



Red Hat Satellite 6.15

监控 Satellite 性能

从 Satellite 收集指标，并允许它们在外部工具中分析

Red Hat Satellite 6.15 监控 Satellite 性能

从 Satellite 收集指标，并允许它们在外部工具中分析

Red Hat Satellite Documentation Team

satellite-doc-list@redhat.com

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

本指南论述了如何从 Red Hat Satellite 6 收集指标进行分析。它是面向 Satellite 管理员的。

目录

对红帽文档提供反馈	3
第 1 章 指标概述	4
第 2 章 指标解决方案组件	5
第 3 章 设置指标监控解决方案	6
3.1. 安装 PCP	6
3.2. 配置 PCP 数据收集	6
3.3. 验证 PCP 配置	7
3.4. 启用 WEB UI 访问指标	8
第 4 章 指标数据保留	9
4.1. 更改默认日志记录间隔	9
4.2. 更改数据保留策略	9
4.3. 查看数据存储统计信息	10
第 5 章 PCP 指标	11
5.1. 识别可用指标	11
5.2. 检索实时指标	11
5.3. 检索归档的指标	12
5.4. 在 WEB UI 中检索指标	13

对红帽文档提供反馈

我们感谢您对我们文档的反馈。让我们了解如何改进它。

使用 Red Hat JIRA 中的 **Create Issue** 表单提供您的反馈。JIRA 问题在 Red Hat Satellite Jira 项目中创建，您可以在其中跟踪其进度。

先决条件

- 确保您已注册了 [红帽帐户](#)。

流程

1. 单击以下链接：[创建问题](#)。如果 Jira 显示登录错误，则登录并在您重定向到表单后继续。
2. 完成 **Summary** 和 **Description** 字段。在 **Description** 字段中，包含文档 URL、章节号以及问题的详细描述。不要修改表单中的任何其他字段。
3. 点 **Create**。

第 1 章 指标概述

您可以设置第三方解决方案从 Satellite 服务器收集实时指标，将其归档为固定的时间，并分析它们。从 Satellite 获取指标对于容量规划和故障排除性能问题非常有用。

如果您需要通过红帽引发支持问题单来解决性能问题，则存档数据会提供宝贵洞察力。请注意，仅在将其上传到支持问题单时，红帽支持只能访问归档的数据。

您可以从 Satellite 收集以下指标：

- 操作系统的基本统计信息，包括系统负载、内存使用率和输入/输出操作
- 进程统计信息，包括内存和 CPU 使用率
- Apache HTTP 服务器活动统计信息
- PostgreSQL 活动统计信息
- Satellite 应用统计

第 2 章 指标解决方案组件

红帽建议使用 Performance Co-Pilot 来收集和归档 Satellite 指标。

Performance Co-Pilot (PCP)

Performance Co-Pilot 是一组用于获取、存储和分析系统级性能测量的工具和库。您可以使用 PCP 在 CLI 中分析实时和历史指标。

性能指标域代理(PMDA)

Performance Metric Domain Agent 是一个 PCP 附加组件，允许访问应用程序或服务的指标。要收集与 Satellite 相关的所有指标，您必须为 Apache HTTP 服务器和 PostgreSQL 安装 PMDA。

Grafana

视觉化 PCP 收集指标的 Web 应用程序。要在 web UI 中分析指标，您必须安装 Grafana 和 Grafana PCP 插件。

第 3 章 设置指标监控解决方案

安装 PCP 软件包并配置 PCP 数据收集。您可以使用 PCP CLI 工具在命令行中检索指标。另外，您可以安装 Grafana 来启用 Web UI 访问指标。

3.1. 安装 PCP

在受管主机上安装 PCP 软件包并启用 PCP 守护进程。

先决条件

- 确保 `/var/log/pcp` 目录中至少有 20 GB 空间。
使用默认的 PCP 数据保留设置，数据存储预计每天需要 100 MB 到 500 MB 磁盘空间，但随着时间的推移可能会消耗几 GB。如需更多信息，请参阅 [第 4 章 指标数据保留](#)。

流程

1. 安装 PCP 软件包：

```
# satellite-maintain packages install pcp \  
pcp-pmda-apache \  
pcp-pmda-openmetrics \  
pcp-pmda-postgresql \  
pcp-pmda-redis \  
pcp-system-tools \  
foreman-pcp
```

2. 启用并启动性能指标收集器守护进程和性能指标日志程序守护进程：

```
# systemctl enable --now pmcd pmlogger
```

3.2. 配置 PCP 数据收集

您可以配置 PCP 以收集有关进程、Satellite、Apache HTTP 服务器和 PostgreSQL 的指标。

流程

1. 将特定于 Satellite 的配置符号链接到 PMDA 进程监控：

```
# ln -s /etc/pcp/proc/foreman-hotproc.conf /var/lib/pcp/pmdas/proc/hotproc.conf
```

