



Fabric Services System

SOFTWARE INSTALLATION GUIDE RELEASE 22.4

3HE 18706 AAAA TQZZA

Issue 1.0

May 2022

© 2022 Nokia.

Use subject to Terms available at: www.nokia.com/terms/.

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.

This document includes Nokia proprietary and confidential information, which may not be distributed or disclosed to any third parties without the prior written consent of Nokia.

This document is intended for use by Nokia's customers ("You"/"Your") in connection with a product purchased or licensed from any company within Nokia Group of Companies. Use this document as agreed. You agree to notify Nokia of any errors you may find in this document; however, should you elect to use this document for any purpose(s) for which it is not intended, You understand and warrant that any determinations You may make or actions You may take will be based upon Your independent judgment and analysis of the content of this document.

Nokia reserves the right to make changes to this document without notice. At all times, the controlling version is the one available on Nokia's site.

No part of this document may be modified.

NO WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY OF AVAILABILITY, ACCURACY, RELIABILITY, TITLE, NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, IS MADE IN RELATION TO THE CONTENT OF THIS DOCUMENT. IN NO EVENT WILL NOKIA BE LIABLE FOR ANY DAMAGES, INCLUDING BUT NOT LIMITED TO SPECIAL, DIRECT, INDIRECT, INCIDENTAL OR CONSEQUENTIAL OR ANY LOSSES, SUCH AS BUT NOT LIMITED TO LOSS OF PROFIT, REVENUE, BUSINESS INTERRUPTION, BUSINESS OPPORTUNITY OR DATA THAT MAY ARISE FROM THE USE OF THIS DOCUMENT OR THE INFORMATION IN IT, EVEN IN THE CASE OF ERRORS IN OR OMISSIONS FROM THIS DOCUMENT OR ITS CONTENT.

Copyright and trademark: Nokia is a registered trademark of Nokia Corporation. Other product names mentioned in this document may be trademarks of their respective owners.

© 2022 Nokia.

Table of contents

1 About this document.....	5
1.1 What's new.....	5
1.2 Precautionary and information messages.....	5
1.3 Conventions.....	5
2 Installation overview.....	7
2.1 Supported deployment models.....	7
2.1.1 Networking for the Fabric Services System nodes.....	7
2.1.1.1 DHCP and ZTP.....	7
2.1.2 Fabric Services System deployer.....	8
2.1.3 Fabric Services System nodes.....	8
2.1.4 Requirements.....	8
2.1.4.1 Fabric Services System deployer requirements.....	8
2.1.4.2 Fabric Services System node requirements.....	9
2.1.5 Installation process overview.....	9
3 Deploying and configuring the Fabric Services System deployer VM.....	10
3.1 Downloading the Fabric Services System deployer image.....	10
3.2 Preparing the Fabric Services System deployer hypervisor.....	10
3.3 Creating the Fabric Services System deployer VM.....	11
3.3.1 Creating the VM on a bridged network.....	11
3.4 Configuring the Fabric Services System deployer VM.....	12
4 Virtual machine-based installation: Using the Fabric Services System base OS image.....	14
4.1 Downloading the Fabric Services System base OS image.....	14
4.2 Networking considerations.....	14
4.3 Creating the Fabric Services System virtual machine.....	14
5 Bare metal-based installation: Preparing the Fabric Services System nodes.....	20
5.1 Disk partitioning.....	20
5.2 Installing the operating system, kernel, and packages.....	20
5.3 Configuring network settings.....	21
5.4 Setting up key-based authentication from the Fabric Services System deployer VM.....	21

6 Installing Fabric Services System.....	22
6.1 Editing the installation configuration file.....	22
6.2 Installing the Fabric Services System environment.....	27
6.3 Troubleshooting a failed installation.....	27
7 Reinstalling software on an existing Kubernetes cluster.....	28
7.1 Uninstalling Fabric Services System software only.....	28
7.2 Reinstalling Fabric Services System software only.....	29
8 Uninstalling software.....	30
8.1 Uninstalling a Fabric Services System deployment.....	30
8.2 Deleting the deployer VM.....	30

1 About this document

This document describes how to deploy the Fabric Services System software and the required software components such as the Kubernetes cluster and storage volumes.

This document is intended for network technicians, administrators, operators, service providers, and others who need to understand how to configure bare metal servers and provision the Fabric Services System for deployment.



Note: This document covers the current release and may also contain some content that will be released in later maintenance loads. See the *Fabric Services System Release Notes* for information about features supported in each load.

1.1 What's new

This is the first release of this document. In future releases, a table will define new or changed information for the release.

1.2 Precautionary and information messages

The following are information symbols used in the documentation.



DANGER: Danger warns that the described activity or situation may result in serious personal injury or death. An electric shock hazard could exist. Before you begin work on this equipment, be aware of hazards involving electrical circuitry, be familiar with networking environments, and implement accident prevention procedures.



WARNING: Warning indicates that the described activity or situation may, or will, cause equipment damage, serious performance problems, or loss of data.



Caution: Caution indicates that the described activity or situation may reduce your component or system performance.



Note: Note provides additional operational information.



Tip: Tip provides suggestions for use or best practices.

1.3 Conventions

Commands use the following conventions.

- **Bold** type indicates a command that the user must enter.

-
- Input and output examples are displayed in Courier text.
 - An open right angle bracket indicates a progression of menu choices or a simple command sequence (often selected from a user interface; for example, **start > connect to**).
 - Angle brackets (< >) indicate an item that is not used verbatim. For example, for the command **show ethernet <name>**, *name* must be replaced with the name of the interface.
 - A vertical bar (|) represents an OR, indicating that only one of the parameters can be selected.
 - Square brackets ([]) indicate optional elements.
 - Braces ({ }) indicate a required choice. When braces are contained within square brackets, they indicate a required choice within an optional element.
 - *Italic* type indicates a variable.

