nCSSV Upgrade Guide

Contents

1 Upgrade Overview 2
1.1 Change Principles and Scope 2
2 Upgrade Procedure 2
2.1 Platform Upgrade Check 2
2.2 Pre-Change Environment Handling 4
2.3 Upgrade Execution 6
2.4 Platform Risk Assessment 7
3 Virtualization Component Upgrade 8
3.1 Detailed Upgrade Steps 8
3.2 Component Upgrade Risk Assessment 11
4 Risk Evaluation 12
4.1 Risk Mitigation Plan 12
5 Rollback Plan 12
6 Emergency Response Plan 14

1 Upgrade Overview

This document describes the maintenance and upgrade procedure for the nCSSV 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 Change Principles and Scope

No changes will impact business operations. This procedure solely involves upgrading the cloud platform, updating QEMU and libvirt components.

2 Upgrade Procedure

2.1 Platform Upgrade Check

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

General Checks

Check the version of the management node:

```
zstack-ctl status
```

Check OS version:

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

```
nkvers
```

Check virtualization components information:

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

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

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

Check the status of the dual management node system:

```
zsha2 status
```

Upgrade Checklist

Version Change Item	Involved	Upgrade Adjustment Strategy
Cloud router fully upgraded to VPC network		
Tenant management account system replaces user groups and users		
Adjusted AD/LDAP integration and management method with accounts		
Qemu + Libvirt component upgrade	Yes	Upgrade via CLI
File backup tasks		
Heterogeneous cluster		
Mixed cluster of C72, C74, C76	No	Evaluate upgrade solution separately
Sharedblock storage	No	
Ceph storage	No	Disable multipath adjustment via CLI in advance
VM instance split-brain check	No	
VM instance qemu hot upgrade		
Check if database tables are too large	Yes	Pre-Check
HTTPS configuration	No	
SSH remote upgrade		Upgrade directly using non-timeout IPMI method
Historical API audit information retention		

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 package to verify basic environmental information and identify potential upgrade risks.

Unzip the inspection package:

```
tar zxvf upgrade-check-master.tar.gz
```

Enter the package directory and 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
```

Check the status of VM instances and the resource status of hosts.

2. Master and Slave Node Consistency Check (zsmap)

Use the `zsmap-v2.3` tool to verify consistency between the control plane and data plane.

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 and **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 high availability in the cloud platform's global settings:
    - **Business Reliability** → **HA Policy**

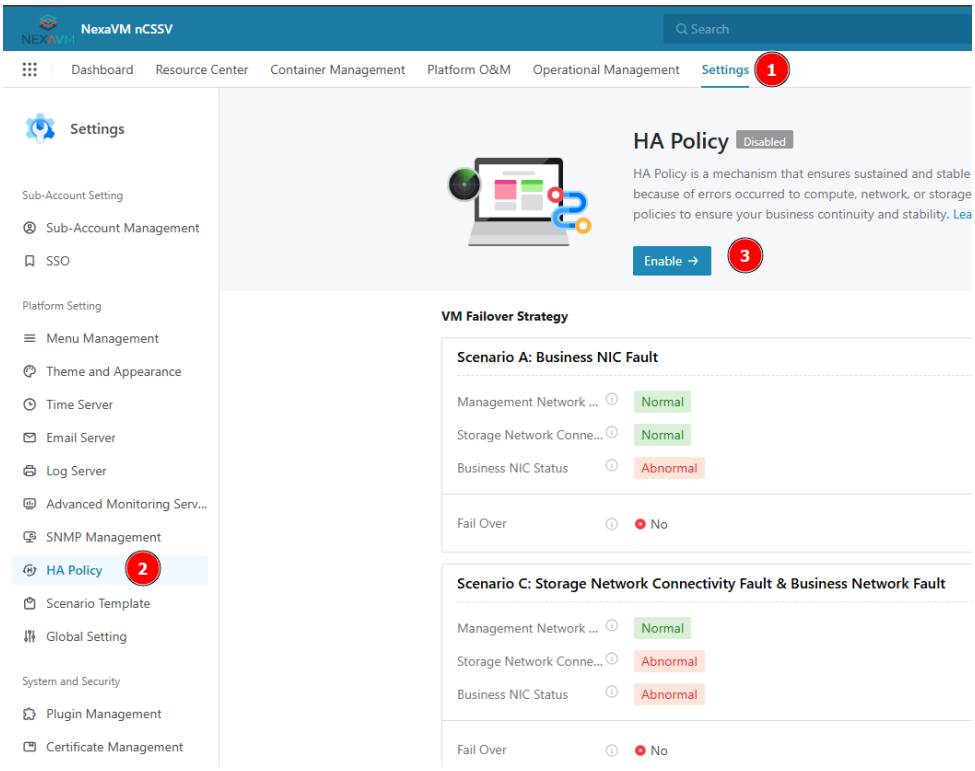

Figure 1: HA Policy

2. Upload the required images, upgrade packages, and high-availability components to both management nodes (or single management node). Upload the bin file for the high-availability suite, verify the commit version of the high-availability package, and ensure the MD5 checksums for all packages are correct.

   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.

3. Backup the database:

   - Dual MN (VIP node):

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

     Dual MN (Slave node):

     ```
     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 to the `/root/` directory (both management nodes):

