



Fabric Services System

Release 23.4.1

Software Installation Guide

3HE 19455 AAAA TQZZA
Edition 1.0
June 2023

© 2023 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.

© 2023 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.....	6
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.....	9
2.1.4.2	Fabric Services System node requirements.....	9
2.1.5	Installation process overview.....	10
3	The Fabric Services System deployer VM.....	11
3.1	Downloading the Fabric Services System deployer image.....	11
3.2	Prepare the Fabric Services System deployer hypervisor.....	11
3.3	Fabric Services System deployer VM creation.....	11
3.3.1	Creating the VM on a bridged network on KVM.....	11
3.3.2	Creating the VM on VMware vSphere.....	12
3.4	Configuring the Fabric Services System deployer VM.....	14
4	Preparing the Fabric Services System virtual machine nodes.....	17
4.1	Downloading the Fabric Services System base OS image.....	17
4.2	Networking considerations.....	17
4.3	Create the Fabric Services System virtual machine.....	17
4.3.1	Creating the VM on bridged networks on KVM.....	17
4.3.2	Creating the VM on bridged networks on VMware vSphere.....	18
4.4	Configuring the Fabric Services System virtual machine.....	20
5	Fabric Services System installation.....	26
5.1	Editing the installation configuration file.....	27

5.2	Installing the Fabric Services System environment.....	34
5.3	How to troubleshoot a failed installation.....	34
6	Software reinstallation on an existing Kubernetes cluster.....	36
6.1	Uninstalling Fabric Services System software only.....	36
6.2	Reinstalling Fabric Services System software only.....	36
7	Uninstalling software.....	38
7.1	Uninstalling a Fabric Services System deployment.....	38
7.2	Deleting the deployer VM.....	38
8	Software upgrade and rollback.....	40
8.1	Preparing for software upgrade.....	40
8.2	Performing a software upgrade.....	42
8.3	Upgrading the Fabric Services System cluster.....	50
8.4	Performing a software rollback.....	50

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 install 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 section lists the changes that were made in this release.

Table 1: What's new in Fabric Services System 23.4.1

Description	Location
Support for remote syslog for user audit logs	Editing the installation configuration file
External access to Fabric Services System alarms	Editing the installation configuration file
IPv6 support for northbound services	Editing the installation configuration file Fabric Services System installation Configuring the Fabric Services System virtual machine
Manage LDAP configuration through CLI script	Editing the installation configuration file
Upgradability	Software upgrade and rollback Preparing for software upgrade Performing a software upgrade Upgrading the Fabric Services System cluster
Other updates	Uninstalling software

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 simple command sequence (often selected from a user interface). Example: **start > connect to**
- Angle brackets (< >) indicate an item that is not used verbatim. For example, for the command **show ethernet <name>**, **name** should be replaced with the name of the interface.
- A vertical bar (|) indicates a mutually exclusive argument.
- 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 appropriate 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 one, three, or six Fabric Services System nodes. The Fabric Services System is deployed as an application on virtual machine servers. In a deployment with six nodes, the Fabric Services System can be deployed with the Digital Sandbox.

Related topics

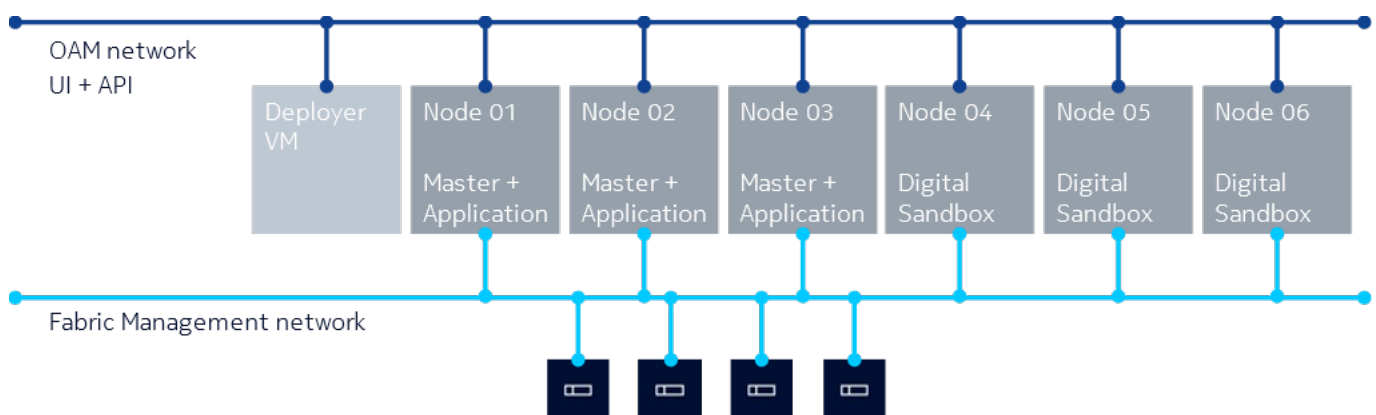
[Fabric Services System node requirements](#)

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
- as the 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:

- 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 Fabric Services System microservices using Helm charts and container repositories hosted in the deployer
- provides technical support tools and backup and restore 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 backup and restore tools and 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 VMware vSphere-based 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:

- an environment with one node, which hosts only the Fabric Services System application for very small scale deployments and does not support the use of Digital Sandbox
- an environment with three nodes, which hosts only the Fabric Services System application and does not support the use of Digital Sandbox
- an environment with a minimum of six nodes, which hosts both the Fabric Services System application and supports 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 used to install Kubernetes and the Fabric Services System application. It is delivered as one of the following images:

- QCOW2: the image that provides the deployer VM and which includes the operating system and the deployer software for the KVM hypervisor
- OVA: the VM image for the VMware vSphere 7.0 hypervisor

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

- CPU: 1 virtual CPU (vCPU) on a VMware vSphere or KVM hypervisor with a modern x86-64 CPU that supports virtualization
Recommended: 2 vCPU
- memory: 8 GB RAM
- storage: 50 GB of storage (used by the deployer QCOW2 image)
- networking:
 - 1 GbE network interface card (NIC)
 - the configured DNS servers must be reachable, functional, and able to resolve the hostnames used for the Fabric Services System nodes

2.1.4.2 Fabric Services System node requirements

The Fabric Services System nodes are deployed as virtual machine servers. The following are the requirements of the Fabric Services System nodes in KVM and VMware hypervisor:

- CPU: 32 vCPUs on a modern x86-64 CPU
- memory: 64 GB
- storage: at least 500 GB of available SSD-based storage
- networking:
 - at least one 10 Gbps NIC
 - the configured DNS servers must be reachable, functional, and able to resolve the hostnames used for the FSS nodes

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

- operating system: VMware vSphere 7.0 or RHEL/Rocky 8
- hypervisor: ESXi 7.0 or KVM
- resource reservation for CPU, memory, and disks must be set to 100% for the Fabric Services System node virtual machines

Related topics

[Supported deployment models](#)

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. [Fabric Services System deployer VM creation](#)

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

3. [Preparing the Fabric Services System virtual machine nodes](#)

This task describes how to prepare and configure the nodes for deployment.

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

This task describes how to modify the variables and enter the expected data in the JSON-based configuration file, which is used to specify the configuration and deployment of the Fabric Services System environment.

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

This task describes the deployment of the Fabric Services System environment using the `sample-input.json` file.

3 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 QCOW2 or OVA image.

Table 2: Deployer VM images

Deployment	Where to download the image
VMware vSphere	Download the OVA image to a host that can reach the VMware v Center or ESXi host on which it will be deployed.
KVM	Download the QCOW2 image to the deployer host.

