Upgrade Health Checker docs

Upgrade Health Checker

NetApp
November 15, 2026

This PDF was generated from https://docs.netapp.com/us-en/upgrade-health-checker/index.html on November 15, 2026. Always check docs.netapp.com for the latest.

Table of Contents

Upgrade Health Checker docs . . . 1
Release notes . . . 2
What's new in Upgrade Health Checker . . . 2
28 October 2026 . . . 2
15 October 2026 . . . 2
13 August 2026 . . . 2
17 July 2026 . . . 2
11 June 2026 . . . 3
15 April 2026 . . . 3
13 February 2026 . . . 3
Get started . . . 5
Learn about Upgrade Health Checker . . . 5
Download and set up Upgrade Health Checker . . . 5
Input parameters for Upgrade Health Checker . . . 8
Input methods priority . . . 8
Input parameters . . . 9
Examples using each input method . . . 13
Additional commands . . . 14
Use Upgrade Health Checker . . . 16
Generate an upgrade plan with Upgrade Health Checker . . . 16
Upgrade Health Checker dashboard . . . 18
Automate generating upgrade plans with Upgrade Health Checker . . . 20
Receive Upgrade Health Checker reports by email . . . 22
Configure using the config.yaml file . . . 22
Configure using command line arguments . . . 23
Maintain, troubleshoot, FAQ . . . 25
Maintain Upgrade Health Checker . . . 25
/tmp space management . . . 25
Run directory cleanup . . . 25
Automatic updates . . . 25
Troubleshoot Upgrade Health Checker . . . 26
Troubleshoot issues . . . 26
FAQ for Upgrade Health Checker . . . 31
How does Upgrade Health Checker differ from Upgrade Advisor? . . . 31
What information does an Upgrade Health Checker report include? . . . 31
Can Upgrade Health Checker be used without an internet connection? . . . 31
What versions of ONTAP does Upgrade Health Checker support? . . . 32
Legal notices for Upgrade Health Checker . . . 33
Copyright . . . 33
Trademarks . . . 33
Patents . . . 33
Privacy policy . . . 33
Open source . . . 33

Upgrade Health Checker docs

Release notes

What's new in Upgrade Health Checker

Learn what's new in Upgrade Health Checker (UHC). You can find additional enhancement information by navigating to the Upgrade Health Checker tile in the NetApp Console Automation Hub and viewing the latest Upgrade Health Checker download files.

28 October 2026

This update adds the ability to automatically select the latest ONTAP patch release version and lists supported ONTAP HotFix upgrade versions in the report.

- Provides the ability to auto-select the latest ONTAP patch release version using `--target-ontap-version=latest_patch` or by using the `config.yaml` file.
- Supported ONTAP HotFix upgrade versions are now listed in the report.

15 October 2026

This update adds several enhancements to target-version risk assessment, report details, recommended ONTAP version selection, and firmware recommendations.

- The UHC assessment now evaluates ONTAP risks against the target version.
- Reports now include CVE IDs, CVSS scores, and recommended fix versions.
- UHC now supports automatically selecting the recommended ONTAP version using the `--target-ontap-version=recommended` parameter or in the `config.yaml` file.
- Reports now include recommended firmware versions.

13 August 2026

This update enhances interoperability checks for SAN and NAS ONTAP clusters and fixes some existing issues.

17 July 2026

This update adds several enhancements including a fleet risk view dashboard, email authentication and encryption support, drive count reporting, recommended ONTAP version support, and NAS and SAN IMT checks.

- Provides support for an email server with authentication and encryption.

  Receive Upgrade Health Checker reports by email.

- Includes a Fleet Risk view dashboard that risk-ranks your entire estate. You can now see which blocked clusters carry the most risk. This allows you to prioritize remediation.
- The cluster report now includes Drive Count, alongside ONTAP version, node count, and capacity. This completes the hardware snapshot for upgrade and capacity planning.
- The UHC assessment now includes the recommended upgrade target ONTAP version.

- Adds support for NAS and SAN IMT checks.

11 June 2026

This update adds email delivery for reports, a dashboard parameter for quick health overviews, and new documentation sections to help you maintain and troubleshoot Upgrade Health Checker.

- UHC supports the direct delivery of reports to your inbox.

  Receive Upgrade Health Checker reports by email.

- Introduces the `--dashboard` parameter that regenerates the dashboard from existing runs at `<runs>/dashboard/index.html` without contacting a cluster. It provides a high-level overview of the health status of all the clusters that have been recently accessed.

  Upgrade Health Checker dashboard, Dashboard parameter.

- Introduces a new Maintain, Troubleshoot, and FAQ section.
  - Maintain Upgrade Health Checker by checking for updates, monitoring logs for recurring issues, and verifying that you configure the tool correctly.

    Maintain Upgrade Health Checker.

  - Troubleshoot common Upgrade Health Checker issues and resolve errors that can block ONTAP upgrade planning.

    Troubleshoot Upgrade Health Checker.

  - Frequently asked questions (FAQ) for Upgrade Health Checker.

    FAQ for Upgrade Health Checker.

15 April 2026

This update adds a new XLSX report format and requires access to a new endpoint for auto-updates to work.

- You must grant UHC access to a new endpoint (https://api.activeiq.netapp.com) for auto-update features to work.

  Download and set up Upgrade Health Checker.

- UHC now provides a new report format (.XLSX - human-editable) that complements the HTML (human-readable) and JSON (machine-readable) formats.

  Generate an ONTAP upgrade report.

  Automate generating upgrade plans with Upgrade Health Checker.

