Red Hat Advanced Cluster Management for Kubernetes 2.10

Multicluster Global Hub

Multicluster Global Hub

Last Updated: 2024-05-24

Red Hat Advanced Cluster Management for Kubernetes 2.10 Multicluster Global Hub

Multicluster Global Hub

法律上の通知

概要

複数のハブクラスターを 1 つのハブで管理できる Mulitcluster Global Hub については、続きをお読みください。

目次

第1章 MULTICLUSTER GLOBAL HUB

Multicluster Global Hub は、1 つ以上のハブクラスターをインポートし、単一のハブクラスターから管理できるようにするコンポーネントのセットです。

ハブクラスターをマネージドハブクラスターとしてインポートした後、Multicluster Global Hub を使用して、すべてのマネージドハブクラスター全体に、次のタスクを実行できます。

- ポリシーのコンプライアンスの状態と傾向を報告する
- 概要ページですべてのマネージドハブとマネージドクラスターのインベントリーを作成する
- ポリシーの異常な動作を検出して警告する

Multicluster Global Hub は、大規模な環境で単一のハブクラスターでは多数のクラスターを管理できない場合に役立ちます。この問題が発生した場合は、クラスターをより小さなクラスターグループに分割し、グループごとにハブクラスターを設定します。

複数のハブクラスター上で、そのハブクラスターによって管理されているマネージドクラスターのデータを表示するのは不便なことがよくあります。Multicluster Global Hub は、複数のハブクラスターをマネージドハブクラスターとして指定することにより、複数のハブからの情報を表示する簡単な方法を提供します。multicluster global hub は、他のハブクラスターを管理し、マネージドハブクラスターから要約された情報を収集します。

注記: Multicluster Global Hub で可観測性は利用できません。クラスターに Multicluster Global Hub をインストールする前にマルチクラスター可観測性機能を有効にしている場合は、マルチクラスター可観測性機能を手動で無効にします。

Multicluster Global Hub の使用方法については、以下のセクションを参照してください。

- Multicluster Global Hub アーキテクチャー
- グローバルハブの要件
- 接続された環境への Multicluster Global Hub のインストール
- 非接続環境での Multicluster Global Hub のインストール
- 既存のコンポーネントの統合
- デフォルトモードでのマネージドハブクラスターのインポート
- Grafana データへのアクセス
- Grafana アラート (テクノロジープレビュー)
- cron ジョブの設定
- 要約プロセスの手動実行
- multicluster global hub のバックアップ (テクノロジープレビュー)

1.1. MULTICLUSTER GLOBAL HUB アーキテクチャー

Multicluster Global Hub は、ハブクラスターへのアクセスと管理に使用される次のコンポーネントで設定されます。

- 管理ツールとコンソールが実行される **グローバルハブクラスター** と呼ばれるサーバーコンポーネント
- Red Hat Advanced Cluster Management にインストールされる **マネージドハブ** と呼ばれるクライアントコンポーネント。グローバルハブクラスターによって管理できます。マネージドハブは他のクラスターも管理します。multicluster global hub クラスター専用のクラスターを使用する必要はありません。

アーキテクチャーの詳細は、次のセクションで説明します。

次の高度なマルチクラスター用語とコンポーネントを参照してください。

- Multicluster Global Hub Operator
- Multicluster Global Hub マネージャー
- Multicluster Global Hub エージェント
- Multicluster Global Hub の視覚化

1.1.1. Multicluster Global Hub Operator

Multicluster Global Hub Operator には、Multicluster Global Hub のコンポーネントが含まれています。Operator は、グローバルマルチクラスター管理に必要なコンポーネントをすべてデプロイします。コンポーネントには 、**multicluster-global-hub-manager**、**multicluster-global-hub-grafana**、および multicluster global hub 内の **Kafka** および **PostgreSQL** の提供バージョンと、マネージドハブクラスター内の **multicluster-global-hub-agent** が含まれます。

また、Operator は **manifestwork** カスタムリソースを利用して、マネージドクラスターに Red Hat Advanced Cluster Management for Kubernetes Operator をデプロイします。Red Hat Advanced Cluster Management Operator がマネージドクラスターにデプロイされると、マネージドクラスターは標準の Red Hat Advanced Cluster Management クラスターになります。このハブクラスターはマネージドハブクラスターになりました。

1.1.2. Multicluster Global Hub マネージャー

Multicluster Global Hub マネージャーは、データを **postgreSQL** データベースに永続化するために使用されます。データは Kafka トランスポートからのものです。また、マネージャーはデータを Kafka トランスポートにポストするため、マネージドハブクラスター上のデータと同期できます。

1.1.3. Multicluster Global Hub エージェント

Multicluster Global Hub エージェントは、マネージドハブクラスター上で実行されます。multicluster global hub クラスターと管理対象ハブクラスター間でデータを同期します。たとえば、エージェントは、マネージドハブクラスターからのマネージドクラスターの情報を multicluster global hub と同期し、グローバルハブクラスターとマネージドハブクラスターからのポリシーまたはアプリケーションを同期します。

