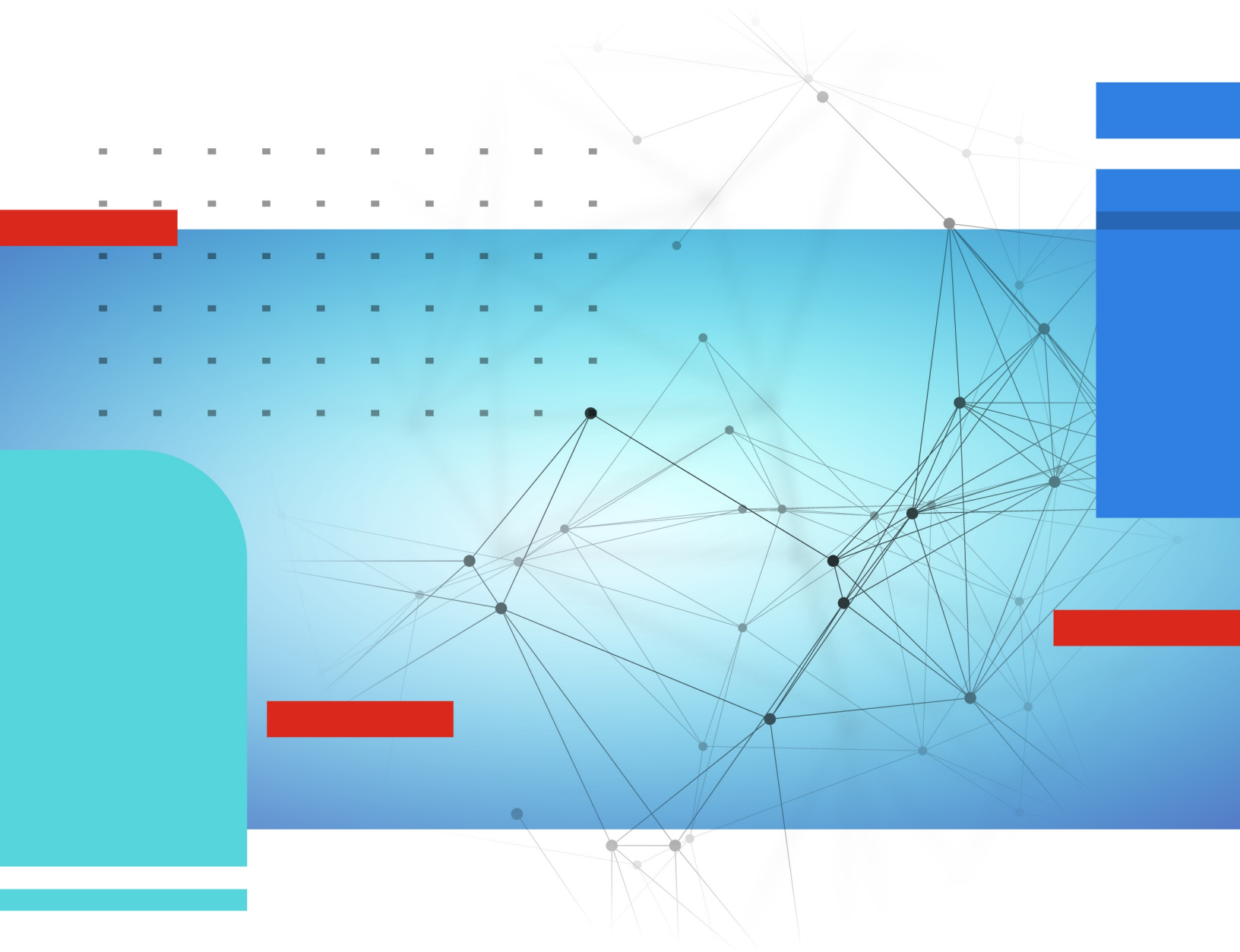
Upgrade Guide

FortiSOAR 7.6.1



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

December, 2024
FortiSOAR 7.6.1 Upgrade Guide
00-400-000000-20210416

TABLE OF CONTENTS

Change Log

Date	Change Description
2025-03-26	Added information to unmount any external mount entries before upgrading FortiSOAR in the Preparing to Upgrade FortiSOAR chapter.
2025-02-21	Updated the Introduction chapter to add that versions 7.5.0 and 7.5.1 can be upgraded to 7.6.1.
2025-02-03	Updated the Preparing to Upgrade FortiSOAR on page 6 chapter to add a note about upgrading a FortiSOAR instance with an external PostgreSQL database.
2024-12-19	Initial release of 7.6.1

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 7.6.1. 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 using Upgrade Framework
  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 7.6.1 from release 7.6.0, 7.5.1, or 7.5.0. Also, once you have upgraded your instance, you must log out from the FortiSOAR UI and log back into FortiSOAR.