13 February 2026

Upgrade Health Checker is now available. This tool generates detailed, onsite reports to help you complete a successful ONTAP upgrade. You can use Upgrade Health Checker with existing automation and orchestration frameworks to save time when upgrading single or multiple clusters. Upgrade Health Checker supports upgrades to ONTAP 9.11.1 and later.

Learn more about Upgrade Health Checker.

Get started

Learn about Upgrade Health Checker

Upgrade Health Checker is an upgrade advisory service tool that provides detailed information and recommendations to help you complete a successful ONTAP upgrade. It conducts onsite, comprehensive health checks that identify blockers and warnings that you must resolve before performing an upgrade.

Upgrade Health Checker can integrate into your existing automation and orchestration frameworks to aid in upgrading a single or multiple clusters. It uses your most recent AutoSupport data to generate a detailed, accurate report on risks associated with your clusters, information about new features and enhancements in the ONTAP version you're upgrading to, and any actions to take before attempting an ONTAP upgrade.

To learn more about Upgrade Health Checker, you can refer to the FAQ for Upgrade Health Checker or watch the following introduction video.

[] | *https://img.youtube.com/vi/5mwmigXAB9c/maxresdefault.jpg*

Download and set up Upgrade Health Checker

You can download Upgrade Health Checker to generate onsite, comprehensive reports before upgrading to a newer version of ONTAP.

About this task

Upgrade Health Checker supports upgrades for on-premises ONTAP versions 9.11.1 and later. If you are using Cloud Volumes ONTAP, refer to Upgrade Cloud Volumes ONTAP for information on upgrading your system.

Before you begin

Set up a virtual machine for Upgrade Health Checker with the following specifications:

Component	Specification
VM	m5.large VM or equivalent with 2 vCPUs and 8 GiB RAM
Operating System	Any Linux OS with glibc library version 2.28 or higher (Ubuntu 22.04, RHEL8, or RHEL9)
Disk space	At least 4 GB free (8 GB recommended) in `/tmp` with execute permissions
Network	HTTPS connectivity to the ONTAP cluster(s) management LIF
ONTAP version	Cluster running ONTAP 9.11.1 or later

Component	Specification
Hosting requirements	Place the VM in a location with cluster connectivity and internet access to allow for automatic tool updates. The machine should have access to the following endpoints through HTTPS (443) to conduct automatic updates and to enable NetApp to receive AutoSupport information about your upgrade plans: https://support.netapp.com/ https://api.uhc.netapp.com https://gql.aiq.netapp.com https://api.activeiq.netapp.com
Recommended packages	Install a web server to facilitate access

Ensure that the virtual machine can connect to the auto-update and telemetry endpoints (https://support.netapp.com/ and https://api.activeiq.netapp.com) through HTTPS (443) to enable NetApp to receive AutoSupport information about your upgrade plans. If internet access is unavailable, you must manually download the latest version of Upgrade Health Checker.

Steps

1. Download the Upgrade Health Checker binary by navigating to the NetApp Console Automation Hub and finding the Upgrade Health Checker tile.
2. Set up the Upgrade Health Checker virtual machine and use SSH to place the binary in the location of your choice.
3. Verify the digital signature of Upgrade Health Checker.

   Upgrade Health Checker has a public code signing certificate (`UHC-Linux-codesigning-certificate-public.pem`) and intermediate and root certificates chain (`UHC-Linux-chain-certificates-public.pem`).

   a. (Optional) Verify the code signing certificate against the chain:

   ```
   openssl verify -CAfile UHC-Linux-chain-certificates-public.pem UHC-Linux-codesigning-certificate-public.pem
   ```

   An output of `OK` confirms a valid chain of trust.

   b. Extract the public key from the code signing certificate:

   ```
   openssl x509 -in UHC-Linux-codesigning-certificate-public.pem -pubkey -noout -out UHC-Linux-public.pub
   ```

   c. Verify the signature file (`uhc.sig`) against the Upgrade Health Checker binary using the public key:

   ```
   openssl dgst -sha256 -verify UHC-Linux-public.pub -signature uhc.sig
   ```

```
uhc
```

An output of `Verified OK` confirms the signature is valid.

4. Configure a service account and role for ONTAP cluster access. For cluster access through Upgrade Health Checker, create the service role as a REST-role.

> You can build an Ansible playbook to automate the deployment of the role and user to all ONTAP clusters.

You must create the service account for the HTTP application. Use the following CLI commands to configure the necessary permissions:

```
security login rest-role create -role uhctool -api /api -access readonly
-vserver

security login rest-role create -role uhctool -api
/api/support/autosupport -access read_create_modify -vserver

security login rest-role create -role uhctool -api
/api/support/autosupport/messages -access read_create_modify -vserver

vserver services web access create -name spi -role uhctool -vserver

security login create -user-or-group-name uhctool -role uhctool
-application http -authentication-method password

security login create -user-or-group-name uhctool -role uhctool
-application ssh -authentication-method password
```

5. (Optional) Set up credential management to secure access to the application.

For example, if you have CyberArk and Conjur, you can configure your environment to avoid passing credentials through a yaml file or command line.

a. Create required CyberArk Safes:
   i. Create a Safe (Main-Conjur-Safe) that holds the application credentials and secrets
   ii. Create a Safe (API-Credentials-Safe) that holds the Conjur Host ID and API Key
   iii. Create a Safe (Conjur-SSL-Certificate) that holds the necessary certificate
b. Create the configuration (Conjur.conf) and identity (Conjur.identity) files for this application:
   i. Conjur.conf

```
account:
plugins:[]
appliance_url: https://FQDN
```

```
cert_file: /etc/conjur.pem
```

ii. Conjur.identity

```
machine https://FQDN/authn
login host /prodvault/devops/<Main-Conjur-Safe>/host1
password XXXXXX
```

Here is an example of how to use CyberArk and Conjur in an Ansible playbook:

a. Install Conjur Ansible Collection, which includes the Conjur Ansible Lookup Plugin, on Ansible Host:

```
ansible-galaxy collection install cyberark.conjur
```

b. Create a task in a yaml file to fetch the username and password from CyberArk:

```
conjur_username: "{{ lookup('cyberark.conjur.conjur_variable',
'prodvault/devops/<Main-Conjur-Safe>/<Account name>/username',
validate_certs=false) }}"
conjur_password: "{{ lookup('cyberark.conjur.conjur_variable',
'prodvault/devops/<Main-Conjur-Safe>/<Account name>/password',
validate_certs=false) }}"
```

Related information

- Learn about the input parameters for Upgrade Health Checker

What's next?