1.1.4. Multicluster Global Hub の視覚化

Grafana は、Multicluster Global Hub の視覚化のメインサービスとして multicluster global hub クラスター上で実行されます。グローバルハブマネージャーによって収集された PostgreSQL データは、デフォルトの DataSource です。**multicluster-global-hub-grafana** というルートを使用してサービスを公開すると、コンソールにアクセスして multicluster global hub Grafana ダッシュボードにアクセスできます。

1.2. グローバルハブ要件

インストールとネットワークに必要なもの、およびサポートされるコンポーネントおよび環境について説明します。

- 一般要件
- ネットワーク要件
- サポートされるコンポーネント

1.2.1. 一般要件

Global Hub をインストールするには、次の要件が必要です。

必要なアクセス権限: クラスターの管理者

OpenShift Container Platform Dedicated 環境に必要なアクセス:cluster-admin パーミッションが必要です。デフォルトで、**dedicated-admin** ロールには OpenShift Container Platform Dedicated 環境で namespace を作成するために必要な権限がありません。

- Red Hat Advanced Cluster Management for Kubernetes をインストールして設定する必要があります。Red Hat Advanced Cluster Management の詳細は、こちらをご覧ください 。

1.2.2. ネットワーク要件

以下のネットワーク要件を参照してください。

- マネージドハブは、Red Hat Advanced Cluster Management のマルチクラスターグローバルハブのマネージドクラスターでもあります。Red Hat Advanced Cluster Management のネットワーク設定が必要です。Red Hat Advanced Cluster Management ネットワークの詳細は、Red Hat Advanced Cluster Management ネットワーク詳細の ネットワーク を参照してください。
- 以下の表は、グローバルハブネットワーク情報の一覧です。

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス	
ユーザーが使用するブラウザーからの受信	HTTPS	ユーザーは Grafana ダッシュボードにアクセスする必要があります。	443	ユーザーが使用するブラウザー	Grafana ルートの IP アドレス	

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス	
Kafka クラスターへのアウトバウンド	HTTPS	グローバルハブマネージャーは、Kafka クラスターからデータを取得する必要があります。	443	multicluster-global-hub-manager-xxx Pod	Kafka ルートホスト	
PostgreSQL データベースへの送信	HTTPS	グローバルハブマネージャーは、データを PostgreSQL データベースに永続化する必要があります。	443	multicluster-global-hub-manager-xxx Pod	PostgreSQL データベースの IP アドレス	

- 次の表に、マネージドハブネットワーク情報を示します。

方向	プロトコル	接続	ポート (指定されている場合)	送信元アドレス	宛先アドレス	
Kafka クラスターへのアウトバウンド	HTTPS	グローバルハブエージェントは、クラスター情報とポリシー情報を Kafka クラスターに同期する必要があります	443	multicluster-global-hub-agent pod	Kafka ルートホスト	

- サイズ設定のガイドラインは、製品ドキュメント の Red Hat Advanced Cluster Management クラスターのサイズ設定 を参照してください。
- **オプション:** ミドルウェアの場合、Multicluster Global Hub には Kafka、PostgreSQL、および Grafana が組み込まれていますが、独自に設定された Kafka、PostgreSQL、および Grafana を使用することもできます。詳細は、既存コンポーネントの統合 を参照してください。

1.2.3. サポートされるコンポーネント

サポートされるプラットフォームとコンポーネントについて説明します。

- Multicluster Global Hub コンソールは統合コンソールを共有するため、OpenShift Container Platform と同じブラウザーをサポートします。サポート対象のブラウザーとバージョンについては、Red Hat OpenShift Container Platform ドキュメントの Web コンソールへのアクセス を参照してください。
- サポートされている Multicluster Global Hub クラスターで利用可能なプラットフォームを次の表に示します。

プラットフォーム	グローバルハブクラスターでのサポート	マネージドハブクラスターのサポート
Red Hat Advanced Cluster Management 2.10 以降の 2.10 リリース	○	○
Red Hat Advanced Cluster Management 2.9 以降の 2.9.x リリース	○	○
Red Hat Advanced Cluster Management 2.8.3 以降の 2.8.x リリース	○	○
Red Hat Advanced Cluster Management on Arm	○	○
IBM Z での Red Hat Advanced Cluster Management	○	○
IBM Power Systems での Red Hat Advanced Cluster Management	○	○

- Multicluster Global Hub は、以下のミドルウェアをサポートします。
  - Kafka 3.4 以降の 3.4.x リリース。
  - PostgreSQL バージョン 13 以降の 13.x リリース。

1.2.4. 関連情報

- 接続された環境への Multicluster Global Hub のインストール
- 非接続環境での Multicluster Global Hub のインストール

