# Dell PowerScale: Non-Disruptive Upgrade Best Practices

June 2024

H17459.3

White Paper

## Abstract

This white paper provides configuration considerations for Dell PowerScale OneFS Non-Disruptive Upgrade (NDU) features including OneFS upgrade and patch upgrade, and covers how NDU can impact different workloads including SMB, NFS, HDFS, FTP, and HTTP.

DELL Technologies

## Copyright

# Contents

# Executive summary

## Overview

This white paper provides configuration considerations and best practices of the Dell PowerScale OneFS Non-Disruptive Upgrade (NDU) including the following:

- Explanation of OneFS NDU mechanism and its general configuration considerations
- Explanation of how OneFS upgrade can impact the client workloads and the best practices, discussing the following workloads:
  - SMB: including SMB1, SMB2 and SMB3 CA
  - NFS: including NFSv3 and NFSv4
  - HDFS
  - FTP
  - HTTP
- Patch upgrade consideration

## Audience

This guide is intended for experienced system and storage administrators who are familiar with file services and network storage administration.

This guide assumes the reader has a working knowledge of the following:

- Network-attached storage (NAS) systems
- The PowerScale scale-out storage architecture and the PowerScale OneFS operating system

The reader should also be familiar with PowerScale documentation resources, including:

- Dell Community Network info hubs
- Dell OneFS release notes, which are available on the Dell support network and contain important information about resolved and known issues.
- Dell PowerScale OneFS Best Practices

## Revisions

| Date | Part number/ revision | Description |
|---|---|---|
| August 2018 | | Initial release |
| April 2019 | | Update to reflect the improvements in OneFS 8.2.0 |
| August 2019 | | Update to reflect the improvements in OneFS 8.2.1 – simplified patch installation and multi- patches installation during OneFS upgrade. |
| January 2020 | | Update to reflect parallel upgrade in OneFS 8.2.2 |
| May 2020 | | Update to reflect rebranding in OneFS 9.0.0 |
| August 2020 | | Update to reflect the improvements in OneFS 9.1.0.0 |

| Date | Part number/ revision | Description |
|---|---|---|
| April 2021 | H17459 | Update to reflect the improvements in OneFS 9.2.0.0 - Drain-based NDU |
| December 2021 | H17459.1 | Update the following areas:<br>• Improvements on firmware upgrade<br>• Firmware combined upgrades |
| April 2022 | H17459.2 | Update for OneFS 9.4.0.0<br>• Signed upgrade |
| June 2024 | H17459.3 | Template update, publishing in HTML |

## We value your feedback

Dell Technologies and the authors of this document welcome your feedback on this document. Contact the Dell Technologies team by email.

**Author:** Vincent Shen

**Note**: For links to other documentation for this topic, see the PowerScale Info Hub.

# OneFS upgrade basics

This section explains the following topics:

- NDU introduction
- Upgrade types including simultaneous upgrade and rolling upgrade
- Supported upgrade path
- Upgrade states and transition

## What is NDU

NDU is a framework which is introduced in OneFS 8.0 to provide better control and predictability to the upgrade process.

From a user perspective, the goal is to provide a seamless and robust upgrade process which is non- disruptive to normal business workflows.

At a high level, the NDU framework implements a state machine, which means at any given time, the PowerScale cluster is at one of the following states:

- Upgrading
- Upgraded
- Rollback
- Committed

NDU supports the state's transition through some user actions. For details of the NDU state machine and its transition, refer to Pre-upgrade check.

At the low level, the NDU framework is implemented by a controlling `Supervisor` process and a servant Agent daemon process on every PowerScale node.

- The `Supervisor` is a short-lived process and it is to assess the current state of the PowerScale cluster and take the appropriate action.
- The `Agent` is a daemon process on every PowerScale node. It is used to take actions on the local node based on received messages from the `Supervisor` process.

---

**Note**: NDU framework is not only used for OneFS upgrade, it is applied to patch and node firmware upgrade as well. This chapter will only focus on the OneFS upgrade. For patch upgrade, see the section Patch upgrade.

---

## Upgrade types

There are three options available for upgrading OneFS:

- Simultaneous upgrade
- Rolling upgrade
- Parallel upgrade

The details of each upgrade type will be explained in the following sections.

### Simultaneous upgrade

A simultaneous upgrade installs the new operating system and restarts all nodes in the PowerScale cluster concurrently.

Simultaneous upgrades are faster than rolling upgrades but require a temporary interruption of service during the upgrade process. All client connections to the cluster must be terminated prior to initiating the upgrade and data is inaccessible until the installation of the new OneFS operating system is complete and the cluster is back online. Based on this, OneFS simultaneous upgrade is a disruptive upgrade path as all of the cluster services will be offline during the upgrade process.

### Rolling upgrade

A rolling upgrade individually upgrades and restarts each node in the PowerScale cluster so that only one node is offline at a time.

A rolling upgrade takes longer to complete than a simultaneous upgrade. You can specify the order in which nodes are upgraded by using the `--nodes` parameter of the `isi upgrade cluster start` command. The `-nodes` parameter can also be used in the scenario to upgrade a specific subset of nodes. The following example command starts a rolling upgrade on logical node number (LNN) 1, 3 and 5 in that order:

```
isi upgrade cluster start <install-image-path> --nodes 1,3,5
```

The following example commands use a dash-separated range to upgrade LNN 1 to node 5

```
isi upgrade cluster start <install-image-path> --nodes 1-5
```

It is required to upgrade all the nodes in order to install a patch, do a node firmware upgrade and do the next OneFS upgrade. If you only upgrade several nodes in the cluster, a weekly alert is sent to confirm that the upgrade is making progress if you have subscribed to the corresponding alert channel. Do not leave the cluster in a partially upgraded state for a prolonged period. Some new features in the upgrade might not be available until all the nodes in the cluster have been upgraded and the upgrade is committed. Refer to the release notes for the OneFS version that you are upgrading to for information about features that require the cluster to be committed to the upgraded version of OneFS.

To add new nodes to a running upgrade process, use the following command:

```
isi upgrade cluster add-nodes -nodes=2,4,6
```

To add all the remaining nodes to an upgrade process, use the following command:

```
isi upgrade cluster add-remaining-nodes
```

If you do not specify an order, nodes are upgraded in ascending order from the node with the lowest Array ID to the node with the highest Array ID. Because Array IDs are never reused, a node's Array ID might not be the same as the node's logical node number (LNN). To check each node's Array ID, run the following command:

```
isi_nodes "%{name}: LNN %{lnn}, Array ID %{id}"
```

A typical outcome of the above command is as below, and in this case, LNN matches Array ID.

```
tme-sandbox-1: LNN 1, Array ID 1
tme-sandbox-2: LNN 2, Array ID 2
tme-sandbox-3: LNN 3, Array ID 3
```

**Important**: During a rolling upgrade, nodes that are not actively being upgraded remain online and can continue serving clients. However, clients that are connected to a restarting node are disconnected and reconnected. How the client connection behaves when a node is restarted depends on several factors including client workload type, client configuration (mount type, timeout settings), IP allocation method, and how the client connected to the cluster. Usually, NDU requires specific configurations on either the PowerScale side or the client-side. For detailed client behavior and the recommended configurations, see the section Client behavior in an upgrade.

Rolling upgrades are not available between all OneFS versions. See the section Supported upgrade path for information about which types of upgrades are supported between OneFS versions.

## Parallel upgrade

The parallel upgrade is introduced in OneFS 8.2.2. It provides some extent of parallelism which is to upgrade at most one node per neighborhood at any time. By doing that, it can reduce upgrade duration and ensure that the end-user can still continue to have access to their data.

As shown in Figure 1, this feature can be enabled through WebUI.

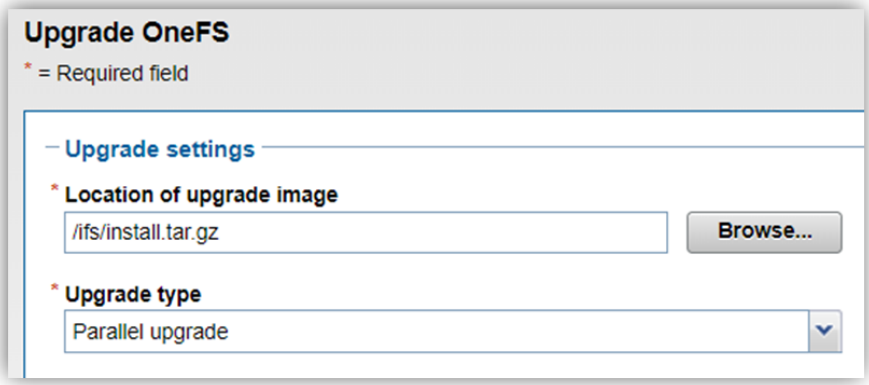


**Figure 1. OneFS parallel upgrade**

You can also leverage the CLI to enable this feature through OneFS upgrade:

```
# isi upgrade start --parallel /ifs/install.tar.gz
```

The parallel upgrade can dramatically improve the OneFS upgrade efficiency without impacting the data availability. You can use the following formula to estimate the duration of the parallel upgrade:

$$Estimation\ time = (per\ node\ upgrade\ duration) \times (highest\ number\ of\ nodes\ per\ neighborhood)$$

In the above formula:

- The first parameter – **per node upgrade duration**, it is 20 minutes on average.
- The second parameter – **the highest number of nodes per neighborhood**, you can get the number by running the following command:

```
#sysctl efs.lin.lock.initiator.coordinator_weights
```

The following is an example:

In a 150 node PowerScale cluster, ideally, there are 15 neighborhoods with 10 PowerScale nodes each. Neighborhood 1st with node number 1 to 10 and Neighborhood 2nd with node number 11 to 20 and so on.

During the parallel upgrade, the upgrade framework will pick at most one node from each neighborhood, to run the upgrading job simultaneously. So in this case, node 1 from neighborhood 1st, node 11 from neighborhood 2nd, node 21 from neighborhood 3rd, and so on, will be upgraded simultaneously. Considering, they are all in different neighborhoods or failure domain, it will not impact the current running workload. After the first pass completes, it will go to the 2nd pass and then 3rd, and so on.

So, in this example the estimated duration of the parallel upgrade is 200 minutes:

$Estimation\ time$ = ($per\ node\ upgrade\ duration$) × ($highest\ number\ of\ nodes\ per\ neighborhood$) = 20 × 10

= 200 $minutes$

## Supported upgrade path

The supported upgrade path is a matrix by which it is easy to see which OneFS releases can be upgraded to from a given source release. This matrix applies to both simultaneous and rolling upgrades - in other words, an upgrade is either possible or not, regardless of the upgrade method chosen. **Error! Reference source not found.** is an example of a high-level matrix of the supported upgrade path from 8.0.0.

Starting from 8.0.0, the supported upgrade path follows the rule below:

Consider a source release version in the format of w.x.y.z. The rule is that a simultaneous or rolling upgrade is supported to any version up to and including x+2, where the w digit either remains the same or increases by one step only.

---

**Note**: starting from OneFS 8.2.0, the release version is in a 3-digit format such as w.x.y.

---

This document will only focus on the upgrade path starting from OneFS 8.0.0. For a detailed supported upgrade path for all OneFS releases, see OneFS Upgrade Paths.

Key:

- "O" : Rolling and Simultaneous upgrades available
- "=" : Parallel, Rolling, and Simultaneous upgrades available
- "x" : Unsupported upgrade path
- "–": For upgrades from OneFS 9.0.0 and earlier, contact your Dell Technologies account team.

Table 1. OneFS Upgrade Paths

| UPGRADE FROM CURRENT VERSION | UPGRADE TO TARGET VERSION | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 9.0.0 | 9.1.0 | 9.2.0 | 9.2.1 | 9.3.0 | 9.4.0 | 9.5.0 | 9.6.0 | 9.7.0 | 9.8.0 |
| 9.0.0 | – | = | = | = | = | = | = | x | x | x |
| 9.1.0 | – | | = | = | = | = | = | x | x | x |
| 9.2.0 | – | | | = | = | = | = | x | = | = |
| 9.2.1 | – | | | | = | = | = | x | = | = |
| 9.3.0 | – | | | | | = | = | x | = | = |
| 9.4.0 | – | | | | | | = | x | = | = |
| 9.5.0 | – | | | | | | | x | = | = |
| 9.6.0 | – | | | | | | | | = | = |
| 9.7.0 | – | | | | | | | | | = |
| 9.8.0 | – | | | | | | | | | |

There are some special upgrade support matrix and path defined. For a complete list of OneFS upgrade paths, see OneFS Upgrade Paths.

