



Red Hat build of Cryostat 2

启用基于 MBean 自定义触发器的动态 JFR 记录

Red Hat build of Cryostat 2 启用基于 MBean 自定义触发器的动态 JFR 记录

法律通告

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.

摘要

将 Cryostat 代理配置为根据 MBean 自定义触发器动态启动 JFR 记录。

目录

前言	3
使开源包含更多	4
第 1 章 自定义触发器	5
第 2 章 为动态记录配置自定义触发器	6
2.1. 定义自定义触发器的选项	6
2.2. 常见表达式语言	6
2.3. 自定义触发器的一般语法规则	6
第 3 章 自动启动动态 JFR 记录	8
第 4 章 手动停止动态 JFR 记录	9
第 5 章 基于同一自定义触发器定义的多个 JFR 记录	10
第 6 章 动态 JFR 记录与代理集成以进行归档	11
第 7 章 为自定义 MBEAN 触发器配置评估周期	13
第 8 章 MBEAN 计数器类型	14

前言

Red Hat build of Cryostat 是 JDK Flight Recorder (JFR)的容器原生虚拟化实现，可用于安全地监控 OpenShift Container Platform 集群上运行的工作负载的 Java 虚拟机(JVM)性能。您可以使用 Cryostat 2.4 使用 web 控制台或 HTTP API 启动、停止、检索、存档、导入和导出容器化应用程序中 JVM 的 JFR 数据。

根据您的用例，您可以使用 Cryostat 提供的内置工具直接在 Red Hat OpenShift 集群上存储和分析记录，或者您可以将记录导出到外部监控应用程序，以对记录的数据进行更深入的分析。



重要

Red Hat build of Cryostat 只是一个技术预览功能。技术预览功能不受红帽产品服务等级协议 (SLA) 支持，且功能可能并不完整。红帽不推荐在生产环境中使用它们。这些技术预览功能可以使用户提早试用新的功能，并有机会在开发阶段提供反馈意见。

有关红帽技术预览功能支持范围的更多信息，请参阅[技术预览功能支持范围](#)。

使开源包含更多

红帽致力于替换我们的代码、文档和 Web 属性中有问题的语言。我们从这四个术语开始：master、slave、黑名单和白名单。由于此项工作十分艰巨，这些更改将在即将推出的几个发行版本中逐步实施。有关更多详情，请参阅[我们的首席技术官 Chris Wright 提供的消息](#)。

第 1 章 自定义触发器

Cryostat 2.4 代理支持基于 MBean 指标值的自定义触发器。您可以将 Cryostat 代理配置为在满足这些自定义触发器条件时动态启动 JFR 记录。

您可以定义在满足此条件时动态启动 JFR 记录的自定义触发器条件。自定义触发器条件基于 MBean 计数器，可覆盖一系列运行时、内存、线程和操作系统指标。您可以包含一个或多个 MBean 计数器类型，作为 JFR 记录的自定义触发器条件的一部分。您还可以指定一个持续时间或时间段作为触发器条件的一部分，这意味着满足条件前必须在指定持续时间内保留条件值。

Cryostat 代理支持智能触发器，持续侦听指定 MBean 计数器的值。如果指定计数器的当前值与指定持续时间的自定义触发器中配置的值匹配，则会触发。如果发生触发，Cryostat 代理会在此时动态启动 JFR 记录。



注意

如果没有满足与此记录关联的自定义触发器条件，则不会动态启动 JFR 记录。

第 2 章 为动态记录配置自定义触发器

当您将目标应用程序配置为加载 Cryostat 代理时，您可以定义一个或多个自定义触发器，然后作为参数传递给代理。

有关配置目标应用程序以加载 Cryostat 代理的更多信息，请参阅[配置 Java 应用程序](#)。

2.1. 定义自定义触发器的选项

您可以使用以下任何方法定义自定义触发器：

将自定义触发器附加到 Cryostat 代理的 JAR 文件路径

以下示例演示了如何将简单的自定义触发器附加到 Cryostat 代理的 JAR 文件路径：

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

如果 **ProcessCpuLoad** 指标的值大于 30 秒，则前面的示例触发器指示代理启动 JFR 记录，该记录的值超过 30 秒：本例还指示代理使用 JFR 记录的[配置集](#) 事件模板。

使用 JVM 系统属性标志

以下示例演示了如何使用 JVM 系统属性标志指定简单的自定义触发器：

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

本例使用与上例相同的自定义触发器标准。

使用环境变量

以下示例演示了如何使用环境变量指定简单的自定义触发器：

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

本例使用与前面示例相同的自定义触发器标准。

2.2. 常见表达式语言

您可以使用通用表达式语言(CEL)来定义自定义触发器条件。CEL 是一个自由格式表达式语法，在定义用于评估数据的规则和约束方面提供了很大的灵活性。例如，您可以使用 CEL 创建关系语句来评估 MBean 计数器类型的任何组合是否有大于、等于或小于指定的可配置值。您还可以包括属于同一触发器条件的不同 MBean 计数器类型之间的 AND (&)&或 OR (||)逻辑语句的任意组合。

