Configuration File Guide

Cloud Execution Environment

USER GUIDE

3/1553-CSA 113 085/5 Uen D

Contents

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 the Configuration File Guide and the System Dimensioning Guides. This has been verified against the Cloud Execution Environment reference configurations. See the following documents:

- *HP System Dimensioning Guide, CEE R6*
- *BSP System Dimensioning Guide, CEE R6*
- *Multi-Server System Dimensioning Guide, CEE R6*
- *Single Server System Dimensioning Guide, CEE R6*

The guide is focused on mandatory parameters that must be set according to the certified configuration.

There are parameters that **can** be changed for other configurations. They are either described in Section 4 on page 49, or 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.

Note: Ericsson Hyperscale Datacenter System (HDS), Hewlett Packard (HP), Blade Server Platform (BSP), Dell multi-servers and Dell single server are the supported hardware configurations in CEE R6.3.

This guide describes the initial configurations needed before CEE deployment. For post-deployment configuration options, see the *Runtime Configuration Guide*.

1.1.1 Hardware Terminology

In the configuration file templates for all hardware configurations, <blade> and <shelf> are used. <blade> and <shelf> map to the following hardware terms:

Table 1 Hardware Terminology

Ericsson Cloud Term	Definition	Dell Rack Servers Term	BSP (EBS) Term	HP BladeSystem c7000 Term	HDS Term
CEE Region	Several blade servers or rack-mounted servers				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	Enclosure	
Server	A basic compute engine to host cloud applications in CEE Region	Server	Blade	Blade	Computer System
Blade (only in context of config.yaml)	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	Bay	Logical definition that corresponds to a ComputerSystem identified with a UUID

1.2 Target Groups

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

- Ericsson field personnel
- Ericsson designers

- Test 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.
- 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
  - The amount of RAM on each blade, in GiB. Only needed if not using the `auto` VM huge page allocation.
- Configuration file templates in the `CEE_RELEASE` folder of the tarball:
  - config.yaml.*<hardware_platform>*
  - Switch configuration templates (*<switch_model>*_switch.yaml) (optional)
  - Host network templates (host_nw_*<hardware_platform>*.yaml)
  - Cabling scheme templates (optional)
  - Neutron configuration templates (neutron_ericsson_*<neutron_template>*.yaml)
- EMC storage username and password (optional)
- IP addresses for the external NTP servers

For more information on the required tarball contents, see Section 1.3.1 on page 4.

Have a local copy of `config.yaml`, *<switch_model>*_switch.yaml, and neutron_ericsson_*<neutrontemplate>*.yaml available. In a Single Server deployment, the `neutron_ericsson_user_spec.yaml` is used. In HDS deployment, `neutron_ericsson_sdn_standard.yaml` is used. Compare the examples in the sections to the relevant part of the files. A file editor that can render indentations well (for example, UltraEdit) is recommended.

Note: EMC storage username and password and `<switch_model>_switch.yaml` is not applicable to HDS or Single Server deployment.

Note: Use the `config.yaml` template that matches the hardware, that is, HP, Dell multi-server, Dell Single Server, HDS, or BSP.

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/cabling_scheme/4_x770_hp.yaml
CEE_RELEASE/cabling_scheme/4_x670v_dell.yaml
CEE_RELEASE/cabling_scheme/4_x670v_hp.yaml
CEE_RELEASE/cabling_scheme/2_x670v_hp.yaml
CEE_RELEASE/cabling_scheme/2_x670v_dell.yaml
CEE_RELEASE/switch_config/4_x670v_switch.yaml
CEE_RELEASE/switch_config/4_x770_switch.yaml
CEE_RELEASE/switch_config/cmx_switch.yaml
CEE_RELEASE/switch_config/2_x670v_switch.yaml
CEE_RELEASE/config.yaml.dell
CEE_RELEASE/config.yaml.bsp
CEE_RELEASE/neutron/neutron_ericsson_basic.yaml
CEE_RELEASE/neutron/neutron_ericsson_user_spec.yaml
CEE_RELEASE/neutron/neutron_ericsson_extreme.yaml
CEE_RELEASE/neutron/neutron_ericsson_cmx.yaml
CEE_RELEASE/neutron/neutron_ericsson_sdn_standard.yaml
CEE_RELEASE/neutron/neutron_ericsson_sdn_tight.yaml
CEE_RELEASE/config.yaml.dell-single_server
CEE_RELEASE/config.yaml.hds
CEE_RELEASE/config.yaml.hp
CEE_RELEASE/scripts/parseyaml.rb
CEE_RELEASE/scripts/migrate_fuel.sh
CEE_RELEASE/scripts/install_vfuel.sh
CEE_RELEASE/host_net_templates/host_nw_hp.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_bsp.yaml
CEE_RELEASE/host_net_templates/host_nw_hds.yaml
```

1.4 Generated and Prefilled Passwords

Some generated passwords are stored in a separate file, see `/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 Note on YAML Syntax

The CEE configuration file is a YAML file. The YAML standard defines anchors and aliases, see section Anchors and Aliases in 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: `&label`. An alias consists of a label prefixed with a `*` character: `*label`.

```
...
 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.

2 Basic Parameter Settings

This chapter 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 CEE region name specified in the configuration file will only be applied at the components deployed by Ansible. Most of 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 8, and Section 4.7 on page 66.

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: 25
    l2_vlan_start: <L2.VLAN.START>
    l2_vlan_end: <L2.VLAN.END>
    neutron_config_yaml_file: <neutron_CONFIG.yaml>
  ...
```

Example 4 Neutron Configuration

Note: For SDN configuration, the parameters `l2_vlan_start` and `l2_vlan_end` are not applicable but currently need to be configured with the value `0` (zero).

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

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

`l2_vlan_start` 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, it cannot include any of the CEE internal VLANs.

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 user specific deployments and for the Single Server deployment of CEE.
- `neutron_ericsson_extreme.yaml`: Used for Extreme traffic switches.
- `neutron_ericsson_sdn_standard.yaml`: Used for HDS deployment of CEE.
- `neutron_ericsson_cmx.yaml`: Used for BSP deployment of CEE.

The following example is a Neutron BSP configuration:

```
ericsson:
  ...
  neutron:
    mgmt_vip: 192.168.2.15
    mgmt_subnetmask: 25
    l2_vlan_start: 130
    l2_vlan_end: 3999
    neutron_config_yaml_file: neutron_ericsson_cmx.yaml
  ...
```

Example 5 Neutron Configuration for BSP

For more information on the Neutron configuration file configuration, see Section 4.5 on page 63.

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 6 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 (HP and Dell multi-server).
- **`cmx`**, if using CMX switches setup (BSP). If `initial_setup` is set to **`cmx`**, `cabling_scheme_yaml_file` is not applicable.
- **`None`**, if setting up unmanaged CEE (HDS, Single Server). If `initial_setup` is set to **`None`**, `switching_scheme_yaml_file` and `cabling_scheme_yaml_file` are not applicable.

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 switches on BSP, the following value is valid:
  - `cmx_switch.yaml`

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 HP server:
  - `2_x670v_hp.yaml`
  - `4_x670v_hp.yaml`
  - `4_x770_hp.yaml`
- For Dell multi-server:
  - `2_x670v_dell.yaml`
  - `4_x670v_dell.yaml`

For advanced hardware switch configuration options, see Section 4.7 on page 66.

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 CIC northbound communication.
- `subrack_om_sp` link network between the traffic switches and BGW.

For more information on these networks, see Section 2.6 on page 30.

The structure of the configuration options is as follows:

```
Switch A

  switching:
    -
      ...
      mgmt_vrrp_config:
        -
          vip: 10.0.3.1/25
          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: 10.0.3.2/25
            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: 10.0.3.1/25
          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: 10.0.3.3/25
            tagged: true
```

```
        bgw_config:
          -
            id: 1
            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 7 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>***).

