NOKIA

NSP
Network Services Platform

Release 21.6

Planning Guide

3HE-17257-AAAB-TQZZA

Issue 1

June 2021

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.

Disclaimer

Open Source Software and Red Hat Enterprise Linux Operating System

In case:

- (i) any "Open Source Software and Red Hat Enterprise Linux Operating System ("FOSS & RHEL") is packaged separately or integrated with any Nokia Software and to which third party license obligations apply; or,
- (ii) any FOSS & RHEL is directly licensed by Customer under a separate license or subscription agreement, and such FOSS & RHEL is interacting or interoperating with any Nokia Software or Product:

information will be available either in the FOSS & RHEL itself or on the website from which the download is available indicating the license under which such FOSS was released, and containing required acknowledgements, legends and/or notices.

It is hereby acknowledged and agreed by the Parties that any FOSS & RHEL is distributed on an "as is" basis under the respective FOSS & RHEL license terms. Nokia will not warrant nor will be liable for, and will not defend, indemnify, or hold Customer harmless for any claims arising out of, or in any case related to FOSS & RHEL and their use (or inability to use) by the Parties. This includes, but is not restricted to, any and all claims for direct, indirect, incidental, special, exemplary, punitive or consequential damages in connection with FOSS & RHEL or its components (whether included in the Nokia Software or not) and their use or inability to use. Also, this includes claims for or in connection with the title in, the non-infringement of or interferences and damages caused to Customer or third parties by FOSS & RHEL.

CUSTOMER SHALL HAVE NO RIGHT TO RECEIVE FROM NOKIA ANY CARE (MAINTENANCE & SUPPORT SERVICES) ON FOSS &

RHEL LICENSED BY CUSTOMER UNDER A SEPARATE LICENSE AGREEMENT OR SUBSCRIPTION CONTRACT AND TO WHICH THIRD PARTY LICENSE OBLIGATIONS APPLY WHETHER OR NOT IT INTERACTS WITH ANY NOKIA SOFTWARE OR PRODUCT.

The above shall also apply in case Customer requires - and Nokia accepts under the terms of this Disclaimer to use its reasonable commercial effort to do so - certain installation services on FOSS & RHEL as directly licensed by Customer under a separate license or subscription agreement; and, such FOSS & RHEL are interacting or interoperating with a Nokia Software or Product. For sake of clarity in such a case the following shall also apply:

- Before starting any installation service, Customer must instruct Nokia to start such installation and must confirm in writing to Nokia that its FOSS & RHEL license or subscription contract (for RHEL: Red Hat Enterprise Agreement) with Customer includes the right to use of the specific FOSS & RHEL and the related support for all platforms running the FOSS & RHEL; that such subscription and support contract is in force (not expired) and allows such installation activities.
- Nokia will not warrant nor will be liable for any cost, expense, damage, and will not defend, indemnify, or hold Customer or any third party harmless for any claims arising out of, or in any case related to FOSS & RHEL (and in connection with the installation activities of such FOSS & RHEL) and their use (or inability to use) by the Customer or by any third party, following the installation of such FOSS & RHEL. This includes, but is not restricted to, any and all claims for direct, indirect, incidental, special, exemplary, punitive or consequential damages in connection with FOSS & RHEL or its components and their use or inability to use.
- Nokia will not provide nor will have any liability or obligation to provide any support, maintenance, care service, warranty or indemnity with respect to any (installed) FOSS & RHEL as licensed by the Customer under a separate license agreement or subscription contract.

Any Care service (maintenance and support service) on FOSS & RHEL licensed by Nokia as packaged separately or integrated with any Nokia Software may be made available by Nokia under specific contractual terms to be agreed upon by the Parties.

Contents

About this document **7**

1 Product overview **9**
1.1 NSP overview **9**
1.2 NSP deployment overview **10**
1.3 NSP key technologies **15**

2 Operating system specifications **19**
2.1 Red Hat Enterprise Linux (RHEL) **19**

3 System resource requirements **21**
3.1 Introduction **21**
3.2 Virtual machine requirements **21**
3.3 VMware virtualization **21**
3.4 KVM virtualization **22**
3.5 OpenStack requirements **23**
3.6 Platform requirements **24**
3.7 System requirements **26**
3.8 Requirements for containerized deployments **27**

4 Network requirements **29**
4.1 Overview **29**
4.2 Network requirements between NSP and other components **29**
4.3 Network requirements for redundancy, high-availability and NSP cluster nodes **30**

5 Scaling and performance **31**
5.1 Overview **31**
5.2 Scale limits for NSP deployments **31**
5.3 Scale limits for Cross Domain resource control deployments **32**
5.4 Scale limits for applications **33**
5.5 Failover performance for HA and redundant deployments **38**

6 Security **39**
6.1 Introduction **39**
6.2 Securing the NSP **39**
6.3 Operating system security for NSP workstations **40**
6.4 Communication between the NSP and external systems **40**
6.5 NSP port information **42**

6.6 NSP cluster deployer and worker node ports 50
6.7 NSP firewall rules 52

7 NSP deployment with multiple network interfaces and IP addresses 61
7.1 Support for multiple network interfaces 61
7.2 Network Address Translation 64

8 Appendix A 65
8.1 Storage-layer I/O performance tests 65

About this document

Purpose

The *NSP Planning Guide* consolidates all pre-installation information required to plan a successful deployment of the Nokia NSP product.

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

- Documentation Feedback

1 Product overview

1.1 NSP overview

1.1.1 Introduction

This chapter provides an overview of the Network Services Platform (NSP) product and the components that comprise an NSP deployment.

1.1.2 NSP architecture

The NSP product consists of multiple interoperating network management components that allow for service provisioning, automation, optimization, and element management functions for both IP and optical networks.

NSP cluster

The NSP cluster is the central component of most NSP deployments. It hosts all the major NSP software functions, and is the location where the common services (nspOS) are located. The nspOS enables system-wide functions, including Single Sign On, Centralized Logging and operator access to the NSP Launchpad. The nspOS also contains common components and services that other NSP components require. In a shared-mode deployment, each component uses the nspOS instance in the NSP cluster.

The following applications are hosted by the NSP cluster when deployed with the corresponding feature packages. For information about feature packages and installation options, see the *NSP System Architecture Guide.*

- **Model-driven Mediation (MDM)** - provides mediation between model-driven NSP applications and Nokia or third-party network devices. MDM provides an adaptation layer which uses adaptors to convert NSP application requests to device specific directives using standard protocols such as gRPC/gNMI, NETCONF, SNMP and CLI over SSH or Telnet. MDM is an optional component in a NSP deployment and can coexist with NFM-P and NFM-T.
- **Workflow Manager (WFM)** - allows for the creation and execution of workflows. A workflow consists of a sequence of tasks to create an automated procedure. A workflow can be executed on demand, scheduled, or triggered to run in response to a Kafka event notification. Some workflow examples include node software upgrades, service activation tests, service fulfillment with pre- and post-deployment workflows, and mass migration of services from one tunnel to another.
- **NSP Baseline Analytics** - monitors network traffic to establish baselines and can flag anomalous traffic patterns. Traffic patterns can be saved for analysis and comparison. Enables automated corrective actions by other NSP applications.

In an IP-only or optical-only deployment, nspOS is installed with NFM-P or NFM-T systems to provide common services such as Single Sign On, Launchpad and common applications. When deployed with NFM-P or NFM-T, nspOS is a RPM based deployment.

Additional components

A deployment of the NSP product can include the following additional components:

- **IP resource control** - service provisioning and activation, plus MPLS path computation and traffic flow management. IP resource control uses flow-based protocols such as OpenFlow and BGP FlowSpec to perform intelligent traffic steering and to automate policy based redirection at the flow or route level. It also manages the creation of LSPs across IP network elements (NEs), and supports RSVP and segment routing LSP technologies. IP resource control provisions services using operator-defined policies across multi-domain networks. It works with other NSP components to perform service provisioning to specific elements. IP resource control requires the deployment of a VSR-NRC.
- **Cross Domain resource control** - optimizes network resources across different layers and domains of IP/MPLS and optical networks. Cross domain resource control processes information from other NSP components to discover the entire transport topology, including the cross layer links between IP routers and optical switches.
- **Simulation tool** - a traffic engineering tool that can be used by network engineers to design a new network, or optimize and simulate failures in an existing network that is imported into the tool. A network topology can be imported into the Simulation server from an IP resource control server.
- **Virtual Service Router - Network Resource Controller (VSR-NRC)** - must be deployed in order for IP resource control to interface with IP NEs for PCE-PCEP and OpenFlow communications. The VSR-NRC is a virtual SROS instance that uses the same software image as the vSIM of the same SROS release number; the VSR-NRC license enables additional code to interact with the NSP; this software can only run on a Linux KVM environment or VMWare ESXi versions 6.5 or 6.7. For platform requirements and installation instructions, see the *Virtualized 7750 SR and 7950 XRS Simulator (vSIM) Installation and Setup Guide*.

Element management systems

A deployment of the NSP product can also include the following element management systems:

- **Network Functions Manager - Packet (NFM-P)** - formerly 5620 SAM. The NFM-P is an advanced IP/MPLS and mobile network management system that has a modular, scalable architecture. The system provides multiple GUI, web, and OSS interfaces, and can integrate with other management systems.
- **Network Functions Manager - Transport (NFM-T)** - formerly 1350 OMS. The NFM-T provides unified optical end-to-end network management and operational support for all network element products in the Nokia's optics portfolio.

1.2 NSP deployment overview

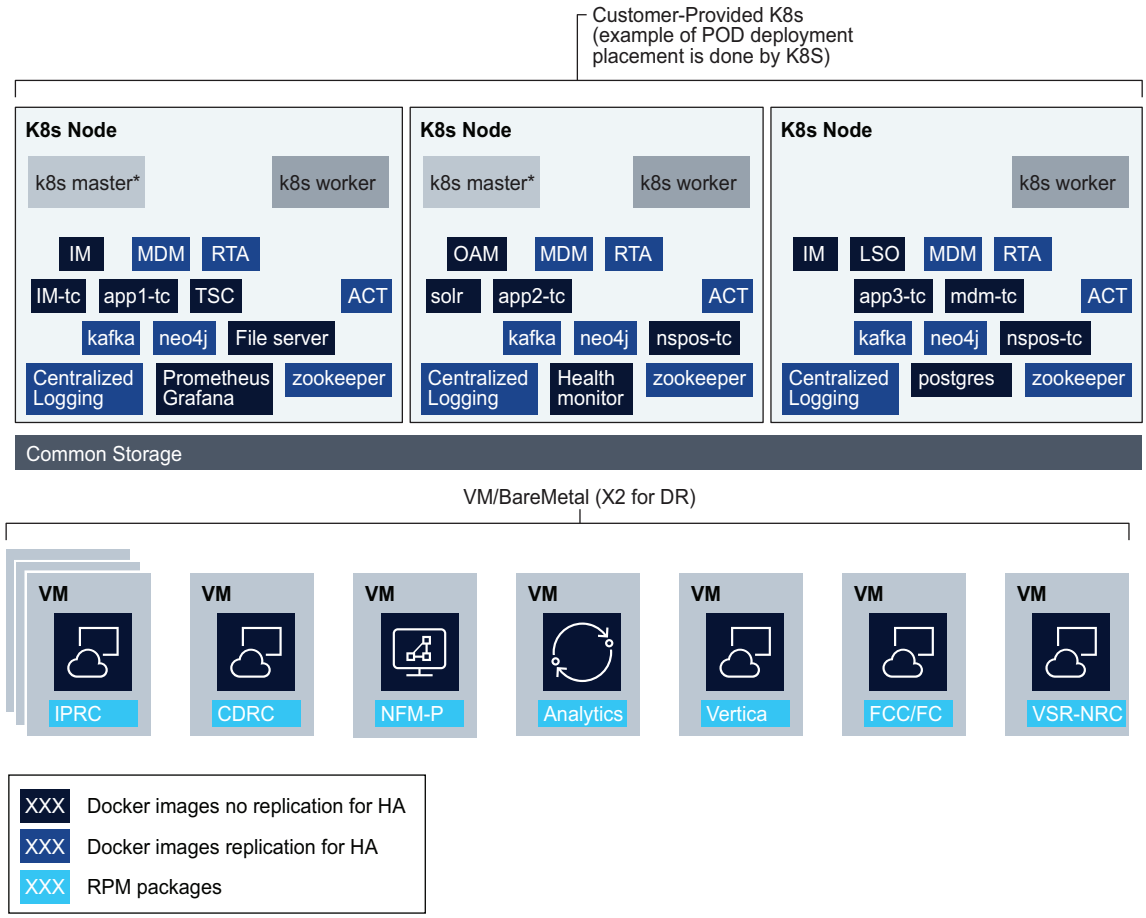
1.2.1 NSP deployment types

A NSP deployment consists of 2 parts:

