Red Hat Advanced Cluster Management for Kubernetes 2.1

Observing environments

Observing environments

Last Updated: 2022-01-27

Red Hat Advanced Cluster Management for Kubernetes 2.1 Observing environments

Observing environments

Legal Notice

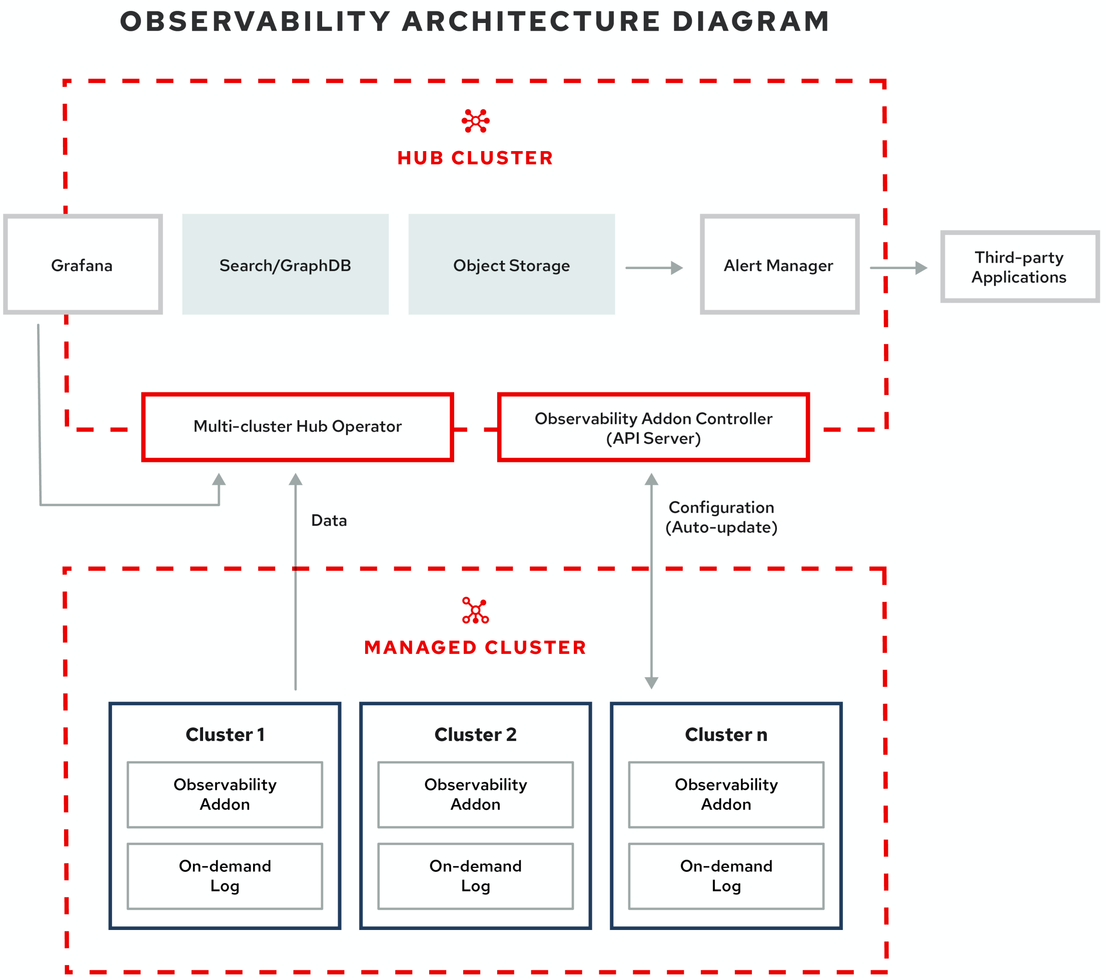
Abstract

Observe environments in Red Hat Advanced Cluster Management for Kubernetes

Table of Contents

CHAPTER 1. OBSERVING ENVIRONMENTS

You can use Red Hat Advanced Cluster Management for Kubernetes to gain insight and optimize your managed clusters. Enable the observability service operator, **multicluster-observability-operator**, to monitor the health of your managed clusters. Learn about the architecture for the multicluster observability service in the following sections.



