ISILON ONEFS NONDISRUPTIVE UPGRADES

An introduction

This whitepaper provides an overview of the nondisruptive upgrade features of the OneFS operating system. It covers details of the different types of upgrades and the upgrade process.

July 2018
TABLE OF CONTENTS

Executive summary .......................................................................................................................................................... 3
What is an upgrade? .......................................................................................................................................................... 4
Background ........................................................................................................................................................................ 4
OneFS 8.0: nondisruptive upgrades ................................................................................................................................................. 4
Upgrade types .................................................................................................................................................................... 4
Simultaneous upgrade .......................................................................................................................................................... 4
Rolling upgrade ...................................................................................................................................................................... 5
Rolling or simultaneous upgrade? .................................................................................................................................................... 5
Rollback .............................................................................................................................................................................. 5
OneFS 8.0 rules for upgrades .......................................................................................................................................................... 5
Cluster upgrade states and transitions ................................................................................................................................. 7
Upgrade state details .......................................................................................................................................................... 8
What happens when a commit is performed? ................................................................................................................................... 8
What functions are available before an upgrade is committed? ................................................................................................. 8
Operating the cluster in UPGRADED state ...................................................................................................................................... 9
Patch upgrades ................................................................................................................................................................................ 9
OneFS upgrade process .......................................................................................................................................................... 9
Upgrading with the WebUI ....................................................................................................................................................... 9
Prepare for upgrade ............................................................................................................................................................... 10
Cluster maintenance window .................................................................................................................................................. 10
Upgrade pre-checks ............................................................................................................................................................... 10
Start upgrade ............................................................................................................................................................................. 10
Commit upgrade ....................................................................................................................................................................... 15
Rollback the upgrade .............................................................................................................................................................. 16
View upgrade errors .............................................................................................................................................................. 17
Conclusion ......................................................................................................................................... 18
Executive summary

Isilon® storage systems run OneFS® as the underlying operating system. Storage systems are upgraded typically to provide new functionality, as well as to receive fixes for known issues. The ability to perform an upgrade nondisruptively - that is, without interrupting data and application services - has become critical in the majority of environments. The ability to revert or rollback an upgrade is also highly desirable. Prior to OneFS 8.0, many upgrades were disruptive as all nodes had to be upgraded simultaneously. An upgrade could only be reverted with substantial assistance from technical support. In OneFS 8.0, the upgrade capability is completely redesigned so that nondisruptive upgrades are always available within designated release levels. Rollback to the previously installed OneFS version is also provided, providing the new version has not been committed. In following OneFS releases, the capabilities are further enhanced for improved reporting and monitoring of the upgrade process. This paper describes the capabilities and operation of nondisruptive upgrade for OneFS releases beginning with OneFS 8.0.

AUDIENCE

This white paper is intended for Isilon administrators who will be upgrading an Isilon OneFS cluster.
What is an upgrade?

In the context of this paper, an upgrade is the process of transitioning a cluster from one release of OneFS to another. In a OneFS cluster, each node runs an independent OneFS image.

To upgrade a node, the new software version is first unpacked to a working area on the node and all needed data for that node is migrated to that area. The node is then set to boot from the working area and finally rebooted. After reboot, some final processes run to complete the upgrade, after which the node is ready to resume outgoing data services. Before OneFS 8.0, many upgrades, particularly major upgrades, required all nodes to be upgraded simultaneously. The reboot process interrupted all data services and client connections, hence the operation was disruptive. This type of upgrade was a monolithic operation – that is, all nodes in a cluster had to be running the same release at any given time. From OneFS 8.0 onwards, all upgrades (including major upgrades) can be performed nondisruptively via ‘rolling upgrade’, where nodes are upgraded one at a time. Since all data protection schemes in OneFS are resilient to one node being down, SMB3 and NFS v3 and v4 data services will continue uninterrupted for the duration of the upgrade.

Note: simultaneous upgrades remain available and supported.

Background

Before OneFS 8.0, upgrades were distinguished as major or minor. Consider a release version in the format of w.x.y.z.

