nSSV Upgrade Guide

Contents

1 Upgrade Overview

This document describes the maintenance and upgrade procedure for the nSSV platform.

The change covers the cloud management layer as well as the virtualization components (QEMU and libvirt). All operations are designed so that no impact is introduced to running production workloads.

1.1 Configuration Information

Environment:

- Dual or Single Management Node (x86)

1.2 Change Principles and Scope

The upgrade procedure:

- Does not modify business workloads.
- Only affects the cloud management platform and virtualization components.
- Does not introduce architectural changes.

2 Upgrade Procedure

2.1 Platform Upgrade Check

Before upgrading, review the platform conditions and perform several validation tasks.

General Checks

```
zstack-ctl status
```

```
cat /etc/*-release
```

```
uname -a
```

```
rpm -qa | grep -i qemu
```

```
rpm -qa | grep -i glib2
```

```
rpm -qa | grep -i libvirt
```

```
zsha2 status
```

Pre-upgrade Validation Tools

Before upgrading, use the self-check packages to inspect the overall health status of the environment.

1. Environment Health Check (pre_upgrade_self_check)

Use the `pre_upgrade_self_check` package to verify basic environmental information and identify potential upgrade risks.

Unzip the inspection package:

```
tar zxvf pre_upgrade_self_check.tar.gz
```

Enter the package directory:

```
cd pre_upgrade_self_check
```

Run the upgrade check script. Replace **[MN_admin_password]** with the management node administrator password:

```
bash exec_tool.sh -p [MN_admin_password]
```

Review the inspection results and collect the log file for internal verification. The `check.log` file contains environment status information and upgrade risk checks:

```
cat ./log/check.log
```

Verify the running status of virtual machine instances and confirm that host resources are healthy.

2. Master and Slave Node Consistency Check (zsmap)

Use the `zsmap-v2.3` tool to verify configuration consistency between the master and slave management nodes.

Unzip the package:

```
tar -zxvf zsmap-v2.3.tar.gz
```

Enter the package directory:

```
cd zsmap/
```

Run the consistency check script. Replace:

- **[MNIP]** with the management node IP address
- **admin / password** with the corresponding login credentials

```
bash check.sh [MNIP] admin password
```

Ensure that the database paths and XML configuration paths are consistent between the master and slave nodes.

2.2 Pre-Change Environment Handling

1. Disable global HA:
   - **Business Reliability** → **HA Policy**

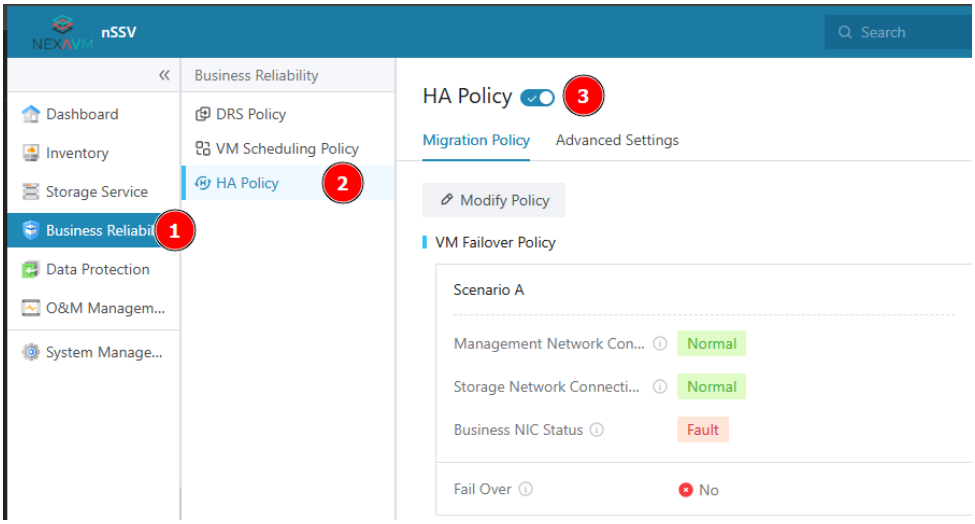

Figure 1: HA Policy

2. Prepare and upload all required upgrade packages.

   Before starting the upgrade, it is essential to ensure that all necessary installation files are available on the management nodes. The exact set of required packages depends on whether the platform is deployed with a **single management node** or a **dual management node (HA)** configuration.

