



OpenShift Container Platform 4.16

パワーモニタリング

OpenShift Container Platform の電力モニタリングの設定と使用

OpenShift Container Platform 4.16 パワーモニタリング

OpenShift Container Platform の電力モニタリングの設定と使用

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[®] 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.

概要

パワーモニタリングを使用すると、OpenShift Container Platform クラスターで実行されている各コンテナの CPU や DRAM などのさまざまなコンポーネントの電力消費をモニタリングします。

目次

第1章 RED HAT OPENSIFT 用パワーモニタリングのリリースノート	3
1.1. パワーモニタリング 0.2 (テクノロジープレビュー)	3
1.2. パワーモニタリング 0.1 (テクノロジープレビュー)	4
第2章 パワーモニタリングの概要	5
2.1. パワーモニタリングについて	5
2.2. パワーモニタリングのアーキテクチャー	5
2.3. KEPLER のハードウェアと仮想化のサポート	5
2.4. 関連情報	6
第3章 RED HAT OPENSIFT 用パワーモニタリングのインストール	7
3.1. POWER MONITORING OPERATOR のインストール	7
3.2. KEPLER のデプロイ	8
第4章 パワーモニタリングの設定	9
4.1. KEPLER の設定	9
4.2. KEPLER のステータスの監視	10
4.3. REDFISH を使用するように KEPLER を設定	11
第5章 パワーモニタリングのメトリクスの視覚化	14
5.1. パワーモニタリングダッシュボードの概要	14
5.2. クラスター管理者として電力監視ダッシュボードにアクセスする	15
5.3. 開発者として電力監視ダッシュボードにアクセスする	16
5.4. パワーモニタリングのメトリクスの概要	16
5.5. 関連情報	19
第6章 パワーモニタリングのアンインストール	20
6.1. KEPLER の削除	20
6.2. POWER MONITORING OPERATOR のアンインストール	20

第1章 RED HAT OPENSIFT 用パワーモニタリングのリリースノート



重要

パワーモニタリングはテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。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。

Red Hat OpenShift 用パワーモニタリングを使用すると、ワークロードの電力使用量を監視し、コンテナレベルで測定された CPU や DRAM などの主要な電力消費メトリクスを使用して、OpenShift Container Platform クラスター内で実行されている消費電力が最も多い namespace を特定できます。

このリリースノートでは、OpenShift Container Platform の Red Hat OpenShift 用パワーモニタリングの開発状況を追跡します。

Power monitoring Operator の概要は、[パワーモニタリングについて](#) を参照してください。

1.1. パワーモニタリング 0.2 (テクノロジープレビュー)

このリリースには、次のバージョンの更新が含まれています。

- Kepler 0.7.10
- Power monitoring Operator 0.13.0

パワーモニタリング 0.2 については、次のアドバイザリーが利用可能です。

- [RHEA-2024:2923](#)

1.1.1. 機能

- この更新により、Kepler で Redfish API を有効にできるようになりました。Kepler は Redfish を使用してノードによって消費される電力を取得します。
- この更新により、任意の namespace に電力監視 Operator をインストールできるようになりました。
- この更新により、**Developer** パースペクティブからパワーモニタリングの **Overview** ダッシュボードにアクセスできるようになりました。

1.1.2. CVE

- [CVE-2022-48554](#)
- [CVE-2023-2975](#)
- [CVE-2023-3446](#)

- [CVE-2023-3817](#)
- [CVE-2023-5678](#)
- [CVE-2023-6129](#)
- [CVE-2023-6237](#)
- [CVE-2023-7008](#)
- [CVE-2024-0727](#)
- [CVE-2024-25062](#)
- [CVE-2024-28834](#)
- [CVE-2024-28835](#)

1.2. パワーモニタリング 0.1(テクノロジープレビュー)

このリリースでは、Red Hat OpenShift 用パワーモニタリングのテクノロジープレビューバージョンが導入されています。パワーモニタリング 0.1については、次のアドバイザリーが利用可能です。

- [RHEA-2024:0078](#)

1.2.1. 機能

- Kepler のデプロイと削除
- Intel ベースのベアメタルデプロイメントからの電力使用量メトリクス
- 電力使用量を視覚化するためのダッシュボード

第2章 パワーモニタリングの概要



重要

パワーモニタリングはテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。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。

