



## Red Hat Satellite 6.15

### Satellite のパフォーマンスの監視

Satellite からメトリクスを収集し、外部ツールで分析できるようにする



## Red Hat Satellite 6.15 Satellite のパフォーマンスの監視

---

Satellite からメトリクスを収集し、外部ツールで分析できるようにする

Red Hat Satellite Documentation Team

[satellite-doc-list@redhat.com](mailto:satellite-doc-list@redhat.com)

## 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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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

このガイドでは、分析のために Red Hat Satellite 6 からメトリクスを収集する方法を説明します。この内容は、Satellite の管理者向けとなります。

---

## 目次

RED HAT ドキュメントへのフィードバック (英語のみ) .....	3
第1章 メトリクスの概要 .....	4
第2章 メトリクスソリューションのコンポーネント .....	5
第3章 メトリクス監視ソリューションの設定 .....	6
3.1. PCP のインストール .....	6
3.2. PCP データ収集の設定 .....	6
3.3. PCP 設定の確認 .....	7
3.4. メトリクスへの WEB UI アクセスの有効化 .....	8
第4章 メトリクスデータの保持 .....	9
4.1. デフォルトのログ間隔の変更 .....	9
4.2. データ保持ポリシーの変更 .....	9
4.3. データストレージ統計情報の表示 .....	10
第5章 PCP メトリクス .....	11
5.1. 利用可能なメトリクスの特定 .....	11
5.2. ライブメトリクスの取得 .....	12
5.3. アーカイブされたメトリクスの取得 .....	12
5.4. WEB UI でのメトリクスの取得 .....	13



## 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。また、改善点があればお知らせください。

フィードバックを提供するには、Red Hat Jira の **Create Issue** フォームを使用します。Jira の問題は Red Hat Satellite Jira プロジェクトに作成され、その進捗状況を追跡できます。

### 前提条件

- [Red Hat アカウント](#) が登録されている。

### 手順

1. **Create Issue** にアクセスします。Jira でログインエラーが表示された場合は、フォームにリダイレクトされた後、ログインして続行します。
2. **Summary** フィールドと **Description** フィールドに入力します。**Description** フィールドに、ドキュメントの URL、章またはセクション番号、および問題の詳しい説明を入力します。フォーム内の他のフィールドは変更しないでください。
3. **Create** をクリックします。

## 第1章 メトリクスの概要

サードパーティーのソリューションを設定して、Satellite Server からライブメトリクスを収集し、一定期間アーカイブして分析することができます。Satellite からメトリクスを取得すると、キャパシティプランニングやパフォーマンスの問題のトラブルシューティングに役立ちます。

パフォーマンスの問題を解決するために Red Hat にサポートケースを起票する必要がある場合は、アーカイブデータが貴重な情報源となります。Red Hat サポートチームがアーカイブデータにアクセスするには、サポートケースにデータをアップロードしていただく必要があることに注意してください。

Satellite からは次のメトリクスを収集できます。

- システム負荷、メモリー使用率、入出力操作など、オペレーティングシステムからの基本的な統計情報
- メモリーや CPU 使用率などのプロセス統計情報
- Apache HTTP Server のアクティビティ統計情報
- PostgreSQL のアクティビティ統計情報
- Satellite アプリケーションの統計情報



## 第2章 メトリクスソリューションのコンポーネント

Red Hat では、Satellite メトリクスの収集とアーカイブに Performance Co-Pilot を使用することを推奨しています。

### Performance Co-Pilot (PCP)

Performance Co-Pilot は、システムレベルのパフォーマンス測定値を取得、保存、分析するためのツールとライブラリーのスイートです。PCP を使用すると、CLI でライブメトリクスと履歴メトリクスを分析できます。

### Performance Metric Domain Agent (PMDA)

Performance Metric Domain Agent は、アプリケーションまたはサービスのメトリクスへのアクセスを可能にする PCP のアドオンです。Satellite に関連するすべてのメトリクスを収集するには、Apache HTTP Server および PostgreSQL 用の PMDA をインストールする必要があります。

### Grafana

PCP によって収集されたメトリクスを視覚化する Web アプリケーション。Web UI でメトリクスを分析するには、Grafana と Grafana PCP プラグインをインストールする必要があります。

## 第3章 メトリクス監視ソリューションの設定

PCP パッケージをインストールし、PCP データ収集を設定します。PCP の CLI ツールを使用すると、コマンドラインでメトリクスを取得できます。必要に応じて、Grafana をインストールして、Web UI からメトリクスにアクセスできます。

### 3.1. PCP のインストール

Satellite Server に PCP パッケージをインストールし、PCP デーモンを有効にします。

#### 前提条件

- `/var/log/pcp` ディレクトリーに少なくとも 20 GB の空き領域がある。  
デフォルトの PCP データ保持設定では、1日あたり 100 MB - 500 MB のディスク領域がデータストレージに使用されると推定されます。ただし、時間が経過すると、最大で数ギガバイト使用される可能性があります。詳細は、[4章メトリクスデータの保持](#)を参照してください。