Examples use generic IP addresses. Replace these with the applicable IP addresses used in your system.

2 Installation overview

This chapter describes the Fabric Services System components, the requirements for these components, and provides an overview of the installation process.

2.1 Supported deployment models

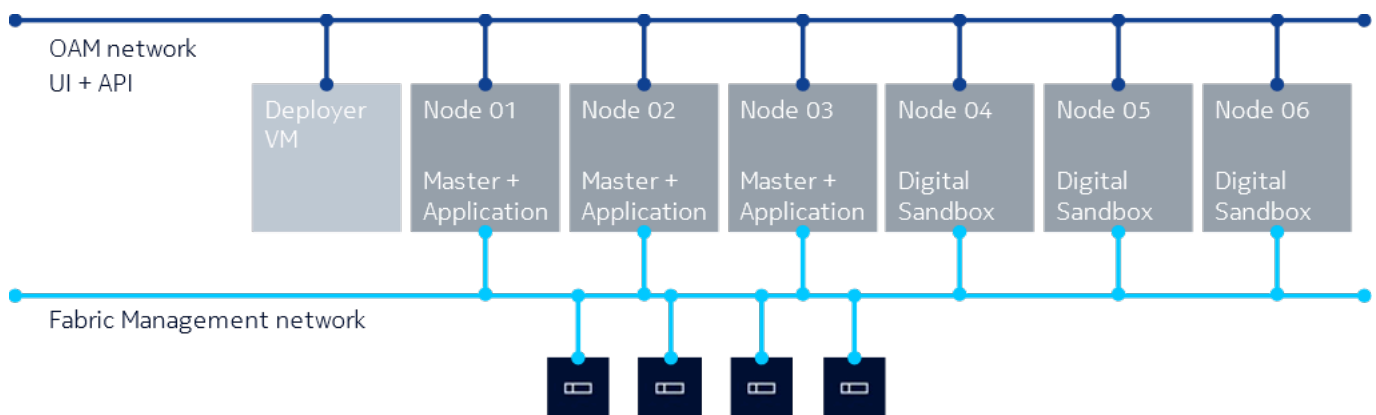
The Fabric Services System is deployed on three or six Fabric Services System nodes which can be deployed as bare metal servers or virtual machine servers.

2.1.1 Networking for the Fabric Services System nodes

Nokia recommends that you use two different networks for the Fabric Services System nodes:

- Fabric Services System OAM interface
This interface is used to access the UI and the API of the Fabric Services System. This is also the network through which the deployer VM reaches the node.
- Fabric management interface
This interface is used to communicate with the management interfaces of the fabric (for example, SR Linux devices). This interface is where the Fabric Services System exposes its DHCP and ZTP services.

Figure 1: Deployment with six Fabric Services System nodes, with Digital Sandbox enabled



2.1.1.1 DHCP and ZTP

The Fabric Services System deployment can serve as a DHCP service for the fabric devices using one of the following methods:

- All the management interfaces of the fabric devices are connected to the same Layer 2 network as the DHCP interface of the Fabric Services System

- DHCP relay

2.1.2 Fabric Services System deployer

The Fabric Services System deployer is a virtual machine (VM) that is used to provision a Kubernetes environment and install Fabric Services System microservices on the Fabric Services System nodes.

It contains all the software packages, container images, and Helm charts that are used by the deployer script to install the entire Fabric Services System environment.

The Fabric Services System deployer VM does the following:

- installs a Kubernetes cluster on the Fabric Services System nodes in your environment
- configures the nodes on the cluster for local storage
- labels the nodes to prepare them for Fabric Services System software deployment
- installs microservices using Helm charts and container repositories hosted in the deployer
- provides technical support tools



Caution: To maintain the security of the Fabric Services System deployment, ensure that access to the deployer host and the deployer VM are restricted and password protected.



Note: The deployer VM must remain powered on and must be reachable by the Fabric Services System cluster throughout its life cycle, as it serves as a container registry for all the Fabric Services System components. Additionally, the technical support scripts are hosted on the deployer VM and must be executed from the deployer VM.

2.1.3 Fabric Services System nodes

The Fabric Services System nodes are the bare metal servers or KVM-based VMs that host the Kubernetes environment on which the Fabric Services System application and Digital Sandbox are run.

The following deployment models are supported:

- A three-node environment that hosts only the Fabric Services System application and does not support the use of Digital Sandbox.
- An environment of a minimum of 6-node which hosts both the Fabric Services System application and does support the use of Digital Sandbox.

2.1.4 Requirements

This section details the specific requirements for the components and their connectivity.

2.1.4.1 Fabric Services System deployer requirements

The Fabric Services System deployer is delivered as a QCOW2 image that is used to launch a VM. The QCOW2 image must be used as the disk of a VM in a Kernel-based virtual machine (KVM) hypervisor.

The minimal requirements for the Fabric Services System deployer virtual machines are:

- CPU: 1 virtual CPU (vCPU) on a KVM hypervisor with a modern x86_64 CPU that supports virtualization. Recommended: 2 vCPU
- Memory: 8GB RAM
- Storage: 50GB of storage (used by the deployer QCOW2 image)
- Networking: 1GbE network interface card (NIC)

2.1.4.2 Fabric Services System node requirements

The Fabric Services System nodes can be deployed either as bare metal servers or as virtual machine servers. The following requirements apply to both types of deployments:

- CPU: 32 cores on a modern x86_64 CPU
- Memory: 64GB
- Storage: At least 500GB of available SSD-based storage
- Networking: At least one 10Gbps NIC

You can run the Fabric Services System nodes as virtual machines using the following virtualization platforms:

- Operating system: RHEL/CentOS 7.9
- Hypervisor: KVM
- Resource reservation for CPU, memory, and disks must be set to 100% for the Fabric Services System node virtual machines

2.1.5 Installation process overview

The installation consists of the following high-level tasks:

1. [Downloading the Fabric Services System deployer image](#)

This task describes how to access the Fabric Services System deployer image so it can be used in the next task.

