



Red Hat build of Cryostat 2

Cryostat のスタートガイド

法律上の通知

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.

概要

Red Hat build of Cryostat は、OpenShift Container Platform で提供される Red Hat 製品です。Cryostat のスタートガイドでは、この製品の概要と、ソフトウェアをインストールして使用を開始する方法を説明します。

目次

はじめに	3
多様性を受け入れるオープンソースの強化	4
第1章 CRYOSTAT の概要	5
第2章 RED HAT BUILD OF CRYOSTAT OPERATOR を使用した RED HAT OPENSIFT への CRYOSTAT のインストール	6
2.1. 複数の NAMESPACE での CRYOSTAT の作成	10
第3章 JAVA アプリケーションの設定	15
Cryostat エージェント	15
リモート Java Management Extensions (JMX) 接続	17
Cryostat エージェントと JMX のハイブリッド	17
3.1. CRYOSTAT エージェントを使用したアプリケーションの設定	17
3.2. JMX 接続を使用したアプリケーションの設定	21
3.3. CRYOSTAT エージェントと JMX 接続を使用したアプリケーションの設定	24
第4章 CRYOSTAT を使用した JFR レコーディングの作成	28
4.1. CRYOSTAT WEB コンソールでの JFR レコーディングの作成	28
4.2. アクティブなレコーディングからのスナップショット作成	32
4.3. JFR レコーディングのラベル	34

はじめに

Red Hat build of Cryostat は、JDK Flight Recorder (JFR) のコンテナネイティブ実装です。これを使用すると、OpenShift Container Platform クラスターで実行されるワークロードで Java 仮想マシン (JVM) のパフォーマンスを安全にモニターできます。Cryostat 2.4 を使用すると、Web コンソールまたは HTTP API を使用して、コンテナ化されたアプリケーション内の JVM の JFR データを起動、停止、取得、アーカイブ、インポート、およびエクスポートできます。

ユースケースに応じて、Cryostat が提供するビルトインツールを使用して、Red Hat OpenShift クラスターに直接レコーディングを保存して分析したり、外部のモニタリングアプリケーションにレコーディングをエクスポートして、レコーディングしたデータをより詳細に分析したりできます。



重要

Red Hat build of Cryostat は、テクノロジープレビュー機能のみです。テクノロジープレビュー機能は、Red Hat 製品のサービスレベルアグリーメント (SLA) の対象外であり、機能的に完全ではないことがあります。Red Hat は、実稼働環境でこれらを使用することを推奨していません。テクノロジープレビュー機能は、最新の製品機能をいち早く提供して、開発段階で機能のテストを行いフィードバックを提供していただくことを目的としています。

Red Hat のテクノロジープレビュー機能のサポート範囲に関する詳細は、[テクノロジープレビュー機能のサポート範囲](#) を参照してください。

多様性を受け入れるオープンソースの強化

Red Hat では、コード、ドキュメント、Web プロパティにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、[Red Hat CTO である Chris Wright のメッセージ](#) をご覧ください。

第1章 CRYOSTAT の概要

Cryostat は、JDK Flight Recorder (JFR) をベースとするコンテナネイティブ Java アプリケーションで、Red Hat OpenShift クラスターで実行されるコンテナ化されたワークロードの Java 仮想マシン (JVM) パフォーマンスを監視できます。

コンテナ化された Java アプリケーションをホストする Red Hat OpenShift プロジェクトのコンテナに Cryostat をデプロイできます。また、コンテナ化されたワークロードの実行に使用する JVM インスタンスに対応する JVM ターゲットを作成できます。さらに、Cryostat を JVM ターゲットに接続して、各 JVM ターゲットのヒープおよびヒープ以外のメモリー使用量、スレッド数、ガベージコレクション、およびその他のパフォーマンスメトリクスに関するデータを記録および分析できます。

Cryostat に含まれるツールを使用して、JVM のパフォーマンスのリアルタイム監視、JDK Flight Recorder (JFR) のレコーディングおよびスナップショットのキャプチャー、自動分析レポートの生成、Grafana ダッシュボードを使用してレコーディングされたパフォーマンスデータの可視化を行えます。

Cryostat Web コンソールおよび HTTP API は、外部モニタリングアプリケーションを使用せずにコンテナ内の JVM パフォーマンスデータを分析する方法を提供します。ただし、クラスター環境外にあるデータのより深い分析を行う必要がある場合は、Cryostat からのレコーディングを JDK Mission Control (JMC) の外部インスタンスにエクスポートすることもできます。

Cryostat は、OpenShift Container Platform の標準機能としてロールベースのアクセス制御 (RBAC) をサポートします。ユーザーロールごとに異なるレベルの承認を設定し、フライトレコーディングデータのプライバシーと整合性を確保できます。

Operator Lifecycle Manager (OLM) を使用して、Red Hat OpenShift プロジェクトに Cryostat をインストールできます。

Red Hat Ecosystem Catalog から最新の Cryostat コンポーネントイメージをダウンロードすることもできます。Red Hat Ecosystem Catalog には、Cryostat 2.4 の次のコンテナイメージが存在します。

- Cryostat
- Red Hat build of Cryostat Operator
- Red Hat build of Cryostat Operator バンドル
- Cryostat レポート
- Cryostat Grafana ダッシュボード
- JFR データソース

関連情報

- [Operator ライフサイクルマネージャー \(OLM\) \(OpenShift Container Platform\)](#)
- [コンテナイメージ \(Red Hat エコシステムカタログ\)](#)

第2章 RED HAT BUILD OF CRYOSTAT OPERATOR を使用した RED HAT OPENSIFT への CRYOSTAT のインストール

Operator Lifecycle Manager (OLM) を使用して、Red Hat build of Cryostat Operator を Red Hat OpenShift クラスターのプロジェクトにインストールできます。Red Hat build of Cryostat Operator を使用すると、シングル namespace またはマルチ namespace の Cryostat インスタンスを作成できます。これらのインスタンスは、Red Hat OpenShift Web コンソールからアクセスできる GUI を使用して制御できます。



重要

Red Hat build of Cryostat Operator のサブスクリプションを Cryostat 2.0 から Cryostat 2.4 に更新する場合は、更新チャンネルを **stable-2.0** から **stable** に変更する必要があります。

前提条件

- OpenShift Container Platform 4.11 以降のクラスターを作成している。
- Red Hat build of Cryostat Operator をプロジェクトにインストールする権限を持つ Red Hat OpenShift ユーザーアカウントを作成している。
- クラスターに Operator Lifecycle Manager (OLM) がインストールされている。
- Red Hat OpenShift の cert-manager Operator を使用して cert-manager をインストールしている。
 - OpenShift Container Platform 4.11 以降を使用している場合は、Red Hat OpenShift の cert-manager Operator をインストールできます。詳細は、[Red Hat OpenShift の cert-manager Operator](#) を参照してください。
- Red Hat OpenShift Web コンソールを使用して Red Hat OpenShift にログインしている。

手順

1. ブラウザーで、Web コンソールを使用して **Home > Projects** に移動します。
2. Red Hat build of Cryostat をインストールするプロジェクトの名前を選択します。
3. Red Hat build of Cryostat Operator をインストールします。
 - a. Web コンソールのナビゲーションメニューで、**Operators > OperatorHub** に移動します。
 - b. リストから **Red Hat build of Cryostat Operator** を選択します。画面上部の検索ボックスを使用すると、Red Hat build of Cryostat Operator を検索できます。
 - c. **Install** をクリックし、プロジェクトに Red Hat build of Cryostat Operator をインストールします。
Red Hat OpenShift Web コンソールで、Cryostat カスタムリソース (CR) を作成するように求められます。



注記

複数の namespace に対して有効な Cryostat インスタンスをインストールする場合は、**Installation mode** エリアで **All namespaces on the cluster (default)** ラジオボタンをクリックします。

CR は手動または自動で作成できます。CR を手動で作成する場合は、ステップ 4 を参照してください。CR を自動的に作成する場合は、ステップ 5 を参照してください。

4. CR を手動で作成する場合は、以下の手順を実行します。
 - i. Web コンソールを使用して **Operators > Installed Operators** に移動し、インストール済み Operator のリストから **Red Hat build of Cryostat Operator** を選択します。

図2.1 インストール済み Operator のリストでの Red Hat build of Cryostat Operator の表示

Project: cryostat-test ▾

Installed Operators

Installed Operators are represented by ClusterServiceVersions within this Namespace. For more information, see the [Understanding Operators documentation](#) or create an Operator and ClusterServiceVersion using the [Operator SDK](#).

Name ▾ Search by name... /

Name	Managed Namespaces	Status	Last updated	Provided APIs
Red Hat build of Cryostat 2.4.0 provided by Red Hat	All Namespaces	● Succeeded Up to date	Dec 5, 2023, 1:26 PM	Cluster Cryostat Cryostat

- ii. **Details** タブをクリックします。
- iii. シングル namespace の Cryostat インスタンスを作成するには、**Provided APIs** セクションに移動します。次に、**Cryostat** で **Create instance** をクリックします。



注記

複数の namespace に対して有効な Cryostat インスタンスを作成する場合は、**Provided APIs** セクションで **Cluster Cryostat** を選択し、**Create instance** をクリックします。**Cluster Cryostat** API には、Cryostat アプリケーションとその関連コンポーネントのデプロイメントを制御する設定オプションがあります。詳細は、[複数の namespace での Cryostat の作成](#) を参照してください。

図2.2 Red Hat build of Cryostat Operator によって提供される Cryostat API の選択

Project: cryostat-test ▾

Installed Operators > Operator details

Red Hat build of Cryostat
2.4.0 provided by Red Hat

Actions ▾

[Details](#) [YAML](#) [Subscription](#) [Events](#) [All instances](#) [Cluster Cryostat](#) [Cryostat](#)

Provided APIs

Cluster Cryostat

ClusterCryostat allows you to install Cryostat for multiple namespaces or cluster-wide. It contains configuration options for controlling the Deployment of the Cryostat application and its related components. A ClusterCryostat or Cryostat instance must be...

[Create instance](#)

Cryostat

Cryostat allows you to install Cryostat for a single namespace. It contains configuration options for controlling the Deployment of the Cryostat application and its related components. A ClusterCryostat or Cryostat instance must be created to instruct the...