#### 手順

1. PCP パッケージを以下のコマンドでインストールします。

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

2. Performance Metrics Collector デーモンと Performance Metrics Logger デーモンを有効にして起動します。

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

### 3.2. PCP データ収集の設定

PCP を設定して、プロセス、Satellite、Apache HTTP Server、PostgreSQL に関するメトリクスを収集できます。

#### 手順

1. Satellite 固有の設定を PMDA プロセスモニタリングにシンボリックリンクします。

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

デフォルトでは、PCP は基本的なシステムメトリクスのみを収集します。このステップを実行すると、次の Satellite プロセスに関する詳細なメトリクスが有効になります。

- Java
- PostgreSQL
- Redis

- Dynflow
- Puma
- Pulpcore

2. プロセスモニタリング PMDA をインストールします。

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

3. Apache HTTP Server からメトリクスを収集するように PCP を設定します。

- a. Apache HTTP サーバーの拡張ステータスモジュールを有効にします。

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

- b. Apache HTTP サーバー PMDA を有効にします。

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

4. PostgreSQL からメトリクスを収集するように PCP を設定します。

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

5. Satellite でテレメトリー機能を有効にします。

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

6. Satellite からデータを収集するように PCP を設定します。

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

7. PCP を再起動してデータ収集を開始します。

```
# systemctl restart pmcd pmlogger
```

### 3.3. PCP 設定の確認

PCP が正しく設定され、サービスがアクティブであることを確認できます。

#### 手順

- アクティブな PCP 設定の概要を出力します。

```
# pcp
```

**pcp** コマンドの出力例:

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
```

この例では、Performance Metrics Collector Daemon (pmcd) サービスと Performance Metrics Proxy Daemon (pmproxy) サービスの両方を実行します。また、メトリクスを収集している PMDA を確認します。最後に、**pmlogger** が現在メトリクスを保存しているアクティブなログ ファイルをリスト表示します。

### 3.4. メトリクスへの WEB UI アクセスの有効化

Grafana をインストールすることで、PCP によって収集されたメトリクスへの Web UI アクセスを有効にすることができます。

#### 手順

1. Satellite Server に Grafana と Grafana PCP プラグインをインストールします。

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

2. Grafana Web サービスと PCP プロキシサービスを起動して有効にします。

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

3. ファイアウォールポートを開いて、Grafana Web インターフェイスへのアクセスを許可します。

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

4. ファイアウォール設定をリロードして、変更を適用します。

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

5. PCP Redis をインストールし、それをロードするように Grafana を設定します。詳細は、Red Hat Enterprise Linux 8 の [システムの状態とパフォーマンスの監視と管理](#) の [PCP Redis の設定](#) を参照してください。
6. Grafana Web UI にアクセスし、PCP プラグインを有効にして、PCP Redis をデータソースとして追加します。詳細は、Red Hat Enterprise Linux 8 の [システムの状態とパフォーマンスの監視と管理](#) の [Grafana Web UI へのアクセス](#) を参照してください。

## 第4章 メトリクスデータの保持

PCP データログに必要なストレージ容量は、以下の要素で決定されます。

- ログに記録されるメトリクス
- ログ間隔
- 保持ポリシー

デフォルトのログ (サンプリング) 間隔は 60 秒です。デフォルトの保持ポリシーでは、1 日以上前のアーカイブを圧縮し、過去 14 日間のアーカイブを保持します。

ストレージ領域を節約するために、ログ間隔を長くしたり、保持ポリシーを短くしたりすることができます。高精度のサンプリングが必要な場合は、ログ間隔を短くすることができます。その場合は、十分なストレージ領域があることを確認してください。

PCP アーカイブログは、`/var/log/pcp/pmlogger/satellite.example.com` ディレクトリーに保存されます。

### 4.1. デフォルトのログ間隔の変更

デフォルトのログ間隔を変更して、PCP メトリクスを記録するサンプリングレートを増減することができます。間隔を長くすると、サンプリングレートが低くなります。

#### 手順

1. `/etc/pcp/pmlogger/control.d/local` 設定ファイルを開きます。
2. **LOCALHOSTNAME** 行を見つけます。
3. **-t XX s** を追加します。XX は必要な時間間隔 (秒単位) です。
4. ファイルを保存します。
5. **pmlogger** サービスを再起動します。

