Red Hat build of Cryostat 3

Creating a JFR recording with Cryostat

Last Updated: 2024-07-02

Legal Notice

Abstract

Use the Red Hat build of Cryostat to create JDK Flight Recorder (JFR) recordings that monitor the performance of Java Virtual Machines (JVMs) in containerized applications. You can also learn how to take a snapshot of a running JFR recording so that you can view data from a specific timeframe.

Table of Contents

PREFACE 3

MAKING OPEN SOURCE MORE INCLUSIVE 4

CHAPTER 1. JFR CREATION OPTIONS FOR CRYOSTAT 5

CHAPTER 2. CREATING A JFR RECORDING IN THE CRYOSTAT WEB CONSOLE 6

CHAPTER 3. CREATING SNAPSHOTS FROM AN ACTIVE RECORDING 11

CHAPTER 4. LABELS FOR JFR RECORDINGS 13

4.1. ADDING LABELS TO JFR RECORDINGS 13

4.2. EDITING A LABEL FOR YOUR JFR RECORDING 14

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. JFR CREATION OPTIONS FOR CRYOSTAT

With Cryostat, you can create a JDK Flight Recorder (JFR) recording that monitors the performance of your JVM in your containerized application. Additionally, you can take a snapshot of an active JFR recording to capture any collected data, up to a specific point in time, for your target JVM application.

Cryostat supports all of the following different ways to create JFR recordings:

- You can use the Cryostat web console to create JFR recordings manually for target JVMs that are using a JMX or agent HTTP connection.
- The Cryostat server can send on-demand requests over JMX or an agent HTTP connection to start JFR recordings dynamically based on automated rules.
- The Cryostat agent can start JFR recordings automatically at agent startup based on a given event template as part of the agent harvester feature.
- From Red Hat build of Cryostat 3.0 onward, the Cryostat agent can start JFR recordings dynamically based on MBean custom triggers and a given event template.

The rest of this document describes how to create a JFR recording manually in the Cryostat web console.

Additional resources

- Using automated rules on Cryostat
- Enabling dynamic JFR recordings based on MBean custom triggers

CHAPTER 2. CREATING A JFR RECORDING IN THE CRYOSTAT WEB CONSOLE

You can create a JFR recording that monitors the performance of your JVM located in your containerized application. After you create a JFR recording, you can start the JFR to capture real-time data for your JVM, such as heap and non-heap memory usage.

Prerequisites

- Installed Cryostat 3.0 on Red Hat OpenShift by using the OperatorHub option.
- Created a Cryostat instance in your Red Hat OpenShift project.
- Logged in to your Cryostat web console.
    - You can retrieve your Cryostat application’s URL by using the Red Hat OpenShift web console.

Procedure

1. On the **Dashboard** panel for your Cryostat web console, select a target JVM from the **Target** list.

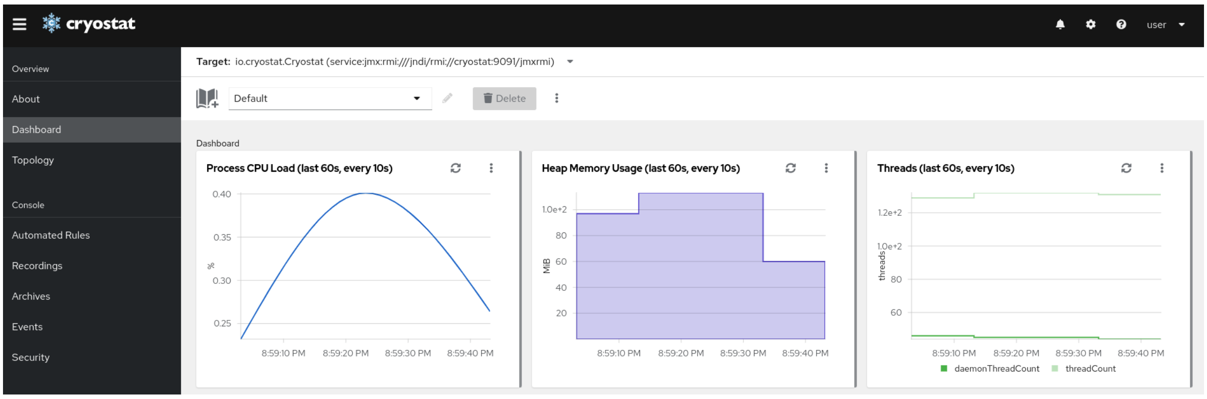
    ### NOTE

    Depending on how you configured your target applications, your target JVMs might be using a JMX connection or an agent HTTP connection. For more information about configuring your target applications, see Configuring Java applications.

