Red Hat Advanced Cluster Management for Kubernetes 2.3

Observability（可观察性）

请参阅更多信息，了解如何启用和自定义可观察性服务来优化受管集群。

Last Updated: 2023-02-24

Red Hat Advanced Cluster Management for Kubernetes 2.3 Observability（可观察性）

请参阅更多信息，了解如何启用和自定义可观察性服务来优化受管集群。

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律通告

摘要

请参阅更多信息，了解如何启用和自定义可观察性服务来优化受管集群。

目录

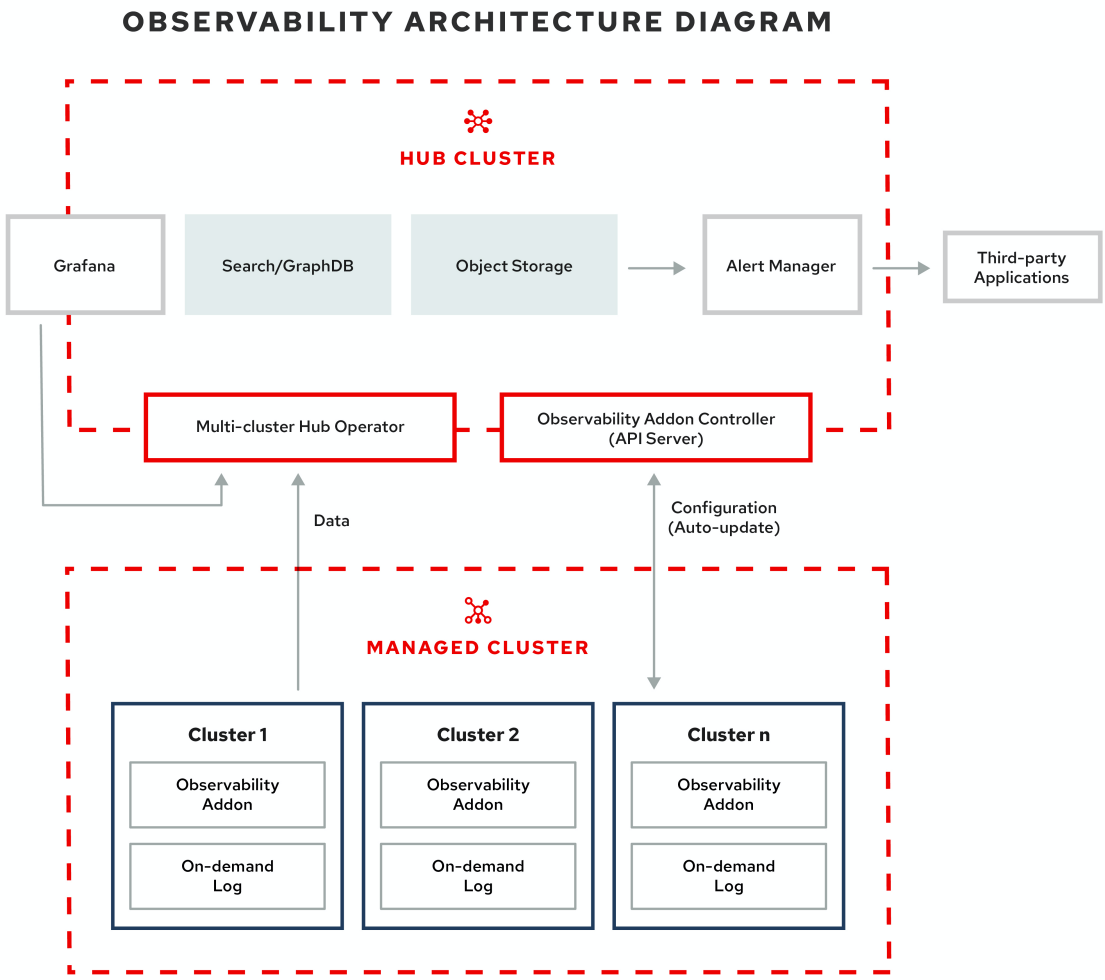
第 1 章 观察环境简介

启用可观察（observability）服务后，您可以使用 Red Hat Advanced Cluster Management for Kubernetes 深入了解受管集群并进行优化。这些信息可以帮助节约成本并防止不必要的事件发生。

- 观察环境
- 启用 observability 服务
- 定制可观察性
- 设计您的 Grafana 仪表板

1.1. 观察环境

您可以使用 Red Hat Advanced Cluster Management for Kubernetes 深入了解受管集群并进行优化。启用 observability 服务 operator（**multicluster-observability-operator**）以监控受管集群的健康状态。在以下部分了解多集群观察服务的架构。



注：*按需（on-demand）日志*允许工程师实时访问指定 pod 的日志。hub 集群的日志不会被聚合。这些日志可以通过搜索服务以及控制台的其他部分进行访问。

- 观察（Observability）服务
- 指标类型
- Observability pod 容量请求
- Observability 服务中使用的持久性存储
- 支持

1.1.1. 观察（Observability）服务

默认情况下，产品安装中包含了可观察性（observability）功能，但不启用它。由于对持久性存储的要求，observability 服务默认不会启用。Red Hat Advanced Cluster Management 支持以下 S3 兼容、稳定的对象存储：

- Amazon S3
  注：Thanos 中的对象存储接口支持兼容 AWS S3 restful API 的 API，或其他 S3 兼容对象存储，如 Minio 和 Ceph。
- Google Cloud Storage
- Azure 存储
- Red Hat OpenShift Container Storage
  重要：当您配置对象存储时，请确保在敏感数据持久化时满足加密要求。如需支持的对象存储的完整列表，请参阅 Thanos 文档。

启用该服务后，**observability-endpoint-operator** 会自动部署到每个导入或创建的集群中。此控制器从 Red Hat OpenShift Container Platform Prometheus 收集数据，然后将其发送到 Red Hat Advanced Cluster Management hub 集群。

如果 hub 集群将自身导入为 **local-cluster**，则也会启用可观察性，并从 hub 集群收集指标。

注：在 Red Hat Advanced Cluster Management 中，只有 Red Hat OpenShift Container Platform 4.x 集群支持 **metrics-collector**。

observability 服务部署了一个 Prometheus AlertManager 实例，它允许通过第三方应用程序转发警报。它还包括一个 Grafana 实例，通过仪表板（静态）或数据探索启用数据视觉化。Red Hat Advanced Cluster Management 支持 Grafana 的版本 7.4.2。您还可以设计自己的 Grafana 仪表板。如需更多信息，请参阅设计 Grafana 仪表板。

