NOKIA

Fabric Services System

SOFTWARE INSTALLATION GUIDE
RELEASE 22.8.2

3HE 19154 AAAA
Issue 1.0

October 2022

Table of contents

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 section lists the changes that were made in this release.

Table 1: What's new in Release FSS 22.8.2

Topic	Location
Support for software upgrade from Release FSS 22.8.1 to FSS 22.8.2	Software upgrade and rollback Preparing for software upgrade Performing a software upgrade Performing a software rollback

Table 2: What's new

Topic	Location
Support for deployment on a single VM without DS for small scale environments	Supported deployment models Fabric Services System nodes
Support for deployment on VMware vSphere	Updated topics in the following sections: • Installation overview • Fabric Services System deployer • Preparing the Fabric Services System virtual machine nodes • Fabric Services System installation • Software reinstallation on an existing Kubernetes cluster • Uninstalling software

Topic	Location
Removed the section *Bare metal-based installation: Preparing the Fabric Services System nodes*.	

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 one, three, or six Fabric Services System nodes. The Fabric Services System is deployed as an application on virtual machine servers.

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
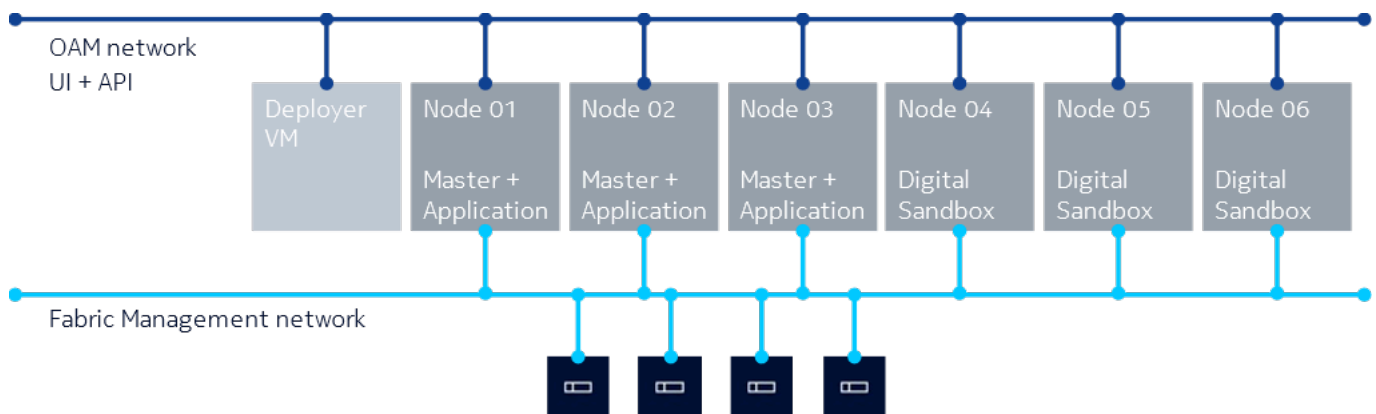
2.1.1.1 DHCP and ZTP

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

- 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 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:

- a single-node environment that hosts only the Fabric Services System application for very small scale deployments and does not support the use of Digital Sandbox

- 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-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 - this file provides the deployer VM. This image 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: 8GB RAM
- Storage: 50GB of storage (used by the deployer QCOW2 image)
- Networking:
    - 1GbE network interface card (NIC)
    - The configured DNS servers must be reachable, functional, and able to resolve the hostnames used for the FSS nodes.

2.1.4.2 Fabric Services System node requirements

The Fabric Services System nodes is 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: 64GB
- Storage: At least 500GB of available SSD-based storage
- Networking:
    - At least one 10Gbps 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/CentOS 7.9
- 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 section describes how to prepare and configure the nodes for deployment.

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 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 (OVA or QCOW2).

Table 3: 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/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. You can configure networking for the deployer VM using a bridged network using one of the following procedures:

- 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 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
```

3.3.2 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 argument.

```
$ ovftool fss-deployer_v22.8.1-16.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.