    ### IMPORTANT

    If your target JVM is using an agent HTTP connection, ensure that you set the **cryostat.agent.api.writes-enabled** property to **true** when you configured your target application to load the Cryostat agent. Otherwise, the Cryostat agent cannot accept requests to start and stop JFR recordings.

    Figure 2.1. Example of selecting a Target JVM for your Cryostat instance

    Target: io.cryostat.Cryostat (service:jmx:rmi:///jndi/rmi://cryostat:9091/jmxrmi)

    Default

2. *Optional:* On the **Dashboard** panel, you can create a target JVM. From the **Target** list, click **Create Target**. The **Create Custom Target** window opens.

    a. In the **Connection URL** field, enter the URL for your JVM’s Java Management Extension

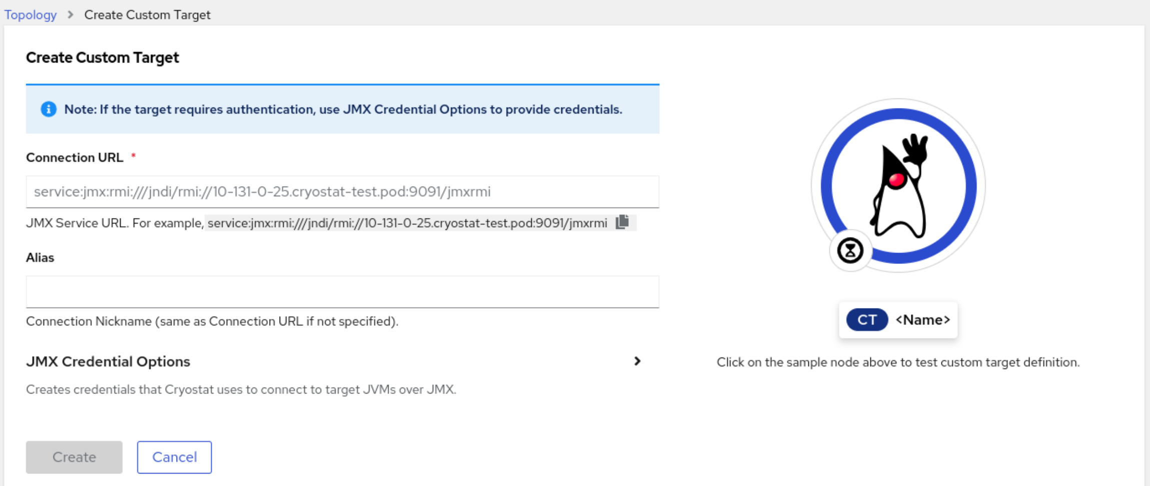
(JMX) endpoint.

b. *Optional:* To test if the **Connection URL** that you specified is valid, click the **Click to test** sample node image. If there is an issue with the **Connection URL**, an error message is displayed that provides a description of the issue and guidance to troubleshoot.

c. *Optional:* In the **Alias** field, enter an alias for your JMX Service URL.

d. Click **Create**.

Figure 2.2. Create Custom Target window



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

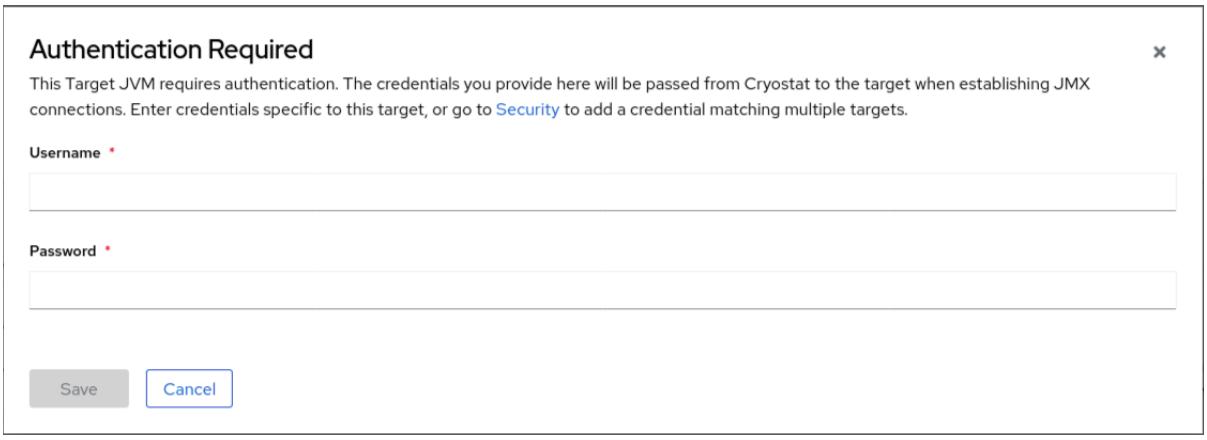
Creates credentials that Cryostat uses to connect to target JVMs over JMX.