2. [Deploying and configuring the Fabric Services System deployer VM](#)

This task describes how to create the VM in a KVM environment and how to configure the Fabric Services System deployer VM.

3. Preparing the Fabric Services System nodes

Complete the applicable procedure for your deployment scenario.

- [Virtual machine-based installation: Using the Fabric Services System base OS image](#)
- [Bare metal-based installation: Preparing the Fabric Services System nodes](#)

4. [Editing the installation configuration file](#)

The Fabric Services System deployer uses a JSON-based configuration file to identify the configuration and deployment of the Fabric Services System environment. This task describes the different variables and the expected data.

5. [Installing the Fabric Services System environment](#)

The deployment of the Fabric Services System environment involves the `sample-input.json` file.

3 Deploying and configuring the Fabric Services System deployer VM

The procedures in this section describe how to deploy and configure the Fabric Services System deployer VM.

3.1 Downloading the Fabric Services System deployer image

Contact Nokia support for the location of the Fabric Services System deployer image. Download the image to the deployer host, then extract the QCOW2 image.

3.2 Preparing the Fabric Services System deployer hypervisor

Prerequisites

Before you install the deployer VM, you must prepare the node on which you are installing the deployer VM. Virtualization must be enabled on the node, and can be enabled in the BIOS/EFI. You must also install a KVM environment on the node before installing the deployer VM.

Commands in this procedure require root privileges.

Procedure

Step 1. Confirm that virtualization is enabled on the node using one of the following commands, depending on your system chip set. If virtualization is not enabled, enable it in the BIOS/EFI before continuing.

- For an Intel system, enter:

```
$ lscpu | grep Virtualization
Virtualization: VT-x
```

- For an AMD system, enter:

```
$ lscpu | grep Virtualization
Virtualization: AMD-V
```

Step 2. Install the KVM environment.

```
$ yum install qemu-kvm libvirt libvirt-python libguestfs-tools virt-install -y
```

Step 3. Enable and start the libvirtd service on the installer node.

```
$ systemctl enable libvirtd
$ systemctl start libvirtd
```

3.3 Creating the Fabric Services System deployer VM

After you have downloaded the QCOW2 image and prepared the deployer node, follow the installation steps to create the VM.

The Fabric Services System nodes contained in the cluster (worker nodes) and the node hosting the deployer VM must communicate with each other. Both the worker nodes and the deployer VM must be able to initiate connections. You can configure networking for the deployer VM using a bridged network; see [Creating the VM on a bridged network](#).

3.3.1 Creating the VM on a bridged network

Prerequisites

This section provides an example script on how the VM can be created in a KVM-based hypervisor. You can use this script or you can use your own procedure as long as the resulting VM meets the requirements for the Fabric Services System VM.

Procedure

Step 1. Create an `fssvm_create.sh` file, then copy the following contents into the file:

```
create_fssvm() {
    BRIDGE="breth0:1"
    VM=fss-deployer
    VMDIR=/var/lib/libvirt/images/$VM
    FSSIMAGE=<path to fss-installer qcow2 image>
    sudo mkdir -vp $VMDIR
    sudo cp $FSSIMAGE $VMDIR/$VM.qcow2
    sudo virsh pool-create-as --name $VM --type dir --target $VMDIR
    sudo virt-install --import --name $VM \
--memory 8096 --vcpus 1 --cpu host \
--disk $VMDIR/$VM.qcow2,format=qcow2,bus=virtio \
--network bridge=$BRIDGE,model=virtio \
--os-type=linux \
--os-variant=centos7.0 \
--noautoconsole --debug
}
VMDIR=.
create_fssvm
```

Step 2. In the script, modify the `FSSIMAGE=<path to fss-installer qcow2 image>` field to show the actual path to the Fabric Services System image on your system.

```
FSSIMAGE=./fss-deployer-x.y.qcow2
```

Step 3. Modify the permissions of the shell script file.

```
chmod 755 fssvm_create.sh
```

Step 4. Execute the shell script.

```
./fssvm_create.sh
```

3.4 Configuring the Fabric Services System deployer VM

Procedure

Step 1. From the KVM console, log in to the deployer VM.

Use the following credentials:

Username: root

Password: N0ki@DCF4bricFSS1!



Note: After the initial login, Nokia recommends that you change this default password to a stronger password to enhance the security of the deployer and the Fabric Services System environment.

Step 2. If your environment does not support or use cloud-init services, disable and stop these services.

```
# systemctl stop cloud-init cloud-init-local cloud-config cloud-final
```

```
# systemctl disable cloud-init cloud-init-local cloud-config cloud-final
```

Step 3. Edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file to configure the correct static IP address, DNS servers, and gateway.

Example

The final content should look similar to the following, except with the IP address, DNS, and domain details specific to the target environment:

```
BOOTPROTO=static
DEVICE=eth0
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
IPADDR=192.0.2.10
PREFIX=24
GATEWAY=192.0.2.1
DNS1=192.0.2.5
DNS2=192.0.2.6
DOMAIN=fss.nokia.local
```

Step 4. Restart the network to apply the new configuration.

Execute the following command:

```
# systemctl restart network
```

Step 5. Configure the appropriate NTP servers.

Edit the `/etc/chrony.conf` configuration file and replace all lines beginning with `server` with the correct server lines for the environment.

