

Multi-Server System Dimensioning Guide, CEE 6

Cloud Execution Environment

CONFIGURATION MODEL

Copyright

© Ericsson AB 2016–2018. All rights reserved. No part of this document may be reproduced in any form without the written permission of the copyright owner.

Disclaimer

The contents of this document are subject to revision without notice due to continued progress in methodology, design and manufacturing. Ericsson shall have no liability for any error or damage of any kind resulting from the use of this document.

Trademark List

All trademarks mentioned herein are the property of their respective owners. These are shown in the document Trademark Information.



Contents

1	Introduction	1
1.1	Target Group	2
1.2	System Characteristics	2
2	Prerequisites	2
3	CEE System	3
3.1	System Configurations	3
4	Hardware Requirements	4
4.1	Network Configuration	6
4.1.1	Generic Network Requirements	6
4.2	CPU Configuration	8
4.2.1	Compute Host	9
4.2.2	Compute Host Examples on BSP Hardware Platform	12
4.2.3	ScaleIO Host with MDM/TB and SDS	14
4.3	RAM Configuration	15
4.3.1	Introduction	15
4.3.2	Compute Host	17
4.3.3	ScaleIO Host with MDM/TB and SDS	21
4.4	Storage Configuration	21
4.4.1	Local Storage Disk Space	21
4.4.2	Local Storage Disk Space on BSP Hardware Platforms	25
4.4.3	Disk Requirements for Atlas	28
4.4.4	Disk Requirements for Nova Snapshots	28
4.4.5	Disk Requirements for ScaleIO	29
4.4.6	Disk Requirements for Software RAID	29
4.4.7	Storage Performance Consideration	29
4.5	Kickstart Server	30
5	Characteristics	30
5.1	General System Limits	31
5.2	Orchestration Interface	33
5.3	Tenant Execution Environment	33
5.3.1	Performance	33
5.3.2	Resiliency	33
5.4	Network	34
5.4.1	Performance	34
5.4.2	Resiliency	39
5.4.3	Tenant Network Limitations	39



5.5	Storage	40
5.5.1	Limitations When Using Local Storage	40
5.5.2	Resiliency	41
5.5.3	Distributed Storage, ScaleIO	42
5.6	In-Service Performance	42
Reference List		44



1 Introduction

This document describes the characteristics of Cloud Execution Environment (CEE) to enable dimensioning and understanding the limitations of CEE. It also describes generic requirements on HW used for running CEE. The application can have additional requirements. The document provides general system dimensioning guidelines and does not describe a specific hardware type or model. Dedicated studies to create optimizations for a specific hardware model are outside of the scope of this document.

Storage is measured in gibibyte (GiB), tebibyte (TiB), and mebibyte (MiB) in this document.

1 GiB is equivalent to 1.074 GB.

The following words are used in this document with the meaning specified below:

CSS	The Cloud SDN Switch (CSS) is the virtual switch (vSwitch) component of CEE. It is based on the opensource project openvswitch (OVS) with functional extensions and performance enhancements. For more information, refer to the CEE Architecture Description, Reference [1], and the CSS documentation.
vNIC	A virtual network interface card (vNIC) provides connectivity between CSS and a Virtual Machine (VM). A configuration can provide several vNICs to a VM.
Interface	A network interface. Can be either a physical NIC (PHY) providing CSS with board external connectivity, or a virtual NIC connecting CSS to a VM.
PMD thread	CSS uses a Poll Mode Driver (PMD) technique that continuously polls incoming packets from the NICs, that is, interrupts are not used. To be able to reliably handle all incoming packets, the NIC queues are continuously polled for packets to be handled. This software is executing in one or more threads that are called PMD threads. The execution environment for the PMD threads resides in the Linux user space and is thus isolated from the Linux scheduler to be able to reliably handle a high sustained packet flow without interrupts or delays caused by being scheduled out.
Compute host	A physical server running the Nova compute daemon.



vCIC host	<p>A compute host with Virtual Cloud Infrastructure Controller (vCIC). There are three vCIC hosts in Multi-Server deployments of CEE.</p> <p>Note: vCIC and vFuel can run on the same compute host.</p>
vFuel host	<p>A compute host with vFuel</p> <p>Note: vCIC and vFuel can run on the same compute host.</p>
ScaleIO host	<p>A physical server dedicated for running the server related components of ScaleIO like MDM/TB, SDS, ScaleIO Gateway or LIA</p> <p>Note: The set of ScaleIO components installed on a ScaleIO host depends on the configuration of the host according to the role of that host in the ScaleIO distributed storage.</p>
Core	<p>A physical core of a processor.</p>

1.1 Target Group

Cloud Infrastructure providers and application designers.

1.2 System Characteristics

For information on the features of CEE, refer to the [CEE Technical Description](#).

For the characteristics of the used hardware, refer to the product documentation of the used hardware.

2 Prerequisites

The following must be specified as prerequisites for using the values provided in this dimensioning guide:

- The maximum amount of physical servers included in the CEE region. Amount of physical servers means the sum of the number of compute hosts and ScaleIO hosts.
- The number of physical servers (compute hosts plus ScaleIO hosts) to be added in a single region expansion procedure.



3 CEE System

This section describes CEE system configurations and capabilities.

3.1 System Configurations

CEE is a scalable system used with hardware products from different vendors. CEE can be used on a single physical server, called a single server configuration, described in the [Single Server System Dimensioning Guide, CEE 6](#) document. CEE can also be used on a set of servers, called multi-server configuration, described in this document.

Multi-server deployments can use different optional configurations. The following matrix lists the possible configurations:

Configuration Option	Conf 1	Conf 2	Conf 3	Conf 4	Conf 5
CEE on HDS and Ericsson Cloud SDN Controller (CSC) as networking backend			X	X	
Modular Layer 2 (ML2) driver as networking backend for direct physical switch management		X			X
ScaleIO as Cinder backend				X	X

4 Hardware Requirements

CEE is a software product which can run on hardware infrastructures that comply with the generic requirements detailed in this section. Different hardware can imply different CEE system characteristics.

A general diagram of the server, switching, and optional distributed storage components of the hardware environment is shown in Figure 1:

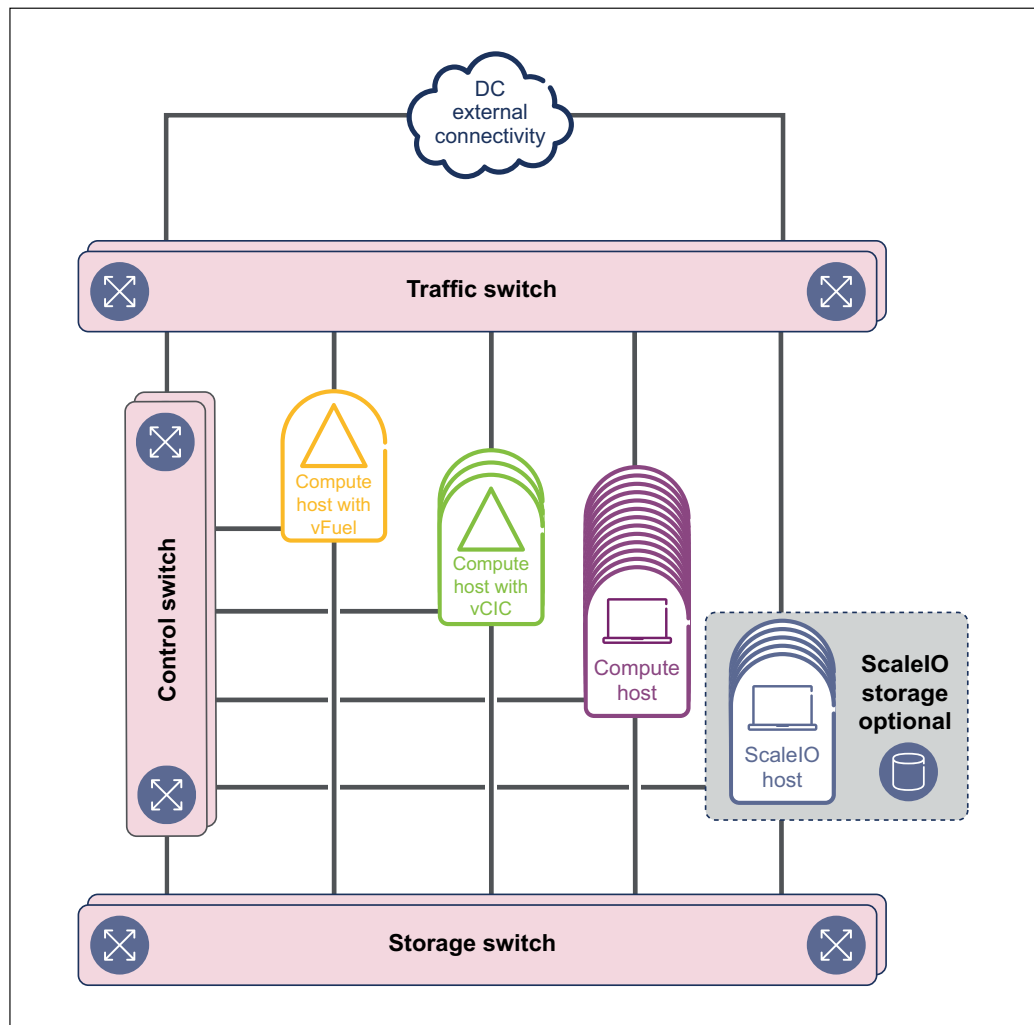


Figure 1 CEE Hardware Environment with Optional Distributed Storage

Figure 2 shows the hardware components of CEE installed on the Ericsson Hyperscale Datacenter System (HDS):

