Centralized License Manager

Release 24.11

CLM Installation and Upgrade Guide

3HE-20359-AAAA-TQZZA
Issue 1
December 2024

Legal notice

Nokia is committed to diversity and inclusion. We are continuously reviewing our customer documentation and consulting with standards bodies to ensure that terminology is inclusive and aligned with the industry. Our future customer documentation will be updated accordingly.

Contents

About this document

Purpose

The *CLM Installation and Upgrade Guide* provides detailed information regarding the installation of the CLM, including pre- and post-installation activities.

Safety information

For your safety, this document contains safety statements. Safety statements are given at points where risks of damage to personnel, equipment, and operation may exist. Failure to follow the directions in a safety statement may result in serious consequences.

Document support

Customer documentation and product support URLs:

- Documentation Center
- Technical support

How to comment

Documentation Feedback

Part I: Getting started

Overview

Purpose

This part of the *CLM Installation and Upgrade Guide* provides an introduction and describes how to commission the host platform.

Contents

1 Before you begin

CLM deployment overview

1.1 Getting started with CLM

1.1.1 The planning phase

Before you consider deploying a new CLM, upgrading or adding to a system, or modifying a system in any way, the reader is strongly encouraged to become familiar with the *CLM Release Notice* for release-specific information that may apply to your deployment.

1.1.2 Using this guide

After the planning phase, you must become familiar with the relevant content in this and the subsequent Part I: “Getting started” chapters, which describe platform preparation.

When the platform configuration is established, your planned deployment action can proceed using the required procedures in Part II: “CLM system deployment”.

1.2 CLM system redundancy

1.2.1 Disaster recovery (DR) deployment

CLM installations can provide redundancy through disaster recovery (DR). DR is available to any deployment size where two identical CLM installations exist in geographically separate data centers, providing fundamental network resiliency through geographical redundancy. In a DR deployment, the geographically dispersed installations must be identical in terms of CLM cluster and redundancy models.

CLM supports standalone or DR deployment, but not HA.

1.3 CLM deployment terms and concepts

1.3.1 Introduction

The CLM guide uses specific terms to describe some basic elements of a CLM system. The following topics define commonly used terms used in the CLM system documentation that may be unfamiliar to the reader.

Note: Although the usage of each term is typically as described, a specific context may include a variant of the term.

1.3.2 Station

A station is a physical processing entity that has one native OS instance, or hosts OS instances in multiple VMs. The term station is typically used only for configuration operations, such as actions that require CLI access.

The term “host station” is sometimes used to indicate the physical station that hosts a specific VM or function.

1.3.3 CLM deployer host

The deployer host is a VM from which you deploy the container environment for a CLM cluster.

1.3.4 CLM cluster

A CLM cluster is a VM in a Kubernetes container environment that hosts CLM software and functions. A CLM system deployment includes at least one CLM cluster.

1.3.5 CLM cluster member

A CLM cluster member is a VM in a CLM cluster. A CLM cluster member is also called a cluster node, depending on the context.

1.3.6 CLM cluster host

The CLM cluster host is a specific CLM cluster member from which CLM deployment operations in the cluster are performed. Node 1 of a cluster is chosen as the CLM cluster host.

1.4 Browser applications

1.4.1 Browser applications

The CLM provides functionality using browser-based applications. These applications use standard REST security mechanisms for authentication and authorization. All applications are HTML-5 based and are supported on the following web browsers:

- Latest version of Google Chrome
- Latest version of Chromium Edge for most applications (see the CLM Release Notice for restrictions)
- Latest version of Mozilla Firefox
- Latest version of Apple Safari

Additional Internet browsers and older versions may function with CLM but are not supported by Nokia.

Note: You cannot switch browsers between clients. You must always use the system default browser.

Note: If you are using Chrome or Firefox on Windows 8.1, it is recommended that you enable ClearType Text for optimal viewing of fonts. To enable, open the Display settings in Windows Control Panel and enable the Turn on ClearType parameter under the Adjust ClearType text settings.

Localized language support

CLM supports localized language display. Localized language display, also known as internationalization, displays GUI text in a specified language. The localized language setting applies to most GUI objects, except system elements and database objects. Contact Nokia technical support for more information about localized language support.

Note: The CLM supports localized language settings using predefined strings, and does not translate data to different languages.

2 System requirements for CLM

2.1 Container environment requirements

2.1.1 Supported CLM cluster environments

A CLM cluster must be deployed in a Nokia provided container environment.

Nokia supports and recommends installing CLM on VMs using VMware ESXi or RHEL KVM, including OpenStack. The guest OS must be a supported RHEL version.

2.1.2 Supported container software versions

The CLM is validated against a container environment that uses the following software versions:

Table 2-1 Supported container software versions

Containerization element	Supported version
Kubernetes core	v1.29.5
calico	v3.27.3
cni	v1.3.0
containerd	v1.7.16
etcd	v3.5.12
Helm	v3.14.2
coredns	v1.11.1
kubespray	2.25.0
k9s	v0.32.5
metallb	v0.13.9
harbor app	v2.11.1
nerdctl	v1.7.4

2.1.3 KVM virtualization

The CLM supports using RHEL 6, RHEL 7, and RHEL 8 based KVM on x86 based servers natively supported by KVM. See the *Host Environment Compatibility Reference for NSP and CLM* for the current KVM compatibility level, requirements, and restrictions. See the RHEL Hardware Compatibility List (HCL) for information about specific hardware support.

Not all features offered by KVM are supported when using the CLM. For example Snapshots, Live Migration and HA are not supported. Contact Nokia to determine if a specific KVM feature is supported with an installation of CLM.

KVM CPU and Memory

The required memory resources must be reserved and dedicated to each guest OS, and cannot be shared or oversubscribed. In a medium development type, CPU resources may be oversubscribed with a KVM cpu_allocation_ratio of 1.25.

Note: CPU oversubscription is only supported on CLM clusters with three or more nodes. Smaller CLM deployments must have reserved and dedicated CPU resources for each node.

KVM configuration

When you configure the KVM, set the parameters listed in the following table to the required values.

Table 2-2 KVM configuration parameters

Parameter	Value
Disk Controller type	virtio
Storage format	raw
Cache mode	none
I/O mode	native
NIC device model	virtio
Hypervisor type	kvm

CPU pinning may be used for CLM virtual machines (deployer and cluster nodes) but may restrict other KVM functions. Please consult KVM documentation for more information.

2.1.4 OpenStack virtualization

The CLM is tested in an open-source OpenStack environment. Nokia supports CLM deployment on VMs provided in any OpenStack distribution that is based on the tested version. Any product issues deemed related to a specific distribution must be pursued by the customer and OpenStack vendor. The supported OpenStack versions include Newton, Queens and Train.

To ensure CLM compatibility with an OpenStack environment, you must follow the requirements in the following topics.

Hypervisor

KVM is the only supported hypervisor for an OpenStack environment. For information about the supported KVM hypervisor versions, see 2.1.3 “KVM virtualization” (p. 15).

CPU and memory resources

The required memory resources must be reserved and dedicated to each guest OS, and cannot be shared or oversubscribed. You must set the ram_allocation_ratio parameter to 1.0 in the OpenStack Nova configuration on the control NE, or on each individual compute node that may host a CLM VM. CPU resources may be oversubscribed using the guidelines for KVM CPU oversubscription in Table 2-2, “KVM configuration parameters” (p. 16).

Simultaneous Multi-threading (SMT)

The usage of CPUs with enabled SMT must be consistent across all compute nodes. If the CPUs do not support SMT, you must disable SMT at the hardware level on each compute node that may host a CLM VM.

CPU pinning

CPU pinning is supported, but may restrict some OpenStack functions such as migration.

Availability zones/affinity/placement

Nokia does not provide recommendations for configuring VM placement in OpenStack.

Migration

The OpenStack environment supports only regular migration; live migration is not supported.

Networking

Basic Neutron functions using Open vSwitch with the ML2 plugin are supported in an NSP deployment, as is the use of OpenStack floating IP addresses.

Storage

All storage must meet the throughput and latency performance criteria in Table 4-1, “Live partitioning scheme, CLM deployer host” (p. 40)

VM storage

The VM storage must be persistent block (Cinder) storage, and not ephemeral. In order to deploy each VM, you must create a bootable Cinder volume. The volume size is indicated in Table 4-1, “Live partitioning scheme, CLM deployer host” (p. 40)

Firewalls

You can enable firewalls using OpenStack Security Groups, or on the VMs using the firewalld service, except as noted. If firewalld is enabled, an OpenStack Security Group that allows all incoming and outgoing traffic must be used.

ARP spoofing

OpenStack ARP spoofing protection must be disabled. MetalLB load balancer looks like an ARP spoofing attempt to OpenStack.

2.1.5 VMware virtualization

The CLM supports using VMware vSphere ESXi 6.5, 6.7, 7.0 and 8.0 only, on x86 based servers natively supported by ESXi. See the *Host Environment Compatibility Reference for NSP and CLM* for the current VMware compatibility level, requirements, and restrictions. See the VMware Hardware Compatibility List (HCL) to determine specific hardware support.

Not all features offered by ESXi are supported when using the CLM. For example, Fault Tolerant, High Availability (HA), Memory Compression, Distributed Resource Scheduler (DRS), and vMotion

features are not supported. VM snapshots are not supported when using CLM. Contact Nokia to determine if a specific ESXi feature is supported with a CLM installation.

CPU pinning is supported on CLM virtual machines (deployer and cluster nodes) but may restrict some VMware functions. Please consult VMware documentation for more information.

Note: The system clocks of the CLM must always be closely synchronized. The RHEL chronyd service is the mandatory time-synchronization mechanism that you must engage on CLM during deployment.

Caution: Some entities - members of an etcd cluster, for example - will not trust the integrity of data if a time difference is detected. As such, failure to closely synchronize system clocks can complicate debugging and cause outages or other unexpected behavior.

Only one time-synchronization mechanism can be active in a CLM system. Before you enable a service such as chronyd on CLM , you must ensure that no other time-synchronization mechanism, for example, the VMware Tools synchronization utility, is enabled.

Virtual Machine Version 11 or above must be used.

See the following table for additional Virtual Machine setting requirements:

Table 2-3 Additional Virtual Machine setting requirements

Resource type	Parameter	Setting
CPU	Shares	Set to High
	Reservation	See VMWare CPU Reservation below
	Limit	Check box checked for unlimited
Memory	Shares	Set to High
	Reservation	Reserve all guest memory
	Limit	Check box checked for unlimited
Disk	Shares	Set to High
	Limit - IOPs	set to Unlimited
	Type	Thick Provision Eager Zeroed
SCSI controller	Type	VMware Paravirtual
Network Adapter	Type	VMXNET 3

VMWare CPU Reservation

CPU resources may be oversubscribed when CLM is deployed on VMWare. The CPU Reservation is determined by: factor X * the number of CPUs * the CPU frequency. For example, with a factor of 0.5 on a 2.4 GHz 8 vCPU configuration, the reservation must be set to (0.5 * 8 * 2400) = 9600 MHz. For a medium deployment type, the VMWare factor X is 0.4.

Note: CPU oversubscription is only supported on CLM clusters with three or more nodes. Smaller CLM deployments must have reserved and dedicated CPU resources for each node.

2.2 Cluster requirements for CLM

2.2.1 CLM VM hardware requirements

CLM deployments are server- and vendor-agnostic, but must meet all CLM hardware criteria and performance targets. Server-class hardware is required; desktop hardware is inadequate. Processor support is limited to specific Intel Xeon-based x86-64 and AMD Epyc x86-64 CPUs that have the required minimum CPU core speed listed in Table 2-4, “VM processor requirements” (p. 18).

Table 2-4 VM processor requirements

Processor microarchitecture	Minimum CPU core speed	Supported deployments
Intel Xeon Haswell or Broadwell	2.4 GHz	Supported for all CLM deployments
Intel Skylake or newer	2.0 GHz	Supported for all CLM deployments
AMD Epyc Zen 3 or newer	2.0 GHz	Supported for all CLM deployments

Provisioned CPU resources are based upon threaded CPUs. The CLM platform requirements will specify a minimum number of vCPUs to be assigned to the VM. VMs are recommended to be configured with all vCPUs on one virtual socket.

A host system requires CPU, memory and disk resources after resources for CLM VMs have been allocated. Contact the hypervisor provider for requirements and best practices related to the hosting environment.

You must provide information about the provisioned VMs to Nokia support. You can provide the information through read-only hypervisor access, or make the information available upon request. Failure to provide the information may adversely affect CLM support.

2.2.2 CLM cluster storage-layer requirements

The storage layer of a CLM cluster requires a minimum read/write IOPS based on deployment type, network size and storage type. Deployments with customer provided storage also have storage capacity requirements for database backups.

The following minimum CLM cluster IOPS requirement applies:

- Minimum storage-layer read/write IOPS for local storage: 2000

Note: IOPS performance can vary in a cloud environment and may degrade over time. Degraded IOPS performance may affect CLM performance. Customers are advised to monitor IOPS performance over time.

2.2.3 Minimum and production platform requirements

WARNING

Service Degradation Risk

The CLM deployer host is a crucial element of a CLM system that holds the required container images and Helm repositories for deployment to the CLM cluster VM. If the CLM deployer host is unavailable, cluster recovery in the event of a failure may be compromised.

Ensure that the CLM deployer host remains operational and reachable by the CLM cluster after the cluster deployment.

Each deployment requires one deployer node and one cluster node.

The minimum requirements are as follows:

- CLM deployer node
  - 4 vCPU
  - 8 GB memory
  - 400 GB disk
- CLM cluster node
  - 4 vCPU
  - 32 GB memory
  - 710 GB disk

2.2.4 VM memory

The virtual memory configuration of each CLM cluster VM requires a parameter change to support Centralized Logging.

The following command entered as the root user displays the current setting:

```
sysctl -a | grep "vm.max_map_count"
```

If the setting is not at least 26 2144, you must enter the following command before you deploy the CLM:

```
sysctl -w vm.max_map_count=262144
```

2.2.5 nsp system user

The CLM deployer and cluster nodes require a system user account named nsp that must have uid 1000. If a user other than nsp is already assigned to uid 1000, then that user needs to be deleted or modified before CLM can be installed.

2.2.6 Using hostnames

The hostname of CLM must meet the following criteria:

- include only alphanumeric ASCII characters and hyphens
- not begin or end with a hyphen
- if an FQDN, FQDN components are delimited using periods

- hostname FQDN does not exceed 63 characters

Note: If you use hostnames or FQDNs to identify CLM in the configuration, the hostnames or FQDNs must be resolvable using DNS.

2.3 Storage

2.3.1 Storage overview

This section provides information about configuring stations that will host CLM software.

When using the RHEL server OS, ext4 is the required file system for all application specific mount points. No other file systems are supported with CLM. OS specific mount points can be either xfs or ext4 as the file system. Windows based clients must use a local file system for client files. Network based files systems, including Samba are not supported.

While Nokia identifies areas of the storage that are not specifically required for CLM and are partitionable for customer use, station resources are expected to be dedicated for CLM. As such, these “Remainder” portions of the storage should only be used for static storage purposes. Consideration should also be made to the expected growth of the network. If the “Remainder” is not to be used, then it should not be created.

In CLM upgrade scenarios, previous disk configurations may still be valid.

2.3.2 Virtualization I/O requirements

When using CLM on a guest operating system of a hosted virtualized installation, specific storage requirements must be met. For optimal performance, storage should be either internal disks (10K or 15K RPM SAS, SAS or NVMe based SSDs), Fiber Channel attached storage (array) with dedicated fiber channel connections between hosts and Storage Arrays, or 10Gb iSCSI using non-shared infrastructure. In VMware environments, storage I/O shares must be set to “High” and IOPs set to “Unlimited” for best performance and low latency.

To verify disk performance for CLM, see 2.4 “To test CLM disk performance” (p. 21) for the minimum required throughput and latency for a collocated CLM configuration.

2.4 To test CLM disk performance

2.4.1 Purpose

Perform this procedure to check the disk performance on a CLM deployer host or CLM cluster VM.

2.4.2 Steps

1

Log in to the VM as the root user.

2

Open a console window.

3

Enter the following:

```
# cd /opt/nsp ↵
```

4

Enter the following:

```
# fio --randrepeat=1 --ioengine=libaio --direct=1 --gtod_reduce=1
--name=test --filename=random_read_write.fio --bs=4k --iodepth=64
--size=4G --readwrite=randrw --rwmixread=50 ↵
```

A file called 'test' is created in the /opt/nsp directory; the file contains disk performance test output like the following:

```
test: (g=0): rw=randrw, bs=(R) 4096B-4096B, (W) 4096B-4096B, (T)
4096B-4096B, ioengine=libaio, iodepth=64
fio-3.7
Starting 1 process
test: Laying out IO file (1 file / 4096MiB)
Jobs: 1 (f=1): [m(1)][100.0%][r=22.1MiB/s,w=22.2MiB/s][r=5645,w=5674
IOPS][eta 00m:00s]
test: (groupid=0, jobs=1): err= 0: pid=32439: timestamp
read: IOPS=6301, BW=24.6MiB/s (25.8MB/s)(2049MiB/83252msec)
   bw (  KiB/s): min=13824, max=39088, per=99.57%, avg=25098.60,
stdev=5316.27, samples=166
   iops        : min= 3456, max= 9772, avg=6274.49, stdev=1329.11,
samples=166
write: IOPS=6293, BW=24.6MiB/s (25.8MB/s)(2047MiB/83252msec)
   bw (  KiB/s): min=13464, max=40024, per=99.56%, avg=25062.73,
stdev=5334.65, samples=166
   iops        : min= 3366, max=10006, avg=6265.57, stdev=1333.67,
samples=166
  cpu          : usr=5.13%, sys=18.63%, ctx=202387, majf=0, minf=26
  IO depths    : 1=0.1%, 2=0.1%, 4=0.1%, 8=0.1%, 16=0.1%, 32=0.1%,
>=64=100.0%
     submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%,
>=64=0.0%
     complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.1%,
>=64=0.0%
     issued rwts: total=524625,523951,0,0 short=0,0,0,0 dropped=0,0,
0,0
     latency   : target=0, window=0, percentile=100.00%, depth=64
Run status group 0 (all jobs):
```

```
   READ: bw=24.6MiB/s (25.8MB/s), 24.6MiB/s-24.6MiB/s (25.8MB/s-25.
8MB/s), io=2049MiB (2149MB), run=83252-83252msec
  WRITE: bw=24.6MiB/s (25.8MB/s), 24.6MiB/s-24.6MiB/s (25.8MB/s-25.
8MB/s), io=2047MiB (2146MB), run=83252-83252msec
Disk stats (read/write):
  vda: ios=523989/526042, merge=0/2218, ticks=3346204/1622070,
in_queue=4658999, util=96.06%
```

5

Review the output to verify that the reported values are within the product specifications.

END OF STEPS

3 Network requirements

3.1 CLM deployment network addressing requirements

3.1.1 CLM cluster virtual IP address requirements

A CLM cluster must be configured with at least one virtual IP for client/ingress traffic, distinct from any cluster node IP addresses. This requirement for virtual IPs applies to CLM deployments with a single network interface, and to multi network interface deployments.

Each CLM cluster node that is an ingress gateway must have a network interface address that belongs to the same subnet for each VIP configured to the cluster.

3.1.2 Using IPv4 and IPv6 in CLM deployments

The CLM supports IPv4 and IPv6 network connectivity in the CLM architecture.

The following limitations and restrictions apply to deploying the CLM with IPv6 network communications:

- The deployer host for a CLM cluster must have IPv4 connectivity to CLM cluster nodes. The CLM cluster can be configured for IPv6 communications for CLM applications, but must have IPv4 connectivity to the deployer node.
- Common web browser applications have security policies that may prevent the use of bracketed IPv6 addresses in the URL browser bar. Customers who use IPv6 networking for client communications to the CLM must use the hostname configuration.
- The CLM Kubernetes cluster communications uses internal addressing in 10.233.0.0/18 subnet. Customers should avoid using this subnet in their CLM deployment on VM network interfaces.

The CLM can be deployed with multiple network interfaces using IPv4 and IPv6 addressing.

3.2 Network requirements for CLM redundancy and communications within a CLM cluster

3.2.1 Communication requirements between redundant CLM deployments

The network requirements between active/standby CLM clusters for round-trip network latency between the pair of redundant servers must be limited to 100 ms.

3.2.2 Communication between nodes in a CLM cluster

In a multi-node deployment of a CLM cluster, it is recommended that the nodes within the cluster have 1Gbps of Ethernet connectivity with less than 1 ms round-trip latency.

4 CLM disk setup and partitioning

4.1 Overview

4.1.1 Purpose

This chapter describes the disk commissioning options, and lists the partitioning requirements for CLM in live network environments.

4.1.2 Contents

CLM disk deployment

4.2 Introduction

4.2.1 Disk commissioning methods

A physical or virtual station that hosts CLM software requires a specific disk partitioning scheme. You can create the required disk partitions on the station:

- during the deployment of a RHEL OS disk image, as described in 4.2.2 “Disk-image deployment for CLM” (p. 28)
- after a manual RHEL OS installation, using the partitioning scheme in “Disk partitioning for live deployments” (p. 40)

Supported file systems

For OS-deployed disk partitions such as /, /home, /tmp, /opt, and /var, the CLM supports ext4 or XFS, with the following exception:

- The /var/log and /var/log/audit partitions require XFS.

For CLM application partitions such as /opt/nsp and child partitions, ext4 is required, with the following exception:

- /extra, which can be mounted as ext4 or XFS

Additional disk configuration

Regardless of the disk deployment method, each partition created after the OS installation requires the additional configuration described in 4.6 “To configure and mount a CLM disk partition” (p. 38).

4.2.2 Disk-image deployment for CLM

The CLM uses NSP OEM disk images for deploying a RHEL OS instance. Images are available in the following formats:

- qcow2
- OVA

The following image is available in each format:

- CLM deployer host / CLM cluster VM image

Note: RHEL image deployment is authorized only for CLM software installation, and not for the installation of any other Nokia or third-party product.

A RHEL disk image:

- contains only the RHEL OS
- has all required and optional OS packages to support CLM software deployment
- does not include any product-specific packages or application files
- has SELinux enabled in permissive mode

qcow2 image deployment

For qcow2 image deployment, see 4.3 “To deploy a RHEL qcow2 disk image” (p. 28).

OVA image deployment

Deploying the CLM on VMWare has the following requirements.

- You must ensure that each non-LVM partition is mounted using the partition UUID, rather than the block device name; see 4.4 “To configure disk partitions using device UUIDs” (p. 35) for information.
- If you use cloud-init to set the IP address of a CLM VM on VMWare, you must perform 4.5 “To apply the VMware cloud-init workaround” (p. 37) before you attempt to install any CLM software on the VM.

For general OVA image deployment information, see the documentation for your virtualization environment.

Note: For a CLM deployer host or CLM cluster VM deployed using the OVA image, it is strongly recommended that you mount the /opt directory on a separate hard disk that has sufficient capacity to allow for future expansion.

4.3 To deploy a RHEL qcow2 disk image

4.3.1 Purpose

Perform this procedure to deploy a qcow2 disk image on a station that is to host CLM software.

Note: A leading # symbol in a command represents the root user prompt, and is not to be included in the command.

4.3.2 Steps

Check host station OS compatibility

1

Check the *CLM Release Notice* to ensure that the OS version of the host station supports the creation of VMs at the RHEL version that the CLM requires.

2

Log in to the VM host station as the root user.