2.1. パワーモニタリングについて

Red Hat OpenShift 用パワーモニタリングを使用すると、電力使用量を監視し、OpenShift Container Platform クラスター内で実行されている消費電力が多いコンテナを特定できます。パワーモニタリングは、CPU や DRAM などのさまざまなコンポーネントからエネルギー関連のシステム統計情報を収集し、エクスポートします。また、Kubernetes Pod、namespace、ノードの詳細な電力消費データを提供します。



警告

テクノロジープレビュー機能であるパワーモニタリングは、ベアメタルデプロイメントでのみ機能します。ほとんどのパブリッククラウドベンダーは、カーネル電源管理サブシステムを仮想マシンに公開していません。

2.2. パワーモニタリングのアーキテクチャー

パワーモニタリングは、次の主要コンポーネントで構成されます。

Power monitoring Operator

Power monitoring Operator は、管理者のために、OpenShift Container Platform クラスターの Kepler のデプロイと管理を簡素化し、ワークロードの電力使用量の監視を効率化します。Power monitoring Operator のセットアップと設定は、Kepler カスタムリソース定義 (CRD) を追加することで簡素化されます。Operator は、Kepler のアップグレード、削除、設定、再デプロイなどの操作も管理します。

Kepler

Kepler はパワーモニタリングの重要なコンポーネントです。OpenShift Container Platform で実行されているコンテナの電力使用量を監視する役割を果たします。ノードとコンテナの両方の電力使用量に関連するメトリクスを生成します。

2.3. KEPLER のハードウェアと仮想化のサポート

Kepler は、パワーモニタリングの重要なコンポーネントであり、次のいずれかの方法でノードからリアルタイムの電力消費データを収集します。

カーネル電源管理サブシステム (推奨)

- **rapl-sysfs**: これには、`/sys/class/powercap/intel-rapl` ホストファイルへのアクセスが必要です。
- **rapl-msr**: これには、`/dev/cpu/*/msr` ホストファイルへのアクセスが必要です。

estimator 電源

Kepler は、カーネルの電力制限サブシステムにアクセスせずに、機械学習モデルを使用してノード上の CPU の電力使用量を推定します。



警告

estimator 機能は実験的なものであり、サポート対象外です。この機能に依拠することがないようにご注意ください。

Power Monitoring / Overview ダッシュボードを使用すると、ノードの電力推定方法を確認できます。

2.4. 関連情報

- [パワーモニタリングダッシュボードの概要](#)

第3章 RED HAT OPENSIFT 用パワーモニタリングのインストール



重要

パワーモニタリングはテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。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。

OpenShift Container Platform Web コンソールに Power monitoring Operator をデプロイすることで、Red Hat OpenShift 用パワーモニタリングをインストールできます。

3.1. POWER MONITORING OPERATOR のインストール

クラスター管理者は、OpenShift Container Platform Web コンソールを使用して、OperatorHub から Power monitoring Operator をインストールできます。



警告

インストール前に、以前にインストールしたバージョンの Power monitoring Operator を削除する必要があります。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- **cluster-admin** ロールを持つユーザーとしてログインしている。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Operators** → **OperatorHub** に移動します。
2. **power monitoring** を検索し、**Power monitoring for Red Hat OpenShift** タイルをクリックして、**Install** をクリックします。
3. もう一度 **Install** をクリックして、Power monitoring Operator をインストールします。
Red Hat OpenShift 用パワーモニタリングが、OpenShift Container Platform クラスターのすべての namespace で利用できるようになります。

検証

1. Power monitoring Operator が **Operators** → **Installed Operators** にリストされていることを確認します。 **Status** は **Succeeded** に解決されるはずですが。

3.2. KEPLER のデプロイ

Kepler をデプロイするには、Power monitoring Operator を使用して **Kepler** カスタムリソース定義 (CRD) のインスタンスを作成します。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- **cluster-admin** ロールを持つユーザーとしてログインしている。
- Power monitoring Operator がインストールされている。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Operators** → **Installed Operators** に移動します。
2. **Installed Operators** リストから **Power monitoring for Red Hat OpenShift** をクリックし、**Kepler** タブに移動します。
3. **Create Kepler** をクリックします。
4. **Create Kepler** ページで、**Name** が **kepler** に設定されていることを確認します。



重要

Kepler インスタンスの名前は、**kepler** に設定する必要があります。他のインスタンスは、すべて Power monitoring Operator によって無視されます。

5. **Create** をクリックして、Kepler およびパワーモニタリングダッシュボードをデプロイします。

第4章 パワーモニタリングの設定



重要

パワーモニタリングはテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。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。

