



Red Hat build of Cryostat 2

Red Hat build of Cryostat 2.0 のリリースノート

リリースノート

Red Hat build of Cryostat 2 Red Hat build of Cryostat 2.0 のリリースノート

リリースノート

Enter your first name here. Enter your surname here.

Enter your organisation's name here. Enter your organisational division here.

Enter your email address here.

法律上の通知

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Release_notes_for_the_Red_Hat_build_of_Cryostat_2.0.ent file |.

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 2.0 のリリースノート」には、Cryostat 2.0 の新機能の概要と、潜在的な既知の問題と考えられる回避策のリストが記載されています。

目次

はじめに	4
第1章 多様性を受け入れるオープンソースの強化	5
RED HAT ドキュメントへのフィードバック (英語のみ)	6
第2章 CRYOSTAT のサポートポリシー	7
第3章 新機能	8
Cryostat アプリケーションの API パーミッション	8
自動化ルール	8
cert-manager API	8
デプロイメントシナリオのツリービュー	9
エンドポイントリソースの監視	9
TargetEventGetHandler でのイベント検索	9
HTTP API の動作変更	10
新しい環境変数	10
新しいハンドラー	10
Cryostat で利用可能な新規イベントテンプレート	10
事前設定された Grafana ダッシュボードイメージ	11
応答ステータスコードメッセージ	11
クライアント側の接続プロトコルのサポート	11
第4章 機能拡張	12
Cryostat Web コンソールの GUI	12
Cryostat Operator トポロジービュー	12
Cryostat Operator コントローラマネージャー	12
Cryostat クラスタ ConsoleLink namespace	12
カスタムイベントテンプレート	12
サポートされる形式による SSL/TLS 証明書のエンコード	12
Fabric8 Kubernetes および OpenShift クライアント	13
Grafana コンテナのバージョン	13
Grafana エラーメッセージ	13
ヘルスチェックリソースの定義	13
initialization-resource アノテーション	13
OLM バンドル記述子	13
SCC (Security Context Constraints)	13
ServiceRef 定義	13
サブプロセス管理	14
サブプロセスレポートの生成	14
truststore 環境変数	14
TargetConnectionManager ハンドラー	14
Vertx サーバーの応答管理	14
WebSocket	15
第5章 サポートされない機能および非推奨の機能	16
ContainerJFR バリエーション	16
イベント指定子の文字列構文	16
対話モードのコマンド	16
TTY/TCP 実行モード	16
WebSocket コマンド	16
WebSever IO クリーンアップ	16

第6章 既知の問題	17
OpenShift SDN デフォルト CNI ネットワークプロバイダー	17
アーカイブアップロード API のメモリー不足 (OOM)	17
統合された Grafana コンポーネントのファイルアップロード制限	18
第7章 セキュリティー修正	19
jsoup Java ライブラリー	19

はじめに

Cryostat は、OpenShift クラスター内で実行している Java ワークロードをデスクトップの JDK Mission Control (JMC) アプリケーションに接続するコンテナネイティブな Java アプリケーションです。

Cryostat 2.0 の主な機能は、OpenShift Operator を使用して OpenShift Container Platform Web コンソールの OperatorHub で Cryostat をインストールし、デプロイできることです。Cryostat をインストールおよびデプロイしたら、フル機能を備えた Cryostat インスタンスにアクセスし、**Red Hat build of Cryostat 2.0 のリリースノート** に記載されている追加機能を確認できます。



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、Red Hat カスタマーポータル[の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。

第1章 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。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するご意見やご感想をお寄せください。フィードバックをお寄せいただくには、ドキュメントのテキストを強調表示し、コメントを追加できます。以下の手順に従って、Red Hat ドキュメントのフィードバックをお寄せください。

前提条件

- Red Hat カスタマーポータルにログインしている。
- Red Hat カスタマーポータルで、**Multi-page HTML** 形式でドキュメントを表示している。