- A containerized portion of the deployment (NSP cluster), that uses an orchestration layer to coordinate deployment of NSP services on a Kubernetes cluster.
- An RPM-based portion of the deployment that installs certain NSP components on different

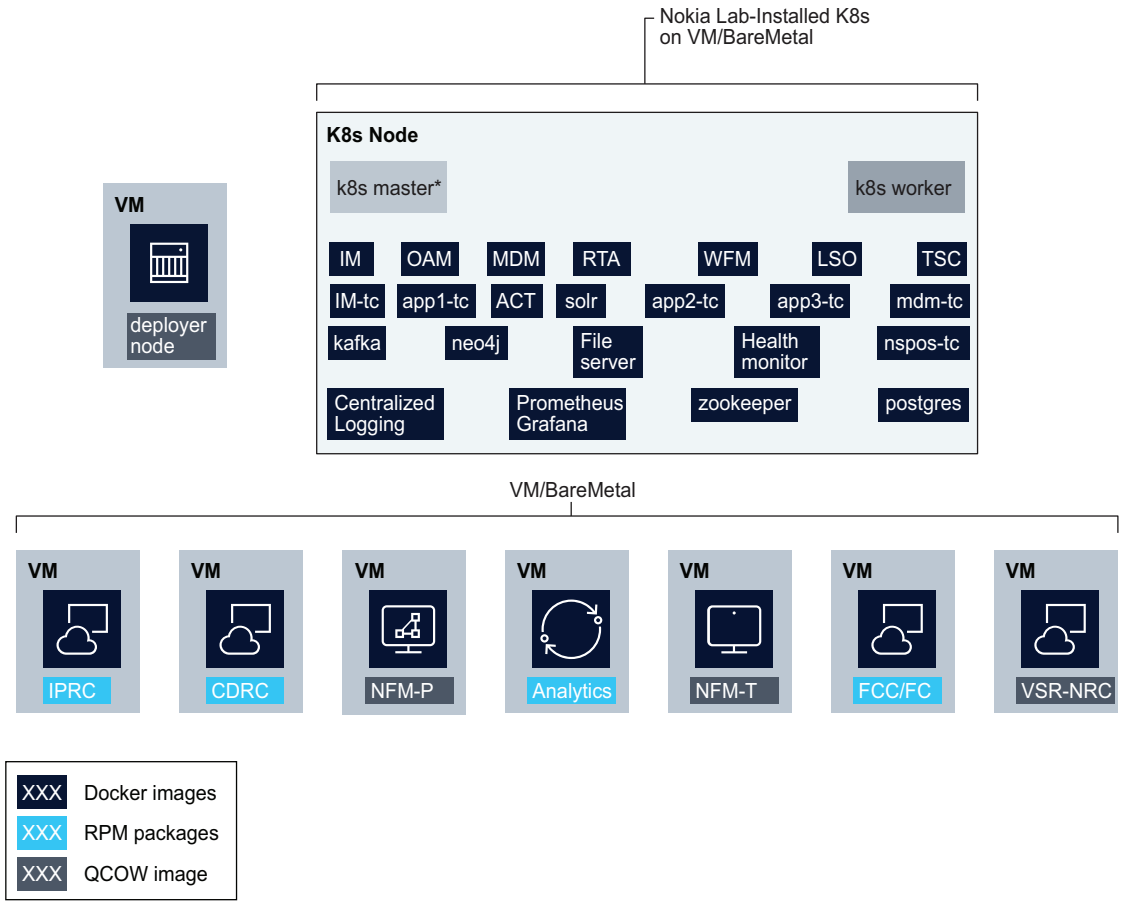
virtual machines. These components (IP resource control, Cross domain resource control) are installed on separate virtual machines.

Figure 1-1 NSP deployment example



As shown in the following figure, NSP also supports a containerized lab deployment:

Figure 1-2 Containerized NSP lab deployment example



See the Containerized Lab Installer Reference for details on installation of the containerized NSP lab deployment.

1.2.2 Redundancy and high availability overview

The NSP components support deployment in standalone or redundant configuration. Some components support high availability of services which are supported in both standalone and redundant configurations. A NSP cluster supports high availability of certain services (called an enhanced cluster) through pod replicas. RPM based components support a high availability cluster configuration on multiple virtual machine instances. High availability of services provides minimal downtime for a primary instance failure and avoids a full system switchover to redundant NSP datacenter.

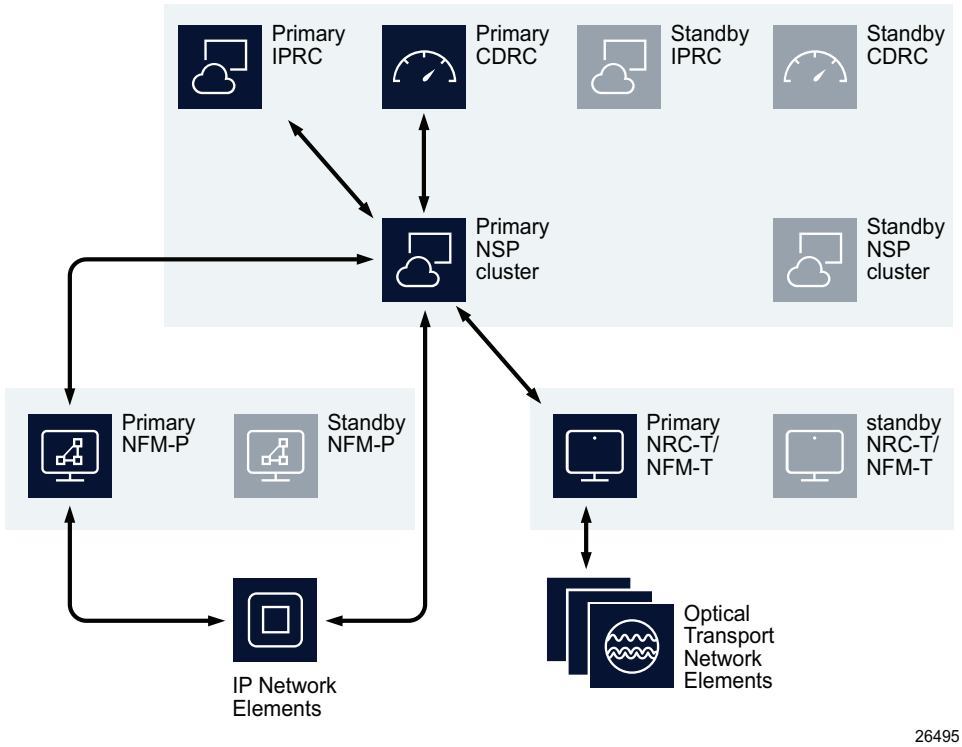
NSP can be deployed with NFM-P and NFM-T systems, which also support standalone and redundant configurations. Where NSP is deployed with NFM-P and/or NFM-T, all components must be deployed in similar configurations - as either standalone or redundant systems. Mixed

redundancy configuration of components is not supported. (A redundant NSP deployment supports both classic HA and fast-HA NFM-T deployment. Refer to the *NSP 21.6 Release Notice* for compatibility.)

In a fully redundant deployment of NSP with NFM-P and NFM-T, the primary NSP cluster instance operates independently of the primary activity of the NFM-P and NFM-T systems. If the NSP cluster performs an activity switch, the new primary instance will reconnect to the primary NFM-P and NFM-T. Likewise, if NFM-P or NFM-T perform an activity switch, the primary NSP cluster will connect to the new primary NFM-P / NFM-T.

The following figure shows a fully redundant deployment of these NSP components:

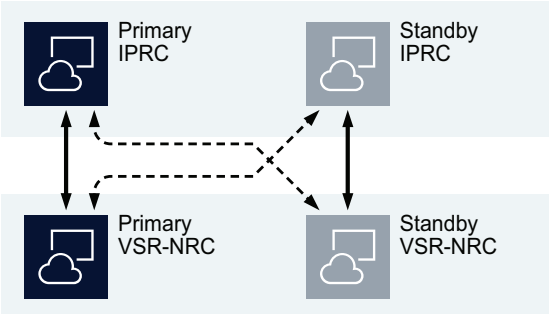
Figure 1-3 Redundant deployment of NSP components



The VSR-NRC can be deployed as a standalone or in a redundant configuration. A standalone IP resource control server can be deployed with a standalone or redundant VSR-NRC. When IP resource control is installed with a redundant VSR-NRC, if the communication channel between IP resource control and primary VSR-NRC fails, then the IP resource control will switch communications to the standby VSR-NRC.

In a DR or HA-DR deployment of IP resource control, a redundant deployment of VSR-NRC is required. The primary IP resource control server will communicate to NEs through the primary VSR-NRC, but if the communication channel to primary VSR-NRC fails, then the primary IP resource control server can switch to the standby VSR-NRC.

Figure 1-4 Redundant NSP deployment with redundant VSR-NRC



When an activity switch takes place between redundant IP resource control servers, the new active IP resource control server communicates with IP NEs through its corresponding VSR-NRC instance.

1.2.3 IP resource control redundancy and high availability

IP resource control can be deployed as a standalone system, as a redundant pair (1+1 or DR), as a high availability cluster, or as a high availability redundant pair (3+3 or HA DR). IP resource control should be deployed with the same redundancy and high availability configuration as the NSP cluster.

In a DR deployment, the activity of the IP resource control component follows the activity of the co-located NSP cluster. If the co-located NSP cluster switches to standby, IP resource control will also switch to standby. If the active IP resource control component fails, the co-located NSP cluster will detect this and automatically switch to standby.

In a HA DR deployment, the cluster leader of IP resource control in a datacenter is determined by the local nspd daemon process. Like the DR IP resource control deployment, the activity of the IP resource control datacenter follows the activity of the co-located NSP cluster.

1.2.4 Cross Domain resource control redundancy and high availability

Cross Domain resource control can be deployed as a standalone system or as a redundant pair. Cross Domain resource control should be deployed with the same redundancy configuration as the NSP cluster. Cross Domain resource control must be deployed in conjunction with an NSP cluster.

Note: Cross Domain resource control does not support high availability deployment in NSP Release 21.6.

In a 1+1 deployment, the activity of the Cross Domain resource control component is independent of the health of the NSP cluster. The primary Cross Domain resource control instance will automatically reconnect to the primary NSP cluster if an activity switch takes place

1.2.5 MDM redundancy and high availability

The MDM is deployed within the NSP cluster. In a NSP cluster, only one MDM pod can be deployed on each node within the cluster (eg. a 3 node cluster can deploy 3 MDM pods). In a redundant NSP cluster deployment, each NSP cluster will have the same number of nodes and MDM instances.

A single node NSP cluster can only deploy one MDM pod. High availability of MDM will not be available in this configuration.