Example

Make sure to 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.

```
$ ovftool --acceptAllEulas -dm=thick -ds=VSAN -n=fss-deployer --net:"OAM=OAM-
network" --net:"FABRIC=Fabric-network" fss-deployer_v22.8.1-16.ova vi://administrator
%40vsphere.local@vcenter.domain.tld/My-Datacenter/host/My-Cluster/Resources/My-
Resource-Group

Opening OVA source: fss-deployer_v22.8.1-16.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 so it can log in over SSH. In any case, this configuration is needed for the deployer VM to reach the nodes.

Step 4. 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 5. Restart the network to apply the new configuration.

Execute the following command:

```
# systemctl restart network
```

Step 6. 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 7. Restart the chronyd service.

```
# systemctl restart chronyd
```

Step 8. 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 9. Synchronize the RTC clock and the system clock.

Example

```
# hwclock --systohc
```

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

```
# timedatectl
```

Step 10. Optional: Change the hostname.

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

Step 11. 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 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).

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 network 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 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
```

What to do next

Configuring the Fabric Services System virtual machine

4.3.2 Creating the VM on bridged network 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 argument.

```
$ ovftool fss-baseos_v22.8.1-16.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.

Example

Make sure to 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 yo power on.

```
$ ovftool --acceptAllEulas -dm=thick -ds=VSAN -n=fss-node01 --net:"OAM=OAM-
network" --net:"FABRIC=Fabric-network" fss-baseos_v22.8.1-16.ova vi://administrator
%40vsphere.local@vcenter.domain.tld/My-Datacenter/host/My-Cluster/Resources/My-
Resource-Group

Opening OVA source: fss-base_v22.8.1-16.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 VMware vCenter and add a new 300GB 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. 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 4. Delete the existing partition.

Example

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

Step 5. 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 6. Set the boot flag for this partition.

Example

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

Step 7. 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 8. 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 9. Reboot the node to register the change in partition sizes in the system.

Example

```
# reboot
```

Step 10. 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 11. 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 12. 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 13. 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 14. Restart the network to apply the new configuration.

Example

```
# systemctl restart network
```

Step 15. 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 16. Restart the chronyd service.

```
# systemctl restart chronyd
```

Step 17. 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 18. Change the hostname.

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

Step 19. 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 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

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

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

```
[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, (`"sdb"` or `"vdb"` 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 4: 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
  },
  "fss": {

    "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": [
              "vdb"
            ]
          },
          {
            "hostip": "192.0.2.15",
            "hostname": "fss-node05",
            "devices": [
              "vdb"
            ]
          },
          {
            "hostip": "192.0.2.16",
            "hostname": "fss-node06",
            "devices": [
              "vdb"
            ]
          }
        ]
}
```

Table 4: Configuration file fields

Heading	Configurable values
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.	• **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
digitalsandbox Specifies Digital Sandbox parameters.	**enabled**: when this flag is set to `true`, the Digital Sandbox component is installed. Ensure that at least one worker node is tagged with the `digitalsandbox` role. When set to `false`, the Digital Sandbox component is not installed.

fss
Specifies Fabric Services System deployment options.
	• **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.
	
	• **dhcpnode**: specifies a node on which the Fabric Services System DHCP pod is scheduled
	
	• **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.
	
	• **httpsenabled**: Specifies whether HTTPS is enabled or disabled on the API and UI. This field is mandatory. Possible values:
	   – `false`: disables HTTPS on the API and UI
	   – `true`: 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.

Heading	Configurable values
	• **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.
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.	• **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 `digitalsandbox`. For Kubernetes master nodes, specify this value as `master`.
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	• **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 `xxx`. 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 `false` indicating that the deployment is a standard 3 or 6 node deployment. If set to `true`, the deployment is set up on a single node and has no redundancy built in.

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 ~]$ fss-uninstall.shfss-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:

Procedure

Step 1. To reinstall the Fabric Services System software, log in to the deployer VM and enter the following command:

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

Step 2. When the re-installation is complete, unset the *SKIP_K8S* environment variable.

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

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:** To quickly uninstall the Fabric Services system cluster and the deployer VM, simply delete the VMs where the Fabric Services System cluster and deployer are running. Another way is to uninstall scripts as discussed 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 ~]$ /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 upgrade 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 rollback after an upgrade, if needed.