您可以通过创建自定义 记录规则 或 警报规则来自定义可观察性服务。

有关启用可观察性的更多信息，请参阅启用可观察性服务。

1.1.2. 指标类型

默认情况下，OpenShift Container Platform 使用 Telemetry 服务向红帽发送指标数据。以下附加指标包括在 Red Hat Advanced Cluster Management 中，它们包含在遥测（telemetry）中，但 *不会*显示在 Red Hat Advanced Cluster Management *Observe environments overview* 仪表板中：

- **visual_web_terminal_sessions_total** 在 hub 集群上收集。
- **acm_managed_cluster_info** 在每个受管集群上收集并发送到 hub 集群。

从 OpenShift Container Platform 文档中了解使用遥测来收集并发送哪些指标类型。如需更多信息，请参阅 Telemetry 收集的信息。

1.1.3. Observability pod 容量请求

Observability 组件需要 2701mCPU 和 11972Mi 内存来安装可观察性服务。下表是启用了 **observability-addons** 的五个受管集群的 pod 容量请求列表：

表 1.1. Observability pod 容量请求

Deployment 或 StatefulSet	容器名称	CPU（mCPU）	内存（Mi）	Replicas	Pod 总计 CPU	Pod 内存总量
observability-alertmanager	alertmanager	4	200	3	12	600
	config-reloader	4	25	3	12	75
	alertmanager-proxy	1	20	3	3	60
observability-grafana	grafana	4	100	2	8	200
	grafana-dashboard-loader	4	50	2	8	100
observability-observatorium-api	observatorium-api	20	128	2	40	256
observability-observatorium-operator	observatorium-operator	100	100	1	10	50
observability-rbac-query-proxy	rbac-query-proxy	20	100	2	40	200
	oauth-proxy	1	20	2	2	40
observability-thanos-compact	thanos-compact	100	512	1	100	512
observability-thanos-query	thanos-query	300	1024	2	600	2048

Deployment 或 StatefulSet	容器名称	CPU（mCPU）	内存（Mi）	Replicas	Pod 总计 CPU	Pod 内存总量
observability-thanos-query-frontend	thanos-query-frontend	100	256	2	200	512
observability-thanos-query-frontend-memcached	memcached	45	128	3	135	384
	exporter	5	50	3	15	150
observability-thanos-receive-controller	thanos-receive-controller	4	32	1	4	32
observability-thanos-receive-default	thanos-receive	300	512	3	900	1536
observability-thanos-rule	thanos-rule	50	512	3	150	1536
	configmap-reloader	4	25	3	12	75
observability-thanos-store-memcached	memcached	45	128	3	135	384
	exporter	5	50	3	15	150
observability-thanos-store-shard	thanos-store	100	1024	3	300	3072

1.1.4. Observability 服务中使用的持久性存储

安装 Red Hat Advanced Cluster Management 时，必须创建以下持久性卷(PV)，以便 PVC 可以自动附加到 PVC。提醒，当没有指定默认存储类或想要使用非默认存储类来托管 PV 时，您必须在 **MultiClusterObservability** CR 中定义存储类。建议您使用 Block Storage，类似于 Prometheus 使用的内容。另外，**alertmanager**、**thanos-compactor**、**thanos-ruler**、**thanos-receive-default** 和 **thanos-store-shard** 的每个副本必须具有自己的 PV。查看下表：

表 1.2. 持久性卷表列表

持久性卷名称	用途

alertmanager	Alertmanager 将 **nflog** 数据和静默的警报信息存储在它的存储中。**nflog** 是一个只能附加的日志，它包括当前有效的和已解决的日志，以及通知的接收者和用来识别内容的哈希摘要数据。
thanos-compact	紧凑器需要本地磁盘空间来存储用于处理的中间数据，以及 bucket 状态缓存。所需空间取决于基础块的大小。紧凑器必须有足够的空间下载所有源块，然后在磁盘上构建紧凑块。磁盘上的数据可以安全地在重新启动之间删除，并且应该是首次尝试使崩溃循环解压器。不过，建议为紧凑器永久磁盘提供压缩器持久磁盘，以便在重启期间高效地使用存储桶状态缓存。
thanos-rule	thanos 标尺通过以固定间隔发出查询来评估 Prometheus 记录和警报规则。规则结果以 Prometheus 2.0 存储格式写回磁盘。在这个有状态集合中保留的数据量在 API 版本 **observability.open-cluster-management.io/v1beta1** 中被修复。它在 **observability.open-cluster-management.io/v1beta2**: ***RetentionInLocal*** 中以一个 API 参数的形式公开
thanos-receive-default	Thanos 接收器接受传入数据（Prometheus 远程写入请求），并将这些数据写入 Prometheus TSDB 的一个本地实例。TSDB 块定期（每 2 小时）上传到对象存储，以进行长期存储和压缩。在这个有状态集合中保留的小时数或天数。有状态集合作为一个本地的缓存，它在 API 版本 **observability.open-cluster-management.io/v1beta** 中被修复。它在 **observability.open-cluster-management.io/v1beta2**: ***RetentionInLocal*** 中以一个 API 参数的形式公开
thanos-store-shard	它主要充当一个 API 网关，因此不需要大量的本地磁盘空间。它在启动时加入 Thanos 集群，并公告它可以访问的数据。它在本地磁盘上保留少量的、与所有远程块相关的信息，并与存储桶保持同步。这些数据通常在重启时会被安全地删除，这会增加启动时间。

注意 ：时间序列历史数据存储在对象存储中。Thanos 使用对象存储作为指标和与其相关的元数据的主存储。有关对象存储和降级的详情，请参阅启用可观察性服务

1.1.5. 支持

Red Hat Advanced Cluster Management 使用 Red Hat OpenShift Container Storage 测试并被完全支持。

Red Hat Advanced Cluster Management 支持在用户提供的兼容 S3 API 的第三方对象存储中多集群可观察 Operator 的功能。