[Create instance](#)

Provider
Red Hat

Created at
Nov 24, 2023, 9:42 PM

Links
Upstream Project
<https://github.com/cryostatio/cryostat>

Website
<https://cryostat.io/>

Maintainers
The Cryostat Authors
cryostat-development@googlegroups.com

- iv. **Form view** または **YAML view** のラジオボタンをクリックします。YAML 設定ファイルに情報を入力する場合は、**YAML view** をクリックします。
- v. 作成する Cryostat のインスタンスの名前を指定します。
- vi. **オプション**: ラベルフィールドで、デプロイするオペランドワークロードのラベルまたはアノテーションを指定します。
デプロイメントに追加するその他の設定オプションも指定できます。

図2.3 Web コンソールでのフォームを使用した Cryostat インスタンスの作成

Project: cryostat-test ▾

Create Cryostat

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: Form view YAML view

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name *

cryostat-sample

Labels

app=frontend

Minimal Deployment *

false
Deploy a pared-down Cryostat instance with no Grafana Dashboard or JFR Data Source.

Enable cert-manager Integration

true
Use cert-manager to secure in-cluster communication between Cryostat components. Requires cert-manager to be installed.

あるいは、フォームを使用する代わりに、YAML テンプレートを使用してインスタンスを作成し、追加の設定オプションを指定することもできます。

図2.4 Web コンソールでの YAML テンプレートを使用した Cryostat インスタンスの作成

Project: cryostat-test ▾

Create Cryostat

Create by manually entering YAML or JSON definitions, or by dragging and dropping a file into the editor.

Configure via: Form view YAML view

```

1 apiVersion: operator.cryostat.io/v1beta1
2 kind: cryostat
3 metadata:
4   name: cryostat-sample
5   namespace: cryostat-test
6 spec:
7   storageOptions:
8     pvc:
9       annotations: {}
10      labels: {}
11      spec: {}
12   trustedCertSecrets: []
13   reportOptions:
14     replicas: 0
15   eventTemplates: []
16   enableCertManager: true
17   minimal: false
18

```

Cryostat

Schema

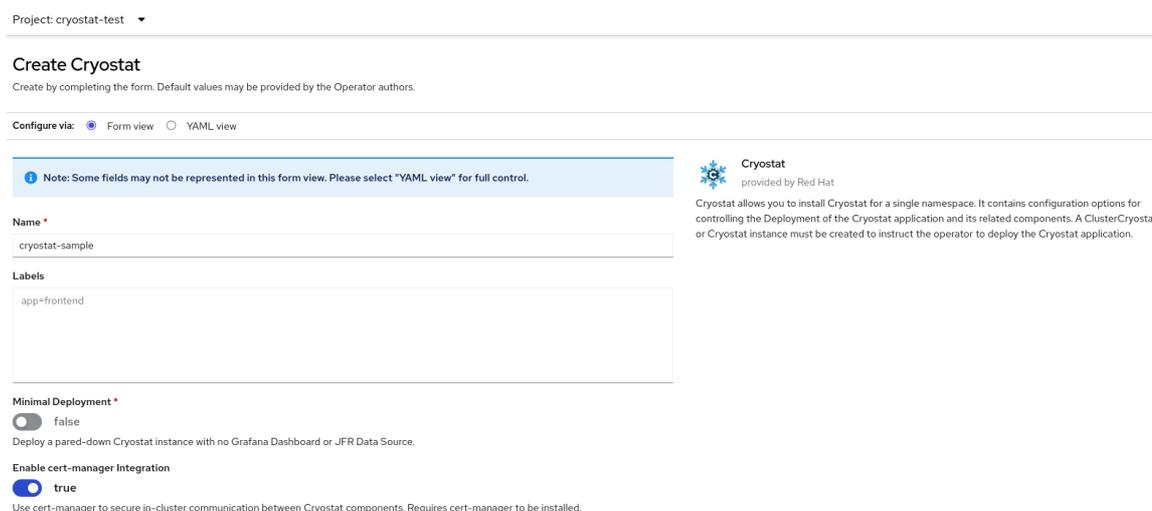
Cryostat allows you to install Cryostat for a single namespace. It contains configuration options for controlling the Deployment of the Cryostat application and its related components. A ClusterCryostat or Cryostat instance must be created to instruct the operator to deploy the Cryostat application.

- apiVersion**
string
APIVersion defines the versioned schema of this representation of an object. Servers should convert recognized schemas to the latest internal value, and may reject unrecognized values. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#resources>
- kind**
string
Kind is a string value representing the REST resource this object represents. Servers may infer this from the endpoint the client submits requests to. Cannot be updated. In CamelCase. More info: <https://git.k8s.io/community/contributors/devel/sig-architecture/api-conventions.md#types-kinds>
- metadata**

5. 自動プロンプトオプションを使用して CR を作成する場合は、プロンプトの指示に従い、次の手順を実行します。
 - i. **Form view** または **YAML view** のラジオボタンをクリックします。YAML 設定ファイルに情報を入力する場合は、**YAML view** をクリックします。
 - ii. 作成する Cryostat のインスタンスの名前を指定します。

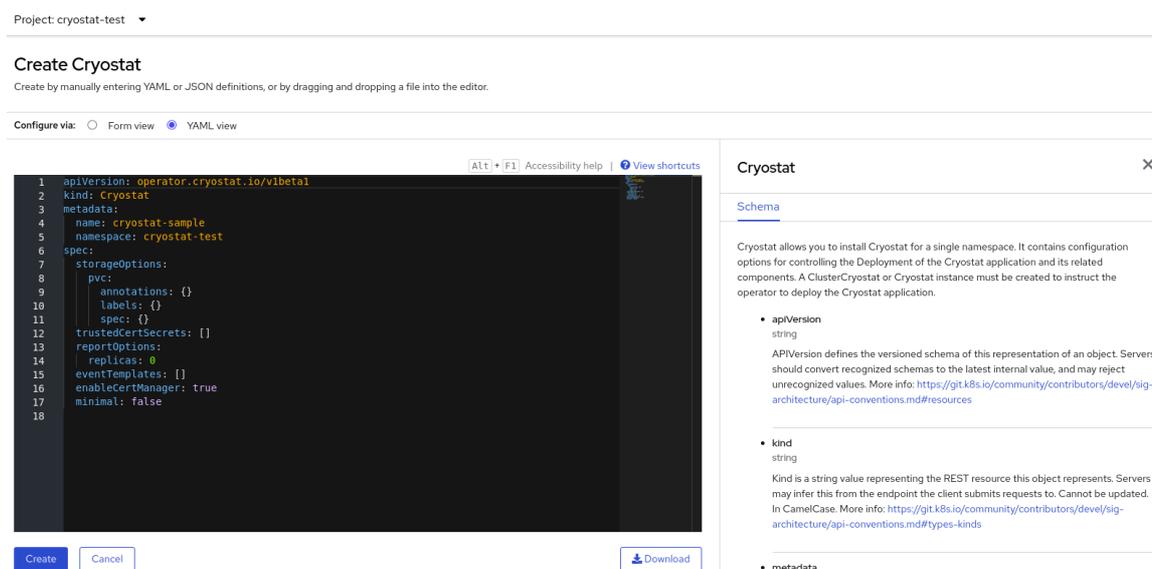
- iii. **オプション:** ラベルフィールドで、デプロイするオペランドワークロードのラベルまたはアノテーションを指定します。
 デプロイメントに追加するその他の設定オプションも指定できます。

図2.5 Web コンソールでのフォームを使用した Cryostat インスタンスの作成



あるいは、フォームを使用する代わりに、YAML テンプレートを使用してインスタンスを作成し、追加の設定オプションを指定することもできます。

図2.6 Web コンソールでの YAML テンプレートを使用した Cryostat インスタンスの作成



6. **Create** をクリックして、Cryostat インスタンスの作成プロセスを開始します。
 Cryostat インスタンスのすべてのリソースの準備が完了しなければアクセスできません。

検証

1. Web コンソールのナビゲーションメニューで、**Operators** をクリックし、**Installed Operators** をクリックします。
2. インストール済み Operator のテーブルから、**Red Hat build of Cryostat Operator** を選択します。
3. **Cryostat** タブを選択します。
 Cryostat インスタンスが、インスタンスのテーブルで開き、次の条件が一覧表示されます。

- **TLSSetupComplete** は **true** に設定されている。
- **MainDeploymentAvailable** は **true** に設定されている。
- オプション: レポートジェネレーターサービスを有効にした場合、**ReportsDeploymentAvailable** が表示され、**true** に設定されている。

図2.7 OpenShift の Cryostat インスタンスの Status 列で True に設定された条件の例

Project: cryostat-test ▾

Installed Operators > Operator details

Red Hat build of Cryostat
2.4.0 provided by Red Hat

Actions ▾

Details YAML Subscription Events All instances Cluster Cryostat **Cryostat**

Cryostats Show operands in: All namespaces Current namespace only [Create Cryostat](#)

Name ▾ Search by name... /

Name	Kind	Namespace	Status	Labels	Last updated
cryostat-sample	Cryostat	cryostat-test	Conditions: TLSSetupComplete, MainDeploymentAvailable, MainDeploymentProgressing, ReportsDeploymentAvailable, ReportsDeploymentProgressing	No labels	Nov 24, 2023, 9:43 PM

4. オプション: Cryostat テーブルから Cryostat インスタンスを選択します。Cryostat Conditions テーブルに移動すると、各条件の詳細情報を確認できます。

図2.8 各条件とその基準をリストした Cryostat Conditions 表の例

Type	Status	Updated	Reason	Message
TLSSetupComplete	True	Nov 24, 2023, 9:43 PM	AllCertificatesReady	All certificates for Cryostat components are ready.
MainDeploymentAvailable	True	Dec 7, 2023, 5:00 AM	MinimumReplicasAvailable	Deployment has minimum availability.
MainDeploymentProgressing	True	Nov 24, 2023, 9:43 PM	NewReplicaSetAvailable	ReplicaSet "cryostat-sample-7dd96f95cd" has successfully progressed.
ReportsDeploymentAvailable	True	Nov 24, 2023, 9:43 PM	MinimumReplicasAvailable	Deployment has minimum availability.
ReportsDeploymentProgressing	True	Nov 24, 2023, 9:43 PM	NewReplicaSetAvailable	ReplicaSet "cryostat-sample-reports-7fbbf6d95f" has successfully progressed.

