Red Hat Advanced Cluster Management for Kubernetes 2.10

多集群全局 hub

多集群全局 hub

Last Updated: 2024-07-01

Red Hat Advanced Cluster Management for Kubernetes 2.10 多集群全局 hub

多集群全局 hub

法律通告

摘要

了解更多有关多集群全局 hub 的信息，它可让您使用单个 hub 管理多个 hub 集群。

目录

第 1 章 多集群全局 HUB

multicluster global hub 是一组组件，可让您从单个 hub 集群中导入一个或多个 hub 集群。

将 hub 集群导入为受管 hub 集群后，您可以使用 multicluster global hub 在所有受管 hub 集群中完成以下任务：

- 报告策略合规状态和趋势
- 在概览页面中将所有受管 hub 和受管集群清单
- 在不规则策略行为的情况下检测和警报

当单个 hub 集群无法在高扩展环境中管理大量集群时，多集群全局 hub 很有用。当发生这种情况时，您可以将集群分成较小的集群组，并为每个组配置 hub 集群。

对于由该 hub 集群管理的受管集群，通常不太方便查看多个 hub 集群上的数据。multicluster global hub 提供了一种更简单的方法，通过将多个 hub 集群指定为受管 hub 集群来查看来自多个 hub 的信息。multicluster global hub 集群管理其他 hub 集群，并从受管集群收集总结信息。

注： Observability 对于多集群全局 hub 不可用。如果您在集群中安装 multicluster global hub 前启用了 multicluster observability 功能，请手动禁用多集群可观察性功能。

要了解如何使用多集群全局 hub，请参阅以下部分：

- 多集群全局 hub 架构
- 全局 Hub 要求
- 在连接的环境中安装多集群全局 Hub
- 在断开连接的环境中安装 Multicluster Global Hub
- 集成现有组件
- 以默认模式导入受管 hub 集群
- 访问 Grafana 数据
- Grafana 警报（技术预览）
- 配置 cron 任务
- 手动运行总结过程
- multicluster global hub 的备份（技术预览）

1.1. 多集群全局 HUB 架构

multicluster global hub 由用于访问和管理 hub 集群的以下组件组成：

- 运行管理工具和控制台的服务器组件，称为 *全局 hub 集群*
- 在 Red Hat Advanced Cluster Management 上安装的客户端组件，名为 *受管 hub*，可由全局 hub 集群管理。受管 hub 还管理其他集群。您不必为多集群全局 hub 集群使用专用集群。

在以下部分了解更多有关架构的信息：

请参阅以下高级别多集群术语和组件：

- multicluster global hub operator
- multicluster global hub manager
- 多集群全局 hub 代理
- 多集群全局 hub 视觉化

1.1.1. multicluster global hub operator

multicluster global hub operator 包含 multicluster global hub 的组件。Operator 为全局多集群管理部署所有需要的组件。组件包括 **multicluster-global-hub-manager**、**multicluster-global-hub-grafana**，以及多集群全局 hub 集群和 **multicluster-global-hub-agent** 中的 **Kafka** 和 **PostgreSQL** 版本。

Operator 还利用 **manifestwork** 自定义资源在受管集群中部署 Red Hat Advanced Cluster Management for Kubernetes operator。在受管集群中部署 Red Hat Advanced Cluster Management Operator 后，受管集群将变为标准的 Red Hat Advanced Cluster Management 集群。这个 hub 集群现在是受管 hub 集群。

1.1.2. multicluster global hub Manager

multicluster 全局 hub Manager 用于将数据持久保留在 **postgreSQL** 数据库中。数据来自 Kafka 传输。管理器还会将数据发布到 Kafka 传输，以便它可以与受管 hub 集群中的数据同步。

1.1.3. multicluster global hub 代理

multicluster global hub 代理在受管 hub 集群上运行。它同步 multicluster global hub 集群和受管 hub 集群之间的数据。例如，代理将受管集群的信息从受管 hub 集群同步到多集群全局 hub 集群，并将策略或应用程序从 multicluster global hub 集群同步到受管 hub 集群。

1.1.4. 多集群全局 hub 视觉化

Grafana 在多集群全局 hub 集群中运行的，作为多集群全局 hub 视觉化的主服务。Global Hub Manager 收集的 PostgreSQL 数据是其默认的 DataSource。通过使用名为 **multicluster-global-hub-grafana** 的路由公开服务，您可以通过访问控制台来访问多集群全局 hub Grafana 仪表板。

1.2. 全局 HUB 要求

了解安装和网络所需的内容，以及支持的组件和环境。

- 常规要求
- 网络要求
- 支持的组件

1.2.1. 常规要求

要安装全局 hub，您需要以下要求：

需要的访问权限：集群管理员

OpenShift Container Platform Dedicated 环境需要访问权限：您必须具有 **cluster-admin** 权限。默认情况下，**dedicated-admin** 角色没有在 OpenShift Container Platform Dedicated 环境中创建命名空间所需的权限。

- 安装并配置 Red Hat Advanced Cluster Management for Kubernetes。了解有关 Red Hat Advanced Cluster Management 的详情。