2.3.1.1 Limitation

Dell R620 and R630 Top of Rack (ToR) traffic switch ports are not configured when a VM is booted in a Neutron network used for SR-IOV. Even if CEE Neutron is installed with managed Extreme switches, the Modular Layer 2 (ML2) mechanism driver does not understand to cover SR-IOV based Neutron ports.

To create SR-IOV network or networks, the following command can be used. The parameter values can be different from the ones used in this example:

```
neutron net-create --provider:physical_network=pool_0000_41_00_0 ⇒
--provider:network_type=vlan --provider:segmentation_id=3666 sriov_net1
```

Manual configuration of the Extreme Switch can be done by using the following command. The value of `segmentation_id` must be used for the `tag` parameter.

```
create vlan <sriov_VLAN_NAME>
configure vlan <sriov_vlan_name> tag 3666
configure vlan <sriov_vlan_name> add ports <sriov_port_1> ⇒
<sriov_port_2> tagged
```

The configuration of the ToR switch can be done when the CEE region has been installed, if the VLAN specifications are available at that time. It is suggested to use a VLAN tag for the Neutron network that is not included in the range used for the default physical network. If external connectivity must be configured as well, add the port or ports that connect to the Border Gateway (BGW) as shown below:

```
configure vlan <sriov_vlan_name> add ports <sriov_port_1> ⇒
<sriov_port_2> <port_to_BGW> tagged
```

The configuration changed manually on the ToR switches can be saved on each ToR switch. However, if the consistency checker function is triggered for other reasons, it will restore the previous configuration that does not contain the SR-IOV settings.

2.3.2 Configuring CMX Switches on BSP

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 8 Hardware Switches

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

```
switching:
-
  model: cmx
  provider_vlan_start: 50
  provider_vlan_end: 129
  provider_name_prefix: provider_
switch_config:
  restore_suffix: preCEETenant
  initial_backup: postCEETenant
  migrated_backup: postMigrateCEETenant
```

Example 9   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.
- `restore_suffix`: is the suffix to the name of the BSP configuration backup saved after BSP jumpstart. The installation script of CEE terminates if the file with the expected suffix is not available in BSP. Refer to the installation document *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 Unmanaged Switch

This section describes unmanaged CEE on HDS hardware or in Single Server deployment. The configuration file template contains the following section:

```
ericsson:
  ...
  hw_switches:
    initial_setup: none
  ...
```

Example 10   Unmanaged HW 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>
      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>
      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 11 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.
- `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 BGW.
- `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: ***<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 VLAN (Non-SDN) deployments

In VLAN deployments, additional `tag` parameters need to be configured for both `nbi` and `sbi`.

```
ericsson:
  ...
  cloud_mgmt:
    nbi:
      name: <NETWORK.NAME>
      tag: <VLAN.TAG>
      ...
    sbi:
      name: <NETWORK.NAME>
      tag: <VLAN.TAG>
      ...
```

Example 12 Cloud Management VLAN Configuration

The `tag` parameter is the unique VLAN ID of the Atlas Northbound or Southbound Network.

2.4.3 Additional Configuration for SDN deployments

In SDN deployment, the following extra `cloud_mgmt` parameters need to be configured:

```
  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

network_type

The mechanism through which the Atlas Northbound and Southbound networks are implemented. Allowed values are `vlan` and `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 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*"**

2.5 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`. 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.

```
ericsson:
  ...
  shelf:
    -
      id: 0
      shelf_mgmt:
        ip: 10.0.3.100
        name: subrack_ctrl_sp
        passwd: firstPassword
        username: firstUsername
      blade:
        ...
    -
      id: 1
      shelf_mgmt:
        ip: 10.0.3.102
        name: subrack_ctrl_sp
        passwd: secondPassword
        username: secondUsername
      blade:
        ...
    -
      id: 2
      shelf_mgmt:
        ip: 10.0.3.104
        name: subrack_ctrl_sp
        passwd: thirdPassword
        username: thirdUserName
      blade:
        ...
```

Example 14   Shelf Management of an HP-based CEE

Zero must be used as the ID of the first shelf, and the shelf ID must be monotonically increased for the further shelves.

HP

The shelf configuration options for HP deployment are as follows:

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

Example 15 HP Shelf Configuration

The following shelf manager information has to be updated for each configured shelf:

- `ip` is the IP address of OA-1 in each enclosure.
- `passwd` is the administrator password.
- `username` is the administrator username.

HP uses the term “enclosure” instead of “shelf”. Zero must be used as the ID of the first shelf, and the shelf ID must be monotonically increased in steps of one for the further shelves.

Note: Values of `01`, or `001` are **not** acceptable as shelf IDs. The installation stops if these values are used.

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

- Blade ID 1 is the blade in bay 1.
- Blade ID 2 is the blade in bay 2.
- Blade ID 5 is the blade in bay 5.

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
        passwd: <PASSWORD-SHELF.MANAGER>
        username: <USERNAME.SHELF.MANAGER>
      blade:
        -
          id: 1
```

Example 16   BSP Shelf Configuration

The following shelf manager information has to be updated:

- `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.

The blade ID is the position of the blade, blade ID=(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 17 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 the Dell blade manager. For example:

- Blade ID 1 is the first Dell blade manager.
- Blade ID 2 is the second Dell blade manager.

Unmanaged Server

Note: HDS deployment uses unmanaged CEE.

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 18 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.
- `control0` is the MAC address of the primary boot NIC of the server.
- `control1` is the MAC address of the alternate boot NIC (optional).

SDN Blade Configuration

For HDS SDN deployment, two extra parameters are needed to be in the `blade` section to indicate the VTEP network where the blade belongs:

```
shelf:
...
  blade:
  ...
    vtep_net: sdn_underlay_sp_<net_id>
    vtep_ip: <IP.OF.THE.VTEP>
```

Example 19 SDN Blade Configuration

- `vtep_net` is the network name, in the form of **`sdn_underlay_sp_`*`<net_id>`***.
- `vtep_ip` is the IP of the blade.