## Pre-upgrade check

It is recommended to have pre-upgrade checks before performing an actual upgrade job. At the time of writing, there are three tools for this purpose:

- IOCA
- CLI for upgrade assessment (`isi upgrade cluster assess`)
- Health Check Framework

## Upgrade state and transition

From OneFS 8.0.0, the entire NDU process can be described by a PowerScale cluster upgrade state machine as demonstrated in Figure 2. This means a PowerScale OneFS cluster is in exactly one of the following upgrade states at any given time:

- Committed
- Upgrading
- Upgraded
- Rollback

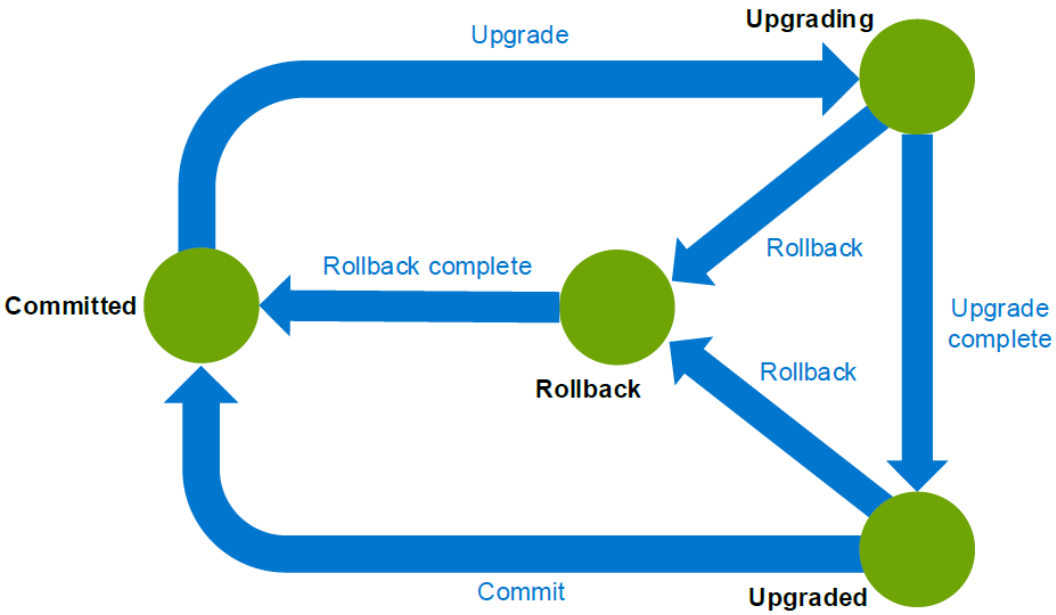


**Figure 2. PowerScale cluster upgrade state and transition paths**

The state will change as it moves through the NDU cycle based on a set of allowed transitions. In Figure 2, transitions are the blue arrows between two upgrade states:

- Upgrade
- Upgrade complete
- Rollback
- Rollback complete
- Commit

Table 1 lists all the details of each OneFS upgrade states and how they can switch from one to another through transitions.

**Table 1. PowerScale cluster upgrade states and transition paths details**

| PowerScale cluster status | Description |
|---|---|
| Committed | • A previous upgrade operation has been completed and committed.<br>• All nodes are running the same version of OneFS and all features of that version are available. Rollback to the previously installed version is not available.<br>• The cluster is ready to start another OneFS upgrade when required.<br>• A cluster remains in this state until another upgrade is initiated.<br>• This is considered the steady state of a cluster, and it is expected that a cluster over its life cycle will spend most of its operational time in this state. |

| PowerScale cluster status | Description |
|---|---|
| Upgrading | • At least one PowerScale node has started upgrading to the target release version.<br>• The required information to roll back to the source release is maintained while the cluster is in Upgrading state.<br>• A cluster remains in Upgrading state until either all nodes are upgraded to the target release, or a rollback is initiated.<br>• In Upgrading state, the cluster is running in mixed mode, as there are now two versions of OneFS present in the cluster.<br>• Nodes which have already upgraded may be able to access some of the functionality of the new release.<br>• Nodes which have not been upgraded cannot access any new release functionality. |
| Upgraded | • All nodes are now running the target release version; however, the upgrade has not been Committed.<br>• The required configuration to roll back to the source release is maintained while the cluster is in an Upgraded state.<br>• A cluster in the Upgraded state can run any new functionality of the target release. |
| Rollback | • The cluster is in the process of rolling back a OneFS upgrade.<br>• Rollback can be initiated by the administrator on a cluster in either the Upgrading or Upgraded state.<br>• Once the upgrade is committed, rollback is no longer available.<br>• In the Rollback state, the cluster restores the saved information associated with the source release and prepares the nodes to reboot to the original source release version. Once the nodes have rebooted, the cluster transitions automatically to the Committed state.<br>• Rollback is available for both rolling and simultaneous upgrades. A cluster can be rolled back only to the previously installed release.<br>• This state should be considered a transition state. Clusters should not be run in this state for extended periods of time. |

To check out the current cluster upgrade state and activity, use the following CLI command:

```
isi upgrade cluster view
```

or

```
isi upgrade view
```

An example of the output from the command above is as below, which indicates the PowerScale cluster is in the committed state:

```
Upgrade Status:

   Cluster Upgrade State: committed
Current Upgrade Activity: -
      Upgrade Start Time: 2018-08-09T07:22:15
   Upgrade Finished Time: 2018-08-14T06:09:35
      Current OS Version: 8.1.0.4_build(57)style(5)
      Upgrade OS Version: N/A
        Percent Complete: 0%

Nodes Progress:

     Total Cluster Nodes: 3
       Nodes On Older OS: 3
          Nodes Upgraded: 0
Nodes Transitioning/Down: 0

LNN          Progress    Version     Status
- - - - - - - - - - - - - - - - - - - - - - -
1            0%          8.1.0.4     committed
2            0%          8.1.0.4     committed
3            0%          8.1.0.4     committed
```

## Pause and resume

Starting from OneFS 8.2.0, a OneFS upgrade can be paused and resume. This usually happens when customers reach the end of a maintenance window and they can pause the upgrade and resume in a later window. To pause a running OneFS upgrade process, run the following command:

```
isi upgrade pause
```

After this command is triggered, upgrade status will be in a **Pausing** status until the current upgrading node is completed. The remaining nodes will not be upgraded until the upgrade process is resumed.

To resume a paused OneFS upgrade process, run the following command:

```
isi upgrade resume
```

To check the **Pausing/Paused** status, use the following CLI command:

```
isi upgrade view
```

Or, to view the PAUSE file data by using the following command:

```
cat /ifs/.ifsvar/upgrade/processes/upgrade/PAUSE
```

A typical output which indicates the PowerScale cluster is still in pausing status is as below:

```
{
"PauseState": "Pausing"
}
```

## Pre-package OneFS image with roll-up patches (RUPs)

Dell provides the OneFS installation bundle for:

- Upgrading
- Reimage
- Manufacture loading

Each bundle contains the OneFS image and the most recent RUPs that have been verified by Dell. We support any types of the RUP except the node firmware package (NFP) and drive support package (DSP). The specific supported types of RUP are:

- GA RUP (including Kernel GA RUP and User-space GA RUP)
- Singleton patch
- HealthCheck RUP

With the most recent RUP, OneFS can provide the most enhanced reliability and stability. You can download these bundles for multiple OneFS releases including:

- OneFS 8.1.2
- OneFS 8.2.0
- OneFS 8.2.1
- OneFS 8.2.2
- OneFS 9.0.0
- OneFS 9.1.0.0 and the following versions

The installation process is the same as the one that does not include RUPs. It supports rolling upgrade, simultaneous upgrade, and parallel upgrade. It also supports rollback operations.

In OneFS 9.1.0.0, this feature provides the following improvements:

- Support the CLI parameter '--patch-paths'
- Support precheck operation

These new features are explained in the following sections.

### Support the CLI parameter '--patch-paths'

Starting from OneFS 8.2, installing patches during a OneFS upgrade is supported by using the newly introduced parameter '--patch-paths'.

OneFS 9.1.0.0 supports using the '--patch-paths' parameter along with the pre-package OneFS image with RUPs.

In this case, the embedded RUP and the user input patches using '--patch-paths' may have conflicts. To deal with these conflicts, internally we have some complicated rules defined. However, in order to simplify these rules, see the following typical cases:

**Table 2. Typical cases for upgrade conflicts**

| Scenarios | Upgrade From | Upgrade image | --patch-paths | Compatibility | Outcome |
|---|---|---|---|---|---|
| Scenario I | OneFS 8.2.2 | Pre-package image:<br>OneFS 9.1.0.0 + RUP v1 + KRUP v1 | RUP v2 and KRUP v2 | Yes | OneFS 9.1.0.0 + RUP v2 + KRUP v2 |
| Scenario II | OneFS 8.2.2 | Pre-package image:<br>OneFS 9.1.0.0 + RUP v2 + KRUP v1 | RUP v1 and KRUP v2 | No | OneFS 9.1.0.0 + RUP v2 + KRUP v2<br>A CELOG event raised |
| Scenario III | OneFS 8.2.0 | Pre-package image:<br>OneFS 9.1.0.0 + RUP v2 | RUP v1 | No | OneFS 9.1.0.0 + RUP v2<br>A CELOG event raised |

### Support precheck operation

Dell Technologies recommends using precheck operation to run the preliminary check between embedded RUP and user input patches to ensure that there is no conflict before starting an upgrade.

### Group change time to one second

Group change happens during the OneFS upgrade and this process can impact the protocol latency in the upgrade process. In OneFS 9.1.0.0, this period of time has been highly improved to only one second to help better data availability.

## Drain-based upgrade

### Overview

During an upgrade workflow, nodes will get a reboot or the protocol service must be stopped temporarily. This leads to a quick disruption to the clients connected to the rebooting node. This feature provides a mechanism by which nodes are prevented from rebooting or restarting protocol services until all SMB clients have disconnected from the node. A single SMB client that does not disconnect can cause the upgrade to be delayed indefinitely and so the user is provided with options to reboot the node despite persisting clients.

The drain-based upgrade supports the following scenarios and is available for WebUI, CLI, and PAPI.

- SMB protocol
- OneFS upgrades
- Firmware upgrades
- Cluster reboots
- Combined upgrades (OneFS and Firmware)

The drain-based upgrade is built upon the parallel upgrade workflow which is introduced in OneFS 8.2.2.0 which offers parallel node upgrade and reboot activity across node neighborhoods. It upgrades at most one node per neighborhood at any time. By doing that, it can reduce upgrade duration and ensure that the end- user can continue to have access to their data. The more node neighborhoods within a cluster the more parallel activity can occur.

Figure 3 shows an example of how it works. In this case, we assume there are two neighborhoods in a given 6 node PowerScale cluster. Node 1 to node 3 belongs to neighborhood 1 and Node 4 to node 6 belongs to neighborhood 2. You can use the following command to identify the correlation between your PowerScale nodes and neighborhoods (failure domains):

```
# sysctl efs.lin.lock.initiator.coordinator_weights
```

Once the drain-based upgrade is started, at most one node from each neighborhood will get the reservation which allows the nodes to upgrade simultaneously and OneFS will not reboot these nodes until the number of SMB clients is “0”. In this example, Node 3 and Node 4 get the reservation for upgrading simultaneously.

