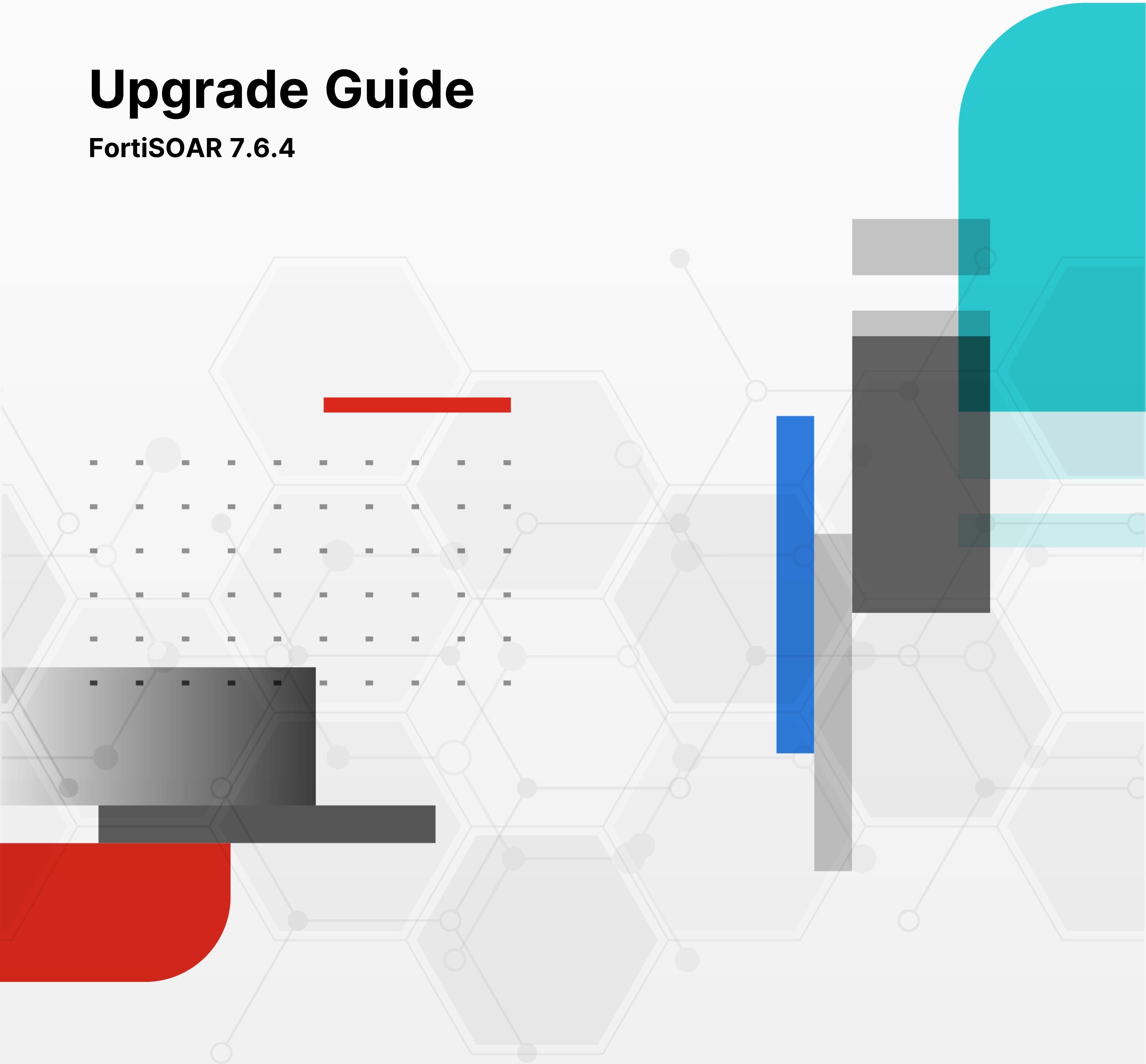
# Upgrade Guide

## FortiSOAR 7.6.4



**FORTINET DOCUMENT LIBRARY**
https://docs.fortinet.com

**FORTINET VIDEO LIBRARY**
https://video.fortinet.com

**FORTINET BLOG**
https://blog.fortinet.com

**CUSTOMER SERVICE & SUPPORT**
https://support.fortinet.com

**FORTINET TRAINING & CERTIFICATION PROGRAM**
https://www.fortinet.com/training-certification

**FORTINET TRAINING INSTITUTE**
https://training.fortinet.com

**FORTIGUARD LABS**
https://www.fortiguard.com

**END USER LICENSE AGREEMENT**
https://www.fortinet.com/doc/legal/EULA.pdf

**FEEDBACK**
Email: techdoc@fortinet.com

October, 2025
FortiSOAR 7.6.4 Upgrade Guide
00-400-000000-20210416

# TABLE OF CONTENTS

# Change Log

| Date | Change Description |
|---|---|
| 2026-01-13 | Updated the Upgrading a FortiSOAR Secure Message Exchange topic in the *Upgrading a FortiSOAR Distributed Multi-Tenancy Configuration* chapter to include steps for upgrading the secure message exchange from the 7.5.x release to the target release. |
| 2025-11-11 | Updated the *Introduction* chapter to reflect support for upgrading from versions 7.5.0–7.5.2 to 7.6.4. |
| 2025-10-14 | Initial release of 7.6.4 |

# Introduction

This guide covers upgrading a FortiSOAR™ enterprise instance, High Availability (HA) cluster, or a distributed multi-tenant configuration.

The FortiSOAR UI displays a notification when a new release (always the latest) is available. The notification also contains a link to that version's release notes so that you can get details about the latest available release. This keeps FortiSOAR users informed about the latest releases and then users can make informed decisions about upgrading to the latest available FortiSOAR version.

This document describes how to upgrade FortiSOAR to 7.6.4. This guide is intended to supplement the FortiSOAR Release Notes, and it includes the following sections:

- Preparing to Upgrade FortiSOAR
- Upgrading a FortiSOAR Enterprise Instance
  An "Upgrade Framework" was introduced in release 7.5.0 to improve the flexibility, usability, and efficiency of the FortiSOAR upgrade process.
- Upgrading a FortiSOAR High Availability Cluster
- Upgrading a FortiSOAR Distributed Multi-Tenancy Configuration
- Upgrading FortiSOAR using the Offline Repository
- Upgrading your FortiSOAR Docker image and upgrading your FortiSOAR Docker on an Amazon Elastic Kubernetes Cluster

You can upgrade your FortiSOAR enterprise instance, High Availability (HA) cluster, or a distributed multi-tenant instance to release 7.6.4 from releases 7.6.0 to 7.6.3 and 7.5.0 to 7.5.2.

# Important considerations before upgrading FortiSOAR

- **Upgrading an HA Cluster from Versions Prior to 7.6.1**: If you are upgrading your FortiSOAR HA cluster from releases 7.6.0, 7.6.0, 7.5.1, 7.5.0, etc. to release 7.6.1 or later, follow the steps in the Upgrading FortiSOAR High Availability Cluster for releases prior to 7.6.1 topic.
- **Upgrading an HA Cluster from Version 7.6.1 onwards**: If you are upgrading your FortiSOAR HA cluster from release 7.6.1 to release 7.6.2 or later, you can use the 'rolling upgrades' process. Steps for rolling upgrade are mentioned in the Upgrading FortiSOAR High Availability Cluster for releases post 7.6.1 topic.
- **Post-Upgrade Action**: After upgrading, log out from the FortiSOAR UI and log back in to complete the process.

- **System Downtime**: The upgrade procedure temporarily takes the FortiSOAR application offline. We recommend notifying users of the scheduled upgrade, as they will be unable to log in to the FortiSOAR during this time.
- **Troubleshooting**: If you encounter issues during the upgrade, see the *Troubleshooting FortiSOAR* chapter in the “Deployment Guide.”

# Preparing to Upgrade FortiSOAR

We recommend performing the following tasks to prepare for a successful FortiSOAR upgrade:

If you are upgrading a FortiSOAR instance with an external PostgreSQL database to FortiSOAR 7.6.0 or later, first upgrade PostgreSQL to version 16 or later. After upgrading PostgreSQL, proceed with the FortiSOAR upgrade. Upgrading to PostgreSQL 16 or later is required because FortiSOAR 7.6.0 and later use the 'pg_squeeze' and 'pg_repack' utilities—available only in PostgreSQL 16 and later—to optimize disk space reclamation. For more information, see the Externalization of your FortiSOAR PostgreSQL database chapter in the "Administration Guide."