手順

1. **Feedback** ボタンをクリックして、既存の読者のコメントを表示します。



注記

フィードバック機能は、**Multi-page HTML** 形式でのみ有効です。

2. フィードバックを提供するドキュメントのセクションを強調表示します。
3. 選択したテキストの近くに表示されるプロンプトメニューで、**Add Feedback** をクリックします。
ページの右側のフィードバックセクションにテキストボックスが開きます。
4. テキストボックスにフィードバックを入力し、**Submit** をクリックします。
ドキュメントに関する問題が作成されます。
5. 問題を表示するには、フィードバックビューで問題トラッカーリンクをクリックします。

第2章 CRYOSTAT のサポートポリシー

Red Hat は、Cryostat のメジャーバージョンを最低 6 カ月間サポートします。Red Hat は、製品が Red Hat カスタマーポータルでリリースされる時期に基づいてこの数値を定めています。

Cryostat は、x86_64 アーキテクチャーで実行される Red Hat OpenShift Container Platform 4.8 以降のバージョンにインストールしてデプロイできます。

関連情報

- Cryostat のライフサイクルポリシーの詳細は、Red Hat OpenShift Container Platform ライフサイクルポリシーの Web ページにある [Red Hat build of Cryostat](#) を参照してください。

第3章 新機能

Cryostat 2.0 では、Cryostat 製品の使用を強化する新機能が導入されています。

Cryostat アプリケーションの API パーミッション

Cryostat Operator はサービスアカウントを作成し、そのサービスアカウントを使用するように Cryostat アプリケーションを設定します。Cryostat Operator はサービスアカウントをチェックして、そのパーミッションが承認チェックによって求められるパーミッションと同期していることを確認します。

Cryostat 1 では、Cryostat Operator は独自のサービスアカウントのパーミッションを Cryostat アプリケーションに適用していました。これにより、Cryostat アプリケーションに過剰なパーミッションが与えられていました。

自動化ルール

Cryostat 2.0 では、自動化ルールを使用して、実行中のターゲットアプリケーションを JFR により継続的にモニターすることができます。アプリケーションの再起動または再デプロイは必要ありません。

JVM フラグを設定することで、アプリケーションの常時継続的なモニタリングを引き続き有効にすることもできますが、この方法ではアプリケーションを再起動する必要があります。

Cryostat 2.0 には、自動化されたルールとテンプレートの作成に使用できる継続的なモニタリングイベントテンプレートがあります。継続的なモニタリングイベントテンプレートを使用すると、JFR を指定してアプリケーションを継続的にモニターするためのダウンタイムを短縮できます。

自動化ルールには、以下の自動化ルール定義を指定する必要があります。

- 1 回限りのルール定義: JFR 設定用の 1 回限りの自動化ルールの簡単な設定を作成できるようになりました。 `eventSpecifier` プロパティには、 `archive` の単純な文字列値が含まれます。
- **matchExpression** 定義: 自動化ルールを受け取る必要がある JVM ターゲットを定義する自動化ルールの必須コンポーネント。Cryostat は、 **matchExpression** 文字列式で定義された正規表現に一致するすべての JVM ターゲットにルールを適用します。有効な文字列式は、区切り文字、演算子、および単純な式を含むものです。
 - OpenShift ラベルまたはアノテーションを **matchExpression** 定義の条件として適用できません。

自動化ルールを JSON 形式で Cryostat にアップロードすると、Cryostat は定義ファイルを検証します。

以前に定義された一致するターゲットのルール定義をすぐにアクティブにすることができます。ルール定義の作成後にターゲットを再度追加または再起動する必要はありません。

HTTP クエリー `clean=true` パラメーターをルール定義の **DELETE** 要求に追加して、ルール定義を削除できます。このパラメーターは、アクティブなレコーディングのみを削除するように Cryostat に指示します。アーカイブされたレコーディングは削除対象ではありません。

cert-manager API

Cryostat Operator の **Create Cryostat** ページにある **Enable cert-manager integration** トグルボタンで、Cryostat アプリケーションの **cert-manager** を使用してエンドツーエンドの暗号化を有効または無効にできるようになりました。

図3.1 Enable cert-manager integration トグルボタン

Name *

Labels

Minimal Deployment *

false

Deploy a pared-down Cryostat instance with no Grafana dashboard or jfr-datasource and no web-client UI.