3.2 Prepare the Fabric Services System deployer hypervisor

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 or EFI. You must also install a VMware vSphere or KVM environment on the node before installing the deployer VM.

3.3 Fabric Services System deployer VM creation

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

Use one of the following procedures to configure networking for the deployer VM using a bridged network:

- [Creating the VM on a bridged network on KVM](#)
- [Creating the VM on VMware vSphere](#)

3.3.1 Creating the VM on a bridged network on KVM

About this task

This section provides an example script used to create a VM 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-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
```

3.3.2 Creating the VM on VMware vSphere

About this task

You can use one of the following methods to deploy the VM on VMware vSphere:

- the VMware vSphere vCenter or ESXi UI
For instructions, see “Deploy an OVF or OVA Template” in the VMware vSphere documentation.
- the VMware Open Virtualization Format Tool CLI
The following section provides an example of how to use the VMware OVF Tool CLI.

Procedure

- Step 1.** Download and install the latest version of the VMware OVF Tool from the VMware Developer website.
- Step 2.** Display details about the OVA image.

Example

Execute the **ovftool** command with just the OVA image name as the argument.

```
$ ovftool fss-deployer_v23.4.1-18.ova
OVF version: 1.0
VirtualApp: false
Name: fss-deployer

Download Size: 9.47 GB

Deployment Sizes:
  Flat disks: 40.00 GB
  Sparse disks: 12.57 GB

Networks:
  Name: OAM
  Description: The Fabric Services System OAM (UI and API) network

  Name: FABRIC
  Description: The Fabric Services System Fabric Management network

Virtual Machines:
  Name: fss-deployer
  Operating System: centos7_64guest
  Virtual Hardware:
    Families: vmx-14
    Number of CPUs: 2
    Cores per socket: 1
    Memory: 7.91 GB

  Disks:
    Index: 0
    Instance ID: 4
    Capacity: 40.00 GB
    Disk Types: SCSI-lsilogic

  NICs:
    Adapter Type: VmxNet3
    Connection: OAM

    Adapter Type: VmxNet3
    Connection: FABRIC

References:
  File: fss-deployer-disk1.vmdk
```

- Step 3.** Deploy the OVA image using the OVF Tool.
For details about command line arguments, see the OVF Tool documentation from the VMware website.



Note: Ensure that you use thick provisioning for the disk and to connect all the interfaces to a network. The secondary interface can be disconnected and disabled after the deployment and before you power on.

Example

```
$ ovftool --acceptAllEulas -dm=thick -ds=VSAN -n=fss-deployer --net:"OAM=OAM-
network" --net:"FABRIC=Fabric-network" fss-deployer_v23.4.1-18.ova vi://administrator
%40vsphere.local@vcenter.domain.tld/My-Datacenter/host/My-Cluster/Resources/My-
Resource-Group

Opening OVA source: fss-deployer_v23.4.1-18.ova
The manifest validates
Enter login information for target vi://vcenter.domain.tld/
Username: administrator%40vsphere.local
Password: *****
Opening VI target: vi://administrator%40vsphere.local@vcenter.domain.tld/My-
Datacenter/host/My-Cluster/Resources/My-Resource-Group
Deploying to VI: vi://administrator%40vsphere.local@vcenter.domain.tld/My-Datacenter/
host/My-Cluster/Resources/My-Resource-Group
Transfer Completed
```

3.4 Configuring the Fabric Services System deployer VM

Procedure

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

Use the following credentials:

Username: root

Password: N0ki@FSSb4se!



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. 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 to log in over SSH. In any case, this configuration is needed for the deployer VM to reach the nodes.

Step 4. Restart SSH.

```
# systemctl restart sshd
```

Step 5. 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
MTU=9000
```

Step 6. Restart the network to apply the new configuration.

Execute the following command:

```
# systemctl restart NetworkManager.service
```

Step 7. 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 8. Restart the `chronyd` service.

```
# systemctl restart chronyd
```

Step 9. Verify that time synchronization is functioning properly.

```
# chronyc tracking
```

Example

```
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
```

Expected outcome

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

Step 10. Synchronize the RTC clock and the system clock.

Example

Ensure that the RTC and the system clock are synchronized after every reboot.

```
# hwclock --systohc
```

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

```
# timedatectl
```

Step 11. Optional: Change the hostname.

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

Step 12. Reboot the Fabric Services System deployer VM to ensure that all services come up with the correct network configuration.

```
# reboot
```


4 Preparing the Fabric Services System virtual machine nodes

The procedures in this section describe how to create and configure Fabric Services System nodes in deployments that use virtual machine servers.

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 Rocky 8.6 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 OVA or QCOW2 image.

4.2 Networking considerations

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

Within the hypervisor, both networks should be available as bridged networks. Both these networks require support for jumbo frames (MTU set to 9000).

Ensure that the MTU is set to 9000 on all the interfaces on the hypervisor, Fabric Service System VM nodes, deployer, and the interconnecting devices.

Related topics

[Networking for the Fabric Services System nodes](#)

4.3 Create the Fabric Services System virtual machine

To deploy a Fabric Services System node as a virtual machine, complete the appropriate procedure for your deployment scenario:

- [Creating the VM on bridged networks on KVM](#)
- [Creating the VM on bridged networks on VMware vSphere](#)

4.3.1 Creating the VM on bridged networks on KVM

About this task

Complete the following steps to deploy a Fabric Services System node as a virtual machine on KVM. The OAM network is referred to as br0 and the fabric management network is referred to as br1.

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 appropriate location on the hypervisor 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 reduce 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-variant=centos7.0 \  
--noautoconsole
```

What to do next

[Configuring the Fabric Services System virtual machine](#)

4.3.2 Creating the VM on bridged networks on VMware vSphere

About this task

You can use one of the following methods to deploy the VM on VMware vSphere:

- the VMware vSphere vCenter or ESXi UI
For instructions, see *Deploy an OVF or OVA Template* in the VMware vSphere documentation.
- the VMware Open Virtualization Format Tool CLI
The following section provides an example of how to use the VMware OVF Tool CLI.

Procedure

Step 1. Download and install the latest version of the VMware OVF Tool from the VMware Developer website.

Step 2. Display details about the OVA image.

Example

Execute the **ovftool** command with just the OVA image name as the argument.

```
$ ovftool fss-baseos_v23.4.1-18.ova
OVF version: 1.0
VirtualApp: false
Name: fss-baseos

Download Size: 2.03 GB

Deployment Sizes:
  Flat disks: 128.00 GB
  Sparse disks: 3.64 GB

Networks:
  Name: OAM
  Description: The Fabric Services System OAM (UI and API) network

  Name: FABRIC
  Description: The Fabric Services System Fabric Management network

Virtual Machines:
  Name: fss-baseos
  Operating System: centos7_64guest
  Virtual Hardware:
    Families: vmx-14
    Number of CPUs: 16
    Cores per socket: 1
    Memory: 32.00 GB

  Disks:
    Index: 0
    Instance ID: 4
    Capacity: 128.00 GB
    Disk Types: SCSI-lsilogic

  NICs:
    Adapter Type: VmxNet3
    Connection: OAM

    Adapter Type: VmxNet3
    Connection: FABRIC

References:
```

File: fss-baseos-disk1.vmdk

Step 3. Deploy the OVA image using the OVF Tool.

For details about command line arguments, see the OVF Tool documentation from the VMware website.



Note: Ensure that you use thick provisioning for the disk and to connect all the interfaces to a network. The secondary interface can be disconnected and disabled after the deployment and before you power on.

Example