3

If the host station OS version supports CLM VM creation, enter the following; otherwise, update the host OS version as required:

```
# osinfo-query os | grep rhel | grep -v - ↵
```

A list of supported RHEL variants is listed, for example:

```
rhel7.8 | Red Hat Enterprise Linux 7.8 | 7.8 | http://redhat.
com/rhel/7.8
rhel7.9 | Red Hat Enterprise Linux 7.9 | 7.9 | http://redhat.
com/rhel/7.9
rhel8.0 | Red Hat Enterprise Linux 8.0 | 8.0 | http://redhat.
com/rhel/8.0
rhel8.1 | Red Hat Enterprise Linux 8.1 | 8.1 | http://redhat.
com/rhel/8.1
rhel8.2 | Red Hat Enterprise Linux 8.2 | 8.2 | http://redhat.
com/rhel/8.2
```

4

Record the appropriate RHEL version number in the left column, which is one of the following:

- the version that matches the CLM-supported RHEL version, if listed
- the version that is less than but closest to the supported RHEL version; in the output example, the version to record is 8.2, as the CLM supports a higher RHEL version that is not listed

Prepare required images

5

Log in to the host station as the root user.

6

Download one of the following files from the NSP downloads page in the OEM_Images subdirectory on the Nokia Support portal to a local directory on the station:

- NSP_K8S_PLATFORM_RHEL8_*yy_mm*.qcow2—for CLM deployer host or CLM cluster VM

where *yy_mm* represents the year and month of issue

7

Open a console window.

8

Enter the following:

```
# dnf -y install virt-install libguestfs-tools ↵
```

9

For each VM that you require, enter the following to create a raw VM disk image file:

```
# qemu-img convert -f qcow2 qcow2_file -O raw -S 0 raw_image.img ↵
```

where

qcow2_file is the name of the downloaded qcow2 file

raw_image is the name that you want to assign to the image; for example, CLM_Server_A

10

Perform one of the following:

a. If you want only one disk to contain all OS, product software, and data files on a VM, you must resize the VM disk image in accordance with Table 4-1, “Live partitioning scheme, CLM deployer host” (p. 40).

   For each one-disk VM that you require, enter the following:

   ```
   # qemu-img resize -f raw "raw_image.img" sizeG ↵
   ```

   where

   raw_image is the raw disk image name specified in Step 9

   size is the required disk size, in GBytes

b. If you want more than one disk in a VM, for example, one for the OS, and one for CLM software and data, or separate disks for specific partitions, you must create a separate raw image for each required disk. The disk size must be in accordance with Table 4-1, “Live partitioning scheme, CLM deployer host” (p. 40).

   For each separate disk image that you require, enter the following:

   ```
   # qemu-img create -f raw "raw_image.img" sizeG ↵
   ```

   where

   raw_image is the name that you want to assign to the disk image; for example, CLM_Server_A_Complete, for an image that is to contain all server partitions, or CLM_Server_A_Software, for an image that is to contain only the /opt/nsp partition

   size is the required disk size, in GBytes

11

The raw image files that you create in Step 10 are in sparse format; you must convert the image to non-sparse format, which provides optimal disk performance.

Perform the following steps for each raw disk image created in Step 10.

1. Enter the following:

   ```
   # cp --sparse=never raw_image.img non-sparse_image.img ↵
   ```

   raw_image is the name of a raw disk image created in Step 10

   non-sparse_image is the name to assign to the non-sparse image

   A *non-sparse_image*.img file is created.

2. Delete the *raw_image*.img file, which is no longer required.

Deploy VMs

12

Enter the following once for each VM to deploy the VM:

Note: One "--network bridge=*bridge_name"* entry is required for each VM interface that you intend to configure.

```
# virt-install --connect qemu:///system --ram RAM --vcpu=vCPUs -n
instance --os-type=linux --os-variant=variant --disk path="image_
name", device=disk,bus=virtio,format=raw,io=native,cache=none
--network bridge=bridge_name --import & ↵
```

where

RAM is the required amount of VM RAM, depending on whether you are creating a deployer or a CLM node, in MBytes; for example, 64 GBytes is expressed as 65536, which is 64 x 1024 MBytes

vCPUs is the required number of vCPU threads, depending on whether you are creating a deployer or a CLM node

instance is the name to assign to the VM

variant is the OS version recorded in Step 4, for example, 8.2

image_name is the name of the raw or non-sparse disk image created for the VM

bridge_name is the name of the network bridge for a VM interface

13

Enter the following to open a console session on the VM:

```
# virsh console VM ↵
```

where VM is the VM name

You are prompted for credentials.

14

Enter the following credentials:

- username—root
- password—*available from technical support*

A virtual serial console session opens on the VM.

15

Configure the RHEL OS as required for the CLM; for example:

- Plumb the required IPv4 or IPv6 addresses.
- Set the hostname.
- Update the /etc/hosts file.

16

Perform one of the following; see "Disk partitioning for live deployments" (p. 40) for the partitioning scheme.

Note: If you are using multiple disks in a VM, you must mount a parent partition before you mount any child partition. For example, you cannot mount the /var/log/audit partition before you mount the /var/log partition.

a. If you are using only one disk per VM, perform the following steps for each such VM.

1. Enter the following commands:

   ```
   # mkdir -p /extra ↵
   # mkdir -p /opt/nsp ↵
   ```

2. Use the RHEL fdisk utility to create the required sub-disks for the following directories:
   - /extra
   - /opt/nsp

   For each directory, enter the following and then respond to the prompts; use the directory size value from Table 4-1, “Live partitioning scheme, CLM deployer host” (p. 40):

   ```
   # fdisk /dev/virtual_device ↵
   ```

   where *virtual_device* is the virtual device name, for example, vda in a KVM VM

3. Enter the following to reboot the VM:

   ```
   # systemctl reboot ↵
   ```

4. After the reboot, perform one of the following.
   a. If you are using LVM, perform the following steps.
      1. Enter the following sequence of commands for each sub-disk:

      ```
      # pvcreate /dev/virtual_devicen ↵
      # vgcreate vg2 /dev/virtual_devicen ↵
      ```

      where
      virtual_device is the virtual device name, for example, vda in a KVM VM
      n is the number associated with the sub-disk
      2. Go to Step 17.
   b. If you are not using LVM, perform the following steps.
      1. Enter the following for each sub-disk:

      ```
      # mkfs fs_type -L path /dev/devicen ↵
      ```

      where
      fs_type is the file system type; see “Supported file systems” (p. 28) for partition-specific file system support
      path is the directory path associated with the sub-disk, for example, /opt/nsp
      device is the device name, for example, vda in a KVM VM
      n is the device number associated with the sub-disk
      2. Open the /etc/fstab file using a plain-text editor such as vi.
      3. Add one line in the following format for each sub-disk:

      ```
      /dev/virtual_devicen  path   fs_type   defaults        0 0
      ```

      where
      device is the device name, for example, vda in a KVM VM
      n is the number associated with the sub-disk
      path is the directory path associated with the sub-disk, for example, /opt/nsp

fs_type is the file system type; see "Supported file systems" (p. 28) for partition-specific file system support

4. Save and close the file.
5. Enter the following:

   # **mount -a** ↵

6. Go to Step 19.

b. If you specify multiple disks per VM and are using LVM, enter the following sequence of commands for each disk in each VM:

pvcreate /dev/*device* ↵

vgcreate *group* /dev/*device* ↵

where

device is the device name for the disk

group is the name to assign to the volume group, and must be unique in the VM

Configure LVM

17

Create the LVM volumes and partitions.

Perform the following steps for each disk in a VM, beginning with the parent disk partitions.

Note: If you are using multiple disks in a VM, you must mount a parent partition before you mount any child partition. For example, you cannot mount the /opt/nsp/nfmp/nebackup partition before you mount the /opt/nsp partition.

Note: The /extra partition is allocated for use as a temporary storage location for downloaded product software.

1. Enter the following to create a logical volume:

   # **lvcreate -n *volume* -L *size*G *group* /dev/*device*** ↵

   where

   volume is the name to assign to the logical volume

   size is the required volume size shown in Table 4-1, "Live partitioning scheme, CLM deployer host" (p. 40)

   group is the name to assign to the volume group, and must be unique in the VM

   device is the device name

2. Enter the following:

   # **mkdir *directory*** ↵

   where *directory* is the name of the directory to associate with the volume, for example, /opt/nsp

3. Enter the following:

   # **mkfs *fs_type* -L *directory* /dev/*group*/*volume*** ↵

   where

fs_type is the file system type; see “Supported file systems” (p. 28) for partition-specific file system support

directory is the directory associated with the volume

group is the volume group

volume is the logical volume name

4. Open the /etc/fstab file using a plain-text editor such as vi.
5. Add an entry in the following format:

   ```
   /dev/group/partition directory fs_type noatime   0 0
   ```

   where

   group is the volume group

   partition is the partition name

   directory is the associated directory path

   fs_type is the file system type; see “Supported file systems” (p. 28) for partition-specific file system support
6. Save and close the file.
7. Enter the following:

   ```
   # mount -a ↵
   ```

Perform optional security hardening

18

Optionally, for greater system security, you can remove the world permissions from RHEL compiler executable files, as described in A.1 “Resetting GCC-compiler file permissions” (p. 161).

19

Close the open console windows.

END OF STEPS

4.4 To configure disk partitions using device UUIDs

4.4.1 Purpose

If you deploy a CLM VM in a VMware environment, you must mount each non-LVM partition using the block-device UUID, and not the block-device name.

Perform this procedure to change the block-device identifier for each non-LVM partition from the device name to the device UUID.

4.4.2 Steps

1

Log in as the root user on the station that hosts the partition.

2

Open a console window.

3

Enter the following:

```
# grep /dev/sd /etc/fstab ↲
```

The devices and associated partitions are listed; a line like the following is displayed for each partition:

```
/dev/device  path    fs_type    noatime    1 2
```

where

device is the block-device name

path is the mount point, for example, /opt

fs_type is the file system type, for example, ext4 or xfs

4

Perform the following steps for each listed partition:

1. Enter the following

   ```
   # blkid | grep /dev/device ↲
   ```

   One or more lines like the following are displayed, depending on the number of partitions on the device:

   ```
   /dev/device: UUID="device_UUID" BLOCK_SIZE="4096" TYPE="fs_type"
   PARTUUID="partition_UUID"
   ```

2. Record the *device_UUID* value.

5

Open the /etc/fstab file using a plain-text editor such as vi.

6

Use the recorded *device_UUID* values to modify the fstab entries.

The fstab entry example in Step 3 changes from:

```
/dev/device  path    fs_type    noatime    1 2
```

to:

```
UUID=device_UUID  path    fs_type    noatime    1 2
```

7

Close the /etc/fstab file.

8

Close the console window.

END OF STEPS

4.5 To apply the VMware cloud-init workaround

4.5.1 Purpose

The following VMware knowledge-base article describes how the cloud-init configuration fails to persist through a reboot, and instead defaults to DHCP.

https://kb.vmware.com/s/article/71264

If you deploy a CLM VM in a VMware environment, you must apply a VMware workaround described in the article before you attempt to install any CLM software on the VM.

Perform this procedure to apply the workaround from VMware.

4.5.2 Steps

1

Log in as the root user on the station that hosts the partition.

2

Open a console window.

3

Open the following file using a text editor such as vi:

/etc/cloud/cloud.cfg

4

Add the following line:

manual_cache_clean: True

5

Save and close the file.

6

Close the console window.

END OF STEPS

4.6 To configure and mount a CLM disk partition

4.6.1 Purpose

Perform this procedure on each CLM disk partition on a station that you create after the RHEL OS installation.

Note: A leading # symbol in a command is the root user prompt, and is not to be included in the command.

4.6.2 Steps

1

Log in as the root user on the station that hosts the partition.

2

Open a console window.

3

Mount the partition; see the RHEL OS documentation for information.

4

Enter the following:

```
# tune2fs -m 0 -o +acl /dev/device ↵
```

where *device* is the name of the device associated with the partition

5

Open the /etc/fstab file using a plain-text editor such as vi.

6

Perform one of the following.

a. For a partition in a physical hardware deployment, add the following entry:

```
/dev/device   mount_point   fs_type  barrier=0,noatime    1 2
```

b. For a partition in an OpenStack VM, add the following entry:

```
/dev/device   mount_point   fs_type  noatime    1 2
```

c. For a non-LVM partition in a VMWare VM, add the following entry:

```
UUID=UUID   mount_point   fs_type  noatime    1 2
```

where

device is the name of the device associated with the partition

mount_point is the partition mount point, for example, /opt/nsp

fs_type is the file system type, for example, ext4 or xfs

UUID is the block-device UUID; see 4.4 “To configure disk partitions using device UUIDs” (p. 35) for information about obtaining a blick-device UUID

7

Optionally, in accordance with ANSSI and CIS specifications, configure the following partitions using the following mount options:

Note: Configuring the mount options is strongly recommended.

Note: If you choose to configure the options, you must do so before any CLM software is installed on the station.

Note: The /var partition options are only partially ANSSI-compliant; see the *NSP Security Hardening Guide* for CIS recommendations and the support for each.

```
/boot xfs nodev,noexec,nosuid 0 0
/home xfs nodev,noexec,nosuid 0 0
/tmp xfs nodev,noexec,nosuid 0 0
/var xfs nodev,nosuid 0 0
```

8

Optionally, to meet the CIS noexec requirement for the /var/tmp directory, add the following line to bind the directory to the /tmp partition; see the *NSP Security Hardening Guide* for information:

```
/tmp /var/tmp none bind 0 0
```

9

Save and close the /etc/fstab file.

10

Enter the following to reboot the station:

```
# systemctl reboot ↵
```

The station reboots.

END OF STEPS

Disk partitioning for live deployments

4.7 Live partitioning requirements, CLM deployer host and cluster VMs

4.7.1 Live CLM deployer host partitioning scheme

CAUTION

Service Disruption

Each disk partition described in this section must be a mounted partition, and not a symbolic link. The use of symbolic links to represent partitions is not supported.

Do not use a symbolic link to represent a CLM partition under any circumstances.

Note: See the *NSP Planning Guide* for information about the supported disk types.

The following table lists the disk partitions required for CLM deployer host VM.

Table 4-1 Live partitioning scheme, CLM deployer host

Partition	Content	Size (GBytes)
/	Root	26
/boot	Boot partition	0.5
/home	User home directories	0.5
/tmp	Temporary files	4
/opt	Registry, operating data, and software for Kubernetes installer and CLM deployer	275
/var	System data	64
/var/log	System logs	6
/var/log/audit	System audit logs	6

4.7.2 Live CLM cluster VM partitioning scheme

A CLM cluster is deployed as one VM that uses local storage.

The following table lists the disk partitions required for the deployment of a CLM cluster VM.

Table 4-2 Live partitioning scheme, CLM cluster VM

Partition	Content	Size (GBytes), by deployment profile
		Basic / Standard
/	Root	26

Table 4-2 Live partitioning scheme, CLM cluster VM (continued)

Partition	Content	Size (GBytes), by deployment profile
		Basic / Standard
/home	User home directories	0.5
/tmp	Temporary files	6
/var	System data	64
/var/log	System logs	6
/var/log/audit	System audit logs	6
/opt	CLM software, operating data, backups	600

5 RHEL OS deployment for the CLM

5.1 Overview

5.1.1 Purpose

This chapter describes the following:

- RHEL OS requirements for CLM deployment
- RHEL OS installation method

5.1.2 Contents

RHEL OS deployment for CLM

5.2 Introduction

5.2.1 Red Hat support

For customers using the RHEL OS image for guest VMs, support for the RHEL instance is available directly from Nokia, not Red Hat. For all other RHEL installations, Red Hat support must be purchased for all platforms running the RHEL server with CLM. It is strongly recommended to purchase a support package from Red Hat that provides 24x7 support.

The RHEL OS image can only be used as a guest VM hosting CLM, and not for the deployment of any other Nokia or third-party product.

5.2.2 RHEL version support

The Nokia-provided RHEL OS image is based on RHEL 8.10 and must only be used for the deployment of CLM software, and not for the deployment of any other Nokia or third-party product. VMs created from the RHEL OS image can only be updated with the Nokia-provided RHEL OS update.

Consider the following:

- The RHEL OS must be installed in English.
- The CLM does not necessarily support all functionality provided in RHEL.
- CLM does not support the requiretty option in /etc/sudoers.
- CLM requires that the server hostname is configured in the */etc/hosts* file.
- RHEL must be installed in 64-bit mode where the CLM is to be installed.

The CLM product team does not support the configuration of OS services, that is, not enabled by default in the RHEL OS image.

With the exception of documented operating system parameter changes for CLM, all other settings must be left at the RHEL default configuration.

5.2.3 Third-party applications

Applications that are not sanctioned by Nokia must not be running on any virtual instance running CLM. Nokia reserves the right to remove any applications that are suspected of causing issues from stations running CLM.

5.2.4 OS deployment methods

CAUTION

System Support Violation

You must ensure that the CLM supports each update that you apply to a RHEL OS in a CLM deployment. An automated update by a subscription manager may deploy an unsupported RHEL version that you must subsequently roll back.

In order to avoid the accidental deployment of an unsupported RHEL version on a CLM station, it is strongly recommended that you lock the supported release in your RHEL subscription manager. See 5.11 "To lock the RHEL OS version" (p. 60) for information.

Before you attempt to deploy the RHEL OS in a CLM system, you must review the *NSP and CLM Host Environment Compatibility Reference* for information about the RHEL OS support by product release and for the latest compatibility information.

Note: It is strongly recommended to install any driver or firmware update that your hardware vendor advises for RHEL.

You can install the required RHEL OS instance for CLM by:

- deploying a CLM disk image, as described in 4.2.2 "Disk-image deployment for CLM" (p. 28)
- manually, as described in "Manual RHEL OS installation for CLM" (p. 51)

Note: Deploying a CLM disk image is the recommended method.

Note: Before you deploy any CLM software in a VMware VM, you must install the latest VMware Tools software.

Note: It is strongly recommended that you verify the message digest of each image file or software bundle that you download from the Nokia Support portal. The download page includes the MD5, SHA256, and SHA512 checksums for comparison with the output of the RHEL md5sum, sha256sum, or sha512sum command. See the associated RHEL man page for command usage information.

Note: The Bash shell is the supported command shell for RHEL CLI operations.

5.2.5 Time synchronization requirement

CAUTION

Service Degradation

Some entities, for example, members of an etcd cluster, fail to trust data integrity in the presence of a time difference. Failing to closely synchronize the system clocks among components complicates troubleshooting and may cause a service outage.

Ensure that you use only the time service described in this section to synchronize the CLM components.

The system clocks of the CLM must always be closely synchronized. The RHEL chronyd service is the mandatory time-synchronization mechanism that you must engage on each CLM component during deployment.

Note: Only one time-synchronization mechanism can be active in a CLM system. Before you enable chronyd on CLM, you must ensure that no other time-synchronization mechanism, for example, the VMware Tools synchronization utility, is enabled.

5.2.6 OS security

The CLM includes various security mechanisms and system hardening options. The following topics describe established or configurable during RHEL OS installation.

RHEL 8 crypto-policy setting

The CLM provides system-wide support for a RHEL 8 crypto-policy setting of FUTURE. The setting is enabled and preconfigured on an OS instance deployed using a RHEL OS OEM image.

A manually deployed OS instance, however, requires the creation of a custom sub-policy, as described in 5.12 “ To enable the CLM crypto-policy function on a manually installed RHEL OS” (p. 61).

SELinux

CLM supports deployment on a RHEL OS that has SELinux enabled in permissive or enforcing mode.

You cannot upgrade CLM on which SELinux is enabled in enforcing mode, so must switch to permissive mode before the upgrade. Switching to SELinux enforcing mode is done only after an installation or upgrade.

Note: A RHEL disk image has SELinux enabled in permissive mode by default.

See “What is SELinux?” in the *NSP System Administrator Guide* for information about enabling and troubleshooting SELinux on the deployer host and cluster VMs, and about switching between SELinux permissive mode and enforcing mode.

Removing executable world permissions

Optionally, you can remove the world permissions from RHEL compiler executable files, as described in A.1 “Resetting GCC-compiler file permissions” (p. 161).

5.2.7 Applying OS updates

WARNING

System Failure

Attempting to apply the OS update described below on a station that is not described in this guide may result in a catastrophic failure.

You must perform the OS-update procedure only on a station whose deployment is described in the CLM Installation and Upgrade guide.

If you are upgrading the CLM in a VM created using a RHEL OS disk image, you must apply a RHEL update to the OS before you can upgrade the CLM deployer host or CLM cluster, as described in 5.3 “To apply a RHEL update to a CLM image-based OS” (p. 46).

Note: If the upgrade includes a migration to a new RHEL OS version, the update is included in the new OS image that you deploy, so you do not need to perform the procedure.

5.3 To apply a RHEL update to a CLM image-based OS

5.3.1 Purpose

WARNING

System Failure

Attempting to apply the OS update described below on a station that is not described in this guide may result in a catastrophic failure.

You must perform the OS-update procedure only on a station whose deployment is described in the CLM Installation and Upgrade guide.

Perform this procedure to update a RHEL OS instance deployed using an RHEL OS disk image. Such an OS update may include RHEL patches or security enhancements, and is typically applied as part of a CLM system upgrade.

Note: The procedure applies only to a RHEL OS instance deployed using an RHEL OS disk image, and is not to be performed on a manually deployed OS.

Note: Upgrading CLM requires the latest available update for the installed RHEL version.

Applying an OS update

In order to apply an OS update, you must shut down either the deployer or cluster node, depending on which one you are updating. During an upgrade, you are directed to shut down a component before you apply an OS update.

CAUTION

Network Visibility Loss

Applying a RHEL OS update requires the shutdown of the entity receiving the update, and may cause a temporary loss of network visibility, depending on the deployment.

You must perform the procedure only during a scheduled maintenance period.

5.3.2 Steps

1

Log in as the root user on the station that hosts the OS.

2

Open a console window.

3

If the station is a CLM deployer host, correct the node_exporter user ID, if required.

1. Enter the following:

   # **id -u node_exporter** ↵

   The node_exporter user ID is displayed.

2. If the user ID is 1000, enter the following sequence of commands:

   # **systemctl stop node_exporter.service** ↵

   # **userdel -r node_exporter** ↵

4

In order to apply the OS update on a CLM deployer host or CLM cluster VM, the RHEL user named nsp requires user ID 1000; otherwise, the update fails.

If ID 1000 is assigned to a user other than nsp, make the ID available to the nsp user, for example, by doing one of the following:

- deleting the user
- using the RHEL usermod command to change the ID of the other user

5

Stop the CLM software on the CLM cluster, as required.

6

Enter the following:

mkdir -p /opt/OSUpdate ↵

7

Download the following compressed file for the new CLM release to the /opt/OSUpdate directory:

NSP_RHEL*n*_OEM_UPDATE_*yy_mm*.tar.gz

where

n is the major release of the RHEL version that you are updating, for example, 8

yy_mm is the issue date of the OS update

8

Enter the following:

cd /opt/OSUpdate ↵

9

Enter the following to expand the downloaded file:

```
# tar -zxvf NSP_RHELn_OEM_UPDATE_yy_mm.tar.gz ↵
```

The update files are extracted to the following directory:

/opt/OSUpdate/*R_r*-RHEL*V.v-yy.mm.dd*

where

R_r is the CLM release that introduces the OS update

V.v is the RHEL version, for example, 8.6

yy.mm.dd is the issue date of the OS update

10

Enter the following:

```
# cd R_r-RHELV.v-yy.mm.dd ↵
```

11

Enter the following to perform the OS update:

```
# ./yum_update.sh ↵
```

12

If the station is a CLM deployer host and you have deleted the node_exporter user in Step 3, enter the following sequence of commands:

```
# useradd -s /sbin/nologin -U node_exporter ↵
# systemctl start node_exporter.service ↵
```

13

CAUTION

Misconfiguration Risk

Performing the procedure on a CLM station running CLM Release 22.11 or earlier may have undesirable effects that include restricted system access.

You must perform the procedure only on a CLM Release 23.4 or later station.

Optionally, to align with OS-hardening best practices, as defined by the Center for Information Security, or CIS, you can change the default login umask on a RHEL OS instance that hosts a CLM deployer host or CLM cluster node to restrict file and directory access for non-root users.

To set the default RHEL login umask to 0027, perform the following steps.