To prepare for upgrading FortiSOAR (summary):

- *Only applicable for upgrades from FortiSOAR release 7.6.0 or earlier to FortiSOARrelease 7.6.1 or later*. Ensure that there is at least twice the current workflow storage capacity available before upgrading to release 7.6.4. This additional space is required because, post-upgrade, existing playbook execution logs are moved to historical storage to optimize the workflow log storage. This feature reduces active storage usage, improves performance, and enhances playbook efficiency.
  To check the workflow log size, run the following command and review the output under the '`workflow`' section:
  `csadm db --getsize`
  Ensure that the available space in the '`pgsql`' partition is at least **twice** the size of the workflow log. For example, if the workflow log size is 10 GB, the '`pgsql`' partition should have at least 20 GB of free space. To verify the available space, run the following command and check the '`var/lib/pgsql`' partition:
  `df -h`
  **NOTE**: This storage requirement does **not** apply if you are upgrading from version 7.6.1 to 7.6.2 or later. It applies **only** when upgrading from versions prior to 7.6.1 to 7.6.1 or later.
  If you do not have enough space, consider the following approaches:
  - **Approach 1: Optimize existing storage by purging playbook logs**
    This approach is suitable if you have limited disk space and can purge older playbook logs. Perform this action during periods with minimal or no data ingestion. Brief steps are as follows:
    i. Purge playbook logs older than one day using the **Executed Playbook Logs** > **Purge Logs** option in the UI.
    ii. Run the Reclaim Disk Space job using the **Storage Space Reclamation** setting found under **System Configuration** > **Application Configuration** in the UI.
  - **Approach 2: Increase Disk Space**
    This approach is recommended if you have sufficient disk space and prefer to retain playbook execution logs. To do this, increase the PostgreSQL logical volume to provide twice the current workflow storage space.
- Ensure that all data ingestion playbooks and schedules are stopped and wait for all existing active playbooks to complete before starting the upgrade process.
- Take a VM snapshot of your current system. Only after you have taken a VM snapshot of your system should you attempt to upgrade FortiSOAR. In case of any failures, these VM snapshots will allow you to revert to the latest working state. Follow the steps mentioned in the documentation of your platform for taking a snapshot and reverting to the current snapshot.

- Ensure that repo.fortisoar.fortinet.com is reachable from your VM. If you are connecting using a proxy, then ensure that proxy details set are correct using the `csadm network list-proxy` command and also ensure that `repo.fortisoar.fortinet.com` is allowed in your proxy. For more information on `csadm` CLI, see the *FortiSOAR Admin CLI* chapter in the "Administration Guide."
- Disable the IPv6 protocol on your VM if it is not being used, prior to upgrading FortiSOAR.
- Ensure that you have reviewed the *Special Notices* chapter in the "Release Notes", so that you are aware of operational and breaking changes made in version 7.6.4.
- Before starting the upgrade, identify and list all configuration files that have been modified from their default package versions.
  This step is critical for comparison with any `.rpmnew` files that might be generated during the upgrade process. See the "Handling Configuration Changes Post-Upgrade" topic in the Post-Upgrade Tasks and Notes chapter for steps on how to review or merge changes from `.rpmnew` files.
  **Example**:
  If you have modified the following files:
  `/opt/cyops-workflow/sealab/sealab/config.ini`
  `/opt/cyops/configs/database/db_config.yml`
  Ensure that you note these paths before proceeding with the upgrade. You will need them for verification after the upgrade.
- **Review Release Notes**: To ensure a smooth and informed upgrade experience, we recommend reviewing the 'Release Notes' before proceeding. Focus especially on the '*Special Notices*' chapter and the '*Known Issues*' chapter well-informed about new features, operational updates, and important changes in 7.6.4.

# Upgrading a FortiSOAR Enterprise Instance

The "Upgrade Framework" is designed to improve the flexibility, usability, and efficiency of the FortiSOAR upgrade process. Its modular architecture allows users to easily integrate custom tasks at different stages of the upgrade, providing a more personalized and adaptable experience. This framework empowers users to tailor the upgrade process by plugging in specific tasks during any phase, ensuring that the upgrade meets their unique requirements.

The framework also offers greater control by allowing users to selectively execute individual phases or tasks independently. This feature is particularly useful for focused testing, validation, and troubleshooting, as users can complete specific tasks without running the entire upgrade cycle. Additionally, the framework validates the feasibility of the upgrade before proceeding, ensuring that potential issues are identified early in the process. This reduces the risk of errors and enhances the overall reliability of the upgrade.

Furthermore, the Upgrade Framework supports the customization of both pre- and post-upgrade tasks, giving users the flexibility to tailor the upgrade process to their specific needs. It also enhances resilience by separating post-upgrade activities, such as database migrations and other services, from the upgrade process. This separation ensures that the code base for all packages is upgraded first, followed by the necessary post-upgrade tasks, which streamlines the overall process.

All upgrade-related logs are saved to a file named '`upgrade-fortisoar-<version>-<timestamp>.log`' in the `/var/log/cyops` directory. If an error occurs during the upgrade, you can resolve it by reviewing the log file and rerunning the upgrade command to resume from the point of failure. This approach enhances the upgrade process by offering inline resolutions, fixing issues in task files, and resuming the upgrade process from the point of failure, etc. For detailed upgrade steps, see the Manage FortiSOAR Upgrades topic.

## Optimizing FortiSOAR Upgrades

Prior to release 7.6.0, FortiSOAR relied on the `yum` repository for updates. However, a slow connection to the repository could cause significant delays in the upgrade process, leading to increased application downtime and user inconvenience. Starting with release 7.6.0, the upgrade process has been optimized to minimize application downtime by offering users the option to pre-download the necessary FortiSOAR upgrade packages and store them locally. During the upgrade, the process accesses these local packages, ensuring a faster and more reliable upgrade experience. Once the upgrade is complete, FortiSOAR will revert to using the `yum` repository for future updates.

It is recommended to upgrade your FortiSOAR instance promptly after downloading the upgrade packages to ensure that the latest packages are installed. Delaying the upgrade could result in outdated packages being installed.

# Manage FortiSOAR Upgrades

To manage your FortiSOAR upgrades post the 7.5.0 release, use the '`upgrade`' subcommand of the FortiSOAR Admin CLI ('`csadm`'). The various arguments that you can use with the `csadm upgrade` subcommand are explained in the Arguments available for the '`csadm upgrade`' subcommand topic.

> Before you start upgrading your FortiSOAR system, make sure to disable the IPv6 protocol on your VM if you are not using the IPv6 protocol and also ensure that you review the tasks mentioned in the Preparing to Upgrade FortiSOAR chapter.

To upgrade your system to FortiSOAR to 7.6.4, perform the following steps:

1. Users who have `root` access must perform the upgrade process.
2. ssh to the VM that you want to upgrade.
3. Check your system to see if `tmux` is installed; if not, use the following command to install it
   ```
   sudo yum install -y tmux
   ```
   Next, check that you are connected to a `tmux` session. A tmux session is needed for situations where network connectivity is less than favorable. You can check your tmux session using the following command:
   ```
   # tmux ls
   ```
   This command returns an output such as the following example:
   ```
   0: 1 windows (created Thu Nov 24 09:37:47 2022) [170x47]
   ```
   Log back into the SSH console and run the following command to reattach the `tmux` session:
   ```
   tmux attach-session -t 0
   ```
   If you do not find any `tmux` session, connect to one using the following command:
   ```
   # tmux
   ```
4. Run the following command to check if your FortiSOAR system is ready for an upgrade to the target release:
   ```
   csadm upgrade check-readiness --target-version [TARGET_VERSION]
   ```
   For example to upgrade to the 7.6.4 release, use the following command:
   ```
   csadm upgrade check-readiness --target-version 7.6.4
   ```
   The check-readiness argument runs various checks including checking if there is sufficient space available for the upgrade. If there is insufficient space in any directory during the space check, appropriate messages will be added to the check-readiness report. The report will contain information on the recommended space, currently available space, and the additional space required to meet the recommendation. The check-readiness report is generated at '`/opt/fsr-elevate/elevate/outputs/`' and will include explanatory messages for any failures.
   After addressing any validation failures to ensure system readiness for the upgrade, rerun the `csadm upgrade check-readiness` command to confirm that the system is prepared for the upgrade.
   **Note**: You must run the `csadm upgrade check-readiness` command prior to upgrading FortiSOAR to ensure a successful upgrade.
5. (Recommended) To download upgrade packages locally first and then upgrade your system at a later time, use the following command:
   `csadm upgrade execute --target-version [TARGET_VERSION] --download-packages] [--local-download-directory [LOCAL_DOWNLOAD_DIRECTORY]]`. This command will verify the availability of sufficient space in the default `/opt/cyops/packages` directory or in a directory specified by the user in the `--local-download-directory` argument. The packages will be downloaded to the designated directory if sufficient space is available. If space is insufficient, the command will exit with an error message indicating the issue.
   Use the `--local-download-directory` argument to specify the absolute path in the local directory where the upgrade packages should be downloaded. By default, the FortiSOAR packages, OS Packages and third-

party packages are downloaded to the `/opt/cyops/packages/fsr-packages` folder, while system-connectors rpms are downloaded to the `/opt/cyops/packages/fsr-connectors` folder.
**NOTE**: If the upgrade packages are successfully downloaded to the local directory, the upgrade process will utilize those packages for the upgrade. Otherwise, the upgrade process will access the `yum` repository for the upgrade. Once the upgrade is complete, FortiSOAR will revert to using the `yum` repository for future updates. Additionally, note that if you re-run the `csadm upgrade` command with the `--download-packages` argument, it will download the packages again, overwriting the previously downloaded ones.