However, there is 1 SMB connection for Node 3 and two SMB connections for Node 4. They cannot reboot until the SMB connections get to “0”. At this stage, there are three options:

- Wait: Wait until the SMB connections get to “0” or it reaches the drain timeout value. The drain timeout value is the configurable parameter for each upgrade process. It is the maximum waiting period. If the drain timeout is set to “0”, it means wait forever.
- Delay drain: Add the node into the delay list to delay client draining. The upgrade process will continue on another node in this neighborhood. After all the non-delayed nodes are upgraded, OneFS will rewind to the node in the delay list.
- Skip drain: Stop waiting for clients to migrate away from the draining node and reboot immediately.

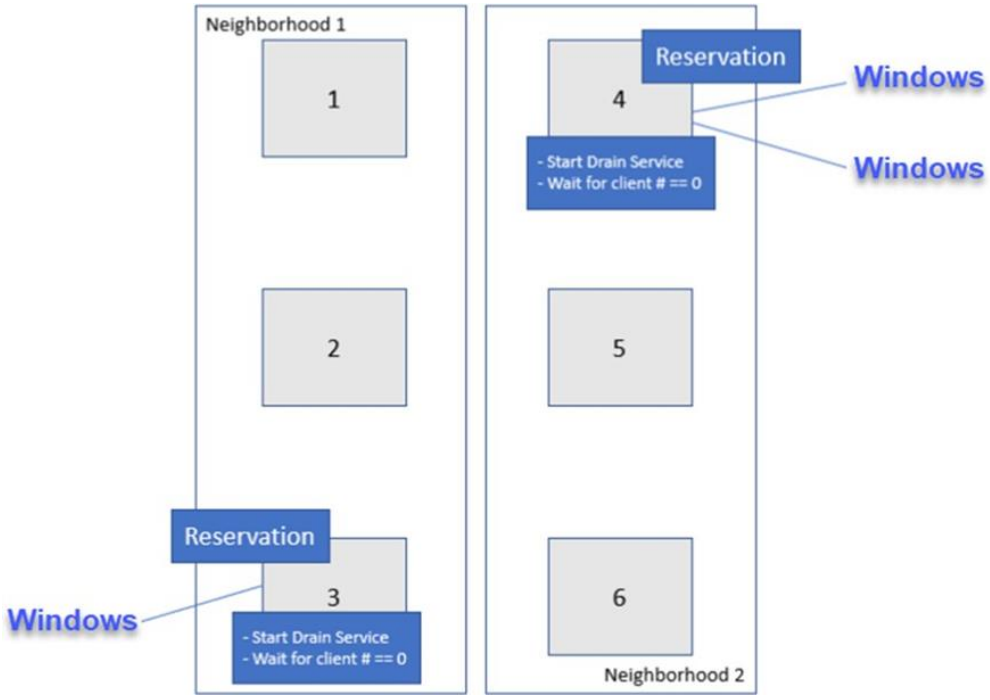


**Figure 3. Example of drain based upgrade**

## Perform drain-based upgrade

1. Run the following command to perform the drain-based upgrade. In this example, we have set the drain timeout value to 60 minutes and the alert timeout value to 45 minutes, which means if there is still some connection after 45 minutes, a CELOG notification will be triggered to the administrator.

```
# isi upgrade start --parallel --skip-optional --install-image-path=/ifs
/data/<installation-file-name> --drain-timeout=60m --alert-timeout=45m
```

2. Wait for a while, in OneFS WebUI, browse to **Upgrade** under **Cluster management**, in this window you will see the node waiting for draining clients. You can either try the action of **Skip** or **Delay**. In this case, **Skip** is selected as shown in Figure 4. In the prompt window click the button **Skip Draining**.

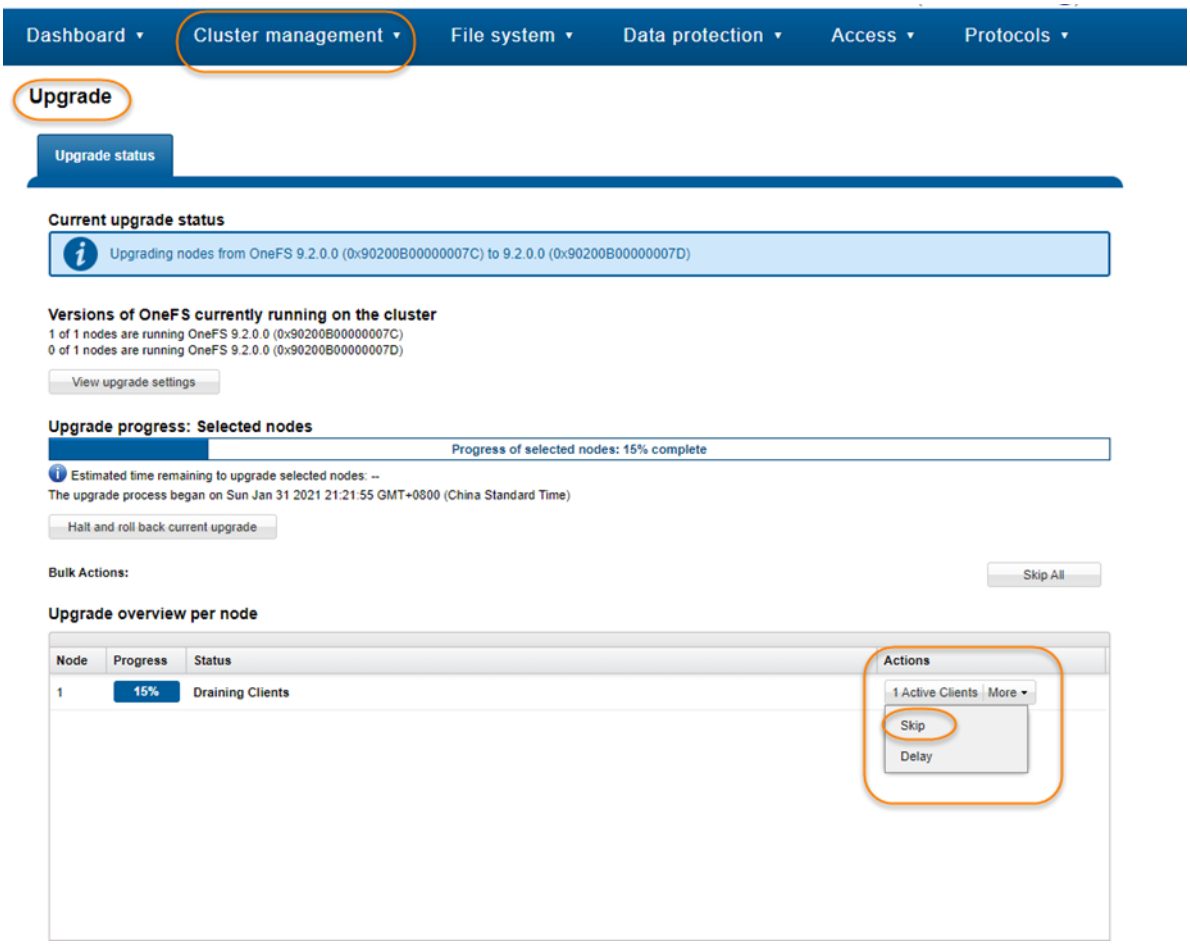


**Figure 4. Skip the draining clients**

## Signed upgrade

### Overview

From OneFS 9.4.0.0, we introduced the signed upgrades feature. By using this, all upgrades like OneFS, firmware, and patches must be signed before they can be installed using the upgrade framework. The following are the new components of this feature:

- A new .isi package format. This format applies to all the packages of OneFS upgrade, patch, firmware, and DSP.
- A new OneFS Catalog. This component has been added to store signed packages on the cluster for use by the upgrade framework. It is designed to protect customers against malicious use of upgrade packages to compromise cluster security.

### the .isi format

The Integrated Signature Install package (.isi) format is designed to enable a customer to download a package, verify its authenticity, review README text and install it on the cluster without needing to unpack anything. It also enables automatic verification and enforcement of signed packages by the upgrade framework.

Starting a signed upgrade is similar to starting an upgrade in previous versions with one exception. All upgrades must be started using a signed .isi package in place of the usual .tar or .pkg file.

The following are a few examples:

```
# isi upgrade start --install-image-path=/ifs/install.isi --parallel
# isi upgrade firmware start --fw-pkg /ifs/firmware.isi --parallel
# isi upgrade patches install /ifs/patch.isi --parallel
# isi_dsp_install /ifs/drive_firmware.isi
```

A .isi file is essentially a tarball file with a security footer as shown in Figure 5. The footer contains a manifest of data regions, cryptographic signature, cryptographic timestamp and a few other values used for verifying the file integrity. The manifest is a list of data regions within the file where the main package tarball is in region 1 and may be followed by other regions that can contain README text data or additional upgrade packages.

The first region is special because it can be accessed by older OneFS versions that are not aware of the .isi format, provided that the data is in tar format or another format that can handle unexpected trailing data.

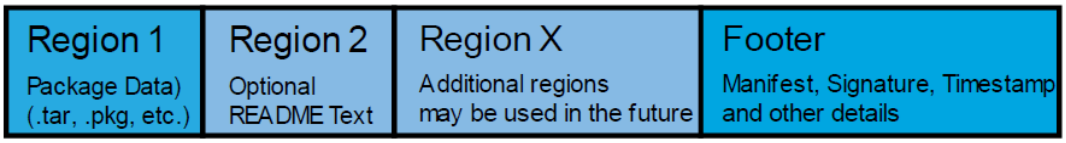


**Figure 5. .isi file format**

## OneFS Catalog

The OneFS Catalog is the repository to store .isi files on the cluster that have been previously verified. Verified items in the catalog are called "artifacts". New files that are added to the catalog will be copied to the temp directory /ifs/.ifsvar/catalog/pending for verification. Once verified, the artifact will be stored in /ifs/.ifsvar/catalog/artifacts with a filename that corresponds to the SHA256 hash of the .isi file. Also, a database entry will be added to /ifs/.ifsvar/catalog/catalog.db.

The catalog serves two main functions:

The first is to provide a location that is only accessible by the root account to store files that have been verified so that they can be installed without worrying about tampering from non-privileged users. All non-root accounts including admin, compadmin or users in the wheel group do not have direct access to the catalog directory.

The second purpose of the catalog is to provide a PAPI interface so that non-privileged users can run approved functions on items in the catalog directory. It is possible for users with the appropriate RBAC role to import, export, verify, and remove items in the catalog. If the catalog directory, contents, or database is damaged, there is also a repair function to correct the issue without the user needing any directory access to the directory. This is especially important on compliance mode clusters where root access is not available.

Users can interact with the catalog using the "isi upgrade catalog" command set or related PAPI handlers. To verify the package signature, use the following command:

```
# isi upgrade catalog verify --file=/ifs/install.isi
Item  Verified

--------------------------
```

```
/ifs/install.isi True
--------------------------
Total: 1
```

To have a deep insight of the package details, use the following command:

```
# isi_packager view --package=/ifs/install.isi
== Region 1 ==
Type: OneFS Install Image
Name: OneFS_Install_0x904005000000001_B_9_4_0_001(RELEASE)
Hash: 35a0901e35f7f3fa6a15c8bde7a91fc575855b164a83ed4938e69d70fd758069
Size: 361651596

== Footer Details == Format Version: 1
 Manifest Size: 296
Signature Size: 3068
Timestamp Size: 1374
 Manifest Hash: 8baf4591f2f68a2a31980c58d243a5781a1e0afaf122881360529b70ffc6ecb7
Signature Hash: 7cf6efb229847e6f7bb5ca52c3810d4c53325b0258f63060a7df6cfbd86e5ac1
Timestamp Hash: 47c653e9db3fef41146ad1cd4cdfbcee953c30add5febc6d0850ec56d9b0b0e7

== Signature Details ==
Fingerprint:    37c57297522afcd5890dfc24990a944598e69077
Issuer:         Entrust Code Signing CA - OVCS2
Subject:        US / Texas / Round Rock / Dell Technologies Inc.
Organization:   Dell Technologies Inc., Powerscale OneFS
Expiration:     2022-08-17 17:44:43
Ext Key Usage:  codeSigning
```

To import a signed package, use the following command. A package will automatically be verified when it is imported.

```
# isi upgrade catalog import --file /ifs/export.isi
```

Use the following list command, to view the ID, type, description and whether it contains the README file of a signed package.

```
# isi upgrade catalog list
ID    Type  Description                    README
-----------------------------------------------------
a75fb OneFS OneFS 9.4.0.0_build(1)style(5) -
-----------------------------------------------------
```

To export the artifact in the catalog to the OneFS file system, use the following command:

```
# isi upgrade catalog export --id=a75fb /ifs/export.isi
```

To view the README file or the release notes of the signed package, use the following command:

```
# isi upgrade catalog readme --file HealthCheck_9.2.1_2021-09.isi | less Updated:
September 02, 2021
```

```
********************************************************************************
HealthCheck_9.2.1_2021-09: Patch for OneFS 9.2.1.x.
This patch contains the 2021-09 RUP for the Isilon HealthCheck System
********************************************************************************
This patch can be installed on clusters running the following OneFS version:
* 9.2.1.x
```

# Client behavior in an upgrade

This section explains the behavior of different workloads during a OneFS rolling upgrade including the following workloads:

- NFS
- SMB
- HDFS
- HTTP
- FTP

For each workload, this section includes best practices and configurations for NDU consideration.