• A major upgrade increments the w or most significant digit between the original and target release, for example, 7.2.1.0 to 8.0.
• A minor upgrade only increments the x, y, or z digits (and not the w digit), for example, 7.2.0.0 to 7.2.1.0.
• Introduction. Appendix B in this template presents an example as a list of styles that are available in this template.

Major upgrades always required a simultaneous upgrade. Some minor upgrades could be performed as a rolling upgrade; however the process would incur client service interruptions. See “Upgrade types” below for more information on the difference between simultaneous and rolling upgrades.

Some version upgrades were supported directly (one hop) with a rolling upgrade; however the particular supported upgrade paths were not always predictable, and varied with each release. It was necessary to carefully consult the appropriate Upgrade Guides and Release Notes to determine if an upgrade from a particular source to a particular target release was available.

OneFS 8.0: nondisruptive upgrades

Starting with OneFS 8.0, the concept of major and minor upgrade is no longer used. Upgrade paths are well defined according to explicit rules derived from the release numbering. An upgrade between two releases is either possible or not. If it is possible, it can be achieved by either rolling or simultaneous upgrade and it can also be rolled back. Upgrade commands are enhanced in each release starting with 8.0.0 to provide better control and predictability to the upgrade process. The goal is to provide a seamless and robust upgrade process which is nondisruptive to normal business workflows.

All supported upgrades from OneFS 7.1.x or 7.2.x to 8.0.x require simultaneous upgrade and will be disruptive. Once a cluster has reached an 8.x release level, nondisruptive rolling upgrades are available going forward, in addition to simultaneous upgrades.

Upgrade types

Upgrades can be performed in two ways – simultaneous upgrade and rolling upgrade. Going forward from OneFS 8.0, it is expected that customers will increasingly adopt rolling upgrade as it is nondisruptive to workflows.

Simultaneous upgrade

In a simultaneous upgrade, all nodes in the cluster are taken out of service together and upgraded. The cluster is unavailable until the upgrade is complete; therefore a simultaneous upgrade is disruptive to workflows.
Rolling upgrade

In a rolling upgrade, nodes are upgraded one by one. Each node drops out of the cluster then rejoins after it is upgraded to the new target release. Since the nodes are upgraded individually and sequentially, cluster services remain available during the upgrade, providing a nondisruptive experience. As each node is upgraded, the clients connected to it are disconnected and reconnected to another running node. In general, NFS v3 and v4 clients and SMB3 clients on shares with Continuous Availability will be transparently (nondisruptively) reconnected. SMB2 clients will experience a disconnect and have to reconnect to re-establish file services. Compared with a simultaneous upgrade, a rolling upgrade will take longer, since the nodes are upgraded one at a time.

Rolling upgrades have been available prior to OneFS 8.0 in certain circumstances; however OneFS provides a completely new upgrade infrastructure and operation to fully support nondisruptive rolling upgrades. Many compatibility and process checks are built in, and the ability to rollback to the previously installed release is provided. A progress indicator is provided with the interface, which provides valuable optics into the process by showing the estimated percent complete and estimate remaining time. Note that the progress indicator is provided as a guideline or estimate – it cannot be completely accurate because of the many variables which can occur during an upgrade. Generally speaking, the accuracy of the indicator increases as the upgrade process moves through the cluster.

The rolling upgrade process provides control over which nodes are upgraded, so you can initiate the upgrade on just a subset of nodes. For example, you may wish to upgrade particular nodes (for example, in a designated node pool) or to upgrade the cluster in a series of separate shorter maintenance events, rather than as a single longer event. The rolling upgrade process executes very conservatively to maintain the required robustness to the process. Each node must complete its upgrade and restart process serially – parallelization is not supported in currently available OneFS releases. As each node is upgraded, there is impact on the remaining nodes to preserve the required data protection levels (restriping). For clusters which are already busy, this impact may be observable to clients.

Rolling or simultaneous upgrade?

In OneFS 8.0 – any supported upgrade can be performed either as a simultaneous or a rolling upgrade. The default is a rolling upgrade; however simultaneous upgrade can play a role where the environment can tolerate a reboot across all nodes. In such environments, consider the tradeoffs associated with each upgrade process. A rolling upgrade, while nondisruptive, takes longer, and any performance impact will be of longer duration, compared to a simultaneous upgrade. That said, a rolling upgrade can be broken up into smaller events for a finer granularity, whereas a simultaneous upgrade requires a single outage for the cluster.

