OpenShift Container Platform 4.14

Power Monitoring

Configuring and using power monitoring for OpenShift Container Platform

Last Updated: 2024-06-28

OpenShift Container Platform 4.14 Power Monitoring

Configuring and using power monitoring for OpenShift Container Platform

Legal Notice

Abstract

Use power monitoring to monitor the power consumption for various components, such as CPU and DRAM, for each container running in an OpenShift Container Platform cluster.

Table of Contents

CHAPTER 1. POWER MONITORING FOR RED HAT OPENSHIFT RELEASE NOTES

IMPORTANT

Power monitoring is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

Power monitoring for Red Hat OpenShift enables you to monitor the power usage of workloads and identify the most power-consuming namespaces running in an OpenShift Container Platform cluster with key power consumption metrics, such as CPU or DRAM, measured at container level.

These release notes track the development of power monitoring for Red Hat OpenShift in the OpenShift Container Platform.

For an overview of the Power monitoring Operator, see About power monitoring.

1.1. POWER MONITORING 0.2 (TECHNOLOGY PREVIEW)

This release includes the following version updates:

- Kepler 0.7.10
- Power monitoring Operator 0.13.0

The following advisory is available for power monitoring 0.2:

- RHEA-2024:2923

1.1.1. Features

- With this update, you can enable the Redfish API in Kepler. Kepler uses Redfish to get the power consumed by nodes.
- With this update, you can install the Power monitoring Operator in the namespace of your choice.
- With this update, you can access the power monitoring **Overview** dashboard from the **Developer** perspective.

1.1.2. CVEs

- 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. POWER MONITORING 0.1 (TECHNOLOGY PREVIEW)

This release introduces a Technology Preview version of power monitoring for Red Hat OpenShift. The following advisory is available for power monitoring 0.1:

- RHEA-2024:0078

1.2.1. Features

- Deployment and deletion of Kepler
- Power usage metrics from Intel-based bare-metal deployments
- Dashboards for plotting power usage

CHAPTER 2. POWER MONITORING OVERVIEW

IMPORTANT

Power monitoring is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

2.1. ABOUT POWER MONITORING

You can use power monitoring for Red Hat OpenShift to monitor the power usage and identify power-consuming containers running in an OpenShift Container Platform cluster. Power monitoring collects and exports energy-related system statistics from various components, such as CPU and DRAM. It provides granular power consumption data for Kubernetes pods, namespaces, and nodes.

WARNING

Power monitoring Technology Preview works only in bare-metal deployments. Most public cloud vendors do not expose Kernel Power Management Subsystems to virtual machines.

2.2. POWER MONITORING ARCHITECTURE

Power monitoring is made up of the following major components:

The Power monitoring Operator

For administrators, the Power monitoring Operator streamlines the monitoring of power usage for workloads by simplifying the deployment and management of Kepler in an OpenShift Container Platform cluster. The setup and configuration for the Power monitoring Operator are simplified by adding a Kepler custom resource definition (CRD). The Operator also manages operations, such as upgrading, removing, configuring, and redeploying Kepler.

Kepler

Kepler is a key component of power monitoring. It is responsible for monitoring the power usage of containers running in OpenShift Container Platform. It generates metrics related to the power usage of both nodes and containers.

2.3. KEPLER HARDWARE AND VIRTUALIZATION SUPPORT

Kepler is the key component of power monitoring that collects real-time power consumption data from a node through one of the following methods:

Kernel Power Management Subsystem (preferred)

- **rapl-sysfs**: This requires access to the **/sys/class/powercap/intel-rapl** host file

- **rapl-sysfs**: This requires access to the **/sys/class/powercap/intel-rapl** host file.
- **rapl-msr**: This requires access to the **/dev/cpu/*/msr** host file.

The estimator power source

Without access to the kernel's power cap subsystem, Kepler uses a machine learning model to estimate the power usage of the CPU on the node.

> WARNING
>
> The **estimator** feature is experimental, not supported, and should not be relied upon.

You can identify the power estimation method for a node by using the **Power Monitoring / Overview** dashboard.