Enable cert-manager Integration

true

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

Event Templates >

List of Flight Recorder Event Templates to preconfigure in Cryostat

Storage Options >

Options to customize the storage for Flight Recordings and Templates

Trusted TLS Certificates >

List of TLS certificates to trust when connecting to targets

Network Options >

Options to control how the operator exposes the application over a network

さらに、Cryostat Operator は、Cryostat をデプロイする前に **cert-manager** API があることを自動的にチェックするようになりました。これにより、Cryostat Operator が Cryostat を安全にデプロイできるようになります。必要性に応じて、Cryostat Operator **cert-manager** API チェックを無効にするように明示的に設定できます。

Cryostat Operator が **cert-manager** を必要とする Cryostat インスタンスを検出したが、**cert-manager** をインストールしていない場合は、OpenShift Web コンソールの **Events** ページに Cryostat インスタンスの警告イベントが表示されます。

デプロイメントシナリオのツリービュー

アプリケーションを OpenShift またはサポートされる Kubernetes ディストリビューションにデプロイした後に、Cryostat の情報を JSON 形式のツリービューで表示できます。ツリービューでは次のコンポーネントを表示できます。

- Pod
- レプリカ
- コンテナ
- エンドポイント

エンドポイントリソースの監視

OpenShiftPlatformClient により、エンドポイントリソースの監視がターゲット JVM の非同期ターゲット検出を自動的に設定できるようになります。

TargetEventGetHandler でのイベント検索

TargetEventGetHandler を使用して、ターゲット JVM によって生成されたイベントタイプをリスト表示または検索できます。

HTTP API の動作変更

Cryostat 2.0 より前のバージョンでは、Vertx サーバーのハンドラーをインストールするかどうかを決定する際に、HTTP リクエストハンドラーがデフォルトの **isAsync** メソッドの設定 **true** をオーバーライドすることがありました。

各 HTTP ハンドラーには、Cryostat の HTTP API エンドポイントを定義するコードが含まれています。**isAsync** メソッドは、HTTP ハンドラーが別のワーカースレッドまたはメインの webserver スレッドによって処理されるかどうかを定義します。

新しい環境変数

Cryostat 2.0 には、次の 2 つの新しい環境変数が追加されています。

- **CRYOSTAT_TARGET_CACHE_MAX_CONNECTIONS**: OpenShift Operator は、この変数を、開いている JMX 接続の最大数を指定する値に設定します。
- **CRYOSTAT_TARGET_CACHE_TTL**: OpenShift Operator は、この変数を、非アクティブ状態が続いて終了するまで JMX 接続をキャッシュする秒数を指定する値に設定します。



注記

OpenShift Operator はこれらの環境変数の値を自動的に判別するため、これらの環境変数について OpenShift Operator によって指定された値を変更することはできません。

新しいハンドラー

Cryostat 2.0 では、次の V2 ハンドラーが導入されています。

- **AbstractV2RequestHandler**: **AbstractAuthenticatedRequestHandler** と同様に動作しますが、V2 ハンドラー間で応答全体の形式を維持します。
- **TargetRecordingOptionsListGetHandler**: ターゲット JVM でサポートされているレコーディングオプションを確認します。ターゲット JVM でサポートされていないレコーディングオプションを使用しないように、レコーディングを作成する前にこのハンドラーを使用してください。
- **TargetSnapshotPostHandler**: カスタマイズされたスナップショットのレコーディングのためにターゲット応答を JSON 形式でエンコードします。この V2 スナップショットハンドラーには、レコーディングの名前、ダウンロード可能なレコーディングファイルの URL、自動分析レポートの URL なども含まれています。



注記

V1 ハンドラーは V2 ハンドラーと同様のスナップショット操作を実行できますが、V1 ハンドラーの HTTP 応答はプレーンの文字列形式でのみレンダリングされ、スナップショットのレコーディングの名前のみを含みます。