Rollback

OneFS 8.0 provides the ability to rollback, or undo an uncommitted upgrade. Rollback is the process of returning the cluster to the previously running release version – that is, the OneFS version which was running before the upgrade was initiated. Both simultaneous and rolling upgrades can be rolled back. Rollback is available once the cluster is upgraded to OneFS 8.0 and higher - rollback to any 7.x release is not supported. Rollback requires all nodes to be restarted, and is supported only to the most recent previously installed version of OneFS (8.0 and higher). Both simultaneous and rolling upgrades can be rolled back. Rollback can be performed at any time up to when the upgrade is committed. An upgrade commit completes the upgrade process, and after that time, rollback is no longer possible. Information about the previous (source) release is maintained in the cluster until the commit is performed, and will be restored as the cluster state in the event of a rollback. See section “Upgrade state details” for more information on committing and rolling back an upgrade.

OneFS 8.0 rules for upgrades

Upgrades bring changes and new functionality – and the greater the distance between the source and target releases, the more complexity and overhead is involved in maintaining the previous release state. Therefore a predictable set of rules exists so it is easily possible to see which releases can be upgraded to from a given source release. These rules apply to both simultaneous and rolling upgrades - in other words, an upgrade is either possible or not, regardless of the upgrade method chosen, and any supported upgrade can be rolled back to the previous release.

Consider a source release version in the format of w.x.y.z. The rule is that a 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.
Exception: A one-step supported upgrade path from 7.1.1.11 to 8.1.1 is available. This is an exception to the x+2 rule and is provided to support customers using NFS encoding, which is introduced in OneFS 8.1.1. This will be a simultaneous upgrade, as for any 7.x to 8.x upgrade.

Considering releases 8.0, 8.0.1, 8.1.x (including 8.1.1), and hypothetical releases 8.2, 8.3 and 9.0, the following upgrade paths would be available as rolling upgrades. The following table is used to demonstrate how this rule set works.

Note: the numbered releases shown in orange italics are solely theoretical, shown only for the purposes of illustration. They represent no commitment or roadmap from Isilon on what future versions may be available.

<table>
<thead>
<tr>
<th>To</th>
<th>8.0</th>
<th>8.0.1</th>
<th>8.1.x</th>
<th>8.2</th>
<th>8.3</th>
<th>9.0</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>8.0</td>
<td></td>
<td>R or S</td>
<td>R or S</td>
<td>R or S</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>8.0.1</td>
<td></td>
<td>R or S</td>
<td>R or S</td>
<td>No</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>8.1.x</td>
<td></td>
<td></td>
<td></td>
<td>R or S</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8.2</td>
<td></td>
<td></td>
<td></td>
<td>R or S</td>
<td>R or S</td>
<td></td>
</tr>
<tr>
<td>8.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R or S</td>
</tr>
</tbody>
</table>

S   =   Simultaneous Upgrade supported  
R   =   Rolling Upgrade supported  
No  =   No direct Upgrade, multi-hop required

Table 1 - Sample upgrade paths

It is clear from the table that:

- All upgrades from 7.x to 8.x are simultaneous upgrades.
- 8.0 can upgrade to any higher 8.0 release, and to 8.1.x and 8.2; this represents the maximum 2 increments in the x digit. 8.0 to 8.3 is not supported, since that would be 3 incremental steps (0 → 1 → 2 → 3)
- 8.1.x can upgrade to 8.2 and 8.3. An upgrade from 8.1.x to 9.0 is not supported, since that would be 3 incremental steps (1 → 2 → 3 → 0)
- 8.2 can upgrade to 8.3 and 9.0; x increases from 2 → 3 → 0, with one increment in the w digit (8 → 9)
- No release of 8.x could upgrade to an even more theoretical 10.x, since that would be two increments in the w digit.

A note on upgrades from OneFS 7.x to 8.0
Always refer to the Upgrade Guide for specific information on supported source releases, as there are some restrictions. Upgrade guides are published at the InfoHub https://community.emc.com/docs/DOC-44007.

