Red Hat build of Cryostat 3

Using Cryostat to manage a JFR recording

Last Updated: 2024-07-02

Legal Notice

Abstract

Learn how you can use Red Hat build of Cryostat to manage JDK Flight Recordings (JFRs) that monitor the performance of Java Virtual Machines (JVMs) that are located in containerized applications. You can use Cryostat 3.0 to start, stop, retrieve, archive, import, and export JFR data for JVMs.

Table of Contents

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. SECURITY OPTIONS

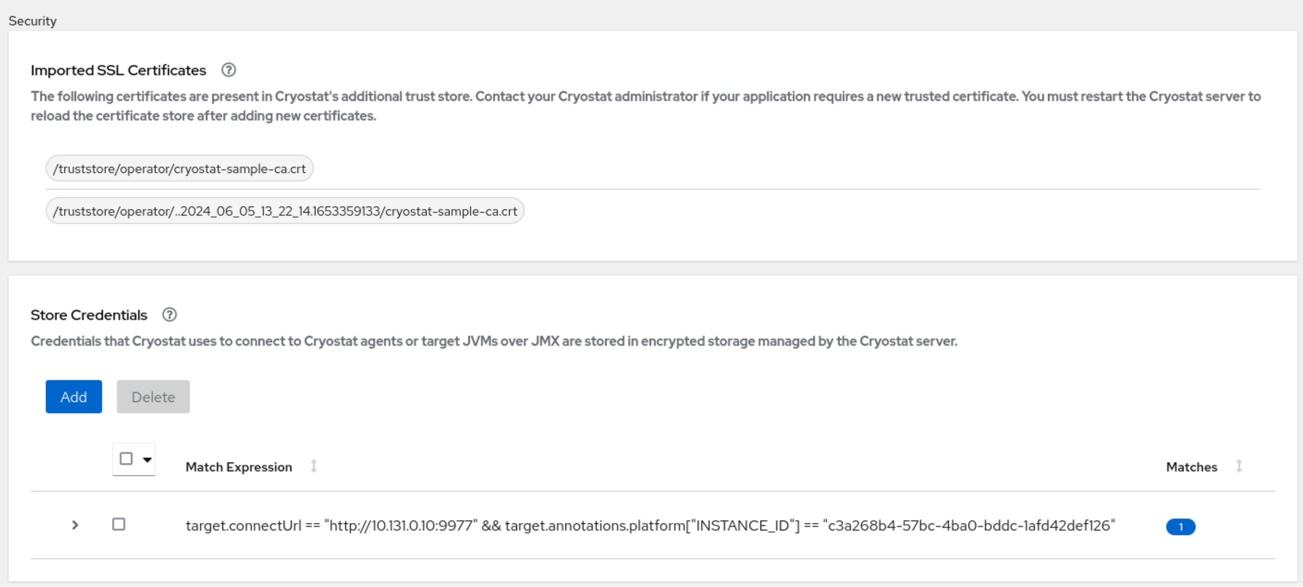
You can configure security settings for Cryostat, so that you can better protect your Cryostat instance.

Cryostat can encrypt and store credentials for a target JVM application in a database that is stored on a persistent volume claim (PVC) on Red Hat OpenShift. Cryostat supports SSL/TLS on the HTTP request that adds credentials to the database and on the JMX connection that uses those credentials to connect to the target application. Cryostat also encrypts the credentials within the database by using a passphrase that is either provided by the user or that is generated by the Red Hat build of Cryostat Operator.

You can use the Cryostat Operator to configure Cryostat to trust SSL/TLS certificates from specific applications by adding these certificates to a secret and by configuring the Cryostat custom resource (CR) to include this secret. For more information, see Using the Red Hat build of Cryostat Operator to configure Cryostat: Configuring TLS certificates.

You can view the list of imported SSL/TLS certificates for a target JVM by clicking the **Security** menu in the Cryostat web console.

Figure 1.1. Viewing the list of imported SSL certificates for a target JVM



Imported SSL Certificates

The following certificates are present in Cryostat's additional trust store. Contact your Cryostat administrator if your application requires a new trusted certificate. You must restart the Cryostat server to reload the certificate store after adding new certificates.

/truststore/operator/cryostat-sample-ca.crt

/truststore/operator/..2024_06_05_13_22_14.1653359133/cryostat-sample-ca.crt

Store Credentials

Credentials that Cryostat uses to connect to Cryostat agents or target JVMs over JMX are stored in encrypted storage managed by the Cryostat server.

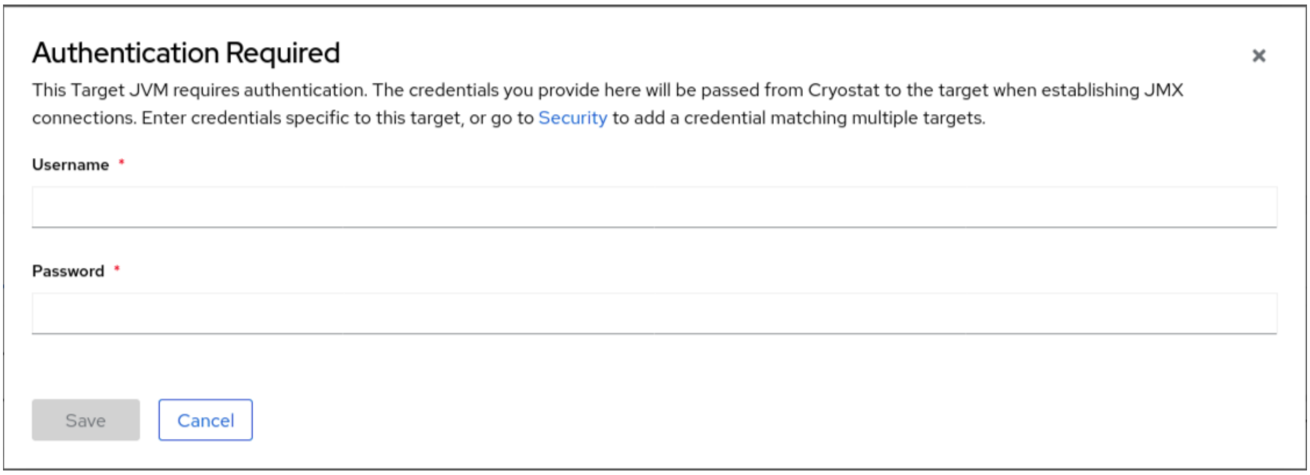
	☐	Match Expression	Matches
›	☐	target.connectUrl == "http://10.131.0.10:9977" && target.annotations.platform["INSTANCE_ID"] == "c3a268b4-57bc-4ba0-bddc-1afd42def126"	1

1.1. STORING AND MANAGING CREDENTIALS

If you enable Java Management Extensions (JMX) authentication or HTTP authentication for your target JVM application, Cryostat prompts you to enter your credentials before Cryostat can access any of the application’s JFR recordings.

