Red Hat build of Cryostat 2

MBean カスタムトリガーに基づく動的 JFR レコーディングの有効化

Last Updated: 2024-01-05

Red Hat build of Cryostat 2 MBean カスタムトリガーに基づく動的 JFR レコーディングの有効化

法律上の通知

概要

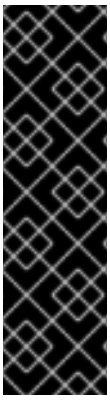
MBean カスタムトリガーに基づいて JFR レコーディングを動的に開始するように Cryostat エージェントを設定します。

目次

はじめに

Red Hat build of Cryostat は、JDK Flight Recorder (JFR) のコンテナー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、コンテナー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

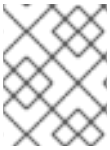
Red Hat では、コード、ドキュメント、Web プロパティーにおける配慮に欠ける用語の置き換えに取り組んでいます。まずは、マスター (master)、スレーブ (slave)、ブラックリスト (blacklist)、ホワイトリスト (whitelist) の 4 つの用語の置き換えから始めます。この取り組みは膨大な作業を要するため、今後の複数のリリースで段階的に用語の置き換えを実施して参ります。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

第1章 カスタムトリガー

Cryostat 2.4 エージェントは、MBean メトリクス値に基づくカスタムトリガーをサポートします。カスタムトリガー条件が満たされたときに JFR レコーディングを動的に開始するように Cryostat エージェントを設定できます。

この条件が満たされたときに JFR レコーディングを動的に開始するカスタムトリガー条件を定義できます。カスタムトリガー条件は、ランタイム、メモリー、スレッド、オペレーティングシステムのメトリックの範囲に対応できる MBean カウンターに基づいています。JFR レコーディングのカスタムトリガー条件の一部として 1 つ以上の MBean カウンタータイプを含めることができます。トリガー条件の一部として期間を指定することもできます。つまり、条件が満たされるまで、条件値は指定された期間存続する必要があります。

Cryostat エージェントは、指定された MBean カウンターの値を継続的にリッスンするスマートトリガーをサポートします。指定されたカウンターの現在の値が、指定された期間のカスタムトリガの設定値と一致すると、トリガが発生します。トリガーが発生すると、Cryostat エージェントはその時点で JFR レコーディングを動的に開始します。



注記

このレコーディングに関連するカスタムトリガー条件が満たされない場合、JFR レコーディングは動的に開始されません。

第2章 動的レコーディングのカスタムトリガーの設定

Cryostat エージェントをロードするようにターゲットアプリケーションを設定する場合に、エージェントに引数として渡される 1 つ以上のカスタムトリガーを定義できます。

Cryostat エージェントをロードするようにターゲットアプリケーションを設定する方法は、Java アプリケーションの設定 を参照してください。

2.1. カスタムトリガーを定義するオプション

カスタムトリガーは以下のいずれかの方法で定義できます。

Cryostat エージェントの JAR ファイルパスへのカスタムトリガーの追加

次の例は、Cryostat エージェントの JAR ファイルパスに単純なカスタムトリガーを追加する方法を示しています。

```
JAVA_OPTS="-javaagent:/deployments/app/cryostat-agent-shaded.jar=\"[ProcessCpuLoad > 0.2 ; TargetDuration > duration('30s')]~profile\""
```

前述のトリガーの例は、**ProcessCpuLoad** メトリクスの値が 30 秒以上、0.2 より大きい場合に、JFR レコーディングを開始するようにエージェントに指示します。この例は、エージェントに、JFR レコーディングに **profile** イベントテンプレートを使用するように指示します。

JVM システムプロパティーフラグの使用

以下の例は、JVM システムプロパティーフラグを使用して簡単なカスタムトリガーを指定する方法を示しています。

```
-Dcryostat.agent.smart-trigger.definitions="[ProcessCpuLoad > 0.2 ; TargetDuration > duration(\"30s\")]~profile"
```

この例では、前述の例と同じカスタムトリガー条件を使用しています。

環境変数の使用

次の例は、環境変数を使用して簡単なカスタムトリガーを指定する方法を示しています。

```
- name: CRYOSTAT_AGENT_SMART_TRIGGER_DEFINITIONS
  value: "[ProcessCpuLoad > 0.2 ; TargetDuration > duration(\"30s\")]~profile"
```