Red Hat Advanced Cluster Management 使用商业、合理的努力来帮助识别根本原因。

如果创建了相关的支持问题，且确定问题的根本原因是客户提供的兼容 S3 对象存储，则需要通过客户支持频道解决问题。

Red Hat Advanced Cluster Management 不会承诺修复客户提供的、问题的根本原因是 S3 兼容对象存储供应商的支持问题单。

1.2. 启用 OBSERVABILITY 服务

监控使用 observability 服务（**multicluster-observability-operator**）的受管集群的监控状态。

需要的访问权限： 集群管理员或 **open-cluster-management:cluster-manager-admin** 角色。

- 先决条件
- 启用可观察性
- 创建 MultiClusterObservability CR
- 从 Red Hat OpenShift Container Platform 控制台启用可观察性
- 使用外部指标查询
- 禁用可观察性

1.2.1. 先决条件

- 您必须安装 Red Hat Advanced Cluster Management for Kubernetes。如需更多信息，请参阅在线安装。
- 您必须配置对象存储来创建存储解决方案。Red Hat Advanced Cluster Management 支持带有稳定对象存储的以下云供应商：
  - Amazon Web Services S3 (AWS S3)
  - Red Hat Ceph (S3 compatible API)
  - Google Cloud Storage
  - Azure 存储
  - Red Hat OpenShift Container Storage
  - Red Hat OpenShift on IBM (ROKS)
    重要：当您配置对象存储时，请确保在敏感数据持久化时满足加密要求。有关 Thanos 支持的对象存储的更多信息，请参阅 Thanos 文档。

1.2.2. 启用可观察性

通过创建一个 **MultiClusterObservability** 自定义资源（CR）实例来启用可观察性服务。在启用可观察性前，请参阅 Observability pod 容量请求 以了解更多信息。完成以下步骤以启用可观察服务：

1. 登录到您的 Red Hat Advanced Cluster Management hub 集群。
2. 使用以下命令，为可观察服务创建一个命名空间：

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

3. 生成 pull-secret。如果在 **open-cluster-management** 命名空间中安装了 Red Hat Advanced Cluster Management，请运行以下命令：

```
DOCKER_CONFIG_JSON=`oc extract secret/multiclusterhub-operator-pull-secret -n open-cluster-management --to=-`
```

如果命名空间中没有定义 **multiclusterhub-operator-pull-secret**，将 **openshift-config** 命名空间中的 **pull-secret** 复制到 **open-cluster-management-observability** 命名空间中。运行以下命令：

```
DOCKER_CONFIG_JSON=`oc extract secret/pull-secret -n openshift-config --to=-`
```

然后，在 **open-cluster-management-observability** 命名空间中创建 pull-secret，运行以下命令：

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

4. 为您的云供应商的对象存储创建 secret。您的 secret 必须包含存储解决方案的凭证。例如，运行以下命令：

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

查看以下受支持对象存储的 secret 示例：

- 对于 Red Hat Advanced Cluster Management，您的 secret 可能类似以下文件：

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

- 对于 Amazon S3 或 S3 兼容，您的 secret 可能类似以下文件：

```
apiVersion: v1
kind: Secret
metadata:
  name: thanos-object-storage
  namespace: open-cluster-management-observability
type: Opaque
stringData:
```

```
  thanos.yaml: |
    type: s3
    config:
      bucket: YOUR_S3_BUCKET
      endpoint: YOUR_S3_ENDPOINT
      insecure: true
      access_key: YOUR_ACCESS_KEY
      secret_key: YOUR_SECRET_KEY
```

详情请参阅 Amazon Simple Storage Service 用户指南 。

- 对于 Google，您的 secret 可能类似以下文件：

```
apiVersion: v1
kind: Secret
metadata:
  name: thanos-object-storage
  namespace: open-cluster-management-observability
type: Opaque
stringData:
  thanos.yaml: |
    type: GCS
    config:
      bucket: YOUR_GCS_BUCKET
      service_account: YOUR_SERVICE_ACCOUNT
```

如需了解更多详细信息，请参阅 Google Cloud Storage。

- 对于 Azure，您的 secret 可能类似以下文件：

```
apiVersion: v1
kind: Secret
metadata:
  name: thanos-object-storage
  namespace: open-cluster-management-observability
type: Opaque
stringData:
  thanos.yaml: |
    type: AZURE
    config:
      storage_account: YOUR_STORAGE_ACCT
      storage_account_key: YOUR_STORAGE_KEY
      container: YOUR_CONTAINER
      endpoint: blob.core.windows.net
      max_retries: 0
```

如需了解更多详细信息，请参阅 Azure Storage 文档。

注 ： 如果您将 Azure 用作 Red Hat OpenShift Container Platform 集群的对象存储，则不支持与集群关联的存储帐户。您必须创建新存储帐户。

- 对于 Red Hat OpenShift Container Storage，您的 secret 可能类似以下文件：

```
apiVersion: v1
kind: Secret
metadata:
```

```
  name: thanos-object-storage
  namespace: open-cluster-management-observability
type: Opaque
stringData:
  thanos.yaml: |
    type: s3
    config:
      bucket: YOUR_OCS_BUCKET
      endpoint: YOUR_OCS_ENDPOINT
      insecure: false
      access_key: YOUR_OSC_ACCESS_KEY
      secret_key: YOUR_OSC_SECRET_KEY
```

如需了解更多详细信息，请参阅 Red Hat OpenShift Container Storage 。

- 对于 IBM 上的 Red Hat OpenShift (ROKS)，您的 secret 可能类似以下文件：

```
apiVersion: v1
kind: Secret
metadata:
  name: thanos-object-storage
  namespace: open-cluster-management-observability
type: Opaque
stringData:
  thanos.yaml: |
    type: s3
    config:
      bucket: YOUR_ROKS_S3_BUCKET
      endpoint: YOUR_ROKS_S3_ENDPOINT
      insecure: true
      access_key: YOUR_ROKS_ACCESS_KEY
      secret_key: YOUR_ROKS_SECRET_KEY
```

如需了解更多详细信息，请参阅 IBM 云文档 Cloud Object Storage。务必使用服务凭据来连接对象存储。如需了解更多详细信息，请参阅 IBM Cloud 文档，云对象存储和服务凭证。