6. Run the following command to upgrade your system:
   ```
    csadm upgrade execute --target-version [TARGET_VERSION]
   ```
   For example to upgrade to 7.6.4 use the `csadm upgrade execute --target-version 7.6.4` command.
   **IMPORTANT**: If your instance can only connect to "repo.fortisoar.fortinet.com" by using a proxy, then ensure that the proxy is set in the `/etc/wgetrc` file. For example,
   ```
   use_proxy=yes
   http_proxy=<proxy_server_ip:port>
   https_proxy=<proxy_server_ip:port>
   ```
   **Important Notes**: To upgrade a high availability cluster in FortiSOAR, you require to upgrade each node individually, one after the other. For more information, see the Upgrading a FortiSOAR High Availability Cluster chapter. For information on how to upgrade a FortiSOAR distributed multi-tenant configuration to 7.6.4, see the Upgrading a FortiSOAR Distributed Multi-Tenancy Configuration chapter. Note that when you upgrade your FortiSOAR enterprise instance, High Availability (HA) cluster, or a distributed multi-tenant configuration, the FortiSOAR appliance hostkey also gets changed.
7. Once your FortiSOAR instance is upgraded, you must log out from the FortiSOAR UI and log back into FortiSOAR.

After upgrading to FortiSOAR 7.6.3 or later, users running `sudo` commands as the 'csadmin' user will be prompted for a password on all systems except AWS.

# Arguments available for the '`csadm upgrade`' subcommand

The '`upgrade`' subcommand was added to the `csadm` utility in release 7.5.0. It is not available in releases prior to 7.5.0. This command can only be used to upgrade from release 7.5.0 to a later release, such as 7.6.0 or higher. For releases prior to 7.5.0, use the upgrade script to upgrade your FortiSOAR system.

Arguments for the '`csadm upgrade`' subcommand are as follows:

- `check-readiness --target-version [TARGET_VERSION]`: This option checks if your FortiSOAR system is prepared for an upgrade to the release specified in the 'target-version' argument. This option executes the pre-upgrade phase and saves a report with the results in JSON format at `/opt/fsr-elevate/elevate/outputs/`. All pre-upgrade validations are performed during the 'pre-upgrade' phase. A sample of a check-readiness report is present in the Example of a check-readiness report topic.
  The format of the JSON file is:
  ```
  {
         <short description of task>:
     {
        "result":<boolean value>,
         "msg":<string value>
  ```

```
   }
}
```

- `execute-task --target-version [TARGET_VERSION] --phase [PHASE] --task-name [TASK_NAME]`: This option allows you to run a specific task during a particular phase of the upgrade process. The possible options for the '`phase`' argument are `pre-upgrade`, `post-upgrade`.
  For instance, to execute a task file named '01_initialize' in the pre-upgrade phase, use the command:
  `csadm upgrade execute-task --target-version 7.6.0 --phase pre-upgrade --task-name 01_initialize`
  This option is useful for testing custom task files or modifications in existing task files.
  Note the following important points
  - Tasks from the 'pre-upgrade' phase can only be executed when the target version is higher than the current version.
  - Tasks from the 'post-upgrade' phase can be executed when the target version is higher than or equal to the current version.
  - Tasks from the 'upgrade' phase cannot be executed using this command.

  **NOTE**: After running the check-readiness command to assess their readiness for upgrading to version 7.6.0 or later, users on release 7.5.0 will also see the execute-task argument. However, users on 7.5.0 cannot use the csadm upgrade command with this argument. To utilize this argument, run the following command using "python3":
  `/opt/fsr-elevate/elevate/.env/bin/python3 /opt/fsr-elevate/elevate/main.py execute-task[--target-version TARGET_VERSION] [--phase PHASE] [--task-name TASK_NAME]`
  For example, to run a task file named '01_remove_security_patch_versions' in the post-upgrade phase, use the following command:
  `/opt/fsr-elevate/elevate/.env/bin/python3 /opt/fsr-elevate/elevate/main.py execute-task --target-version 7.6.0 --phase post-upgrade --task-name 01_remove_security_patch_versions`
- `execute-phase --target-version [TARGET_VERSION] --phase [PHASE]`: This option executes the specified phase in the upgrade process of the given version. The possible options for the '`phase`' argument are `pre-upgrade`, `post-upgrade`.
  For example, `csadm upgrade execute-phase --target-version 7.6.4 --phase pre-upgrade` executes the pre-upgrade phase of upgrading your FortiSOAR instance to release 7.6.4. This assists in anticipating problems and taking proactive measures to fix them before moving forward with the full upgrade.
  The 'post-upgrade' phase can be executed after upgrading the FortiSOAR instance to the target version. This phase will not be executed if the target version is not the same as the current version. For example, if your FortiSOAR instance is on version 7.6.4, then you can run post-upgrade for version 7.6.4 using the command `csadm upgrade execute-phase --target-version 7.6.4 --phase post-upgrade`.
  However, if you are on lower version than 7.6.4 such as 7.5.0, 7.5.1, 7.6.0, 7.6.1, and so on, and you try to run post-upgrade phase for target-version 7.6.4 , it will fail with a message such as the command `csadm upgrade execute-phase --target-version 7.6.4 --phase post-upgrade` will fail with a message such as "`The post-upgrade phase can only be executed for the 7.6.4 version after upgrading FortiSOAR to the 7.6.4 version. The current version is 7.6.1, and the post-upgrade phase can be executed for the current version.`"
  Additionally, if the `execute-phase` or `execute` options encounter a failure while executing a specific task, the subsequent execution of the same phase starts with the tasks that failed. Tasks that completed successfully prior to the failure are not executed again.
- `execute --target-version [TARGET_VERSION] [--download-packages] [--local-download-directory [LOCAL_DOWNLOAD_DIRECTORY]]`: This option downloads packages that are required during the upgrade of your FortiSOAR instance to the specified release on your system.
  Use the `csadm upgrade execute --target-version 7.6.4 --download-packages` command to download

upgrade packages required to upgrade your system to 7.6.4 to the default `/opt/cyops/packages` directory. Additionally, you can use the `--local-download-directory` argument to specify the absolute path in the local directory where the upgrade packages should be downloaded.

Use the `csadm upgrade execute --target-version [TARGET_VERSION]` command to upgrade your system. For example, to upgrade FortiSOAR from release 7.6.0 to release 7.6.4 after downloading the upgrade packages locally, run the `csadm upgrade execute --target-version 7.6.4` command.

**NOTE**: When you execute the command `csadm upgrade execute --target-version <TARGET_VERSION>`, a log file named '`upgrade-fortisoar-<target_version><timestamp>.log`' is created in the `/var/log/cyops` folder. For example, running the command `csadm upgrade execute --target-version 7.6.4` will generate the `upgrade-fortisoar-7.6.4-2024-05-22-1708597271.log` file in the `var/log/cyops` folder. This log file contains the complete CLI output, allowing you to review all the steps of the FortiSOAR upgrade process and can also be viewed during a '`tmux`' session. You can also use '`tail -f`' to monitor the update from a different system.

If a failure occurs during the upgrade process, the upgrade process is terminated, and errors are logged in '`upgrade-fortisoar-<target_version><timestamp>.log` file. Resolving these errors and executing the upgrade command again resumes the process from the point of failure.

- `create-task --phase [PHASE] --task-name [TASK_NAME] --cls-name [CLS_NAME]`: This option adds a new task file to the specified upgrade phase based on the task name and class name you have specified.
- `create-shell-script --phase [PHASE] --shell-script-name [SHELL_SCRIPT_NAME]`: This option adds a new shell script file to the specified upgrade phase based on the shell script name you have specified.