Kepler リソースは、Kubernetes カスタムリソース定義 (CRD) です。これを使用してデプロイメントを設定し、Kepler リソースのステータスを監視できます。

4.1. KEPLER の設定

Kepler リソースの **spec** フィールドを使用して Kepler を設定できます。



重要

Kepler インスタンスの名前は、必ず **kepler** に設定してください。その他のすべてのインスタンスは、電力監視 Operator Webhook によって拒否されます。

以下は設定オプションのリストです。

表4.1 Kepler の設定オプション

名前	Spec	説明	デフォルト
port	exporter.deployment	Prometheus メトリクスが公開されるノード上のポート。	9103
nodeSelector	exporter.deployment	Kepler エクスポーター Pod がスケジュールされるノード。	kubernetes.io/os: linux
tolerations	exporter.deployment	Kepler エクスポーターの容認は、特定の特性を持つノード上で Pod をスケジュールすることを許可するものです。	- operator: "Exists"

デフォルト設定の Kepler リソースの例

```
apiVersion: kepler.system.sustainable.computing.io/v1alpha1
kind: Kepler
metadata:
```

```

name: kepler
spec:
  exporter:
    deployment:
      port: 9103 ❶
      nodeSelector:
        kubernetes.io/os: linux ❷
      Tolerations: ❸
      - key: ""
        operator: "Exists"
        value: ""
        effect: ""

```

- ❶ Prometheus メトリクスはポート 9103 で公開されます。
- ❷ Kepler Pod は Linux ノード上でスケジュールされます。
- ❸ デフォルトの容認は、Kepler を任意のノードでスケジュールすることを許可します。

4.2. KEPLER のステータスの監視

Kepler リソースの **status** フィールドを使用すると、Kepler エクスポートの状態を監視できます。

status.exporter フィールドには、次のような情報が含まれています。

- 現在 Kepler Pod を実行しているノードの数
- Kepler Pod を実行する必要があるノードの数
- Kepler リソースの健全性を表す条件

これにより、**spec** フィールドを通じて行った変更についての貴重な情報が得られます。

Kepler リソースの状態の例

```

apiVersion: kepler.system.sustainable.computing.io/v1alpha1
kind: Kepler
metadata:
  name: kepler
status:
  exporter:
    conditions: ❶
    - lastTransitionTime: '2024-01-11T11:07:39Z'
      message: Reconcile succeeded
      observedGeneration: 1
      reason: ReconcileSuccess
      status: 'True'
      type: Reconciled
    - lastTransitionTime: '2024-01-11T11:07:39Z'
      message: >-
        Kepler daemonset "kepler-operator/kepler" is deployed to all nodes and
        available; ready 2/2
      observedGeneration: 1
      reason: DaemonSetReady

```

```

status: 'True'
type: Available
currentNumberScheduled: 2 2
desiredNumberScheduled: 2 3

```

- 1** Kepler リソースの健全性。この例では、Kepler は正常に調整され、準備が完了しています。
- 2** 現在 Kepler Pod を実行しているノードの数は2です。
- 3** Kepler Pod を実行する必要があるノードの数は2です。

4.3. REDFISH を使用するように KEPLER を設定

コンテナを実行またはホストするためのソースとして Redfish を使用するように Kepler を設定できます。Kepler はこれらのコンテナの電力使用量を監視できます。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- **cluster-admin** ロールを持つユーザーとしてログインしている。
- Power monitoring Operator がインストールされている。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Operators** → **Installed Operators** の順にクリックします。
2. **Installed Operators** リストから **Power monitoring for Red Hat OpenShift** をクリックし、**Kepler** タブをクリックします。
3. **Create Kepler** をクリックします。Kepler インスタンスがすでに作成されている場合は、**Kepler** の **編集** をクリックします。
4. 必須の **secretRef** フィールドを指定して、Kepler インスタンスの **.spec.exporter.redfish** を設定します。ニーズに合わせてオプションの **probeInterval** フィールドと **skipSSLVerify** フィールドを設定することもできます。

Kepler インスタンスの例

```

apiVersion: kepler.system.sustainable.computing.io/v1alpha1
kind: Kepler
metadata:
  name: kepler
spec:
  exporter:
    deployment:
# ...
  redfish:
    secretRef: <secret_name> required 1

```

```

probeInterval: 60s 2
skipSSLVerify: false 3
# ...

