



Red Hat build of Cryostat 3

Red Hat build of Cryostat 3.0 のリリースノート

Red Hat build of Cryostat 3 Red Hat build of Cryostat 3.0 のリリースノート

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 3.0 のリリースノートには、Cryostat 3.0 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	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 のサポートポリシー	5
第2章 新機能	6
Cryostat DB コンテナ	6
Cryostat ストレージコンテナ	6
リバースプロキシアーキテクチャー	6
ルートホスト名のカスタマイズのサポート	6
JVM への動的割り当て	7
Cryostat エージェントをスタンドアロンプロセスとして起動するためのサポート	8
第3章 機能拡張	9
Cryostat コンテナの機能強化	9
マルチ namespace の Cryostat インスタンスの作成に対する Cryostat API のサポート	9
Cryostat にアクセスするための RBAC の拡張機能	9
Cryostat CR 検証の機能強化	9
Cryostat Helm チャートの設定の強化	9
Cryostat エージェント組み込み Web サーバー	10
Cryostat エージェントポートの強化	10
第4章 サポートされない機能および非推奨の機能	11
ターゲット TLS 証明書のアップロード	11
API リクエスト経由で渡される JMX ターゲットクレデンシャル	11
Cryostat の自動検出	11
単一の namespace 用の Cryostat Operator のインストール	11
Cluster Cryostat API	11
第5章 既知の問題	13
URL パスで直接アクセスされると /recordings/create などのサブビューにアクセスできない	13
replace=always パラメーターでレコーディングを再起動すると、アクティブな Recordings テーブルが更新に失敗する	13
第6章 このリリースに関連するアドバイザリー	14

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 3.0 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、用語の置き換えは、今後の複数のリリースにわたって段階的に実施されます。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 または ARM64 アーキテクチャーで実行される Red Hat OpenShift Container Platform 4.12 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2章 新機能

このセクションでは、Cryostat 3.0 リリースが提供する新機能について説明します。

Cryostat DB コンテナ

Cryostat 3.0 以降では、[Red Hat Ecosystem Catalog](#) には Cryostat DB コンテナイメージ(**cryostat-db**)も含まれています。Cryostat Operator または Helm チャートのいずれかを使用して Cryostat をインストールすると、**cryostat-db** コンテナも自動的にデプロイされます。

cryostat-db コンテナは、若干カスタマイズされた Postgres データベースを提供します。Cryostat は、暗号化されたターゲット JMX 認証情報キーリング、自動化ルール定義、検出されたターゲット、検出プラグインなどの情報を保存するためにこのデータベースを使用するようになりました。

以前のリリースでは、Cryostat は JMX 認証情報キーリングなどの情報を単純な H2 ファイルベースのデータベースに保存し、その他の情報は、Automated Rules 定義などの他の情報がディスクに直接ファイルとして保存されていました。**cryostat-db** コンテナの導入により、Cryostat は異なるタイプの情報を同じ Postgres データベースに保存できるようになりました。

Cryostat ストレージコンテナ

Cryostat 3.0 以降では、[Red Hat Ecosystem Catalog](#) には Cryostat ストレージコンテナイメージ(**cryostat-storage**)も含まれています。Cryostat Operator または Helm チャートのいずれかを使用して Cryostat をインストールすると、**cryostat-storage** コンテナも自動的にデプロイされます。

cryostat-storage コンテナは、S3 互換ストレージプロバイダーとして機能する、若干カスタマイズされた SeaweedFS ストレージソリューションを提供します。

以前のリリースでは、Cryostat はアーカイブされた Flight Recordings およびカスタムイベントテンプレートに、ディスク上のファイルの直接ストレージを使用していました。**cryostat-storage** コンテナの導入により、Cryostat はこのタイプの情報に対して直接のファイルシステムアクセスを使用する必要がなくなりました。

リバースプロキシアーキテクチャー

Cryostat Operator または Helm チャートのいずれかを使用して Cryostat 3.0 をインストールすると、Cryostat のリバースプロキシ(**openshift-oauth-proxy** または **oauth2_proxy**)が Pod に含まれるようになりました。このプロキシのみがサービスを介してクラスタトラフィックに公開されます。つまり、Cryostat へのすべての API リクエストと、Cryostat Web コンソールまたは Grafana ダッシュボードのすべてのユーザーがプロキシを介して転送されます。