1. Back up the following files to a secure location on a station outside the management network for safekeeping:
   - /etc/bashrc
   - /etc/csh.cshrc
   - /etc/login.defs

- /etc/profile

2. Enter the following:

   ```
   # sed -i 's/^\([[:space:]]*\)\(umask\|UMASK\)[[:space:]][[:space:]]
   *[0-9][0-9][0-9]/\1\2 027/'  /etc/bashrc /etc/csh.cshrc
   /etc/login.defs /etc/profile ↵
   ```

3. Log out.
4. Log in as the root user.
5. Enter the following:

   ```
   # umask ↵
   ```

   The current umask value is displayed.
6. Verify that the umask value is 0027.

14

Enter the following:

```
# systemctl reboot ↵
```

The station reboots.

15

Close the console window.

END OF STEPS

Manual RHEL OS installation for CLM

5.4 Overview

5.4.1 Purpose

This section describes the manual rollout of a RHEL OS instance for use in a CLM deployment.

5.4.2 Contents

5.5 Manually installing the RHEL OS for the CLM

5.5.1 RHEL OS installation requirements

CAUTION

Upgrade Failure

The CLM system locale must remain unchanged after the initial system installation; otherwise, a system upgrade fails.

Ensure that you set the system locale on a component only before CLM software installation, and not afterward.

CAUTION

Risk of excessive resource consumption

The RHEL gnome desktop may consume excessive memory and result in system performance degradation.

The CLM does not require the gnome desktop, which is provided for customer and support convenience. It is recommended that you disable the gnome desktop in each RHEL OS instance in a CLM deployment if you do not require the gnome desktop.

You can stop the gnome desktop using the following command as the root user:

```
systemctl stop gdm ↲
```

To disable the gnome desktop so that it does not start after a reboot, enter the following as the root user:

```
systemctl disable gdm ↲
```

CAUTION

Deployment Failure

CLM software deployment may fail if the RHEL OS configuration includes a parameter setting that the CLM does not support.

Each RHEL OS configuration setting for a CLM entity must remain at the default unless otherwise specified in the CLM documentation.

Each CLM VM or physical station requires a specifically configured base OS environment and set of RHEL OS packages that are described in this section.

Note: A manually installed RHEL OS instance must be established as described in this section; otherwise, CLM deployment on the OS fails.

Note: After you successfully install a RHEL OS instance, you can optionally install one or more packages listed in 5.10 “Optional RHEL OS packages” (p. 59).

5.5.2 Installing the OS packages

The procedures and examples in the CLM documentation use the RHEL dnf utility to orchestrate RHEL OS package installation and management.

If an OS package has dependencies on any additional packages that are not listed in the package documentation, the dnf utility installs the packages to resolve the dependencies.

dnf installs packages from RHEL ISO images or package repositories. A package repository is one of the following:

- local—created during RHEL OS installation
- Internet-based—accessible by registering with the Red Hat Network

See the RHEL documentation for information about setting up a dnf repository.

Note: dnf uses the RHEL rpm utility, which requires hardware driver files in binary format. If the RHEL driver files provided by your server hardware vendor are in source rpm format, you may need to install additional packages in order to compile the files into binary format. See the station hardware documentation for information.

Using dnf

You can use one dnf command to install or uninstall multiple OS packages. If you do not specify the -y option shown in the command examples below, dnf prompts you before installing or uninstalling each package.

The package installation syntax is:

```
dnf -y install package_1 package_2 ... package_n ↲
```

The package uninstallation syntax is:

```
dnf -y remove package_1 package_2 ... package_n ↲
```

5.6 Workflow for manual RHEL OS installation

5.6.1 Purpose

The following is the sequence of high-level actions required to install an instance of the RHEL OS for use in a CLM system.

5.6.2 Stages

1

Using the RHEL installer, choose “Minimal Install” as the Software Selection for the OS.

2

Prevent the accidental deployment of an unsupported RHEL version, perform 5.11 “To lock the RHEL OS version” (p. 60).

3

Install the required OS package set.

For a CLM deployer host or CLM cluster member, install the packages listed in 5.7 “Required RHEL OS packages for CLM container elements” (p. 54).

4

Remove specific OS packages that are installed by default but not required.

a. For a CLM deployer host or CLM cluster member, remove the packages listed in 5.8 “RHEL OS packages to remove from CLM container elements” (p. 58).

b. For an entity that is deployed outside the container environment, remove the packages listed in .

5

Perform any additional package configuration described in 5.9 “Special OS requirements” (p. 59), as required.

6

Perform 5.12 “ To enable the CLM crypto-policy function on a manually installed RHEL OS” (p. 61) to secure the OS using a RHEL crypto-policy setting.

Note: You must enable the crypto-policy function before you attempt to install any CLM software on a station.

7

Perform 5.13 “To set the default Python version” (p. 63) to configure the default Python language version to the version required by the CLM.

8

On a CLM deployer host or CLM cluster station, perform 5.14 “To create the nsp user on a manually installed CLM cluster RHEL OS” (p. 63) to create the RHEL nsp user.

9

Perform 5.15 “To disable the RHEL firewalld service” (p. 64) to ensure that the RHEL firewalld service is inactive during deployment.

10

Optionally, for greater OS security, perform 5.17 “To disable RHEL user namespaces” (p. 66) to disable the use of RHEL user namespaces.

5.7 Required RHEL OS packages for CLM container elements

5.7.1 OS packages for CLM deployer host or cluster member

The OS for a CLM deployer host or a member node in a CLM cluster requires the RHEL OS package set listed in Table 5-1, “Required OS packages, CLM container element” (p. 55). A listed package is available from the RHEL BaseOS repository, RHEL AppStream repository, or the RHEL ISO disk image.

To facilitate the package installation, you can paste the following command block in a CLI:

Note: Specific versions of some packages are required, as described in 5.9 “Special OS requirements” (p. 59).

```
dnf -y install @base aide.x86_64 autofs.x86_64 bpftool.x86_64 c-ares.x86_64
dnf -y install cloud-utils-growpart.noarch
dnf -y install container-selinux.noarch copy-jdk-configs.noarch conntrack-tools.x86_64
dnf -y install createrepo_c.x86_64 cups-client.x86_64 cups-libs.x86_64 dialog.x86_64
dnf -y install elfutils.x86_64 fio.x86_64 flac-libs.x86_64 ftp.x86_64 gc.x86_64
dnf -y install gtk2.x86_64 guile.x86_64 haproxy.x86_64 hdparm.x86_64 hyphen-en.noarch
dnf -y install ipvsadm.x86_64 irqbalance.x86_64 javapackages-tools.noarch
```

```
dnf -y install keepalived.x86_64 libkadm5.x86_64 libnetfilter_cthelper.x86_64
dnf -y install libnetfilter_cttimeout.x86_64 libnetfilter_queue.x86_64 libquadmath.x86_64
dnf -y install libselinux-devel.x86_64 libverto-libevent.x86_64 lksctp-tools.x86_64
dnf -y install lshw.x86_64 lsof.x86_64 man mcelog.x86_64 net-snmp.x86_64
dnf -y install net-snmp-utils.x86_64 network-scripts.x86_64 nfs-utils.x86_64
dnf -y install nspr.x86_64 nss-softokn.x86_64 nss-softokn-freebl.x86_64 nss-util.x86_64
dnf -y install ntpstat.noarch openssh.x86_64 openssh-askpass.x86_64
dnf -y install openssh-clients.x86_64 openssh-server.x86_64 policycoreutils.x86_64
dnf -y install pcsc-lite-libs.x86_64 procps python3 python3-babel.noarch
dnf -y install python3-jinja2.noarch python3-jmespath.noarch python3-jsonpatch.noarch
dnf -y install python3-jsonpointer.noarch python3-ldb.x86_64 python3-libselinux.x86_64
dnf -y install python3-markupsafe.x86_64 python3-netaddr.noarch python3-oauthlib.noarch
dnf -y install python3-prettytable.noarch python3-pytz.noarch python3-tdb.x86_64
dnf -y install python3-urllib3.noarch redhat-lsb-core.x86_64
dnf -y install redhat-lsb-submod-security.x86_64 rng-tools.x86_64 rsync.x86_64
dnf -y install selinux-policy-devel.noarch selinux-policy-doc.noarch
dnf -y install setroubleshoot-server.x86_64 setools-console.x86_64 sshpass.x86_64
dnf -y install socat.x86_64 tcpdump.x86_64 tzdata-java.noarch unzip.x86_64 which.x86_64
dnf -y install zip.x86_64
```

Table 5-1 Required OS packages, CLM container element

Package	Description
@base	Roles and playbooks to deploy FreeIPA servers, replicas and client
aide.x86_64	Intrusion detection environment
autofs.x86_64	Libraries for avahi run-time use
bpftool.x86_64	A 2D graphics library
c-ares.x86_64	A library that performs asynchronous DNS operations
cloud-utils-growpart.noarch	Script for growing a partition
conntrack-tools.x86_64	Userspace tools for interacting with the Connection Tracking System
container-selinux.noarch	SELinux policies for container runtimes
copy-jdk-configs.noarch	JDKs configuration files copier
createrepo_c.x86_64	Creates a common metadata repository
cups-client.x86_64	CUPS printing system - client programs
cups-libs.x86_64	CUPS printing system - libraries
dialog.x86_64	A utility for creating TTY dialog boxes
elfutils.x86_64	A collection of utilities and DSOs to handle ELF files and DWARF data
fio.x86_64	Multithreaded IO generation tool
flac-libs.x86_64	Libraries for the Free Lossless Audio Codec
ftp.x86_64	The standard UNIX FTP client

Table 5-1 Required OS packages, CLM container element (continued)

Package	Description
gc.x86_64	A garbage collector for C and C++
gtk2.x86_64	GTK+ graphical user interface library
guile.x86_64	A GNU implementation of Scheme for application extensibility
haproxy.x86_64	HAProxy reverse proxy for high availability environments
hdparm.x86_64	A utility for displaying and/or setting hard disk parameters
hyphen-en.noarch	English hyphenation rules
ipvsadm.x86_64	Utility to administer the Linux Virtual Server
irqbalance.x86_64	IRQ balancing daemon
javapackages-tools.noarch	Macros and scripts for Java packaging support
keepalived.x86_64	High Availability monitor built upon LVS, VRRP and service pollers
libkadm5.x86_64	Kerberos 5 Administrative libraries
libnetfilter_cthelper.x86_64	User-space infrastructure for connection tracking helpers
libnetfilter_cttimeout.x86_64	Timeout policy tuning for Netfilter/conntrack Fedora Rawhide for x86_64
libnetfilter_queue.x86_64	Netfilter queue userspace library, Fedora Rawhide for x86_64
libquadmath.x86_64	GCC __float128 shared support library
libselinux-devel.x86_64	Header files and libraries used to build SELinux
libverto-libevent.x86_64	libevent module for libverto
lksctp-tools.x86_64	User-space access to Linux Kernel SCTP
lshw.x86_64	Hardware lister
lsof.x86_64	A utility that lists open files on a Linux/UNIX system
man	Linux kernel and C library user-space interface documentation
mcelog.x86_64	Tool to translate x86-64 CPU Machine Check Exception date
net-snmp.x86_64	A collection of SNMP protocol tools and libraries
net-snmp-utils.x86_64	Network management utilities using SNMP, from the NET-SNMP project
network-scripts.x86_64	Legacy scripts for manipulating of network devices
nfs-utils.x86_64	NFS utilities and supporting clients and daemons for the kernel NFS server
nspr.x86_64	Netscape Portable Runtime
nss-softokn.x86_64	Network Security Services Softoken Module
nss-softokn-freebl.x86_64	Freebl library for the Network Security Services
nss-util.x86_64	Network Security Services Utilities Library
ntpstat.noarch	Utility to print NTP synchronization status

Table 5-1 Required OS packages, CLM container element (continued)

Package	Description
openssh.x86_64	An open source implementation of SSH protocol version 2
openssh-askpass.x86_64	A passphrase dialog for OpenSSH and X
openssh-clients.x86_64	An open source SSH client application
openssh-server.x86_64	An open source SSH server daemon
policycoreutils.x86_64	SELinux policy core utilities
pcsc-lite-libs.x86_64	PC/SC Lite libraries
procps	System and process monitoring utilities
python3	Interpreter of the Python programming language
python3-babel.noarch	Library for internationalizing Python applications
python3-jinja2.noarch	General purpose template engine for Python 3
python3-jmespath.noarch	JSON Matching Expressions
python3-jsonpatch.noarch	Applying JSON Patches in Python 3
python3-jsonpointer.noarch	Resolve JSON Pointers in Python
python3-ldb.x86_64	Python bindings for the LDB library
python3-libselinux.x86_64	SELinux python 3 bindings for libselinux Fedora Rawhide for x86_64
python3-markupsafe.x86_64	Implements a XML/HTML/XHTML Markup safe string for Python 3
python3-netaddr.noarch	A pure Python network address representation and manipulation library
python3-oauthlib.noarch	An implementation of the OAuth request-signing login
python3-prettytable.noarch	Python library to display tabular data in tables
python3-pytz.noarch	World Timezone Definitions for Python
python3-tdb.x86_64	Python3 bindings for the Tdb library
python3-urllib3.noarch	Python3 HTTP module with connection pooling and file POST abilities.
redhat-lsb-core.x86_64	LSB Core module support
redhat-lsb-submod-security.x86_64	LSB Security sub-module support
rng-tools.x86_64	Random number generator related utilities
rsync.x86_64	A program for synchronizing files over a network
selinux-policy-devel.noarch	SELinux policy devel
selinux-policy-doc.noarch	SELinux policy documentation
setroubleshoot-server.x86_64	SELinux troubleshooting
setools-console.x86_64	Policy analysis command-line tools for SELinux
sshpass.x86_64	Non-interactive SSH authentication utility

Table 5-1 Required OS packages, CLM container element (continued)

Package	Description
socat.x86_64	Bidirectional data relay between two data channels 'netcat++'
tcpdump.x86_64	A network traffic monitoring tool
tzdata-java.noarch	Timezone data for Java
unzip.x86_64	A utility for unpacking zip files
which.x86_64	Displays where a particular program in your path is located
zip.x86_64	A file compression and packaging utility compatible with PKZIP

5.8 RHEL OS packages to remove from CLM container elements

5.8.1 OS packages to remove from CLM deployer host or cluster member

After you install the required RHEL base environment and OS packages for a CLM deployer host or a member node in a CLM cluster, you must remove the packages listed in Table 5-2, “RHEL packages to remove, CLM container element” (p. 58). The packages are installed by default, but not required by the CLM.

To facilitate the package removal, you can paste the following command block in a CLI:

```
dnf -y remove esmtp.x86_64 gnupg2-smime iwl6000-firmware
dnf -y remove libconfig.x86_64 libesmtp.x86_64
dnf -y remove liblockfile.x86_64 libstoragemgmt.x86_64
dnf -y remove nmap-ncat python3-beautifulsoup4.noarch
dnf -y remove python3-cssselect.noarch python3-html5lib
dnf -y remove python3-webencodings realmd timedatex trousers
dnf -y remove trousers-lib
```

Table 5-2 RHEL packages to remove, CLM container element

Package	Description
esmtp.x86_64	User-configurable send-only Mail Transfer Agent
gnupg2-smime	CMS encryption and signing tool and smart card support for GnuPG
iwl6000-firmware	Firmware for Intel(R) Wireless WiFi Link 6000 AGN Adapter
libconfig.x86_64	C/C++ configuration file library
libesmtp.x86_64	SMTP client library
liblockfile.x86_64	This implements a number of functions found in -lmail on SysV system
libstoragemgmt.x86_64	Storage array management library Fedora Rawhide for aarch64
nmap-ncat	Nmap's Netcat replacement Fedora Rawhide for aarch64
python3-beautifulsoup4.noarch	HTML/XML parser for quick-turnaround applications like screen-scraping
python3-cssselect.noarch	Serializer for literal Python expressions

Table 5-2 RHEL packages to remove, CLM container element (continued)

Package	Description
python3-html5lib	A python based HTML parser/tokenizer
python3-webencodings	Documentation for python-webencodings
realmd	Kerberos realm enrollment service Fedora Rawhide for aarch64
timedatex	D-Bus service for system clock and RTC settings CentOS 8-stream BaseOS for aarch64
trousers	TSS (TCG Software Stack) access daemon for a TPM chip OpenSuSE Leap 15.4 for aarch64
trousers-lib	TrouSerS libtspi library Fedora Rawhide for aarch64

5.9 Special OS requirements

5.9.1 Required OS package versions

The CLM requires the minimum version or later of each RHEL package listed in Table 5-3, “Required RHEL OS package versions” (p. 59). If an installed package version is lower than the minimum, you must upgrade the package to at least the minimum,

You can use the following commands to check the current package versions:

```
# dnf list installed | grep container-selinux ↲
```

```
# dnf list installed | grep tzdata-java ↲
```

Note: If the minimum version is not installed, CLM deployment may fail.

As required, use the following CLI commands to upgrade one or more packages:

```
dnf -y install container-selinux
```

```
dnf -y install tzdata-java
```

Table 5-3 Required RHEL OS package versions

Package	Minimum required version
container-selinux	2.191.0-1
tzdata-java	2023d

5.10 Optional RHEL OS packages

5.10.1 Optional RHEL OS packages

Table 5-4, “Optional RHEL OS packages” (p. 60) lists the packages that you can opt to install for any entity without affecting CLM operation.

Note: The packages are not included in the RHEL OS disk images.

To facilitate the package installation, you can paste the following command block in a CLI:

```
dnf -y install @gnome-desktop @legacy-x @base-x firefox.x86_64
dnf -y install tigervnc-server.x86_64
```

Table 5-4 Optional RHEL OS packages

Package	Description
@gnome-desktop	Gnome package group
@legacy-x	Legacy X package group
@base-x	X11 package group
firefox.x86_64	Mozilla Firefox web browser
tigervnc-server.x86_64	Server for the VNC remote display system

5.11 To lock the RHEL OS version

5.11.1 Purpose

If the RHEL OS version is not locked in your RHEL subscription manager, a RHEL OS version not yet supported by the CLM may be accidentally deployed during the OS installation.

Perform the following procedure to prevent the deployment of an unsupported RHEL OS version on a CLM station.

5.11.2 Steps

1

It is recommended that you lock the RHEL OS at the latest version that the CLM supports. The *Host Environment Compatibility Reference for NSP and CLM* includes a support matrix for CLM and RHEL version compatibility.

Use the support matrix to identify the latest supported RHEL version for the CLM release.

2

Log in as the root user on the station that hosts the OS.

3

Open a console window.

4

Enter the following:

subscription-manager release --set=*version* ↵

where *version* is the RHEL version to lock, for example, 8.10

The version is locked.

5

Close the console window.

END OF STEPS

5.12 To enable the CLM crypto-policy function on a manually installed RHEL OS

5.12.1 Purpose

Perform this procedure to configure the minimum RSA cryptography key length for the RHEL crypto-policy function on a CLM OS instance.

Note: The crypto-policy function is not enabled on the OS until you perform the procedure.

Note: You must perform the procedure before you install any CLM software on the OS.

5.12.2 Steps

1

Log in as the root user on the station that hosts the OS.

2

Open a console window.

3

Enter the following:

```
# cat /etc/crypto-policies/config ↵
```

The following crypto-policy setting is displayed:

```
DEFAULT
```

4

Create the following file using a plain-text editor such as vi:

/etc/crypto-policies/policies/modules/NSP_CUSTOM_RSA_SIZE.pmod

5

Edit the file to read as follows:

```
min_rsa_size = 2048
```

6

Save and close the file.

7

Enter the following:

```
# cat NSP_CUSTOM_RSA_SIZE.pmod ↵
```

The edited file is displayed.

8

Ensure that the file reads as follows:

```
min_rsa_size = 2048
```

9

Enter the following:

```
# update-crypto-policies --set FUTURE:NSP_CUSTOM_RSA_SIZE ↵
```

Messages like the following are displayed.

```
Setting system policy to FUTURE:NSP_CUSTOM_RSA_SIZE
Note: System-wide crypto policies are applied on application start-up.
It is recommended to restart the system for the change of policies to
fully take place.
```

If the output is as shown, the crypto-policy configuration is successful.

10

If the crypto-policy configuration succeeds, enter the following:

```
# systemctl reboot ↵
```

The station reboots.

11

Log in as the root user.

12

Open a console window.

13

Enter the following:

```
# cat /etc/crypto-policies/config ↵
```

The crypto-policy setting is displayed.

14

Verify that the crypto-policy setting reads as follows:

```
FUTURE:NSP_CUSTOM_RSA_SIZE
```

15

Close the console window.

END OF STEPS

5.13 To set the default Python version

5.13.1 Purpose

Perform the procedure to set the default version of the Python language that the CLM requires. The setting is required after a manual RHEL OS installation for a CLM component, and before any CLM software is installed on the station.

Note: You must perform the procedure on each station in a CLM deployment that has a manually installed RHEL OS.

Note: You do not need to perform the procedure on a RHEL OS deployed using the CLM qcow2 OS image, as the OS image includes the setting.

5.13.2 Steps

1

Log in as the root user on the station.

2

Open a console window.

3

Enter the following:

```
# alternatives --set python /usr/bin/python3 ↵
```

The setting is applied.

4

Close the console window.

END OF STEPS

5.14 To create the nsp user on a manually installed CLM cluster RHEL OS

5.14.1 Purpose

Perform the procedure to create the Linux nsp user as the owner of files and processes on a station that are otherwise associated by default with user ID 1000.

The procedure applies only to a manually installed RHEL OS on the following:

- CLM deployer host
- CLM cluster node

You must perform the procedure on each such CLM station after a manual RHEL OS installation, and before any CLM software is installed on the station.

Note: You do not need to perform the procedure on a RHEL OS deployed using the CLM qcow2 OS image, as the OS image includes the setting.

5.14.2 Steps

1

Log in as the root user on the station.

2

Open a console window.

3

Enter the following:

```
# useradd --shell /sbin/nologin --no-create-home --uid 1000
--user-group nsp ↵
```

The nsp user account is created in the nsp user group.

4

Close the console window.

END OF STEPS

5.15 To disable the RHEL firewalld service

5.15.1 Purpose

You must stop and disable the RHEL firewalld service on each station in a CLM deployment before you attempt to install a CLM component on a station.

Perform this procedure to disable firewalld on a CLM station.

5.15.2 Steps

1

Log in as the root user on the station.

2

Open a console window.

3

Enter the following:

```
# systemctl stop firewalld ↲
```

The firewalld service stops.

4

Enter the following:

```
# systemctl disable firewalld ↲
```

The firewalld service is disabled.

5

Close the console window.

END OF STEPS

5.16 To set the default umask to 0027

5.16.1 Purpose

To align with OS-hardening best practices, as defined by the Center for Information Security, or CIS, you can change the default login umask on a RHEL OS instance that hosts a CLM deployer host, CLM cluster node, or CLM entity deployed outside the CLM cluster, to restrict file and directory access for non-root users.

Perform this procedure to set the default login umask on a RHEL OS instance to 0027.

CAUTION

Misconfiguration Risk

Performing the procedure on a CLM station running CLM Release 22.11 or earlier may have undesirable effects that include restricted system access.

You must perform the procedure only on a CLM Release 23.4 or later station.

5.16.2 Steps

1

Log in as the root user on the station.

2

Open a console window.

3

Back up the following files to a secure location on a station outside the management network for safekeeping:

- /etc/bashrc
- /etc/csh.cshrc
- /etc/login.defs
- /etc/profile

4

Enter the following:

```
# sed -i 's/^\([[:space:]]*\)\(umask\|UMASK\)[[:space:]][[:space:]]*
[0-9][0-9][0-9]/\1\2 027/'  /etc/bashrc /etc/csh.cshrc /etc/login.defs
/etc/profile ↵
```

5

Log out.

6

Log in as the root user.

7

Enter the following:

```
# umask ↵
```

The current umask value is displayed.

8

Verify that the umask value is 0027.

9

Close the console window.

END OF STEPS

5.17 To disable RHEL user namespaces

5.17.1 Purpose

Current and future RHEL OS vulnerabilities may be mitigated by disabling namespaces for RHEL users.

