



Red Hat build of Cryostat 2

Release notes for the Red Hat build of Cryostat 2.1

Red Hat build of Cryostat 2 Release notes for the Red Hat build of Cryostat 2.1

Legal Notice

Copyright © 2023 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

Abstract

The Release notes for the Red Hat build of Cryostat 2.1 document provides an overview of new features in Cryostat 2.1 and a list of potential known issues and possible workarounds.

Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	5
CHAPTER 1. SUPPORT POLICY FOR CRYOSTAT	6
CHAPTER 2. NEW FEATURES	7
Automated rules user console (UI)	7
Attachment of metadata and labels to JFR recordings	7
Control of client-side notifications	7
Custom target resource definition	7
Environment variables for the Cryostat Operator	8
JMC Agent plug-in support	8
Red Hat OpenShift authentication for Cryostat 2.1	8
Red Hat OpenShift credentials	9
Manage Java Management Extensions credentials in Cryostat 2.1	9
New automated rules environment variables	9
Resource requirements	9
Sidecar report container	10
CHAPTER 3. FEATURE ENHANCEMENTS	11
Archives view	11
cert-manager API	11
Create Target dialog box	11
Cryostat Operator service customization	12
Cryostat Operator details	12
Download file behavior of Cryostat	13
File upload functionality	13
jfr-datasource container	14
Netty performance regression	14
Red Hat OpenShift cluster connections with external JVMs	15
RecordingPostHandler behavior change	15
Security menu item	15
setCachedTargetSelect implementation	16
Username on GUI masthead	16
WebSocket API	16
CHAPTER 4. UNSUPPORTED AND DEPRECATED FEATURES	17
ContainerJFR variants	17
CHAPTER 5. KNOWN ISSUES	18
Cryostat Operator upgrade	18
Deleted entries in the Archived Recordings table	18
Cryostat's API handling behavior	18
Duplicate file name displays under the Archived Recordings table	20
CHAPTER 6. ADVISORIES RELATED TO THIS RELEASE	21

PREFACE

The Red Hat build of Cryostat is a container-native implementation of JDK Flight Recorder (JFR) that you can use to securely monitor the Java Virtual Machine (JVM) performance in workloads that run on an OpenShift Container Platform cluster. You can use Cryostat 2.1 to start, stop, retrieve, archive, import, and export JFR data for JVMs inside your containerized applications by using a web console or an HTTP API.

Depending on your use case, you can store and analyze your recordings directly on your Red Hat OpenShift cluster by using the built-in tools that Cryostat provides or you can export recordings to an external monitoring application to perform a more in-depth analysis of your recorded data.



IMPORTANT

Red Hat build of Cryostat is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process. For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#) (Red Hat Customer Portal).

MAKING OPEN SOURCE MORE INCLUSIVE

Red Hat is committed to replacing problematic language in our code, documentation, and web properties. We are beginning with these four terms: master, slave, blacklist, and whitelist. Because of the enormity of this endeavor, these changes will be implemented gradually over several upcoming releases. For more details, see [our CTO Chris Wright's message](#).

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate your feedback on our documentation. To provide feedback, you can highlight the text in a document and add comments. Follow the steps in the procedure to learn about submitting feedback on Red Hat documentation.

Prerequisites

- Log in to the Red Hat Customer Portal.
- In the Red Hat Customer Portal, view the document in **Multi-page HTML** format.

Procedure

1. Click the **Feedback** button to see existing reader comments.



NOTE

The feedback feature is enabled only in the **Multi-page HTML** format.

2. Highlight the section of the document where you want to provide feedback.
3. In the prompt menu that opens near the text you selected, click **Add Feedback**.
A text box opens in the feedback section on the right side of the page.
4. Enter your feedback in the text box and click **Submit**.
You have created a documentation issue.
5. To view the issue, click the issue tracker link in the feedback view.

CHAPTER 1. SUPPORT POLICY FOR CRYOSTAT

Red Hat supports a major version of Cryostat for a minimum of 6 months. Red Hat bases this figure on the time that the product gets released on the Red Hat Customer Portal.

You can install and deploy Cryostat on Red Hat OpenShift Container Platform 4.8 or a later version that runs on an x86_64 architecture.

Additional resources

- For more information about the Cryostat life cycle policy, see [Red Hat build of Cryostat](#) on the Red Hat OpenShift Container Platform Life Cycle Policy web page.

CHAPTER 2. NEW FEATURES

Cryostat 2.1 introduces new features that enhance your use of the Cryostat product.

Automated rules user console (UI)

Cryostat 2.1 includes a user console (UI) for the automated rules API. The UI includes the following key features:

- A form view that simplifies user interaction with the API by supporting typed inputs.
- A match expression wizard, where you can create custom match expressions that target specific applications.
- A JSON format view for the selected target application, so that you can view high-level application information. This information forms an integral part of the creation of match expressions.
- A color-coded response system that indicates whether your expression matches the selected target application.



NOTE

The match expression wizard includes an example of a custom match expression that you can reference if you need to familiarize yourself with the custom match expression syntax.

Attachment of metadata and labels to JFR recordings

When you create a JFR recording on Cryostat 2.1, you can add metadata with key-value label pairs to the recording. Additionally, you can attach custom labels to JFR recordings that are inside a target JVM, so that you can easily identify and better manage your JFR recordings.