When you click the **Recordings** or **Events** menu item on the Cryostat web console, an **Authentication Required** window opens on the console. You must enter the username and password of the target JVM application. You can then view the recordings or perform any additional recording operations on the application.

Figure 1.2. Example of a Cryostat Authentication Required window



Authentication Required

This Target JVM requires authentication. The credentials you provide here will be passed from Cryostat to the target when establishing JMX connections. Enter credentials specific to this target, or go to Security to add a credential matching multiple targets.

Username *

Password *

Cryostat stores credentials that it uses to connect to Cryostat agents or target JVMs.

IMPORTANT

If you need to restart your target JVM application, ensure that you complete one of the following tasks to avoid losing JFR recording data for the application:

- Click the **Recordings** menu item on the Cryostat web console and archive your JFR recording.
- Create an automated rule that schedules Cryostat to copy a snapshot recording to the storage location for the Cryostat archives.

When you want to monitor multiple target JVMs by creating an automated rule, you can configure Cryostat to store and then reuse your credentials for each target JVM connection. By using this configuration, you do not need to re-enter your credentials whenever you want to revisit the JFR recording for your application on the Cryostat web console.

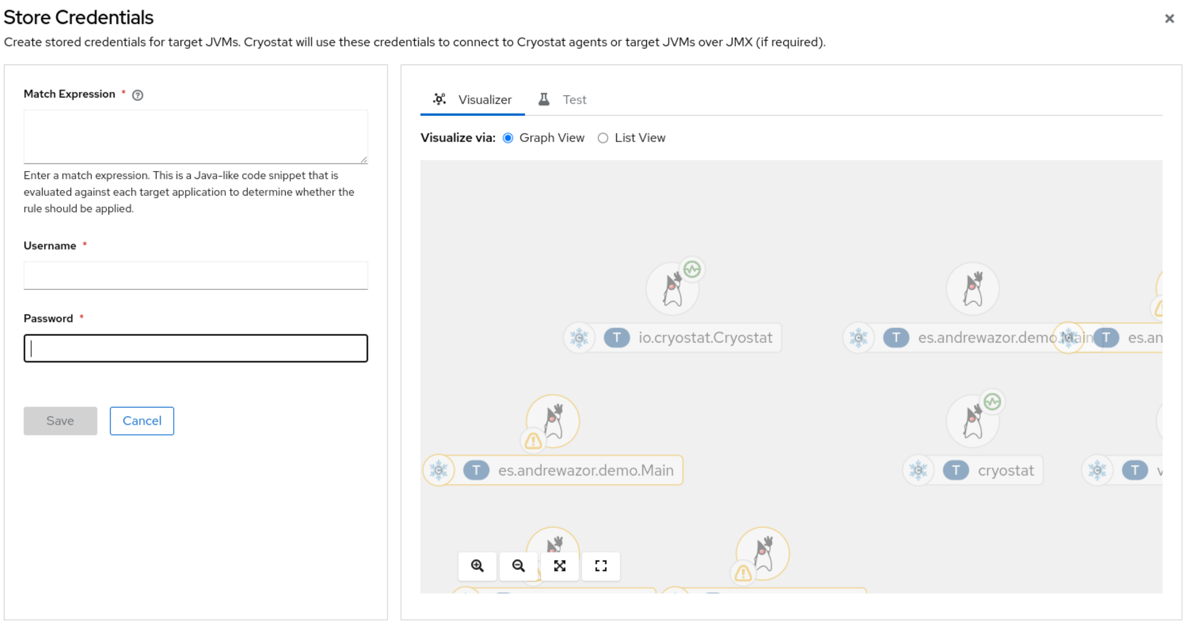
Prerequisites

- Enabled JMX or HTTP authentication for your target JVM application.

Procedure

1. Click the **Security** menu item.
2. From the **Store Credentials** window, click the **Add** button. The **Store Credentials** window opens.

Figure 1.3. Example of a Store Credentials window



Store Credentials

Create stored credentials for target JVMs. Cryostat will use these credentials to connect to Cryostat agents or target JVMs over JMX (if required).

Match Expression *

Enter a match expression. This is a Java-like code snippet that is evaluated against each target application to determine whether the rule should be applied.

Username *

Password *

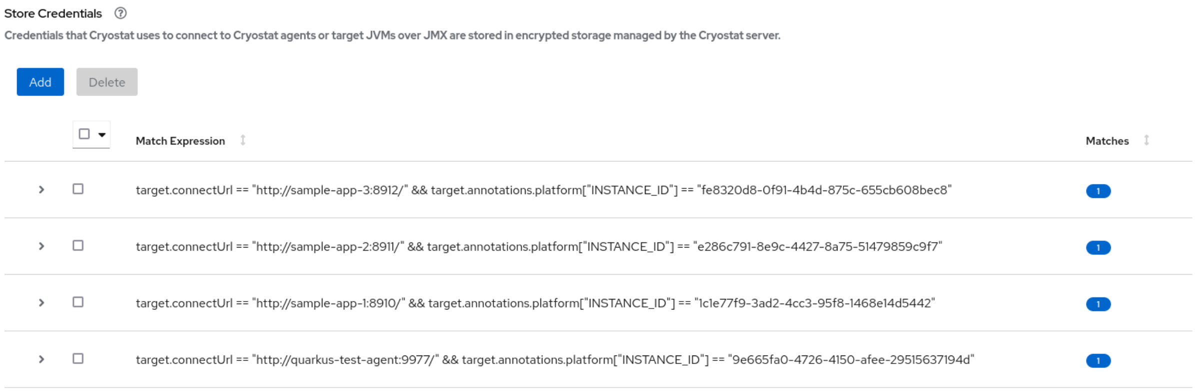
Visualize via: ● Graph View ○ List View

3. 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.

4. Click **Save**. A table entry is displayed in the **Store Credentials** window that shows the **Match Expression** for your target JVM.

Figure 1.4. Example of a table entry on the Store Credentials pane



Store Credentials

Credentials that Cryostat uses to connect to Cryostat agents or target JVMs over JMX are stored in encrypted storage managed by the Cryostat server.

	□	Match Expression	Matches
›	□	target.connectUrl == "http://sample-app-3:8912/" && target.annotations.platform["INSTANCE_ID"] == "fe8320d8-0f91-4b4d-875c-655cb608bec8"	1
›	□	target.connectUrl == "http://sample-app-2:8911/" && target.annotations.platform["INSTANCE_ID"] == "e286c791-8e9c-4427-8a75-51479859c9f7"	1
›	□	target.connectUrl == "http://sample-app-1:8910/" && target.annotations.platform["INSTANCE_ID"] == "1c1e77f9-3ad2-4cc3-95f8-1468e14d5442"	1
›	□	target.connectUrl == "http://quarkus-test-agent:9977/" && target.annotations.platform["INSTANCE_ID"] == "9e665fa0-4726-4150-afee-29515637194d"	1

> **IMPORTANT**
>
> For security purposes, a table entry does not display your username or password.