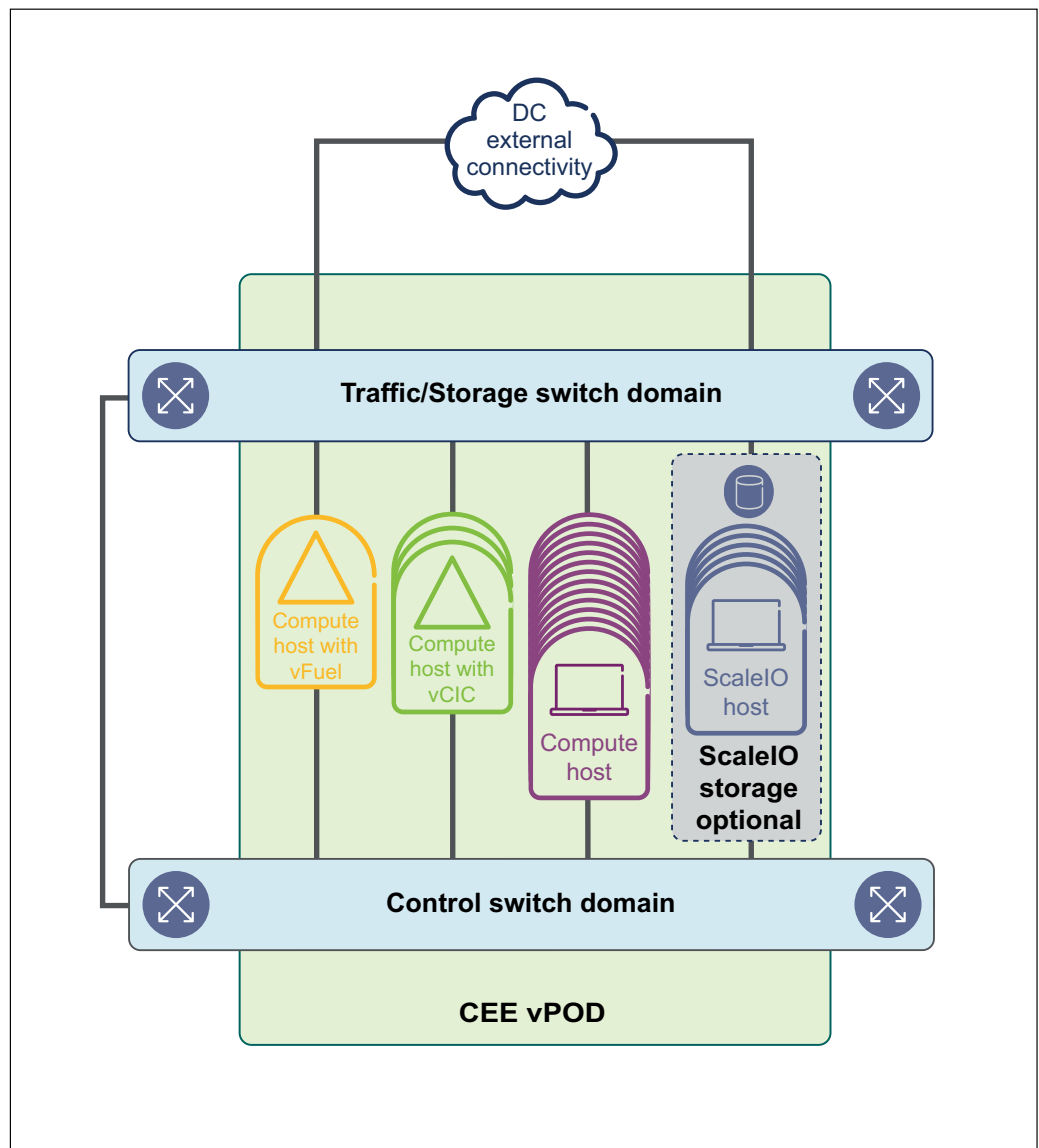


Figure 2 CEE on HDS Hardware Environment

Figure 3 shows the hardware components of CEE installed on the BSP hardware platforms.

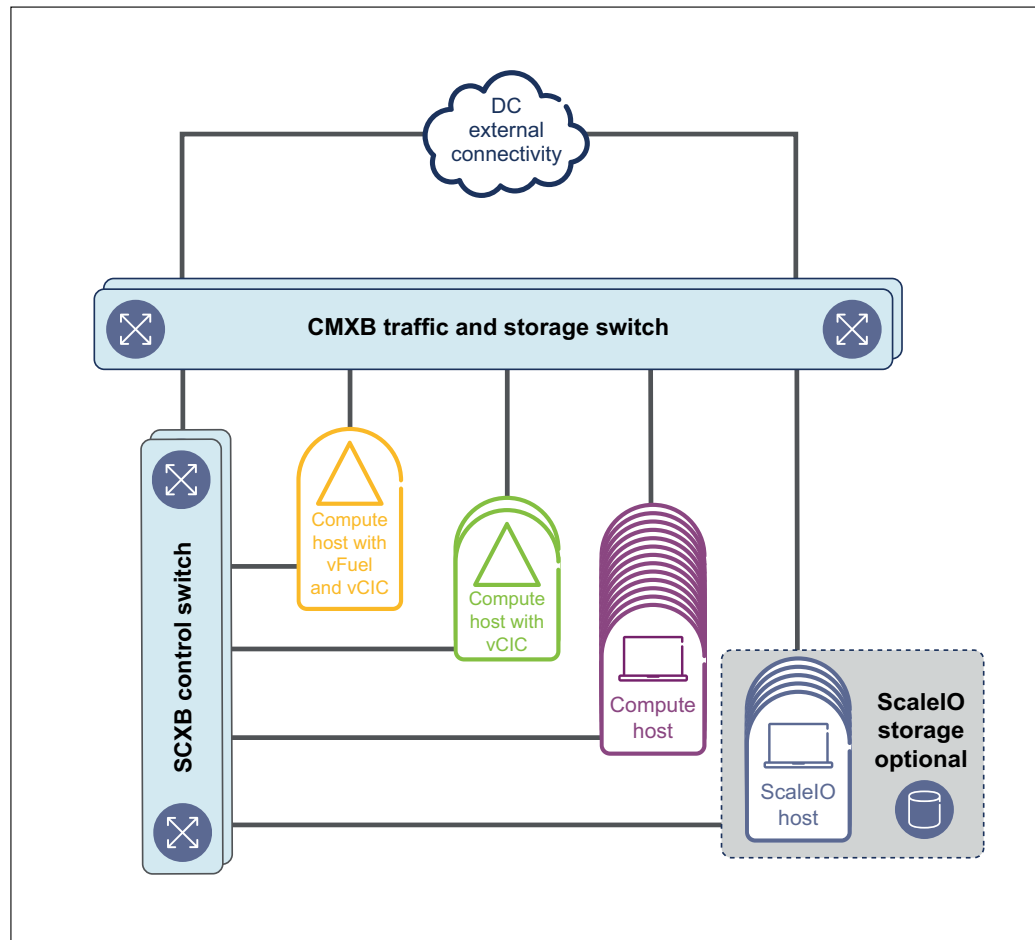


Figure 3 CEE HW Environment with BSP Hardware

4.1 Network Configuration

4.1.1 Generic Network Requirements

CEE provides a degree of flexibility among different hardware configurations. However, the hardware configurations of the compute nodes and hardware switches must comply with the following generic requirements:

- Two dedicated Ethernet ports with nominal bandwidth of 1 Gb/s or greater to be used for CEE internal control traffic



Note: Some operations like `openstack server create` or `openstack volume create --image` consume most of the 1 Gb/s bandwidth due to image data transfer from vCIC to compute or between vCICs. Frequent tenant VM image data transfer or transfer of big images might lead to time periods when the control network is congested. The congestion of the control network of CEE affects the number of operations per unit of time the VIM can achieve. When operation rate performance is of concern, it is recommended to study network bandwidth occupancy with care. If 1Gb/s is not enough, use a 10Gb/s NIC for the CEE management network at least on the compute nodes hosting the vCICs. If the servers or switches do not have 10Gb/s ports for the control traffic, it is still possible to configure Glance and Swift endpoints on the physical NICs used for storage network (if present) with a dedicated VLAN and subnet. With this solution the image data transfer will use the 10 Gb/s NICs dedicated to storage. For configuration details, refer to [Configuration File Guide](#). Consider that in this case the control network of CEE is extended to an additional physical domain.

- Two dedicated Ethernet ports to be used for application traffic and VIM O&M interfaces, with the following requirements:
 - Nominal bandwidth of 10 Gb/s or greater
 - Based on NICs that support Data Plane Development Kit (DPDK)
- Two dedicated Ethernet ports with nominal bandwidth of 10 Gb/s or greater, to be used for storage network traffic

Note: See Section 4.1.1.2 on page 8 for more information. Refer to [Configuration File Guide](#) in order to install CEE without NICs for storage traffic on compute nodes without vCIC.

- Optionally, additional dedicated Ethernet ports with support of DPDK and Single-Root Input/Output Virtualization (SR-IOV) for high throughput application traffic using SR-IOV
- Optionally, additional dedicated Ethernet ports for high throughput application traffic using PCI passthrough
- A non-blocking switching infrastructure
- Consistency of MTU settings across all network components used for the CEE region

4.1.1.1

Network Requirements for Distributed Storage, ScaleIO Nodes

The nodes dedicated to ScaleIO must comply with the following requirements:

- Two dedicated Ethernet ports with nominal bandwidth of 1 Gb/s or greater to be used for ScaleIO management traffic



- Minimum two dedicated Ethernet ports to be used for storage traffic. For an optimal setup, use two dedicated Ethernet ports for frontend storage traffic, plus two dedicated Ethernet ports for backend storage traffic.
- The Ethernet switch supports the required bandwidth between network nodes.
- The available network bandwidth and latency are acceptable among all the nodes, according to the application demands.
- MTU settings are consistent across all servers and switches.
- The following TCP ports are not used by any other application, and are open in the local firewall of the server:
 - Meta Data Manager (MDM): 6611 and 9011
 - ScaleIO Data Server (SDS): 7072, for multiple SDS: 7073-7076
 - ScaleIO Gateway (including REST Gateway, Installation Manager, and SNMP trap sender): 80 and 443
 - Light Installation Agent (LIA): 9099

Parameters and configuration in ScaleIO that affect dimensioning:

- SDS network limits can be set to avoid overloading the storage network with huge amount of rebuild or rebalance traffic. Refer to the CLI and REST API command reference section in the [Dell EMC ScaleIO Version 2.x User Guide](#).
- For further details on network setup, refer to the [Networks](#) section in the [Configuration File Guide](#).