1.3. 接続された環境への MULTICLUSTER GLOBAL HUB のインストール

Multicluster Global Hub は、Multicluster Global Hub を含むコンポーネントのインストール、アップグレード、および削除を管理する Operator Lifecycle Manager を通じてインストールされます。

必要なアクセス権限: クラスターの管理者

1.3.1. 前提条件

- OpenShift Container Platform Dedicated 環境の場合、**クラスター管理者** 権限があり、この環境にアクセスできるデフォルトで、**dedicated-admin** ロールには OpenShift Container Platform Dedicated 環境で namespace を作成するために必要な権限がありません。
- Red Hat Advanced Cluster Management for Kubernetes をインストールおよび設定しておく。詳細は、インストールとアップグレード を参照してください。
- Red Hat Advanced Cluster Management ネットワークを設定している。マネージドハブは、Red Hat Advanced Cluster Management の multicluster global hub のマネージドクラスターでもあります。詳細は、ハブクラスターネットワーク設定 を参照してください。

1.3.1.1. コンソールを使用した Multicluster Global Hub のインストール

OpenShift Container Platform コンソールを使用して接続環境に Multicluster Global Hub Operator をインストールするには、以下の手順を実行します。

1. **cluster-admin** ロールを持つユーザーとして OpenShift Container Platform コンソールにログインします。
2. ナビゲーションメニューから **Operators** > **OperatorHub** アイコンを選択します。
3. **Multicluster global hub operator** を見つけて選択します。
4. **Install** をクリックしてインストールを開始します。
5. インストールが完了したら、**Installed Operators** ページでステータスを確認します。
6. **Multicluster global hub operator** をクリックして、**Operator** ページに移動します。
7. **Multicluster global hub** タブをクリックして、**multicluster global hub** インスタンスを表示します。
8. **Create multicluster global hub** をクリックして、**multicluster global hub** インスタンスを作成します。
9. 必要な情報を入力し、**Create** をクリックして **multicluster global hub** インスタンスを作成します。

注記:

- Multicluster Global Hub は、x86 プラットフォームでのみ使用できます。
- Multicluster Global Hub のインストール後、Red Hat Advanced Cluster Management でポリシーとアプリケーションが無効になります。

1.3.2. 関連情報

- Operator カタログのミラーリングの詳細は、 Operator カタログのミラーリング を参照してください。
- プライベートレジストリーからのイメージへのアクセスの詳細は、プライベートレジストリーからオペレータのイメージにアクセスする を参照してください。
- カタログソースの追加の詳細は、クラスターへのカタログソースの追加 を参照してください。

- 切断された環境での Red Hat Advanced Cluster Management のインストールの詳細は、非接続ネットワーク環境でのインストール を参照してください。
- イメージのミラーリングの詳細は、非接続インストールのイメージのミラーリング を参照してください。
- Operator SDK と OLM の統合の詳細は、Operator SDK と Operator Lifecycle Manager の統合 を参照してください。

1.4. 非接続環境での MULTICLUSTER GLOBAL HUB のインストール

クラスターが制限付きネットワーク内にある場合は、非接続環境に multicluster global hub operator をデプロイメントできます。

必要なアクセス権限: クラスターの管理者

1.4.1. 前提条件

非接続環境に multicluster global hub をインストールする前に、次の要件を満たす必要があります。

- インターネットとミラーレジストリーの両方にアクセスできるイメージレジストリーと bastion ホスト
- Operator Lifecycle Manager をクラスターにインストールする。Operator Lifecycle Manager (OLM) を参照してください。
- Red Hat Advanced Cluster Management for Kubernetes をインストールする。
- 以下のコマンドラインインターフェイスをインストールする。
  - OpenShift Container Platform コマンドライン。OpenShift Container Platform CLI のスタートガイド を参照してください。
  - **opm** コマンドライン。opm CLI のインストール を参照してください。
  - **oc-mirror** プラグイン。oc-mirror プラグインを使用した非接続インストールのイメージのミラーリング を参照してください。

1.4.2. ミラーレジストリーの設定

非接続環境に multicluster global hub をインストールするには、ローカルミラーイメージレジストリーを使用する必要があります。この時点では、OpenShift Container Platform クラスターのインストール中にミラーレジストリーを設定済みであることを前提としています。

以下の手順に従って、multicluster global hub のミラーレジストリーをプロビジョニングします。

1.4.2.1. oc-mirror プラグインを使用したミラーカタログでの Operator パッケージの作成

Red Hat は、Multicluster Global Hub および AMQ Streams Operator を Red Hat Operator カタログで提供しており、これらは、**registry.redhat.io/redhat/redhat-operator-index** インデックスイメージによって提供されます。このカタログインデックスイメージのミラーを準備する場合に、Red Hat が提供するカタログ全体をミラーリングするか、使用する Operator パッケージのみを含むサブセットをミラーリングするかを選択できます。