Cryostat で利用可能な新規イベントテンプレート

Cryostat Web コンソールの利用可能な **Target JVM** テンプレートのリストから **cryostat.jfc** イベントテンプレートを選択できるようになりました。

cryostat.jfc イベントテンプレートを選択すると、以下の Cryostat 設定が自動的に有効になります。

- ステータスコード、期間などの HTTP リクエスト情報。

- 期間、メッセージタイプなどの WebSocket メッセージ情報。
- オープン、クローズ、失敗などの JMX 接続の状態。

事前設定された Grafana ダッシュボードイメージ

Cryostat 2.0 には、アプリケーションのメトリクスを表示するための事前設定された Grafana ダッシュボードイメージが含まれています。必要に応じて、独自の Grafana ダッシュボードイメージを作成することもできます。

Cryostat 1 では、Cryostat コンテナのデプロイ時に OpenShift Operator による基本的なカスタマイズを提供する Grafana イメージを提供していました。

応答ステータスコードメッセージ

ClientUrlGetHandler エラーメッセージの応答ステータスコードメッセージが、エラーメッセージの前に表示されるようになりました。これはデフォルト設定です。カスタム例外メッセージは、エラーメッセージの応答ボディ部に残っています。

クライアント側の接続プロトコルのサポート

Cryostat 2 では、**clientlib** ディレクトリーのマウントポイントが追加されました。ここに Cryostat クラスパスにロードする追加の JAR ファイルを追加できます。これは、JMX RMI の代替として、クライアント定義の接続プロトコルをサポートします。

第4章 機能拡張

Cryostat 2.0 には、Cryostat 1 の機能に基づいて構築された機能拡張が含まれています。

Cryostat Web コンソールの GUI

Cryostat Web コンソールの右上隅にある **Help** アイコンを使用して Cryostat 情報ページにアクセスできます。このコンソールページで、Cryostat のバージョンを表示できます。

また、アーカイブされたレコードの生成された名にターゲットエイリアスが含まれるようになりました。ターゲットエイリアスにより、エイリアスがレコードの生成された名前にリンクされるため、レコードの取得が向上します。

Cryostat Operator トポロジレビュー

Cryostat Operator が、正しい **app.openshift.io/connects-to** アノテーションを Cryostat デプロイメントに適用するようになりました。この設定により、OpenShift Container Platform Web コンソールのトポロジレビューで、Cryostat デプロイメントが Cryostat Operator デプロイメントにリンクされます。

Cryostat Operator コントローラーマネージャー

cert-manager インストールしなかったために Cryostat Operator が Cryostat のデプロイに失敗した場合でも、Cryostat カスタムリソースを削除するときに Cryostat Operator がハングしなくなりました。

Cryostat クラスター ConsoleLink namespace

Cryostat 2.0 では、Cryostat Operator クラスターが Cryostat クラスター用に重複した **ConsoleLink** namespace を作成する場合があります。この問題を回避するために、Cryostat Operator は **GenerateName** オブジェクトを **ConsoleLink** リソース定義の **Name** オブジェクトで定義された **SH256** ハッシュ値に置き換えるようになりました。

カスタムイベントテンプレート

カスタムイベントテンプレートを使用して、すべての Cryostat JFR レコーディングを設定できます。カスタムイベントテンプレートは、以下の方法で使用できます。

- Cryostat Operator に、保存された **ConfigMap** オブジェクトのテンプレートファイルを指定して Cryostat カスタムイベントテンプレートを事前設定するよう指示します。
- Cryostat Web アプリケーションを使用してカスタムイベントテンプレートをアップロードします。

サポートされる形式による SSL/TLS 証明書のエンコード

ターゲット JVM アプリケーションの Cryostat Web コンソールインターフェイスに SSL/TLS 証明書を追加できます。

有効な SSL/TLS 証明書は、DER でエンコードされた **base-64** または **binary** 形式です。どちらの形式でも次の拡張子がサポートされています。