In a multi-node NSP cluster, the MDM pods are deployed in a N+M deployment (where N+M equals the number of nodes in the cluster), with N active MDM instances and M standby instances. If an active MDM instance fails, a standby MDM instance in the active cluster will take over management of the nodes that were managed by the failed MDM instance. When the failed instance recovers, it becomes a standby instance (it will not automatically revert to active). When more than M active MDM instances fail, a manual activity switch to the standby NSP cluster will be required. Each NSP cluster must have the same N+M configuration of MDM.

1.3 NSP key technologies

1.3.1 Java virtual machine

The NSP and additional components use Java technology. The installation package contains a Java Virtual Machine which is installed with the software. This is a dedicated Java Virtual Machine and does not conflict with other JVMs which may be installed on the same workstation. The NSP uses OpenJDK 8.

1.3.2 Databases

A NSP deployment has multiple databases. IP resource control has a Neo4j database for network topology information. The nspOS component has a PostgreSQL database for policy management and common applications data, and a Neo4j database for topology data for the Map Server. Cross domain resource control contains a Neo4j database for network topology information.

A Neo4j database contains a graphical representation of the network topology and its elements in a multi-layer graph. The installation of the Neo4j database is customized for, and must be dedicated to, the NSP. Data redundancy and replication within a high availability cluster is managed within the neo4j instances.

The PostgreSQL database contains non-topological NSP information, including policies, templates, and nspOS common model data. PostgreSQL is an open source database application.

PostgreSQL database redundancy is managed by the role-manager. In a redundant configuration of the NSP, the active NSP cluster hosts the primary PostgreSQL database. The standby NSP cluster hosts the standby PostgreSQL database.

In a high availability deployment of the NSP cluster, one PostgreSQL database in the NSP cluster is selected as the primary database and the other in the NSP cluster is standby. If the active pod fails, then the standby member is promoted to primary database. In a redundant high availability configuration of NSP, the standby datacenter databases are updated as standby databases.

In a NSP cluster deployment where the customer provides the orchestration layer, each NSP cluster has one PostgreSQL database instance. Data replication of PostgreSQL database needs to be provided by the storage layer.

Note: Nokia does not support any PostgreSQL database configuration that deviates from the NSP installation procedure.

Nokia does not support direct customer access to the Neo4j and PostgreSQL databases.

1.3.3 Browser applications

NOTICE

View of network can be affected

Browser applications' view of the network can be affected whenever activities are drawing heavily on CPU and memory usage.

This can happen when a large number of services are being created, modified, or deleted via the NSP REST APIs.

The NSP provides functionality using browser-based applications. The NSP uses standard REST security mechanisms for authentication and authorization. All NSP applications are HTML-5 based and are supported on the latest desktop version of Google Chrome. The browser applications require that WebGL be enabled.

Localized language support

All NSP applications support 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 components and database objects. Contact Nokia technical support for more information about localized language support.

Note: The NSP components support localized language settings using predefined strings, and do not translate data to different languages.

1.3.4 API

The NSP applications provide a northbound REST API with Swagger-compliant documentation. The northbound API supports queries, service creation requests, and other functions. See the *NSP Developer Portal* for more information.

1.3.5 Network mediation

The NSP application has southbound interfaces that consist of plug-ins that interact with the NFM-P using CPROTO and HTTP protocols secured with TLS. The NFM-P manages IP network elements using SNMP.

The NSP communicates with MDM using gRPC, and MDM communicates with network elements using gRPC/gNMI, NETCONF, SNMP and CLI over SSH or Telnet.

For LSP management functions of the NSP, a VSR-NRC communicates with the PCC network elements via PCEP, IGP, and BGPLS. For flow control functions, the VSR-NRC OpenFlow Controller communicates with OpenFlow Switches using the OpenFlow protocol.

The NRC-T is installed with NFM-T to provide a TLS secured REST API for optical network discovery and service provisioning. The NFM-T uses TL-1 and SNMP to manage optical switches.

1.3.6 Container-based deployment technologies

Containerized deployments of NSP can be deployed in either:

- a Nokia-provided Kubernetes/Docker/Helm environment
- a customer-provided Kubernetes/Docker/Helm environment

Note: In NSP Release 21.6, the containerized deployment of NSP in customer-provided environments is Limited Availability. Contact your Nokia representative for more information.

2 Operating system specifications

2.1 Red Hat Enterprise Linux (RHEL)

2.1.1 Introduction

This chapter defines the operating system requirements for a deployment of NSP. These requirements also apply to NSP optional components. Please refer to NFM-P and NFM-T documentation for system requirements of those product deployments.

2.1.2 RHEL description and recommendations

Release 21.6 of the NSP is supported on the following RHEL server 7 x86-64 versions

Component	Supported RHEL versions
NSP cluster	7.7, 7.8, 7.9
IP resource control, Cross domain resource control	7.3, 7.4, 7.5, 7.6, 7.7, 7.8, 7.9

Previous releases, or other variants of Red Hat, and other Linux variants are not supported.

The Nokia provided RHEL OS qcow2 image is based upon RHEL 7.9 and is only available for KVM and Openstack hypervisors. The NSP RHEL qcow2 image can be used only for the deployment of NSP software, and not for the deployment of any other Nokia or third-party product.

The NSP does not necessarily support all functionality provided in RHEL. SELinux and Network Manager are not supported in NSP deployments. The NSP should use a time synchronization mechanism, such as NTP, to ensure accurate time. The NSP also requires that the server hostname is configured in the */etc/hosts* file. RHEL must be installed in 64 bit mode where NSP will be installed.

Customers are expected to purchase RHEL software and support for all platforms running RHEL Server with the NSP. It is strongly recommended to purchase a support package from Red Hat that provides 24x7 support.

Nokia recommends the installation of any OS, driver, or firmware updates that the hardware vendor advises for RHEL.

With the exception of documented Operating System parameter changes for NSP, all other settings must be left at the RHEL default configuration.

The *NSP Deployment and Installation Guide* provides detailed instructions for the RHEL OS installation.

2.1.3 Third-party applications

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

3 System resource requirements

3.1 Introduction

3.1.1 Overview

This chapter defines the system resource requirements for a successful deployment of NSP. Follow these guidelines to ensure that NSP performs adequately.

3.2 Virtual machine requirements

3.2.1 Overview

Nokia recommends that NSP components be installed on virtual machines using VMWare ESXi or RHEL KVM, including OpenStack. The Guest Operating System for a NSP deployment must be a supported version of RHEL.

Installations are server- and vendor-agnostic, but must meet any defined hardware criteria and performance targets to be used with the NSP modules. Server class hardware must be used, not desktops. Processor support is limited to specific Intel Xeon based x86-64 CPUs with a minimum CPU core speed of 2.4GHz. The CPU must be from the Haswell microarchitechture, or newer.

Defined CPU and Memory resources must be reserved for the individual Guest OSs and cannot be shared or oversubscribed. This includes individual vCPUs which must be reserved for the VM.

Provisioned CPU resources are based upon threaded CPUs. The NSP Platform Requirements will specify a minimum number of vCPUs to be assigned to the Virtual Machine. Virtual Machines should be configured with all vCPUs on one virtual socket.

A guest virtual machine must use only one time synchronization protocol such as NTP. Additional time synchronization applications must be disabled to ensure the proper operation of NSP.

Nokia support personnel must be provided with the details of the provisioned Virtual Machine. These details can either be provided through read-only access to the hypervisor or must be available to Nokia support when requested. Failure to provide these details could impact support of the NSP.

3.3 VMware virtualization

3.3.1 Overview

The NSP supports using VMware vSphere ESXi 6.7 only, on x86 based servers natively supported by ESXi. VMware’s Hardware Compatibility List (HCL) should be consulted to determine specific hardware support.

Not all features offered by ESXi are supported when using the NSP. For example, Fault Tolerant, High Availability (HA), Memory Compression, Distributed Resource Scheduler (DRS), and vMotion features are not supported. Contact Nokia to determine if a specific ESXi feature is supported with a NSP installation.

If using NTP or a similar time synchronization protocol on the guest virtual machine, then you must disable VMwareTools time synchronization.

Virtual Machine Version 11 or above must be used. The deployer host OVA file provided in the NSP software bundle is built using Virtual Machine Version 14.

See the following table for additional Virtual Machine setting requirements:

Table 3-1 Additional Virtual Machine setting requirements

Resource type	Parameter	Setting
CPU	Shares	Set to High
	Reservation	Must be set to the number of vCPUs * the CPU frequency. For example, on a 2.4 GHz 8 vCPU configuration, the reservation must be set to (8*2400) = 19200 MHz.
	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

3.4 KVM virtualization

3.4.1 Overview

The NSP supports using RHEL 6.3 through 6.7 KVM using QEMU version 0.12.1.2 and RHEL 7.2 through 7.9 KVM using QEMU version 1.5.3, 2.0.0, 2.3.0, 2.10.0 or 2.12.0 only, on x86 based servers natively supported by KVM. Consult the RHEL's Hardware Compatibility List (HCL) to determine specific hardware support.

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

3.4.2 Configuration

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

Table 3-2 KVM configuration parameters

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

3.5 OpenStack requirements

3.5.1 OpenStack support

NSP tests on open source OpenStack and will support the application running on any OpenStack distribution that is based on the version we test with. Any product issues deemed to be related to the specific distribution will need to be pursued by the customer with their OpenStack vendor. Supported versions include Newton, Queens and Train.

To ensure the stability of NSP and compatibility with OpenStack, you must follow the recommendations provided in this section.

3.5.2 Hypervisor

The only hypervisor supported within an OpenStack environment is KVM. For details about the KVM hypervisor supported versions, see 3.4 “KVM virtualization” (p. 22).

3.5.3 CPU and memory resources

Defined CPU and memory resources must be reserved and dedicated to the individual Guest OSs, and cannot be shared or oversubscribed. You must set both the cpu_allocation_ratio and ram_allocation_ratio parameters to 1.0 in the OpenStack Nova configuration either on the control NE or on each individual compute node where a VM hosting the NSP host server could reside.

3.5.4 HyperThreading

The usage of CPUs with enabled HyperThreading must be consistent across all compute nodes. If there are CPUs that do not support HyperThreading, then you must disable HyperThreading at the hardware level on all compute nodes where the NSP could be deployed.

3.5.5 CPU pinning

CPU pinning is supported but may restrict some OpenStack functionality like migration.

3.5.6 Availability zones/affinity/placement

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

3.5.7 Migration

The OpenStack environment supports only the regular migration. Live migration is not supported.

3.5.8 Networking

Basic Neutron functionality using Open vSwitch with the ML2 plugin can be used in a deployment of NSP. The use of OpenStack floating IP addresses is supported for the NSP.

3.5.9 Storage

All storage must meet the performance metrics provided with the NSP Platform Sizing Response. Performance must meet the documented requirements for both throughput and latency.

3.5.10 VM storage

The VM storage must be persistent block (Cinder) storage and not ephemeral. For each VM to be deployed, a bootable Cinder volume must be created. The size of the volume is indicated in the NSP Platform Sizing Response.

3.5.11 Flavors

Flavors must be created for each “Station Type” indicated in the NSP Platform Sizing Response.

3.5.12 Firewalls

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

3.6 Platform requirements

3.6.1 Overview

The virtual machine requirements for a deployment of NSP and its additional components depends on, but is not limited to, the following factors:

- Number of managed LSPs and services
- Number of managed elements
- Number of MDM learned services
- Number of simultaneous user and API sessions

- Expected number of flows, monitored routers, number of ASs, number of ports with real-time statistics collection

3.6.2 Minimum and production platform requirements

The following table lists the minimum and production platform requirements for deployment of IP resource control, VSR-NRC and Cross Domain resource control servers as described in the *NSP Deployment and Installation Guide*. The minimum and production platforms support the network dimensions described in Chapter 5, “Scaling and performance”.