```
$ ovftool --acceptAllEulas -dm=thick -ds=VSAN -n=fss-node01 --net:"OAM=OAM-network" --net:"FABRIC=Fabric-network" fss-baseos_v23.4.1-18.ova vi://administrator%40vsphere.local@vcenter.domain.tld/My-Datacenter/host/My-Cluster/Resources/My-Resource-Group

Opening OVA source: fss-base_v23.4.1-18.ova
The manifest validates
Enter login information for target vi://vcenter.domain.tld/
Username: administrator%40vsphere.local
Password: *****
Opening VI target: vi://administrator%40vsphere.local@vcenter.domain.tld/My-Datacenter/host/My-Cluster/Resources/My-Resource-Group
Deploying to VI: vi://administrator%40vsphere.local@vcenter.domain.tld/My-Datacenter/host/My-Cluster/Resources/My-Resource-Group
Transfer Completed
Completed successfully
```

Step 4. Optional: If you are using this node as a storage node, create the necessary extra disk for the storage cluster to be formed.

To create the extra disk, edit the VM in the VMware vCenter and add a new 300 GB disk.

Step 5. Enable 100% resource reservation for both CPU and memory for the VM.

You can configure the resource reservation for CPU and memory by editing the VM in vCenter.

What to do next

[Configuring the Fabric Services System virtual machine](#)

4.4 Configuring the Fabric Services System virtual machine

About this task

Complete the following steps to configure a Fabric Services System node.

Procedure

Step 1. From the VMware vSphere or KVM console, log in to the node VM.

Use the following credentials:

Username: root

Password: N0ki@FSSb4se!

Step 2. 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 3. 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 to log in over SSH. In any case, this configuration is needed for the deployer VM to reach the nodes.

Step 4. Restart SSH.

```
# systemctl restart sshd
```

Step 5. Edit the `/etc/sysconfig/network-scripts/ifcfg-eth0` file to configure the correct static IP address, DNS servers, and gateway for the OAM network.

Note: If you are deploying a dual-stack system, provide the IPv6 details in the `ifcfg-eth0` file. Additionally, ensure that the default gateway is configured for both IPv4 and IPv6 and that both gateways are functional before installing the Fabric Services System.

Example

The final content should look similar to the following example, 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 6. Edit the `/etc/sysconfig/network-scripts/ifcfg-eth1` file to configure the correct static IP address for the fabric management network.

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

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
```

Step 7. Restart the network to apply the new configuration.

Example

```
# systemctl restart NetworkManager.service
```

Step 8. Resize the root partition.

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 correct disk.

Example

This example uses `/dev/sda` for VMware vSphere; in KVM, `/dev/vda` is used instead.

```
# 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):

Last sector, +sectors or +size{K,M,G} (2048-419430399, default 419430399):

Created a new partition 1 of type 'Linux' and of size 200 GiB.
Partition #1 contains a xfs signature.

Do you want to remove the signature? [Y]es/[N]o: Y

The signature will be removed by a write command.

```

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.

Example

This example uses `/dev/sda` for VMware vSphere; in KVM, `/dev/vda` is used instead.

```

# 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. 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 17. Restart the `chronyd` service.

```
# systemctl restart chronyd
```

Step 18. 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 19. Synchronize the RTC clock and the system clock.

Ensure that the RTC and the system clock are synchronized after every reboot.

Example

```
# hwclock --systohc
```

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

```
# timedatectl
```

Step 20. Change the hostname.

Example

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

Step 21. Set up key-based authentication from the deployer VM to Fabric Services System compute VMs.

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 Fabric Services System installation

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

The certificate provided during the installation must also be able to sign certificate requests itself; the CA option must be set to `true`. This is also referred to as a SubCA or Intermediate CA certificate.

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:

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 to use HTTPS and to use that specific FQDN. If a different hostname is used, the request fails.

Dual-stack networks

The system supports a dual-stack network, where each VM has an IPv4 and an IPv6 address; the system does not support a pure IPv6 network.



Note:

- The network must be ready for IPv6 and IPv4 IP addresses.
- Each node must be configured with an IPv4 and IPv6 default gateway configured and the gateways must be functional.
- The pods running in the VMs need to connect to SR Linux, which is in a different network.
- After creating the VMs, Nokia recommends that you verify the required connectivity over IPv4 and IPv6.

To enable support for a dual-stack network, set the **enable_dual_stack_networks** parameter to `True` in the `sample-input.json` file. You must also set an IP v6 address in the **ipv6** parameter for each worker node and storage node.

Related topics

[Editing the installation configuration file](#)

5.1 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 file.

```
[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 node settings
- worker node settings
- storage node settings



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


- time synchronization


- replica count
- optional: Digital Sandbox installation characteristics
- optional: remote syslog settings


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 table below describes the fields in the `sample-input.json` file. Examples of `sample-input.json` files for IPv4 and dual-stack deployments follow.

Table 3: Field definitions

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: the IP address of the deployer VM. • gateway: the gateway address of the interface on the deployer node. • netmask: the netmask of the interface on the deployer node.
<p>rsyslog Specifies the remote syslog server settings.</p>	<ul style="list-style-type: none"> • host: the IP address or FQDN of the remote syslog server. • port: the port that the rsyslog utility uses for network connectivity. • proto: the protocol used for syslog traffic, either TCP or UDP. <p> Note: The system currently supports one remote syslog server.</p>
<p>digitalsandbox Specifies Digital Sandbox parameters.</p>	<p>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.</p>
<p>fss Specifies Fabric Services System deployment options.</p>	<ul style="list-style-type: none"> • 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. • ztpv6address: specifies the IPv6 address associated with the node for SR Linux to connect using IPv6. • dhcpcnode: specifies a node on which the Fabric Services System DHCP pod is scheduled.

Heading	Configurable values
	<ul style="list-style-type: none"> • dhcpinterface: 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. • dhcpv6interface: specifies the IPv6 address of the DHCPv6 relay agent. • 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>true</code>: enables HTTPS on the API and UI – <code>false</code>: disables HTTPS on the API and UI • certificate: contains the path on the deployer VM where the installer can find the certificate <code>.crt</code> 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. • truststoreFilename: specifies the location of the truststore filename with the absolute path information. The JKS file must be generated to access the LDAP server from the Fabric Services System instance. The alternate names in the certificate should match the name and IP address configured for the federation provider (using the Fabric Services System UI or REST API). • truststorePassword: specifies the password used to access the truststore • kafkaconfig Configures the parameters that enable third-party tools to access Fabric Services System alarms. <p> Note: HTTPS must be enabled on the Fabric Services System.</p> <ul style="list-style-type: none"> – port: the port number used by the client to connect to the Kafka service; specify a value between 30000 and 32767. – groupprefix: the user group prefix for the client to use to connect to Kafka service. – user and password: the credentials to use to authenticate.