フルミラーカタログを作成する場合、Multicluster Global Hub および AMQ Streams のインストールに必要なすべてのパッケージが含まれているため、特別な考慮事項はありません。ただし、特定のパッ

ケージを含めるかを特定するために、一部または絞り込んだミラーリングカタログを作成する場合には、**multicluster-global-hub-operator-rh** と **amq-streams** のパッケージ名をリストに含める必要があります。

multicluster-global-hub-operator-rh および **amq-streams** パッケージのローカルミラーレジストリーを作成するには、以下の手順を実行します。

1. **ImageSetConfiguration** YAML ファイルを作成し、Operator イメージを設定して追加します。YAML ファイルは次の内容のようになり、現在のバージョンを **4.x** に置き換えます。

```
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. 次のコマンドを使用して、イメージセットをターゲットミラーレジストリーに直接ミラーリングします。

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

3. 完全な非接続環境でイメージセットをミラーリングします。詳細は、Mirroring images for a disconnected installationを参照してください。

1.4.2.2. 非接続クラスターへのレジストリーとカタログの追加

非接続クラスターでミラーレジストリーとカタログを使用できるようにするには、以下の手順を実行します。

1. Operator Hub のデフォルトのカタログソースを無効にします。以下のコマンドを実行して **OperatorHub** リソースを更新します。

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

2. Operator カタログのミラーリング の手順を実行して、Operator カタログをミラーリング します。

3. ミラーリングされたカタログの **CatalogSource** リソースを **openshift-marketplace** namespace に追加します。**CatalogSource** YAML ファイルは以下の例のようになります。

```
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>
```

- **注記: metadata.name** フィールドの値に注意してください。

4. 更新したファイルを保存します。

5. 利用可能な **PackageManifest** リソースをクエリーして、非接続クラスターから必要なパッケージが利用できることを確認します。以下のコマンドを実行します。

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

表示されるリストには **multicluster-global-hub-operator-rh** および **amq-streams** パッケージがミラーカタログのカタログソースによって提供されていることを示すエントリーが含まれているはずです。

1.4.3. イメージレジストリーの設定

クラスターがインターネットでホストされているレジストリーからではなく、ローカルミラーレジストリーからMulticluster Global Hub Operator のコンテナーイメージを取得できるようにするには、非接続クラスターで **ImageContentSourcePolicy** リソースを設定し、イメージ参照をミラーレジストリーにリダイレクトする必要があります。**ImageContentSourcePolicy** は、digest イメージを使用したイメージミラーのみをサポートします。

oc adm catalog mirror コマンドを使用してカタログをミラーリングした場合に、必要なイメージコンテンツソースポリシー設定は、そのコマンドによって作成される **manifests-*** ディレクトリー内の **imageContentSourcePolicy.yaml** ファイルにあります。

代わりに **oc-mirror** プラグインを使用してカタログをミラーリングした場合に、**imageContentSourcePolicy.yaml** ファイルは **oc-mirror** プラグインによって作成された **oc-mirror-workspace/results-** ディレクトリー内にあります。

いずれの場合も、**oc replace -f ./<path>/imageContentSourcePolicy.yaml** のような **oc apply** または **oc replace** コマンドを使用して、ネットワーク接続なしのコマンドにポリシーを適用できます。

必要なイメージコンテンツソースポリシーステートメントは、ミラーレジストリーの作成方法によって異なりますが、次の例のようになります。

```
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
```

ManagedClusterImageRegistry を使用して、マネージドハブごとに異なるイメージレジストリーを設定できます。**ManagedClusterImageRegistry** API を使用してエージェントイメージを置き換える方法については、ManagedClusterImageRegistry を持つクラスターのインポート を参照してください。

前の手順を完了すると、選択した **ManagedCluster** にラベルとアノテーションが追加されます。これは、クラスター内のエージェントイメージがミラーイメージに置き換えられることを意味します。

- ラベル: **multicluster-global-hub.io/image-registry=<namespace.managedclusterimageregistry-name>**
- アノテーション: **multicluster-global-hub.io/image-registries: <image-registry-info>**

1.4.3.1. イメージプルシークレットを設定する

サブスクライブされた Operator によって参照される Operator イメージまたはオペランドイメージにプライベートレジストリーへのアクセスが必要な場合は、クラスター内のすべての namespace または個々のターゲットテナント namespace のいずれかにアクセスを提供 できます。

1.4.3.1.1. OpenShift Container Platform クラスターでの multicluster global hub イメージプルシークレットの設定

イメージプルシークレットを既存の OpenShift Container Platform クラスターに設定できます。

注記: 既存のクラスターにイメージプルシークレットを適用すると、すべてのノードがローリング再起動されます。

プルシークレットを設定するには、以下の手順を実行します。

1. プルシークレットからユーザー名をエクスポートします。

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

2. プルシークレットからパスワードをエクスポートします。

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

3. プルシークレットをコピーします。

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

4. プルシークレットを使用してログインします。

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

5. multicluster global hub イメージプルシークレットを指定します。

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

6. 古いプルシークレットを削除します。