次のステップ

- [Web コンソールを使用して Cryostat にアクセスする](#)

2.1. 複数の NAMESPACE での CRYOSTAT の作成

Red Hat build of Cryostat Operator は、複数の namespace で機能する Cryostat インスタンスを作成できる **Cluster Cryostat API** を提供します。

前提条件

- OpenShift Container Platform 4.11 以降のクラスターを作成している。
- Red Hat build of Cryostat Operator をプロジェクトにインストールする権限を持つ Red Hat OpenShift ユーザーアカウントを作成している。
- クラスターに Operator Lifecycle Manager (OLM) がインストールされている。
- Red Hat OpenShift の cert-manager Operator を使用して cert-manager をインストールしている。
 - OpenShift Container Platform 4.11 以降を使用している場合は、Red Hat OpenShift の cert-manager Operator をインストールできます。詳細は、[Red Hat OpenShift \(OpenShift](#)

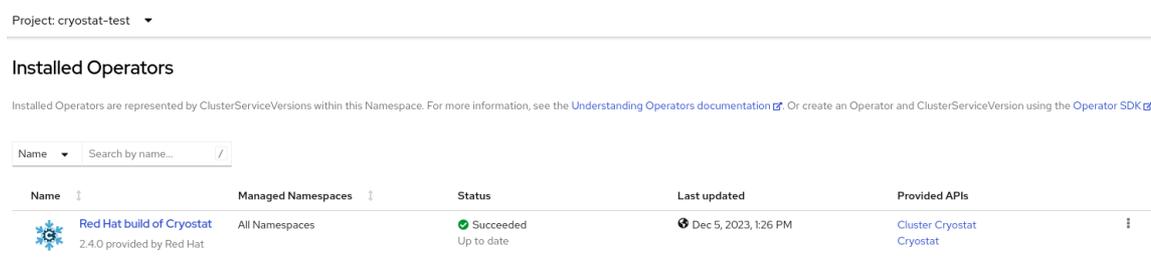
manager Operator をインストールしてください。詳細は、[Red Hat OpenShift \(OpenShift Container Platform\) の cert-manager Operator](#) を参照してください。

- Red Hat OpenShift Web コンソールを使用して Red Hat OpenShift にログインしている。

手順

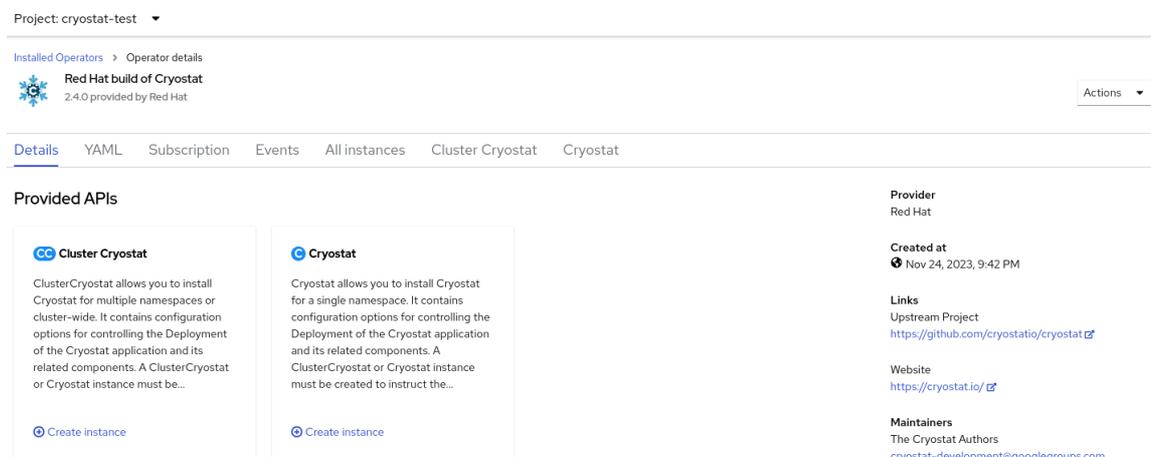
1. ブラウザーで、Web コンソールを使用して **Home > Projects** に移動します。
2. Red Hat build of Cryostat Operator をインストールするプロジェクトの名前を選択します。
3. Red Hat build of Cryostat Operator をインストールします。
Red Hat OpenShift Web コンソールで、Cryostat カスタムリソース (CR) を作成するように求められます。
4. CR を作成するには、以下の手順を実行します。
 - i. Web コンソールを使用して **Operators > Installed Operators** に移動し、インストール済み Operator のリストから **Red Hat build of Cryostat Operator** を選択します。

図2.9 インストール済み Operator のリストでの Red Hat build of Cryostat Operator を表示



- ii. **Details** タブをクリックします。
- iii. Cryostat のマルチ namespace インスタンスを作成するには、**Provided APIs** セクションに移動します。次に、**Cluster Cryostat** で **Create instance** をクリックします。

図2.10 Red Hat build of Cryostat Operator によって提供される Cluster Cryostat API の選択



- iv. **Form view** または **YAML view** のラジオボタンをクリックします。YAML 設定ファイルに情報を入力する場合は、**YAML view** をクリックします。
- v. 作成する Cluster Cryostat インスタンスの一意的な名前を指定します。



注記

Cluster Cryostat インスタンスに指定する名前が一意であり、Cluster Cryostat インスタンスのインストール namespace またはターゲット namespace ですすでに作成されているシングル namespace の Cryostat インスタンスの名前と競合しないことを確認してください。

- vi. **オプション:** ラベルフィールドで、デプロイするオペランドワークロードのラベルまたはアノテーションを指定します。
デプロイメントに追加するその他の設定オプションも指定できます。

図2.11 Web コンソールでのフォームを使用した Cluster Cryostat インスタンスの作成

Project: cryostat-test ▾

Create ClusterCryostat

Create by completing the form. Default values may be provided by the Operator authors.

Configure via: Form view YAML view

Note: Some fields may not be represented in this form view. Please select "YAML view" for full control.

Name *

Labels

Install Namespace *

Namespace where Cryostat should be installed. On multi-tenant clusters, we strongly suggest installing Cryostat into its own namespace.

Minimal Deployment *

 false
Deploy a pared-down Cryostat instance with no Grafana Dashboard or JFR Data Source.

Cluster Cryostat
provided by Red Hat

ClusterCryostat allows you to install Cryostat for multiple namespaces or cluster-wide. It contains configuration options for controlling the Deployment of the Cryostat application and its related components. A ClusterCryostat or Cryostat instance must be created to instruct the operator to deploy the Cryostat application.

- vii. **Install Namespace** フィールドで、Cryostat のこのインスタンスをインストールする namespace を選択します。

ヒント

Red Hat build of Cryostat Operator は、Cryostat アプリケーションと比較してより多くの権限セットを使用します。また、Cryostat にはターゲットワークロードよりも多くの権限がある場合があります。したがって、セキュリティを最適化するために、Red Hat build of Cryostat Operator がインストールされている場所およびターゲットワークロードが配置されている場所とは異なる namespace に Cryostat インスタンスをインストールしてください。

- viii. **Target Namespaces** フィールドで、Cryostat のこのインスタンスにアクセスして操作することを許可するワークロードの namespace を選択します。必要に応じて、Cryostat をインストールしたのと同じ namespace を選択することも、別の namespace を選択することもできます。さらに namespace を追加するには、**+Add Target Namespace** をクリックします。

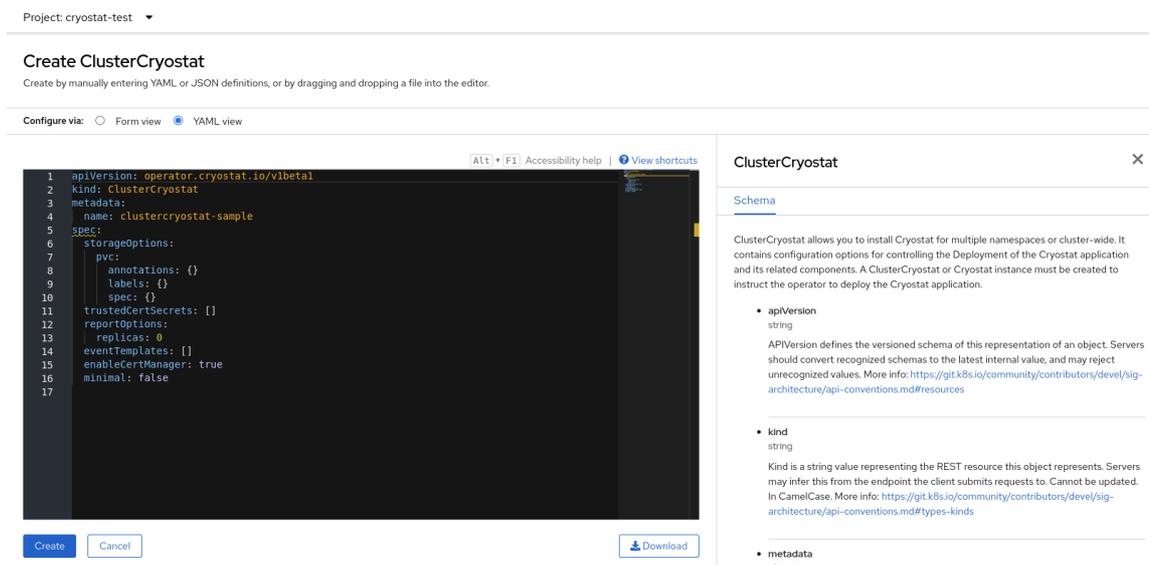


重要

Cryostat インスタンスにアクセスできるユーザーは、その Cryostat インスタンスに認識される namespace 内のすべてのターゲットアプリケーションにアクセスできます。したがって、マルチ namespace の Cryostat インスタンスをデプロイする場合は、監視対象にどの namespace を選択するか、Cryostat をどの namespace にインストールするか、およびどのユーザーにアクセスを許可するかを考慮する必要があります。