プロキシは、ユーザーセッションを処理してアプリケーションへのアクセスを制御し、Cryostat Web コンソールと Grafana ダッシュボードの両方に統一されたアクセス制御とユーザーセッションを提供します。これらのユーザーインターフェイスはいずれも同じルートからアクセスでき、同じ TLS 証明書を表示します。

Red Hat OpenShift にデプロイすると、プロキシは Cryostat インストール namespace を使用してロールベースのアクセス制御(RBAC)を実行し、Red Hat OpenShift クラスタ SSO プロバイダーと統合することで、ユーザー認証と承認をチェックします。オプションで **htpasswd** ファイルを使用して認証プロキシを設定し、Basic 認証を有効にすることができます。Red Hat OpenShift では、これにより、Red Hat OpenShift SSO RBAC アクセスの有無以外に Cryostat にアクセスできる追加のユーザーアカウントを定義できます。

ルートホスト名のカスタマイズのサポート

デフォルトでは、Red Hat OpenShift Container Platform は、ホストを指定しないルートに対して、クラスタのデフォルトの Ingress ドメイン名に基づいてホスト名を自動的に割り当てます。要件に応じて、Cryostat Operator が Cryostat デプロイメント用に作成するルートに特定のホスト名を使用するこ

とができます。Cryostat 3.0 では、Cryostat カスタムリソース(CR)の新しい **.spec.networkOptions.coreConfig.externalHost** プロパティを使用して、Cryostat ルートのカスタムホスト名を指定できます。

Red Hat OpenShift コンソールでは、Cryostat CR の作成時にこのプロパティにアクセスできます。

Project: cryostat-test ▾

Network Options ▾

Options to control how the operator exposes the application outside of the cluster, such as using an Ingress or Route.

Core Config ▾

Specifications for how to expose the Cryostat service, which serves the Cryostat application.

External Host

cryostat.example.com

Externally routable host to be used to reach this Cryostat service. Used to define a Route's host on OpenShift when it is first created. On Kubernetes, define this using "spec.ingressSpec".

Ingress Spec >

Configuration for an Ingress object. Currently subpaths are not supported, so unique hosts must be specified (if a single external IP is being used) to differentiate between ingresses/services.

または、Cryostat CR を YAML 形式で作成することもできます。以下に例を示します。

```
apiVersion: operator.cryostat.io/v1beta2
kind: Cryostat
metadata:
  name: cryostat-sample
spec:
  networkOptions:
    coreConfig:
      externalHost: cryostat.example.com
```

ルートが Red Hat OpenShift Container Platform に作成されると、ルートのホスト名を変更することはできません。Cryostat CR の作成後にルートのホスト名を変更する必要がある場合は、Cryostat CR を削除し、変更したホスト名で新しい CR を作成する必要があります。

JVM への動的割り当て

Cryostat 3.0 以降では、Cryostat エージェントは、アプリケーションを再起動しなくても、すでに実行中のアプリケーション JVM に動的に割り当てることができます。この動的割り当て機能には、以下の要件があります。

- エージェントの JAR ファイルが JVM のファイルシステムにコピーされていることを確認する必要があります (たとえば、**oc cp** コマンドを使用して)。
- エージェントを、(**oc exec** コマンドを使用して)同じホスト上または同じアプリケーションで別のプロセスとして実行できる必要があります。

動的アタッチメントは、JVM の起動時に毎回エージェントを割り当てる必要がない可能性のあるアドホックワнтаイムプロファイリングまたはトラブルシューティングワークフローをサポートします。動的アタッチメントは、エージェントをアタッチする目的でアプリケーションの再設定を行うことができないか、または再設定したくない状況に適しています。エージェントはアプリケーションの再起動を必要とせず、実行中の JVM にアタッチできるため、アプリケーションのダウンタイムも発生しません。



注記

以前のリリースでは、JVM の起動時にアプリケーションの JVM を読み込み、初期化できるようにすることが唯一のオプションでした。これには、Cryostat エージェントの JAR ファイルへのパスを指定して **-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。要件に応じて、このタイプの静的アタッチメントの JVM への引き続き使用できます。