A simultaneous upgrade will always be required, regardless of the source release, since this is a major upgrade. At the time of writing, the following releases can be simultaneously upgraded to 8.0.0.z or 8.0.1.z.

- 7.1.0.7+
- 7.1.1.4+
- 7.2.0.2+
- 7.2.1.0+

Clusters which are not running one of these releases need to be first upgraded to the closest supported starting release – for example, from 7.2.0.1 to 7.2.0.2 and from there to the desired 8.0.x.x release. As noted above, an exception is available only for simultaneous upgrade from 7.1.1.11 to 8.1.1 which may be performed in a single operation.

Cluster upgrade states and transitions

OneFS 8.0 introduces the concept of upgrade states. A OneFS cluster at release 8.x and higher is in exactly one of the following states at any given time. The state will change as it moves through the OS upgrade cycle, based on a set of allowed transitions. Transitions occur either by an explicit administrative operation, or automatically as a process completes.

The following diagram shows the states and transitions of the OneFS upgrade process. Cluster upgrade states are shown in upper case (COMMITTED, UPGRADING, ROLLBACK and UPGRADED). Operations are shown in italics (upgrade, rollback and commit). These represent explicit actions which the administrator performs during the upgrade process and cause a state transition. The “upgrade complete” and “rollback complete” steps shown in lower case are conditions that cause the cluster to transition to another state. The upgrade is complete when the final node is upgraded to the target release. Rollback is complete when the rollback process has finished, and the cluster automatically transitions to the COMMITTED state.

![Cluster upgrade states and available transitions](image)
Upgrade state details

COMMITTED: A previous upgrade operation, if any, has been completed and committed. 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. The cluster is ready to start another OneFS upgrade when required. A cluster remains in this state until another upgrade is initiated. This is considered the steady state of a cluster, and it is expected that a cluster over its lifecycle will spend the majority of its operational time in this state.

UPGRADING: A OneFS upgrade has started — that is, at least one node has started upgrading to the target release version. The required information to rollback to the source release is maintained while the cluster is in UPGRADING state. A cluster remains in UPGRADING state until either all nodes are upgraded to the target release (at which time the cluster automatically transitions to UPGRADED state), or a rollback is initiated. While in UPGRADING state, the cluster is running in mixed mode, as there are now two versions of OneFS present in the cluster. Nodes which have already upgraded may be able to access some of the functionality of the new release. Nodes which have not been upgraded cannot access any new release functionality. See below “What functions are available before an upgrade is committed” for more details.

UPGRADED: All nodes are now running the target release version; however the upgrade has not been COMMITTED. The required configuration to rollback to the source release is maintained while the cluster is in UPGRADED state. A cluster remains in this state until either the upgrade is COMMITTED or a ROLLBACK is initiated. A cluster in UPGRADED state can run any new functionality of the target release, provided this functionality safely interacts with the source release version, and does not prevent the possibility of a rollback. See below “What functions are available before an upgrade is committed” for more details.

ROLLBACK: The cluster is in the process of rolling back a OneFS upgrade. Rollback can be initiated by the administrator on a cluster in either the UPGRADING or UPGRADED state. Once the upgrade is committed, rollback is no longer available. In 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. Rollback is available for both rolling and simultaneous upgrades. A cluster can be rolled back only to the previously installed release.

What happens when a commit is performed?
The commit operation modifies or removes all saved information such that the previous version can no longer recovered. At this point, all new functionality of the new release is available. The cluster transitions to the COMMITTED state.

What functions are available before an upgrade is committed?
The availability of new functionality introduced in a release depends on the cluster’s state. In general, functionality from a new release falls into one of the following three categories. The Release Notes for each OneFS version should provide guidance on individual features, to determine at what point they become available.

- Functionality which is available on an individual node as soon as it is UPGRADED to the target release (i.e. after it reboots to the target release). These functions execute on an individual node, rather than on the cluster as a whole. Nodes still to be upgraded will have the same interfaces and functionality as before the upgrade was initiated. Upgraded nodes may have a different WebUI, or different CLI/SDK functionality available according to the new release functionality. The functionality available therefore depends which node is the target of the HTTP/SSH/PAPI/SDK connection.

