

# IBM Spectrum Virtualize Hot-Spare Node and NPIV Target Ports

Alex Ainscow

Fiona Crowther

Gareth Jones

Anil Palled

Graham Woodward

Jon Tate



**Storage**





## IBM Spectrum Virtualize: Hot-Spare Node and NPIV Target Ports

Historically, each physical Fibre Channel port on any IBM Spectrum Virtualize product has presented a single worldwide port name (WWPN) to the Fibre Channel fabric, and this port is capable of being used as a target for host I/O, an initiator for back-end controller I/O, and for internode and intercluster communications.

This multi-use port design is not ideal from the host point of view because an IBM Spectrum Virtualize node must message other nodes of the cluster before it knows its own configuration, including the volumes which are mapped to hosts. As such, it must bring online all Fibre Channel ports at software start up. The host therefore will log in and immediately request the list of volumes that are available. At this point, the IBM Spectrum Virtualize node is forced to delay the SCSI command until the cluster is configured.

If any failure of the node occurs, hardware or software, the port will be disabled while the recovery takes place. In the case of hardware failure, this might take several days at worst.

While a node is unavailable, the system is non-redundant, which will then affect other areas:

- ▶ Any additional hardware failures will lead to an outage.
- ▶ The performance of the system will be affected, partly due to the reduced hardware resources, but also the write cache must be disabled to provide the required levels of data protection.

The use of N\_Port ID Virtualization (NPIV) to provide host-only ports (NPIV target ports) and spare nodes improves the host failover characteristics by separating out host communications from communication tasks on the same port and providing standby hardware, which can be automatically introduced into the cluster to reintroduce redundancy.

Because the host ports are not used for internode communications, they can freely move between nodes, and this includes spare nodes that are added to the cluster automatically.

This IBM Redpaper™ publication describes the requirements for NPIV target ports for host communications, deployment of spare nodes, expected behavior, and how common maintenance procedures are affected by the deployment of hot-spare nodes. It covers both deployment of new systems, but also the introduction of these features into existing clusters.

## Terminology: Ports and nodes

The following terminology is used throughout this paper:

<b>NPIV target port</b>	A virtual port used only by hosts. This port is owned by a physical port, but can be moved between nodes.
<b>Primary port</b>	The WWPN associated with the physical hardware; this port cannot move.
<b>Spare node</b>	A node that is configured as a spare in the cluster.
<b>Original node</b>	A node that was replaced by a spare.

## Basic functionality

This section describes some of the basic functionality.

### Minimum IBM Spectrum Virtualize versions

The NPIV target port behavior is available in IBM Spectrum™ Virtualize V7.7.0 and later. Hot-spare nodes are available in IBM Spectrum Virtualize™ V8.1.0 and later.

### Spare nodes

In a cluster with spare nodes, each spare node is configured in the cluster with spare status. Unlike conventionally configured nodes, these nodes are not associated with any I/O groups. As such, under normal conditions, spare nodes do not participate in I/O services to hosts. The nodes that a spare is covering are listed in the *protected nodes* field in the node view.

Example 1 shows the use of the **lsnode** command to list the protected nodes.

*Example 1 Node view on lsnode*

---

```
IBM_2145:ITS0_DH8_LAB:superuser>lsnode spare_node
.lines removed for brevity
spare yes
failover_source
protected_nodes 1,2
IBM_2145:ITS0_DH8_LAB:superuser>
```

---

Figure 1 shows use of the GUI to view the protected node.

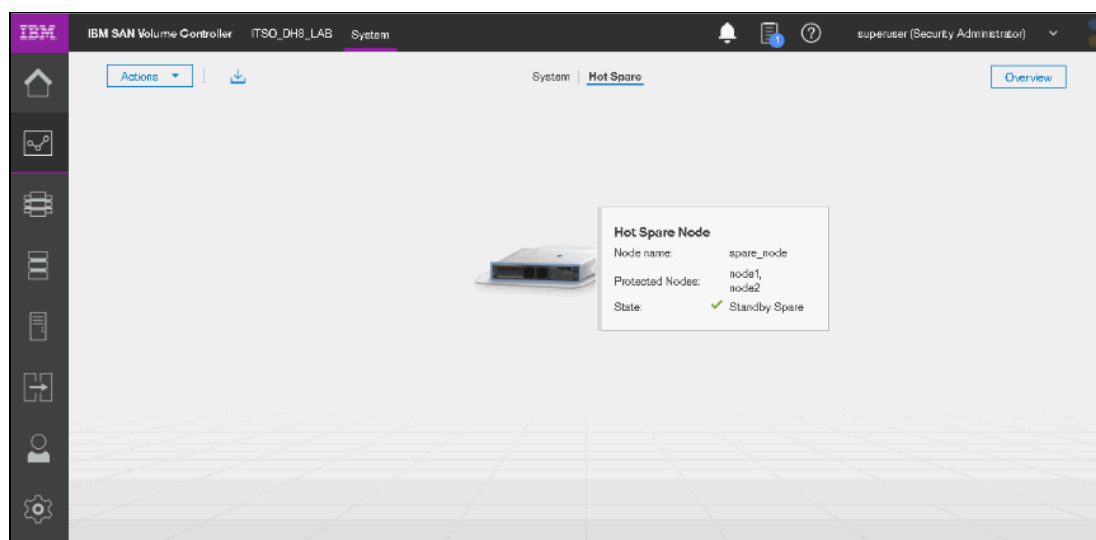


Figure 1 Node view showing protected node

A spare can be used as a temporary replacement for an original node. The spare is always only a temporary replacement with the expectation that the original node is repaired and used as the replacement node. When a spare is in use, its status is `online_spare`. The node for which a spare is online in place of is shown in the Protected Nodes field in the node view.

No mechanism is available for directly transitioning a spare node to an original node, other than by removing and re-adding that node to the I/O group.

## Failover

Where a node is offline and a suitable spare is available and online, a spare will be automatically configured in place of the offline node after a period of 1 - 5 minutes, depending on the reason for being offline.

A spare node will be considered as suitable if its hardware is compatible with the hardware of the failed node. This information is addressed in “Hardware compatibility” on page 9.

When a spare is online, its status becomes `online_spare`, indicating that this node is a temporary replacement for the original node.

A spare will never be brought online if the action will result in data loss. For example, if both nodes in an I/O group fail nearly simultaneously, the write cache will exist on only the offline nodes, so a spare cannot be brought online in these scenarios.

A failover can be initiated by the user manually by using the **swapnode** command.

## Failback

Failback occurs automatically when the original node returns and attempts to join the cluster. When failback occurs, the spare node is taken offline before the original node is permitted to rejoin the cluster.

The system never allows volume resources to be taken offline due to an original node returning to an online state, for example, if the I/O group containing the online spare is non-redundant, or has not yet completed a cache flush. In this scenario an event will be logged and a directed maintenance procedure (DMP) will guide the user through manually re-adding the original spare node.

**Note:** A returning node never changes position within an I/O group. This means that if an original node is replaced by an online spare, it cannot now become the partner node.

A failback can be forced by using the **swapnode** command. This is permitted even if the original node is not available.

## NPIV target ports

This section describes the characteristics of NPIV target ports.

### Modes

The NPIV target port feature can be enabled or disabled; three modes are available:

<b>Disabled</b>	No NPIV target ports are started and behavior is unchanged from a previous release.
<b>Transitional</b>	NPIV target ports are enabled. Volumes are presented through both physical and NPIV target ports.
<b>Enabled</b>	NPIV target ports are enabled. Volumes are presented through NPIV target ports only.

### Port behavior

In an NPIV enabled cluster, each physical port is associated with two WWPNs: the WWPN associated with the primary port and the WWPN associated with the NPIV target port.

The primary port will be logged into the fabric at all points that the IBM Spectrum Virtualize software is running, including in service mode. The WWPN of the primary port is equal to the WWPN of a primary port of a system with NPIV disabled and this value has not changed since previous releases of IBM Spectrum Virtualize.

The NPIV target port logs into the fabric when the IBM Spectrum Virtualize software is ready to begin processing I/O. It is presented to the fabric as *target only*, which means hosts may use it for I/O, but IBM Spectrum Virtualize cannot use this port to access back-end storage.

Figure 2 shows a standard topology.

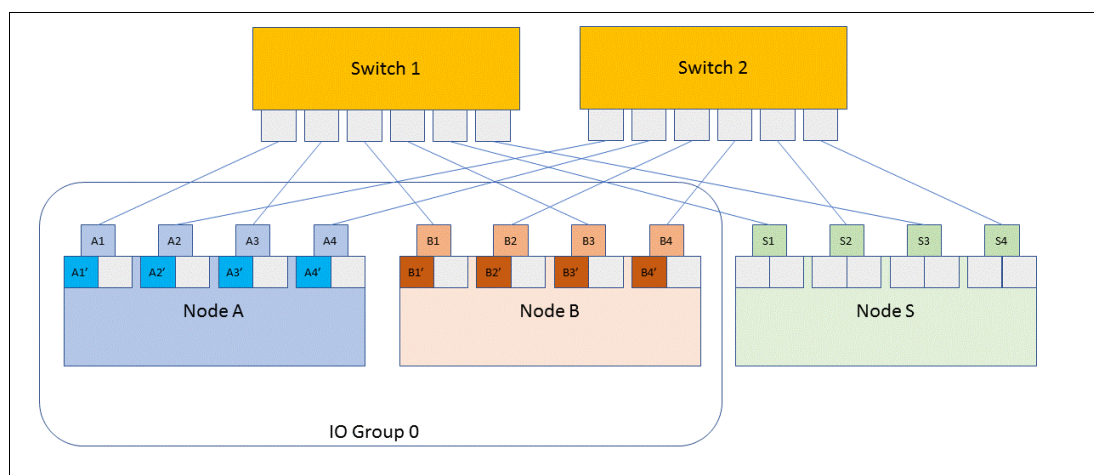


Figure 2 NPIV ports standard topology

The NPIV port is brought online on the first available node that is able to process I/O. The priority list is as follows:

1. The original node
2. A spare node replacing the original node
3. The partner node
4. A spare node replacing the original partner node

The delays that are associated with bringing a node online automatically mean that multiple failovers might occur. For example, if an original node suffers a hardware failure, the sequence will be as follows:

1. NPIV target ports failover to partner node
2. Five-minute delay
3. Spare brought into cluster
4. NPIV target ports failover to online spare node

Assuming the Fibre Channel network is correctly configured, this process is mostly transparent to the host.

## Failover behavior

When a node is offline, another node will attempt to bring online NPIV ports with the WWPNs of the missing ports. If a spare node is online in the cluster, that node will start the ports, otherwise it will be brought up on the partner node.