Use cases for metadata and labels include running queries or batch operations on recordings. You can navigate to the **Recordings** menu on the Cryostat web console and edit the label and its metadata of your JFR recording. You can also edit the label and metadata for a JFR recording that you uploaded to archives.

Control of client-side notifications

Cryostat 2.1 broadcasts notifications for all actions and state changes that can occur, resulting in a greater number of notifications appearing in the Cryostat web client. As a result, the **Settings** page has a control for client-side notifications.

Cryostat 2.1 users can enable or disable notifications by category and bulk enable or disable all graphical notifications.

The Cryostat backend still sends the notification messages, and the web client receives them. Disabling notifications prevents the messages from appearing on the console. Enabling the notifications again allows you to read any previous notifications.

Custom target resource definition

You can now create a custom target resource definition for your Cryostat instance. This allows the Cryostat Operator to connect to target applications by using a JMX protocol other than the Cryostat default protocol. The default protocol is typically JMX-RMI.

The custom target resource definition consists of a YAML file, where you can specify any of the following attributes for the definition:

- **alias**, which sets an optional name for the resource definition.
- **annotation.cryostat**, which defines optional annotations for the definition. An automated rule can use these annotations to apply a rule to a target JVM.
- **connectUrl**, which specifies a target URL, such as a JMX service URL or a **host:port** pair, that Cryostat must use when it opens a JMX connection to a target JVM application. This attribute is mandatory.

When you create a custom target object, the Cryostat Operator uses a RESTful API endpoint, **POST /api/v2/targets**, for the object. After the object is created, the object's **connectUrl** can be used as a **targetId** URL parameter in the REST HTTP API.

You can use **TargetDeleteHandler** to remove a custom target resource definition from the Cryostat Operator. This handler reads a **DELETE /api/v2/targets/:connectUrl** endpoint request and attempts to remove the definition from the Cryostat Operator.

Both **TargetsPostHandler** and **TargetDeleteHandler** include coded error messages that provide detailed error messages if a handler cannot handle a request.

Environment variables for the Cryostat Operator

Cryostat 2.1 includes the following environment variables that you can set to change the behavior of the Cryostat Operator:

- **CRYOSTAT_REPORT_GENERATION_MAX_HEAP**: Defaults to **200** MiB. Sets the maximum heap size that is used by the container subprocess to generate an automated rules analysis report.
- **CRYOSTAT_MAX_WS_CONNECTIONS**: Defaults to **unlimited**. Sets the maximum number of WebSocket client connections that your Cryostat application can support.
- **CRYOSTAT_TARGET_CACHE_SIZE**: Default to **-1**, which indicates an unlimited caching. Sets the maximum number of JMX connections that the OpenShift Operator can cache to your Cryostat application.
- **CRYOSTAT_TARGET_CACHE_TTL**: Defaults to **10**, which indicates the amount of time in seconds that a JMX connection remains cached in your Cryostat instance's memory.

JMC Agent plug-in support

Cryostat 2.1 supports the JMC Agent Plugin by using a set of API handlers for managing probe templates.

After you have installed a JMC Agent application and then built it to produce a JAR file, you can access the agent functionality for your Cryostat application by using the JMC Agent Plugin. This plug-in provides JMC Agent functionality to your Cryostat instance, such as adding JDK Flight Recorder (JFR) functionality to a running application.

Red Hat OpenShift authentication for Cryostat 2.1

Cryostat 2.1 integrates the Red Hat OpenShift built-in OAuth server into its framework. When enabled, users can log in to Cryostat by using their Red Hat OpenShift username and password. This integrated capability provides a better mechanism than that offered in Cryostat 2.0, where you had to manually copy your Red Hat OpenShift authorization token from the Red Hat OpenShift web console and then paste the token's details in the Cryostat **Application URL** section on the console.

Additionally, you can limit access to Cryostat features by using role-based access control (RBAC) roles that were assigned in Red Hat OpenShift.

The Cryostat 2.1 release includes the following keys for the **GET /health** response object:

- **DATASOURCE_CONFIGURED**
- **DASHBOARD_CONFIGURED**
- **REPORTS_CONFIGURED**
- **REPORTS_AVAILABLE**

Red Hat OpenShift credentials

After you log in to the Cryostat 2.1 web console from the Red Hat OpenShift web console, the Cryostat Operator temporarily stores your username and password credentials on your Red Hat OpenShift account for the duration of the session. This prevents your Cryostat web console session from ending before you log out of the Cryostat web console.

Manage Java Management Extensions credentials in Cryostat 2.1

You can store and manage the Java Management Extension (JMX) credentials that are used to authenticate to containerized Java Virtual Machines (JVM). This feature is useful when you want Cryostat to remember and reuse your credentials for multiple JVMs.

When you add JMX credentials to Cryostat, you cannot view the credentials anymore. This keeps your credentials secure, because your credentials do not remain visible after you enter them on the Cryostat web console. If you want to replace the credentials, you must delete the credentials and add them again.

New automated rules environment variables

Before Cryostat 2.1, JMX connections for automated rules would close if these JMX connections were previously cached to your Cryostat instance's memory. This issue occurred because of the behavior of the **CRYOSTAT_TARGET_CACHE_MAX_CONNECTIONS** environment variable.