この例では、前述の例と同じカスタムトリガー条件を使用しています。

2.2. COMMON EXPRESSION LANGUAGE

Common Expression Language (CEL) を使用して、カスタムトリガー条件を定義できます。CEL は、データを評価するためのルールと制約を柔軟に定義できる自由形式の式構文です。たとえば、CEL を使用して、MBean カウンタータイプの組み合わせの現在の値が、指定した設定可能な値以上か、以下かを評価するためのリレーショナルステートメントを作成できます。同じトリガー条件に含まれる、異なる MBean カウンタータイプ間で、AND (**&&**) または OR (||) 論理ステートメントの任意の組み合わせを含めることもできます。

CEL の詳細は、CEL 言語の仕様 を参照してください。

2.3. カスタムトリガーの一般的な構文ルール

カスタムトリガーを定義するための次の構文ガイドラインを考慮してください。

- カスタムトリガー定義は、全体的なトリガー条件と、JFR レコーディングに使用されるイベントテンプレートの名前の両方で設定される必要があります。
- トリガー式全体を角括弧で囲む必要があります (例: **ProcessCpuLoad > 0.2 ; TargetDuration <duration ("30s")**)。
- 前の例に示すように、トリガー式に空白を使用して可読性を高めることはできますが、これは必須ではありません。
- イベントテンプレートの名前は、トリガー式の後に定義し、その前にチルダ (~) 文字を付ける必要があります (例: ~**profile**)。
- トリガー式は、1 つ以上の制約とターゲット期間で設定できます。一連の制約とターゲット期間はセミコロン (**;**) 文字で区切る必要があります。
- 各制約には、MBean カウンターの名前、**>** (より大きい)、**=** (等しい)、**<** (より小さい) などの関係演算子、および指定された値など、関係演算子などを含める必要があります。指定できる関係演算子のタイプと値は、関連付けられた MBean カウンターのタイプ (例: **ProcessCpuLoad > 0.2**) によって異なります。
- **&&** (AND)、|| (OR)、または **!** などの論理演算子を使用して制約をグループ化できます。(NOT) ロジック。操作の順序と演算子の優先順位を読みやすく明確にするために、グループ化された制約を丸括弧で囲む場合がありますが、これは必須ではありません。以下に例を示します。

  > [(**MetricA** > **value1** && **MetricB** < **value2**) || **MetricC** == '**stringvalue**' ; TargetDuration > duration("30s")]

- カスタムトリガーの一部として指定される各 MBean カウンターの名前は、スペルと大文字の使用に関して正確な構文規則に従う必要があります。指定できる MBean メトリクスの完全なリストは、MBean カウンタータイプ を参照してください。
- カスタムトリガーに対して定義できるターゲット期間は 1 つだけです。対象の期間は、角かっこで囲まれたトリガー式全体に適用されます。
- 対象の間は、秒、分、または時間で表すことができます。たとえば、**30 秒** は 30 秒、**5 分** は 5 分、**2h** は 2 時間を意味します。
- 対象の期間は任意です。対象の期間が指定されていない場合、トリガー条件が満たされるとすぐにトリガーが発生します。
- 複数のカスタムトリガー定義を一緒に指定でき、それぞれが個別の JFR レコーディングに関連します。異なるカスタムトリガー定義はコンマ (**,**) 文字で区切る必要があります。以下に例を示します。

  > [ProcessCpuLoad>0.2]~profile,[ThreadCount>30]~Continuous

第3章 動的な JFR レコーディングの自動起動

Cryostat エージェントが JFR レコーディングを開始できるように設定されており、動的レコーディングのカスタムトリガー条件が満たされると、Cryostat エージェントはターゲットアプリケーション内からレコーディングを自動的に開始します。

Cryostat エージェントは、JFR レコーディングに自動的に名前を割り当てます。この名前は常に **cryostat -smart-trigger -X** 形式になります (**X** は、レコーディング ID を表します)。JVM は、レコーディング ID を自動的に生成します。これは、JVM 内で開始される各 JFR レコーディングに固有の増分数値です。

Cryostat エージェントが動的 JFR 記録を開始すると、その後、Cryostat Web コンソールの **Active Recordings** タブでこのレコーディングを表示できます。**Active Recordings** タブの使用の詳細は、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。