```
rm pull_secret.yaml
```

1.4.3.1.2. multicluster global hub イメージプルシークレットの個別の namespace への設定

以下の手順を実行することで、イメージプルシークレットを個別の namespace に設定できます。

1. 次のコマンドを実行して、テナントの namespace にシークレットを作成します。

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

2. シークレットをオペレーターまたはオペランドのサービスアカウントにリンクします。

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

1.4.3.2. グローバルハブオペレーターのインストール

Red Hat OpenShift Container Platform Web コンソールを使用して、OperatorHub から Operator をインストールしてサブスクライブできます。手順は、クラスターへのオペレーターの追加 を参照してください。Operator を追加した後、次のコマンドを実行して、multicluster global hub Operator のステータスを確認できます。

```
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. 関連情報

- ミラーレジストリーの作成の詳細は、ミラーレジストリーの作成 を参照してください。
- イメージのミラーリングの詳細は、非接続インストールのイメージのミラーリング を参照してください。
- Operator カタログのミラーリングの詳細は、 Operator カタログのミラーリング を参照してください。

1.5. 既存のコンポーネントの統合

Multicluster Global Hub には、ミドルウェアコンポーネント Kafka と PostgreSQL に加えて、ポリシーコンプライアンスビューを提供する可観測性プラットフォームとして Grafana が必要です。
Multicluster Global Hub は、Kafka、PostgreSQL、および Grafana のバージョンを提供します。また、独自の既存の Kafka、PostgreSQL、および Grafana を統合することもできます。

- 既存の Kafka バージョンの統合
- 既存の PostgreSQL バージョンの統合

- 既存の Grafana バージョンの統合

1.5.1. 既存の Kafka バージョンの統合

独自の Kafka インスタンスがある場合は、Multicluster Global Hub のトランスポートとして使用できます。Kafka 3.3 はテスト済みバージョンです。Kafka のインスタンスを統合するには、次の手順を実行します。

1. Kafka インスタンス用の永続ボリュームがない場合は、永続ボリュームを作成する必要があります。
2. **multicluster-global-hub** namespace に **multicluster-global-hub-transport** という名前のシークレットを作成します。
   a. 次の必須フィールドの情報を展開します。
      - **bootstrap.servers**: Kafka ブートストラップサーバーを指定します。
      - **ca.crt**: **KafkaUser** カスタムリソースを使用して認証情報を設定する場合に必要です。シークレットから **ca.crt** 証明書を抽出するために必要な手順は、STRIMZI ドキュメントの **ユーザー認証** トピックを参照してください。
      - **client.crt**: 必須。シークレットから **user.crt** 証明書を抽出する手順は、STRIMZI ドキュメントの **ユーザー認証** トピックを参照してください。
      - **client.key**: 必須。シークレットから **user.key** を抽出する手順は、STRIMZI ドキュメントの **ユーザー認証** トピックを参照してください。
3. 次のコマンドを実行してシークレットを作成し、必要に応じて値を抽出した値に置き換えます。

```
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. Kafka インスタンスでトピックの自動作成が設定されている場合は、この手順をスキップしてください。設定されていない場合は、**spec**、**status**、および **event** トピックを手動で作成します。
5. Kafka にアクセスするグローバルハブユーザーに、トピックからデータを読み取り、トピックにデータを書き込む権限があることを確認します。

1.5.2. 既存の PostgreSQL バージョンの統合

独自の PostgreSQL リレーショナルデータベースがある場合は、Multicluster Global Hub のストレージとして使用できます。PostgreSQL 13 はテスト済みバージョンです。

最小要件のストレージサイズは 20GB です。この量では、250 のマネージドクラスターを含む 3 つのマネージドハブと、マネージドハブごとに 50 のポリシーを 18 か月間保存できます。**multicluster-global-hub** namespace に **multicluster-global-hub-storage** という名前のシークレットを作成する必要があります。シークレットには以下のフィールドが含まれている必要があります。

- **database_uri**: データベースを作成し、データを挿入するために使用されます。値は、**postgres://<user>:<password>@<host>:<port>/<database>?sslmode=<mode>** の形式にする必要があります。

- **database_uri_with_readonlyuser**: Multicluster Global Hub で使用される Grafana のインスタンスによってデータのクエリーに使用されます。これは、オプションの値です。値は **postgres://<user>:<password>@<host>:<port>/<database>?sslmode=<mode>** の形式にする必要があります。

- **sslmode** に基づく **ca.crt** は、オプションの値です。

1. クラスターに最低限必要なストレージサイズが 20GB であることを確認します。この量では、250 のマネージドクラスターを含む 3 つのマネージドハブと、マネージドハブごとに 50 のポリシーを 18 か月間保存できます。

2. 以下のコマンドを実行してシークレットを作成します。