## NFS behavior and configuration consideration

This section explains how the PowerScale OneFS upgrade process can impact the NFS workloads including both NFSv3 and NFSv4.

**Note**: NFS version 2 is not supported in OneFS 7.2.0 and above. Due to this reason, it is not in this white paper.

Before explaining how the PowerScale OneFS NDU process can impact the NFS workloads, it is important to understand the following three points:

- PowerScale dynamic IP pool for NFS workloads
- NFS recovery or retry mechanism
- Performance impact

Best practices are in the conclusion of this section.

### PowerScale dynamic IP pool for NFS workloads

#### *NFSv3 with dynamic IP pool*

Dynamic IP pools assign out all the IP addresses within a given range to all the available NICs across the entire PowerScale cluster. Dynamic IP addresses can move from one NIC to another, when a node goes to an unhealthy state. This ensures that dynamic IP addresses are always available during failover and failback. For a stateless protocol like NFSv3, the best practice is to use a dynamic IP pool for business continuity.

During the OneFS NDU process, if the rolling upgrade is selected, it will individually upgrade and restart each node in the PowerScale cluster so that only one node is offline at a time. Once a node is offline, the IP address of this node will move to one of the remaining available nodes by using the dynamic IP pool.

As shown in 0, in a 4 nodes PowerScale cluster, once node 1 is offline, both of the dynamic IPs on node 1 will move to the remaining nodes to ensure the business continuity. If the NFS clients use 192.168.200.241 as the NFS server IP to mount NFS exports, during the node 1 offline, it is actually accessing node 2 in the PowerScale cluster and this is transparent to the NFS clients.

**Important**: This will introduce a noticeable pause of the NFS workload. Usually, it only takes less than 20 seconds, which is the amount of time that it usually takes the network ARP cache to flush. This NFS workload pause only happens in the clients which connect to the PowerScale node being rebooted. The other clients will not be affected. In this period of time, you will see the throughput between the NFS client and the NFS server is 0. And after that, it will restore automatically.

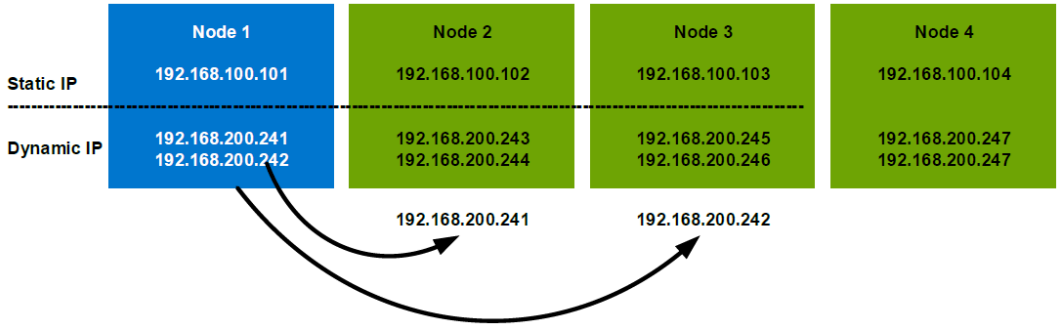


**Figure 6. Dynamic IP example**

Figure 7 is an example of how it works in the rolling upgrade: NFS is mounted at 192.168.200.241. After initiating the OneFS rolling upgrade, node 1 reboots first and causes the NFS mount IP to move from Node 1 to Node 2, which will introduce a noticeable pause of NFS workload. Then, Node 2 starts to upgrade and follows a reboot. This will introduce another IP reallocation from Node 2 to Node 3 and a second short pause. In this case, during the rolling upgrade process, there will be 4 short pauses in total. This is for a 4 nodes PowerScale cluster. If it is a large cluster, the interruptions will be much more frequent.

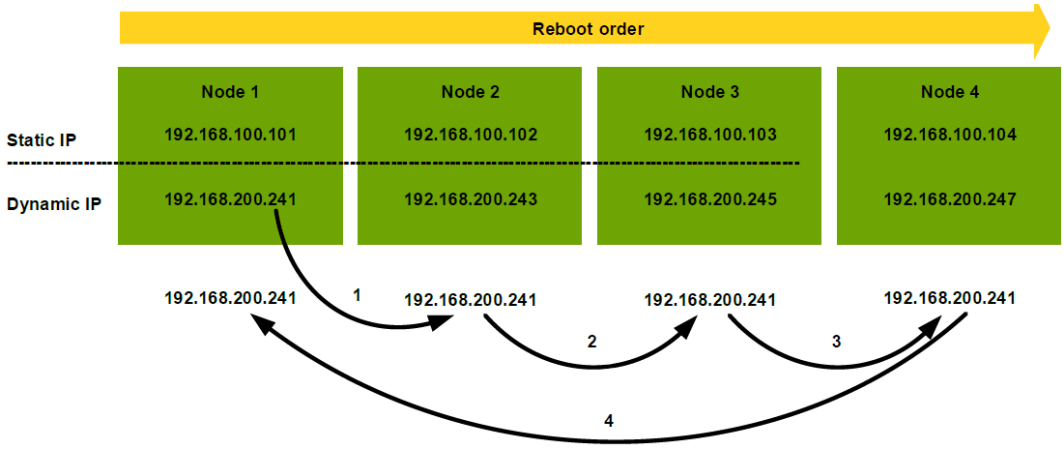


**Figure 7. Worst case of OneFS rolling upgrade**

The above example is the worst case which tends to be unlikely to happen. This is because from OneFS 8.0, when performing NFS failover using SmartConnect dynamic IP,

we tend to favor nodes that are already upgraded. For example, on a three-node PowerScale cluster with 8 IPs in a dynamic pool, if PowerScale node 3 has been upgraded and node 1 and node 2 have not, OneFS will rebalance IPs so that node 3 will have 4 IPs and at the same time node 1 and node 2 only have two IPs each. This awareness reduces the overall rolling upgrade process to the overall services.

On the other hand, we have multiple dynamic IPs per pool to spread the load across multiple nodes to mitigate the impact. Determining the number of IP addresses within a dynamic allocation pool varies depends on the PowerScale node count, the estimated number of clients that would be in a failover event and so on. For detailed best practices, refer to PowerScale Network Design Considerations.

#### *NFSv4 with dynamic IP pool*

NFSv4 is a stateful protocol and in this case, it expects the NFS server to maintain session state information. This means each PowerScale node runs its own NFS daemon and the session information is unique per node. For this reason, we usually recommended using PowerScale static IP pool for NFSv4. However, beginning in PowerScale OneFS 8.0, the NFSv4 session state information is kept in sync across multiple nodes. In the OneFS 8.0 and later, it is recommended to use a dynamic IP pool for NFSv4 connections.

In the case where a static IP pool is applied for NFSv4 workloads, there will be a much longer time during which NFS clients will not receive any response from a PowerScale node. This is because static IPs cannot move among the interfaces. Clients that cannot communicate to the specific PowerScale node may receive an “NFS server not responding” message until the PowerScale node comes back online. In some cases, the NFS client may timeout.

### NFS recovery/retry mechanism

The behavior of NFS recovery is determined by several NFS mount options as below. These mount options apply to both NFSv3 and NFSv4.

#### *timeo=n*

The `timeo` is measured in deciseconds (tenths of a second) and it means how long the NFS client waits for a response before it retries an NFS request. In this period of time, NFS clients will see an “NFS server not responding” response.

For NFS over TCP the default `timeo` value is 600 (60 seconds). As shown in Figure 8, the NFS client performs linear backoff algorithm for timeout value, which means after each retransmission the timeout is increased by `timeo` up to the maximum of 600 seconds. Figure 8 shows an example where `timeo` equals 600.

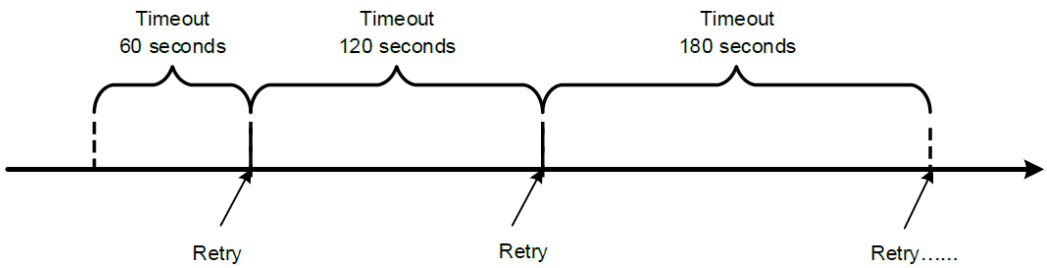


**Figure 8. An example of an NFS timeout linear backoff algorithm (timeo=600)**

### *retrans=n*

The `retrans` is the number of times the NFS client retries a request before it attempts further recovery action. If the `retrans` option is not specified, the NFS client tries each request three times. Figure 9 shows an example of retrains equal to 2 and timeo equaling to 600.

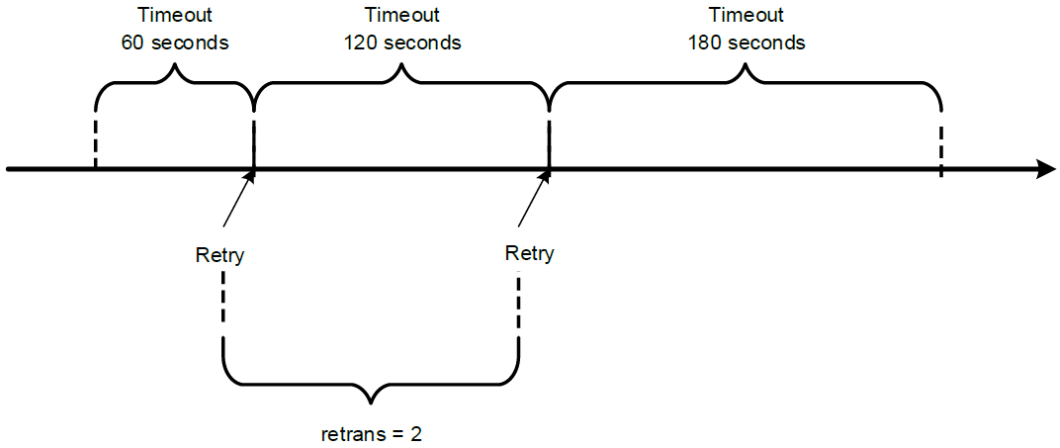


**Figure 9. An example of NFS retrans = 2**

### *Soft/hard mount*

The soft or hard mount option determines the recovery behavior of the NFS client after an NFS request times out as described in Table 3. For most clients, Dell Technologies recommends using the hard mount option and avoid soft mount.