## Example of a check-readiness report

```
{
    "metadata": {
        "Current FortiSOAR version": "7.5.0",
        "File creation time": "24/06/2024, 06:40:03",
        "File modification time": "24/06/2024, 07:05:26"
    },
    "Verify Operating System Compatibility": {
        "result": true,
        "message": "The current operating system is Rocky Linux, which is supported for upgrade."
    },
    "Verify Yum Repo Connection": {
        "result": true,
        "message": "Connection to 'https://repo.fortisoar.fortinet.com' repo is successful"
    },
    "Verify Instance Type Compatibility": {
        "result": true,
        "message": "Current instance is of type 'enterprise'."
    },
    "Check '/' Directory Free Space": {
        "result": true,
        "message": "Required free space is available in '/'."
    },
    "Check '/boot' Directory Free Space": {
        "result": true,
        "message": "Required free space is available in '/boot'."
    },
    "Check '/var/log' Directory Free Space": {
        "result": true,
        "message": "Required free space is available in '/var/log'."
    },
    "Check '/opt' Directory Free Space": {
        "result": true,
        "message": "Required free space is available in '/opt'."
    },
    "Check '/var/tmp' Directory Free Space": {
        "result": true,
        "message": "Required free space is available in '/var/tmp'."
    },
    "Verify Cyops RPM Installation": {
        "result": true,
        "message": "'cyops' rpm is installed on this instance."
    },
    "Verify Publish Status Of All Modules": {
        "result": true,
        "message": "All modules in current system are in published state"
    },
    "Verify presence of cluster": {
        "result": true,
        "message": "No other cluster nodes found."
    },
    "Install Cyops Repo Update": {
        "result": true,
        "message": "Successfully installed /opt/fsr-elevate/elevate/cyops-repo-update-
7.6.0.el9.x86_64.rpm."
    }
}
```

The '`metadata`' key in the check-readiness report contains the following data:

- The "`Current FortiSOAR version`" key contains the version of FortiSOAR on which the report is generated
- The "`File creation time`" key contains the date and time when the report was generated.
- The "`File modification time`" key contains the date and time when the report was modified upon rerunning the `csadm upgrade check-readiness` command. It will be empty when the report is first generated.

# Upgrading a FortiSOAR High Availability Cluster

This section describes the procedure to upgrade a FortiSOAR High Availability (HA) cluster, assuming the HA setup includes a Reverse Proxy or Load Balancer, such as "HAProxy".

Before you start upgrading your FortiSOAR HA cluster, refer to the Preparing to Upgrade FortiSOAR section to ensure all the prerequisites are met. The upgrade installer will manage all FortiSOAR services.

Starting from release 7.6.1, FortiSOAR supports rolling upgrades for high availability (HA) clusters, reducing downtime from approximately 30 minutes to just 2 minutes. This optimization ensures minimal disruption during upgrades.

If you are upgrading your FortiSOAR HA cluster from releases 7.6.0, 7.5.1, or 7.5.0 to release 7.6.1 or later, follow the steps in the Upgrading FortiSOAR High Availability Cluster for releases prior to 7.6.1 topic.
If you are upgrading your FortiSOAR HA cluster from release 7.6.1 to release 7.6.2 or later such as releases 7.6.3, 7.6.4, you can use the 'rolling upgrades' process. Steps for rolling upgrade are mentioned in the Upgrading FortiSOAR High Availability Cluster for releases post 7.6.1 topic.

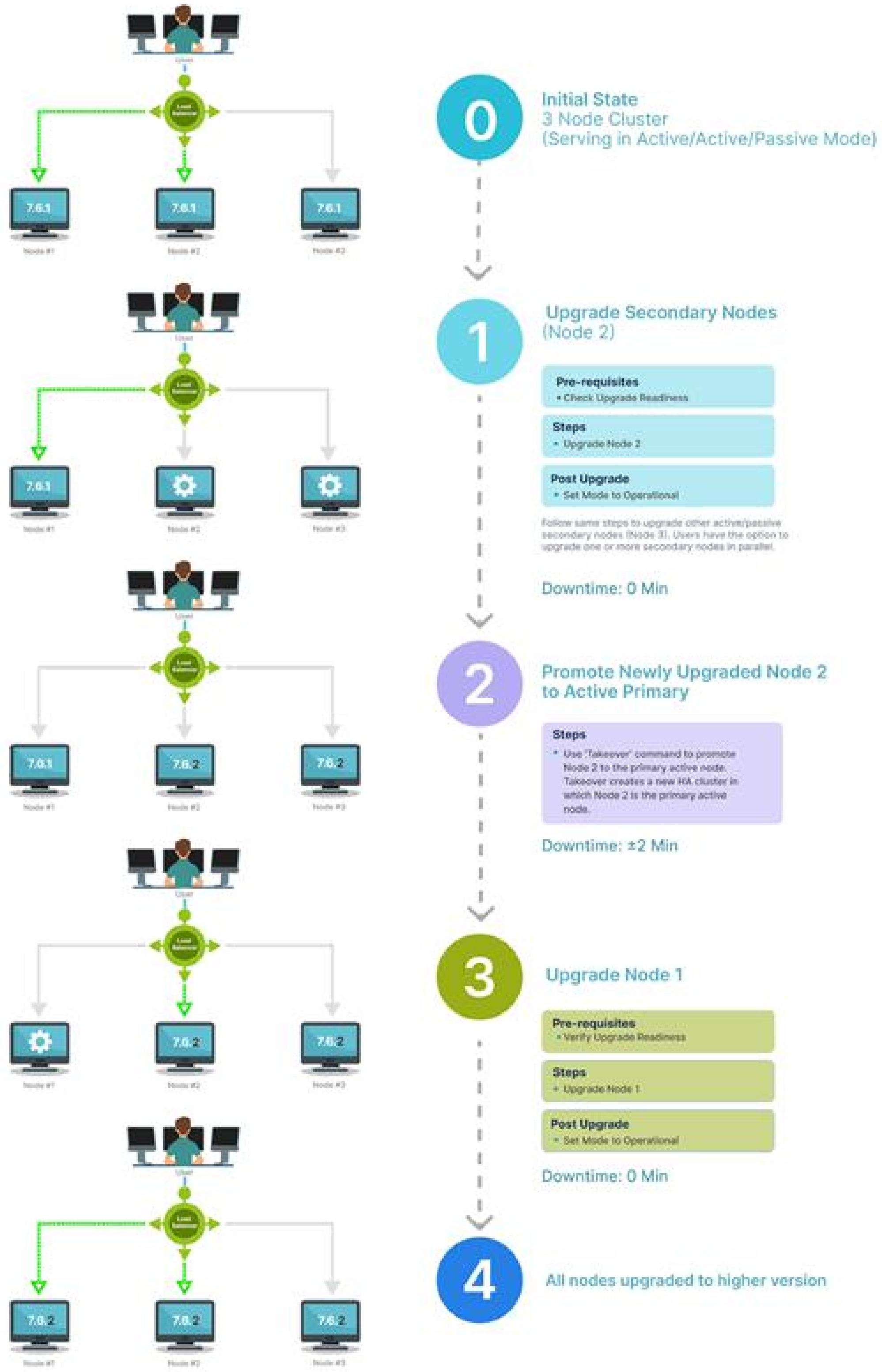
## Upgrading FortiSOAR High Availability Cluster for releases post 7.6.1

This section outlines the procedure for upgrading a FortiSOAR HA cluster for releases after 7.6.1 (e.g. from 7.6.1 to 7.6.4). The upgrade steps are the same for both configurations, i.e., Active-Active or Active-Passive HA clusters.

For the purpose of the following procedure:

- *Node1* is set as the Active Primary node
- *Node2* is set as the Active Secondary node,
- *Node 3* is set as the Passive Secondary node.
  **NOTE**: All the nodes are fronted by a Reverse Proxy or Load Balancer, such as "HAProxy".

The following diagram provides a high-level overview of the rolling upgrade process. The 'Rolling Upgrade' feature was introduced in FortiSOAR release 7.6.1, and it applies to all subsequent releases.



Initial State
3 Node Cluster
(Serving in Active/Active/Passive Mode)

Upgrade Secondary Nodes
(Node 2)

**Pre-requisites**
- Check Upgrade Readiness

**Steps**
- Upgrade Node 2

**Post Upgrade**
- Set Mode to Operational

Follow same steps to upgrade other active/passive secondary nodes (Node 3). Users have the option to upgrade one or more secondary nodes in parallel.

Downtime: 0 Min

Promote Newly Upgraded Node 2 to Active Primary

**Steps**
- Use 'Takeover' command to promote Node 2 to the primary active node. Takeover creates a new HA cluster in which Node 2 is the primary active node.

Downtime: ±2 Min

Upgrade Node 1

**Pre-requisites**
- Verify Upgrade Readiness

**Steps**
- Upgrade Node 1

**Post Upgrade**
- Set Mode to Operational

Downtime: 0 Min

All nodes upgraded to higher version

### Procedure

> Perform all steps in this procedure on all secondary active nodes, except point 3, which is the 'Takeover' operation. The 'Takeover' operation should be performed on only one of the upgraded active secondary nodes.