The supported upgrade path is as follows:

22.8.1 to 22.8.x (where x > 1)

8.1 Preparing for software upgrade

About this task

Use this procedure to prepare for the software upgrade from Fabric Services System 22.8.x to Fabric Services System 22.8.y (where y > x).

Procedure

Step 1. Display system information and status.

Log in to the 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                                   APP VERSION
prod     default        1               2022-10-27 17:53:04.124504577 +0000 UTC
 deployed        fss-FSS_22_8_B1-charts-v22.8.1-108      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 (6m31s
 ago)   9m10s
prod-cp-kafka-1                                      2/2      Running   1 (6m45s
 ago)   8m5s
prod-cp-kafka-2                                      2/2      Running   0
      7m29s
prod-cp-zookeeper-0                                  2/2      Running   0
      9m10s
prod-cp-zookeeper-1                                  2/2      Running   0
      7m14s
```

```
prod-cp-zookeeper-2                                    2/2      Running   0
     6m51s
prod-fss-alarmmgr-6678f4d49c-xz77v                     1/1      Running   0
     9m7s
prod-fss-auth-79ff5f495c-s4t7m                         1/1      Running   0
     9m6s
prod-fss-catalog-7658bc4ccf-wp7vn                      1/1      Running   0
     9m9s
prod-fss-cfggen-5d8f94f74b-57nd7                       1/1      Running   0
     9m7s
prod-fss-cfgsync-5b69b7f854-l9lx5                      1/1      Running   0
     9m8s
prod-fss-connect-75df8ff7f9-vwfgj                      1/1      Running   0
     9m7s
prod-fss-da-69f8676b6d-4kjsn                           1/1      Running   0
     9m8s
prod-fss-da-69f8676b6d-cp5v6                           1/1      Running   0
     9m9s
prod-fss-da-69f8676b6d-fj7db                           1/1      Running   0
     9m8s
prod-fss-da-69f8676b6d-mnz9b                           1/1      Running   0
     9m8s
prod-fss-da-69f8676b6d-ph42r                           1/1      Running   0
     9m8s
prod-fss-da-69f8676b6d-vgflj                           1/1      Running   0
     9m9s
prod-fss-da-69f8676b6d-zg2pb                           1/1      Running   0
     9m9s
prod-fss-da-69f8676b6d-zt7z6                           1/1      Running   0
     9m8s
prod-fss-deviationmgr-acl-d8949f794-2dl79              1/1      Running   0
     9m8s
prod-fss-deviationmgr-bfd-68f5d7988b-q8vks             1/1      Running   0
     9m6s
prod-fss-deviationmgr-interface-769c5d46f-l9wrc        1/1      Running   0
     9m6s
prod-fss-deviationmgr-netinst-6f455d75d5-gqftv         1/1      Running   0
     9m9s
prod-fss-deviationmgr-platform-589dd98c4f-l67dp        1/1      Running   0
     9m10s
prod-fss-deviationmgr-qos-567cbfc7ff-twblt             1/1      Running   0
     9m6s
prod-fss-deviationmgr-routingpolicy-7cbcfc858b-rxf7p   1/1      Running   0
     9m10s
prod-fss-deviationmgr-system-6549bc48ff-2qttp          1/1      Running   0
     9m10s
prod-fss-dhcp-c595df9fc-lx4ln                          1/1      Running   0
     9m10s
prod-fss-filemgr-687fc88b4d-hc8bz                      1/1      Running   0
     9m8s
prod-fss-imagemgr-7555dbbbdc-qb5ns                     1/1      Running   0
     9m8s
prod-fss-intentmgr-78fcdf84df-fzmzm                    1/1      Running   0
     9m7s
prod-fss-inventory-58f8579d6f-whbmf                    1/1      Running   0
     9m6s
prod-fss-labelmgr-7db67c68cf-482kw                     1/1      Running   0
     9m10s
prod-fss-maintmgr-76d755c8cc-m6njl                     1/1      Running   0
     9m7s
prod-fss-mgmtstack-68b8fc9fdf-6xt7f                    1/1      Running   0
     9m10s
prod-fss-oper-da-6556b58fc5-29tw6                      1/1      Running   0
     9m10s
```