1.1. OBSERVABILITY SERVICE

By default, observability is included with the product installation, but not enabled. Due to the requirement for persistent storage, the observability service is not enabled by default. Red Hat Advanced Cluster Management supports the following stable object stores:

- Amazon S3 (or other S3 compatible object stores like Ceph)
- Google Cloud Storage
- Azure storage

When the service is enabled the **observability-endpoint-controller** is automatically deployed to each imported or created cluster. This controller collects the data from Red Hat OpenShift Container Platform Prometheus, then sends it to the Red Hat Advanced Cluster Management hub cluster.

Note: In Red Hat Advanced Cluster Management the **metrics-collector** is only supported for Red Hat OpenShift Container Platform 4.x clusters.

The observability service deploys an instance of Prometheus AlertManager, which enables alerts to be forwarded with third-party applications. It also includes an instance of Grafana to enable data visualization via dashboards (static) or data exploration.

Customize the observability. You can create custom recording rules or alerting rules.

For more information about enabling observability, see Enable observability service.

1.2. OBSERVABILITY CONSOLE PAGES

You can also view metric data from your managed clusters by selecting the Grafana link from the following console pages:

- *Overview*: Operation details across providers
- *Topology*: Visual data for clusters, applications, and policies

1.2.1. Observe environments Overview page

You can view the following information about your clusters on the *Overview* dashboard:

- Cluster, node, and pod counts across all clusters and for each provider
- Cluster status
- Cluster compliance
- Pod status

Many clickable elements on the dashboard open a search for related resources. Click on a provider card to view information for clusters from a single provider. You can personalize your view of the *Overview* dashboard by clicking and dragging to reorganize the cards.

1.2.2. Topology page

The *Topology* page displays related Kubernetes resources within a cluster. As you configure managed clusters, you see more clusters in the Topology view.

To reduce the graphics on the page, you can filter the view by Clusters, Namespaces, and Labels. You can also filter the design by selecting the icon that represents the Kubernetes resources.

Learn more about the tabs that are available from the Topology page:

- Clusters: You can monitor your cluster network, object network, and security policies in a graphical format. View your hub clusters, all your managed clusters, and monitor security violations.
- Policies: View the policy, policy placement, and clusters that are being validated. Check for violations for the selected policy.

1.3. ENABLE OBSERVABILITY SERVICE

Monitor the health of your managed clusters with the observability service (**multicluster-observability-operator**).

Required access: Cluster administrator or the **open-cluster-management:cluster-manager-admin** role.

Prerequisites:

- You must install Red Hat Advanced Cluster Management for Kubernetes. See Installing while connected online for more information.

- You must configure an object store to create a storage solution. Red Hat Advanced Cluster Management only supports cloud providers with stable object stores, such as Amazon S3 (or other S3 compatible object stores like Ceph), Google Cloud Storage, and Azure storage. **Important**: When you configure your object store, ensure that you meet the encryption requirements necessary when sensitive data is persisted. For a complete list of the supported object stores, see Thanos documentation.