```
# systemctl restart pmlogger
```

### 4.2. データ保持ポリシーの変更

データ保持ポリシーを変更して、PCP データがアーカイブおよび削除されるまでの期間を制御できます。

#### 手順

1. `/etc/sysconfig/pmlogger_timers` ファイルを開きます。
2. **PMLOGGER\_DAILY\_PARAMS** 行を見つけます。
3. 行がコメント化されている場合は、その行をコメント解除します。
4. 次のパラメーターを設定します。
  - デフォルトの **-E** パラメーターが存在することを確認します。

- **-x** パラメーターを追加し、データがアーカイブされるまでの必要な日数の値を追加します。
- **-k** パラメーターを追加し、データが削除されるまでの日数の値を追加します。

たとえば、**-x 4 -k 7** というパラメーターの場合は、データが 4 日後に圧縮され、7 日後に削除されます。

5. ファイルを保存します。

### 4.3. データストレージ統計情報の表示

利用可能なすべてのメトリクスを、ログに記録される頻度ごとにグループ化してリスト表示できます。各グループについて、リスト表示されたメトリクスを保存するのに必要なストレージを 1 日単位で表示することもできます。

ストレージ統計情報の例

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

#### 手順

- データストレージの統計情報を表示するには、Satellite Server で次のコマンドを入力します。

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

## 第5章 PCP メトリクス

メトリクスはツリーのような構造で保存されます。たとえば、すべてのネットワークメトリクスは、**network** という名前のノードに保存されます。各メトリクスは、1つの値か、インスタンスと呼ばれる値のリストです。たとえば、カーネルの負荷には平均1分、5分、15分の3つのインスタンスがあります。

PCP は、すべてのメトリクスエントリーについて、データとメタデータの両方を保存します。メタデータには、メトリクスの説明、データタイプ、単位、ディメンションが含まれます。メタデータにより、PCP は、たとえばディメンションが異なる複数のメトリクスを出力することが可能になります。

メトリクスがカウンターの場合、その値は増加するだけです。たとえば、特定のデバイス上のディスク書き込み操作回数は、増加するだけです。カウンターメトリクスの値をクエリーすると、PCP はデフォルトでこれをレート値に変換します。

CPU、メモリー、カーネル、XFS、ディスク、ネットワークなどのシステムメトリクスに加えて、次のメトリクスが設定されます。

メトリクス	説明
hotproc.*	主要 Satellite プロセスの基本的なメトリクス
apache.*	Apache HTTP サーバーのメトリクス
postgresql.*	PostgreSQL の基本的なメトリクス
openmetrics.foreman.fm_rails_*	Satellite のメトリクス

### 5.1. 利用可能なメトリクスの特定

- PCP を通じて利用可能なすべてのメトリクスをリスト表示するには、次のコマンドを入力します。

```
# pminfo
```

- すべての Satellite メトリクスとその説明をリスト表示するには、次のコマンドを入力します。

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

- アーカイブされたメトリクスをリスト表示するには、次のコマンドを入力します。

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

- **pmlogger** デーモンは、設定に従って、受信したデータをアーカイブします。アクティブなログファイルの名前を取得するには、次のコマンドを入力します。

```
# pcp | grep logger
```

出力に、アクティブなログファイルのファイル名が含まれています。以下に例を示します。

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

-

## 5.2. ライブメトリクスの取得

PCP CLI ツールを使用してライブメトリクスを取得できます。

### 手順

- 特定のメトリクスの現在の値を出力するには、次のコマンドを入力します。

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

この例では、ディスクパーティションへの書き込みに関するメトリクスインスタンスが表示されます。PCP は、ディスクパーティションへの書き込み回数をカウンター値からレート値に変換します。**-f 1** 引数は、値を小数点第1位まで表示することを指定します。

出力例:

```
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
```

- システムパフォーマンスの概要を2秒ごとに出力するには、次のコマンドを入力します。

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

## 5.3. アーカイブされたメトリクスの取得

PCP CLI ツールを使用して、アーカイブファイルからメトリクスを取得できます。

### 例

- アーカイブファイルの作成時に有効になっていたすべてのメトリクスをリスト表示するには、次のコマンドを入力します。

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

- アーカイブファイルに含まれるホストと期間を表示するには、次のコマンドを入力します。

```
# pmdumplog -l archive_file
```

- アーカイブファイルがカバーしている期間の各パーティションのディスク書き込みをリスト表示するには、以下のコマンドを実行します。



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

- 14:00 から 14:15 までの期間にわたる、2 秒間隔のパーティションごとのディスク書き込み操作をリスト表示するには、次のコマンドを実行します。

```
# 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
```

- 14:00 から 14:30 までの期間にわたる、最小値/最大値の時間と値を含むすべてのパフォーマンスメトリクスの平均値をリスト表示し、値を表形式にフォーマットするには、次のコマンドを実行します。

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

- アーカイブに保存されている 14:00 以降のシステムメトリクスを、**top** ツールと同様にインタラクティブな方法で表示するには、次のコマンドを実行します。

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

## 5.4. WEB UI でのメトリクスの取得

Grafana ダッシュボードを表示すると、メトリクスの値を含むウィジェットを表示できます。メトリクスは要件に合わせて追加および削除できます。各ウィジェットに表示される時間範囲を選択することもできます。

### 手順

- ブラウザーで URL `http://satellite.example.com:3000` を開きます。

### 関連情報

- Grafana の使用方法に関する詳細は、[Grafana Labs Web サイト](#)を参照してください。