This information can be fetched from the vPOD definition of HDS. If the blade IP is missing, any free IP can be set for the blade in the IP range defined for the VTEP, see Section 2.6 on page 30.

ScaleIO Blade Configuration

To dedicate a blade to be part of the ScaleIO cluster, 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 2.9.3 on page 40. The following `blade.scaleio` parameters are available:

- `mode` has two available values, **`dedicated`** and **`shared`**. Currently only **`dedicated`** is supported.
- `roles` has three parameters, `mdm`, `gw` and `sds`. For `sds` role, the following parameters need to be specified:
  - Protection domain, as defined in the `scaleio` storage section
  - The list of devices to be used for storage
  - The pool name the device belongs to, as defined in the `scaleio` storage section

```
ericsson:
  shelf:
    blade
      - id:  ..
        scaleio:
          mode: dedicated
          roles:
            mdm:
            gw:
            sds:
              - protection_domain: domain1
                devices:
                  - name: /dev/sdb
                    pool: pool1
```

Example 20   ScaleIO Blade Configuration

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 (huge pages), 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 and/or vFuel VMs. The allocation of resources depends on whether the host contains these infrastructure VMs or not.

hw_uuid

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 21   Hardware System UUID Assignment

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

2.5.3 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: *HP_GEN8_nic_assignment
          ....
```

Example 22   NIC Assignment of an HP Server

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

Example 23   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 from the predefined values. If the hardware is not listed in `nic_assignments:`, a new NIC mapping must be added to the list. See Section 4.2 on page 55 for information on how to define a new NIC mapping.

2.5.4 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 huge pages. 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 huge pages is defined in `config.yaml` for each blade. The remainder of the physical memory that is not allocated in huge pages is accessible in 4 KiB pages. If using the keyword **`auto`** for the VM huge page allocation, the installer calculates the huge page count.

To reserve huge pages 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: *BSP_GEP7_reservedHugepages_with_vcic_and_vfuel
          reservedCPUs: *auto_reservedCPUs_with_vcic_and_vfuel
          reservedDisk: *reservedDisk_for_vcic_and_vfuel
          cfm_role: active
          virt:
            cic:
              id: 1
        -
          id: 2
          nic_assignment: *BSP_GEP5_nic_assignment
          reservedHugepages: *BSP_GEP5_reservedHugepages_with_vcic_and_vfuel
          reservedCPUs: *auto_reservedCPUs_with_vcic_and_vfuel
          reservedDisk: *reservedDisk_for_vcic_and_vfuel
          cfm_role: active
          virt:
            cic:
              id: 2
        -
          id: 3
          nic_assignment: *BSP_GEP5_nic_assignment
          reservedHugepages: *BSP_GEP5_reservedHugepages_with_vcic
          reservedCPUs: *auto_reservedCPUs_with_vcic
          reservedDisk: *reservedDisk_for_vcic
          cfm_role: active
          virt:
            cic:
              id: 3
        -
          id: 4
          nic_assignment: *BSP_GEP5_nic_assignment
          reservedHugepages: *BSP_GEP5_reservedHugepages
          reservedCPUs: *auto_reservedCPUs
          cfm_role: passive
        ...
```

Example 24 Huge Page Reservations for Three Different Blades in BSP

The amount of memory to reserve in huge pages depends on whether the Compute host contains a vCIC and/or vFuel. The examples above illustrate different alternatives. Recommended huge page reservations are listed in the `reservedHugepages` section. Refer to one of the available predefined settings by using an alias as shown in the examples.

There is a lower limit on the amount of memory for the host OS. In other words, this is the amount of physical memory that must not be allocated to huge pages. The actual value depends on whether a vCIC is hosted on the Compute host. The limits are defined by the `compute_os_reserved_mem` and `compute_with_vcic_os_reserved_mem` configuration parameters. The values are expressed in MiB:

```
compute_os_reserved_mem: 8192
compute_with_vcic_os_reserved_mem: 14336
```

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
```

2.5.5 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: *DELL_630_OEM_reservedHugepages_with_vcic
          reservedCPUs: *auto_reservedCPUs_with_vcic
          reservedDisk: *reservedDisk_for_vcic
          virt:
            cic:
              id: 3
        -
          id: 4
          blade_mgmt:
            ...
          nic_assignment: *DELL_630_OEM_nic_assignment
          reservedHugepages: *DELL_630_OEM_reservedHugepages
          reservedCPUs: *auto_reservedCPUs
        -
          id: 5
          blade_mgmt:
            ...
          nic_assignment: *DELL_630_OEM_nic_assignment
          reservedHugepages: *DELL_630_OEM_reservedHugepages_with_vfuel
          reservedCPUs: *auto_reservedCPUs_with_vfuel
          reservedDisk: *reservedDisk_for_vfuel
          vfuel: ""
          ...
```

Example 25 CPU Allocation

If the Compute host contains a vCIC and/or vFuel, CPUs must be reserved for those as well. The examples above illustrate different alternatives. Recommended CPU reservations are listed at the beginning of the configuration template in the `reservedCPUs` section. Refer to one of the available predefined settings by using an alias as shown in the examples.

Note: The default CPU reservation uses automatic CPU allocation in Multi-Server configurations but uses manual CPU allocation for Single Server deployments. See Section 4.1 on page 50 for details on the supported CPU allocation methods.

2.5.6 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: *HP_GEN9_nic_assignment
          reservedHugepages: *HP_GEN9_reservedHugepages_with_vcic
          reservedCPUs: *auto_reservedCPUs_with_vcic
          reservedDisk: *reservedDisk_for_vcic
          virt:
            cic:
              id: 2
        -
          id: 2
          ...
          nic_assignment: *HP_GEN9_nic_assignment
          reservedHugepages: *HP_GEN9_reservedHugepages_with_vfuel
          reservedCPUs: *auto_reservedCPUs_with_vfuel
          reservedDisk: *reservedDisk_for_vfuel
          vfuel: ""
          ...
```

Example 26 Disk reservation for vCIC and vFuel

The value set for vFuel disk size is 50 GB in the `config.yaml` templates. The recommendation is to increase it to 70 GB if there is enough disk space on the vFuel host. A vFuel disk size of 50 GB might not be enough for two generations of CEE software to be installed simultaneously, which is needed during an update procedure.

```
ericsson:
  ...
  - &reservedDisk_for_vfuel
    - owner: vfuel
      size: 70G
      ...

Or...

  - &reservedDisk_for_vcic_and_vfuel
    - owner: vfuel
      size: 70G
      ...
```

Example 27 vFuel Disk Size 70 GB

2.5.7 Cloud Infrastructure Controller (CIC)

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 huge pages, CPU, and disk storage suitable for vCIC.

2.5.8 Host Networking

The used host networking yaml file must correspond to the used hardware.