```

- 1 必須: Redfish サーバーにアクセスするための認証情報を含むシークレットの名前を指定します。
- 2 オプション: Redfish から電力情報を照会する頻度を制御します。デフォルト値は **60s** です。
- 3 オプション: Kepler が Redfish サーバー証明書の検証をスキップするかどうかを制御します。デフォルト値は **false** です。



注記

Kepler がデプロイされると、**openshift-power-monitoring** namespace 名前空間が作成されます。

5. 次のデータ形式で **redfish.csv** ファイルを作成します。

```

<your_kubelet_node_name>,<redfish_username>,
<redfish_password>,https://<redfish_ip_or_hostname>/

```

redfish.csv ファイルの例

```

control-plane,exampleuser,examplepass,https://redfish.nodes.example.com
worker-1,exampleuser,examplepass,https://redfish.nodes.example.com
worker-2,exampleuser,examplepass,https://another.redfish.nodes.example.com

```

6. **openshift-power-monitoring** namespace の下にシークレットを作成します。次の条件でシークレットを作成する必要があります。
 - シークレットタイプは **Opaque** です。
 - 認証情報は、シークレットの **data** フィールドの **redfish.csv** キーの下に保存されます。

```

$ oc -n openshift-power-monitoring \
  create secret generic redfish-secret \
  --from-file=redfish.csv

```

出力例

```

apiVersion: v1
kind: Secret
metadata:
  name: redfish-secret
data:
  redfish.csv: YmFyCg==
# ...

```




重要

Kepler のデプロイメントは、Redfish シークレットが作成されるまで続行されません。この情報は、Kepler インスタンスの **status** で確認できます。

第5章 パワーモニタリングのメトリクスの視覚化



重要

パワーモニタリングはテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。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。

OpenShift Container Platform Web コンソールでパワーモニタリングのメトリクスを視覚的に確認するには、パワーモニタリングダッシュボードにアクセスするか、**Observe** タブの **Metrics** を確認します。

5.1. パワーモニタリングダッシュボードの概要

パワーモニタリングダッシュボードは 2 種類あります。どちらも、1 つのクラスターの消費電力メトリクスに関するさまざまなレベルの詳細情報を提供します。

Power Monitoring / Overview ダッシュボード

このダッシュボードを使用すると、次の情報を確認できます。

- CPU アーキテクチャーおよびその電源 (**rapl-sysfs**、**rapl-msr**、または **estimator**) と、この構成の総ノード数の集約ビュー
- 過去 24 時間のクラスターの総エネルギー消費量 (キロワット時単位で測定)
- 過去 24 時間にクラスター内の上位 10 の namespace によって消費された電力量
- CPU アーキテクチャーやコンポーネントの電源などのノード詳細情報

これらの機能を使用すると、各 namespace を個別に調査することなく、クラスターのエネルギー消費を効果的に監視できます。



警告

Components Source 列に **estimator** が電源として表示されていないことを確認してください。

図5.1 ノード詳細情報の表 (rapl-sysfs をコンポーネントの電源として使用)

Node	CPU Architecture	Component Power Source
master-00.power-mon-dev-bm.qe.devcluster.openshift.com	Cypress Cove	rapl-sysfs
master-01.power-mon-dev-bm.qe.devcluster.openshift.com	Cypress Cove	rapl-sysfs
master-02.power-mon-dev-bm.qe.devcluster.openshift.com	Cypress Cove	rapl-sysfs
worker-00.power-mon-dev-bm.qe.devcluster.openshift.com	Cypress Cove	rapl-sysfs
worker-01.power-mon-dev-bm.qe.devcluster.openshift.com	Cypress Cove	rapl-sysfs

Kepler がハードウェアの電力消費メトリクスを取得できない場合、Components Source 列には電源として **estimator** が表示されます。この電源はテクノロジープレビューではサポート対象外です。このような場合、ノードからの値が正確ではありません。

Power Monitoring / Namespace ダッシュボード

このダッシュボードでは、namespace および Pod ごとにメトリクスを表示できます。次の情報を確認できます。

- DRAM や PKG の消費などの電力消費メトリクス
- 過去1時間のエネルギー消費メトリクス (コアおよび非コアコンポーネントの DRAM および PKG の消費量など)