Also, note that the upgrade procedure temporarily takes the FortiSOAR application offline while the upgrade operations are taking place. We recommend that you send a prior notification to all users of a scheduled upgrade as users are unable to log in to the FortiSOAR Platform during the upgrade. Release 7.6.0 optimizes the upgrade process to minimize application downtime and provide a faster and more reliable upgrade experience. For more information, see the Upgrading a FortiSOAR Enterprise Instance using the Upgrade Framework chapter.

Before you upgrade your FortiSOAR instance, it is highly recommended that you review the *Special Notices* chapter in the "Release Notes", so that you are aware of operational and breaking changes made in version 7.6.1.

For information on upgrading FortiSOAR using the offline repository and upgrading your FortiSOAR Docker image, see the "Deployment Guide."

To solve common issues that occur during the upgrade process, 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):

- Ensure that there is at least twice the current workflow storage capacity available before upgrading to release 7.6.1. 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 helps reduce the size of the active storage, improving performance, and making playbooks more efficient.
  NOTE: The space requirement is only necessary when upgrading to the 7.6.1 release and is not needed for subsequent upgrades
  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.
- Take a backup of your FortiSOAR Built-in connectors' (SSH, IMAP, Database, Utilities, etc.) configuration, since the configuration of your FortiSOAR Built-in connectors might be reset, if there are changes to the configuration parameters across versions.
- 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."
- Ensure that there are no duplicate entries in the `/etc/fstab` file.
- Unmount any external mount entries listed in the `/etc/fstab` file before upgrading FortiSOAR. After the upgrade is complete, you can re-mount these entries.
- 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 7.6.1.

Upgrading a FortiSOAR Enterprise instance using the Upgrade Framework

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.

