



Cost Management Service 1-latest

Google Cloud データの Cost Management への 統合

Google Cloud インテグレーションを追加して設定する方法

Cost Management Service 1-latest Google Cloud データの Cost Management への統合

Google Cloud インテグレーションを追加して設定する方法

法律上の通知

Copyright © 2024 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux[®] is the registered trademark of Linus Torvalds in the United States and other countries.

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

概要

Google Cloud のインテグレーションを Cost Management に追加する方法を説明します。Cost Management は、Red Hat Insights ポートフォリオサービスに含まれます。高度な分析ツールである Red Hat Insights スイートは、運用、セキュリティー、およびビジネスへの影響を特定して優先順位を付けるのに役立ちます。

目次

第1章 GOOGLE CLOUD インテグレーションの作成	3
1.1. GOOGLE CLOUD アカウントのインテグレーションとしての追加	3
1.2. GOOGLE CLOUD プロジェクトの作成	4
1.3. GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT のロール作成	4
1.4. GOOGLE CLOUD プロジェクトへの課金サービスアカウントメンバーの追加	5
1.5. GOOGLE CLOUD BIGQUERY のデータセット作成	6
1.6. GOOGLE CLOUD の課金データの BIGQUERY への書き出し	6
第2章 フィルタリングされた GOOGLE CLOUD データの COST MANAGEMENT への統合	8
2.1. GOOGLE CLOUD アカウントのインテグレーションとしての追加	8
2.2. GOOGLE CLOUD プロジェクトの作成	9
2.3. GOOGLE CLOUD バケットの作成	9
2.4. GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT のロール作成	10
2.5. GOOGLE CLOUD プロジェクトへの課金サービスアカウントメンバーの追加	11
2.6. GOOGLE CLOUD BIGQUERY のデータセット作成	12
2.7. GOOGLE CLOUD の課金データの BIGQUERY への書き出し	12
2.8. フィルターされたデータをストレージバケットに POST する関数の作成	13
2.9. フィルターされたデータをストレージバケットに POST する関数のトリガー	16
第3章 コストを管理するための次のステップ	18
3.1. COST MANAGEMENT リソースへのアクセス制限	18
3.2. インテグレーションのタグ付けの設定	18
3.3. コストを正確にレポートするためのコストモデルの設定	19
3.4. COST EXPLORER を使用したコストの可視化	19
RED HAT ドキュメントへのフィードバック (英語のみ)	20

第1章 GOOGLE CLOUD インテグレーションの作成

Google Cloud アカウントを Cost Management に追加するには、メトリクスを提供するように Google Cloud を設定し、[Red Hat Hybrid Cloud Console](#) ユーザーインターフェイスからインテグレーションとして Google Cloud アカウントを追加する必要があります。



注記

Cost Management にインテグレーションを追加するには、クラウド管理者のエンタイトルメントを持つユーザーが必要です。

Google Cloud アカウントをデータインテグレーションとして Cost Management に追加する前に、Google Cloud アカウントで以下のサービスを設定し、Cost Management がメトリクスにアクセスできるようにする必要があります。

- Cost Management Google Cloud プロジェクト。
- [Red Hat Hybrid Cloud Console](#) にデータをエクスポートするための適切なロールを持つ課金サービスアカウントメンバー。
- コストデータを格納する BigQuery データセット。
- コストデータを BigQuery のデータセットに送信する課金エクスポート。

Google Cloud コンソールで以下の手順の一部と、Cost Management ユーザーインターフェイスの手順の一部を完了するため、両方のアプリケーションを Web ブラウザーで開いた状態にしてください。

[Integrations](#) ページから、Google Cloud インテグレーションを Cost Management に追加します。



注記

サードパーティ製品およびドキュメントは変更される可能性があるため、提供されるサードパーティ統合を設定するための手順は一般的な内容であり、公開時点では正しいものです。最新情報は、[Google Cloud Platform のドキュメント](#) を参照してください。

1.1. GOOGLE CLOUD アカウントのインテグレーションとしての追加

Google Cloud アカウントをインテグレーションとして追加できます。Google Cloud インテグレーションを追加すると、Cost Management アプリケーションは Google Cloud アカウントからのコストと使用状況のデータを処理し、表示できるようにします。