You can use Upgrade Health Checker to help you plan for an ONTAP upgrade by generating an upgrade report.

Input parameters for Upgrade Health Checker

You can learn more about the input parameters for Upgrade Health Checker and how to provide them through CLI arguments, configuration files, or interactive prompts to assist you with generating an upgrade report for your ONTAP cluster.

Input methods priority

Upgrade Health Checker provides several input options for all parameters. Its priority order for accepting inputs is:

1. CLI arguments (highest priority)
2. Configuration file (`config.yaml`)
3. Interactive prompts (lowest priority)

When a parameter is provided through multiple methods, the tool will use the value from the highest priority source.

Input parameters

Cluster IP

The `--cluster-ip` parameter specifies the IP address or hostname of the ONTAP cluster to connect to.

The priority order for providing this parameter is:

1. CLI argument: `--cluster-ip`
2. Config file: `CLUSTER.IP`
3. Interactive prompt

▼ Examples

- CLI argument:

  ```
  ./uhc --cluster-ip 192.168.1.100
  ```

- Config.yaml:

  ```
  CLUSTER:
    IP: "192.168.1.100"
  ```

- Interactive mode (if not provided above):

  Tool will prompt: `Enter cluster IP address:`

Cluster username

The `--cluster-username` parameter specifies the username for authenticating with the ONTAP cluster.

The priority order for providing this parameter is:

1. CLI argument: `--cluster-username`
2. Config file: `CLUSTER.USERNAME`
3. Interactive prompt

▼ Examples

- CLI argument:

  ```
  ./uhc --cluster-username admin
  ```

- Config.yaml:

  ```
  CLUSTER:
  ```

```
  USERNAME: "admin"
```

- Interactive mode (if not provided above):

  Tool will prompt: `Enter cluster username:`

Cluster password

The `--cluster-password` parameter specifies the password for authenticating with the ONTAP cluster.

The priority order for providing this parameter is:

1. CLI argument: `--cluster-password`
2. Config file: `CLUSTER.PASSWORD`
3. Interactive prompt (secure input)

▼ **Examples**

- CLI argument (not recommended for security):

  ```
  ./uhc --cluster-password mypassword
  ```

- Config.yaml (ensure file permissions are restricted):

  ```
  CLUSTER:
    PASSWORD: "mypassword"
  ```

- Interactive mode (recommended - password hidden):

  Tool will prompt: `Enter cluster password:`

Target ONTAP version

The `--target-ontap-version` parameter specifies the ONTAP version you want to upgrade to for analysis. Use "current" to keep the existing cluster ONTAP version.

The priority order for providing this parameter is:

1. CLI argument: `--target-ontap-version`
2. Config file: `CLUSTER.TARGET_ONTAP_VERSION`
3. Interactive selection menu

▼ **Examples**

- CLI argument:

  Updated version of ONTAP: `./uhc --target-ontap-version 9.15.1`

  Keep current version of ONTAP: `./uhc --target-ontap-version current`

- Config.yaml:

```
CLUSTER:
  TARGET_ONTAP_VERSION: "9.15.1"
```

- Interactive mode (if not provided above):

  Tool will display available versions and prompt for selection

Accept EULA

The `--accept-eula` parameter specifies whether to accept the End User License Agreement. Must be set to `true` to proceed.

The priority order for providing this parameter is:

1. CLI argument: `--accept-eula`
2. Config file: `CLUSTER.ACCEPT_EULA`
3. Interactive prompt

▼ Examples

- CLI argument:

  ```
  ./uhc --accept-eula true
  ```

- Config.yaml:

```
CLUSTER:
  ACCEPT_EULA: true
```

- Interactive mode (if not provided above):

  Tool will display EULA and prompt for acceptance

Configuration file path

The `--config-path` parameter specifies the path to a custom configuration YAML file.

The priority order for providing this parameter is:

1. CLI argument: `--config-path`
2. Default: `config.yaml`

▼ Examples

- CLI argument:

  ```
  ./uhc --config-path /path/to/custom_config.yaml
  ```

- Default (if not provided):

  Tool will look for config.yaml in the current directory

Run output path

The `--runs-path` parameter specifies a custom directory for storing run outputs and reports.

The priority order for providing this parameter is:

1. CLI argument: `--runs-path`
2. Config file: `APP.RUNS_PATH`
3. Default: `./runs`

▼ **Examples**

- CLI argument:

  ```
  ./uhc --runs-path /custom/output/path
  ```

- Config.yaml:

  ```
  APP:
    RUNS_PATH: "/custom/output/path"
  ```

- Default (if not provided):

  Tool will use `./runs` directory

Skip compatibility check

The `--skip-compatibility-check` parameter bypasses hardware compatibility checks and uses the specified version of ONTAP you want to upgrade to.

Only use this option if you are certain the target ONTAP version is compatible with your hardware.