Step 6. Restart the chronyd service.

```
# systemctl restart chronyd
```

Step 7. Verify that time synchronization is functioning properly.

```
# chronyc tracking
```

Expected outcome

If the **Reference ID** field does not refer to any of the configured servers, but instead refers to something like 127.127.1.1, time synchronization is not functioning properly.

```
Reference ID      : 87E30FFE (192.0.2.5)
Stratum          : 4
Ref time (UTC)   : Wed Feb 16 01:20:36 2022
System time      : 0.000014215 seconds slow of NTP time
Last offset      : -0.000001614 seconds
RMS offset       : 0.000106133 seconds
Frequency        : 11.863 ppm slow
Residual freq    : -0.071 ppm
Skew             : 0.187 ppm
Root delay       : 0.063009784 seconds
Root dispersion  : 0.018440660 seconds
Update interval  : 64.5 seconds
Leap status      : Normal
```

Step 8. Synchronize the RTC clock and the system clock.

Example

```
# hwclock --systohc
```

Then, verify that local time and the RTC time are synchronized.

```
# timedatectl
```

Step 9. Optional: Change the hostname.

```
# hostnamectl set-hostname new-hostname.domain.tld
```

Step 10. Reboot the Fabric Services System deployer VM to ensure that all services come up with the correct network configuration.

```
# reboot
```

4 Virtual machine-based installation: Using the Fabric Services System base OS image

For deployments that use virtual machine servers for the Fabric Services System nodes, follow this chapter to configure these nodes properly.

You must use the Fabric Services System base OS image. This image is specially designed for use with the Fabric Services System deployment and comes with the necessary software and components, pre-installed in a minimally-hardened CentOS 7.9 operating system.

Complete the procedure for each individual Fabric Services System node, ensuring that each node is running on a separate hypervisor to minimize the risk of any impact if a hypervisor fails.

4.1 Downloading the Fabric Services System base OS image

Contact Nokia support for the location of the Fabric Services System base OS image. Download the image to the deployer host, then extract the QCOW2 image.

4.2 Networking considerations

Nokia recommends that you use two different networks for the Fabric Services System nodes as described in [Networking for the Fabric Services System nodes](#).

Within the hypervisor, both networks should be available as bridged networks. In [Creating the Fabric Services System virtual machine](#), the first network is referred to as br0 and the second network is br1. Both these networks require support for jumbo frames (MTU 9000).

4.3 Creating the Fabric Services System virtual machine

About this task

Complete the following steps to deploy a Fabric Services System node as a virtual machine.

Procedure

Step 1. Ensure that the virt-install tool is installed on the KVM hypervisor.
If you need to install the tool, use the following command:

```
# yum install virt-install
```

Step 2. Copy the base OS image to the hypervisor in the appropriate location where the virtual disks should be stored.

Step 3. Resize the base OS image.

By default, the Fabric Services System base OS image comes with a small partition to lower the download size of the image. To assign the appropriate size to the image, execute the following command:

```
# qemu-img resize /path/to/fss-node01.qcow2 200G
```

The actual root partition is resized later in the procedure.

- Step 4.** Optional: If the node is also going to be used as a storage node, create the necessary extra disk for the storage cluster to be formed.

Example

Create the virtual disk using the following command:

```
# qemu-img create -f qcow2 /path/to/fss-node01-storage.qcow2 300G
```

- Step 5.** Create the virtual machine.

Example

The following command creates a node that also serves as a storage node. If a storage node is not needed, omit the second line that starts with `--disk`.

```
# virt-install --import --name fss-node01 \  
--memory 65536 --vcpus 32 --cpu host \  
--disk /path/to/fss-node01.qcow2,format=qcow2,bus=virtio \  
--disk /path/to/fss-node01-storage.qcow2,format=qcow2,bus=virtio \  
--network bridge=br0,model=virtio \  
--network bridge=br1,model=virtio \  
--os-type=linux \  
--os-variant=centos7.0 \  
--noautoconsole
```

- Step 6.** From the KVM console, log in to the node VM.

Use the following credentials:

Username: root

Password: N0ki@FSSb4se!

- Step 7.** If your environment does not support or use the cloud-init services, disable and stop these services.

Example

```
# systemctl stop cloud-init cloud-init-local cloud-config cloud-final  
# systemctl disable cloud-init cloud-init-local cloud-config cloud-final
```

- Step 8.** Resize the root partition.

Example

The base OS image comes with a smaller partition than needed. Use the following commands to resize the root partition. Verify that you are using the right disk (in this example `/dev/sda`).

```
# fdisk /dev/sda  
Welcome to fdisk (util-linux 2.23.2).
```

```
Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.
```

```
Command (m for help): p
```

```
Disk /dev/sda: 214.7 GB, 214748364800 bytes, 419430400 sectors
Units = sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disk label type: dos
Disk identifier: 0x000b050e
```

Device	Boot	Start	End	Blocks	Id	System
/dev/sda1	*	2048	16777215	8387584	83	Linux

Step 9. Delete the existing partition.

Example

```
Command (m for help): d
Selected partition 1
Partition 1 is deleted
```

Step 10. Create a new partition that fills the entire disk.

Example

Accept the default settings presented.

```
Command (m for help): n
Partition type:
  p primary (0 primary, 0 extended, 4 free)
  e extended
Select (default p):
Using default response p
Partition number (1-4, default 1):
First sector (2048-419430399, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-419430399, default 419430399):
Using default value 419430399
Partition 1 of type Linux and of size 200 GiB is set
```

Step 11. Set the boot flag for this partition.

Example

```
Command (m for help): a
Selected partition 1
```

Step 12. Write the partition table.