```
prod-fss-oper-da-6556b58fc5-789nx                    1/1      Running   0
        9m10s
prod-fss-oper-da-6556b58fc5-bpnvg                    1/1      Running   0
        9m9s
prod-fss-oper-da-6556b58fc5-bvvkc                    1/1      Running   0
        9m9s
prod-fss-oper-da-6556b58fc5-k7xqz                    1/1      Running   0
        9m10s
prod-fss-oper-da-6556b58fc5-rlqlv                    1/1      Running   0
        9m9s
prod-fss-oper-da-6556b58fc5-wn2bb                    1/1      Running   0
        9m9s
prod-fss-oper-da-6556b58fc5-zvkx2                    1/1      Running   0
        9m9s
prod-fss-oper-topomgr-94b897c4f-r9tjt                1/1      Running   0
        9m5s
prod-fss-topomgr-77cd566695-5w5mx                    1/1      Running   0
        9m8s
prod-fss-transaction-5756574f99-clfv5                1/1      Running   0
        9m6s
prod-fss-version-b565f896-kcktt                      1/1      Running   0
        9m10s
prod-fss-web-5d5bbf4c67-gd6bf                         1/1      Running   0
        9m9s
prod-fss-workloadmgr-c99db6bb6-fft5q                 1/1      Running   0
        9m10s
prod-fss-ztp-59675c6ffd-pzpxj                        1/1      Running   0
        9m10s
prod-keycloak-0                                      1/1      Running   0
        9m10s
prod-mongodb-arbiter-0                               1/1      Running   0
        9m10s
prod-mongodb-primary-0                               1/1      Running   0
        9m10s
prod-mongodb-secondary-0                             1/1      Running   0
        9m10s
prod-neo4j-core-0                                    1/1      Running   0
        9m10s
prod-postgresql-0                                    1/1      Running   0
        9m10s
[root@fss-deployer ~]#
```

Step 2. Back up the necessary files from 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`

- Input JSON configuration file that was used to install the Fabric Services System, for example:

  `/root/customer-input.json`

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

  `/root/tls.crt`

  `/root/tls.key`

- Optional files

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. Shut down the old deployer VM.

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

Step 5. 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 6. 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 ~]# ./bin/fss-upgrade.sh
Usage: ./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.

Prerequisites

Ensure that you have completed the steps described in Preparing for software upgrade. You execute the steps in this procedure 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 ~]# ./bin/fss-upgrade.sh configure customer-input.json
    Timesync service is running on 192.168.59.195  Time difference is 0 seconds
    Timesync service is running on 192.168.59.129  Time difference is 0 seconds
    Timesync service is running on 192.168.59.127  Time difference is 0 seconds
  Maximum time difference between nodes 0 seconds
```

```
Successfully configured. Please run /root/bin/fss-upgrade.sh upgrade
```

> **Note:** Do not execute the **fss-upgrade.sh upgrade** command as indicated in the output. Continue with Step 2.

Step 2.   Discover the cluster.

Example

```
[root@fss-deployer ~]# ./bin/fss-upgrade.sh discover
Trying to discover 192.168.59.195
Trying to discover 192.168.59.129
Trying to discover 192.168.59.127
NAME                      STATUS   ROLES                 AGE   VERSION
fss-compute-customer-01   Ready    control-plane,master  12h   v1.23.1
fss-compute-customer-02   Ready    control-plane,master  12h   v1.23.1
fss-compute-customer-03   Ready    control-plane,master  12h   v1.23.1
FSS release discovered fss-FSS_22_8_B1-charts-v22.8.1-108
```

Step 3.   Upgrade the Fabric Services System application.