Table 3-3 Platform requirements for IP resource control, VSR-NRC and Cross domain resource control deployment

Component	Minimum platform	Production platform
IP resource control	CPU: 8 vCPU Memory: 39 GB Disk space: 320 GB or more	CPU: 24 vCPU Memory: 64 GB Disk space: 400 GB
VSR-NRC	CPU: 4 vCPU Memory: 4 GB Disk space: 5 GB	CPU: 4 vCPU Memory: 8 GB Disk space: 5 GB
Cross Domain resource control	CPU: 8 vCPU Memory: minimum 16 GB Disk space: 210 GB or more	CPU: 16 vCPU Memory: 32 GB Disk space: 320 GB

Note: Verify that the VSR-NRC platform specifications are consistent with the specifications provided in the *Virtualized 7750 SR and 7950 XRS Simulator (vSIM) Installation and Setup Guide* for this release.

A deployment of a container based NSP software component in a Nokia-provided container environment will be sized according to the deployment type and the number of feature packages enabled. Each deployment will require a deployer node and one or more worker nodes. The worker nodes will require vCPU, memory and disk space as specified by the NSP Sizing Tool. These platform requirements support the network dimensions described in Chapter 5, “Scaling and performance”.

Note: Service Degradation Risk

The NSP deployer host is a crucial element of a NSP deployment that must remain reachable and operational by the NSP cluster after the initial deployment; otherwise, cluster recovery in the event of a failure may be compromised. The NSP deployer host holds the required Docker and Helm repositories, and pushes a containerization environment to each NSP host VM.

The following table shows the NSP cluster deployment types with number of nodes in a Nokia provided container environment.

Table 3-4 Platform requirements for NSP cluster deployment

Deployment	Basic/Medium	Standard	Enhanced (HA)
Kubernetes node count	1 node	3 or 4 nodes	7 nodes
Cluster deployer	CPU: 2 Memory: 4 GB Disk: 150GB	CPU: 2 Memory: 4 GB Disk: 150GB	CPU: 2 Memory: 4 GB Disk: 150GB

The NSP Sizing Tool will specify the overall vCPU, memory and disk space requirements, but generally, a Kubernetes worker node is a virtual machine with the following specifications:

- vCPU: 24
- Memory: 64GB
- Disk: 900GB (storage performance must have IOPS > 3000)

A Basic NSP cluster deployment requires 32 vCPU and 80 GB of memory and supports the NSP Platform feature package plus one additional feature package.

The Simulation Tool NSP deployment must be deployed as type “Lab”.

Note: The Kubernetes node count represents the number of nodes in a single NSP cluster. A redundant NSP cluster will require that number of nodes at each datacenter.

A NSP cluster deployment is supported in the following configurations with IP resource control.

- lab/basic/medium/standard NSP cluster with standalone IP resource control
- standard DR NSP cluster with DR IP resource control
- enhanced NSP cluster with HA IP resource control
- enhanced DR NSP cluster with HA DR IP resource control

3.7 System requirements

3.7.1 Overview

The hostname of an NSP server must meet the following criteria:

- can contain only ASCII alphanumeric characters and hyphens
- cannot begin or end with a hyphen
- if the hostname is a FQDN, period characters delimit the FQDN components
- the FQDN of the hostname cannot exceed 63 characters

A NSP server must be configured to perform name-service lookups before reverting to network-based name service such as NIS, NIS+ or DNS. The station also requires enough available user IDs to create new system users for NSP applications.

The layout and partitioning of the disks that contain the NSP software, operating data, and backups must be identical on each peer station in redundant and HA deployments.

3.8 Requirements for containerized deployments

3.8.1 Overview

The NSP cluster is deployed in either a Nokia provided container environment, or in a customer provided container environment. This section will identify system requirements specific to containerized NSP cluster deployments.

3.8.2 Platform requirements

A containerized deployment of the NSP cluster should follow guidelines of the customer's existing cloud infrastructure documentation and size the platform according to the requirements as provided by the Nokia NSP Platform Sizing Tool.

The virtual machine(s) with NSP cluster application requires kernel version kernel-3.10.0-1075.el7 or greater in combination with the following kernel setting:

```
--args=cgroup.memory=nokmem
```

The storage layer of a containerized NSP deployment requires a minimum read/write IOPS of 3000.

Note: Refer to Appendix A for commands on how to determine storage layer performance.

3.8.3 Kubernetes, Docker, and Helm requirements

NSP has been validated with the following versions of Kubernetes, Docker, and Helm:

kubeadm	v1.16.3
Docker client Docker server	18.09.7 18.09.7
Helm client Helm server	v2.16.1 v2.16.1

3.8.4 Operating system requirements

The operating system of the virtual machine(s) running the NSP applications requires a change to the virtual memory configuration (vm.max_map_count) to support Centralized Logging.

The current system setting can be determined with the following command as root user:

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

If the vm.max_map_count is not set to a minimum of 262144, then the following command should be executed before NSP deployment on the host OS:

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

4 Network requirements

4.1 Overview

4.1.1 Introduction

This chapter describes network bandwidth and latency requirements for a NSP deployment.

Nokia supports the deployment of NSP using the RHEL IP Bonding feature. The support for IP Bonding is intended only to provide network interface redundancy configured in active-backup mode for IP Bonding. All other modes of IP Bonding are not supported. RHEL documentation should be consulted on how to configure IP Bonding.

Note: The NSP only supports IPv4 connectivity with other components in the NSP architecture.

4.2 Network requirements between NSP and other components

4.2.1 NSP and OSS clients

The bandwidth requirements between NSP and OSS clients depends on the number of concurrent connections and on the type of transactions that are performed. For a single provisioning thread, Nokia recommends to provide 50 kbps of bandwidth from the OSS client to the NSP server (IP resource control, Cross Domain resource control, NSP cluster). An OSS client that performs frequent query operations (for example, port or service inventory) must be provided additional bandwidth.

4.2.2 NSP and GUI clients

The bandwidth requirements between NSP and GUI clients mostly depends on the size of the network. A larger network with more nodes and services will require more data to download to GUI clients. Optimal GUI performance is achieved with 10 Mbps of bandwidth with minimal network latency. Nokia recommends to provide a minimum of 2.5 Mbps of bandwidth.

High network latency between the NSP and GUI clients slows GUI performance. Nokia recommends to limit the round-trip network latency time to 100 ms.

4.2.3 NSP and NFMP

The bandwidth requirements between NSP and NFM-P depends on the following factors:

- the number of NEs, LSPs, and services configured on the NFM-P
- the frequency of NE updates to the NSP

When an NSP system re-synchronizes with NFM-P, optimal performance is achieved with 50 Mbps of bandwidth between NSP and NFM-P. Nokia recommends to provide a minimum of 25 Mbps of bandwidth.

Network latency impacts the time it takes for the NSP to re-synchronize a large amount of data from

the NFM-P. Nokia recommends to limit the round-trip network latency time to 100 ms.

4.3 Network requirements for redundancy, high-availability and NSP cluster nodes

4.3.1 Redundant deployments

The network requirements between active/standby IP resource control servers, Cross Domain resource control servers, and NSP clusters depends on the network size (number of NEs and configured services) and the rate of service provisioning activities. The peak bandwidth requirement between redundant servers is 50 Mbps, with sustained bandwidth of 25 Mbps. Round-trip network latency between the redundant pair must be limited to 100ms.

4.3.2 IP resource control high-availability deployment

IP resource control can be deployed in a high availability cluster on virtual machines from the same or different hosts and share a virtual IP address. Cluster members require connectivity with a minimum of 50 Mbps bandwidth and less than 1 ms round trip latency. The bandwidth and network latency requirements between active and standby high-availability clusters are the same as those in a redundant deployment.

4.3.3 NSP cluster nodes

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

5 Scaling and performance

5.1 Overview

5.1.1 Introduction

The following sections present the network dimension parameters for the minimum and production platforms described in section 3.6.2 “Minimum and production platform requirements” (p. 25).

5.2 Scale limits for NSP deployments

5.2.1 NSP deployment with classic and model-driven IP management

The following tables present key dimension details for a NSP deployment with classic and model-driven IP management as described in the *NSP Deployment and Installation Guide*.

Table 5-1 Dimensioning details for a NSP deployment with classic and model-driven IP management

Key dimension	Minimum platform	Production platform
Number of IP services managed	3000	300 000
Number of L2 access interfaces	5000	500 000
Number of L3 access interfaces	300	30 000
Combined total of RSVP-TE and SR-TE LSPs (non PCE controlled)	400	40 000
Number of service tunnels	600	60 000
Number of NFM-P managed services	17 000	1 700 000
Number of supported NEs	500	50 000

5.2.2 Control plane-only deployment

The following table presents key dimension details for a control plane-only deployment as described in the *NSP Deployment and Installation Guide*.

Table 5-2 Dimensioning details for control plane-only deployment

Key dimension	Minimum platform	Production platform
Total Number of LSPs	5000	120 000

Table 5-2 Dimensioning details for control plane-only deployment (continued)

Key dimension	Minimum platform	Production platform
Number of delegated RSVP-TE or SR-TE LSPs or both (PCE-Control, PCE-Compute and PCE-Report)	2000	50 000
Number of un-delegated RSVP-TE LSPs (only PCE-Report)	3000	70 000
Number of IP NEs (BGP-LS and PCEP)	1000	4500
Number of IP links	2000	22 000
Number of flows (applies to Traffic Steering Controller Application)	10 000	1 000 000

The following table presents key dimension details for a Simulation tool deployment.

Table 5-3 Dimensioning details for Simulation tool deployment

Key dimension	Minimum platform	Production platform
Total Number of LSPs	5000	20 000
Number of IP NEs	1000	3000
Number of IP links	3000	10 000

5.2.3 Scale limits for WFM deployment

The following table presents scaling dimensions for deployment of WFM.

Table 5-4 Dimensioning details for WFM deployment

Key dimension	Lab Platform	Production Platform
Number of stored executions (results)	500 000	1 000 000

5.3 Scale limits for Cross Domain resource control deployments

5.3.1 Cross Domain resource control scaling within NSP classic and model-driven IP + optical deployment

The following table presents key dimension details for Cross Domain resource control in a NSP classic and model-driven IP + optical deployment as described in the *NSP Deployment and Installation Guide*.

Table 5-5 Dimensioning details for Cross Domain resource control in a NSP classic and model-driven IP + optical deployment

Key dimension	Minimum platform	Production platform
IP NEs	100	3000
Optical NEs	100	3000
Ports	2400	240 000
Links	250	25 000
CDLs	100	10 000
Optical services	200	20 000
LLI	50	5000
IP-optical correlation	100	10 000

5.4 Scale limits for applications

5.4.1 Concurrent session limits

The following table represents the concurrent session limit for applications in a deployment with NSP.

Table 5-6 Concurrent session limits for NSP applications

Application	Maximum number of concurrent sessions
Analytics	10
Fault Management	125
Network Supervision	50
Service Supervision	125
Telemetry Monitor	100
Network Health Dashboard	20

Note: The combined concurrent session limit for all applications is 125.

The NSP can support up to 100 concurrent NE sessions on NFMP managed nodes, and 100 concurrent NE sessions on MDM managed nodes. Users must have execute level access control for a NE to create a NE session.

5.4.2 Scale limits for Telemetry

The maximum number of OSS subscriptions is 200. This number of OSS subscriptions includes, but is not limited to, Telemetry data.

Non-MDM Telemetry has one additional scaling limit:

- The maximum number of Telemetry notifications per second is 14,000, where one statistics counter update equals one Telemetry notification.