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

3. From the navigation menu on the Cryostat web console, click **Recordings**.

4. *Optional:* Depending on how you configured your target JVM, an **Authentication Required** dialog might open on your web console. In the **Authentication Required** dialog box, enter your **Username** and **Password**. To provide your credentials to the target JVM, click **Save**.

Figure 2.3. 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 *

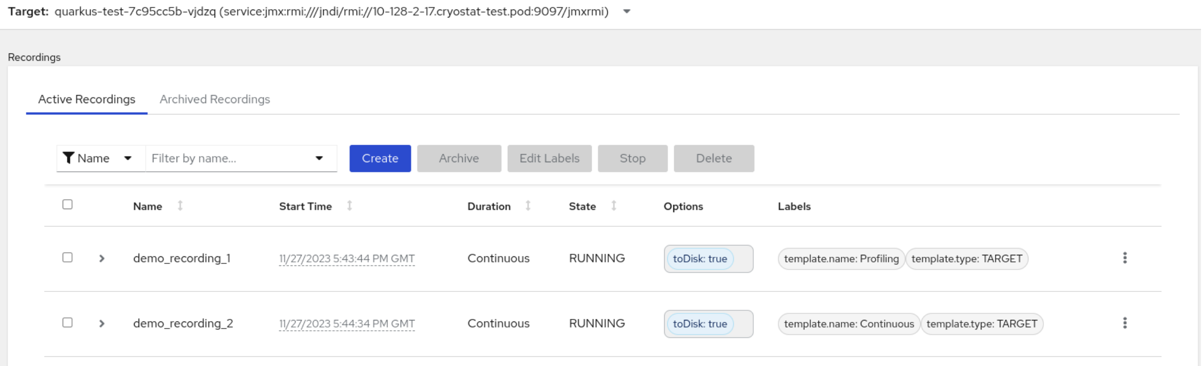
NOTE

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

Cryostat encrypts and stores credentials for a target JVM application in a database that is stored on a persistent volume claim (PVC) on Red Hat OpenShift. See Storing and managing credentials (Using Cryostat to manage a JFR recording).

5. On the **Active Recordings** tab, click **Create**.

Figure 2.4. Example of creating an active 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	11/27/2023 5:43:44 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling template.type: TARGET
□	›	demo_recording_2	11/27/2023 5:44:34 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Continuous template.type: TARGET

6. On the **Custom Flight Recording** tab:

   a. In the **Name** field, enter the name of the recording you want to create. If you enter a name in an invalid format, the web console displays an error message.

   b. If you want Cryostat to automatically restart an existing recording, select the **Restart if recording already exists** check box.