1. Upgrade the active secondary node, *Node 2*, as follows:
   a. (Recommended) Run the following command to verify if *Node 2* is ready for an upgrade to the target release:
   `csadm upgrade check-readiness --target-version [TARGET_VERSION]`
   For example, `csadm upgrade check-readiness --target-version 7.6.4`
   For details on the `check-readiness` option, see the Upgrading a FortiSOAR Enterprise Instance chapter.
   b. Upgrade *Node 2* using the following command:
   `csadm upgrade execute --target-version [TARGET_VERSION]`
   For example to upgrade to release 7.6.4, use the `csadm upgrade execute --target-version 7.6.4` command.
   In the case of our example, while *Node 2* is being upgraded, traffic will be directed to healthy nodes by the load balancer, i.e., *Node 1* will continue to serve requests, ensuring no downtime. Also, as the nodes are in cluster, data replication will synchronize data on *Node 2*.
   After running the command, the Upgrade Framework begins the upgrade process on *Node 2* and its UI becomes inaccessible. Note that during the upgrade process, some operations on other active nodes in the cluster become temporarily unavailable. These operations include:
      - Publishing of modules
      - Creating a connector
      - Uploading a connector
      - Installing/Uninstalling a connector
      - Publishing a connector/widget
      - Deleting code within a connector/widget

   During the upgrade process, a toaster message will be displayed on the UI of the other nodes in the cluster:
   `'A node in the HA cluster is undergoing an upgrade, temporarily affecting operations such as publishing modules, creating or uploading connectors, installing or uninstalling connectors, and publishing or deleting connectors and widgets.'`
   **NOTE**: If the UI remains unresponsive for operations such as publishing modules, installing connectors, etc., after the upgrade, log out and log back in to refresh the system.
2. After upgrading, verify *Node 2's* UI is accessible using *Node 2's* hostname or IP address (and not load balancer). Additionally, perform basic sanity checks and once the sanity checks on the system are completed, set *Node 2* to 'Operational' using the following command:
   `csadm system env --mode operational`
   Setting the mode to 'Operational', unmasks and starts the `celeryd` service on *Node 2*. This command also checks all other nodes in the cluster that are not upgraded and sets their mode to 'Upgrade.' This ensures the load balancer directs traffic to the newly upgraded node while blocking traffic to nodes, which are not upgraded.
   Repeat this procedure for any other active or passive secondary nodes. Passive nodes, such as *Node 3*, can be upgraded at any time. You may upgrade one or more active/passive secondary nodes in parallel.
3. Run the `csadm ha takeover` command on the active secondary node, *Node 2* to promote it the Active Primary node. This will cause a brief downtime of approximately 2 minutes.
   **NOTE**: The 'Takeover' operation should be performed on only one of the upgraded active secondary nodes.
   **IMPORTANT**: You must promote an active secondary node to the active primary node, before upgrading

the original primary node to ensure that requests are served by the new active primary node.
In our example, you must promote *Node 2* before upgrading *Node 1* to ensure that requests are served by *Node 2* during the upgrade.
For details on the Takeover process, see the Takeover section in the *High Availability and Disaster Recovery support in FortiSOAR* chapter of the "Administration Guide."
Once the takeover is complete, *Node 2* will become the new primary node and begin serving requests. Users will then be prompted to join all other nodes in the cluster to the new primary, and all operational nodes will resume load sharing.

4. Upgrade the previous active primary node (*Node1*) using the same steps:
   - **a.** (Recommended) Verify if the node is ready for an upgrade to the target release: `csadm upgrade check-readiness --target-version [TARGET_VERSION]`
   - **b.** Upgrade the node: `csadm upgrade execute --target-version [TARGET_VERSION]`
   - **c.** Once *Node 1* is upgraded, verify that the FortiSOAR UI is accessible and perform basic sanity checks, using *Node 1's* hostname or IP address (and not load balancer).
   - **d.** *Only applicable for upgrades from release 7.6.1.* Create the RabbitMQ cluster using the following steps:
     - **i.** On the new primary node, run the following command to allowlist the previous primary node:
       `csadm ha allowlist --nodes <previous-primary-hostname>`
     - **ii.** On the previous primary node, run the following command to allowlist the new primary node:
       `csadm ha allowlist --nodes <new-primary-hostname>`
     - **iii.** Create the RabbitMQ cluster using one of the following methods:
       - *Join using the csadm mq command*:
         - **i.** Run the following command on the previous primary node:
           `csadm mq join-cluster --primary-node <new-primary-hostname`
       - *Join using the config file*:
         - **i.** On the new primary node, generate the config:
           `csadm ha export-conf`
         - **ii.** Copy the exported config file to the previous primary node.
         - **iii.** On the previous primary node, run the following command:
           `csadm mq join-cluster –-config-file <conf-file-path>`
   - **e.** Restart the services on the previous primary node:
     `csadm services --restart`
   - **f.** Set *Node 1* to 'Operational' mode:
     `csadm system env --mode operational`

# Troubleshooting: Cluster rejoin fails on new primary node during rolling upgrade in HA Cluster

During a rolling upgrade of a High Availability (HA) cluster from version 7.6.2 to 7.6.4, you may encounter a failure when attempting to rejoin nodes to the new primary.

When the user attempts to run `join-cluster` manually from the previous primary or any secondary node(s), which is on release 7.6.2, the operation fails with the following error message:

```
"Validating SSH credentials
Fetching HA configuration details for qa-env4-swati.fortisoar.in
```

```
[2025-10-06 12:34:49] ERROR manager: start(): 492: -----------------------------------------------
-
Error:
There was an issue accessing all files in the PostgreSQL data directory of the primary node.
Please ensure that all files in the PostgreSQL directory have the correct permissions assigned to
the postgres user to avoid operation failure."
```

Steps to resolve this issue:

1. On the current active primary node, export the HA configuration using the following command:
   `csadm ha export-conf`
2. On the previous primary node or another secondary node where the issue is occurring, import the exported `ha.conf` file.
3. Use the command below to rejoin the node as a secondary in the cluster:
   `csadm ha join-cluster --status active --role secondary --conf ha.conf`
4. Check the cluster status to confirm the node has successfully rejoined as a secondary node.

# Upgrading FortiSOAR High Availability Cluster for releases prior to 7.6.1

This section outlines the procedure for upgrading a FortiSOAR HA cluster for releases prior to 7.6.1, such as from 7.6.0 to 7.6.1 or later.

## Upgrading an Active-Active HA Cluster

This topic provides instructions for upgrading an Active-Active High Availability (HA) cluster for releases prior to 7.6.1. In this configuration, *Node1* acts as the Active Primary node, and *Node2* is the Active Secondary node. Both the nodes are fronted by a Reverse Proxy or Load Balancer such as "HAProxy".

Upgrading the primary node will result in downtime, which varies based on the amount of data on your system. Additionally, it is highly recommended to upgrade the primary node immediately after the secondary node.

### Prerequisites

Before beginning the upgrade, adjust the `wal_keep_size` parameter on all nodes, to prevent WAL rotation during the process:

1. Open the PostgreSQL configuration file:
   `/var/lib/pgsql/16/data/postgresql.conf`
2. Increase the wal_keep_size to 15GB:
   `wal_keep_size = 15360`

3. Reload PostgreSQL configuration to apply the changes:
   `systemctl reload postgresql-16`

## Procedure

To upgrade your active-active HA cluster from releases prior to 7.6.1 to release 7.6.1 or later, for example upgrading FortiSOAR release 7.6.0 to release 7.6.2, perform the following steps:

1. Set the Reverse Proxy to direct all requests to *Node1*.
   This ensures that FortiSOAR traffic is handled by *Node1*, allowing *Node2* to be upgraded without interrupting service.
2. Log in as the '*root*' user and suspend the cluster on *Node2*:
   `csadm ha suspend-cluster`
   This command isolates *Node2*, enabling it to be upgraded.
3. Upgrade *Node2* using the procedure outlined in the Upgrading a FortiSOAR Enterprise Instance chapter.
4. (Optional) Upgrade any remaining secondary nodes, by following the same process used for *Node2*. To do this, first run the `csadm ha suspend-cluster` on each remaining secondary node, and then upgrade the nodes using the procedure outlined in the Upgrading a FortiSOAR Enterprise Instance chapter.
5. Once *Node2* and all other secondary nodes (if any) are upgraded, proceed with upgrading *Node1* using the process described in the Upgrading a FortiSOAR Enterprise Instance chapter.
   **Important**: Upgrading *Node1* will incur downtime.
6. Add the host names of all HA cluster nodes (except for the node itself) to the 'allowlist' using the following command:
   `csadm ha allowlist --nodes <comma-separated list of host names>`
   For example, on *Node 1* run `csadm ha allowlist --nodes <hostname of Node2>`
   For example, on *Node 2* run `csadm ha allowlist --nodes <hostname of Node1>`
   On a 3-node HA system, on the primary node run `csadm ha allowlist --nodes <all-secondary-hostnames>`
   On a 3-node HA system, on the secondary node run `csadm ha allowlist --nodes <all-secondary-hostname-except-itself>,<primary-hostname>`
7. Once both the nodes are upgraded, run the following command on *Node2* to resume the HA cluster:
   `csadm ha resume-cluster`
8. Create the MQ cluster by running the following command on each secondary node:
   `csadm mq join-cluster --primary-node <primary-hostname>`
9. Restart all services after the MQ cluster has been created on each secondary node:
   `csadm services --restart`
10. Adjust the Reverse Proxy settings to route requests to both *Node1* and *Node2*.