5. *Optional:* If you want to delete your stored credentials for a target JVM, you can select the checkbox next to the table entry for this target JVM and then click **Delete**.

CHAPTER 2. ARCHIVE JFR RECORDINGS

You can archive active JFR recordings to avoid potential data loss from JFR recordings. You can download or upload the archived JFR recording, so that you can analyze the recording to suits your needs.

You can find archived JFR recordings from the **Archives** menu in chronological order under one of three headings: **All Targets**, **All Archives**, and **Uploads**. Depending on what actions you performed on a JFR recording, the recording might display under each table.

2.1. ARCHIVING JDK FLIGHT RECORDER (JFR) RECORDINGS

You can archive active JFR recordings to avoid potential data loss from JFR recordings. Data loss might occur when Cryostat replaces legacy JFR recording data with new data to save storage space or when a target JVM abruptly stops or restarts.

When you create an archived recording, Cryostat copies the active JFR recording's data and stores the data in a persistent storage location on your Cryostat instance. The Red Hat build of Cryostat Operator builds this persistent storage location onto the associated persistent volume claim (PVC) on the Red Hat OpenShift cluster.

You can archive any JFR recording, regardless of its configuration. Additionally, you can archive snapshots from a JFR recording.

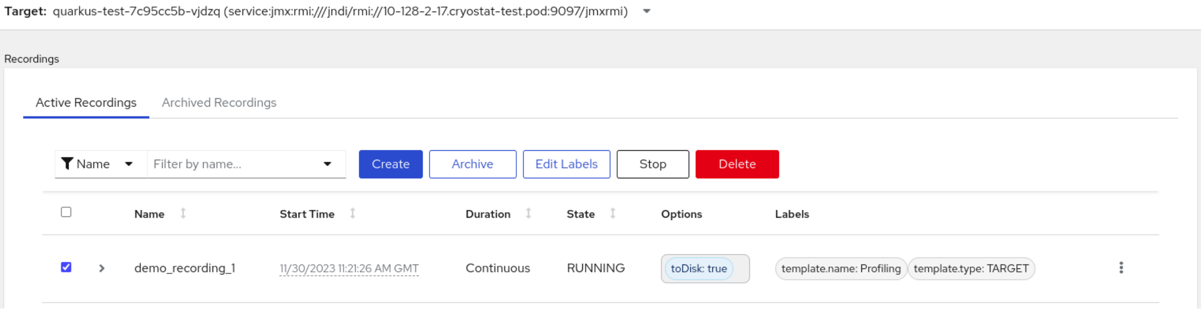
Prerequisites

- Entered your authentication details for your Cryostat instance.
- Created a target JVM recording and entered your authenticated details to access the **Recordings** menu. See Creating a JDK Flight Recorder (JFR) recording (Creating a JFR recording with Cryostat).

Procedure

1. On the **Active Recordings** tab, select the checkbox for your JFR recording. The **Archive** button is activated in the **Active Recordings** toolbar.

Figure 2.1. Archive button for your JFR recording



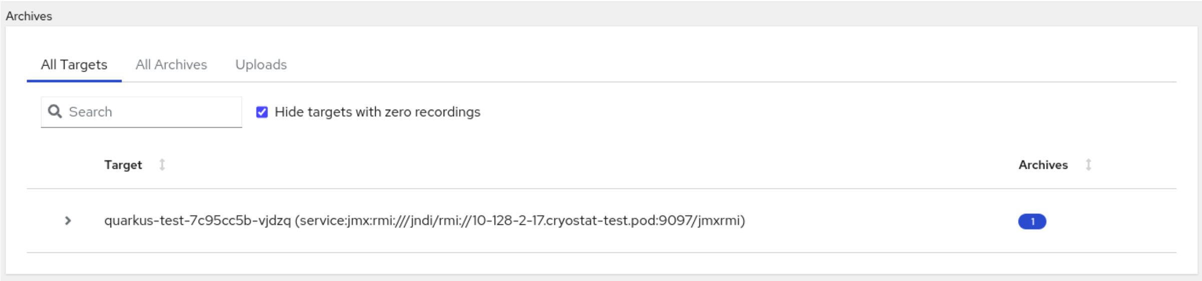
Target: quarkus-test-7c95cc5b-vjdzq (service:jmx:rmi:///jndi/rmi://10-128-2-17.cryostat-test.pod:9097/jmxrmi)

☐		Name	Start Time	Duration	State	Options	Labels
☑	›	demo_recording_1	11/30/2023 11:21:26 AM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling template.type: TARGET

2. Click the **Archive** button. Cryostat creates an archived recording of your JFR recording. You can view your archived recording from under the **Archived Recordings** tab along with any other recording that relates to your selected target JVM.

   Alternatively, you can view your archived recording from under the **All Targets** table.

Figure 2.2. Example of a listed target JVM application that is under the All Targets table



Archives

All Targets | All Archives | Uploads

Search ☑ Hide targets with zero recordings

	Target	Archives
›	quarkus-test-7c95cc5b-vjdzq (service:jmx:rmi:///jndi/rmi://10-128-2-17.cryostat-test.pod:9097/jmxrmi)	1

TIP

To remove a target JVM entry that does not have an archived recording, select the **Hide targets with zero recordings** checkbox.

After you click on the twistie (**v**) beside the JVM target entry, you can access a filter function, where you can edit labels to enhance your filter or click the **Delete** button to remove the filter.

3. From the **All Targets** table, select the checkbox beside each target JVM application that you want to review. The table lists each archived recording and its source location.

