

Upgrade ExtraHop firmware through the REST API with ExtraHop Cloud Services

Published: 2024-09-04

You can automate upgrades to the firmware on your appliances through the ExtraHop REST API. You can also view available firmware versions and download firmware directly to your appliance through ExtraHop Cloud Services.



Note: Before you can complete the steps in this guide, the appliance must be connected to ExtraHop Cloud Services. For information on upgrading without ExtraHop Cloud Services, see [Upgrade ExtraHop firmware through the REST API](#).

While the firmware upgrade process is similar across all ExtraHop appliances, some appliances have additional considerations or steps that you must address before you install the firmware in your environment. If you need assistance with your upgrade, contact ExtraHop Support.

All appliances must meet the following requirements:

- The firmware version must be compatible with your appliance model.
- The firmware version on your appliance must be supported by the upgrade version.
- Command appliances must be running firmware that is greater than or equal to their connected appliances.
- Discover appliances must be running firmware that is greater than or equal to connected Explore and Trace appliances.

If your deployment only includes a sensor, proceed to the [API Explorer](#) or [Python](#) upgrade instructions.

If your deployment includes additional appliance types, you must address the following dependencies before proceeding with the upgrade instructions.

If your deployment includes...	Pre-upgrade tasks	Upgrade order
Command appliances	Reserve a maintenance window of an hour for Command appliances managing 50,000 devices or more.	<ul style="list-style-type: none"> • Command appliance • Discover appliances • All Explore appliances (manager nodes, then data nodes)
Explore appliances	See Upgrading ExtraHop recordstores .	<ul style="list-style-type: none"> • Trace appliances
Trace appliances	None	

Upgrade ExtraHop firmware through the REST API Explorer

View available firmware versions

1. In a browser, navigate to the REST API Explorer.
The URL is the hostname or IP address of your sensor or console, followed by `/api/v1/explore/`. For example, if your hostname is `seattle-eda`, the URL is `https://seattle-eda/api/v1/explore/`.
2. Click **Enter API Key** and then paste or type your API key into the **API Key** field.
3. Click **Authorize** and then click **Close**.
4. Click **ExtraHop**.
5. Click **GET /extrahop/firmware/next**.
6. Click **Try it out**.

7. Click **Send Request**.

In the Response body section, view the versions for each release and make a note of the version you want to upgrade to.



Note: For sensors, the list includes only firmware versions that are compatible with the version running on the connected console.

Download firmware and upgrade the appliance

1. Click **POST /extrahop/firmware/download/version**.
2. Click **Try it out**.
3. In the body field, specify the following fields:
 - **version:** The firmware version you want to upgrade to.
 - **upgrade:** Indicates whether to upgrade the appliance after the firmware download completes. Set this field to true.

The following body field upgrades the appliance to firmware version 8.7.1.1282:

```
{
  "upgrade": true,
  "version": "8.7.1.1282"
}
```

4. Click **Send Request**.

In the Response headers, note the value after the last forward slash in the `location` header. You will need this value to monitor the progress of the upgrade job. For example, the job ID in the following example is `ebdbbc9e-7113-448c-ab9b-cc1ec2407702`:

```
/api/v1/jobs/ebdbbc9e-7113-448c-ab9b-cc1ec2407702
```

Monitor the progress of the upgrade job

1. Click **Jobs**.
2. Click **GET /jobs/{id}**.
3. In the id field, type the value you copied from the `location` header in the previous task.
4. Click **Send Request**.
5. In the Response body, view information about the job.
The `status` field is `DONE` when the job is complete.

Retrieve and run the example Python script

The ExtraHop GitHub repository contains an example Python script that upgrades multiple appliances by reading URLs and API keys from a CSV file.



Note: The script does not automatically disable record ingest for ExtraHop recordstores. You must manually disable record ingest before running the script for an ExtraHop recordstores.



Important: The example python script authenticates to the sensor or console through an API key, which is not compatible with the RevealX 360 REST API. To run this script with RevealX 360, you must modify the script to authenticate with API tokens. See the [py_rx360_auth.py](#) script in the ExtraHop GitHub repository for an example of how to authenticate with API tokens.

1. Go to the [ExtraHop code-examples GitHub repository](#) and download the contents of the `upgrade_system_cloud` directory to your local machine.

- In a text editor, open the `systems.csv` file and replace the example values with the hostnames and API keys of your appliances.
- Run the `upgrade_system_cloud.py` script.

You must specify one of the following arguments:

--latest-hotfix

Upgrades the system to the latest hotfix on the release that is currently installed on the ExtraHop system. For example, if the system is running version 8.7.1, and versions 8.7.2, 8.7.3, 8.8.0, and 8.8.1 are available, the system is upgraded to 8.7.3.

--latest-release

Upgrades the system to the latest version on the latest release that the system can be upgraded to. For example, if the system is running version 8.7.1, and versions 8.7.2, 8.7.3, 8.8.0, and 8.8.1 are available, the system is upgraded to 8.8.1.

--version {str}

Upgrades the system to the specified version.

The following arguments are optional:

--force

Does not prompt for confirmation before upgrading.

--max-threads {int}

Specifies the maximum number of concurrent threads. The default value is 2.

--wait {float}

Specifies the number of minutes to wait before checking the progress of an upgrade job. The default value is 0.5.

For example, the following command upgrades appliances to the latest hotfix on the release that is currently installed:

```
python3 upgrade_system_cloud.py --latest-hotfix
```



Note: If the script returns an error message that the TLS certificate verification failed, make sure that **a trusted certificate has been added to your sensor or console** [🔗](#). Alternatively, you can add the `verify=False` option to bypass certificate verification. However, this method is not secure and is not recommended. The following code sends an HTTP GET request without certificate verification:

```
requests.get(url, headers=headers, verify=False)
```

The system displays output similar to the following text:

```
The following systems will be upgraded:
extrahop.example.com: 8.7.5.2941
extrahop.example2.com: 8.7.5.2941
Do you want to continue?
y/n
```

To upgrade the system, type `y` and then press ENTER.

Upgrading ExtraHop recordstores

Pre-upgrade tasks

Before upgrading an ExtraHop recordstore, you must halt record ingest. You can halt record ingest for all of the nodes in a cluster from a single node.



Note: The message Could not determine ingest status on some nodes and Error might appear on the Cluster Data Management page in the Administration settings of the upgraded nodes until all nodes in the cluster are upgraded. These errors are expected and can be ignored.

1. Open a terminal application.
2. Run the following command, where `YOUR_KEY` is the API for your user account, and `HOSTNAME` is the hostname of your ExtraHop recordstore:

```
curl -X PATCH "https://HOST/api/v1/extrahop/cluster" -H "accept: application/json" -H "Authorization: ExtraHop apikey=YOUR_KEY" -H "Content-Type: application/json" -d '{"ingest_enabled": false}'
```

Post-upgrade tasks

After you have upgraded all of the nodes in the recordstore cluster, enable record ingest.

1. Open a terminal application.
2. Run the following command, where `YOUR_KEY` is the API for your user account, and `HOSTNAME` is the hostname of your ExtraHop recordstore:

```
curl -X PATCH "https://HOST/api/v1/extrahop/cluster" -H "accept: application/json" -H "Authorization: ExtraHop apikey=YOUR_KEY" -H "Content-Type: application/json" -d '{"ingest_enabled": true}'
```