Note: Disabling RHEL namespaces is described in a STIG recommendation.

For greater OS security, perform the following steps to disable the use of namespaces by any RHEL user on any CLM station.

5.17.2 Steps

1

Log in as the root user on the station.

2

Open a console window.

3

Enter the following to display the current namespace setting:

```
# cat /proc/sys/user/max_user_namespaces ↵
```

A numeric value is displayed.

4

RHEL namespaces are enabled if the value is greater than zero.

If RHEL namespaces are enabled, perform the following steps.

1. Enter the following:

   ```
   # echo user.max_user_namespaces=0 >/etc/sysctl.d/97-nsp-rhel-oem.conf ↵
   ```

2. Enter the following:

   ```
   # cat /etc/sysctl.d/97-nsp-rhel-oem.conf ↵
   ```

   The /etc/sysctl.d/97-nsp-rhel-oem.conf file content is listed.

3. If the file content is anything other than the following, return to substep 1 to correct any input error and recreate the file.

   ```
   user.max_user_namespaces=0
   ```

4. Enter the following:

   ```
   # sudo sysctl --system ↵
   ```

   Namespace usage is disabled.

5

Enter the following to verify that namespaces are disabled:

```
# cat /proc/sys/user/max_user_namespaces ↵
```

A numeric value is displayed.

6

If the value is not 0, contact technical support for assistance.

7

Close the console window.

END OF STEPS

6 Configuring CLM security

6.1 Overview

6.1.1 Purpose

This chapter describes fundamental CLM system security elements, and includes important information that you must consider as you deploy a CLM system or entity.

For additional CLM security information such as post-deployment configuration, see the *NSP Security Hardening Guide*.

6.1.2 Contents

CLM system security

6.2 Introduction

6.2.1 Overview

This section provides general information about platform security for a deployment of CLM. Recommendations in this section apply to CLM except where indicated.

The CLM implements a number of safeguards to ensure the protection of private data. Additional information can be found in the NSP Data Privacy section of the *NSP System Architecture Guide*.

6.3 Securing the CLM

6.3.1 Overview

Nokia recommends performing the following steps to achieve station security for the CLM:

- Install the latest recommended patch cluster for RHEL. For customers using the Nokia-provided RHEL OS image, only the RHEL OS update can be used for applying OS patches. For customer-sourced and manually deployed RHEL OS instances, the patches must be obtained from Red Hat.
- CLM has no ingress or egress requirements to access the public internet and should be isolated with properly configured firewalls.
- Implement traffic management policies to control access to ports on CLM systems, as detailed in this section
- Use TLS certificates with strong hashing algorithms.
- Enforce minimum password requirements and password renewal policies on user accounts that access the CLM.
- Configure a warning message in the Launchpad Security Statement.
- OAUTH2 authentication module provides login protection mechanisms to prevent denial of service attacks, lockout users for consecutive failed logins and configure maximum sessions for GUI and OSS users. See 6.9.1 “OAUTH2 user authentication” (p. 77) for details.
- When using custom TLS certificates for CLM deployment, ensure that the server private key file is protected when not in use by nsp configurator.
- Optional: Revoke world permissions from compiler executables. See Appendix A, “Removing world permissions from compiler executables”.

See the *NSP Security Hardening Guide* for RHEL OS compliance with CIS benchmarks. The supported CIS benchmark best practices are already implemented on RHEL OS images.

6.4 Operating system security for CLM stations

6.4.1 RHEL patches

For customer-sourced and manually deployed RHEL OS instances, Nokia supports customers applying RHEL patches provided by Red Hat which include security fixes as well as functional fixes. If a patch is found to be incompatible with the CLM, the patch may need to be removed until a solution to the incompatibility is provided by Red Hat or Nokia. See the *CLM Release Notice* for up-to-date information about the recommended RHEL maintenance update and patch levels.

For customers using the Nokia-provided RHEL OS images, only the RHEL OS update provided by Nokia can be applied.

6.4.2 Platform hardening

Additional efforts to secure the system could impact CLM operation or future upgrades of the product. Customers must perform some level of basic testing to validate that additional platform hardening does not impact the operation of the CLM. The CLM Product Group makes no commitment to make the CLM compatible with a customer's hardening requirements.

6.5 CLM Kubernetes Platform Communications

6.5.1 Overview

The tables provided in this section identify the listening ports on a deployer node and on worker nodes for a CLM cluster deployment. These ports must be accessible between the deployer and worker nodes within a CLM deployment. The SSH ports on all servers must be accessible by a system administrator for installation and maintenance functions.

Table 6-1 Ports used by deployer node

Default port(s)	Type	Application
22	TCP	SSH
111	TCP	rpcbind
443	TCP	HTTPS
6443	TCP	kubernetes API server
8443	TCP	helm repo, container registry
9100	TCP	node exporter
10250	TCP	kubelet metrics
30000-32767	TCP	kube proxy

Table 6-2 Ports used by worker nodes

Default Port(s)	Type	Application
22	TCP	sshd
53	TCP	node-cache
111	TCP	rpcbind
179	TCP	bird
2375	TCP	containerd
2379	TCP	etcd
2380	TCP	etcd
6443	TCP	kubernetes API server
7472	TCP	metalLB (metrics)
7946	TCP	metalLB (load balancer communications)
8081	TCP	nginx
9100	TCP	node exporter
9253	TCP	node cache
9254	TCP	node cache
9353	TCP	node-cache
10250	TCP	kubelet metrics
10251	TCP	kube-scheduler
10256	TCP	kube-proxy
10257	TCP	kube controller
10259	TCP	kube scheduler
30000-32767	TCP	kube-proxy

6.6 CLM platform user accounts

6.6.1 Default user accounts

CLM system deployment, configuration, and support require low-level operations by a privileged user such as root. The root user typically performs most platform-level and CLM system-level operations.

For greater CLM system security, you can prevent root-user access to CLM system-level deployment and configuration commands, and enable access to only specific privileged non-root users. See 6.7 “Restricting root-user system access” (p. 73) for information.

CLM also requires the user accounts described below for CLM runtime functions and UI-based system administration. Each account is created during system deployment.

6.6.2 RHEL runtime user, nsp

Internal CLM functions require a local ‘nsp’ RHEL user group and an ‘nsp’ user in the group. The nsp user owns the local CLM processes, and has administrative control over CLM runtime functions.

The nsp user home directory is the CLM installation base directory, /opt/nsp. The initial nsp user password is randomly generated, and must be changed during the initial login attempt.

Note: CLM software uninstallation does not remove the nsp user account, user group, or home directory.

6.6.3 CLM UI administrator account, admin

By default, a new CLM system has one user account for CLM UI access, the ‘admin’ account. The admin user has full CLM UI and application administration privileges, and access to all CLM UI functions.

Note: Only the admin user, or a user with equivalent administrative privileges, has access to CLM administrative functions in the UI.

6.7 Restricting root-user system access

6.7.1 Description

To restrict root-user access to CLM system commands, you can create privileged users that can execute only specific CLM system commands. If a user other than the privileged non-root user attempts to execute a restricted command, the command fails and an error message is displayed.

Restricted root-user access:

- meets an important CIS security benchmark
- assigns sudo privileges for only the required commands per user
- ensures that any configuration or control actions are traceable to a specific user

You can restrict root-user access on the CLM deployer host and cluster VMs.

Note: Client delegate servers do not support restricted root access.

CLM commands named in the documentation or used in a CLM script support execution as a non-root user under the conditions described in this topic.

Note: The root user creates the privileged users, but is not required afterward for CLM system deployment, administration, or support. A superuser account is created for support use.

6.7.2 Defining privileged CLM user accounts

Restricted CLM root access requires multiple privileged non-root user accounts for different types of system access. You can enlist technical support to create and enable a default set of privileged non-root user accounts, or use accounts that you create.

You can also opt to disable only remote root access to the CLM cluster VMs from the CLM deployer host, as described in 6.7.5 “Disabling remote root access” (p. 75).

Note: Each user must belong to the local ‘nsp’ user group.

The following directory on a CLM deployer host contains CLM specific example sudoers files that list the restricted commands and functions:

/opt/nsp/NSP-CN-DEP-*release-ID*/NSP-CN-*release-ID*/tools/sudoers

Privileged CLM cluster accounts

The account types required include the following, depending on the CLM entity:

- CLM admin user—on CLM deployer host and cluster VMs; performs deployment operations
- CLM ansible user—on CLM cluster VMs only; for software rollout operations; can be disabled when system is operational
- CLM super user—for general access with root-level privileges; in wheel group

Note: The CLM super user is required by customer support teams for troubleshooting the system, so must always be enabled.

6.7.3 Restricted commands

The following are the restricted commands and scripts, by element; the list is not exhaustive, but summarizes the commands used in the CLM documentation.

- **general**—SELinux commands
- **CLM deployer host and cluster VMs**
  - Kubernetes deployment—kubectl, nerdctl, helm
  - CLM deployment—nspregistryctl, nspk8sctl, nspdeployerctl

6.7.4 Users and commands in procedure steps

For simplicity, CLM procedure steps that involve restricted root access are minimally formatted, as shown in the following examples.

Note: Depending on the sudoers configuration, the operator may be prompted for the password of the privileged user when executing a command.

CLM example

1. Log in as the root or CLM admin user on the station.

   The step text indicates that if root access is restricted, you must log in as the CLM admin user.

2. Enter the following:

   ```
   # cd /opt/nsp/nsp-k8s-deployer-release-ID/bin ↵
   ```

Basic commands like 'cd' and 'ls' are entered as shown.

3. Enter the following:

   # **./nspk8sctl config -l** ↵

   You must enter a CLM command such as 'nspk8sctl' using 'sudo', so the command you type is:

   # **sudo ./nspk8sctl config -l** ↵

4. As the root or CLM admin user, enter the following:

   # **kubectl get pods -A** ↵

   If root access is restricted, the CLM admin user must execute the command and use 'sudo'., so the command you type is:

   # sudo **kubectl get pods -A** ↵

6.7.5 Disabling remote root access

To comply with the CIS RHEL benchmark "Ensure SSH root login is disabled", you can also disable root-user SSH, and by extension, SCP, access on the CLM deployer host and cluster VMs.

To disable remote root access, during CLM cluster deployment you specify a designated non-root user and security key in the cluster deployment configuration. The user requires sudo privileges on each CLM station.

6.8 HTTPS Strict-Transport Security (HSTS)

6.8.1 Enabling HSTS for the CLM

CAUTION

Security Risk

Without HSTS, a browser that receives an invalid TLS certificate displays a warning that the user can circumvent. If HSTS is enabled, however, the browser blocks CLM access, and does not allow the user to circumvent the warning.

If HSTS is enabled, the system administrator must monitor and manage the TLS certificates carefully to ensure that, for example, a certificate is not expired, self-signed, or signed by an unknown CA.

HSTS is a mechanism that returns a header with specific instructions for any browser that attempts to connect using HTTP. The HSTS header instructs the browser to access the site using HTTPS instead of HTTP for all subsequent connections to the site or any child domain.

When HSTS is enabled, all CLM web interfaces are protected.

Note: HSTS is disabled by default in a CLM system, and can be enabled only during system installation; you cannot enable HSTS in a deployed CLM system.

6.8.2 HSTS TLS certificate management

In addition to ensuring that the current TLS certificate recognized by HSTS is not expired or nearing expiry, the same level of security must be applied to a certificate that replaces an expired certificate.

For example, if HSTS is enabled in the CLM, and you then change from a trusted root-CA-signed certificate to a self-signed certificate, browsers that attempt to connect to the CLM may prevent access because the new certificate is not trusted.

6.8.3 Configuring HSTS

You can enable HSTS during CLM system installation in the **hsts** section of the CLM configuration file.

Note: HSTS is disabled by default.

CLM user authentication

6.9 Overview

6.9.1 OAUTH2 user authentication

The CLM employs OAUTH2 user authentication, which is based on Keycloak open-source identity and access management using the OAuth 2.0 protocol.

OAUTH2 supports local user authentication, and authentication using external authentication agents such as RADIUS, LDAP/S, and TACACS+ servers. Windows Active Directory is also supported.

CLM user authentication includes configurable mechanisms that guard against unwanted system access by maintaining strict control over repeated login attempts. See 6.10.2 “CLM login protection” (p. 78) for information.

The CLM also supports the forwarding of user activity log events, as described in 6.11 “CLM user activity logging” (p. 79).

See 9.3 “Configuring single sign-on” (p. 119) for specific OAUTH2 configuration information.

6.9.2 Kafka user authentication

The CLM Kafka subsystem reports events to internal clients and systems. The internal Kafka communication is secured using TLS.

Kafka authentication for internal clients is configurable in the **nsp**—**modules**—**nspos**—**kafka** section of the CLM configuration file.

The following parameter in the CLM configuration file enables or disables the support for the deprecated TLS versions:

- tlsv1ProtocolsEnabled

Internal Kafka client authentication

Kafka authentication for internal clients is based on two-way mTLS, rather than CLM user credentials.

The following parameter in the CLM configuration file enables or disables the support:

- internalClientAuth

The following parameter in the CLM configuration file enables or disables the support:

- configure nspos mtls-kafka-enabled

6.10 CLM user authentication functions

6.10.1 Local and remote user management

CLM Users and Security is the interface for local user creation and administration.

CLM username convention

To be valid for CLM access, a local or remote authentication source username must consist of only lowercase characters, for example, johndoe. The convention is enforced as follows:

- You cannot create a local username that includes an uppercase character.
- The CLM cannot authenticate a remote authentication username that includes uppercase characters. During CLM login, the username is converted to lowercase before authentication is attempted.

CLM remote authentication

You can define multiple LDAP, RADIUS, and TACACS+ remote authentication sources.

The CLM first attempts to verify a set of user credentials against the local user database. If the user account is not found, or lacks the correct credentials, the credentials are verified against the remote authentication sources.

Note: During a remote user login attempt, if the remote authentication source returns a user group that does not exist in the CLM, the user is denied CLM access.

Note: Remote authentication servers can communicate with the CLM using IPv4 or IPv6.

CLM remote authentication has the following characteristics.

- You can define multiple servers for each type of remote authentication source, for example, two LDAP servers.
- RADIUS and TACACS+ authentication sources cannot be used in the same OAUTH2 deployment.
- LDAP immediately follows local user authentication in priority, and is always above RADIUS or TACACS+.
- RADIUS or TACACS+ is always the last authentication source to be tried.

6.10.2 CLM login protection

CLM Users and Security provides functions for temporarily or permanently locking out users for login failures. Login failure management is configured during CLM deployment.

You cannot enable both temporary and permanent user lockout. If user lockout is to be enforced, only one mechanism can be active at any time.

Note: Temporary user lockout is enabled by default.

User login failures and permanent lockout

The CLM can automatically lock out a user after a specified number of consecutive login failures. The user is prevented from logging in until an administrator unsuspends the user account. The user lockout applies only to local CLM users, and not to users defined in external authentication sources.

User login throttling and temporary lockout

A user that reaches a specified number of consecutive failed login attempts can be temporarily disabled for a specified wait interval. During the wait interval, further login attempts by the user are

not processed. After the wait interval, the CLM processes new login attempts by the user. If the user login attempts continue to fail, the login attempts are subsequently disabled for incrementally longer periods, up to a configurable maximum.

Note: Temporary lockout applies to local and external authentication source users.

6.11 CLM user activity logging

6.11.1 Local logging

The CLM logs user activity events such as login, logout, and system configuration changes.

6.11.2 Log forwarding

The CLM supports the forwarding of user log entries to a third-party processing system. You can enable log forwarding using the following CLM configuration file parameters:

- "NSP Platform - Logging and Monitoring" installation option
- one option in the **nsp**—**modules**—**logging**—**forwarding**—**applicationLogs** section:
  - openSearch
  - splunk
  - syslog

CLM Transport Layer Security (TLS)

6.12 CLM TLS overview

6.12.1 Introduction

CLM communication is secured using Transport Layer Security, or TLS. The TLS-secured elements of a CLM deployment include the following:

- Kubernetes infrastructure—Kubernetes registry, interfaces between CLM deployer host and CLM cluster VMs
- internal CLM subsystem and service endpoints
- external CLM cluster endpoints for communication with:
  - UI and API clients
  - other management systems and target servers

Note: A CLM system upgrade preserves the TLS artifacts.

6.12.2 CLM TLS certificates

The CLM TLS certificates include the following:

Note: You specify the issuer and server TLS artifacts during CLM cluster deployment. The Kubernetes certificates are automatically generated, and require no configuration.

- Kubernetes infrastructure certificates, applied to:
  - Kubernetes registry
  - CLM deployer host
  - CLM cluster control plane
- CLM issuer certificates, applied to:
  - internal CLM service and subsystem endpoints
  - external-facing CLM application endpoints
- CLM server certificates, applied to:
  - CLM cluster gateway for client access
  - mediation interfaces

CLM issuer certificates

CLM issuer certificates are CA signing certificates that provide session-level security for the internal and external-facing CLM application endpoints.

CLM server certificates

CLM server certificates secure the ingress gateway for external client access. Using a custom server certificate has special requirements, as described in 6.13.2 “Using custom TLS certificates” (p. 81).

6.12.3 Kubernetes secrets

The TLS artifacts of a CLM cluster are stored in Kubernetes secrets to prevent the exposure of sensitive security information. The '**nspdeployerctl secret**' command on a CLM deployer host facilitates secret creation, update, and replacement, and includes other functions such as secret backup and restore. You can also use the command to display the secret content.

The CLM system deployment procedures include steps for creating the required secrets, and "What is NSP TLS administration?" in the *NSP System Administrator Guide* describes post-deployment TLS certificate and Kubernetes secret management.

See 6.13 "CLM TLS configuration requirements" (p. 80) and "CLM TLS configuration procedures" (p. 83) for information about creating the required TLS artifacts for a CLM deployment.

6.12.4 CLM PKI-server service

A CLM cluster hosts a PKI-server service that uses the CA certificates from the internal and external issuer secrets to sign certificates. The service uses an access control list based on the CLM cluster configuration; consequently, the service responds only to certificate requests from known addresses in the nsp-config.yml file of the local cluster.

6.13 CLM TLS configuration requirements

6.13.1 TLS deployment options

During CLM system deployment, you can choose to use one or more TLS certificates that the CLM generates and signs, or can provide one or more of your own signed certificates, which are called custom certificates.

The CLM cluster software package provides a PKI server that can be used to simplify the TLS certificate distribution to NSP components.

Note: If the CLM clusters use advertised hostnames, the SAN field of a CLM server certificate must include the advertised hostname of each CLM cluster.

Note: The private key and certificate files for a CLM deployment must be in unencrypted PEM format.

6.13.2 Using custom TLS certificates

A custom TLS certificate for the CLM must:

- be CA-signed
- be a 2048-bit RSA key
- include serverAuth in the ExtendedKeyUsages field

Note: A custom CLM server certificate must be unique to a CLM cluster.

See 6.14 "To generate custom TLS certificate files for the CLM" (p. 83) for configuration information.

6.13.3 Using intermediate signing certificates

The CLM PKI service can act as an intermediate CA. The supported intermediate key type is a 4096-bit RSA key.

The required and recommended key extensions are the following:

- Required:
  - CA:TRUE
  - certificate sign key usage
  - chained .pem file in which the CLM Intermediate cert is first in the chain, followed by the intermediate certificates, and ending with the root certificate
- Recommended:
  - path length = 0, which signifies that the PKI server can sign only end-entity certificates

For example:

Note: Required restrictions are in **boldface** type:

X509v3 Basic Constraints: critical

CA:TRUE, pathlen:0

X509v3 Key Usage: critical

Digital Signature, **Certificate Sign**, CRL Sign

6.13.4 TLS version and cipher support

By default, only TLS 1.2 is enabled. However, external systems such as OSS clients may use deprecated TLS versions. For CLM compatibility with such systems, you can enable older TLS versions.

The following parameter in the CLM configuration file enables or disables the support for the deprecated TLS versions:

- tlsv1ProtocolsEnabled

CLM TLS configuration procedures

6.14 To generate custom TLS certificate files for the CLM

6.14.1 Purpose

Perform this procedure to generate a set of TLS key and certificate files to provide as security artifacts in a CLM deployment.

The locations of the custom TLS files that the procedure generates are the required inputs to the '**nspdeployerctl secret**' prompts when you create or update the custom TLS secret, as shown below:

- **tls.key**=*customKey*
- **tls.cert**=*customCert*
- **ca.crt**=*customCaCert*

where

customKey is the location of the private server key file extracted in Step 9

customCert—location of one of the following:

- server.pem file obtained in Step 7
- server-chained.pem obtained fn Step 10, if using intermediate CA

customCaCert—location of CA.pem key file obtained in Step 7

In addition, the CA key file location must be specified as shown below in the **tls** section of the CLM cluster configuration file:

customCaCert—location of CA.pem key file obtained in Step 7

Using the Java keytool utility

The procedure uses the Java keytool utility, which is included in each Java Development Kit, or JDK, and Java Runtime Environment, or JRE. The keytool utility is described on the Oracle website.

You can run the keytool command from any directory on a CLM deployer host or cluster VM. If the CLM is not yet deployed, ensure that the keytool utility on the station that you use is from the supported Java version specified in the *NSP Planning Guide*.

Note: The keytool utility that you use must be from the Java version that the CLM uses.

Note: You require root user privileges to run the keytool command.

6.14.2 Steps

Note: The Bash shell is the supported command shell for RHEL CLI operations.

Note: A leading # character in a command line represents the root user prompt, and is not to be included in a typed command.

Generate TLS certificate

1

Log in as the root user on the station that hosts the keytool utility.

2

Open a console window.

3

Generate a keystore file that contains the certificate.

Note: A file path in the *keystore_file* value, or in the name of any file generated in a subsequent step, must not include /opt/nsp/os. If you do not include a path, the file is generated in the current working directory, which must not be below /opt/nsp/os.

Note: You must enclose a password that contains a special character in single quotation marks; for example:
-keypass 'Mypa$$word' -storepass 'Mypa$$word'

```
# keytool -genkeypair -alias alias -keyalg RSA -keypass password
-storepass password -keystore keystore_file -validity days -dname
"CN=server_name, OU=org_unit, O=org_name, L=locality, S=state,
C=country" -ext bc=ca:true -ext san=DNS:DNS_name↵
```

where

alias is a case-insensitive alias that is required for subsequent keytool operations

password is the password for the key and keystore

Note: The keypass and storepass passwords must be identical.

keystore_file is the name of the keystore file to generate

days is the number of days for which the certificate is to be valid

server_name is the common name or hostname of the server

org_unit is a department or division name

org_name is a company name

locality is a city name

state is a state or region name

country is a country code, for example, US

DNS_name is the DNS-resolvable server hostname or FQDN

Note: A custom certificate is unique to a CLM cluster.

Note: In a CLM deployment that uses separate client and internal interfaces, and does not use mTLS, the SAN field must also include the internal advertised address of the CLM

cluster. The address to include is one of the following in the **platform**—**ingressApplications**—**ingressController** section of the config.yml file on the local CLM deployer host:
In the **internalAddresses** subsection, if configured, otherwise, in the **clientAddresses** subsection:

- if configured, the **advertised** value
- otherwise, the **virtualIp** value

4

Record the *alias* and *password* values that you specify.

Export certificate

5

Enter the following to export the certificate from the keystore to a certificate file:

Note: You must enclose a password that contains a special character in single quotation marks; for example:
-storepass 'Mypa$$word'

```
# keytool -export -alias alias -keystore keystore_file -storepass
password -file certificate_file ↵
```

where

alias is the alias specified during keystore creation

keystore_file is the source keystore file, for example, /opt/samserver.keystore

password is the keystore password

certificate_file is the name of the certificate file to generate

Generate and submit CSR

6

Generate a certificate signing request, or CSR.