**Table 3. Soft and hard mount options**

| Mount type | Description |
|---|---|
| Hard (or not specify) | After an NFS request timeout, it will attempt to retry and NFS requests are retried indefinitely. |
| Soft | Once an NFS request timeout, it will attempt to retry. But after `retrans` retransmissions have been sent, the NFS client fails an NFS request, causing the NFS client to return an error to the calling application. For example if `retrans` equals 2, the NFS client will return an error after two attempts to retry. This example is also shown in Figure 10. |

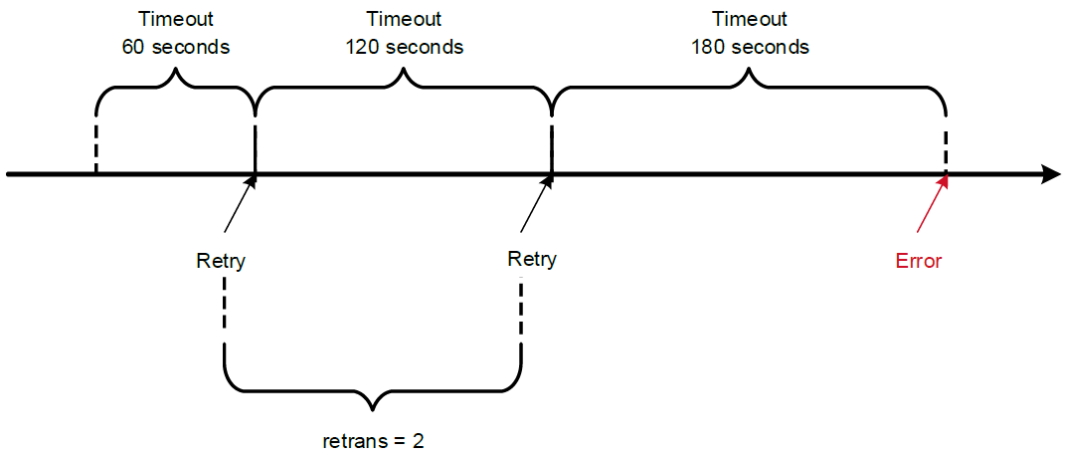


**Figure 10. An example of soft mount failure (retrains = 2)**

It is obvious to see how the client behaves during the noticeable pause in a rolling upgrade is determined by the above three mount options. The detailed explanation is as the following:

- In the case of a hard mount, because the NFS client request will attempt to retry indefinitely, there will be no error message in the NFS layer during the noticeable pause in a rolling upgrade process.

---

**Note**: Although in the NFS layer, there will be no errors and the NFS client will try to retry indefinitely for hard mount, some applications may still encounter errors and this depends on how the application is implemented. Consult your application vendors for this situation.

---

- In the case of a soft mount, if the noticeable pause ends in the green area as shown in 0, there will be no error message in the NFS client application. If the noticeable pause ends beyond the green area as shown in Figure 12, the NFS client will send an error message during the OneFS rolling upgrade process. Usually, it recommends using hard mount instead of using soft mount

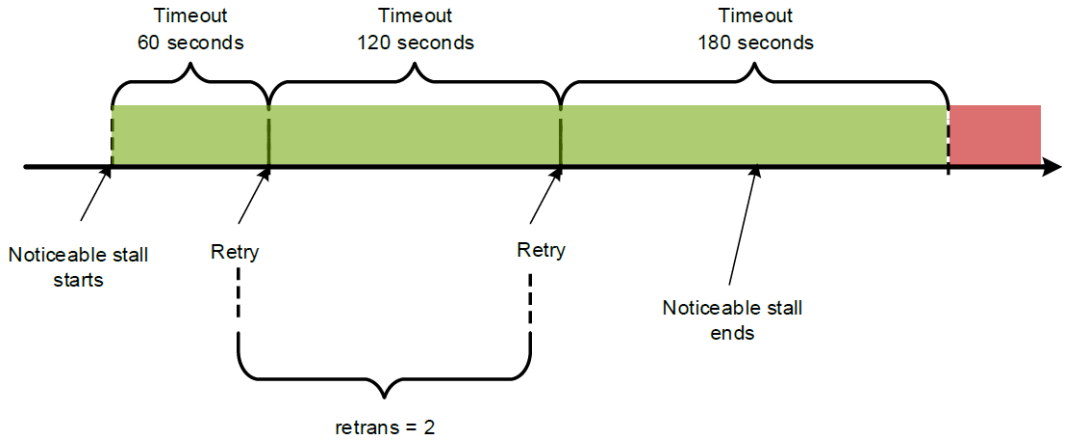


**Figure 11. The noticeable pause within the timeout range**

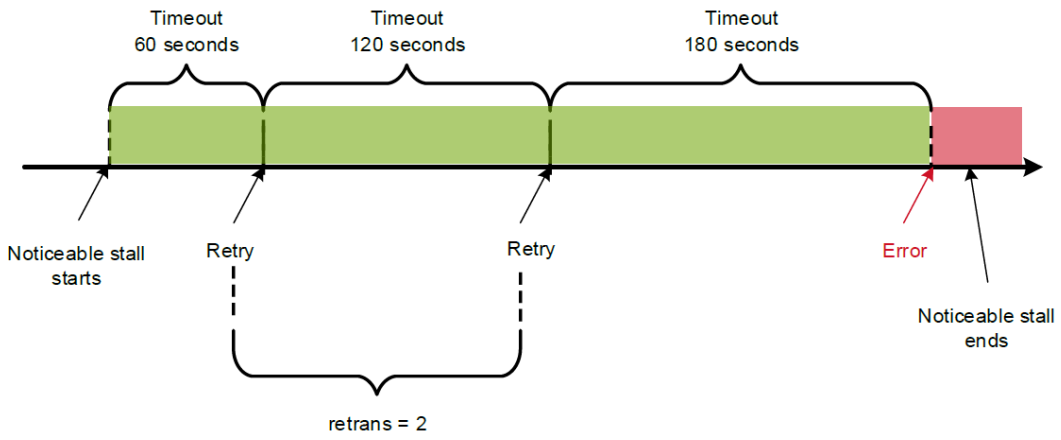


**Figure 12. The noticeable pause beyond the timeout range**

### Performance impact

Dell Technologies recommends all non-disruptive upgrades be performed at a time of low I/O. This is identified as the target maintenance window. If you perform the OneFS rolling upgrade during the maintenance window, you will see minimal performance impact during the overall process.

In case OneFS rolling upgrade is initiated at a time when the cluster is under heavy workload, you will see limited performance impact due to the PowerScale node reboots, since you now have (n-1) PowerScale nodes in the cluster to serve the workload during the reboot time. Performance impacts will be lessened as PowerScale cluster size increases.

### NDU best practices concluded for NFSv3/v4

With the knowledge of the section PowerScale dynamic IP pool for NFS workloads, and the section NFS recovery/retry mechanism, we can conclude the following NDU best practices for NFSv3/v4:

- Use PowerScale dynamic IP pool for NFSv3.
- Use PowerScale dynamic IP pool for NFSv4, if the OneFS version is 8.0 and above
- Leverage SmartConnect multiple dynamic IPs and SSIP to spread the load across multiple nodes to mitigate the impact of the OneFS rolling upgrade process.
- Use the NFS hard mount option and the default NFS mount option is good enough for NDU consideration.

## SMB behavior and configuration consideration

This section explains how the PowerScale OneFS upgrade process can impact the SMB workloads including:

- SMB1 and SMB2
- SMB3 with continuous availability (CA)
- Best practices concluded

### SMB1 and SMB2: always disruptive

SMB is a stateful protocol which means it maintains a session state for all the open files in the PowerScale node where the client connects to. This session state is not shared

across the nodes. For a stateful protocol like SMB, it is recommended using OneFS static IP pools. But in certain workflows, SMB is preferred to use a dynamic IP pool. SMB preserves complex state information per session on the server side. If a connection is lost and a new connection is established with dynamic failover to another node, the new node may not be able to continue the session where the previous one had left off. If the SMB workflow is primarily reads, the impact of a dynamic failover will not be as drastic, as the client can re-open the file and continue reading. Conversely, if an SMB workflow is primarily writes, the state information is lost and the writes could be lost as well.

By using the static IP pool, IP addresses assigned to the node will not reallocate to other nodes in the event of hardware failure or reboot. The client behavior of SMB1 and SMB2 during rolling upgrade is listed in the following table:

**Table 4. SMB1/SMB2 client behavior during a rolling upgrade: always disruptive**

| Access methodology | Client behavior |
|---|---|
| Direct IP access<br>Such as \\<PowerScale Node IP>\< share name> | • Connection will drop.<br>• The application may send an error message.<br>• The client will wait for the SMB service to resume on the node. |
| Access through SmartConnect zone<br>Such as \\<smartconnct zone name\<share name> | • Connection will drop.<br>• The application may send an error message.<br>• The client can quickly re-establish the connection to another PowerScale node by leveraging the SmartConnect failover policy. |

Therefore, the recommended configuration for SMB1 or SMB2 is to use SmartConnect with a SmartConnect Service IP and an IP failover policy to quickly re-establish the connection between the client and the PowerScale cluster.

When the SmartConnect failover policy is used, the connection will drop and re-established to another PowerScale node in the cluster. In the rolling upgrade process, the node where the new connection is established will also have a chance to reboot later on. And the worst case is that in an n-node PowerScale cluster, this disruptive failover will happen n times. Although re-establishing the connection using SmartConnect is usually instantaneous, there is still a brief disruption of the client application, which means that the client application is aware of the disruption and will send an error message. To resume the client workload, the connection must be re-established.

### SMB3 CA

In OneFS 8.0, PowerScale offers the SMB continuously available (CA) option. This allows SMB clients the ability to transparently fail over to another node in the event of a network or node failure. This feature applies to Microsoft Windows 8, Windows Server 2012 and later clients. This feature is part of PowerScale's non- disruptive operation initiative to give customers more options for continuous work and less downtime. The SMB CA option allows seamless movement from one node to another and no manual intervention on the client side. This enables a continuous workflow from the client side with no disruption from error messages.

Dell Technologies recommends using static IP pool with SMB3 CA for transparent failover and NDU consideration. But using dynamic IP pool can also work, but there is a risk with SMB3 CA Witness sending confusing signals. The behavior really depends on the client implementation. For example, it probably causes just one failover to another IP, after which the client loses interest in the original address, but it could also potentially make the client jump around with reconnections for no good reason if its interest is not lost and it keeps watching.

The SMB CA feature needs to be enabled at share creation time. To enable SMB CA, the following preconditions need to be met:

- • SMB3 is supported
- • The cluster is running OneFS 8.0 or later
- • Clients are running Windows 8 or Windows Server 2012 R2 or later

**Note**: It is recommended to enable SMB Witness feature for transparent failover, which can dramatically shorten the time to detect the failure. A common way to enable SMB Witness on PowerScale OneFS is to set the SmartConnect zone name and access the SMB share with the name. This is because SMB Witness can get the failure notification from SmartConnect and FlexNet.

If any precondition in the above list is not met, SMB3 CA will not function.

You can use the following command to create an SMB file share with CA enabled:

```
isi smb shares create <name> <path> --continuously-available=yes
```

In case the existing share is not SMB CA enabled, you can still enable it on an existing file share by using the following command:

```
isi_smb_ca_share --enable-ca --share=<the name of SMB share>
```

Using this command to make a change will actually delete and recreate the share without losing any data. But it will result in a quick disconnection for all current clients and this is a disruptive command. After the OneFS rolling upgrade is finished, if you want to revert it back, use the following command:

```
isi_smb_ca_share --disable-ca --share=<the name of SMB share>
```

To verify that the SMB CA and SMB Witness is enabled at the client level, check the Windows Event Log in the following path:

```
Applications and Services Logs, Microsoft, Windows, SMBClient, Connectivity.
```

Figure 13 shows an example of the Windows Event Log message of successful Witness registration.

