



# Red Hat Advanced Cluster Management for Kubernetes 2.10

상태 메트릭

상태 메트릭



상태 메트릭

## 법적 공지

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.

## 초록

클러스터 및 구성 요소 전체의 메트릭 및 모니터링에 대해 자세히 알아보려면 자세히 알아보십시오.

## 차례

1장. 상대 메트릭 .....	3
1.1. 메트릭 서비스 사용	3



## 1장. 상태 메트릭

메트릭을 사용하여 구성 요소의 상태를 모니터링할 수 있습니다.

다음 설명서를 참조하십시오.

- [메트릭 서비스 사용](#)

### 1.1. 메트릭 서비스 사용

메트릭을 사용하여 Red Hat Advanced Cluster Management for Kubernetes에서 구성 요소 상태를 모니터링할 수 있습니다. 많은 사용자 정의 메트릭은 [Metrics chronicle 개요](#)에 설명되어 있습니다.

#### 1.1.1. 허브 클러스터 메트릭 서비스에 액세스

수집된 지표를 보려면 hub 클러스터에 메트릭 서비스를 노출해야 합니다. Grafana 대시보드에 메트릭이 이미 노출되어 있는 경우 이 절차는 선택 사항입니다.

OpenShift Container Platform 콘솔에서 metrics 서비스를 찾습니다. **모니터링 > 메트릭을 클릭합니다.**

Grafana 대시보드, Grafana Explorer 또는 OpenShift Container Platform 콘솔에 메트릭이 표시되지 않는 경우 메트릭을 스크랩하도록 Prometheus가 구성되지 않을 수 있습니다. 메트릭을 노출하려면 *Prometheus를 사용하여 Scrapping*을 계속합니다.

#### 1.1.2. Prometheus를 사용하여 스크랩

Prometheus를 사용하여 제품 콘솔에서 노출하지 않는 지표를 노출할 수 있습니다. 허브 및 관리 클러스터 메트릭 모두에 대한 절차를 참조하십시오.

##### 1.1.2.1. 허브 클러스터 스크랩

hub 클러스터에 대한 메트릭을 노출하려면 다음 절차를 참조하십시오. 이러한 파일은 **openshift-monitoring** 네임스페이스에 있습니다.

1. 서비스 수집 및 메트릭 노출을 위한 **ServiceMonitor**를 생성합니다. 다음 YAML 예제를 참조하십시오.

```
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. 다음 명령을 실행하여 파일을 적용합니다.

```
oc apply -f
```

3. 모니터링에 대한 권한을 설정하기 위한 역할을 만듭니다. 다음 YAML 파일을 참조하십시오.

```
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. 다음 명령을 실행하여 파일을 적용합니다.

```
oc apply -f
```

5. 다음 예와 같이 Prometheus 모니터링 **ServiceAccount** 에 역할을 바인딩하기 위한 **RoleBinding** 을 생성합니다.

```
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
```

- 다음 명령을 실행하여 파일을 적용합니다.

```
oc apply -f
```

- 확인하려면 대시보드에서 다음 쿼리를 실행하여 Subscription Operator Metrics 서비스에서 보고한 메트릭을 찾습니다.

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

### 1.1.2.2. 관리 클러스터 스크랩

관리 클러스터에 대한 메트릭을 노출하려면 다음 절차를 참조하십시오. 이러한 파일은 **openshift-monitoring** 네임스페이스에 있습니다.

- 메트릭을 노출하는 서비스 수집을 위한 **ServiceMonitor** 를 생성합니다. 다음 YAML 파일 예제를 참조하십시오.

```
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
```

- 다음 명령을 실행하여 파일을 적용합니다.

```
oc apply -f
```

- 모니터링에 대한 권한을 설정하기 위한 **역할**을 만듭니다. 다음 YAML 파일 예제를 참조하십시오.

```
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. 다음 명령을 실행하여 파일을 적용합니다.

```
oc apply -f
```

5. 역할을 Prometheus 모니터링 **ServiceAccount** 에 바인딩 하기 위한 **RoleBinding** 을 생성합니다.

```

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. 다음 명령을 실행하여 파일을 적용합니다.

```
oc apply -f
```

7. Subscription Operator Metrics 서비스에서 보고되는 지표를 찾기 위해 다음 쿼리를 실행하여 **Prometheus** 대시보드에서 확인합니다.

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

### 1.1.3. 독립 실행형 클러스터 스크랩

1. 메트릭을 노출하는 서비스 수집을 위한 **ServiceMonitor** 를 생성합니다.

```

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. 모니터링에 대한 권한을 설정하기 위한 역할을 생성합니다.

```
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

```

3. 역할을 Prometheus 모니터링 **ServiceAccount** 에 바인딩 하기 위한 **RoleBinding** 을 생성합니다. 다음 YAML 파일 예제를 참조하십시오.

■

```
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
```

4. 다음 명령을 실행하여 파일을 적용합니다.

```
oc apply -f
```

5. Subscription Operator Metrics 서비스에서 보고되는 지표를 찾기 위해 다음 쿼리를 실행하여 **Prometheus** 대시보드에서 확인합니다.

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