Cryostat エージェントをスタンドアロンプロセスとして起動するためのサポート

Cryostat 3.0 以降では、Cryostat エージェントをすでに実行中のアプリケーション JVM に動的に割り当てる場合は、スタンドアロン Java プロセスとしてエージェントを起動できます。この機能により、エージェントの JAR ファイルを JVM のファイルシステム（たとえば **oc cp** コマンドを使用して）にすでにコピーする必要があります。

エージェントを起動するには、次のコマンドを実行します。< **agent_jar_file** > はエージェントの JAR ファイル名を表し、< **pid** > は JVM のプロセス ID (PID) を表します。

```
$ java -jar target/<agent_jar_file> <pid>
```

以下に例を示します。

```
$ java -jar target/cryostat-agent-0.4.0.jar 1234
```

エージェントプロセスは、そのアタッチプロバイダーを使用して指定された PID を検索します。指定した PID が見つかった場合、エージェントプロセスはこの PID に接続し、エージェントの JAR ファイルをこの JVM にロードし、通常のエージェント起動プロセスにブートストラップします。

コマンドラインオプションを使用して、エージェントランチャーに追加の遅延設定オプションを指定することもできます。以下に例を示します。

```
$ java -jar target/cryostat-agent-0.4.0.jar \  
-Dcryostat.agent.baseuri=http://cryostat.local \  
--smartTrigger=[ProcessCpuLoad>0.2]~profile \  
@/deployment/app/moreAgentArgs \  
1234
```

利用可能なオプションとその動作の詳細は、**java -jar target/cryostat-agent-0.4.0.jar -h help** コマンドを実行してください。with **-D** で指定するシステムプロパティーは、挿入されたエージェントが設定値の読み取りを試行する前にホスト JVM に設定されます。これは、これらのシステムプロパティーや同等の環境変数をホスト JVM プロセス自体に設定するのと同じ効果があります。

第3章 機能拡張

Cryostat 3.0 には、Cryostat 2.4 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 コンテナの機能強化

本リリースでは、Quarkus を使用して、メインの Cryostat コンテナ(**cryostat**)が再実装されました。以前のリリースでは、**cryostat** コンテナは Eclipse Vert.x を使用して直接ビルドされていました。

今回の機能拡張により、Cryostat は Quarkus フレームワークの長所と機能を活用できるようになり、Cryostat は JDK Flight Recorder へのパフォーマンスへのアクセスを向上させ、データの整合性とセキュリティを向上させることができるようになりました。

マルチ namespace の Cryostat インスタンスの作成に対する Cryostat API のサポート

Cryostat 3.0 以降では、**Cryostat** API はシングル namespace とマルチ namespace の両方の Cryostat インスタンスの作成をサポートします。Cryostat Operator を使用して Cryostat インスタンスをインストールすると、**Cryostat** API でターゲット namespace のオプションのリストを指定できるようになりました。これは、**Cryostat** API が単一 namespace インスタンスの作成のみをサポートしていた以前のリリースの動作に優先します。



注記

以前のリリースでは、**Cluster Cryostat** API を使用して、マルチ namespace の Cryostat インスタンスを作成できました。Cryostat 3.0 では、個別の **Cluster Cryostat** API が提供されなくなりました。

Cryostat にアクセスするための RBAC の拡張機能

Cryostat は、製品へのアクセスを許可または拒否するために、すべてのユーザーに同じロールベースのアクセス制御(RBAC)パーミッションチェックを適用するようになりました。デフォルトでは、Cryostat アプリケーションのデプロイメント namespace に必要な RBAC ロールは **create pods/exec** です。

必要な RBAC ロールが割り当てられている Red Hat OpenShift ユーザーアカウントに、Cryostat Web コンソールとすべての Cryostat 機能に完全にアクセスできるようになりました。Red Hat OpenShift アカウントに必要な RBAC ロールがない場合、このユーザーは Cryostat へのアクセスをブロックします。

Cryostat Operator を使用して Cryostat インスタンスをインストールする場合、Cryostat カスタムリソース(CR)の **.spec.authorizationOptions.openShiftSSO.accessReview** フィールドを使用して、Cryostat にアクセスするために必要な RBAC パーミッションをカスタマイズできます。