- Functionality which is available once the cluster reaches the UPGRADED state (all nodes running the target release). Certain features operate only when the entire cluster has been upgraded. At this point all nodes have the same WebUI and CLI/PAPI options. Functionality in this category and the previous category can be tested, and then rolled back with the release if required.

- Functionality which is available only when the cluster is in a COMMITTED state. These features are incompatible with the previous release or would prevent a rollback from being possible.
Operating the cluster in UPGRADED state

A cluster in UPGRADED state will operate normally so that you can test for backward compatibility of applications as well as available new functionality. If issues are found, the upgrade can be rolled back. Only functionality in the third category of the previous section will not be available until the upgrade is committed. You should aim to complete all testing and to either commit or rollback the upgrade as soon as possible. If you do not commit or rollback the upgrade, reminder or ‘nag’ messages about the uncommitted upgrade will be issued via CELOG starting seven days after the cluster enters the UPGRADED state. These messages will reappear periodically until the upgrade is committed or rolled back. There is no enforced requirement to commit the upgrade within a particular time period, and the commit operation will never be performed automatically – it will always require explicit administrator action. However Isilon engineering focuses its testing of clusters in UPGRADED/non-COMMITTED state for a period of 10 days after the upgrade is completed. For this reason, the best practice is to commit or rollback OneFS upgrades within 10 days. Should an issue arise requiring assistance from Technical Support, it is likely that you will be advised to either commit or rollback the upgrade. You will not be able to initiate a subsequent OneFS upgrade or to install cluster wide patches until any existing upgrade in progress has been committed.

Patch upgrades

While this paper focuses on OneFS upgrades, the infrastructure introduced in OneFS 8.0 also applies to the patch and firmware upgrade process, making it simpler and more transparent. Patches which do not change the file system or OS kernel can be installed without requiring a node reboot. Wherever possible, patches are packaged to minimize the services which will be impacted during patch installation. The command `isi upgrade patches view` provides easy to interpret output, showing which services are affected (such as SMB, NFS) and indicating whether the patch requires a node reboot. Patches can be installed rolling (without a cluster reboot) and can be rolled back.

OneFS upgrade process

This section works through a rolling upgrade of a OneFS cluster from 8.0 to 8.1.0. While both a WebUI and CLI are provided, the WebUI is recommended for better usability.

A note on the CLI

In OneFS 8.0, the previous `isi update` command is no longer available. The command `isi upgrade` is used to upgrade OneFS in 8.0 and all future releases. Refer to the CLI Administration Guide and Upgrade Guide for detailed syntax to run an upgrade.

Upgrading with the WebUI

Click Cluster Management → Upgrade to display the current status of the cluster. The current version of OneFS 8.0.0.0 has been committed, therefore the cluster is ready to begin a new upgrade cycle.

![Figure 2 - Cluster in stable state before starting the upgrade](image-url)
Prepare for upgrade

The Upgrade Guide for the target release provides detailed instructions for prerequisite tasks and checks. Use this as the prime reference during the upgrade.

Cluster maintenance window

OneFS 8.0 provides a capability to designate an internal Maintenance Window for the duration of the upgrade, by specifying a start and finish time. During the designated period, alerts will be suppressed. This prevents the event log from being flooded by benign or expected alerts which are a normal part of the upgrade process. To set a maintenance window, navigate to Cluster Management → Events and Alerts → Settings in the WebUI. Alternatively, use the command `isi event settings modify --maintenance-start <start-time> --maintenance-duration <duration>`.

![Figure 3 - Set a maintenance window to mute event alerts](image)

Upgrade pre-checks

It is recommended to run the pre-upgrade checks before starting the upgrade, using the `isi upgrade cluster assess` command. The pre-upgrade checks verify that the cluster is in a steady state before commencing a new upgrade. For example, any previous upgrade must have been committed, and there must be no recovery processes running for failed hardware. This command will not affect any user or application I/O or change the upgrade state of the cluster. If the check runs clean, you are ready to proceed with the upgrade. Otherwise, use the messages and the Upgrade Guide to fix the issue before proceeding.