The port ID, as represented by the `lstargetportfc` command, of the ports does not change. This means that the NPIV target port that is associated with port ID 1 on the offline node will move to port ID 1 on the partner or spare node. In cases such as a software upgrade, where the cluster is deliberately taking a node offline, NPIV target ports will move between nodes prior to the node shutting down. This minimizes the time during which the port is offline, and typically the failover takes less than four seconds. In cases where a node suffers a sudden failure, the downtime might need to be longer to maintain data integrity.

When a node fails, the NPIV target port will initially move to the partner nodes (Figure 3 shows the initial failover). Note how the port IDs stay the same (B1 moves to port A1, B2 to port A2, and so on).

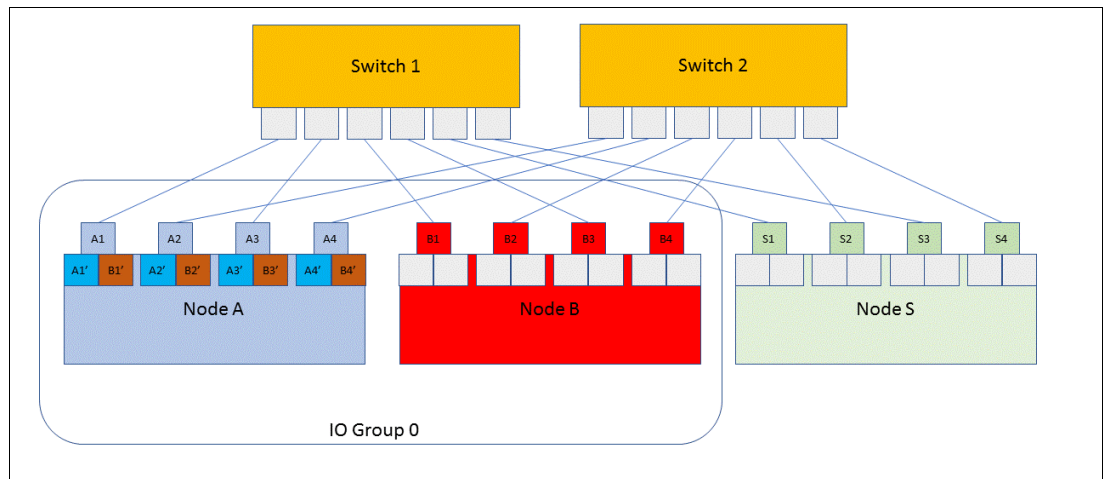


Figure 3 Node failure and target port move to the partner nodes

If a suitable spare node is available, that node will be brought online in place of the failed node. After that node is initialized as part of the cluster, the NPIV target ports will fail over to the spare online node, as Figure 4 shows. Note how the port ID stays the same again (port B1' moves to port S1, B2' to port S2 and so on).

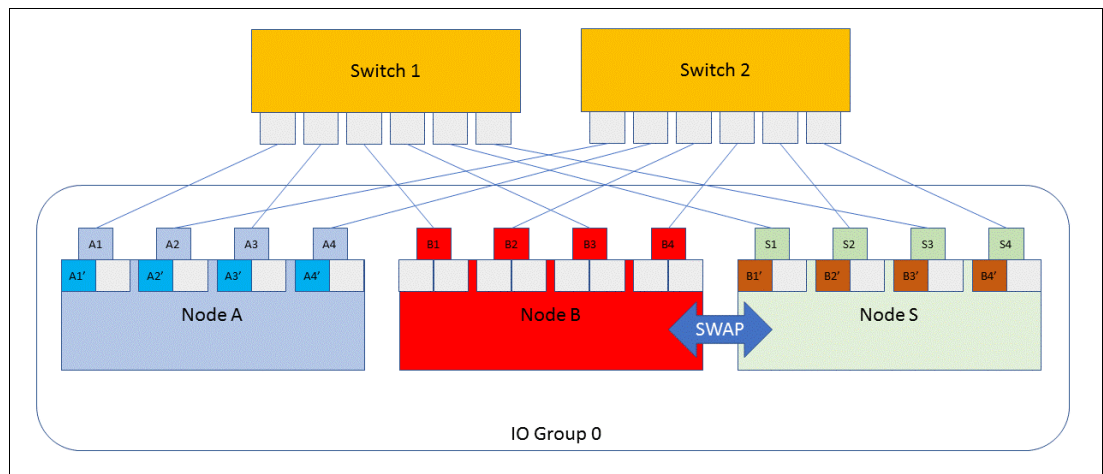


Figure 4 NPIV target ports failover to the spare node



In this state, the system is redundant once again. If a second failure occurs to node A, then the NPIV target ports will also be failed over to the spare node. At this point, the spare node will act in place of both the original nodes (Figure 5).

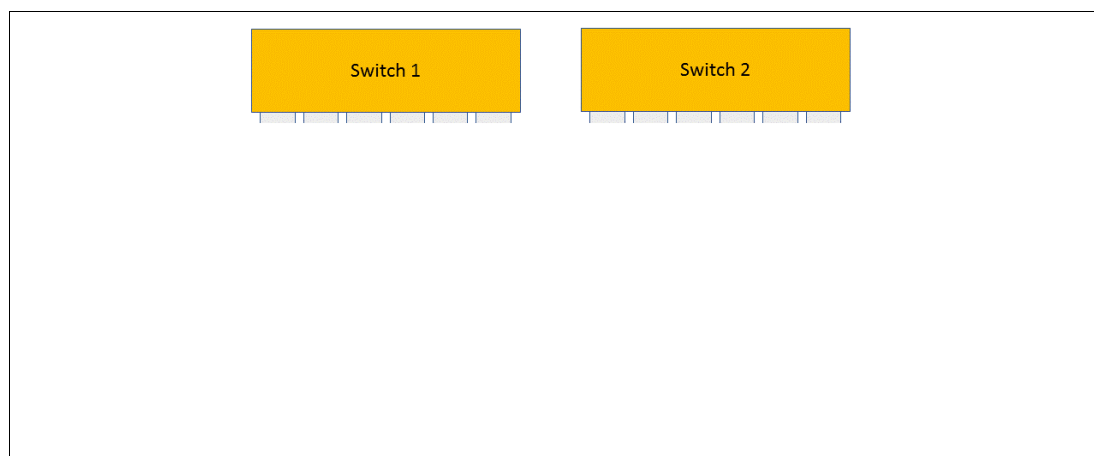


Figure 5 System is redundant again

## Hardware compatibility

For a spare node to be selected, the hardware must be compatible with the node it is replacing. If multiple spares are available, with varying hardware, then the most similar node is chosen from the pool of spares.

For a node to be considered at all, the following conditions must be true:

- ▶ Memory must match exactly, for example:
  - A 64 GiB SV1 node *can* replace a 64 GiB DH8 node.
  - A 128 GiB SV1 node *cannot* replace a 64 GiB SV1 node.
- ▶ For each Fibre Channel port on the original node, a Fibre Channel port with the same ID must exist on the spare node.
- ▶ Compression hardware existence. If compression hardware exists on the original node, then the spare must also have the hardware. Furthermore, if no compression hardware exists on the original, the spare must *not* have compression hardware either.

In addition, IBM Spectrum Virtualize selects a node with the best possible hardware match. The decision is made based on these factors:

- ▶ Hardware Type. The same hardware type is preferred (that is, an SV1 to replace an SV1).
- ▶ CPU count. The number of CPUs is preferred to be the same.
- ▶ Exact FC port. Exact FC port matches are preferred (that is, for nodes with ports 1,2,3,4, a spare with ports 1,2,3,4 is preferred over a spare with 1,2,3,4,5,6,7,8).

## Upgrade behavior

During an upgrade, a spare node, where available, is used to replace an original node that was taken offline for upgrade.

The spare will be brought online approximately one minute after the original node is taken offline for upgrade and will be removed when the upgrading node returns into the cluster.

After the upgrade is completed, the spare nodes are upgraded simultaneously. This happens for all nodes after the upgrade is committed.

More details about the upgrade process are in “Software upgrade” on page 17.

## Multi-site clusters

IBM Spectrum Virtualize clusters can be split across multiple data centers. When such a configuration is implemented, each node (including spares) is associated with a *site* parameter. IBM Spectrum Virtualize will choose only spares that have the same site parameter as the original node. This means that a spare from site A will never be used to cover a node from site B.

NPIV target port behavior is unaffected by site configuration.

## Enhanced stretched cluster (ESC)

In an ESC configuration, the two nodes of an I/O group are physically located at different sites. During an NPIV failover, NPIV ports move between sites, potentially leading to additional cross-site traffic. Where traffic from hosts to failover ports is not what you want, you can use zoning to prevent the cross-site traffic.

**Note:** Spare nodes are never used for site failures because the site setting must match. This means that spare nodes can be used to provide additional redundancy for hardware failures, but do not provide redundancy when a site failure occurs.

The fabric topology is largely unchanged when compared to a standard ESC topology. The spare node should be connected identically to the other nodes in the fabric to ensure that port failover occurs correctly.

Figure 6 shows an example configuration with two spare nodes, one at each site.

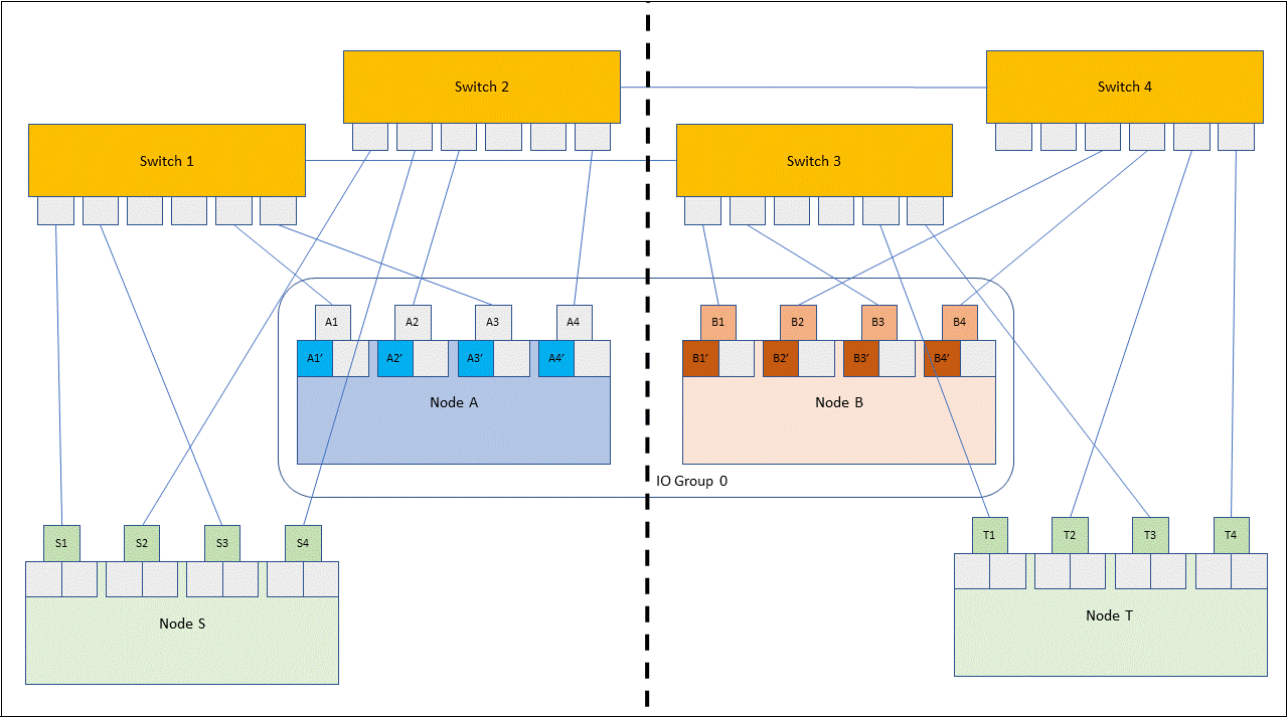


Figure 6 Two spare nodes, one at each site

Following a node failure, a node from the same site is always used. A node without a valid site setting will not be used for either site (Figure 7).