```
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>
```

ホストは、multicluster global hub クラスターからアクセスできる必要があります。PostgreSQL データベースが Kubernetes クラスターにある場合は、**nodePort** または **LoadBalancer** でサービスタイプを使用してデータベースを公開することを検討してください。詳細は、トラブルシューティングのためのプロビジョニングされた postgres データベースへのアクセス を参照してください。

1.5.3. 既存の Grafana バージョンの統合

独自の Grafana に依存して Prometheus などの複数のソースからメトリクスを取得し、異なるクラスターからメトリクスを自分で集計する場合、既存の Grafana インスタンスを使用すると、Multicluster Global Hub で機能する可能性があります。Multicluster Global Hub データを独自の Grafana に取得するには、データソースを設定し、ダッシュボードをインポートする必要があります。

1. 次のコマンドを実行して、Multicluster Global Hub Grafana **datasource** シークレットから PostgreSQL 接続情報を収集します。

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

出力は以下の例のようになります。

```
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. PostgreSQL などのソースを追加して独自の Grafana インスタンスに **datasource** を設定し、以前に抽出した情報を必須フィールドに入力します。
以下の必須フィールドを参照してください。

   - 名前
   - Host
   - Database
   - User
   - Password
   - TLS/SSL Mode
   - TLS/SSL Method
   - CA Cert

3. Grafana が Multicluster Global Hub クラスターにない場合は、**LoadBalancer** を使用して PostgreSQL を公開して、外部から PostgreSQL にアクセスできるようにする必要があります。以下の値を **PostgresCluster** オペランドに追加できます。

```
service:
  type: LoadBalancer
```

   そのコンテンツを追加すると、**postgres-ha** サービスから **EXTERNAL-IP** を取得できるようになります。以下の例を参照してください。

```
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
```

   このコマンドを実行すると、PostgreSQL 接続ホストとして **xxxx.us-east-1.elb.amazonaws.com:5432** を使用できます。

4. 既存のダッシュボードをインポートします。

   a. 公式 Grafana ドキュメントの Export and import dashboards の手順に従って、既存の Grafana インスタンスからダッシュボードをエクスポートします。

   b. 公式 Grafana ドキュメントの Export and import dashboards の手順に従って、ダッシュボードを Multicluster Global Hub Grafana インスタンスにインポートします。

1.5.4. 関連情報

シークレットから **ca.crt** 証明書を抽出する方法は、STRIMZI ドキュメントの User authentication を参照してください。

シークレットから **user.crt** 証明書を抽出する手順は、STRIMZI ドキュメントの User authentication を参照してください。

1.6. デフォルトモードでのマネージドハブクラスターのインポート

既存のハブクラスターをマネージドハブクラスターとしてインポートするには、以下の手順を実行します。

1. **multiclusterhub** カスタムリソースの **disableHubSelfManagement** 設定を **true** に指定して、既存の Red Hat Advanced Cluster Management for Kubernetes ハブクラスターのクラスター自己管理を無効にします。この設定により、ハブクラスターのマネージドクラスターとしての自動インポートが無効になります。

2. クラスターの インポートの概要 の手順を完了して、マネージドハブクラスターをインポートします。

3. マネージドハブクラスターをインポートしたら、次のコマンドを実行して、multicluster global hub エージェントのステータスをチェックし、管理対象ハブクラスターでエージェントが実行されていることを確認します。

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

1.7. GRAFANA データへのアクセス

Grafana データはルート経由で公開されます。次のコマンドを実行して、ログイン URL を表示します。

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

この URL の認証方法は、Red Hat OpenShift Container Platform コンソールに対する認証と同じです。

1.7.1. Grafana ダッシュボードを使用したポリシーステータスの表示

グローバルハブ Grafana データにアクセスした後、管理されているハブクラスター環境を通じて設定されたポリシーを監視できます。

multicluster global hub ダッシュボードから、選択した時間範囲におけるシステムのポリシーのコンプライアンスステータスを特定できます。ポリシーのコンプライアンスステータスは毎日更新されるため、ダッシュボードには当日のステータスが翌日まで表示されません。

multicluster global hub ダッシュボードを操作するには、**policy** または **cluster** 別にグループ化することで、ポリシーデータの確認および絞り込みが可能です。

policy グループを使用してポリシーデータを調べる場合は、**Global Hub - Policy Group Compliancy Overview** というダッシュボードから開始します。

このダッシュボードでは、**standard**、**category** および **control** に基づいてポリシーデータをフィルタリングできます。グラフで特定の時点を選択すると、**Global Hub - Offending Policies** ダッシュボードに移動します。**Global Hub - Offending Policies** ダッシュボードには、その時点で準拠していないポリ

シーまたは不明なポリシーがリストされます。ターゲットポリシーを選択した後、**Global Hub - What's Changed / Policies** ダッシュボードにアクセスして、関連イベントを表示し、変更内容を確認できます。