Example

```
[root@fss-deployer ~]# /root/bin/fss-upgrade.sh upgrade
NAME                      STATUS   ROLES                 AGE   VERSION
fss-compute-customer-01   Ready    control-plane,master  12h   v1.23.1
fss-compute-customer-02   Ready    control-plane,master  12h   v1.23.1
fss-compute-customer-03   Ready    control-plane,master  12h   v1.23.1
FSS will be upgraded from fss-FSS_22_8_B1-charts-v22.8.1-108 to fss-FSS_22_8_B1-charts
 v22.8.2-6 : Are you sure [YyNn]? y
Upgrade in progress...
W1027 18:06:09.880444    7001 warnings.go:67] policy/v1beta1 PodDisruptionBudget is
 deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W1027 18:06:09.888237    7001 warnings.go:67] policy/v1beta1 PodDisruptionBudget is
 deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W1027 18:06:09.905507    7001 warnings.go:67] policy/v1beta1 PodDisruptionBudget is
 deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W1027 18:06:09.910756    7001 warnings.go:67] policy/v1beta1 PodDisruptionBudget is
 deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W1027 18:06:09.918327    7001 warnings.go:67] policy/v1beta1 PodDisruptionBudget is
 deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W1027 18:06:09.927503    7001 warnings.go:67] policy/v1beta1 PodDisruptionBudget is
 deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W1027 18:06:09.932171    7001 warnings.go:67] policy/v1beta1 PodDisruptionBudget is
 deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W1027 18:06:09.938165    7001 warnings.go:67] policy/v1beta1 PodDisruptionBudget is
 deprecated in v1.21+, unavailable in v1.25+; use policy/v1 PodDisruptionBudget
W1027 18:06:09.945057    7001 warnings.go:67] 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: Thu Oct 27 18:06:07 2022
NAMESPACE: default
STATUS: deployed
REVISION: 2
NOTES:
[root@fss-deployer ~]#
```

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 ~]# ./bin/fss-upgrade.sh status
NAME                     STATUS   ROLES                  AGE    VERSION
fss-compute-customer-01  Ready    control-plane,master   12h    v1.23.1
fss-compute-customer-02  Ready    control-plane,master   12h    v1.23.1
fss-compute-customer-03  Ready    control-plane,master   12h    v1.23.1
FSS release discovered: fss-FSS_22_8_B1-charts-v22.8.2-6 ; newer release: fss-FSS_
22_8_B1-charts v22.8.2-6
```

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                                APP VERSION
prod    default         2               2022-10-27 18:06:07.835241335 +0000 UTC
 deployed        fss-FSS_22_8_B1-charts-v22.8.2-6        1.0
```

c. Check the status of system services and processes.