1.2.2. 网络要求

请参见以下网络要求：

- 受管 hub 也是 Red Hat Advanced Cluster Management 中多集群全局 hub 的受管集群。Red Hat Advanced Cluster Management 中的网络配置是必需的。请参阅 Red Hat Advanced Cluster Management 网络详情。
- 下表列出了全局 hub 网络信息：

方向	协议	连接	端口（如果指定）	源地址	目标地址	
从用户的浏览器中入站	HTTPS	用户需要访问 Grafana 仪表板	443	用户的浏览器	Grafana 路由的 IP 地址	
到 Kafka 集群的外向流量	HTTPS	全局 hub 管理器需要从 Kafka 集群获取数据	443	multicluster-global-hub-manager-xxx pod	Kafka 路由主机	
出站到 PostgreSQL 数据库	HTTPS	全局 hub 管理器需要保留数据到 PostgreSQL 数据库	443	multicluster-global-hub-manager-xxx pod	PostgreSQL 数据库的 IP 地址	

- 下表列出了 Managed hub 网络信息：

方向	协议	连接	端口（如果指定）	源地址	目标地址	
到 Kafka 集群的外向流量	HTTPS	全局 hub 代理需要将集群信息和策略信息同步到 Kafka 集群	443	multicluster-global-hub-agent pod	Kafka 路由主机	

- 请参阅产品文档中 调整 Red Hat Advanced Cluster Management 集群的大小 指南。
- **可选：**对于中间件，多集群全局 hub 具有内置 Kafka、PostgreSQL 和 Grafana，但您可以使用自己的配置的 Kafka、PostgreSQL 和 Grafana。如需了解更多详细信息，请参阅 集成现有组件。

1.2.3. 支持的组件

了解支持的平台和组件。

- 由于它们共享集成控制台，多集群全局 hub 控制台支持与 OpenShift Container Platform 相同的浏览器。如需有关支持的浏览器和 版本的信息，请参阅 Red Hat OpenShift Container Platform 文档中的访问 Web 控制台。
- 下表中显示了支持的多集群全局 hub 集群的平台：

平台	全局 hub 集群支持	支持受管 hub 集群
Red Hat Advanced Cluster Management 2.10 和更新的 2.10.x 版本	是	是
Red Hat Advanced Cluster Management 2.9 及更新的 2.9.x 版本	是	是
Red Hat Advanced Cluster Management 2.8.3 及更新的版本 2.8.x 版本	是	是
Red Hat Advanced Cluster Management on Arm	是	是
Red Hat Advanced Cluster Management on IBM Z	是	是
Red Hat Advanced Cluster Management on IBM Power Systems	是	是

- multicluster global hub 支持以下中间件：
  - Kafka 3.4 及更新的版本 3.4.x 版本。
  - PostgreSQL 版本 13 及更新的版本 13.x 版本。

1.2.4. 其他资源

- 在连接的环境中安装多集群全局 Hub
- 在断开连接的环境中安装 Multicluster Global Hub

1.3. 在连接的环境中安装多集群全局 HUB

multicluster global hub 通过 Operator Lifecycle Manager 安装，用于管理组成 Operator 的组件的安装、升级和删除。

需要的访问权限：集群管理员

1.3.1. 先决条件

- 对于 OpenShift Container Platform Dedicated 环境，必须具有 **cluster-admin** 权限来访问环境。默认情况下，**dedicated-admin** 角色没有在 OpenShift Container Platform Dedicated 环境中创建命名空间所需的权限。
- 您必须安装并配置 Red Hat Advanced Cluster Management for Kubernetes。如需了解更多详细信息，请参阅 安装和升级。
- 您必须配置 Red Hat Advanced Cluster Management 网络。受管 hub 集群也是 Red Hat Advanced Cluster Management 中多集群全局 hub 的受管集群。如需了解更多详细信息，请参阅 Hub 集群网络配置。

1.3.1.1. 使用控制台安装多集群全局 hub

要使用 OpenShift Container Platform 控制台在连接的环境中安装 multicluster global hub operator，请完成以下步骤：

1. 以具有 **cluster-admin** 角色的用户身份登录 OpenShift Container Platform 控制台。
2. 在导航菜单中选择 Operators & **gt**; **OperatorHub** 图标。
3. 找到并选择 **Multicluster global hub operator**。
4. 点 **Install** 开始安装。
5. 安装完成后，检查 *Installed Operators* 页中的状态。
6. 点 **Multicluster global hub operator** 进入 *Operator* 页面。
7. 点 **Multicluster global hub** 选项卡查看 **Multicluster Global Hub** 实例。
8. 点 **Create Multicluster Global Hub** 创建 **Multicluster Global Hub** 实例。
9. 输入所需信息并点 **Create** 创建 **Multicluster Global Hub** 实例。

备注：

- multicluster global hub 仅适用于 x86 平台。
- 安装 multicluster 全局 hub 后，Red Hat Advanced Cluster Management 中禁用了策略和应用程序。