The JMX cache component for Cryostat 2.1 now uses the **CRYOSTAT_TARGET_CACHE_SIZE** environment variable instead of the **CRYOSTAT_TARGET_CACHE_MAX_CONNECTIONS** environment variable. This means that any JMX connections that you open do not get automatically cached by automated rules to your Cryostat instance's memory. This prevents automated rules from overfilling the cache storage space and causing in-use JMX connections to close, which can lead to degradation of performance with increased latency and response times.

The **CRYOSTAT_TARGET_CACHE_SIZE** environment variable specifies the maximum number of JMX connections to cache in your Cryostat instance's memory. You can specify the following values for this environment variable:

- < 0 : default value is **-1**. Values less than **0** indicate an unlimited cache size. This means that JMX connections only get removed from memory when they reach an inactivity limit.
- **0**: A value of zero indicates JMX connections are immediately removed from memory when these connections are closed.
- > 0 : Values greater than **0** indicates that a Cryostat instance can cache a set number of JMX connections in its memory. If a new connection is created when the cache amount has reached its level, the oldest JMX connection is closed and removed from memory to facilitate storage of the new connection.

Automated rules can re-use any previously cached JMX connections. If no JMX connection exists then the Cryostat Operator creates a new JMX connection for the automated rule. This connection does not get cached to memory.

Resource requirements

By default, the Cryostat Operator deploys your Cryostat application without specifying any resource requests or limits for each of the three containers that operate in your Cryostat instance's main pod on Red Hat OpenShift. Cryostat 2.1 includes a capability where you can use a Cryostat custom resource (CR) to specify resource requests or limits for each of the following three containers:

- **core**, which runs the Cryostat backend service and the web application.
- **datasource**, which runs the JFR data source that converts your JFR recordings into a file format that is supported by Grafana.
- **grafana**, which runs the Grafana instance that is associated with your Cryostat application.

Sidecar report container

With Cryostat 2.1, you can use the sidecar report container to generate automated analysis reports for JDK flight recordings (JFR).

Before Cryostat 2.1, you had to rely on the main Cryostat container to generate analysis reports. This approach is resource intensive and could impact the performance of running your Cryostat application because you might need to provision additional resources for the main Cryostat container.

By generating analysis reports in the sidecar report container, you can efficiently use the Cryostat Operator to provision resources for your Cryostat application. This provides your Cryostat container with a lower resource footprint, because the Cryostat Operator that interacts with the container can focus on running low-overhead operations over HTTP and JMC connections.

Additionally, you can duplicate a sidecar report container and then configure this duplicate to meet your needs.

CHAPTER 3. FEATURE ENHANCEMENTS

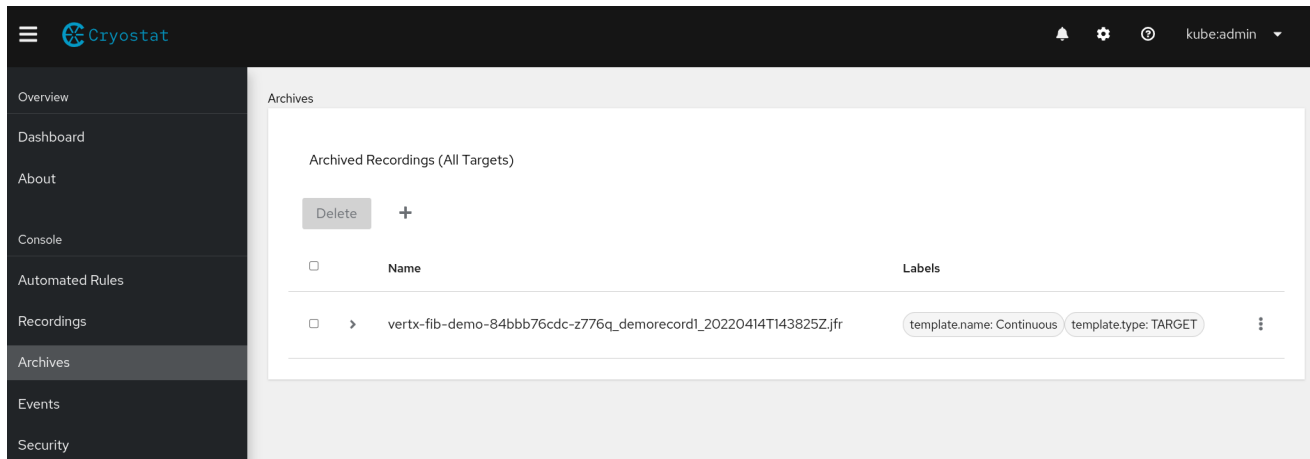
Cryostat 2.1 includes feature enhancements that build upon the Cryostat 2.1 offerings.

Archives view

The Cryostat 2.1 web console includes an **Archives** menu item. After you select this menu item, an **Archives Recording** table displays on your console. This table improves upon the Cryostat 2.0 table in that it adheres to a split view and uses a GraphQL query to populate table data.

The **Archived Recordings** table on the **Archives** menu item is different from the **Archived Recordings** table that displays on the **Recordings** menu in that it displays archives for all target JVMs.

Figure 3.1. Archives view on the Cryostat web console



cert-manager API

Cryostat 2.1 supports version **1.5.3**, so that the Cryostat Operator now uses the cert-manager API to set TLS certificates for a target JVM.

