



Red Hat build of Cryostat 3

Using automated rules on Cryostat

Legal Notice

Copyright © 2024 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 Red Hat build of Crio is a Red Hat offering on OpenShift Container Platform. The Using automated rules on Crio document is for users that want to use the automated rules feature to enable JFR to continuously monitor a running target application. Additionally, this document describes continuous monitoring event templates that you can use to create automated rules and templates.

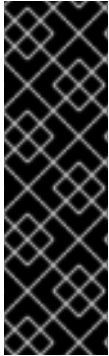
Table of Contents

PREFACE	3
MAKING OPEN SOURCE MORE INCLUSIVE	4
CHAPTER 1. OVERVIEW OF AUTOMATED RULES	5
CHAPTER 2. CREATING DEFINITIONS	6
2.1. ENABLING OR DISABLING EXISTING AUTOMATED RULES	6
2.2. CREATING AN AUTOMATED RULE DEFINITION	7
2.3. CRYOSTAT MATCH EXPRESSION VISUALIZER PANEL	9
2.4. UPLOADING AN AUTOMATED RULE IN JSON	9
2.5. METADATA LABELS	10
CHAPTER 3. ADDITIONAL AUTOMATED RULE FUNCTIONS	12
3.1. COPYING JFR DATA	12
3.2. DELETING AN AUTOMATED RULE	12

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 3.0 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. OVERVIEW OF AUTOMATED RULES

You can use automated rules to enable JFR to continuously monitor a running target application. You do not need to restart or redeploy the application.

Continuous monitoring event templates exist in Cryostat that you can use to create automated rules and templates. By using continuous monitoring event templates, you can reduce any downtime for specifying a JFR to continuously monitoring an application.

You can define automated rules regardless of whether you configure your target applications to use a Java Management Extensions (JMX) connection or an agent HTTP API connection. For more information about configuring your target applications, see [Configuring Java applications](#).

Consider the following guidelines:

- If your target JVMs use an agent HTTP API connection, ensure that you set the **cryostat.agent.api.writes-enabled** property to **true** in your target application's configuration. Otherwise, the Cryostat agent cannot accept on-demand requests to start, stop, and delete JFR recordings based on automated rules.
- If your target JVMs use a JMX connection, before you create an automated rule that applies to multiple target JVMs that each require JMX credentials, consider storing credentials for each JVM on the Cryostat web console. Storing credentials ensures that your automated rule starts, because Cryostat maintains a connection with each target JVM. For more information, see [Storing and managing JMX credentials \(Using Cryostat to manage a JFR recording\)](#) . When your target JVMs use a JMX connection, Cryostat stores JMX credentials in a **keyring** database. In this database, JMX credentials are encrypted by a user-provided passphrase that Cryostat supplies with the **PG_ENCRYPT_KEY** environment variable.



NOTE

From Cryostat 3.0 onward, the passphrase is supplied through the **PG_ENCRYPT_KEY** environment variable in the Cryostat DB container. In previous releases, the passphrase was supplied through the **CRYOSTAT_JMX_CREDENTIALS_DB_PASSWORD** environment variable in the Cryostat container.

When installing a Cryostat instance, you can specify an encryption key manually by using the **spec.databaseOptions.secretName** property in the Cryostat custom resource (CR). Otherwise, if you leave the **spec.databaseOptions.secretName** property blank, the Cryostat Operator generates a key automatically.

CHAPTER 2. CREATING DEFINITIONS

When creating an automated rule definition, you can configure numerous options. Cryostat uses an automated rule to apply rules to any JVM targets that match regular expressions defined in the **matchExpression** string expression. You can apply Red Hat OpenShift labels or annotations as criteria for a **matchExpression** definition.

After you specify a rule definition for your automated rule, you do not need to re-add or restart matching targets. If you have defined matching targets, you can immediately activate a rule definition.

If you want to reuse an existing automated rule definition, you can upload your definition in JSON format to Cryostat.



NOTE

From Cryostat 3.0 onward, you must use [Common Expression Language \(CEL\)](#) syntax when defining match expressions in automated rules. In previous releases, you could use JavaScript syntax to define match expressions.

2.1. ENABLING OR DISABLING EXISTING AUTOMATED RULES

You can enable or disable existing automated rules by using a toggle switch on the Cryostat web console.

Prerequisites

- Logged in to the Cryostat web console.
- Created an automated rule.