1. Enter the following:

   ```
   # path/keytool -certreq -alias alias -keystore keystore_file -file
   CSR_file -storetype JKS -ext san=DNS:DNS_name -ext
   ExtendedKeyUsage=serverAuth,clientAuth ↵
   ```

   where

   alias is the keystore alias

   keystore_file is the keystore file generated in Step 3

   CSR_file is the name of the CSR file to generate

   DNS_name is the DNS-resolvable server hostname or FQDN

   Note: Multiple server SAN entries are separated using semicolons; for example,

san=DNS:*DNS_name_1*;DNS:*DNS_name_2*

The following prompt is displayed:

```
Enter keystore password:
```

2. Enter the keystore password. The following prompt is displayed:

   ```
   Enter key password for alias
   ```

3. Enter the key password. The utility generates a CSR file.

7

Send the CSR file to a CA for authentication. The CA returns the following certificate files that contain a trusted root certificate in a hierarchical certificate chain.

- server.pem—public server key
- CA.pem—public CA key

Generate CLM cluster TLS artifacts

8

Enter the following to convert the keystore to PKCS12 format:

```
# keytool -importkeystore -noprompt -srckeystore keystore_file
-destkeystore file_name.pkcs12 -deststoretype PKCS12 -deststorepass
storepass -destkeypass keypass -srcstorepass storepass -srckeypass
keypass -alias alias ↵
```

where

alias is the keystore alias

keystore_file is the keystore file generated in Step 3

file_name is the name of the new keystore file in PKCS12 format

keypass is the keystore password

storepass is the truststore password

9

Enter the following to extract the private key from the PKCS12 keystore to a file:

```
# openssl pkcs12 -in file_name.pkcs12 -passin pass:keypass -nodes
-nocerts -descert -out private_key.key ↵
```

where

file_name is the name of the keystore file in PKCS12 format

private_key is the name to assign to the private key file

10

If you are using an intermediate CA, enter the following to generate the chained server .pem file:

```
# cat server.pem ca-chained.pem > server-chained.pem ↵
```

Note: The certificate order is important; the server certificate must be first in the chain of certificates in the file in order for the CLM installer to read the certificates correctly.

11

Close the console window.

End of steps

6.15 To suppress security warnings in CLM browser sessions

6.15.1 Purpose

The following steps describe how to prevent the repeated display of security warnings in a browser that connects to the CLM using a private-CA-signed or self-signed TLS certificate.

Note: You do not need to perform the procedure if the certificate is signed by a public root CA, which is trusted by default.

6.15.2 Steps

1

Transfer the CLM ca.pem certificate file to each client host on which you want to suppress the browser warnings

2

Perform one of the following.

a. Import the certificate to the certificate store of a client OS.

Note: This method suppresses the display of CLM-related security warnings for all client browsers.

Perform the appropriate procedure in the OS documentation to import the certificate; specify the certificate file as the certificate source.

Note: Such a procedure varies by OS type and version.

b. Import the certificate to the certificate store of a client browser.

Perform the appropriate procedure in the browser documentation to import the certificate; specify the certificate file as the certificate source.

Note: Such a procedure varies by browser type and version.

3

Open a browser session and verify that the CLM opens without the display of security warnings.

End of steps

7 CLM deployment with multiple network interfaces and IP addresses

7.1 Support for multiple network interfaces

7.1.1 Introduction

CLM supports configuring different network interfaces to handle the following types of traffic in a multi-homed system.

- A client network interface can be used for connecting users to CLM GUI and to connect external OSS systems to CLM.
- An internal network interface can be used to handle traffic between CLM systems that does not need to be accessed by external systems or with managed network elements. Internal traffic includes, but is not limited to, resync of network topology information, security communications, application registration and data synchronization between redundant entities.
- A mediation network interface can be used to communicate with network elements (provisioning, NE database backups, monitoring, operations, etc).

7.1.2 Support and limitations of CLM deployer nodes and CLM clusters

A CLM cluster can be configured with network interfaces for client traffic, for internal network management traffic, and for managed network traffic. In a multi-node CLM cluster, each node must have the same number of interfaces. Each node of the CLM cluster must have mediation network connectivity to all managed NE devices

A deployer node must have connectivity to all CLM cluster nodes on the internal network. The deployer node must also have access to the client network.

There is no requirement on the CLM cluster to use the first network interface (eg. eth0, bge0) to communicate with client applications.

Additional network interfaces can be configured on the CLM cluster, at the customer's discretion, for other operations such as archiving database backups or activity logs.

When using custom TLS certificates in a multi-network configuration, the CLM server certificate requires the IP address or hostname or FQDN of the client network interface (or virtual IP) and the IP address or hostname or FQDN of the internal network interface (or virtual IP) in the certificate SAN field.

7.1.3 Multi-interface support in IPv4 and IPv6 networks

The CLM cluster can use IPv4 or IPv6 addressing on the client, internal and mediation network interfaces. In addition to the limitations and restrictions documented in section 4.2.1, the following conditions apply:

- The CLM cluster can only use IPv4 or IPv6 communications on the client network interface and

on the internal network interface. The system network interfaces can have both IPv4 and IPv6 addresses assigned, but CLM communications on those interfaces can only use IPv4 or IPv6.

- The CLM cluster mediation interface supports IPv4 only, IPv6 only and IPv4 and IPv6 simultaneously. When CLM is configured with IPv4 and IPv6 mediation simultaneously, the CLM must have a dedicated mediation interface not shared with client and internal network communications.
- In a CLM deployment with separate network interfaces for client and internal communications, the client and internal networks must both use IPv4 or IPv6 addressing. Example, client communications on IPv4 and internal communications on IPv6 is not supported.

7.1.4 Multi-interface CLM deployment and traffic management policies

The following table summarizes the traffic management policies for a CLM cluster deployment by each network or network interface.

Table 7-1 CLM cluster communications by network interface

Network description	Permitted communications
Client network	Client communications Kafka communications on ports 9092, 9093, 9094, 9192, 9193, 9194
Internal network	All communications between cluster nodes and deployer node Kafka communications on ports 9292, 9293, 9294
Mediation network	Mediation communications

The CLM cluster can communicate with some external elements on any network interface, including

- remote authentication servers (LDAP, RADIUS, TACACS)
- syslog server
- email server

Each node in an NSP cluster must allow the same traffic on each network interface.

Part II: CLM system deployment

Overview

Purpose

This part of the *CLM Installation and Upgrade Guide* describes the CLM deployment environment, and provides information about performing various CLM system deployment operations.

Contents

8 CLM deployment basics

8.1 Overview

8.1.1 Purpose

This chapter describes the container-based CLM environment and fundamental CLM deployment considerations. Also included is information about upgrading the deployment environment.

The following contains important information about the initial platform setup in advance of deploying the CLM container environment:

- Part I: “Getting started”—platform configuration, for example, preparing disk partitions, installing the RHEL OS, and implementing platform security

8.1.2 Contents

CLM system elements

8.2 Introduction

8.2.1 nspOS resource base

The nspOS, which is the common resource base of a CLM system, is deployed in a CLM cluster of one VM in the Kubernetes container environment. A disaster-recovery, or DR, deployment, consists of matching CLM clusters in geographically separate data centers. A DR deployment is also called a geo-redundant deployment.

8.2.2 Time synchronization

CAUTION

Service Degradation

Some entities, for example, members of an etcd cluster, fail to trust data integrity in the presence of a time difference. Failing to closely synchronize the system clocks among components complicates troubleshooting and may cause a service outage.

Ensure that you use only the time service described in this section to synchronize the CLM components.

The system clocks of the CLM components must always be closely synchronized. The RHEL chronyd service is the mandatory time-synchronization mechanism that you must engage on each CLM component during deployment.

Note: Only one time-synchronization mechanism can be active in a CLM system. Before you enable chronyd on a CLM component, you must ensure that no other time-synchronization mechanism, for example, the VMware Tools synchronization utility, is enabled.

8.3 Containerized CLM cluster

8.3.1 Introduction

The system elements described in the following topics are common to all CLM deployments.

You can use a disk image to instantiate CLM elements and functions as VMs, as described in 4.2.2 “Disk-image deployment for CLM” (p. 28).

Note: The RHEL OS image deployment steps in a procedure are specific to a RHEL KVM environment; however, alternative virtualization environments are supported, as described in 2.1.3 “KVM virtualization” (p. 15).

To deploy a RHEL OS image in an environment other than KVM, you must observe the requirements in 2.1.4 “OpenStack virtualization” (p. 16) or 2.1.5 “VMware virtualization” (p. 17)for the environment, and perform the deployment as directed in the virtualization product documentation.

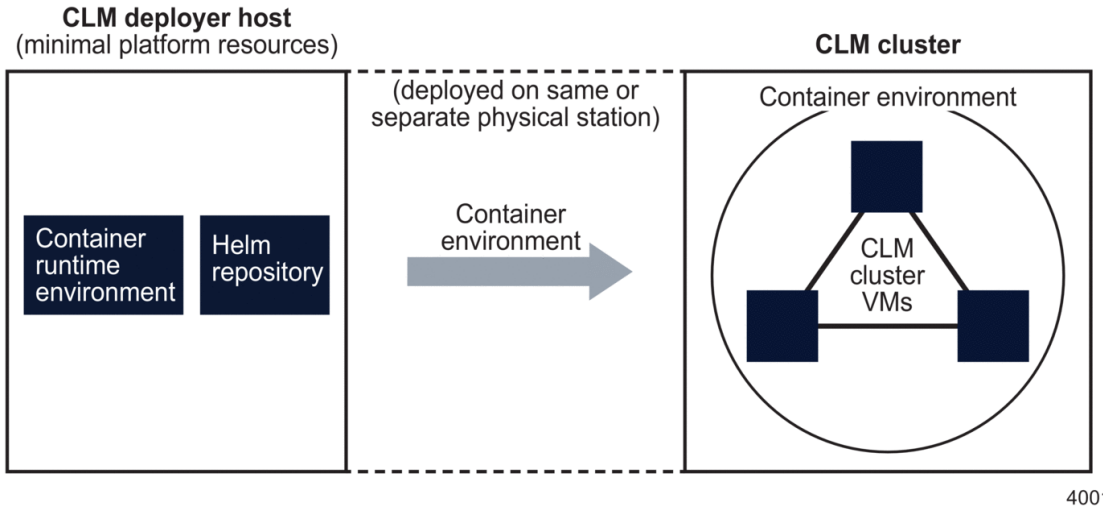
The main elements of a CLM system are the following:

- CLM deployer host; deploys containerization environment for CLM cluster; one CLM deployer host is required in each data center of a DR deployment
- CLM cluster VMs; host the main CLM functions

The CLM deployer host and CLM cluster VM can be hosted on one physical station, or on separate stations. Figure 8-1, “CLM deployer host and CLM cluster” (p. 94) shows a standalone deployment on one physical station that hosts the CLM deployer host and CLM cluster VM.

Note: Communication between the CLM deployer host and the CLM cluster VM is IPv4-only.

Figure 8-1 CLM deployer host and CLM cluster



8.3.2 CLM deployer host

CAUTION

Service degredation risk

The CLM deployer host is a crucial element of a CLM cluster deployment that must remain reachable by each CLM cluster VM after the initial deployment; otherwise, cluster recovery in the event of a failure may be compromised.

You must ensure that the CLM deployer host remains operational and reachable by the CLM cluster VMs at all times.

The CLM deployer host holds the required container image repository and Helm repository, and pushes a containerization environment for the CLM cluster.

8.3.3 CLM cluster VMs

The CLM software runs on the CLM cluster VMs, and is load-balanced among the VMs, depending on the deployment configuration.

CLM cluster host

The CLM cluster host is a specific CLM cluster VM from which CLM configuration operations are performed. The VM requires direct network access to the CLM deployer host.

Note: The CLM cluster host is functionally no different from the other VMs in a CLM cluster; the VM is merely the designated cluster member for performing cluster and software management actions. The designation helps to prevent operator confusion, and simplifies the logging of maintenance actions.

CLM deployment infrastructure

8.4 Kubernetes deployment environment

8.4.1 Introduction

The CLM software can be deployed only in a supported version of Kubernetes environment. A CLM software bundle includes the latest supported version as of the CLM release date; a CLM system installation uses the bundled Kubernetes version.

A CLM system may be compatible with a Kubernetes version that is released after the CLM deployment. In such a case, you can upgrade the Kubernetes deployment environment without upgrading the CLM software, as described in 8.4.2 “Upgrading Kubernetes” (p. 97).

8.4.2 Upgrading Kubernetes

A CLM system upgrade typically includes an upgrade of the Kubernetes deployment environment. Each CLM system upgrade procedure has a link to 8.5 “To upgrade the CLM Kubernetes environment” (p. 98), which describes the upgrade process for a standalone or DR CLM deployment.

Supported upgrade schemes

A Kubernetes upgrade must be version-sequential, which means that you cannot upgrade directly from version A to version C and skip version B; you must first upgrade from version A to version B, and only then can you upgrade to version C.

Rather than performing a series of sequential Kubernetes upgrades, you can choose to uninstall your current Kubernetes environment and then install the new version.

Note: Performing a series of sequential upgrades preserves the CLM cluster data in the Kubernetes etcd database, while uninstalling Kubernetes deletes the etcd data.

Off-cycle upgrades

CAUTION

Misconfiguration Risk

The Kubernetes upgrade procedure in this guide is specific to an upgrade to the Kubernetes version introduced in the CLM release of this guide, and cannot be used for upgrading to a later version. Using the Kubernetes upgrade procedure in this guide to upgrade to a Kubernetes version issued with a later CLM release is not supported, and may result in system misconfiguration.

To upgrade to a later Kubernetes version, you must use the Kubernetes upgrade procedure in the CLM Installation and Upgrade Guide for the CLM release associated with the new Kubernetes version.

After a CLM installation or upgrade, if a new Kubernetes version is made available that your CLM release supports, you can perform an “off-cycle” Kubernetes upgrade to the new version without

upgrading the CLM software. See the *Host Environment Compatibility Reference for NSP and CLM* for information about the supported Kubernetes versions for various CLM releases.

8.5 “To upgrade the CLM Kubernetes environment” (p. 97) describes how to upgrade your Kubernetes version to the version introduced with the CLM release of the CLM release of this guide..

8.5 To upgrade the CLM Kubernetes environment

8.5.1 Purpose

Perform this procedure to upgrade the Kubernetes deployment environment in a CLM system. The procedure upgrades only the deployment infrastructure, and not the CLM software.

Note: You must upgrade Kubernetes in each CLM cluster of a DR deployment, as described in the procedure.

Note: *release-ID* in a file path has the following format:
R.r.p-rel.*version*
where
R.r.p is the CLM release, in the form *MAJOR.minor.patch*
version is a numeric value

8.5.2 Steps

Download Kubernetes upgrade bundle

1

Download the following from the NSP downloads page on the Nokia Support portal to a local station that is not part of the CLM deployment:

Note: The download takes considerable time; while the download is in progress, you may proceed to Step 2.

- NSP_CLM_K8S_DEPLOYER_*R_r*.tar.gz—software bundle for installing the registry and deploying the container environment
- associated .cksum file

where
R_r is the CLM release ID, in the form *Major_minor*

Verify CLM cluster readiness

2

Perform the following steps on each CLM cluster to verify that the cluster is fully operational.

1. Log in as the root user on the CLM cluster host.
2. Open a console window.

3. Enter the following to display the status of the CLM cluster nodes:

   ```
   # kubectl get nodes -A ↵
   ```

   The status of each cluster node is displayed.

   The CLM cluster is fully operational if the status of each node is Ready.

4. If any node is not in the Ready state, you must correct the condition; contact technical support for assistance, if required.

   Do not proceed to the next step until the issue is resolved.

5. Enter the following to display the CLM pod status:

   ```
   # kubectl get pods -A ↵
   ```

   The status of each pod is displayed.

   The CLM cluster is operational if the status of each pod is Running or Completed.

6. If any pod is not in the Running or Completed state, you must correct the condition; see the *NSP Troubleshooting Guide* for information about troubleshooting an errored pod.

Back up CLM databases

3

On the standalone CLM cluster, or the primary cluster in a DR deployment, perform “Back up the existing CLM” (p. 153).

Back up system configuration files

4

Perform the following on the CLM deployer host in each data center.

Note: In a DR deployment, you must clearly identify the source cluster of each set of backup files.

1. Log in as the root or CLM admin user.
2. Back up the Kubernetes deployer configuration file:

   /opt/nsp/nsp-k8s-deployer-*release-ID*/config/k8s-deployer.yml

3. Back up the CLM deployer configuration file:

   /opt/nsp/NSP-CN-DEP-*release-ID*/config/nsp-deployer.yml

4. Back up the CLM configuration file:

   /opt/nsp/NSP-CN-DEP-*release-ID*/NSP-CN-*release-ID*/appliedConfigs/nspConfiguratorConfigs.zip

5. Copy the backed-up files to a separate station that is not part of the CLM deployment.

Back up Kubernetes secrets

5

If the CLM is at Release 24.8 or later, back up the Kubernetes secrets, if not done as part of the database backup in Step 3.

1. Enter the following on the CLM deployer host:

   # cd /opt/nsp/NSP-CN-DEP-*old-release-ID*/bin ↵

2. Enter the following:

   # ./nspdeployerctl secret -o *backup_file* backup ↵

   where *backup_file* is the absolute path and name of the backup file to create

   As the secrets are backed up, messages like the following are displayed for each Kubernetes namespace:

   ```
   Backing up secrets to /opt/backupfile...
     Including secret namespace:ca-key-pair-external
     Including secret namespace:ca-key-pair-internal
     Including secret namespace:nsp-tls-store-pass
   ```

   When the backup is complete, the following prompt is displayed:

   ```
   Please provide an encryption password for backup_file
   enter aes-256-ctr encryption password:
   ```

3. Enter a password.

   The following prompt is displayed:

   ```
   Verifying - enter aes-256-ctr encryption password:
   ```

4. Re-enter the password.

   The backup file is encrypted using the password.

5. Record the password for use when restoring the backup.
6. Record the name of the data center associated with the backup.
7. Transfer the backup file to a secure location in a separate facility for safekeeping.

Verify checksum of downloaded file

6

It is strongly recommended that you verify the message digest of each CLM file that you download from the Nokia Support portal. The downloaded .cksum file contains checksums for comparison with the output of the RHEL md5sum, sha256sum, or sha512sum commands.

When the file download is complete, verify the file checksum.

1. Enter the following:

   # *command file* ↵

   where

   command is md5sum, sha256sum, or sha512sum

file is the name of the downloaded file

A file checksum is displayed.

2. Compare the checksum and the associated value in the .cksum file.
3. If the values do not match, the file download has failed. Retry the download, and then repeat Step 6.

Upgrade CLM registry

7

Perform Step 8 to Step 17 on the CLM deployer host in each data center, and then go to Step 18.

Note: In a DR deployment, you must perform the steps first on the CLM deployer host in the primary data center.

8

If the CLM deployer host is deployed in a VM created using a RHEL OS disk image, perform 5.3 “To apply a RHEL update to a CLM image-based OS” (p. 47).

9

Copy the downloaded NSP_CLM_K8S_DEPLOYER_*R_r*.tar.gz file to the /opt/nsp directory.

10

Expand the software bundle file.

1. Enter the following:

   ```
   # cd /opt/nsp ↵
   ```

2. Enter the following:

   ```
   # tar -zxvf NSP_CLM_K8S_DEPLOYER_R_r.tar.gz ↵
   ```

   The bundle file is expanded, and the following directories are created:
   - /opt/nsp/nsp-registry-*new-release-ID*
   - /opt/nsp/nsp-k8s-deployer-*new-release-ID*
3. After the file expansion completes successfully, enter the following to remove the bundle file, which is no longer required:

   ```
   # rm -f NSP_CLM_K8S_DEPLOYER_R_r.tar.gz ↵
   ```

11

If you are not upgrading Kubernetes from the immediately previous version supported by the CLM, but from an earlier version, you must uninstall the Kubernetes registry; otherwise, you can skip this step. See the *Host Environment Compatibility Guide for NSP and CLM* for information about Kubernetes version support.

Enter the following:

```
# /opt/nsp/nsp-registry-old-release-ID/bin/nspregistryctl uninstall ↵
```

The Kubernetes software is uninstalled.

12

Enter the following:

```
# cd /opt/nsp/nsp-registry-new-release-ID/bin ↵
```

13

Enter the following to perform the registry upgrade:

Note: During the registry upgrade, the registry may be temporarily unavailable. During such a period, a CLM pod that restarts on a new cluster node, or a pod that starts. is in the ImagePullBackOff state until the registry upgrade completes. Any such pods recover automatically after the upgrade, and no user intervention is required.

```
# ./nspregistryctl install ↵
```

14

If you did not perform Step 11 to uninstall the Kubernetes registry, go to Step 17.

15

Enter the following to import the original Kubernetes images.

```
# /opt/nsp/NSP-CN-DEP-base_load/bin/nspdeployerctl import ↵
```

where *base_load* is the initially deployed version of the installed CLM release

16

If you have applied any CLM service pack since the original deployment of the installed release, you must import the Kubernetes images from the latest applied service pack.

Enter the following to import the Kubernetes images from the latest applied service pack.

```
# /opt/nsp/NSP-CN-DEP-latest_load/bin/nspdeployerctl import ↵
```

where *latest_load* is the version of the latest applied CLM service pack

Verify CLM cluster initialization

17

When the registry upgrade is complete, verify the cluster initialization.

1. Enter the following:

   ```
   # kubectl get nodes ↵
   ```

   CLM deployer node status information like the following is displayed:

   ```
   NAME          STATUS     ROLES                    AGE      VERSION
   node_name     status     control-plane,master     xxdnnh    version
   ```

2. Verify that *status* is Ready; do not proceed to the next step otherwise.
3. Enter the following periodically to monitor the CLM cluster initialization:

```
# kubectl get pods -A ↵
```

The status of each pod is displayed.

The CLM cluster is fully operational when the status of each pod is Running or Completed.

4. If any pod fails to enter the Running or Completed state, correct the condition; see the *NSP Troubleshooting Guide* for information about troubleshooting an errored pod.

Prepare to upgrade CLM Kubernetes deployer

18

Perform Step 19 to Step 32 on the CLM deployer host in each cluster, and then go to Step 33.

Note: In a DR deployment, you can perform the steps on each CLM deployer host concurrently; the order is unimportant.

19

You must merge the current k8s-deployer.yml settings into the new k8s-deployer.yml file.

Open the following files using a plain-text editor such as vi:

- old configuration file—/opt/nsp/nsp-k8s-deployer-*old*-release-*ID*/config/k8s-deployer.yml
- new configuration file—/opt/nsp/nsp-k8s-deployer-*new*-release-*ID*/config/k8s-deployer.yml

20

Apply the settings in the old file to the same parameters in the new file.

21

Close the old k8s-deployer.yml file.

22

In the new k8s-deployer.yml file, edit the following line in the **cluster** section to read:

```
hosts: "/opt/nsp/nsp-k8s-deployer-new-release-ID/config/hosts.yml"
```

23

If you have disabled remote root access to the CLM cluster VMs,configure the following parameters in the **cluster** section, **sshAccess** subsection:

```
sshAccess:
  userName: "user"
  privateKey: "path"
```

where

user is the designated CLM ansible user

path is the SSH key path, for example, /home/*user*/.ssh/id_rsa

24

Each CLM cluster VM has a parameter block like the following in the **hosts** section; configure the parameters for each VM, as required:

```
- isIngress: value
  nodeIp: private_IP
  accessIp: public_address
  nodeName: node_name
```

where

value is true or false, and indicates whether the node acts as a load-balancer endpoint

private_IP is the VM IP address

public_IP is the public VM address; required only in a NAT environment

node_name is the VM name

25

In the following section, specify the virtual IP addresses for the CLM to use as the internal load-balancer endpoints.

Note: A single-node CLM cluster requires at least the *client_IP* address.

The addresses are the following values for CLM client, internal, and mediation access that you specify in the **platform**—**ingressApplications** section of the nsp-config.yml file during CLM cluster deployment.