```
host_networking:
  template_yaml_file: host_nw_hp.yaml
```

Example 28 Host Networking, HP

2.5.9 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, no vFuel is used.

To be able to contain a vFuel, the blade definition must contain resource reservations for huge pages, 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).

2.6 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 29 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 4.13 on page 74.

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

Table 2 Hardware-dependent Network Configuration

Network name	HDS	Dell multi-server	HP	BSP	Single server
`subrack_ctrl_sp`	N/A	to be configured	to be configured	N/A	to be configured
`subrack_om_sp`	N/A	to be configured	to be configured	N/A	N/A
`cee_om_sp`	to be configured	to be configured	to be configured	to be configured	to be configured
`cee_ctrl_sp`	to be configured	preconfigured	preconfigured	preconfigured	preconfigured
`iscsi_san_pda`	to be configured	preconfigured	preconfigured	preconfigured	N/A
`iscsi_san_pdb`	to be configured	preconfigured	preconfigured	preconfigured	N/A
`swift_san_sp`	to be configured	preconfigured	preconfigured	preconfigured	N/A
`migration_san_sp`	to be configured	preconfigured	preconfigured	preconfigured	N/A
`hds_agent`	to be configured	N/A	N/A	N/A	N/A
`sdn_underlay__sp_`*<net_id>*	to be configured	N/A	N/A	N/A	N/A

subrack_ctrl_sp

Note: The use of `subrack_ctrl_sp` is specific to setups using Extreme switches.

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.

```
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 30 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 controller for the CM-HA fencing feature.
- `end` is the last IP address in range for the subnet offered to the controller 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 use of `subrack_om_sp` is specific to multi-server deployments using Extreme switches.

The network `subrack_om_sp` is used as a link network between the traffic switches and the border gateways. 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 31   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 CIC 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 32   CIC 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 7 including the endpoints:

- The first address in the range is always the vRouter address.

- The second IP address in the range is the CIC HA proxy.
- The third, fourth, and fifth are individual addresses of the three CICs.
- The last two IP addresses are reserved for CIC internal use.

cee_ctrl_sp

Network used for CEE internal system functions. `cee_ctrl_sp` VLAN needs to be configured on HDS.

iscsi_san_pda

Network used for persistent block storage (Cinder) in a centralized storage array. `cee_ctrl_sp` VLAN needs to be configured on HDS.

swift_san_sp

Network used for persistent block storage (Cinder) in a centralized storage array. `cee_ctrl_sp` VLAN needs to be configured on HDS.

migration_san_sp

Network used for Swift traffic on the storage switching domain. `cee_ctrl_sp` VLAN needs to be configured on HDS.

hds_agent

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.

sdn_underlay_sp

The network `sdn_underlay_sp` is used for SDN managed networks to handle the tenant VM traffic. The network is declared in HDS per (pair of) leaf switch for every vPOD.

```
ericsson:
  ...
  networks:
    ...
    -
      name: sdn_underlay_sp_<net_id>
      tag: <VLAN,TAG>
      cidr: <NETWORK.IP/PREFIXSIZE>
      gateway: <IP.OF.THE.GW>
   ...
```

Example 33 SDN Managed Networks for VTEP Handling

The `<net_id>` and `cidr` IP range is different for each `sdn_underlay_sp` network.

The `gateway` must be an existing, reachable gateway of this network and come from HDS.

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>]
    orphan_mode_stratum: <ORPHAN.MODE.STRATUM>
  ...
```

Example 34 NTP Configuration

Define up to four external NTP server IP addresses. A minimum of one NTP server IP address is required, and at least three are recommended. 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 35 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 36 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:4|BoCdm;S-A1]a>o
  ...
```

Example 37 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 login 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 38 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 Centralized storage

Note: Centralized storage is not applicable to Single Server CEE, BSP and HDS.

Centralized storage is available for CEE hardware including EMC storage solution (VNX5400). The structure of the `centralized` storage subsection is as follows:

```
ericsson:
  ...
  storage:
    centralized:
      type: <CENTRALIZED.STORAGE>
      cee_managed: true
      hw_type: VNX5400
      storagepool_name: <STORAGE.POOL.NAME>
      mgmt_ip_A: 192.168.2.12
      mgmt_ip_B: 192.168.2.13
      emc_admin:
        user: <EMC.USER>
        passwd: <EMC.PASSWORD>
      ...
```

Example 39   Storage Pool Type

type

To set up centralized storage, set the ***type*** parameter for ***<CENTRALIZED.STORAGE>*** to **EMC2**. If centralized storage is not available or not wanted, set the ***type*** to **None**.

cee_managed

Set `cee_managed` to **true** to enable CEE to manage the centralized storage array. Only set to **false** after careful consideration. Scripts setting and cleaning up the VNX prior to an installation do **not** run if the parameter is set to **false**. LDAP is not configured for authentication on the VNX. Contact Ericsson support for more information.

storagepool_name

The variable ***<STORAGE.POOL.NAME>*** must match the name of the Storage Pool for Cinder, created on the VNX as written in the document *VNX5400 SW Installation*, Reference [2]. The admin user ***<EMC.USER>*** created during VNX5400 installation has password ***<EMC.PASSWORD>***.

emc_admin

The EMC admin refers to the admin user created during the VNX5400 installation. ***<EMC.USER>*** has to be replaced with the admin username and ***<EMC.PASSWORD>*** with the password.

2.9.2 Adding iSCSI Ports to Centralized Storage

In the `config.yaml` template, four iSCSI ports are defined, two for each of the Storage Processors (A and B).

See the pre-defined iSCSI ports:

```
ericsson:
  ...
  storage:
    centralized:
      iscsi_ports:
        -
          port: 0
          SP: A
          name: iscsi_san_pda
          target_ip: 192.168.11.1
          tagged: true
        -
          port: 1
          SP: A
          name: iscsi_san_pdb
          target_ip: 192.168.12.1
          tagged: true
        -
          port: 0
          SP: B
          name: iscsi_san_pda
          target_ip: 192.168.11.2
          tagged: true
        -
          port: 1
          SP: B
          name: iscsi_san_pdb
          target_ip: 192.168.12.2
          tagged: true
      ...
```

Example 40 Pre-defined iSCSI Ports

Up to 12 additional ports can be added in the `iscsi_ports` section using this structure:

```
          port: <PORT.NUMBER>
          vport: <PORT.NUMBER>
          SP: <STORAGE.PROCESSOR>
          name: <NETWORK.NAME>
          target_ip: <TARGET.IP>
          tagged: true
```

Example 41

port

The iSCSI port number depends on the slot used for the interface card, and the port used on the interface card. Valid values are 0–7.

vport

The optional parameter `vport` is only to be used if the virtual port number configured on the VNX is non-zero. This is normally not the case, but can be relevant if CEE does not manage the centralized storage array (that is if parameter `cee_managed` is set to **`false`**).