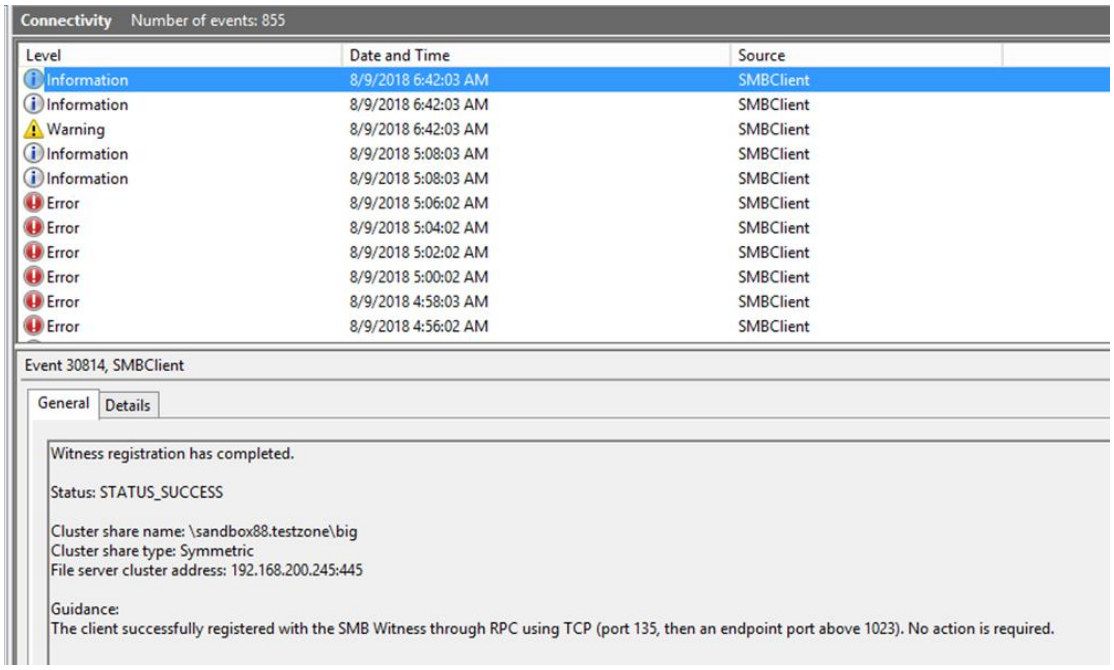


**Figure 13. SMB3 Witness registration**

During the rolling upgrade process, the PowerScale node reboots one by one. If SMB3 CA is enabled on a share, when the PowerScale node reboots, the connection to the share will not be disrupted and thus there will be no error message sent by the application. There will still be a short period of time when all the workload on the share is paused and automatically resumed in several seconds.

For the performance impact, it is similar to an NFS workload. See the section Performance impact for more details.

## NDU best practices concluded for SMB1/SMB2/SMB3

As a summary of 2.2.1SMB1 and SMB2 and 2.2.2 SMB3 CA, we recommend the following NDU best practices for SMB1/SMB2/SMB3:

- Use PowerScale static IP pool for SMB1/SMB2/SMB3
- Access the SMB share through SmartConnect zone name
- Use SmartConnect failover policy and connect to SmartConnect zone for SMB1/SMB2
- Use SMB3 CA for SMB3 share

**Note**: Due to the nature of SMB CA, this feature will bring some performance impact especially on write I/O. The impact depends on the factors like the PowerScale node type, the configuration of PowerScale OneFS like endurant cache (EC), workload profile and so on.

### FTP behavior and configuration consideration

FTP is a stateful protocol which means PowerScale should keep the session state between the client and itself. For this reason, the recommendation is to use a static IP pool. The IP will not failover or failback during the PowerScale node reboot. The client has to wait for FTP serviceability to resume on the node that it is connected to. In this case, using SmartConnect Service IPs (SSIP) can help minimize the impact. SSIP is implemented by a way of DNS delegation and it can help to redirect the request to the right PowerScale nodes which are still alive. However, if the rolling upgrade reboot happens in the middle of file transmission, the transmission will get stopped with errors and need to be manually re-establish the connection. The recommendation is as below:

- Use static IP pool for FTP workload
- Use SSIP enabled subnet for FTP workload to minimize the impact

**Note**: With all the recommendations above, the OneFS upgrade process still provides a disruptive upgrade. But it can dramatically minimize the impact.

### HTTP behavior and configuration consideration

PowerScale OneFS has a built-in web service and we can easily access the files by using the HTTP protocol. At the time of writing, PowerScale only supports HTTP 1.1 which is a stateless protocol. See RFC 7230 for details of HTTP 1.1.

Since it is a stateless protocol, it is recommended using dynamic IP pool to make sure that all the IPs in the pool are accessible during the reboot of OneFS rolling upgrade process. However, if a file is in the transmission status, it will get disconnected by errors and you have to retry and re-establish the connection by manually refresh the page and reinitiate the file transfer. An alternative way is to use SmartConnect zone name to make sure that the client HTTP request can always find the right PowerScale nodes that are still alive. However, it has the same side effect. Since SSIP is a way of a delegation of DNS, it will not support HTTP requests by directly accessing the IP address. Best practices include the following:

- Use dynamic IP pool for HTTP workload
- Use an SSIP enabled subnet for HTTP workloads, if all the HTTP requests are through the zone name

**Note**: With all the recommendations above, the OneFS upgrade process still provides a disruptive upgrade. But it can dramatically minimize the impact.

# Patch upgrade

The OneFS patch system provides a method to deploy a set of changes to all nodes in the PowerScale cluster in a simple and revisable manner, which is also under the control of the OneFS NDU framework. It allows a user to apply a patch in a simultaneous way or a rolling sequence. The details of rolling and simultaneous patch upgrade are shown in the section Rolling and simultaneous patch upgrade.

### Roll-Up Patches overview

A monthly cadence for Roll-Up Patches (RUPs) has been established to deliver critical fixes to customers.

In general, there are three kinds for RUPs delivered for each of the OneFS releases listed above every month. They are shown in Table 5:

**Table 5. RUPs category overview**

| RUPs category | Userspace/Kernel patch | Require reboot | Description |
|---|---|---|---|
| Userspace GA RUPs | Userspace patch | No | Highest priority fixes with minimum risk and maximum benefits |
| Userspace DA RUPs | Userspace patch | No | Broader fixes coverage |
| Kernel GA RUPs | Kernel patch | Yes | Fixes in the kernel space, for example, drivers bug or security bug. It will not conflict with DA or GA Userspace RUPs. |

The relationship among all the three RUPs categories is concluded as below:

- Each month's Userspace GA RUP is a superset of the Userspace GA RUP for the previous month.
- Each month's Userspace DA RUP is a superset of the Userspace DA RUP for the previous month.
- Each month's Userspace DA RUP is a superset of the Userspace GA RUPs of the current month.
- Kernel GA RUPs will not conflict with either Userspace DA RUP or Userspace GA RUP of the same month

## General best practices

Here list several general best practices and configuration considerations for patch upgrade:

- The NDU framework was originally created to handle OneFS upgrades. The same framework is also used for the patch system and to deploy firmware packages. As designed for the 8.0 and later releases, the NDU framework can only be used to perform one action at a time. This means that once a OneFS upgrade has been started, NDU will not be available to deploy patches or firmware packages until the upgrade has been committed.
- Beginning with the 8.0 release, the NDU framework creates the possibility to upgrade a cluster and then roll back to the previously installed version anytime before the new version is committed. This is accomplished by creating a rollback image from one node in the cluster and then deploying that rollback image to all the nodes during the rollback operation. When the rollback file is created, it will include any patches installed on the node and the local patch databases. If the OneFS upgrade is rolled back, the exact patches that were previously installed on each node will be replaced with the patches contained in the rollback file. This will only be an issue if there is a different patch set installed on different nodes in the cluster. Dell Technologies recommends that you use consistent patch among all the PowerScale nodes in the cluster.
- Dell Technologies recommends that you install the patch during an off-hours maintenance window to minimize the disruption of service to clients.

## Rolling and simultaneous patch upgrade

Before OneFS 8.2.0, in the patch upgrade command `isi upgrade patch`, there is a parameter `-- rolling=true/false` which controls how the patch is applied. With a different purpose of the patch, this parameter behaves differently.

1. If a patch impacts certain services and only requires services to restart (user space patch), it does the following:

   When the parameter `--rolling=false` is set, a simultaneous patch request is made and a patch will be installed simultaneously across all nodes. The patch will typically run an `isi services` command to disable and then later to re-enable the affected services. Since services that are affected by the patch are simultaneously restarted on all the PowerScale nodes in the cluster, this will affect the specified services across the entire cluster causing temporary service disruption. For a simultaneous patch request, the PowerScale nodes will not be rebooted.

   When the parameter `--rolling=true` is set, a rolling patch upgrade request is made. In this case, the patch will be installed and the node will be rebooted in succession. For rolling patch upgrade request, the specified services will not be restarted. Instead, NDU will migrate all user connections away from the nodes before starting the patch request. This migration process can be disruptive. NFS with dynamic pool and SMB CA can help to make this migration transparent to the client application. For other workloads, they will get disconnected and need to re-establish the connection when the node they are connected to reboots.

   Simultaneous patch upgrade request can save time and act more efficient, but the specified service will be restarted which causes a temporary service disruption. At the same time, rolling patch upgrade request is the default setting and can take much longer especially when the PowerScale cluster is large. But, it can be less disruptive with the combination of NFS dynamic pool or SMB CA.

   The guideline is for the specified service which will not impact the workload like WebUI, PAPI, and so on, use the parameter `--rolling=false` for the patch upgrade. This will make the upgrade more efficient and will not impact the real workload. For a service that can impact the workload, such as NFS or SMB, use the parameter `--rolling=true` with the combination of NFS dynamic IP pool or SMB CA to minimize the impact to the client application.

2. In the case of a patch requiring PowerScale node reboots (kernel patch), it does the following:

   When the parameter `--rolling=false` is set, a simultaneous patch request is made and a patch will be installed simultaneously across all nodes. And in this case, it will reboot all nodes in the cluster simultaneously.

   When the parameter `--rolling=true` is set, a rolling patch upgrade request is made. In this case, it will install the patch and then reboot each node in succession.

   The guideline is to use `–rolling=true` to have minimal impact on the workload. However, if customers are willing to have a maintenance window with the disconnected workload, use `– rolling=false` to make this patch upgrade more efficient.

   Starting from OneFS 8.2.0, the parameter `--rolling` is no more and now it uses `--simultaneous` for the same purpose. It is important to carefully read the `Readme` file for each patch, which explains the behavior of the patch installation process and its impact in details.

## Installation of patches during a OneFS upgrade

Starting with version 8.2.0, OneFS supports automatically installing a patch during a OneFS upgrade. Use the newly added parameter `--patch-paths` of `isi upgrade start` to include a patch to install when staring a OneFS upgrade. The following is an example:

```
isi upgrade start --install-image-path=<OneFS image path> --patch-paths=<patch path>
```

There are some considerations to be aware of:

- This parameter only takes a single patch to be installed during the upgrade.
- The path of the patch should be within `/ifs`.
- The installation of the patch happens only after each node is rebooted into a new version but before it goes to `COMMIT` states. This means it may require a second reboot for patch installation depending on which category the patch falls in. If it is a kernel patch, it then requires a second reboot.
- Once it is triggered, the patch will be listed in `isi upgrade patches list` with a partial status until fully installed on all nodes. See the following example:

  ```
  tme-sandbox-2# isi upgrade patches list
  Patch Name    Description          Status
  ----------------------------------------
  hp-provision  Halfpipe test files  Partial
  ----------------------------------------
  Total: 1
  ```

  After the patch has been installed successfully on all nodes, the status will be changed to installed.

  ```
  tme-sandbox-2# isi upgrade patches list
  Patch Name    Description          Status
  ----------------------------------------
  hp-provision  Halfpipe test files  Installed
  ----------------------------------------
  ```

- Before OneFS 8.2.1, the parameter `--patch-paths` can only take a single patch. If there is a requirement to install additional patches during the OneFS upgrade process, it is recommended to use `isi_patch register` command to achieve this.

---

**Note**: Dell Technologies recommends adding additional patches be requested after upgrade has been started but before any nodes have been upgraded before commit. Otherwise it only applies to the remaining nodes. The nodes already upgraded will need to be reconciled manually on each node.