Heading	Configurable values
	<ul style="list-style-type: none"> – maxConnections: the maximum number of clients that can connect to the Kafka service. The maximum allowed value is 10.
K8s Specifies whether dual-stack network is supported	enable_dual_stack_networks : set to <code>True</code> to enable support for IPv6 networks.  Note: The system supports only a dual-stack network, where each VM has an IPv4 and IPv6 address; the system does not support a pure IPv6 network.
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.	<ul style="list-style-type: none"> • hostip: the IP address of the specific worker node . • ip6: the IPv6 address of the worker node; required if the enable_dual_stack_networks parameter is set to <code>True</code>. • hostname: the hostname of the worker node. • role: the 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>.
replicacount Specifies the replica count for Gluster volumes, including the active volume.	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.
storagenodes Specifies the list of nodes used to create a storage pool. The number of storage nodes must match the value of replicacount , if configured. Nokia recommends that you configure a minimum of three storage nodes.	<ul style="list-style-type: none"> • hostip: the IP address of the specific storage node. • ip6: the IPv6 address of the specific storage node; required if the enable_dual_stack_networks parameter is set to <code>True</code>. • hostname: the hostname of the storage node. • devices: separate block devices must be configured. Configure a raw partition as <code>xxx</code>. If an existing file system is present on the device, the setup cannot proceed.
singlenode Specifies whether the deployment consists of only a single node for extra small deployments.	The default value is <code>false</code> , indicating that the deployment is a standard three- or six-node deployment. If set to <code>true</code> , the deployment is set up on a single node and has no redundancy built in.

Example

IPv4

The following is an example of a `sample-input.json` configuration file for an IPv4 network.

```
{
  "deployernode": {
    "ipaddr": "192.0.2.200",
    "gateway": "192.0.2.1",
```

```
"netmask": "255.255.254.0"
},
"digitalsandbox": {
  "enabled": true,
  "volumenode": "fss-node04"
},
"fss": {
  "heketinode": "fss-node04",
  "dhcpnode": "fss-node01",
  "dhcpinterface": "128.66.0.201/24",
  "ztpaddress": "128.66.0.201",
  "httpsenabled": true,
  "certificate": "/root/certs/fss-tls.crt",
  "privatekey": "/root/certs/fss-tls.key",
  "domainhost": "fss-dev.phd.fss.dev",
  "kafkaconfig": {
    "port": "31000",
    "groupprefix": "fsskafka",
    "user": "fssalarms",
    "password": "fssalarms",
    "maxConnections": 2
  }
},
"rsyslog": {
  "host": "192.0.2.161",
  "port": 51400,
  "proto": "udp"
},
"k8s": {
  "enable_dual_stack_networks": false
},
"replicacount": 2,
"workernodes": [
  {
    "hostip": "192.0.2.201",
    "hostname": "fss-node01",
    "role": "master"
  },
  {
    "hostip": "192.0.2.202",
    "hostname": "fss-node02",
    "role": "master"
  },
  {
    "hostip": "192.0.2.203",
    "hostname": "fss-node03",
    "role": "master"
  },
  {
    "hostip": "192.0.2.204",
    "hostname": "fss-node04",
    "role": "digitalsandbox"
  },
  {
    "hostip": "192.0.2.205",
    "hostname": "fss-node05",
    "role": "digitalsandbox"
  },
  {
    "hostip": "192.0.2.206",
    "hostname": "fss-node06",
    "role": "digitalsandbox"
  }
],
```

```

"storagenodes": [
  {
    "hostip": "192.0.2.204",
    "hostname": "fss-node04",
    "devices": [
      "sdb1"
    ]
  },
  {
    "hostip": "192.0.2.205",
    "hostname": "fss-node05",
    "devices": [
      "sdb1"
    ]
  },
  {
    "hostip": "192.0.2.206",
    "hostname": "fss-node06",
    "devices": [
      "sdb1"
    ]
  }
]
}

```

Example

Dual stack

The following is an example of a `sample-input.json` configuration file for dual stack IPv4 and IPv6 network.

```

{
  "deployernode": {
    "ipaddr": "192.0.2.200",
    "gateway": "192.0.2.1",
    "netmask": "255.255.254.0"
  },
  "digitalsandbox": {
    "enabled": true,
    "volumenode": "fss-node04"
  },
  "fss": {
    "heketinode": "fss-node04",
    "dhcpnode": "fss-node01",
    "dhcpinterface": "128.66.0.201/24",
    "ztpaddress": "128.66.0.201",
    "httpsenabled": true,
    "certificate": "/root/certs/fss-tls.crt",
    "privatekey": "/root/certs/fss-tls.key",
    "domainhost": "fss-dev.phd.fss.dev",
    "kafkaconfig": {
      "port": "31000",
      "groupprefix": "fsskafka",
      "user": "fssalarms",
      "password": "fssalarms",
      "maxConnections": 2
    },
    "truststoreFilename": "/root/fss.truststore.jks",
    "truststorePassword": "fss123"
  },
  "k8s": {
    "enable_dual_stack_networks": true
  },
  "replicacount": 2,

```



```
"rsyslog": {
  "host": "192.0.2.161",
  "port": 514,
  "proto": "udp"
},
"workernodes": [
  {
    "hostip": "192.0.2.201",
    "ip6": "2001:db8:f685:0::201",
    "hostname": "fss-node01",
    "role": "master"
  },
  {
    "hostip": "192.0.2.202",
    "ip6": "2001:db8:f685:0::202",
    "hostname": "fss-node02",
    "role": "master"
  },
  {
    "hostip": "192.0.2.203",
    "ip6": "2001:db8:f685:0::203",
    "hostname": "fss-node03",
    "role": "master"
  },
  {
    "hostip": "192.0.2.204",
    "ip6": "2001:db8:f685:0::204",
    "hostname": "fss-node04",
    "role": "digitalsandbox"
  },
  {
    "hostip": "192.0.2.205",
    "ip6": "2001:db8:f685:0::205",
    "hostname": "fss-node05",
    "role": "digitalsandbox"
  },
  {
    "hostip": "192.0.2.206",
    "ip6": "2001:db8:f685:0::206",
    "hostname": "fss-node06",
    "role": "digitalsandbox"
  }
],
"storagenodes": [
  {
    "hostip": "192.0.2.204",
    "ip6": "2001:db8:f685:0::204",
    "hostname": "fss-node04",
    "devices": [
      "sdb1"
    ]
  },
  {
    "hostip": "192.0.2.205",
    "ip6": "2001:db8:f685:0::205",
    "hostname": "fss-node05",
    "devices": [
      "sdb1"
    ]
  },
  {
    "hostip": "192.0.2.206",
    "ip6": "2001:db8:f685:0::206",
    "hostname": "fss-node06",
```

```
"devices": [  
  "sdb1"  
]  
}  
]  
}
```

What to do next

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

5.2 Installing the Fabric Services System environment

Procedure

Step 1. Initiate the setup.

Example

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

5.3 How to troubleshoot 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*.

6 Software reinstallation 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

6.1 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

To uninstall Fabric Services System software and the related data, log in to the deployer VM and run following command:

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

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.

Related topics

[Reinstalling Fabric Services System software only](#)

6.2 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.

- To reinstall the Fabric Services System software, log in to the deployer VM and enter the following command:

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

Related topics

[Uninstalling Fabric Services System software only](#)

7 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



Note: If you want to bring a new FSS cluster, Nokia recommends that you delete and recreate the cluster.

To quickly uninstall the Fabric Services system cluster and the deployer VM, delete the VMs where the Fabric Services System cluster and deployer are running or use the uninstall scripts as described in the following procedures.

7.1 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.

7.2 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, 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.

8 Software upgrade and rollback

The Fabric Services System supports software upgrades from one release to a newer one in a KVM or VMware vSphere environment running in a supported configuration (with one deployer VM and one, three, or six Fabric Services System virtual machine nodes).

The procedures in this section describe how to prepare for the software upgrade and execute the software upgrade. The rollback procedure describes how to roll back after an upgrade, if needed.

The supported upgrade path is as follows:

Release 22.12.2 to Release 23.4.1



Note: In the software upgrade and rollback procedures, Release 22.12.2 is referred to as the *old* release and Release 23.4.1 is the *new* release.