4.1.1.2 Network Configuration Without Storage Switching Domain

CEE on multi-server platforms supports configurations without storage interfaces (storage switching domain) defined on compute servers that are not hosting vCICs. This configuration is applicable to the whole CEE region. As a result, the free storage interfaces can be used for other purposes. However, only local storage can be used on the compute servers with this configuration, as it is not possible to attach remote storage volumes to the tenant VMs. For more information and configuration details, refer to the Host Networking section in [Configuration File Guide](#).

4.2 CPU Configuration

Refer to the [Configuration File Guide](#) for more information about the configuration procedure.

The number of available CPU IDs depends on the CPU model.



See the relevant section:

- For compute host, see Section 4.2.1 on page 9.
- For compute host CPU configuration examples specific to BSP, see Section 4.2.2 on page 12.
- For ScaleIO host with MDM/TB and SDS, see Section 4.2.3 on page 14.

Note: For CPU core allocation to vFuel on the kickstart server, see Section 4.5 on page 29.

4.2.1 Compute Host

The vCIC CPU load has a complex dependency on the number of physical servers (compute hosts plus ScaleIO hosts), the number of tenant VMs, and the orchestration traffic profile.

The dimensioning values shown in this section are based on the assumption that hyperthreading is enabled on the compute hosts.

The tables in this section show the minimum number of cores required for the CPU owners in a host processor depending on the used solution:

- For CEE with tightly integrated Software Defined Networking (SDN), see Table 1.
- For CEE without tightly integrated SDN, see Table 2.

Use Table 1 and Table 2 for the following cases:

- Compute host without vCIC and vFuel
- Compute host with vCIC
- Compute host with vFuel
- Compute host with vCIC and vFuel

Table 1 Minimum Number of Cores Required in a Compute Host for CEE with Tightly Integrated SDN

CPU Owner	Minimum Number of Cores Required
Mandatory Components on Each Compute Host:	
Host Operating System (OS)	2 ⁽¹⁾⁽²⁾
CSS ⁽³⁾	See Section 5.4.1 on page 34 for recommendations.
CSS control process ⁽⁴⁾	0
Optional Components:	



CPU Owner	Minimum Number of Cores Required	
vCIC ⁽⁵⁾⁽⁶⁾	Up to 16 physical servers ⁽⁷⁾	14 ⁽⁸⁾
	17–48 physical servers ⁽⁷⁾	16 ⁽⁸⁾
	49–80 physical servers ⁽⁷⁾	18 ⁽⁸⁾
vFuel	To add 1–16 new physical servers ⁽⁹⁾ in one expansion procedure	1 ⁽¹⁰⁾
	To add 17–48 new physical servers ⁽⁹⁾ in one expansion procedure:	4 ⁽¹⁰⁾
Atlas VM ⁽¹¹⁾	1	
Tenant VM:		
Tenant VMs	The remaining cores	

(1) It can be 1 in some use-cases as explained in Section 4.2.1.2 on page 12.

(2) If the compute host includes ScaleIO Data Client (SDC), the SDC runs 2 threads on the host OS CPUs, which increases the load on the CPU depending on the amount of block IOs the VMs execute. Add one additional core for host OS when running SDC. For some workloads it is possible that the additional core is not needed. The recommendation is to measure the host OS CPU utilization in the actual deployment scenario before deciding to not allocate the additional core to host OS.

(3) See Section 5.4.1.2 on page 34 for more information.

(4) The process does not get a CPU for its exclusive use. A configuration parameter specifies one of the host OS CPUs to be used by the CSS control process.

(5) No tenant VM deployment is possible on the physical servers hosting vCICs.

(6) If ScaleIO is used as Swift storage backend, the SDC runs 2 threads on the vCIC CPUs, which increases the load on the CPU depending on the amount of block IOs the VM execute. Add one additional core for vCIC (2 vCPUs) when running SDC in vCIC. For some workloads it is possible that the additional core is not needed. The recommendation is to measure the total CPU utilization of vCIC in the actual deployment scenario with the actual rate of image upload, download and image copy to volumes operations before deciding to not allocate the additional core for the vCIC.

(7) To select the correct range, use the planned maximum number of physical servers (compute hosts plus ScaleIO hosts) including future expansions.

(8) If vCIC is not used on the server, this amount is allocated to the tenant VMs.

(9) Sum of compute hosts and ScaleIO hosts

(10) If vFuel is not used on the server, this amount is allocated to the tenant VMs.

(11) Atlas configurations with more than 2 allocated vCPUs have not been tested.

The CPU allocation values in Table 2 are examples from a deployment where 12 VMs per host with 10 vNICs per VM (excluding the hosts running vCIC) are deployed in a sequential manner with no pause between two consecutive instantiations.

Table 2 Minimum Number of Cores Required in a Compute Host, for CEE without Tightly Integrated SDN

CPU Owner	Minimum Number of Cores Required
Mandatory Components on Each Compute Host:	
Host Operating System (OS)	2 ⁽¹⁾⁽²⁾



CPU Owner	Minimum Number of Cores Required	
CSS ⁽³⁾	See Section 5.4.1 on page 34 for recommendations.	
CSS control process ⁽⁴⁾	0	
Optional Components:		
vCIC ⁽⁵⁾⁽⁶⁾	Up to 16 physical servers ⁽⁷⁾	6 ⁽⁸⁾
	17–48 physical servers ⁽⁷⁾	8 ⁽⁸⁾
	49–80 physical servers ⁽⁷⁾	10 ⁽⁸⁾
vFuel	To add 1–16 new physical servers ⁽⁹⁾ in one expansion procedure	1 ⁽¹⁰⁾
	To add 17–48 new physical servers ⁽⁹⁾ in one expansion procedure	4 ⁽¹⁰⁾
Atlas VM ⁽¹¹⁾	1	
Tenant VM:		
Tenant VMs	The remaining cores	

(1) It can be 1 in some use-cases as explained in Section 4.2.1.2 on page 12.

(2) If the compute host includes ScaleIO Data Client (SDC), the SDC runs 2 threads on the host OS CPUs, which increases the load on the CPU depending on the amount of block IOs the VMs execute. Add one additional core for host OS when running SDC. For some workloads it is possible that the additional core is not needed. The recommendation is to measure the host OS CPU utilization in the actual deployment scenario before deciding to not allocate the additional core to host OS.

(3) See Section 5.4.1.2 on page 34 for more information.

(4) The process does not get a CPU for its exclusive use. A configuration parameter specifies one of the host OS CPUs to be used by the CSS control process.

(5) The vCIC can be allocated on cores that are shared with the application. In such cases, it must be ensured that the application sharing resources with the vCIC does not exhaust the vCIC resources. Refer to the [Configuration File Guide](#) to configure accordingly.

(6) If ScaleIO is used as Swift storage backend, the SDC runs 2 threads on the vCIC CPUs, which increases the load on the CPU depending on the amount of block IOs the VM execute. Add one additional core for vCIC (2 vCPUs) when running SDC in vCIC. For some workloads it is possible that the additional core is not needed. The recommendation is to measure the total CPU utilization of vCIC in the actual deployment scenario with the actual rate of image upload, download and image copy to volumes operations before deciding to not allocate the additional core for the vCIC.

(7) To select the correct range, use the planned maximum number of physical servers (compute hosts plus ScaleIO hosts) including future expansions.

(8) If vCIC is not used on the server, this amount is allocated to the tenant VMs.

(9) Sum of compute hosts and ScaleIO hosts

(10) If vFuel is not used on the server, this amount is allocated to the tenant VMs.

(11) Atlas configurations with more than 2 allocated vCPUs have not been tested.

If vCIC, vFuel, or ScaleIO Data Client (SDC) are not used on the server, their CPU cores are allocated to the tenant VMs.



4.2.1.1 Host OS Cores

The host OS, monitoring processes, OpenStack agents and ScaleIO client use the CPUs not reserved for the tenant VMs.

The number and IDs of the CPUs assigned to the host OS is configurable. For more information, refer to the [Configuration File Guide](#).

4.2.1.2 Allocating One Core to the Host OS

The recommended CPU allocation to the host OS is two cores for each compute host, but for some use-cases, where a single or very few VMs are used on each compute host, it is possible to reduce the host OS allocation to one core. Having a single core allocated for the host OS can impact the characteristics of the system. For example, actions such as starting, stopping, and migrating VMs can become slower, and it can also impact the performance of other VMs on the same host. Before using a system with a single core allocated to the Host OS in a production environment, sufficient testing must be performed. Beside other checks, this testing must include the capacity and performance test of the VM during the time of performing life cycle management operations for other VMs running on the same host. While performing such a test, the processor load for the host OS must be monitored carefully for deviation from a steady state.

4.2.2 Compute Host Examples on BSP Hardware Platform

The examples in this section describes the CPU dimensioning for a CEE region working with 16 physical servers.

Table 3 and Figure 4 give an example on CPU allocation for CSS, vCIC, and vFuel for CEE on GEP5 or GEP7L.

Table 4 and Figure 5 give an example on CPU allocation for VMs, CSS, vCIC, and vFuel for CEE on GEP7.

If vCIC and vFuel are not used on the server, their CPUs are allocated for the tenant VMs. vCIC and vFuel are by default pinned to dedicated CPUs. The pinning is achieved using automatic configuration, as described in [Configuration File Guide](#).

Table 3 Example of CPU Allocation on GEP5 for CSS, vCIC, and vFuel in a Compute Host

CPU Owner	Allocated CPU ID
Tenant VM	-
CSS ⁽¹⁾	1,11
CSS control process ⁽²⁾	0 ⁽²⁾
vCIC	4,14,5,15,6,16,7,17,8,18,9,19



CPU Owner	Allocated CPU ID
vFuel	3,13
Host OS ⁽³⁾⁽⁴⁾	0,10,2,12

(1) In this example, CSS is configured with normal-perf. Refer to the **Configuration File Guide** for more details.

(2) The process does not get a CPU for its exclusive use. A configuration parameter specifies one of the host OS CPUs to be used by the CSS control process.

(3) In some use-cases only one core is allocated to the host OS as explained in Section 4.2.1.2 on page 12.

(4) If the compute host includes ScaleIO Data Client (SDC), the SDC runs 2 threads on the host OS CPUs, which increases the load on the CPU depending on the amount of block IOs the VMs execute. Add one additional core for host OS when running SDC. For some workloads it is possible that the additional core is not needed. The recommendation is to measure the host OS CPU utilization in the actual deployment scenario before deciding to not allocate the additional core to host OS.