- **.der**
- **.cer**
- **.pem**

生成された証明書を実行時に指定できるようになりました。これにより、ターゲット JVM アプリケーションが JMX 接続に SSL/TLS を使用できるようになります。Cryostat は、SSL/TLS 証明書を使用するターゲット JVM への JMX 接続を開くを試みます。JMX 接続を成功させるには、実行時に指定したターゲット JVM 証明書に対するすべての認証チェックに Cryostat が合格する必要があります。

POST ハンドラーを使用して、証明書の受け入れ、検証、書き込みを行うことができます。

Fabric8 Kubernetes および OpenShift クライアント

Cryostat 2.0 は、Fabric8 Kubernetes および OpenShift クライアントのバージョン 4.12.0 をサポートします。このバージョンでは、Cryostat とのアプリケーションの互換性が強化され、ダウンストリームのビルドエラーが減少します。

Grafana コンテナのバージョン

Cryostat 2.0 では、Grafana コンテナのバージョン 6.4.4 がバージョン 7.3.6 に置き換わります。

Grafana エラーメッセージ

502 エラーメッセージは失敗した JMX 接続に関連するものであり、**512** エラーメッセージは JFR コンテナに送信された無効な応答に関連するものです。

ヘルスチェックリソースの定義

Cryostat Operator の **api/v1/clienturl** リソース定義が **health** リソース定義に置き換わります。Cryostat Operator は、コンテナ化された JVM ヘルスチェックの実行時に **health** エンドポイントを使用するようになりました。

initialization-resource アノテーション

Cryostat Operator の CSV ファイルの設定に **initialization-resource** アノテーションが含まれるようになりました。このアノテーションは、Cryostat クラスターの Cryostat カスタムリソースを作成するためのグラフィカルなヒントを提供して、OpenShift Container Platform Web コンソールで実行される Cryostat Operator インスタンスを強化します。

OLM バンドル記述子

Cryostat 2.0 の OLM バンドルでは、JFR レコーディングの期間をチェックするために、**eventOptions** 記述子に整数値を設定する必要がなくなりました。代わりに、OLM バンドルは **EventOptions** 文字列で定義される期間形式の値を使用するようになりました。

サポートされる期間単位には、**s**、**m**、および **h** があります。たとえば、**2h15m** は、2 時間 15 分の JFR レコーディングの長さを示します。

SCC (Security Context Constraints)

Cryostat Operator はデフォルトで **restricted** SCC 設定を使用するようになりました。Cryostat クラスターに含まれる Pod が、**restricted** SCC 設定にリスト表示されている許可された **fsGroup** の値を使用できるようになりました。これにより、永続ストレージ (PV) にマウントされた Cryostat Pod が、そのディレクトリーへの読み取り/書き込みアクセス権を持つことができます。

Pod の読み取り/書き込みアクセスレベルは、Pod の **fsGroup** GID の値により異なります。Cryostat Operator は、この値をビルトインの制限付き **SecurityContextConstraint** に準拠するように設定します。**SecurityContext** 要素には、Pod レベルのセキュリティ属性が含まれます。

Cryostat 2.0 より前のバージョンでは、Cryostat Operator が **default** 設定に設定されていたため、Cryostat クラスターで実行する Pod の **fsGroup** アクセスの問題が生じました。

Cryostat クラスターで許可される **fsGroup** 値の範囲についての詳細は、OpenShift ドキュメントの Web サイトの [About pre-allocated security context constraints values](#) を参照してください。

ServiceRef 定義

ServiceRef 定義には、HTTP API ハンドラーを使用した **GET** 通信から返されたすべての **ServiceRef** オブジェクトに含まれる以下の新しいプロパティが含まれています。

- Java アプリケーション名、ラベル、ポート番号などのアノテーション。
- エイリアスや **connectURL** などのユーザー固有の **ServiceRef** 値。

以下の 2 つのハンドラーを使用して **ServiceRef** 定義を作成または削除できます。

- **POST:** 同じサービス URL を持つ既存のターゲットが存在しない場合に、新規の **ServiceRef** を作成します。
- **DELETE:** **targetID** パスパラメーターで指定された値と一致する場合に、**CustomTargetPlatformClient** から **ServiceRef** 定義を削除します。