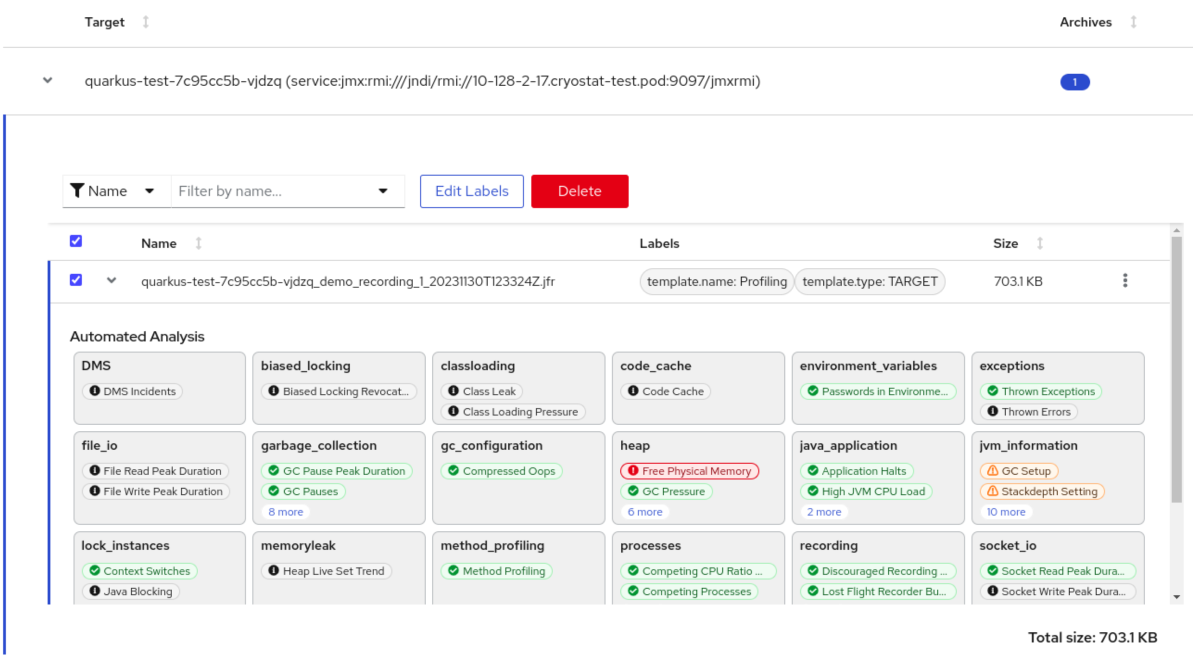
4. Go to the **All Archives** table. This table looks similar to the **All Targets** table, but the **All Archives** table lists target JVM applications from files that Cryostat archived inside Cryostat.

NOTE

If an archived file has no recognizable JVM applications, it is still listed on the **All Archives** table but opens within a nested table under the heading **lost**.

5. *Optional:* To delete an archived recording, select the checkbox next to the specific archived JFR recording item, and click **Delete** when prompted.

Figure 2.3. Deleting an archived JFR recording



	Target	Archives
⌄	quarkus-test-7c95cc5b-vjdzq (service:jmx:rmi:///jndi/rmi://10-128-2-17.cryostat-test.pod:9097/jmxrmi)	1

☑		Name	Labels	Size
☑	⌄	quarkus-test-7c95cc5b-vjdzq_demo_recording_1_20231130T123324Z.jfr	template.name: Profiling template.type: TARGET	703.1 KB

Automated Analysis

DMS: DMS Incidents

biased_locking: Biased Locking Revocat...

classloading: Class Leak; Class Loading Pressure

code_cache: Code Cache

environment_variables: Passwords in Environme...

exceptions: Thrown Exceptions; Thrown Errors

file_io: File Read Peak Duration; File Write Peak Duration

garbage_collection: GC Pause Peak Duration; GC Pauses; 8 more

gc_configuration: Compressed Oops

heap: Free Physical Memory; GC Pressure; 6 more

java_application: Application Halts; High JVM CPU Load; 2 more

jvm_information: GC Setup; Stackdepth Setting; 10 more

lock_instances: Context Switches; Java Blocking

memoryleak: Heap Live Set Trend

method_profiling: Method Profiling

processes: Competing CPU Ratio ...; Competing Processes

recording: Discouraged Recording ...; Lost Flight Recorder Bu...

socket_io: Socket Read Peak Dura...; Socket Write Peak Dura...

Total size: 703.1 KB

NOTE

Cryostat assigns names to archived recordings based on the address of the target JVM's application, the name of the active recording, and the timestamp of the created archived recordings.

Additional resources

- See Persistent storage using local volumes (Red Hat OpenShift)

2.2. DOWNLOADING AN ACTIVE RECORDING OR AN ARCHIVED RECORDING

You can use Cryostat to download an active recording or an archived recording to your local system.

Prerequisites

- Entered your authentication details for your Cryostat instance.
- Created a JFR recording. See Creating a JDK Flight Recorder (JFR) recording (Creating a JFR recording with Cryostat).
- *Optional:* Uploaded an SSL certificate or provided your credentials to the target JVM.
- *Optional:* Archived your JFR recording. See Archiving JDK Flight Recorder (JFR) recordings (Using Cryostat to manage a JFR recording).

Procedure

1. Navigate to the **Recordings** menu or the **Archives** menu on your Cryostat instance.

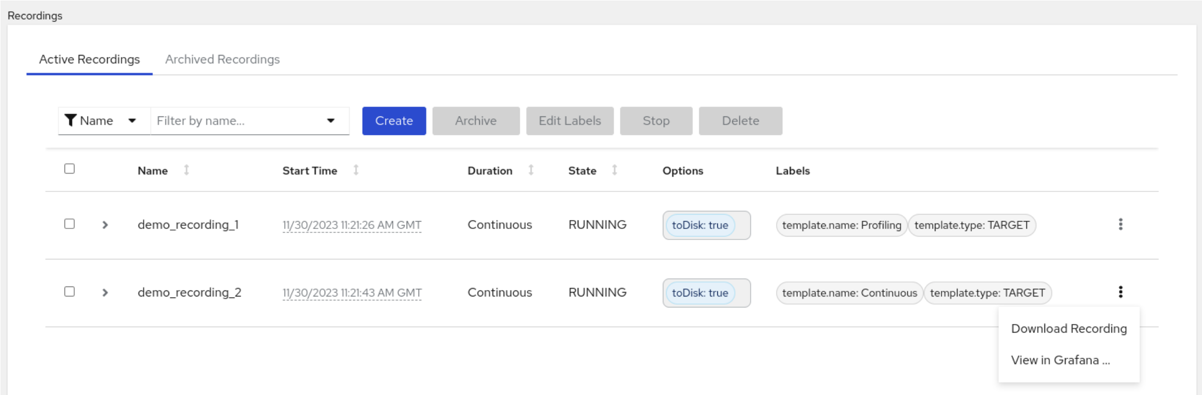
NOTE

The remaining steps use the **Recordings** menu as an example, but you can follow similar steps on the **Archives** menu.

2. Determine the recording you want by clicking either the **Active Recordings** tab or the **Archived Recordings** tab.

3. Locate your listed JFR recording and then select its overflow menu.

Figure 2.4. Viewing a JFR recording's overflow menu



Name	Start Time	Duration	State	Options	Labels
demo_recording_1	11/30/2023 11:21:26 AM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling template.type: TARGET
demo_recording_2	11/30/2023 11:21:43 AM GMT	Continuous	RUNNING	toDisk: true	template.name: Continuous template.type: TARGET

4. Choose one of the following options:

    a. From the overflow menu, click **Download Recording**. Depending on how you configured your operating system, a file-save dialog opens. Save the JFR binary file and the JSON file to your preferred location.

    b. From the **All Targets** table, select the overflow menu for your listed JFR recordings. Click **Download** to save the archived file along with its JSON file, which contains metadata and label information, to your local system.