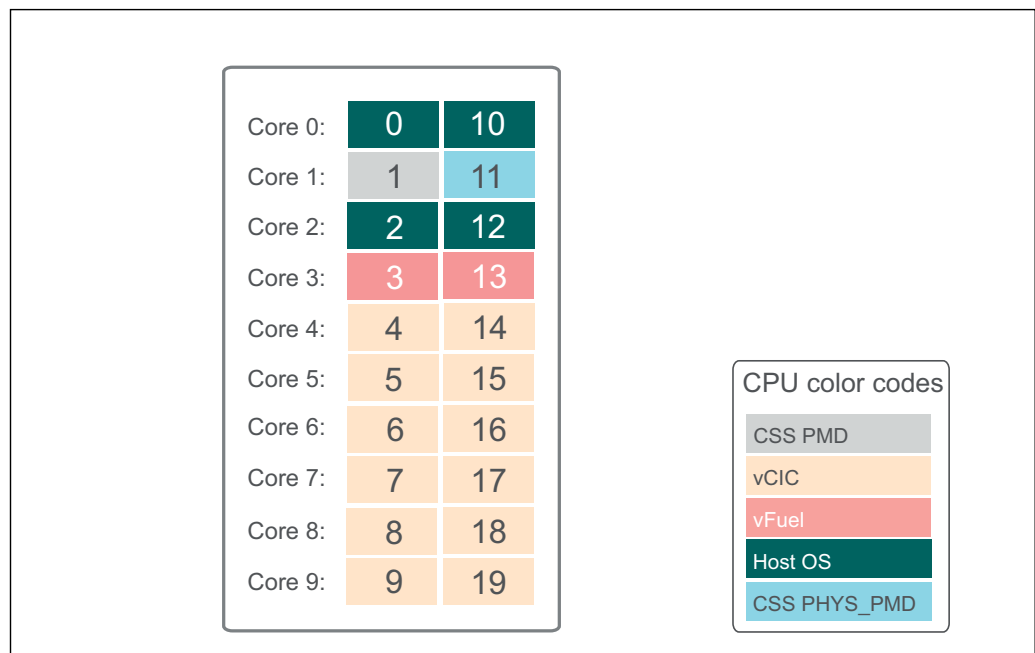


Figure 4 Example of CPU Allocation of the Respective Resource Owner on GEP5 for CSS, vCIC, and vFuel in a Compute Host

Table 4 Example of CPU Allocation on GEP7 for VMs, CSS, vCIC, and vFuel in a Compute Host

CPU Owner	Allocated CPU ID
Tenant VM	10,24, 11,25, 12,26, 13,27
CSS ⁽¹⁾	1,15
CSS control process ⁽²⁾	0 ⁽²⁾
vCIC ⁽³⁾	4,18, 5,19, 6,20, 7,21, 8,22, 9,23



CPU Owner	Allocated CPU ID
vFuel	3,17
Host OS ⁽⁴⁾⁽⁵⁾	0,14, 2,16

(1) In this example, CSS is configured with normal-perf. Refer to the [Configuration File Guide](#) for more details.

(2) The process does not get a CPU for its exclusive use. A configuration parameter specifies one of the host OS CPUs to be used by the CSS control process.

(3) The vCIC can be allocated on cores that are shared with the application. In such cases, it must be ensured that the application sharing resources with the vCIC does not exhaust the vCIC resources. Refer to the [Configuration File Guide](#) to configure accordingly.

(4) In some use-cases only one core is allocated to the host OS as explained in Section 4.2.1.2 on page 12.

(5) If the compute host includes ScaleIO Data Client (SDC), the SDC runs 2 threads on the host OS CPUs, which increases the load on the CPU depending on the amount of block IOs the VMs execute. Add one additional core for host OS when running SDC. For some workloads it is possible that the additional core is not needed. The recommendation is to measure the host OS CPU utilization in the actual deployment scenario before deciding to not allocate the additional core to host OS.

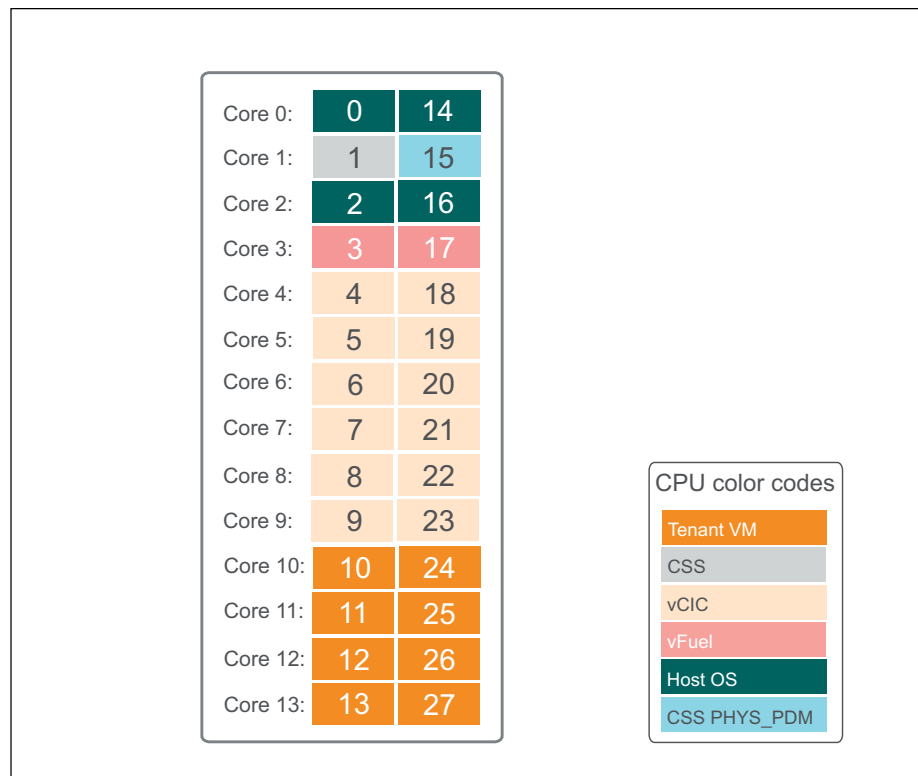


Figure 5 Example of CPU Allocation of the Respective Resource Owner on GEP7 for VMs, CSS, vCIC, and vFuel in a Compute Host

4.2.3 ScaleIO Host with MDM/TB and SDS

Table 5 shows the minimum number of cores required for the ScaleIO host.



Table 5 Minimum Number of Cores Required in a ScaleIO Host

CPU Owner	Minimum Number of Cores Required
Host OS	2
Meta Data Manager/Tie-Breaker (MDM/TB)	1
SDS	2–4

4.3 RAM Configuration

This section describes the optimal RAM configuration for the CEE.

Refer to the [Configuration File Guide](#) for more information about the configuration procedure.

Each Neutron network created consumes RAM in the vCIC, and it influences the maximum number of virtual tenant networks. See Section 5.4.3 on page 39 for more information.

The minimum required RAM size of the compute host is 64 GiB. The recommended RAM size is 128 GiB or more.

Running vCIC, vFuel, or both on the server modifies the optimal memory allocation.

More memory must be allocated to the vCIC in a system with tightly integrated SDN.

See the relevant sub-section:

- For general information about RAM configuration in CEE, see Section 4.3.1 on page 15.
- For compute host, see Section 4.3.2 on page 17.
- For ScaleIO host with MDM/TB and SDS, see Section 4.3.3 on page 21.

Note: For memory allocation to vFuel on the kickstart server, see Section 4.5 on page 29.

4.3.1 Introduction

This section provides general information about RAM configuration in CEE.

Note: Some considerations in this section are not applicable to systems with single NUMA node.

A certain amount of memory is reserved for booting a server, for example, BIOS-reserved memory, page tables, and memory-mapped devices. This reserved memory is called unmanaged. The total memory the system can use is called managed, and it is the difference between the nominal physical memory and the



unmanaged part. The sum of managed and unmanaged parts is relevant for planning purposes, for example, to plan the amount of physical memory to be installed on the servers. The unmanaged part cannot be changed. The managed memory allocated to the host OS on the compute nodes is 6 GiB by default. More managed memory can be reserved for the host OS by setting the relevant configuration parameter. The needed amount of memory depends on the values of a number of other parameters as described below.

The memory used for the VMs is allocated to hugepages. This is the memory visible from the inside of the VMs. The 1 GiB hugepages are referred to as **Tenant VM** in the RAM reservation tables of the upcoming sections. In addition to the 1 GiB hugepages, the VMs need memory allocated from the host OS. This memory is used, for example, to emulate devices used by the VM. It is hard to predict the amount of host OS memory used by the emulator since, for example, it depends on the type and number of the used devices. A small VM consumes less than 100 MiB, while it can grow to several hundred MiB in specific cases. About 300 MiB host OS memory would be enough for each VM but we must double it and calculate with 600 MiB as explained below.

In a system using the NUMA architecture, the NUMA location of VMs must be considered. The available memory, that is, the hugepages and the Host OS memory, are evenly distributed between the NUMA nodes. By design, OpenStack Nova allocates VMs on the first NUMA node that fits the VM. Apart from the VMs running on both NUMA nodes, the VMs allocate memory from the NUMA node on which they are running. In a worst case scenario where all VMs are allocated on the same NUMA node, all the memory for the VMs will be allocated from the same NUMA node. In such a scenario most of the memory on the other NUMA node will be unused, and half of the memory on the compute node will be free. To be on the safe side in a dual socket system, the 300 MiB host OS memory per VM must be doubled to cover the case where all VMs are allocated on the same NUMA node.

The processes running on the compute host use 4 KiB memory pages. In addition, CSS also uses 2 MiB hugepages, and QEMU processes of each tenant VM deployed with hugepages use additional 1 GiB pages. Hugepages for tenant VMs are always reserved symmetrically on each NUMA node, if the required number of hugepages is even. If the required number is odd, one more page is reserved on NUMA node 0.

In multi-server configurations the infrastructure VMs (vFuel and vCIC) are set up to use CPU pinning and 1 GiB hugepages. Memory and CPU allocation for the infrastructure VMs is configurable at CEE installation.