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. In 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 7.6.1, 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 7.6.1 release, use the following command:
   `csadm upgrade check-readiness --target-version 7.6.1`
   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.
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 7.6.1 use the `csadm upgrade execute --target-version 7.6.1` 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>
```
You can also set the proxy while running the FortiSOAR Configuration Wizard or by using the `csadm network` command.
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 7.6.1, 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.

Arguments available for the '`csadm upgrade`' subcommand

'`upgrade`' is a new subcommand 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 7.6.1 --phase pre-upgrade` executes the pre-upgrade phase of upgrading your FortiSOAR instance to release 7.6.1. This assists in anticipating problems and taking proactive measures to fix them before moving forward with the full upgrade.
  IMPORTANT: The 'post-upgrade' phase must 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 7.6.1, then you can run post-upgrade for version 7.6.1 using the command `csadm upgrade execute-phase --target-version 7.6.1 --phase post-upgrade`. However, the command `csadm upgrade execute-phase --target-version 9.9.9 --phase post-upgrade` will fail with a message such as "`The post-upgrade phase can only be executed for the 9.9.9 version after upgrading FortiSOAR to the 9.9.9 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 7.6.1 --download-packages` command to download upgrade packages required to upgrade your system to 7.6.1 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 7.6.1 after downloading the upgrade packages locally, run the `csadm upgrade execute --target-version 7.6.1` 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 7.6.1` will generate the `upgrade-fortisoar-7.6.1-2024-05-22-1708597271.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.

Upgrading FortiSOAR High Availability Cluster for releases post 7.6.1

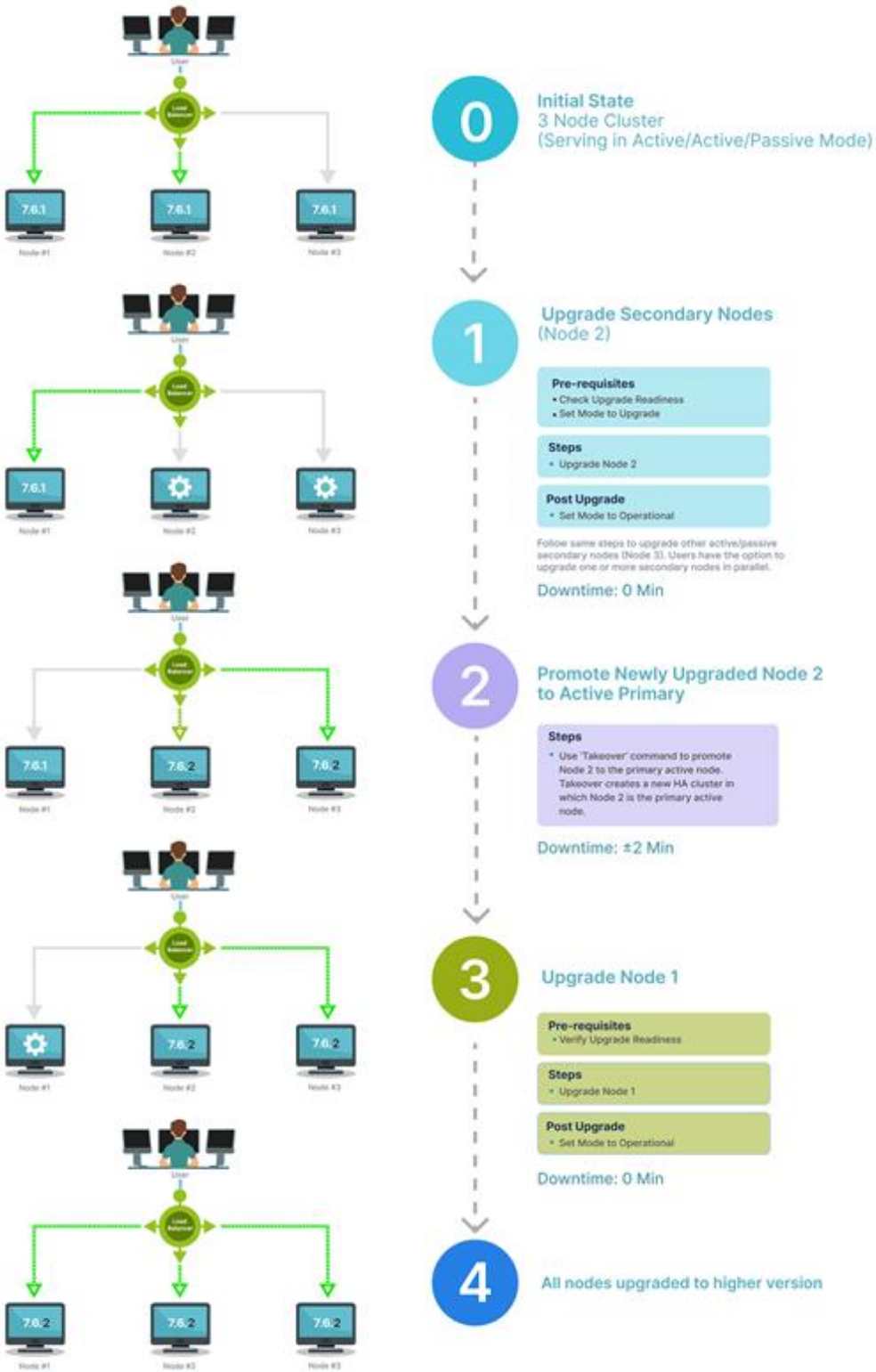
This section outlines the procedure for upgrading a FortiSOAR HA cluster (Active-Active or Active-Passive) for releases after 7.6.1, such as from 7.6.1 to 7.6.2. The upgrade steps are the same for both configurations, i.e., Active-Active or Active-Passive HA clusters.

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".

IMPORTANT: It is recommended to upgrade all secondary nodes before upgrading the primary node. Additionally, it is mandatory to upgrade at least one secondary node before upgrading the primary node.

The following diagram illustrates a high-level view of the rolling upgrades process:



Procedure

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 9.9.9`
   For details on the `check-readiness` option, see the Upgrade Framework chapter.
   b. Set the node's mode to 'Upgrade' using the following command to prepare for the upgrade:
   `csadm system env --mode upgrade`
   Setting the environment mode to '`upgrade`' on a node will cause its health check to fail. As a result, the load balancer will redirect traffic to the healthy nodes. For more information, see the '*Health Checks in FortiSOAR HA nodes*' topic in the High Availability and Disaster Recovery support in FortiSOAR of the "Administration Guide."
   In this example, the health check will fail on *Node 2*, causing the load balancer to redirect all requests to *Node 1*. Additionally, the `celeryd` service on Node 2 will be stopped and masked to prevent new playbooks from executing. Wait for this command to complete, which may take up to 30 minutes.
   c. Upgrade *Node 2* using the following command:
   `csadm upgrade execute --target-version [TARGET_VERSION]`
   For example to upgrade to release 9.9.9, use the `csadm upgrade execute --target-version 9.9.9` command.
   In the case of our example, while *Node 2* is being upgraded, *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
   d. During the upgrade process, a toaster message such as '`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.`' is displayed on the UI of the other nodes in the cluster.
   NOTE: If the upgrade of a node in the cluster is complete but the UI remains unresponsive for operations such as publishing modules, installing connectors, etc., log out and log back in to ensure the changes are applied correctly.
2. Once *Node 2* is upgraded, verify that the FortiSOAR UI is accessible and perform basic sanity checks, using *Node 2's* hostname or IP address (and not load balancer). *Node 2* will be in 'Upgrade' mode during this stage.
3. Once the FortiSOAR UI is accessible and you have completed the sanity checks on the system, 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.
   Use the same procedure to upgrade additional active/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.
4. 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.
   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. 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 serve requests as the new primary node, *Node 2*, and all nodes will again be joined to form the HA cluster, and all active nodes will resume sharing the load.

5. 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. Set *Node 1* to 'Operational' mode:
   `csadm system env --mode operational`