2.4. ADDITIONAL RESOURCES

- Power monitoring dashboards overview

CHAPTER 3. INSTALLING POWER MONITORING FOR RED HAT OPENSHIFT

IMPORTANT

Power monitoring is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

You can install power monitoring for Red Hat OpenShift by deploying the Power monitoring Operator in the OpenShift Container Platform web console.

3.1. INSTALLING THE POWER MONITORING OPERATOR

As a cluster administrator, you can install the Power monitoring Operator from OperatorHub by using the OpenShift Container Platform web console.

WARNING

You must remove any previously installed versions of the Power monitoring Operator before installation.

Prerequisites

- You have access to the OpenShift Container Platform web console.
- You are logged in as a user with the **cluster-admin** role.

Procedure

1. In the **Administrator** perspective of the web console, go to **Operators → OperatorHub**.
2. Search for **power monitoring**, click the **Power monitoring for Red Hat OpenShift**tile, and then click **Install**.
3. Click **Install** again to install the Power monitoring Operator.
   Power monitoring for Red Hat OpenShift is now available in all namespaces of the OpenShift Container Platform cluster.

Verification

1. Verify that the Power monitoring Operator is listed in **Operators → Installed Operators**. The **Status** should resolve to **Succeeded**.

3.2. DEPLOYING KEPLER

You can deploy Kepler by creating an instance of the **Kepler** custom resource definition (CRD) by using the Power monitoring Operator.

Prerequisites

- You have access to the OpenShift Container Platform web console.
- You are logged in as a user with the **cluster-admin** role.
- You have installed the Power monitoring Operator.

Procedure

1. In the **Administrator** perspective of the web console, go to **Operators → Installed Operators**.
2. Click **Power monitoring for Red Hat OpenShift** from the **Installed Operators** list and go to the **Kepler** tab.
3. Click **Create Kepler**.
4. On the **Create Kepler** page, ensure the **Name** is set to **kepler**.

   IMPORTANT

   The name of your Kepler instance must be set to **kepler**. All other instances are ignored by the Power monitoring Operator.

5. Click **Create** to deploy Kepler and power monitoring dashboards.

CHAPTER 4. CONFIGURING POWER MONITORING

IMPORTANT

Power monitoring is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

The **Kepler** resource is a Kubernetes custom resource definition (CRD) that enables you to configure the deployment and monitor the status of the Kepler resource.

4.1. THE KEPLER CONFIGURATION

You can configure Kepler with the **spec** field of the **Kepler** resource.

IMPORTANT

Ensure that the name of your Kepler instance is **kepler**. All other instances are rejected by the Power monitoring Operator Webhook.

The following is the list of configuration options:

Table 4.1. Kepler configuration options

Name	Spec	Description	Default
port	**exporter.deployment**	The port on the node where the Prometheus metrics are exposed.	**9103**
nodeSelector	**exporter.deployment**	The nodes on which Kepler exporter pods are scheduled.	**kubernetes.io/os: linux**
tolerations	**exporter.deployment**	The tolerations for Kepler exporter that allow the pods to be scheduled on nodes with specific characteristics.	**- operator: "Exists"**

Example Kepler resource with default configuration

```
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 The Prometheus metrics are exposed on port 9103.

❷ Kepler pods are scheduled on Linux nodes.

❸ The default tolerations allow Kepler to be scheduled on any node.

4.2. MONITORING THE KEPLER STATUS

You can monitor the state of the Kepler exporter with the **status** field of the **Kepler** resource.

The **status.exporter** field includes information, such as the following:

- The number of nodes currently running the Kepler pods
- The number of nodes that should be running the Kepler pods
- Conditions representing the health of the Kepler resource

This provides you with valuable insights into the changes made through the **spec** field.

Example state of the Kepler resource