この機能を使用すると、重要なピークを調査し、消費量が多い主な根本原因を簡単に特定できます。

5.2. クラスタ管理者として電力監視ダッシュボードにアクセスする

OpenShift Container Platform Web コンソールの Administrator パースペクティブから、パワーモニタリングダッシュボードにアクセスできます。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- **cluster-admin** ロールを持つユーザーとしてログインしている。
- Power monitoring Operator がインストールされている。
- Kepler がクラスタにデプロイされている。
- ユーザー定義プロジェクトのモニタリングが有効になっている。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Observe** → **Dashboards** に移動します。
2. **Dashboard** ドロップダウンリストから、表示するパワーモニタリングダッシュボードを選択します。
 - **Power Monitoring / Overview**
 - **Power Monitoring / Namespace**

5.3. 開発者として電力監視ダッシュボードにアクセスする

OpenShift Container Platform Web コンソールの **Developer** パースペクティブから、パワーモニタリングダッシュボードにアクセスできます。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- 開発者またはユーザーとしてクラスターにアクセスできる。
- Power monitoring Operator がインストールされている。
- Kepler がクラスターにデプロイされている。
- ユーザー定義プロジェクトのモニタリングが有効化されている。
- Kepler がデプロイされている namespace **openshift-power-monitoring** に対する **view** 権限があります。

手順

1. Web コンソールの **Developer** パースペクティブで、**Observe** → **Dashboard** に移動します。
2. **Dashboard** ドロップダウンリストから、表示するパワーモニタリングダッシュボードを選択します。
 - **Power Monitoring / Overview**

5.4. パワーモニタリングのメトリクスの概要

Power Monitoring Operator は以下のメトリクスを公開します。これらのメトリクスは、OpenShift Container Platform Web コンソールの **Observe** → **Metrics** タブを使用して表示できます。



警告

以下に示す公開されるメトリクスのリストは、最終的なものではありません。今後のリリースでメトリクスが追加または削除される可能性があります。

表5.1 Power monitoring Operator のメトリクス

メトリクス名	説明
kepler_container_joules_total	コンテナによる CPU、DRAM、およびその他のホストコンポーネントのパッケージまたはソケットのエネルギー消費量の合計。
kepler_container_core_joules_total	コンテナによって使用された CPU コア全体の合計エネルギー消費量。システムが RAPL_ メトリクスにアクセスできる場合、このメトリクスは、RAPL Power Plan 0 (PP0) の比例コンテナエネルギー消費量を示します。これは、ソケット内のすべての CPU コアによって消費されるエネルギー量です。
kepler_container_dram_joules_total	コンテナによる DRAM の総エネルギー消費量。
kepler_container_uncore_joules_total	コンテナによって使用された非コアコンポーネントによる累積エネルギー消費量。コンポーネントの数はシステムによって異なる場合があります。非コアメトリクスはプロセッサモデルに固有のものであり、サーバーの CPU によっては使用できない場合があります。
kepler_container_package_joules_total	コンテナが使用する CPU ソケットによって消費された累積エネルギー。これには、すべてのコアコンポーネントと非コアコンポーネントが含まれます。
kepler_container_other_joules_total	コンテナによって使用された、CPU と DRAM を除くホストコンポーネントの累積エネルギー消費量。通常、このメトリクスは ACPI ホストのエネルギー消費量です。
kepler_container_bpf_cpu_time_us_total	BPF トレーシングを使用するコンテナによって使用された合計 CPU 時間。
kepler_container_cpu_cycles_total	ハードウェアカウンターを利用するコンテナによって使用された合計 CPU サイクル。CPU サイクルは、CPU 周波数に直接関連するメトリクスです。プロセッサが固定周波数で実行されるシステムでは、CPU サイクルと合計 CPU 時間がほぼ同等になります。プロセッサがさまざまな周波数で実行されるシステムでは、CPU サイクルと合計 CPU 時間の値が異なります。
kepler_container_cpu_instructions_total	ハードウェアカウンターを利用するコンテナによって使用された CPU 命令の合計数。CPU 命令数は、CPU がどのように使用されているかを示す指標です。
kepler_container_cache_miss_total	ハードウェアカウンターを使用するコンテナで発生したキャッシュミスの合計。