Start upgrade

From the Upgrade OneFS status screen shown previously in *Error! Reference source not found.*, click Upgrade OneFS to start an upgrade. Enter the location of the downloaded image for the target OneFS release.
Here you can select either a rolling or simultaneous upgrade, and also control which nodes will be upgraded if you want to upgrade in stages. The default selection is a rolling upgrade of all nodes. The nodes will be upgraded in order of their logical node number. Note the checkbox “Skip optional pre-upgrade checks”. If you have run the `isi upgrade cluster assess` command immediately prior, you could check this box so that the upgrade starts faster. The most critical upgrade checks (for example whether any hardware recovery process is running) will still execute, even if this box is checked. It is recommended not to skip the checks. The Upgrade Planning and Process Guide contains the complete list of Upgrade compatibility checks which will be run.

If the cluster is determined to be not ready to start the upgrade process, the pre-check will fail with messages indicating the reason for failure. After you have cleared the condition, you can run the pre-check again until it successfully completes. It is recommended to run the pre-check step manually (as described in the section “Prepare for upgrade”) before starting the actual upgrade, as well as in this step, to detect any last minute state changes in the cluster which could affect the upgrade.

For more granular control over the upgrade process, click Specify nodes and reboot order. In the panel displayed, you can exert control over which specific nodes will be upgraded, and the order they should be upgraded in. This option is useful if for example, you want to upgrade only the nodes in a particular node pool, or you want to first run the upgrade on some nodes used for archive, followed by nodes running production workloads. The order and selection of nodes for upgrade can be modified while the upgrade is in progress.

The following figure shows an example of 2 nodes selected for upgrade. The up and down Order arrows are used to change the upgrade order. It is included for illustration only – in the rest of this example, the default “All nodes” is used to upgrade the entire cluster. Click Cancel at the bottom to exit the node selection.

Check you have selected the correct upgrade image, and click Start Upgrade. During a rolling upgrade, since each node reboots in turn, the WebUI will disconnect when the node it is running on is rebooted. You can reconnect on another node to continue monitoring the process. You will see a popup message to this effect – click Upgrade to proceed.
Once the upgrade starts, you will see a status message that an upgrade is in progress. This banner will be shown everywhere in the WebUI as a visual reminder of the state of the cluster.

The pre-checks start - either just the mandatory checks, or the full set, if you did not select to skip the optional pre-checks. This will take a few minutes to determine if all pre-checks are passed. The button **Halt and Roll Back Current Upgrade** is available – at any point, you can terminate the upgrade process and roll it back to the starting release. The display automatically refreshes periodically to show the latest status of the upgrade or you can manually refresh the browser window.

If the pre-checks succeed, the upgrade starts automatically on the first node. The status will automatically update as major stages are passed – for example, as each node completes its upgrade or performs the post-upgrade processes.

As the upgrade proceeds, this display changes dynamically to show the progress. The following figure shows that the first node has started the upgrade and is 16% completed. The total estimated time to complete the upgrade is blank – this will first be populated after the first node finishes its upgrade.

Note the cluster is now in an UPGRADING state.
After the first node is upgraded and completes the reboot, the second node starts to upgrade automatically. Notice the status display at the top of the window changes to indicate the number of nodes running the original and target release.

Another subtle change can be seen – when you connect your browser to the WebUI of an upgrade node, the message at the top of the cluster similarly changes to indicate the cluster is at the new release.

Figure 10 - Second node upgrading

Figure 11 - Cluster OneFS version changes
Also when connected to an upgraded node, you will see the progress bar. The upgrade progress bar changes accordingly, as shown in the following figure. A simple algorithm to estimate the total time remaining for the upgrade is used, based on the average time taken by the nodes upgraded so far. The algorithm does not take into account the current load on the cluster, or any jobs which might run during the upgrade, or be started during the upgrade. Therefore the estimate should be taken as a guideline only. For example, in a 3 node cluster if the first node took 10 minutes (T1), the estimated time to upgrade the remaining two nodes would be in the order of 20 minutes (2 x T1). Say the second node actually took 8 minutes (T2). It took 18 minutes (T1+T2) upgrade the first 2 nodes, with an average of 9 minutes per node. Therefore the estimated time to complete the final node would be 9 minutes. For larger size clusters in particular, the completion time displayed will become more accurate as the upgrade progresses through more and more nodes.