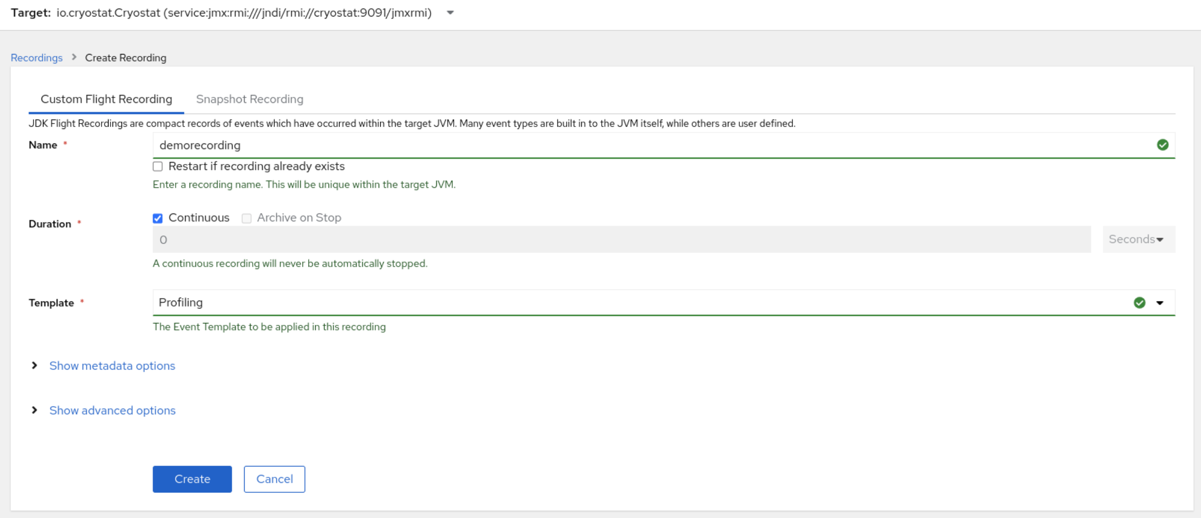
   > **NOTE**
   >
   > If you enter a name that already exists but you do not select **Restart if recording already exists**, Cryostat refuses to create a custom recording when you click the **Create** button.

   c. In the **Duration** field, select whether you want this recording to stop after a specified duration or to run continuously without stopping. If you want Cryostat to automatically archive your new JFR recording after the recording stops, click **Archive on Stop**.

   d. In the **Template** field, select the template that you want to use for the recording.

The following example shows continuous JVM monitoring, which you can enable by selecting **Continuous** from above the **Duration** field. This setting means that the recording will continue until you manually stop the recording. The example also shows selection of the **Profiling** template from the **Template** field. This provides additional JVM information to a JFR recording for troubleshooting purposes.

Figure 2.5. Example of creating a custom flight recording



Target: io.cryostat.Cryostat (service:jmx:rmi:///jndi/rmi://cryostat:9091/jmxrmi)

Recordings > Create Recording

Custom Flight Recording | Snapshot Recording

JDK Flight Recordings are compact records of events which have occurred within the target JVM. Many event types are built in to the JVM itself, while others are user defined.

Name * demorecording

□ Restart if recording already exists

Enter a recording name. This will be unique within the target JVM.

Duration * ☑ Continuous □ Archive on Stop

0 Seconds

A continuous recording will never be automatically stopped.

Template * Profiling

The Event Template to be applied in this recording

> Show metadata options

> Show advanced options

Create | Cancel

7. To access more options, click the following expandable hyperlinks:

   - **Show advanced options**, where you can select additional options for customizing your JFR recording.

   - **Show metadata options**, where you can add custom labels and metadata to your JFR recording.

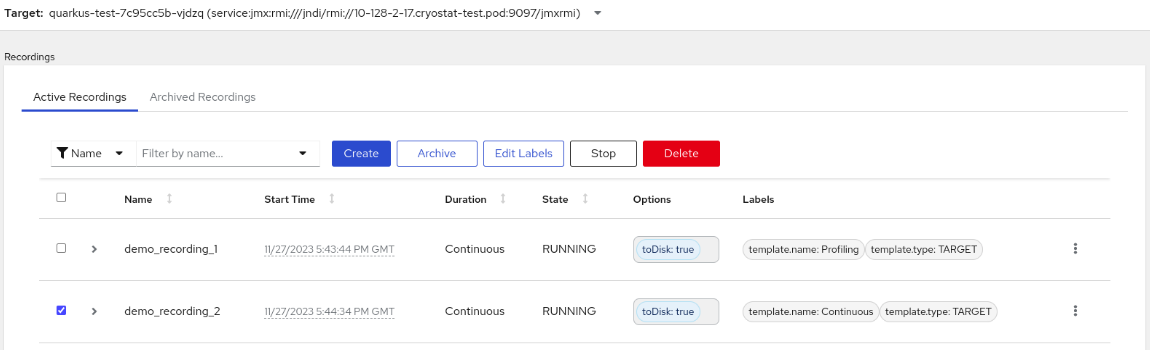
8. To create your JFR recording, click **Create**. The **Active Recordings** tab opens and lists your JFR recording.
   Your active JFR recording starts collecting data on the target JVM location inside your containerized application. If you specified a fixed duration for your JFR recordings, the target JVM stops the recording when it reaches the fixed duration setting. Otherwise, you must manually stop the recording.