```
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. Run the command on both management nodes and update the source:

```
bash /root/zstack-upgrade -r nCSSV[version].iso
```

2. Check the VIP node:

```
zsha2 status
```

3. Unzip the HA suite at the VIP node:

```
tar zxvf NexaVM-Multinode-HA-Suite.tar.gz
```

4. Grant execution privileges at the VIP node:

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

5. Upgrade the HA suite at the VIP node (the Cloud Platform UI will be inaccessible during the upgrade):

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

6. Run the command on both management nodes and stop the MN service:

```
zsha2 stop-node
```

7. Upgrade the cloud platform at the VIP node (the Cloud UI is inaccessible during this time):

```
zsha2 upgrade-mn Cloud-installer-[version].bin
```

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

```
zstack-ctl status
```

```
zsha2 status
```

9. Log in to the web UI and check the status of physical hosts, primary storage, imagestore, etc.
10. 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."**
11. After the physical host reconnects successfully, check the specific storage according to the recorded data.
12. 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 local repository first, then perform the upgrade:

```
zstack-upgrade -r nCSSV[version].iso
```

```
bash Cloud-installer-[version].bin -u
```

2. After the upgrade is complete, verify the status and check the version:

```
zstack-ctl status
```

2.4 Platform Risk Assessment

1. The platform upgrade process must not affect the operation of online VM services.
2. It is recommended that the upgrade be performed by the technical support team to avoid uncontrollable risks.
3. Schedule the platform upgrade during off-peak business hours to avoid resource application or business change requests during the upgrade.
4. During the upgrade process, management nodes cannot be accessed normally, and no new cloud hosts or cloud disk resources can be requested.
5. The management node database should be properly backed up in advance to ensure data security.
6. If any anomalies occur during the upgrade that cause interruptions, the upgrade can be re-executed as needed or rolled back according to the rollback plan.
7. Database operations must be backed up separately.
8. After the upgrade, the cloud platform may fail to connect to the VM management service. Users need to manually reconnect.
9. After the upgrade, due to UI framework updates, if the browser retains previous cache data, it may prevent the platform from being accessed via the domain name. Users need to manually clear the browser cache.

