



## Red Hat build of Cryostat 2

# Release notes for the Red Hat build of Cryostat 2.4



# Red Hat build of Cryostat 2 Release notes for the Red Hat build of Cryostat 2.4

---

## 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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 2.4 document provides an overview of new features in Cryostat 2.4 and a list of potential known issues and possible workarounds.

# Table of Contents

|  |           |
|--|-----------|
| <b>PREFACE</b> .....   | <b>3</b>  |
| <b>MAKING OPEN SOURCE MORE INCLUSIVE</b> .....   | <b>4</b>  |
| <b>CHAPTER 1. SUPPORT POLICY FOR CRYOSTAT</b> .....                                      | <b>5</b>  |
| <b>CHAPTER 2. NEW FEATURES</b> .....   | <b>6</b>  |
| ARM64 architecture support   | 6         |
| Dynamic JFR recordings based on MBean custom triggers                                    | 6         |
| Agent HTTP API support for JFR write operations  | 6         |
| Dual HTTP and JMX discovery registration   | 6         |
| Standard agent JAR file without any dependencies   | 6         |
| Choice of agent JAR file distributions   | 7         |
| Default restriction on server host or IP address ranges that the agent can register with | 7         |
| <b>CHAPTER 3. FEATURE ENHANCEMENTS</b> .....   | <b>9</b>  |
| Topology view enhancements for displaying Cryostat agents                                | 9         |
| Topology and Dashboard views display additional target details                           | 9         |
| Automated Analysis enhancements for JFR recordings                                       | 9         |
| Parameter enhancements for restarting JFR recordings                                     | 9         |
| Enhanced error and warning messages  | 10        |
| <b>CHAPTER 4. FIXED ISSUES</b> .....   | <b>11</b> |
| Rule activation failures on late-connecting targets                                      | 11        |
| Discovery plug-in registration failures  | 11        |
| Server startup failures due to unavailable agent instances                               | 11        |
| Unwanted override of container-tuning properties by cryostat-reports and jfr-datasource  | 11        |
| Truncation of JFR data when restarting flight recordings by using automated rules        | 11        |
| Container startup failure due to small CPU limits  | 12        |
| <b>CHAPTER 5. KNOWN ISSUES</b> .....   | <b>13</b> |
| Cryostat agent cannot accept the ALL event template in recording start requests          | 13        |
| Red Hat Insights integration fails on ARM64 architecture                                 | 13        |
| <b>CHAPTER 6. ADVISORIES RELATED TO THIS RELEASE</b> .....                               | <b>15</b> |



## 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 2.4 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](#).

## 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](#).



## 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 4.11 or a later version that runs on an x86\_64 or ARM64 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

Crioostat 2.4 introduces new features that enhance your use of the Crioostat product.

### ARM64 architecture support

Crioostat 2.4 introduces support for using Crioostat on Red Hat OpenShift Container Platform 4.11 or a later version that runs on an ARM64 (aarch64) architecture.

### Dynamic JFR recordings based on MBean custom triggers

In Crioostat 2.4, you can enable the Crioostat agent to start JFR recordings dynamically based on MBean custom triggers.

A custom trigger condition is based on MBean counters that can cover a range of runtime, memory, thread, and operating system metrics. You can include one or more MBean counter types as part of the custom trigger condition for a JFR recording. You can also specify a duration or time period as part of the trigger condition, which means the conditional values must persist for the specified duration before the condition is met. Each custom trigger definition must also include an event template that the Crioostat agent can use for the recording.

The Crioostat agent supports smart triggers that continually listen to the value of the specified MBean counters. Triggering occurs if the current values of the specified counters match the configured values in the custom trigger for the specified duration. If triggering occurs, the Crioostat agent dynamically starts the JFR recording at that point based on the given event template.



#### NOTE

A JFR recording will not start dynamically if the custom trigger condition associated with this recording is not met.

### Agent HTTP API support for JFR write operations

The Crioostat 2.4 agent provides an HTTP API that the Crioostat server can use as an alternative to an application's JMX port. By attaching a properly configured Crioostat 2.4 agent to the workload applications that you deploy, you can use the full Crioostat feature set without any need for your target applications to expose a JMX port.