前提条件

- Cost Management にデータインテグレーションを追加するには、Cloud Administrator 権限を持つ Red Hat アカウントが必要です。

手順

1. [Red Hat Hybrid Cloud Console](#) から、**Settings Menu**  > **Integrations** をクリックします。
2. **Settings** ページの **Cloud** タブで、**Add integration** をクリックします。

3. **Add a cloud integration** ウィザードで、**Google Cloud** をクラウドプロバイダータイプとして選択し、**Next** をクリックします。
4. インテグレーションの名前を入力します。**Next** をクリックします。
5. **Select application** の手順で、**Cost Management** を選択し、**Next** をクリックします。

1.2. GOOGLE CLOUD プロジェクトの作成

Google Cloud プロジェクトを作成し、コストレポートを収集して Cost Management に送信します。

前提条件

- **resourcemanager.projects.create** 権限で Google Cloud Console にアクセスできる。

手順

1. [Google Cloud Console](#) で **IAM & Admin** → **Create a Project** をクリックします。
2. 表示される新しいページで **Project Name** を入力し、課金アカウントを選択します。
3. **Organization** を選択します。
4. **Location** ボックスに親組織を入力します。
5. **Create** をクリックします。

検証手順

1. Google Cloud Console Dashboard に移動します。
2. プロジェクトがメニューバーにあることを確認します。

関連情報

- プロジェクトの作成に関する詳細は、Google Cloud のドキュメント [プロジェクトの作成と管理](#) を参照してください。

1.3. GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT のロール作成

Cost Management 用のカスタム Identity and Access Management (IAM) ロールは、Google Cloud Platform インテグレーションを有効にするために必要な特定のコスト関連リソースへのアクセスを許可し、他のリソースへのアクセスを禁止します。

前提条件

- 以下の権限で Google Cloud Console にアクセスできる。
 - **resourcemanager.projects.get**
 - **resourcemanager.projects.getIamPolicy**
 - **resourcemanager.projects.setIamPolicy**

- Google Cloud [プロジェクト](#)

手順

1. [Google Cloud Console](#) で、**IAM & Admin**→ **Roles** をクリックします。
2. メニューバーのドロップダウンから Cost Management プロジェクトを選択します。
3. **+ Create role** をクリックします。
4. ロールの **Title**、**Description**、**ID** を入力します。この例では、**customer-data-role** を使用します。
5. **+ ADD PERMISSIONS** をクリックします。
6. **Enter property name or value** フィールドを使用して、カスタムロールのこれら 4 つの権限を検索して選択します。
 - **bigquery.jobs.create**
 - **bigquery.tables.getData**
 - **bigquery.tables.get**
 - **bigquery.tables.list**
7. **Add** をクリックします。
8. **CREATE** をクリックします。

関連情報

- ロールとその使用方法の詳細は、Google Cloud のドキュメント [ロールについて](#) と [カスタムロールの作成と管理](#) を参照してください。

1.4. GOOGLE CLOUD プロジェクトへの課金サービスアカウントメンバーの追加

[Red Hat Hybrid Cloud Console](#) にコストレポートをエクスポートできる課金サービスアカウントメンバーをプロジェクトに作成する必要があります。

前提条件

- 以下の権限で Google Cloud Console にアクセスできる。
 - **resourcemanager.projects.get**
 - **resourcemanager.projects.getIamPolicy**
 - **resourcemanager.projects.setIamPolicy**
- Google Cloud [プロジェクト](#)
- Cost Management Identity and Access Management (IAM) の [ロール](#)

手順

1. [Google Cloud Console](#) で、IAM & Admin→ IAM をクリックします。
2. メニューバーのドロップダウンから Cost Management プロジェクトを選択します。
3. **Add** をクリックします。
4. 作成した IAM ロールを **New principals** フィールドに貼り付けます。

```
billing-export@red-hat-cost-management.iam.gserviceaccount.com
```

5. **Assign roles** セクションで、作成した IAM ロールを割り当てます。この例では、**customer-data-role** を使用します。
6. **SAVE** をクリックします。

検証手順

1. IAM & Admin→ IAM に移動します。
2. 新しいメンバーが正しいロールで存在することを確認します。

関連情報

- ロールとその使用方法の詳細は、Google Cloud のドキュメント [ロールについて](#) と [カスタムロールの作成と管理](#) を参照してください。