默认情况下，PCP 仅收集基本系统指标。此步骤启用以下 Satellite 进程的详细指标：

- Java
- PostgreSQL
- Redis
- dynflow
- puma

- Pulpcore

2. 安装监控 PMDA 的进程：

```
# cd /var/lib/pcp/pmdas/proc
# ./Install
```

3. 配置 PCP 以从 Apache HTTP 服务器收集指标。

- 启用 Apache HTTP 服务器扩展状态模块：

```
# satellite-installer --enable-apache-mod-status
```

- 启用 Apache HTTP 服务器 PMDA：

```
# cd /var/lib/pcp/pmdas/apache
# ./Install
```

4. 配置 PCP 以从 PostgreSQL 收集指标：

```
# cd /var/lib/pcp/pmdas/postgresql
# ./Install
```

5. 在 Satellite 中启用遥测功能：

```
# satellite-installer --foreman-telemetry-prometheus-enabled true
```

6. 配置 PCP 以从 Satellite 收集数据：

```
# cd /var/lib/pcp/pmdas/openmetrics
# echo "https://satellite.example.com/metrics" > config.d/foreman.url
# ./Install
```

7. 重启 PCP 以开始数据收集：

```
# systemctl restart pmcd pmlogger
```

3.3. 验证 PCP 配置

您可以验证 PCP 是否已正确配置，服务是否活跃。

流程

- 显示活跃 PCP 配置的摘要：

```
# pcp
```

pcp 命令的输出示例：

```
Performance Co-Pilot configuration on satellite.example.com:

platform: Linux satellite.example.com 4.18.0-372.32.1.el8_6.x86_64 #1 SMP Fri Oct 7
```

```
12:35:10 EDT 2022 x86_64
hardware: 16 cpus, 2 disks, 1 node, 31895MB RAM
timezone: UTC
services: pmcd pmproxy
  pmcd: Version 5.3.7-17, 13 agents, 4 clients
  pmda: root pmcd proc pmproxy xfs redis linux apache mmv kvm
        postgresql jbd2 openmetrics
pmlogger: primary logger: /var/log/pcp/pmlogger/satellite.example.com/20230831.00.10
pmie: primary engine: /var/log/pcp/pmie/satellite.example.com/pmie.log
```

在本例中，性能指标收集器守护进程(pmcd)和性能指标代理守护进程(pmproxy)服务都在运行。它还确认正在收集指标的 PMDA。最后，它列出了活跃的日志文件，其中 **pmlogger** 目前在其中存储指标。

3.4. 启用 WEB UI 访问指标

您可以通过安装 Grafana 来启用对 PCP 收集的指标的 Web UI 访问。

流程

1. 在 Satellite 服务器上安装 Grafana 和 Grafana PCP 插件：

```
# satellite-maintain packages install grafana grafana-pcp
```

2. 启动并启用 Grafana Web 服务和 PCP 代理服务：

```
# systemctl enable --now pmproxy grafana-server
```

3. 打开防火墙端口以允许访问 Grafana Web 界面：

```
# firewall-cmd --permanent --add-service=grafana
```

4. 重新载入防火墙配置，以应用更改：

```
# firewall-cmd --reload
```

5. 安装 PCP Redis 并配置 Grafana 以加载它。如需更多信息，请参阅在 *Red Hat Enterprise Linux 8 监控和管理系统状态和性能* 中配置 PCP Redis。
6. 访问 Grafana Web UI，启用 PCP 插件，并将 PCP Redis 添加为数据源。如需更多信息，请参阅在 *Red Hat Enterprise Linux 8 监控和管理系统状态和性能* 中访问 Grafana Web UI。

第 4 章 指标数据保留

PCP 数据日志记录所需的存储容量由以下因素决定：

- 日志记录的指标
- 日志记录间隔
- 保留策略

默认日志记录(sampling)间隔为 60 秒。默认保留策略是压缩超过一天的存档，并在最后 14 天内保留存档。

您可以增加日志记录间隔或缩短保留策略来节省存储空间。如果需要高分辨率抽样，您可以减少日志记录间隔。在这种情况下，请确保您有足够的存储空间。

PCP 归档日志存储在 `/var/log/pcp/pmlogger/satellite.example.com` 目录中。

4.1. 更改默认日志记录间隔

您可以更改默认的日志记录间隔，以增加或减少抽样率，在其中记录 PCP 指标。较大的间隔会降低抽样率。

流程

1. 打开 `/etc/pcp/pmlogger/control.d/local` 配置文件。
2. 找到 `LOCALHOSTNAME` 行。
3. 附加 `-t XXs`，其中 `XX` 是所需的时间间隔（以秒为单位）。
4. 保存该文件。
5. 重启 `pmlogger` 服务：

```
# systemctl restart pmlogger
```

4.2. 更改数据保留策略

您可以更改数据保留策略，以便在 PCP 数据存档和删除后控制。

流程

1. 打开 `/etc/sysconfig/pmlogger_timers` 文件。
2. 找到 `PMLOGGER_DAILY_PARAMS` 行。
3. 如果行被注释，请取消注释行。
4. 配置以下参数：
 - 确保存在默认的 `-E` 参数。
 - 附加 `-x` 参数，并为数据存档后所需的天数添加一个值。