5. *Optional:* View the downloaded file with the Java Mission Control (JMC) desktop application.

> NOTE
>
> If you do not want to download the **.jfr** file, but instead want to view the data from your recording on the Cryostat application, you can click the **View in Grafana** option.

2.3. UPLOADING A JFR RECORDING TO THE CRYOSTAT ARCHIVES LOCATION

You can upload a JFR recording from your local system to the archives location of your Cryostat.

To save Cryostat storage space, you might have scaled down or removed your JFR recording. If you downloaded a JFR recording, you can upload it to your Cryostat instance when you scale up or redeploy the instance.

Additionally, you can upload a file from a previous Cryostat instance to a new Cryostat instance. Cryostat analysis tools work on the recording uploaded to the new Cryostat instance.

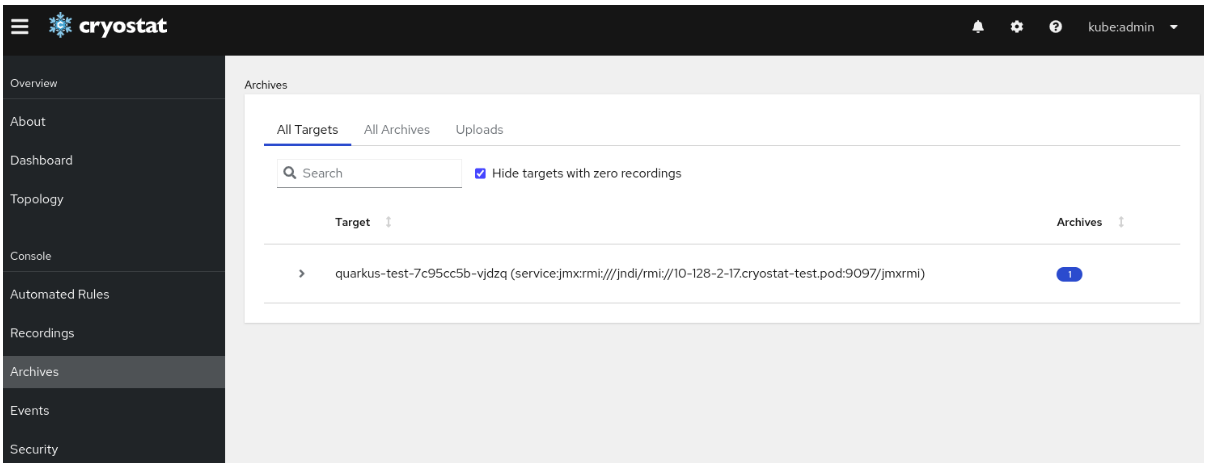
Prerequisites

- Entered your authentication details for your Cryostat instance.
- Created a JFR recording. See Creating a JDK Flight Recorder (JFR) recording (Creating a JFR recording with Cryostat).
- See Downloading an active recording or an archived recordings (Using Cryostat to manage a JFR recording).

Procedure

1. Go to the **Archives** menu on your Cryostat instance.

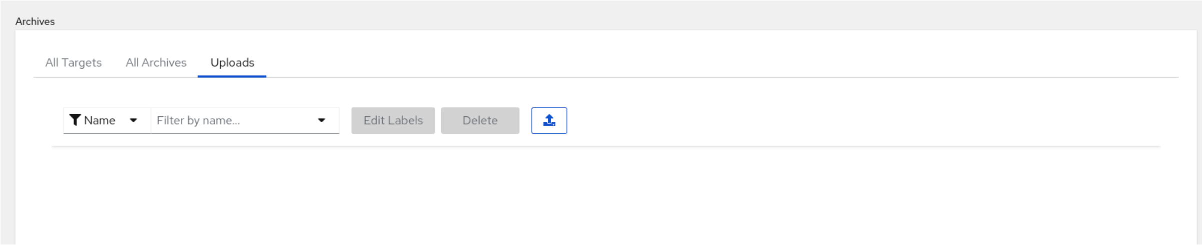
Figure 2.5. Archives menu on the Cryostat web console



	Target	Archives
›	quarkus-test-7c95cc5b-vjdzq (service:jmx:rmi:///jndi/rmi://10-128-2-17.cryostat-test.pod:9097/jmxrmi)	1

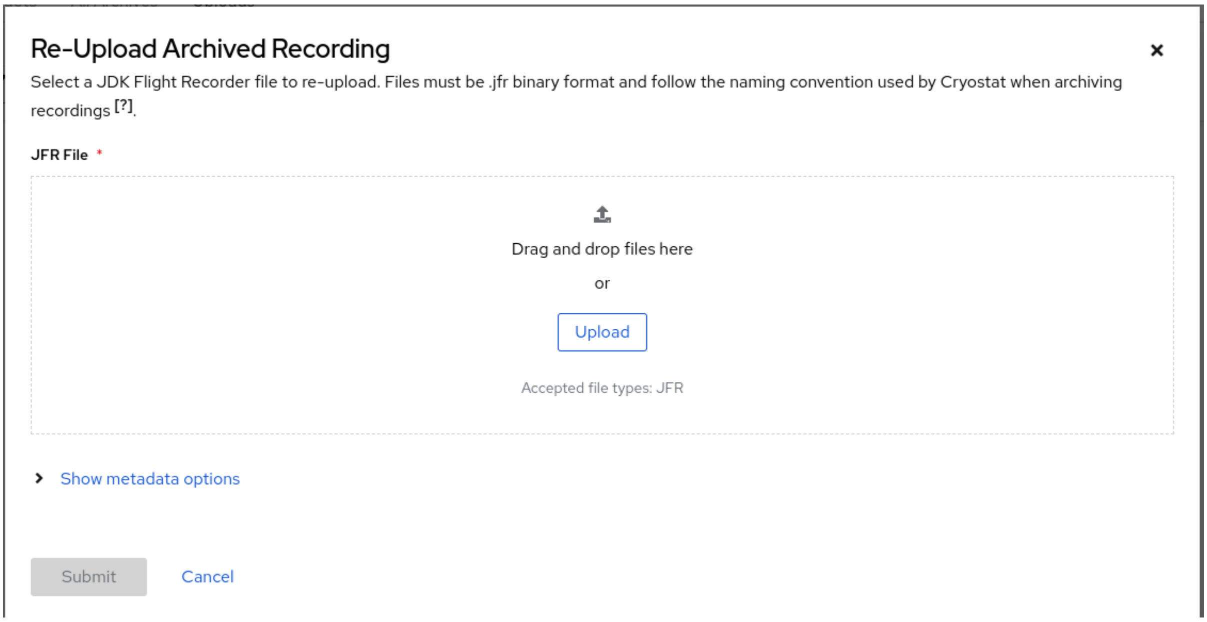
2. *Optional:* From the **Uploads** table, you can view all of your uploaded JFR recordings. The **Uploads** table also includes a filtering mechanism similar to other tables, such as the **All Targets** table, and other output. You can also use the filtering mechanism on the **Archives** menu to find an archived file that might have no recognizable target JVM application.