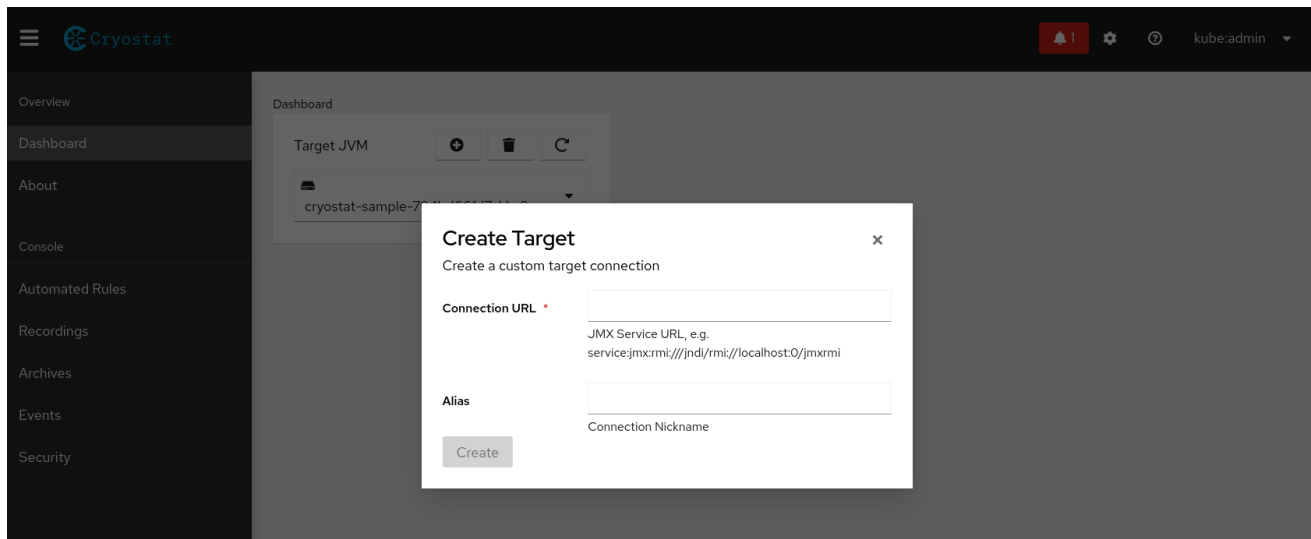
See, [v1.5.3 \(cert-manager\)](#)

Create Target dialog box

Cryostat 2.1 disables the **Create** button on the **Create Target** dialog box until you enter a value in the **Connection URL** field.

Additionally, the **Connection URL** field includes an example of a JMX Service URL that you can refer to when you need to enter a valid URL in the field.

Figure 3.2. The Create Target dialog box in the Dashboard menu item



Cryostat Operator service customization

The Cryostat Operator now includes a **spec.serviceOptions** property in its YAML configuration file, so that you can change the following service options for the operator:

- Annotations
- Labels
- Port numbers
- Service type

After you make changes to the default service option values, the Cryostat Operator creates services for the following components:

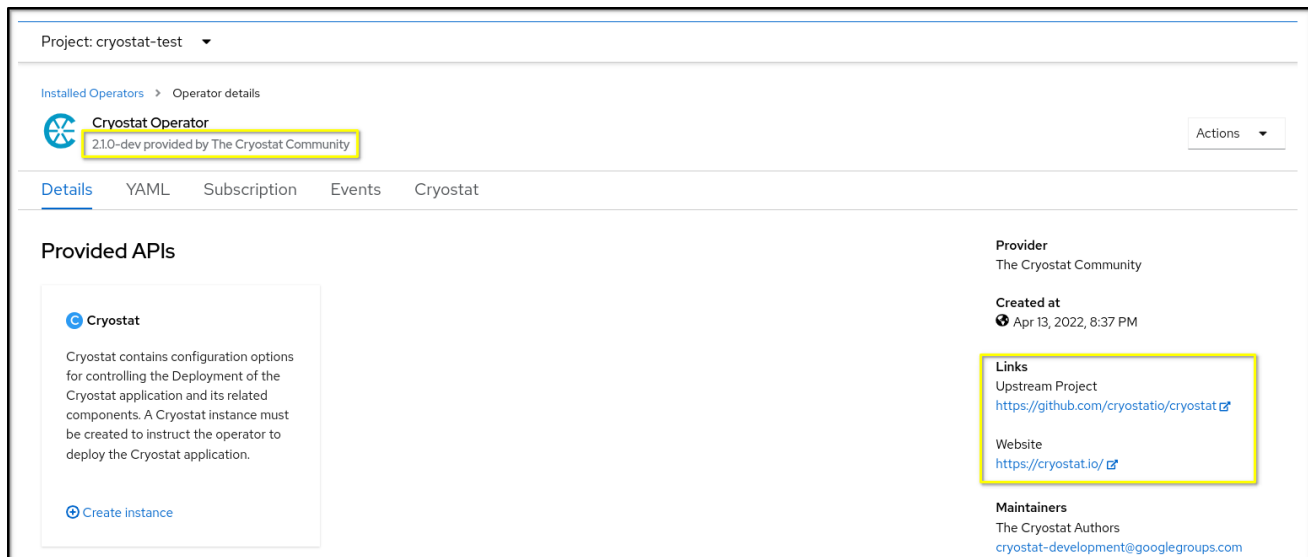
- Cryostat application
- Grafana application
- Report generator microservice

Cryostat Operator details

The Cryostat Operator details page on the OpenShift Container Platform (OCP) web console includes the following enhancements:

- An updated name reference for the Cryostat Operator. Before the Cryostat 2.1 release, the Cryostat application and Cryostat Operator were named similarly on OCP.
- A link to the Cryostat website

Figure 3.3. New enhancements on the Cryostat Operator details page



Download file behavior of Cryostat

When you downloaded a file from your Cryostat 2.0 web console, such as selecting the **Download Recording** item from the **Active Recordings** overflow menu, you would need to complete the following steps:

1. Download the remote file into your default web browser's memory.
2. Create a local object URL for the blob file item.

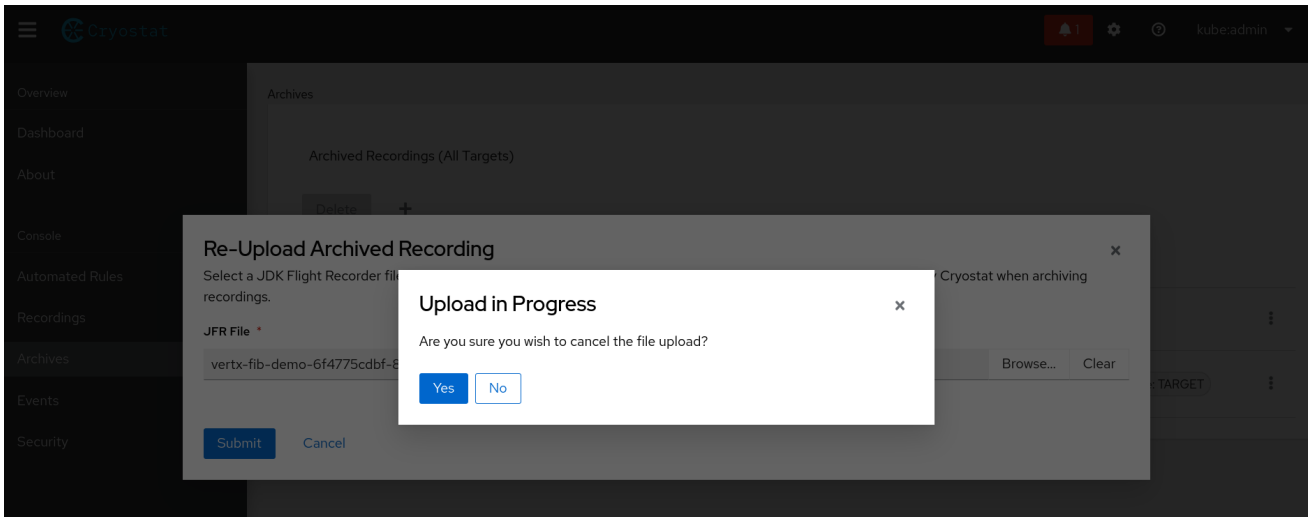
This behavior voided the purpose of the **Cancel** option from your web browser's downloads menu. This might be problematic if you wanted to cancel the download operation of the JFR binary file.

Cryostat 2.1 relies on the HTML 5 **download** attribute that is available with your web browser to manage a file download. This attribute reads the anchor element from the **href** attribute and then instructs your web browser to download the file. This download operation decreases the time it takes for your web browser to display the **Save File** menu, so that you can choose to cancel the download operation before saving the file to your local system.

File upload functionality

During a large file upload operation on Cryostat 2.1, such as re-uploading an archived recording from the **Re-Upload Archived Recordings** dialog box, you can click the **Cancel** button to stop the file upload operation. An **Upload in Progress** dialog box displays, where you must choose to proceed with the cancel operation.