---

The following is an example:

Start a OneFS upgrade using the following command.

```
isi upgrade start --nodes=1 --install-image-
path=/ifs/PipelineBeta/OneFS_v8.2.0.BETA.2_Install.tar.gz --skip-
optional
```

Before any PowerScale node gets rebooted, run the following two commands to add to additional patch during the OneFS upgrade process:

```
isi_patch register /ifs/PipelineBeta/hp-provision.pkg
isi_patch register /ifs/PipelineBeta/hp-base.pkg
```

The two patches are added to the patch list with partial status:

```
tme-sandbox-3# isi upgrade patches list
Patch Name   Description                    Status
-------------------------------------------------
hp-base      Patch interaction - Base patch Partial
hp-provision Halfpipe test files            Partial
-------------------------------------------------
```

After the patches have been installed successfully on all PowerScale nodes, the status will be changed to `Installed`.

```
tme-sandbox-2# isi upgrade patches list
Patch Name    Description                    Status
-------------------------------------------------
hp-base       Patch interaction - Base patch Installed
hp-provision  Halfpipe test files            Installed
-------------------------------------------------
```

- Starting from OneFS 8.2.1, the parameter `--patch-paths` can take multiple patches and install them during the OneFS upgrade.
- After the rollback, the patches which have been installed during the OneFS upgrade will automatically be uninstalled.

## Simplified patch installation process

Starting from OneFS 8.2.1, the patch installation process has been simplified and enhanced. Prior to OneFS 8.2.1, you will see the following behaviors when you apply a new RUP on top of an existing one:

- The previous patch has to be uninstalled first and then you can install the new patch no matter whether it is for kernel or userspace, DA or GA, RUP or singleton.
- Multiple reboots or service restart will be required during this process.
- There will be a less protected time window that in the middle of the process where existing security fixes in the current patch will be uninstalled.
- Patches are not able to patch patch-system.

In OneFS 8.2.1, the simplified patch installation process includes the following:

- Support installing the patch without uninstalling the previous one.
- Reduce node reboot to only once.
- Reduce service restart to only once.

- In most cases, eliminate the less protected window in the upgrade process.
- Support for patching the patch system

**Note**: This process cannot apply to firmware or OneFS upgrades.

The following is an example to explain how this feature works:

In this example, the test patch, `patch-1234`, has been installed as shown below:

```
# isi upgrade patches list
Patch Name Description                  Status
-------------------------------------------------
patch-1234 base patch to be deprecated. Installed
-------------------------------------------------
```

Install a new patch, `test-deprecated-patch`, which is to deprecate the previous one `patch-1234`. To do this, leverage the command `isi upgrade patches install` as shown in the following:

```
# isi upgrade patches install /ifs/test-deprecated-patch.pkg
The below patches are deprecated by this patch and will be removed
automatically:
- patch-1234
Would you like to proceed? (yes/[no]): yes
Requested install of patch test-deprecated-patch.
```

In this case, it will automatically detect an existing deprecated patch installed in the cluster and ask you if you want to proceed to remove it.

During the installation, the status of `patch-1234` is `AdHoc` as shown below.

```
# isi upgrade patch list
Patch Name             Description                Status
-----------------------------------------------------------
patch-1234  base       patch to be deprecated.    AdHoc
test-deprecated-patch  patch with deprecated tag  Installing
-----------------------------------------------------------
Total: 2
```

`AdHoc` status means:

- This patch is to be deprecated and removed by a new patch installation.
- This patch is still taking effect on the PowerScale cluster

After the new patch has been installed and the deprecated patch has been removed, the patch status will change to the following:

```
# isi upgrade patch list
Patch Name             Description                Status
-----------------------------------------------------------
test-deprecated-patch  patch with deprecated tag  Installed
-----------------------------------------------------------
```

This feature can dramatically simplify the upgrade process by removing unnecessary steps, node reboots, and service restart. As a result, it makes the patch upgrade process more efficient and secure. Based on this, Dell Technologies recommends you to leverage this feature when it applies.

# Firmware upgrade

Prior to OneFS 8.2.0, the only way to upgrade node firmware was through a rolling (serial) firmware upgrade process. Serial firmware upgrades have the least impact to business continuity, yet require extended maintenance windows that increase linearly with cluster size.

In OneFS 8.2.0, a new parameter `--simultaneous` is introduced for the CLI command `isi upgrade firmware`. This parameter along `with –nodes-to-upgrade` allows upgrading firmware on multiple PowerScale nodes simultaneously. OneFS will check if the simultaneous upgrade parameter is valid or not. The following is an example where node 1 and node 2 are PowerScale node pairs. OneFS will stop the upgrade process when there is an attempt to start a simultaneous firmware upgrade for these two nodes. This is to ensure data integrity and that we do not update peer nodes simultaneously.

```
# isi upgrade cluster firmware start --no-verify --no-burn --
simultaneous -- nodes-to-upgrade=1,2
You are about to start a Simultaneous Firmware UPGRADE, are you
sure? (yes/[no]): yes
Invalid nodes specified for simultaneous upgrade. Please run
isi_upgrade_helper for possible valid commands
```

To support simultaneous firmware upgrades, a new tool, `isi_upgrade_helper`, is also included in the OneFS 8.2.0. This tool can help end users decide how to use the newly introduced simultaneous firmware upgrade mechanism to meet their business requirements. This tool will give three firmware upgrade recommendations:

- Least Disruptive Firmware Upgrade Recommendation
- Fastest Firmware Upgrade Recommendation
- Balanced Firmware Upgrade Recommendation

Within each recommendation, the output of the command – `isi_upgrade_helper` also includes the corresponding CLI commands which can be used directly for firmware upgrades.

From OneFS 9.0, the tool `isi_upgrade_helper` is removed and the node firmware upgrade process has been changed. For details, see the section Node firmware upgrade workflow from OneFS 9.0.0.

## Least Disruptive Firmware Upgrade Recommendation

This option is to upgrade one PowerScale node at a time, which causes the lowest impact on availability and performance during the firmware upgrade process. It has no difference with the firmware upgrade mechanism before OneFS 8.2, which will take a long time for a large PowerScale cluster. The following command is used for this option:

```
isi upgrade cluster firmware start
```

### Fastest Firmware Upgrade Recommendation

This recommendation which is also known as simultaneous firmware upgrade is to ensure the data integrity during the firmware upgrade process by preventing any Gen 6 Node Pairs from being upgraded concurrently. For example, the simultaneous firmware upgrade will be run in the following sequence:

```
isi upgrade cluster firmware start --simultaneous --nodes-to-
upgrade <odd number of slots>
isi upgrade cluster firmware start --simultaneous --nodes-to-
upgrade <even number of slots>
```

**Note**: For Isilon Gen 4 or Gen 5 nodes which have no Node Pairs, you can add these nodes to either `<odd number of slots>` or `<even number of slots>`. By default, `isi_upgrade_helper` will add them to the `<odd number of slots>`.

It is not recommended to use this option to upgrade PowerScale node firmware in a production environment.

### Balanced Firmware Upgrade Recommendation

This recommendation which is also known as intermediate firmware upgrade is a balance between simultaneous and rolling firmware upgrade. It runs faster than a rolling upgrade but is less intrusive than a simultaneous firmware upgrade. The recommendation adheres to the following restrictions:

- No more than one node per Disk Pool is added to the firmware upgrade list
- No more than one node per Gen 6 Node Pair is added to the firmware upgrade list
- PowerScale nodes without redundant power suppliers are not added to the firmware upgrade list

To check the power redundancy for all the PowerScale node in the cluster, use the following command:

```
isi_for_array isi_hw_status | grep "Power Supp"
```

The following example indicates that all four nodes in the PowerScale cluster have redundant power supplies.

```
hop-isi-n-4: Power Supplies OK
hop-isi-n-4: Power Supply Slot3-PS0 good
hop-isi-n-4: Power Supply Slot4-PS1 good
hop-isi-n-2: Power Supplies OK
hop-isi-n-2: Power Supply Slot1-PS0 good
hop-isi-n-2: Power Supply Slot2-PS1 good
hop-isi-n-3: Power Supplies OK
hop-isi-n-3: Power Supply Slot3-PS0 good
hop-isi-n-3: Power Supply Slot4-PS1 good
hop-isi-n-1: Power Supplies OK
hop-isi-n-1: Power Supply Slot1-PS0 good
hop-isi-n-1: Power Supply Slot2-PS1 good
```

This recommendation will ensure that there is minimal impact during the simultaneous upgrade process.

## Workflow

### Node firmware upgrade workflow before OneFS 9.0.0

The overall workflow to upgrade firmware for PowerScale is shown in Figure 14. The rolling firmware upgrade part is only for the PowerScale OneFS prior to 8.2.0. For OneFS 8.2.0, all the three options, rolling, simultaneous, and intermediate firmware upgrade are available.

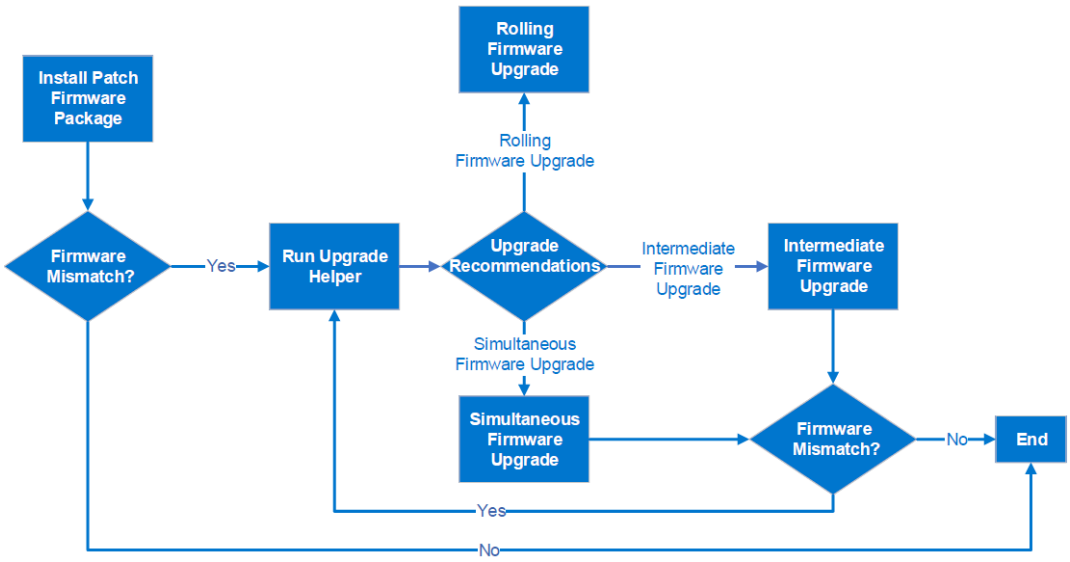


**Figure 14. Firmware upgrade workflows**

**Step 1**: It is required to install a firmware package prior to being able to upgrade. Starting from OneFS 8.0, use `isi upgrade patches` for this purpose. The following is an example:

```
isi upgrade patches install --simultaneous <firmware path>
```

**Note**: Since no reboot or service restart required for firmware patch install, Dell Technologies recommends using simultaneous parameter to go faster.

**Step 2**: Prior to OneFS 8.2.0, the only option is to run a rolling firmware upgrade. In OneFS 8.2.0, a new tool, `isi_upgrade_helper` is introduced. You can use the following command to leverage to tool to give recommendations:

```
isi_upgrade_helper
```

The output of this tool includes three recommendations and the corresponding CLI command as introduced earlier in this section.

- Least Disruptive Firmware Upgrade Recommendation
- Fastest Firmware Upgrade Recommendation
- Balanced Firmware Upgrade Recommendation

**Step 3**: Choose a recommendation from the helper tool and run the firmware upgrade command according to the recommendation.