5. 您可以使用以下命令为云供应商检索 S3 access key 和 secret 密钥：

```
YOUR_CLOUD_PROVIDER_ACCESS_KEY=$(oc -n open-cluster-management-observability get secret <object-storage-secret> -o jsonpath="{.data.thanos\.yaml}" | base64 --decode | grep access_key | awk '{print $2}')

echo $ACCESS_KEY

YOUR_CLOUD_PROVIDER_SECRET_KEY=$(oc -n open-cluster-management-observability get secret <object-storage-secret> -o jsonpath="{.data.thanos\.yaml}" | base64 --decode | grep secret_key | awk '{print $2}')

echo $SECRET_KEY
```

您必须在 secret 中解码、编辑并编码您的 **base64** 字符串。

1.2.2.1. 创建 MultiClusterObservability CR

完成以下步骤，为您的受管集群创建 **MultiClusterObservability** 自定义资源 (CR)：

1. 创建名为 ***multiclusterobservability_cr.yaml*** 的 **MultiClusterObservability** 自定义资源 YAML 文件。
   查看以下默认 YAML 文件以查看可观察性：

```
apiVersion: observability.open-cluster-management.io/v1beta2
kind: MultiClusterObservability
metadata:
  name: observability
spec:
  observabilityAddonSpec: {}
  storageConfig:
    metricObjectStorage:
      name: thanos-object-storage
      key: thanos.yaml
```

   您可能需要修改 **advanced** 部分中的 **retentionConfig** 参数的值。如需更多信息，请参阅 Thanos Downsampling 分辨率和保留时间。根据受管集群的数量，您可能需要为有状态的集合更新存储量。如需更多信息，请参阅 Observability API。

2. 要在基础架构机器集上部署，您必须通过更新 **MultiClusterObservability** YAML 中的 ***nodeSelector*** 来为设置设置一个标签。您的 YAML 可能类似以下内容：

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

   如需更多信息，请参阅创建基础架构机器集。

3. 运行以下命令，将可观察 YAML 应用到集群：

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

   用于 Thanos、Grafana 和 AlertManager 的所有 pod 在 **open-cluster-management-observability** 命名空间中创建。所有连接到 Red Hat Advanced Cluster Management hub 集群的受管集群都会被启用，以将指标数据发送回 Red Hat Advanced Cluster Management Observability 服务。

4. 要验证是否已启用了 observabilty 服务，启动 Grafana 仪表板，查看其中是否包括了相关的数据。完成以下步骤：

   a. 登录到 Red Hat Advanced Cluster Management 控制台。

   b. 在导航菜单中选择 **Home** > **Overview**。

   c. 点击位于控制台标头旁的 Grafana 链接，从您的受管集群中查看指标。
      注：如果要排除特定的受管集群收集可观察性数据,请在集群中添加以下集群标签：**observability: disabled**。

observability 服务被启用。启用 observability 服务后，会启动以下功能：

- 所有来自受管集群的警报管理器都转发到 Red Hat Advanced Cluster Management hub 集群。
- 所有连接到 Red Hat Advanced Cluster Management hub 集群的受管集群都会被启用，以将警报发送回 Red Hat Advanced Cluster Management observability 服务。您可以配置 Red Hat Advanced Cluster Management Alertmanager 来处理重复数据删除、分组和将警报路由到正确的接收器集成，如电子邮件、PagerDuty 或 OpsGenie。您还可以处理静默和禁止警报。
  注 ： 只有 Red Hat OpenShift Container Platform 版本 4.8 或更高版本的受管集群支持将警报转

发到 Red Hat Advanced Cluster Management hub 集群功能。在安装启用了可观察性服务的 Red Hat Advanced Cluster Management 后，来自 OpenShift Container Platform v4.8 及更新的版本的警报会自动转发到 hub 集群。

请参阅转发警报以了解更多信息。

1.2.3. 从 Red Hat OpenShift Container Platform 控制台启用可观察性

1. 登录您的 Red Hat OpenShift Container Platform 集群。
2. 在导航菜单中选择 **Home** > **Projects**。
3. 点 **Create Project** 按钮。您必须为项目名称输入 *open-cluster-management-observability*。
4. 点击 **Create**。
5. 创建镜像 pull-secret：
   a. 在 **open-cluster-management-observability** 项目中创建名为 **multiclusterhub-operator-pull-secret** 的镜像 pull-secret。在 OpenShift Container Platform 控制台导航菜单中选择 **Workloads** > **Secrets**。
   b. 选择 **Create** 按钮 > **Image Pull Secret**。
   c. 完成 *Create Image Pul Secret* 表单，然后点 **Create**。
6. 在 **open-cluster-management-observability** 项目中创建名为 **thanos-object-storage** 的对象存储 secret, 。在本例中，为可观察性服务创建一个 Amazon S3 对象存储 secret：
   a. 在 OpenShift Container Platform 导航菜单中点 **Workloads** > **Secrets**。
   b. 点 **Create** 按钮 > **From YAML**。
   c. 输入您的对象存储 secret 详细信息，然后单击 **Create**。
   注 ： 请参阅 Enabling observability 部分的第 4 步来查看 secret 的示例。
7. 创建 **MultiClusterObservability** CR：
   a. 在 OpenShift Container Platform 导航菜单中选择 **Home** > **Explore**。
   b. 通过查询 *MultiClusterObservability* 来搜索 **MultiClusterObservability** API 资源。
   c. 选择带有 **v1beta2** 版本的 **MultiClusterObservability** 来查看资源详情。
   d. 选择 *Instances* 选项卡，然后点 **Create MultiClusterObservability** 按钮。
   e. 输入您的 **MultiClusterObservability** 实例详情，然后点 **Create**
   f. 查看 *Conditions* 部分，检查 **MultiClusterObservability** 实例的状态。当您收到以下消息时，代表 obseravbility 服务已被成功启用：**Observability components are deployed and running**

从 OpenShift Container Platform 控制台启用可观察性。

1.2.3.1. 使用外部指标查询

Observability 提供了一个外部 API，用于通过 OpenShift 路由（**rbac-query-proxy**）查询指标。查看以下使用 **userbac-query-proxy** 路由的任务：

- 您可以使用以下命令获取路由的详情：

  ```
  oc get route rbac-query-proxy -n open-cluster-management-observability
  ```