サブプロセス管理

Cryostat は自動分析レポートを生成する前に、子サブプロセスを作成します。子サブプロセスは、レポート生成によって消費される大量のメモリー負荷を蓄積することで親プロセスを保護します。

たとえば、Linux オペレーティングシステムでは、OOM (Out-of-memory) Killer が、システムから利用できない追加メモリーを要求するプロセスを検出し、プロセスを停止します。子サブプロセスが存在する場合、OOM Killer はこのサブプロセスを停止しますが、実行中の親プロセスには干渉しません。

Cryostat 環境変数 **CRYOSTAT_REPORT_GENERATION_MAX_HEAP** を使用して、子サブプロセスの最小または最大の JVM ヒープサイズを設定できます。最小値が低いとレポートを生成する前に子サブプロセスが停止する可能性があり、最大値が高いと親プロセスで追加のメモリー制約が発生する可能性があることに注意してください。

サブプロセスレポートの生成

Cryostat 2.0 は、**ActiveRecordingReportCache** または **ArchivedRecordingReportCache** のいずれかを使用して Java アプリケーションのサブプロセスレポートを生成する拡張機能を提供します。これらのクラスは以下のように簡素化されます。

- **ActiveRecordingReportCache** に、ディスクのレコーディングへのパスが自動的に含まれるようになりました。
- **ArchivedRecordingReportCache** は、親プロセスの接続マネージャーを使用してレコードをローカルディスクファイルにコピーするユーティリティーメソッドを呼び出します。このクラスはローカルパスの場所をサブプロセスに渡します。

サブプロセスでは、以下のタスクを完了する必要がなくなりました。

- JMX 接続を確立する。
- SSL/TLS 証明書を処理する。
- JMX 認証のクレデンシャルを処理する。

これらのタスクが不要になると、次の面でサブプロセスのワークロードが改善されます。

- エンドツーエンドのレポート生成が高速化されます。
- コードの複雑さが軽減され、コードベースの保守が容易になります。
- SSL/TLS 認証および JMX 認証メカニズムが適切かつセキュアに処理されます。

truststore 環境変数

Cryostat 2.0 では、**TRUSTSTORE** 環境変数名が **SSL_TRUSTSTORE_DIR** に置き換わります。

TargetConnectionManager ハンドラー

TargetConnectionManager は、ターゲット要求と Cryostat 間の接続に JMX 接続にタイムアウトしたキャッシュ値を使用することで、同時接続をサポートするようになりました。デフォルトでは、時間のキャッシュは 90 秒です。

Vertx サーバーの応答管理

Vertx サーバーは、**TimeoutHandler** 実装を使用して、Cryostat リクエストハンドラーによって送信された遅延応答メッセージを自動的に終了できるようになりました。これにより、クライアントが Cryostat リクエストハンドラーからの応答を無期限に待機する必要がなくなります。

TimeoutHandler 実装は、Cryostat リクエストハンドラーからの遅延応答を検出すると、**500** エラーメッセージを出力します。

WebSocket

WebSocket には以下の更新が含まれます。

- WebSocket 接続が、双方向のインタラクティブチャネルから一方方向のプッシュ通知チャネル (NC) にアップグレードされました。
- WebSocket が一方方向のプッシュ通知に WebSocket 通知チャネルを使用するようになったため、WebSocket 通知チャネルが **api/v1/command** から **api/v1/notifications** に変更されました。
- JFR データの分析にレコーディングとイベントテンプレートを使用する場合、WebSocket は通知チャネルでイベントを送信できます。

WebSocket クライアントが一方方向プッシュ NC に接続すると、クライアントは同じチャネルで他の接続クライアントが実行したアクションに関する情報を自動的に受信します。

第5章 サポートされない機能および非推奨の機能

Cryostat 2.0 では、メンテナンスコストの高さ、コミュニティの関心の低さ、より優れた代替ソリューションにより、一部の機能が削除されています。

ContainerJFR バリエーション

Cryostat namespace が **ContainerJFR** namespace に置き換わるため、Cryostat 2.0 は **ContainerJFR** namespace のバリエーションを指定しなくなりました。