あるいは、フォームを使用する代わりに、YAML テンプレートを使用してインスタンスを作成し、追加の設定オプションを指定することもできます。

図2.12 Web コンソールでの YAML テンプレートを使用した Cluster Cryostat インスタンスの作成



5. **Create** をクリックして、Cryostat のマルチ namespace インスタンスの作成プロセスを開始します。
 インスタンスにアクセスするには、Cluster Cryostat インスタンスのすべてのリソースの準備が完了するまで待つ必要があります。

検証

1. Web コンソールのナビゲーションメニューで、**Operators > OperatorHub** に移動します。
2. インストール済み Operator のテーブルから、**Red Hat build of Cryostat Operator** を選択します。
3. **Cluster Cryostat** タブをクリックします。
 Cryostat インスタンスが、インスタンスのテーブルで開き、次の条件が一覧表示されます。
 - **TLSSetupComplete** は **true** に設定されている。
 - **MainDeploymentAvailable** は **true** に設定されている。
 - オプション: レポートジェネレーターサービスを有効にした場合、**ReportsDeploymentAvailable** が表示され、**true** に設定されている。

図2.13 OpenShift の Cluster Cryostat インスタンスの Status 列で True に設定された条件の例

Project: cryostat-test ▾

Installed Operators > Operator details

 Red Hat build of Cryostat
2.4.0 provided by Red Hat Actions ▾

Details YAML Subscription Events All instances Cluster Cryostat Cryostat

ClusterCryostats Create ClusterCryostat

Name ▾ Search by name... /

Name	Kind	Status	Labels	Last updated
 clustercryostat-sample	ClusterCryostat	Conditions: TLSSetupComplete, MainDeploymentAvailable, MainDeploymentProgressing	No labels	🕒 Dec 7, 2023, 3:21 PM

4. オプション: Cluster Cryostat テーブルから、Cryostat インスタンスを選択します。Conditions テーブルに移動して、各条件の詳細情報を確認します。

図2.14 各条件とその基準をリストした Cryostat Conditions 表の例

Cryostat Conditions

Type	Status	Updated	Reason	Message
TLSSetupComplete	True	🕒 Nov 24, 2023, 9:43 PM	AllCertificatesReady	All certificates for Cryostat components are ready.
MainDeploymentAvailable	True	🕒 Dec 7, 2023, 5:00 AM	MinimumReplicasAvailable	Deployment has minimum availability.
MainDeploymentProgressing	True	🕒 Nov 24, 2023, 9:43 PM	NewReplicaSetAvailable	ReplicaSet "cryostat-sample-7dd96f95cd" has successfully progressed.
ReportsDeploymentAvailable	True	🕒 Nov 24, 2023, 9:43 PM	MinimumReplicasAvailable	Deployment has minimum availability.
ReportsDeploymentProgressing	True	🕒 Nov 24, 2023, 9:43 PM	NewReplicaSetAvailable	ReplicaSet "cryostat-sample-reports-7fbbf6d95f" has successfully progressed.

関連情報

- [さまざまなクラスター設定で Cryostat をセットアップするためのベストプラクティス \(Red Hat ナレッジベース\)](#)
- [Web コンソールを使用して Cryostat にアクセスする](#)

第3章 JAVA アプリケーションの設定

Cryostat が Java 仮想マシン (JVM) 上で実行されるターゲットアプリケーションに関する Java Flight Recorder (JFR) データを収集、保存、分析できるようにするには、Cryostat がアプリケーションを検出して接続できるようにアプリケーションを設定する必要があります。

アプリケーションは次のいずれかの方法で設定できます。

- Cryostat エージェントコンポーネントを検出と接続性に使用します。これは Java インストールメンテーションエージェントとして実装され、JVM 上で実行されるアプリケーションのプラグインとして機能します。
- Java Management Extensions (JMX) 接続を許可するようにアプリケーションを設定し、検出には OpenShift Service を、接続性には JMX を使用します。
- 検出には Cryostat エージェントを使用し、接続性には JMX を使用します。

Cryostat エージェント

Red Hat build of Cryostat 2.4 以降では、Cryostat エージェントは、Cryostat サーバーがアプリケーションの JMX ポートの代わりに使用できる HTTP API を提供します。適切に設定された Cryostat エージェントをデプロイするワークロードアプリケーションに接続すると、ターゲットアプリケーションが JMX ポートを公開する必要がなく、Cryostat 機能セットをすべて使用できます。



注記

Red Hat build of Cryostat 2.4 の前は、Cryostat エージェントは、限られた JFR 操作のみをサポートする読み取り専用 HTTP API を提供していました。

Cryostat エージェントの HTTP API は、JMX ポートと比較して次の利点があります。

- API 領域の縮小によるセキュリティの向上
- Cryostat エージェントが Cryostat 検出プラグインとして 2 つの役割をはたすことでデプロイメントの柔軟性を向上

Cryostat エージェントがアプリケーション上でも JMX が設定されていることを検出すると、エージェントはエージェント HTTP API 定義と JMX URL 定義の両方を使用して自身を Cryostat サーバーに公開します。この場合は、任意の設定オプションを使用できます。

ワークロードのアプリケーションで Cryostat エージェントを使用するには、Cryostat エージェントの JAR ファイル (例: `-javaagent:/deployment/app/lib/cryostat-agent.jar`) へのパスを使用して `-javaagent` JVM フラグを渡すようにアプリケーションを設定する必要があります。この設定により、ワークロードアプリケーションの JVM が起動時に Cryostat エージェントを読み込み、初期化できるようになります。Cryostat エージェントの基本的な初期化が完了すると、ワークロードアプリケーションの通常の起動プロセスが通常どおり開始されます。

エージェントの JAR ファイルをワークロードアプリケーションに含めるためのオプション

Cryostat エージェントの JAR ファイルは、さまざまな方法でワークロードアプリケーションに含めることができます。

- 最も簡単な方法は、JAR ファイルを `pom.xml` または `build.gradle` ファイルのアプリケーションの依存関係に追加することです。ビルドツール (Maven または Gradle) は、アプリケーションのビルド出力に含めるための JAR ファイルをダウンロードします。

- **maven-dependency-plugin** などの Maven プラグインを使用して、アプリケーションのビルド出力に JAR ファイルのダウンロードおよび組み込みをより詳細に制御できます。
- JAR ファイルを含めた PersistentVolume ストレージボリュームを作成できます。次に、アプリケーションの **Deployment/DeploymentConfig** を再設定して PersistentVolume をマウントし、**-javaagent:/path/to/persistentvolume/cryostat-agent.jar** を使用します。このタスクを実行する方法は、OpenShift クラスタで有効になっている PersistentVolume プロバイダーのタイプによって異なります。

Cryostat エージェントがアプリケーションコンテナに正常に追加されてロードされると、アプリケーションの **stdout** および **コンソール** ログが Cryostat エージェントからのログメッセージを表示し始めます。

エージェント設定プロパティ

Cryostat エージェントの設定プロパティは、次の 2 つの方法のいずれかで指定できます。

- アプリケーションで JVM システムプロパティフラグを使用します (例: **-Dcryostat.agent.api.writes-enabled=true**)。
- すべての文字を大文字にし、句読点をアンダースコアに置き換えて環境変数を使用します (例: **CRYOSTAT_AGENT_API_WRITES_ENABLED=true**)。

Cryostat エージェントが正常に動作できるようにするには、次のプロパティを設定する必要があります。

cryostat.agent.baseuri	これは、Cryostat エージェントが自身をアドバタイズする Cryostat サーバーバックエンド (つまり、内部 OpenShift サービスオブジェクト) の URL の場所 (https://my-cryostat.my-namespace.svc.cluster.local など) を指定します。
cryostat.agent.callback	これは、Cryostat エージェントインスタンスまたはアプリケーション自体の URL の場所を指定します。Cryostat は、この URL を使用してヘルスチェックを実行し、エージェントにデータを要求します。OpenShift/Kubernetes Downward API を使用して、これを動的に判別できます。詳細は、 status.podIP の Kubernetes Downward API ドキュメント を参照してください。

設定要件に応じて、次のエージェントプロパティを設定することもできます。

cryostat.agent.api.writes-enabled	これは、Cryostat エージェントが書き込み操作を許可するかどうかを示します。デフォルトで false に設定されています。Cryostat エージェントが JFR フライトレコーディングの開始、停止、または削除のリクエストを受け入れるようにするには、このプロパティを true に設定する必要があります。
	<p>注記</p> <p>このプロパティが false に設定されている場合でも、エージェントはフライトレコーディングをリストしたり、個々のレコーディングファイルをダウンロードしたりするリクエストを実行できます。</p>

cryostat.agent.webserver.port	これは、エージェントが HTTP API をバインドするために使用する HTTP ポート番号を指定します (デフォルトでは 9977)。これがアプリケーションまたは別のツールエージェントが使用する既存のポートと競合する場合は、別のポート番号を指定する必要があります。
cryostat.agent.app.name	これは、この Cryostat エージェントインスタンスがどのアプリケーションにアタッチされているかを識別するためのラベルを指定します (デフォルトでは、 cryostat-agent)。これには、Downward API の metadata.name フィールド または metadata.labels'app' フィールドを使用できます。詳細は、Kubernetes Downward API ドキュメント Kubernetes Downward API ドキュメント を参照してください。

リモート Java Management Extensions (JMX) 接続

JMX は、JVM 上で実行されるターゲットアプリケーションを監視および管理できる JVM の標準機能です。Cryostat が JMX を使用するには、JVM の起動時に JMX を有効にして設定する必要があります。これは、ターゲットアプリケーションによる JMX ポートの公開を Cryostat が必要とするためです。

Cryostat は、この JMX ポートを介してターゲットアプリケーションと通信して、JFR レコーディングを開始および停止し、ネットワーク経由で JFR データを取得することで、この JFR データを Cryostat で保存および分析できるようにします。リモートモニタリングには、承認されていないユーザーがアプリケーションにアクセスできないようにセキュリティが必要です。Cryostat がアプリケーションの JFR レコーディングにアクセスする前に、Cryostat は認証情報の入力を求めるプロンプトを表示します。