8.1 Preparing for software upgrade

About this task

Use this procedure to prepare for the software upgrade from the *old* release to the *new* release.

Procedure

Step 1. Display the system information and status.

Log in to the old deployer VM using SSH.

a. Verify that the Helm charts are running the current release.

Example

```
[root@fss-deployer ~]# export KUBECONFIG=/var/lib/fss/config.fss
[root@fss-deployer ~]# helm list
NAME NAMESPACE REVISION UPDATED                               STATUS  CHART
prod default 1      2023-05-29 06:08:37.917207827 +0000 UTC deployed fss-FSS_22_
12_B1-charts-v22.12.2-11 1.0
```

b. Verify that the pods are all up and running.

Example

```
[root@fss-deployer ~]# kubectl get pods
NAME                                READY  STATUS   RESTARTS  AGE
prod-cp-kafka-0                     2/2    Running  2 (32m ago)  35m
prod-cp-kafka-1                     2/2    Running  1 (32m ago)  33m
prod-cp-kafka-2                     2/2    Running  0           32m
prod-cp-zookeeper-0                 2/2    Running  0           35m
prod-cp-zookeeper-1                 2/2    Running  0           32m
prod-cp-zookeeper-2                 2/2    Running  0           32m
prod-ds-apiserver-75ffbd8dd8-bbgt2  1/1    Running  0           35m
prod-ds-cli-67bc6c89c7-dfdbn        1/1    Running  0           35m
prod-ds-docker-registry-589fb466d5-tdltk  1/1    Running  0           35m
prod-ds-imgsvc-deploy-66dc945bf6-9k24k  1/1    Running  0           35m
```


prod-fss-alarmmgr-7b6ff65f97-mj5x8	1/1	Running	0	35m
prod-fss-auth-57648dff7c-vv8j2	1/1	Running	0	35m
prod-fss-catalog-75b6b87c69-rqg87	1/1	Running	0	35m
prod-fss-cfggen-6df55b9fd9-hnj99	1/1	Running	0	35m
prod-fss-cfgsync-6ccdd5474f-ppgfj	1/1	Running	0	35m
prod-fss-connect-686589bb65-vx8wj	1/1	Running	0	35m
prod-fss-da-58b4fbbd5-42d52	1/1	Running	0	35m
prod-fss-da-58b4fbbd5-674tl	1/1	Running	0	35m
prod-fss-da-58b4fbbd5-7vzw8	1/1	Running	0	35m
prod-fss-da-58b4fbbd5-fskd8	1/1	Running	0	35m
prod-fss-da-58b4fbbd5-lj5wz	1/1	Running	0	35m
prod-fss-da-58b4fbbd5-plx6s	1/1	Running	0	35m
prod-fss-da-58b4fbbd5-tg67x	1/1	Running	0	35m
prod-fss-da-58b4fbbd5-w8gg7	1/1	Running	0	35m
prod-fss-deviationmgr-acl-66c754c6cc-h7c25	1/1	Running	0	35m
prod-fss-deviationmgr-bfd-65c5ff9457-45sgz	1/1	Running	0	35m
prod-fss-deviationmgr-interface-58fc8bb77d-xfdlp	1/1	Running	0	35m
prod-fss-deviationmgr-netinst-b8cbccb9d-9gtl6	1/1	Running	0	35m
prod-fss-deviationmgr-platform-7c9f6bf67b-x96mn	1/1	Running	0	35m
prod-fss-deviationmgr-qos-6b97dcc884-8w5dv	1/1	Running	0	35m
prod-fss-deviationmgr-routingpolicy-67df5489cc-w6tpc	1/1	Running	0	35m
prod-fss-deviationmgr-system-67b85d8974-bgd8l	1/1	Running	0	35m
prod-fss-dhcp-7b77b55746-27mh7	1/1	Running	0	35m
prod-fss-digitalandbox-7c84dbfb85-wgngf	1/1	Running	0	35m
prod-fss-filemgr-6d6cfcc68b-4wz4j	1/1	Running	0	35m
prod-fss-imagmgr-c8d7cdfb5-hsgnq	1/1	Running	0	35m
prod-fss-intentmgr-858587457f-ppk6g	1/1	Running	0	35m
prod-fss-inventory-69bdbf979-wp852	1/1	Running	0	35m
prod-fss-labelmgr-56cf479f99-ljbkm	1/1	Running	0	35m
prod-fss-maintmgr-569c964d89-4j7l7	1/1	Running	0	35m
prod-fss-mgmtstack-7dcddf4479-hlpd8	1/1	Running	0	35m
prod-fss-oper-da-fdb879d95-28xj4	1/1	Running	0	35m
prod-fss-oper-da-fdb879d95-gpvd4	1/1	Running	0	35m
prod-fss-oper-da-fdb879d95-hslvh	1/1	Running	0	35m
prod-fss-oper-da-fdb879d95-nsbgd	1/1	Running	0	35m
prod-fss-oper-da-fdb879d95-qtplh	1/1	Running	0	35m
prod-fss-oper-da-fdb879d95-x7rgg	1/1	Running	0	35m
prod-fss-oper-da-fdb879d95-xd964	1/1	Running	0	35m
prod-fss-oper-da-fdb879d95-xdczn	1/1	Running	0	35m
prod-fss-oper-topomgr-598c95fbc4-cm5hl	1/1	Running	0	35m
prod-fss-protocolmgr-5d67f4548b-9vxs7	1/1	Running	0	35m
prod-fss-topomgr-5d8dd6597f-n76xq	1/1	Running	0	35m
prod-fss-transaction-b678ff785-czgcz	1/1	Running	0	35m
prod-fss-version-758dfd6c49-7wxvg	1/1	Running	0	35m
prod-fss-web-59c677689-44dtr	1/1	Running	0	35m
prod-fss-workloadmgr-7469c5b46b-whgdw	1/1	Running	0	35m
prod-fss-ztp-646cd8df76-9gmt4	1/1	Running	0	35m
prod-keycloak-0	1/1	Running	0	35m
prod-mongodb-arbiter-0	1/1	Running	0	35m
prod-mongodb-primary-0	1/1	Running	0	35m
prod-mongodb-secondary-0	1/1	Running	0	35m
prod-neo4j-core-0	1/1	Running	0	35m
prod-postgresql-0	1/1	Running	0	35m
prod-sftpserver-b68849d55-gsv7m	1/1	Running	0	35m

Step 2. Back up the necessary files from the deployer VM.

Copy the following files to a directory that is outside of the deployer VM:

- SSH files
 - /root/.ssh/id_rsa
 - /root/.ssh/id_rsa.pub

/root/.ssh/known_hosts

- the input JSON configuration and user values YAML file (if any) that were used to install the Fabric Services System, for example:

/root/customer-input.json

/root/user_values.yaml

- the HTTPS related certificate files that were used to install Fabric Services System, for example:

/root/tls.crt

/root/tls.key

- optional files; for example, SRL image tar files that were added to the old deployer VM

Step 3. Back up the Fabric Services System.

For instructions, see “Backup and restore” in the *Fabric Services System User Guide*.

Step 4. Temporarily remove all ACL and QoS profiles configured in the system.

See the *Release Notes* for background information about this step. For instructions, see the “ACL profile management” and “QoS profile management” sections in the *Fabric Services System User Guide*.

Step 5. Shut down the old deployer VM.



Note: Ensure that the old deployer VM is not configured to automatically restart.

Step 6. Create a new deployer VM using the image for the new release.

For instructions, see [Fabric Services System deployer VM creation](#). The new deployer should have the same network configuration and the same IP address of old deployer VM.