If running a simultaneous firmware upgrade, repeat the command: `isi_upgrade_helper` once the firmware upgrade is complete to obtain the next set of nodes to upgrade. Repeat until no firmware mismatches remain. To check the firmware mismatch, use the following commands and exam the Mismatch column.

```
isi upgrade firmware devices
Device                      Type    Firmware                 Mismatch  Lnns
-------------------------------------------------------------------------
Mellanox-EN1                40GigE  2.40.5030+EMC1090111023 No        1-4
Mellanox-EN0                40GigE  2.40.5030+EMC1090111023 No        1-4
EPspime_warthog             ePOST   07.00                   No        1-4
EPadpt_moons                ePOST   01.50                   No        1-4
EPbios_warthog              ePOST   28.14                   No        1-4
EPcmd_mam_64pin             ePOST   02.01.64.02             No        1-4
EPps0_gen2_artesyn          ePOST   02.14                   No        1,3
EPvrd0_warthog_stmicro      ePOST   01.01                   No        1-4
EPbcc_infinity              ePOST   00.00                   Yes       3
EPvrd1_warthog_stmicro      ePOST   01.01                   No        1-4
```

**Note**: You may see `Upgrade framework is re-gathering firmware status, please try again` as the response to `isi upgrade firmware devices`. This is normal behavior.

## Node firmware upgrade workflow from OneFS 9.0.0

From OneFS 9.0.0, the tool `isi_upgrade_helper` is removed and the node firmware upgrade process has been simplified. The overall workflow is:

1. Assess a node firmware.
2. Check the firmware assessment report.
3. Run the firmware upgrade process
4. Monitor the upgrade status

The following example shows how it works in OneFS 9.0.0

1. Assess a node firmware. In this step, the upgrade check is run without starting a real upgrade process.

   ```
   # isi upgrade firmware assess --fw-pkg
   /ifs/<node_firmware_package>.tar
   ```

**Note**: The assessment may take an extended amount of time to complete based on cluster size, activity, platform types, and so on.

2. Check the firmware assessment report. Step 1 will be running for a while and you can use `isi upgrade view` to check if the assessment is complete or not. After it finishes, run the following command to check the assessment report:

   ```
   # isi_upgrade_logs --get-fw-report
   ```

3. If there are no issues or conflicts in the report, run the real upgrade:

   ```
   # isi upgrade cluster firmware start --fw-
   pkg=/ifs/<node_firmware_package>.tar
   ```

**Note**: In OneFS 9 and OneFS 9.1 releases, to perform a 'Balanced/Parallel' or 'Fastest/Simultaneous' node firmware update, run the commands provided in the results of the firmware assessments.

4. Use the following command to monitor the upgrade process:

```
# isi upgrade view
# isi upgrade firmware view
```

## Parallel node firmware upgrade

In OneFS 9.2 and later releases, the firmware assessment no longer provides commands needed to perform a 'Balanced/Parallel' or 'Fastest/Simultaneous' node firmware update. The functionality to perform updates in these manners is now built into the firmware update commands as additional arguments. Use the additional argument, --parallel for Balanced/Parallel, --simultaneous for Fastest/Simultaneous, or --rolling for Least Disruptive. For example to run a parallel firmware upgrade, use the following command:

```
# isi upgrade cluster firmware start --parallel --fw-
pkg=/ifs/<node_firmware_package>.tar
```

## Combined node firmware upgrade with OneFS upgrade

From OneFS 9.2.0.0, node firmware upgrade can be combined into the OneFS upgrade process to make it more efficient and save maintenance windows. The reboot times will be minimized to once per node and in this case to largely reduce the disruption to the workload and clients.

This feature is available in the CLI and WebUI:

```
# isi upgrade cluster start <install-image-path> --parallel --fw-
pkg <path>
```

Use the parameter `--fw-pkg` to specify the path for node firmware. Figure 15 demonstrates how to run combined upgrades in the WebUI.

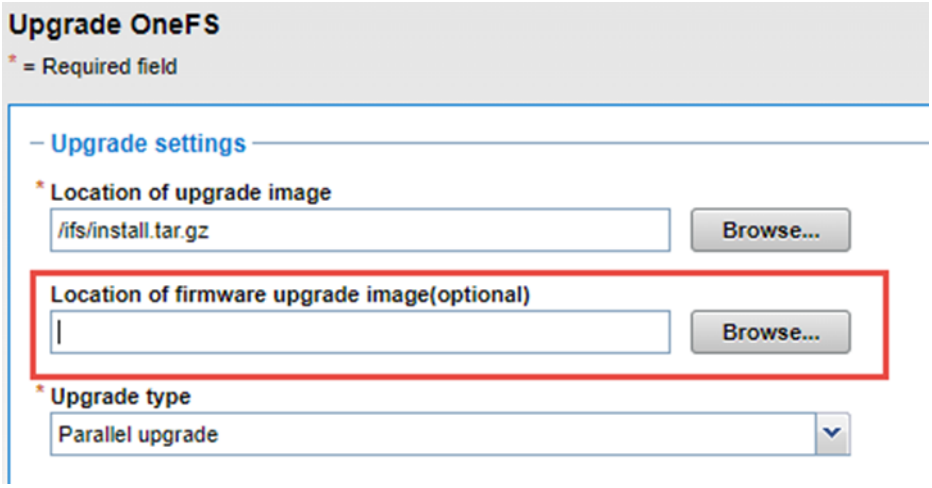


**Figure 15. Combined upgrade**

**Note**: rollback does not revert any applied firmware updates.

### General recommendations

Here list several general best practices and configuration considerations for PowerScale firmware upgrade:

- Always consult Dell Remote Proactive for PowerScale node firmware upgrade.
- Perform an initial simultaneous firmware upgrade on a subset of the PowerScale nodes before moving onto the remainder of the cluster.
- For **Fastest Firmware Upgrade Recommendation**, it is not recommended for a production environment with customer data.
- For **Balanced Firmware Upgrade Recommendation**, it is recommended for a production environment with customer data.

# Troubleshooting

This section explains the troubleshooting methodologies for the following upgrades:

- OneFS upgrade
- Patch upgrade
- Firmware upgrade

### OneFS upgrade

This section will focus on the guideline and methodologies of troubleshooting OneFS upgrade issues.

#### Monitor the OneFS upgrade

To monitor the high-level OneFS upgrade status use the following command

```
isi upgrade view –interactive
```

Starting from OneFS 8.2.0, it is possible to detect upgrade hangs. If a OneFS upgrade is not making any progress after 15 minutes, the CELOG sends a notification. In general, CELOG Alerts will be generated for three causes of upgrade hangs:

- PowerScale node is down or not responding over backend network.
- Upgrade agent is not ready to run the upgrade command.
- Upgrade hook takes a long time to complete and can get stuck in a start state for an indefinite amount of time.

For these three reasons, there are three newly introduced CELOG events:

```
400150007: Upgrade Hang – unable to communicate with Upgrade Agent on devids:
{devids}
400150006: Upgrade Hang – Upgrade Agent unable to make progress on devids:
{devids}
400150008: Upgrade Hook Hang – {hook} on devids: {devids}
```

Use the following command to check the event list for unresolved Upgrade Hang events:

```
isi event events list
```

## OneFS upgrade logs

The location of the OneFS upgrade logs varies between MRs. They can be categorized into two classes:

### *Pre 8.1.1.0*

- Error logs are at `/var/log/upgrade.log`
- Verbose logs are at `/ifs/.ifsvar/upgrade/Agent-*` and `/ifs/.ifsvar/upgrade/Supervisor-*`
- Pre-upgrade logs and hook script logs can be found at `/var/ifs/upgrade/logs/hook-logs`
- Use `isi_upgrade_logs` to gather all the logs when an error is reported by `isi upgrade view`

### *8.1.1.0 and later*

- All the logs are at `/ifs/.ifsvar/upgrade/logs` directory
- Use `isi_upgrade_logs` to display errors reported by `isi upgrade view`

Starting from OneFS 8.2.0, `isi_upgrade_logs` supports to filter and extract for specific upgrade logging information. The following list shows the supported filters. The filters can be used in combination.

`--guid` - dump the logs for the node with the supplied guid

`--devid` - dump the logs for the node/s with the supplied devid/s

`--lnn` - dump the logs for the node/s with the supplied lnn/s

`--process` - dump the logs for the node with the supplied process name

`--level` - dump the logs for the supplied level range

`--time` - dump the logs for the supplied time range

`--metadata` - dump the logs matching the supplied regex

An example is shown below to extract all the upgrade log generated by the process `isi_upgrade_agent_d` on PowerScale node 3 and 4:

```
isi_upgrade_logs --lnn=3,4 --process=/usr/bin/isi_upgrade_agent_d
```

**Note**: When upgrading into 8.1.1.0 or later release from a pre 8.1.1.0 release, the logging will follow the old format.

## Failure handling

In case you encounter some errors during the OneFS upgrade process, it is recommended to investigate the latest upgrade log first. After you determine the root cause and get the issue fixed, you can use the following command to retry the last upgrade action.

```
tme-sandbox-2# isi upgrade retry-last-action all
```

If it does not work, use the following command to roll back the upgrade:

```
tme-sandbox-2# isi upgrade rollback
```

**Note**: Rollback can lead to a simultaneous reboot process of all the nodes in the PowerScale cluster. Be careful to initiate OneFS rollback.

## Patch upgrade

This section focuses on the guideline and methodologies of troubleshooting patch upgrade issues.

### Monitor the patch upgrade

Since patch system activity is asynchronous, you will drop back to the command prompt immediately after issuing a patch system request. You can monitor the status of the patch installation or uninstallation by using the following commands:

```
tme-sandbox-2# isi upgrade node list
tme-sandbox-2# isi upgrade cluster view
```

You can also use the following commands to view the detailed messages of the patch upgrade:

```
isi_upgrade_status
```

To collect the log bundles of patch upgrade, use the following command:

```
isi_upgrade_logs
```

### Failure handling

It may be possible for a patch request to fail for some reasons. When this happens, both the patch system and the NDU framework will remain in use until either the patch request is completed or aborted. Both of these options can be completed using the following command:

1. The best course of action in case of failure is to examine the logs for both the NDU framework and the patch system to try to determine the root cause of the failure. If the issue can be corrected, the patch request can be restarted using the following command:
    a. To retry the last patch upgrade action on all the PowerScale nodes, use the following command:

    ```
    tme-sandbox-2# isi upgrade retry-last-action all
    ```

    b. To retry the last patch upgrade action on PowerScale nodes 2, 4, and 6, use the following command:

    ```
    tme-sandbox-2# isi upgrade retry-last-action 2,4,6
    ```

2. In rare circumstances, it may not be possible to correct for the failure and the only option will be to abort. In this case, it is necessary to abort the patch request using the following command:

    ```
    tme-sandbox-2# isi upgrade patches abort
    ```

Once this is completed, you can use the following command to archive and clean up the NDU status:

```
tme-sandbox-2# isi upgrade cluster archive --clear
```

**Note**: After the commands in option 2 have completed, the patch may still be installed on some nodes. This is likely if you were doing a rolling patch install and it failed in the middle. To uninstall from the remaining nodes, run a normal patch uninstall command: `isi upgrade patches uninstall <patch name>`.

## Firmware upgrade

This section focuses on the guideline and methodologies of troubleshooting firmware upgrade issues.

For `isi_upgrade_helper`, the logs are located on the PowerScale node where the tool is run at `/var/log/isi_upgrade_helper.log`. It is recommended to add the `--debug` parameter for verbose log information.

To monitor the upgrade process, use the following command:

```
isi upgrade view
```

The behaviors are different between rolling firmware upgrade and simultaneous firmware upgrade. The details are:

- PowerScale nodes that are selected for rolling firmware upgrade will quickly proceed through:
  a. Committed status
  b. Upgrade Ready status
  c. Committed status
- PowerScale nodes that are selected for simultaneous firmware upgrade will proceed through the following stats and it will run much slower:
  a. Committed status
  b. Upgrade Ready status
  c. Non-responsive (Rebooting)
  d. Upgrade Ready status
  e. Committed status

# References

## Dell Technologies documentation

The following Dell Technologies documentation provides other information related to this document. Access to these documents depends on your login credentials. If you do not have access to a document, contact your Dell Technologies representative.

- PowerScale OneFS Upgrade Planning and Process Guide
- OneFS Technical Overview