イベント指定子の文字列構文

Cryostat 2.0 には、イベント固有の文字列構文が含まれていません。イベントテンプレートが、イベント固有の文字列構文に代わる機能を提供します。

対話モードのコマンド

HTTP API および通知チャンネルがコマンドチャンネルに置き換わります。Cryostat 2.0 では、コマンドチャンネル実装やその **CommandExecutor** インターフェイスを使用できません。

TTY/TCP 実行モード

Cryostat 2.0 には、TTY および TCP ソケット実行モードやモードの設定は含まれていません。これらの実行モードは、プラットフォーム認証もターゲット認証もサポートしていませんでした。

サポートされている WebSocket 実行モードを引き続き使用できます。

WebSocket コマンド

Cryostat 2.0 では、すべての WebSocket コマンドが削除されました。これらのコマンドは Cryostat 1 リリースで非推奨になりました。

WebSever IO クリーンアップ

Cryostat 2.0 では、組み込み WebSever IO の低メモリー不足モードが削除されました。このモードを使用して、固定メモリーサイズのマッピングを組み込み WebServer IO に適用することはできなくなりました。

第6章 既知の問題

Cryostat のリリースには、Red Hat が認識し、後の製品開発段階で修正される可能性がある問題が含まれている場合があります。既知の各問題の説明と解決策を確認してください。

OpenShift SDN デフォルト CNI ネットワークプロバイダー

Cryostat 2.0 は、Cryostat ノードで実行しているノードとは異なるノードにある JVM の以下の設定を持つ OpenShift クラスターを接続できません。

- ソフトウェア定義ネットワーク (SDN) メソッドを、クラスター上の Pod 間の通信を行うための統合クラスターネットワークとして使用する。
- デフォルトの Container Network Interface (CNI) ネットワークプロバイダーを、コンテナのネットワークインターフェイスを設定するためのプラグインを作成するために使用する。

この既知の問題を解決するには、SDN メソッドではなく Open Virtual Network (OVN) メソッドを使用するようにクラスターを設定します。OVN メソッドには、SDN メソッドと同様の次の設定が含まれています。

- Open vSwitch (OVS) を使用してネットワークトラフィックを管理する。
- CNI ネットワークプロバイダーをデフォルトとして設定するプラグインを使用する。

関連情報

- SND メソッドの詳細は、Red Hat OpenShift ドキュメントの [About the OpenShift SDN default CNI network provider](#) を参照してください。
- OVN メソッドの詳細は、Red Hat OpenShift ドキュメントの [About the OVN-Kubernetes default Container Network Interface \(CNI\) network provider](#) を参照してください。

アーカイブアップロード API のメモリー不足 (OOM)

クライアントが Cryostat HTTP **POST /api/v1/recordings** ハンドラーにリクエストを送信すると、Cryostat 2.0 は予想よりも多くのメモリーを消費します。このハンドラーは Cryostat 2.0 インスタンスの **/opt/cryostat.d/recordings.d** ディレクトリーを参照しており、ユーザーはこのハンドラーを使用して **.jfr** バイナリーファイルをこのディレクトリーにアップロードできます。

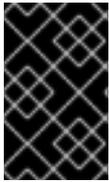
Cryostat Operator は、OpenShift プロジェクトにデプロイされた Cryostat インスタンスに対して 512 MB のデフォルトのメモリー制限を設定します。150 MB 以上の **.jfr** ファイルを Cryostat 2.0 インスタンスにアップロードすると、OpenShift クラスターの OOM (Out of Memory) Killer が、デプロイされた Cryostat インスタンスを含む Pod を停止します。

この既知の問題は、**.jfr** バイナリーファイルを Cryostat 2.0 インスタンスの永続ストレージの場所にコピーすることで解決できます。この方法を使用すると、**/opt/cryostat.d/recordings.d** ディレクトリーに **.jfr** バイナリーを保存するためにクライアント要求を Cryostat HTTP **POST /api/v1/recordings** ハンドラーに送信する必要がなくなります。