1.3.2. 其他资源

- 如需有关镜像 Operator 目录的更多信息，请参阅 镜像 Operator 目录。
- 有关从私有 registry 访问镜像的更多信息，请参阅从私有 registry 访问 Operator 的镜像。
- 有关添加目录源的更多信息，请参阅 在集群中添加目录源。
- 有关在断开连接的环境中安装 Red Hat Advanced Cluster Management 的更多信息，请参阅 在断开连接的网络环境中安装。
- 有关镜像镜像的更多信息，请参阅 为断开连接的安装镜像镜像。

- 如需有关 Operator SDK 与 OLM 协调的更多信息，请参阅 Operator SDK 与 Operator Lifecycle Manager 集成。

1.4. 在断开连接的环境中安装多集群全局 HUB

如果集群位于受限网络中，您可以在断开连接的环境中部署 multicluster global hub operator。

需要的访问权限：集群管理员

1.4.1. 先决条件

在断开连接的环境中安装多集群全局 hub 前，您必须满足以下要求：

- 镜像 registry 和堡垒主机必须可以同时访问互联网和您的镜像 registry。
- 在集群上安装 Operator Lifecycle Manager。请参阅 Operator Lifecycle Manager (OLM)。
- 安装 Red Hat Advanced Cluster Management for Kubernetes。
- 安装以下命令行界面：
  - OpenShift Container Platform 命令行。请参阅 OpenShift Container Platform CLI 入门。
  - **opm** 命令行。请参阅安装 opm CLI。
  - **oc-mirror** 插件。请参阅使用 oc-plugin 为断开连接的安装镜像镜像。

1.4.2. 配置镜像 registry

在断开连接的环境中安装多集群全局 hub 涉及使用本地镜像 registry。此时假设您已在 OpenShift Container Platform 集群安装过程中设置了镜像 registry。

完成以下步骤，为多集群全局 hub 置备镜像 registry：

1.4.2.1. 使用 oc-mirror 插件在镜像目录中创建 operator 软件包

红帽在 Red Hat operator 目录中提供多集群全局 hub 和 AMQ Streams operator，由 **registry.redhat.io/redhat/redhat-operator-index** 索引镜像提供。当您准备此目录索引镜像的镜像时，您可以选择将镜像整个目录作为红帽提供的镜像，或者镜像只包含您要使用的 Operator 软件包的子集。

如果您要创建完整镜像目录，则不需要特别考虑安装多集群全局 hub 和 AMQ Streams 所需的所有软件包。但是，如果您要创建部分或过滤了镜像目录，您需要为其识别要包含的特定软件包，则必须在列表中包含 **multicluster-global-hub-operator-rh** 和 **amq-streams** 软件包名称。

完成以下步骤以创建 **multicluster-global-hub-operator-rh** 和 **amq-streams** 软件包的本地镜像 registry：

1. 创建 **ImageSetConfiguration** YAML 文件来配置并添加 Operator 镜像。您的 YAML 文件可能类似以下内容，使用当前版本替换 **4.x** ：

```
kind: ImageSetConfiguration
apiVersion: mirror.openshift.io/v1alpha2
storageConfig:
  registry:
    imageURL: myregistry.example.com:5000/mirror/oc-mirror-metadata
mirror:
```

```
  platform:
    channels:
    - name: stable-4.x
      type: ocp
  operators:
  - catalog: registry.redhat.io/redhat/redhat-operator-index:v4.13
    packages:
    - name: multicluster-global-hub-operator-rh
    - name: amq-streams
  additionalImages: []
  helm: {}
```

2. 使用以下命令将镜像直接设置为目标镜像 registry：

```
oc mirror --config=./imageset-config.yaml docker://myregistry.example.com:5000
```

3. 镜像在完全断开连接的环境中设置的镜像。如需了解更多详细信息，请参阅为断开连接的安装镜像镜像。

1.4.2.2. 将 registry 和目录添加到断开连接的集群中

要使您的镜像 registry 和目录在断开连接的集群中可用。完成以下步骤：

1. 禁用 Operator Hub 的默认目录源。运行以下命令以更新 **OperatorHub** 资源：

```
oc patch OperatorHub cluster --type json \
-p '[{"op": "add", "path": "/spec/disableAllDefaultSources", "value": true}]'
```

2. 通过完成镜像 Operator 目录的步骤来 镜像 Operator 目录。

3. 将镜像目录的 **CatalogSource** 资源添加到 **openshift-marketplace** 命名空间中。您的 **CatalogSource** YAML 文件可能类似以下示例：

```
apiVersion: operators.coreos.com/v1alpha1
kind: CatalogSource
metadata:
  name: my-mirror-catalog-source
  namespace: openshift-marketplace
spec:
  image: myregistry.example.com:5000/mirror/my-operator-index:v4.13
  sourceType: grpc
  secrets:
  - <global-hub-secret>
```

- **注：** 记录 **metadata.name** 字段的值。

4. 保存更新的文件。

5. 通过查询可用的 **PackageManifest** 资源，验证所需的软件包是否可以从断开连接的集群中可用。运行以下命令：

```
oc -n openshift-marketplace get packagemanifests
```

显示的列表应包含显示 **multicluster-global-hub-operator-rh** 和 **amq-streams** 软件包由镜像目录的目录源提供的条目：