This enhancement enables the Crioostat agent to accept on-demand requests to start, stop, and delete JFR recordings for any target JVMs that have an agent HTTP connection. This supersedes the behavior in the previous release where the Crioostat agent provided a read-only HTTP API that supported a limited set of JFR operations only.



#### NOTE

By default, the Crioostat agent is not enabled to start and stop JFR recordings based on on-demand requests from the server. If you want the Crioostat agent to service these types of requests, you must set the **criostat.agent.api.writes-enabled** property to **true** in your target application's configuration.

### Dual HTTP and JMX discovery registration

If the Crioostat 2.4 agent detects that JMX is also configured on your target application, the agent publishes itself to the Crioostat server with both agent HTTP API definitions and JMX URL definitions. In this situation, you can use whichever configuration option you prefer.

### Standard agent JAR file without any dependencies

In Cryostat 2.4, you can choose to deploy the Cryostat agent by using a standard JAR file that contains the agent code without any dependencies. For more information, see [Choice of agent JAR file distributions](#).

### Choice of agent JAR file distributions

Cryostat 2.4 distributes two different variations of the Cryostat agent's JAR file. Depending on your setup requirements, you can use either of the following types of agent JAR file:

- An all-in-one "shaded" JAR file that is self-contained and includes the agent code and all of its dependencies  
This provides the most convenient form of Cryostat agent to include in your existing applications, because you need to include only one additional agent JAR file.
- A standard JAR file that contains the agent code without any dependencies  
This is useful if you know that dependency conflicts exist between the agent and your workload applications, and you intend to apply your own strategies to resolve any dependency conflicts.

This supersedes the behavior in the previous release, which provided only one distribution of the agent as an all-in-one "shaded" JAR file.

### Default restriction on server host or IP address ranges that the agent can register with

Cryostat 2.4 introduces a new **cryostat.agent.baseuri-range** system property that enables you to set limits around the resource location (URL) of the Cryostat server that the Cryostat agent advertises itself to. The Cryostat agent can use the **cryostat.agent.baseuri-range** property to check if the URL that you specified in the **cryostat.agent.baseuri** property is within an acceptable range of host names or IP addresses.

This feature helps to protect users from accidentally misconfiguring the server's base URI, which might otherwise result in the agent advertising itself to an incorrect service or a service outside your control. Misconfiguration of the server's URL could lead to a data leak from your Cryostat agent instance to a service outside of your cluster or data being routed over the open internet.

Depending on the level of security restrictions you want to apply, you can set the **cryostat.agent.baseuri-range** property to any of the following values:

- **loopback**  
This provides the highest level of restrictions. The agent connects to the server only if the server URL has an address such as **localhost** or **127.0.0.1**.
- **link\_local**  
This provides the second-highest level of restrictions. The agent connects to the server only if the server URL has an address in the link-local range or an address that is already within the highest level of restrictions.
- **site\_local**  
This provides the third-highest level of restrictions. The agent connects to the server only if the server URL has an IPv6 "site-local" or IPv4 "private" address or an address that is already within a higher level of restrictions.
- **dns\_local**  
This provides the fourth-highest level of restrictions. The agent connects to the server only if the server URL has an address with a **.local** or **.localhost** suffix or an address that is already within a higher level of restrictions.



## NOTE

This is the default setting. This level of restrictions suits Cryostat server and agent deployments in Red Hat OpenShift Container Platform, because the Cryostat server is typically referenced by an internal OpenShift Service address such as **`https://cryostat.my-namespace.svc.cluster.local`**.

- **public**

This removes any restrictions. This setting matches the behavior in the previous release where the agent could connect to any configured host or IP address.

## CHAPTER 3. FEATURE ENHANCEMENTS

Cryostat 2.4 includes feature enhancements that build upon the Cryostat 2.3 offerings.

### Topology view enhancements for displaying Cryostat agents

In the Topology view of the Cryostat web console, Cryostat agents are now displayed with the Cryostat icon. This helps to differentiate the Cryostat agents from JMX targets, which are still displayed with the OpenJDK Duke icon.