1.3.1. Enabling observability

Enable the observability service by creating a MultiClusterObservability CustomResource (CR) instance. Complete the following steps to enable the observability service:

1. Log in to your Red Hat Advanced Cluster Management hub cluster.

2. Create a namespace for the observability service with the following command:

   ```
   oc create namespace open-cluster-management-observability
   ```

3. Generate your pull-secret. If Red Hat Advanced Cluster Management is installed in the **open-cluster-management** namespace, run the following command:

   ```
   DOCKER_CONFIG_JSON=`oc extract secret/multiclusterhub-operator-pull-secret -n open-cluster-management --to=-`
   oc create secret generic multiclusterhub-operator-pull-secret \
       -n open-cluster-management-observability \
       --from-literal=.dockerconfigjson="$DOCKER_CONFIG_JSON" \
       --type=kubernetes.io/dockerconfigjson
   ```

   If the **multiclusterhub-operator-pull-secret** is not defined in the namespace, copy the **pull-secret** from the **openshift-config** namespace into the **open-cluster-management-observability** namespace. Run the following command:

   ```
   DOCKER_CONFIG_JSON=`oc extract secret/pull-secret -n openshift-config --to=-`
   oc create secret generic multiclusterhub-operator-pull-secret \
       -n open-cluster-management-observability \
       --from-literal=.dockerconfigjson="$DOCKER_CONFIG_JSON" \
       --type=kubernetes.io/dockerconfigjson
   ```

4. Create a secret for your object storage. Your secret must contain the credentials to your storage solution. For example, run the following command:

   ```
   oc create -f thanos-object-storage.yaml -n open-cluster-management-observability
   ```

   a. View the following examples of secrets for the supported object stores:

i. For Amazon S3 or S3 compatible, your secret might resemble the following file:

```
apiVersion: v1
kind: Secret
metadata:
  name: thanos-object-storage
type: Opaque
stringData:
  thanos.yaml: |
    type: s3
    config:
      bucket: YOUR_S3_BUCKET
      endpoint: YOUR_S3_ENDPOINT
      insecure: false
      access_key: YOUR_ACCESS_KEY
      secret_key: YOUR_SECRET_KEY
```

ii. For Google, your secret might resemble the following file:

```
type: GCS
config:
  bucket: YOUR_GCS_BUCKET
  service_account: YOUR_SERVICE_ACCOUNT
```

iii. For Azure your secret might resemble the following file:

```
type: AZURE
config:
  storage_account: YOUR_STORAGE_ACCT
  storage_account_key: YOUR_STORAGE_KEY
  container: YOUR_CONTAINER
  endpoint: YOUR_ENDPOINT
  max_retries: 0
```

5. Create the **MultiClusterObservability** custom resource (mco CR) for your managed cluster by completing the following steps:

   a. Create the **MultiClusterObservability** custom resource YAML file named **multiclusterobservability_cr.yaml**.
   View the following default YAML file for observability:

```
apiVersion: observability.open-cluster-management.io/v1beta1
kind: MultiClusterObservability
metadata:
  name: observability #Your customized name of MulticlusterObservability CR
spec:
  availabilityConfig: High # Available values are High or Basic
  imagePullPolicy: Always
  imagePullSecret: multiclusterhub-operator-pull-secret
  observabilityAddonSpec: # The ObservabilityAddonSpec defines the global settings for
all managed clusters which have observability add-on enabled
    enableMetrics: true # EnableMetrics indicates the observability addon push metrics to
hub server
    interval: 60 # Interval for the observability addon push metrics to hub server
  retentionResolution1h: 30d # How long to retain samples of 1 hour in bucket
```

```
 retentionResolution5m: 14d
 retentionResolutionRaw: 5d
 storageConfigObject: # Specifies the storage to be used by Observability
  metricObjectStorage:
   name: thanos-object-storage
   key: thanos.yaml
  statefulSetSize: 10Gi # The amount of storage applied to the Observability
StatefulSets, i.e. Amazon S3 store, Rule, compact and receiver.
  statefulSetStorageClass: gp2
```

You might want to modify the value for the **retentionResolution** parameter. For more information, see Thanos Downsampling resolution and retention. Depending on the number of managed clusters, you might want to update **statefulSetSize**, see Observability API for more information.

b. To deploy on infrastructure machine sets, you must set a label for your set by updating the ***nodeSelector*** in the MultiClusterObservability YAML. Your YAML might resemble the following content:

```
nodeSelector:
  node-role.kubernetes.io/infra:
```

For more information, see Creating infrastructure machine sets.

c. Apply the observability YAML to your cluster by running the following command:

```
oc apply -f multiclusterobservability_cr.yaml
```

All the pods in **open-cluster-management-observability** namespace for Thanos, Grafana and AlertManager are created. All the managed clusters connected to the Red Hat Advanced Cluster Management hub cluster are enabled to send metrics back to the Red Hat Advanced Cluster Management Observability service.