第4章 動的な JFR レコーディングの手動停止

Cryostat エージェントは、現在、動的 JFR レコーディングの自動停止をサポートしていません。このリリースでは、記録をトリガーした条件が満たされなくなった場合でも、動的な JFR レコーディングは停止しません。この状況で、動的な JFR レコーディングを停止する場合は、Cryostat Web コンソールの **Active Recordings** タブから記録を手動で停止する必要があります。

Cryostat Web コンソールを使用して JFR 記録を停止する方法の詳細は、Cryostat を使用した JFR 記録の作成 を参照してください。

第5章 同じカスタムトリガー定義に基づく複数の JFR レコーディング

Cryostat 2.4 エージェントは、カスタムトリガー定義ごとに 1 回だけ JFR 記録を動的に開始できます。このリリースでは、Cryostat エージェントは、同じカスタムトリガー条件に対して複数の JFR レコーディングを定期的に開始できません。Cryostat エージェントが特定のカスタムトリガー定義の JFR レコーディングを開始すると、エージェントは残りのエージェントセッションでこのトリガー定義を無視します。

この状況で、以前に記録をトリガーしたカスタムトリガー条件に基づいて Cryostat エージェントが新しい JFR レコーディングを開始できるようにする場合は、Cryostat エージェントを再起動する必要があります。

第6章 アーカイブに使用するエージェント HARVESTER と動的 JFR レコーディングの統合

Cryostat エージェントが MBean カスタムトリガーに基づいて動的 JFR レコーディングを開始できるようにすると、これらの JFR レコーディングをエージェント harvester システムと統合することもできます。このように統合すると、MBean カスタムトリガーから得られる JFR レコーディングデータが定期的に JFR スナップショットにキャプチャーされ、設定された harvester のスケジュールに基づいてアーカイブのために Cryostat サーバーにプッシュされます。

エージェント harvester の期間を使用した MBean カスタムトリガー

エージェント harvester は、もう 1 つの設定可能な機能であり、Cryostat エージェントが、指定されたイベントテンプレートに基づいて、エージェントの起動時に JFR レコーディングを自動的に開始できるようにします。エージェント harvester には、レコーディングのスナップショットをキャプチャーして Cryostat サーバーにアップロードするスケジュールを定義するために使用できる設定可能なプロパティーが含まれています。

harvester テンプレートを使用せずに MBean カスタムトリガーとエージェントハーベスター期間を定義すると、エージェントが次の両方を実行する設定を実現できます。

- エージェントは、MBean カスタムトリガーに基づいて JFR レコーディングを動的に開始します。
- Aent は、設定された harvester の期間を使用して、レコーディングデータのスナップショットを定期的にキャプチャーし、このデータを Cryostat サーバーにアップロードします。

このような状況では、動的 JFR レコーディングを手動で停止するか、ホスト JVM がシャットダウンするまで、エージェントはレコーディングデータをキャプチャーし続けます。

エージェント harvester の期間設定

Cryostat エージェントをロードするようにターゲットアプリケーションを設定する場合に、定期的な JFR レコーディングデータのアップロードを可能にするエージェント harvester 期間を設定することもできます。harvester 期間の値は、ミリ秒単位で指定できます。デフォルトでは、Cryostat エージェントは、JFR レコーディングデータのスケジュールされた収集アップロードを実行できません。

エージェント harvester の期間は、次のいずれかの方法で設定できます。

JVM システムプロパティーフラグの使用

次の例は、JVM システムプロパティーフラグを使用して harvester の期間を設定する方法を示しています。

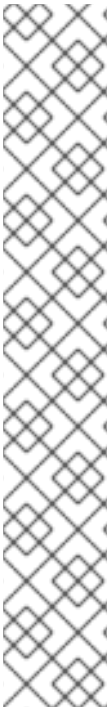
```
-Dcryostat.agent.harvester.period-ms=1000
```

環境変数の使用

次の例は、環境変数を使用して harvester の期間を設定する方法を示しています。

```
- name: CRYOSTAT_AGENT_HARVESTER_PERIOD_MS
  value: 1000
```

前述の例では、harvester の期間値 1000 を示しています。この例に基づいて、エージェントは、アーカイブ用の JFR レコーディングデータを 1000 ミリ秒ごと (一定の 1 秒間隔) にアップロードします。