```
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 ❷
  desiredNumberScheduled: 2 ❸
```

❶ The health of the Kepler resource. In this example, Kepler is successfully reconciled and ready.

❷ The number of nodes currently running the Kepler pods is 2.

❸ The wanted number of nodes to run the Kepler pods is 2.

4.3. CONFIGURING KEPLER TO USE REDFISH

You can configure Kepler to use Redfish as the source for running or hosting containers. Kepler can then monitor the power usage of these containers.

Prerequisites

- You have access to the OpenShift Container Platform web console.
- You are logged in as a user with the **cluster-admin** role.
- You have installed the Power monitoring Operator.

Procedure

1. In the **Administrator** perspective of the web console, click **Operators** → **Installed Operators**.
2. Click **Power monitoring for Red Hat OpenShift** from the **Installed Operators** list and click the **Kepler** tab.
3. Click **Create Kepler**. If you already have a Kepler instance created, click **Edit Kepler**.
4. Configure **.spec.exporter.redfish** of the Kepler instance by specifying the mandatory **secretRef** field. You can also configure the optional **probeInterval** and **skipSSLVerify** fields to meet your needs.

   Example Kepler instance

   ```
   apiVersion: kepler.system.sustainable.computing.io/v1alpha1
   kind: Kepler
   metadata:
     name: kepler
   spec:
     exporter:
       deployment:
   # ...
       redfish:
         secretRef: <secret_name> required ❶
         probeInterval: 60s ❷
         skipSSLVerify: false ❸
   # ...
   ```

1 Required: Specifies the name of the secret that contains the credentials for accessing the Redfish server.

2 Optional: Controls the frequency at which the power information is queried from Redfish. The default value is **60s**.

3 Optional: Controls if Kepler skips verifying the Redfish server certificate. The default value is **false**.

NOTE

After Kepler is deployed, the **openshift-power-monitoring** namespace is created.

5. Create the **redfish.csv** file with the following data format:

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

Example redfish.csv file

```
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. Create the secret under the **openshift-power-monitoring** namespace. You must create the secret with the following conditions:

- The secret type is **Opaque**.
- The credentials are stored under the **redfish.csv** key in the **data** field of the secret.

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

Example output

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

IMPORTANT

The Kepler deployment will not continue until the Redfish secret is created. You can find this information in the **status** of a Kepler instance.

CHAPTER 5. VISUALIZING POWER MONITORING METRICS

IMPORTANT

Power monitoring is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

You can visualize power monitoring metrics in the OpenShift Container Platform web console by accessing power monitoring dashboards or by exploring **Metrics** under the **Observe** tab.

5.1. POWER MONITORING DASHBOARDS OVERVIEW

There are two types of power monitoring dashboards. Both provide different levels of details around power consumption metrics for a single cluster:

Power Monitoring / Overview dashboard

With this dashboard, you can observe the following information:

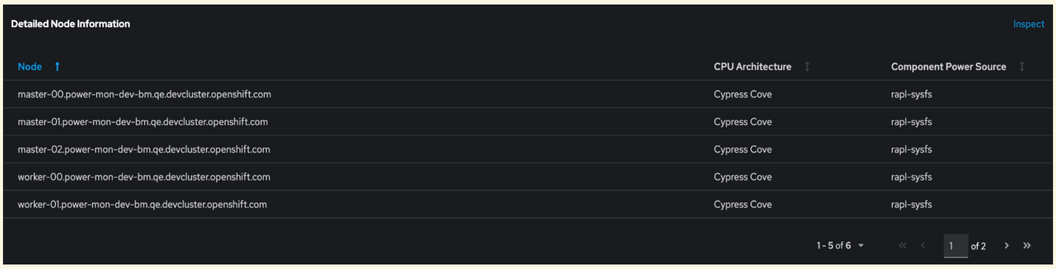
- An aggregated view of CPU architecture and its power source (**rapl-sysfs**, **rapl-msr**, or **estimator**) along with total nodes with this configuration
- Total energy consumption by a cluster in the last 24 hours (measured in kilowatt-hour)
- The amount of power consumed by the top 10 namespaces in a cluster in the last 24 hours
- Detailed node information, such as its CPU architecture and component power source

These features allow you to effectively monitor the energy consumption of the cluster without needing to investigate each namespace separately.

WARNING

Ensure that the **Components Source** column does not display **estimator** as the power source.