メトリクス名	説明
kepler_container_cgroupfs_cpu_usage_us_total	コントロールグループの統計情報を読み取るコンテナによって使用された合計 CPU 時間。
kepler_container_cgroupfs_memory_usage_bytes_total	コントロールグループの統計情報を読み取るコンテナによって使用された合計メモリー (バイト単位)。
kepler_container_cgroupfs_system_cpu_usage_us_total	コントロールグループの統計情報を読み取るコンテナによって使用されたカーネル空間の合計 CPU 時間。
kepler_container_cgroupfs_user_cpu_usage_us_total	コントロールグループの統計情報を読み取るコンテナによって使用されたユーザー空間の合計 CPU 時間。
kepler_container_bpf_net_tx_irq_total	BPF トレーシングを使用するコンテナのネットワークカードに送信されたパケットの総数。
kepler_container_bpf_net_rx_irq_total	BPF トレーシングを使用するコンテナのネットワークカードから受信したパケットの総数。
kepler_container_bpf_block_irq_total	BPF トレーシングを使用するコンテナのブロック I/O 呼び出しの合計数。
kepler_node_info	ノードの CPU アーキテクチャーなどのノードのメタデータ。
kepler_node_core_joules_total	ノードおよびオペレーティングシステム上で実行されているすべてのコンテナによって使用された CPU コア全体の合計エネルギー消費量。
kepler_node_uncore_joules_total	ノードおよびオペレーティングシステム上で実行されているすべてのコンテナによって使用された非コアコンポーネントによる累積エネルギー消費量。コンポーネントの数はシステムによって異なる場合があります。
kepler_node_dram_joules_total	ノードおよびオペレーティングシステム上で実行されているすべてのコンテナによる DRAM の総エネルギー消費量。
kepler_node_package_joules_total	ノードおよびオペレーティングシステム上で実行されているすべてのコンテナによって使用される CPU ソケットによって消費された累積エネルギー。これには、すべてのコアコンポーネントと非コアコンポーネントが含まれます。

メトリクス名	説明
kepler_node_other_host_components_joules_total	ノードおよびオペレーティングシステム上で実行されているすべてのコンテナによって使用された、CPU および DRAM を除くホストコンポーネントの累積エネルギー消費量。通常、このメトリクスは ACPI ホストのエネルギー消費量です。
kepler_node_platform_joules_total	ホストの総エネルギー消費量。通常、このメトリクスは Redfish BMC または ACPI からのホストのエネルギー消費量です。
kepler_node_energy_stat	モデルサーバーで使用されるコンテナリソース使用率コントロールグループメトリクスのラベルが付いたノードから得られた複数のメトリクス。
kepler_node_accelerator_intel_qat	特定のノードにおける Intel QAT アクセラレーターの使用率。システムに Intel QAT が搭載されている場合、Kepler はテレメトリーを通じてノードの QAT の使用率を計算できます。

5.5. 関連情報

- [ユーザー定義プロジェクトのモニタリングの有効化](#)

第6章 パワーモニタリングのアンインストール



重要

パワーモニタリングはテクノロジープレビュー機能としてのみご利用いただけます。テクノロジープレビュー機能は、Red Hat 製品サポート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場合があります。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。

パワーモニタリングをアンインストールするには、OpenShift Container Platform Web コンソールで Kepler インスタンスを削除してから、Power monitoring Operator を削除します。

6.1. KEPLER の削除

Kepler を削除するには、OpenShift Container Platform Web コンソールから **Kepler** カスタムリソース定義 (CRD) の Kepler インスタンスを削除します。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- **cluster-admin** ロールを持つユーザーとしてログインしている。

手順

1. Web コンソールの **Administrator** パースペクティブで、**Operators** → **Installed Operators** に移動します。
2. **Installed Operators** リストから **Power monitoring for Red Hat OpenShift** をクリックし、**Kepler** タブに移動します。
3. リストで Kepler インスタンスのエントリーを見つけます。
4. このエントリーの  をクリックし、**Delete Kepler** を選択します。
5. **Delete Kepler?** ダイアログで、**Delete** をクリックして Kepler インスタンスを削除します。

6.2. POWER MONITORING OPERATOR のアンインストール

OperatorHub を使用して Power monitoring Operator をインストールした場合は、OpenShift Container Platform Web コンソールからアンインストールできます。

前提条件

- OpenShift Container Platform Web コンソールにアクセスできる。
- **cluster-admin** ロールを持つユーザーとしてログインしている。


手順

1. Kepler インスタンスを削除します。



警告

Power monitoring Operator をアンインストールする前に、Kepler インスタンスが削除されていることを確認してください。

2. **Operators** → **Installed Operators** に移動します。
3. リストで **Power monitoring for Red Hat OpenShift** エントリーを見つけます。
4. このエントリーの  をクリックし、**Uninstall Operator** を選択します。
5. **Uninstall Operator?** ダイアログで、**Uninstall** をクリックして Power monitoring Operator をアンインストールします。