In the **internalAddresses** subsection, if configured, otherwise, in the **clientAddresses** subsection:

- if configured, the **advertised** value
- otherwise, the **virtualIp** value

```
loadBalancerExternalIps:
    - client_IP
    - internal_IP
```

26

Configure the following parameter, which specifies whether dual-stack NE management is enabled:

Note: Dual-stack NE management can function only when the network environment is appropriately configured, for example:

- Only valid, non-link-local static or DHCPv6-assigned addresses are used.
- A physical or virtual IPv6 subnet is configured for IPv6 communication with the NEs.

```
enable_dual_stack_networks: value
```

where *value* must be set to true if the cluster VMs support both IPv4 and IPv6 addressing

27

Save and close the new k8s-deployer.yml file.

28

Enter the following:

```
# cd /opt/nsp/nsp-k8s-deployer-new-release-ID/bin ↵
```

29

Enter the following to create the new hosts.yml file:

```
# ./nspk8sctl config -c ↵
```

30

Enter the following to list the node entries in the new hosts.yml file:

```
# ./nspk8sctl config -l ↵
```

Output like the following example for a one-node cluster is displayed:

Note: If NAT is used in the cluster:

- The ip value is the private IP address of the cluster node.
- The access_ip value is the public address of the cluster node.
  Otherwise:
- The ip value is the private IP address of the cluster node.
- The *access_ip* value matches the ip value.

Note: The ansible_host value must match the access_ip value.

```
Existing cluster hosts configuration is:
node_1_name:
  ansible_host: 203.0.113.11
  ip: private_IP
  access_ip: public_IP
  node_labels:
    isIngress: "true"
```

31

Verify the IP addresses.

32

Enter the following to import the Kubernetes images to the repository:

```
.# ./nspk8sctl import ↵
```

The images are imported.

Stop and undeploy CLM cluster

33

Perform Step 34 to Step 36 on each CLM cluster, and then go to Step 37.

Note: In a DR deployment, you must perform the steps first on the standby cluster.

34

Perform the following steps on the CLM deployer host to preserve the existing cluster data.

1. Open the following file using a plain-text editor such as vi:

   /opt/nsp/NSP-CN-DEP-*old-release-ID*/NSP-CN-*old-release-ID*/config/nsp-config.yml
2. Edit the following line in the **platform** section, **kubernetes** subsection to read:

   ```
   deleteOnUndeploy:false
   ```
3. Save and close the file.

35

Enter the following on the CLM deployer host to undeploy the CLM cluster:

Note: If you are upgrading a standalone CLM system, or the primary CLM cluster in a DR deployment, this step marks the beginning of the network management outage associated with the upgrade.

Note: If the CLM cluster VMs do not have the required SSH key, you must include the --ask-pass argument in the command, as shown in the following example, and are subsequently prompted for the root password of each cluster member:
nspdeployerctl --ask-pass uninstall --undeploy --clean

```
# /opt/nsp/NSP-CN-DEP-old-release-ID/bin/nspdeployerctl uninstall
--undeploy --clean ↵
```

The CLM cluster is undeployed.

36

On the CLM cluster host, enter the following periodically to display the status of the Kubernetes system pods:

Note: You must not proceed to the next step until the output lists only the following:

- pods in kube-system namespace
- nsp-backup-storage pod

```
# kubectl get pods -A ↵
```

The pods are listed.

Deploy new CLM Kubernetes software

37

Perform Step 38 to the end of the procedure on each CLM cluster.

Note: In a DR deployment, you must perform the steps first on the primary cluster.

38

If the new Kubernetes version is more than one version later than the existing version, you cannot upgrade the software; instead, you must completely replace the existing version by uninstalling the software and then installing the new version.

Perform the following steps.

Note: The software replacement on a cluster takes approximately 30 minutes, and is the recommended option.

Note: If the CLM cluster VMs do not have the required SSH key, you must include the --ask-pass argument in a command, as shown in the following example, and are subsequently prompted for the root password of each cluster member:
nspk8sctl --ask-pass uninstall

1. Enter the following on the CLM deployer host:

   # **cd /opt/nsp/nsp-k8s-deployer-*old-release-ID*/bin** ↵

2. Enter the following:

   # **./nspk8sctl uninstall** ↵

   The existing Kubernetes software is uninstalled.

39

Enter the following:

cd /opt/nsp/nsp-k8s-deployer-*new-release-ID*/bin ↵

40

If the CLM is at Release 24.8 or later, restore the backed-up Kubernetes secrets.

1. Enter the following to restore the Kubernetes secrets:

   ./nspdeployerctl secret -i *backup_file* restore ↵

   where *backup_file* is the secrets backup file created earlier in the procedure

   The following prompt is displayed:

   ```
   Please provide the encryption password for /opt/backupfile
   enter aes-256-ctr decryption password:
   ```

2. Enter the password recorded during the backup creation.

   As the secrets are restored, messages like the following are displayed for each Kubernetes namespace:

   ```
   Restoring secrets from backup_file...
   ```

```
secret/ca-key-pair-external created
  Restored secret namespace:ca-key-pair-external
secret/ca-key-pair-internal created
  Restored secret namespace:ca-key-pair-internal
secret/nsp-tls-store-pass created
  Restored secret namespace:nsp-tls-store-pass
```

41

Enter the following to install or upgrade the Kubernetes software:

Note: If you do not uninstall Kubernetes in Step 38, the software is upgraded rather than installed. An upgrade takes considerable time; during the upgrade process, each cluster node is individually cordoned, drained, upgraded, and uncordoned. The operation on each node may take 15 minutes or more.

```
# ./nspk8sctl install ↵
```

The new CLM Kubernetes environment is deployed.

42

Enter the following on the CLM cluster host periodically to display the status of the Kubernetes system pods:

Note: You must not proceed to the next step until each pod STATUS reads Running or Completed.

```
# kubectl get pods -A ↵
```

The pods are listed.

43

Enter the following periodically on the CLM cluster host to display the status of the CLM cluster nodes:

Note: You must not proceed to the next step until each node STATUS reads Ready.

```
# kubectl get nodes -o wide ↵
```

The CLM cluster nodes are listed, as shown in the following example:

```
NAME    STATUS   ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP
node1   Ready    master   nd    version   int_IP   ext_IP
```

44

Update the CLM cluster configuration.

1. Open the following files using a plain-text editor such as vi:
   - old configuration file—/opt/nsp/NSP-CN-DEP-*old-release-ID*/config/nsp-deployer.yml
   - new configuration file—/opt/nsp/NSP-CN-DEP-*new-release-ID*/config/nsp-deployer.yml
2. Merge the settings from the old file into the new file.

3. Edit the following line in the new file to read:

   `hosts: "/opt/nsp/nsp-k8s-deployer-`*`new-release-ID`*`/config/hosts.yml"`

4. Save and close the new nsp-deployer.yml file.
5. Close the old nsp-deployer.yml file.

Redeploy CLM software

45

Enter the following on the CLM deployer host:

```
# cd /opt/nsp/NSP-CN-DEP-new-release-ID/bin ↵
```

46

Enter the following:

```
# ./nspdeployerctl config ↵
```

47

Enter the following:

Note: If the CLM cluster VMs do not have the required SSH key, you must include the --ask-pass argument in the command, as shown in the following example, and are subsequently prompted for the root password of each cluster member:
nspdeployerctl --ask-pass install --config --deploy

```
# ./nspdeployerctl install --config --deploy ↵
```

The CLM starts.

Verify CLM initialization

48

On the CLM cluster host, monitor and validate the CLM cluster initialization.

Note: You must not proceed to the next step until each CLM pod is operational.

1. Enter the following every few minutes:

   ```
   # kubectl get pods -A ↵
   ```

   The status of each CLM cluster pod is listed; all pods are running when the displayed STATUS value is Running or Completed.

   The nsp deployer log file is /var/log/nspdeployerctl.log.

2. If any pod fails to enter the Running or Completed state, see the *NSP Troubleshooting Guide* for information about troubleshooting an errored pod.

49

Enter the following on the CLM cluster host to display the status of the CLM cluster members:

Note: You must not proceed to the next step until each node is operational.

```
# kubectl get nodes ↵
```

The status of each node is listed; all nodes are operational when the displayed STATUS value is Ready.

The CLM Kubernetes deployer log file is /var/log/nspk8sctl.log.

Verify upgraded CLM cluster operation

50

Use a browser to open the CLM cluster URL.

51

Verify the following.

- In a DR deployment, if you specify the standby cluster address, the browser is redirected to the primary cluster address.
- The CLM sign-in page opens.
- The CLM UI opens after you sign in.

52

As required, use the CLM to monitor device discovery and to check network management functions.

Note: You do not need to perform the step on the standby CLM cluster.

Note: If you are upgrading Kubernetes in a standalone CLM cluster, or the primary CLM cluster in a DR deployment, the completed CLM cluster initialization marks the end of the network management outage.

Purge Kubernetes image files

53

Note: You must perform this and the following step only after you verify that the CLM system is operationally stable and that an upgrade rollback is not required.

Enter the following on the CLM deployer host:

```
# cd /opt/nsp/nsp-k8s-deployer-new-release-ID/bin ↵
```

54

Enter the following on the CLM deployer host:

```
# ./nspk8sctl purge-registry -e ↵
```

The images are purged.

55

Close the open console windows.

END OF STEPS

IP version support

8.6 Introduction

8.6.1 Using IPv4 and IPv6

The CLM supports IPv4 or IPv6 communication with external clients, and internally among CLM entities, in single- and multiple-interface deployments. The NEs can communicate with the CLM using IPv4, IPv6, or both concurrently.

Some restrictions may apply in a multi-interface deployment; see 8.8.4 “CLM cluster multi-interface configuration” (p. 112) for information.

Note: The use of compressed IPv6 addresses, for example, 2001:E7A3::6502:0DA8, is fully supported.

8.7 Addressing requirements

8.7.1 Client and internal addressing

The client and internal networks must be in the same IP family: IPv4 or IPv6. CLM functions communicate internally and externally using only one protocol version.

Note: Only IPv4 is supported for communication between a CLM deployer host and a CLM cluster.

8.7.2 Mediation addressing

Concurrent IPv4 and IPv6 NE mediation is supported using separate interfaces, or using one interface that supports both protocols.

Using multiple CLM interfaces

8.8 Multi-interface configuration

8.8.1 Introduction

For greater security, you can configure multiple network interfaces to segregate the different types of CLM traffic.

When the CLM uses only one network for all communication, the CLM client traffic shares the same network as the NE mediation traffic and the internal communication between CLM elements. Such a configuration may pose a considerable security risk.

You can segregate the CLM client, mediation, and internal traffic by configuring the CLM to use interfaces in separate networks for each traffic type.

8.8.2 Traffic isolation

The multi-interface implementation isolates different traffic types to one or more of the following networks:

- client—UI, browser, and other northbound client traffic
- mediation—for direct communication with NEs
- internal—for communication such as the following:
  - between CLM cluster members
  - CLM DR functions such as data replication and keepalive messaging

Using separate networks enables you to apply additional security policies. For example, the CLM PostgreSQL service is an internal service only, and the only legitimate clients are CLM components, and not northbound browser or API clients. To help secure the PostgreSQL service from unintended access, you can apply a firewall rule to block the PostgreSQL port on the client interface.

8.8.3 System conversion to multi-interface

You can convert an existing CLM system from a single-interface deployment to a multi-interface deployment.

8.8.4 CLM cluster multi-interface configuration

You specify the CLM cluster interface addresses for your deployment in the **platform**—**ingressApplications** section of the CLM configuration file. The configuration steps are described in each CLM deployment procedure, and the parameters are shown below for network planning purposes.

Note: The *client_IP* value is mandatory; the address is used for interfaces that remain unconfigured, such as in a single-interface deployment.

Note: If the client network uses IPv6, you must specify the CLM cluster hostname as the *client_IP* value.

Note: The trapForwarder addresses that you specify must differ from the *client_IP* value, even in a single-interface deployment.

```
ingressApplications:
  ingressController:
    clientAddresses:
      virtualIp: "client_IP"
      advertised: "client_public_address"
    internalAddresses:
      virtualIp: "internal_IP"
      advertised: "internal_public_address"
```

where

client_IP is the address for external client access

internal_IP is the address for internal communication

each *public_address* value is an optional address to advertise instead of the associated *_IP* value, for example, in a NAT environment

Centralized logging

8.9 Introduction

8.9.1 CLM logging functions

The CLM includes centralized functions for logging CLM application and user activity. By default, the following are enabled:

- CLM user-activity log forwarding to the Kafka messaging subsystem
- CLM application-log forwarding to OpenSearch

"NSP logging and monitoring" in the *NSP System Administrator Guide* describes using OpenSearch and CLM Logviewer.

In order to enable one or more CLM centralized logging functions, the following CLM Installation Option must be enabled:

NSP Platform - Logging and Monitoring

Additional logging options

You can also configure the CLM to forward the following:

- application log records to Splunk servers
- CLM user-activity records to a remote syslog server
- CLM application logs to a remote syslog server
- CLM application logs to a remote Elasticsearch server

Note: You can specify separate syslog servers for application and user activity log forwarding, as described in 8.13.1 "Description" (p. 116) and 8.14 "User activity log forwarding to syslog servers" (p. 116).

Note: The forwarding of CLM application logs to a syslog server is supported over a TLS-secured or non-secure connection.

8.10 CLM application log forwarding to OpenSearch

8.10.1 Description

By default, the CLM is configured to forward CLM application logs to the internal OpenSearch engine, which makes the information available in an OpenSearch Dashboards view available from the CLM Log Viewer.

CLM application log forwarding to OpenSearch is configurable in the **nsp—modules—logging—forwarding—applicationLogs—opensearch** section of the CLM configuration file.

Note: OpenSearch-Dashboards access requires CLM user credentials.

8.11 CLM application log forwarding to Elasticsearch

8.11.1 Description

CLM application log forwarding to a remote Elasticsearch server is disabled by default. To enable CLM application-log forwarding to an Elasticsearch server, you configure the parameters in the **nsp**—**modules**—**logging**—**forwarding**—**applicationLogs**—**elasticsearch** section of the CLM configuration file.

8.11.2 Activation and security

In order to activate Elasticsearch application-log forwarding, you must copy the required TLS certificate files from the Elasticsearch server to the following location on the CLM deployer host:

/opt/nsp/NSP-CN-DEP-*release*-I*D*/NSP-CN-*release*-*ID*/tls/fluent

If mTLS is enabled on the internal CLM interface, the following TLS files are required for the mutual authentication:

- root CA certificate
- client certificate
- client key

If basic TLS is enabled on the internal CLM interface, the root CA certificate file is mandatory, and the client files are optional.

The files transferred to the CLM deployer host must be named as follows:

- root CA certificate file—ca_cert.pem
- client certificate file—client_cert.pem
- client key file—client.key

During initialization, the CLM imports the required TLS certificates to the local trust store.

8.12 CLM application log forwarding to Splunk

8.12.1 Description

A CLM cluster can forward application logs to a remote Splunk server using the Splunk HEC, or HTTP Event Collector. During CLM deployment, you can enable the log forwarding by configuring the Splunk forwarding parameters in the **nsp**—**modules**—**logging**—**forwarding**—**applicationLogs**—**splunk** section of the CLM configuration file.

When log forwarding to Splunk is enabled, you can use the CLM cluster address as a Splunk query criterion for the CLM application logs. The address to use is one of the following values in the **platform**—**ingressApplications**—**ingressController** section of the config.yml file on the local CLM deployer host:

In the **internalAddresses** subsection, if configured, otherwise, in the **clientAddresses** subsection:

- if configured, the **advertised** value
- otherwise, the **virtualIp** value

For example:

index="*k8s_log*" and nspHost="*cluster_address*"

where

cluster_address is the advertised client address in the CLM configuration file described above

k8s_log is the Splunk HEC index

For information about setting up Splunk HEC, see the Splunk documentation.

8.13 CLM application log forwarding to syslog servers

8.13.1 Description

To enable CLM application-log forwarding to a syslog server, you must configure the parameters in the **nsp**—**modules**—**logging**—**forwarding**—**applicationLogs**—**syslog** section of the CLM configuration file.

Note: A syslog server address can be an IPv4 or IPv6 address, or a hostname or FQDN that the local CLM cluster can resolve.

To secure the application-log forwarding, you must generate a TLS certificate on the syslog server and transfer the certificate to the **caCertPath** location that you specify in the **applicationLogs** section of the CLM configuration file. During initialization, the CLM imports the certificate to the local trust store.

"What is the syslog record format for NSP application log forwarding?" in the *NSP System Administrator Guide* describes the application log record format.

8.14 User activity log forwarding to syslog servers

8.14.1 Description

You enable the forwarding of CLM user activity logs to a remote syslog server by specifying the syslog server parameters in the **nsp**—**modules**—**logging**—**forwarding**—**activityLogs**—**syslog** section of the CLM configuration file.

Note: A syslog server address can be an IPv4 or IPv6 address, or a hostname or FQDN that the local CLM cluster can resolve.

In order to secure the forwarding of logs to a syslog server, you must generate a TLS certificate on the syslog server, and transfer the certificate to the **caCertPath** location that you specify in the **activityLogs** section of the CLM configuration file. During initialization, the CLM imports the certificate to the local TLS truststore.

9 CLM software configuration

9.1 CLM configuration file

9.1.1 nsp-config.yml file format

In order to deploy a CLM cluster, you must specify system parameters, installation options, and additional components in the following CLM configuration file on the CLM deployer host:

/opt/nsp/NSP-CN--DEP-*release-ID*/NSP-CN-*release-ID*/config/nsp-config.yml

The nsp-config.yml file has the following main sections:

- **platform**—defines the following for the mandatory *platform-baseServices* installation option:
  - deployment type, for example, standalone or DR
  - CLM IP addressing scheme
  - container environment
- **nsp**—defines the CLM deployment; includes the following subsections:
  - **deployment**—system scale, license, DR, backup configuration
  - **installationOptions**—CLM installation options to include in the deployment
  - **modules**—internal nspOs functions such as security and health monitoring, optional system functions
  - **sso**—single sign-on configuration for system access

A section begins with a header and section label, followed by descriptive text and one or more parameter lines. An example section layout is shown below.

```
################################################################
## platform - Platform specific configs                       ##
################################################################
platform:
   ## parameter_1     - This is the parameter 1 description.
   ## parameter_2     - This is the parameter 2 description.
   ## parameter_3     - This is the parameter 3 description.
   # parameter_1=
   # parameter_2=
    parameter_3=true
```

To enable a parameter, delete the leading # symbol from the parameter line. In the example, only *parameter_3* is enabled and configured.

Note: You must preserve the lead spacing of each line. Ensure that you delete only the # symbol, and no spaces, from a parameter line.

Note: In the event of a discrepancy between information in a configuration file and the CLM documentation, or if the documentation fails to adequately describe a specific configuration, the configuration file information is to be followed and considered correct. Also, the *CLM Release Notice* describes configuration updates and corrections that are not captured in the core documentation.

9.1.2 Enabling installation options

The **installationOptions** section lists all available installation options. You enable the deployment of an option by removing the leading # character from the name and id lines of the option.

In **installationOptions**, choose the following:

nsp - installationOptions

- centralizedLicenseManager-standalone

nsp - deployment - mode

- centralized-license-manager

nsp - deployment - type

- basic

9.2 Configuring database backups

9.2.1 Description

It is strongly recommended that you configure a database backup storage location that is not local to the CLM cluster. To do so, you must configure NFS using the following parameters in the **backups** section of the nsp-config.yml file:

Note: If the NFS parameters are configured, the capacity parameter must be unconfigured, as shown below.

```
storage:
  existingClaim: ""
  capacity: ""
  nfs:
    server: "server"
    path: "path"
```

where

server is the NFS server IP address

path is the local path of the exported file system

9.3 Configuring single sign-on

9.3.1 Introduction

The CLM supports single sign-on, or SSO access, as described in . Multiple authentication sources of the same or different type are supported.

Note: The descriptive text in the nsp-config.yml file includes additional configuration information.

Configuring LDAPS or secure AD

TLS certificates for LDAPS communication must be copied to the /tls/ldap directory below the CLM installation directory.

Using LDAPS requires that an LDAPS certificate contains the IP or hostname of the LDAP server in the certificate SAN field, and that the same IP or hostname is specified in the nsp-config.yml.

9.3.2 CLM SSO configuration parameters

Table 9-1, “SSO parameters, CLM configuration file” (p. 120) lists and describes the configuration parameters in the **sso** subsection, **nsp** section of the nsp-config.yml file. You can use the parameters to enable HSTS for secure web-browser access, and configure brute-force detection parameters for managing repeated failed login attempts, as described in6.10.2 “CLM login protection” (p. 78).

See for remote authentication configuration examples.

Table 9-1 SSO parameters, CLM configuration file

Section and parameters	Description
hsts	Whether to enable HSTS headers that tell client browsers to use only HTTPS and a valid CA certificate Default: false
bruteForceDetection parameters	

Table 9-1 SSO parameters, CLM configuration file (continued)

Section and parameters		Description
	enabled	Whether to enable brute-force protection Default: true
	permanentLockout	Whether to enable permanent user lockout after the maxLoginFailures number of login failures Default: false
	maxLoginFailures	Number of allowed login failures before temporary or permanent lockout Default: 5
	waitIncrement	Temporary lockout time, in seconds, after maxLoginFailures failed login attempts reached Default: 60
	quickCheck	Number of milliseconds during which two consecutive login failures enable lockout period defined by minQuickWait parameter Default: 1000
	minQuickWait	Lockout duration, in seconds, triggered by quickCheck violation Default = 60
	maxWait	Maximum temporary lockout duration, in minutes Default: 15
	failureResetTime	Number of hours after which to reset the login-failure counts Default: 12
ldap — LDAP parameters		
enabled		Whether LDAP is to be used for authentication Default: false
servers		List of LDAP servers; specify a server using the parameters below

Table 9-1 SSO parameters, CLM configuration file (continued)

Section and parameters		Description	
	type	LDAP server type; valid values are: • AD • AUTHENTICATED	
	name	LDAP server name; text string	
	url	LDAP server URL with IP address or hostname and port, for example: ldap://203.0.113.172:389 Default: none	
	priority	LDAP server priority, 0 is highest Default: 0	
	usernameLdapAttribute	LDAP attribute to map to OAUTH2 username, for example, cn, uid, or userPrincipalName	
	rdnLdapAttribute	LDAP attribute to use as rdn for typical user dn, typically cn	
	uuidLdapAttribute	LDAP attribute that uniquely identifies LDAP objects	
	userObjectClasses	Comma-separated list of user objectClasses	
	customUserLdapFilter	Additional filter for user searches	
	searchScope	Scope of user search in userDn; valid values are: • 1—scope limited to specified userDN • 2—scope is entire sub-tree	
	security	LDAP server security type; valid values are: • TLS • None	
	timeout	Timeout period for receiving LDAP server response, in milliseconds Default: 5000	
	userDn	DN of LDAP tree in which to find users	
	userFilter	User filter criteria	
	groupDn	DN of LDAP tree in which to find groups	
	groupNameLdapAttribute	LDAP attribute to map to user group	
	groupsLdapFilter	Groups filter criteria	
	groupObjectClasses	Comma-separated list of objectClasses for groups	
	groupMembershipLdapAttribute	Group attribute for user search	
	groupMembershipUserLdapAttribute	Username attribute in group membership	
	groupMemberOfLdapAttribute	User attribute that indicates group membership, usually memberOf	
	bind	LDAP bind credentials; for AUTHENTICATED server type only	
		dn	Bind user DN
		credential	Bind user credential

Table 9-1 SSO parameters, CLM configuration file (continued)

Section and parameters	Description
radius — RADIUS parameters	
enabled	Whether RADIUS is to be used for authentication Default: none
address	Comma-separated list of colon-separated RADIUS-server IP addresses or hostnames and ports; for example: 203.0.113.150:1812,radius-server-a:1812 Default: none
secret	RADIUS server secret You can specify a unique secret for each RADIUS server. Default: none
protocol	Protocol to use—PAP or CHAP Default: none
retries	Maximum number of attempts to reach server Default: 3
timeout	Timeout, in milliseconds, for RADIUS-server connection attempts Default: 5000
vendorId	Vendor ID for VSA search Default: 123
roleVsaId	VSA ID used to identify group Default: 3
nasId	ID of the RADIUS Network Access Server (optional)
nasIp	IP address of the RADIUS Network Access Server (optional)
nasIpv6	IPv6 address of the RADIUS Network Access Server (optional)
tacacs — TACACS+ parameters	
enabled	Whether TACACS+ authentication is to be used Default: none
address	Comma-separated list of colon-separated TACACS+-server IP addresses or hostnames and ports; for example: 203.0.113.167:1812,tacacs-server-a:1812 Default: none
secret	Shared TACACS+ server secret The secret must be common to all TACACS+ servers. Default: none
protocol	Protocol to use Default: PAP

Table 9-1 SSO parameters, CLM configuration file (continued)

Section and parameters	Description
timeout	Timeout, in milliseconds, for TACACS+-server connection attempts Default: 7000
defaultGroup	Default group to assign if no group is defined on remote server for user The group is assigned to a TACACS+ user if the vsaEnabled parameter is set to false. Default: none
vsaEnabled	Whether VSA search is enabled If set to true, a user group attribute is expected in the user authentication response/ Default: true
roleVsaId	Role used for VSA search Default: sam-security-group
vsaServiceId	VSA search service identifier Default: sam-app

10 CLM system installation

10.1 Supported installation scenarios

10.1.1 Introduction

CAUTION

Service Disruption

A CLM system installation requires a thorough understanding of CLM system administration and platform requirements, and is supported only under the conditions described in this guide, the CLM Installaton and Upgrade Guide, and the CLM Release Notice.