## Post-Upgrade

After upgrading all the nodes and resuming the cluster, you can verify that the replication lag is zero by running the following command:
`csadm ha get-replication-stat`

Additionally, after the upgrade, FortiSOAR restores the `wal_keep_size` parameter to its original value of 1GB (`wal_keep_size = 1024`).

# Upgrading an Active-Passive HA Cluster

This topic provides instructions for upgrading an Active-Passive High Availability (HA) cluster for releases prior to 7.6.1. In this configuration, *Node1* acts as the Active Primary node, and *Node2* is the Passive Secondary node. Both the nodes are fronted by a Reverse Proxy or Load Balancer such as "HAProxy".

Upgrading the primary node will result in downtime, which varies based on the amount of data on your system. Additionally, it is highly recommended to upgrade the primary node immediately after the secondary node.

## Prerequisites

Before starting the upgrade, adjust the `wal_keep_size` parameter, on all nodes, to prevent WAL rotation, as outlined in the Prerequisites section of the *Upgrading an Active-Active HA Cluster* topic.

## Procedure

To upgrade your active-passive HA cluster from releases prior to 7.6.1 to release 7.6.1 or later, for example upgrading FortiSOAR release 7.6.0 to release 7.6.2, perform the following steps:

1. Log in as the '*root*' user and suspend the cluster on *Node2*:
   ```
   csadm ha suspend-cluster
   ```
   This command isolates *Node2*, enabling it to be upgraded.
2. Upgrade *Node2* using the process mentioned in the Upgrading a FortiSOAR Enterprise Instance chapter.
3. (Optional) Upgrade any remaining secondary nodes, by following the same process used for *Node2*. To do this, first run the `csadm ha suspend-cluster` on each remaining secondary node, and then upgrade the nodes using the procedure outlined in the Upgrading a FortiSOAR Enterprise Instance chapter.
4. Once *Node2* and all other secondary nodes (if any) are upgraded, proceed with upgrading *Node1* using the process described in the Upgrading a FortiSOAR Enterprise Instance chapter.
   **Important**: Upgrading *Node1* will incur downtime.
5. Add the host names of all HA cluster nodes (except for the node itself) to the allowlist using the following command:
   ```
   csadm ha allowlist --nodes <comma-separated list of host names>
   ```
6. Once both the nodes are upgraded, run the following command on *Node2* to resume the HA cluster:
   ```
   csadm ha resume-cluster
   ```

## Post-Upgrade

After upgrading all the nodes and resuming the cluster, restore the `wal_keep_size` setting to its original value, on all nodes, as described in the Post-Upgrade section of the *Upgrading an Active-Active HA Cluster* topic.

# Upgrading a FortiSOAR Distributed Multi-Tenancy Configuration

This section describes the procedure to upgrade a FortiSOAR distributed multi-tenant configuration for managed security services providers (MSSPs) or Distributed SOC configuration.

You must first upgrade the master node of your FortiSOAR distributed multi-tenant configuration and only then upgrade the tenant nodes of your FortiSOAR multi-tenancy setup.

In case of a distributed deployment, both the master and the tenant nodes must be upgraded.

## Upgrading a FortiSOAR master node

Before you upgrade your FortiSOAR master node, ensure the following:

- All playbooks have completed their execution on the master.
- The tenant node(s) are deactivated from the master node before upgrading the master node, and tenant nodes have disabled communication to the master node from the "`Master Configuration`" page.

If the master node of your multi-tenant configuration is part of an HA setup, i.e., MSSP +HA, then follow the steps mentioned in the Upgrading a FortiSOAR High Availability Cluster chapter.

If the master node of your multi-tenant configuration is not part of an HA setup, then follow the steps mentioned in the Upgrading a FortiSOAR Enterprise Instance chapter.

## Upgrading a FortiSOAR Tenant node

Before you upgrade your FortiSOAR tenant node, ensure the following:

- Data replication from the tenant node to the master node is stopped. You can stop data replication by logging on to the tenant node and clicking **Settings** to open the `System` page, then in the `Multi Tenancy` section, click the **Master Configuration** menu item and then in the `Communication With Master Node` section, toggle the **Enabled** button to **NO**.
- All playbooks have completed their execution on the tenant.
- All schedule playbooks that fetch data from data sources to the tenant are stopped.
- Any application that pushes data from data sources to the tenant is stopped.

If the tenant node of your multi-tenant configuration is part of an HA setup, i.e., MSSP +HA, then follow the steps mentioned in the Upgrading a FortiSOAR High Availability Cluster section.

If the tenant node of your multi-tenant configuration is not part of an HA setup, then follow the steps mentioned in the Upgrading a FortiSOAR Enterprise Instance section.

After the tenant node has been successfully upgraded, you must toggle the **Allow Module Management** setting to **NO** and then back to **YES**. This is needed only if you were already using the 'Allow Module Management' feature and is required to synchronize the tenant module metadata with the master instance. You can ignore this step, if your 'Allow Module Management' setting was already disabled before the upgrade.

# Upgrading a FortiSOAR Secure Message Exchange

A secure message exchange establishes a secure channel that is used to relay information to the agents or tenant nodes. To create a dedicated secure channel, you are required to add the reference of the installed and configured secure message exchange, when you add agent or tenant nodes to your environment. For information on agents see the *Segmented Network support in FortiSOAR* chapter in the "Administration Guide," and for more information on secure message exchange and tenants, see the "Multi-Tenancy support in FortiSOAR Guide".

1. Ensure that you stop data replication between the master and the tenant nodes. You can stop data replication by logging on to the tenant node and clicking **Settings** to open the `System` page, then in the `Multi Tenancy` section, click the **Master Configuration** menu item and then in the `Communication With Master Node` section, toggle the **Enabled** button to **NO**.
2. SSH to the secure message exchange VM that you want to upgrade.
3. Check your system to see if `tmux` is installed; if not, use the following command to install it
   ```
   sudo yum install -y tmux
   ```
   Next, check that you are connected to a `tmux` session. A tmux session is needed for situations where network connectivity is less than favorable. You can check your tmux session using the following command:
   ```
   # tmux ls
   ```
   This command returns an output such as the following example:
   ```
   0: 1 windows (created Thu Nov 24 09:37:47 2022) [170x47]
   ```
   Log back into the SSH console and run the following command to reattach the `tmux` session:
   ```
   tmux attach-session -t 0
   ```
   If you do not find any `tmux` session, connect to one using the following command:
   ```
   # tmux
   ```
4. (Recommended) Upgrade the secure message exchange to the target release:
   - If your secure message exchange version 7.5.0, 7.5.1, or 7.5.2, then upgrade your external secure message exchange to version 7.6.4 as follows:
     i. Download the `upgrade-fortisoar-7.6.4.bin` script from the FortiSOAR repo using `wget`
     ii. Run the `upgrade-fortisoar-7.6.4.bin` script using the "sh" command under `tmux`

- If your secure message exchange version is 7.6.0 or 7.6.1, then upgrade your external secure message exchange to version 7.6.4 as follows:
  - **i.** Run the following command to verify if the secure message exchange is ready for an upgrade to the target release:
    `csadm upgrade check-readiness --target-version [TARGET_VERSION]`
    For details on the `csadm upgrade` command, see the Upgrading a FortiSOAR Enterprise Instance chapter.
  - **ii.** (Recommended) From release 7.6.2 onwards, you can choose to download upgrade packages locally first and then upgrade your SME at a later time, use the following command:
    `csadm upgrade execute --target-version [TARGET_VERSION]--download-packages][--local-download-directory [LOCAL_DOWNLOAD_DIRECTORY]].`
    By default, the upgrade packages are downloaded to the `/opt/cyops/packages/fsr-packages` folder.
    **NOTE**: If the upgrade packages are successfully downloaded to the local directory, the upgrade process will utilize those packages for the upgrade. Otherwise, the upgrade process will access the `yum` repository for the upgrade.
    Once the upgrade is complete, any packages downloaded during the upgrade process will be discarded. FortiSOAR will no longer reference these local packages and will instead use the configured `yum` repositories to check for and install future updates.
    Additionally, note that if you re-run the `csadm upgrade` command with the `--download-packages` argument, it will download the packages again, overwriting the previously downloaded ones.
  - **iii.** Upgrade the secure message exchange:
    `csadm upgrade execute --target-version [TARGET_VERSION]`

5. After successfully upgrading the secure message exchange, restart data replication between the master and tenant nodes by clicking **Master Configuration** in the left navigation. Then, in the **Communication With Master Node** section, toggle the **Data Replication** button to **ON**. Finally, verify that the replication is working correctly.