   Single Management Node Deployment:

   For a standalone management node, upload the following packages to the server:

   - **nSSV ISO** of the target version (used to update the local repository)
   - **nSSV installer bin** of the target version

   These two components are sufficient to complete the upgrade procedure on a single-node environment.

   Dual Management Node Deployment (HA):

   For environments using dual management nodes, the following packages must be up-

loaded to **both management nodes**:

- **nSSV ISO** of the target version (to update the repository sources on each node)
- **nSSV installer bin** of the target version (used during the management service upgrade stage)
- **High-Availability Suite Package** (required to upgrade the HA framework before updating the management service)

Ensure that all files are transferred to each node and that their integrity is confirmed via MD5 checksum validation before proceeding.

3. Backup the database:
   - Dual MN:

     ```
     zstack-ctl dump_mysql --file-name zstack-db-backup-master
     ```

     ```
     zstack-ctl dump_mysql --file-name zstack-db-backup-slave
     ```

   - Single MN:

     ```
     zstack-ctl dump_mysql --file-name zstack-db-backup
     ```

4. Backup the upgrade script:

   ```
   cp /usr/local/bin/zstack-upgrade /root/zstack-upgrade-bk
   ```

2.3 Upgrade Execution

Choose the appropriate procedure depending on your management node topology.

Dual Management Node

1. Update source on both MNs:

   ```
   bash /root/zstack-upgrade -r nexavm-nssv-x86_64-dvd-[
      version_number]-h84r.iso
   ```

2. Check VIP node:

   ```
   zsha2 status
   ```

3. Extract HA suite on VIP node:

```
tar zxvf nexavm-nssv-multinode-ha-suite-[version_number].tar.gz
```

```
chmod +x zsha2 zstack-hamon
```

4. Upgrade HA:

```
./zsha2 upgrade-ha -gencfg=true
```

5. Stop MN services on both nodes:

```
zsha2 stop-node
```

6. Upgrade cloud platform on VIP node (the UI will be inaccessible during the time):

```
zsha2 upgrade-mn nexavm-nssv-installer-[version_number].bin
```

7. Check the status of the nodes and version after upgrading:

```
zstack-ctl status
```

```
zsha2 status
```

8. Log in to the web UI, check the status of inventory, storage service, clusters, etc.
9. Select physical hosts running non-critical business for priority UI manual reconnection. After a successful reconnection, check the kvmagent status and the running/business status of the VM instance. At this time, the upgrade status should display “Upgraded.”
10. After the physical host reconnects successfully, check the specific storage according to the recorded data.
11. Check the VM status, log in to the console, and use ping to verify whether the network is normal. Ensure that business operations are not affected.

Single Management Node

1. Upgrade repository and platform:

```
zstack-upgrade -r nexavm-nssv-x86_64-dvd-[version_number]-h84r.
   iso
```

```
bash nexavm-nssv-installer-[version_number].bin -u
```

2. Validate upgrade:

```
zstack-ctl status
```

2.4 Platform Risk Assessment

- VM services remain unaffected during the upgrade.
- Perform the upgrade during off-peak hours.
- MN services will be temporarily unavailable.
- UI may require cache clearing due to framework updates.
- In case of unexpected issues, the process can be repeated or rolled back following the rollback plan.

3 Virtualization Component Upgrade

3.1 Detailed Upgrade Steps

1. Check current versions of virtualization components on the target host or cluster:

   ```
   rpm -qa | grep qemu
   ```

   ```
   rpm -qa | grep libvirt
   ```

   ```
   rpm -qa | grep glib2
   ```

2. Retrieve platform UUID and authenticate to the cloud platform:

   ```
   zstack-cli
   ```

   ```
   >>> LogInByAccount accountName=admin password=password
   ```

3. Upgrade components using `UpdateClusterOS` commands:

   Before upgrading the virtualization components, it is necessary to retrieve the identifiers of both the cluster and the individual hosts on which the upgrade will be performed.

   Retrieve the Cluster UUID: From the cloud platform interface, navigate to the Cluster details page and copy the **Cluster UUID**.