Figure 3.4. Cancel button on the Re-Upload Archived Recordings dialog box

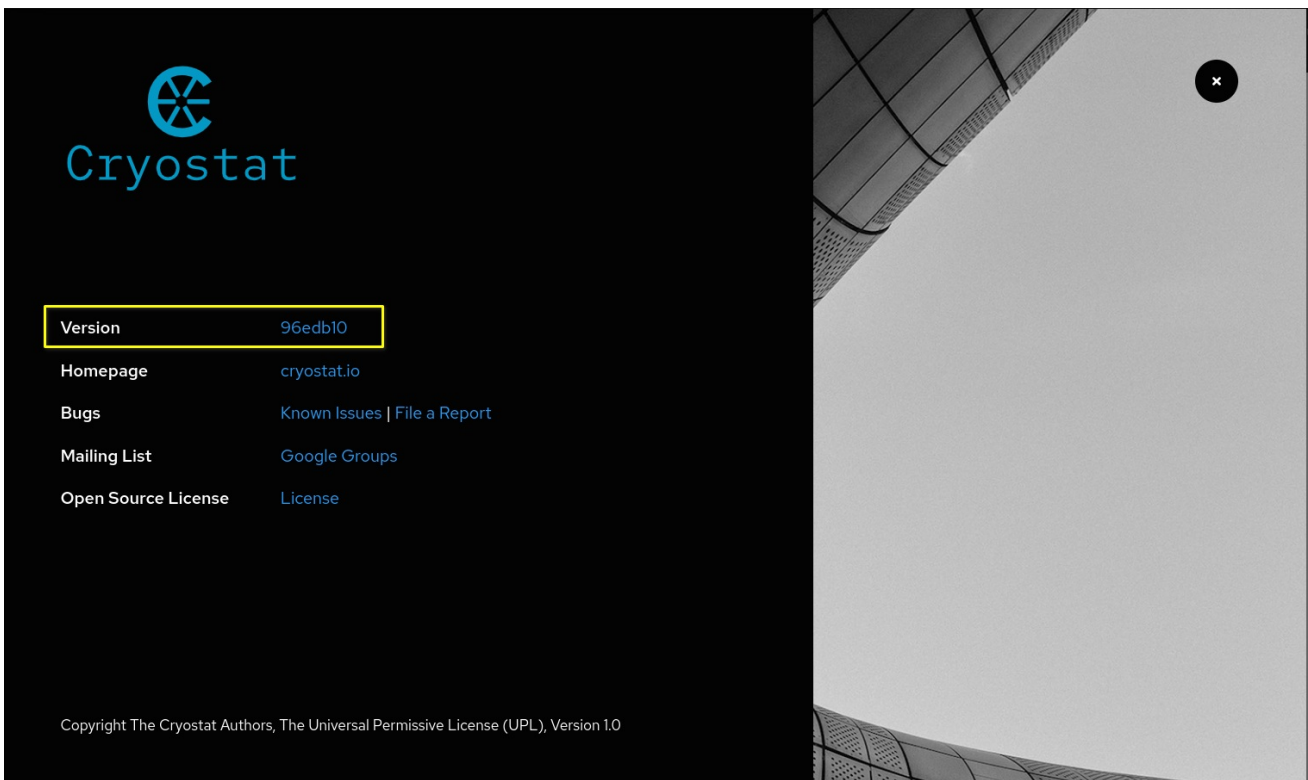


After the cancel operation completes, your web browser displays the size of the JFR file that was not transferred to your Cryostat application.

jfr-datasource container

Cryostat 2.0 contained an issue where the Cryostat web console would not display the version number on the **About** page. Cryostat 2.1 fixes this issue by changing the codebase to ensure that the version number displays on this page, regardless of the **jfr-datasource** or **grafana** dashboard configuration settings.

Figure 3.5. About page on the Cryostat web console



Netty performance regression

Handler implementations that use the Vert.x **BodyHandler** class no longer experience the performance issues that were evident in Cryostat 2.0, such as accepting file uploads where a standard HTTP form upload was expected by a handler. These file uploads could lead to resource constraints for Vert.x,

because the handlers might permanently store such files in the temporary file-upload storage location on Vert.x. Additionally, Netty's parsing of **POST** form bodies could lead to higher than expected memory usage while processing API requests.

Cryostat 2.1 uses Vert.x version **3.9.9**, which includes an upgrade to Netty, version **4.1.67**. This upgrade improves both the speed and logic on how handlers upload files to Vert.x.

Red Hat OpenShift cluster connections with external JVMs

Cryostat 2.0 had a known issue with connecting an Red Hat OpenShift cluster with JVMs located on nodes different to those running on a Cryostat node.

Cryostat 2.1 resolves this issue with the new **CRYOSTYAT_ENABLE_JDP_BROADCAST** environment variable, which is set to **false** as default. The default configuration of this environment variable disables the Java Discovery Protocol (JDP) on Red Hat OpenShift, so that Cryostat 2.1 can now connect to JVMs that are located on any nodes.

See, [Known issues \(Cryostat 2.0\)](#)

RecordingPostHandler behavior change

Cryostat 2.1 enhances the **RecordingPostHandler** implementation so that it now sequentially parses a JFR binary. The implementation in Cryostat 2.0 parsed data and then constructed a list of events.

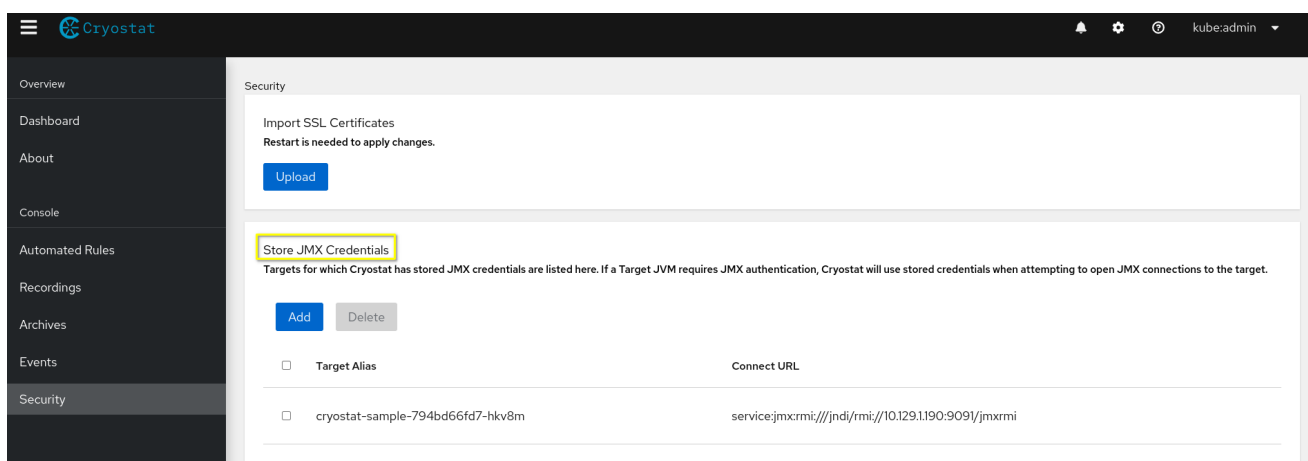
The new implementation has the following advantages:

- Provides a simple method
- Requires less resources to run
- Validates uploaded data much faster than the previous behavior

Security menu item

After you select the **Security** menu item on your Cryostat 2.1 instance, you can access the **Store JMX Credentials** tile.