# Upgrading a FortiSOAR Secure Message Exchange Cluster

RabbitMQ supports clustering, which, when combined with Queue Mirroring, enables an Active-Active configuration. For detailed setup instructions and guidance on monitoring queues, see the Clustering Guide and the Highly Available (Mirrored) Queues article. For optimal performance, the clustered instances should be managed by a TCP Load Balancer such as HAProxy, and clients should connect to the cluster via the proxy's address. For more information, see the *Multi-tenancy support in FortiSOAR* guide.

This procedure covers upgrading a two-node mirrored MQ cluster, both configured with the Reverse Proxy.

1. Configure the Reverse Proxy to route requests exclusively to *Node1*, which is the primary node of the MQ cluster.
   This ensures *Node1* handles all requests, while *Node2* (the secondary node) is available for maintenance.

2. Before upgrading, break the cluster on the secondary node (*Node2*) by executing the following commands:
   a. `rabbitmqctl stop_app`
   b. `rabbitmqctl reset`
   c. `rabbitmqctl start_app`
3. Log into the *Node2* terminal session as the `root` user, and upgrade *Node2* following the steps in the Upgrading a FortiSOAR Secure Message Exchange section.
4. Remove the *Node1* entry from the Reverse Proxy.
   **NOTE**: Downtime begins at this stage.
5. Log into *Node1* terminal session as the `root` user, and upgrade *Node1* following the steps in the Upgrading a FortiSOAR Secure Message Exchange section.
6. Add the *Node1* entry back to the Reverse Proxy.
   **NOTE**: Downtime ends at this stage.
7. Once the SME cluster is upgraded, use the `join-cluster` command to create the SME cluster. For details, see the Setting up High Availability of the Secure Message Exchange topic in the *Multi-tenancy support in FortiSOAR* guide.
8. Reconfigure the Reverse Proxy to load balance requests between *Node1* and *Node2*.

# Troubleshooting upgrade issues for MSSP setups

## Replication from tenant to master stops once you upgrade an MSSP with an HA setup

If you have upgraded an MSSP+HA setup, then post-upgrade the replication from tenant nodes to the master node stopped.

**Resolution**

To resolve this issue, once you have upgraded your MSSP setup and created the HA cluster, you must restart all services on the primary master node and the primary tenant node using the following command:

```
csadm services --restart
```

## Troubleshooting issues after running the `db flush` command

If you have reset the RabbitMQ node, you must reconfigure the server using the `csadm mq db flush` command. However, running this command on a node that is already part of a RabbitMQ cluster requires additional steps:

**Resolution**

1. On a different active node within the cluster, run the following command to forget the node where the `db flush` command was executed:
   ```
   csadm mq forget-node --nodename <rabbit@hostname>
   ```
2. On the node where you ran the `db flush` command, run the following command:
   ```
   csadm mq join-cluster --primary-node 'other-cluster-nodes-hostname'
   ```

And then restart all the services using the following command:

```
csadm services --restart
```

## Post-upgrade, FSR Agent status remains 'Awaiting Remote Node'

After upgrading FortiSOAR, the agent status remains stuck at 'Awaiting Remote Node' instead of 'Remote Node Connected'. This indicates that the agent is unable to establish a connection with the node using the SME.

**Resolution**

To resolve this issue and update the agent status to 'Remote Node Connected' after upgrading your MSSP setup, restart the RabbitMQ service on the external SME instance using the following command:

```
systemctl restart rabbitmq-server
```

## Embedded SME status displays as 'Not Configured' after upgrading FortiSOAR

After upgrading FortiSOAR, the embedded SME status might appear as 'Not Configured'.

**Resolution**

To update the embedded SME status to 'Configured', navigate to the `Secure Message Exchanges` page and refresh the SME list grid by clicking the **Refresh** button.

# Upgrading FortiSOAR using the Offline Repository

1. Ensure that the offline repository host is accessible from the FortiSOAR appliance and to ensure that the upgrade is not affected if the session times out, run the `tmux` command:
   `[root@localhost ~]# tmux`
2. If you are using your private repository to upgrade FortiSOAR, then specify the offline repo URL in the "`custom_yum_url`" key that is present in the `/opt/cyops/configs/fsr-elevate/config.yml` file.
3. Upgrade to FortiSOAR 7.6.4 using the process mentioned in the Upgrading a FortiSOAR Enterprise Instance chapter.
4. If you are using a self-signed certificate, then you must add your custom CA certificate in the OS and python trust store as a trusted certificate. For detailed steps, see the Adding a custom CA (self-signed) certificate in Rocky Linux or RHEL as a trusted certificate topic in the Additional configuration settings for FortiSOAR chapter of the "Deployment Guide."

# Upgrading your FortiSOAR Docker image, FortiSOAR Docker HA, and upgrading your FortiSOAR Docker on an Amazon Elastic Kubernetes Cluster

> Do not use the 'Upgrade Framework' to upgrade the Docker images. Instead, follow the steps outlined in this chapter to upgrade Docker instances.
> Additionally, also note that the 'rolling upgrade' process, which minimizes downtime for high availability (HA) clusters, ***is not supported*** for Docker images.

## Upgrading your FortiSOAR Docker image

1. Download the FortiSOAR docker image from https://support.fortinet.com; details are in the Downloading the FortiSOAR Docker image section of the "Deployment Guide".
2. Load the downloaded Docker image using the following command:
   ```
   docker load -i <image-path>
   ```
3. Download the FortiSOAR Docker installer from `https://repo.fortisoar.fortinet.com/7.6.4/install-fortisoar-docker-<release_version>.bin`
   For example, `https://repo.fortisoar.fortinet.com/7.6.4/install-fortisoar-docker-7.6.4.bin`
4. Update the `fortisoar.env` file with the ID of the Docker Image that is loaded in step 2. For more information, see Understanding the `fortisoar.env` file topic in the *Deploying FortiSOAR on a Docker Platform* chapter in the "Deployment Guide."
   **Important**: Ensure that the value of the `PROJECT_NAME` field in the `fortisoar.env` file must be the same as the value in the earlier version of the Docker image.
5. Before you begin the upgrade, it is recommended to take a backup of your FortiSOAR Docker as listed in the following commands.
   **Note**: The following commands uses `fortisoar_fortisoar_1` as the Docker name. You must replace this sample name with your own Docker name, which you can find using the `docker ps` command.
   a. 
   ```
   docker exec -ti fortisoar_fortisoar_1 bash -c 'export LANG=en_US.UTF-8;csadm db --backup /home/csadmin'
   ```
   **Note**: This command stores the backup file at `/home/csadmin/DR_BACKUP_<release_version>-*_*_*.tgz` (for example, `/home/csadmin/DR_BACKUP_7.6.4-*_*_*.tgz`) inside your FortiSOAR Docker.
   b. Copy the backup file from your FortiSOAR Docker on the Docker host using the following command:
   ```
   docker cp fortisoar_fortisoar_1:/home/csadmin/DR_BACKUP_<release_version>-*_*_*.tgz /data
   ```
   For example, `docker cp fortisoar_fortisoar_1:/home/csadmin/DR_BACKUP_7.6.4-*_*_*.tgz`

`/data`
**Note**: The FortiSOAR Docker backup file is stored in the `/data` directory on your Docker host.

6. Stop your FortiSOAR Docker using the following command:
   `docker stop fortisoar_fortisoar_1`
7. Remove your FortiSOAR Docker using the following command:
   `docker rm fortisoar_fortisoar_1`
8. Run the FortiSOAR Docker using the updated `fortisoar.env` file that contains the ID of the new Docker Image using the following command:
   `./install-fortisoar-docker-<release_version>.bin --env-file fortisoar.env`
   For example, `./install-fortisoar-docker-7.6.4.bin --env-file fortisoar.env`

After upgrading to FortiSOAR 7.6.3 or later, users running `sudo` commands as the 'csadmin' user will be prompted for a password on all systems except AWS.