1.5. GOOGLE CLOUD BIGQUERY のデータセット作成

Cost Management のための課金データを収集、保存するための BigQuery データセットを作成します。

前提条件

- **bigquery.datasets.create** 権限で Google Cloud Console にアクセスできる。
- Google Cloud [プロジェクト](#)

手順

1. [Google Cloud Console](#) で、**Big Data** → **BigQuery** をクリックします。
2. **Explorer** パネルで Cost Management プロジェクトを選択します。
3. **CREATE DATASET** をクリックします。
4. **Dataset ID** 欄に、データセットの名前を入力します。この例では、**CustomerData** を使用します。
5. **CREATE DATASET** をクリックします。

1.6. GOOGLE CLOUD の課金データの BIGQUERY への書き出し

BigQuery への課金エクスポートを有効にすると、Google Cloud の課金データ (使用量、見積もりコスト、価格データなど) が Cost Management の BigQuery データセットに自動的に送信されます。

前提条件

- 課金アカウント管理者ロールで Google Cloud Console にアクセスできる。
- Google Cloud [プロジェクト](#)
- Cost Management Identity and Access Management (IAM) [ロール](#) を持つ [課金サービスメンバー](#)
- [BigQuery データセット](#)

手順

1. [Google Cloud Console](#) で **Billing** → **Billing export** をクリックします。
2. **Billing export** タブをクリックします。
3. **Detailed usage cost** セクションで **EDIT SETTINGS** をクリックします。
4. ドロップダウンメニューから、作成した Cost Management **Project** と **Billing export dataset** を選択します。
5. **SAVE** をクリックします。

検証手順

1. **Detailed usage cost** セクションに **Enabled** のチェックマークがあり、**Project name** と **Dataset name** が正しいことを確認します。

1.6.1. BigQuery で課金テーブルの表示

収集して Cost Management に送信するメトリクスを確認する場合があります。また、Cost Management でデータが不正な場合や、欠けている場合のトラブルシューティングにも役立ちます。



注記

Google は、課金データを BigQuery のデータセットにエクスポートするのに数時間かかる場合があります。

前提条件

- **bigquery.dataViewer** ロールで Google Cloud コンソールにアクセスできる。

手順

1. [Google Cloud Console](#) で **Big Data** → **BigQuery** へ移動します。
2. **Explorer** パネルで Cost Management **プロジェクト** を選択します。
3. Cost Management データセット下の **gcp_billing_export_v1_XXXXXX_XXXXXX** テーブルをクリックします。
4. **Preview** タブをクリックすると、メトリクスが表示されます。

第2章 フィルタリングされた GOOGLE CLOUD データの COST MANAGEMENT への統合

エクスポートとオブジェクトストレージバケットをコピーし、データをフィルタリングして請求情報のサブセットのみを Red Hat と共有するには、Google Cloud で関数スクリプトを設定します。



注記

Cost Management にデータインテグレーションを追加するには、Cloud Administrator 権限を持つ Red Hat アカウントが必要です。

Google Cloud アカウントを Cost Management インテグレーションとして設定するには、以下を行います。

- Cost Management データ用の Google Cloud プロジェクトを作成します。
- フィルターされたレポートのバケットを作成します。
- データを Cost Management にエクスポートするための適切なロールを持つ課金サービスアカウントメンバーを用意します。
- コストデータを格納する BigQuery データセットを作成します。
- Cost Management データを BigQuery データセットに送信する請求エクスポートを作成します。

Google Cloud コンソールで以下の手順の一部と、Cost Management ユーザーインターフェイスの手順の一部を完了するため、両方のアプリケーションを Web ブラウザーで開いた状態にしてください。



注記

サードパーティー製品およびドキュメントは変更される可能性があるため、提供されるサードパーティー統合を設定するための手順は一般的な内容であり、公開時点では正しいものです。最新情報は、[Google Cloud Platform のドキュメント](#) を参照してください。

[Integrations](#) ページから、Google Cloud インテグレーションを Cost Management に追加します。

2.1. GOOGLE CLOUD アカウントのインテグレーションとしての追加

Google Cloud アカウントをインテグレーションとして追加できます。Google Cloud インテグレーションを追加すると、Cost Management アプリケーションは Google Cloud アカウントからのコストと使用状況のデータを処理し、表示できるようにします。

