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

Last Updated: 2024-04-24

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

法律上の通知

概要

本ガイドでは、Red Hat Satellite 6 から分析に使用するメトリックを収集する方法について説明します。この内容は、Satellite の管理者向けとなります。

目次

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。この取り組みは膨大な作業を要するため、これらの変更による更新は可能な範囲で段階的に行われます。詳細は、Red Hat CTO である Chris Wright のメッセージ をご覧ください。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに対するご意見をお聞かせください。ドキュメントの改善点があればお知らせください。

Bugzilla でチケットを作成することでフィードバックを送信できます。

1. Bugzilla のWeb サイトに移動します。
2. **Component** フィールドで、**Documentation** を使用します。
3. **Description** フィールドに、ドキュメントの改善に関するご意見を記入してください。ドキュメントの該当部分へのリンクも追加してください。
4. **Submit Bug** をクリックします。

第1章 メトリックの概要

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、容量計画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。

Red Hat にサポートケースを起票する必要がある場合には、アーカイブしたデータを確認することで、価値のある見識が得られます。Red Hat サポートは、サポートケースにアップロードした場合にのみアーカイブされたデータにアクセスできることに注意してください。

Satellite から以下のメトリックを収集できます。

- 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統計
- メモリーや CPU 使用率などのプロセス統計
- Apache HTTP Server アクティビティー統計
- PostgreSQL アクティビティー統計
- Satellite アプリケーションの統計

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 の使用を推奨しています。

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。

パフォーマンスメトリクスドメインエージェント (PMDA)

パフォーマンスメトリクスドメインエージェントは、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 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 CLI ツールを使用して、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るようにすることができます。

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 の空き容量があることを確認してください。デフォルトの PCP データ保持設定では、データストレージは 1 日あたり 100 MB - 500 MB のディスク領域を使用すると推定されますが、時間の経過とともに最大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. パフォーマンスメトリクスコレクターデーモンとパフォーマンスメトリクスロガー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 プロセス監視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、以下の Satellite プロセスの詳細なメトリックが収集されるようになります。

- Java
- PostgreSQL
- Redis

- Dynflow
- プーマ
- パルプコア

2. プロセス監視 PMDA をインストールします。

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

3. PCP が Apache HTTP サーバーからメトリックを収集するように設定します。
   a. Apache HTTP Server 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 Server 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. データ収集を開始するには PCP を再起動します。

```
# 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、パフォーマンスメトリクスコレクターデーモン (pmcd) サービスとパフォーマンスメトリクスプロキシーデーモン (pmproxy) サービスの両方が実行されています。また、メトリクスを収集している PMDA も確認します。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 アクセスを有効にする

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 プロキシー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 での PCP Redis の設定、**システムの状態とパフォーマンスの監視および管理** 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 での Grafana Web UI へのアクセスおよび**システムステータスとパフォーマンスの監視と管理** 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。

- 記録されたメトリクス
- 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 日以上経過した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。

ストレージスペースを節約するために、ログ記録間隔を長くしたり、保持ポリシーを短くしたりすることができます。高解像度のサンプリングが必要な場合は、ログ記録間隔を短くすることができます。このような場合は、十分なストレージスペース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. デフォルトのロギング間隔の変更

デフォルトのログ記録間隔を変更して、PCP メトリクスが記録されるサンプリングレートを増減することができます。間隔が広くなると、サンプリングレートは低くなります。

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。ここで、**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、その行のコメントを解除します。
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、パラメーター **-x 4 -k 7 は**、データが 4 日後に圧縮され、7 日後に削除されることを指定します。

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、リストされたメトリクスを保存するために必要なストレージを 1 日あたりに表示することもできます。

ストレージ統計の例:

```
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

メトリックは、ツリー構造で保存されます。たとえば、ネットワークメトリックはすべて、**network** というノードに保存され、各メトリックはインスタンスと呼ばれ、単一または複数の値を使用できます。各メトリックは、値 を1つまたはリストで指定できます (インスタンスと呼ばれる)。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 は各メトリック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。これにはメトリックの説明、データタイプ、単位、ディメンションなどが含まれます。たとえば、メタデータを使うと、PCP は異なるディメンションで複数のメトリックを出力できます。

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、特定デバイス上でのディスクの書き込み操作のカウントは、増加のみです。カウンターメトリックの値をクエリーすると、PCP はこれをデフォルトで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ック
apache.*	Apache HTTP サーバーのメトリック
postgresql.*	PostgreSQL の基本的なメトリック
openmetrics.foreman.fm_rails_*	Satellite のメトリック

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 メトリックと説明をリスト表示するには、以下のコマンドを実行します。

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

- アーカイブ化されたメトリックをリスト表示するには、以下のコマンドを実行します。

```
# 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
```

出力には、アクティブなログファイルのファイル名が含まれます。例:

```
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印刷するには、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 位に省略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 秒ごとに印刷するには、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 ツールを使ってアーカイブファイルからメトリックを取得できます。

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、2 秒間隔で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、値をテーブルとしてフォーマットするには、次のようにします。

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

- アーカイブに保存されているシステムメトリクスを、14:00 から、**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、次のようにします。

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

5.4. WEB UI でメトリクスを取得する

メトリクスの値を含むウィジェットを表示する Grafana ダッシュボードを表示できます。要件に合わせてメトリックを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

- ブラウザーで次の URL を開きます: http://**satellite.example.com**:3000

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。