### Topology and Dashboard views display additional target details

The Topology and Dashboard views of the Cryostat web console now display the following additional types of information for your target JVMs:

- Operating system name
- Total physical memory
- Total swap space
- Class path
- Library paths
- Input arguments
- System properties



#### NOTE

You can view this information in either of the following ways:

- In the Topology panel, click the twistie (>) icons to expand down to a target endpoint and then click the **Details** tab for this endpoint.
- In the Dashboard panel, add the Target JVM Details card to the dashboard.

### Automated Analysis enhancements for JFR recordings

In the Cryostat 2.4 web console, the Automated Analysis information for a JFR recording reuses components from the Automated Analysis dashboard card. This supersedes the behavior in previous releases where Automated Analysis data provided a more limited set of information.



#### NOTE

To view Automated Analysis information for a JFR recording, open any panel that includes a list of recordings and click the twistie (>) icon next to the recording you are interested in.

### Parameter enhancements for restarting JFR recordings

When creating a JFR flight recording, the Cryostat server typically rejects requests to create a recording if another recording with the same name already exists on the same target. Clients can modify this behavior by specifying a request parameter to restart the recording.

In previous releases, clients could include a **restart** parameter in requests to create a recording. The **restart** parameter accepted **true** or **false** values and was set to **false** by default. If the client request included a **restart=false** setting, the Cryostat server rejected the request if another recording with the

same name already existed. If the client request included a **restart=true** setting, the Cryostat server would stop, delete, and recreate any existing recording with the same name.

Cryostat 2.4 introduces a new **replace** parameter, which supersedes the **restart** parameter that was available in previous releases. The **replace** parameter accepts **always**, **never**, or **stopped** values. In this situation, a setting of **replace=always** matches the old **restart=true** behavior, and a setting of **replace=never** matches the old **restart=false** behavior. The **replace=stopped** setting provides a third type of behavior, by instructing the server to recreate a recording with the same name only if the existing recording is in a **STOPPED** state. Otherwise, if the client request includes a **replace=stopped** setting, but an existing recording with the same name is in another state such as **RUNNING**, the server rejects the request.

This enhancement helps to avoid situations where enabling automated rules to recreate JFR recordings might lead to the loss of non-archived data for recordings that are not in a **STOPPED** state.

### Enhanced error and warning messages

Cryostat 2.4 includes enhancements to various error and warning messages.

## CHAPTER 4. FIXED ISSUES

The Cryostat release might include fixes for issues that were identified in earlier releases of Cryostat. Review each fixed issue note for a description of the issue and the subsequent fix.

The following issues have been fixed in the Cryostat 2.4 release:

### Rule activation failures on late-connecting targets

Before Cryostat 2.4, Cryostat might fail to trigger automated rules on discovered JVM targets. This issue could occur when the discovery mechanism (for example, OpenShift endpoints querying) informed Cryostat about a new target container before the JVM in the container was ready to accept incoming JMX requests.

Cryostat 2.4 resolves this issue by performing rechecks for a target JVM that was previously non-connectible and reattempting rule triggering at regular frequent intervals for a set period. This fix helps to resolve any connection issues between Cryostat and JVMs in a faster timeframe. This fix also helps to avoid inconsistent or unpredictable rule-triggering behavior when toggling between the enabling or disabling of rules.

### Discovery plug-in registration failures

Before Cryostat 2.4, instances of the Cryostat agent might fail to register with the Cryostat server. Even though the registration protocol has defined behaviors to help identify such failures and reset the registration, these defined behaviors might also fail in other situations. If both of these failures occurred, the Cryostat server would have an invalid agent registration record that the server considered to be valid. In this situation, the agent might make further attempts to register based on the assumption that the previous registration failed. However, the server would subsequently reject the agent's attempts to register again based on the assumption that the agent was already registered. This left the agent and server unable to recognise each other or to reset the registration status.

Cryostat 2.4 resolves this issue and provides more reliable agent discovery registration.