9. *Optional:* On the **Active Recording** tab, you can also stop the recording.

   a. Select the checkbox next to the JFR recording’s name. On the toolbar in the **Active Recordings** tab, the Cryostat web console activates the **Stop** button.

   b. Click **Stop**. The JFR adopts the **STOPPED** status, so it stops monitoring the target JVM. The JFR still shows under the **Active Recording** tab.

Figure 2.6. Example of stopping an active recording



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

Recordings

Active Recordings | Archived Recordings

Name | Filter by name... | Create | Archive | Edit Labels | Stop | Delete

□		Name	Start Time	Duration	State	Options	Labels
□	>	demo_recording_1	11/27/2023 5:43:44 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling template.type: TARGET
☑	>	demo_recording_2	11/27/2023 5:44:34 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Continuous template.type: TARGET

IMPORTANT

JFR recording data might be lost in the following situations:

- Target JVM fails
- Target JVM restarts
- Target JVM Red Hat OpenShift Deployment is scaled down

Archive your JFR recordings to ensure you do not lose your JFR recording’s data.

Additional resources

- 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).

CHAPTER 3. CREATING SNAPSHOTS FROM AN ACTIVE RECORDING

You can take a snapshot of an active JFR recording to capture any collected data, up to a specific point in time, for your target JVM application. A snapshot is like a checkpoint marker that has a start point and an end point for a given time segment in a running JFR recording.

A snapshot gets stored in the memory of a target JVM application. This differs from an archive in that Cryostat stores an archive on a cloud storage disk, which is a more permanent solution for storing a JFR recording's data.

You can take snapshots of recordings if you want to experiment with different configuration changes among active JFR recordings.

When you create a snapshot for your JFR recording, Cryostat creates a new target JVM named **snapshot -*<snapshot_number>***, where ***<snapshot_number>*** is the number that Cryostat automatically assigns to your snapshot.

A target JVM recognizes a snapshot as an active recording. Cryostat sets any JFR snapshots in the **STOPPED** state, which means that the JFR snapshot does not record new data to the target JVM. Depending on the JFR configuration, active JFR recordings can continue to monitor the target JVM regardless of the number of snapshots taken.

NOTE

For a JFR recording that you set for continuous monitoring of a target JVM application, ensure that you create archived recordings to avoid losing JFR recording data.

If you choose to take regular snapshots to store your JFR recording data, the target JVM application might free some of its data storage space by replacing older recording data with newer recording data.

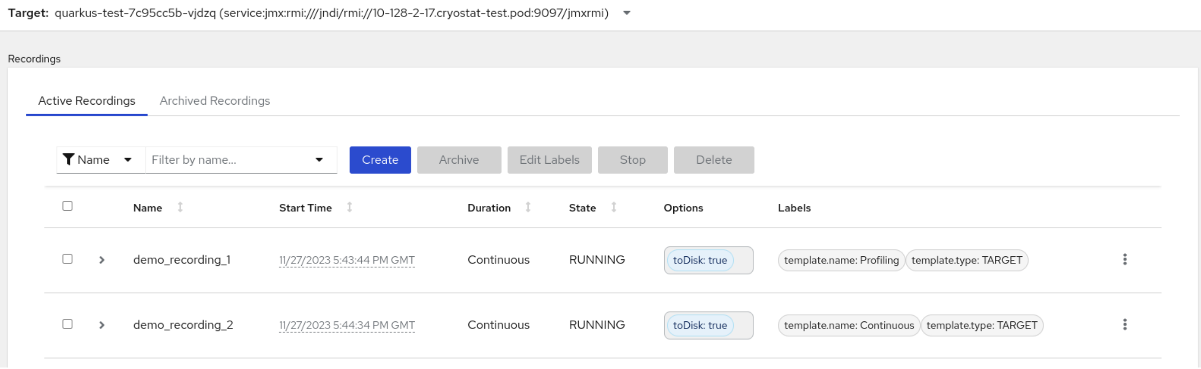
Prerequisites

- 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, click the **Create** button. A new window opens on the web console.