1.4.3. 配置镜像 registry

要让集群从本地镜像 registry 获取 multicluster global hub operator 的容器镜像，而不是从互联网托管的 registry 中配置 **ImageContentSourcePolicy** 资源，以将镜像引用重定向到您的镜像 registry。**ImageContentSourcePolicy** 仅支持带有 **镜像摘要** 的镜像镜像。

如果您使用 **oc adm catalog mirror** 命令镜像目录，所需的镜像内容源策略配置位于由该命令创建的 **manifests-** 目录中的 **imageContentSourcePolicy.yaml** 文件中。

如果您使用 **oc-mirror** 插件来镜像目录，则 **imageContentSourcePolicy.yaml** 文件位于由 **oc-mirror-workspace/results-** 目录中创建的 **oc-mirror** -workspace/results- 目录中。

在这两种情况下，您可以使用 **oc apply** 或 oc replace 命令将策略应用到断开连接的命令中，如 **oc replace -f ./<path>/imageContentSourcePolicy.yaml**

所需的镜像内容源策略语句会根据您创建镜像 registry 的方式而有所不同，但与以下示例类似：

```
apiVersion: operator.openshift.io/v1alpha1
kind: ImageContentSourcePolicy
metadata:
  labels:
    operators.openshift.org/catalog: "true"
  name: global-hub-operator-icsp
spec:
  repositoryDigestMirrors:
  - mirrors:
    - myregistry.example.com:5000/multicluster-globalhub
    source: registry.redhat.io/multicluster-globalhub
  - mirrors:
    - myregistry.example.com:5000/openshift4
    source: registry.redhat.io/openshift4
  - mirrors:
    - myregistry.example.com:5000/redhat
    source: registry.redhat.io/redhat
```

您可以使用 **ManagedClusterImageRegistry** 为不同的受管 hub 配置不同的镜像 registry。请参阅 导入具有 ManagedClusterImageRegistry 的集群，以使用 **ManagedClusterImageRegistry** API 替换代理镜像。

通过完成上一步，会将标签和注解添加到所选 **ManagedCluster** 中。这意味着集群中的代理镜像被镜像(mirror)替换。

- label: **multicluster-global-hub.io/image-registry=<namespace.managedclusterimageregistry-name>**
- annotation: **multicluster-global-hub.io/image-registries: <image-registry-info>**

1.4.3.1. 配置镜像 pull secret

如果订阅的 Operator 引用的 Operator 或 Operand 镜像需要访问私有 registry，您可以 提供对集群中的所有命名空间或单独的目标租户命名空间的访问权限。

1.4.3.1.1. 在 OpenShift Container Platform 集群中配置多集群全局 hub 镜像 pull secret

您可以在现有 OpenShift Container Platform 集群中配置镜像 pull secret。

注： 在预先存在的集群中应用镜像 pull secret 会导致所有节点的滚动重启。

完成以下步骤以配置 pull secret：

1. 从 pull secret 导出用户名：

```
export USER=<the-registry-user>
```

2. 从 pull secret 导出密码：

```
export PASSWORD=<the-registry-password>
```

3. 复制 pull secret：

```
oc get secret/pull-secret -n openshift-config --template='{{index .data ".dockerconfigjson" | base64decode}}' > pull_secret.yaml
```

4. 使用 pull secret 登录：

```
oc registry login --registry=${REGISTRY} --auth-basic="$USER:$PASSWORD" --to=pull_secret.yaml
```

5. 指定 multicluster global hub 镜像 pull secret：

```
oc set data secret/pull-secret -n openshift-config --from-file=.dockerconfigjson=pull_secret.yaml
```

6. 删除旧的 pull secret：

```
rm pull_secret.yaml
```

1.4.3.1.2. 将 multicluster global hub 镜像 pull secret 配置为单个命名空间

您可以通过完成以下步骤，将镜像 pull secret 配置为单独的命名空间：

1. 运行以下命令，在租户命名空间中创建 secret：

```
oc create secret generic <secret_name> -n <tenant_namespace> \
--from-file=.dockerconfigjson=<path/to/registry/credentials> \
--type=kubernetes.io/dockerconfigjson
```

2. 将 secret 链接到 Operator 或操作对象的服务帐户：

```
oc secrets link <operator_sa> -n <tenant_namespace> <secret_name> --for=pull
```

1.4.3.2. 安装 Global Hub Operator

您可以使用 Red Hat OpenShift Container Platform Web 控制台从 OperatorHub 安装并订阅 Operator。详情请参阅将 Operator 添加到集群。添加 Operator 后，您可以运行以下命令来检查 multicluster global hub Operator 的状态：

```
oc get pods -n multicluster-global-hub
NAME                                         READY   STATUS    RESTARTS   AGE
multicluster-global-hub-operator-687584cb7c-fnftj   1/1     Running   0          2m12s
```

1.4.4. 其他资源

- 有关创建镜像 registry 的更多信息，请参阅 创建镜像 registry。
- 有关镜像镜像的更多信息，请参阅 为断开连接的安装镜像镜像。
- 如需有关镜像 Operator 目录的更多信息，请参阅 镜像 Operator 目录。