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. Log in to the cloud platform and record the platform UUID:

```
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**.

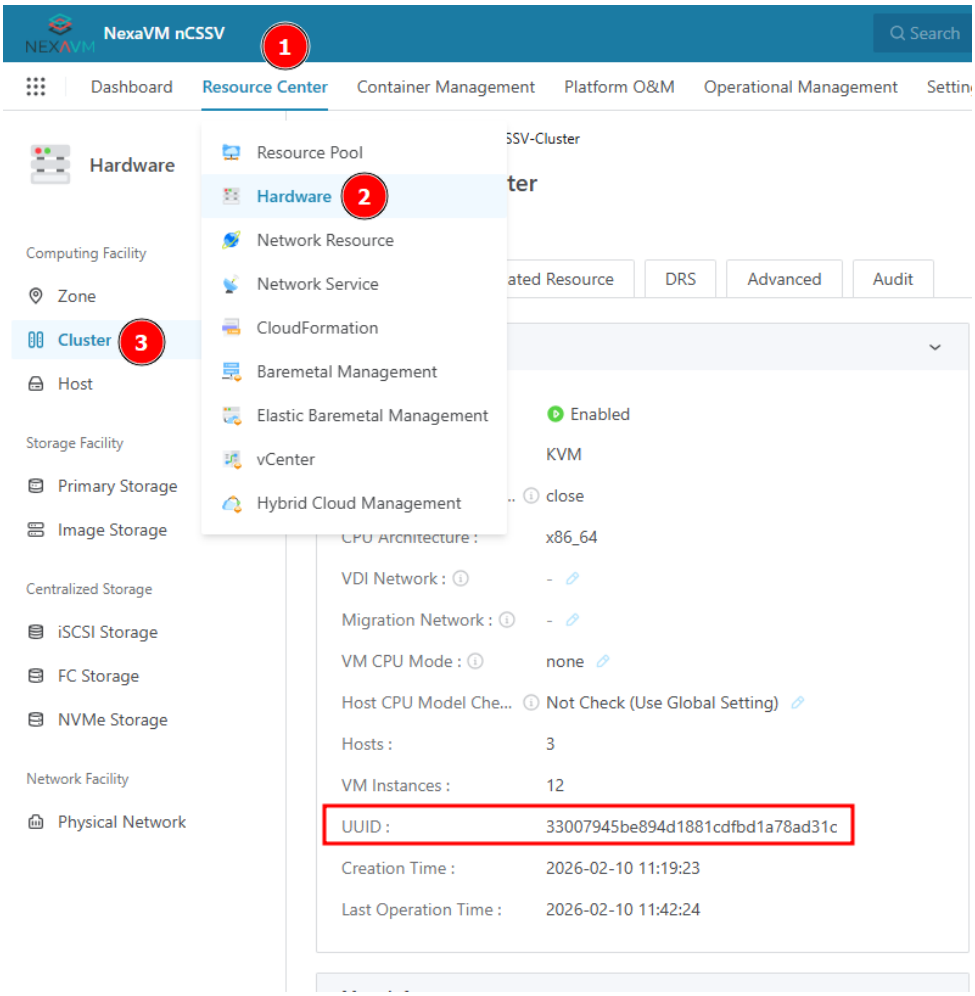

Figure 2: Cluster UUID Location

Retrieve the Host UUID: Open the details page of the target physical host and copy the **Host UUID**.

> **Note**
>
> `hostUuid` parameter in the following commands is required only when more than one cluster is present in the environment. If the environment contains a single cluster, the `hostUuid` parameter can be omitted from all commands below.

Once 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 virtualization components:

Upgrade `qemu-storage-daemon`:

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

Upgrade `qemu-kvm`:

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

Upgrade `glib2`:

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

Upgrade `libvirt`:

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

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. After the upgrade is complete, verify whether the cluster upgrade was successful:

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

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

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

5. After completing the qemu upgrade, migrate VM instances 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.

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

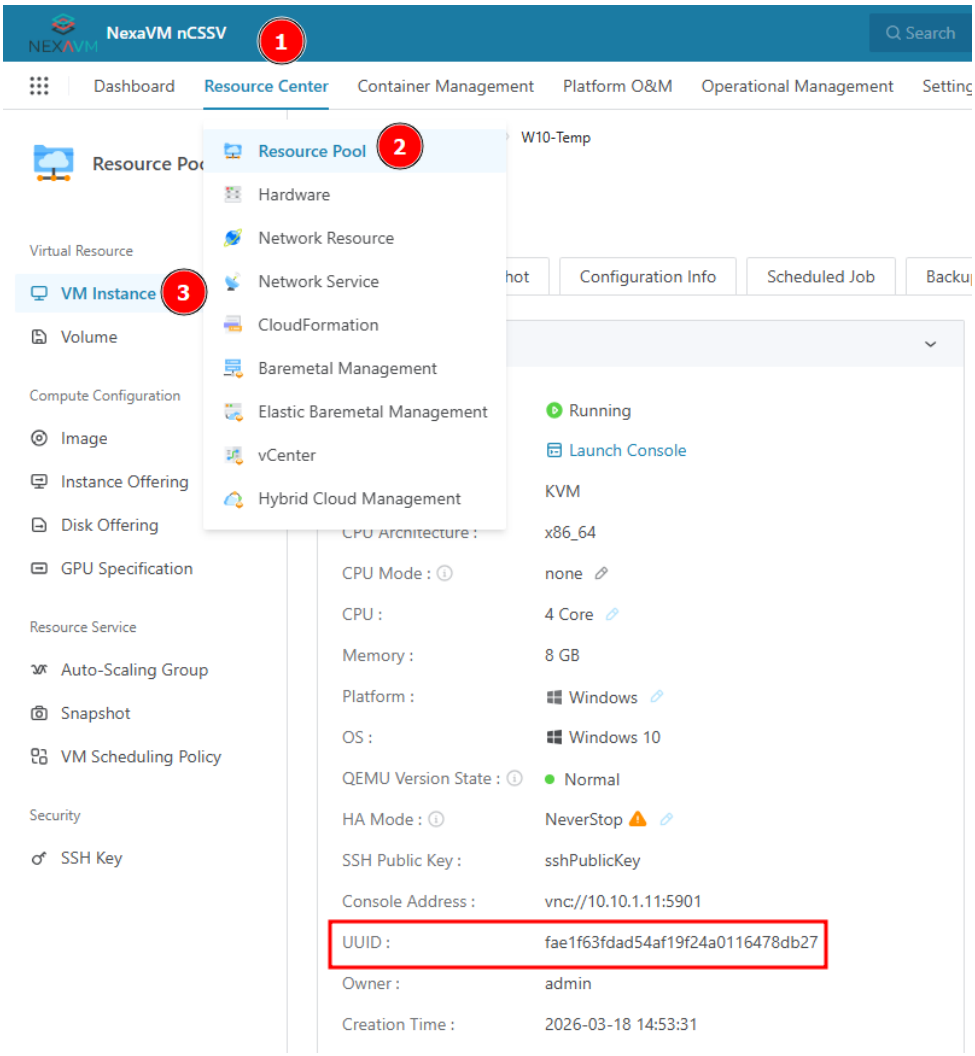

Figure 3: VM UUID Location

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.

3.2 Component Upgrade Risk Assessment

1. QEMU and libvirt upgrades cannot be rolled back.
2. After upgrading QEMU, perform a live migration or reboot the VM instance for the upgrade to take effect.