Such an operation must be planned, documented, and tested in advance on a trial system that is representative of the target live network. Contact CLM professional services to assess the requirements of your CLM deployment, and for information about the upgrade service, which you are strongly recommended to engage for any type of deployment.

The following scenario supports the initial deployment of CLM clusters:

- CLM system installation that does not include legacy CLM components; see 10.1.2 “New CLM system deployment” (p. 125)

10.1.2 New CLM system deployment

A completely new CLM system deployment, sometimes called a greenfield deployment, includes only new CLM components. Such a scenario is described in 10.2 “Workflow for new CLM system deployment” (p. 126), which provides a comprehensive view of the installation activities for planning purposes. The workflow includes links to CLM installation procedures.

Licensing

Your CLM system requires a license. It is recommended that you contact Nokia early in the planning process to obtain the required license for your deployment.

DR system installation

10.4 “To install the CLM” (p. 129) describes the installation of one CLM cluster. To install a DR CLM system, you must perform the procedure once in each data center, as described in 10.2 “Workflow for new CLM system deployment” (p. 126).

Note: In a DR deployment, the first data center in which you perform 10.4 “To install the CLM” (p. 129) is designated the primary data center and hosts the active nspOs instance.

10.2 Workflow for new CLM system deployment

10.2.1 Purpose

The following is the sequence of high-level actions that you must perform in order to install a standalone or DR CLM system.

10.2.2 Stages

1

If using physical hosts for the VMs, perform 10.3 “To provision the network bridge for CLM VMs” (p. 126) on each physical host to create a network bridge for KVM access to the guest VMs.

2

Commission the required stations for the CLM VMs.

3

Ensure that the RHEL chronyd time-synchronization service is running on each component, and that chronyd is actively tracking a central time source. See the RHEL documentation for information about using the chronyc command to view the chronyd synchronization status.

Note: CLM deployment is blocked if the chronyd service is not active.

4

Contact Nokia to obtain the required license for your deployment.

5

If you are using a custom TLS certificate for the deployment, perform 6.14 “To generate custom TLS certificate files for the CLM” (p. 83) to generate the required TLS files.

6

Install the CLM in the standalone or primary data center; perform 10.4 “To install the CLM” (p. 129).

7

In a DR deployment, install the CLM in the standby data center; perform 10.4 “To install the CLM” (p. 129).

10.3 To provision the network bridge for CLM VMs

10.3.1 Purpose

Perform the following steps on a physical host to create a network bridge for communication with the guest VMs.

Note: There is no requirement for the VM host station to be at the same RHEL OS release as the guest CLM VMs, which use RHEL 8. The configuration commands in the procedure are specific to RHEL 7, the predominantly deployed RHEL version in data centers that have not yet migrated to RHEL 8.

Note: It is strongly recommended that you perform the procedure using the local console or ILO interface. Using a local session ensures that the session remains active in the event that a misconfiguration or network disruption isolates the station.

Note: You require root user privileges on the station.

Note: Command lines use the # symbol to represent the RHEL CLI prompt for the root user. Do not type the leading # symbol when you enter a command.

10.3.2 Steps

1

Ensure that the RHEL firewalld, iptables, and netfilter configurations allow traffic to and from the network bridge; see the RHEL documentation for information.

2

If you are configuring the network bridge on a RHEL 8 station, perform the following steps.

1. See the RHEL 8 documentation for information about configuring a network bridge and assigning an interface to the bridge.
2. Go to Step 16.

3

Log in as the root user on the station.

4

Open a console window.

5

Enter the following sequence of commands:

```
chkconfig NetworkManager off
chkconfig network on
systemctl stop NetworkManager
systemctl start network
```

6

Open the following file using a plain-text editor such as vi:

/etc/sysconfig/network-scripts/ifcfg-*interface*

where *interface* is the physical network interface that the host is to use for connectivity as a bridge member, for example, eno1

7

Add the following line:

`BRIDGE=bridge_name`

where *bridge_name* is the name to assign to the bridge

8

Record the *bridge_name* value.

9

Save and close the file.

10

Create the following file using a plain-text editor such as vi:

/etc/sysconfig/network-scripts/ifcfg-*bridge_name*

where *bridge_name* is the network bridge name specified in Step 7

11

Enter the following as the file content:

```
TYPE=Bridge
BOOTPROTO=static
DEFROUTE=yes
DEVICE=bridge_name
ONBOOT=yes
IPADDR=n.n.n.n
PREFIX=mm
GATEWAY=g.g.g.g
DNS1=d.d.d.d
DOMAIN=domain
```

where

bridge_name is the network bridge name specified in Step 7

n.n.n.n is the IP address of *interface* specified in Step 6

mm is the *interface* subnet mask

g.g.g.g is the gateway IP address

d.d.d.d is a DNS IP address

domain is the DNS server FQDN

12

Save and close the ifcfg-*bridge_name* file.

13

Enter the following to instantiate the network bridge:

```
# brctl addbr bridge_name ↵
```

where *bridge_name* is the network bridge name

14

Enter the following to restart the network service:

```
# systemctl restart network ↵
```

15

Close the console window.

16

After you create the required VMs for the CLM deployment, verify the bridge connectivity between the host and the guest VMs.

END OF STEPS

10.4 To install the CLM

10.4.1 Purpose

Perform this procedure to deploy a new standalone or DR CLM system.

Note: To create a DR deployment, you must perform the procedure on the CLM cluster in each data center. The CLM cluster on which you first perform the procedure initializes as the primary cluster.

Note: You require root user privileges on the CLM deployer host, and on each VM that you create.

Note: *release-ID* in a file path has the following format:
R.r.p-rel.*version*
where
R.r.p is the CLM release, in the form *MAJOR.minor.patch*
version is a numeric value

Note: Command lines use the # symbol to represent the RHEL CLI prompt for the root user. Do not type the leading # symbol when you enter a command.

10.4.2 Steps

Create CLM deployer host VM

1

Download the following from the CLM downloads page on the Nokia Support portal:

Note: You must also download the .cksum file associated with each.

Note: This step applies only when using a CLM OEM disk image.

- NSP_CLM_K8S_DEPLOYER_*R_r*.tar.gz—bundle for installing the registry and deploying the container environment
- one of the following RHEL OS images for creating the CLM deployer host and CLM cluster VMs:
  - NSP_K8S_PLATFORM_RHEL8_*yy_mm*.qcow2
  - NSP_K8S_PLATFORM_RHEL8_*yy_mm*.ova
- NSP_CLM_DEPLOYER_*R_r*.tar.gz—bundle for installing the CLM application software

where

R_r is the CLM release ID, in the form *Major_minor*

yy_mm represents the year and month of issue

2

It is strongly recommended that you verify the message digest of each CLM image file or software bundle that you download from the Nokia Support portal. The download page includes checksums for comparison with the output of the RHEL md5sum, sha256sum, or sha512sum command.

To verify a file checksum, perform the following steps.

1. Enter the following:

   ```
   # command file ↵
   ```

   where

   command is md5sum, sha256sum, or sha512sum

   file is the name of the file to check

   A file checksum is displayed.
2. Compare the checksum value and the value in the .cksum file.
3. If the values do not match, the file download has failed. Download a new copy of the file, and then repeat this step.

3

Log in as the root user on the station designated to host the CLM deployer host VM.

4

Open a console window.

5

If the downloaded NSP_CLM_DEPLOYER_R_r.tar.gz file has multiple parts, enter the following to create one NSP_CLM_DEPLOYER_R_r.tar.gz file from the partial image files:

```
# cat filename.part* >filename.tar.gz ↵
```

where *filename* is the image file name

A *filename*.tar.gz file is created in the current directory.

6

Perform one of the following to create the CLM deployer host VM.

Note: The CLM deployer host VM requires a hostname; you must change the default of 'localhost' to an actual hostname.

a. Deploy the downloaded NSP_K8S_PLATFORM_RHEL8_*yy_mm*.qcow2 disk image; perform Step 6 to Step 16 of 4.3 "To deploy a RHEL qcow2 disk image" (p. 29).

b. Deploy the NSP_K8S_PLATFORM_RHEL8_*yy_mm*.ova disk image; see the documentation for your virtualization environment for information.

Note: For OVA-image deployment, it is strongly recommended that you mount the /opt directory on a separate hard disk that has sufficient capacity to allow for future expansion.

c. Manually install the RHEL OS and configure the disk partitions, as described in "Manual RHEL OS installation for CLM" (p. 51) and Chapter 4, "CLM disk setup and partitioning".

Configure CLM deployer host networking

7

Enter the following to open a console session on the CLM deployer host:

```
# virsh console deployer_host ↵
```

You are prompted for credentials.

8

Enter the following credentials:

- username—root
- password—*available from technical support*

A virtual serial console session opens on the deployer host VM.

9

Enter the following:

```
# ip a ↵
```

The available network interfaces are listed; information like the following is displayed for each:

```
if_n: if_name: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq
state UP group default qlen 1000
    link/ether MAC_address
    inet IPv4_address/v4_netmask brd broadcast_address scope global
noprefixroute if_name
       valid_lft forever preferred_lft forever
    inet6 IPv6_address/v6_netmask scope link
       valid_lft forever preferred_lft forever
```

10

Record the *if_name* and *MAC_address* values of the interface that you intend to use.

11

Enter the following:

 # nmcli con add con-name *con_name* ifname *if_name* type ethernet mac *MAC_address* ↵

where

con_name is a connection name that you assign to the interface for ease of identification

if_name is the interface name recorded in Step 10

MAC_address is the MAC address recorded in Step 10

12

Enter the following:

 # nmcli con mod *con_name* ipv4.addresses *IP_address/netmask* ↵

where

con_name is the connection name assigned in Step 11

IP_address is the IP address to assign to the interface

netmask is the subnet mask to assign

13

Enter the following:

 # nmcli con mod *con_name* ipv4.method static ↵

14

Enter the following:

 # nmcli con mod *con_name* ipv4.gateway *gateway_IP* ↵

gateway_IP is the gateway IP address to assign

15

Enter the following:

Note: You must specify a DNS name server. If DNS is not deployed, you must use a non-routable IP address as a nameserver entry.

Note: Any hostnames used in a CLM deployment must be resolved by a DNS server.

Note: A CLM deployment that uses IPv6 networking for client communication must use a hostname configuration.

nmcli con mod *con_name* ipv4.dns *nameserver_1,nameserver_2... nameserver_n* ↵

where *nameserver_1* to *nameserver_n* are the available DNS name servers

16

To optionally specify one or more DNS search domains, enter the following:

nmcli con mod *con_name* ipv4.dns-search *search_domains* ↵

where *search_domains* is a comma-separated list of DNS search domains

17

Enter the following to reboot the VM:

systemctl reboot ↵

Install CLM Kubernetes registry

18

Log in as the root or CLM admin user on the CLM deployer host.

19

Enter the following:

mkdir /opt/nsp ↵

20

Copy the downloaded NSP_CLM_K8S_DEPLOYER_R_r.tar.gz bundle file to the following directory:

/opt/nsp

21

Enter the following:

cd /opt/nsp ↵

22

Enter the following:

tar xvf NSP_CLM_K8S_DEPLOYER_R_*r*.tar.gz ↵

where *R_r* is the CLM release ID, in the form *Major_minor*

The bundle file is expanded, and the following directories are created:

- /opt/nsp/nsp-k8s-deployer-*release-ID*
- /opt/nsp/nsp-registry-*release-ID*

23

Remove the bundle file to save disk space; enter the following:

rm -f NSP_CLM_K8S_DEPLOYER_*R_r*.tar.gz ↵

The file is deleted.

24

Enter the following:

cd nsp-registry-*release-ID*/bin ↵

25

Enter the following:

./nspregistryctl install ↵

The following prompt is displayed.

```
Enter a registry admin password:
```

26

Create a registry administrator password, and enter the password.

The following prompt is displayed.

```
Confirm the registry admin password:
```

27

Re-enter the password.

The registry installation begins, and messages like the following are displayed.

```
✓ New installation detected.
✓ Initialize system.
date time Copy container images ...
date time Install/update package [container-selinux] ...
✓ Installation of container-selinux has completed.
date time Install/update package [k3s-selinux] ...
✓ Installation of k3s-selinux has completed.
date time Setup required tools ...
```

```
✓ Initialization has completed.
date time Install k3s ...
date time Waiting for up to 10 minutes for k3s initialization ...
.........................................
✓ Installation of k3s has completed.
➜ Generate self-signed key and cert.
date time Registry TLS key file:
/opt/nsp/nsp-registry/tls/nokia-nsp-registry.key
date time Registry TLS cert file:
/opt/nsp/nsp-registry/tls/nokia-nsp-registry.crt
date time Install registry apps ...
date time Waiting for up to 10 minutes for registry services to be
ready ...
.........
✓ Registry apps installation is completed.
date time Generate artifacts ...
date time Apply artifacts ...
date time Setup registry.nsp.nokia.local certs ...
date time Setup a default project [nsp] ...
date time Setup a cron to regenerate the k3s certificate [nsp] ...
✓ Post configuration is completed.
✓ Installation has completed.
```

28

Enter the following periodically to display the status of the Kubernetes system pods:

Note: You must not proceed to the next step until each pod STATUS reads Running or Completed.

```
# kubectl get pods -A ↵
```

The pods are listed.

Create CLM cluster VMs

29

For each required CLM cluster VM, perform one of the following to create the VM.

Note: Each CLM cluster VM requires a hostname; you must change the default of 'localhost' to an actual hostname.

a. Deploy the downloaded NSP_K8S_PLATFORM_RHEL8_*yy_mm*.qcow2 disk image; perform Step 6 to Step 16 of 4.3 "To deploy a RHEL qcow2 disk image" (p. 29).

b. Deploy the NSP_K8S_PLATFORM_RHEL8_*yy_mm*.ova disk image; see the documentation for your virtualization environment for information.

Note: For OVA-image deployment, it is strongly recommended that you mount the /opt directory on a separate hard disk that has sufficient capacity to allow for future expansion.

c. Manually install the RHEL OS and configure the disk partitions, as described in "Manual RHEL OS installation for CLM" (p. 51) and Chapter 4, "CLM disk setup and partitioning".

30

Record the MAC address of each interface on each VM.

31

Perform Step 32 to Step 50 for each CLM cluster VM to configure the required interfaces.

Configure CLM cluster networking

32

Enter the following to open a console session on the VM:

```
# virsh console NSP_cluster_VM ↵
```

where *NSP_cluster_VM* is the VM name

You are prompted for credentials.

33

Enter the following credentials:

- username—root
- password—*available from technical support*

A virtual serial console session opens on the CLM cluster VM.

34

Enter the following:

```
# ip a ↵
```

The available network interfaces are listed; information like the following is displayed for each:

```
if_n: if_name: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq
state UP group default qlen 1000
    link/ether MAC_address
    inet IPv4_address/v4_netmask brd broadcast_address scope global
noprefixroute if_name
       valid_lft forever preferred_lft forever
    inet6 IPv6_address/v6_netmask scope link
       valid_lft forever preferred_lft forever
```

35

Record the *if_name* and *MAC_address* values of the interfaces that you intend to use.

36

Enter the following for each interface:

```
# nmcli con add con-name con_name ifname if_name type ethernet mac MAC_address ↵
```

where

con_name is a connection name that you assign to the interface for ease of identification; for example, ClientInterface or MediationInterface

if_name is the interface name recorded in Step 35

MAC_address is the MAC address recorded in Step 35

37

Enter the following for each interface:

```
# nmcli con mod con_name ipv4.addresses IP_address/netmask ↵
```

where

con_name is the connection name assigned in Step 36

IP_address is the IP address to assign to the interface

netmask is the subnet mask to assign

38

Enter the following for each interface:

```
# nmcli con mod con_name ipv4.method static ↵
```

39

Enter the following for each interface:

```
# nmcli con mod con_name ipv4.gateway gateway_IP ↵
```

gateway_IP is the gateway IP address to assign

Note: This command sets the default gateway on the primary interface and the gateways for all secondary interfaces.

40

Enter the following for all secondary interfaces:

```
# nmcli con mod con_name ipv4.never-default yes ↵
```

41

Enter the following for each interface:

Note: You must specify a DNS name server. If DNS is not deployed, you must use a non-routable IP address as a nameserver entry.

Note: Any hostnames used in a CLM deployment must be resolved by a DNS server.

Note: A CLM deployment that uses IPv6 networking for client communication must use a hostname configuration.

```
# nmcli con mod con_name ipv4.dns nameserver_1,nameserver_2...
nameserver_n ↵
```

where *nameserver_1* to *nameserver_n* are the available DNS name servers

42

To optionally specify one or more DNS search domains, enter the following for each interface:

```
# nmcli con mod con_name ipv4.dns-search search_domains ↵
```

where *search_domains* is a comma-separated list of DNS search domains

43

Open the following file with a plain-text editor such as vi:

/etc/sysctl.conf

44

Locate the following line:

```
vm.max_map_count=value
```

45

Edit the line to read as follows; if the line is not present, add the line to the end of the file:

```
vm.max_map_count=262144
```

46

Save and close the file.

47

If you are installing in a KVM environment, enter the following:

```
# mkdir /opt/nsp ↵
```

48

It is essential that the disk I/O on each VM in the CLM cluster meets the CLM specifications.

On each CLM cluster VM, perform the tests described in “To verify disk performance for NSP” in the *NSP Troubleshooting Guide*.

If any test fails, contact technical support for assistance.

49

Enter the following to reboot the CLM cluster VM:

```
# systemctl reboot ↵
```

50

Close the console session by pressing Ctrl+] (right bracket).

Deploy Kubernetes environment

51

Enter the following on the CLM deployer host

```
# cd /opt/nsp/nsp-k8s-deployer-release-ID/config ↵
```

52

Open the following file using a plain-text editor such as vi:

k8s-deployer.yml

53

Configure the following parameters for the CLM cluster VM.

```
- nodeName: noden
  nodeIp: private_IP_address
  accessIp: public_IP_address
  isIngress: value
```

Note: The nodeName value:

- can include only ASCII alphanumeric and hyphen characters
- cannot include an upper-case character
- cannot begin or end with a hyphen
- cannot begin with a number
- cannot include an underscore
- must end with a number

54

In the following section, specify the virtual IP addresses for the CLM to use as the internal load-balancer endpoints.

Note: A single-node CLM cluster requires at least the *client_IP* address.

The addresses are the **virtualIP** values for CLM client, internal, and mediation access that you intend to specify in Step 75 in the nsp-config.yml file.

```
loadBalancerExternalIps:
```

- *client_IP*
- *internal_IP*

The following addresses are not used in CLM and must be left empty.

- *trapV4_mediation_IP*
- *trapV6_mediation_IP*
- *flowV4_mediation_IP*
- *flowV6_mediation_IP*

55

Configure the following parameter, which specifies whether dual-stack NE management is enabled:

Note: Dual-stack NE management can function only when the network environment is appropriately configured, for example:

- Only valid, non-link-local static or DHCPv6-assigned addresses are used.
- A physical or virtual IPv6 subnet is configured for IPv6 communication with the NEs.

```
enable_dual_stack_networks: value
```

where *value* must be set to true if the cluster VMs support both IPv4 and IPv6 addressing

56

Configure the following parameter in the **cluster** section:

```
hosts: "path"
```

where *path* is the location of the hosts file for deploying the CLM cluster

57

If you have disabled remote root access to the CLM cluster VMs, configure the following parameters in the **cluster** section, **sshAccess** subsection:

```
sshAccess:
  userName: "user"
  privateKey: "path"
```

where

user is the designated CLM ansible user

path is the SSH key path, for example, /home/*user*/.ssh/id_rsa

58

Save and close the k8s-deployer.yml file.

59

Create a backup copy of the updated k8s-deployer.yml file, and transfer the backup copy to a station that is separate from the CLM system, and preferably in a remote facility.

Note: The backup file is crucial in the event of a CLM deployer host failure, and must be copied to a separate station.

60

Enter the following:

```
# cd /opt/nsp/nsp-k8s-deployer-release-ID/bin ↲
```

61

Enter the following to create the cluster configuration:

```
# ./nspk8sctl config -c ↲
```

The following is displayed when the creation is complete:

```
✓ Cluster hosts configuration is created at:
/opt/nsp/nsp-k8s-deployer-release-ID/config/hosts.yml
```

62

Enter the following to import the Kubernetes container images to the registry:

```
# ./nspk8sctl import ↲
```

Messages like the following are displayed as the import proceeds:

```
✓ Pushing artifacts to registry (it takes a while) ...
date time Load container image from
[/opt/nsp/nsp-k8s-deployer-release-ID/artifact/nsp-k8s-R.r.0-rel.tar.
gz] ...
date time Push image [image_name] to registry.nsp.nokia.local/library
...
date time Push image [image_name] to registry.nsp.nokia.local/library
...
.
.
.
date time Push image [image_name] to registry.nsp.nokia.local/library
...
```

63

For password-free CLM deployer host access to the CLM cluster VMs, you require an SSH key.

To generate and distribute the SSH key, perform the following steps.

1. Enter the following:

   ```
   # ssh-keygen -N "" -f path -t rsa ↲
   ```

   where *path* is the SSH key file path, for example, /home/*user*/.ssh/id_rsa

   An SSH key is generated.

2. Enter the following for each CLM cluster VM to distribute the key to the VM.

```
# ssh-copy-id -i key_file user@address ↵
```

where

user is the designated CLM ansible user configured in Step 57 , if root-user access is restricted; otherwise, *user@* is not required

key_file is the SSH key file, for example, /home/*user*/.ssh/id_rsa.pub

address is the CLM cluster VM IP address

64

Enter the following:

Note: If the CLM cluster VMs do not have the required SSH key, you must include the --ask-pass argument in the following command, and are subsequently prompted for the root password of each cluster member:
nspk8sctl --ask-pass install

```
.# ./nspk8sctl install ↵
```

The CLM Kubernetes environment is deployed.

65

The CLM cluster member named node1 is designated the CLM cluster host for future configuration activities; record the CLM cluster host IP address for future reference.

Check CLM cluster status

66

Open a console window on the CLM cluster host.

67

Enter the following periodically to display the status of the Kubernetes system pods:

Note: You must not proceed to the next step until each pod STATUS reads Running or Completed.

```
# kubectl get pods -A ↵
```

The pods are listed.