The system, as a general rule, allocates memory pages that are local to the NUMA node.

For example, if the vCIC requires 20 cores in total, 10 per NUMA node and 60 huge pages, the system will reserve 30 huge pages per NUMA node.

If 30 GiB are required for vCIC with all CPUs on NUMA node 0 and there are no 30×1 GiB huge pages available on NUMA 0, CEE installation will fail..

For details regarding memory allocation in `config.yaml`, refer to the [Configuration File Guide](#).



Note: In order to allocate as little memory for the host OS as possible, memory profiling of the host OS for the specific scenario is recommended.

4.3.2 Compute Host

This section specifies the volumes allocated to the resource owners in a compute host.

Use the tables of this section for the following cases:

- Compute host without vCIC and vFuel
- Compute host with vCIC
- Compute host with vFuel
- Compute host with vCIC and vFuel

More memory must be allocated to the vCIC in a system with tightly integrated SDN:

- Table 6 specifies the volumes to be allocated to the resource owners in a system with tightly integrated SDN.
- Table 7 specifies the volumes to be allocated to the resource owners in a system without tightly integrated SDN.

Table 6 Minimum Required Memory Allocation for System with Tightly Integrated SDN

RAM Resource Owner	Hugepage size (MiB)	Number of Hugepages (Count)		Total Size (GiB)
Mandatory Components on Each Compute Host:				
CSS	2	Up to MTU=2140 bytes	1024	2
		Up to MTU=4400 bytes	2048	4
		Up to MTU=6700 bytes	3072	6
		Up to MTU=9000 bytes	4096	8



RAM Resource Owner	Hugepage size (MiB)	Number of Hugepages (Count)		Total Size (GiB)	
Unmanaged memory ⁽¹⁾				Up to 64 GiB nominal physical memory	2 ⁽¹⁾
				Up to 128 GiB nominal physical memory	3 ⁽¹⁾
				Up to 256 GiB nominal physical memory	5 ⁽¹⁾
				Up to 384 GiB nominal physical memory	7 ⁽¹⁾
Host OS ⁽²⁾				X (Integer) ⁽²⁾	
Optional Components:					
vCIC ⁽³⁾	1024	Up to 16 physical servers ⁽⁴⁾	46	46	
		17–48 physical servers ⁽⁴⁾	72	72	
		49–80 physical servers ⁽⁴⁾	82	82	
vFuel	1024	To add 1–16 new physical servers ⁽⁵⁾ in one expansion procedure	3	3	
		To add 17–48 new physical servers ⁽⁵⁾ in one expansion procedure	8	8	



RAM Resource Owner	Hugepage size (MiB)	Number of Hugepages (Count)	Total Size (GiB)
Tenant VM:			
Tenant VM ⁽⁶⁾	1024	The same number as the rest amount of memory in GiB ⁽⁷⁾	The rest amount of memory

(1) For more information, see Section 4.3.1 on page 15.

(2) For more information, see Section 4.3.2.1 on page 20.

(3) No tenant VM deployment is possible on the physical servers hosting vCICs.

(4) To select the correct range, use the planned maximum number of physical servers (compute hosts plus ScaleIO hosts) including future expansions.

(5) Sum of compute hosts and ScaleIO hosts.

(6) If Atlas is used, it runs at one of the hosts, and it uses 4 GiB of the RAM configured for the tenant VMs.

(7) The rest amount of memory (GiB) divided by the 1 GiB hugepage size.

Table 7 Minimum Required Memory Allocation for System without Tightly Integrated SDN

RAM Resource Owner	Hugepage size (MiB)	Number of Hugepages (Count)	Total Size (GiB)
Mandatory Components on Each Compute Host:			
CSS	2	Up to MTU=2140 bytes	2
		Up to MTU=4400 bytes	4
		Up to MTU=6700 bytes	6
		Up to MTU=9000 bytes	8
Unmanaged memory ⁽¹⁾			Up to 64 GiB nominal physical memory
			2 ⁽¹⁾
			Up to 128 GiB nominal physical memory
			3 ⁽¹⁾
			Up to 256 GiB nominal physical memory
			5 ⁽¹⁾
			Up to 384 GiB nominal physical memory
			7 ⁽¹⁾



RAM Resource Owner	Hugepage size (MiB)	Number of Hugepages (Count)		Total Size (GiB)
Host OS ⁽²⁾				X (Integer) ⁽²⁾
Optional Components:				
vCIC	1024	Up to 16 physical servers ⁽³⁾	30	30
		17–48 physical servers ⁽³⁾	40	40
		49–80 physical servers ⁽³⁾	50	50
vFuel	1024	To add 1–16 new physical servers ⁽⁴⁾ in one expansion procedure	3	3
		To add 17–48 new physical servers ⁽⁴⁾ in one expansion procedure	8	8
Tenant VM:				
Tenant VM ⁽⁵⁾	1024	The same number as the remaining amount of memory in GiB ⁽⁶⁾		The remaining amount of memory

(1) For more information, see Section 4.3.1 on page 15.

(2) For more information, see Section 4.3.2.1 on page 20.

(3) To select the correct range, use the planned maximum number of physical servers (compute hosts plus ScaleIO hosts) including future expansions.

(4) Sum of compute hosts and ScaleIO hosts.

(5) If Atlas is used, it runs at one of the hosts, and it uses 4 GiB of the RAM configured for the tenant VMs.

(6) The remaining amount of memory (GiB) divided by the 1 GiB hugepage size

4.3.2.1 RAM Required for Host OS

The variable X corresponding to the minimum size of RAM in GiBs allocated to the Host OS can be calculated using the following formula:

Note: The formulae described in this section are used to calculate the minimum amount of RAM required for the host OS. Depending on the configuration, more memory can be reserved for the host OS.

If ScaleIO is installed, SDC uses 0.05 GiB of the RAM configured for the host OS.

- For systems with a single NUMA node:

$$X = 5 + n * 0.3 \text{ rounded up to the next integer,}$$
where n is the maximum number of VMs planned on the compute host.

For example, if the compute host is to host seven VMs, $5 + 7 * 0.3 = 7.1$ is rounded up to the next integer, therefore $X = 8$



- For systems with two NUMA nodes:

$$X = 6 + n * 0.6 \text{ rounded up to the next odd integer,}$$
 where n is the maximum number of VMs planned on the compute host.

For example, if the compute host is to host 10 VMs, $6 + 10 * 0.6 = 12$ is rounded up to the next odd integer, therefore $X = 13$

4.3.3 ScaleIO Host with MDM/TB and SDS

Table 8 specifies the minimum amount of memory required by ScaleIO services.

Table 8 Memory Requirement for Services in a ScaleIO Host

RAM Resource Owner	Total Size (GiB)
MDM/TB	0.5
SDS	0.5

4.4 Storage Configuration

This section provides information on centralized, local and distributed storage implementations, and disk requirements.

Note: The Glance storage must always have empty space left for storing images temporarily. The minimum required space for temporary storage is the size of the largest image. This minimum requirement is enough to start one VM from volume at a time. To achieve a decent performance even if more VMs are started from volume at the same time, the Glance storage capacity must be dimensioned carefully.

For local storage disk space requirements on BSP hardware platforms, see Section 4.4.2 on page 25.

4.4.1 Local Storage Disk Space

Note: This section, apart from the database partition calculation for vCIC in Section 4.4.1.1 on page 23, is not applicable for CEE deployments on BSP hardware platforms. See Section 4.4.2 on page 25 for BSP-specific information and requirements for local storage disk space.

This section lists requirements on disk space in Table 9 and Table 10.

Table 9 Compute Host Disk Allocation

Use	Compute Host with vCIC	Compute Host without vCIC	Note
Root partition (host OS)	50 GiB	50 GiB	



Use	Compute Host with vCIC	Compute Host without vCIC	Note
Logs and core/crash dumps	40 GiB	40 GiB	When 40 GiB is allocated to log partition, 10 GiB is allocated to logs and 30 GiB is allocated to core/crash dumps.
vCIC total	Sum of the relevant items in Table 10	-	
Host total without vFuel and Atlas	Sum of the items above	90 GiB	
vFuel	80 GiB	80 GiB	<p>vFuel can be run on any compute host and it uses disk from the ephemeral storage.</p> <p>Fuel Snapshot: If Fuel snapshot is planned to be used, additional disk space must be configured accordingly, on top of the 80 GiB. The recommended disk space for vFuel with Fuel snapshot is 80+10=90 GiB. This Fuel feature allows collecting Fuel-specific diagnostic information and logs from the compute nodes. By default, it tries to fetch logs from all compute nodes. It can be configured to collect logs from a subset of compute nodes or even from no compute nodes.</p>
Atlas	>20 GiB	>20 GiB	<p>Atlas can be run on any compute host and it uses disk from the ephemeral storage. The disk for Atlas is allocated by the CEE.</p> <p>For more details, refer to Section 4.4.3 on page 28.</p>
Remaining storage is for ephemeral storage for tenant VMs.	Dimensioned depending on application need.	Dimensioned depending on application need.	<p>Valid for boot from image.</p> <p>Calculated from total disk reduced by used space.</p>

Table 10 vCIC Disk Allocation

Use	vCIC	Note
Root partition (host OS)	50 GiB	
Logs and crash dumps	40 GiB	



Use	vCIC	Note
Database for OpenStack and Zabbix (MySQL)	Use the formula provided in Section 4.4.1.1 on page 23.	The size must be adjusted to the storage needs of Zabbix. The database size depends on the number of physical servers ⁽¹⁾ .
Database for Ceilometer (MongoDB)	Use the formula provided in Section 4.4.1.1 on page 23.	The size needs to be adjusted to the storage needs of Ceilometer.
Glance repository in Swift	> 40 GiB ⁽²⁾	<ul style="list-style-type: none"> The size needs to be adjusted depending on the amount and size of images planned to be stored in Glance. Atlas data backup is stored in the same partition. The backup is typically below 1 GiB.

(1) Sum of compute hosts and ScaleIO hosts

(2) Dimensioned depending on application need.

4.4.1.1 Database Partition Calculation for vCIC

This section specifies the formulas for calculating the disk partition size for database in the local storage.

The formulas are valid for a CEE region with default counters configuration.