Figure 2.6. The Uploads table in the Archives menu



3. Click the upload icon. A **Re-Upload Archived Recording** window opens in your Cryostat web console:

Figure 2.7. Re-Upload Archived Recording window



Re-Upload Archived Recording

Select a JDK Flight Recorder file to re-upload. Files must be .jfr binary format and follow the naming convention used by Cryostat when archiving recordings [?].

4. In the **JFR File** field, click **Upload**.

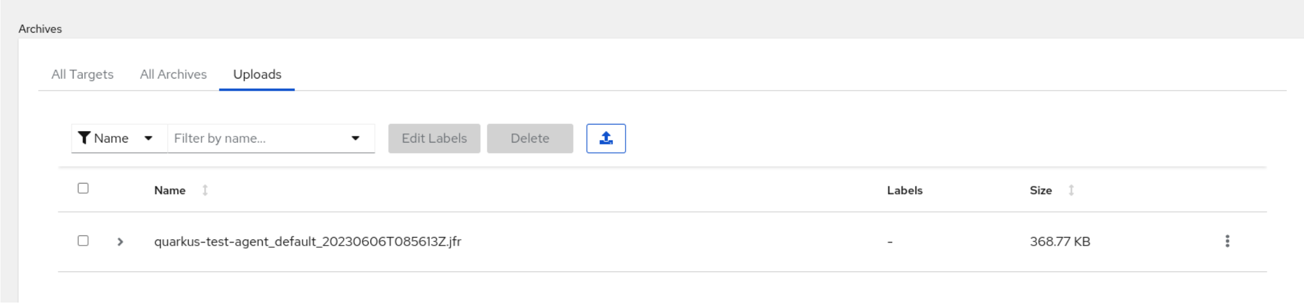
5. Locate the JFR recording files, which are files with a **.jfr** extension, and then click **Submit**.

NOTE

Alternatively, you can drag and drop **.jfr** files into the **JFR File** field.

Your JFR recording files open in the **Uploads** table.

Figure 2.8. Example of a JFR recording that is in the Uploads table



□		Name	Labels	Size
□	›	quarkus-test-agent_default_20230606T085613Z.jfr	-	368.77 KB

CHAPTER 3. EVENT TEMPLATES

Cryostat includes default event templates that you can use to quickly create a JFR recording for monitoring your target JVM's performance.

3.1. USING CUSTOM EVENT TEMPLATES

You can choose either one of the following default event templates when creating a JDK Flight Recorder (JFR) recording:

- Continuous template, which collects basic target Java Virtual Machine (JVM) data for either a fixed duration or until it is explicitly stopped.
- Profiling template, which collects in-depth target JVM data for either a fixed duration or until it is explicitly stopped.

By using either of these default event templates, you can quickly create a JFR recording for monitoring your target JVM's performance. You can edit either event template at a later stage to suit your needs. For example, the default event templates do not contain application-specific custom events, so you must add these custom events to the custom template.

Cryostat also supports the **ALL** meta-template, which enables a JFR to monitor all event types for a target JVM. Default values exist for each event type. The **ALL** meta-template does not contain an XML definition, so you cannot download an XML file for the **ALL** meta-template.

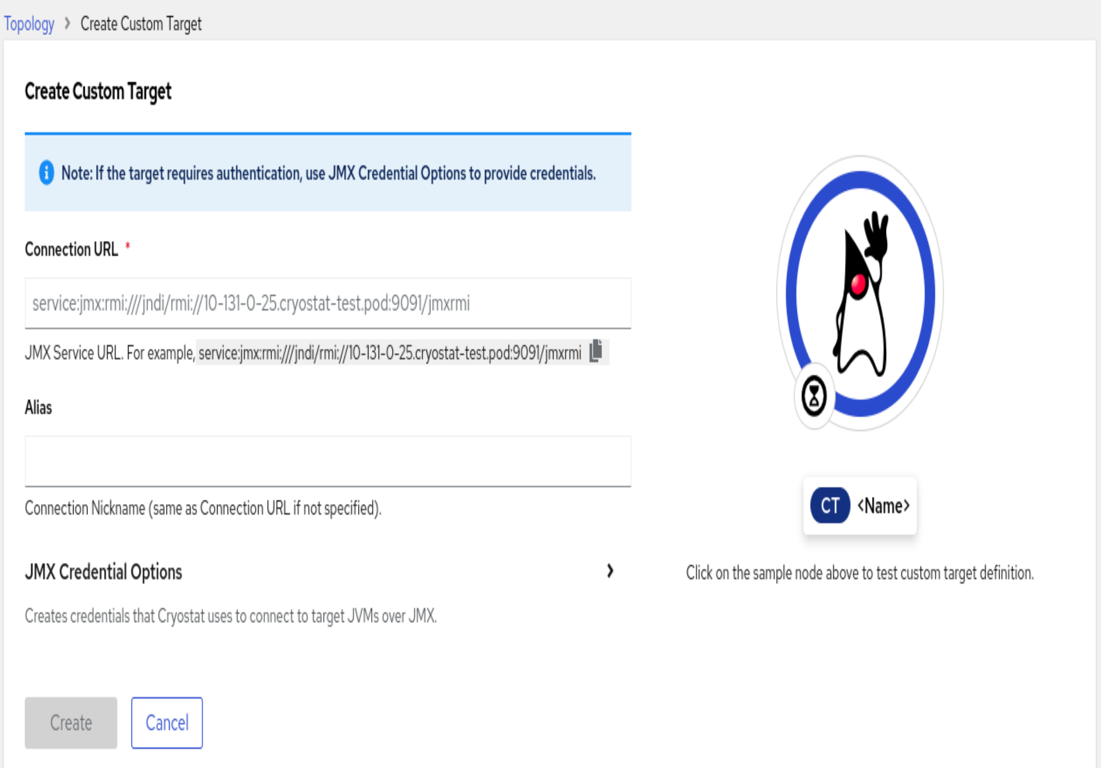
Prerequisites

- Installed Cryostat 3.0 on Red Hat OpenShift by using the **Installed Operators** option.
- Created a Cryostat instance in your Red Hat OpenShift project.

Procedure

1. On the **Dashboard** panel for your Cryostat instance, select a **Target JVM** from the drop-down list.
2. *Optional:* On the **Topology** panel, you can define a target JVM by selecting the **Add to view** icon. After you select the icon, a window opens for defining a custom target connection URL.
   a. In the **Connection URL** field, enter the URL for your JVM's Java Management Extension (JMX) endpoint.
   b. *Optional:* In the **Alias** field, enter an alias for your JMX Service URL.
   c. Click **Create**.