同様に、**cluster** グループごとにポリシーデータを調べる場合は、まず **Global Hub - Cluster Group Compliancy Overview** ダッシュボードを使用します。ナビゲーションフローは **policy** グループ化フローと同じですが、マネージドクラスターの **labels** や **values** など、クラスターに関連するフィルターを選択します。すべてのクラスターのポリシーイベントを表示する代わりに、**Global Hub - What's Changed / Clusters** ダッシュボードに到達した後、個々のクラスターに関連するポリシーイベントを表示できます。

1.8. GRAFANA アラート (テクノロジープレビュー)

3 つの Grafana アラートを設定でき、これらは **multicluster-global-hub-default-alerting** config map に保存されます。これらのアラートは、不審なポリシー、不審なクラスターのコンプライアンスステータスの変更、および失敗した cron ジョブを通知します。

次のアラートの説明を参照してください。

- Suspicious policy change - このアラートルールは、不審なポリシーの変更を監視します。以下のイベントが 1 時間以内に 5 回以上発生すると、通知が作成されます。
  - ポリシーが有効または無効になった。
  - ポリシーが更新された。
- Suspicious cluster compliance status change - このアラートルールは、クラスターのクラスターコンプライアンスステータスとポリシーイベントを監視します。このアラートには、次の 2 つのルールがあります。
  - Cluster compliance status changes frequently: クラスターのコンプライアンスステータスが 1 時間に 3 回以上 **compliance** から **compliance** に変化すると、通知が作成されます。
  - Too many policy events in a cluster: クラスター内のポリシーの場合、5 分間に 20 を超えるイベントがある場合、通知が作成されます。このアラートが常に発生している場合、**event.local_policies** テーブル内のデータが急増します。
- Cron Job Failed - このアラートは、失敗したイベントの cron ジョブの設定 で説明されている cron ジョブを監視します。このアラートには、次の 2 つのルールがあります。
  - Local compliance job failed: このアラートルールが通知を作成する場合、ローカルコンプライアンスステータス同期ジョブが失敗したことを意味します。これにより、**history.local_compliance** テーブルのデータが失われる可能性があります。必要に応じてジョブを手動で実行します。
  - Data retention job failed: このアラートルールが通知の作成を開始した場合、データ保持ジョブが失敗したことを意味します。これは手動で実行できます。

1.8.1. デフォルトの Grafana アラートルールの削除

デフォルトの Grafana アラートルールで要件にあった情報が得られない場合は、**multicluster-global-hub-custom-alerting** configmap に **deleteRules** セクションを含めることによって、Grafana アラートルールを削除できます。**multicluster-global-hub-custom-alerting** configmap の詳細は、Grafana アラートリソースのカスタマイズ を参照してください。

デフォルトのアラートをすべて削除するには、**deleteRules** 設定セクションは次の例のようになります。

```
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. Grafana アラートのカスタマイズ

Multicluster Global Hub は、カスタム Grafana アラートの作成をサポートします。Grafana アラートをカスタマイズするには、次の手順を実行します。

1.8.2.1. grafana.ini ファイルのカスタマイズ

grafana.ini ファイルをカスタマイズするには、Multicluster Global Hub Operator をインストールした namespace に **multicluster-global-hub-custom-grafana-config** という名前のシークレットを作成します。次の例に示すように、シークレットデータキーは **grafana.ini** です。必要な情報を独自の認証情報に置き換えます。

```
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> **SMTP** ダイアログの **EHLO** ID。デフォルトは、**instance_name** です。

注: セクションに **multicluster-global-hub-default-grafana-config** シークレットがすでに含まれている場合は、設定できません。

1.8.2.2. Grafana アラートルールのカスタマイズ

Multicluster Global Hub は、Grafana ドキュメントの ファイルプロビジョニングを使用したアラートリソースの作成と管理 で説明されているアラートリソースのカスタマイズをサポートしています。

アラートリソースをカスタマイズするには、**multicluster-global-hub-custom-alerting** という名前の config map を **multicluster-global-hub** namespace に作成します。