次のコマンドを発行して、**.jfr** バイナリーファイルを Cryostat 2.0 インスタンスの永続ストレージの場所にコピーできます。

```
oc exec -i -n <your_namespace> -c <cryostat_container_name> <cryostat_pod_name> -- mkdir /opt/cryostat.d/recordings.d/unlabelled
```

```
oc cp vertx-fib-demo-6f4775cdbf-82dvl_150mb_20211006t152006z.jfr <cryostat_pod_name>:/opt/cryostat.d/recordings.d/unlabelled/vertx-fib-demo-6f4775cdbf-82dvl_150mb_20211006t152006z.jfr -c <cryostat_container_name>
```



重要

`/opt/cryostat.d/recordings.d/` パスに **unlabelled** ディレクトリーがすでに存在する場合、前述の **oc exec** コマンドは失敗します。失敗したコマンドメッセージを無視して、**oc cp** コマンドを続行することを選択できます。

.jfr バイナリーファイルを PVC アーカイブの場所に直接コピーした後、**curl**、**httpie**、または **wget** コマンドを使用して、Cryostat 2.0 インスタンスに **.jfr** ファイルが存在することを確認できます。

次の例は、**oc cp** コマンドによって、永続ストレージの場所にコピーされたアップロード済みファイルを Cryostat が認識しているか **curl** コマンドで確認する方法を示しています。この例の `<cryostat_url>` 値は <https://cryostat-sample-myproject.apps-crc.testing:443> に解決されますが、`<cryostat_url>` 値はお使いのアプリケーションの URL に置き換えることができます。**oc status** コマンドを発行すると、アプリケーションの URL を判別できます。

```
$ curl -kv -H Authorization:"Bearer $(oc whoami -t)" <cryostat_url>/api/v1/recordings
```

関連情報

- Cryostat コンテナのメモリ制限の問題を解決するために使用できるコマンドの詳細は、Red Hat OpenJDK Jira プロジェクトの [OPENJDK-495](#) を参照してください。

統合された Grafana コンポーネントのファイルアップロード制限

Cryostat 2.0 の場合、統合された **View in Grafana** コンポーネントは、**jfr-datasource** コンテナの設定の問題により、10 MB を超える JFR ファイルを受け入れることができません。

デプロイされた Cryostat 2.0 Pod の **jfr-datasource** コンテナは、デフォルトの **quarkus.http.limits.max-body-size** パラメーターを含むデフォルトの Quarkus 設定を使用します。このパラメーターは、Quarkus 上のファイルの最大サイズ制限を設定します。パラメーターのデフォルト値は 10 MB です。

クライアントが 10 MB を超える JFR ファイルをアップロードしようとするすると、**jfr-datasource** Web サーバーはファイルを拒否し、**HTTP 413** エラーメッセージを出力します。

この既知の問題は、次の手順を完了することで解決できます。

1. Cryostat 2.0 インスタンスの **Recordings** メニューで、リストされているアクティブまたはアーカイブのレコーディングに移動します。
2. 対象のレコーディングのオーバーフローメニューから、**Download Recording** オプションをクリックします。
3. ローカルシステムの任意の場所にファイルを保存します。
4. ダウンロードした JFR ファイルを Java Mission Control (JMC) デスクトップアプリケーションで開きます。

第7章 セキュリティー修正

Cryostat 2.0 には、Cryostat の以前のリリースで提供されていたセキュリティーを強化するセキュリティー修正が含まれています。各セキュリティー修正の説明を確認してください。

jsoup Java ライブラリー

Cryostat 2.0 で事前にインストールされている **jsoup** Java ライブラリーは、バージョン 1.13.1 からバージョン 1.14.2 にアップグレードされました。この新しいバージョンは、HTML および XML 構文の **jsoup** Java ライブラリーの処理を強化します。

関連資料

jsoup 1.14.2 リリースに関連するバグ修正の詳細は、**jsoup** ドキュメントの Web サイト [jsoup Java HTML Parser release 1.14.2](#) を参照してください。

改訂日時: 2022-12-03 22:20:45 +1000