1.5. 集成现有组件

multicluster global hub 需要中间件组件、Kafka 和 PostgreSQL，以及 Grafana 作为 Observability 平台来提供策略合规视图。multicluster global hub 提供 Kafka、PostgreSQL 和 Grafana 的版本。您还可以集成您自己的现有 Kafka、PostgreSQL 和 Grafana。

- 集成现有 Kafka 版本
- 集成现有 PostgreSQL 版本
- 集成现有 Grafana 版本

1.5.1. 集成现有 Kafka 版本

如果您有自己的 Kafka 实例，可以使用它作为多集群全局 hub 的传输。Kafka 3.3 是经过测试的版本。完成以下步骤以集成 Kafka 实例：

1. 如果没有 Kafka 实例的持久性卷，则需要创建一个。
2. 在 **multicluster-global-hub** 命名空间中创建一个名为 **multicluster-global-hub-transport** 的 secret。
   a. 在以下必填字段中提取信息：
      - **bootstrap.servers**: 指定 Kafka bootstrap 服务器。
      - **ca.crt**: 如果您使用 **KafkaUser** 自定义资源配置身份验证凭据，则需要此项。有关从 secret 提取 **ca.crt** 证书所需的步骤，请参阅 STRIMZI 文档中的 *用户身份验证* 主题。
      - **client.crt**: 必需，请参阅 STRIMZI 文档中的 *User authentication* 主题，以了解从 secret 中提取 **user.crt** 证书的步骤。
      - **client.key**: 必需，请参阅 STRIMZI 文档中的 *用户身份验证* 主题，以了解从 secret 中提取 **user.key** 的步骤。
3. 运行以下命令来创建 secret，根据需要将值替换为您提取的值：

```
oc create secret generic multicluster-global-hub-transport -n multicluster-global-hub \
    --from-literal=bootstrap_server=<kafka-bootstrap-server-address> \
```

```
--from-file=ca.crt=<CA-cert-for-kafka-server> \
--from-file=client.crt=<Client-cert-for-kafka-server> \
--from-file=client.key=<Client-key-for-kafka-server>
```

4. 如果在 Kafka 实例中配置了自动主题创建，则跳过这一步。如果没有配置，请手动创建 **spec**、**status** 和 **event** 主题。

5. 确保访问 Kafka 的全局 hub 用户具有从主题中读取数据的权限，并将数据写入主题。

1.5.2. 集成现有 PostgreSQL 版本

如果您有自己的 PostgreSQL 关系数据库，您可以使用它作为多集群全局 hub 的存储。PostgreSQL 13 是经过测试的版本。

最低所需的存储大小为 20GB。这个数量可以存储有 250 个受管集群的 3 个受管 hub，每个受管 hub 需要 50 个策略(18 个月)。您需要在 **multicluster-global-hub** 命名空间中创建一个名为 **multicluster-global-hub-storage** 的 secret。secret 必须包含以下字段：

- **database_uri** ：用于创建数据库和插入数据。您的值必须类似以下格式： **postgres://<user>:<password>@<host>:<port>/<database>?sslmode=<mode>**。
- **database_uri_with_readonlyuser** ：它用于查询由多集群全局 hub 使用的 Grafana 实例的数据。它是可选值。您的值必须类似以下格式： **postgres://<user>:<password>@<host>:<port>/<database>?sslmode=<mode>**。
- **ca.crt** （基于 **sslmode** ）是一个可选值。

1. 验证您的集群是否具有最低需要 20GB 的存储大小。这个数量可以存储有 250 个受管集群的三个受管 hub，每个托管 hub 需要 18 个月。

2. 运行以下命令来创建 secret：

```
oc create secret generic multicluster-global-hub-storage -n multicluster-global-hub \
  --from-literal=database_uri=<postgresql-uri> \
  --from-literal=database_uri_with_readonlyuser=<postgresql-uri-with-readonlyuser> \
  --from-file=ca.crt=<CA-for-postgres-server>
```

主机必须可从多集群全局 hub 集群访问。如果您的 PostgreSQL 数据库位于 Kubernetes 集群中，您可以考虑使用带有 **nodePort** 或 **LoadBalancer** 的服务类型来公开数据库。如需更多信息，请参阅 访问置备的 postgres 数据库以进行故障排除。

1.5.3. 集成现有 Grafana 版本

如果您依赖自己的 Grafana 从不同的集群（如 Prometheus）获取指标数据，且如果您自行聚合指标，请使用现有的 Grafana 实例使用多集群全局 hub。要将多集群全局 hub 数据进入您自己的 Grafana 中，您需要配置数据源并导入仪表板。

1. 运行以下命令，从 multicluster global hub Grafana **数据源** secret 收集 PostgreSQL 连接信息：

```
oc get secret multicluster-global-hub-grafana-datasources -n multicluster-global-hub -
ojsonpath='{.data.datasources\.yaml}' | base64 -d
```