Step 7. Copy the files that you saved in Step 2 to the new deployer VM in the same locations.

What to do next

[Performing a software upgrade](#)

8.2 Performing a software upgrade

About this task

You use the **fss-upgrade.sh** script to perform the software upgrade steps. The command syntax is as follows:

```
[root@fss-deployer ~]# /root/bin/fss-upgrade.sh
Usage: /root/bin/fss-upgrade.sh configure <config-file> - Reads config file and configures accordingly.
./bin/fss-upgrade.sh discover - Discovers cluster.
./bin/fss-upgrade.sh upgrade - Upgrades FSS application.
./bin/fss-upgrade.sh status - Prints status of cluster and FSS Chart information.
./bin/fss-upgrade.sh clean - Cleans temporary files.
./bin/fss-upgrade.sh [help | -h] - Prints usage.
```

This script updates the entire Fabric Services System application using the new Helm charts.



Note: The Fabric Services System 23.4 release significantly improves the infrastructure of several components. Consequently, the upgrade process requires that you update these component in several steps; the high-level steps are as follows:

1. Using the new Deployer VM, upgrade the Fabric Services System application in place on the existing Kubernetes environment.
2. Take a backup of the new release using the Deployer VM, as described in “Backup and restore” in the *Fabric Services System User Guide*.
3. Shut down and remove the Fabric Services System virtual machine nodes.
4. Use the latest base image to deploy new Fabric Services System virtual machine nodes, as described in [Preparing for software upgrade](#).
5. Restore the application from the back taken in Step 2, as described in “Backup and restore” in the *Fabric Services System User Guide*.

Prerequisites

Ensure that you have completed the steps described in [Preparing for software upgrade](#). Execute the following steps while logged in to the new deployer VM that you created in [Preparing for software upgrade](#).

Procedure

Step 1. Update the configuration.

Example

In the following example, the input configuration file is `customer-input.json`.

```
[root@fss-deployer ~]# /root/bin/fss-upgrade.sh configure customer-input.json
Timesync service is running on 10.254.106.74 Time difference is 0 seconds
Timesync service is running on 10.254.106.75 Time difference is 0 seconds
Timesync service is running on 10.254.106.76 Time difference is 0 seconds
Timesync service is running on 10.254.106.77 Time difference is 0 seconds
Timesync service is running on 10.254.106.78 Time difference is 0 seconds
Timesync service is running on 10.254.106.79 Time difference is 0 seconds
Maximum time difference between nodes 0 seconds
Successfully configured. Please run /root/bin/fss-upgrade.sh discover
```

Step 2. Discover the cluster.

Example

```
[root@fss-deployer ~]# /root/bin/fss-upgrade.sh discover
NAME           STATUS    ROLES                    AGE   VERSION
fss-node01     Ready    control-plane,master    72m   v1.23.1
fss-node02     Ready    control-plane,master    71m   v1.23.1
fss-node03     Ready    control-plane,master    71m   v1.23.1
fss-node04     Ready    <none>                   70m   v1.23.1
fss-node05     Ready    <none>                   70m   v1.23.1
fss-node06     Ready    <none>                   70m   v1.23.1
FSS release discovered fss-FSS_22_12_B1-charts-v22.12.2-11
```

Step 3. Upgrade the Fabric Services System application.

Example

```
[root@fss-deployer ~]# /root/bin/fss-upgrade.sh upgrade
```

```

NAME                STATUS    ROLES                    AGE   VERSION
fss-node01          Ready    control-plane,master    75m   v1.23.1
fss-node02          Ready    control-plane,master    74m   v1.23.1
fss-node03          Ready    control-plane,master    74m   v1.23.1
fss-node04          Ready    <none>                  73m   v1.23.1
fss-node05          Ready    <none>                  73m   v1.23.1
fss-node06          Ready    <none>                  73m   v1.23.1
FSS will be upgraded from fss-FSS_22_12_B1-charts-v22.12.2-11 to fss-FSS_23_4_B1-
charts v23.4.1-109 : Are you sure [YyNn]? y
Upgrade in progress...
Upgrading fss-logs
Upgrading traefik and ingress routes
traefik release discovered: traefik-10.14.2 ; Deployer packages traefik release:
traefik 21.0.0
Upgrading kafka and kafkaop if required
KAFKA not present. Installing fresh
Error from server (NotFound): secrets "kafka-fss-cluster-ca-cert" not found
Error from server (NotFound): secrets "kafka-fss-clients-ca-cert" not found
W0529 07:14:29.567723    7676 warnings.go:70] policy/v1beta1 PodDisruptionBudget is
  deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W0529 07:14:29.575490    7676 warnings.go:70] policy/v1beta1 PodDisruptionBudget is
  deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
Release "prod" has been upgraded. Happy Helming!
NAME: prod
LAST DEPLOYED: Mon May 29 07:14:18 2023
NAMESPACE: default
STATUS: deployed
REVISION: 2
NOTES:

  Checking for FSS pods
    Waiting for default fss-auth 1 Running
    Waiting for FSS pods to comeup
    Waiting for default fss-auth 1 Running
    Waiting for FSS pods to comeup
    Waiting for default fss-cfgsync 1 Running
    Waiting for FSS pods to comeup
  All FSS pods are running
  Checking for FSS digitalsandbox pods
  FSS digital sandbox pods are running
  Checking for digitalsandbox pods
  Digital sandbox pods are running

  FSS is ready, you can access FSS using https://prod.k8s.nokia.com

```

Step 4. Verify that the upgrade is successful.

- a. Check the status of the compute nodes.

Example

The STATUS field should show Ready.

```

[root@fss-deployer ~]# /root/bin/fss-upgrade.sh status
NAME                STATUS    ROLES                    AGE   VERSION
fss-node0#          Ready    control-plane,master    27m   v1.23.1
fss-node02          Ready    control-plane,master    27m   v1.23.1
fss-node03          Ready    control-plane,master    27m   v1.23.1
fss-node04          Ready    <none>                  26m   v1.23.1
fss-node05          Ready    <none>                  26m   v1.23.1
fss-node06          Ready    <none>                  26m   v1.23.1
FSS release discovered: fss-FSS_23_4_B1-charts-v23.4.1-yyy ; Deployer packages FSS
release: fss-FSS_23_4_B1-charts v23.4.1-yyy
Checking for FSS pods

```

```
All FSS pods are running
Checking for FSS digitalsandbox pods
FSS digital sandbox pods are running
Checking for digitalsandbox pods
Digital sandbox pods are running

FSS is ready, you can access FSS using https://myhost.mydomain.com
```

- b. Display the Helm chart version.

Example

```
[root@fss-deployer ~]# export KUBECONFIG=/var/lib/fss/config.fss
[root@fss-deployer ~]# helm list
NAME          NAMESPACE REVISION UPDATED                               STATUS  CHART
fss-logs     default   1       2023-05-25 19:49:37.968802894 +0000 UTC deployed fluent-bit-0.20.9
kafka        default   1       2023-05-25 20:00:16.161414151 +0000 UTC deployed fss-strimzi-kafka-0.1.8
prod         default   1       2023-05-25 20:02:21.883150466 +0000 UTC deployed fss-FSS_23_4_B1-charts-
v23.4.1-xxx v23.4.1
```

- c. Check the status of system services and processes.