Cryostat エージェントと JMX のハイブリッド

Cryostat エージェントと JMX の両方を使用するハイブリッドアプローチを使用するようにターゲットアプリケーションを設定できます。このアプローチでは、Cryostat エージェントを使用してターゲットアプリケーションを検出し、JMX を使用して JFR データを Cryostat に公開するため、高い柔軟性が得られます。

たとえば、エージェントを使用すると、特定のポート番号に依存せずにアプリケーションを検出したり、JMX 接続を使用して JFR フライトレコーディングをオンデマンドで開始および停止したりできます。

3.1. CRYOSTAT エージェントを使用したアプリケーションの設定

Java インストルメンテーションエージェントとして実装した Cryostat エージェントを使用すると、Cryostat がアプリケーションを検出し、データを収集し、分析のためにデータを Cryostat に送信できるようにターゲットアプリケーションを設定できます。オプションで、Cryostat エージェントが Cryostat サーバーからの JFR レコーディングの開始、停止、削除リクエストを受け入れることもできます。

Red Hat build of Cryostat 2.4 は、Cryostat エージェントの JAR ファイルに 2 つの異なるバリエーションを配布しています。設定要件に応じて、次のタイプのエージェント JAR ファイルのいずれかを使用できます。

- 自己完結型で、エージェントコードとそのすべての依存関係が含まれるオールインワンの "shaded" JAR ファイル
この "shaded" JAR ファイルは、追加のエージェント JAR ファイルを 1 つだけ含める必要があるため、既存のアプリケーションに組み込むのに最も便利な形式の Cryostat エージェントを提供します。これは、同様のエージェントやツールの一般的な配布パターンです。

- 依存関係のないエージェントコードを含む標準 JAR ファイル
このタイプの JAR ファイルは、エージェントとワークロードアプリケーションの間に依存関係の競合が存在することがわかっている場合に役立ちます。独自のストラテジーを適用して、エージェントとアプリケーションの両方の要件を満たすように、各依存関係の正しいバージョンを提供する場合は、スタンドアロン JAR ファイルを使用できます。



注記

以前のリリースでは、オールインワンの "shaded" JAR ファイルである Cryostat エージェントの1つのディストリビューションが提供されていました。次の手順では、Cryostat 2.4 エージェントの "shaded" JAR ファイルディストリビューションをインストールする方法について説明します。

[Java アプリケーションの設定: Cryostat エージェント](#) で説明されているように、Cryostat 2.4 エージェントは、エージェントの JAR ファイルをワークロードアプリケーションに含めるためのさまざまなオプションをサポートしています。以下の手順では、"shaded" JAR ファイルを **pom.xml** または **build.gradle** ファイルのアプリケーションの依存関係に追加する方法を説明します。

前提条件

- Cryostat Web コンソールにログインしている。
- JDK バージョン 11 以降がインストールされている。

手順

1. Cryostat エージェントをインストールします。アプリケーションのビルドに応じて、次のオプションのいずれかを選択します。
 - Maven の使用:
アプリケーションの **pom.xml** ファイルを Cryostat エージェントの JAR ファイルの情報で更新します。

pom.xml の例

```
<project>
...
<repositories>
  <repository>
    <id>redhat-maven-repository</id>
    <url>https://maven.repository.redhat.com/earlyaccess/all/</url>
  </repository>
</repositories>
...
<build>
  <plugins>
    <plugin>
      <artifactId>maven-dependency-plugin</artifactId>
      <version>3.3.0</version>
      <executions>
        <execution>
          <phase>prepare-package</phase>
          <goals>
            <goal>copy</goal>
          </goals>
        </execution>
      </executions>
    </plugin>
  </plugins>
</build>
```

```

</goals>
<configuration>
  <artifactItems>
    <artifactItem>
      <groupId>io.cryostat</groupId>
      <artifactId>cryostat-agent</artifactId>
      <version>0.3.0.redhat-00001</version>
      <classifier>shaded</classifier>
    </artifactItem>
  </artifactItems>
  <stripVersion>true</stripVersion>
</configuration>
</execution>
</executions>
</plugin>
</plugins>
...
</build>
...
</project>

```

次回アプリケーションをビルドするときに、Cryostat エージェント JAR ファイルが **target/dependency/cryostat-agent-shaded.jar** で利用可能になります。

- Gradle の使用:
build.gradle ファイルを更新します。

build.gradle ファイルの例

```

repositories {
  ...
  maven {
    url "https://maven.repository.redhat.com/earlyaccess/all/"
    credentials {
      username "myusername"
      password "mytoken"
    }
  }
}

```

エージェント JAR ファイルをアプリケーションにパッケージ化する方法は、ビルドに使用する Gradle プラグインによって異なります。たとえば、Jib プラグインを使用している場合は、次のように **build.gradle** ファイルを更新します。

build.gradle ファイルの例

```

plugins {
  id 'java'
  id 'application'
  id 'com.google.cloud.tools.jib' version '3.3.1'
  id 'com.ryandens.javaagent-jib' version '0.5.0'
}
...

```

```
dependencies {
  ...
  javaagent 'io.cryostat:cryostat-agent:0.3.0.redhat-00001:shaded'
```

2. Docker ファイルを更新します。次の例では、**JAVA_OPTS** 環境変数を使用して、関連する JVM 情報を渡します。

例

```
...
COPY target/dependency/cryostat-agent.jar /deployments/app/
...
ENV JAVA_OPTS="-javaagent:/deployments/app/cryostat-agent-shaded.jar"
```

3. アプリケーション固有のコンテナイメージを再構築します。

```
docker build -t docker.io/myorg/myapp:latest -f src/main/docker/Dockerfile
```

4. Crio エージェントの設定に必要な JVM システムプロパティまたは環境変数を指定するには、更新されたイメージをプッシュし、アプリケーションのデプロイメントを変更します。

例

```
apiVersion: apps/v1
kind: Deployment
...
spec:
  ...
  template:
    ...
    spec:
      containers:
        - name: sample-app
          image: docker.io/myorg/myapp:latest
          env:
            - name: CRYOSTAT_AGENT_APP_NAME
              value: "myapp"
              # Replace this with the Kubernetes DNS record
              # for the Crio Service
            - name: CRYOSTAT_AGENT_BASEURI
              value: "http://cryostat.mynamespace.mycluster.svc:8181"
            - name: POD_IP
              valueFrom:
                fieldRef:
                  fieldPath: status.podIP
            - name: CRYOSTAT_AGENT_CALLBACK
              value: "http://$(POD_IP):9977" ❶
              # Replace "abcd1234" with a base64-encoded authentication token
            - name: CRYOSTAT_AGENT_AUTHORIZATION ❷
              value: "Bearer abcd1234"
            - name: CRYOSTAT_AGENT_API_WRITES_ENABLED ❸
              value: true
          ports:
            - containerPort: 9977
```

```

protocol: TCP
resources: {}
restartPolicy: Always
status: {}

```

- <1>: ポート番号 **9977** は、Cryostat のリクエストを処理する内部 Web サーバーに対してエージェントが公開するデフォルトの HTTP ポートです。エージェントがインストールされているターゲットアプリケーションとこのポート番号が競合する場合は、番号を変更できます。
- <2>: **CRYOSTAT_AGENT_AUTHORIZATION** 値は、エージェントが自身の存在をアドバタイズするために、または JFR データをプッシュするために Cryostat への API リクエストに含める認証情報を示します。また、この目的で Kubernetes **Service Account** を作成し、**abcd1234** を、サービスアカウントに関連付けられている Base64 エンコードされた認証トークンに置き換えることもできます。
- <3>: **CRYOSTAT_AGENT_API_WRITES_ENABLED** 変数はデフォルトで **false** に設定されます。Cryostat エージェントが Cryostat サーバーからの JFR フライト記録の開始、停止、または削除のリクエストを受け入れるようにするには、この変数を **true** に設定する必要があります。

3.2. JMX 接続を使用したアプリケーションの設定

Cryostat がターゲット Java アプリケーションを検出して通信できるようにする場合、リモート Java Management Extensions (JMX) 接続を許可するようにアプリケーションを設定できます。

前提条件

- Cryostat Web コンソールにログインしている。
- プロジェクトに Cryostat インスタンスを作成している。

手順

1. リモート JMX 接続を有効にするには、次の手順を実行します。
 - a. アプリケーションで、次の Java システムプロパティを定義します。

```
-Dcom.sun.management.jmxremote.port=<port_num>
```



注記

ターゲットアプリケーションを再構築せずに -

Dcom.sun.management.jmxremote.port=<port_num> プロパティを追加するには、アプリケーションで **JAVA_OPTS_APPEND** 環境変数を設定します。**JAVA_OPTS_APPEND** は、Red Hat Universal Base Images (UBI) のみが使用する環境変数です。

Red Hat UBI を使用してアプリケーションイメージをビルドする場合は、ビルド時にアプリケーション Docker ファイルで **JAVA_OPTS_APPEND** 変数を設定するか、ランタイム時に次のコマンドを実行して、**JAVA_OPTS_APPEND** 変数を設定します。

```
oc set env deployment <name> JAVA_OPTS_APPEND="..."
```

Red Hat UBI を使用してアプリケーションイメージをビルドしない場合は、ベースイメージのドキュメントを参照し、ビルド時またはランタイム時に Java システムプロパティを追加する方法を確認してください。

- b. アプリケーションへのトラフィックを許可して、アプリケーションがリモート JMX 接続をリッスンするように指定します。Red Hat OpenShift Service を使用し、そのリモート JMX ポートに次の値を指定します。

service.yaml の例

```
apiVersion: v1
kind: Service
...
spec:
  ports:
    - name: "jfr-jmx"
      port: 9091
      targetPort: 9091
  ...
```