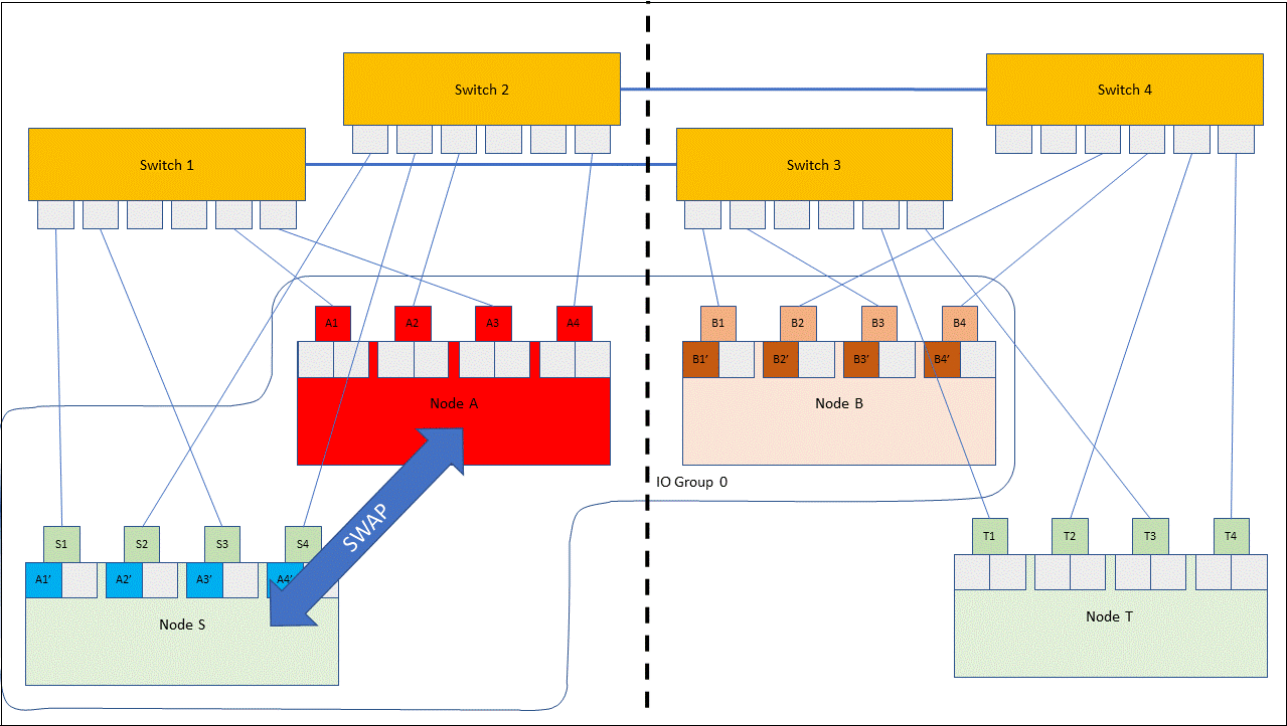


Figure 7 After a node failure

## HyperSwap

For an IBM HyperSwap® topology, at least one spare node should be at each site. NPIV will not cause additional intersite traffic because NPIV target port and spare failovers never occur cross-site (that is to say, NPIV failover will never happen between I/O Groups).

Following a node failure, a node from the same site will always be used. At least one spare is required on each site.

## Standard stretched cluster

A non-enhanced stretched cluster is a legacy support configuration where the nodes are physically located in separate sites, however the system topology and node site settings were not specified by the user.

**Important:** Spare nodes should not be added to such a setup because a spare from the wrong site could be selected, which will likely lead to performance problems.

### Interaction with service mode

A service mode node is treated as offline from the perspective of a spare node.

### Pinned data

In some double-failure scenarios, pinned data might possibly exist solely on an offline node. This is data, such as the SAN Volume Controller write cache, which is essential to the integrity of the volumes.

A spare node will never replace a node where non-redundant data exists, even if that node is offline.

An example of this scenario is as follows:

1. Node A fails.
2. Node B starts flushing cache.
3. Node B goes offline (before cache is flushed).

At this point, the data from the write cache exists only on node B, and a spare node has no way of accessing this data (because node B is offline). Therefore, there is no point of that spare being brought online.

### SNAP

Snaps collect data from spare nodes depending on their status:

<b>offline</b>	No data is collected.
<b>spare</b>	No live dump is possible; other data is collected.
<b>online_spare</b>	Treated like normal online node.
<b>service</b>	Treated like any other service mode node.
<b>adding/pending</b>	Snap will wait for this transitional period to pass.

# Implementation considerations

This section describes considerations before implementation.

## Supported hardware

NPIV target ports are supported on all Fibre Channel cards that are currently supported by IBM Spectrum Virtualize. NPIV target ports are not supported on any Fibre Channel over Ethernet (FCoE) cards, but FCoE hosts that are connected to FC ports through an FCoE-to-FC bridge are supported. Switches are required in order to support NPIV with a minimum supported port count of two NPIV ports per physical port. All 8 Gb and 16 Gb FC switches that are supported by the IBM Spectrum Virtualize software meet this requirement, and no special hardware is needed by the hosts.

## Restrictions

For interoperability restrictions on host support, always refer to the latest release notes.

Spare nodes are currently not supported if these conditions exist:

- ▶ SAS enclosures are attached.
- ▶ Any enclosure-based product (for example, IBM Storwize® V7000) is used.
- ▶ iSCSI is in use.
- ▶ Direct Fibre Channel attach is in use.

## Encryption

When using encryption and spare nodes, a feature code is required for all nodes, including spares. Adding a node to a cluster with encryption enabled is *not* possible unless a license and encryption feature code is available for that node.

Encryption keys will failover automatically to spare nodes. Standard recommendations for the location of USB drives containing encryption keys apply, although the USB drives should not be connected to the spare node, because inactive spare nodes cannot communicate keys to the active cluster.

## SAN fabric, zoning, and controller configuration

Where NPIV is enabled, the NPIV target port never changes its `port_id` as specified by the `lstargetportfc` command. That is, if the NPIV target port associated with port ID 1 on node 1 will failover to port ID 1 on node 2 (assuming node 1 and 2 are in the same I/O group).

As such, the fabric cabling should be configured so that port 1 on every node is connected to the same SAN and that zoning is always configured by WWPN.

### Retrieving WWPNs for zoning

When preparing zoning rules for NPIV-enabled clusters, use the `lstargetportfc` command. Two lines are output for each line in `lspportfc`, one with `virtualized=no`, and one with `virtualized=yes`. Ports that list `virtualized=no` correspond to the ports in `lspportfc`. These should be zoned between nodes, between clusters with replication relationships, and to back-end controllers.

When NPIV feature is set to transitional nodes, all storage is also mapped to hosts through the primary port. This way enables migration of existing configurations to those that use the NPIV feature, as reflected in `lstorageportfc` (Figure 8).

```
IBM_2145:BETA77_SVC:superuser>chiogrp -fctargetportmode transitional 0
IBM_2145:BETA77_SVC:superuser>lstorageportfc
```

id	WWPN	WWNN	port_id	owning_node_id	current_node_id	nportid	host_io_permitted	virtualized
1	500507680C510801	500507680C000801	1	1	1	3F0360	yes	no
2	500507680C550801	500507680C000801	1	1	1	3F0361	yes	yes
3	500507680C520801	500507680C000801	2	1	1	3F06E0	yes	no
4	500507680C560801	500507680C000801	2	1	1	3F0600	yes	yes
5	500507680C530801	500507680C000801	3	1	1	DF0082	yes	no
6	500507680C570801	500507680C000801	3	1	1	DF0A00	yes	yes
7	500507680C540801	500507680C000801	4	1	1	DF0001	yes	no
8	500507680C580801	500507680C000801	4	1	1	DF09C0	yes	yes
33	500507680C511013	500507680C001013	1	2	2	3F06C0	yes	no
34	500507680C551013	500507680C001013	1	2	2	3F0620	yes	yes
35	500507680C521013	500507680C001013	2	2	2	3F09E0	yes	no
36	500507680C561013	500507680C001013	2	2	2	3F0640	yes	yes
37	500507680C531013	500507680C001013	3	2	2	DF0040	yes	no
38	500507680C571013	500507680C001013	3	2	2	DF0A20	yes	yes
39	500507680C541013	500507680C001013	4	2	2	DF0020	yes	no
40	500507680C581013	500507680C001013	4	2	2	DF0AA0	yes	yes

Figure 8 The `lstorageportfc` transitional view 1

When NPIV feature is set to enabled, storage is mapped to hosts only through the NPIV target ports. This is again reflected in the `lstorageportfc` view (Figure 9).

```
IBM_2145:BETA77_SVC:superuser>chiogrp -fctargetportmode enabled 0
IBM_2145:BETA77_SVC:superuser>lstorageportfc
```

id	WWPN	WWNN	port_id	owning_node_id	current_node_id	nportid	host_io_permitted	virtualized
1	500507680C510801	500507680C000801	1	1	1	3F0360	no	no
2	500507680C550801	500507680C000801	1	1	1	3F0361	yes	yes
3	500507680C520801	500507680C000801	2	1	1	3F06E0	no	no
4	500507680C560801	500507680C000801	2	1	1	3F0600	yes	yes
5	500507680C530801	500507680C000801	3	1	1	DF0082	no	no
6	500507680C570801	500507680C000801	3	1	1	DF0A00	yes	yes
7	500507680C540801	500507680C000801	4	1	1	DF0001	no	no
8	500507680C580801	500507680C000801	4	1	1	DF09C0	yes	yes
33	500507680C511013	500507680C001013	1	2	2	3F06C0	no	no
34	500507680C551013	500507680C001013	1	2	2	3F0620	yes	yes
35	500507680C521013	500507680C001013	2	2	2	3F09E0	no	no
36	500507680C561013	500507680C001013	2	2	2	3F0640	yes	yes
37	500507680C531013	500507680C001013	3	2	2	DF0040	no	no
38	500507680C571013	500507680C001013	3	2	2	DF0A20	yes	yes
39	500507680C541013	500507680C001013	4	2	2	DF0020	no	no
40	500507680C581013	500507680C001013	4	2	2	DF0AA0	yes	yes

Figure 9 The `lstorageportfc` enabled view 2

Ports that list `virtualized=yes` correspond to the NPIV target ports. These should be zoned only to hosts, including nodes from other clusters for which this cluster is acting as a controller. The NPIV target ports must never be zoned with other SAN Volume Controller clusters in the same storage layer.