Figure 3.6. Store JMX Credentials on the Security menu item



The **Store JMX Credentials** tile provides a convenient way to easily view any target JVMs that have stored JMX credentials.

Additionally, on this tile item, you can add stored credentials to a specific target JVM. For a target JVM that requires JMX authentication, you must provide your username and password when prompted. Cryostat can use stored credentials when attempting to open JMX connections to a target JVM.

setCachedTargetSelect implementation

Before the Cryostat 2.1 release, when you logged into your Cryostat web console and navigated to the **Dashboard**, the JVM you selected from your previous session would display as the default value under the **Target JVM** drop-down list. This would occur even if Cryostat can no longer connect to this JVM.

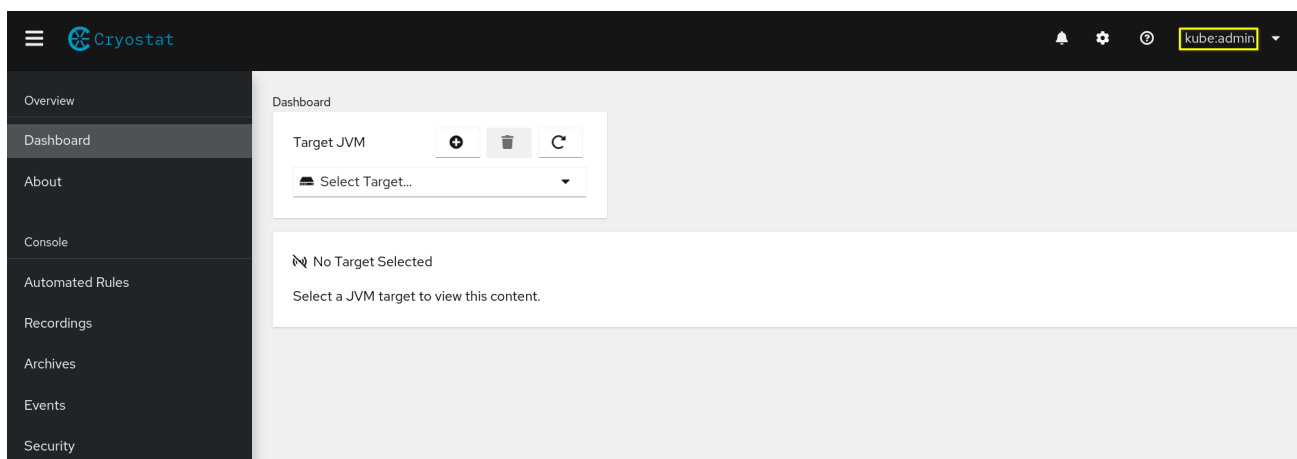
Cryostat 2.1 resolves this issue by refreshing the list of target JVMs at the start of each new session and then only lists JVMs where it can establish a connection.

You can configure the refresh period for your Cryostat web console by navigating to **Settings > Auto-Refresh**. In the provided field, you can specify a value in seconds, minutes, or hours. You must select the **Enable** checkbox to complete the configuration.

Username on GUI masthead

Cryostat 2.1 fetches a username from the `/v2.1/auth` endpoint, so it can display a username in the Cryostat web console masthead. In Cryostat 2.0, you could only view your username when you start a Cryostat instance in basic authentication mode.

Figure 3.7. Username displaying on the masthead of the Cryostat web console



WebSocket API

Cryostat 2.1 updates its WebSocket API to support unlimited WebSocket client connections. Before this release, the WebSocket API could only support a maximum of 64 client connections.

For Cryostat 2.1, the WebSocket API can now automatically receive information about actions performed by an unlimited number of connected clients that are using the same one-way push Notification Channel (NC) channel.

CHAPTER 4. UNSUPPORTED AND DEPRECATED FEATURES

Cryostat 2.1 removes some features because of their high maintenance costs, low community interest, and better alternative solutions.

ContainerJFR variants

Cryostat 2.1 removes the **TimeoutHandler** implementation. The Vertx server no longer uses this implementation to automatically end a delayed response message that was sent by a Cryostat request handler. The **WebClientAssetsHandler** and the **StaticAssetsHandler** now prevent these delayed response issues.

CHAPTER 5. KNOWN ISSUES

Sometimes a Cryostat release might contain an issue or issues that Red Hat acknowledges and might fix at a later stage during the product's development. Review each known issue for its description and its resolution.

Cryostat Operator upgrade

Description

When updating your Cryostat Operator subscription from Cryostat 2.0 to Cryostat 2.1, you must change the update channel from **stable-2.0** to **stable**.

After this upgrade process, some Red Hat OpenShift objects that were created by the OpenShift Operator from Cryostat 2.0 will contain some outdated definitions. These definitions can result in connection issues among Cryostat components when you interact with your Cryostat 2.1 instance.

Workaround

Remove Cryostat 2.0 services and deployments that exist in your Cryostat 2.1 instance.