Use the relevant formula depending on the used solution.

OpenStack and Zabbix (MySQL):

Table 11 Formula for OpenStack and Zabbix (MySQL)

Formula for Partition Size	Variable
$0.7 \times N + 40$ GiB	N is the number of physical servers, that is, the sum of compute hosts and ScaleIO hosts.

Examples for OpenStack and Zabbix (MySQL):

Planned Number of Physical Servers ⁽¹⁾ in the CEE Region	Database Partition Size, vCIC
16 physical servers ⁽¹⁾	51 GiB
48 physical servers ⁽¹⁾	74 GiB
80 physical servers ⁽¹⁾	96 GiB

(1) Sum of compute hosts and ScaleIO hosts

**Formula for Ceilometer (MongoDB):**

The MongoDB storage partition size can be dimensioned according to the formula:

Actual file size + Journal size + Local database size + 10 GiB

The formula consists of the following:

— Actual file size: $3 \times (\text{Actual data size} + \text{Index size})$

- Actual data size: $([\sum_i R_i \times N_i / P_i] \times T \times M) / 1024^3$, where:

R is the number of resources for which the counters are produced,
N is the number of counters produced,
P is the interval of collection in seconds,
T is the retention period in the database (2x86400 seconds), and
M is the metric average size (2018 bytes).

For details on the values for **R**, **N** and **P**, see Table 12.

- Index size: $0.3 \times (\text{Actual data size})$

— Journal size: 5 GiB

— Local database size: $0.05 \times (\text{Actual file size} + \text{Journal size})$

Note: A partition size of minimum 40 GiB must be allocated for MongoDB.

Table 12 Operands for MongoDB Actual Data Size Formula

i (Index)	R	N	P
1..n	Number of VMs with k number of disk partitions in guest OS (for example, vda, vdb)	$45 + (k - 1) \times 8$	300
n+1	Number of compute hosts	5	300
n+2	Number of Neutron ports ⁽¹⁾	7	900
n+3	Number of compute hosts ⁽²⁾	1	900
n+4	Number of Neutron networks	3	300
n+5	Number of Neutron subnets	3	300
n+6	Number of Neutron ports	3	300
n+7	Number of Neutron routers	3	300
n+8	Number of floating IPs	3	300
n+9	Number of stacks	5	300
n+10	Number of images	6	300
n+11	Number of volumes	11	300
n+12	Number of snapshots	4	300



i (Index)	R	N	P
n+13	Number of Keystone users	6	300
n+14	Number of Keystone roles	5	300
n+15	Number of Keystone trusts	5	300
n+16	Number of Keystone groups	3	300
n+17	Number of Keystone projects	3	300
n+18	Number of Swift storage objects	8	300

(1) SDN port meters. Applicable only to CEE with Tightly Integrated SDN.

(2) SDN switch meters. Applicable only to CEE with Tightly Integrated SDN.

4.4.2 Local Storage Disk Space on BSP Hardware Platforms

This section lists requirements on local disk space on BSP hardware platforms.

- For compute hosts based on GEP5-64, see Table 13.
- For compute hosts based on GEP7-128-X and GEP7L-64-X, see Table 14.
- For compute hosts based on GEP5-64-400, GEP5-64-1200, GEP7-128-X16, GEP7L-64-X16 and GEP7L-64-X04, see Table 15.
- For vCIC, see Table 16.

The disk space of the GEP boards is used as follows:

- **GEP5-64** has a 8 GiB disk. CEE infrastructure uses 4.9 GiB, and only 2 GiB is available for ephemeral storage for applications.
- **GEP7-128-X** has a 16 GiB disk. CEE infrastructure uses 4.4GiB, and only 9.6 GiB is available for ephemeral storage for applications.

The disk space on GEP5-64 and GEP7-128 is not enough to store debug logs. Instead, the logs from compute nodes are forwarded to the vCIC and stored in the vCIC. When using these GEP board versions as compute nodes, CEE can be configured to forward and store the logs on the vCIC.

Among the various GEP models, only GEP5-64-1200, GEP7-128-X16, GEP7L-64-X16 can be selected as compute hosts for the vCICs, due to availability of additional disk space. The servers hosting the vCICs must have identical hardware characteristics in the CEE region, mixed configurations are not possible.

Table 13 Compute Host with 8 GiB Disk, GEP5-64

Use	Compute Host
Root partition (host OS)	4.9 GiB



Use	Compute Host
Logs and crash dumps	0 GiB
Remaining storage is for ephemeral storage for VMs	2 GiB

Table 14 Compute Host with 16 GiB Disk, GEP7-128-X and GEP7L-64-X

Use	Compute Host
Root partition (host OS)	4.4 GiB
Logs and crash dumps	0 GiB
Remaining storage is for ephemeral storage for VMs	9.6 GiB

Table 15 Compute Host, GEP Boards

Use	Compute Host with vCIC GEP5-64-1200 GEP7-128-X16 GEP7L-64-X16	Compute Host without vCIC GEP5-64-400 or GEP5-64-1200 GEP7-128-X GEP7L-64-X04 GEP7L-64-X16	Note
Root partition (host OS)	50 GiB	50 GiB	
Logs and crash dumps	40 GiB	40 GiB	
vCIC Total	Sum of the relevant items in Table 16	-	See Table 16.
Host total without vFuel and Atlas	Sum of the items above	90 GiB	



Use	Compute Host with vCIC GEP5-64-1200 GEP7-128-X16 GEP7L-64-X16	Compute Host without vCIC GEP5-64-400 or GEP5-64-1200 GEP7-128-X GEP7L-64-X04 GEP7L-64-X16	Note
vFuel	80 GiB	80 GiB	vFuel can be run on any compute host and it uses disk from the ephemeral storage. Fuel Snapshot: If Fuel snapshot is planned to be used, additional disk space must be configured accordingly, on top of the 80 GiB. This Fuel feature allows collecting Fuel-specific diagnostic information and logs from the compute nodes. By default, it tries to fetch logs from all compute nodes. It can be configured to collect logs from a subset of compute nodes or even from no compute nodes.
Atlas	>20 GiB	>20 GiB	Atlas can be run on any compute host and it uses disk from the ephemeral storage. The disk for Atlas is allocated by the CEE. For more details, refer to Section 4.4.3 on page 28.
Remaining storage is used as ephemeral storage for VMs.	Dimensioned depending on application need.	Dimensioned depending on application need.	Valid for boot from image. Calculated from total disk reduced by used space.

Table 16 vCIC Disk Allocation

Note: This configuration is optimized for a BSP system equipped with 72 blades. Other configurations are possible, provided they comply with the general guidelines included in this document.	
Use	vCIC
Root partition (host OS)	50 GiB
Logs and crash dumps	244 GiB



Database for OpenStack and Zabbix (MySQL)	Use the formula provided in Section 4.4.1.1 on page 23.
Database for Ceilometer (MongoDB)	Use the formula provided in Section 4.4.1.1 on page 23.
Glance repository in Swift	100 GiB

4.4.3 Disk Requirements for Atlas

Atlas uses a fixed disk size value of 10GiB and a configurable ephemeral storage size with default value of 120GiB. When images are loaded using Atlas from Images or Catalog Panel (as part of an .ova file), the image is temporarily stored in ephemeral storage in Atlas. To support loading of large images, the recommendation is to use 120 GiB for the Atlas ephemeral storage.

In case the local disk is used as ephemeral storage (no centralized storage or distributed storage), the Atlas VM occupies 130 GiB (10 GiB disk + 120 GiB ephemeral) of the local disk on the compute host where it is running.

To reduce the ephemeral disk allocated to Atlas, the size of the ephemeral disk can be reduced from 120 GiB to a minimum of 10 GiB.

Note: 30% of the ephemeral disk in Atlas is used as a temporary storage for images or .ova files. Consequently, the size of the ephemeral disk needs to be adjusted according to the size of images to be loaded. Using a reduced ephemeral disk size of 10 GiB implies that it can be impossible to load images or .ova files that contain images larger than 3 GiB.

4.4.4 Disk Requirements for Nova Snapshots

Nova snapshots are stored in the `/var/lib/glance` partition of CIC nodes.

There are certain disk requirements for the Nova snapshots to work. Depending on the requirements and frequency of Nova snapshots, the system must be dimensioned with free disk space, according to the following guidelines:

- Disk partition `/var/lib/nova` in the compute host where the VM is hosted, must have **at least** double the space of the snapshot/VM size, for a successful Nova snapshot. The reason is that the snapshot is first extracted locally in the compute node before it is uploaded to the Glance/Swift store.
- The disk space needed in the `/var/lib/nova` partition of the compute disk must have free space **at least** twice the size of VMs root disk. The reason is that during the extraction of the snapshot, first the delta of the VM disk will be extracted, after which the complete disk will be extracted.
- Disk partition `/var/lib/glance` in each CIC node must have free space **at least** equal to the root disk size of the VM in order to accommodate the snapshot.



4.4.5 Disk Requirements for ScaleIO

In certain configurations, the EMC² ScaleIO Value Package provides distributed storage on dedicated hosts. ScaleIO is optional and it is used as the backend for Cinder.

The following requirements must be fulfilled:

- Needed disk space on each ScaleIO host for various ScaleIO components like MDM/TB, SDS, ScaleIO Gateway or LIA: 1 GB

Note: The set of ScaleIO components installed on a ScaleIO host depends on the configuration of the host according to the role of that host in the ScaleIO distributed storage. 1 GB disk space is enough for any ScaleIO server configurations.

- Minimum disk space to be added as device to one SDS: 100 GB (this must be a physical disk)
- Minimum number of SDSs: 3

Note: The feature `Swift store` on ScaleIO uses space from the storage pool configured for Cinder. In the ScaleIO distributed storage, the feature uses the same amount of disk space as the sum of the amount of disk space available for the Swift storage on the three vCICs together. So the space used by the feature is three times larger than the space for Swift storage on one vCIC.

4.4.6 Disk Requirements for Software RAID

The software RAID feature uses the local storage disk space. The feature is optional.

The following requirements must be fulfilled to enable software RAID in CEE:

- Two physical disks on the compute blade
- Each disks must satisfy the minimum space requirement as described in Section 4.4.1 on page 21
- Each disk must have more than 15 GiB free space available

After software RAID is configured on the compute blade, the disk capacity is reduced to half. For more details and implementation, refer to the [Configuration File Guide](#).

