disk names without the leading `/dev`.

**Note:** It is recommended to use `disk/by-path` or `disk/by-id` names. Do not use standard device names such as `sdb`, as their mapping to the physical disk is not guaranteed to be consistent.

## 3.23 Deployment Debugging Information

**Note:** The debugging logs can contain sensitive data, including password hashes, and user-password pairings. As such, the use of `debug_deploy` is not recommended in production environments.



Debugging information can be logged during CEE installation, by defining the optional `debug_deploy` key. The logs can be used to troubleshoot deployment issues.

```
ericsson:
 ...
 debug_deploy: true
```

#### Example 99 Enabling Deployment Debugging

If the parameter is **true**, logs about the deployment will be stored in the `/var/log/puppet.log` file on each node apart from vFuel (compute host, vCIC, ScaleIO). The default value for `debug_deploy` is **false**.



## Reference List

- [1] IP and VLAN plan, 2/102 62-CRA 119 1862/5 Uen
- [2] VNX5400 SW Installation, 3/1531-CSA 113 125/5 Uen
- [3] BSP External Network Connectivity, 2/1553-APP 111 01 Uen
- [4] YAML Specification, <http://www.yaml.org/spec/1.2/spec.html>
- [5] CEE Network Infrastructure, 1/102 62-CRA 119 1862/5 Uen
- [6] CEE Architecture Description, 5/155 53-AZE 102 01 Uen
- [7] DoD Internet Host Table Specification, <https://tools.ietf.org/html/rfc952>
- [8] Requirements for Internet Hosts -- Application and Support, <https://tools.ietf.org/html/rfc1123>
- [9] CSS User Guide, 1553-AXT 901 11/2-V1
- [10] Manage VLAN, 12/1543-APR 901 0549/1 Uen
- [11] Hyperscale Datacenter System 8000 Customer Product Information, <https://ewstest.ericsson.com/hyperscale/cloud-infrastructure/hyperscale-datacenter-system/hds-8000-product-information/>