## Host zoning rules for new installations

For new host installations, where NPIV feature is enabled, or in transitional mode, the host should be placed only in zones with NPIV target ports. If all hosts are configured to be zoned with just NPIV target ports, then the NPIV feature should be set to enabled, rather than transitional.

## Host zoning rules for existing installations not using NPIV

Where a host was previously configured to use the primary ports, transitional mode can be enabled. This allows the host to use either the primary or NPIV target ports. The zoning should be configured to allow access to both primary and NPIV target ports while multipathing on the host is reconfigured to use the NPIV target ports. Be sure to remove the primary ports from the zone after the multipath reconfiguration is completed.

## Zoning between IBM Spectrum Virtualize clusters for remote copy

Communication between IBM Spectrum Virtualize nodes, including between different clusters, takes place over primary ports, so the primary ports must always be zoned to each other as in previous releases regardless of the NPIV mode or the relationship between clusters.

For this purpose, spare nodes should be included as nodes in the cluster. When a spare node comes online, its primary ports will be used for remote copy relationships and, as such, must be zoned with the remote cluster.

## Zoning between IBM Spectrum Virtualize and back-end controllers

Back-end controllers must be zoned to the primary ports on IBM Spectrum Virtualize nodes where a spare node is in use, and that node must be configured for use by the back-end controller as with any other controller.

### Notes:

- ▶ Currently, the zoning configuration for spare nodes is not policed while the spare is inactive, and no errors will be logged if the zoning or back-end configuration is incorrect.
- ▶ Clusters running a version of IBM Spectrum Virtualize software *earlier* than V7.7.0 should not be zoned in to the NPIV target ports of an IBM Spectrum Virtualize V7.7.0 or *later* cluster.

## Zoning between IBM Spectrum Virtualize replication layer and storage layer clusters

A common use case is to use IBM Spectrum Virtualize appliances, such as SAN Volume Controller, as a virtualization layer for IBM Spectrum Virtualize based back-end controllers, such as the Storwize V7000. In such cases, the virtualizing layer is referred to as the *replication* layer and the storage is referred to as the *storage* layer.

In such a configuration, those rules continue to apply, with the replication layer device being considered a host for the storage layer device.

As such, the primary ports on the replication layer device should be zoned with the NPIV target ports on the storage layer systems.

**Note:** Clusters running a version of IBM Spectrum Virtualize software *older* than V7.7.0 should not be zoned in to the NPIV target ports of an IBM Spectrum Virtualize V7.7.0 or *newer* cluster.

## Back-end controller configuration

IBM Spectrum Virtualize uses the primary ports to communicate with the back-end controller, including the spare. This means that all MDisk must be mapped to all IBM Spectrum Virtualize nodes, including spares. For IBM Spectrum Virtualize based back-end controllers, such as Storwize V7000, use the *host clusters* functionality, with each node forming one host within this cluster. This way will ensure that each volume is mapped identically to each IBM Spectrum Virtualize node.

Ensure that volumes are mapped in these ways:

- ▶ Volumes are mapped to all nodes.
- ▶ Volumes are mapped with the same SCSI ID through all paths.

## Configuring NPIV target port functionality

This section describes how to configure NPIV target port functionality.

### Enabling NPIV target port functionality on a new cluster

Complete these steps:

1. After you create the cluster, but before you begin adding hosts, issue the following command on each populated I/O group:  

```
chiogrp fctargetportmode transitional <I/O group>
```
2. When that is completed, issue the following command on each populated I/O group:  

```
chiogrp fctargetportmode enabled <I/O group>
```
3. If you are using WWPN based zoning, issue the **lstargetportfc** command to view the set of FC ports. All ports that display `virtualized=yes` must then be zoned into hosts as described in “SAN fabric, zoning, and controller configuration” on page 13.

### Enabling NPIV target port functionality

Before beginning the process of enabling NPIV ports, first audit the Fibre Channel fabric to ensure it conforms to the required cabling and zoning rules. If it does not, make any required changes, ensuring that hosts are zoned to the correct non-NPIV ports after the fabric changes.

Then, complete these steps:

1. If using WWPN based zoning, add the NPIV target ports to the hosts zones.
2. Check the number of available paths between each host and the storage. If all NPIV and non-NPIV ports are zoned to the hosts, the number of available paths will double while the NPIV mode is set to transitional. If this will result in the number of paths exceeding supported maximums, ensure that hosts are not zoned to the NPIV target ports at this point.  
  
If using WWPN based zoning, and you are not limited by the maximum supported path count, add the NPIV target ports to the host zones.
3. Issue the following command on each I/O group for which you want to activate NPIV:  

```
chiogrp fctargetportmode transitional <I/O group>
```
4. If hosts are not zoned to all NPIV target ports due to the maximum supported path count being exceeded, continue zoning now, one port at a time, by removing a non-NPIV port from the zone and then adding the corresponding NPIV target port.

At this point, the hosts automatically switch over to using the NPIV ports for their I/O.



5. Issue the **lsfabric** command for each host, and ensure that each has logins to the expected set of NPIV ports. At least one of those logins per host should be active if the host is currently performing I/O. Hosts that do not respect optimized paths might still be using non-NPIV ports for I/O at this point, in which case the NPIV target port logins will not be active.
6. After all hosts start using the NPIV target ports, issue the following command on each I/O group for which you want to activate NPIV:  

```
chiogrp fctargetportmode enabled <IO group>
```

If during this step, a host I/O outage occurs, immediately switch back to transitional mode.

**Attention:** Do not switch back to a disabled state, because this risks an outage on hosts that switched over to using the NPIV target ports.

If you use WWPN-based zoning, remove all non-NPIV ports from the host zones.

### Disabling NPIV target port functionality

Before starting, check the number of available paths between each host and storage, as you did in the procedure to enable NPIV. Then, complete these steps:

1. Issue the following command on each I/O group for which you want to deactivate NPIV:  

```
chiogrp fctargetportmode transitional <IO group>
```

At this point, hosts should still be using the NPIV target ports for their I/O.
2. Issue the **lsfabric** command for each host, and ensure that each has logins to the expected set of NPIV target ports. Logins to the non-NPIV ports might be marked as inactive.
3. Issue the following command on each I/O group for which you want to deactivate NPIV:  

```
chiogrp fctargetportmode disabled <IO group>
```

### Adding spare nodes

A spare can be added by using the **addnode** command. Unlike normal nodes, spares are not associated with any I/O group and as such do not need to have one specified, for example:

```
addnode -panelname <panelname> -spare
```

Note that any candidate node can be nominated as a spare.

## Software upgrade

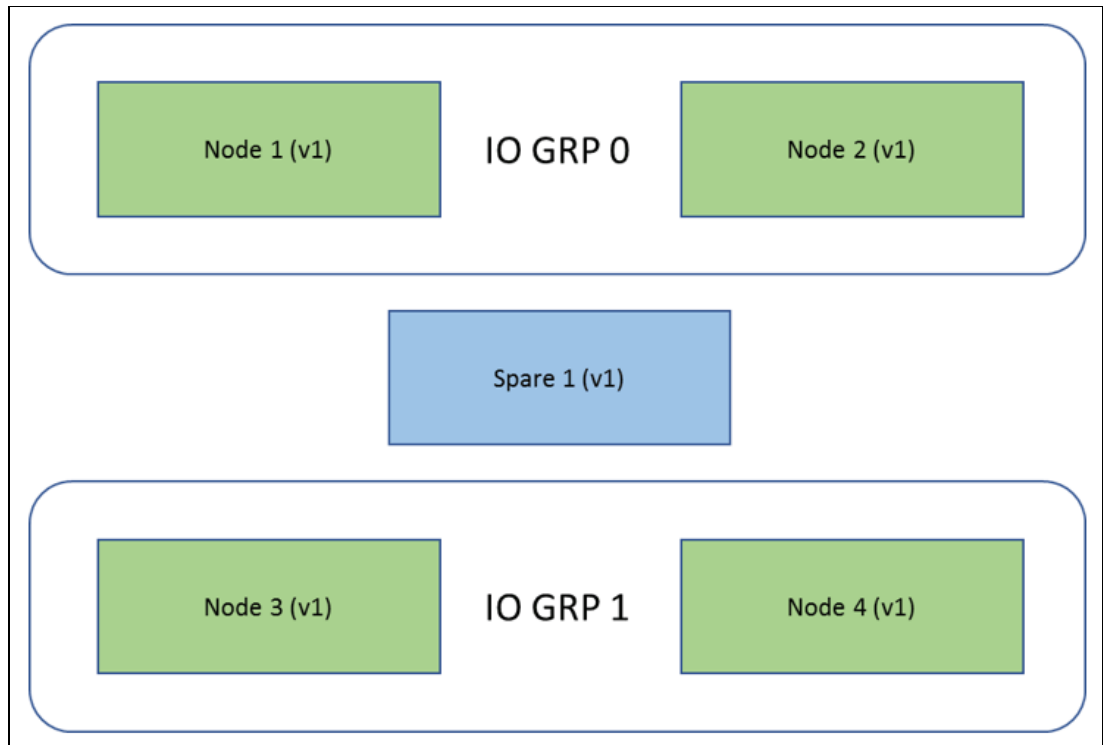
SAN Volume Controller supports two methods to upgrade the cluster code level concurrently:

- Automatic concurrent code upgrade (CCU): This method is preferred. Normally, you run an automatic CCU by using the **applysoftware** command.
- Manual CCU: This alternative is a more complex sequence that involves upgrading each node. Using the automatic mechanism is highly suggested.