4.4.7 Storage Performance Consideration

To achieve optimal read and write performance in storage aspect, the usage of Solid-State Drives, if possible high-performance NVMe ones, is strongly recommended, especially for compute hosts hosting vCICs.



4.5 Kickstart Server

Table 17, Table 18, and Table 19 describe the resources to be allocated to vFuel on the kickstart server used for the installation of CEE. These settings are used in the procedure described in [Preparation of Kickstart Server](#).

Table 17 vFuel CPU Core Allocation on the Kickstart Server

Number of Physical Servers ⁽¹⁾ to be Installed	Minimum Number of Cores Dedicated to vFuel
Up to 16	2
17–48	4
49–80	6

(1) Sum of compute hosts and ScaleIO hosts

Table 18 vFuel RAM Allocation on the Kickstart Server

Number of Physical Servers ⁽¹⁾ to be Installed	Total vFuel RAM (GiB)
Up to 16	3
17–48	8
49–80	12

(1) Sum of compute hosts and ScaleIO hosts

Table 19 vFuel Disk Space Allocation on the Kickstart Server

Number of Physical Servers ⁽¹⁾ to be Installed	Disk Size for vFuel
1–80	<p>The minimum size is the same as the amount allocated for vFuel at a compute host in Table 9.</p> <p>If the reserved disk size for vFuel is increased by configuring the <code>config.yaml</code> file, then the configured disk size must be available for vFuel on the kickstart server. Refer to the Configuration File Guide for more information.</p>

(1) Sum of compute hosts and ScaleIO hosts

5 Characteristics

This section describes the system characteristics of CEE.



5.1 General System Limits

For the list of system limits, see Table 20.

Table 20 General System Limits

Slogan	Limit
Number of physical servers ⁽¹⁾	CEE has been verified for working with 80 physical servers ⁽¹⁾ . Larger configurations can be used but those require configuration and tuning as a System Integration activity.
Number of compute hosts dedicated to CEE	<p>For CEE installations with tightly integrated SDN, the three physical servers hosting the vCICs are dedicated to CEE services. Tenant VMs must not be deployed on these compute hosts. Optionally, vFuel and Atlas can be co-located with vCIC.</p> <p>Refer to CEE on HDS Installation for the compute node segregation procedure.</p>



Slogan	Limit
RAM used by the infrastructure ⁽²⁾	<p>See Section 4.3 on page 15 for more information.</p> <p>The minimum memory requirement depends on the following:</p> <ul style="list-style-type: none">• System with or without tightly integrated SDN• The amount of nominal physical memory• vCIC on the server• vFuel on the server• The planned maximum number of physical servers (compute hosts plus ScaleIO hosts)• The number of physical servers (compute hosts plus ScaleIO hosts) to be added in one expansion procedure
Number of cores occupied by the infrastructure	<p>CEE infrastructure uses the following on all compute hosts (including vCIC and vFuel hosts):</p> <ul style="list-style-type: none">• 2 cores occupied by Host OS⁽³⁾• 1–4 cores occupied by CSS, see Section 5.4.1 on page 34. <p>vCIC and vFuel use additional cores at the hosting compute servers. One vCIC and one vFuel instance can be combined on one compute host configured for such usage. The following number of cores is used:</p> <ul style="list-style-type: none">• The vCIC uses 6–18 cores depending on the following parameters of the used CEE configuration:<ul style="list-style-type: none">– System with or without tightly integrated SDN– The planned maximum number of physical servers (compute hosts plus ScaleIO hosts)Three instances of vCIC are needed, and each of these must run on a separate compute host that is configured as a vCIC host. See Section 4.2.1 on page 9 for more information.• Two instances of vFuel must be allocated. Depending on the number of physical servers (compute hosts plus ScaleIO hosts) to be added in one expansion procedure, a vFuel instance uses the following amount of cores at the compute host on which it runs:<ul style="list-style-type: none">1–16 physical servers in one step: 1 core1–48 physical servers in one step: 4 cores



- (1) Sum of compute hosts and ScaleIO hosts
- (2) If Atlas is used, it runs at one of the physical servers, and it uses 4 GiB of the RAM configured for the tenant VMs.
- (3) The default configuration use 2 cores for the host OS. Depending on the load generated and the characteristics required by the Virtual Network Function (VNF), this can be reduced to 1 core. Careful load measurements are needed for such tuning, and the instructions for performing such measurements are outside the scope of this document.

5.2 Orchestration Interface

The system limits for orchestration are listed in Table 21.

Table 21 Orchestration Limits

Slogan	Limits
Number of tenants	The maximum number of verified tenants is 50.

5.3 Tenant Execution Environment

This section describes the tenant-related limits on the environment.

5.3.1 Performance

Performance limits are listed in Table 22.

Table 22 Tenant Execution Performance

Slogan	Limits
Oversubscription	CPU, memory, and disk overcommit are not supported.

5.3.2 Resiliency

Resiliency-related tenant limits are listed in Table 23.

Table 23 Tenant Execution Resiliency

Slogan	Limits
Execution environment resiliency	<p>The execution environment resiliency is relying on VM evacuation. States not conserved in storage are lost.</p> <p>Each hypervisor instance is not redundant, and, apart from attached storage, assumed to be a knock-out unit.</p>



5.4 Network

5.4.1 Performance

5.4.1.1 Performance Indicators

CSS processes the packets transmitted by physical ports and the packets transmitted by the VM network interfaces. The following indicators characterize the performance of a virtual switch:

- Throughput
- Packet loss
- Latency
- Jitter
- Packet re-ordering

The maximum throughput of a virtual switch is measured by the packet rate (packet per second) processed with a certain level of packet loss (for example, 10 ppm), and depends on the following factors:

- Compute host hardware processor generation
- Clock speed of CPU cores
- CPU caches
- Number of PMDs
- NUMA allocation of PMDs, VMs, and NICs
- VNF behavior as “noisy neighbor”
- Traffic pattern

It is recommended to perform the following steps:

1. Measure the throughput with the actual VNFs as tenant VMs using the actual traffic type and profile.
2. Use the results to dimension the cloud, for example, to optimize the number of PMD threads and VMs per host.

5.4.1.2 CSS PMD Allocation Options

CSS executes a configurable number of threads running in endless loops, called PMD threads. The allocation of PMD threads must be decided before the installation of the CEE region. Each PMD thread polls interfaces that are



automatically assigned to it, processes the incoming packets and puts them into a queue to be transmitted.

5.4.1.2.1 Automatic CPU Allocation Option

It is recommended to use the automatic allocation option for CSS PMDs at CEE installation.

For more information, refer to the [Advanced CPU Allocation](#) section in the [Configuration File Guide](#) and the [CSS User Guide, Reference \[4\]](#).

5.4.1.3 DPDK Physical Interface Driver

The supported DPDK physical interface driver in the multi-server deployments of CEE is the generic `vfio-pci`, that is, the driver for Virtual Function I/O (VFIO).

To provide a stable operation and the full functionality of CSS, the `vfio-pci` driver must be configured on the physical interfaces used by DPDK on each compute host. Check the [DPDK Physical Interface Driver](#) section in the [Configuration File Guide](#) for the configuration details.

5.4.1.4 NUMA Balancing

Not applicable to systems with single NUMA node.

Automatic NUMA balancing is a Linux kernel feature that improves the performance of applications running on NUMA hardware systems.

This feature interrupts CSS PMDs through soft page faults, triggering packet drops. Hence, it can be switched off per compute host or for the whole system. Refer to the [NUMA Balancing](#) section of the [Configuration File Guide](#) for configuration details.

5.4.1.5 Customized QEMU

The customized QEMU with increased VirtIO queue size reduces the vulnerability of certain applications (VNFs) with high throughput requirements to early packet drop (in the order of 100 ppm) before saturating the CPU in VNF or CSS (vSwitch). This can significantly increase the application throughput for packet-loss sensitive VNFs.

Not all VNFs benefit from the customized QEMU. In addition, some existing VNFs that work on the standard QEMU in CEE 6 may not be compatible with the increased VirtIO queue size even though that is in the range specified by VirtIO 1.0. Such VNFs cannot be deployed on hosts with the custom QEMU.

Due to this, CEE supports the following two types of QEMU:

— Standard QEMU

- Customized QEMU with VirtIO queue size increased to 1024 bytes

Configure Customized QEMU:

In CEE, the QEMU type to be installed can be configured per compute hosts. For more information, refer to section [Increasing Virtio Queue Size in the Configuration File Guide](#).

Note: The customized QEMU is to be installed only on those compute hosts that are designated to host performance-critical VNFs that have been certified to be compatible with the custom QEMU. At the time of this release, the following Ericsson VNFs have been certified: vMSP and vEPG. This list is expected to grow as VNFs are on-boarded to CEE 6. VNFs that are not yet certified to be compatible must be deployed on hosts with the standard QEMU.

Isolate Compute Hosts with Customized QEMU:

If the deployment consists of compute hosts with customized QEMU and compute hosts with standard QEMU, it is important to separate them in two groups to allow scheduling for these two sets of hosts in a controlled manner.

This can be achieved in multiple ways by using Nova scheduling mechanisms depending on the deployment.

A simple way to achieve this is to create an availability zone by grouping the hosts with customized QEMU. This way, VMs instantiated on the specific availability zone will only be scheduled on hosts with customized QEMU. However, as a specific host can only be added to one availability zone, for this option no other availability zones can be used for hosts with customized QEMU.

Another method is to create an availability zone according to the needs of the deployment and create host-aggregates among the availability zones. In this option the compute hosts using customized QEMU must be grouped into a host-aggregate and VMs can be scheduled on this aggregate host using the appropriate extra specs in the flavor.

The procedure to create the availability zone and host-aggregates depends on the dimensioning of the deployment, the usage of availability zones in the deployment, and the orchestration (for example on usage of VNFM or NFVO).

An example procedure to create host-aggregates through CEE OpenStack CLI is provided in section [Isolating Compute Hosts According to VirtIO Queue Size in SW Installation in Multi-Server Deployment](#).