Procedure

1. From the Cryostat web console, click **Automated Rules**. The **Automated Rules** window opens and displays your automated rule in a table.

Figure 2.1. Example of match expression output from completing an automated rule

Ena...	Name	Descri...	Match Expression	Event Specifier	Archiv...	Initial...	Preser...	Maxim...	Max...
<input checked="" type="checkbox"/>	testing!		target.alias == 'agent-test' ('PORT' in target.annotations.cryostat && target.annotations.cryostat.PORT == 9977)	template=Profiling.type=TARGET	0	0	0	0	0

2. In the **Enabled** column, view the **Enabled** status of the listed automated rules. Depending on the status, choose one of the following actions:
 - To enable the automated rule, click the toggle switch to **On**. Cryostat immediately evaluates each application that you defined in the automated rule against its match expression. If a match expression applies to an application, Cryostat starts a JFR recording that monitors the performance of the application.

- To disable the automated rule, click the toggle switch to **Off**. The **Disable your Automated Rule** window opens. To disable the selected automated rule, click **Disable**. If you want to also stop any active recordings that were created by the selected rule, select **Clean** then click **Disable**.

2.2. CREATING AN AUTOMATED RULE DEFINITION

While creating an automated rule on the Cryostat web console, you can specify the match expression that Cryostat uses to select all the applications. Then, Cryostat starts a new recording by using a JFR event template that was defined by the rule.

If you previously created an automated rule and Cryostat identifies a new target application, Cryostat tests if the new application instance matches the expression and starts a new recording by using the associated event template.

Prerequisites

- Created a Cryostat instance in your Red Hat OpenShift project.
- Created a Java application.
- Installed Cryostat 3.0 on Red Hat OpenShift by using the OperatorHub option.
- Logged in to your Cryostat web console.

Procedure

1. In the navigation menu on the Cryostat web console, click **Automated Rules**. The **Automated Rules** window opens.
2. Click **Create**. A **Create** window opens.

Figure 2.2. The Create window (Graph View) for an automated rule

The screenshot shows the 'Create' window for an automated rule in the Cryostat web console. The window is divided into two main panels. The left panel contains the following fields and controls:

- Enter a rule name:** A text input field.
- Description:** A text area with a placeholder: "Enter a rule description. This is only used for display purposes to aid in identifying rules and their intentions."
- Match Expression:** A text area containing the expression: `target.alias == 'agent-test' || ('PORT' in target.annotations.cryostat && target.annotations.cryostat.PORT == 9977)`. A green checkmark is visible to the right of the text.
- Enabled:** A toggle switch that is currently turned on (blue).
- Template:** A dropdown menu with the text "Select a Template".
- Maximum size:** A text input field with the value "0" and a "B" dropdown.

The right panel is titled "Match Expression visualizer" and includes a "Visualize via:" section with radio buttons for "Graph view" (selected) and "List view". Below this is a graph visualization showing two nodes: "agent-test" and "quarkus-te...d6d-9h4gw". The graph area has a search icon, a zoom in icon, a zoom out icon, and a refresh icon at the bottom.

3. Enter a rule name in the **Name** field.
4. In the **Match Expression** field, specify the match expression details.



NOTE

Select the question mark icon to view suggested syntax in a **Match Expression Hint** snippet.

In the **Match Expression Visualizer** panel, the **Graph View** option highlights the target JVMs that are matched. Unmatched target JVMs are greyed out.

5. *Optional:* In the **Match Expression Visualizer** panel, you can also click **List View**, which displays the matched target JVMs as expandable rows.

Figure 2.3. The Create window (List View) for an automated rule

6. From the **Template** list, select an event template.
7. To create your automated rule, click **Create**. The **Automated Rules** window opens and displays your automated rule in a table.

Figure 2.4. Example of match expression output from completing an automated rule

If a match expression applies to an application, Crystat starts a JFR recording that monitors the performance of the application.

8. *Optional:* You can download an automated rule by clicking **Download** from the automated rule's overflow menu. You can then configure a rule definition in your preferred text editor or make additional copies of the file on your local file system.

2.3. CRYOSTAT MATCH EXPRESSION VISUALIZER PANEL

You can use the **Match Expression Visualizer** panel on the web console to view information in a JSON structure for your selected target JVM application. You can choose to display the information in a **Graph View** or a **List View** mode. The **Graph View** highlights the target JVMs that are matched. Unmatched target JVMs are greyed out. The **List View** displays the matched target JVM as expandable rows.