Figure 3.1. Create Target dialog box



Topology › Create Custom Target

Create Custom Target

Note: If the target requires authentication, use JMX Credential Options to provide credentials.

Connection URL *

service:jmx:rmi:///jndi/rmi://10-131-0-25.cryostat-test.pod:9091/jmxrmi

JMX Service URL. For example, service:jmx:rmi:///jndi/rmi://10-131-0-25.cryostat-test.pod:9091/jmxrmi

Alias

Connection Nickname (same as Connection URL if not specified).

JMX Credential Options ›

Creates credentials that Cryostat uses to connect to target JVMs over JMX.

Create Cancel

Click on the sample node above to test custom target definition.

3. From the navigation menu on the Cryostat web console, click **Events**. An **Authentication Required** dialog might open on your web console. If prompted, enter your **Username** and **Password** in the **Authentication Required** dialog box, and click **Save** to provide your credentials to the target JVM.

NOTE

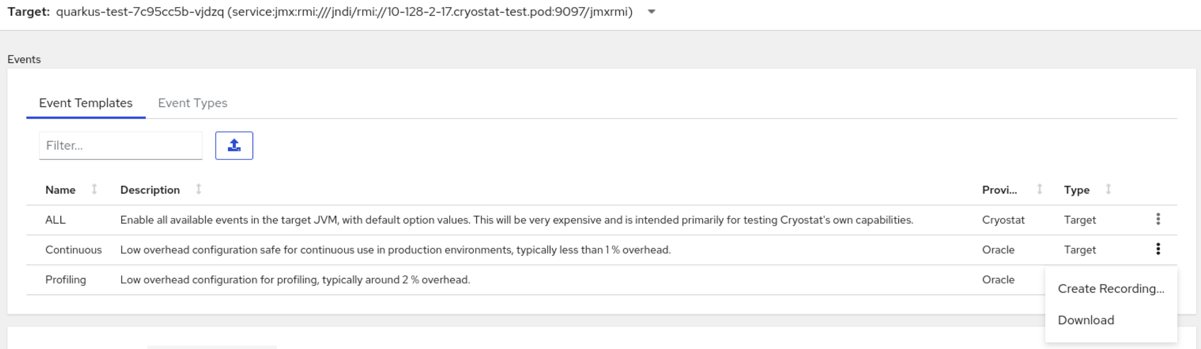
If the selected target JMX has SSL certification enabled for JMX connections, you must add its certificate when prompted.

Cryostat can encrypt and store credentials for a target JVM application in a database that is stored on a persistent volume claim (PVC) on Red Hat OpenShift.

4. Under the **Event Templates** tab, locate your listed event template and then select its more options menu.

5. From the more options menu, click **Download**. Depending on how you configured your operating system, a file-save dialog opens. Save the file to your preferred location.

Figure 3.2. Example of an event template’s more options menu



Target: quarkus-test-7c95cc5b-vjdzq (service:jmx:rmi:///jndi/rmi://10-128-2-17.cryostat-test.pod:9097/jmxrmi)

Events

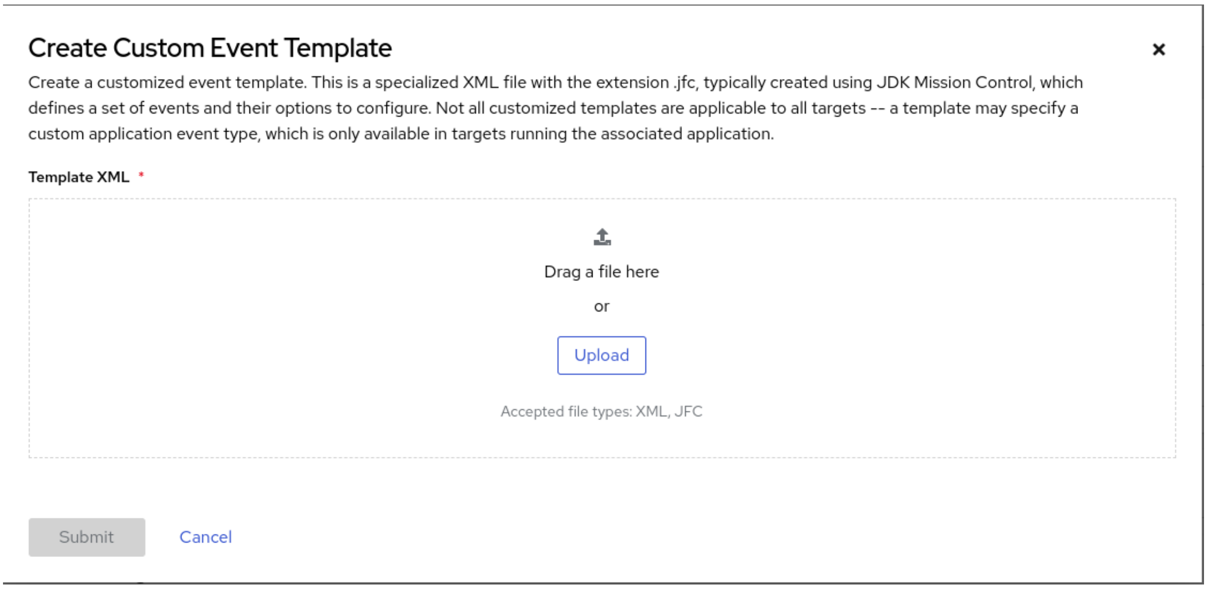
Name	Description	Provi...	Type
ALL	Enable all available events in the target JVM, with default option values. This will be very expensive and is intended primarily for testing Cryostat's own capabilities.	Cryostat	Target
Continuous	Low overhead configuration safe for continuous use in production environments, typically less than 1 % overhead.	Oracle	Target
Profiling	Low overhead configuration for profiling, typically around 2 % overhead.	Oracle	

6. Open the file with your default file editor and edit the file to meet your needs. You must save your file to retain your configuration changes.

   > NOTE
   >
   > You can add values to the **description** and **provider** attributes that can help with identifying your file at a later stage.

7. From the **Events** menu, go to the **Event Templates** tab then click the **Upload** icon. A **Create Custom Event Template** window opens in your Cryostat web console.

   Figure 3.3. Create Custom Event Template window

   Create Custom Event Template

   Create a customized event template. This is a specialized XML file with the extension .jfc, typically created using JDK Mission Control, which defines a set of events and their options to configure. Not all customized templates are applicable to all targets -- a template may specify a custom application event type, which is only available in targets running the associated application.

   Template XML *

8. Click **Upload** and use your default file editor to upload one or more configured event template files to the Cryostat web console. You can also drag and drop the files into the **Template XML** window.

9. Click the **Submit** button. The **Event Templates** tab opens on your Cryostat web console, where you can now view your custom event template.