6. To validate that the observability service is enabled, launch the Grafana dashboards to make sure the data is populated. Complete the following steps:

   a. Log in to the Red Hat Advanced Cluster Management console.

   b. From the navigation menu, select **Observe environments** > **Overview**.

   c. Click the Grafana link that is near the console header to view the metrics from your managed clusters.
   Note: If you want to exclude specific managed clusters from collecting the observability data, add the following cluster label to your clusters: **vendor: OpenShift**.

1.3.2. Disabling observability

To disable the observability service, uninstall the **observability** resource. See step 1 of Removing a MultiClusterHub instance by using commands for the procedure.

To learn more about customizing the observability service, see Customizing observability.

1.4. CUSTOMIZING OBSERVABILITY

Review the following sections to learn more about customizing, managing, and viewing data that is collected by the observability service.

Collect logs about new information that is created for observability resources with the **must-gather** command. For more information, see the *Must-gather* section in the Troubleshooting documentation.

- Creating custom rules
- Configuring rules for AlertManager
- Viewing and exploring data
- Disable *metrics-collector*

1.4.1. Creating custom rules

You can create custom rules for the observability installation by adding Prometheus recording rules and alerting rules to the observability resource. For more information, see Prometheus configuration.

Note: You can only create custom rules on the metrics that are collected from all managed clusters. View a list of of the metrics that are collected by running the following command: **kubectl describe cm observability-metrics-whitelist**.

Define custom rules with Prometheus to create alert conditions, and send notifications to an external messaging service. Complete the following steps to create a custom rule:

1. Log in to your Red Hat Advanced Cluster Management hub cluster.

2. Create a ConfigMap named **thanos-rule-custom-rules** in the **open-cluster-management-observability** namespace. The key must be named, **thanos-ruler-custom-rules.yaml**, as shown in the following example. You can create multiple rules in the configuration:
   By default, the out-of-the-box alert rules are defined in the ConfigMap in the **open-cluster-management-observability** namespace.

   For example, you can create a custom alert rule that notifies you when your CPU usage passes your defined value:

```
data:
  custom_rules.yaml: |
    groups:
      - name: cluster-health
        rules:
        - alert: ClusterCPUHealth-jb
          annotations:
            summary: Notify when CPU utilization on a cluster is greater than the defined
utilization limit
            description: "The cluster has a high CPU usage: {{ $value }} core for {{ $labels.cluster
}} {{ $labels.clusterID }}."
          expr: |
            max(cluster:cpu_usage_cores:sum) by (clusterID, cluster, prometheus) > 0
          for: 5s
          labels:
            cluster: "{{ $labels.cluster }}"
            prometheus: "{{ $labels.prometheus }}"
            severity: critical
```

Note: If this is the first new custom rule, it is created immediately. For changes to the ConfigMap, you must restart the observability pods with the following command: **kubectl rollout restart statefulset observability-observatorium-thanos-rule -n open-cluster-management-observability**.

3. If you want to verify that the alert rules is functioning appropriately, complete the following steps:

    a. Access your Grafana dashboard and select the **Explore** icon.

    b. In the Metrics exploration bar, type in "ALERTS" and run the query. All the ALERTS that are currently in pending or firing state in the system are displayed.

    c. If your alert is not displayed, revisit the rule to see if the expression is accurate.

A custom rule is created.