### Automatic CCU

During an automatic CCU, the system will handle taking each node offline one by one and upgrading them. One half of each I/O group will be upgraded initially, followed by a configurable delay (which defaults to 30 minutes), and finally the second half of each I/O group is upgraded.

The example system in Figure 10 shows two I/O groups, each with two nodes.



*Figure 10 Two I/O groups, each with two nodes*

The upgrade time of each node is 15 - 30 minutes, depending on the hardware involved and the size of the upgrade; during this time, the node will be offline. An arbitrary node will upgrade first. The example in Figure 11 on page 19 shows node 1 upgrading.

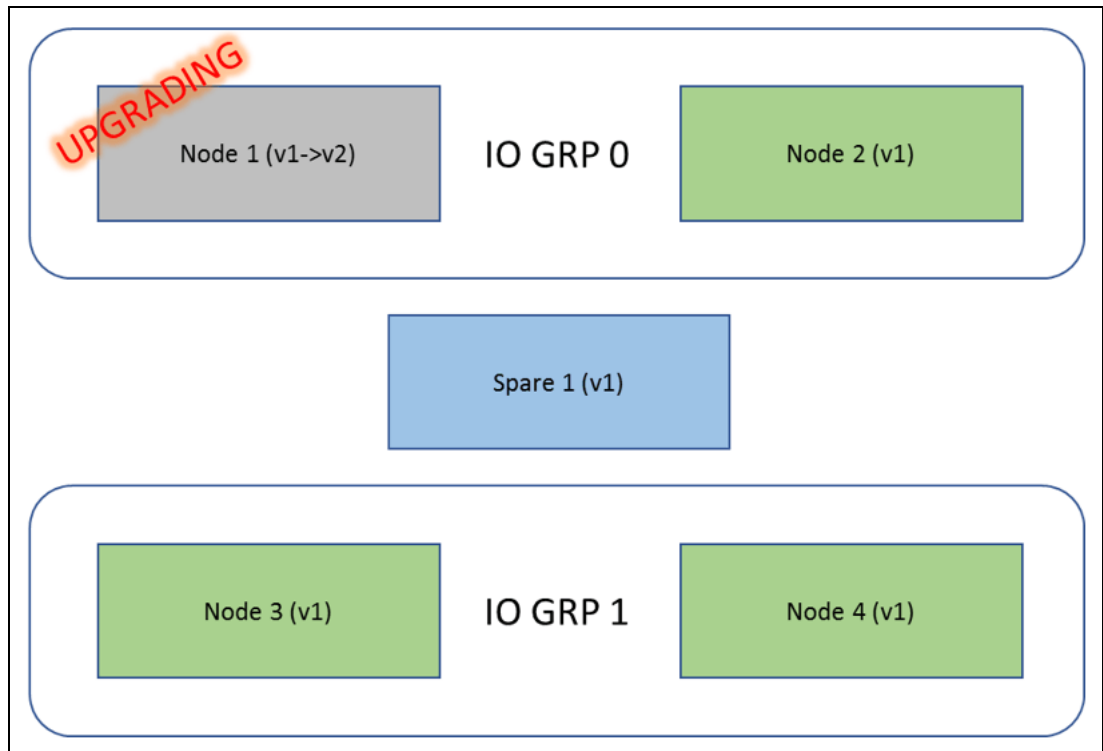


Figure 11 Node 1 upgrading

If a spare node is available, it will be brought online approximately one minute after the node to be upgraded goes offline for its upgrade. Logically, spare 1 will be part of I/O group 0 and will be swapping out node 1 (Figure 12).

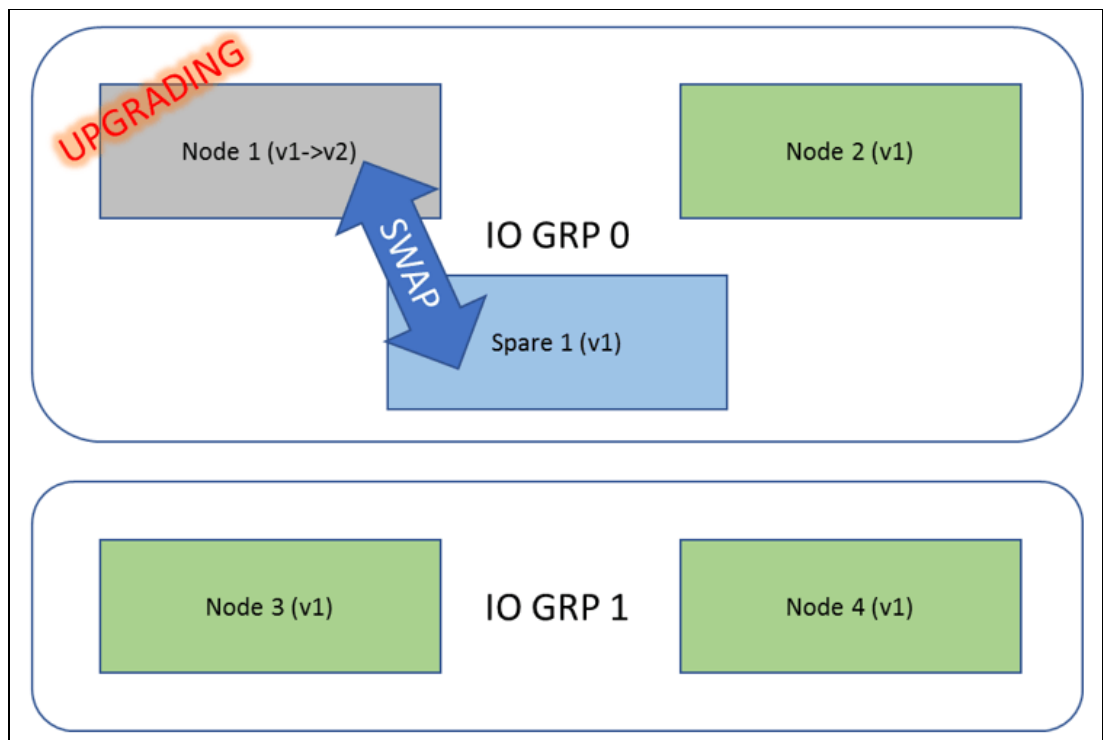


Figure 12 Swapping out node 1

After the node is successfully upgraded, the spare is removed from the cluster and the original, the upgraded node will rejoin. At this point, the spare will be reconfigured into I/O group 1 (Figure 13).

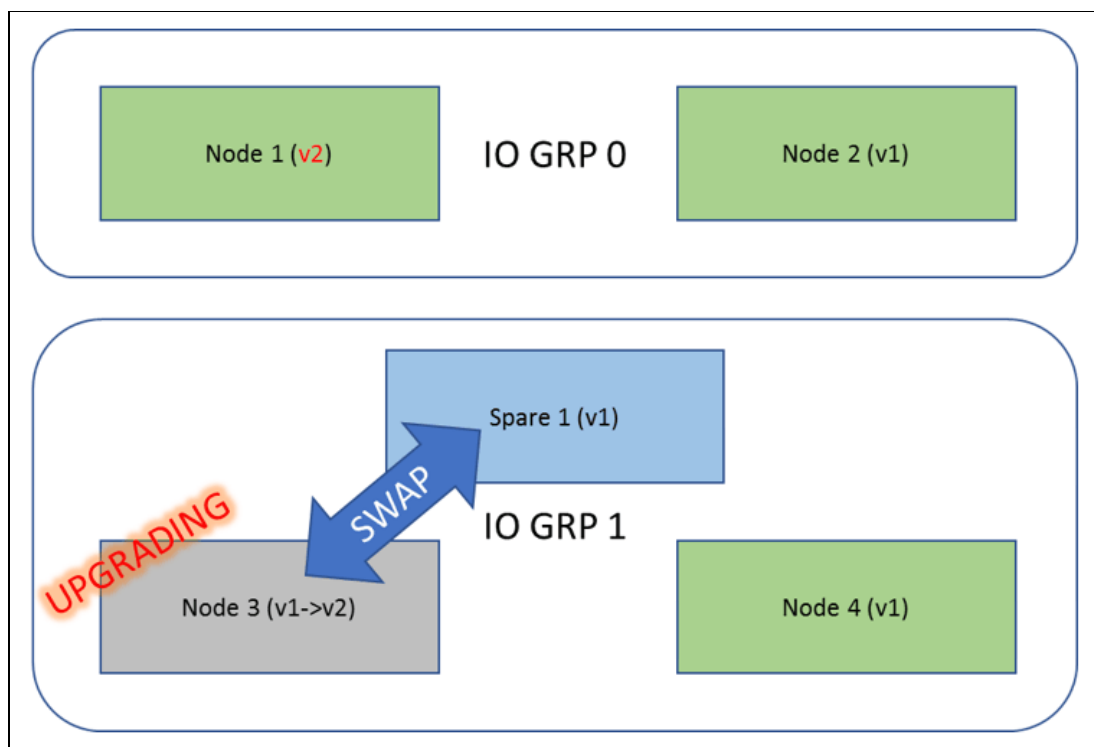


Figure 13 Spare reconfigured into I/O group 1

After all original nodes are upgraded, the CCU completes and commits, which means a downgrade is no longer possible. After successfully committed, all configured spares are upgraded to the new code level. While the spares are offline upgrading, they are not used to replace any offline cluster members.

### Recovering from failed automatic upgrade

At any point in the sequence, a node can fail. This section describes various recovery procedures to help recover from failure.

**Important:** The decision as to which recovery mechanism to use is critical; always consult IBM® support before taking any action.

This paper does not attempt to describe recovery from every possible scenario, but leaves that to the documentation used by the IBM Level 3 (L3) support staff. Instead this paper describes the most likely failure scenarios, and those directly affected by the spare node functionality.

## Understanding the software upgrade status

The software upgrade status will be one of these:

<b>inactive</b>	The upgrade is no longer running.
<b>upgrading</b>	The upgrade is progressing normally.
<b>downgrading</b>	The user aborted an upgrade and the system is currently downgrading.
<b>stalled_non_redundant</b>	The upgrade stalled and the system is not redundant (the removal of a single node will result in volumes going offline).
<b>abort_stalled_non_redundant</b>	Identical to <code>stalled_non_redundant</code> , but detected during an abort/downgrade.
<b>stalled</b>	A problem was detected, but the system recovered to a redundant state.

Typically, a failed upgrade causes the upgrade to go into a stalled state. This can be identified by using the **lsupdate** command.