2. リモート JMX 接続をセキュリティー保護します。
 - a. アプリケーションで、リモート JMX 接続の認証と SSL/TLS を有効にして設定します。

```
-Dcom.sun.management.jmxremote.port=<port_num>

# enable JMX authentication
-Dcom.sun.management.jmxremote.authenticate=true

# define users for JMX auth
-Dcom.sun.management.jmxremote.password.file=</path/to/jmxremote.password>

# set permissions for JMX users
-Dcom.sun.management.jmxremote.access.file=</path/to/jmxremote.access>

# enable JMX SSL
-Dcom.sun.management.jmxremote.ssl=true

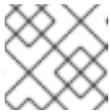
# enable JMX registry SSL
-Dcom.sun.management.jmxremote.registry.ssl=true
```

```
# set your SSL keystore
-Djavax.net.ssl.keyStore=</path/to/keystore>

# set your SSL keystore password
-Djavax.net.ssl.keyStorePassword=<password>
```

- b. アプリケーションの TLS 証明書を信頼するように Croyostat を設定します。Croyostat アプリケーションと同じ namespace にアプリケーションのシークレットを作成し、そのシークレットを参照するように Croyostat を設定します。証明書のシークレットを作成するには、次のコマンドを実行します。

```
oc create secret generic myapp-cert --from-file=tls.crt=/path/to/cert.pem
```



注記

証明書は **.pem** ファイル形式である必要があります。

- c. Croyostat インスタンスの作成時に、信頼できる TLS 証明書のリストにシークレットを追加します。詳細は、[TLS 証明書の設定](#) を参照してください。
- d. パスワード認証以外の方法で Croyostat がアプリケーションに接続していることをアプリケーションが検証できるようにするために、TLS クライアント認証を有効にします。

```
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true
-Djavax.net.ssl.trustStore=</path/to/truststore>
-Djavax.net.ssl.trustStorePassword=<password>
```



注記

TLS クライアント認証には、Red Hat OpenShift の cert-manager Operator が必要です。

- e. リモート JMX 接続に TLS クライアント認証を使用する場合、アプリケーションのトラストストアに Croyostat 証明書が含まれている必要があります。Croyostat Operator の cert-manager の統合により、Croyostat デプロイメント用の自己署名証明書が作成されます。この証明書は **<croyostat>-tls** シークレットにあります。<croyostat> は作成した Croyostat インスタンスの名前です。



注記

cert-manager Operator は、このシークレットに Java キーストアトラストストアも配置します。

このトラストストアをアプリケーションデプロイメントにマウントするには、次のコマンドを実行します。"<myapp>" はアプリケーションデプロイメントの名前に置き換え、"<croyostat>" は Croyostat インスタンスの名前に置き換えてください。

```
oc set volumes deploy <myapp> --add --name=truststore \
  --secret-name=<croyostat>-tls --sub-path=truststore.p12 \
  --mount-path=/var/run/secrets/<myapp>/truststore.p12
```

- f. Cryostat Operator はトラストストアのパスワードを生成します。このパスワードは、**<cryostat>-keystore** シークレットにあります。これをアプリケーションのデプロイメントで環境変数としてマウントするには、次のコマンドを実行します。

```
oc set env deploy <myapp> --from='secret/<cryostat>-keystore'
```

- g. コンテナの Java 引数を設定します。以下のコマンドを実行します。

```
-Dcom.sun.management.jmxremote.ssl.need.client.auth=true  
-Djavax.net.ssl.trustStore=/var/run/secrets/<myapp>/truststore.p12  
-Djavax.net.ssl.trustStorePassword="$(KEYSTORE_PASS)"
```



警告

Cryostat とアプリケーションをテスト環境にデプロイした場合は、JMX または TLS 認証を使用せずにターゲットアプリケーションを設定することもできます。これは、次の Java システムプロパティのセットを使用して行うことができますが、この設定はセキュアではないため、推奨されません。

```
-Dcom.sun.management.jmxremote.port=<port_num>  
-Dcom.sun.management.jmxremote.ssl=false  
-Dcom.sun.management.jmxremote.authenticate=false
```

関連情報

- [TLS 証明書の設定](#)
- [JMX クレデンシャルの保存および管理](#)

3.3. CRYOSTAT エージェントと JMX 接続を使用したアプリケーションの設定

Cryostat エージェントと Java Management Extensions (JMX) 接続を組み合わせ使用してターゲットアプリケーションを検出および通信するように、Java 仮想マシン (JVM) 上で実行されるターゲットアプリケーションを設定できます。

Cryostat エージェントを使用してターゲットアプリケーションを検出および通信し、JMX を使用して Java Flight Recorder (JFR) データを公開します。

Cryostat エージェント自体について Cryostat と通信し、HTTP 経由ではなく JMX 経由でエージェントに到達できるように、Cryostat エージェントを設定する必要があります。

前提条件

- Cryostat Web コンソールにログインしている。
- プロジェクトに Cryostat インスタンスを作成している。

手順

1. Cryostat エージェントをインストールします。Maven を使用するアプリケーションビルドの場合は、Cryostat エージェントの JAR ファイルの情報を使用してアプリケーションの **pom.xml** ファイルを更新します。

pom.xml ファイルの例

```

<project>
...
<repositories>
  <repository>
    <id>redhat-maven-repository</id>
    <url>https://maven.repository.redhat.com/earlyaccess/all/</url>
  </repository>
</repositories>
...
<build>
  <plugins>
    <plugin>
      <artifactId>maven-dependency-plugin</artifactId>
      <version>3.3.0</version>
      <executions>
        <execution>
          <phase>prepare-package</phase>
          <goals>
            <goal>copy</goal>
          </goals>
          <configuration>
            <artifactItems>
              <artifactItem>
                <groupId>io.cryostat</groupId>
                <artifactId>cryostat-agent</artifactId>
                <version>0.3.0.redhat-00001</version>
                <classifier>shaded</classifier>
              </artifactItem>
            </artifactItems>
            <stripVersion>true</stripVersion>
          </configuration>
        </execution>
      </executions>
    </plugin>
  </plugins>
...
</build>
...
</project>

```

2. アプリケーションのデプロイメントを変更します。

例

```

apiVersion: apps/v1
kind: Deployment
...

```

```

spec:
  ...
  template:
    ...
    spec:
      containers:
        - name: sample-app
          image: docker.io/myorg/myapp:latest
          env:
            - name: CRYOSTAT_AGENT_APP_NAME
              value: "myapp"
              # Replace this with the Kubernetes DNS record
              # for the Crio Service
            - name: CRYOSTAT_AGENT_BASEURI
              value: "http://criostat.mynamespace.mycluster.svc:8181"
            - name: POD_IP
              valueFrom:
                fieldRef:
                  fieldPath: status.podIP
            - name: CRYOSTAT_AGENT_CALLBACK
              value: "http://$(POD_IP):9977"
            - name: CRYOSTAT_AGENT_AUTHORIZATION
              # Replace "abcd1234" with a base64-encoded authentication token
              value: "Bearer abcd1234"
              # This environment variable is key to the Crio agent
              # and JMX "hybrid" setup.
              # Set the Crio agent to register itself with Crio
              # as reachable through JMX, rather than reachable through HTTP.
            - name: CRYOSTAT_AGENT_REGISTRATION_PREFER_JMX
              value: "true"
              # Configure the application to load the agent JAR file and
              # to enable JMX, so that the Crio agent can register
              # itself as reachable through JMX.
              # To configure authentication and SSL/TLS for the JMX
              # connections, see <1>.
            - name: JAVA_OPTS
              value: >-
                -javaagent:/deployments/app/criostat-agent-shaded.jar
                -Dcom.sun.management.jmxremote.port=9091 ①
          ports:
            - containerPort: 9977
              protocol: TCP
          resources: {}
          restartPolicy: Always
        status: {}

```

<1>: JMX 接続の認証と SSL/TLS を設定し、その他の設定オプションを表示するには、[JMX 接続を使用したアプリケーションの設定](#) を参照してください。

- Crio がターゲットアプリケーションを検出し、Crio エージェントに接続できるようにするために、アプリケーション **Service** を設定します。

例

```

apiVersion: v1
kind: Service

```

```
...
spec:
  ports:
    - name: "jfr-jmx"
      port: 9091
      targetPort: 9091
    - name: "cryostat-agent"
      port: 9977
      targetPort: 9977
  ...
```

関連情報

- [Cryostat エージェントを使用したアプリケーションの設定](#)
- [JMX 接続を使用したアプリケーションの設定](#)

第4章 CRYOSTAT を使用した JFR レコーディングの作成

Cryostat を使用すると、コンテナ化されたアプリケーションで JVM のパフォーマンスを監視する JDK Flight Recorder (JFR) レコーディングを作成できます。さらに、アクティブな JFR レコーディングのスナップショットを作成して、ターゲット JVM アプリケーションの特定の時点までの収集データをキャプチャーできます。

4.1. CRYOSTAT WEB コンソールでの JFR レコーディングの作成

コンテナ化されたアプリケーション内にある JVM のパフォーマンスを監視する JFR レコーディングを作成できます。JFR レコーディングの作成後は、JFR を起動して、ヒープおよび非ヒープのメモリー使用量など、JVM のリアルタイムデータをキャプチャーできます。

前提条件

- OperatorHub オプションを使用して、Cryostat 2.4 を Red Hat OpenShift にインストールしている。
- Red Hat OpenShift プロジェクトに Cryostat インスタンスを作成した。
- Cryostat Web コンソールにログインしている。
 - Red Hat OpenShift Web コンソールを使用して Cryostat アプリケーションの URL を取得できます。