3. After upgrading libvirt, check the status of VM instances. For VMs marked as “no stat,” apply for a business downtime window and manually restart them using the `virsh` command.

4 Risk Evaluation

4.1 Risk Mitigation Plan

In response to the risks involved in this change, the following measures will be taken before executing the change to minimize potential impact:

1. Before performing the formal change operation, coordinate with the customer’s operation and maintenance personnel to agree on the change window period, ensuring that it is scheduled for the time with the least business impact.
2. Prior to the change, verify that the cloud platform and storage platform are in a healthy and normal state, and fully test the change plan in advance.

5 Rollback Plan

The rollback plan is used to quickly restore the system to its pre-failure state after an upgrade failure, ensuring service recovery and minimizing the impact of incidents.

1. On both management nodes, prepare a system ISO of the same version as before the platform upgrade, install upgrade packages, and ensure the management nodes are available.
2. Both management nodes should have separate database backups for recovery in case of emergencies.
3. On one of the management nodes, run the following commands to check which node the high-availability VIP is currently located on and obtain the clock source IP address of the platform:

   ```
   zsha2 status
   ```

   ```
   zstack-ctl show_configuration | grep chrony.serverIp
   ```

   The above commands will retrieve the IP address of the clock source used by the platform.
4. When planning to roll back the dual management node setup, first unbind the nodes to convert the dual-node configuration into a single-node one. Stop related services on both management nodes:

   ```
   zsha2 stop-node
   ```

   ```
   systemctl disable keepalived
   ```