前提条件

- Cost Management にデータインテグレーションを追加するには、Cloud Administrator 権限を持つ Red Hat アカウントが必要です。

手順

1. [Red Hat Hybrid Cloud Console](#) から、**Settings Menu**  > **Integrations** をクリックします。

2. **Settings** ページの **Cloud** タブで、**Add integration** をクリックします。
3. **Add a cloud integration** ウィザードで、**Google Cloud** をクラウドプロバイダータイプとして選択し、**Next** をクリックします。
4. インテグレーションの名前を入力します。**Next** をクリックします。
5. **Select application** の手順で、**Cost Management** を選択し、**Next** をクリックします。

2.2. GOOGLE CLOUD プロジェクトの作成

Google Cloud プロジェクトを作成し、コストレポートを収集して Cost Management に送信します。

前提条件

- **resourcemanager.projects.create** 権限で Google Cloud Console にアクセスできる。

手順

1. [Google Cloud Console](#) で **IAM & Admin** → **Create a Project** をクリックします。
2. 表示される新しいページで **Project Name** を入力し、課金アカウントを選択します。
3. **Organization** を選択します。
4. **Location** ボックスに親組織を入力します。
5. **Create** をクリックします。
6. Cost Management の **Add a cloud integration** ウィザードの **Project** ページで、**Project ID** を入力します。
7. データを Red Hat に送信する前にデータをフィルタリングするように Google Cloud を設定するには、**I wish to manually customize the data set sent to Cost Management** を選択し、**Next** をクリックします。

検証手順

1. Google Cloud Console Dashboard に移動します。
2. プロジェクトがメニューバーにあることを確認します。

関連情報

- プロジェクトの作成に関する詳細は、Google Cloud のドキュメント [プロジェクトの作成と管理](#) を参照してください。

2.3. GOOGLE CLOUD バケットの作成

フィルタリング処理されたレポート用にバケットを作成します。このレポートは後ほど作成します。バケットは、データを保存するコンテナです。

手順

1. [Google Cloud Console](#) で **Buckets** をクリックします。

2. **Create bucket** をクリックします。
3. バケット情報を入力します。バケットに名前を付けます。この例では、**customer-data** を使用します。
4. **Create** をクリックし、確認ダイアログで **Confirm** をクリックします。
5. Cost Management の **Add a cloud integration** ウィザードの **Create cloud storage bucket** ページで、**Cloud storage bucket name** を入力します。

関連情報

- バケットの作成の詳細は、Google Cloud のドキュメント [バケットの作成](#) を参照してください。

2.4. GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT のロール作成

Cost Management 用のカスタム Identity and Access Management (IAM) ロールは、Google Cloud Platform インテグレーションを有効にするために必要な特定のコスト関連リソースへのアクセスを許可し、他のリソースへのアクセスを禁止します。

前提条件

- 以下の権限で Google Cloud Console にアクセスできる。
 - **resourcemanager.projects.get**
 - **resourcemanager.projects.getIamPolicy**
 - **resourcemanager.projects.setIamPolicy**
- Google Cloud [プロジェクト](#)

手順

1. [Google Cloud Console](#) で、**IAM & Admin** → **Roles** をクリックします。
2. メニューバーのドロップダウンから Cost Management プロジェクトを選択します。
3. **+ Create role** をクリックします。
4. ロールの **Title**、**Description**、**ID** を入力します。この例では、**customer-data-role** を使用します。
5. **+ ADD PERMISSIONS** をクリックします。
6. **Enter property name or value** フィールドを使用して、カスタムロールのこれら 4 つの権限を検索して選択します。
 - **storage.objects.get**
 - **storage.objects.list**
 - **storage.buckets.get**

7. **Add** をクリックします。
8. **CREATE** をクリックします。

関連情報

- ロールとその使用方法の詳細は、Google Cloud のドキュメント [ロールについて](#) と [カスタムロールの作成と管理](#) を参照してください。

2.5. GOOGLE CLOUD プロジェクトへの課金サービスアカウントメンバーの追加

[Red Hat Hybrid Cloud Console](#) にコストレポートをエクスポートできる課金サービスアカウントメンバーをプロジェクトに作成する必要があります。