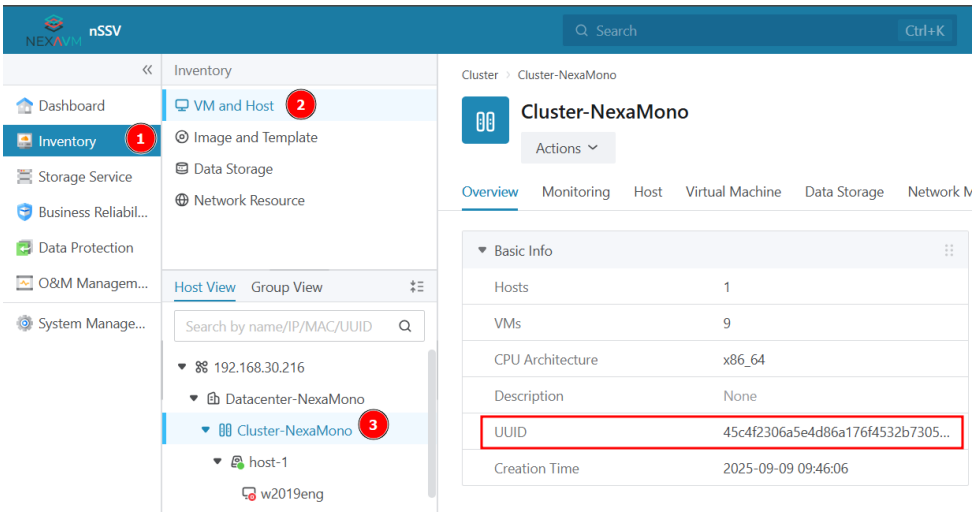

Figure 2: Cluster UUID Location

Retrieve the Host UUID: Open the details page of the target physical host and copy the **Host UUID**. This UUID is required only when more than one cluster is present in the environment. If there is only a single cluster, in the following commands it will be sufficient to specify the cluster UUID.

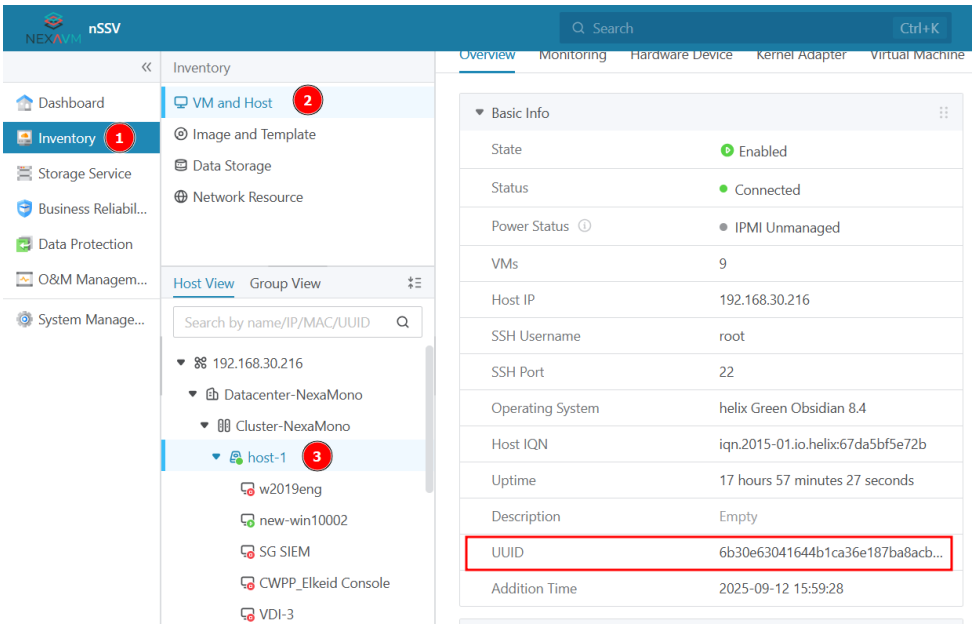

Figure 3: Host UUID Location

Once both values have been retrieved, proceed with the component upgrade commands shown below. In each command, replace:

- **[CLUSTER_UUID]** → with the actual Cluster UUID
- **[HOST_UUID]** → with the actual Host UUID of the node being upgraded

Upgrade qemu-storage-daemon on a specific host:

```
>>> UpdateClusterOS uuid=[CLUSTER_UUID] excludePackages=python2-
    crypto updatePackages=qemu-storage-daemon hostUuid=[HOST_UUID]
```