Example

The upgrade is in progress until the STATUS field for all pods shows Running.

```
[root@fss-deployer ~]# kubectl get pods
NAME                                             READY   STATUS
 RESTARTS       AGE
prod-cp-kafka-0                                  2/2     Running     2 (11m
 ago)   13m
prod-cp-kafka-1                                  2/2     Running     1 (11m
 ago)   12m
prod-cp-kafka-2                                  2/2     Running     0
        12m
prod-cp-zookeeper-0                              2/2     Running     0
        13m
prod-cp-zookeeper-1                              2/2     Running     0
        11m
prod-cp-zookeeper-2                              2/2     Running     0
        11m
prod-fss-alarmmgr-6678f4d49c-xz77v               1/1     Terminating 0
        13m
prod-fss-auth-6875676df7-9b74r                   1/1     Running     0
        33s
prod-fss-auth-79ff5f495c-s4t7m                   0/1     Terminating 0
        13m
prod-fss-catalog-7c5478d8ff-t546z                0/1     Init:0/1    0
        27s
prod-fss-cfggen-5d8f94f74b-57nd7                 1/1     Running     0
        13m
prod-fss-cfggen-6f5958bdf4-m6s54                 0/1     Init:0/1    0
        33s
prod-fss-cfgsync-5b69b7f854-l9lx5                1/1     Running     0
        13m
prod-fss-cfgsync-6889dfc69d-t99mg                0/1     Init:0/1    0
        33s
prod-fss-connect-69d8d49799-96w7k                0/1     Init:0/1    0
        33s
```

```
prod-fss-connect-75df8ff7f9-vwfgj                        1/1    Running       0
        13m
prod-fss-da-69f8676b6d-4kjsn                             0/1    Terminating   0
        13m
prod-fss-da-69f8676b6d-cp5v6                             1/1    Terminating   0
        13m
prod-fss-da-69f8676b6d-fj7db                             1/1    Running       0
        13m
prod-fss-da-69f8676b6d-mnz9b                             1/1    Running       0
        13m
prod-fss-da-69f8676b6d-ph42r                             1/1    Running       0
        13m
prod-fss-da-69f8676b6d-vgflj                             0/1    Terminating   0
        13m
prod-fss-da-69f8676b6d-zg2pb                             1/1    Running       0
        13m
prod-fss-da-69f8676b6d-zt7z6                             1/1    Terminating   0
        13m
prod-fss-da-bf8f8d456-6pmlr                              0/1    Init:0/1      0
        33s
prod-fss-da-bf8f8d456-h9z6j                              1/1    Running       0
        33s
prod-fss-da-bf8f8d456-l9fjh                              0/1    Init:0/1      0
        17s
prod-fss-da-bf8f8d456-lqzmn                              1/1    Running       0
        32s
prod-fss-da-bf8f8d456-ptd7t                              0/1    Init:0/1      0
        32s
prod-fss-da-bf8f8d456-tghht                              0/1    Init:0/1      0
        22s
prod-fss-deviationmgr-acl-6c88f46886-pl4bm              0/1    Init:0/1      0
        33s
prod-fss-deviationmgr-acl-d8949f794-2dl79              1/1    Running       0
        13m
prod-fss-deviationmgr-bfd-68f5d7988b-q8vks             1/1    Running       0
        13m
prod-fss-deviationmgr-bfd-79bff7cd9d-5qst2             0/1    Init:0/1      0
        33s
prod-fss-deviationmgr-interface-769c5d46f-l9wrc        1/1    Running       0
        13m
prod-fss-deviationmgr-interface-77574648bf-zhrwk       0/1    Init:0/1      0
        32s
prod-fss-deviationmgr-netinst-6f455d75d5-gqftv         1/1    Running       0
        13m
prod-fss-deviationmgr-netinst-79d9484657-jsp67         0/1    Init:0/1      0
        32s
prod-fss-deviationmgr-platform-589dd98c4f-l67dp        1/1    Running       0
        13m
prod-fss-deviationmgr-platform-868fc5fdc7-z7xzk        0/1    Init:0/1      0
        32s
prod-fss-deviationmgr-qos-567cbfc7ff-twblt            1/1    Running       0
        13m
prod-fss-deviationmgr-qos-6ffcfcbff5-2f2mj            0/1    Init:0/1      0
        32s