Expected outcome

```
[root@fss-deployer ~]# kubectl get pods -A
NAMESPACE     NAME                                     READY   STATUS
RESTARTS      AGE
cert-manager   cert-manager-78759ccb7c-4pssc          1/1     Running
0             12h
cert-manager   cert-manager-cainjector-7d487f784-zkr55 1/1     Running
0             12h
cert-manager   cert-manager-webhook-6dbcf856cc-62l4w   1/1     Running
0             12h
default        fss-logs-fluent-bit-6sn2k              1/1     Running
0             12h
default        fss-logs-fluent-bit-8h6wb              1/1     Running
0             12h
default        fss-logs-fluent-bit-9wmtr              1/1     Running
0             12h
default        fss-logs-fluent-bit-ltnc8              1/1     Running
0             12h
default        fss-logs-fluent-bit-rlzts              1/1     Running
0             12h
default        fss-logs-fluent-bit-w8fgk              1/1     Running
0             12h
default        prod-ds-apiserver-67cd5696dc-6vdhp      1/1     Running
0             12h
default        prod-ds-cli-d98576cc8-pbzxm            1/1     Running
0             12h
default        prod-ds-docker-registry-5b467bbf67-29bb2 1/1     Running
0             12h
default        prod-ds-imagsvc-deploy-958b4bb6f-f9cgg   1/1     Running
0             12h
default        prod-fss-alarmmgr-79669d987c-4m26p      1/1     Running
0             12h
default        prod-fss-auth-68d7797465-dvbqb         1/1     Running
0             12h
default        prod-fss-catalog-d586f956-tl8z7        1/1     Running
0             12h
```

default		prod-fss-cfggen-7654fd9679-g9qn7	1/1	Running
0	12h			
default		prod-fss-cfgsync-5b7487bcfd-757n9	1/1	Running
0	12h			
default		prod-fss-connect-5786db898c-jlnnr	1/1	Running
0	12h			
default		prod-fss-da-0	1/1	Running
0	12h			
default		prod-fss-da-1	1/1	Running
0	12h			
default		prod-fss-da-2	1/1	Running
0	12h			
default		prod-fss-da-3	1/1	Running
0	12h			
default		prod-fss-da-4	1/1	Running
0	12h			
default		prod-fss-da-5	1/1	Running
0	12h			
default		prod-fss-da-6	1/1	Running
0	12h			
default		prod-fss-da-7	1/1	Running
0	12h			
default		prod-fss-deviationmgr-acl-5c5cfc76c-fwpx8	1/1	Running
0	12h			
default		prod-fss-deviationmgr-bfd-6d7b554f9f-xml4n	1/1	Running
0	12h			
default		prod-fss-deviationmgr-interface-6c79fcc578-4748w	1/1	Running
0	12h			
default		prod-fss-deviationmgr-netinst-57d885f474-dzbsc	1/1	Running
0	12h			
default		prod-fss-deviationmgr-platform-f8cb8c5f6-f5dw9	1/1	Running
0	12h			
default		prod-fss-deviationmgr-qos-78986d7f56-77jbp	1/1	Running
0	12h			
default		prod-fss-deviationmgr-routingpolicy-568bc7b5c5-nt4b7	1/1	Running
0	12h			
default		prod-fss-deviationmgr-system-86d9459cc4-q85n8	1/1	Running
0	12h			
default		prod-fss-dhcp-5f4bd96ffb-475rv	1/1	Running
0	12h			
default		prod-fss-dhcp6-b65449855-mqf5v	1/1	Running
0	12h			
default		prod-fss-digitalsandbox-8fd7c5b55-2zf5v	1/1	Running
0	12h			
default		prod-fss-filemgr-75c6dfc75-sm77t	1/1	Running
0	12h			
default		prod-fss-imagmgr-69d997488c-ph9p2	1/1	Running
0	12h			
default		prod-fss-intentmgr-c644856c8-x82x5	1/1	Running
0	12h			
default		prod-fss-inventory-7ff597c8dc-66ql9	1/1	Running
0	12h			
default		prod-fss-labelmgr-85cd5d646c-cwvzk	1/1	Running
0	12h			
default		prod-fss-maintmgr-7874767b7c-rg5qv	1/1	Running
0	12h			
default		prod-fss-mgmtstack-659c7755b4-5mw68	1/1	Running
0	12h			
default		prod-fss-oper-da-0	1/1	Running
0	12h			
default		prod-fss-oper-da-1	1/1	Running
0	12h			
default		prod-fss-oper-da-2	1/1	Running
0	12h			

default		prod-fss-oper-da-3	1/1	Running
0	12h			
default		prod-fss-oper-da-4	1/1	Running
0	12h			
default		prod-fss-oper-da-5	1/1	Running
0	12h			
default		prod-fss-oper-da-6	1/1	Running
0	12h			
default		prod-fss-oper-da-7	1/1	Running
0	12h			
default		prod-fss-oper-topomgr-7b7f9fc7fd-dcqfr	1/1	Running
0	12h			
default		prod-fss-protocolmgr-5c877747bb-r5b7z	1/1	Running
0	12h			
default		prod-fss-topomgr-d8b5c899-zxcsl	1/1	Running
0	12h			
default		prod-fss-transaction-65f595d6df-ghfhx	1/1	Running
0	12h			
default		prod-fss-version-6d87bf6cc5-9fvbz	1/1	Running
0	12h			
default		prod-fss-web-7dcfdccb5d-r85cp	1/1	Running
0	12h			
default		prod-fss-workloadmgr-55df864cc-d8spn	1/1	Running
0	12h			
default		prod-fss-ztp-75f44cc674-4p6lv	1/1	Running
0	12h			
default		prod-keycloak-0	1/1	Running
0	12h			
default		prod-mongodb-arbiter-0	1/1	Running
0	12h			
default		prod-mongodb-primary-0	1/1	Running
0	12h			
default		prod-mongodb-secondary-0	1/1	Running
0	12h			
default		prod-neo4j-core-0	1/1	Running
0	12h			
default		prod-postgresql-0	1/1	Running
0	12h			
default		prod-sftpserver-77cd8696d5-8j689	1/1	Running
0	12h			
digital-sandbox-system		prod-netsegmd-5dvk6	1/1	Running
0	12h			
digital-sandbox-system		prod-netsegmd-72hzp	1/1	Running
1 (12h ago)	12h			
digital-sandbox-system		prod-netsegmd-c7bh4	1/1	Running
0	12h			
digital-sandbox-system		prod-netsegmd-gzztk	1/1	Running
1 (12h ago)	12h			
digital-sandbox-system		prod-netsegmd-hrmw8	1/1	Running
1 (12h ago)	12h			
digital-sandbox-system		prod-netsegmd-jr9zf	1/1	Running
0	12h			
digital-sandbox-system		srlfabrics-controller-manager-579ccbfb8b-v77gk	2/2	Running
0	12h			
kafka		kafka-fss-entity-operator-b6757b664-kshcl	3/3	Running
0	12h			
kafka		kafka-fss-kafka-0	1/1	Running
0	12h			
kafka		kafka-fss-kafka-1	1/1	Running
0	12h			
kafka		kafka-fss-kafka-2	1/1	Running
0	12h			
kafka		kafka-fss-zookeeper-0	1/1	Running
0	12h			