前提条件

- 以下の権限で Google Cloud Console にアクセスできる。
 - `resourcemanager.projects.get`
 - `resourcemanager.projects.getIamPolicy`
 - `resourcemanager.projects.setIamPolicy`
- Google Cloud [プロジェクト](#)
- Cost Management Identity and Access Management (IAM) の [ロール](#)

手順

1. [Google Cloud Console](#) で、**IAM & Admin**→**IAM** をクリックします。
2. メニューバーのドロップダウンから Cost Management プロジェクトを選択します。
3. **Add** をクリックします。
4. 作成した IAM ロールを **New principals** フィールドに貼り付けます。

```
billing-export@red-hat-cost-management.iam.gserviceaccount.com
```

5. **Assign roles** セクションで、作成した IAM ロールを割り当てます。この例では、**customer-data-role** を使用します。
6. **SAVE** をクリックします。

検証手順

1. **IAM & Admin**→**IAM** に移動します。
2. 新しいメンバーが正しいロールで存在することを確認します。

関連情報

- ロールとその使用方法の詳細は、Google Cloud のドキュメント [ロールについて](#) と [カスタムロールの作成と管理](#) を参照してください。

2.6. GOOGLE CLOUD BIGQUERY のデータセット作成

Cost Management のための課金データを収集、保存するための BigQuery データセットを作成します。

前提条件

- **bigquery.datasets.create** 権限で Google Cloud Console にアクセスできる。
- Google Cloud [プロジェクト](#)

手順

1. [Google Cloud Console](#) で、**Big Data** → **BigQuery** をクリックします。
2. **Explorer** パネルで Cost Management プロジェクトを選択します。
3. **CREATE DATASET** をクリックします。
4. **Dataset ID** 欄に、データセットの名前を入力します。この例では、**CustomerFilteredData** を使用します。
5. **CREATE DATASET** をクリックします。

2.7. GOOGLE CLOUD の課金データの BIGQUERY への書き出し

BigQuery への課金エクスポートを有効にすると、Google Cloud の課金データ (使用量、見積もりコスト、価格データなど) が Cost Management の BigQuery データセットに自動的に送信されます。

前提条件

- **課金アカウント管理者** ロールで Google Cloud Console にアクセスできる。
- Google Cloud [プロジェクト](#)
- Cost Management Identity and Access Management (IAM) [ロール](#) を持つ [課金サービスメンバー](#)
- [BigQuery データセット](#)

手順

1. [Google Cloud Console](#) で **Billing** → **Billing export** をクリックします。
2. **Billing export** タブをクリックします。
3. **Detailed usage cost** セクションで **EDIT SETTINGS** をクリックします。
4. ドロップダウンメニューから、作成した Cost Management **Project** と **Billing export dataset** を選択します。
5. **SAVE** をクリックします。

検証手順

1. **Detailed usage cost** セクションに **Enabled** のチェックマークがあり、**Project name** と **Dataset name** が正しいことを確認します。

2.8. フィルターされたデータをストレージバケットに POST する関数の作成

データをフィルターする関数を作成し、Red Hat と共有するために作成したストレージアカウントにデータを追加します。サンプル Python スクリプトを使用して、Red Hat の経費に関連するコストエクスポートからコストデータを収集し、ストレージアカウントに追加できます。このスクリプトは、BigQuery で作成したコストデータをフィルタリングし、Red Hat 以外の情報を削除してから、**.csv** ファイルを作成し、作成したバケットに保存して、データを Red Hat に送信します。

手順

1. [Google Cloud Console](#) で、**Secret** を検索し、**Secret manager** の結果を選択して、関数に認証情報を保存せずに Red Hat で関数を認証するためのシークレットを設定します。
 - a. Secret Manager ページで **Create Secret** をクリックします。
 - b. シークレットに名前を付け、Red Hat のユーザー名を追加して、**Create Secret** をクリックします。
 - c. この手順を繰り返して、Red Hat パスワードのシークレットを保存します。
2. [Google Cloud Console](#) 検索バーで、**functions** を検索し、**Cloud Functions** 結果を選択します。
3. [Cloud Functions](#) ページで、**Create function** をクリックします。
4. 関数に名前を付けます。この例では、**customer-data-function** を使用します。
5. **Trigger** セクションで、**Save** をクリックして HTTP Trigger タイプを確定します。
6. **Runtime, build, connections and security settings** で、**セキュリティとイメージリポジトリ** をクリックし、作成したシークレットを参照し、**Done** をクリックして、**Next** をクリックします。
7. [Cloud Functions Code](#) ページで、ランタイムを **Python 3.9** に設定します。
8. **requirements.txt** ファイルを開きます。以下の行をファイルの最後に貼り付けます。

```
requests
google-cloud-bigquery
google-cloud-storage
```