10. *Optional:* After you create your event template, you can choose one of the following options for using your template to create a JFR recording:

    - From the **Automated Rules** menu, click **Create** and then select an event template from the **Template** list.
    - From the **Events** menu, locate your listed event template, then from the more options menu, select **Create Recording**.
    - From the **Recordings** menu, under the **Active Recordings** tab, click **Create**.

Additional resources

- See Creating a JDK Flight Recorder (JFR) recording (Creating a JFR recording with Cryostat)
- See Uploading an SSL certificate (Using Cryostat to manage a JFR recording)
- See Archiving JDK Flight Recorder (JFR) recordings (Using Cryostat to manage a JFR recording)

- See Enabling or disabling automated rules(Using automated rules on Cryostat)

CHAPTER 4. INTEGRATED APPLICATIONS

Cryostat integrates with specific applications that can enhance how you analyze data from your JFR recording.

4.1. VIEWING A JFR RECORDING ON GRAFANA

Cryostat 3.0 integrates with the Grafana application, so you can plot JFR recording data in Grafana. You can view plot data in time interval sections to precisely analyze the performance of your target JVM application.

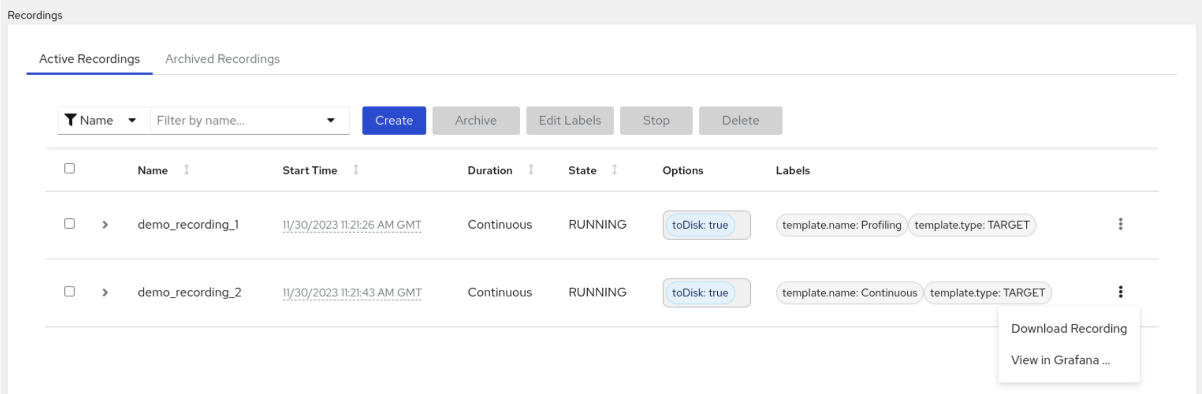
Prerequisites

- Entered your authentication details for your Cryostat instance.
- Created a JFR recording. See Creating a JFR recording in the Cryostat web console .

Procedure

1. Go to the **Recordings** menu or the **Archives** menu on your Cryostat instance.
2. Depending on your needs, click either the **Active Recordings** tab or the **Archived Recordings** tab.
3. Locate your JFR recording and then select the overflow menu.

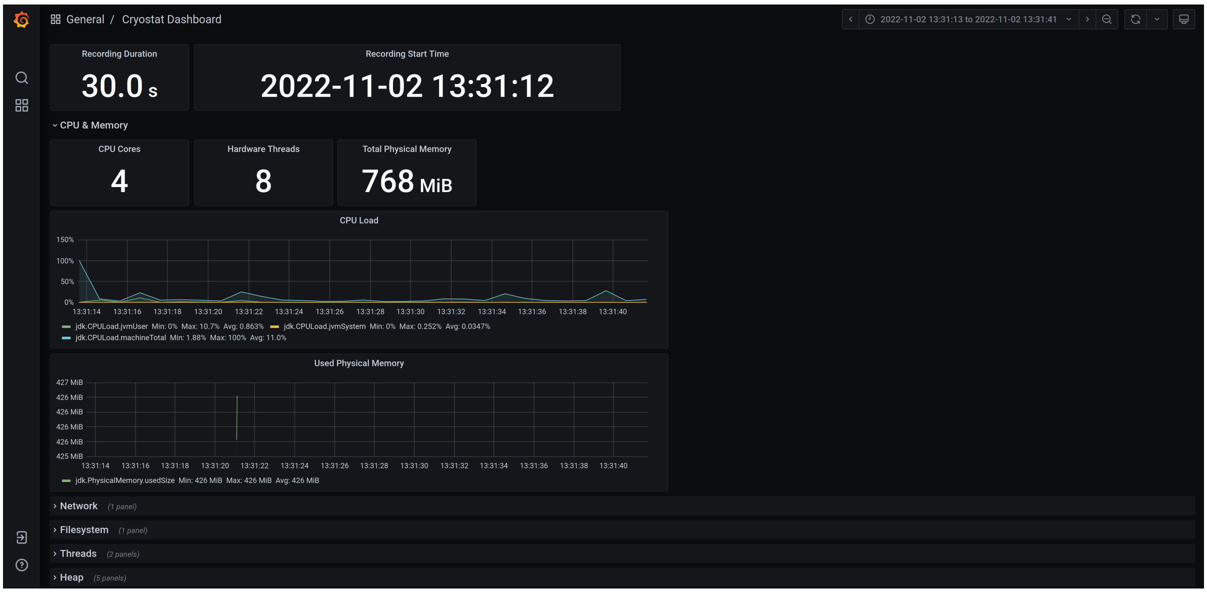
Figure 4.1. Overflow menu items available for an example JFR recording



□		Name	Start Time	Duration	State	Options	Labels	
□	›	demo_recording_1	11/30/2023 11:21:26 AM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling template.type: TARGET	⋮
□	›	demo_recording_2	11/30/2023 11:21:43 AM GMT	Continuous	RUNNING	toDisk: true	template.name: Continuous template.type: TARGET	⋮

4. From the overflow menu, click the **View in Grafana** option. The Grafana application opens in a new web browser window.
5. Enter your Red Hat OpenShift credentials in the Grafana web console login page, if prompted. A dashboard window opens and shows your JFR recording’s data in various time-series plots.
6. *Optional:* Interact with any plot by selecting a time-series segment on the plot. Grafana expands the on-screen data to show only the data for that time interval.

Figure 4.2. Example of a Grafana dashboard with plotted graphs



General / Cryostat Dashboard

2022-11-02 13:31:13 to 2022-11-02 13:31:41

Recording Duration: 30.0 s

Recording Start Time: 2022-11-02 13:31:12

CPU & Memory

CPU Cores: 4

Hardware Threads: 8

Total Physical Memory: 768 MiB

› Network (1 panel)

› Filesystem (1 panel)

› Threads (2 panels)

› Heap (5 panels)

Revised on 2024-07-02 13:35:46 UTC