prod-fss-deviationmgr-routingpolicy-7cbcfc858b-rxf7p   1/1    Running       0
        13m
prod-fss-deviationmgr-routingpolicy-c9c5d56db-5zm6r    0/1    Init:0/1      0
        31s
prod-fss-deviationmgr-system-6549bc48ff-2qttp         1/1    Running       0
        13m
prod-fss-deviationmgr-system-6cd48bdb67-42lb8         0/1    Init:0/1      0
        31s
prod-fss-dhcp-c595df9fc-lx4ln                           1/1    Running       0
        13m
```

```
prod-fss-filemgr-687fc88b4d-hc8bz                    1/1    Running       0
        13m
prod-fss-filemgr-7546d4945b-xhs2h                    0/1    Init:0/1      0
        31s
prod-fss-imagemgr-5bbddc7995-fj5vm                   0/1    Init:0/1      0
        19s
prod-fss-intentmgr-78fcdf84df-fzmzm                  0/1    Terminating   0
        13m
prod-fss-inventory-7d5544bf6f-d6cm5                  0/1    Init:0/1      0
        21s
prod-fss-labelmgr-7db67c68cf-482kw                   1/1    Running       0
        13m
prod-fss-labelmgr-bdd847459-b5qkm                    0/1    Init:0/1      0
        30s
prod-fss-maintmgr-76d755c8cc-m6njl                   1/1    Terminating   0
        13m
prod-fss-mgmtstack-68b8fc9fdf-6xt7f                  1/1    Running       0
        13m
prod-fss-mgmtstack-7f67b85c97-hj6rd                  0/1    Init:0/1      0
        30s
prod-fss-oper-da-6556b58fc5-29tw6                    1/1    Running       0
        13m
prod-fss-oper-da-6556b58fc5-789nx                    0/1    Terminating   0
        13m
prod-fss-oper-da-6556b58fc5-bpnvg                    1/1    Running       0
        13m
prod-fss-oper-da-6556b58fc5-bvvkc                    1/1    Running       0
        13m
prod-fss-oper-da-6556b58fc5-k7xqz                    1/1    Running       0
        13m
prod-fss-oper-da-6556b58fc5-rlqlv                    0/1    Terminating   0
        13m
prod-fss-oper-da-6556b58fc5-wn2bb                    1/1    Running       0
        13m
prod-fss-oper-da-6556b58fc5-zvkx2                    1/1    Running       0
        13m
prod-fss-oper-da-7cc49477f-fglt4                     0/1    Init:0/1      0
        29s
prod-fss-oper-da-7cc49477f-lmghk                     0/1    Init:0/1      0
        30s
prod-fss-oper-da-7cc49477f-pmkkc                     0/1    Init:0/1      0
        28s
prod-fss-oper-da-7cc49477f-wprbk                     0/1    Init:0/1      0
        28s
prod-fss-oper-topomgr-665bdb7bcc-hwgnm               0/1    Pending       0
        3s
prod-fss-topomgr-7756dbc5dc-pnfhp                    0/1    Init:0/1      0
        29s
prod-fss-topomgr-77cd566695-5w5mx                    1/1    Running       0
        13m
prod-fss-transaction-5756574f99-clfv5                1/1    Terminating   0
        13m
prod-fss-version-b565f896-kcktt                      0/1    Terminating   0
        13m
prod-fss-web-5d5bbf4c67-gd6bf                        1/1    Running       0
        13m
prod-fss-web-7967d75595-x7dpb                        0/1    Init:0/1      0
        29s
prod-fss-workloadmgr-77f49c9dbb-gbz74                0/1    Init:0/1      0
        29s
prod-fss-workloadmgr-c99db6bb6-fft5q                 1/1    Running       0
        13m
prod-fss-ztp-59675c6ffd-pzpxj                        1/1    Running       0
        13m
```

```
prod-fss-ztp-644d8c689d-cdn5s                        0/1      Init:0/1         0
        29s
prod-keycloak-0                                      1/1      Running          0
        13m
prod-mongodb-arbiter-0                               1/1      Running          0
        13m
prod-mongodb-primary-0                               1/1      Running          0
        13m
prod-mongodb-secondary-0                             1/1      Running          0
        13m
prod-neo4j-core-0                                    1/1      Running          0
        13m
prod-postgresql-0                                    1/1      Running          0
        13m
[root@fss-deployer ~]#
```