次の例に示すように、config map データキーは **alerting.yaml** です。

```
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. CRON ジョブの設定

Multicluster Global Hub の cron ジョブを設定できます。

Multicluster Global Hub オペランドをインストールすると、グローバルハブマネージャーが実行され、次の cron ジョブをスケジュールするためのジョブスケジューラーが表示されます。

- Local compliance status sync job: この cron ジョブは、前日にマネージャーによって収集されたポリシーステータスとイベントに基づいて、毎日午前 0 時に実行されます。このジョブを実行すると、クラスター上のポリシーのコンプライアンスステータスと変更頻度が要約され、それらが Grafana ダッシュボードのデータソースとして **history.local_compliance** テーブルに保存されます。

- Data retention job: multicluster global hub 内の一部のデータテーブルは時間の経過とともに増大し続けるため、通常、テーブルが大きくなりすぎると問題が発生する可能性があります。次の 2 つの方法は、サイズが過剰なテーブルに起因する問題を最小限に抑えるのに役立ちます。

  - 不要になった古いデータを削除する

  - 大規模なテーブルでパーティション設定を有効にして、素早くクエリーや削除を実行する **event.local_policies** や **history.local_compliance** など、毎日サイズが増加するイベントテーブルの場合、範囲パーティション分割により大きなテーブルが小さなパーティションに分割されます。このプロセスは、実行されるたびに翌月のパーティションテーブルも作成します。**local_spec.policies** や **status.managed_clusters** などのポリシーおよびクラスターテーブルの場合、ハード削除時のパフォーマンスを向上させるために、テーブルに **delete_at** インデックスがあります。

    multicluster global hub オペランドの **retention** 設定を変更することで、データが保持される期間を変更できます。推奨される最小値は 1 カ月で、デフォルト値は 18 カ月です。このジョブの実行間隔は 1 か月未満である必要があります。

一覧表示されている cron ジョブは、multicluster global hub マネージャーが起動するたびに実行されます。ローカルコンプライアンスステータス同期ジョブは 1 日に 1 回実行され、結果を変えることなく 1 日に複数回実行できます。データ保持ジョブは週に 1 回実行されますが、結果に変化がなければ月に何度も実行することもできます。

これらのジョブのステータスは、**multicluster_global_hub_jobs_status** という名前のメトリックに保存され、Red Hat OpenShift Container Platform クラスターのコンソールから表示できます。値 **0** はジョブが正常に実行されたことを示し、値 **1** は失敗を示します。

失敗したジョブがある場合は、ログテーブル (**history.local_compliance_job_log**、**event.data_retention_job_log**) を使用してトラブルシューティングを行うことができます。サービスを手動で実行するかどうかを決定するためのガイダンスや詳細は、コンプライアンスデータの復元 を参照してください。

1.10. 要約プロセスの手動実行

要約プロセスを手動で実行することもできます。これは、問題を調査しようとしている場合や、次にスケジュールされているルーチンよりも早くレポートが必要な場合に役立ちます。

手動要約プロセスは、次の 2 つのサブタスクで設定されます。

- その日のクラスターポリシーデータを マテリアライズドビュー**local_compliance_view_<yyyy_MM_dd>** から **history.local_compliance** に挿入します。

- その日の **compliance** とポリシー切り替えの **frequency** を **event.local_policies** に基づいて **history.local_compliance** に更新します。

要約プロセスを手動で実行するには、次の手順を実行します。

1. データベースに接続します。
   pgAdmin、tablePlush などのクライアントを使用して multicluster global hub データベースに接続し、次のいくつかの手順で SQL ステートメントを実行できます。次のコマンドを実行すると、クラスター上のデータベースに直接接続できます。

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

2. **2023-07-06** など、実行する必要がある日付を決定します。
   ダッシュボードに **2023-07-06** のコンプライアンス情報がない場合は、**history.local_compliance_job_log** でこの日の翌日のジョブ失敗情報を見つけます。この場合、それは **2023-07-07** です。**2023-07-06** は、要約プロセスを手動で実行する必要がある日付であると判断できます。

3. 次のコマンドを実行して、**history.local_compliance_view_2023_07_06** の Materialized View が存在するかどうかを確認します。

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

   ビューが存在する場合は、次のコマンドを実行してビューレコードを **history.local_compliance** にロードします。

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

   ビューが存在しない場合は、その日の前日 (この例では **2023_07_05)** の履歴コンプライアンスレコードを継承します。

   ```
   -- 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. その日の **compliance** および **frequency** 情報を **history.local_compliance** に更新します。

   ```
   -- 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. **history.local_compliance** で生成されたその日の記録を見つけます。次のコマンドを実行すると、Materialized View **history.local_compliance_view_2023_07_06** を安全に削除できます。

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

1.11. MULTICLUSTER GLOBAL HUB のバックアップ (テクノロジープレビュー)

リカバリーソリューションや基本リソースを利用するには、Red Hat Advanced Cluster Management のバックアップおよび復元機能を備えた multicluster global hub を使用します。これらの機能がどのように役立つかの詳細は、バックアップと復元 を参照してください。

multicluster global hub は **acm-hub-pvc-backup** を使用した **postgres PVC** のバックアップもサポート

します。マルチクラスターグローバルハブがバックアップ **postgres pvc** をサポートできるようにするには、現在のバージョンの VolySync および Red Hat Advanced Cluster Management が必要です。データのバックアップに関する詳細な手順は、acm-hub-pvc-backup を参照してください。

1.11.1. multicluster global hub のバックアップおよび復元の復元

multicluster global hub を復元する必要がある場合は、新しいハブクラスターの準備 を参照してください。multicluster global hub Operator をインストールしますが、CR は自動的に復元されるため、multicluster global hub カスタムリソース (CR) は作成しないでください。