MDM Telemetry has the following two additional scaling limits per active MDM instance:

- The maximum number of Telemetry notifications per second is 1500, where one Telemetry record (a collection of statistics counters) update equals one Telemetry notification.
- The maximum number of rows uploaded to the database per minute is 90,000, where one row equals one Telemetry record.

These limits are per MDM instance and scale horizontally with the number of MDM instances in a cluster. Factors that may result in fluctuations of these targets include network activity, database activity and latency.

5.4.3 Scale limits for Fault Management

The following table defines the alarm limits for the Fault Management application:

Key dimension	Maximum number of alarms
Historical alarms from non-NFM-P systems (eg. NFM-T, MDM, NSP)	10 million
Active alarms from NFM-P, NFM-T, and/or MDM-managed nodes	500 thousand

The following table defines the squelching limits for the Fault Management application:

Key dimension	Maximum number of objects
Port squelching	1000 ports
Network Element squelching	1000 network elements
Resource group squelching	250,000 ports and/or network elements combined

Note: Because the maximum size for a port group is currently 100k (100,000) ports, multiple resource groups are needed to achieve the 250k squelching limit.

5.4.4 Scale limits for Network Supervision

The Network Supervision application supports up to 50 thousand monitored NEs.

- The maximum number of NEs per group is 2000.
- The maximum number of groups per view is 250.

The Link Utilization map view has limits on the number of supported endpoints and links that can subscribe for stats simultaneously. For NFM-P discovered nodes, a maximum of 400 ports (200 links) in a supervision group is supported for real time stats collection. If the number of qualified ports in the supervision group exceeds that limit, it may affect performance and topology rendering times.

The size of the supervision group (number of NEs and links) may affect performance and topology rendering time.

Multi-layer maps support a recommended maximum of 4000 objects. Users should expect the following multi-layer map loading times with different numbers of NEs.

- For 250 NEs (125 physical links), approximately six seconds for the initial page loading and four seconds to reload.
- For 500 NEs (250 physical links), approximately nine seconds for the initial page loading and six seconds to reload.
- For 2000 NEs (1000 physical links), approximately 50 seconds for the initial page loading and 28 seconds to reload.

5.4.5 Scale limits for Service Supervision

The Service Supervision application supports up to 1.7 million services.

- Maximum number of services per group is 50,000
- Maximum number of groups per view is 5000.

5.4.6 Scale limits of Group Manager

Where Cross Domain Resource Control is deployed, the following scaling limits for Map Layout will apply:

- Maximum number of nodes per region is 250.
- Maximum number of links per region is 1200.

Group directories have the following scaling limits.

- There is no limit on the number of directories for each directory type (Network Element, Port, Service, Analytics Resource).
- Maximum number of groups per directory is 5000.
- Maximum number of objects per group is 100,000.

5.4.7 Scale limits of Help Center application

The maximum number of concurrent sessions for Help Center Application is 250.

5.4.8 Scale limits for NSP Baseline Analytics application

The NSP Baseline Analytics application can support collection storage in the Postgres database or in the Auxiliary database. Baselines are supported on NFM-P and MDM managed nodes.

Key dimension	Postgres database storage	AuxDB storage
Number of baselines	10,000	100,000
Retention time	35 days	403 days
Collection Interval	900 seconds	900 seconds

The following limits apply for 20 Indicator rules.

Key dimension	Postgres database storage	AuxDB storage
Number of resources	10,000	100,000
NSP Indicator objects	2000	20,000

5.4.9 Scale limits for Large Scale Operations

The Large Scale Operations application has scaling limits for framework and for device operations.

The following table summarizes the framework limits.

Key dimension	Maximum
Number of parallel LSO executions	20
Number of stored operations (running and historical)	500
Number of operation types	100
Number of executions per operation	2000

The following table summarizes the device operation limits.

Key dimension	Maximum
Number of software images for Model Driven devices	20
Number of NE backups stored for Model Driven devices (with retention period of 7 days)	3657

The Large Scale Operations application has the following limitations in this release:

- Role Based Access Control will not apply to user operations
- Pre-check for deprecated hardware is not performed prior to NE upgrade on MDM managed nodes

5.4.10 Scale limits for ACT event processing

The following limits apply for ACT event processing.

Key dimension	standard NSP deployment	enhanced NSP deployment
Maximum kafka source notifications per second	2000	4000
Maximum kafka notifications per second for a single kafka source topic	2000	3000

5.4.11 Scale limits for Zero Touch Provisioning

The following limits apply to Day 0 Zero Touch Provisioning (ZTP):

Key dimension	Maximum
NE instances created per second	5
Simultaneous downloads from file server	10
ZTP instances in various provisioning states	1000

5.4.12 Scale limits for Generic Mediator

The Generic Mediator application has the following scaling limits:

Key dimension	Maximum
Concurrent threads	10
Request queue size	50

5.4.13 Scale limits for Network Health Dashboard

The Network Health Dashboard is supported to the following network scale limits;

- 15,000 network elements
- 15,000 physical links
- 300,000 ports
- 800,000 services
- 1.2 million sites
- 1 million SAPs
- 1 million SDP bindings
- 20,000 tunnels
- 15,000 LSPs
- 200,000 alarms
- Average trap rate of 35 traps/second

5.4.14 Web application performance and User Access Control

In a NSP deployment with User Access Control enabled, and more than 10 user groups are defined, in large networks (> 2000 NEs), NSP web application performance may be affected if the resource groups contain a very large number of equipment and/or service objects.

5.5 Failover performance for HA and redundant deployments

5.5.1 Overview

NSP components in a redundant configuration will experience application down time during an activity switch. NSP components in a high availability configuration may experience application down time when switching to a new cluster leader or during pod reselection.

In a NSP deployment with classic and model-driven IP management, the Service Fulfillment application enables service provisioning. Operators will experience a service provisioning outage during a leader switch within a high availability cluster, or during an activity switch in an active/standby deployment. The actual down time will vary based on network size.

Service Fulfillment		
	no MDM learned services 60k PCEP LSPs 3k NEs and 11k links	65k services learned from MDM 60k PCEP LSPs 3k NEs and 11k links
Down time for a switch of leader activity within an HA cluster	5 - 20 minutes	33 - 80 minutes
Down time for an activity switch from active to standby (HA cluster or single node) in a redundant deployment	16- 30 minutes	45 - 90 minutes

Note: Performance assumes that NFM-P is running release 18.12 SP5 or later.

Users and client applications that need access to Launchpad and NSP applications will also experience downtime during an activity switch and during pod re-selection in an enhanced deployment. Once Launchpad has been restored to service, northbound clients can authenticate and access applications.

Launchpad	
Down time for pod reselection	< 1s (see note)
Down time for an activity switch from active to standby (enhanced or single node) in a redundant deployment	9 minutes

Note: The Launchpad will remain available to users during a pod re-selection in an enhanced deployment; some applications on Launchpad may be temporarily unavailable due to pod rescheduling.

6 Security

6.1 Introduction

6.1.1 Overview

This chapter provides general information about platform security for a deployment of NSP. Recommendations in this section apply to NSP and its optional components except where indicated. Please refer to product documentation of NFM-P and NFM-T for deployments of those products.

The NSP 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*.

6.2 Securing the NSP

6.2.1 Overview

Nokia recommends that you to perform the following steps to achieve workstation security for the NSP:

- Install the latest recommended patch cluster from Red Hat (not supported on RHEL OS qcow2 images)
- NSP has no ingress or egress requirements to access the public internet and should be isolated with properly configured firewalls.
- Implement firewall rules to control access to ports on NSP systems, as detailed in this section
- Use SSL certificates with strong hashing algorithms.
- Enforce minimum password requirements and password renewal policies on user accounts that access the NSP applications.
- Configure a warning message in the Launchpad Security Statement.
- Configure login throttling to prevent denial of service attacks (see *NSP Deployment and Installation Guide*).
- Configure maximum session limits for administrators and users (see *NSP Deployment and Installation Guide*).
- Configure user lockout after a threshold of consecutive failed login attempts (see *NSP Deployment and Installation Guide*).
- When using custom TLS certificates for NSP deployment, ensure that the server private key file is protected when not in use by nsp configurator.
- Optional: Revoke world permission on compiler executables (see *NSP Deployment and Installation Guide*).

Refer to the *NSP System Architecture Guide* for NSP RHEL OS compliance with CIS Benchmarks. The supported CIS Benchmark best practices are already implemented on RHEL OS provided qcow2 images.

6.2.2 TLS communications

Communications of the NSP is secured using TLS. The NSP supports TLS versions TLSv1.2, TLSv1.1, and TLSv1.0.

The NSP supports the use of custom TLS certificates for client communications with NSP applications. Internal communications between NSP components can be secured with the use of a PKI server which can create, sign and distribute certificates. The NSP cluster software package provides a PKI server that can be used to simplify the TLS certificate distribution to NSP components.

A NSP cluster will check the expiry date of TLS certificates every 24h and raise an alarm in the Fault Management application if the certificate is expired or nearing expiry. Refer to the *NSP System Administrator Guide* for further information.

Refer to the *NSP Deployment and Installation Guide* for instructions on the configuration of custom TLS certificates and the provided PKI server application.

6.3 Operating system security for NSP workstations

6.3.1 RHEL patches

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

Operating system patches of NSP provided qcow2 images must be obtained from the NSP product group.

6.3.2 Platform hardening

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

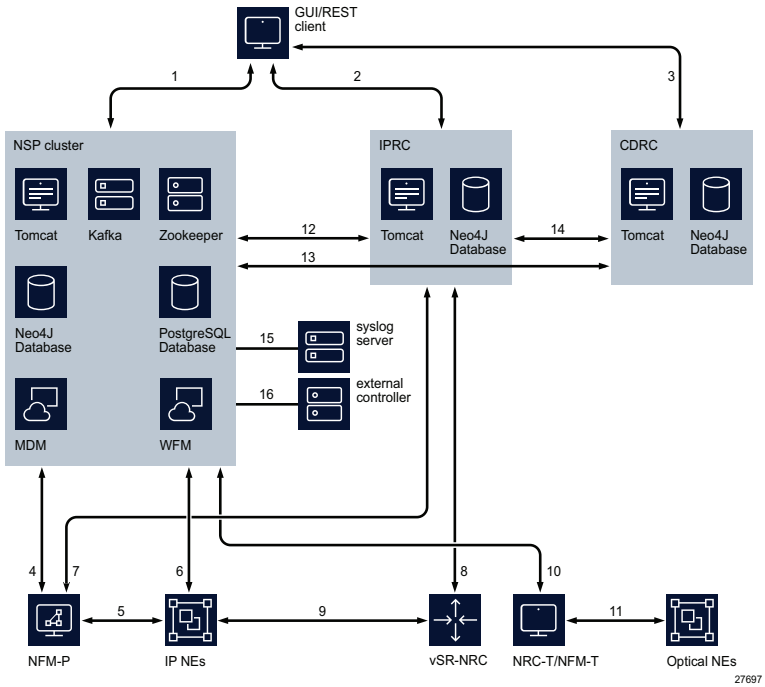
6.4 Communication between the NSP and external systems

6.4.1 Overview

The following diagrams illustrate the various components of the NSP and its internal communications, as well as communications with external systems.

The following figure shows a standalone NSP deployment and its communications with external systems.