### Server startup failures due to unavailable agent instances

Before Cryostat 2.4, if you shut down and restarted the Cryostat server, the server might fail to restart successfully. This issue occurred if you shut down a server that had valid agent registration records and then you shut down these agent instances without restarting the agents. In this situation, at server startup, the server could see the preexisting agent registration records and expected these agent instances to still exist. However, because the agents were no longer available, the server startup subsequently failed.

Cryostat 2.4 resolves this issue. In this release, at server startup, if agent registration records still exist for agent instances that are no longer available, the server deletes these registration records and the server startup continues as normal.

### Unwanted override of container-tuning properties by `cryostat-reports` and `jfr-datasource`

The `cryostat-reports` and `jfr-datasource` containers both use the OpenJDK UBI runtime image. Before Cryostat 2.4, these containers were mistakenly configured to override the various container-tuning properties that the base image's entry-point script performed.

In Cryostat 2.4, the `cryostat-reports` and `jfr-datasource` containers no longer override the container-tuning properties, and these parameters are now active.

### Truncation of JFR data when restarting flight recordings by using automated rules

Before Cryostat 2.4, when a client sent a request to create a JFR recording by using automated rules, this might lead to unexpected truncation of JFR data for existing recordings. This could occur if the client request included a `restart=true` setting but an existing recording with the same name was not

already in a **STOPPED** state. In this situation, the server automatically stopped, deleted, and recreated the recording, which resulted in a loss of non-archived recording data.

Cryostat 2.4 resolves this issue by introducing a **replace** parameter, which supersedes the **restart** parameter that was available in previous releases. When you define an automated rule to create a JFR recording, you can now include a **replace=stopped** setting in the client request. This setting instructs the server to restart a JFR recording only if the existing recording is in a **STOPPED** state. If the existing recording is in another state such as **RUNNING**, the server rejects the request.

### **Container startup failure due to small CPU limits**

Before Cryostat 2.4, if Cryostat was deployed with small CPU limits, the container might fail to start.

Cryostat 2.4 resolves this issue by ensuring that the container can start even if Cryostat is deployed with small CPU limits.



## CHAPTER 5. KNOWN ISSUES

Sometimes a Crioostat 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.

### Crioostat agent cannot accept the ALL event template in recording start requests

#### Description

When creating a JFR recording for a target JVM that is using the Crioostat agent, if you select the ALL event template, the Crioostat agent returns an HTTP 400 error.

#### Workaround

Depending on your requirements, you can do either of the following:

- If you want to use the ALL event template, use a JMX connection with your target JVM.
- If you want to use a Crioostat agent connection with your target JVM, use a different event template such as Profiling.

### Red Hat Insights integration fails on ARM64 architecture

#### Description

In the Crioostat Operator's namespace, a pod whose name begins with **insights-proxy** is in the **ImagePullBackOff** error state. This is because the **APICast** container used for this pod is not built for the ARM64 architecture. However, Crioostat and all applications that are using the Crioostat agent should still function normally.

#### Workaround

To disable Red Hat Insights integration:

1. Navigate to **Operators > Installed Operators**
2. Select the Red Hat build of Crioostat operator.
3. Select the **Subscription** tab.
4. From the **Actions** drop-down menu, select **Edit Subscription**.
5. Add the **INSIGHTS\_ENABLED** environment variable to the **Subscription** object:

```
apiVersion: operators.coreos.com/v1alpha1
kind: Subscription
metadata:
  name: cryostat-operator
  namespace: openshift-operators
spec:
  config:
    env:
      - name: INSIGHTS_ENABLED
        value: "false"
  channel: stable
  installPlanApproval: Automatic
  name: cryostat-operator
```

source: my-operator-catalog  
sourceNamespace: openshift-marketplace  
startingCSV: cryostat-operator.v2.4.0

6. Click **Save**.

This workaround disables Red Hat Insights integration in the Cryostat Operator, which removes the **insights-proxy** deployment.

## CHAPTER 6. ADVISORIES RELATED TO THIS RELEASE

The following advisory has been issued to document bug fixes and CVE fixes included in the Cryostat 2.4 release:

- [RHSA-2023:7669](#)

*Revised on 2023-12-06 21:34:38 UTC*