```
systemctl disable zstack-ha
```

5. Modify the database mode separately on both management nodes:

```
mysql -uroot -pzstack.mysql.password zstack
```

```
MariaDB [zstack]> stop slave;
MariaDB [zstack]> reset slave all;
MariaDB [zstack]> show slave status;
```

6. On both management nodes, configure the database to point to the original IP address of the single management node. Replace **single_mn_ip** with the real IP address of the current management node:

```
zstack-ctl config_ui --db-url jdbc:mysql://single_mn_ip:3306
```

```
zstack-ctl configure DB.url=jdbc:mysql://single_mn_ip:3306
```

7. After completing the above steps, reinstall each node as an independent management node. Delete the existing ZStack cloud platform environment and run:

```
zstack-ctl stop
```

```
cp /usr/local/zstack/dnsmasq /root/
```

```
rm -rf /usr/local/zstack
```

Use the zstack-upgrade script to update to the original repo source:

```
bash /root/zstack-upgrade -r ZStack-Cloud-aarch64-DVD-4.4.9-
   kylinv10sp2.iso
```

```
mount ZStack-Cloud-aarch64-DVD-4.4.9-kylinv10sp2.iso /mnt/
```

```
cp /mnt/zstack-installer.bin /opt/zstack-installer.bin
```

```
bash /opt/zstack-installer.bin -D
```

8. After completing the above steps, stop management services on the VIP node and perform database recovery operations to restore the management node.

   Stop the management node:

   ```
   zstack-ctl stop
   ```

   Recover the database:

   ```
   zstack-ctl restore_mysql -f /var/lib/zstack/mysql-backup/mn1
      -2023-04-27_10-46-37.gz --mysql-root-password zstack.mysql.
      password
   ```

   Run the management node to restore the cloud platform as a single management node:

   ```
   zstack-ctl start
   ```

9. On the management node where the VIP resides, install the high-availability suite using the original zsha2 package:

   ```
   ./zsha2 install-ha -nic br_bond0 -slave root:password@172
      .31.250.5 -gateway 172.31.250.1 -vip 172.31.250.253 -db-root-
      pw zstack.mysql.password -time-server 172.31.250.6 -force
   ```

   Parameter reference:

   - **br_bond0**: The network interface name of the management node's management IP.
   - **root:password@172.31.250.5**: SSH login information of the slave management node.
   - **172.31.250.1**: The gateway of the management node.
   - **172.31.250.253**: The VIP (Virtual IP) address.
   - **zstack.mysql.password**: The root password of the database.
   - **172.31.250.6**: The IP address of the time synchronization source (retrieved in Step 3).

   Important: This operation will overwrite the database on the slave node. Do not execute it from any other slave node, or all data on that node will be cleared.

10. After completing the rollback operation, use port 5000 on the VIP node to log in to the platform and verify the connection status of each resource.
11. Verify that business operations are functioning normally.

6 Emergency Response Plan

If a major failure occurs during the change operation, the technical support team will immediately report the issue to the team lead. The team lead will convene within 15 minutes to coordinate technical experts to form an emergency response group.

The group will analyze the fault, assist in continuing the change process, or perform a rollback operation based on customer requirements.

After completing the rollback, technical experts will continue to analyze the root cause of the fault and submit a fault analysis report.