Figure 3.1. Example of creating an active recording

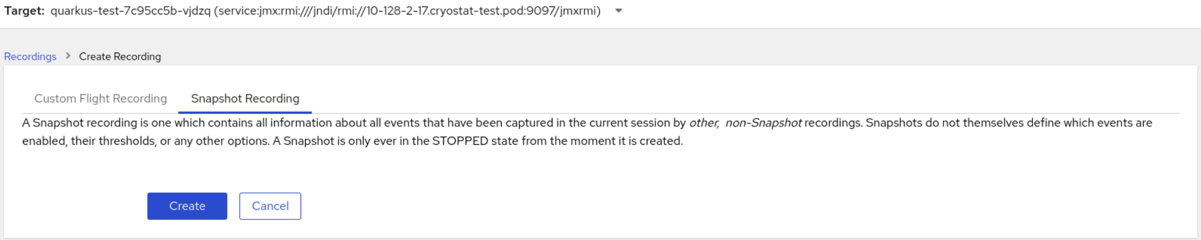


Target: quarkus-test-7c95cc5b-vjdzq (service:jmx:rmi:///jndi/rmi://10-128-2-17.cryostat-test.pod:9097/jmxrmi)

□		Name	Start Time	Duration	State	Options	Labels
□	›	demo_recording_1	11/27/2023 5:43:44 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling template.type: TARGET
□	›	demo_recording_2	11/27/2023 5:44:34 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Continuous template.type: TARGET

2. Click the **Snapshot Recording** tab.

Figure 3.2. Example of creating a snapshot recording

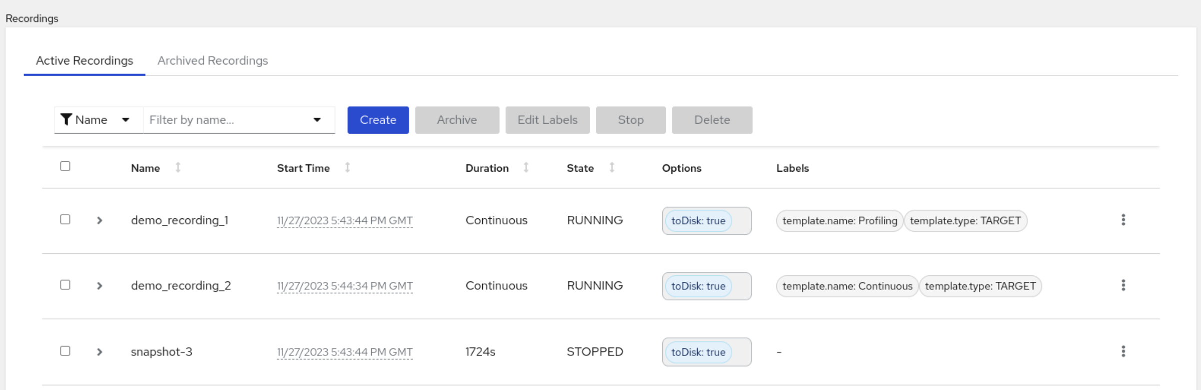


Target: quarkus-test-7c95cc5b-vjdzq (service:jmx:rmi:///jndi/rmi://10-128-2-17.cryostat-test.pod:9097/jmxrmi)

A Snapshot recording is one which contains all information about all events that have been captured in the current session by *other, non-Snapshot* recordings. Snapshots do not themselves define which events are enabled, their thresholds, or any other options. A Snapshot is only ever in the STOPPED state from the moment it is created.

3. Click **Create**. The **Active Recordings** table opens and it lists your JFR snapshot recording. The following example shows a JFR snapshot recording named **snapshot-3**.

Figure 3.3. Example of a completed snapshot recording



□		Name	Start Time	Duration	State	Options	Labels
□	›	demo_recording_1	11/27/2023 5:43:44 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling template.type: TARGET
□	›	demo_recording_2	11/27/2023 5:44:34 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Continuous template.type: TARGET
□	›	snapshot-3	11/27/2023 5:43:44 PM GMT	1724s	STOPPED	toDisk: true	-

NOTE

You can identify snapshots by the *snapshot* prefix from the list of active recordings.

Next steps

- To archive your JFR snapshot recording, see Archiving JDK Flight Recorder (JFR) recordings.