![Figure 12 - Progress indicator on an upgraded node](image)

The upgrade proceeds, automatically upgrading nodes 2, 3, and 4. You might temporarily see a node status turn to orange and show Rebooting before it turns to green.

![Figure 13 - Upgrade in progress](image)

When all cluster nodes have upgraded, the status changes again. The cluster transitions to UPGRADED state. Note the options on the screen – the **Commit upgrade to <release-level>** button is now present, as well as the Rollback button.
The process from here on is the same whether a simultaneous or rolling upgrade was performed. At this point, you might initiate some testing (leaving the cluster in UPGRADED state), and return at a later time to either commit or rollback the upgrade. Or you might commit the upgrade immediately. Eventually you must make the decision to either rollback or commit, as discussed in the section “Operating the cluster in UPGRADED state”.

**Commit upgrade**

When you are ready to commit the upgrade, click **Commit upgrade to <release-level>**. The popup message informs you about the consequences of committing the upgrade. Click Commit to proceed.

The current upgrade status changes to reflect the commit.

The commit process executes various tasks to finish the upgrade – any necessary file system changes are made so that all functionality of the target release is available, and the configuration and other saved information required to do a rollback to the previous release is removed, so that a rollback is no longer possible. Typically, these tasks execute almost instantaneously. For OneFS releases which include significant file system changes, the commit may take longer and potentially temporarily impact performance. The Release Notes or Upgrade guide may provide some guidance if this is a possibility.
After the release is committed, the cluster reaches the COMMITTED state. The status shows all nodes running the target release. The cluster is now in a steady state and will remain so until a subsequent software upgrade is initiated.

**Rollback the upgrade**

If you decide to rollback the release, and you have been exercising any new release capabilities, you will first need to delete any created objects or configurations associated with these new capabilities. This is because such objects cannot exist on the cluster running the source release. As stated previously, any rollback requires all nodes to be restarted, temporarily interrupting cluster services and data availability.

Click the **Rollback to <release-version>** button - you are advised that all nodes will restart with interruption to client connections. Click **Roll Back** to continue.

This affects the WebUI - after the initial message shown in the figure, at some point, the WebUI will disconnect until the nodes are available again to run the WebUI services. There may still be some processes related to the rollback running in background at this point.
After the nodes reboot and the WebUI is available again, the cluster will again be at the original release and will be in the steady COMMITTED state. Processes will still run for some time in the background – the Upgrade OneFS button will not appear until these have completed, at which time, the cluster is ready to begin another upgrade cycle.

As indicated, the initial troubleshooting step is to run `isi_upgrade_logs` at the CLI. This command copies the appropriate logs for each node to the directory `/ifs/upgrade_error_logs`.

Cluster1-1# isi_upgrade_logs
Copied log to /ifs/upgrade_error_logs/1_pre-upgrade-optional_isi_upgrade_checker.log
Copied log to /ifs/upgrade_error_logs/2_pre-upgrade-optional_isi_upgrade_checker.log
Copied log to /ifs/upgrade_error_logs/3_pre-upgrade-optional_isi_upgrade_checker.log
Copied log to /ifs/upgrade_error_logs/4_pre-upgrade-optional_isi_upgrade_checker.log

Cluster1-1# ls -ls
  total 126
  26 -rw-rw-r--    1 root  wheel   3994 Nov 08 16:10 1_pre-upgrade-optional_isi_upgrade_checker.log
  26 -rw-rw-r--    1 root  wheel   3994 Nov 08 16:10 2_pre-upgrade-optional_isi_upgrade_checker.log
  26 -rw-rw-r--    1 root  wheel   3994 Nov 08 16:10 3_pre-upgrade-optional_isi_upgrade_checker.log
  26 -rw-rw-r--    1 root  wheel   3994 Nov 08 16:10 4_pre-upgrade-optional_isi_upgrade_checker.log

View the log to point to the error found by the upgrade process. The Upgrade Guide includes a troubleshooting section which will help you determine the error and take appropriate action. Once the error is resolved, you can restart the upgrade.

Conclusion

This paper has described the design and operation of upgrade in OneFS 8.0 and higher.