Expected outcome

```
[root@fss-deployer ~]# kubectl get pods
NAME                                                 READY    STATUS     RESTARTS
    AGE
prod-cp-kafka-0                                      2/2      Running    2 (14m ago)
    16m
prod-cp-kafka-1                                      2/2      Running    1 (14m ago)
    15m
prod-cp-kafka-2                                      2/2      Running    0
    15m
prod-cp-zookeeper-0                                  2/2      Running    0
    16m
prod-cp-zookeeper-1                                  2/2      Running    0
    14m
prod-cp-zookeeper-2                                  2/2      Running    0
    14m
prod-fss-alarmmgr-5dd8bb8b4f-h6rtb                   1/1      Running    0
    2m52s
prod-fss-auth-6875676df7-9b74r                       1/1      Running    0
    3m42s
prod-fss-catalog-7c5478d8ff-t546z                    1/1      Running    0
    3m36s
prod-fss-cfggen-6f5958bdf4-m6s54                     1/1      Running    0
    3m42s
prod-fss-cfgsync-6889dfc69d-t99mg                    1/1      Running    0
    3m42s
prod-fss-connect-69d8d49799-96w7k                    1/1      Running    0
    3m42s
prod-fss-da-bf8f8d456-6pmlr                          1/1      Running    0
    3m42s
prod-fss-da-bf8f8d456-h9z6j                          1/1      Running    0
    3m42s
prod-fss-da-bf8f8d456-l9fjh                          1/1      Running    0
    3m26s
prod-fss-da-bf8f8d456-lqzmn                          1/1      Running    0
    3m41s
prod-fss-da-bf8f8d456-nqbnz                          1/1      Running    0
    75s
prod-fss-da-bf8f8d456-ptd7t                          1/1      Running    0
    3m41s
prod-fss-da-bf8f8d456-tghht                          1/1      Running    0
    3m31s
prod-fss-da-bf8f8d456-xn5s5                          1/1      Running    0
    83s
prod-fss-deviationmgr-acl-6c88f46886-pl4bm           1/1      Running    0
    3m42s
```

```
prod-fss-deviationmgr-bfd-79bff7cd9d-5qst2              1/1    Running   0
    3m42s
prod-fss-deviationmgr-interface-77574648bf-zhrwk        1/1    Running   0
    3m41s
prod-fss-deviationmgr-netinst-79d9484657-jsp67          1/1    Running   0
    3m41s
prod-fss-deviationmgr-platform-868fc5fdc7-z7xzk         1/1    Running   0
    3m41s
prod-fss-deviationmgr-qos-6ffcfcbff5-2f2mj              1/1    Running   0
    3m41s
prod-fss-deviationmgr-routingpolicy-c9c5d56db-5zm6r     1/1    Running   0
    3m40s
prod-fss-deviationmgr-system-6cd48bdb67-42lb8           1/1    Running   0
    3m40s
prod-fss-dhcp-c595df9fc-lx4ln                           1/1    Running   0
    16m
prod-fss-filemgr-7546d4945b-xhs2h                       1/1    Running   0
    3m40s
prod-fss-imagemgr-5bbddc7995-fj5vm                      1/1    Running   0
    3m28s
prod-fss-intentmgr-5c55d858d9-62cc6                     1/1    Running   0
    2m55s
prod-fss-inventory-7d5544bf6f-d6cm5                     1/1    Running   0
    3m30s
prod-fss-labelmgr-bdd847459-b5qkm                       1/1    Running   0
    3m39s
prod-fss-maintmgr-d95457cdf-45vtx                       1/1    Running   0
    2m39s
prod-fss-mgmtstack-7f67b85c97-hj6rd                     1/1    Running   0
    3m39s
prod-fss-oper-da-7cc49477f-ds2hk                        1/1    Running   0
    80s
prod-fss-oper-da-7cc49477f-fglt4                        1/1    Running   0
    3m38s
prod-fss-oper-da-7cc49477f-lmghk                        1/1    Running   0
    3m39s
prod-fss-oper-da-7cc49477f-mp8cz                        1/1    Running   0
    61s
prod-fss-oper-da-7cc49477f-pmkkc                        1/1    Running   0
    3m37s
prod-fss-oper-da-7cc49477f-rl2f8                        1/1    Running   0
    72s
prod-fss-oper-da-7cc49477f-vffjz                        1/1    Running   0
    72s
prod-fss-oper-da-7cc49477f-wprbk                        1/1    Running   0
    3m37s
prod-fss-oper-topomgr-665bdb7bcc-hwgnm                  1/1    Running   0
    3m12s
prod-fss-topomgr-7756dbc5dc-pnfhp                       1/1    Running   0
    3m38s
prod-fss-transaction-66fc8f4d75-bjxdp                   1/1    Running   0
    2m54s
prod-fss-version-5bc8c875f6-5cpb8                       1/1    Running   0
    2m53s
prod-fss-web-7967d75595-x7dpb                           1/1    Running   0
    3m38s
prod-fss-workloadmgr-77f49c9dbb-gbz74                   1/1    Running   0
    3m38s
prod-fss-ztp-644d8c689d-cdn5s                           1/1    Running   0
    3m38s
prod-keycloak-0                                         1/1    Running   0
    16m
prod-mongodb-arbiter-0                                  1/1    Running   0
    16m
```

```
prod-mongodb-primary-0                                    1/1      Running   0
    16m
prod-mongodb-secondary-0                                  1/1      Running   0
    16m
prod-neo4j-core-0                                         1/1      Running   0
    16m
prod-postgresql-0                                         1/1      Running   0
    16m
[root@fss-deployer ~]#
```

Step 5. The upgrade procedure replaces the software catalog. Changes that you made to the software catalog will be lost. If you have added new SRL images in the software catalog, you need to modify the software catalog and then upload the SRL images.

For instructions, see *Updating software catalog to include new versions* in the *Fabric Services System User Guide*. You need to complete this step before you access the Fabric Services System for any changes.

8.3 Performing a software rollback

About this task

If a software upgrade is unsuccessful, use the following steps to rollback 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 Release 22.8.1.

For instructions, see 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 Release 22.8.1.

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