この拡張機能は、異なるユーザーアカウントに異なるレベルの認証を設定できる以前のリリースの動作に取って代わります。

Cryostat CR 検証の機能強化

Cryostat Operator は、これらのオブジェクトを処理のために受け入れる前に、Cryostat CR オブジェクトに対して追加の検証チェックを実行するようになりました。注目すべき検証チェックの1つは、ターゲット namespace のリストを使用して Cryostat CR を作成するユーザーに、これらのターゲット namespace にシングル namespace の Cryostat CR を作成するための十分な権限があることです。

Cryostat Helm チャートの設定の強化

Cryostat Helm チャートに次の設定パラメーターを設定できるようになりました。

- **authentication.openshift.enabled**

このプロパティは **openshift-oauth-proxy** のデプロイメントを有効にし、デフォルトで無効にされます。このプロパティが無効な場合には、**oauth2_proxy** は代わりにデプロイされます。



注記

authentication.**basicAuth** プロパティを使用して Basic 認証を有効にするために **openshift-oauth-proxy** と **oauth2-proxy** の両方を設定できます。ただし、Red Hat OpenShift に Cryostat をデプロイする場合は、**openshift-oauth-proxy** は Red Hat OpenShift クラスター SSO との統合もサポートします。

- **authentication.basicAuth**

このプロパティは、認証プロキシに Basic 認証を設定します。**openshift-oauth-proxy** のデプロイメントを有効にする場合、この Basic 認証は Red Hat OpenShift SSO に追加されます。**oauth2_proxy** のデプロイメントを有効にすると、この基本認証は、使用可能な唯一のユーザー認証メカニズムです。

- **openshiftOauthProxy.accessReview**

このプロパティは、Red Hat OpenShift SSO を介した Cryostat へのクライアントアクセスをテストするための **SubjectAccessReview** を設定します。

設定パラメーターの完全なリストは、[Cryostat Helm Chart readme ファイル](#) を参照してください。

Cryostat エージェント組み込み Web サーバー

Cryostat エージェントの起動時に、エージェントは組み込み Web サーバーを起動します。これは、Cryostat サーバーからの要求を処理するために使用されます。Basic 認証を使用して、埋め込み Web サーバーは自身のセキュリティを保護します。

以前のリリースでは、基本的なユーザー名は常に **ユーザー** で、ランダムに生成されるパスワードは 24 つの ASCII 文字で設定されていました。Cryostat 3.0 では、デフォルトのユーザー名は **user** で、デフォルトのパスワードの長さは 24 文字ですが、ユーザー名とパスワードの長さはどちらも設定可能です。このリリースでは、無作為に生成されるパスワードも、以前のリリースと比較して大きな文字セットに基づいています。

Cryostat エージェントポートの強化

Cryostat エージェントを使用するようにアプリケーションを設定する場合、エージェントのベース URI はデフォルトでポート 4180 を使用するようになりました。これは、エージェントのベース URI がポート 8181 を使用していた以前のリリースの動作に優先します。

この機能拡張は、Cryostat 3.0 の [リバースプロキシアーキテクチャー](#) が導入されたためです。ポート 4180 は、認証プロキシの HTTP ポートであり、承認されたリクエストを Cryostat に渡します。したがって、Cryostat HTTP ポートがプロキシの背後で非表示になっているため、Cryostat エージェントをポート 4180 に直接ポート 4180 に送信するように設定する必要があります。

第4章 サポートされない機能および非推奨の機能

Cryostat 3.0 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。

ターゲット TLS 証明書のアップロード

Cryostat Web コンソールの Security ビューは、SSL/TLS 証明書を Cryostat サーバートラストストアに直接アップロードする方法を提供しなくなりました。Security ビューには、すでにロードされている証明書の一覧のみが表示されます。

Cryostat 3.0 以降では、Cryostat が起動時に読み取るストレージボリュームに新しい証明書を追加する必要があります。Cryostat CR で既存の **TrustedCertSecrets** プロパティを使用して、新しい証明書を設定できます。