Upgrade qemu-kvm on a specific host:

```
>>> UpdateClusterOS uuid=[CLUSTER_UUID] excludePackages=python2-
    crypto updatePackages=qemu-kvm hostUuid=[HOST_UUID]
```

Upgrade glib2 on a specific host:

```
>>> UpdateClusterOS uuid=[CLUSTER_UUID] updatePackages=glib2
    hostUuid=[HOST_UUID]
```

Upgrade libvirt on a specific host:

```
>>> UpdateClusterOS uuid=[CLUSTER_UUID] excludePackages=python2-
    crypto updatePackages=libvirt hostUuid=[HOST_UUID]
```

Upgrade Scope Clarification:

The virtualization components can be upgraded either on:

- A **single physical host**, by specifying the `hostUuid` parameter
- An **entire cluster**, by executing the upgrade command without the `hostUuid` parameter

It is recommended to perform a single-host upgrade first to validate the process before proceeding with a cluster-wide upgrade.

Monitor Upgrade Task Execution:

The upgrade operation is executed asynchronously. Use the following command to monitor the execution status of the upgrade task:

```
>>> QueryLongJob uuid=[JOB_UUID]
```

Ensure that the job status is reported as **Succeeded** before continuing.

4. Validate upgraded versions after the upgrade task completes:

```
rpm -qa | grep qemu
```

```
rpm -qa | grep libvirt
```

```
rpm -qa | grep glib2
```

5. Migrate VMs to activate the new QEMU version.

   Important Operational Notice:

   The virtual machine migration operation introduces operational risks and must be performed with caution. It is strongly recommended to execute VM migration activities **only during off-peak business hours** and preferably on non-critical workloads.

   Ensure that cluster resource utilization is stable before proceeding.

   Before verifying the QEMU version of a running virtual machine, obtain the **VM UUID** from the VM details page in the management interface.

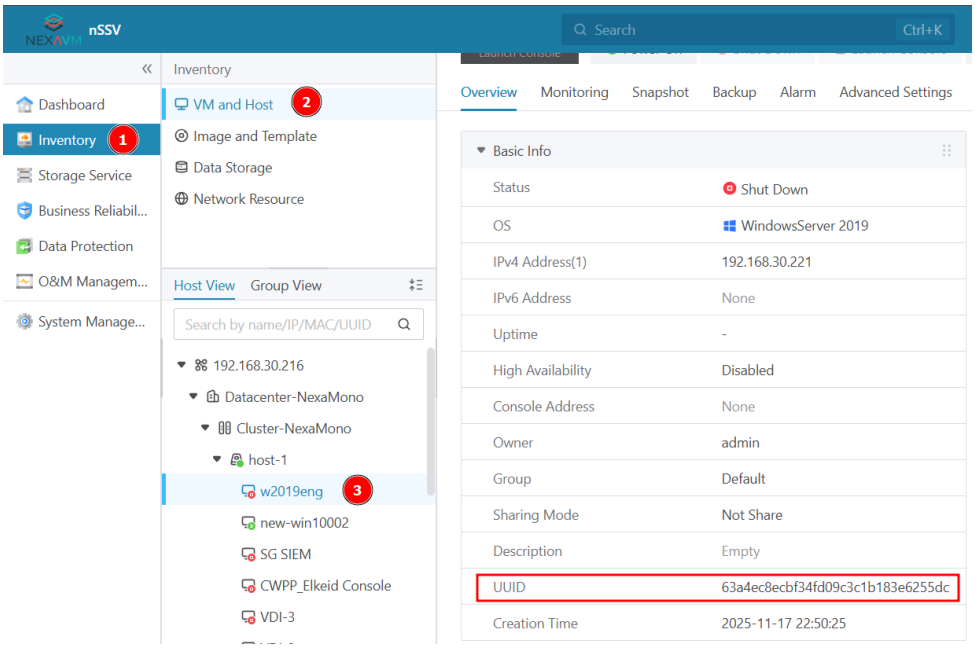

Figure 4: VM UUID Location

Once the VM UUID has been retrieved, run the following command on the physical host where the VM is currently running. Replace:

- **[VM_UUID]** → with the actual UUID of the virtual machine

```
virsh qemu-monitor-command [VM_UUID] --hmp info version
```

During the entire virtualization component upgrade process, running virtual machine workloads remain operational, and business services are not interrupted when best practices are followed.

6. Re-enable global HA.