输出类似以下示例：

```
apiVersion: 1
```

```
datasources:
- access: proxy
  isDefault: true
  name: Global-Hub-DataSource
  type: postgres
  url: postgres-primary.multicluster-global-hub.svc:5432
  database: hoh
  user: guest
  jsonData:
    sslmode: verify-ca
    tlsAuth: true
    tlsAuthWithCACert: true
    tlsConfigurationMethod: file-content
    tlsSkipVerify: true
    queryTimeout: 300s
    timeInterval: 30s
  secureJsonData:
    password: xxxxx
    tlsCACert: xxxxx
```

2. 通过添加源（如 PostgreSQL）在您自己的 Grafana 实例中配置数据源，并使用之前提取的信息完成必填字段。

   请参见**以下必填字段：**

   - **Name**
   - **主机**
   - **数据库**
   - **用户**
   - **密码**
   - **TLS/SSL 模式**
   - **TLS/SSL 方法**

- CA 认证

3. 如果您的 Grafana 不在多集群全局 hub 集群中，您需要使用 LoadBalancer 公开 PostgreSQL，以便可以从外部访问 PostgreSQL。您可以将以下值添加到 PostgresCluster 操作对象中：

```
service:
  type: LoadBalancer
```

添加该内容后，您可以从 postgres-ha 服务获取 EXTERNAL-IP。请参见以下示例：

```
oc get svc postgres-ha -n multicluster-global-hub
NAME          TYPE           CLUSTER-IP     EXTERNAL-IP                          PORT(S)          AGE
postgres-ha   LoadBalancer   172.30.227.58   xxxx.us-east-1.elb.amazonaws.com   5432:31442/TCP   128m
```

运行该命令后，您可以使用 xxxx.us-east-1.elb.amazonaws.com:5432 作为 PostgreSQL Connection Host。

4. 导入现有的仪表板。

   a. 按照官方 Grafana 文档中的 导出和导入 仪表板中的步骤，从现有的 Grafana 实例导出仪表板。

   b. 按照官方 Grafana 文档中的 导出和导入 仪表板中的步骤，将仪表板导入到多集群全局 hub Grafana 实例中。

1.5.4. 其他资源

有关如何从 secret 提取 ca.crt 证书的更多信息，请参阅 STRIMZI 文档中的 用户身份验证。

如需了解从 secret 中提取 user.crt 证书的步骤，请参阅 STRIMZI 文档中的 用户身份验证。

1.6. 以默认模式导入受管 HUB 集群

要将现有 hub 集群导入为受管 hub 集群，请完成以下步骤：

1. 通过将 multiclusterhub 自定义资源中的 disableHubSelfManagement 设置为 true 来禁用现有 Red Hat Advanced Cluster Management for Kubernetes hub 集群中的集群自助管理。此设置禁用自动导入 hub 集群作为受管集群。

2. 通过完成集群导入 简介 中的步骤导入 受管 hub 集群。

3. 导入受管 hub 集群后，通过运行以下命令检查 multicluster global hub 代理状态，以确保代理在受管 hub 集群中运行：

```
oc get managedclusteraddon multicluster-global-hub-controller -n
$<managed_hub_cluster_name>
```

1.7. 访问 GRAFANA 数据

Grafana 数据通过路由公开。运行以下命令以显示登录 URL：

```
oc get route multicluster-global-hub-grafana -n <the-namespace-of-multicluster-global-hub-instance>
```

此 URL 的身份验证方法与向 Red Hat OpenShift Container Platform 控制台进行身份验证相同。

1.7.1. 使用 Grafana 仪表板查看策略状态

访问全局 hub Grafana 数据后，您可以监控通过管理的 hub 集群环境配置的策略。

在多集群全局 hub 仪表板中，您可以通过所选时间范围识别系统的策略的合规性状态。策略合规状态每日更新，因此控制面板不会显示当前当天的状态，直到下日为止。

要浏览多集群全局 hub 仪表板，您可以通过按策略或集群对策略数据进行分组来观察和过滤 策略 数据。

如果您希望使用策略分组来检查策略数据，请从 和名为 Global Hub - Policy Group Compliancy Overview 的控制面板开始。

此仪表板允许您根据 标准、类别 和 控制 来过滤策略数据。在图形上选择特定时间点后，您会被定向到 Global Hub - Offending Policies 仪表板。Global Hub - 官方策略 仪表板会在那个时候列出不合规或未知的策略。选择目标策略后，您可以查看相关事件，并通过访问 Global Hub - Changed / Policies 仪表板来查看更改的事件。

同样，如果要按 集群 分组检查策略数据，请从使用 Global Hub - Cluster Group Compliancy Overview 仪表板开始。导航流程 与策略 分组流程相同，但您可以选择与集群相关的过滤器，如受管集群标签和 值。在到达 Global Hub - 什么是 Changed / Clusters 仪表板后，您可以查看与单个集群相关的策略事件，而不是查看所有集群的策略事件。

1.8. GRAFANA 警报（技术预览）

您可以配置三个 Grafana 警报，它们存储在 multicluster-global-hub-default-alerting 配置映射中。这些警报通知您可疑策略、可疑集群合规状态更改和失败的 cron 作业。