API リクエスト経由で渡される JMX ターゲットクレデンシャル

X-JMX-Authorization ヘッダーに対応しなくなりました。つまり、Cryostat はターゲットアプリケーションからの API リクエストを受け入れて、Cryostat が自身を認証して、アプリケーションへの JMX 接続期間中、認証情報をメモリーに保存できることを意味します。

Cryostat 3.0 以降では、ターゲットアプリケーションの JMX クレデンシャルは常に、Red Hat OpenShift の永続ボリュームクレーム(PVC)に保存される暗号化されたデータベースに保存されます。Cryostat Web コンソールの Settings ビューには、使用する認証メカニズムを選択する高度な設定が提供されなくなりました。

Cryostat の自動検出

Cryostat Operator または Helm チャートを使用して Cryostat をインストールすると、Cryostat はデフォルトでターゲットアプリケーションとして検出されなくなります。以前のリリースでは、Cryostat は Kubernetes サービスと Cryostat Operator で JMX ポートを公開し、このポートのセキュリティーを保護するための TLS 証明書が割り当てられていました。

Cryostat 3.0 以降では、Cryostat によって公開される JMX ポートが無効になり、対応する Service ポートが削除されます。つまり、Cryostat は接続可能なターゲットとして自身を検出できなくなります。これは、Cryostat がターゲット選択リストまたは Cryostat Web コンソールの Topology ビューに表示されないことも意味します。



注記

Cryostat を自身に接続してパフォーマンスを確認する場合は、URL 値が **localhost:0** の **Custom Target** を作成できます。この値は、ネットワークにポートを公開せずにローカル JMX 接続を開くよう JVM に指示します。つまり、追加の認証や TLS 暗号化は必要ありません。

単一の namespace 用の Cryostat Operator のインストール

単一の namespace またはクラスター namespace のサブセットに Cryostat Operator をインストールするためのサポートが提供されなくなりました。

Cryostat 3.0 以降では、Cryostat Operator はクラスター全体でのみインストールできます。クラスター全体のインストールは、Operator Lifecycle Manager の推奨モードであり、namespace ごとのインストールは非推奨の機能です。

Cluster Cryostat API

Cluster Cryostat API はサポートされなくなりました。このリリースでは、Cryostat Operator を使用して Cryostat インスタンスをインストールするときに、**Details** タブの Provided APIs セクションで **Cluster Cryostat** オプションを選択できなくなりました。

Crio 3.0 以降では、**Crio** API を使用して、シングル namespace とマルチ namespace の両方の Crio インスタンスを作成できます。Crio Operator を使用して Crio インスタンスをインストールすると、**Crio** API でターゲット namespace のオプションのリストを指定できるようになりました。

第5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。

URL パスで直接アクセスされると /recordings/create などのサブビューにアクセスできない

説明

`/topology/create-custom-target`、`/rules/create`、`/recordings/create` などのサブビューを含む URL パスを介してコンソールにアクセスしようとする、Cryostat Web コンソールには Web ブラウザーからアクセスできません。

たとえば、`https://my_cryostat_domain/recordings/create` などの URL パスを入力すると、Cryostat コンソールには空のページが表示されます。

回避策

URL パスでサブビューを指定しないでください。

たとえば、`https://my_cryostat_domain/recordings/create` にアクセスする場合は、Web ブラウザーに `https://my_cryostat_domain/recordings` の URL パスを入力し、Cryostat Web コンソールで **Create** をクリックします。

replace=always パラメーターでレコーディングを再起動すると、アクティブな Recordings テーブルが更新に失敗する

説明

クライアントが **replace=always** パラメーターを含むリクエストを送信して既存のレコーディングを再作成すると、Cryostat Web コンソールの Active Recordings テーブルが更新されて、新しいレコーディングの詳細が表示されません。**Recording created** 通知が表示されても、新しいレコーディングは Active Recordings テーブルに自動的に表示されません。

回避策

Active Recordings ページを再度読み込んでから現在のページに戻ります。Active Recordings テーブルには、新しいレコーディングが正しく表示されます。

第6章 このリリースに関連するアドバイザリー

Cryostat 3.0 リリースに含まれるバグ修正と CVE 修正を文書化するために、次のアドバイザリーが発行されています。

- [RHSA-2024:4173](#)

改訂日時: 2024-07-02