To view details about a matched target JVM, select the target JVM that is highlighted. In the window that appears, information specific to the metadata for your application is shown in the **Details** tab. You can use any of this information as syntax in your match expression. A match expression is a rule definition parameter that you can specify for your automated rule.

After you specify match expressions and created the automated rule, Cryostat immediately evaluates each application that you defined in the automated rule against its match expression. If a match expression applies to an application, Cryostat starts a JFR recording that monitors the performance of the application.

2.4. UPLOADING AN AUTOMATED RULE IN JSON

You can reuse an existing automated rule by uploading it to the Cryostat web console, so that you can quickly start monitoring a running Java application.

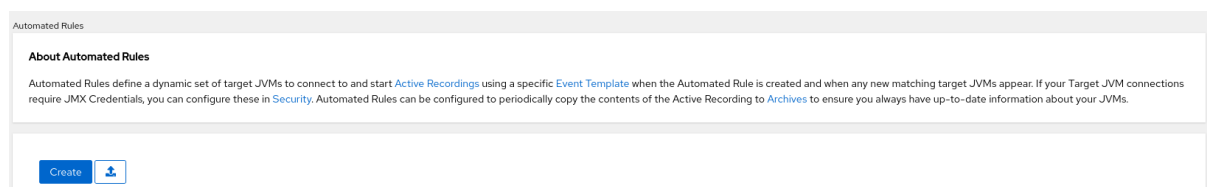
Prerequisites

- Created a Cryostat instance in your project. See [Installing Cryostat on OpenShift using an operator](#) (Installing Cryostat).
- Created a Java application.
- Created an automated rules file in JSON format.
- Logged in to your Cryostat web console.

Procedure

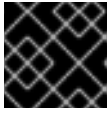
1. In the navigation menu on the Cryostat web console, click **Automated Rules**. The **Automated Rules** window opens.
2. Click the file upload icon, which is located beside the **Create** button.

Figure 2.5. The automated rules upload button



The **Upload Automated Rules** window opens.

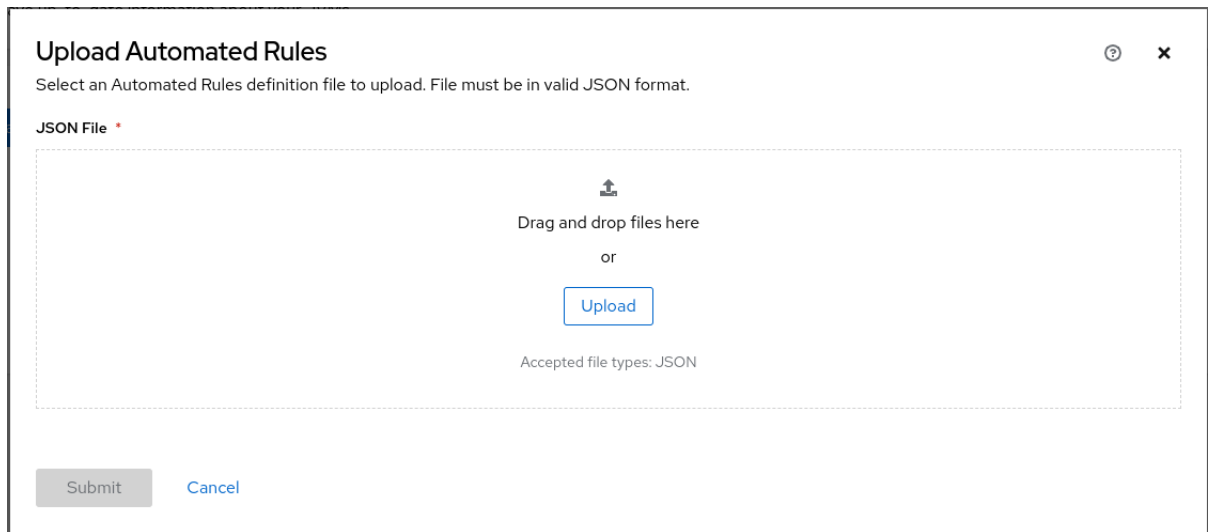
3. Click **Upload** and locate your automated rules files on your local system. You can upload one or more files to Cryostat. Alternatively, you can drag files from your file explorer tool and drop them into the **JSON File** field on your web console.