Figure 6-1 Standalone NSP deployment



Connection	Usage
1, 2, 3	Web Client/REST API client connections. REST over HTTPS secured with TLS
4	SSO authentication (secure), zookeeper registration (secure), neo4j database (non-secure), kafka (secure), NFM-P API (secure), Data connection – CPROTO protocol secured with TLS
5	NE mediation using SNMP and FTP/SCP
6	NE mediation using gRPC/gNMI, SNMP, NETCONF, SSH
7	Data connection – CPROTO protocol secured with TLS
8	Data connection – not secured
9	BGP (supports GTSM), PCEP (secured by TLS), OpenFlow communications (secured by TLS) * Note

Connection	Usage
10	SSO authentication (secure), zookeeper registration (non-secure), REST over HTTPS secured with TLS, proprietary HTTP with NFM-T
11	NE mediation with SNMP and TL-1
12	SSO authentication (secure), zookeeper registration (secure), kafka (secure), PostgreSQL (secure), gRPC (secure), REST over HTTPS secured with TLS
13	SSO authentication (secure), zookeeper registration (secure), kafka (secure), gRPC (secure), REST over HTTPS secured with TLS
14	REST over HTTPS secured with TLS
15	syslog notifications secured with TLS
16	Mediator communications with external controller, REST/RESTCONF secured with TLS

Note: VSR-NRC supports secure PCEP and OpenFlow communications in specific releases. Refer to SR OS documentation for details.

6.5 NSP port information

6.5.1 Overview

The tables provided in this section identify the listening ports in a NSP deployment. For a complete listing of listening ports and firewall rules for NFM-P, refer to the *NFM-P Planning Guide*. For a complete listing of listening ports for NFM-T, refer to the *NFM-T Administration Guide: System Maintenance and Troubleshooting*.

Note: Host-based firewall applications should not block traffic on loopback interface ports declared in this guide.

Port changes for release 21.6:

- Addition of NSP cluster ports 7473, 7474, 7687, 8002, 8621, 9093, 9094, 9193, 9194, 9292, 9293, 9294
- Remove NSP cluster port 8500
- Addition of IPRC port 8180

Table 6-1 Port information for a NSP deployment

Default port(s)	Type	Encryption	Description	Deployment type
All applications				
22	TCP	Dynamic Encryption	SSH/SCP/SFTP Used for remote access and secure file transfer	All
NSP cluster				
80	TCP	None	HTTP port redirects to 443	All
162	UDP	None	SNMP traps	All
443	TCP	Dynamic Encryption provided by TLS	nspOS tomcat HTTPS port	All
2181	TCP	None	zookeeper (non-secure)	All
2281	TCP	Dynamic Encryption provided by TLS	zookeeper (secure)	All
2391	TCP	None	PKI server	All
4160	TCP	Dynamic Encryption provided by TLS	NF Management Function	All
5000	TCP	None	nspOS-neo4j	All
5001	TCP	None	nspOS-neo4j	All
5002	TCP	None	nspOS-neo4j	All
5601	TCP	Dynamic Encryption provided by TLS	Kibana web interface	All
6000	TCP	None	nspOS-neo4j	All
6001	TCP	None	nspOS-neo4j	All
6002	TCP	None	nspOS-neo4j	All
6432	TCP	Dynamic Encryption provided by TLS	PostgreSQL database	All
7000	TCP	None	nspOS-neo4j	All
7001	TCP	None	nspOS-neo4j	All
7002	TCP	None	nspOS-neo4j	All

Table 6-1 Port information for a NSP deployment (continued)

Default port(s)	Type	Encryption	Description	Deployment type
7473	TCP	Dynamic Encryption provided by TLS	nspOS-neo4j	All
7474	TCP	None	nspOS-neo4j	All
7576	TCP	None	Simulation tomcat neo4j browser	All
7687	TCP	Dynamic Encryption provided by TLS	nspOS-neo4j	All
7689	TCP	None	Simulation tomcat neo4j shell	All
8001	TCP	Dynamic Encryption provided by TLS	NSP Role Manager	All
8002	TCP	Dynamic Encryption provided by TLS	NSP Role Manager	All
8110	TCP	None	HTTP port for trap metrics	All
8120-8149	TCP	None	MD resync	All
8150-8179	TCP	None	MD deployer	All
8182	TCP	Dynamic Encryption provided by TLS	Analytics tomcat HTTPS port	All
8183	TCP	Dynamic Encryption provided by TLS	Analytics HTTPS port	All
8544	TCP	Dynamic Encryption provided by TLS	app1-tomcat HTTPS port	All
8545	TCP	Dynamic Encryption provided by TLS	app2-tomcat HTTPS port : Insights Administrator	All
8546	TCP	Dynamic Encryption provided by TLS	app3-tomcat HTTPS port : WFM	All
8547	TCP	Dynamic Encryption provided by TLS	MDT tomcat HTTPS port	All

Table 6-1 Port information for a NSP deployment (continued)

Default port(s)	Type	Encryption	Description	Deployment type
8548	TCP	Dynamic Encryption provided by TLS	MDM tomcat HTTPS port	All
8549	TCP	Dynamic Encryption provided by TLS	TSC tomcat HTTPS port	All
8551	TCP	Dynamic Encryption provided by TLS	LSOM tomcat HTTPS port	All
8555	TCP	None	Grafana	All
8561	TCP	Dynamic Encryption provided by TLS	NSP file service	All
8562	TCP	Dynamic Encryption provided by TLS	NSP file service HTTPS	All
8565	TCP	Dynamic Encryption	NSP file service SFTP	All
8566	TCP	None	NSP file service FTP	All
8567	TCP	Dynamic Encryption provided by TLS	NSP file service	All
8568	TCP	Dynamic Encryption provided by TLS	NSP file service RSYNC	All
8575	TCP	Dynamic Encryption provided by TLS	System Token Server	All
8617	TCP	Dynamic Encryption provided by TLS	nspOS AuxDB agent	All
8619	TCP	Dynamic Encryption provided by TLS	Insights Viewer HTTPS port	All
8620	TCP	Dynamic Encryption provided by TLS	MD OAM HTTPS port	All

Table 6-1 Port information for a NSP deployment (continued)

Default port(s)	Type	Encryption	Description	Deployment type
8621	TCP	Dynamic Encryption provided by TLS	Intent Based Service Fulfillment	All
9000	TCP	Dynamic Encryption provided by TLS	gRPC server	All
9092	TCP	None	kafka (non-secure)	All
9093, 9094	TCP	None	kafka (non-secure)	High availability
9192	TCP	Dynamic Encryption provided by TLS	kafka (secure)	All
9193, 9194	TCP	Dynamic Encryption provided by TLS	kafka (secure)	High availability
9200	TCP	Dynamic Encryption provided by TLS	Elasticsearch	All
9292	TCP	None / Dynamic Encryption provided by TLS	kafka (unsecure / secure)	All
9293, 9294	TCP	None / Dynamic Encryption provided by TLS	kafka (unsecure / secure)	High availability
9443	TCP	Dynamic Encryption provided by TLS	nspOS tomcat haproxy	All
9510	TCP	Dynamic Encryption provided by TLS	MDT Resource Manager HTTPS port	All
9543	TCP	Dynamic Encryption provided by TLS	Simulation tomcat	All
24224	TCP	None	fluentd	All
30000	TCP	Dynamic Encryption provided by TLS	MDM	All
40450	TCP	None	Analytics windower app	All

Table 6-1 Port information for a NSP deployment (continued)

Default port(s)	Type	Encryption	Description	Deployment type
40451	TCP	None	Analytics trainer app	All
40452	TCP	None	Analytics anomaly app	All
IP resource control				
2390	TCP	Dynamic Encryption provided by TLS	nspdctl	All
5001	TCP	None	Neo4j database	All
5798	TCP	None	Ignite communication within cluster	HA 3+3
6017	TCP	None	Neo4j database	All
6018	TCP	None	Neo4j database	1+1 3+3
6362	TCP	None	Neo4j database Local port to the host	All
7575	TCP	None	Neo4j database Local port to the host	All
7676	TCP	None	Neo4j database Local port to the host	1+1 3+3
7688	TCP	None	Neo4j database Local port to the host	All
7689	TCP	None	Neo4j database Local port to the host	1+1 3+3
8105	TCP	None	Java Tomcat Local port to the host	All
8180	TCP	None	HA Proxy	HA 3+3
8223	TCP	Dynamic Encryption provided by TLS	Java Tomcat	All
8224	TCP	Dynamic Encryption provided by TLS	Java Tomcat Local port to the host	All

Table 6-1 Port information for a NSP deployment (continued)

Default port(s)	Type	Encryption	Description	Deployment type
8543	TCP	Dynamic Encryption provided by TLS	Java Tomcat, secure HTTPS port for GUI and REST API	All
10800	TCP	None	Java Tomcat	All
11211	TCP	None	Ignite cache	All
47100–47199	TCP	None	Ignite cache	All
47500–47599	TCP	None	Ignite cache	All
Cross Domain resource control				
2390	TCP	Dynamic Encryption provided by TLS	nspdctl	All
5001	TCP	None	Neo4j database	All
6017	TCP	None	Neo4j database	All
6018	TCP	None	Neo4j database	1+1
6362	TCP	None	Neo4j database Local port to the host	All
7575	TCP	None	Neo4j database Local port to the host	All
7676	TCP	None	Neo4j database Local port to the host	1+1
7688	TCP	None	Neo4j database Local port to the host	All
7689	TCP	None	Neo4j database Local port to the host	1+1
8105	TCP	None	Java Tomcat Local port to the host	All
8223	TCP	Dynamic Encryption provided by TLS	Java Tomcat	All
8543	TCP	Dynamic Encryption provided by TLS	Java Tomcat, secure HTTPS port for GUI and REST API	All
10800	TCP	None	Java Tomcat	All

Table 6-1 Port information for a NSP deployment (continued)

Default port(s)	Type	Encryption	Description	Deployment type
11211	TCP	None	Ignite cache	All
47100 - 47199	TCP	None	Ignite cache	All
47500 - 47599	TCP	None	Ignite cache	All
VSR-NRC				
179	TCP	None	BGP	N/A
4189	TCP	None	PCEP	N/A
4199	TCP	None	CPROTO	N/A
6653	TCP	None	OpenFlow	N/A
NFM-P				
7879	TCP	Dynamic Encryption provided by TLS	CPROTO	N/A
8087	TCP	Dynamic Encryption provided by TLS	Web applications communications	N/A
8089	TCP	Dynamic Encryption provided by TLS	Web applications communications	N/A
8543	TCP	Dynamic Encryption provided by TLS	Web applications communications	N/A
NRC-T				
8543	TCP	Dynamic Encryption provided by TLS	REST API	N/A
PKI server				
2391	TCP	None	PKI server	N/A
NSP Analytics Server				
8005	TCP	None	tomcat shutdown port Local port to the host	N/A
8080	TCP	None	HTTP web user interface	N/A

Table 6-1 Port information for a NSP deployment (continued)

Default port(s)	Type	Encryption	Description	Deployment type
8443	TCP	Dynamic Encryption provided by TLS	HTTPS web user interface	N/A
10990	TCP	Dynamic Encryption provided by TLS	HTTPS for JMX console	N/A

6.6 NSP cluster deployer and worker node ports

6.6.1 Overview

The tables provided in this section identify the listening ports on a deployer node and on worker nodes for a NSP cluster deployment.

Table 6-2 Ports used by deployer node

Default port(s)	Type	Application
22	TCP	SSH
80	TCP	HTTP
111	TCP	rpcbind
443	TCP	HTTPS
6443	TCP	kubernetes
6444	TCP	kubernetes
8080	TCP	HTTP
8443	TCP	HTTPS
10010	TCP	kubernetes
10248	TCP	kubernetes
10249	TCP	kubernetes
10250	TCP	kubernetes
10251	TCP	kubernetes
10252	TCP	kubernetes
10256	TCP	kubernetes
30219	TCP	kubernetes
30817	TCP	kubernetes

Table 6-2 Ports used by deployer node (continued)

Default port(s)	Type	Application
31768	TCP	kubernetes
31910	TCP	kubernetes

Table 6-3 Ports used by worker nodes

Default Port(s)	Type	Application
22	TCP	sshd
53	TCP	node-cache
111	TCP	rpcbind
179	TCP	bird
2379	TCP	etcd
2380	TCP	etcd
6443	TCP	nginx
8081	TCP	nginx
9099	TCP	calico-node
9100	TCP	node exporter
9253	TCP	node cache
9254	TCP	node cache
9353	TCP	node-cache
10248	TCP	kubelet
10249	TCP	kube-proxy
10250	TCP	kubelet
10251	TCP	kubernetes
10252	TCP	kubernetes
10256	TCP	kube-proxy
10257	TCP	kubernetes
10259	TCP	kubernetes
30000	TCP	kube-proxy
30368	TCP	kube-proxy
31443	TCP	kube-proxy

6.7 NSP firewall rules

6.7.1 Overview

A firewall can be deployed in a NSP deployment to protect the NSP from different networks and applications. The firewall rules to be applied to an NSP deployment will depend on the deployment configuration (as described in the *NSP Deployment and Installation Guide*) and network topology.