Example

```
Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.

WARNING: Re-reading the partition table failed with error 16: Device or resource busy.
```



```
The kernel still uses the old table. The new table will be used at
the next reboot or after you run partprobe(8) or kpartx(8)
Syncing disks.
```

Step 13. Optional: Configure a standard Linux partition on the storage disk, /dev/sdb in this example.

Example

```
# fdisk /dev/sdb
Welcome to fdisk (util-linux 2.23.2).

Changes will remain in memory only, until you decide to write them.
Be careful before using the write command.

Device does not contain a recognized partition table
Building a new DOS disklabel with disk identifier 0x8891800b.

Command (m for help): n
Partition type:
  p primary (0 primary, 0 extended, 4 free)
  e extended
Select (default p):
Using default response p
Partition number (1-4, default 1):
First sector (2048-629145599, default 2048):
Using default value 2048
Last sector, +sectors or +size{K,M,G} (2048-629145599, default 629145599):
Using default value 629145599
Partition 1 of type Linux and of size 300 GiB is set

Command (m for help): w
The partition table has been altered!

Calling ioctl() to re-read partition table.
Syncing disks.
```

Step 14. Reboot the node to register the change in partition sizes in the system.

Example

```
# reboot
```

Step 15. Grow the root partition file system to its full size.

Example

```
# xfs_growfs -d /
meta-data=/dev/sda1 isize=512 agcount=4, agsize=524224 blks
 = sectsz=512 attr=2, projid32bit=1
 = crc=1 finobt=0 spinodes=0
data = bsize=4096 blocks=2096896, imaxpct=25
 = sunit=0 swidth=0 blks
naming =version 2 bsize=4096 ascii-ci=0 ftype=1
log =internal bsize=4096 blocks=2560, version=2
 = sectsz=512 sunit=0 blks, lazy-count=1
realtime =none extsz=4096 blocks=0, rtextents=0
data blocks changed from 2096896 to 52428544
```

Step 16. Enable SSH.

The base image is a hardened image so SSH is disabled by default for the root user. To enable SSH, update the `/etc/ssh/sshd_config` file and change the following lines:

```
PasswordAuthentication no
PermitRootLogin no
```

to:

```
PasswordAuthentication yes
PermitRootLogin yes
```



Note: You can keep password authentication disabled to provide extra security. In this case, only key-based authentication works, and you must configure the appropriate public SSH keys for the root user so it can log in over SSH. In any case, this configuration is needed for the deployer VM to reach the nodes.

Step 17. Edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file to configure the correct static IP address, DNS servers, and gateway for the OAM network.

Example

The final content should look similar to the following, except with the IP address, DNS, and domain details specific to the target environment:

```
BOOTPROTO=static
DEVICE=eth0
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
IPADDR=192.0.2.10
PREFIX=24
GATEWAY=192.0.2.1
DNS1=192.0.2.5
DNS2=192.0.2.6
DOMAIN=fss.nokia.local
MTU=9000
```

Step 18. Edit the `/etc/sysconfig/network-scripts/ifcfg-eth1` file to configure the correct static IP address for the Fabric Management network.

Example

The final content should look similar to the following, except with the IP address, DNS, and domain details specific to the target environment:

```
BOOTPROTO=static
DEVICE=eth0
ONBOOT=yes
TYPE=Ethernet
USERCTL=no
IPADDR=192.0.2.10
PREFIX=24
MTU=9000
```

Ensure that you set the MTU parameter to 9000 for all the interfaces.

Step 19. Restart the network to apply the new configuration.

Example

```
# systemctl restart network
```

Step 20. Configure the appropriate NTP servers.

Edit the `/etc/chrony.conf` configuration file and replace all lines that begin with `server` with the correct server lines for the environment.

Step 21. Restart the `chronyd` service.

```
# systemctl restart chronyd
```

Step 22. Verify that time synchronization is functioning properly.

```
# chronyc tracking
```

Expected outcome

If the **Reference ID** field is not set to any of the configured servers, but instead refers to something like `127.127.1.1`, time synchronization is not functioning properly.

```
Reference ID      : 87E30FFE (192.0.2.5)
Stratum          : 4
Ref time (UTC)   : Wed Feb 16 01:20:36 2022
System time      : 0.000014215 seconds slow of NTP time
Last offset      : -0.000001614 seconds
RMS offset       : 0.000106133 seconds
Frequency        : 11.863 ppm slow
Residual freq    : -0.071 ppm
Skew             : 0.187 ppm
Root delay       : 0.063009784 seconds
Root dispersion  : 0.018440660 seconds
Update interval  : 64.5 seconds
Leap status      : Normal
```

Step 23. Change the hostname.

```
# hostnamectl set-hostname fss-node01.domain.tld
```

Step 24. Set up key-based authentication from the Fabric Services System deployer VM.

Example

If password authentication has been enabled on the node for SSH, enter the following command from the deployer VM.

```
# ssh-copy-id root@<node IP/FQDN>
```

5 Bare metal-based installation: Preparing the Fabric Services System nodes

For deployments that use bare metal servers for the Fabric Services System nodes, follow this chapter to configure these nodes.

You must install the CentOS 7.9 operating system on these nodes and prepare them so they can run the Kubernetes environment, the Fabric Services System application, and the Digital Sandbox.

5.1 Disk partitioning

Each Fabric Services System node can be configured as a storage node or as a non-storage node. In each environment, to ensure redundancy, at least two of the nodes must be configured as storage nodes.

Depending on the storage role of a node (non-storage node or storage node), the following requirements are in place for the disk partitions:

- Storage and non-storage nodes: Single root (/) partition of 200GB SSD storage. Do not create separate /opt, /var, or /usr partitions. A separate /boot partition is strongly suggested.
- Storage nodes only: Separate Linux type partition of 300GB SSD storage without a file system.



Tip: As best practice, do not include a swap partition as the nodes are running Kubernetes.