Figure 5.1. The Detailed Node Information table with**rapl-sysfs** as the component power source



Detailed Node Information

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

If Kepler is unable to obtain hardware power consumption metrics, the **Components Source** column displays **estimator** as the power source, which is not supported in Technology Preview. If that happens, then the values from the nodes are not accurate.

Power Monitoring / Namespace dashboard

This dashboard allows you to view metrics by namespace and pod. You can observe the following information:

- The power consumption metrics, such as consumption in DRAM and PKG
- The energy consumption metrics in the last hour, such as consumption in DRAM and PKG for core and uncore components

This feature allows you to investigate key peaks and easily identify the primary root causes of high consumption.

5.2. ACCESSING POWER MONITORING DASHBOARDS AS A CLUSTER ADMINISTRATOR

You can access power monitoring dashboards from the **Administrator** perspective of the OpenShift Container Platform web console.

Prerequisites

- You have access to the OpenShift Container Platform web console.
- You are logged in as a user with the **cluster-admin** role.
- You have installed the Power monitoring Operator.
- You have deployed Kepler in your cluster.
- You have enabled monitoring for user-defined projects.

Procedure

1. In the **Administrator** perspective of the web console, go to **Observe → Dashboards**.
2. From the **Dashboard** drop-down list, select the power monitoring dashboard you want to see:
   - **Power Monitoring / Overview**
   - **Power Monitoring / Namespace**

5.3. ACCESSING POWER MONITORING DASHBOARDS AS A DEVELOPER

You can access power monitoring dashboards from the **Developer** perspective of the OpenShift Container Platform web console.

Prerequisites

- You have access to the OpenShift Container Platform web console.
- You have access to the cluster as a developer or as a user.
- You have installed the Power monitoring Operator.
- You have deployed Kepler in your cluster.
- You have enabled monitoring for user-defined projects.
- You have **view** permissions for the namespace **openshift-power-monitoring**, the namespace where Kepler is deployed to.

Procedure

1. In the **Developer** perspective of the web console, go to **Observe → Dashboard**.
2. From the **Dashboard** drop-down list, select the power monitoring dashboard you want to see:
   - **Power Monitoring / Overview**

5.4. POWER MONITORING METRICS OVERVIEW

The Power monitoring Operator exposes the following metrics, which you can view by using the OpenShift Container Platform web console under the **Observe → Metrics** tab.

> **WARNING**
>
> This list of exposed metrics is not definitive. Metrics might be added or removed in future releases.

Table 5.1. Power monitoring Operator metrics

Metric name	Description
kepler_container_joules_total	The aggregated package or socket energy consumption of CPU, DRAM, and other host components by a container.
kepler_container_core_joules_total	The total energy consumption across CPU cores used by a container. If the system has access to **RAPL_** metrics, this metric reflects the proportional container energy consumption of the RAPL Power Plan 0 (PP0), which is the energy consumed by all CPU cores in the socket.
kepler_container_dram_joules_total	The total energy consumption of DRAM by a container.
kepler_container_uncore_joules_total	The cumulative energy consumption by uncore components used by a container. The number of components might vary depending on the system. The uncore metric is processor model-specific and might not be available on some server CPUs.
kepler_container_package_joules_total	The cumulative energy consumed by the CPU socket used by a container. It includes all core and uncore components.
kepler_container_other_joules_total	The cumulative energy consumption of host components, excluding CPU and DRAM, used by a container. Generally, this metric is the energy consumption of ACPI hosts.
kepler_container_bpf_cpu_time_us_total	The total CPU time used by the container that utilizes the BPF tracing.
kepler_container_cpu_cycles_total	The total CPU cycles used by the container that utilizes hardware counters. CPU cycles is a metric directly related to CPU frequency. On systems where processors run at a fixed frequency, CPU cycles and total CPU time are roughly equivalent. On systems where processors run at varying frequencies, CPU cycles and total CPU time have different values.
kepler_container_cpu_instructions_total	The total CPU instructions used by the container that utilizes hardware counters. CPU instructions is a metric that accounts how the CPU is used.
kepler_container_cache_miss_total	The total cache miss that occurs for a container that uses hardware counters.
kepler_container_cgroupfs_cpu_usage_us_total	The total CPU time used by a container reading from control group statistics.