请参阅以下警报描述：

- 可疑策略更改：此警报规则监视可疑策略更改。如果以下事件在一小时内发生超过五分钟，它会创建通知。
  - 一个策略被启用或禁用。
  - 一个策略已更新。
- 可疑集群合规状态更改：此警报规则监视集群的合规状态和策略事件。此警报有两个规则：
  - 集群合规状态经常更改：如果集群合规状态在一小时内从 合规 更改为 非合规 多次，它会创建通知。
  - 集群中的策略事件太多：对于集群中的策略，如果五分钟内有超过 20 个事件，它会创建通知。如果此警报始终触发，则 event.local_policies 表中的数据会变得太快。
- Cron Job failed ：此警报监视 cron 任务，如 为失败的事件 配置 cron 作业 中所述。此警报有两个规则：

- **本地合规作业失败：如果此警报规则创建通知，这意味着本地合规状态同步作业失败。可能会导致 history.local_compliance 表中的数据丢失。如有必要，手动运行作业。**

- **数据保留作业失败：如果此警报规则开始创建通知，这意味着数据保留作业失败。您可以手动运行它。**

1.8.1. 删除默认的 Grafana 警报规则

如果默认的 Grafana 警报规则不提供有用的信息，您可以通过在 multicluster-global-hub-custom-alerting 配置映射中包含 deleteRules 部分来删除 Grafana 警报规则。如需有关 multicluster-global-hub-custom-alerting 配置映射的更多信息，请参阅自定义 Grafana 警报资源。

要删除所有默认警报，deleteRules 配置部分应类似以下示例：

```
deleteRules:
  - orgId: 1
    uid: globalhub_suspicious_policy_change
  - orgId: 1
    uid: globalhub_cluster_compliance_status_change_frequently
  - orgId: 1
    uid: globalhub_high_number_of_policy_events
  - orgId: 1
    uid: globalhub_data_retention_job
  - orgId: 1
    uid: globalhub_local_compliance_job
```

1.8.2. 自定义 Grafana 警报

multicluster global hub 支持创建自定义 Grafana 警报。完成以下步骤以自定义 Grafana 警报：

1.8.2.1. 自定义 grafana.ini 文件

要自定义 grafana.ini 文件，请在安装 multicluster global hub operator 的命名空间中创建一个名为 multicluster-global-hub-custom-grafana-config 的 secret。secret 数据密钥是 grafana.ini，如下例所示。将所需信息替换为您自己的凭证：

```
apiVersion: v1
kind: Secret
metadata:
  name: multicluster-global-hub-custom-grafana-config
  namespace: multicluster-global-hub
type: Opaque
```

```
stringData:
  grafana.ini: |
    [smtp]
    enabled = true
    host = smtp.google.com:465
    user = <example@google.com>
    password = <password>
    ;cert_file =
    ;key_file =
    skip_verify = true
    from_address = <example@google.com>
    from_name = Grafana
    ;ehlo_identity = dashboard.example.com 1
```

<1> the EHLO 身份在 SMTP 对话框中，默认为 instance_name。

注： 您无法配置已包含 multicluster-global-hub-default-grafana-config secret 的部分。

1.8.2.2. 自定义 Grafana 警报资源

multicluster global hub 支持自定义警报资源，如 Grafana 文档中的使用文件置备创建和管理警报资源 中所述。

要自定义警报资源，请在 multicluster-global-hub -hub 命名空间中创建一个名为 multicluster-global-hub-custom-alerting 的配置映射。

配置映射数据键是 alerting.yaml，如下例所示：

```
apiVersion: v1
data:
  alerting.yaml: |
    contactPoints:
      - orgId: 1
        name: globalhub_policy
        receivers:
          - uid: globalhub_policy_alert_email
            type: email
            settings:
              addresses: <example@redhat.com>
              singleEmail: false
          - uid: globalhub_policy_alert_slack
            type: slack
            settings:
              url: <Slack-webhook-URL>
              title: |
```

```
          {{ template "globalhub.policy.title" . }}
        text: |
          {{ template "globalhub.policy.message" . }}
  policies:
    - orgId: 1
      receiver: globalhub_policy
      group_by: ['grafana_folder', 'alertname']
      matchers:
        - grafana_folder = Policy
      repeat_interval: 1d
  deleteRules:
    - orgId: 1
      uid: [Alert Rule Uid]
  muteTimes:
    - orgId: 1
      name: mti_1
      time_intervals:
        - times:
            - start_time: '06:00'
              end_time: '23:59'
              location: 'UTC'
          weekdays: ['monday:wednesday', 'saturday', 'sunday']
          months: ['1:3', 'may:august', 'december']
          years: ['2020:2022', '2030']
          days_of_month: ['1:5', '-3:-1']
kind: ConfigMap
metadata:
  name: multicluster-global-hub-custom-alerting
  namespace: multicluster-global-hub
```