The priority order for providing this parameter is:

1. CLI argument: `--skip-compatibility-check`
2. Config file: `CLUSTER.SKIP_COMPATIBILITY_CHECK`
3. Default: `false`

▼ **Examples**

- CLI argument:

  ```
  ./uhc --skip-compatibility-check true
  ```

- Config.yaml:

```
CLUSTER:
  SKIP_COMPATIBILITY_CHECK: true
```

Dashboard

The `--dashboard` parameter regenerates the dashboard from existing runs at `<runs>/dashboard/index.html` without contacting a cluster. You use it to refresh the fleet view outside of a normal run. The dashboard provides a high-level overview of the health status of all the clusters that have been assessed recently, allowing you to easily monitor and compare the upgrade readiness of multiple clusters in one place. You can open the dashboard in a web browser to view the health status, potential blockers, and risks for each cluster at a glance.

- CLI argument: `--dashboard`

▼ Examples

- CLI argument:

```
./uhc --dashboard
./uhc --dashboard --runs-path /custom/output
./uhc --dashboard --config-path /path/to/custom_config.yaml
```

Disable telemetry

To disable telemetry, add the following to your `config.yaml` file:

```
TELEMETRY:
  ENABLED: false
```

Disable automatic updates

To disable automatic updates for Upgrade Health Checker, add the following to your `config.yaml` file:

```
AUTO_UPDATE:
  ENABLED: false
```

Examples using each input method

Combined CLI arguments

```
./uhc \
  --cluster-ip 192.168.1.100 \
  --cluster-username admin \
  --cluster-password mypassword \
```

```
  --target-ontap-version 9.15.1 \
  --accept-eula true \
  --config-path /path/to/custom_config.yaml \
  --runs-path /custom/output \
  --skip-compatibility-check false
```

Config.yaml file

```
CLUSTER:
  IP: "192.168.1.100"
  USERNAME: "admin"
  PASSWORD: "mypassword"
  TARGET_ONTAP_VERSION: "9.15.1"
  ACCEPT_EULA: true
  SKIP_COMPATIBILITY_CHECK: false

APP:
  RUNS_PATH: "/custom/output"
```

Interactive mode

To prompt all required input parameters interactively, run Upgrade Health Checker without arguments:

```
./uhc
```

Additional commands

These commands provide some additional capabilities beyond the full run of the tool:

- Display help

  ```
  ./uhc --help
  ```

- Display version

  ```
  ./uhc --version
  ```

- Test cluster connectivity

  ```
  ./uhc --test-connectivity cluster
  ```

- Test telemetry connectivity

  ```
  ./uhc --test-connectivity telemetry
  ```

- Test auto-update connectivity

```
./uhc --test-connectivity autoupdate
```

- Run all tests

```
./uhc --test all
```

Use Upgrade Health Checker

Generate an upgrade plan with Upgrade Health Checker

You can use Upgrade Health Checker to generate an upgrade plan for a single or multiple ONTAP clusters.

You can also automate generating upgrade reports to assist with maintaining up-to-date upgrade plans if you have a large, complex environment with multiple clusters.

About this task

Upgrade Health Checker supports upgrades for on-premises ONTAP versions 9.11.1 and later. If you are using Cloud Volumes ONTAP, refer to Upgrade Cloud Volumes ONTAP for information on upgrading your system.

Before you begin

To learn more about a specific Upgrade Health Checker parameter and the priority order for how the tool accepts input, refer to Learn about the input parameters for Upgrade Health Checker.

Steps

1. Set the execute permission on the binary before running it the first time:

```
chmod +x uhc
```

> Upgrade Health Checker is a self-contained package and will need to unpack itself before executing. This can take a few seconds.

2. Before completing a full run of Upgrade Health Checker, perform a comprehensive check to ensure the tool can connect to the cluster and required endpoints:

```
--test all
```

   The comprehensive check ensures the virtual machine hosting Upgrade Health Checker has:

   - Connectivity to the ONTAP cluster IP address through HTTPS
   - Connectivity to the telemetry endpoint (https://support.netapp.com/ and https://api.activeiq.netapp.com) through HTTPS
   - Connectivity to the auto-update endpoints (https://api.uhc.netapp.com, https://gql.aiq.netapp.com, and https://api.activeiq.netapp.com) through HTTPS (443)
   - At least 4 GB of available `/tmp` space

3. (Optional) If you want to use a configuration file to store parameters, rename the `config.yaml.example` file located in the same location you downloaded the binary to `config.yaml`.

   Here is an example config.yaml file:

```
# Application Configuration
```

```
APP:
  RUNS_PATH: "/opt/uhc/runs"

# Cluster Credentials
CLUSTER:
  IP: "x.x.x.x"
  USERNAME: "admin"
  PASSWORD: "xyz"
  TARGET_ONTAP_VERSION: ""  # Optional: Specify target ONTAP version
(e.g., "9.16.1" or "current" to keep existing version). Leave empty to
prompt user.
  ACCEPT_EULA: false  # Optional: Set to true to accept EULA through
config. If false/empty, user will be prompted interactively.
```

4. Run Upgrade Health Checker by entering the ONTAP version you want to upgrade to and accepting the EULA, including additional parameters as needed.

   The EULA is a multi-page document. Enter `all` to see the entire document at once and enter `y` to accept it.

   After the AutoSupport log is downloaded for a cluster, processing generally takes one to two minutes depending on the number of cluster nodes. Because different ONTAP software versions handle AutoSupport collection differently, each Upgrade Health Checker run depends on the current load of the cluster and the current ONTAP version.

   a. Familiarize yourself with the available Upgrade Health Checker parameters, input priority order, and default values for certain parameters by referring to Learn about the input parameters for Upgrade Health Checker.