IMPORTANT

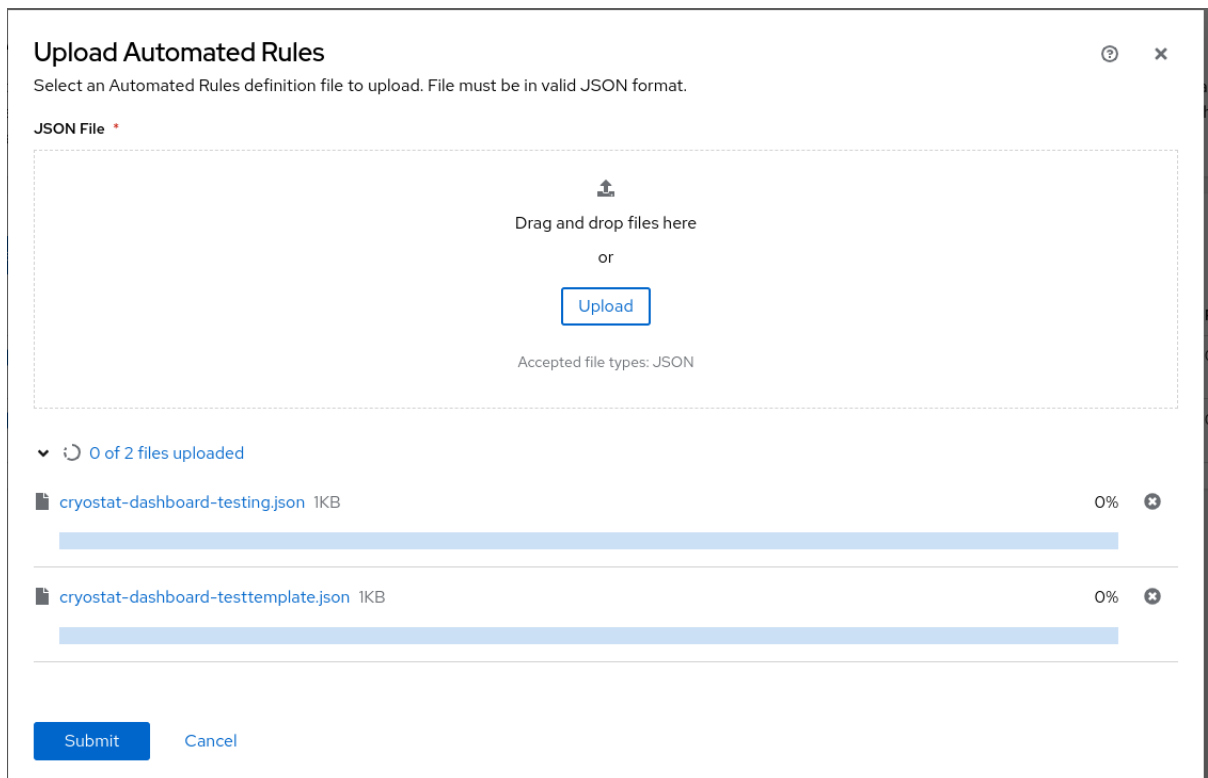
The **Upload Automated Rules** function only accepts files in JSON format.

Figure 2.6. A window prompt where you can upload JSON files that contains your automated rules configuration



4. *Optional:* If you need to remove a file from the **Upload Automated Rules** function, click the **X** icon on the selected file.

Figure 2.7. Example of uploaded JSON files



5. Click **Submit**.

2.5. METADATA LABELS

When you create an automated rule to enable JFR to continuously monitor a running target application, the automated rule automatically generates a metadata label. This metadata label indicates the name of

the automated rule that generates the JFR recording. After you archive the recording, you can run a query on the metadata label to locate the automated rule that generated the recording.

Cryostat preserves metadata labels for the automated rule in line with the lifetime of the archived recording.

Additional resources

- [Creating definitions](#)
- [Archiving JDK Flight Recorder \(JFR\) recordings \(Using Cryostat to manage a JFR recording\)](#)

CHAPTER 3. ADDITIONAL AUTOMATED RULE FUNCTIONS

From the Cystostat web console, you access additional automated rule capabilities, such as deleting an automated rule or copying JFR.