手順

1. Cryostat Web コンソールの **Dashboard** パネルで、**Target** リストからターゲット JVM を選択します。



注記

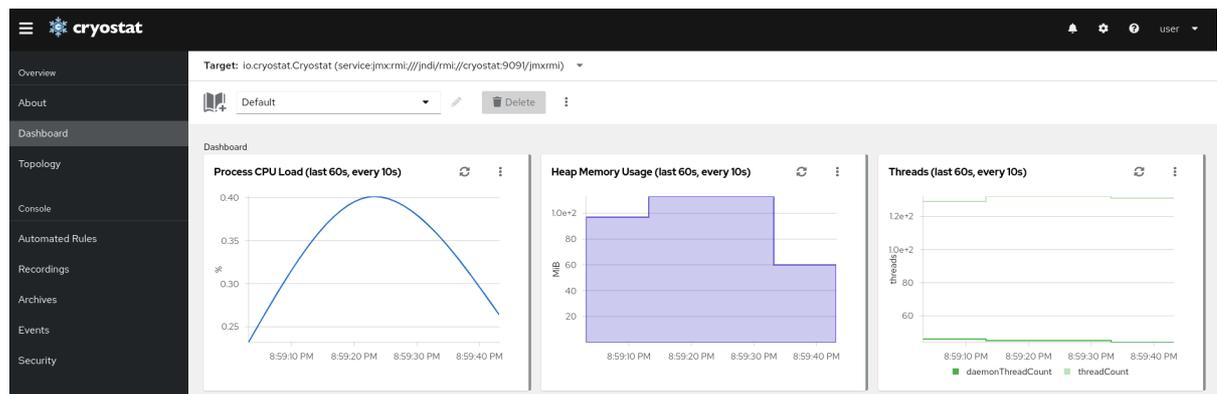
ターゲットアプリケーションの設定方法によっては、ターゲット JVM が JMX 接続またはエージェント HTTP 接続を使用している場合があります。ターゲットアプリケーションの設定の詳細は、[Java アプリケーションの設定](#) を参照してください。



重要

ターゲット JVM がエージェント HTTP 接続を使用しており、ターゲットアプリケーションが Cryostat エージェントをロードするように設定した場合は、**cryostat.agent.api.writes-enabled** プロパティを **true** に指定していることを確認してください。そうしないと、Cryostat エージェントは、JFR レコーディングの開始および停止のリクエストを受け入れることができません。

図4.1 Cryostat インスタンスの Target JVM 選択例



2. オプション: Dashboard パネルで、ターゲット JVM を作成できます。Target リストから **Create Target** をクリックします。Create Custom Target ウィンドウが開きます。
 - a. **Connection URL** フィールドに、JVM の Java Management Extension (JMX) エンドポイントの URL を入力します。
 - b. オプション: 指定した **Connection URL** が有効かどうかをテストするには、**Click to test** をクリックし、サンプルノードイメージをテストします。**Connection URL** に問題がある場合は、問題の説明とトラブルシューティングのガイダンスを示すエラーメッセージが表示されます。
 - c. オプション: **Alias** フィールドに、JMX Service URL のエイリアスを入力します。
 - d. **Create** をクリックします。

図4.2 Create Custom Target ウィンドウ

The screenshot shows the 'Create Custom Target' dialog box in the Cryostat web console. The dialog contains the following fields and options:

- 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:** (Empty field)
- Connection Nickname:** (same as Connection URL if not specified).
- JMX Credential Options:** Creates credentials that Cryostat uses to connect to target JVMs over JMX.
- Buttons:** Create, Cancel, and a 'CT <Name>' button.
- Instructions:** Click on the sample node above to test custom target definition.

3. Cryostat Web コンソールのナビゲーションメニューから、**Recordings** をクリックします。
4. オプション: ターゲット JVM の設定方法によっては、Web コンソールで **Authentication Required** ダイアログが開く場合があります。**Authentication Required** ダイアログボックスで **Username** および **Password** を入力します。**Save** をクリックして、認証情報をターゲット JVM に提供します。

図4.3 Cryostat Authentication Required ウィンドウの例



注記

選択したターゲット JMX で、JMX 接続に対して Secure Socket Layer (SSL) 証明書が有効になっている場合は、プロンプトが表示されたらその証明書を追加する必要があります。

Cryostat は、ターゲット JVM アプリケーションのクレデンシャルを暗号化し、Red Hat OpenShift の永続ボリューム要求 (PVC) に格納されているデータベースに保存します。[認証情報の保存および管理](#) (Cryostat を使用した JFR レコーディングの管理) を参照してください。

5. **Active Recordings** タブで、**Create** をクリックします。

図4.4 アクティブなレコーディングの作成例

Target: quarkus-test-7c95cc5b-vidzq (service:jmxrmi://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

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

6. **Custom Flight Recording** タブで、以下を行います。

- a. **Name** フィールドに、作成するレコーディングの名前を入力します。無効な形式で名前を入力すると、Web コンソールにエラーメッセージが表示されます。
- b. Cryostat が既存のレコーディングを自動的に再起動するようにする場合は、**Restart if recording already exists** のチェックボックスを選択します。



注記

すでに存在する名前を入力し、**Restart if recording already exists** を選択しないと、**Create** ボタンをクリックしても Cryostat はカスタムレコーディングの作成を拒否します。

- c. **Duration** フィールドで、指定された期間後にこのレコーディングを停止するか、停止せずに継続的に実行するかを選択します。レコーディングの停止後に Cryostat が新しい JFR レコーディングを自動的にアーカイブするようにするには、**Archive on Stop** をクリックします。
- d. **Template** フィールドで、レコーディングに使用するテンプレートを選択します。

次の例は、継続的な JVM モニタリングを示しています。これは、**Continuous** フィールドの上から **Duration** を選択することで有効にできます。この設定では、手動でレコーディングを停止するまでレコーディングが継続されます。また、この例では、**Template** フィールドで **Profiling** テンプレートが選択されています。これにより、トラブルシューティングを目的として、JFR レコーディングに追加の JVM 情報が提供されます。

図4.5 カスタムフライトレコーディングの作成例

Target: io.cryostat.Cryostat (service:jmx:rmi:///jndi/rmi://cryostat9091/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

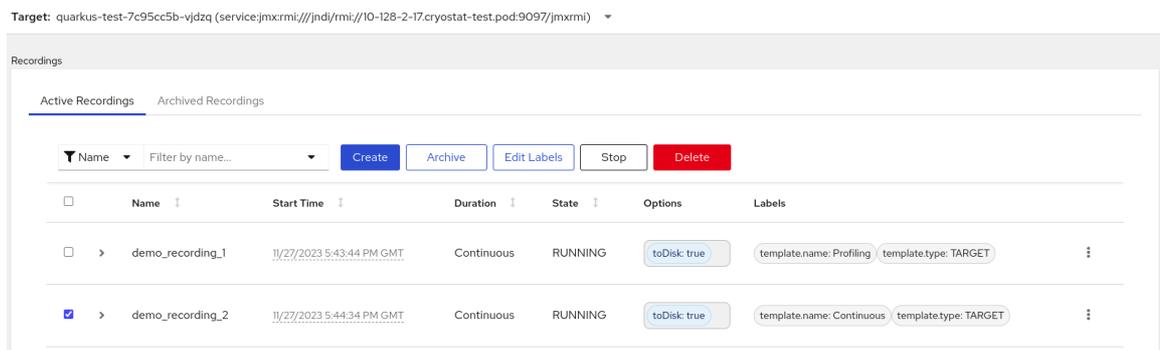
> Show advanced options

Create
Cancel

7. その他のオプションにアクセスするには、次の展開可能なハイパーリンクをクリックします。
 - **Show advanced options:** JFR レコーディングをカスタマイズするための追加オプションを選択できます。
 - **Show metadata options:** カスタムラベルとメタデータを JFR レコーディングに追加できます。
8. **Create** をクリックして、JFR レコーディングを作成します。**Active Recordings** タブが開き、JFR レコーディングがリスト表示されます。

アクティブな JFR レコーディングが、コンテナ化されたアプリケーション内のターゲット JVM の場所でデータ収集を開始します。JFR レコーディングに固定期間を指定した場合、ターゲット JVM は設定された固定期間に達するとレコーディングを停止します。それ以外の場合は、手動でレコーディングを停止する必要があります。
9. **オプション:** **Active Recording** タブで、レコーディングを停止することもできます。
 - a. JFR レコーディングの名前の横にあるチェックボックスを選択します。Cryostat Web コンソールで、**Active Recordings** タブのツールバーにある **Stop** ボタンが有効になります。
 - b. **Stop** をクリックします。JFR が **STOPPED** ステータスになり、ターゲット JVM のモニタリングが停止します。JFR はその後も **Active Recording** タブに表示されます。

図4.6 アクティブなレコーディングの停止例



重要

以下の状況が発生すると、JFR レコーディングのデータが失われる可能性があります。

- ターゲット JVM が失敗する
- ターゲット JVM が再起動する
- ターゲット JVM の Red Hat OpenShift デプロイメントがスケールダウンされる

JFR レコーディングをアーカイブして、JFR レコーディングのデータが失われないようにします。

関連情報

- [SSL 証明書のアップロード](#) (Cryostat を使用して JFR レコーディングの管理) を参照してください。
- [JDK Flight Recorder \(JFR\) レコーディングのアーカイブ](#) (Cryostat を使用した JFR レコーディングの管理) を参照してください。

4.2. アクティブなレコーディングからのスナップショット作成

アクティブな JFR レコーディングのスナップショットを作成して、ターゲット JVM アプリケーションの特定の時点までの収集データをキャプチャできます。スナップショットは、実行中の JFR レコーディングの特定の時間セグメントの開始点と終了点を持つチェックポイントマーカーのようなものです。

スナップショットは、ターゲット JVM アプリケーションのメモリーに保存されます。これはアーカイブとは異なり、Cryostat はアーカイブをクラウドストレージディスクに保存します。アーカイブは、JFR レコーディングのデータを保存するためのより永続的なソリューションです。

アクティブな JFR レコーディング間でさまざまな設定変更を試す場合は、レコーディングのスナップショットを作成するとよいでしょう。

JFR レコーディングのスナップショットを作成すると、Cryostat は **snapshot -<snapshot_number>** という名前の新しいターゲット JVM を作成します。この場合、**<snapshot_number>** は Cryostat がスナップショットに自動的に割り当てる番号です。

ターゲット JVM はスナップショットをアクティブなレコーディングとして認識します。Cryostat は、JFR スナップショットを **STOPPED** 状態に設定します。これは、JFR スナップショットがターゲット

JVM に新しいデータをレコーディングしないことを意味します。JFR 設定によっては、アクティブな JFR レコーディングは、作成されたスナップショットの数にかかわらず、ターゲット JVM の監視を継続できます。



注記

ターゲット JVM アプリケーションの継続的な監視用に設定した JFR レコーディングでは、JFR レコーディングデータの損失を回避するために、レコーディングのアーカイブを作成してください。

JFR レコーディングデータを保存するために通常のスナップショットを作成することを選択した場合、ターゲット JVM アプリケーションが、古いレコーディングデータを新しいレコーディングデータに置き換えて、データストレージ領域の一部を解放する場合があります。

前提条件

- Cryostat インスタンスの認証情報を入力した。
- ターゲット JVM レコーディングを作成し、認証情報を入力しており、**Recordings** メニューにアクセスできる。[JDK Flight Recorder \(JFR\) レコーディングの作成](#) (Cryostat を使用した JFR レコーディングの作成) を参照してください。

手順

1. **Active Records** タブで、**Create** ボタンをクリックします。Web コンソールに新しいウィンドウが開きます。

図4.7 アクティブなレコーディングの作成例

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

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

2. **Snapshot Recording** タブをクリックします。

図4.8 スナップショットレコーディングの作成例

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

Recordings > Create Recording

Custom Flight Recording Snapshot Recording

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.