68

Enter the following periodically to display the status of the CLM cluster nodes:

Note: You must not proceed to the next step until each node STATUS reads Ready.

```
# kubectl get nodes -o wide ↵
```

The CLM cluster nodes are listed, as shown in the following example:

```
NAME    STATUS   ROLES    AGE   VERSION   INTERNAL-IP   EXTERNAL-IP
node1   Ready    master   nd    version   int_IP   ext_IP
```

Configure CLM software

69

Open a console window on the CLM deployer host.

70

Enter the following:

```
# cd /opt/nsp ↵
```

71

Enter the following:

```
# tar xvf NSP_CLM_DEPLOYER_R_r.tar.gz ↵
```

where *R_r* is the CLM release ID, in the form *Major_minor*

The bundle file is expanded, and the following directory is created:

/opt/nsp/NSP-CN-DEP-*release-ID*/NSP-CN-*release-ID*

72

Enter the following:

```
# rm -f NSP_CLM_DEPLOYER_R_r.tar.gz ↵
```

The bundle file is deleted.

73

Open the following file using a plain-text editor such as vi to specify the system parameters and enable the required installation options:

/opt/nsp/NSP-CN-DEP-*release-ID*/NSP-CN-*release-ID*/config/nsp-config.yml

Note: See 9.1.1 “nsp-config.yml file format” (p. 119) for configuration information.

Note: You must preserve the leading spaces in each line.

74

Configure the following parameter in the **platform** section as shown below:

```
  clusterHost: "cluster_host_address"
```

where

cluster_host_address is the address of CLM cluster member node1, which is subsequently used for cluster management operations

75

Configure the following CLM cluster address parameters in the **platform** section, **ingressApplications** subsection as shown below.

Each address is an address from the **ingressApplications** section of the k8s-deployer.yml file described in Step 54.

Note: The *client_IP* value is mandatory; the address is used for interfaces that remain unconfigured, such as in a single-interface deployment.

Note: If the client network uses IPv6, you must specify the CLM cluster hostname as the *client_IP* value.

Note: The trapForwarder addresses are not required.

```
ingressApplications:
  ingressController:
    clientAddresses:
      virtualIp: "client_IP"
      advertised: "client_public_address"
    internalAddresses:
      virtualIp: "internal_IP"
      advertised: "internal_public_address"
```

where

client_IP is the address for external client access

internal_IP is the address for internal communication

each *public_address* value is an optional address to advertise instead of the associated *_IP* value, for example, in a NAT environment

76

Configure the remaining parameters in the **platform** section as shown below:

platform section, **docker** subsection:

```
repo: "registry.nsp.nokia.local/nsp/images"
pullPolicy: "IfNotPresent"
```

platform section, **helm** subsection:

```
repo: "oci://registry.nsp.nokia.local/nsp/charts"
timeout: "300"
```

77

Configure the **type** parameter in the **deployment** section as shown below:

```
deployment:
    type: "deployment_type"
```

where *deployment_type* is one of the parameter options listed in the section

78

If you are using a custom server certificate, configure the following **tls** parameter in the **deployment** section:

```
tls:
  customCaCert: certificate_path
```

where *certificate_path* is the file path of the custom root CA certificate file

79

If the CLM system is a DR deployment, configure the parameters in the **dr** section as shown below:

```
dr:
   dcName: "data_center"
   mode: "deployment_mode"
   peer: "peer_address"
   internalPeer: "peer_internal_address"
   peerDCName: "peer_data_center"
```

where

data_center is the unique alphanumeric name to assign to the cluster

deployment_mode is the case-sensitive deployment type, dr or standalone

peer_address is the address at which the peer data center is reachable over the client network

peer_internal_address is the address at which the peer data center is reachable over the internal network

peer_data_center is the unique alphanumeric name of the peer cluster

80

Configure the user authentication parameters in the **sso** section; see 9.3.2 “CLM SSO configuration parameters” (p. 121) for configuration information.

81

Save and close the nsp-config.yml file.

82

Ensure that your license.zip file is on the CLM deployer host in the location specified in the nsp-config.yml file.

Configure hosts file

83

Log in as the root or CLM admin user on the CLM deployer host.

84

Open the following file using a plain-text editor such as vi:

/opt/nsp/NSP-CN-DEP-*release-ID*/config/nsp-deployer.yml

85

Configure the following parameters:

```
hosts: "hosts_file"
labelProfile: "../ansible/roles/apps/nspos-labels/vars/labels_file"
```

where

hosts_file is the absolute path of the hosts.yml file created in Step 61, typically /opt/nsp/nsp-k8s-deployer-*release-ID*/config/hosts.yml

labels_file is the file name below that corresponds to the cluster deployment type specified in Step 77:

- node-labels-basic-1node.yml

86

If you have disabled remote root access to the CLM cluster VMs, configure the following parameters in the **cluster** section, **sshAccess** subsection:

```
sshAccess:
  userName: "user"
  privateKey: "path"
```

where

user is the designated CLM ansible user

path is the SSH key path, for example, /home/*user*/.ssh/id_rsa

87

Save and close the nsp-deployer.yml file.

Install Kubernetes secrets

88

If you are configuring the standby cluster in a DR deployment, go to Step 97.

89

Open a console window on the standalone or primary CLM deployer host.

90

Enter the following:

```
# cd /opt/nsp/NSP-CN-DEP-release-ID/bin ↵
```

91

Enter the following:

```
# ./nspdeployerctl secret install ↵
```

The following prompt is displayed:

```
Would you like to use your own CA key pair for the NSP Internal
Issuer? [yes,no]
```

92

Perform one of the following.

a. Enter no ↵.

The CLM generates the internal key and certificate files.

b. Provide your own certificate to secure the internal network.

1. Enter yes ↵.

The following messages and prompt are displayed:

2.
```
Building secret 'ca-key-pair-internal-nspdeployer'

The CA key pair used to sign certificates generated by the NSP
Internal Issuer.

Please enter the internal CA private key:
```

3. Enter the full path of the internal private key.

The following prompt is displayed:

```
Please enter the internal CA certificate:
```

4. Enter the full path of the internal certificate:

The following messages are displayed for each Kubernetes namespace:

```
Adding secret ca-key-pair-internal-nspdeployer to namespace
namespace...

secret/ca-key-pair-internal-nspdeployer created
```

The following prompt is displayed:

```
Would you like to use your own CA key pair for the NSP External
Issuer? [yes,no]
```

93

Perform one of the following.

a. Enter no ↵.

The CLM generates the external key and certificate files.

b. Provide your own certificate to secure the external network.

1. Enter yes ↵.

The following messages and prompt are displayed:

```
Building secret 'ca-key-pair-external-nspdeployer'
```

```
The CA key pair used to sign certificates generated by the NSP
External Issuer.

Please enter the external CA private key:
```

2. Enter the full path of the external private key.

   The following prompt is displayed:

   ```
   Please enter the external CA certificate:
   ```

3. Enter the full path of the external certificate:

   The following messages are displayed for each Kubernetes namespace:

   ```
   Adding secret ca-key-pair-external-nspdeployer to namespace
   namespace...

   secret/ca-key-pair-external-nspdeployer created
   ```

```
Would you like to provide a custom private key and certificate for use
by NSP endpoints when securing TLS connections over the client
network? [yes,no]
```

94

Perform one of the following.

a. Enter no ↵.

   The CLM generates the client key and certificate files.

b. Provide your own certificate for the client network.

   1. Enter yes ↵

      The following messages and prompt are displayed:

      ```
      Building secret 'nginx-nb-tls-nsp'

      TLS certificate for securing the ingress gateway.

      Please enter the ingress gateway private key:
      ```

   2. Enter the full path of the private key file for client access. The private key file is the *customKey* file from 6.14 “To generate custom TLS certificate files for the CLM” (p. 83).

      The following prompt is displayed:

      ```
      Please enter the ingress gateway public certificate:
      ```

   3. Enter the full path of the public certificate file for client access. The public certificate file is the *customCert* file from 6.14 “To generate custom TLS certificate files for the CLM” (p. 83).

      The following prompt is displayed:

      ```
      Please enter the ingress gateway public trusted CA certificate
      bundle:
      ```

   4. Enter the full path of the public trusted CA certificate bundle file. The public trusted CA certificate bundle file is the *customCaCert* file from 6.14 “To generate custom TLS certificate files for the CLM” (p. 83).

      The following message is displayed:

      ```
      Adding secret nginx-nb-tls-nsp to namespace namespace...
      ```

95

Your deployment does not include MDM. The following prompt is displayed:

```
Would you like to provide mTLS certificates for the NSP mediation
interface for two-way TLS authentication? [yes,no]
```

Perform one of the following.

a. Enter no ↵

96

Back up the Kubernetes secrets.

1. Enter the following:

   # ./nspdeployerctl secret -o *backup_file* backup ↵

   where *backup_file* is the absolute path and name of the backup file to create

   As the secrets are backed up, messages like the following are displayed for each Kubernetes namespace:

   ```
   Backing up secrets to /opt/backupfile...
     Including secret namespace:ca-key-pair-external
     Including secret namespace:ca-key-pair-internal
     Including secret namespace:nsp-tls-store-pass
   ```

   When the backup is complete, the following prompt is displayed:

   ```
   Please provide an encryption password for backup_file
   enter aes-256-ctr encryption password:
   ```

2. Enter a password.

   The following prompt is displayed:

   ```
   Verifying - enter aes-256-ctr encryption password:
   ```

3. Re-enter the password.

   The backup file is encrypted using the password.

4. Record the password for use when restoring the backup.
5. Record the name of the data center associated with the backup.
6. Transfer the backup file to a secure location in a separate facility for safekeeping.

Restore secrets on standby cluster, DR deployment

97

If you are configuring the standby cluster in a DR deployment, obtain and restore the CLM secrets backup file from the CLM cluster in the primary data center.

1. Enter the following on the standby CLM deployer host:

   # scp *address*:*path/backup_file* /tmp/ ↵

   where

   address is the address of the CLM deployer host in the primary cluster

path is the absolute file path of the backup file created in Step 96

backup_file is the secrets backup file name

The backup file is transferred to the local /tmp directory.

2. Enter the following:

   # cd /opt/nsp/NSP-CN-DEP-*release-ID*/bin ↵

3. Enter the following:

   ./nspdeployerctl secret -i /tmp/*backup_file* restore ↵

   The following prompt is displayed:

   ```
   Please provide the encryption password for /opt/backupfile
   enter aes-256-ctr decryption password:
   ```

4. Enter the password recorded in Step 96.

   As the secrets are restored, messages like the following are displayed for each Kubernetes namespace:

   ```
   Restoring secrets from backup_file...
   secret/ca-key-pair-external created
     Restored secret namespace:ca-key-pair-external
   secret/ca-key-pair-internal created
     Restored secret namespace:ca-key-pair-internal
   secret/nsp-tls-store-pass created
     Restored secret namespace:nsp-tls-store-pass
   ```

5. If you answer yes to the Step 94 prompt for client access during the primary CLM cluster configuration, you must update the standby server secret for client access using the custom certificate and key files that are specific to the standby cluster.

   Enter the following:

   # ./nspdeployerctl secret -s nginx-nb-tls-nsp -n psaRestricted -f tls.key=*customKey* -f tls.crt=*customCert* -f ca.crt=*customCaCert* update ↵

   where

   customKey is the full path of the private server key

   customCert is the full path of the server public certificate

   customCaCert is the full path of the CA public certificate

   Custom certificate and key files are created by performing 6.14 “To generate custom TLS certificate files for the CLM” (p. 83).

   Messages like the following are displayed as the server secret is updated:

   ```
   secret/nginx-nb-tls-nsp patched
   The following files may contain sensitive information. They are no
   longer required by NSP and may be removed.
     customKey
     customCert
   ```

customCaCert

Deploy CLM software, monitor initialization

98

Enter the following to apply the node labels to the CLM cluster:

```
# ./nspdeployerctl config ↵
```

99

Enter the following to import the CLM images and Helm charts to the CLM Kubernetes registry

```
# ./nspdeployerctl import ↵
```

100

Enter the following to deploy the CLM software in the CLM cluster:

Note: If the CLM cluster VMs do not have the required SSH key, you must include the --ask-pass argument in the command, as shown in the following example, and are subsequently prompted for the root password of each cluster member:
nspdeployerctl --ask-pass install --config --deploy

```
# ./nspdeployerctl install --config --deploy ↵
```

The specified CLM functions are installed and initialized.

101

Monitor and validate the CLM cluster initialization.

Note: You must not proceed to the next step until each CLM pod is operational.

1. On the CLM cluster host, enter the following every few minutes:

   ```
   # kubectl get pods -A ↵
   ```

   The status of each CLM cluster pod is displayed; the CLM cluster is operational when the status of each pod is Running or Completed, with the following exception.

2. Check PVCs are bound to PVs and PVs are created with STORAGECLASS as shown below

   ```
   # kubectl get pvc -A
   NAMESPACE            NAME                          STATUS   VOLUME
   CAPACITY   ACCESS MODES   STORAGECLASS    AGE
   nsp-psa-privileged   data-volume-mdm-server-0   Bound    pvc-ID
   5Gi        RWO            storage_class  age
   nsp-psa-restricted   data-nspos-kafka-0         Bound    pvc-ID
   10Gi       RWO            storage_class   age
   nsp-psa-restricted   data-nspos-zookeeper-0     Bound    pvc-ID
   2Gi        RWO            storage_class  age
   ...
   ```

```
# kubectl get pv
NAME                        CAPACITY   ACCESS MODES   RECLAIM
POLICY   STATUS   CLAIM
nspos-fluentd-logs-data   50Mi
ROX            Retain           Bound
nsp-psa-restricted/nspos-fluentd-logs-data
pvc-ID                    10Gi       RWO            Retain
Bound    nsp-psa-restricted/data-nspos-kafka-0
pvc-ID                    2Gi        RWO            Retain
Bound    nsp-psa-restricted/data-nspos-zookeeper-0
...
```

3. Verify that all pods are in the Running state.
4. If any pod fails to enter the Running or Completed state, see the *NSP Troubleshooting Guide* for information about troubleshooting an errored pod.

102

Close the open console windows.

END OF STEPS

11 CLM system upgrade

11.1 Upgrading the CLM system

11.1.1 Purpose

This chapter describes the steps that must be performed in order to upgrade the CLM.

Note: Ensure that the old CLM is running during the upgrade.

11.1.2 Steps

Install the new CLM

1

Install the new CLM as described in Chapter 10, “CLM system installation”

Back up the existing CLM

2

To manually backup the contents of the PostgreSQL database, do the following:

1. Log in to the pre-24.11 primary CLM server as the nsp user.
2. Enter the following:
   nspdctl --host *IP_address* backup -d nspos_migration -f ↲
   where
   IP_address is the IP address of the desired CLM server
3. Verify that the backup has completed successfully. Execute:
   nspdctl --host *IP_address* backup status ↲
   where
   IP_address is the IP address of the desired CLM server

Transfer the backup to the new CLM deployer host

3

Transfer the backup file to the CLM deloyer host:

```
scp /opt/nsp/backup/nspos_migration/nspospostgresql_backup_timestamp.
tar.gz root@<deployer ip>:/root/
```

where

timestamp is the backup creation date and time

Restore the CLM portion of the database

4

Enter the following on the CLM deployer host:

cd /opt/nsp/NSP-CN-DEP-*release-ID*/NSP-CN-*release-ID*/tools/database ↵

5

Note: In a DR deployment, you must perform the steps in the data center that is the pririmary data center.

Enter the following to restore the CLM PostgreSQL database:

./nspos-db-restore-k8s.sh license-manager *backup_dir/backup_file* ↵

where

backup_dir is the directory that contains the backup file

backup_file is the backup file name, for example, for PostgreSQL, the name is /root/nspos-postgresql_backup_*timestamp*.tar.gz

END OF STEPS

12 CLM system uninstallation

12.1 Introduction

12.1.1 Description

The procedures in this chapter describe how to remove the CLM software from a CLM cluster, and how to remove the CLM deployment environment from the CLM host stations.

Note: To uninstall the software or environment in a DR CLM system, you must perform the procedures in each data center.

12.2 Workflow to uninstall a CLM cluster

12.2.1 Purpose

The following is the sequence of high-level actions required to uninstall the CLM software, and optionally, the Kubernetes deployment environment, in a data center that hosts a CLM cluster.

12.2.2 Stages

1

Uninstall the CLM software from the CLM cluster, perform 12.3 “To uninstall the CLM software from a CLM cluster” (p. 155).

2

Uninstall the CLM Kubernetes environment, for example, if the cluster host stations are to be recommissioned for another purpose, perform 12.4 “To uninstall the CLM Kubernetes software” (p. 157).

3

Uninstall the Kubernetes registry from the CLM deployer host, for example, if you intend to replace the Kubernetes environment of your CLM system, perform 12.5 “To uninstall the CLM Kubernetes registry” (p. 158).

12.3 To uninstall the CLM software from a CLM cluster

12.3.1 Purpose

Perform this procedure to remove the CLM software from the nodes in a CLM cluster.

Note: You require root or CLM admin user privileges on each CLM cluster station.

Note: *release-ID* in a file path has the following format:

R.r.p-rel.*version*
where
R.r.p is the CLM release, in the form *MAJOR.minor.patch*
version is a numeric value

12.3.2 Steps

1

Log in as the root or CLM admin user on the CLM deployer host.

2

Open a console window.

3

Enter the following:

```
# cd /opt/nsp/NSP-CN-DEP-release-ID/bin ↵
```

4

Back up the Kubernetes secrets.

1. Enter the following:

   ```
   # ./nspdeployerctl secret -o backup_file backup ↵
   ```

   where *backup_file* is the absolute path and name of the backup file to create

   As the secrets are backed up, messages like the following are displayed for each Kubernetes namespace:

   ```
   Backing up secrets to /opt/backupfile...
     Including secret namespace:ca-key-pair-external
     Including secret namespace:ca-key-pair-internal
     Including secret namespace:nsp-tls-store-pass
   ```

   When the backup is complete, the following prompt is displayed:

   ```
   Please provide an encryption password for backup_file
   enter aes-256-ctr encryption password:
   ```

2. Enter a password.

   The following prompt is displayed:

   ```
   Verifying - enter aes-256-ctr encryption password:
   ```

3. Re-enter the password.

   The backup file is encrypted using the password.

4. Record the password for use when restoring the backup.
5. Record the name of the data center associated with the backup.
6. Transfer the backup file to a secure location in a separate facility for safekeeping.

5

To preserve the cluster configuration, perform the following steps.

1. Open the following file using a plain-text editor such as vi:
   /opt/nsp/NSP-CN-DEP-*release-ID*/NSP-CN-*release-ID*/config/nsp-config.yml
2. Edit the following line in the **platform** section, **kubernetes** subsection to read:
   ```
   deleteOnUndeploy:false
   ```
3. Save and close the file.

6

Enter the following:

Note: If the CLM cluster VMs do not have the required SSH key, you must include the --ask-pass argument in the command, as shown in the following example, and are subsequently prompted for the root password of each cluster member:
nspdeployerctl --ask-pass --undeploy --clean

```
# ./nspdeployerctl uninstall --undeploy --clean ↵
```

The CLM software is removed from each CLM cluster member.

END OF STEPS

12.4 To uninstall the CLM Kubernetes software

12.4.1 Purpose

Perform this procedure to remove the CLM Kubernetes software.

Note: You require root or CLM admin user privileges on the CLM deployer host.

Note: *release-ID* in a file path has the following format:
R.r.p-rel.*version*
where
R.r.p is the CLM release, in the form *MAJOR.minor.patch*
version is a numeric value

12.4.2 Steps

1

Log in as the root or CLM admin user on the CLM deployer host.

2

Open a console window.

3

Enter the following:

```
# cd /opt/nsp/nsp-k8s-deployer-release-ID/bin ↵
```

4

Enter the following:

Note: The action affects all CLM cluster VMs specified in the hosts.yml file.

Note: If the CLM cluster VMs do not have the required SSH key, you must include the --ask-pass argument in the command, as shown in the following example, and are subsequently prompted for the root password of each cluster member:
nspk8sctl --ask-pass uninstall

```
# ./nspk8sctl uninstall ↵
```

The CLM Kubernetes software is uninstalled.

END OF STEPS

12.5 To uninstall the CLM Kubernetes registry

12.5.1 Purpose

Perform this procedure to remove the CLM Kubernetes registry and environment from a CLM deployer host.

Note: You require root or CLM admin user privileges on the CLM deployer host.

Note: *release-ID* in a file path has the following format:
R.r.p-rel.*version*
where
R.r.p is the CLM release, in the form *MAJOR.minor.patch*
version is a numeric value

12.5.2 Steps

1

Log in as the root or CLM admin user on the CLM deployer host.

2

Open a console window.

3

Enter the following:

```
# cd /opt/nsp/nsp-registry-release-ID/bin ↵
```

4

Enter the following to remove the CLM Kubernetes environment:

Note: The CLM Kubernetes registry log is /var/log/nspregistryctl.log.

```
# ./nspregistryctl uninstall -y ↵
```

The CLM Kubernetes registry and environment are uninstalled.

5

Close the console window.

END OF STEPS

A Removing world permissions from compiler executables

A.1 Resetting GCC-compiler file permissions

A.1.1 Description

This appendix describes the post-deployment configuration of specific file permissions for additional CLM system security.

Read and execute privileges for the “other” user type may be enabled on specific RPM files during RHEL OS installation, upgrade, or update.

As a security-hardening measure after such a RHEL OS operation, you can revoke the privileges, as described in A.2 “To remove world permissions from compiler executables” (p. 161).

In the event that you need to roll back the security hardening, see A.3 “To restore compiler world permissions” (p. 162).

A.2 To remove world permissions from compiler executables

A.2.1 Purpose

Perform this procedure to clear the “other” user permissions on specific GCC-compiler files on a CLM component station.

Note: It is recommended that you perform the procedure only during a scheduled maintenance period.

Note: You require root user privileges on the station.

A.2.2 Steps

1

Log in to the CLM component station as the root or CLM admin user.

2

Open a console window.

3

Paste the following command block into the console window:

```
chmod 750 /usr/bin/c89
chmod 750 /usr/bin/c99
chmod 750 /usr/bin/cc
```

```
chmod 750 /usr/bin/f95
chmod 750 /usr/bin/gcc
chmod 750 /usr/bin/gcc-ar
chmod 750 /usr/bin/gcc-nm
chmod 750 /usr/bin/gcc-ranlib
chmod 750 /usr/bin/gcov
chmod 750 /usr/bin/x86_64-redhat-linux-gcc
chmod 750 /usr/bin/c++
chmod 750 /usr/bin/g++
chmod 750 /usr/bin/x86_64-redhat-linux-c++
chmod 750 /usr/bin/x86_64-redhat-linux-g++
chmod 750 /usr/bin/gfortran
```

The file permissions are reset.

4

Close the console window.

End of steps

A.3 To restore compiler world permissions

A.3.1 Purpose

Perform this procedure to restore the original file permissions reset by performing A.2 “To remove world permissions from compiler executables” (p. 163) on a CLM component station.

Note: It is recommended that you perform the procedure only during a scheduled maintenance period.

Note: You require root user privileges on the station.

A.3.2 Steps

1

Log in to the station as the root or CLM admin user.

2

Open a console window.

3

Paste the following command block into the console window:

```
chmod 755 /usr/bin/c89
```

```
chmod 755 /usr/bin/c99
chmod 755 /usr/bin/cc
chmod 755 /usr/bin/f95
chmod 755 /usr/bin/gcc
chmod 755 /usr/bin/gcc-ar
chmod 755 /usr/bin/gcc-nm
chmod 755 /usr/bin/gcc-ranlib
chmod 755 /usr/bin/gcov
chmod 755 /usr/bin/x86_64-redhat-linux-gcc
chmod 755 /usr/bin/c++
chmod 755 /usr/bin/g++
chmod 755 /usr/bin/x86_64-redhat-linux-c++
chmod 755 /usr/bin/x86_64-redhat-linux-g++
chmod 755 /usr/bin/gfortran
```

The original file permissions are restored.

4

Close the console window.

END OF STEPS