- 若要访问 **therbac-query-proxy** 路由，您必须具有 OpenShift OAuth 访问令牌。该令牌应当与用户或服务帐户关联，该帐户有权获取命名空间。如需更多信息，请参阅管理用户拥有的 OAuth 访问令牌。

- 获取默认 CA 证书，并将 key **tls.crt 的内容** 存储在本地文件中。运行以下命令：

  ```
  oc -n openshift-ingress get secret router-certs-default -o jsonpath="{.data.tls\.crt}" | base64 -d > ca.crt
  ```

- 运行以下命令以查询指标：

  ```
  curl --cacert ./ca.crt -H "Authorization: Bearer {TOKEN}" https://{PROXY_ROUTE_URL}/api/v1/query?query={QUERY_EXPRESSION}
  ```

  注：QUERY_EXPRESSION 是标准的 Prometheus 查询表达式。例如，通过替换上述命令中的 URL 来查询指标 **cluster_infrastructure_provider**，使用以下 URL：**https://{PROXY_ROUTE_URL}/api/v1/query?query=cluster_infrastructure_provider** 替换前面提到的命令中的 URL。如需了解更多详细信息，请参阅查询 prometheus。

- 您还可以替换 **therbac-query-proxy** 路由的证书：

  - 请参阅 OpenSSL 命令以生成证书以创建证书。当您自定义 **csr.cnf** 时，将 **DNS.1** 更新为 **therbac-query-proxy** 路由的主机名。

  - 运行以下命令，使用生成的证书创建 **proxy-byo-ca** 和 **proxy-byo-cert secret**：

    ```
    oc -n open-cluster-management-observability create secret tls proxy-byo-ca --cert ./ca.crt --key ./ca.key

    oc -n open-cluster-management-observability create secret tls proxy-byo-cert --cert ./ingress.crt --key ./ingress.key
    ```

1.2.4. 禁用可观察性

要禁用可观察性服务，请卸载 **observability** 资源。请参阅使用命令删除 MultiClusterHub 实例的第 1 步。

要了解更多有关如何定制可观察性的信息，请参阅定制可观察性。

1.3. 定制可观察性

以下部分介绍了对可观察性服务所收集的数据进行自定义、管理和查看的信息。

使用 **must-gather** 命令收集有关为可观察性资源创建的新信息的日志。如需更多信息，请参阅故障排除文档中的 *Must-gather* 部分。

- 创建自定义规则

- 配置 AlertManager
- 添加自定义指标
- 删除默认指标
- 添加*高级*配置
- 从控制台更新 *multiclusterobservability* CR 副本
- 转发警报
- 查看和查找数据
  - 查看 etcd 表
- 禁用可观察性

1.3.1. 创建自定义规则

您可以通过在可观察性资源中添加 Prometheus 记录规则和 警报规则，为可观察性安装创建自定义规则。如需更多信息，请参阅 Prometheus 配置。

- 记录规则可让您根据需要预先计算或计算昂贵的表达式。结果保存为一组新的时间序列。
- 通过警报规则，您可以根据如何将警报发送到外部服务来指定警报条件。

使用 Prometheus 定义自定义规则来创建警报条件，并将通知发送到外部消息服务。**注**：当您更新自定义规则时，**observability-thanos-rule** pod 会自动重启。

完成以下步骤以创建自定义规则：

1. 登录到您的 Red Hat Advanced Cluster Management hub 集群。
2. 在 **open-cluster-management-observability** 命名空间中创建一个名为 **thanos-ruler-custom-rules** 的 ConfigMap。键必须被命名为 **custom_rules.yaml**，如下例所示。您可以在配置中创建多个规则：
   - 默认情况下，开箱即用的警报规则在 **open-cluster-management-observability** 命名空间中的 **thanos-ruler-default-rules** ConfigMap 中定义。
     例如，您可以创建一个自定义警报规则，在 CPU 使用量超过了您定义的值时通知您。您的 YAML 可能类似以下内容：

```
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
```

- 您还可以在 **thanos-ruler-custom-rules** ConfigMap 中创建自定义记录规则。例如，您可以创建一个记录规则，让您可以获取 pod 的容器内存缓存的总和。您的 YAML 可能类似以下内容：

```
data:
  custom_rules.yaml: |
    groups:
      - name: container-memory
        rules:
        - record: pod:container_memory_cache:sum
          expr: sum(container_memory_cache{pod!=""}) BY (pod, container)
```

注：如果这是第一个新的自定义规则，它会被立即创建。对于 ConfigMap 的更改，会自动重新加载配置。由于 **observability-thanos-ruler** sidecar 中的 **config-reload**，所以会重新加载配置。

3. 如果要验证警报规则是否正常工作，请完成以下步骤：

   a. 访问 Grafana 仪表板，然后选择 **Explore** 图标。

   b. 在 *Metrics* 探索栏中，键入 **ALERTS** 并运行查询。显示系统中所有处于待处理或启动状态的 **ALERTS**。

   c. 如果没有显示警报，查看规则来检查表达式是否正确。

已创建一个自定义规则。

1.3.2. 配置 AlertManager

集成外部消息工具，如 email、Slack 和 PagerDuty 以接收来自 AlertManager 的通知。您必须覆盖 **open-cluster-management-observability** 命名空间中的 **alertmanager-config** secret 来添加集成，并为 AlertManager 配置路由。完成以下步骤以更新自定义接收器规则：

1. 从 **alertmanager-config** secret 中提取数据。运行以下命令：

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

2. 运行以下命令，编辑并保存 **alertmanager.yaml** 文件配置：

```
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=-
```

更新的 secret 可能与以下类似：

```
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
```

您的更改会在修改后立即生效。如需 AlertManager 的示例，请参阅 prometheus/alertmanager。

1.3.3. 添加自定义指标

将指标添加到 **metrics_list.yaml** 文件中，用来从受管集群中收集数据。

完成以下步骤以添加自定义指标：

1. 登录到您的集群。
2. 验证 **mco observability** 已启用。在 **status.conditions.message** 中检查以下消息：**Observability components are deployed and running**运行以下命令：

   ```
   oc get mco observability -o yaml
   ```

3. 创建名为 **observability-metrics-custom-allowlist.yaml** 的文件，其中包含以下内容：将自定义指标的名称和记录规则添加到 **metrics_list.yaml** 参数。例如，从受管集群收集 **node_memory_MemTotal_bytes** 和**apiserver_request_duration_seconds:histogram_quantile_90**。ConfigMap 的 YAML 可能类似以下内容：