Metric name	Description
kepler_container_cgroupfs_memory_usage_bytes_total	The total memory in bytes used by a container reading from control group statistics.
kepler_container_cgroupfs_system_cpu_usage_us_total	The total CPU time in kernel space used by the container reading from control group statistics.
kepler_container_cgroupfs_user_cpu_usage_us_total	The total CPU time in user space used by a container reading from control group statistics.
kepler_container_bpf_net_tx_irq_total	The total number of packets transmitted to network cards of a container that uses the BPF tracing.
kepler_container_bpf_net_rx_irq_total	The total number of packets received from network cards of a container that uses the BPF tracing.
kepler_container_bpf_block_irq_total	The total number of block I/O calls of a container that uses the BPF tracing.
kepler_node_info	The node metadata, such as the node CPU architecture.
kepler_node_core_joules_total	The total energy consumption across CPU cores used by all containers running on a node and operating system.
kepler_node_uncore_joules_total	The cumulative energy consumption by uncore components used by all containers running on the node and operating system. The number of components might vary depending on the system.
kepler_node_dram_joules_total	The total energy consumption of DRAM by all containers running on the node and operating system.
kepler_node_package_joules_total	The cumulative energy consumed by the CPU socket used by all containers running on the node and operating system. It includes all core and uncore components.
kepler_node_other_host_components_joules_total	The cumulative energy consumption of host components, excluding CPU and DRAM, used by all containers running on the node and operating system. Generally, this metric is the energy consumption of ACPI hosts.
kepler_node_platform_joules_total	The total energy consumption of the host. Generally, this metric is the host energy consumption from Redfish BMC or ACPI.

Metric name	Description
kepler_node_energy_stat	Multiple metrics from nodes labeled with container resource utilization control group metrics that are used in the model server.
kepler_node_accelerator_intel_qat	The utilization of the accelerator Intel QAT on a certain node. If the system contains Intel QATs, Kepler can calculate the utilization of the node’s QATs through telemetry.

5.5. ADDITIONAL RESOURCES

- Enabling monitoring for user-defined projects

CHAPTER 6. UNINSTALLING POWER MONITORING

IMPORTANT

Power monitoring is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope .

You can uninstall power monitoring by deleting the Kepler instance and then the Power monitoring Operator in the OpenShift Container Platform web console.

6.1. DELETING KEPLER

You can delete Kepler by removing the Kepler instance of the **Kepler** custom resource definition (CRD) from the OpenShift Container Platform web console.

Prerequisites

- You have access to the OpenShift Container Platform web console.
- You are logged in as a user with the **cluster-admin** role.

Procedure

1. In the **Administrator** perspective of the web console, go to **Operators → Installed Operators**.
2. Click **Power monitoring for Red Hat OpenShift** from the **Installed Operators** list and go to the **Kepler** tab.
3. Locate the Kepler instance entry in the list.
4. Click ⋮ for this entry and select **Delete Kepler**.
5. In the **Delete Kepler?** dialog, click **Delete** to delete the Kepler instance.

6.2. UNINSTALLING THE POWER MONITORING OPERATOR

If you installed the Power monitoring Operator by using OperatorHub, you can uninstall it from the OpenShift Container Platform web console.

Prerequisites

- You have access to the OpenShift Container Platform web console.
- You are logged in as a user with the **cluster-admin** role.

Procedure

1. Delete the Kepler instance.

   > **WARNING**
   >
   > Ensure that you have deleted the Kepler instance before uninstalling the Power monitoring Operator.

2. Go to **Operators** → **Installed Operators**.
3. Locate the **Power monitoring for Red Hat OpenShift** entry in the list.
4. Click ⋮ for this entry and select **Uninstall Operator**.
5. In the **Uninstall Operator?** dialog, click **Uninstall** to uninstall the Power monitoring Operator.