注記

Cryostat は、JFR レコーディングを開始する次のさまざまな方法をそれぞれサポートしています。

- Cryostat Web コンソールからレコーディングを手動で開始できます。
- Cryostat エージェントは、MBean のカスタムトリガーに基づいてレコーディングを動的に開始できます。
- Cryostat エージェントは、指定された harvester テンプレートに基づいて、エージェントの起動時に自動的にレコーディングを開始できます。
- Cryostat サーバーは、JMX またはエージェント HTTP 接続経由でオンデマンドリクエストを送信し、自動化されたルールに基づいてレコーディングを開始できます。

この状況では、システム内で JFR レコーディングがどの方法で開始されたかに関係なく、エージェント harvester 設定により、すべての JFR レコーディングデータのキャプチャーとアップロードが制御されます。

第7章 カスタム MBEAN トリガーの評価期間の設定

Cryostat エージェントは、指定された MBean カウンターの現在値を継続的にリッスンするスマートトリガーをサポートします。カウンターの現在の値は、カスタムトリガー定義で指定できます。トリガー条件は、定期的な設定可能な間隔でポーリングに基づき評価されます。デフォルトでは、トリガー条件は通常の 1 秒間隔で評価されます。

評価期間 (ポーリング頻度) を 1 秒に 1 回に指定すると、条件が満たされてからエージェントがこの条件が満たされたことを評価できるまでの間に最大 1 秒の時間遅延が生じる可能性があります。

Cryostat エージェントをロードするようにターゲットアプリケーションを設定する場合、MBean カスタムトリガーに対して別の評価期間を設定できます。評価期間の値をミリ秒単位で指定できます。

評価期間は、以下のいずれかの方法で設定できます。

JVM システムプロパティーフラグの使用

以下の例は、JVM システムプロパティーフラグを使用して評価期間を設定する方法を示しています。

```
-Dcryostat.agent.smart-trigger.evaluation.period-ms=500
```

環境変数の使用

次の例は、環境変数を使用して harvester の期間を設定する方法を示しています。

```
- name: CRYOSTAT_AGENT_SMART-TRIGGER_EVALUATION_PERIOD_MS
  value: 500
```

前述の例では、評価期間値 500 を示しています。この例に基づいて、トリガー条件は 500 ミリ秒ごと (つまり、一定の 0.5 秒間隔) で評価されます。

第8章 MBEAN カウンタータイプ

カスタムトリガー定義の一部として指定できる MBean カウンタータイプは、JDK で一般にデフォルトで使用できる標準 MBean メトリクスです。これらの MBean カウンターは、ランタイム、メモリー、スレッド、オペレーティングシステムのさまざまなメトリクスに対応します。

以下は、カスタムのトリガー定義に指定できる MBean カウンタータイプの完全なリストです。

注記

カスタムトリガー定義の一部として指定する各 MBean カウンターの名前は、次のリストの MBean カウンター名のスペルおよび大文字小文字の区別と正確に一致する必要があります。

オペレーティングシステムのメトリクス

- Arch
- AvailableProcessors
- Name
- SystemLoadAverage
- バージョン
- CommittedVirtualMemorySize
- FreePhysicalMemorySize
- FreeSwapSpaceSize
- ProcessCpuLoad
- ProcessCpuTime
- SystemCpuLoad
- TotalPhysicalMemorySize
- TotalSwapSpaceSize

スレッドメトリクス

- AllThreadIds
- CurrentThreadCpuTime
- CurrentThreadUserTime
- DaemonThreadCount
- PeakThreadCount
- ThreadCount

- TotalStartedThreadCount
- CurrentThreadCpuTimeSupported
- ObjectMonitorUsageSupported
- SynchronizerUsageSupported
- ThreadContentionMonitoringEnabled
- ThreadContentionMonitoringSupported
- ThreadCpuTimeEnabled
- ThreadCpuTimeSupported

ランタイムメトリクス

- BootClassPathSupported
- BootClassPath
- ClassPath
- InputArguments
- LibraryPath
- ManagementSpecVersion
- 名前
- SpecName
- SpecVersion
- SystemProperties
- StartTime
- Uptime
- VmName
- VmVendor
- VmVersion

メモリーメトリクス

- HeapMemoryUsage
- NonHeapMemoryUsage
- ObjectPendingFinalizationCount
- FreeHeapMemory

- FreeNonHeapMemory
- HeapMemoryUsagePercent
- Verbose

改訂日時: 2024-01-02