5.4.1.6

CPU Pinning

Since certain VNFs/VMs use data plane intensive applications, sharing CPUs among different VMs leads to poor and non-deterministic performance. CPU pinning for the VNFs/VMs must be used to avoid interference between tenant VMs running on the same host. CEE is tested with CPU pinned VMs. Use the relevant



flavor extra spec: hw:cpu_policy = dedicated in accordance with OpenStack Compute API in CEE.

5.4.1.7 Libvirt Real Time Instances

Some applications use dedicated vCPUs for DPDK PMD threads. If the VNF/VM is instantiated with CPU pinning, one-to-one mapping is used between the CPUs and vCPUs. Even if the flavor extra spec hw:cpu_policy = dedicated is used (see Section 5.4.1.6 on page 36), the emulator thread is pinned to all of the CPUs dedicated to the guest.

Note: The usage of the emulator thread depends on the specific QEMU implementation. For example, in QEMU 2.5, a prolonged blocking of the emulator thread has been observed.

When the guest uses dedicated vCPUs for DPDK PMD threads and the emulator also uses the corresponding CPUs, bursts of packets or even packet drops inside the guest can be experienced.

Pinning emulator threads to CPUs that are not used for data plane intensive applications can be a significant improvement. The user, for example VNFM, can instantiate VNFs/VMs with the following settings:

- The emulator thread is pinned to a subset of CPUs that are not used for DPDK PMD thread among all CPUs dedicated to the guest.
- A subset of vCPUs (for example, the vCPUs dedicated by the application to DPDK PMD threads) uses first in, first out (FIFO) as kernel scheduling policy, and priority 1.

The above configuration is achieved by using a real-time mask via nova flavor extra spec when booting the VNF/VM.

Note: Ensure that the VM can handle real time scheduling. The VNF/VM must support such configuration to be able to use this feature. Refer to section [Libvirt Real Time Instances in OpenStack Compute API in CEE](#) for the configuration options at VNF/VM instantiation.

5.4.1.8 vSwitch Maximum Throughput for Bandwidth-based Scheduling

The bandwidth-based scheduling feature makes it possible to schedule VMs based on the required outbound and inbound bandwidth per virtual interface of a tenant VM. The function selects a certain host for use if the available outbound and inbound bandwidth that are not used by other processes on the host are enough to fulfill the request.

To configure a certain compute host, the maximum achievable bandwidth is assigned to the compute host. Refer to the [Bandwidth Based Scheduling](#) section of the [Configuration File Guide](#) for assigning the bandwidth to the compute host.

For the NICs used by the CSS, the following parameters must be configured:



- The nominal bandwidth of the NIC port in kbit per second as the value for the **bandwidth** parameter.
- The unidirectional throughput capacity of the vSwitch in kilo packets per second as the value for the **vswitch_capacity**.

vswitch_capacity is to quantify the throughput capacity of the vSwitch for traffic flowing:

- From one physical interface to one virtual interface
- From one virtual interface to one physical interface
- From virtual to virtual interface

The same value is used for both inbound and outbound capacity. Consider the results of the measurements performed in Section 5.4.1.1 on page 34 to apply the most appropriate value. Bandwidth based scheduling compares the free (unused) inbound bandwidth on the host with the value configured for **vswitch_capacity**. The free outbound bandwidth is checked in the same way.

Note: The unidirectional throughput capacity of the vSwitch **vswitch_capacity** must not be confused with the vSwitch forwarding capacity that is the total amount of packets processed by the vSwitch. The vSwitch forwarding capacity includes all hops involved in the traffic case. For example, one traffic generator sends packets to a load balancer VM that then dispatches it to a traffic processor in the same compute node. The traffic processor then returns it back to the traffic generator. In this case the flow traverses the vSwitch 3 times. If the traffic generator sends and receives 1 Mpps, the vSwitch forwarding capacity is 3 Mpps.

The following example can be used for early dimensioning, before the actual measurements are available:

Table 24 vSwitch Forwarding Capacity

Application Type	vSwitch Forwarding Capacity
DPDK driver based application	3-4 Mpps
Kernel driver based application	2-2.4 Mpps

The above figures were obtained from the measurement of real VNF traffic, with the following configuration specifications:

- Turbo mode disabled in BIOS
- CEE without SDN
- One PMD thread per NUMA node
- Idle hyper thread siblings
- NUMA balancing disabled



- Customized QEMU
- Emulator thread pinned to first CPU (corresponding to vCPU 0) for the VM running DPDK application
- Mix of packet sizes
- Average packet size of 800 bytes/packet

5.4.1.9 BIOS Settings

To achieve optimal and deterministic networking performance, it is recommended to configure the physical servers before installing CEE:

- Disable CPU power management (enable the “Maximum Performance” option)
- Disable CPU Turbo Boost

5.4.2 Resiliency

Network resiliency is listed in Table 25.

Table 25 Network Resiliency

Slogan	Limits
Self-healing network	The network solution is self-healing, including network fault detection and automated failover.

5.4.3 Tenant Network Limitations

Limitations of the tenant network are listed in Table 26.

Table 26 Tenant Network Limitations

Slogan	Limits
Number of virtual networks	The theoretical aggregated maximum number of virtual tenant networks per CEE region is 4050 for segmentation type v1an. Since each Neutron network created consumes RAM in the vCIC, this theoretical maximum cannot be reached. The default configuration of RAM for vCIC allows 1000 networks. Additional memory is needed if more Neutron Networks are created.
Number of vNICs per guest VM	The maximum number of vNICs per guest VM is 10 (+ 1 Trunk vNIC).



Slogan	Limits
Number of Trunk vNIC attached vLANs	The number of Trunk vNIC attached vLANs is limited to 100.
Number of vNICs per server	CSS supports up to 64 vNICs per NUMA node with default RAM allocation of 1 GiB for CSS per NUMA node.
MTU	The infrastructure MTU size for tenant traffic is 2140 bytes by default. For LAB experiments it can be configured in CEE. The configured value is used by all vSwitches in the CEE region. See Section 4.3.2 on page 17 for memory allocation to CSS depending on the configured MTU size.
Number of static routes ⁽¹⁾	<p>The default maximum number of static routes is 1000. It is specified by the <code>ext_max_routes</code> value in the <code>neutron.conf</code> file.</p> <ul style="list-style-type: none">• For setting the value at CEE installation, refer to the Neutron Configuration Options in the Configuration File Guide.• For setting the value in a running system, refer to the Static Routes section in the Runtime Configuration Guide.

(1) Only applicable to configurations with Neutron managed Extreme switches.

5.5 Storage

This section describes CEE characteristics on storage.

5.5.1 Limitations When Using Local Storage

The following limitations apply when using local storage:

- For tenants, ephemeral storage (non-persistent block storage) is supported on local disks of the compute hosts.
- If the compute host must be replaced (for example, in case of a hardware failure), any change in the VM data stored on the ephemeral disk is lost.
- Shared file system is not available in CEE. For distributed storage, see separate sections.
- Management of VM images is supported by the OpenStack image service.
- Object storage through Swift is only used for CEE infrastructure.
- Boot from volume is not possible if the only available storage is the local storage on the compute nodes. Boot from volume requires Cinder-managed



volumes stored on a backend storage system. The supported storage backend type for Cinder is ScaleIO.

- If a VM has managed-on-host set as ha-policy and is booted from local storage, it will not be started again after a CEE SW rollback, since the /var/lib/nova partition is not preserved.

Table 27 shows where data is stored for both deployment options.

Table 27 Storage Locations

Data	Storage Location
CEE infrastructure backups (incl. Fuel backups)	On local disks of vCIC hosts ⁽¹⁾
Ephemeral storage	On local disks of compute hosts ⁽¹⁾
vCIC storage (OpenStack infrastructure)	On local disks of vCIC hosts ⁽¹⁾
Core/crash dumps, logs	On local disks of all hosts

(1) Local storage is limited by the local non-scalable disk capacity.

If data is stored on a local disk, it is erased in case of disk failure or rollback from a failed update, meaning that the VM disappears. The application must be designed/configured accordingly.

5.5.2

Resiliency

Storage resiliency characteristics are listed in Table 28.

Table 28 Storage Resiliency

Slogan	Characteristics
Centralized storage resiliency	All components of the centralized storage array are redundant.



Slogan	Characteristics
Swift storage resiliency	<p>Glance uses Swift as back-end. Swift uses the local disks or the distributed storage ScaleIO:</p> <ul style="list-style-type: none">• Local disks: Swift storage is replicated over the local disks that run vCIC.• ScaleIO: Swift storage is replicated by the ScaleIO storage resiliency.
ScaleIO storage resiliency	<p>The smallest granularity of ScaleIO resilient entity is a logical entity called Fault Set. Data mirroring for all devices in a Fault Set takes place on SDSs that are outside of that Fault Set. A Fault Set is a set of units that usually get failed simultaneously.</p> <p>To decide how to create and dimension a Fault Set, refer to the Dell EMC ScaleIO Version 2.x User Guide.</p> <p>To configure a Fault Set refer to the Shelf and Blade Management section of the Configuration File Guide</p>

5.5.3 Distributed Storage, ScaleIO

Distributed Storage Limits are listed in Table 29.

Table 29 Distributed Storage Limits

Slogan	Characteristics
Maximum device size	8 TB
Maximum capacity per SDS	64 TB
Maximum number of devices per SDS	64

5.6 In-Service Performance

This section lists the characteristics on in-service performance.

Table 30 In-Service Performance

Slogan	Characteristics
Guest execution retainability	Guest execution is not interrupted at a virtual infrastructure management cluster restart. At CEE software update the compute nodes are restarted sequentially and this causes VM evacuations or VM restart.



Slogan	Characteristics
Update availability	When the CEE software update is running, the OpenStack API service is migrated to each vCIC sequentially, therefore the service is unavailable for about one minute during the migration.
Rollback availability	The OpenStack API service is unavailable during the backup procedure.
Restart availability	The OpenStack API service is not available during the restart of the Virtual Infrastructure Manager (VIM) cluster.



Reference List

- [1] CEE Architecture Description, 5/155 53-AZE 102 01, available at Ericsson Support
- [2] BSP Hardware Baseline, 1090-CRA 119 1772, available at Ericsson Support
- [3] BSP Technical Product Description, 221 02-FGC 101 2255
- [4] CSS User Guide, 1553-AXT 901 11/2