# Upgrading your FortiSOAR Docker HA image

For the following procedure, *Node1* is considered as the Active Primary node, *Node2* is considered as the Active Secondary node. Both nodes are fronted by a Reverse Proxy or Load Balancer, such as "HAProxy".

Upgrading the primary node will result in downtime, which varies based on the amount of data on your system. Additionally, it is highly recommended to upgrade the primary node immediately after the secondary node.

To upgrade your active-active HA cluster from releases prior to 7.6.0 to FortiSOAR 7.6.2 or later perform the following steps:

1. Set the Reverse Proxy to direct all requests to *Node1*.
   This ensures that FortiSOAR requests are passed only to *Node1*, and *Node2* can be upgraded without interruption.
2. Use the `#csadm ha` command as a '*root*' user to run the `suspend-cluster` command on *Node2*.
   This command makes *Node2* a standalone system, allowing for it to be upgraded.
3. Upgrade *Node2* using the process mentioned in the Upgrading your FortiSOAR Docker image topic.
   **Important**: The `fortisoar.env` file must be updated as per the information present in the FortiSOAR High Availability Support on Dockers topic in the *Deploying FortiSOAR on a Docker Platform* chapter of the "Deployment Guide."
   Once *Node2* is upgraded, proceed to upgrade *Node1*.
   **Note**: Upgrading *Node1* will incur downtime.
4. Add the host names of all HA cluster nodes (except for the node itself) to the allowlist using the following command:
   `csadm ha allowlist --nodes <comma-separated list of host names>`
   For example, on *Node 1* run `csadm ha allowlist --nodes <hostname of Node2>`
   For example, on *Node 2* run `csadm ha allowlist --nodes <hostname of Node1>`
   On a 3-node HA system, on the primary node run `csadm ha allowlist --nodes <all-secondary-hostnames>`

On a 3-node HA system, on the secondary node run `csadm ha allowlist --nodes <all-secondary-hostname-except-itself>,<primary-hostname>`

5. After upgrading both nodes, run the `resume-cluster` command on *Node2* to create the HA cluster.
6. SSH to the primary node and export its HA config file using the following command:
   ```
   csadm ha export-conf
   ```
7. Create the MQ cluster by running the following command on each secondary node:
   ```
   csadm mq join-cluster --conf-file <ha.conf file>
   ```
8. Restart all services using the following command:
   ```
   csadm services --restart
   ```
9. Adjust the Reverse Proxy settings to route requests to both *Node1* and *Node2*.

# Reverting the upgrade on your FortiSOAR Docker image

In case the FortiSOAR Docker image upgrade fails, and you want to revert to the previous release, then you need to restore the backup of the previous version that was taken in the `/data` directory on your Docker host. Following are the steps for restoring the backup.

**Note**: The following commands uses `fortisoar_fortisoar_1` as the Docker name. You must replace this sample name with your own Docker name, which you can find using the `docker ps` command.

1. Stop the running FortiSOAR Docker using the following command:
   ```
   docker stop fortisoar_fortisoar_1
   ```
2. Remove the FortiSOAR Docker using the following command:
   ```
   docker rm fortisoar_fortisoar_1
   ```
3. Remove the FortiSOAR Docker volumes using the following command:
   ```
   docker volume rm $(docker volume ls --filter name=fortisoar_fortisoar_* -q)
   ```
4. Update the `fortisoar.env` file with the ID of the previous Docker Image.
5. Run the previous version FortiSOAR Docker using the updated `fortisoar.env` file that contains the ID of the previous Docker Image using the following command:
   ```
   ./install-fortisoar-docker-<release_version>.bin --env-file fortisoar.env
   ```
   For example, `./install-fortisoar-docker-7.6.4.bin --env-file fortisoar.env`
6. Wait until you see EULA page on the UI at `https://<docker-host-hostname>:<PORT_UI>/`
7. Copy the FortiSOAR Docker backup file from the `/data` directory on your Docker host using the following command:
   ```
   docker cp /data/DR_BACKUP_<release_version>-*_*_*.tgz fortisoar_fortisoar_1:/home/csadmin/
   docker cp /data/DR_BACKUP_7.6.4-*_*_*.tgz fortisoar_fortisoar_1:/home/csadmin/
   ```
8. Restore the Docker image using the following command:
   ```
   docker exec -it fortisoar_fortisoar_1 bash -c “export LANG=en_US.UTF-8;csadm db --restore
   /home/csadmin/DR_BACKUP_<release_version>-*_*_*.tgz”
   ```
   For example, `docker exec -it fortisoar_fortisoar_1 bash -c “export LANG=en_US.UTF-8;csadm db --restore /home/csadmin/DR_BACKUP_7.6.4-*_*_*.tgz”`

# Upgrading your FortiSOAR Docker on an Amazon Elastic Kubernetes Cluster

1. Download the FortiSOAR docker image from https://support.fortinet.com; details are in the Downloading the FortiSOAR Docker image section of the "Deployment Guide".
2. Upload the downloaded FortiSOAR Docker image to your AWS Elastic container registry or any other Docker repository that is accessible from within your Kubernetes cluster. For example:
   ```
   # docker push <account-id>.dkr.ecr.<region>.amazonaws.com/fortisoar/fortisoar:7.6.4
   ```
3. Before you begin the upgrade process, it is recommended to take a backup of your FortiSOAR pod as listed in the following commands.
   The following commands uses `fsr-0` as the pod name. You must replace this sample name with your own pod name that you can find using the `#kubectl get pods -o=name -n fsr` command.
   ```
   #kubectl exec -ti -n fsr -c fsr fsr-0 -- bash -c "csadm db --backup /\home/\csadmin/\ "
   ```
   **Note**: This command stores the backup file in the `/home/csadmin/DR_BACKUP_<release_version>-*_*_*.tgz` folder inside your FortiSOAR pod. For example, `/home/csadmin/DR_BACKUP_7.3.2-*_*_*.tgz` inside your FortiSOAR pod.
   Copy the backup file from your FortiSOAR pod on the EKS cluster node to your machine using the following command:
   ```
   #kubectl cp fsr-0:/home/csadmin/DR_BACKUP_<release_version>-*_*_*.tgz -c fsr -n fsr DR_BACKUP_<release_version>.tgz
   ```
4. Stop the running FortiSOAR pod using the following command:
   ```
   #kubectl delete statefulset <statefulset_name> -n <fortisoar_namespace>
   ```
5. Update the path of the new FortiSOAR image in the `fortisoar-statefulset.yaml` file.
6. Run the following command to deploy a new statefulset with the latest docker image:
   ```
   #kubectl apply -f fortisoar-statefulset.yaml
   ```

After upgrading to FortiSOAR 7.6.3 or later, users running `sudo` commands as the 'csadmin' user will be prompted for a password on all systems except AWS.

# Post-Upgrade Tasks and Notes

## Handling Configuration Changes Post-Upgrade

After a software upgrade, new configuration parameters may be introduced. It's crucial to ensure that these changes are correctly integrated into your existing configuration files.

Follow these steps to verify and apply any configuration changes:

1. Check for `.rpmnew` files
   After a successful upgrade, verify if any `.rpmnew` files have been generated for the configuration files you noted during the pre-upgrade phase (see Preparing to Upgrade FortiSOAR chapter for details).
   These `.rpmnew` files usually contain updated configurations introduced in the new release. If a configuration file was previously modified, the system might generate a corresponding `.rpmnew` file during the upgrade.
   If you did not note all modified files earlier, you can locate all `.rpmnew` files, using the following command:
   ```
   cd /opt
   find . -type f -name *.rpmnew
   ```
2. Compare and merge configuration files
   If `.rpmnew` files are found, compare them with the corresponding active configuration files and manually merge any new settings.
   Example: Updating `das.ini`
   If `das.ini` was modified prior to the upgrade, perform the following steps:
   1. Navigate to the utilities directory:
      ```
      cd /opt/cyops-auth/utilities
      ```
   2. Check if `das.ini.rpmnew` exists:
      ```
      ls -l | grep das.ini.rpmnew
      ```
   3. If the `das.ini.rpmnew` file exists, compare it with the current `das.ini`:
      ```
      diff das.ini das.ini.rpmnew
      ```
   4. Identify any new sections or parameters in the `.rpmnew` file.
      Example of a new configuration:
      ```
      [APPLICATION]
      squeeze_space_notification=true
      sealab_config_location=/opt/cyops-workflow/sealab/sealab/config.ini
      ```
      In this example the `[APPLICATION]` section is new. Add this section to the current `das.ini` file.
3. Apply Changes
   After merging the new configurations, restart the relevant service to apply changes.
   In this case, restart the `cyops-auth` service:
   ```
   systemctl restart cyops-auth
   ```

www.fortinet.com