If you created Cystostat 2.4, and then upgraded from Cystostat 2.4 to Cystostat 3.0, Cystostat 3.0 automatically detects these automated rules.

3.1. COPYING JFR DATA

You can copy information from a JVM application's memory to Cystostat's archive storage location on the OpenShift Container Platform (OCP).

During the creation of an automated rule through the Cystostat web console, you can set a value in the **Archival Period** field. You can specify a numerical value in seconds, minutes, or hours. After you create the automated rule with a specified archival period, Cystostat re-connects with any targeted JVM applications that match the rule. Cystostat then copies any generated JFR recording data from the application's memory to Cystostat's archive storage location.

Additionally, you can populate the **Preserved Archives** field with a value. This field sets a limit to the amount of copies of a JFR recording that Cystostat can move from an application's memory to Cystostat's archive storage location. For example, if you set a value of **10** in the **Preserved Archives** field, Cystostat will not store more than 10 copies of the file in the archive storage location. When Cystostat generates a new copy of the file that exceeds the limit, Cystostat replaces the oldest version with the newest version of the file.

You can also set a size limit for a JFR recording file and specify a time limit for how long a file is stored in the target JVM application's memory by specifying values for the **Maximum Size** and **Maximum Age** parameters.

Prerequisites

- Created a Cystostat instance in your Red Hat OpenShift project.
- Created a Java application.
- Logged in to your Cystostat web console.

Procedure

1. In the navigation menu on the Cystostat web console, click **Automated Rules**. The **Automated Rules** window opens.
2. Click **Create**. The **Create** window opens.
3. Enter values in any mandatory fields, such as the **Match Expression** field.
4. In the **Archival Period** field, specify a value in seconds, minutes, or hours.
5. In the **Preserved Archives** field, enter the number of archived recording copies to preserve.
6. To create the automated rule, click **Create**. The **Automated Rules** window opens and displays your automated rule in a table.

3.2. DELETING AN AUTOMATED RULE

The Cryostat web console that runs on the OpenShift Container Platform (OCP) provides a simplified method for deleting a rule definition.

You can also use the **curl** tool to delete an automated rule. The **curl** tool communicates with your Cryostat instance by using the **DELETE** endpoint. In the request, you can specify the **clean=true** parameter, which stops all active Java Flight Recordings (JFRs) started by the selected rule.

Prerequisites

- Logged in to the Cryostat web console.
- Created an automated rule.

Procedure

1. In the navigation menu on the Cryostat web console, click **Automated Rules**. The **Automated Rules** window opens and displays all existing automated rules in a table.

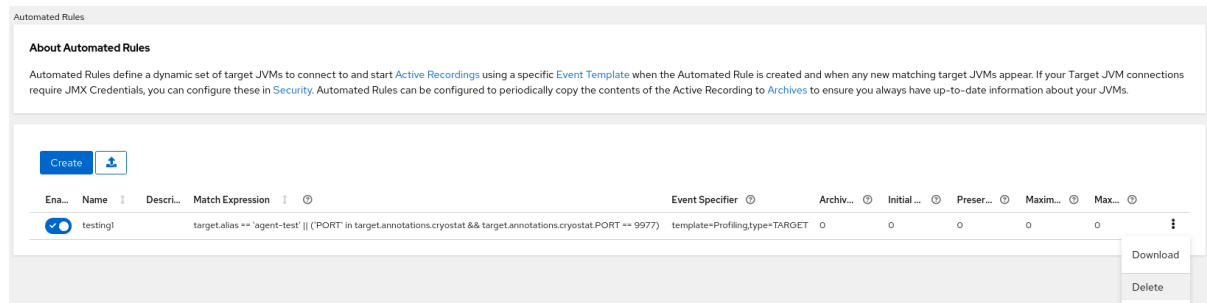


NOTE

If you have not created an automated rule, only a **Create** button appears on the **Automated Rules** window.

2. In the table, select the automated rule that you want to delete.
3. Click the more options icon (**⋮**), then click **Delete**.

Figure 3.1. Delete option from the Automated Rules table



The **Permanently delete your Automated Rule** window opens.

1. To delete the selected automated rule, click **Delete**. If you want to also stop any active recordings that were created by the selected rule, select **Clean** then click **Delete**.

Cryostat deletes your automated rule permanently.

Revised on 2024-07-02 13:35:53 UTC