In some circumstances, a node might fail while being upgraded and never return to the cluster. If any single node fails to upgrade in approximately 90 minutes, it can be considered as having had a hardware failure during the installation. An upgrade can be considered implicitly stalled at this point.

The following sections explore the common use-cases for recovering from a failed upgrade.

**Note:** In all failed upgrades, consult IBM support. IBM support will advise of the reason for any failure. Be sure to have at least a basic understanding of what caused the failure before executing any recovery.

### Case 1: Upgrade stalled, all nodes online

The simplest recovery case is where all original nodes are online. In this case, the upgrade can be either aborted (it downgrades to the previous level) or resumed (it continues the upgrade procedure). See Figure 14.

L3 support will advise about the root cause of the failure and whether an abort or resume is recommended.

The abort/resume can either be executed through the GUI, or by using the **applysoftware** command, and will cause the automated upgrade procedure to continue.

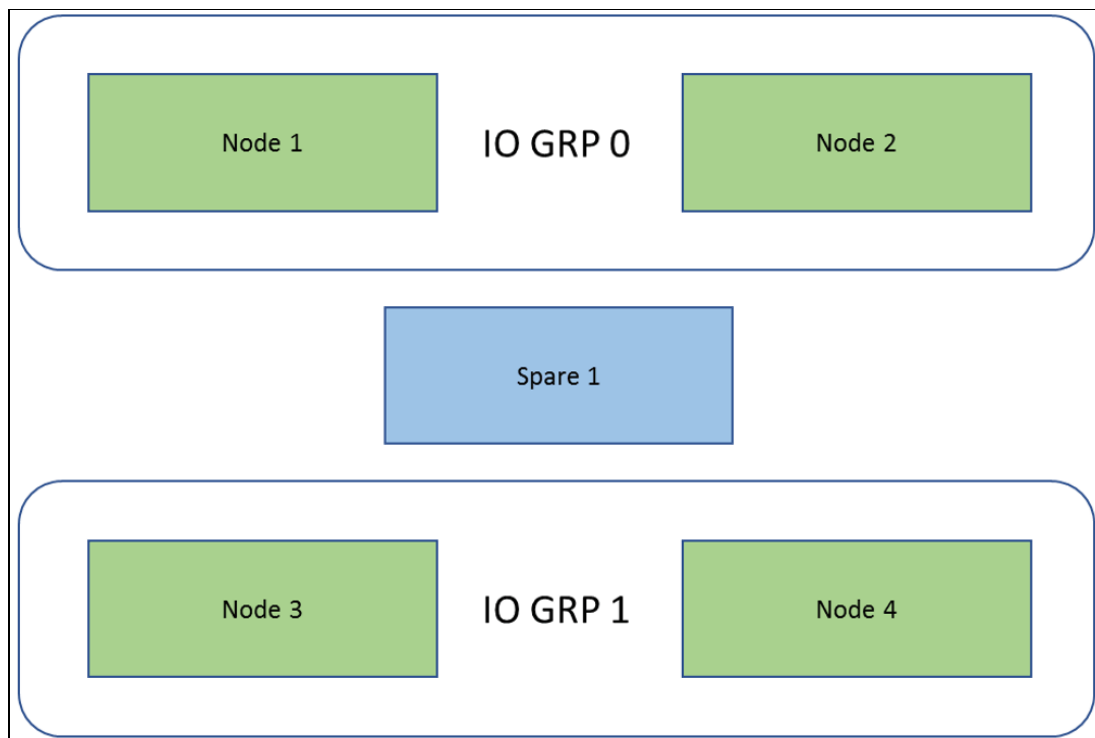


Figure 14 Upgrade stalled, all nodes online

## Case 2: Upgrading node fails to return

In this case, the upgrading node either failed to upgrade, in which case the upgrade appears to be continuing to upgrade but is stuck for 90 minutes or more, or returned into the cluster and then immediately failed, in which case the upgrade stalls.

If the upgrade is stalled, a possibility is that the failed node is downgrading and will rejoin the cluster soon. The system should be allowed at least 30 minutes from the point of stalling to determine whether the node rejoins the cluster. If the node does rejoin, follow case 1 because all nodes will now be online. See Figure 15.

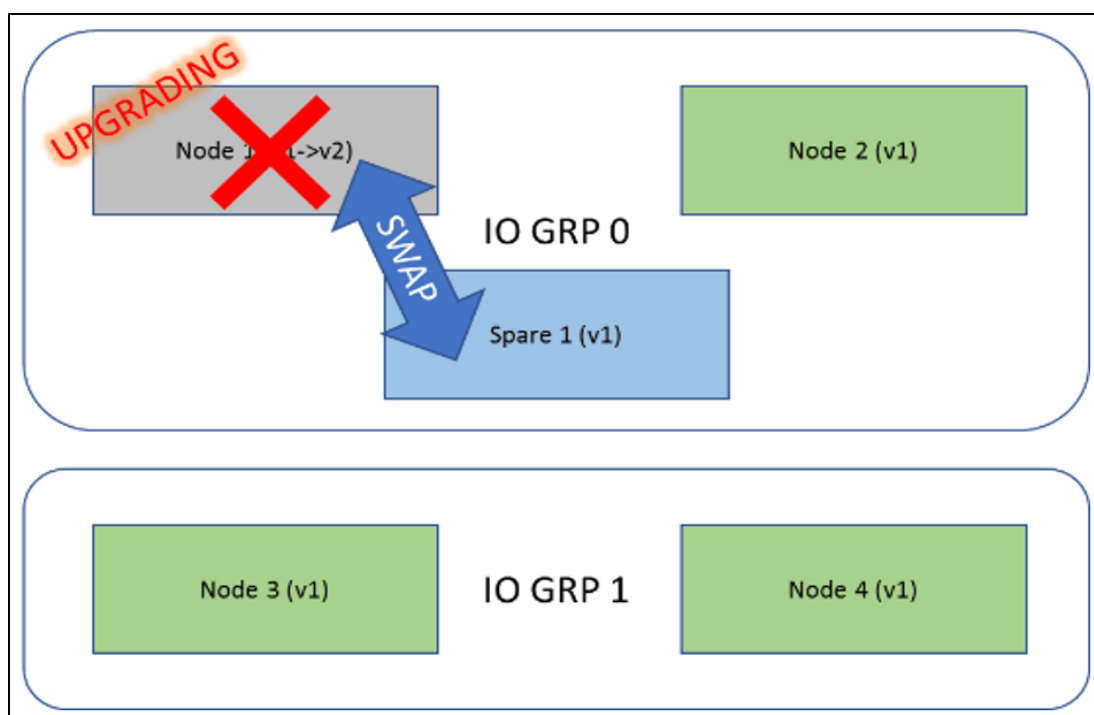


Figure 15 Upgrading node fails to return

If the upgrading node never returns to the cluster, several options to repair are available.

### Option 1: Repair the node without wiping it

If the failed node can be recovered, for example by replacing hardware, in a manner that does not wipe the node, then the node will automatically rejoin the cluster. Then, one of the following results might happen:

- ▶ Node reenters cluster at an upgraded level and the upgrade was not stalled. The upgrade will continue, and no further action is required.
- ▶ Node reenters the cluster at an upgraded level and the upgrade was stalled. The upgrade should be resumed; you can then follow Case 1.
- ▶ Node reenters cluster at the old level. The upgrade will stall; you can then follow Case 1.

### ***Option 2: Reinstall or replace node***

If the failed node cannot be recovered, and must be installed or wiped with the **satask leavecluster** command, it will be returned to the candidate state. At this point the spare node can be reintroduced into the cluster by using the following command:

```
swapnode -replace <original node id>
```

This command deactivates the spare from the cluster and the original node will rejoin at the old level. The upgrade will stall, you can then follow Case 1.

### ***Option 3: Node cannot be recovered, promote spare to original node***

In this option, the original node cannot be recovered for any reason. If you want, the spare can be promoted to become an original node. The procedure described here takes the spare out of the cluster and changes its worldwide node name (WWNN) to match that of the original failed node.

1. Be sure that the original node is powered off during this procedure.
2. Remove the original node by using the following command:

```
rmnode <spare node id>
```

3. Deactivate service mode if required by using this command:

```
satask stopservice <spare panel name>
```

4. Change the WWNN of the spare node to match the original node with this command:

```
satask chwwnn -wwnnsuffix <wwnn suffix of original node> <spare panel name>
```

5. Replace the original node with the new candidate by using this command:

```
swapnode -replace <original node id>
```

### ***Option 4: Remove original node***

If options 1 - 3 are not available, the original node can be simply removed from the cluster by using the following command:

```
rmnode -deactivatespare <original node id>
```

At this point the system will be non-redundant because it is without an active spare.



### Case 3: Partner node fails during upgrade

In this scenario, node A is down for upgrade and is covered by a spare, and its partner node B suffers a fatal hardware failure. The upgrading node is assumed to have upgraded successfully. See Figure 16.

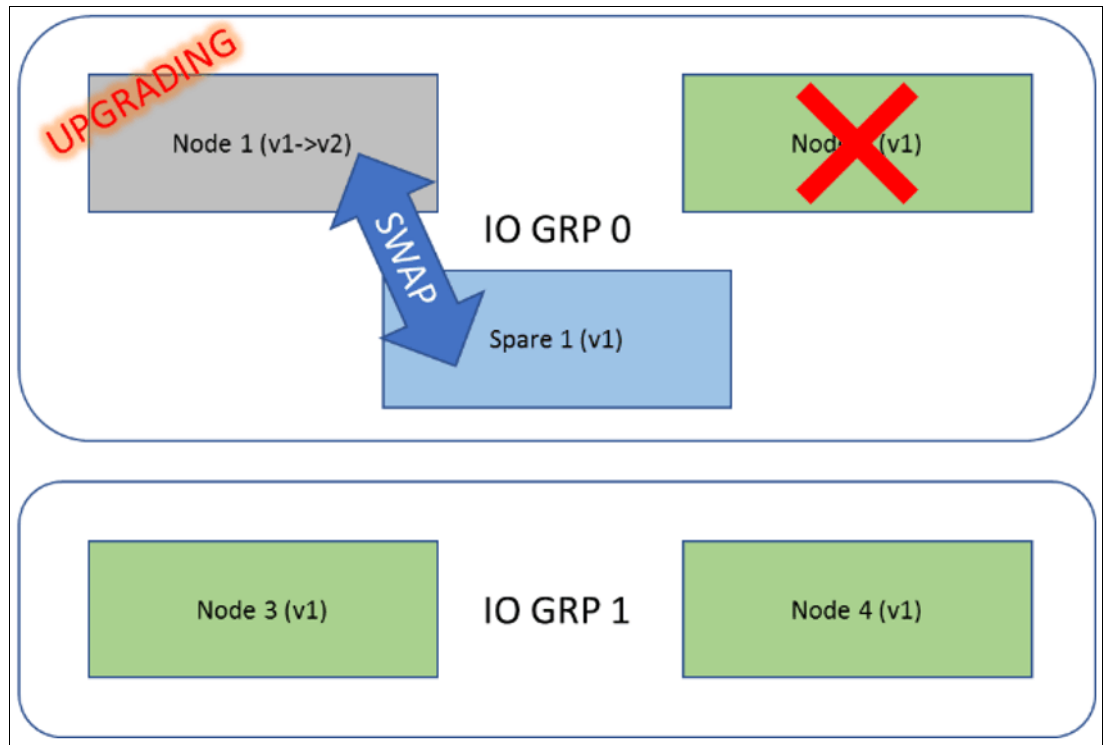


Figure 16 Partner node fails during upgrade

This is the situation:

- ▶ The upgrade state is `stalled_non_redundant`.
- ▶ The upgrading node is unable to rejoin the cluster (because the system is dependent on the spare).