5.2 Installing the operating system, kernel, and packages

Procedure

Step 1. Download the CentOS 7.9 image.

The CentOS 7.9 image is available from http://isoredirect.centos.org/centos/7/isos/x86_64/. Download the CentOS-7-x86_64-DVD-2009.iso file.

The Fabric Services System nodes must run on a Linux kernel of 4.19.72-300.el7.x86_64.

Step 2. Install the CentOS 7.9 operating system on each Fabric Services System node.

During installation, select the **Infrastructure server** profile under **Software selection** (also referred to as “base” and “core”).

Step 3. Install additional packages.

- libcgroup
- psmisc
- policycoreutils-python
- rpcbind

Use the **yum install -y libcgrouppsmisc policycoreutils-python <package>** command. For example:

```
# yum install -y libcgrouppsmisc policycoreutils-python rpcbind
```

5.3 Configuring network settings

Prerequisites

All the Fabric Services System nodes must be part of the same subnet. The Fabric Services System nodes must be able to reach each other and the Fabric Services System deployer VM.

Procedure

Step 1. Configure network settings on all the nodes.

Configure the following network settings:

- Static IPv4 address
- DNS servers, with a maximum of two search domains
- Gateway IP address, if the deployer VM is not in the same subnet
- MTU on all interfaces set to 9000
- Firewalls disabled, for example, firewalld

Step 2. Verify the connectivity between the nodes.

Use the following command to test the MTU and connectivity:

```
# ping -M do -s 8970 <node-ip>
```

5.4 Setting up key-based authentication from the Fabric Services System deployer VM

Procedure

Step 1. From the deployer VM, enable passwordless login for each node in your deployment.

To enable passwordless login for the nodes, run the following command from the deployer VM:

```
$ ssh-copy-id root@<node IP/FQND>
```

Step 2. Verify that passwordless login has been enabled.

Log in to each node as root from the deployer VM. If passwordless login has been enabled successfully, the login attempt succeeds without prompting you for a password.

```
$ ssh <node IP/FQND>
```

6 Installing Fabric Services System

After the Fabric Services System deployer VM and the Fabric Services System nodes have been installed and configured, the environment is ready to be installed with the Fabric Services System application.

Using HTTPS for the UI and API

Nokia strongly recommends that you enable HTTPS for the UI and API to enforce the use of TLS encryption (v1.2 or v1.3) for all communication to the Fabric Services System management interfaces. Enabling HTTPS guarantees that all information is secured against snooping or changes during transit.

To enable HTTPS, a server certificate and private key must be provided to the deployer VM and installation tool. This certificate can either be provided by a trusted Certificate Authority within the environment, or it can be a self-signed certificate.

Ensure that the following properties of the certificate specify the FQDN that is used to access the Fabric Services System UI and API:

- CN
- SubjectAltName

If a self-signed certificate is preferred, the following command is an example of how to generate a self-signed certificate and accompanying private key. Replace the variables, (enclosed in brackets, < >) with the appropriate values for your deployment scenario. This command requires openssl 1.1.1 or later:

```
openssl req -newkey rsa:4096 -x509 -sha512 -days
<validity-period-days> -nodes -out fss-tls.crt -keyout fss-tls.key -subj
"/C=<country-code>/ST=<state-province>/L=<city>/O=<company>/OU=<organizational-unit>/
CN=<FSS-FQDN>"
-addext "subjectAltName=DNS:<FSS-FQDN>"
```

To enable HTTPS, the following settings must be configured in the installation configuration file. For related information, see [Editing the installation configuration file](#).

httpsenabled

Set this parameter to true (Boolean; do not enclose with quotes).

certificate

The full path of the certificate crt file as it is stored on the deployer VM.

privatekey

The full path of the private key file as it is stored on the deployer VM.

domainhost

The FQDN that is used to access the UI and API of the environment.



Note: When HTTPS is enabled, the Fabric Services System only allows connections over HTTPS that use the FQDN as the hostname in the request. Additionally, all HTTP requests are redirected toward HTTPS and to that specific FQDN. If a different hostname is used, the request fails.

6.1 Editing the installation configuration file

About this task

As part of the deployment, you must provide specific details about the configurable portions of the installation using a configuration file. The details you provide instruct the deployer how to proceed when setting up the Kubernetes deployment, the Fabric Services System software, and the Digital Sandbox.



Note: You can rename the input configuration file to a name specific to your deployment. In the examples that follow, the input configuration file is named `sample-input.json`.

Procedure

Step 1. From the deployer, access the input configuration

```
[root@fss-deployer ~] vi sample-input.json
```

Step 2. Edit the configuration file.

Update the file with the following settings:

- IP addresses of the nodes to be used in your Fabric Services System deployment
- deployer nodes
- worker nodes
- storage nodes



Note: When you set the **devices** parameter, specify only the partition name, ("`sdb1`" in the example below). You do not need to specify the path.

- time synchronization
- replica count
- Optional: Digital Sandbox installation characteristics

The deployer creates three Kubernetes master nodes. By default, the master nodes are the first three worker nodes specified in the configuration file. However, you can specify which worker nodes you want to act as the Kubernetes master nodes by tagging those nodes with the `master` role in the configuration file.