The firewall rules in this section define traffic from the originating port to destination port. Where firewall rules are applied bidirectionally, the return path must also be permitted.

Firewall rules may be required for the software installation of IPRC and CDRC servers. The install server will need TCP access from an ephemeral port to port 22 on all IPRC and CDRC servers in the deployment.

The ephemeral port range of different server types may vary. Many Linux kernels use the port range 32768 - 61000. To determine the ephemeral port range of a server, execute

cat /proc/sys/net/ipv4/ip_local_port_range

Some NSP operations require idle TCP ports to remain open for long periods of time. Therefore, a customer firewall that closes idle TCP connections should adjust OS TCP keep-alives to ensure that the firewall will not close sockets that are in use by the NSP.

Communications with Zookeeper and Kafka are secure by default. The firewall tables below will reference both secure and unsecure Zookeeper and Kafka ports. Customers should only enable those ports as required for their configuration.

Note: For an IP + Optical deployment, the NSP cluster needs firewall rules for NSP/NFM-P and NSP/NFM-T.

6.7.2 NSP cluster firewall rules

This section lists the firewall rules for NSP cluster communications with other components.

In a multi-node deployment of NSP cluster, each node in the NSP cluster will require a firewall rule to allow communications from that node's IP address to the destination IP/port. Communications originating from another system will only need to connect to the NSP cluster virtual IP (and not the IP addresses of each node in the cluster). This also applies for communications between active/standby NSP clusters.

When NSP is deployed as an enhanced configuration, Kafka clients will require communications to the following NSP cluster ports

- unsecure Kafka client communications will connect to NSP cluster ports 9092, 9093, 9094, 9292, 9293, 9294
- secure Kafka client communications will connect to NSP cluster ports 9192, 9193, 9194, 9292, 9293, 9294

The firewall tables in this section will only reflect the non-enhanced NSP cluster Kafka ports.

Note: The use of firewalld is not supported on NSP cluster virtual machines. Nokia recommends using Calico policies to control traffic to a NSP cluster deployment.

Table 6-4 Firewall rules for traffic between NSP cluster and IP resource control

Protocol	From port	From component	To port	To component
TCP	>32768	IPRC	443	NSP
TCP	>32768	IPRC	2181 or 2281	NSP
TCP	>32768	IPRC	6432	NSP
TCP	>32768	IPRC	8544	NSP
TCP	>32768	IPRC	8575	NSP
TCP	>32768	IPRC	9092 or 9192, 9292	NSP
TCP	>32768	IPRC	30000	NSP
TCP	>32768	NSP	2390	IPRC
TCP	>32768	NSP	8543	IPRC

Notes:

1. In a HA deployment of IPRC, each IPRC server will require a firewall rule for each NSP destination port. Where NSP initiates a connection to IPRC, the destination IP address will be the virtual IP of the IPRC HA cluster.

Table 6-5 Firewall rules for traffic between NSP cluster and NFM-P

Protocol	From port	From component	To port	To component
TCP	>15000	NFM-P	443	NSP
TCP	>15000	NFM-P	2181 or 2281	NSP
TCP	>15000	NFM-P	6432	NSP
TCP	>15000	NFM-P	7473	NSP
TCP	>15000	NFM-P	7687	NSP
TCP	>15000	NFM-P	8544	NSP
TCP	>15000	NFM-P	8575	NSP
TCP	>15000	NFM-P	9092 or 9192, 9292	NSP
TCP	>32768	NSP	7879	NFM-P
TCP	>32768	NSP	8087	NFM-P
TCP	>32768	NSP	8089	NFM-P
TCP	>32768	NSP	8443	NFM-P

Table 6-5 Firewall rules for traffic between NSP cluster and NFM-P (continued)

Protocol	From port	From component	To port	To component
TCP	>32768	NSP	8543	NFM-P

Table 6-6 Firewall rules for traffic between NSP cluster and Cross Domain resource control

Protocol	From port	From component	To port	To component
TCP	>32768	CDRC	443	NSP
TCP	>32768	CDRC	2181 or 2281	NSP
TCP	>32768	CDRC	6432	NSP
TCP	>32768	CDRC	8544	NSP
TCP	>32768	CDRC	8575	NSP
TCP	>32768	CDRC	9092 or 9192, 9292	NSP
TCP	>32768	CDRC	30000	NSP
TCP	>32768	NSP	8543	CDRC

Table 6-7 Firewall rules for traffic between NSP cluster and NEs

Protocol	From port	From component	To port	To component
UDP	Any	NE	162	NSP
TCP	>32768	NSP	21	NE
TCP	>32768	NSP	22	NE
UDP	Any	NSP	161	NE
TCP	>32768	NSP	830	NE
TCP	>32768	NSP	57400	NE

Table 6-8 Firewall rules for traffic between NSP cluster and NRC-T/NFM-T

Protocol	From port	From component	To port	To component
TCP	>49192	NFM-T	443	NSP
TCP	>49192	NFM-T	2181 (see note)	NSP
TCP	>49192	NFM-T	6432	NSP
TCP	>49192	NFM-T	9092, 9292 (see note)	NSP

Table 6-8 Firewall rules for traffic between NSP cluster and NRC-T/NFM-T (continued)

Protocol	From port	From component	To port	To component
TCP	>32768	NSP	443	NFM-T
TCP	>32768	NSP	8443	NFM-T
TCP	>32768	NSP	8543	NRC-T

Notes:

1. Only unsecure communications is supported between NSP cluster and NFM-T.

Table 6-9 Firewall rules for traffic between NSP cluster and syslog server

Protocol	From port	From component	To port	To component
TCP	>32768	NSP	514	syslog server

Table 6-10 Firewall rules for traffic between NSP cluster and Analytics server

Protocol	From port	From component	To port	To component
TCP	>32768	NSP	8080	Analytics server
TCP	>32768	NSP	8443	Analytics server
TCP	>32768	Analytics server	443	NSP
TCP	>32768	Analytics server	2181 or 2281	NSP
TCP	>32768	Analytics server	6432	NSP
TCP	>32768	Analytics server	8544	NSP
TCP	>32768	Analytics server	9092 or 9192, 9292	NSP

Table 6-11 Firewall rules for traffic between NSP cluster and NFM-P Auxiliary server

Protocol	From port	From component	To port	To component
TCP	>15000	NFM-P Auxiliary server	2181 or 2281	NSP
TCP	>15000	NFM-P Auxiliary server	9092 or 9192, 9292	NSP

Table 6-12 Firewall rules for traffic between NSP cluster and NFM-P Auxiliary database

Protocol	From port	From component	To port	To component
TCP	>32768	NSP	5433	NFM-P Auxiliary database
TCP	>32768	NSP	7299-7309	NFM-P Auxiliary database

Table 6-13 Firewall rules for traffic between active and standby NSP clusters

Protocol	From port	From component	To port	To component
TCP	>32768	NSP	443	NSP
TCP	>32768	NSP	5000	NSP
TCP	>32768	NSP	5001	NSP
TCP	>32768	NSP	6000	NSP
TCP	>32768	NSP	6001	NSP
TCP	>32768	NSP	6432	NSP
TCP	>32768	NSP	8001	NSP

6.7.3 IP resource control firewall rules

When IP resource control is deployed in a high availability configuration, each IP resource control server will need to communicate to each destination IP/port in these firewall rules. Other components in a NSP deployment that need to communicate with IP resource control will need to communicate with the virtual IP of the IP resource control HA cluster. This also applies to communications between active and standby IP resource control systems.

Table 6-14 Firewall rules for traffic between IP resource control and NFM-P

Protocol	From port	From component	To port	To component
TCP	>32768	IPRC	7879	NFM-P

Table 6-15 Firewall rules for traffic between IP resource control and VSR-NRC

Protocol	From port	From component	To port	To component
TCP	>32768	IPRC	4199	VSR-NRC

Table 6-16 Firewall Rules for traffic between IP resource control and Cross Domain resource control

Protocol	From port	From component	To port	To component
TCP	>32768	IPRC	8543	CDRC
TCP	>32768	CDRC	8543	IPRC

Table 6-17 Firewall Rules for traffic between active and standby IP resource control servers

Protocol	From port	From component	To port	To component
TCP	>32768	IPRC	2390	IPRC
TCP	>32768	IPRC	5001	IPRC
TCP	>32768	IPRC	6017	IPRC
TCP	>32768	IPRC	6018	IPRC
TCP	>32768	IPRC	8223	IPRC

Table 6-18 Firewall Rules for traffic between IP resource control HA cluster members

Protocol	From port	From component	To port	To component
TCP	>32768	IPRC	2390	IPRC
TCP	>32768	IPRC	5001	IPRC
TCP	>32768	IPRC	5798	IPRC
TCP	>32768	IPRC	6017	IPRC
TCP	>32768	IPRC	8180	IPRC
TCP	>32768	IPRC	8223	IPRC
TCP	>32768	IPRC	8543	IPRC
TCP	>32768	IPRC	47100-47199	IPRC

6.7.4 Cross Domain resource control firewall rules

Table 6-19 Firewall Rules for traffic between active and standby Cross Domain resource control servers

Protocol	From port	From component	To port	To component
TCP	>32768	CDRC	2390	CDRC
TCP	>32768	CDRC	5001	CDRC
TCP	>32768	CDRC	6017	CDRC

Table 6-19 Firewall Rules for traffic between active and standby Cross Domain resource control servers (continued)

Protocol	From port	From component	To port	To component
TCP	>32768	CDRC	6018	CDRC
TCP	>32768	CDRC	8223	CDRC

6.7.5 NSP client connection firewall rules

Table 6-20 Firewall Rules for traffic between clients (GUI, API) and NSP components

Protocol	To port	To component	Purpose
TCP	80	NSP	HTTP port, redirects to 443
TCP	443	NSP	HTTPS port for Launchpad
TCP	5601	NSP	Kibana
TCP	8182	NSP	Analytics
TCP	8183	NSP	Analytics
TCP	8544	NSP	Common web applications
TCP	8545	NSP	MDM applications
TCP	8546	NSP	WFM
TCP	8547	NSP	MDM applications
TCP	8548	NSP	MDM applications
TCP	8549	NSP	TSC application
TCP	8551	NSP	LSO application
TCP	8555	NSP	Grafana
TCP	8619	NSP	Insights Viewer
TCP	8620	NSP	MD OAM application
TCP	9092 or 9192	NSP	kafka client connections
TCP	9510	NSP	MDT Resource Manager
TCP	9543	NSP	Simulation Tool GUI and REST API

Table 6-20 Firewall Rules for traffic between clients (GUI, API) and NSP components (continued)

Protocol	To port	To component	Purpose
TCP	8543	IPRC	GUI and REST API
TCP	8543	CDRC	GUI and REST API
TCP	8443	NFM-T	GUI
TCP	8543	NRC-T	REST API
TCP	8443	NFM-P	XML API
TCP	8543	NFM-P	Web applications and REST API

6.7.6 Simulation Tool firewall rules

A deployment of the Simulation Tool is a standalone deployment of NSP cluster with the Simulation app. Simulation Tool needs to authenticate with production network NSP cluster then use REST API to pull topology from IP resource control.

