



Red Hat Advanced Cluster Management for Kubernetes 2.10

ヘルスメトリクス

ヘルスメトリクス

Red Hat Advanced Cluster Management for Kubernetes 2.10 ヘルスメ トリクス

ヘルスマトリクス

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

クラスターとコンポーネントのメトリクスと全体のモニタリングを説明します。

目次

第1章 ヘルスメトリクス	3
1.1. メトリクスサービスの使用	3

第1章 ヘルスメトリクス

メトリクスを使用してコンポーネントの正常性を監視できます。

以下のドキュメントを参照してください。

- [メトリクスサービスの使用](#)

1.1. メトリクスサービスの使用

メトリクスを使用して、Red Hat Advanced Cluster Management for Kubernetes 全体のコンポーネントの正常性を監視できます。多くのカスタムメトリックについては、[メトリッククロニクルの概要](#)に記載されています。

1.1.1. ハブクラスターメトリクスサービスへのアクセス

収集されたメトリクスを表示するには、ハブクラスター上でメトリクスサービスを公開する必要があります。メトリクスがすでに Grafana ダッシュボードに公開されている場合、この手順はオプションです。

OpenShift Container Platform コンソールから、メトリクスサービスを見つけます。 **Observe > Metrics** をクリックします。

Grafana ダッシュボード、Grafana Explorer、または OpenShift Container Platform コンソールにメトリクスが表示されない場合は、Prometheus がメトリクスを取得するように設定されていない可能性があります。 **Prometheus を使用したスクラッチ** に進み、メトリクスを公開してください。

1.1.2. Prometheus を使用したスクレイピング

Prometheus を使用すると、製品コンソールから公開されていないメトリクスを公開できます。ハブとマネージドクラスターのメトリクスの手順を参照してください。

1.1.2.1. ハブクラスターのスクレイピング

ハブクラスターのメトリクスを公開する手順は次のとおりです。これらのファイルは、**openshift-monitoring** namespace 内にあります。

1. サービスを収集し、メトリクスを公開するための **ServiceMonitor** を作成します。以下は、YAML の例です。

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: hub-subscription-metrics
  namespace: openshift-monitoring
spec:
  endpoints:
    - port: metrics
  namespaceSelector:
    matchNames:
      - open-cluster-management
  selector:
    matchLabels:
      app: hub-subscription-metrics
```

2. 次のコマンドを実行して、ファイルを適用します。

```
oc apply -f
```

3. モニタリング権限を設定する **Role** を作成します。以下は、その YAML ファイルです。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: prometheus-k8s-monitoring
  namespace: open-cluster-management
rules:
- apiGroups:
  - ""
  resources:
  - services
  - endpoints
  - pods
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - extensions
  resources:
  - ingresses
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses
  verbs:
  - get
  - list
  - watch
```

4. 次のコマンドを実行して、ファイルを適用します。

```
oc apply -f
```

5. 次の例のように、Prometheus モニタリング **ServiceAccount** にロールをバインドするための **RoleBinding** を作成します。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: prometheus-k8s-monitoring-binding
  namespace: open-cluster-management
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: prometheus-k8s-monitoring
```

```
subjects:  
- kind: ServiceAccount  
  name: prometheus-k8s  
  namespace: monitoring
```

6. 次のコマンドを実行して、ファイルを適用します。

```
oc apply -f
```

7. 検証するには、ダッシュボードで次のクエリーを実行し、Subscription Operator Metrics Service により報告されるメトリクスを見つけます。

```
{service="hub-subscription-metrics"}
```

1.1.2.2. マネージドクラスターのスクレイピング

マネージドクラスターのメトリクスを公開するには、次の手順を実行します。これらのファイルは、**openshift-monitoring** namespace 内にあります。

1. メトリクスを公開するサービスを収集する **ServiceMonitor** を作成します。以下は、YAML ファイルの例です。

```
apiVersion: monitoring.coreos.com/v1  
kind: ServiceMonitor  
metadata:  
  name: mc-subscription-metrics  
  namespace: openshift-monitoring  
spec:  
  endpoints:  
    - port: metrics  
  namespaceSelector:  
    matchNames:  
    - open-cluster-management-agent-addon  
  selector:  
    matchLabels:  
      app: mc-subscription-metrics
```

2. 次のコマンドを実行してファイルを適用します。

```
oc apply -f
```

3. モニタリング権限を設定する **Role** を作成します。以下は、YAML ファイルの例です。

```
apiVersion: rbac.authorization.k8s.io/v1  
kind: Role  
metadata:  
  name: prometheus-k8s-monitoring  
  namespace: open-cluster-management-agent-addon  
rules:  
- apiGroups:  
  - ""  
  resources:  
  - services  
  - endpoints
```

```

- pods
verbs:
- get
- list
- watch
- apiGroups:
- extensions
resources:
- ingresses
verbs:
- get
- list
- watch
- apiGroups:
- networking.k8s.io
resources:
- ingresses
verbs:
- get
- list
- watch

```

4. 次のコマンドを実行して、ファイルを適用します。

```
oc apply -f
```

5. Prometheus モニタリング **ServiceAccount** に **Role** をバインドする **RoleBinding** を作成します。

```

apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: prometheus-k8s-monitoring-binding
  namespace: open-cluster-management-agent-addon
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: prometheus-k8s-monitoring
subjects:
- kind: ServiceAccount
  name: prometheus-k8s
  namespace: monitoring

```

6. 次のコマンドを実行して、ファイルを適用します。

```
oc apply -f
```

7. **Prometheus** ダッシュボードで次のクエリーを実行し、Subscription Operator Metrics Service により報告されるメトリクスを見つけて検証します。

```
{service="mc-subscription-metrics"}
```

1.1.3. スタンドアロンクラスターのスクレイピング

1. メトリクスを公開するサービスを収集するための **ServiceMonitor** を作成します。

```
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  name: standalone-subscription-metrics
  namespace: openshift-monitoring
spec:
  endpoints:
    - port: metrics
  namespaceSelector:
    matchNames:
      - open-cluster-management
  selector:
    matchLabels:
      app: standalone-subscription-metrics
```

2. モニタリング権限を設定するための **Role** を作成します。

```
oc apply -f
```

```
apiVersion: rbac.authorization.k8s.io/v1
kind: Role
metadata:
  name: prometheus-k8s-monitoring
  namespace: open-cluster-management
rules:
- apiGroups:
  - ""
  resources:
  - services
  - endpoints
  - pods
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - extensions
  resources:
  - ingresses
  verbs:
  - get
  - list
  - watch
- apiGroups:
  - networking.k8s.io
  resources:
  - ingresses
  verbs:
  - get
  - list
  - watch
```

- Prometheus センタリング **ServiceAccount** に **Role** をバインドするための **RoleBinding** を作成します。以下は、YAML ファイルの例です。

```
apiVersion: rbac.authorization.k8s.io/v1
kind: RoleBinding
metadata:
  name: prometheus-k8s-monitoring-binding
  namespace: open-cluster-management
roleRef:
  apiGroup: rbac.authorization.k8s.io
  kind: Role
  name: prometheus-k8s-monitoring
subjects:
- kind: ServiceAccount
  name: prometheus-k8s
  namespace: monitoring
```

- 次のコマンドを実行して、ファイルを適用します。

```
oc apply -f
```

- Prometheus** ダッシュボードで次のクエリーを実行し、Subscription Operator Metrics Service により報告されるメトリクスを見つけて検証します。

```
{service="standalone-subscription-metrics"}
```