1.4.1.1. Configuring rules for AlertManager

Integrate external messaging tools such as email, Slack, and PagerDuty to receive notifications from AlertManager. You must override the **alertmanager-config** secret in the **open-cluster-management-observability** namespace to add integrations, and configure routes for AlertManager. Complete the following steps to update the custom receiver rules:

1. Extract the data from the **alertmanager-config** secret. Run the following command:

```
oc -n open-cluster-management-observability get secret alertmanager-config --template='{{
index .data "alertmanager.yaml" }}' |base64 -d > alertmanager.yaml
```

2. Edit and save the **alertmanager.yaml** file configuration by running the following command:

```
oc -n open-cluster-management-observability create secret generic alertmanager-config --
from-file=alertmanager.yaml --dry-run -o=yaml |  oc -n open-cluster-management-
observability replace secret --filename=-
```

Your updated secret might resemble the following content:

```
global
  smtp_smarthost: 'localhost:25'
  smtp_from: 'alertmanager@example.org'
  smtp_auth_username: 'alertmanager'
  smtp_auth_password: 'password'
templates:
- '/etc/alertmanager/template/*.tmpl'
route:
  group_by: ['alertname', 'cluster', 'service']
  group_wait: 30s
  group_interval: 5m
  repeat_interval: 3h
  receiver: team-X-mails
  routes:
  - match_re:
      service: ^(foo1|foo2|baz)$
    receiver: team-X-mails
```

Your changes are applied immediately after it is modified. For an example of AlertManager, see prometheus/alertmanager.

1.4.2. Viewing and exploring data

View the data from your managed clusters by accessing Grafana. Complete the following steps to view the Grafana dashboards from the console:

1. Log in to your Red Hat Advanced Cluster Management hub cluster.
2. From the navigation menu, select **Observe environments** > **Overview** > **Grafana link**. You can also access Grafana dashboards from the *Clusters* page. From the navigation menu, select **Automate infrastructure** > **Clusters** > **Grafana**.
3. Access the Prometheus metric explorer by selecting the **Explore** icon from the Grafana navigation menu.

1.4.3. Disable *metrics-collector*

You can disable the **metrics-collector**, which stops it from collecting the data and sending the collection data to the observability service.

1.4.3.1. Disable *metrics-collector* on all clusters

Disable the **metrics-collector** pod to stop data from being collected and sent to the observability service on the Red Hat Advanced Cluster Management hub cluster.

When you disable the **metrics-collector** deployment is scaled to zero, and all managed clusters with the **vendor:OpenShift** label are disabled. View the following options to disable the **metrics-collector**:

Update the **multicluster-observability-operator** resource by setting **enableMetrics** to **false**. Your updated resource might resemble the following change:

```
spec:
  availabilityConfig: High # Available values are High or Basic
  imagePullPolicy: Always
  imagePullSecret: multiclusterhub-operator-pull-secret
  observabilityAddonSpec: # The ObservabilityAddonSpec defines the global settings for all managed
clusters which have observability add-on enabled
    enableMetrics: false #indicates the observability addon push metrics to hub server
```

1.4.3.2. Disable *metrics-collector* on a single cluster

You can disable the **metrics-collector** on specific managed clusters by completing one of the following procedures:

- Add the **observability: disabled** label to the custom resource, **managedclusters.cluster.open-cluster-management.io**.
- From the Red Hat Advanced Cluster Management console *Clusters* page, add the **observability: disabled** label by completing the following steps:
  1. In the Red Hat Advanced Cluster Management console navigation, select **Automate infrastructure** > **Clusters**.

2. Select the name of the cluster for which you want to disable data collection that is sent to observability.

3. Select **Labels**.

4. Create the label that disables the observability collection by adding the following label:

```
observability=disable
```

5. Select **Add** to add the label.

6. Select **Done** to close the list of labels.

Note: When a managed cluster with the observability component is detached, the **metrics-collector** deployments are removed.

For more information on monitoring data from the console with the observability service, see Observing environments.