有关 CEL 的更多信息，请参阅[CEL 语言规格](#)。

2.3. 自定义触发器的一般语法规则

请考虑以下语法规则来定义自定义触发器：

- 自定义触发器定义必须包含定义总体触发器条件和用于 JFR 记录的事件模板名称的表达式。

- 整个触发器表达式必须用方括号括起（例如，`[ProcessCpuLoad > 0.2 ; TargetDuration < duration("30s")]`）。
- 为便于阅读，您可以在触发器表达式中使用空格，如上例中所示，但这不是必需的。
- 事件模板的名称必须在触发器表达式后定义，并以波形符(~)字符（如 `~profile`）前面定义。
- 触发器表达式可由一个或多个约束和目标持续时间组成。约束和目标持续时间集合必须以分号(;)字符分隔。
- 每个约束都必须包括：MBean 计数器的名称；一个关系运算符，如 `>` (greater than), `=`（等于）、`<`（不少于）等），以及指定的值。可以指定关系运算符和值的类型和值取决于关联的 MBean 计数器类型（如 `ProcessCpuLoad > 0.2`）。

- 可以使用一个逻辑运算符（如 `&&` (AND)、`||` (OR)）或 `!`（注意）逻辑。为了提高操作顺序和运算符优先级的可读性和清晰性，分组的限制可能会用圆括号括起来，但这并不是要求。例如：

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

- 作为自定义触发器的一部分指定的每个 MBean 计数器的名称必须在拼写和大写方面遵循精确的语法规则。有关您可以指定的 MBean 指标的完整列表，请参阅 [MBean 计数器类型](#)。

- 对于自定义触发器，只能定义一个目标持续时间。目标持续时间应用到包含在方括号中的整个触发器表达式。

- 目标持续时间可以用秒、分钟或小时表示。例如，`30s` 表示 30 秒，`5m` 表示五分钟，`2h` 表示两个小时，以此类推。

- 目标持续时间是可选的。如果没有指定目标持续时间，则触发器会在满足触发器条件后立即发生。

- 可以一起指定多个自定义触发器定义，每个定义都与单独的 JFR 记录相关。不同的自定义触发器定义必须用逗号分开(,)字符。例如：

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

第 3 章 自动启动动态 JFR 记录

当 Cryostat 代理启用 JFR 记录并满足动态记录的自定义触发器条件时，Cryostat 代理会在目标应用程序中自动启动记录。

Cryostat 代理自动为 JFR 记录分配一个名称，该记录始终为 `cryostat-smart-trigger-X` 格式，其中 `X` 代表记录 ID。JVM 会自动生成记录 ID，它是 JVM 中启动的每个 JFR 记录的唯一增量数字值。

当 Cryostat 代理启动动态 JFR 记录时，您可以在 Cryostat web 控制台的 **Active Recordings** 选项卡中查看这个记录。有关使用 **Active Recordings** 选项卡的更多信息，[请参阅使用 Cryostat 创建 JFR 记录](#)。

第 4 章 手动停止动态 JFR 记录

Cryostat 代理目前不支持自动停止动态 JFR 记录。在本发行版本中，即使不再满足触发记录的条件，也会停止动态 JFR 记录。在这种情况下，如果您想要动态 JFR 记录停止，则必须从 Cryostat web 控制台中的 Active Recordings 选项卡中手动停止记录。

有关使用 Cryostat web 控制台停止 JFR 记录的更多信息，[请参阅使用 Cryostat 创建 JFR 记录。](#)

第 5 章 基于同一自定义触发器定义的多个 JFR 记录

Cryostat 2.4 代理只能为每个自定义触发器定义动态启动 JFR 记录一次。在本发行版本中，Cryostat 代理无法重复为同一自定义触发器条件启动多个 JFR 记录。当 Cryostat 代理为特定自定义触发器定义启动 JFR 记录后，代理会忽略这个触发器定义。

在这种情况下，如果要启用 Cryostat 代理，根据之前触发记录的自定义触发器条件启动新的 JFR 记录，您必须重启 Cryostat 代理。

第 6 章 动态 JFR 记录与代理集成以进行归档

当您启用 Cryostat 代理来基于 MBean 自定义触发器启动动态 JFR 记录时，您还可以将这些 JFR 记录与代理 harvester 系统集成。此集成意味着，任何由 MBean 自定义触发器生成的 JFR 记录数据都会定期捕获在 JFR 快照中，并推送到 Cryostat 服务器进行存档，具体取决于 harvester 的调度。

带有代理 harvester 周期的 MBean 自定义触发器

代理 harvester 是另一个可配置的功能，它允许 Cryostat 代理根据给定事件模板在代理启动时自动启动 JFR 记录。代理 harvester 包含一个可配置的属性，您可以使用它来定义捕获和上传记录快照到 Cryostat 服务器的调度。

在没有 harvester 模板的情况下定义 MBean 自定义触发器和代理 harvester 周期，您可以实现代理执行以下操作的设置：

- 代理动态根据 MBean 自定义触发器启动 JFR 记录。
- Aent 使用配置的 harvester 周期定期捕获记录数据的快照，并将这些数据上传到 Cryostat 服务器。