Table 6-21 Simulation Tool firewall rules

Protocol	From port	From component	To port	To component
TCP	>32768	Simulation Tool	443	NSP
TCP	>32768	Simulation Tool	8543	IPRC

6.7.7 NSP remote authentication firewall rules

Protocol	From port	From server	To port	To server	Purpose
TCP/UDP	>32768	NSP	389	LDAP	LDAP authentication
TCP/UDP	>32768	NSP	636	LDAP	TLS secured LDAP authentication
UDP	>32768	NSP	1812	RADIUS	RADIUS authentication
TCP	>32768	NSP	49	TACACS+	TACACS+ authentication

6.7.8 Firewall rules for PKI server

The NSP cluster deployment will use a locally installed PKI server for certificate generation. Deployment of other NSP components can use the same PKI server or another PKI server with the same internally generated private root CA.

Table 6-22 Firewall rules for PKI server

Protocol	From port	From server	To port	To server
TCP	>32768	IPRC	2391	PKI server
TCP	>32768	CDRC	2391	PKI server
TCP	>15000	NFM-P	2391	PKI server
TCP	>49192	NFM-T	2391	PKI server
TCP	>32768	Analytics Server	2391	PKI server

6.7.9 Firewall rules for deployer and worker nodes communications

The following firewall rules are required for communications between a deployer node and worker nodes when deploying a NSP cluster.

Protocol	From port	From server	To port	To server
TCP	>32768	deployer	22	worker
TCP	>32768	worker	80	deployer
TCP	>32768	worker	443	deployer
TCP	>32768	worker	8080	deployer

7 NSP deployment with multiple network interfaces and IP addresses

7.1 Support for multiple network interfaces

7.1.1 Introduction

The NSP and its associated components communicate with different entities that usually exist in different network spaces. Isolating different types of traffic to different networks provides better security and helps manage traffic volume on different networks.

NSP 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 NSP GUI and to connect external OSS systems to NSP.
- An internal network interface can be used to handle traffic between NSP 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 components.
- A mediation network interface can be used to communicate with network elements (provisioning, NE database backups, monitoring, operations, etc).

7.1.2 Component support and limitations

A NSP cluster can be configured with network interfaces for client traffic, for internal network management traffic, and for managed network traffic. In a multi-node NSP cluster, each node must have the same number of interfaces. A deployer node must be available to the NSP cluster nodes on the internal network.

An IP resource control server can be deployed network interfaces for client traffic and for internal network management traffic. In a high availability deployment of IP resource control, each server must have the same number of network interfaces. An IP resource control server can communicate with a VSR-NRC using any local network interface.

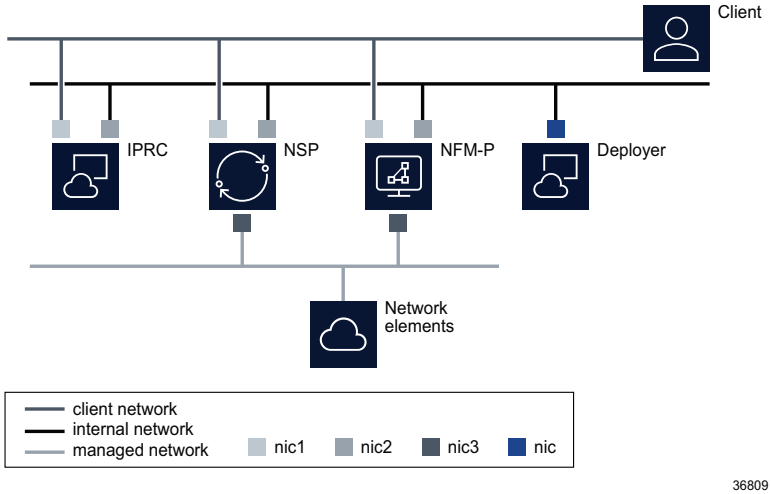
A Cross Domain resource control server can be deployed with network interfaces for client traffic and for internal network management traffic.

The NFM-P supports integrated deployment with NSP cluster and IP resource control server using network interfaces for client, internal and managed network traffic. Details of NFM-P support for multiple network interfaces is described in the *NFM-P Planning Guide*.

An integrated deployment of multi-interface NSP cluster and IP resource control with an older release NFM-P is supported but a workaround procedure is required on the NFM-P system. The workaround will enable the separation of client and internal network traffic between NFM-P and NSP cluster and IP resource control server. Refer to the *NSP Deployment and Integration Guide* for details on the integration procedure for older release NFM-P.

The following figure shows a multi network interface deployment of NSP cluster, deployer host, IP resource control server and NFM-P.

Figure 7-1 Multi-interface NSP deployment



The NSP Cluster, IP resource control and Cross Domain resource control can be deployed with one interface for all client, network management and NE traffic. The NSP cluster can be deployed with one interface for client and internal network traffic, and a second interface for NE traffic. All servers in a NSP deployment must support the same network configuration for client and internal networks. For example, a NSP cluster deployed with network interfaces for client and internal networks cannot be deployed with an IP resource control server that is configured for one interface supporting client and internal communications.

When installing NSP components on workstations with multiple network interfaces, each interface must reside on a separate subnet, with the exception of interfaces that are to be used in IP Bonding.

There is no requirement on the NSP cluster, IP resource control or Cross Domain resource control servers to use the first network interface (eg. eth0, bge0) to communicate with client applications.

Additional network interfaces can be configured on the NSP cluster, IP resource control and Cross Domain resource control servers, at the customer's discretion, for other operations such as archiving database backups or activity logs.

When a NSP cluster is deployed with NFM-T, the separation of client and internal network traffic is not supported. NSP and NFM-T must use a single network for client and internal communications.

When using custom TLS certificates in a multi-network configuration, the NSP cluster certificate will require the IP address of the client network interface (or virtual IP) and the IP address of the internal network interface (or virtual IP) in the certificate SAN field (parameters advertisedAddress and internalAdvertisedAddress in nsp-config.yml).

7.1.3 Multi-interface NSP deployment and firewalls

Customers can use firewall applications to protect NSP components but should be applied with care to ensure that NSP applications are not negatively impacted. NSP firewall rules are defined in Section 6.7 of this guide but need to be applied on the correct networks and network interfaces.

The following table summarizes the firewall rules for a NSP cluster deployment by each network or network interface.

Table 7-1 NSP cluster firewall communications by network interface

Network description	Permitted communications
client network	Client communications as defined in table 6-20 Kafa communications on ports 9092, 9093, 9094, 9192, 9193, 9194
internal network	All communications with NFMP, IPRC, CDRC and with redundant datacenter All communications between cluster nodes and deployer node as defined in section 6.7.9 Kafka communications on ports 9292, 9293, 9294
mediation network	Mediation communications as defined in table 6-7

Communications between the NSP cluster and remote authentication servers (LDAP, RADIUS, TACACS+), and with a remote syslog server, can use any network interface on the NSP cluster.

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

The following table summarizes the firewall rules for a IP resource control server by each network or network interface.

Table 7-2 IP resource control firewall communications by network interface

Network description	Permitted communications
client network	Client communications as defined in table 6-20
internal network	All communications with NSP, NFMP, CDRC and with redundant datacenter All traffic between HA IPRC cluster members as defined in table 6-18
mediation network	not applicable (see note)

Notes:

1. A dedicated network interface for mediation is not required on IP resource control server.

The IP resource control application can communicate with a VSR-NRC, and a PKI server, on any network interface.

When IP resource control is deployed in a high availability configuration, each server must allow the same traffic on each network interface.

The following table summarizes the firewall rules for a Cross Domain resource control server by each network or network interface.

Table 7-3 Cross Domain resource control firewall communications by network interface

Network description	Permitted communications
client network	Client communications as defined in table 6-20
internal network	All communications with NSP, IPRC and with redundant datacenter
mediation network	not applicable (see note)

Notes:

1. A dedicated network interface for mediation is not required on Cross Domain resource control server.

7.2 Network Address Translation

7.2.1 Overview

NSP supports the use of Network Address Translation (NAT) between the following components:

- NSP cluster and clients (web application users, REST API clients)
- NSP cluster and network elements
- NSP cluster and other components in the NSP deployment (eg. IPRC, NFM-P, CDRC)

NSP does not support the use of NAT between nodes within a NSP cluster deployment, including the deployer host.

8 Appendix A

8.1 Storage-layer I/O performance tests

8.1.1 Introduction

Use the commands in this section to determine if the storage-layer performance meets the NSP cluster deployment recommendations.

8.1.2 Determine disk speed

In this example, the /test directory is on the same disk where etcd runs.

Enter the following as the root user to run the test:

```
# fio --rw=write --ioengine=sync --fdatasync=1 --directory=/test
--size=22m --bs=2300 --name=mytest ↵
```

The command produces output like the following:

```
Starting 1 process
mytest: Laying out IO file (1 file / 22MiB)
Jobs: 1 (f=1)
mytest: (groupid=0, jobs=1): err= 0: pid=40944: Mon Jun 15 10:23:23 2020
  write: IOPS=7574, BW=16.6MiB/s (17.4MB/s)(21.0MiB/1324msec)
    clat (usec): min=4, max=261, avg= 9.50, stdev= 4.11
     lat (usec): min=4, max=262, avg= 9.67, stdev= 4.12
    clat percentiles (nsec):
     |  1.00th=[ 5536],  5.00th=[ 5728], 10.00th=[ 5920], 20.00th=[
6176],
     | 30.00th=[ 7584], 40.00th=[ 8896], 50.00th=[ 9408], 60.00th=[
9792],
     | 70.00th=[10432], 80.00th=[11584], 90.00th=[12864], 95.00th=
[14528],
     | 99.00th=[20352], 99.50th=[23168], 99.90th=[28800], 99.95th=
[42752],
     | 99.99th=[60672]
   bw (  KiB/s): min=16868, max=17258, per=100.00%, avg=17063.00,
stdev=275.77, samples=2
   iops        : min= 7510, max= 7684, avg=7597.00, stdev=123.04,
samples=2
```

```
  lat (usec)   : 10=64.21%, 20=34.68%, 50=1.08%, 100=0.02%, 500=0.01%
```

In the second block of output, which is shown below, the 99th percentile durations must be less than 10ms. In this block, each durations is less than 1ms.

```
fsync/fdatasync/sync_file_range:
    sync (usec): min=39, max=1174, avg=120.71, stdev=63.89
    sync percentiles (usec):
     |  1.00th=[   42],  5.00th=[   45], 10.00th=[   46], 20.00th=[
48],
     | 30.00th=[   52], 40.00th=[   71], 50.00th=[  153], 60.00th=[
159],
     | 70.00th=[  167], 80.00th=[  178], 90.00th=[  192], 95.00th=[
206],
     | 99.00th=[  229], 99.50th=[  239], 99.90th=[  355], 99.95th=[
416],
     | 99.99th=[  445]
  cpu          : usr=2.95%, sys=29.93%, ctx=15663, majf=0, minf=35
  IO depths    : 1=200.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%,
>=64=0.0%
     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=0.0%,
>=64=0.0%
     issued rwts: total=0,10029,0,0 short=10029,0,0,0 dropped=0,0,0,0
     latency   : target=0, window=0, percentile=100.00%, depth=1
```

8.1.3 Determine read / write IOPS

To run this test, first change to the directory where the test is to be performed. The test will create a local file. The output from the command contains read and write IOPS values. It is recommended that the storage layer provide a minimum IOPS of 3000.

Enter the following as the root user to run the test:

```
# 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 ↵
```

The command produces 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: Mon Sep 21 10:25:11 2020

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%
```