SP

The Storage Processor connected. Valid values are `A` and `B`.

name

The name of the network used on this port. Valid values depend on the names defined in the `networks` section of `config.yaml`, see Section 2.6 on page 30.

target_ip

The iSCSI target IPs must be in the CIDR but outside the specified range in `iscsi-left` and `iscsi-right`.

tagged

VLAN tagging `tagged` must be set to **`true`**.

2.9.3 ScaleIO Configuration

ScaleIO block storage is configured in the `scaleio` storage subsection in `config.yaml`. If the `scaleio` section exists during deployment, ScaleIO is deployed and configured as storage backend. The structure of the `scaleio` section is as follows:

```
ericsson:
  storage:
    scaleio:
      license:
      protection_domains:
          - name: <PROTECTION_DOMAIN_NAME_1>
            pools:
              - name: <STORAGE_POOL_NAME_1>
                type: <PROVISIONING_TYPE_1>
      cluster_name: <SCALEIO CLUSTER NAME>
      frontend_networks:
          - <FRONTEND_NETWORK_1>
          - <FRONTEND_NETWORK_2>
      backend_networks:
          - <BACKEND_NETWORK_1>
          - <BACKEND_NETWORK_2>
      password: <SCALEIO_PASSWORD>
      gateway_admin_password: <SCALEIO_PASSWORD>
      lia_token: <SCALEIO_PASSWORD>
      gateway_port: 443
      gateway_user: admin
      round_volume_capacity: True
      verify_server_certificate: False
```

Example 42 Configuring ScaleIO

The following parameters can be configured for ScaleIO:

license

This parameter is not in use in R6.3.

protection_domains

The protection domain has a `name`. and storage `pools`. `pools` has two parameters:

- `name` is the storage pool name
- `type` is the storage pool type, which can be `thick` or `thin`.

cluster_name

Cluster name

frontend_networks

List of ScaleIO frontend networks

backend_networks

List of ScaleIO backend networks

password

password is the password for the admin user.

gateway_admin_password

gateway_admin_password is the password for the gateway admin user. It has the same value as password.

lia_token

lia_token is the LIA password for node management. It has the same value as password.

gateway_port

Gateway port. The value must be 443.

gateway_user

Gateway user. The value must be admin.

To dedicate a blade to be part of the ScaleIO cluster, the role of the blade in the cluster has to be defined in the blade section, see Page 23.

2.10 Local Disks

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 CEE Backup.

The partition sizes must be dimensioned according to the site-specific needs. For example, if the data retention times for Zabbix are to be increased, then the size for the MySQL partition (parameter mysql_size) must be increased. For information on MongoDB and MySQL (database size), refer to the System Dimensioning Guides:

- *HP System Dimensioning Guide, CEE R6*
- *BSP System Dimensioning Guide, CEE R6*
- *Multi-Server System Dimensioning Guide, CEE R6*
- *Single Server System Dimensioning Guide, CEE R6*

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

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

Example 43 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:
       passwd:'<IDAM.CEEBACKUP.PASSWD>'
     ceeadm:
       passwd:'<IDAM.CEEADM.PASSWD>'
     ...
```

Example 44 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: `ceebackup` is not applicable to a 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.yml` 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 45 LDAP

2.13 VNX Users

`vnx-log-fetcher` is set up when managed VNX (EMC VNX storage) is present, if these credentials are left blank in `config.yaml`, the entries in `/etc/openstack_deploy/user_secrets.yml` are used.

```
ericsson:
   vnx-log-fetcher:
     mysql:
       password: ''
     vnx:
       password: ''
   ...
```

Example 46 VNX Users

- Account name is always `vnx-log-fetcher`.
- `mysql` password is used to access the database where logs from VNX are temporarily stored. `idam_user_vnxlf_vnx_key` is used if this credential is left blank.

- vnx password is used to access the VNX. idam_user_vnxlf_galera_password is used if this credential is left blank.

2.14 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.

2.15 Swift Configuration Options

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

Note: The prerequisite to configure Swift on backend storage is as follows:

- To use EMC VNX as storage backend: a properly configured centralized storage, see Section 2.9.1 on page 37.

- To use ScaleIO as storage backend: a properly configured ScaleIO block storage, see Section 2.9.3 on page 40.

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 47 Swift Configuration Options

type

The type of the backend storage system. The supported values are **centralized** and **scaleio**:

- To configure Swift to use EMC VNX as storage backend, the `type` must be set to **centralized**.
- To configure Swift to use ScaleIO as storage backend, 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 VNX manually after installation, see the *Swift Store on VNX Activation* operating instructions. To deploy Swift on ScaleIO manually after installation, see the *Swift Store on ScaleIO Activation* operating instructions.

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

lun_size

`lun_size` specifies the LUN size of Swift on the storage backend. 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**. The maximum initial size of the Swift store on EMC VNX backend is 6000 GiB / 6 TiB.

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 Standard 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 alongside a remote SDN Controller without any other extension package. The remote SDN controller will be used to manage the tenant network and create tunnels on compute blades for tenants network isolation. Software VTEPs needs to be created manually at the CSC CLI after CEE deployment. The parameters needed for SDN configuration are as follows:

```
sdn:
  type: standard
  sdnc_sbi_vip: <SBI.VIP.OF.SDNC>
  sdnc_nbi_vip: <NBI.VIP.OF.SDNC>
  sdnc_sbi_gw_ip: <SBI.GW.IP>
  sdnc_admin_username: <SDNC.ADMIN.USERNAME>
  sdnc_admin_password: <SDNC.ADMIN.PASSWORD>
  remote_gre_term: <REMOTE.GRE.TERM.IP>
```

Example 48 SDN Configuration

type

The SDN integration type. Only the `standard` type is supported. In standard mode, CSC is installed on dedicated nodes.

sdnc_sbi_vip

The IP of the southbound interface used for CSS-CSC communication

sdnc_nbi_vip

The IP of the northbound interface used for Neutron-CSC communication

sdnc_sbi_gw_ip

The IP used to set the static route on the compute blades in order to let CSS access CSC SBI IP

sdnc_admin_username

Username for CSC authentication

sdnc_admin_password

Password for CSC authentication

remote_gre_term

The remote GRE term IPs. If there are multiple IPs, this should be a yaml list, for example: ['10.33.215.30', '10.33.215.31']

Additional configuration is needed for SDN in the following sections of `config.yaml`:

- Additional `blade` parameters are needed to indicate the VTEP network, see Page 22.
- The `sdn_underlay_sp_`*<net_id>* network has to be configured, see Page 34.
- The

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 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 after finishing the evacuation of the VMs. This function can help recover the failed compute. Default value: **`true`**

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 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 49 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>
 ...
```

3 Post-Configuration Activities

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.

4 Advanced Parameter Settings

This chapter is for non-certified configurations, for information about parameters that **can** be changed. The settings described in this section are considered to work, but they have not necessarily been formally verified by CEE Integration and Verification.