      The Input Parameters Guide also provides information for specifying a custom configuration file path and run path to streamline your workflow. The best practice is to create a custom output directory for each cluster to organize your reports and logs effectively.

   b. Run Upgrade Health Checker by accepting the EULA and entering the target ONTAP version and any additional parameters.

      ```
      ./uhc
      ```

5. After Upgrade Health Checker completes its checks, navigate to the `runs` folder to view the upgrade plan and cluster report.

   Each execution of the program is a "unique run" and will have all the associated data, logs and results in the associated `runs` folder.

6. List the generated report files:

   ```
   ls -lrt
   ```

7. Open the report file (named `uhc_<cluster-name>_<YYYYMMDDHHMMSS>`) in a web browser, Microsoft Excel, or JSON viewer to view the report. If you are running it on a remote host, first download the report file to a machine where you can view the report on a web browser, in Excel, or in a JSON viewer.

   The log path and report path are as follows:

   - Log Path: `<output-path>/<unique-run-dir>/<cluster-name>/logs`
   - HTML Report Path: `<output-path>/<unique-run-dir>/<cluster-name>/results/uhc_<cluster-name>_<YYYYMMDDHHMMSS>.html`
   - Excel Report Path: `<output-path>/<unique-run-dir>/<cluster-name>/results/uhc_<cluster-name>_<YYYYMMDDHHMMSS>.xlsx`
   - JSON Report Path: `<output-path>/<unique-run-dir>/<cluster-name>/results/uhc_<cluster-name>_<YYYYMMDDHHMMSS>.json`

What's next

After you've generated your upgrade plan and cluster report, you can view the Upgrade Health Checker dashboard for details of the health status of clusters assessed during the run.

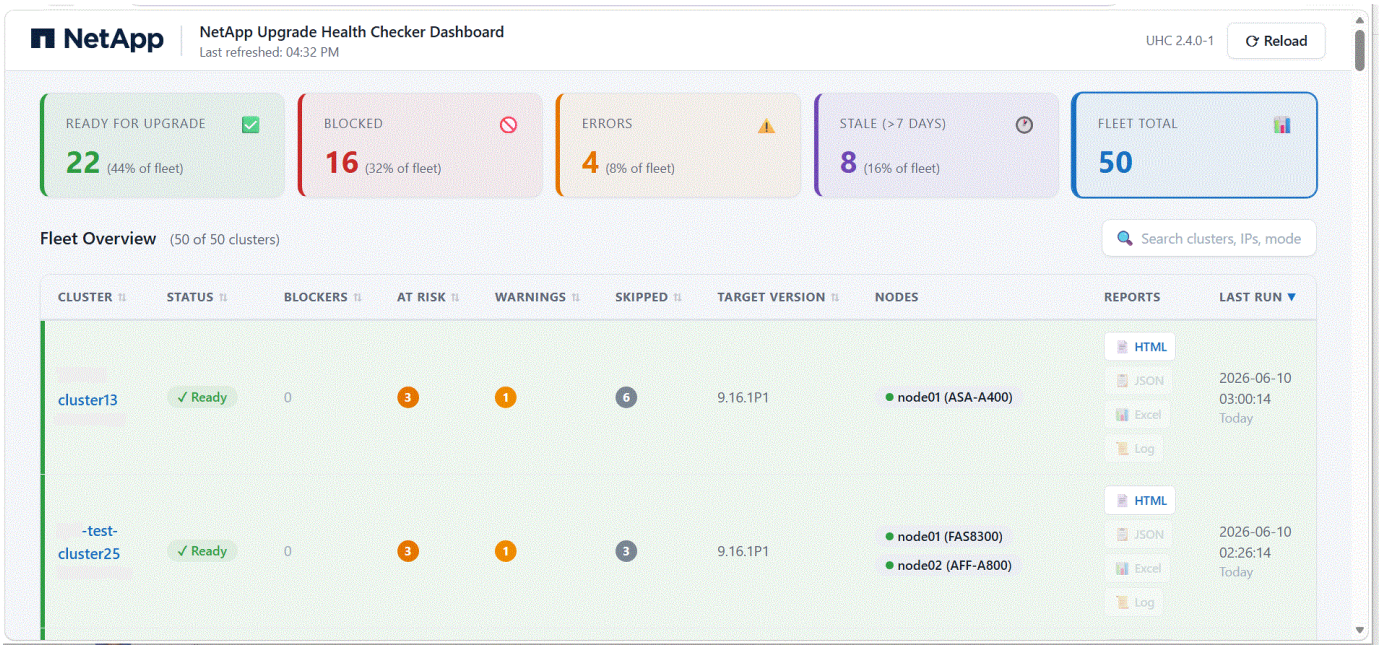
Upgrade Health Checker dashboard

A static HTML dashboard is generated at `runs/dashboard/index.html` at the end of every Upgrade Health Checker run. The dashboard provides a high-level overview of the health status of clusters assessed recently, allowing you to monitor and compare upgrade readiness across clusters.

To regenerate the dashboard later (without contacting a cluster), run:

```
./uhc --dashboard
./uhc --dashboard --runs-path /custom/output
./uhc --dashboard --config-path /path/to/custom_config.yaml
```

The following is a view of a recently generated Upgrade Health Checker dashboard.



At the top of the page there are five color-coded summary cards (each one is also a selectable filter) showing:

- **Ready for upgrade:** Number of clusters whose latest health check found no blockers, i.e. ready to upgrade.
- **Blocked:** Number of clusters blocked and unable to upgrade until the blockers are resolved.
- **Errors:** Number of clusters whose latest UHC run failed to complete (e.g. connection, authentication, or data-collection failure).
- **Stale (>7):** Number of clusters whose last UHC run was more than 7 days ago (the health-check data is stale).
- **Fleet total:** Total number of clusters in the fleet, (those with a run in the dashboard window, default is the last 30 days).