在这种情况下，代理将继续捕获记录数据，直到您手动停止动态 JFR 记录或主机 JVM 关闭。

配置代理 harvester 周期

当您为目标应用程序配置为加载 Cryostat 代理时，您还可以配置代理 harvester 周期，以启用 JFR 记录数据的定期上传。您可以以毫秒为单位指定 harvester 周期的值。默认情况下，Cryostat 代理没有启用来执行任何调度的 JFR 记录数据上传。

您可以使用以下方法之一配置代理 harvester 周期：

使用 JVM 系统属性标志

以下示例演示了如何使用 JVM 系统属性标志配置 harvester 周期：

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

使用环境变量

以下示例演示了如何使用环境变量配置 **harvester** 周期：

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

前面的示例显示 **harvester period** 值为 **1000**。根据这个示例，代理上传 **JFR** 记录数据以每 **1000** 毫秒归档一次（即，常规的 **1** 秒间隔）。

注意

Cryostat 支持使用以下不同方法启动 **JFR** 记录：

- 您可以从 **Cryostat web** 控制台手动启动记录。
- **Cryostat** 代理可以根据 **MBean** 自定义触发器动态开始记录。
- **Cryostat** 代理可根据给定的 **harvester** 模板在代理启动时自动启动记录。
- **Cryostat** 服务器可以通过 **JMX** 或代理 **HTTP** 连接发送按需请求，以便根据自动规则开始记录。

在这种情况下，代理制裁设置控制捕获和上传所有 **JFR** 记录数据，无论系统启动 **JFR** 记录的方式。

第 7 章 为自定义 MBean 触发器配置评估周期

Cryostat 代理支持智能触发器，持续侦听您可以在自定义触发器定义中定义的指定 MBean 计数器的当前值。触发器条件在轮询上以常规可配置的间隔评估。默认情况下，触发器条件会定期评估 1 秒间隔。

每秒一次评估周期（转换频率）意味着满足条件和代理可以评估满足此条件之间最多有 1 秒的时间延迟。

当您为目标应用程序配置为加载 **Cryostat** 代理时，您可以选择为 MBean 自定义触发器配置不同的评估周期。您可以以毫秒为单位指定评估周期的值。

您可以使用以下方法之一配置评估周期：

使用 JVM 系统属性标志

以下示例演示了如何使用 JVM 系统属性标志配置评估周期：

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

使用环境变量

以下示例演示了如何使用环境变量配置 **harvester** 周期：

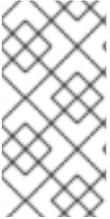
```
- name: CRYOSTAT_AGENT_SMART-TRIGGER_EVALUATION_PERIOD_MS  
value: 500
```

前面的示例显示了一个评估周期值 500。根据这个示例，触发器条件会每 500 毫秒评估一次（即，常规的半秒的间隔）。

第 8 章 MBean 计数器类型

您可以作为自定义触发器定义的一部分指定的 MBean 计数器类型是 JDK 中通常可用的标准 MBean 指标。这些 MBean 计数器涵盖运行时、内存、线程和操作系统指标的范围。

以下是您可以在自定义触发器定义中指定的 MBean 计数器类型的完整列表：



注意

您在自定义触发器定义中指定的每个 MBean 计数器的名称必须与以下列表中 MBean 计数器名称的拼写和大写完全匹配。

操作系统指标

- **arch**
- **AvailableProcessors**
- **Name**
- **SystemLoadAverage**
- **Version**
- **CommittedVirtualMemorySize**
- **FreePhysicalMemorySize**
- **FreeSwapSpaceSize**
- **ProcessCpuLoad**

- **ProcessCpuTime**
- **SystemCpuLoad**
- **TotalPhysicalMemorySize**
- **TotalSwapSpaceSize**

线程指标

- **AllThreadIds**
- **CurrentThreadCpuTime**
- **CurrentThreadUserTime**
- **DaemonThreadCount**
- **PeakThreadCount**
- **ThreadCount**
- **TotalStartedThreadCount**
- **CurrentThreadCpuTimeSupported**
- **ObjectMonitorUsageSupported**
- **SynchronizerUsageSupported**

- **ThreadContentionMonitoringEnabled**
- **ThreadContentionMonitoringSupported**
- **ThreadCpuTimeEnabled**
- **ThreadCpuTimeSupported**

运行时指标

- **BootClassPathSupported**
- **BootClassPath**
- **classpath**
- **InputArguments**
- **LibraryPath**
- **ManagementSpecVersion**
- **Name**
- **SpecName**
- **SpecVersion**
- **SystemProperties**

- **StartTime**
- **uptime**
- **vmName**
- **VmVendor**
- **VmVersion**

内存指标

- **HeapMemoryUsage**
- **NonHeapMemoryUsage**
- **ObjectPendingFinalizationCount**
- **FreeHeapMemory**
- **FreeNonHeapMemory**
- **HeapMemoryUsagePercent**
- 详细

更新于 2024-01-02