CHAPTER 4. LABELS FOR JFR RECORDINGS

When you create a JDK Flight Recorder (JFR) recording on Cryostat 3.0, you can add metadata to the recording by specifying a series of key-value label pairs.

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.

The following list details some common recording label use cases:

- Attach metadata to your JFR recording.
- Perform batch operations on recordings that contain identical labels.
- Use labels when running queries on recordings.

You can use Cryostat to create a JFR recording that monitors the performance of your JVM in your containerized application. Additionally, you can take a snapshot of an active JFR recording to capture any collected data, up to a specific point in time, for your target JVM application.

4.1. ADDING LABELS TO JFR RECORDINGS

When you create a JFR recording on Cryostat 3.0, you can use labels to add metadata that contain key-value label pairs to the recording.

Cryostat applies default recording labels to a created JFR recording. These default labels capture information about the event template that Cryostat used to create the JFR recording.

You can add custom labels to your JFR recording so that you can run specific queries that meet your needs, such as identifying specific JFR recordings or performing batch operations on recordings with the same applied labels.

Prerequisites

- Logged in to your Cryostat web console.
- Created or selected a target JVM for your Cryostat instance.

Procedure

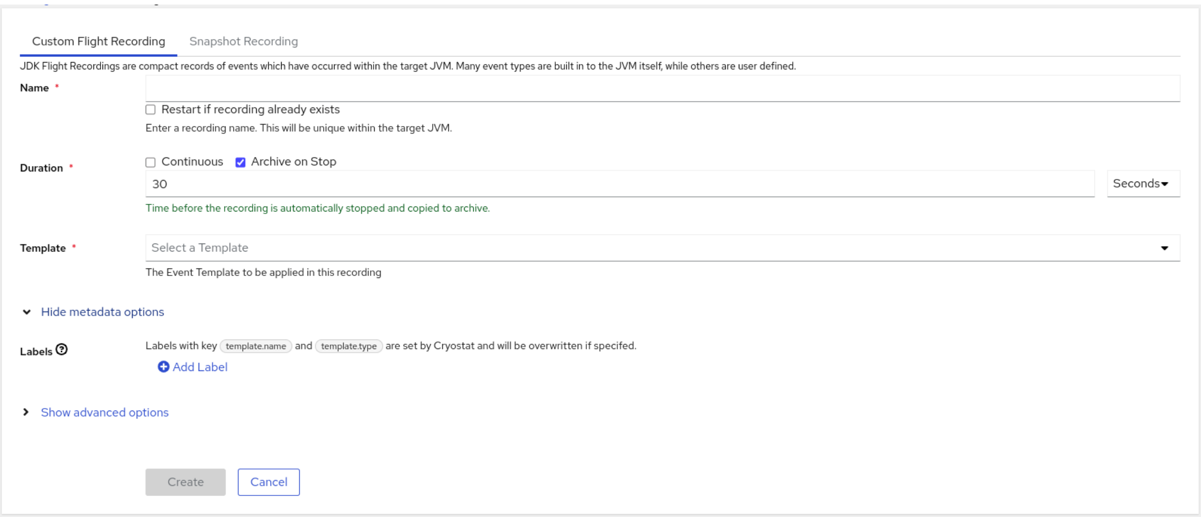
1. From your Cryostat web console, click **Recordings**.
2. Under the **Active Recordings** tab, click **Create**.
3. On the **Custom Flight Recording** tab, expand **Show metadata options**.

   NOTE

   On the **Custom Flight Recording** tab, you must complete any mandatory field that is marked with an asterisk.

4. Click **Add label**.

Figure 4.1. The Add Label button that is displayed under the Custom Flight Recording tab



Custom Flight Recording | Snapshot Recording

JDK Flight Recordings are compact records of events which have occurred within the target JVM. Many event types are built in to the JVM itself, while others are user defined.

Name *

□ Restart if recording already exists

Enter a recording name. This will be unique within the target JVM.

Duration *

□ Continuous ☑ Archive on Stop

30 Seconds

Time before the recording is automatically stopped and copied to archive.

Template *

Select a Template

The Event Template to be applied in this recording

Hide metadata options

Labels

Labels with key template.name and template.type are set by Cryostat and will be overwritten if specifed.