The Fleet overview section details the current state of the fleet. For each cluster, the following are provided (the table shows the latest run per cluster):

- **Cluster name:** Name of the cluster (with its management IP shown below the name).
- **Status:** Current status of the cluster: Ready, Blocked, Error, or Stale.
- **Blockers:** Number of blockers on the cluster. These are critical issues that must be resolved before upgrading.
- **At risk:** Number of at-risk findings on the cluster. These are risks identified during the pre-update check.
- **Warnings:** Number of warnings on the cluster. The pre-update check completed with some warnings; you can choose to fix these warnings or continue with the update.
- **Skipped:** Number of checks that were skipped on the cluster.
- **Target version:** The ONTAP version the user intends to upgrade to. (It can be the same as the current version if you only want to run a health check on the running release.)
- **Nodes:** The cluster's nodes, each shown as node name with its hardware model (e.g. eurbkpcla (FAS8200)).
- **Reports:** Available report formats for the run: HTML, JSON, Excel, Log.

- **Last run:** Date and time of the last Upgrade Health Checker run (with a relative "X days ago").

You can select **Reload** to refresh the page. Use the **Search** bar to search for details of a specific cluster.

Automate generating upgrade plans with Upgrade Health Checker

You can automate generating Upgrade Health Checker reports to reduce your manual effort when planning ONTAP upgrades in large, complex environments.

About this task

Upgrade Health Checker supports upgrades for on-premises ONTAP versions 9.11.1 and later. If you are using Cloud Volumes ONTAP, refer to Upgrade Cloud Volumes ONTAP for information on upgrading your system.

Steps

1. To ensure the successful creation of your upgrade reports, complete any necessary set up steps and one-time tasks outlined in Generate an ONTAP upgrade report.
2. Create a script to run Upgrade Health Checker with the appropriate parameters for your environment.

```
./uhc \
  --cluster-ip=<cluster-ip> \
  --cluster-username=<cluster-username> \
  --cluster-password=<cluster-password> \
  --target-ontap-version=<target-ontap-version> \
  --accept-eula=true
```

3. Create an automation script for cron schedules to run Upgrade Health Checker. The following example script that runs Upgrade Health Checker for multiple clusters. The binary and config.yaml files have been installed into /opt/uhc/tool/.

```
#!/bin/bash
# /opt/uhc/tool/uhccron.sh
# Automated weekly Upgrade Health Checker run for all managed clusters
# Schedule: Every Monday to Friday at 4am

cd /opt/uhc/tool/

# Cluster1 - keep current version (health check only)
/opt/uhc/tool/uhc --cluster-ip cluster1-mgmt.example.com \ --target
-ontap-version current

# Cluster2 - target upgrade to 9.16.1
/opt/uhc/tool/uhc --cluster-ip cluster2-mgmt.example.com \ --target
-ontap-version 9.16.1

# Cluster3 - target upgrade to 9.17.1
```

```
/opt/uhc/tool/uhc --cluster-ip cluster3-mgmt.example.com \ --target
-ontap-version 9.17.1

# Cluster4 - by IP address
/opt/uhc/tool/uhc --cluster-ip 10.20.30.40 --target-ontap-version 9.16.1

# Add additional clusters as needed

# Make the script executable (one-time setup step)
chmod +x /opt/uhc/tool/uhccron.sh
```

4. Set up the cron job to run the script Monday through Friday at 4:00 am:

```
# Edit the crontab for the uhctool user
crontab -e

# Add the following line to schedule the script to run Monday through
Friday at 4:00 am
0 4 * * 1-5 /opt/uhc/tool/uhccron.sh
```

For ongoing health monitoring, use `--target-ontap-version current` without a specific upgrade target.

5. After Upgrade Health Checker completes its checks, navigate to the `runs` folder to view the upgrade plan and cluster report.

Each execution of the program is a unique run and will have all the associated data, logs, and results in the associated `runs` folder.

6. Open the report file (named `uhc_<cluster-name>_<YYYYMMDDHHMMSS>`) in a web browser, Microsoft Excel, or JSON viewer to view the report. If you are running Upgrade Health Checker on a remote host, first download the report file to a machine where you can view the report on a web browser, in Excel, or in a JSON viewer.

   The log path and report path are as follows:

   - Log Path: `<output-path>/<unique-run-dir>/<cluster-name>/logs`
   - HTML Report Path: `<output-path>/<unique-run-dir>/<cluster-name>/results/uhc_<cluster-name>_<YYYYMMDDHHMMSS>.html`
   - Excel Report Path: `<output-path>/<unique-run-dir>/<cluster-name>/results/uhc_<cluster-name>_<YYYYMMDDHHMMSS>.xlsx`
   - JSON Report Path: `<output-path>/<unique-run-dir>/<cluster-name>/results/uhc_<cluster-name>_<YYYYMMDDHHMMSS>.json`

Receive Upgrade Health Checker reports by email

Upgrade Health Checker can email reports directly to your inbox. Enable this feature through the config.yaml file or a command line argument. When enabled, Upgrade Health Checker sends generated reports (HTML, Excel, or JSON) to one or more recipients via an SMTP server. When disabled, no reports are sent. If report generation fails and email is enabled, Upgrade Health Checker emails the log files to the configured recipients.

Configure using the config.yaml file

You can configure email settings in the config.yaml file by enabling email reports, adding SMTP server details, and specifying recipient addresses and report formats.