1.9. 配置 CRON 任务

您可以配置 multicluster global hub 的 cron 作业设置。

安装 multicluster global hub 操作对象后，多集群全局 hub 管理器会运行，并显示您要调度以下 cron 作业的作业调度程序：

- 本地合规性状态同步作业：此 cron 作业每天根据前一天收集的策略状态和事件每天的午夜运行。运行此作业会总结了合规状态以及集群中策略的更改频率，并将其存储在 history.local_compliance 表中，作为 Grafana 仪表板的数据源。
- 数据保留作业：多集群全局 hub 中的一些数据表随着时间的推移继续增长，这通常可能会在表过大时造成问题。以下两种方法有助于最小化表导致的问题：
  - 删除不再需要的旧数据

- 在大型表中启用分区，以便更快地运行查询和删除

  对于事件表，如 event.local_policies 和 history.local_compliance，它每天增大，范围分区会将大表划分为较小的分区。这个过程还会在每次运行时为下一个月创建分区表。对于像 local_spec.policies 和 status.managed_clusters 等策略和集群表，表上有 deleted_at 索引，以便在硬删除时提高性能。

  您可以通过更改 multicluster global hub 操作对象上的 retention 设置来更改数据保留的时间持续时间。推荐的最小值为 1 个月，默认值为 18 个月。此作业的运行间隔应小于一个月。

每次多集群全局 hub 管理器启动时，列出的 cron 作业都会运行。本地合规状态同步作业每天运行一次，可以在不更改结果的情况下在一天内多次运行。数据保留作业每周运行一次，也可以每月多次运行，而无需更改结果。

这些作业的状态保存在名为 multicluster_global_hub_jobs_status 的指标中，该指标可从 Red Hat OpenShift Container Platform 集群的控制台查看。值 0 表示作业成功运行，而 1 表示失败。

如果有失败的作业，您可以使用日志表 (history.local_compliance_job_log,event.data_retention_job_log)进行故障排除。如需了解更多详细信息，请参阅恢复合规数据，以及决定是否手动运行该服务的信息。

1.10. 手动运行总结过程

您还可以手动运行总结过程。当您尝试调查问题或需要比下一个调度的例程更早的报告时，这非常有用。

手动总结过程由两个子任务组成：

- 将当天的集群策略数据从 Materialized View local_compliance_view_<yyyy_MM_dd> 插入 history.local_compliance。
- 更新 compliance 和策略会根据 event.local_policies 将那天的 frequency 改为 history.local_compliance。

完成以下步骤以手动运行总结过程：

1. **连接到数据库。**

   您可以使用 pgAdmin、tablePlush 等客户端连接到多集群全局 hub 数据库，在接下来的几个步骤中运行 SQL 语句。您可以运行以下命令来直接连接到集群中的数据库：

   ```
   oc exec -it multicluster-global-hub-postgres-0 -n multicluster-global-hub -- psql -d hoh
   ```

2. **确定它需要运行的日期，如 2023-07-06。**

   如果您在 2023-07-06 的仪表板中没有合规性信息，请在 history.local_compliance_job_log 中查找本当天的作业失败信息。在这种情况下，它是 2023-07-07。可以确定 2023-07-06 是我们需要手动运行摘要进程的日期。

3. **运行以下命令，检查 history.local_compliance_view_2023_07_06 的 Materialized View 是否存在：**

   ```
   select * from history.local_compliance_view_2023_07_06;
   ```

   如果存在视图，请运行以下命令将视图记录加载到 history.local_compliance 中：

   ```
   -- exec the insert func for that day '2023_07_06'
   SELECT history.insert_local_compliance_job('2023_07_06');
   ```

   如果视图不存在，请在一天之前继承一天的历史记录合规性记录，在本示例中，可能是 2023_07_05。

   ```
   -- call the func to generate the data of '2023_07_06' by inheriting '2023_07_05'
   CALL history.inherit_local_compliance_job('2023_07_05', '2023_07_06');
   ```

4. **将当天 的合规性和 频率 信息更新为 history.local_compliance。**

   ```
   -- call the func to update records start with '2023-07-06', end with '2023-07-07'
   SELECT history.update_local_compliance_job('2023_07_06', '2023_07_07');
   ```

5. **在 history.local_compliance 中查找该日期生成的记录。您可以运行以下命令来安全地删除**

Materialized View history.local_compliance_view_2023_07_06 ：

```
DROP MATERIALIZED VIEW IF EXISTS history.local_compliance_view_2023_07_06;
```

1.11. 多集群全局 HUB 备份（技术预览）

使用带有 Red Hat Advanced Cluster Management 备份和恢复功能的多集群全局 hub，用于恢复解决方案并访问基本资源。如需了解更多有关这些功能的信息，请参阅 备份和恢复。

多集群全局 hub 还支持使用 acm-hub-pvc-backup 备份 postgres pvc。为确保多集群全局 hub 可以支持备份 postgres pvc，您必须有当前版本的 VolySync 和 Red Hat Advanced Cluster Management。有关备份数据的详细步骤，请参阅 acm-hub-pvc-backup。

1.11.1. 恢复多集群全局 hub 备份和恢复

如果需要恢复多集群全局 hub 集群，请参阅准备 新的 hub 集群。安装 multicluster global hub operator，但不要创建 multicluster global hub 自定义资源(CR)，因为 CR 会自动恢复。