Add Label

Show advanced options

Create Cancel

5. Enter values in the provided **Key** and **Value** fields. For example, if you want to file an issue with the recordings, you could enter the reason in the **Key** field and then enter the issue type in the **Value** field.

6. Click **Create** to create your JFR recording. Your recording is then shown under the **Active Recordings** tab along with any specified recording labels and custom labels.

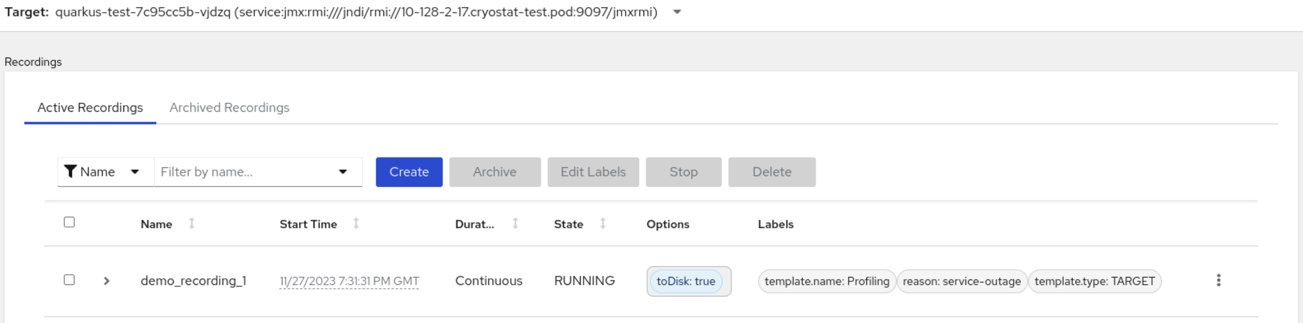
TIP

You can access archived JFR recordings from the **Archives** menu. See Uploading a JFR recording to Cryostat archives location (Using Cryostat to manage a JFR recording).

Example

The following example shows two default recording labels, **template.name: Profiling** and **template.type: TARGET**, and one custom label, **reason:service-outage**.

Figure 4.2. Example of an active recording with defined recording labels and a custom label



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

Recordings

Active Recordings | Archived Recordings

Name | Filter by name... | Create | Archive | Edit Labels | Stop | Delete

□		Name	Start Time	Durat...	State	Options	Labels
□	›	demo_recording_1	11/27/2023 7:31:31 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling reason: service-outage template.type: TARGET

4.2. EDITING A LABEL FOR YOUR JFR RECORDING

On the Cryostat web console, you can navigate to the **Recordings** menu and then edit a label and its metadata for your JFR recording. You can also edit the label and metadata for a JFR recording that you uploaded to archives.

Prerequisites

- Logged in to your Cryostat web console.

- Created a JFR recording and attach labels to this recording.

Procedure

1. On your Cryostat web console, click the **Recording** menu.
2. From the **Active Recordings** tab, locate your JFR recording, and then select the checkbox next to it.
3. Click **Edit Labels**. An **Edit Recording Label** pane opens in your Cryostat web console, which you can use to add, edit, or delete labels for your JFR recording.

   TIP

   You can select multiple JFR recordings by selecting the checkbox that is next to each recording. Click the **Edit Labels** button if you want to bulk edit recordings that contain the same labels or add new identical labels to multiple recordings.

4. *Optional*: You can perform any of the following actions from the **Edit Recording Labels** pane:
   a. Click **Add** to create a label.
   b. Delete a label by clicking the **X** next to the label.
   c. Edit a label by modifying any content in a field. After you edit content, a green tick is shown in the field to indicate an edit.
5. Click **Save**.
6. *Optional*: You can archive your JFR recordings along with their labels by completing the following steps:
   a. Select the checkbox next to the recording's name.
   b. Click the **Archive** button. You can locate your recording under the **Archived Recordings** tab.
      By archiving your recording with its labels, you can enhance your search capabilities when you want to locate the recording at a later stage. You can also add additional labels to any recording that you uploaded to the Cryostat archives.

      NOTE

      Cryostat preserves any labels with the recording for the lifetime of the archived recording.

Verification

- From the **Active Recordings** tab, check that your changes display under the **Labels** section for your recording.

Additional resources

- Archiving JDK Flight Recorder (JFR) recordings (Using Cryostat to manage a JFR recording)

Revised on 2024-07-02 14:08:19 UTC