1. Enable email reports in the config.yaml file.

```
# Email Report Configuration
EMAIL:
  ENABLED: true
```

2. Add the SMTP configuration settings, sender details, and subject:

```
# SMTP Server Configuration
EMAIL:
  ENABLED: true
  FROM: "uhc-notify@example.com"
  SUBJECT: "UHC Upgrade Health Report"
  REPLY_TO: <email>
  SMTP:
    HOST: "smtp.example.com"
    PORT: [<port-number>]
```

3. Add recipient email addresses and set which report formats the system sends by email, for example, HTML, EXCEL, JSON, or ALL. You can also use authentication and encryption:

```
# Recipient Email Addresses, report formats
EMAIL:
  ENABLED: true
  TO: <recipient>
  FROM: "uhc-notify@example.com"
  SUBJECT: "UHC Upgrade Health Report"
  REPLY_TO: <email>
  RECIPIENTS: "user1@example.com, user2@example.com"
  REPORT_FORMATS: "HTML,EXCEL,JSON"   # valid values: HTML, EXCEL, JSON
or ALL
  SMTP:
```

```
    HOST: "smtp.example.com"
    PORT: [PORT]
    USE_SMTP_AUTH: true
    USE_TLS: true
    USE_SSL: true
    USERNAME: "user@example.com"
    PASSWORD: "********"
```

Configure using command line arguments

You can configure basic email settings using command line arguments to enable reports, specify recipients, and select report formats.

1. Enable email reports using command line arguments:

```
./uhc --email-enabled=true
```

2. Provide recipient email addresses:

```
./uhc --email-recipients="user1@example.com, user2@example.com"
```

3. Configure which report formats are sent by email, for example, HTML, EXCEL, JSON:

```
./uhc --email-report-formats=HTML,EXCEL,JSON
```

You can configure the following three attributes using command line arguments or the config.yaml file. You can configure all remaining attributes only in the config.yaml file:

- Email Enabled
- Email Recipients
- Email Report Formats

Note that command line arguments override config.yaml file values.

For example:

```
./uhc \
  --email-enabled=true
  --email-recipients="user1@example.com, user2@example.com" \
  --email-report-formats=HTML,EXCEL,JSON \
```

```
EMAIL:
```

```
FROM: "uhc-notify@example.com"
SUBJECT: "UHC Upgrade Health Report"
REPLY_TO: <email>
SMTP:
  HOST: "smtp.example.com"
  PORT: [PORT]
```

Maintain, troubleshoot, FAQ

Maintain Upgrade Health Checker

Maintain Upgrade Health Checker by regularly checking for updates, monitoring logs for any recurring issues, and ensuring that the tool is configured correctly for your environment. This ensures that the tool runs reliably and efficiently. If you encounter new issues or need further assistance, refer to the NetApp support resources or contact NetApp support.

/tmp space management

Upgrade Health Checker extracts its components to the `/tmp` directory during execution. If you encounter issues related to insufficient space in `/tmp`, consider the following:

- Check available space in `/tmp`, ensuring that there is a minimum of 4 GB of free space for the tool to operate effectively.
- Remove any unnecessary files from `/tmp` to free up space.

Run directory cleanup

Upgrade Health Checker creates a unique run directory for each execution, which contains logs and results. To manage disk space, configure an automated cleanup process to remove old run directories after a certain period, for example, 30 days.

```
# Delete UHC data older than 30 days (recommended for automated workflows)
find /mnt/nfs/uhc/runs -type d -mtime +30 -exec rm -rf {} + 2>/dev/null

# Schedule as a weekly cron job
0 3 * * 0 find /mnt/nfs/uhc/runs -type d -mtime +30 -exec rm -rf {} +
2>/dev/null
```

Automatic updates

Upgrade Health Checker checks for updates automatically after each run when internet connectivity is available. This ensures that you have the latest features, improvements, and bug fixes. Where there is no internet access, you can manually download the latest version from the NetApp Console Automation Hub.

To disable automatic updates, add the following to the `config.yaml` file:

```
AUTO_UPDATE:
  ENABLED: false
```

Troubleshoot Upgrade Health Checker

Learn how to troubleshoot common Upgrade Health Checker issues and resolve errors that can block ONTAP upgrade planning.

Troubleshoot issues

Issue: Authentication or credential errors

Symptoms:

```
Failed to fetch cluster details for <cluster-ip>
Error connecting to ONTAP cluster: 401 Unauthorized
Missing cluster credentials
```

These errors indicate that Upgrade Health Checker is unable to authenticate with the ONTAP cluster using the provided credentials. This can be caused by an incorrect username or password, insufficient permissions for the user account, or an account that is locked or disabled.

Solutions:

- Verify that the username and password are correct
- Ensure the user has sufficient ONTAP REST API permissions
- Check if the account is locked or disabled

```
./uhc --test-connectivity cluster
```

Issue: Auto-update fails

Symptoms:

```
Auto-update failed: ...
Failed to download update
```

These errors indicate that Upgrade Health Checker is unable to connect to the internet to check for or download updates. This can be caused by network connectivity issues, firewall rules blocking access, or proxy settings that are not configured correctly.

Solutions:

- Check internet connectivity:

```
./uhc --test-connectivity autoupdate
```

- Check disk space (requires 1 GB):

```
df -h <location of uhc>
```

Issue: Binary takes a long time to start

Cause: The binary is self-contained and needs to unpack itself before executing.

Expected Behavior: First execution might take a few seconds to load. This is normal.

Issue: "Permission denied" or "Cannot execute binary" on /tmp

Symptoms:

```
[Errno 13] Permission denied
OSError: [Errno 13] Permission denied: '/tmp/_MEI...'
Cannot execute binary file
```

This error might occur when the tool is unable to execute files in the `/tmp` directory, which is used for extracting and running the tool's components. This can be caused by restrictive permissions on the `/tmp` directory or security policies that prevent execution from this location.