Complete these steps:

1. On the upgrading node, issue **satask leavecluster**, to deactivate it from the cluster. The node will then downgrade.
2. Power-off the failed node B.
3. Modify the WWNN of node A to that of node B.
4. If you want to resume the upgrade, issue the **satask installsoftware -pacedccu** command on node A.
5. Run the **swapnode -replace <node B>** command.
6. Remove offline node A from the cluster by using the **rmnode -deactivatespare** command.
7. Resume or abort the upgrade as appropriate.

Now the system is online, upgrade is complete, but it is non-redundant because the spare cannot take over from a deactivated node.

If node B can be repaired quickly, do the following steps:

1. Issue the **satask leavecluster <node B>** command, if the node is not already in a candidate state.
2. Change the WWNN of node B to that of node A.
3. Add the node to the cluster by using the **addnode** command.

If spare parts are not available, the spare can be added into the cluster as an original node:

1. Remove the node with by using the **rmnode <spare node>** command.
2. Add the spare node as an original node into the cluster.

#### Case 4: Original node offline/deactivated. Spare fails

In this scenario, no spare is available or an original node was removed from the cluster by the user.

Observe the state of the partner node to the offline node. If it upgraded, the upgrade should be continued. Otherwise, the upgrade should be aborted.

Any other choice will result in the system becoming unavailable while the non-redundant node is upgraded or downgraded.

## Manual upgrade

The manual upgrade procedure is only for advanced users and should only be executed where extreme care must be taken over the timing and ordering of nodes being removed from the cluster.

**Note:** If the only reason to perform a manual CCU is to control the length of the mid-CCU pause and upgrading from V8.1.0.0 release or later, then consider using the **-delay** parameter on the **applysoftware** command.

To manually upgrade, each node must be removed from the cluster, upgraded and re-added. When a spare node is configured into the cluster, that spare can be used to substitute for the original node while it is removed; however, because the **rmnode** command deactivates any NPIV ports, a different procedure must be used if the spare is required. Each node in the cluster must be upgraded in turn. Use the same sequence as described in “Automatic CCU” on page 17. For each node, complete the following steps:

1. Use SCP upgrade file into /file on node.
2. Swap in a spare. Use the **-spare** parameter if a particular spare is required:  
`swapnode -failover <node to be upgraded>`
3. Wait for the upgrading node to enter the service state.
4. Force the service state node out of the cluster by using this command:  
`satask leavecluster -force <upgrading node panel name>`
5. Wait for the upgrading node to warm start.
6. Reinstall the node using the following command:  
`satask installsoftware -pacedccu <upgrade file> <upgrading node panel name>`
7. After being upgraded, the node will be in a candidate state. Replace the original node with the upgraded version of itself by using the following command:  
`swapnode -replace <target node id>`

## Recovering from a failed manual upgrade

The recovery procedures for a failed manual upgrade are identical to that of an automatic upgrade, except that in case 1, the resume/abort operation must be carried out manually.

In a case where the upgrade must be backed out, the upgraded node must be manually removed from the cluster, downgraded (using the **satask installsoftware** command) and reentered into the cluster.

## Special cases

This section describes special cases.

### Single-spare active

In a cluster with a single I/O group and with a single active spare, the possibility exists that both real nodes will be in service state and the only node active in that cluster is the spare.

Because failing back implies that a period exists in which neither the spare nor the node it has replaced are active, failing back the spare under these circumstances is not possible, and any attempt to do so will fail.

Ensure the node that was not replaced is restored to the cluster first. This establishes a two-node cluster, containing itself and the spare, allowing the spare to then fail back safely because any redundancy issues are now avoided.

### Replacing a spare with a spare

A spare that is currently active goes into a service state, then it can be replaced with another spare. The second spare is a direct swap for the first.

On failing back, the only necessary task is to failback from the spare that is currently active to the original node. Multiple failbacks will not occur, and need not be manually implemented.

## Manual failover

You can use the **swapnode** command to bring the spare nodes online. This command can be used to force failovers. In addition, it can be used for maintenance procedures to avoid running **rmnode/addnode** commands.

The **swapnode** command has the following options:

- ▶ **swapnode -failover [-spare <spare node id>] <original node id>**  
Forces a failover of original node to spare.
- ▶ **swapnode -failback <spare node id>**  
Forces a failback of a spare node to an original node
- ▶ **swapnode -replace <original node id>**  
Coordinates the replacement of an original node with a new (candidate) node on the fabric.
- ▶ **swapnode -service <original node id>**  
Puts an original node into service status.

## Failover

The **swapnode -failover** command forces an original node into service mode. A compatible spare must be available and the most similar node (see “Hardware compatibility” on page 9) will be used.

If you specify the **-spare** parameter, that spare will be used in place of other spares.

Note that the **-force** flag overrides redundancy checking (that is, a spare will be brought online, even if the action will cause volumes to go offline). This flag does not override the minimal hardware match requirements (that is, memory configuration in the spare must match the original node).

## Failback

The **swapnode -failback** command will immediately take a spare node offline, causing the returning original node to take over if available.

The **-force** parameter can be used to force this operation, even if offline volumes will result.

If the original node is in service mode, that node will not be taken out of service mode by the execution of this command and a service mode node is treated identically to an offline node.

If the original node is in an offline or service state, a spare node will come online after a delay of 5 minutes. This might be the same node that failed back by **swapnode -failback** command.

The **failback** command is designed to be used as a tool for service, or in cases where an automatic failover cannot occur for any reason.

## Replace

The **swapnode -replace** command replaces maintenance procedures that historically used the **rmnode** or **addnode** commands to replace a node in the cluster.

The command initially checks that a replacement node with the same WWNN as the original node is available on the fabric and appears as in the **lscnodecandidate** command. It also checks that the specified node is currently online and that no non-redundant hardened data was on that node.

If the original node was replaced by a spare node, then this command will complete and the process of replacing the node will be completed as a background task:

1. The original node is deactivated from the cluster.
2. After the spare is deactivated, the original node will be logically removed from the cluster and replaced by the candidate node.

If at any time, the candidate node fails or otherwise becomes unavailable, the process is aborted. A spare node will be brought back online after approximately 5 minutes.

If no spare node is available, the command initiates the replacement of the original node with the candidate node before the command is completed.

In both cases described, the system watches that the amount of memory on the new node is identical to the amount of memory of the original node; however, no other hardware checks are made on the candidate node.

This command can also be used for hardware maintenance that loses the hardened data on the node. For example, replacing all HDDs on a node will return that node to a candidate state, at which time the **swapnode -replace** command must be used to bring that node back into the cluster.

## Service

The **swapnode -service** command simply places a node into service mode. This task is distinct from the **satask startservice** command in the following ways:

- ▶ Better redundancy checking. The reason is because this command is coordinated by the cluster rather than the node and it is better able to check whether the action would result in offline volumes.
- ▶ Proactive failover. The NPIV ports are proactively failed over before the node goes offline.

## Upgrading hardware

IBM Spectrum Virtualize software allows upgrading hardware with a new node in these ways: with and without spare nodes.

### With spare nodes

Complete these steps:

1. The cluster must have at least one spare node that protects all active nodes in it.
2. Make a note of the WWNN of the node to be upgraded; issue one of these commands:
  - Issue **sainfo lsservicestatus** on the node.
  - Issue **svcinfo lsnode** on the config node.
  - Issue **svcinfo lstargetportfc** on the config node.
3. Put the node that needs to be upgraded into service state; issue one of these commands:
  - Issue **satask startservice** for that node to put the node in a service state; the spare takes over its place after 5 minutes.
  - Issue **swapnode -failover** for that node to put the node in a service state; the spare takes over its place after 1 minute.
4. Determine if the node to be replaced is in a service state and the spare now has the `online_spare` status. Use one of the following methods:
  - The **svcinfo lsnode** command
  - The **sainfo lsservicenodes** command
  - The management GUI
5. Power down the node that is in a service state:  
`satask stopnode -poweroff <panel_name>`
6. After the node is powered off, remove that node from the rack.
7. Add the new node into the rack and complete all cabling.
8. Power on the new node.
9. Log in to this new node and change its WWNN. Assign the old node's WWNN to the new node:
  - Issue **satask chwwnn -wwnnsuffix <wwnn\_suffix> <panel\_name>** if it is a CG8 or CF8.
  - Issue **satask chvpd -wwnn <WWNN> <panel\_name>** if it is a DH8 or SV1.
10. Check the WWNN of the new node, using the commands described in step 2.
11. Replace the offline node in the cluster with the new node:
  - Issue **swapnode -replace <offline node ID>** on the config node.
12. Wait for the new node to replace the old node and for the spare node status to transition from `online_spare` to `spare` status:
  - Issue **svcinfo lsnode** on the config node after it is transitioned.

- 13.Repeat steps 1 - 12 for all the active nodes in the cluster.
- 14.After all active nodes are upgraded, upgrade all spare nodes, by using the following steps and commands:
  - a. Remove the spare node: **svctask rmnode <spare node ID>**
  - b. Power-off the spare: **satask stopnode -poweroff <panel\_name>**
  - c. Remove the old node and insert the new node, and complete all cabling.
  - d. Power on the new node.
  - e. Add the new node as a spare: **svctask addnode -panelname <panel\_name> -spare**
  - f. Check that the spare node was added: **svcinfo lsnode**
- 15.Repeat these steps for all the spare nodes.