- 附加 **-k** 参数，再为数据删除的天数添加一个值。

例如，参数 **-x 4 -k 7** 指定在 4 天后将压缩数据，并在 7 天后删除。

5. 保存该文件。

4.3. 查看数据存储统计信息

您可以列出所有可用的指标，按日志记录的频率分组。对于每个组，您还可以查看每天存储列出的指标所需的存储。

存储统计示例：

```
logged every 60 sec: 61752 bytes or 84.80 Mbytes/day
```

流程

- 要查看数据存储统计信息，请在 Satellite 服务器上输入以下命令：

```
# less /var/log/pcp/pmlogger/satellite.example.com/pmlogger.log
```

第 5 章 PCP 指标

指标存储在树形结构中。例如，所有网络指标都存储在名为 **network** 的节点中。每个指标可以是单个值，也可以是一个值列表，称为实例。例如，内核负载有三个实例，即 1 分钟、5 分钟和 15 分钟平均值。

对于每个指标条目，PCP 同时存储其数据和元数据。这包括指标描述、数据类型、单位和维度。例如，元数据可让 PCP 输出具有不同维度的多个指标。

如果指标是计数器，则其值只能增加。例如，特定设备上的磁盘写入操作计数仅增加。当您查询计数器指标的值时，PCP 会默认将其转换为速率值。

除了系统指标（如 CPU、内存、内核、XFS、磁盘和网络）外，还会配置以下指标：

指标	描述
hotproc.*	关键 Satellite 进程的基本指标
apache.*	Apache HTTP 服务器指标
postgresql.*	基本 PostgreSQL 统计
openmetrics.foreman.fm_rails_*	Satellite 指标

5.1. 识别可用指标

- 要列出 PCP 可用的所有指标，请输入以下命令：

```
# pminfo
```

- 要列出所有 Satellite 指标及其描述，请输入以下命令：

```
# foreman-rake telemetry:metrics
```

- 要列出归档的指标，请输入以下命令：

```
# less /var/log/pcp/pmlogger/satellite.example.com/pmlogger.log
```

- pmlogger** 守护进程存档数据（根据其配置）。要检索活跃日志文件的名称，请输入以下命令：

```
# pcp | grep logger
```

输出包括活跃日志文件的文件名，例如：

```
pmlogger: primary logger: /var/log/pcp/pmlogger/satellite.example.com/20230831.00.10
```

5.2. 检索实时指标

您可以使用 PCP CLI 工具来检索实时指标。

流程

- 要输出特定指标的当前值，请输入以下命令：

```
# pmval -f 1 disk.partitions.write
```

在本例中，会显示写入磁盘分区中的指标实例。PCP 将磁盘分区的写入数量从计数器值转换为速率值。**f 1** 参数指定将值缩写为一个十进制位置。

输出示例：

```
metric: disk.partitions.write
host: satellite.example.com
semantics: cumulative counter (converting to rate)
units: count (converting to count / sec)
samples: all

      vda1          vda2          sr0
      0.0           12.0           0.0
      0.0           1.0            0.0
      0.0           1.0            0.0
      0.0           2.0            0.0
```

- 要每 2 秒打印系统性能概述，请输入以下命令：

```
# pmstat -t 2sec
```

5.3. 检索归档的指标

您可以使用 PCP CLI 工具从存档文件中检索指标。

例子

- 要列出创建存档文件时启用的所有指标，请输入以下命令：

```
# pminfo --archive archive_file
```

- 要查看存档文件涵盖的主机和时间段，请输入以下命令：

```
# pmdumplog -l archive_file
```

- 要列出每个分区的磁盘写入，在存档文件覆盖的时间周期内：

```
# pmval --archive /var/log/pcp/pmlogger/satellite.example.com/20230831.00.10 \
-f 1 disk.partitions.write
```

- 要列出每个分区的磁盘写入操作，在 14:00 到 14:15 之间有一个 2 秒的间隔：

```
# pmval --archive /var/log/pcp/pmlogger/satellite.example.com/20230831.00.10 \
-d -t 2sec \
-f 3 disk.partitions.write \
-S @14:00 -T @14:15
```


- 要列出所有性能指标的平均值，包括 minimum/maximum 的时间和值，在 14:00 到 14:30 之间，将值格式化为表：

```
# pmlogsummary /var/log/pcp/pmlogger/satellite.example.com/20230831.00.10\  
-HfilmM\  
-S @14:00\  
-T @14:30\  
disk.partitions.write\  
mem.freemem
```

- 以互动方式显示存储在存档中的系统指标（从 14:00 开始），类似于 **top** 工具：

```
# pcp --archive /var/log/pcp/pmlogger/satellite.example.com/20230831.00.10\  
-S @14:00\  
atop
```

5.4. 在 WEB UI 中检索指标

您可以查看 Grafana 仪表盘，其中显示带有指标值的小部件。您可以添加和删除指标以满足您的要求。您还可以选择为每个小部件显示的时间范围。

流程

- 在浏览器中打开以下 URL：<http://satellite.example.com:3000>

其他资源

- 有关使用 Grafana 的详情，请查看 [Grafana Labs](#) 网站。