kafka		kafka-fss-zookeeper-1	1/1	Running
0	12h			
kafka		kafka-fss-zookeeper-2	1/1	Running
0	12h			
kafka		strimzi-cluster-operator-5bc66cb4f9-w6nkd	1/1	Running
0	12h			
kube-system		coredns-7947bbfb58-27pgt	1/1	Running
0	12h			
kube-system		coredns-7947bbfb58-snjff	1/1	Running
0	12h			
kube-system		dns-autoscaler-74d7b6c6c-l4f2h	1/1	Running
0	12h			
kube-system		kube-apiserver-fss-node01	1/1	Running
1	12h			
kube-system		kube-apiserver-fss-node02	1/1	Running
1	12h			
kube-system		kube-controller-manager-fss-node03	1/1	Running
1	12h			
kube-system		kube-controller-manager-fss-node04	1/1	Running
1	12h			
kube-system		kube-flannel-g8tr2	1/1	Running
0	12h			
kube-system		kube-flannel-hvm7p	1/1	Running
0	12h			
kube-system		kube-flannel-n6fpp	1/1	Running
0	12h			
kube-system		kube-flannel-p86hz	1/1	Running
0	12h			
kube-system		kube-flannel-rjw8z	1/1	Running
0	12h			
kube-system		kube-flannel-rn8wr	1/1	Running
0	12h			
kube-system		kube-proxy-b646r	1/1	Running
0	12h			
kube-system		kube-proxy-dv6dx	1/1	Running
0	12h			
kube-system		kube-proxy-kzm7m	1/1	Running
0	12h			
kube-system		kube-proxy-qk5lp	1/1	Running
0	12h			
kube-system		kube-proxy-x74mh	1/1	Running
0	12h			
kube-system		kube-proxy-zh5hd	1/1	Running
0	12h			
kube-system		kube-scheduler-fss-node01	1/1	Running
1	12h			
kube-system		kube-scheduler-fss-node02	1/1	Running
1	12h			
kube-system		nginx-proxy-fss-node03	1/1	Running
0	12h			
kube-system		nginx-proxy-fss-node04	1/1	Running
0	12h			
kube-system		nginx-proxy-fss-node05	1/1	Running
0	12h			
kube-system		nginx-proxy-fss-node06	1/1	Running
0	12h			
kube-system		snapshot-controller-55dbff48d-28ptd	1/1	Running
0	12h			
kube-system		snapshot-controller-55dbff48d-jzvtw	1/1	Running
0	12h			
kube-system		traefik-2f8d9	1/1	Running
0	12h			
kube-system		traefik-2j7hr	1/1	Running
0	12h			

kube-system		traefik-dwf27	1/1	Running
0	12h			
kube-system		traefik-q545f	1/1	Running
0	12h			
kube-system		traefik-qgp74	1/1	Running
0	12h			
kube-system		traefik-vfj8m	1/1	Running
0	12h			
rook-ceph		csi-cephfsplugin-6pfsm	3/3	Running
0	12h			
rook-ceph		csi-cephfsplugin-95bm4	3/3	Running
0	12h			
rook-ceph		csi-cephfsplugin-ddgw2	3/3	Running
0	12h			
rook-ceph		csi-cephfsplugin-fc6sn	3/3	Running
0	12h			
rook-ceph		csi-cephfsplugin-nj776	3/3	Running
0	12h			
rook-ceph		csi-cephfsplugin-provisioner-59dd67b6c-c2b4f	6/6	Running
0	12h			
rook-ceph		csi-cephfsplugin-provisioner-59dd67b6c-x2cdh	6/6	Running
0	12h			
rook-ceph		csi-cephfsplugin-w5wdn	3/3	Running
0	12h			
rook-ceph		rook-ceph-mds-ceph-filesystem-a-86bbf4ff-kdt24	1/1	Running
0	12h			
rook-ceph		rook-ceph-mds-ceph-filesystem-b-77df4fbfcc-thc26	1/1	Running
0	12h			
rook-ceph		rook-ceph-mgr-a-7554d5cdbc-qpq6w	2/2	Running
0	12h			
rook-ceph		rook-ceph-mgr-b-7f6ccbc99f-zxzj5	2/2	Running
0	12h			
rook-ceph		rook-ceph-mon-a-6666946d94-c86ck	1/1	Running
0	12h			
rook-ceph		rook-ceph-mon-b-744574b89f-hjvfw	1/1	Running
0	12h			
rook-ceph		rook-ceph-mon-c-7585f86975-jqqb5	1/1	Running
0	12h			
rook-ceph		rook-ceph-operator-77cf896f-7slvw	1/1	Running
0	12h			
rook-ceph		rook-ceph-osd-0-775f95d9f5-shrpl	1/1	Running
0	12h			
rook-ceph		rook-ceph-osd-1-59d94965b8-hg75j	1/1	Running
0	12h			
rook-ceph		rook-ceph-osd-2-84bb4966db-nv6mb	1/1	Running
0	12h			
rook-ceph		rook-ceph-osd-prepare-fss-node01	0/1	
Completed	0	12h		
rook-ceph		rook-ceph-osd-prepare-fss-node02	0/1	
Completed	0	12h		
rook-ceph		rook-ceph-osd-prepare-fss-node03	0/1	
Completed	0	12h		
rook-ceph		rook-ceph-tools-c859b5678-hnpsb	1/1	Running
0	12h			

Step 5. Restore previously added software catalog images.

The upgrade procedure replaces the software catalog. Changes that you made to the software catalog are lost after the upgrade. If you previously added new SR Linux images to the software catalog, you need to modify the software catalog and then upload the SR Linux images. For instructions, see “Adding a network operating system version to the software catalog” in the *Fabric Services System User Guide*. You must complete this step before you can access the Fabric Services System to make any change.

What to do next

[Upgrading the Fabric Services System cluster](#)

8.3 Upgrading the Fabric Services System cluster

About this task

After you complete the procedure [Performing a software upgrade](#), the Fabric Services System software is upgraded to the new release.



Note: Release 23.4.1 also supports new versions of the compute VM OS and Kubernetes, so you need to migrate data in the cluster to the new schema, as described in this procedure.

Procedure

Step 1. Back up the Fabric Services System.

For instructions, see “Backup and restore” in the *Fabric Services System User Guide*.

Step 2. Make a copy of the `customer-input.json` file, certificates, SR Linux images and other user-created files from the old deployer VM to a safe location.

Step 3. Delete all the compute VMs.

Step 4. Recreate the compute VM with the Release 23.4.1 base image.

The number of compute VMs, hostname and IP address details should be same as the previous FSS cluster.

Step 5. Copy the `customer-input.json` file, certificates, SR Linux images and other user-created files to the new deployer VM.

Step 6. Reset the old cluster configurations.

Log in to the deployer VM running the new release and execute the `/root/bin/fss-install.sh reset` command.

Step 7. Set up key-based authentication from the deployer VM to all the compute VMs.

Example

If password authentication is enabled on the node for SSH, enter the following command from the deployer VM:

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

Step 8. Restore the backup that you created in Step 1.

For instructions, see “Backup and restore” in the *Fabric Services System User Guide*.

What to do next

If you want to configure the LDAP settings or remote syslog capabilities, complete the following procedures in the *Fabric Services System User Guide*:

- “Configuring LDAP after software upgrade”
- “Configuring a remote syslog server for user audit logs”

8.4 Performing a software rollback

About this task

If a software upgrade is unsuccessful, use the following steps to roll back from the new release to the old release.

Procedure

- Step 1.** Shut down and delete the deployer VM and Fabric Services System virtual machine nodes for the upgraded deployment.
- Step 2.** Deploy and configure the deployer VM with the old release.
For instructions, see the procedures in [The Fabric Services System deployer VM](#).
- Step 3.** Deploy new base virtual machine nodes.
For instructions, see [Preparing the Fabric Services System virtual machine nodes](#).
- Step 4.** Restore the saved backup of the old release.
For instructions, see “Backup and restore” in the *Fabric Services System User Guide*.

Customer document and product support



Customer documentation

[Customer documentation welcome page](#)



Technical support

[Product support portal](#)



Documentation feedback

[Customer documentation feedback](#)