4.1 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:

- *HP System Dimensioning Guide, CEE R6*
- *BSP System Dimensioning Guide, CEE R6*
- *Multi-Server System Dimensioning Guide, CEE R6*
- *Single Server System Dimensioning Guide, CEE R6*

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, this practice is not recommended. If the goal is to get a CPU allocation that works well for a wide range of workloads, then all reservations must use the `count` key with value **`auto`**. Optionally, the default number of CPU cores can be overruled by specifying the exact number of CPU cores to allocate. If the goal is to optimize the performance for a specific workload or configuration, then all reservations must use the `cpus` key that provides full control for the cloud administrator. The automatic CPU allocation scheme is not recommended for a Single Server deployment.

4.1.1 Allocating CPUs for OVS

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.

OVS has two constraints on the CPU allocation related to the NUMA topology of the server:

- At least one CPU must be reserved for PMD threads on each NUMA node where a physical NIC is located. This rule only applies to those NICs that are handled by OVS/DPDK. In CEE, the NICs for tenant traffic are affected.
- The OVS control process must run on the same NUMA node with at least one PMD thread. That is, at least one CPU reserved for PMD threads must be on the same NUMA as the CPU assigned to the OVS control process.

Failure to fulfill these constraints leads to system malfunction.

CPUs reserved for the owner **`ovs`** in `config.yaml` are used for PMD threads. If the `count` key is used for reserving the OVS PMD CPUs, at least one CPU core is allocated to OVS PMD threads on each NUMA node. Therefore, at least two cores are required on Dell and HP blades, while only one is required on BSP. No more cores are allocated if the CPU reservation is `count:` **`auto`**. If the number of CPU cores to allocate is specified explicitly, then the given value must not be lower than the number of NUMA nodes. Any additional cores are allocated sequentially from the available cores starting with NUMA node 0 and then continuing with NUMA node 1, if necessary.

Note: Providing two CPUs located on the same physical core for OVS can result in decreased throughput capacity. As such, it is recommended to allocate one CPU for OVS (or one per NUMA node, depending on the hardware). Refer to the relevant *System Dimensioning Guides*.

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:
  - &OVS_single_thread_DELL_630_reservedCPUs
    - owner: vm
      cpus: 1,3,5-22,25,27,29-46
    - owner: ovs
      cpus: 2,23
      cpus_nonpmd: 0
    - owner: idle
      cpus: 26,47
      notick: true
```

The actual CPU IDs depend on the server and CPU model.

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

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
```

After CEE deployment, interrupt handling has to be adjusted for the CPUs configured as **`idle`**. Refer to *SW Installation in Multi-Server Deployment* or *SW Installation in Single Server Deployment*.

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 huge pages 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 supported to configure a CEE Region without DPDK acceleration for traffic network (provided by the corresponding OVS bridges).

4.1.2 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, the vCIC uses dedicated physical CPUs, and the VMs virtual CPUs are mapped one-to-one to the reserved physical CPUs. If `isolated:` **`false`** is configured, the parameter `count` only defines the number of virtual CPUs for the VM. The number of vCPUs to use is calculated as **`count×HTs/core`**. These vCPUs can float over the pool of non-isolated physical CPUs. If `isolated:` **`false`** is defined, the reservation can only use `count`, and the use of `cpus` is not supported in this case.

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

Example 50 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 the memory allocation for the **`vcic`** owner, the `memnode` key can 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 given, the VM memory is allocated from NUMA node 0.

```
ericsson:
  ...
  reservedHugepages:
    - owner: vcic
      size: 1GB
      count: 10
      memnode: 1
  ...
```

Example 51 Reserve 10 GiB RAM for a vCIC using physical memory on NUMA node 1

4.1.3 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 reserved for 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, 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.

Four CPU cores are reserved for vCIC, and one CPU core is reserved for vFuel, if `count: auto` is used.

The first CPU core on NUMA node 0 is assigned to the host OS (left non-isolated). Then one CPU core is reserved for OVS PMD threads on each NUMA node. On NUMA node 0, the first available core is reserved for OVS PMD threads. On NUMA node 1, the last available CPU core is reserved for OVS PMD if necessary. Then the remaining cores are reserved for the host OS. After that, the CPUs are reserved for vFuel, vCIC, and tenant VMs in this order.

The first CPU used by the host OS is also assigned to the OVS control process unless it is explicitly defined. See Section 4.1.1 on page 51 for details.

4.1.4 CPU Allocation for Single Server Deployment

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

```
ericsson:
  ...
  reservedCPUs:

    - owner: vm
      cpus: 0,24,2,26,4,28,5,29,7,31,9,33,11,35,13,37,15,39,17,41,19,43
    - owner: ovs
      cpus: 3,27
      cpus_nonpmd: 1
    - owner: vcic
      count: 2
      isolated: false
    ...
```

Example 52 Allocating vCPUs Reserved for VMs

This example illustrates some of the concepts discussed before:

- CPUs are assigned manually using the `count` 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 varies among systems. This example assumes a Dell R630 server equipped with 2 Intel Xeon E5-2680 v3 processors.
- Only one CPU core is reserved for OVS PMD threads on NUMA node 1. Therefore, only physical NICs connected to NUMA node 1 can be used for tenant traffic.
- The CPU to be used for the OVS control process is explicitly set to CPU 1 to override the system default. The system default would be CPU 6 on NUMA node 0. But this results in a violation of the rule of using CPUs for the OVS control process on a NUMA node that contains OVS PMD threads.
- No dedicated CPUs are reserved for the vCIC, it uses any of the non-isolated CPUs.

For more information on the Single Server configuration, refer to *Single Server System Dimensioning Guide, CEE R6*.

4.2 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`, add or modify the relevant element in the list. This is mandatory.

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

Each assignment must be labeled with an anchor. See Section 1.5 on page 5 for the use of anchors and aliases.

```
ericsson:
  nic_assignments:
  - &HP_GEN9_nic_assignment
    control0: "0000:06:00.0"
    control1: "0000:06:00.1"
    data0:    "0000:08:00.0"
    data1:    "0000:08:00.1"
    storage0: "0000:87:00.0"
    storage1: "0000:87:00.1"
  - &HP_GEN8_nic_assignment
    control0: "0000:04:00.0"
    control1: "0000:04:00.1"
    data0:    "0000:05:00.0"
    data1:    "0000:05:00.1"
    storage0: "0000:21:00.0"
    storage1: "0000:21:00.1"
  - &DELL_630_OEM_nic_assignment
    control0: "0000:01:00.0"
    control1: "0000:01:00.1"
    data0:    "0000:81:00.0"
    data1:    "0000:03:00.0"
    storage0: "0000:81:00.1"
    storage1: "0000:03:00.1"
  - &DELL_630_OEM_nic_assignment_with_sriov
    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_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"
  - &BSP_GEP5_nic_assignment
    control0: "0000:02:00.1"
    control1: "0000:02:00.2"
    data0:    "0000:01:00.0"
    data1:    "0000:01:00.1"
    storage0: "0000:04:00.0"
    storage1: "0000:04:00.1"
  ...
```