   ```
   kind: ConfigMap
   apiVersion: v1
   metadata:
     name: observability-metrics-custom-allowlist
   data:
     metrics_list.yaml: |
       names:
         - node_memory_MemTotal_bytes
       rules:
       - record: apiserver_request_duration_seconds:histogram_quantile_90
         expr:
   histogram_quantile(0.90,sum(rate(apiserver_request_duration_seconds_bucket{job=\"apiserver
   \",
           verb!=\"WATCH\"}[5m])) by (verb,le))
   ```

   - 在 **names** 部分中，添加要从受管集群收集的自定义指标的名称。
   - 在 **rules** 部分中，仅为 **expr** 和 **record** 参数对输入一个值来定义查询表达式。指标作为来自受管集群的 **record** 参数中定义的名称来收集。返回的指标值是运行查询表达式后的结果。

- **name** 和 **rules** 部分是可选的。您可以使用其中一个或两个部分。

4. 运行以下命令，在 **open-cluster-management-observability** 命名空间内创建 **observability-metrics-custom-allowlist** ConfigMap：

```
oc apply -n open-cluster-management-observability -f observability-metrics-custom-allowlist.yaml
```

5. 通过在 Grafana 仪表板上查看指标，验证您的自定义指标是否从受管集群收集数据。在 hub 集群中，选择 **Grafana dashboard** 链接。

6. 在 Grafana 搜索栏中输入您要查看的指标。收集自定义指标数据。

7. 如果在 Grafana 仪表板中使用更新的指标，请参阅设计 Grafana 仪表板以更新仪表板。

1.3.4. 删除默认指标

如果不收集特定指标的数据，您可以从 **observability-metrics-custom-allowlist.yaml** 文件中删除相应的指标。删除指标时，您也会删除指标。您可以在 metrics 名称的开头使用连字符 **-** 将默认指标名称添加到 **metrics_list.yaml** 参数中。

完成以下步骤以删除默认指标：

1. 登录到您的集群。

2. 验证 **mco observability** 已启用。在 **status.conditions.message** 中检查以下消息：**Observability components are deployed and running**运行以下命令：

```
oc get mco observability -o yaml
```

3. 在 **observability-metrics-custom-allowlist.yaml** 文件中，将默认的指标名称添加到 **metrics_list.yaml** 参数，并在指标名称前带有一个连字符 **-**。例如，将 **-cluster_infrastructure_provider** 添加到指标列表中。ConfigMap 的 YAML 可能类似以下内容：

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: observability-metrics-custom-allowlist
data:
  metrics_list.yaml: |
    names:
      - node_memory_MemTotal_bytes
      - -cluster_infrastructure_provider
```

4. 运行以下命令，在 **open-cluster-management-observability** 命名空间内创建 **observability-metrics-custom-allowlist** ConfigMap：

```
oc apply -n open-cluster-management-observability -f observability-metrics-custom-allowlist.yaml
```

5. 通过在 Grafana 仪表板上查看指标，验证您的默认指标是否没有从受管集群收集。在 hub 集群中，选择 **Grafana dashboard** 链接。

6. 在 Grafana 搜索栏中输入您要检查的指标。不再收集来自默认指标的数据。

7. 如果在 Grafana 仪表板中使用更新的指标，您可以从 ConfigMap 中删除指标。请参阅使用 ConfigMap 设计 Grafana 仪表板以更新仪表板。

1.3.5. 添加*高级*配置

您可以添加 **advanced** 配置部分来更新每个可观察性组件的保留。完成以下步骤：

1. 登录到您的集群。
2. 编辑 **mco observability**。运行以下命令：

```
oc edit mco observability -o yaml
```

3. 将 **advanced** 配置添加到 **mco observability** YAML。您的 YAML 文件可能类似以下内容：

```
spec:
  advanced:
    retentionConfig:
      blockDuration: 2h
      deleteDelay: 48h
      retentionInLocal: 24h
      retentionResolutionRaw: 30d
      retentionResolution5m: 180d
      retentionResolution1h: 0d
    receive:
      resources:
        limits:
          memory: 4096Gi
      replicas: 3
```

有关可添加到**高级**配置中的所有参数的描述，请参阅 Observability API。

1.3.6. 从控制台更新 *multiclusterobservability* CR 副本

如果工作负载增加，增加可观察 pod 的副本数。完成以下步骤以更新副本：

1. 登录您的 Red Hat Advanced Cluster Management 集群。
2. 在控制台标头中点击 *Applications* 按钮 > **OpenShift Container Platform**。
3. 在 OpenShift Container Platform 导航菜单中选择 **Administration** > **CustomerResourceDefinitions**。
4. 搜索 **multiclusterobservability**。
5. 从*实例*选项卡中，选择 **observability** 实例。
6. 编辑 YAML 选项卡中的 *YAML* 文件。更新的 YAML 可能类似以下内容：

```
spec:
  advanced:
    receive:
      replicas: 6
```

这意味着环境中有六个接收器。

有关 **mco observability** CR 中的参数的更多信息，请参阅 Observability API。

1.3.7. 转发警报

启用可观察性后，来自 OpenShift Container Platform 受管集群的警报会自动发送到 hub 集群。您可以使用 **alertmanager-config** YAML 文件，为警报配置外部通知系统。完成以下步骤以访问 **alertmanager-config** YAML 文件：

1. 以管理员身份登录到您的 Red Hat Advanced Cluster Management hub 集群。
2. 在导航菜单中，选择 **Infrastructure** > **Clusters** 来查看受管集群。
3. 选择您要查看的受管集群。
4. 在 *Details* 选项卡中，选择 OpenShift Container Platform *Console URL* 的链接。
5. 在 OpenShift Container Platform 菜单导航中选择 **Secrets**。选择 **alertmanager-config** secret 以查看 YAML 文件。
   注 ：如果您更改 **alertmanager-config** secret，评估间隔大约为一分钟。
6. 查看 **alertmanager-config** YAML 文件示例：