Solutions:

1. Check if `/tmp` has `noexec` enabled:

```
mount | grep /tmp
# If you see noexec in the output, this is the issue.
```

2. Remount `/tmp` with `exec` (requires System administrator root permissions):

```
# Temporary fix
sudo mount -o remount,exec /tmp
# Permanent fix - edit /etc/fstab
```

```
# Change "noexec" to "exec" for /tmp mount point
```

3. If the previous step cannot be performed due to constraints, you can configure the tool to use an alternative temporary directory with appropriate permissions:

```
mkdir -p /custom-tmp-path
# This only needs to be done one time.
TMPDIR=/custom-tmp-path ./uhc
# The TMPDIR prefix has to be added every time.
```

The custom path must first exist for this workaround to work. If it doesn't already exist, it is not created and falls back to using `/tmp`.

4. After adjusting permissions or changing the temporary directory, try running the tool again to see if the issue is resolved.

Issue: Connection timeout

Symptoms:

```
Connection timeout
Request timeout
```

Solutions:

- Check network connectivity to the cluster
- Verify that no firewall is blocking HTTPS (443) traffic
- Check the cluster is responsive and not under a heavy load

Issue: Insufficient disk space

Symptoms:

```
Not enough disk space available
OSError: [Errno 28] No space left on device
```

Solutions:

- Check disk space:

```
df -h /tmp
```

```
df -h .
```

- Clean old runs:

```
# Remove old run directories
rm -rf runs/<old_run_directories>
```

- Clean temporary files:

```
# Remove temporary files
rm -rf /tmp/_MEI*
```

Issue: Invalid runs path

Symptoms:

```
Invalid basepath_runs: <error>
RUNS path is not set
Cannot create tarball: basepath_runs '<path>' does not exist
```

Solutions:

- Ensure that the runs output directory exists and is writable
- Specify a valid path via CLI: `--runs-path /valid/path`
- Configure in config.yaml: `APP.RUNS_PATH: "/valid/path"`

Issue: Invalid target ONTAP version

Symptoms:

```
Invalid ONTAP version: '<version>' does not exist
Invalid ONTAP version: '<version>' is not a recognized ONTAP version
Downgrade is not supported. Target version must be greater than or equal
to the current version.
```

Solutions:

- Verify the target version string format (e.g. "9.16.1")

- Ensure that the target version is newer than or equal to the current cluster version
- Use "current" to keep the existing ONTAP version: `--target-ontap-version=current`

Issue: Signature verification fails

Symptoms:

```
Signature verification failed
Invalid code signature
```

Cause: Downloaded update file might be corrupted or have been tampered with.

Solution:

- Manual update - download from the NetApp support site
- Verify signature manually:

```
openssl dgst -sha256 -verify UHC-Linux-public.pub -signature uhc.sig uhc
```

Issue: Telemetry upload failure

Symptoms:

```
body.7z upload failed
Telemetry endpoint is not reachable
```

Solutions:

- Check connectivity to the telemetry endpoint:

```
./uhc --test-connectivity telemetry
```

Issue: "UPDATE IN PROGRESS" lock file

Symptoms:

```
UPDATE IN PROGRESS
```

```
Another instance of UHC auto-update is currently running.
Please wait for the update to complete before running again.
```

Cause: Lock file exists from previous update process.

Solutions:

1. **Wait:** Update typically completes in 1-2 minutes.
2. **Check, if stale:** (automatic cleanup after 1 hour):

```
ls -la uhc_update.lock
# If older than 1 hour, it will be auto-cleaned
```

3. **Manual cleanup, if necessary:**

```
rm uhc_update.lock
```

FAQ for Upgrade Health Checker

Frequently asked questions (FAQ) for Upgrade Health Checker.

How does Upgrade Health Checker differ from Upgrade Advisor?

Upgrade Health Checker is an onsite tool suitable for users with large or complex environments, limited access to the internet, or existing automation and orchestration frameworks. Upgrade Advisor is suitable for users with a smaller environment who prefer a cloud-based, UI-centered experience.

You can learn more about about Upgrade Health Checker and Upgrade Advisor to determine which tool best fits your needs.

What information does an Upgrade Health Checker report include?

Upgrade Health Checker reports provide detailed information on potential blockers and warnings that need to be addressed before upgrading your ONTAP clusters, risks associated with your clusters, and information about new features and enhancements in the ONTAP version you're upgrading to.

Can Upgrade Health Checker be used without an internet connection?

You don't need an internet connection to run Upgrade Health Checker. It generates upgrade plans and reports onsite using your ONTAP cluster's current configuration data.

However, internet access is required for automatic tool updates. If your environment lacks internet connectivity, you can manually download the latest version of the tool from the Upgrade Health Checker tile on the NetApp Console Automation Hub.

What versions of ONTAP does Upgrade Health Checker support?

Upgrade Health Checker supports upgrades for on-premises ONTAP 9.11.1 and later.

Legal notices for Upgrade Health Checker

Legal notices provide access to copyright statements, trademarks, patents, and more.

Copyright

https://www.netapp.com/company/legal/copyright/

Trademarks

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

https://www.netapp.com/company/legal/trademarks/

Patents

A current list of NetApp owned patents can be found at:

https://www.netapp.com/pdf.html?item=/media/11887-patentspage.pdf

Privacy policy

https://www.netapp.com/company/legal/privacy-policy/

Open source

Notice files provide information about third-party copyright and licenses used in NetApp software.

Copyright information

Copyright © 2026 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at http://www.netapp.com/TM are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.