Example that shows removal of Cryostat 2.0 services and deployments from a Cryostat 2.1 instance

```
oc project <cryostat_project>
cryostats=$(oc get cryostat --template \
'{{range .items}}{{.metadata.name}}{\n"}}{{end}}')
for cryostat in ${cryostats}; do
  oc delete svc,deploy -lapp="${cryostat}"
done
```

After you remove these definitions, your Cryostat Operator creates services and deployments by using the Cryostat 2.1 object definitions. This restores connectivity among Cryostat components, and preserves JFR recordings that were created by Cryostat 2.0 in Cryostat 2.1.

Additional resources

- See [Upgrading installed Operators \(Red Hat Red Hat OpenShift 4.8 documentation\)](#)

Deleted entries in the Archived Recordings table

Description

The **Archived Recordings** table on the **Recordings** menu item does not immediately remove any archived recordings that you have deleted.

Workaround

Select any menu item, except **Recordings**, and then select the **Recordings** menu item. The **Archived Recordings** table now displays all the available archived recordings.

Cryostat's API handling behavior

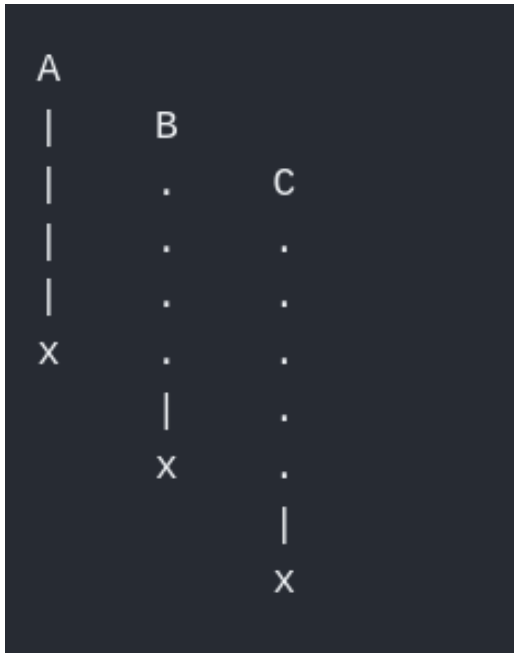
Description

Cryostat 2.1 does not concurrently handle API requests, which can lead to performance issues for your Cryostat instance.

In the following diagrams, the following symbols indicate actions:

- A parallel line (|): Cryostat is processing an API request.
- A dotted line (.): Cryostat has not yet processed a request.
- An "x": Cryostat has processed a request.

Figure 5.1. Example of the current behavior of how Cryostat handles API requests



API request A, a slow request, is running while a client issues API request B and then API request C, fast requests. Cryostat cannot process API request B until it has finished processing API request A. This activity is similar for how Cryostat handles API requests B and C.

Figure 5.2. Example of the expected behavior of how Cryostat handles API requests



The client sent each request at different time intervals. Cryostat processes API requests B and C before API request A, because Cryostat can process fast requests before any slow requests. This behavior reduces any performance issues with Cryostat, because Cryostat can concurrently handle API requests and send responses back to the client.

Workaround

Currently no workaround exists for this issue.

Duplicate file name displays under the Archived Recordings table

Description

An issue occurs under the **Archived Recordings** table on the **Recordings** menu item, when you re-upload a file with an identical file name to a file that already exists on the table.

If you attempt to delete or edit your re-uploaded file with this known issue, Cryostat applies the changes to both files. These changes can impact how you interact with the files, because you might be using Cryostat to analyze the incorrect file.



NOTE

This issue does not occur under the following scenarios:

- If you immediately archive two active recordings that relate to the same target JVM.
- If you re-upload an identical file more than once to Cryostat. Cryostat stores each re-uploaded file in a storage location that does not relate to a specific target JVM.

For both these scenarios, Cryostat appends a number to the second file to distinguish it from the first file.

The following directory structure example shows the **20220401T212052Z.jfr** archive recording file in a storage location that corresponds to the selected target JVM, **archive**. When you re-upload an identically named file from your local system to Cryostat's archives location, Cryostat stores the file in a storage location that does not correspond to any target JVM, **unlabelled**.

```

archive/
├── file-uploads
├──
├── ONSXE5TJMNSTU2TNPA5HE3LJHIXS6L3KNZSGSL3SNVUTULZPMNZHS33TORQXIORZGA4
├── TGL3KNV4HE3LJ
├── ┌── es-andrewazor-demo-Main_foo_20220401T212052Z.jfr
├── └── unlabelled
├──     ├── es-andrewazor-demo-Main_50mb_20211224T172734Z.2.jfr
├──     ├── es-andrewazor-demo-Main_50mb_20211224T172734Z.jfr
├──     └── es-andrewazor-demo-Main_foo_20220401T212052Z.jfr

```

Based on the previous example, the current behavior of the re-upload operation on the **Archived Recordings** table shows two files with the same name. Cryostat applies the same name to both files, because Cryostat identifies each file as having different storage locations.

The expected behavior of the re-upload operation on Cryostat would append a subsequent number to the re-uploaded file. This means that this file does not share the same name as the file that exists before it in the table.

Workaround

Currently no workaround exists for this issue.

CHAPTER 6. ADVISORIES RELATED TO THIS RELEASE

The following advisories have been issued to bugfixes and CVE fixes included in this release:

- [RHSA-2022:1679](#)

Revised on 2023-08-31 09:22:30 UTC