## Without spare nodes

Complete these steps:

1. Make a note of the WWNN of the node to be upgraded; issue one of these commands:
  - Issue **sainfo lsservicestatus** on the node.
  - Issue **svcinfo lsnode** on the config node.
  - Issue **svcinfo lstargetportfc** on the config node.
2. Put the node that needs to be upgraded into service state:
  - Issue **satask startservice** for that node. This causes the host I/O port to fail over to the partner node in that I/O group.
3. Determine if the node to be replaced is in a service state; use one of these methods:
  - Use the **svcinfo lsnode** command.
  - Use the **sainfo lsservicenodes** command.
  - Use the management GUI.
4. Power-down the node that is in a service state:
 

```
satask stopnode -poweroff <panel_name>
```
5. After the node is powered-off, remove this node from the rack.
6. Add the new node in the rack and complete all cabling.
7. Power-on the new node.
8. Log in to this node and change the WWNN of this node. Assign the old node's WWNN:
  - Issue **satask chwwnn -wwnnsuffix <wwnn\_suffix> <panel\_name>** on the CG8 or CF8.
  - Issue **satask chvpd -wwnn <WWNN> <panel\_name>** on the DH8 or SV1.
9. Check the WWNN of the new node by using the commands in 1.
- 10.Replace the offline node in the cluster with the new node:
  - Issue **swapnode -replace <offline node ID>** on the config node.
- 11.Wait for the new node to replace the old node:
  - Issue **svcinfo lsnode** on the config node.
- 12.Repeat steps 1 - 11 for each node in the cluster.

## Hardware maintenance

The hot-spare node functionality can be used to provide redundancy while hardware maintenance is performed on a node. The procedure depends on the effects that the maintenance procedure will have on the node. Some activities (such as replacing the internal disk) remove the node's knowledge of its cluster membership.

### Running nodes

A running node that needs maintenance can be swapped out using the **swapnode** command:

- ▶ For systems with a hot-spare node:  
`swapnode -failover <node requiring maintenance>`
- ▶ For systems without a hot-spare node:  
`swapnode -service <node requiring maintenance>`

Both commands result in placing the specified node in service mode. Then, the node can be switched off and hardware replaced. When that node is restarted, it is automatically returned to the cluster.

### Maintenance where internal node state is lost

Some maintenance procedures, for example, replacing the internal disk, will result in the node not being aware of which cluster to join. If that happens, the previous procedure will result in a candidate node and the node will not be automatically returned into the cluster.

In such an event, the **swapnode** command can be used to re-add this node back to the cluster:

```
swapnode -replace <offline node ID>
```

## T3 or T4 recovery

Recovery scenario can be T3 or T4:

- |                    |  |
|--------------------|--|
| <b>T3 recovery</b> | A mechanism where the configuration of the system is restored from a backup contained on the quorum disks. The data on the volumes is kept intact, although some write cache might be lost during the procedure. (See "What happens if the config node is a spare".) |
| <b>T4 recovery</b> | A mechanism where the configuration is restored from a backup file. The configuration is restored, but the data is not.  |

In both scenarios, any online spare nodes will be deactivated during the process. Only the nodes that are online at the time of the procedure will be rebuilt into the configuration.

In a scenario where an original node is missing, is covered by a spare, and cannot be recovered before a T3 recovery procedure, that node will not be recovered and will not be covered by a spare.

Never run a T3 or T4 recovery procedure from a spare node.

### What happens if the config node is a spare

Although unusual but not impossible, the configuration node might be the spare node.

If that is the scenario, the T3 recovery must *not* be run from the spare node. Instead the configuration backup file must first be copied to one of the original nodes from where the T3 restore can now be run.

## Testing the implementation

A strong suggestion is that a failover test be run in order to test that the spare nodes will take over as expected in the event of a node failure.

Complete these steps:

1. The spare should be configured, NPIV target ports enabled, and the hosts configured to use the NPIV target ports. After configuration is correct, first check that the system is healthy:
  - No unfixed errors are in the error log; ensure controller redundancy and no 88002 errors
  - All controllers are online
  - All MDisks are online
  - All volumes are online
  - Check that no volume is dependent on any single node by using the following command (the result should be empty):  

```
svcinfoldependentvdisks -node <node id>
```
2. After the system is established as fully redundant, test the failover by placing a node into service mode:  

```
satask startservice -force <panel name>
```

After 5 minutes, the cluster observes that this node is offline, not returning, and will attempt to replace it with a spare, which will transition to the state `online_spare`.
3. Now, make the following checks:
  - All MDisks and controllers are online:  
If any MDisks are degraded, check the back-end configuration (see “Zoning between IBM Spectrum Virtualize and back-end controllers” on page 15 and “Back-end controller configuration” on page 16).
  - All volumes are online

Be sure all spares configured in the cluster are tested in this way following initial deployment.

## Restrictions

Some restrictions exist with SAS enclosures, enclosure based products, and maximum limits.

### SAS enclosures

Spare nodes are not supported on any cluster where an SAS enclosure is connected to any I/O group. This limitation is not currently monitored and behavior with an SAS enclosure present is unspecified.

### Enclosure based products

Spare nodes are not supported on any enclosure-based product, because of physical hardware limitations.

### Maximum limits

On any cluster, up to four nodes can be assigned as spares. The maximum size of a cluster is twelve nodes: eight originals and four spares.



## Events: Errors, alerts, and messages

Errors, alerts, and message events are described in Table 1.

Table 1 Events and their explanation

Event	Explanation
Error Code 1300: IO port configuration issue	Occurs if the IBM Spectrum Virtualize software is not able to bring online an NPIV target port, despite the physical Fibre Channel port being operational. The usual cause for this error is a switch that does not support NPIV, has NPIV disabled in its configuration, or has limited NPIV resources. Check the switch configuration.
Event ID 88000: An IO port cannot be started	An NPIV target port cannot be started on the original port.
Event ID 88001: NPIV transition stalled	An NPIV target port cannot be started for another reason. Examples might be an incompatibility with the fabric, or another problem with the switch.
Error Code 1380: Spare node configuration issue	Occurs if a spare node is added to the cluster that has a hardware mismatch when compared to the nodes in the cluster. The usual cause for this is either a mismatch of memory or a compression card being available on active nodes and not on spare or vice-versa.
Alert Event ID 088003: A spare node in this cluster is not providing additional redundancy	Occurs when a spare node was added in the cluster that does not provide protection for any active nodes. This event can be corrected either by adding a node in the cluster as an active node or by removing this spare node.
Service error code 3180: Spare node failback required	If the spare node is the only node in the cluster and the failed node that is covered by an online spare comes online, then IBM Spectrum Virtualize will not failback to the active node, generating error code 3180. If the failback is performed manually with the <b>swapnode -failback</b> command, offline volumes can result. The failback will occur automatically, within 10 minutes of redundancy being restored, or when the <b>swapnode -failback</b> command is run.
Event ID 088004: A spare node could not be automatically removed from the cluster	This is the only event ID for error code 3180.

Event	Explanation
Error code 3220: Equivalent ports may be on different fabrics	Occurs if the IBM Spectrum Virtualize software suspects that an NPIV failover will result in an NPIV target port moving from one Fibre Channel fabric to another. Note that this event might generate false positives if two switches on the same physical fabric report a different fabric name, and zoning prevents a node-to-node login through the non-NPIV port. The directed maintenance procedure (DMP) for this error code will allow an override to confirm that the cabling is actually correct, after which IBM Spectrum Virtualize software will note the fabric names reported by the switches involved, and not raise the event again for those fabric names.
Message Event code 088005: A spare node has automatically come online	Occurs if an online node is being protected by a spare node and that node fails then the spare takes its place. This event is also generated when a node is replaced by using the <b>swapnode</b> command.
Message Event code 088006: A spare node has automatically gone back into standby	Occurs when an online spare node is replaced by the original node either by automatic failback or failback using the <b>swapnode</b> command.

## Authors

This paper was produced by a team of specialists from around the world working at the International Technical Support Organization, Hursley Center.

**Alex Ainscow** is a Senior Technical Staff Member in the United Kingdom. He has 14 years of experience in the storage field. He has worked at IBM for 15 years; his areas of expertise include storage virtualization, flash technology, and enterprise storage systems.

**Fiona Crowther** is part of the development team for IBM Spectrum Virtualize working in IBM Manchester. She has a Masters degree in Information Systems from the Robert Gordon University in Aberdeen, Scotland, and has worked as a Software Engineer in IBM for 21 years. After a number of years with IBM WebSphere® in Hursley, she moved to IBM Spectrum Virtualize in the UK Manchester Lab in 2013; she now leads the configuration development team.

**Gareth Jones** is a Software Engineer in the United Kingdom. He has five years of experience in the storage field, working for IBM, and his areas of expertise include storage virtualization, storage enclosure management, and software performance.

**Anil Palled** works for IBM Systems in Pune, India.

**Graham Woodward** works for IBM Systems in Manchester, UK.

Thanks also to these contributors:

Torsten Rothenwaldt  
**IBM ATS, Germany**

Martin Gingras  
**IBM System Storage®, Canada**

This project was managed by:

**Jon Tate** is a Project Manager for IBM System Storage SAN Solutions at the ITSO, San Jose Center. Before joining the ITSO in 1999, he worked in the IBM Technical Support Center, providing Level 2/3 support for IBM mainframe storage products. Jon has 32 years of experience in storage software and management, services, and support. He is an IBM Certified IT Specialist, an IBM SAN Certified Specialist, and is Project Management Professional (PMP) certified. He is also the UK Chairman of the Storage Networking Industry Association (SNIA).

## Now you can become a published author, too!

Here's an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

[ibm.com/redbooks/residencies.html](http://ibm.com/redbooks/residencies.html)

## Stay connected to IBM Redbooks

- ▶ Find us on Facebook:  
<http://www.facebook.com/IBMRedbooks>
- ▶ Follow us on Twitter:  
<http://twitter.com/ibmredbooks>
- ▶ Look for us on LinkedIn:  
<http://www.linkedin.com/groups?home=&gid=2130806>
- ▶ Explore new IBM Redbooks® publications, residencies, and workshops with the IBM Redbooks weekly newsletter:  
<https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm>
- ▶ Stay current on recent Redbooks publications with RSS Feeds:  
<http://www.redbooks.ibm.com/rss.html>



# Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing, IBM Corporation, North Castle Drive, MD-NC119, Armonk, NY 10504-1785, US*

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

## COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

# Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.


HyperSwap®

IBM®

IBM Spectrum™

IBM Spectrum Virtualize™

Redbooks®

Redbooks (logo) ®

Redpaper™

Storwize®

System Storage®

WebSphere®

The following terms are trademarks of other companies:

Other company, product, or service names may be trademarks or service marks of others.





REDP-5477-00

ISBN 0738456616

Printed in U.S.A.

Get connected