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.

Upgrading an Active-Active HA Cluster

For the purpose of 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".

Approximately 30 minutes of downtime is required for the upgrade.

To upgrade your active-active HA cluster to FortiSOAR 7.6.1, 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 makes *Node2* a standalone system.
3. Upgrade *Node2* using the process mentioned in the Upgrading a FortiSOAR Enterprise Instance using the Upgrade Framework chapter.
   Once the upgrade of *Node2* is complete, proceed to upgrade *Node1*.
   Important: The upgrade of *Node1* will incur downtime.
4. After upgrading both nodes, run the `resume-cluster` command from *Node2*.
5. Adjust the Reverse Proxy settings to handle requests from both *Node1* and *Node2*.

Upgrading an Active-Passive HA Cluster

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

Approximately 30 minutes of downtime is required for the upgrade.

To upgrade your active-passive HA cluster to FortiSOAR 7.6.1, perform the following steps:

1. Reverse Proxy is configured to designate *Node2* as the backup system. Therefore, comment out that part from the Reverse Proxy configuration.
2. Use the `#csadm ha` command as a '`root`' user to run the `suspend-cluster` command on *Node2*.
   This makes *Node2* a standalone system.
3. Upgrade *Node2* using the process mentioned in the Upgrading a FortiSOAR Enterprise Instance using the Upgrade Framework chapter.
   After *Node2* is upgrade successfully, proceed to upgrade *Node1*.
   Important: The upgrade of *Node1* will incur downtime.
4. Once both nodes are upgraded, run the `resume-cluster` command from *Node2*.
5. Reconfigure the Reverse Proxy to set *Node2* as the backup server.

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. A version mismatch will not work if either of them upgrades to 7.6.1.

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 using the Upgrade Framework 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**.
  Once you have completed the upgrade process, i.e., upgrading both your master and tenant nodes from a release prior to 7.4.2 to release 7.4.2 or later, you must restart the `cyops-integrations-agent` service on tenant nodes using the following command:
  `systemctl restart cyops-integrations-agent`
  You must restart the `cyops-integrations-agent` service at tenant nodes before you download the agent's or tenant's logs, from the master node's console..
- 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 using the Upgrade Framework 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 as a '`root`' user.
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) 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 Upgrade Framework chapter.
5. Upgrade the secure message exchange:
   `csadm upgrade execute --target-version [TARGET_VERSION]`
6. Once you have successfully upgraded the secure message exchange, start the data replication between the master and the tenant nodes again by toggling the **Data Replication** button to **ON**, and then verify the replication.

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`
   NOTE: These commands will reset the RabbitMQ node. You will need to reconfigure the server using the `csadm mq db flush` command and add the '`admin`' user when prompted.
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.
   NOTE: Downtime begins when the upgrade process starts on *Node2*.
4. Remove the *Node1* entry from the Reverse Proxy.
5. Log into *Node1* terminal session as the `root` user, and upgrade *Node2* following the steps in the Upgrading a FortiSOAR Secure Message Exchange section.
   NOTE: Downtime ends when the upgrade process is completed on *Node1*.
6. Add the *Node1* entry back to the Reverse Proxy.
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
```

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 7.6.1 using the process mentioned in the Upgrading a FortiSOAR Enterprise Instance using the Upgrade Framework 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 and upgrading your FortiSOAR Docker on an Amazon Elastic Kubernetes Cluster

Do not use the 'Upgrade Framework' to upgrade the Docker images from release 7.5.0 to 7.6.0. Follow the steps outlined in this chapter to upgrade the Docker instances.

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/7.6.1/install-fortisoar-docker-<release_version>.bin`
   For example, `https://repo.fortisoar.fortinet.com/7.6.1/install-fortisoar-docker-7.6.1.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_7.6.1-*_*_*.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_7.6.1-*_*_*.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-7.6.1.bin --env-file fortisoar.env`

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 take 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-7.6.1.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_7.6.1-*_*_*.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_7.6.1-*_*_*.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:7.6.1`
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`

Post-Upgrade Tasks and Notes

Playbook execution logs movement to historical storage

After upgrading to FortiSOAR 7.6.1, existing playbook execution logs will be moved to historical storage to optimize workflow logs storage. This background process may take some time, depending on the size of the logs, but it will not impact system functionality.

www.fortinet.com