The following is an example of a `sample-input.json` configuration file. [Table 1: Configuration file fields](#) describes the fields you must configure.

```
{
  "deployernode": {
    "ipaddr": "192.0.2.10",
    "gateway": "192.0.2.1",
    "netmask": "255.255.255.0"
  },
  "digitalsandbox": {
    "enabled": true,
    "volumenode": "fss-node04"
  },
  "timesync": {
    "install": true,
    "server": "192.0.2.5"
  },
}
```

```
"fss": {
  "heketinode": "fss-node04",
  "dhcpnode": "fss-node01",
  "dhcpinterface": "198.51.100.11/24",
  "ztpaddress": "198.51.100.11",
  "httpsenabled": true,
  "certificate": "/root/certs/fss-tls.crt",
  "privatekey": "/root/certs/fss-tls.key",
  "domainhost": "fss.nokia.tld"
},
"workernodes": [
  {
    "hostip": "192.0.2.11",
    "hostname": "fss-node01",
    "role": "master"
  },
  {
    "hostip": "192.0.2.12",
    "hostname": "fss-node02",
    "role": "master"
  },
  {
    "hostip": "192.0.2.13",
    "hostname": "fss-node03",
    "role": "master"
  },
  {
    "hostip": "192.0.2.14",
    "hostname": "fss-node04",
    "role": "digitalsandbox"
  },
  {
    "hostip": "192.0.2.15",
    "hostname": "fss-node05",
    "role": "digitalsandbox"
  },
  {
    "hostip": "192.0.2.16",
    "hostname": "fss-node06",
    "role": "digitalsandbox"
  }
],
"replicacount": 3,
"storagenodes": [
  {
    "hostip": "192.0.2.14",
    "hostname": "fss-node04",
    "devices": [
      "sdb1"
    ]
  },
  {
    "hostip": "192.0.2.15",
    "hostname": "fss-node05",
    "devices": [
      "sdb1"
    ]
  },
  {
    "hostip": "192.0.2.16",
    "hostname": "fss-node06",
    "devices": [
      "sdb1"
    ]
  }
]
```



```

    }
  }
}

```

Table 1: Configuration file fields

Heading	Configurable values
<p>deployernode Specifies the IP address, gateway, and netmask configured on the network interface of the deployer VM. The deployer VM must be reachable by all of the Fabric Services System nodes, and the Fabric Services System nodes must be reachable by the deployer VM.</p>	<ul style="list-style-type: none"> • ipaddr: IP address of the deployer VM • gateway: gateway address of the interface on the deployer node • netmask: netmask of the interface on the deployer node
<p>timesync Specifies time synchronization parameters. The system requires all servers to be time-synchronized to function properly. If not completed already, the Fabric Services System deployer must be configured to install Chrony for time synchronization.</p>	<ul style="list-style-type: none"> • install: when set to <code>true</code>, Chrony is installed on all compute nodes with a default configuration • server: address of the NTPD server. This optional value configures an additional NTPD server as well as the default servers.
<p>digitalsandbox Specifies Digital Sandbox parameters.</p>	<ul style="list-style-type: none"> • enabled: when this flag is set to <code>true</code>, the Digital Sandbox component is installed. Ensure that at least one worker node is tagged with the <code>digitalsandbox</code> role. When set to <code>false</code>, the Digital Sandbox component is not installed. • volumenode: specifies one of the storage nodes on which the Digital Sandbox volume is created using Gluster. If not specified, the first node in the storage pool is used.
<p>fss Specifies Fabric Services System deployment options.</p>	<ul style="list-style-type: none"> • heketinode: specifies a node in the storage pool for the Heketi component. Heketi is used for provisioning storage requirements. This value specifies where to run the Heketi service. If no value is provided, the first node in the storage pool is used. • ztpaddress: specifies an address associated with the node running Traefik. The node can be any of the Fabric Services System cluster nodes. The SR Linux nodes connect to this IP address during the BOOT process to get the software image and the configuration. This IP address must be reachable from the SR Linux management network. • dhcpcnode: specifies a node on which the Fabric Services System DHCP pod is scheduled

Heading	Configurable values
	<ul style="list-style-type: none"> • dhcpiinterface: specifies the address that the DHCP server listens to for any DHCP requests coming from the DHCP relay agent. Optionally, you can connect SR Linux nodes via the relay agent to reach the Fabric Services System if they are not on the management network. • httpsenabled: Specifies whether HTTPS is enabled or disabled on the API and UI. This field is mandatory. Possible values: <ul style="list-style-type: none"> – <code>false</code>: disables HTTPS on the API and UI – <code>true</code>: enables HTTPS on the API and UI • certificate: contains the path on the deployer VM where the installer can find the crt (certificate) file. This certificate can be provided by a trusted Certificate Authority within your environment, or can be self-signed. The settings for its CN and SubjectAltName fields must be set to the domainhost value. • privatekey: contains the path on the deployer VM where the installer can find the private key file associated with the certificate. • domainhost: contains the full FQDN on which the Fabric Services System API and UI is served.
<p>workernodes Specifies the list of nodes intended to be part of the deployment, except for the deployer host. Worker nodes include storage nodes and Digital Sandbox nodes.</p>	<ul style="list-style-type: none"> • hostip: IP address of the specific worker node • hostname: hostname of the worker node • role: specified role of the worker node For Digital Sandbox nodes, specify this value as <code>digitalsandbox</code>. For Kubernetes master nodes, specify this value as <code>master</code>.
<p>replicacount Specifies the replica count for Gluster volumes, including the active volume.</p>	<p>The default value is 1, indicating no replica (active volume only). A replica count higher than 1 creates the respective number of replica storage volumes. The value cannot be greater than the number of storage nodes.</p>
<p>storagenodes Specifies the list of nodes used to create a storage pool. The number of storage nodes must match the value of <code>replicacount</code>, if configured. Nokia recommends that you configure a minimum of three storage nodes</p>	<ul style="list-style-type: none"> • hostip: IP address of the specific storage node • hostname: hostname of the storage node • devices: separate block devices must be set up. Set up a raw partition as <code>xxx</code>. If an existing file system is present on the device, the setup cannot proceed.

What to do next

After you finish editing the input configuration file, you can install the Fabric Services System environment.

6.2 Installing the Fabric Services System environment

Procedure

Step 1. Initiate the setup.