Example 53   Information on Available NIC Assignments

```
ericsson:
  nic_assignments:
  - &DELL_630_OEM_single_server_nic_assignment
    control0: "0000:01:00.0"
    data0: "0000:81:00.0"
  ...
```

Example 54   Single Server NIC Assignment

4.3 SR-IOV

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 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 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: "PHY1"
    - pci_address: "0000:41:00.1"
      bandwidth: 10000000
      physical_network: "PHY1"
  - &DELL_630_OEM_sriov_info
    - pci_address: "0000:83:00.0"
      bandwidth: 10000000
      physical_network: "PHY1"
    - pci_address: "0000:83:00.1"
      bandwidth: 10000000
      physical_network: "PHY1"
  - &DELL_630_sriov_info
    - pci_address: "0000:84:00.0"
      bandwidth: 10000000
      physical_network: "PHY1"
    - pci_address: "0000:84:00.1"
      bandwidth: 10000000
      physical_network: "PHY1"
  ...
  shelf
    -
      blade:
        -
          id: 2
          sriov:
            devices:*DELL_630_sriov_info
            vf: 8
      ...
```

Example 55   8 VFs with 2 SR-IOV devices in blade 2

sriov_configs

The dictionaries containing the PCI addresses of SR-IOV NICs (Physical Functions) 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.

SR-IOV Blade Configuration

Note: The `sriov` section is optional, and only configurable on Dell hardware platforms.

- `devices` is a list of dictionaries containing the PCI addresses of SR-IOV NICs (Physical Functions). The list of devices cannot contain more than two devices.

- `vf` specifies the number of Virtual Functions on Compute nodes assigned to each PF (NIC). The maximum value is 16. This is a mandatory parameter if `sriov` is defined.

sriov_segmentation_type

The global SR-IOV segmentation needs to be set if the above `sriov` configuration is enabled on at least one of the blades:

```
ericsson:
  ...
  sriov_segmentation_type: vlan
```

Example 56 SR-IOV Segmentation

The value of `sriov_segmentation` can be **`vlan`** or **`flat`**.

The following configuration is needed:

- For **`vlan`** type:

```
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
```

- For **flat** type:

```
ericsson:
  ...
  sriov_segmentation_type: flat
  neutron:
    ...
    neutron_config_yaml_file: neutron_ericsson_user_spec.yaml
  ...
  hw_switches:
    initial_setup: extreme
    switching_scheme_yaml_file: 2_x670v_switch.yaml
    cabling_scheme_yaml_file: 2_x670v_dell.yaml
```

`neutron_ericsson_user_spec.yaml` is to contain only the following line:
`conf_type: ericsson_user_spec`

The cabling scheme yaml file must be updated according to the allocation of SR-IOV ports in the ToR 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 57 SR-IOV Cabling Scheme Configuration

4.3.1 Limitation

Dell R620 and R630 ToR traffic switch ports are not configured when a VM is booted in a Neutron network used for SR-IOV. See Section 2.3.1.1 on page 12 for more information.

4.4 Bandwidth-Based Scheduling

4.4.1 Neutron Networks

The `neutron_networks` section describes the characteristics of the physical Neutron networks on a host. 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: Currently the only supported Neutron physical network is the `default` network. `devices:` 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: *HP_GEN9_nic_assignment
          reservedHugepages:
            …
          reservedCPUs:
            ...
          vswitch_capacity: <vswitch capacity>
          neutron_networks: *neutron_networks_std_limit
```

Example 58   Neutron_networks

4.4.2 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: *HP_GEN9_nic_assignment
          reservedHugepages:
            ...
          reservedCPUs:
            …
          vswitch_capacity: <vswitch capacity>
```

Example 59   vswitch_capacity

Note: The value of `vswitch_capacity` is documented in the System Dimensioning Guides:

- *HP System Dimensioning Guide, CEE R6*
- *BSP System Dimensioning Guide, CEE R6*
- *Multi-Server System Dimensioning Guide, CEE R6*
- *Single Server System Dimensioning Guide, CEE R6*

4.5 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 60 neutron_ericsson_user_spec.yaml Template File

```
conf_type: ericsson_user_spec
additional_packages: ["neutron-plugin-bsp"]
neutron_configuration_files:
  -
    name: ml2_conf.ini
    option:
      ml2/mechanism_drivers: openvswitch,l2population,bsp
  -
    name: ml2_conf_bsp.ini
    option:
      ml2_bsp/management_ip: 192.168.2.2
      ml2_bsp/bsp_tenant: CEE
```

Example 61 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>***

```
option:
 # Comma-Separated list of <vni_min>:<vni_max> tuples enumeratingranges
 #of VXLAN VNI IDs that are available for tenant network allocation
 M12_type_vxlan/vni_ranges: 100:10000
```

Example 62 neutron_ericsson_sdn_standard.yaml Template File

Different ***<vni_min>:<vni_max>*** ranges need to be chosen for the different vPODs, with the following limitations:

- ***<vni_min>*** and ***<vni_max>*** must be integers in the range of [1 ... 16777216].
- ***<vni_min>*** must be lower than ***<vni_max>***.

4.5.1 Prevent ARP Spoofing

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

The ARP spoofing rules can be enabled or disabled globally per deployment. If unspecified, prevention of ARP spoofing is by default enabled. If the IP addresses of the VNFs are manually configured, the ARP spoofing rules must be switched off. For more information, refer to the section about port security of the document *CEE Network Infrastructure*, Reference [5].

To disable the ARP spoofing prevention function, add the `prevent_arp_spoofing` parameter to the Neutron configuration file template with `False` value:

```
conf_type: <NEUTRON_CONFIG.YAML>
neutron_configuration_files:
  -
    name: ml2_conf.ini
    option:
      agent/prevent_arp_spoofing: False
```

This workaround is only applicable if using `neutron_ericsson_user_spec.yaml` template.

4.6 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 63 Nova Configuration Options

4.7 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 HP and Dell multi-server deployments, and CMXB in BSP hardware. No initial hardware switch configuration is used for the 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	**`neutron_ericsson_cmx.yaml`**	**`cmx`**

Hardware Deployment	`neutron:` `  neutron_config_yaml_file:`	`hw_switches:` `  initial_setup:`
HP Dell multi-server	**`neutron_ericsson_extreme.yaml`**	**`extreme`**
Single Server	**`neutron_ericsson_user_spec.yaml`**	**`none`**
HDS	**`neutron_ericsson_sdn_standard.yaml`**	**`none`**
Other, user specific	**`neutron_ericsson_user_spec.yaml`**	**`none`**

See the following sections for more information:

- Section 2.3 on page 8 for hardware switch configuration
- Section 2.2 on page 7 for Neutron configuration files
- Section 4.5 on page 63 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 64 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 65 No Hardware Switch Configuration, Single Server

4.8 Multiple Border Gateways

Two Border Gateways (BGWs) are specified in the cabling schema, `4_x770_hp.yaml`, `4_x670V_hp.yaml`, or `2_x670V_hp.yaml`:

```
external_components:
  border_gateways:
    - id: 1
      name: BGW-1
      switch_id: 1
      ports: [85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96]
      master: 85
      partition: 4x10G
    - id: 2
      name: BGW-2
      switch_id: 2
      ports: [85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96]
      master: 85
      partition: 4x10G
  ...
```

Example 66 Two Border Gateways 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 67 Border Gateways in Switch Configuration File

Adding further border gateways 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 68 Additional Border Gateway 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 69 Switch Configuration File Update for Multiple BGWs

4.9 Change of Border Gateway Settings

To configure BGW with settings different from the default in CEE, the startup configuration of the 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_R10
```
<---- must be replaced with higher number, for example 15B_R11

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.

4.10 Change of Linux I/O Scheduler

The user can select I/O scheduler for computes. This changes the strategy used for scheduling IO requests.

```
ericsson:
  ...
  timezone: Etc/UTC
  compute_io_scheduler: deadline
  ...
  neutron:
  ...
```

Example 70 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.

4.11 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 71 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

4.12 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-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256⇒
-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:
                     ECDHE-RSA-AES256-GCM-SHA384'
      protocols: TLSv1.2
    gnutls:
      priority: 'NONE:+SUITEB128:+SUITEB192:+VERS-TLS-1.2'
    haproxy:
      sslprotocols: no-sslv3 no-tlsv10 no-tlsv11
      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 72 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)

Recommended setting:
`ECDHE-ECDSA-AES128-GCM-SHA256:ECDHE-ECDSA-AES256-GCM-SHA384:ECDHE-RSA-AES128-GCM-SHA256:ECDHE-RSA-AES256-GCM-SHA384`

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-sslv3 no-tlsv10 no-tlsv11`

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 CICs. 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.

Hostnames for the controllers 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.

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.

4.13 Fuel Administration Network

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

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

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`:

```
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 73 Fuel Administration Network

4.14 Location of Logs

Note: Not applicable to HDS.

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 74 Crashes Local

```
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 75 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`**. HP, Dell, and Single Server store crashes locally in each blade/server by default. BSP stores most crashes in the CICs, 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 controller and Compute to Fuel or not. Boolean parameter, default value: **`false`**

forward_to_controller

Indicates whether to forward logs from Compute to controller or not. Boolean parameter, default value: **`false`** on HP/Dell/Single Server, **`true`** on BSP

forward_to_external

Indicates whether to forward logs from both controller and Compute to 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 controller. Boolean parameter, default value: **`true`**

local_on_compute

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

4.15 Link Monitoring for CEE on BSP

4.15.1 Control Network

4.15.1.1 Link Redundancy Based on ARP Monitoring

ARP setup is only applicable to BSP. To enable link monitoring, configure the ARP settings as follows:

```
ericsson:
  ...
  arp_setup:
    control:
      schema: <schema>
      network: fw-admin
      [router: <cmx_virtual_router>]
      [target_ips:]
        [-<ip_1>]
        [- ...]
        [-<ip_n>]
    ...
```

Example 76 ARP Link Redundancy

ARP monitoring is only implemented for control networks with the following parameters:

- `schema`: Possible values: **`open-single`**, **`open-multi`**, **`closed-single`**, **`closed-multi`**, and **`semi-open`**. Currently the **`closed-single`** and **`open-single`** are supported, and the **`open-single`** schema is the recommended one.

- `network`: The network tag must be defined as **`fw-admin`**.
- `router`: Configured only for open- schemas, and must not be configured for other schemas.
  <cmx_virtual_router>: The name of the virtual router on a CMX, defined as **`om_sibb_vr`**.
- `target_ips`: Configured only for open- schemas, and must not be configured for other schemas. Define at least one IP address as an ARP target. These IP addresses are configured on the target virtual router.
  <ip_x>: Target IP addresses to be reached by the ARP monitor to check if the link is alive. Define at least one IP address in `target_ips`. The subnet mask is 255.255.255.0 by default.

```
ericsson:
  ...
  arp_setup:
    control:
      schema: open-single
      network: fw-admin
      router: om_sibb_vr
      target_ips:
        - 192.168.9.111
        - 192.168.9.112
  ...
```

Example 77 ARP Monitoring Example Setup

4.15.2 Traffic Network

4.15.2.1 Enable Link Redundancy Based on CFM in CEE Region

Connectivity Fault Management (CFM) link redundancy is only implemented for traffic networks. To enable CFM link redundancy, configure CFM settings based on the following:

```
ericsson:
  ...
  cfm_setup:
    traffic:
      enabled: true
      ccm_interval: 100
  ...
```

Example 78 CFM Link Redundancy

Valid `ccm_interval:` possible values are `3`, `10`, `100`, `1000`, `10000`, `60000`, `600000` ms. The recommended value is `100` ms.

To configure blades with CFM, see Section 4.15.2.2 on page 79.

Set `ccm_interval` to **`100`**, this makes sure that the failover is quick enough.

4.15.2.2 Configure CFM Roles

Link monitoring based on CFM is only applicable to BSP. To enable CFM on the traffic network of the blades, set the following:

```
ericsson:
  ...
      blade:
        -
          id: 2
          cfm_role: active
        ...
```

Example 79   Active role

```
ericsson:
  ...
      blade:
        -
          id: 2
          cfm_role: passive
        ...
```

Example 80   Passive role

Enable CFM on all blades in a BSP deployment. Configure three blades as active and ensure that all these blades are located in subrack 0. Configure all remaining blades as passive.

4.16 Reduced Footprint Monitoring Data Collection

`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 ommitted, the default value applies.

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

Example 81 Reduced Footprint

4.17 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 82 Zabbix CEE User

`zabbix_user_group`: String, the name of the user group.

`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 83 Zabbix CEE User Example

4.18 Initial Memory Amount of vCIC

Fuel handles the `virt` role as follows:

- First the VM is created with memory outside of hugepages.
- Next, the VMs are moved to hugepages during deployment.

8 GB is set for initial memory, which is consumed from the compute host memory solely for the creation of the VM. In some cases it is advised to set a different value for initial memory, for example:

- If the compute host has over 256 GB of memory, it is possible that the vCIC is unable to start because of memory allocation. In this case, lower the initial vCIC memory.

The initial vCIC memory amount can be changed by adding the `initial_vcic_memory` parameter to config.yaml. The value is in MiB.

```
ericsson
  ...
  initial_vcic_memory: 8192
```

Example 84   Configure Initial vCIC Memory

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