Create Cancel

3. **Create** をクリックします。**Active Recordings** タブが開き、JFR スナップショットのレコーディングが一覧表示されます。次の例は、**snapshot-3** という名前の JFR スナップショットのレコーディングを示しています。

図4.9 完成したスナップショットレコーディングの例

Active Recordings		Archived Recordings				
Name	Filter by name...	Create	Archive	Edit Labels	Stop	Delete
<input type="checkbox"/>	Name ↑	Start Time ↑	Duration ↑	State ↑	Options	Labels
<input type="checkbox"/>	> demo_recording_1	11/27/2023 5:43:44 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Profiling template.type: TARGET
<input type="checkbox"/>	> demo_recording_2	11/27/2023 5:44:34 PM GMT	Continuous	RUNNING	toDisk: true	template.name: Continuous template.type: TARGET
<input type="checkbox"/>	> snapshot-3	11/27/2023 5:43:44 PM GMT	1724s	STOPPED	toDisk: true	-



注記

スナップショットは、アクティブなレコーディングのリストから **snapshot** の接頭辞で識別できます。

次のステップ

- JFR スナップショットレコーディングのアーカイブについては、[JDK Flight Recorder \(JFR\) のレコーディングのアーカイブ](#) を参照してください。

4.3. JFR レコーディングのラベル

Cryostat 2.4 で JDK Flight Recorder (JFR) レコーディングを作成する場合、一連のキーと値のラベルのペアを指定することで、記録にメタデータを追加できます。

さらに、ターゲット JVM 内にある JFR レコーディングにカスタムラベルを割り当てることができるため、JFR レコーディングを簡単に識別してより適切に管理できます。

一般的なレコーディングラベルの使用例を以下に示します。

- JFR レコーディングにメタデータを割り当てる。
- 同一のラベルを含むレコーディングに対してバッチ操作を実行する。
- レコーディングに対してクエリーを実行するときにラベルを使用する。

Cryostat を使用すると、コンテナ化されたアプリケーションで JVM のパフォーマンスを監視する JFR レコーディングを作成できます。さらに、アクティブな JFR レコーディングのスナップショットを作成して、ターゲット JVM アプリケーションの特定の時点までの収集データをキャプチャーできます。

4.3.1. JFR レコーディングへのラベルの追加

Cryostat 2.4 で JFR レコーディングを作成する場合、ラベルを使用して、キーと値のラベルペアを含むメタデータをレコーディングに追加できます。

Cryostat は、作成された JFR レコーディングにデフォルトのレコーディングラベルを適用します。これらのデフォルトラベルは、Cryostat が JFR レコーディングの作成に使用したイベントテンプレートに関する情報を取得します。

カスタムラベルを JFR レコーディングに追加すると、特定の JFR レコーディングを識別したり、同じラベルが適用されたレコーディングでバッチ操作を実行したりするなど、ニーズを満たす特定のクエリーを実行できます。

前提条件

- Cryostat Web コンソールにログインしている。
- Cryostat インスタンスのターゲット JVM を作成または選択している。

手順

1. Cryostat Web コンソールから、**Recordings** をクリックします。
2. **Active Recordings** タブで、**Create** をクリックします。
3. **Custom Flight Recording** タブで、**Show metadata options** を展開します。



注記

Custom Flight Recording タブでは、アスタリスクが付いている必須フィールドに入力する必要があります。

4. **Add label** をクリックします。

図4.10 Custom Flight Recording タブに表示される Add Label ボタン

The screenshot shows the 'Custom Flight Recording' configuration page. It has two tabs: 'Custom Flight Recording' (selected) and 'Snapshot Recording'. Below the tabs is a descriptive text: '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.' The form contains several sections:

- Name**: A text input field with a red asterisk. Below it is a checkbox 'Restart if recording already exists' and a note: 'Enter a recording name. This will be unique within the target JVM.'
- Duration**: A section with a checkbox 'Continuous' and a checked checkbox 'Archive on Stop'. Below is a text input field with '30' and a 'Seconds' dropdown menu. A note says: 'Time before the recording is automatically stopped and copied to archive.'
- Template**: A dropdown menu with 'Select a Template' and a note: 'The Event Template to be applied in this recording'.
- Hide metadata options**: A collapsed section.
- Labels**: A section with a note: 'Labels with key `template.name` and `template.type` are set by Cryostat and will be overwritten if specified.' Below this is a blue '+ Add Label' button.
- Show advanced options**: A link to expand more options.

 At the bottom of the form are 'Create' and 'Cancel' buttons.

5. 表示された **Key** フィールドと **Value** フィールドに値を入力します。たとえば、レコーディングに関する問題を報告する場合は、**Key** フィールドに理由を入力し、**Value** フィールドに問題の種類を入力できます。
6. **Create** をクリックして、JFR レコーディングを作成します。レコーディングは、指定されたレコーディングラベルおよびカスタムラベルとともに **Active Recordings** タブに表示されます。

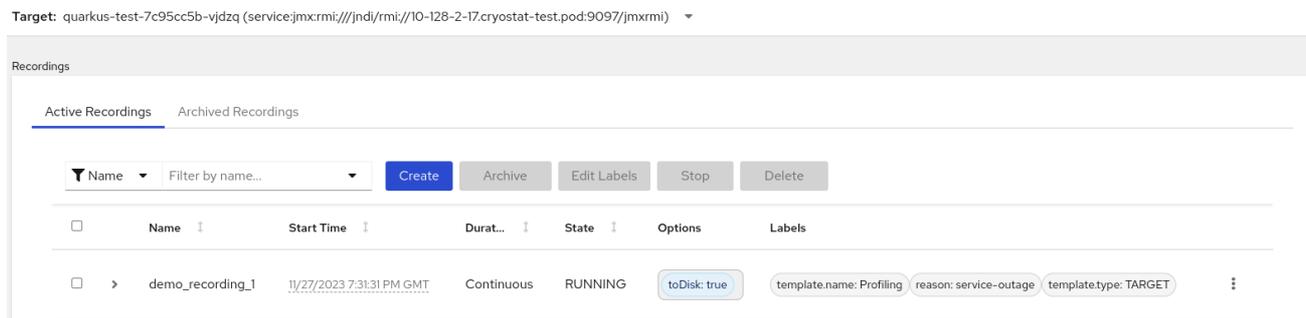
ヒント

Archives メニューから、アーカイブされた JFR レコーディングにアクセスできます。 [Cryostat アーカイブの場所への JFR レコーディングのアップロード](#) (Cryostat を使用した JFR レコーディングの管理) を参照してください。

例

次の例は、2つのデフォルトのレコーディングラベル **template.name: Profiling** および **template.type: TARGET** と、1つのカスタムラベル **reason:service-outage** を示しています。

図4.11 定義済みのレコーディングラベルとカスタムラベルを使用したアクティブなレコーディングの例



4.3.2. JFR レコーディングのラベルの編集

Crioostat Web コンソールで、**Recordings** メニューに移動して、JFR レコーディングのラベルとそのメタデータを編集できます。アーカイブにアップロードした JFR レコーディングのラベルとメタデータを編集することもできます。

前提条件

- Crioostat Web コンソールにログインしている。
- JFR レコーディングを作成し、このレコーディングにラベルを添付する。

手順

1. Crioostat Web コンソールで、**Recording** メニューをクリックします。
2. **Active Recordings** タブから、JFR レコーディングを見つけて、その横にあるチェックボックスを選択します。
3. **Edit Labels** をクリックします。Crioostat Web コンソールで **Edit Recording Label** ペインが開きます。このペインを使用して、JFR レコーディングのラベルを追加、編集、または削除できます。

ヒント

各レコーディングの横にあるチェックボックスを選択すると、複数の JFR レコーディングを選択できます。同じラベルを含むレコーディングを一括編集する場合、または複数のレコーディングに新しい同一のラベルを追加する場合は、**Edit Labels** ボタンをクリックします。

4. **オプション: Edit Recording Labels** ウィンドウから、次のいずれかのアクションを実行できます。
 - a. **Add** をクリックして、ラベルを作成します。
 - b. ラベルの横にある **X** をクリックして、ラベルを削除します。
 - c. フィールドの内容を変更して、ラベルを編集します。内容を編集すると、編集したことを示す緑色のチェックマークがフィールドに表示されます。

5. **Save** をクリックします。
6. **オプション**: 次の手順を実行して、JFR レコーディングをそのラベルとともにアーカイブできます。
 - a. レコーディングの名前の横にあるチェックボックスを選択します。
 - b. **Archive** ボタンをクリックします。レコーディングは **Archived Recordings** タブで見つけることができます。
レコーディングをラベル付きでアーカイブすると、後でレコーディングを見つける場合に検索機能を強化できます。Cryostat アーカイブにアップロードした任意のレコーディングにラベルを追加することもできます。



注記

Cryostat は、アーカイブされたレコーディングの存続期間中、レコーディングのラベルを保持します。

検証

- **Active Recordings** タブで、レコーディングの **Labels** セクションに変更内容が表示されていることを確認します。

関連情報

- [JDK フライトレコーダー \(JFR\) のレコーディングのアーカイブ \(Cryostat を使用した JFR レコーディングの管理\)](#)

改訂日時: 2024-03-06