Example

```
[root@fss-deployer ~]$ /root/bin/fss-install.sh configure sample-input.json
```

The CLI prompt indicates when the configuration is complete.

Step 2. Start the installation of Kubernetes, the Fabric Services System software, and the Digital Sandbox.

Example

```
[root@fss-deployer ~]$ /root/bin/fss-install.sh
```

The installation time varies depending on the capacity of your system.

Step 3. After the installation script is completed, verify the installation by logging in to the Fabric Services System user interface.

If HTTPS is enabled, use `https://<domainhost_in_the_sample-json.input>`;
otherwise, use one of the node the IP addresses.

Log in using the following default username and password:

Username: admin

Password: NokiaFss1!



Note: After the initial login, Nokia recommends that you change this default admin password to a stronger password to secure the platform properly.

6.3 Troubleshooting a failed installation

If the Fabric Services System installation fails for any reason, you can use a script that is bundled with the system to generate information about the installation status. For assistance with troubleshooting, contact your Nokia support team.

The technical support script is included with the Fabric Services System.

For more information about the script and how to run it, see "Capturing troubleshooting data" in the *Fabric Services System User Guide*

7 Reinstalling software on an existing Kubernetes cluster

The Fabric Services System Deployer provides a fast method to reinstall the system software without reinstalling the Kubernetes deployment. Use this re-installation procedure only if the Kubernetes deployment is functional and the hardware topology has not changed after the initial installation.



Note: The re-installation of the Fabric Services System application can only be done on a Kubernetes cluster that was previously installed using the same Fabric Services System deployer.

When you execute this procedure, all Fabric Services System software and data is removed except for the Kubernetes cluster. This procedure installs the same version of the Fabric Services System software with the same images used in the previous installation.

Complete the following tasks:

1. [Uninstalling Fabric Services System software only](#)
Removes all Fabric Services System software and intent data while the Kubernetes cluster remains running.
2. [Reinstalling Fabric Services System software only](#)
Reinstalls the Fabric Services System software on the existing Kubernetes cluster.

7.1 Uninstalling Fabric Services System software only

About this task

This procedure removes all the Fabric Services System software and intent data in the cluster. Only the Kubernetes cluster remains running.

Procedure

- Step 1.** Set the `SKIP_K8S` environment variable.
This ensures that the Kubernetes cluster is not uninstalled with the software. Log in to the deployer VM and run the following command:

```
[root@fss-deployer ~]$ export SKIP_K8S=true
```

- Step 2.** Uninstall Fabric Services System software and the related data.

```
[root@fss-deployer ~]$ fss-uninstall.sh
```

- Step 3.** When the uninstall process is complete, unset the `SKIP_K8S` environment variable.

```
[root@fss-deployer ~]$ unset SKIP_K8S
```

What to do next

After the software has been uninstalled, you can reinstall the same version of Fabric Services System software without reinstalling the Kubernetes deployment. See [Reinstalling Fabric Services System software only](#).

7.2 Reinstalling Fabric Services System software only

Prerequisites

This procedure reinstalls Fabric Services System software on the existing Kubernetes cluster. You must first uninstall the Fabric Services System software; see [Uninstalling Fabric Services System software only](#). To reinstall the Fabric Services System software:

Procedure

Step 1. Ensure the `SKIP_K8S` environment variable is set to `true`.

```
[root@fss-deployer ~]$ SKIP_K8S=true
```

Step 2. Reinstall the Fabric Services System software.

```
[root@fss-deployer ~]$ fss-install.sh
```

Step 3. When the re-installation is complete, unset the `SKIP_K8S` environment variable.

```
[root@fss-deployer ~]$ unset SKIP_K8S
```

8 Uninstalling software

This section describes how to uninstall a Fabric Services System deployment. Topics include:

- [Uninstalling a Fabric Services System deployment](#)
Uninstalls a previous Fabric Services System installation.
- [Deleting the deployer VM](#)
Deletes the deployer VM from the node on which it is hosted.

8.1 Uninstalling a Fabric Services System deployment

About this task

After completing the initial installation, you can uninstall a Fabric Services System deployment.

Procedure

Step 1. Uninstall Fabric Services System services, Kubernetes, Digital Sandbox, and storage volumes.

Example

From the deployer VM, run the following command:

```
[root@fss-deployer ~]$ /root/bin/fss-uninstall.sh all
```

Step 2. Reboot the storage nodes.

As part of the Fabric Services System uninstall procedure, the file system cleanup requires the storage nodes to be rebooted so the partition or disk can be used for Fabric Services System installation or other purposes.

8.2 Deleting the deployer VM

Prerequisites

To completely remove a Fabric Services System deployment, after uninstalling the Fabric Services System software, Digital Sandbox, and Kubernetes components (see [Uninstalling a Fabric Services System deployment](#)), you must delete the deployer VM from the node on which it is hosted.

To delete the deployer VM:

Procedure

Step 1. Log in to the deployer host.

Step 2. Create an `fssvm_delete.sh` file, then copy the following contents into the file:

```
#!/bin/bash  
delete_fssvm() {
```

```
D=/var/lib/libvirt/images
VM=$1

sudo virsh shutdown $VM
sudo virsh undefine $VM
sudo virsh pool-destroy $VM
sudo rm -ri $D/$VM
}
```

Step 3. Modify the permissions of the shell script file.

```
chmod 755 fssvm_delete.sh
```

Step 4. Execute the shell script.

```
./fssvm_delete.sh <name of the VM>
```

Expected outcome

The deployer VM is deleted from the deployer node.

Customer document and product support



Customer documentation

[Customer documentation welcome page](#)



Technical support

[Product support portal](#)



Documentation feedback

[Customer documentation feedback](#)