```
global:
  slack_api_url: '<slack_webhook_url>'

route:
  receiver: 'slack-notifications'
  group_by: [alertname, datacenter, app]

receivers:
- name: 'slack-notifications'
  slack_configs:
  - channel: '#alerts'
    text: 'https://internal.myorg.net/wiki/alerts/{{ .GroupLabels.app }}/{{ .GroupLabels.alertname
}}'
```

7. 如果要为警报转发配置代理，请将以下 **全局** 条目添加到 **alertmanager-config** YAML 文件中：

```
global:
  slack_api_url: '<slack_webhook_url>'
  http_config:
    proxy_url: http://****
```

如需更多信息，请参阅 Prometheus Alertmanager 文档。

1.3.8. 查看和查找数据

通过访问 Grafana 来查看来自受管集群的数据。完成以下步骤，从控制台查看 Grafana 仪表板：

1. 登录到您的 Red Hat Advanced Cluster Management hub 集群。
2. 在导航菜单中点 **Infrastructure** > **Clusters**。
3. 点 Grafana 链接访问 **Grafana** 仪表板。

4. 通过在 Grafana 导航菜单中选择 **Explore** 图标来访问 Prometheus 指标浏览器。

5. 要查询来自单一节点集群的指标，请在查询表达式中添加以下标签：**{clusterType="SNO"}**。例如，对于来自单一节点集群中的 *cluster_infrastructure_provider*，使用以下查询表达式：**cluster_infrastructure_provider{clusterType="SNO"}**
注 ： 如果在单一节点受管集群中启用了可观察性，则不要设置 **ObservabilitySpec.resources.CPU.limits** 参数。当您设置 CPU 限制时，observability pod 会计算为受管集群的容量。如需更多信息，请参阅管理工作负载分区。

1.3.8.1. 查看 etcd 表

您可以通过完成以下步骤，从 Grafana 中的 hub 集群仪表板查看 etcd 表：

1. 登录到您的 Red Hat Advanced Cluster Management hub 集群。

2. 在导航菜单中选择 **Overview**。点 **Grafana** 链接。

3. 查看 hub 集群仪表板中的 *etcd 表，以查看受管集群的领导选举更改*。

4. 选择特定的集群来查看更多详情。

1.3.9. 禁用可观察性

您可以禁用可观察性，在 Red Hat Advanced Cluster Management hub 集群中停止数据收集。

1.3.9.1. 在所有集群中禁用可观察性

通过删除所有受管集群中的可观察性组件来禁用可观察性。

通过将 **enableMetrics** 设置为 **false** 来更新 **multicluster-observability-operator** 资源。更新的资源可能类似如下：

```
spec:
  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
```

1.3.9.2. 在单个集群中禁用可观察性

通过完成以下步骤之一禁用特定受管集群上的可观察性：

- 在自定义资源 **managedclusters.cluster.open-cluster-management.io** 中添加 **observability: disabled** 标签。

- 在 Red Hat Advanced Cluster Management 控制台的 *Clusters* 页面中，完成以下步骤来添加 **observability: disabled** 标签：

  1. 在 Red Hat Advanced Cluster Management 控制台导航中，选择 **Infrastructure** > **Clusters**。

  2. 选择您要禁用发送到可观察性的数据收集的集群名称。

  3. 选择 **Labels**。

4. 通过添加以下标签来创建禁用可观察性集合的标签：

```
observability=disabled
```

5. 选择 **Add** 添加该标签。

6. 选择 **Done** 以关闭标签列表。

备注：当一个带有可观察性组件的受管集群被分离时，**metric-collector** 部署会被删除。

有关使用可观察性服务监控控制台数据的更多信息，请参阅观察环境介绍。

1.4. 设计您的 GRAFANA 仪表板

您可以通过创建一个 **grafana-dev** 实例来设计 Grafana 仪表板。

- 设置 Grafana 开发人员实例
- 设计您的 Grafana 仪表板
- 卸载 Grafana 开发者实例

1.4.1. 设置 Grafana 开发人员实例

首先，克隆 **stolostron/multicluster-observability-operator/** 存储库，以便您可以运行 **tools** 文件夹中的脚本。完成以下步骤以设置 Grafana 开发人员实例：

1. 运行 **setup-grafana-dev.sh** 来设置 Grafana 实例。运行脚本时，会创建以下资源：**secret/grafana-dev-config**、**deployment.apps/grafana-dev**、**service/grafana-dev**、**ingress.extensions/grafana-dev**、**persistentvolumeclaim/grafana-dev**：

```
./setup-grafana-dev.sh --deploy
secret/grafana-dev-config created
deployment.apps/grafana-dev created
service/grafana-dev created
ingress.extensions/grafana-dev created
persistentvolumeclaim/grafana-dev created
```

2. 使用 **switch-to-grafana-admin.sh** 脚本将用户角色切换到 Grafana 管理员。

   a. 选择 Grafana URL **https://$ACM_URL/grafana-dev/** 并登录。

   b. 然后，运行以下命令来添加切换的用户作为 Grafana 管理员。例如，在使用 **kubeadmin** 登录后，运行以下命令：

```
./switch-to-grafana-admin.sh kube:admin
User <kube:admin> switched to be grafana admin
```

Grafana 开发人员实例已设置。

1.4.2. 设计您的 Grafana 仪表板

设置 Grafana 实例后，您可以设计仪表板。完成以下步骤以刷新 Grafana 控制台并设计您的仪表板：

1. 在 Grafana 控制台中，通过在导航面板中选择 **Create** 图标来创建仪表板。选择 **Dashboard**，然后单击 **Add new panel**。
2. 在 *New Dashboard/Edit Panel* 视图中导航到 *Query* 选项卡。
3. 从数据源选择器中选择 **Observatorium** 并输入 PromQL 查询来配置查询。
4. 在 Grafana 仪表板标头中点击仪表板标头中的 **Save** 图标。
5. 添加一个描述性名称并点 **Save**。

1.4.2.1. 使用 ConfigMap 设计 Grafana 仪表板

完成以下步骤，使用 ConfigMap 设计 Grafana 仪表板：

1. 您可以使用 **generate-dashboard-configmap-yaml.sh** 脚本在本地生成仪表板 ConfigMap，并在本地保存 ConfigMap:

```
./generate-dashboard-configmap-yaml.sh "Your Dashboard Name"
Save dashboard <your-dashboard-name> to ./your-dashboard-name.yaml
```

   如果您没有运行上述脚本的权限，请完成以下步骤：

   a. 选择一个仪表板并点 **Dashboard settings** 图标。
   b. 在导航框中，点 **JSON Model** 图标。
   c. 复制仪表板 JSON 数据，并将它粘贴到 ***data*** 部分。
   d. 修改 ***name*** 并替换 ***$your-dashboard-name***。ConfigMap 可能类似以下文件：

```
kind: ConfigMap
apiVersion: v1
metadata:
  name: $your-dashboard-name
  namespace: open-cluster-management-observability
  labels:
    grafana-custom-dashboard: "true"
data:
  $your-dashboard-name.json: |-
    $your_dashboard_json
```

   注： 如果您的仪表板不在 *General* 文件夹中，您可以在此 ConfigMap 的 **annotations** 部分中指定文件夹名称：

```
annotations:
  observability.open-cluster-management.io/dashboard-folder: Custom
```

   完成 ConfigMap 的更新后，您可以安装它，将仪表板导入到 Grafana 实例。

1.4.3. 卸载 Grafana 开发者实例

卸载实例时，相关资源也会被删除。运行以下命令：

```
./setup-grafana-dev.sh --clean
```

```
secret "grafana-dev-config" deleted
deployment.apps "grafana-dev" deleted
service "grafana-dev" deleted
ingress.extensions "grafana-dev" deleted
persistentvolumeclaim "grafana-dev" deleted
```

1.5. RED HAT INSIGHTS 的可观察性

Red Hat Insights 与 Red Hat Advanced Cluster Management observability 集成，并被启用来帮助识别集群中的现有或潜在问题。Red Hat Insights 可帮助您识别、确定和解决稳定性、性能、网络和安全风险。Red Hat OpenShift Container Platform 通过 OpenShift Cluster Manager 提供集群健康监控。OpenShift Cluster Manager 会以匿名的形式收集有关集群健康、使用和大小的聚合信息。如需更多信息，请参阅 Red Hat Insights 产品文档 。

当您创建或导入 OpenShift 集群时，受管集群中的匿名数据会自动发送到红帽。这些信息用于创建提供集群健康信息的智能分析工具。Red Hat Advanced Cluster Management 管理员可以使用这个健康信息根据严重性创建警报。

需要的访问权限： 集群管理员

1.5.1. 先决条件

- 确保启用了 Red Hat Insights。如需更多信息，请参阅修改全局集群 pull secret 以禁用远程健康报告。
- 安装 OpenShift Container Platform 版本 4.0 或更高版本。
- 在 OpenShift Cluster Manager 中注册的 hub 集群用户必须能够管理 OpenShift Cluster Manager 中的所有 Red Hat Advanced Cluster Management 受管集群。

1.5.2. 来自 Red Hat Advanced Cluster Management 控制台的 Red Hat Insights

继续阅读以查看集成的功能描述：

- 当您从 *Clusters* 页面选择集群时，您可以从 *Status* 卡中选择**识别出的问题的数量**。*Status* 卡显示有关 *节点*、*应用程序*、*策略违规*和 *识别出的问题*的信息。*Identified issues* 卡包括了 Red Hat insights 中的信息。*Identified issues* 状态按严重性显示问题的数量。问题严重性的分类级别如下：*Critical*、*Major*、*Low* 和 *Warning*
- 在点数量后会显示 *Potential issue* 侧面板。面板中显示了所有问题的摘要和图表信息。您还可以使用搜索功能搜索推荐的补救方法。补救选项显示漏洞的 *Description*、与漏洞关联的 *Category*，以及 *Total risk*。
- 在 *Description* 部分中，您可以选择到这个漏洞的链接。选择 *How tomediate 选项卡* 来查看解决您的漏洞的步骤。您还可以单击 *Reason* 选项卡来查看漏洞发生的原因。

如需更多信息，请参阅管理 Insights **PolicyReports**。

1.6. 管理 INSIGHTS *POLICYREPORTS*

查看以下部分以了解如何管理和查看 Insights **PolicyReports** ：

- 搜索 insight 策略报告
- 从控制台查看潜在的漏洞

1.6.1. 搜索 insight 策略报告

您可以在受管集群中搜索具有冲突的特定 Insights **PolicyReport**。完成以下步骤以搜索 insight 策略报告：

1. 登录到您的 Red Hat Advanced Cluster Management hub 集群。
2. 单击控制台标头中的*搜索*图标进入 *Search* 页。
3. 在 *Search* 页中输入以下查询： **kind:policyreport**
4. 按回车后，*Policyreport* 部分中会显示 insight 策略报告的列表。
   注 ： PolicyReport 名称与集群名称匹配。
5. 您还可以通过洞察策略违反和类别进一步指定查询。
6. 当您选择一个 **PolicyReport** 名称时，您会被重定向到关联的集群的 *Details* 页面。*Insights* 侧栏会自动显示。
7. 如果禁用了搜索服务，且您要搜索 insight 信息，在 hub 集群中运行以下命令：

```
oc get policyreport --all-namespaces
```

1.6.2. 从控制台查看发现的问题

您可以通过完成以下步骤来查看特定集群中的漏洞：

1. 登录您的 Red Hat Advanced Cluster Management 集群。
2. 在导航菜单中选择 **Overview**。您可以在 *Cluster issues* 概况卡中查看集群问题的详细信息和严重程度。
3. 选择一个严重性来查看与严重性关联的 **PolicyReports**。您会被定向到 *Search* 页面，其中显示 **PolicyReport** 详情。
4. 从 *Clusters* 页面选择要定向到 Insights 的 **PolicyReport** 名称。
5. 另外，您还可以从导航菜单中选择 **Clusters**。
6. 从表中选择一个受管集群来查看更多详细信息。
7. 在 *Status* 卡中查看确定的问题数量。
8. 选择潜在问题的数量来查看严重性图表，以及推荐的对问题进行补救的方法。
9. 点漏洞的链接来查看与漏洞相关的 *How to remediate* 和 *Reason* 信息。
10. 如果有多个要查看的漏洞，请点击 **Back** 查看潜在问题列表。
    注 ： 在解决了这个问题后，Red Hat Advanced Cluster Management 每 30 分钟会接收 Red Hat Insights，Red Hat Insights 每两小时更新一次。

如需更多信息，请参阅为 AlertManager 配置规则。