9. **main.py** ファイルを開きます。
 - a. **Entry Point** を **get_filtered_data** に設定します。
 - b. 以下の python スクリプトを貼り付けます。 **# Required vars to update** とマークされたセクションの値を変更して、環境の値に更新します。

```
import csv
import datetime
```

```

import uuid
import os
import requests
from google.cloud import bigquery
from google.cloud import storage
from itertools import islice
from dateutil.relativedelta import relativedelta

query_range = 5
now = datetime.datetime.now()
delta = now - relativedelta(days=query_range)
year = now.strftime("%Y")
month = now.strftime("%m")
day = now.strftime("%d")
report_prefix="{year}/{month}/{day}/{uuid.uuid4()}"

# Required vars to update
USER = os.getenv('username')      # Cost management username
PASS = os.getenv('password')      # Cost management password
INTEGRATION_ID = "<integration_id>" # Cost management integration_id
BUCKET = "<bucket>"                # Filtered data GCP Bucket
PROJECT_ID = "<project_id>"        # Your project ID
DATASET = "<dataset>"              # Your dataset name
TABLE_ID = "<table_id>"           # Your table ID

gcp_big_query_columns = [
    "billing_account_id",
    "service.id",
    "service.description",
    "sku.id",
    "sku.description",
    "usage_start_time",
    "usage_end_time",
    "project.id",
    "project.name",
    "project.labels",
    "project.ancestry_numbers",
    "labels",
    "system_labels",
    "location.location",
    "location.country",
    "location.region",
    "location.zone",
    "export_time",
    "cost",
    "currency",
    "currency_conversion_rate",
    "usage.amount",
    "usage.unit",
    "usage.amount_in_pricing_units",
    "usage.pricing_unit",
    "credits",
    "invoice.month",
    "cost_type",
    "resource.name",
    "resource.global_name",

```

```

]
table_name = ".".join([PROJECT_ID, DATASET, TABLE_ID])

BATCH_SIZE = 200000

def batch(iterable, n):
    """Yields successive n-sized chunks from iterable"""
    it = iter(iterable)
    while chunk := tuple(islice(it, n)):
        yield chunk

def build_query_select_statement():
    """Helper to build query select statement."""
    columns_list = gcp_big_query_columns.copy()
    columns_list = [
        f"TO_JSON_STRING({col})" if col in ("labels", "system_labels", "project.labels",
"credits") else col
        for col in columns_list
    ]
    columns_list.append("DATE(_PARTITIONTIME) as partition_date")
    return ",".join(columns_list)

def create_reports(query_date):
    query = f"SELECT {build_query_select_statement()} FROM {table_name} WHERE
DATE(_PARTITIONTIME) = {query_date} AND sku.description LIKE '%RedHat%' OR
sku.description LIKE '%Red Hat%' OR service.description LIKE '%Red Hat%' ORDER
BY usage_start_time"
    client = bigquery.Client()
    query_job = client.query(query).result()
    column_list = gcp_big_query_columns.copy()
    column_list.append("partition_date")
    daily_files = []
    storage_client = storage.Client()
    bucket = storage_client.bucket(BUCKET)
    for i, rows in enumerate(batch(query_job, BATCH_SIZE)):
        csv_file = f"{report_prefix}/{query_date}_part_{str(i)}.csv"
        daily_files.append(csv_file)
        blob = bucket.blob(csv_file)
        with blob.open(mode='w') as f:
            writer = csv.writer(f)
            writer.writerow(column_list)
            writer.writerows(rows)
    return daily_files

def post_data(files_list):
    # Post CSV's to console.redhat.com API
    url = "https://console.redhat.com/api/cost-management/v1/ingress/reports/"
    json_data = {"source": INTEGRATION_ID, "reports_list": files_list, "bill_year": year,
"bill_month": month}
    resp = requests.post(url, json=json_data, auth=(USER, PASS))
    return resp

def get_filtered_data(request):
    files_list = []
    query_dates = [delta + datetime.timedelta(days=x) for x in range(query_range)]
    for query_date in query_dates:

```

```
files_list += create_reports(query_date.date())
resp = post_data(files_list)
return f'Files posted! {resp}'
```

10. **Deploy** をクリックします。

2.9. フィルターされたデータをストレージバケットに POST する関数のトリガー

スケジューラージョブを作成して、フィルターされたデータをスケジュールに従って Red Hat に送信するために作成した関数を実行します。

手順

- 作成した関数の **Trigger URL** をコピーして、コストレポートを Post します。Google Cloud Scheduler に追加する必要があります。
 - [Google Cloud Console](#) で、**functions** を検索し、**Cloud Functions** の結果を選択します。
 - Cloud Functions** ページで、関数を選択し、Trigger タブをクリックします。
 - HTTP セクションで、**Copy to clipboard** をクリックします。
- スケジューラージョブを作成します。[Google Cloud Console](#) で、**cloud scheduler** を検索し、**Cloud Scheduler** の結果を選択します。
- Create job** をクリックします。
 - スケジューラージョブに名前を付けます。この例では、**CustomerFilteredDataSchedule** を使用します。
 - Frequency** フィールドに、関数を実行するタイミングの cron 式を設定します。この例では、**09***** を使用して、関数を毎日 9 AM で実行します。
 - タイムゾーンを設定し、**Continue** をクリックします。
- 次のページで実行を設定します。
 - Target type** フィールドで **HTTP** を選択します。
 - URL フィールドに、コピーした Trigger URL を貼り付けます。
 - body フィールドに、関数に渡す以下のコードを貼り付けてトリガーします。

```
{"name": "Scheduler"}
```
 - Auth header フィールドで、**Add OIDC token** を選択します。
 - Service account** フィールドをクリックし、**Create** をクリックしてスケジューラージョブのサービスアカウントおよびロールを作成します。
- Service account details** ステップで、サービスアカウントに名前を付けます。この例では、**scheduler-service-account** を使用します。デフォルトの **Service account ID** を受け入れ、**Create and Continue** をクリックします。

- a. Grant this service account access to project で、アカウントに2つのロールを選択します。
 - b. **ADD ANOTHER ROLE** をクリックしてから、**Cloud Scheduler Job Runner** および **Cloud Functions Invoker** を検索して選択します。
 - c. **Continue** をクリックします。
 - d. **Done** をクリックして、サービスアカウントの作成を終了します。
6. プロジェクトのサービスアカウントページで、作業していたスケジューラージョブを選択します。この例では、名前は **scheduler-service-account** です。
 7. **Configure the execution** ページで **Service account** フィールドを選択し、作成した **scheduler-service-account** を選択します。
 8. **Continue** をクリックしてから **Create** をクリックします。

第3章 コストを管理するための次のステップ

OpenShift Container Platform と Google Cloud インテグレーションを追加すると、[Cost Management Overview](#) ページで、コストデータが **OpenShift** タブと **Infrastructure** タブに分類されます。コストデータのさまざまなビューを切り替えるには、**Perspective** を選択します。

グローバルナビゲーションメニューを使用して、クラウドプロバイダーごとのコストに関する追加の詳細を表示することもできます。

関連情報

- [Cost Management への OpenShift Container Platform データの統合](#)
- [Amazon Web Services \(AWS\) データの Cost Management への統合](#)
- [Microsoft Azure データの Cost Management への統合](#)
- [Oracle Cloud データの Cost Management への統合](#)

3.1. COST MANAGEMENT リソースへのアクセス制限

Cost Management でインテグレーションを追加して設定した後、コストデータとリソースへのアクセスを制限できます。

ユーザーがすべてのコストデータにアクセスできる状況は避ける必要がある場合もあります。代わりに、プロジェクトまたは組織に固有のデータにだけアクセスできるようにユーザーにアクセス権を付与できます。ロールベースのアクセス制御を使用すると、Cost Management レポートでのリソースの表示を制限できます。たとえば、ユーザーのビューを環境全体ではなく、AWS インテグレーションのみに制限できます。

アクセスを制限する方法の詳細は、[Cost Management リソースへのアクセス制限](#) を参照してください。

3.2. インテグレーションのタグ付けの設定

Cost Management アプリケーションは、タグを使用してクラウドとインフラストラクチャーのコストを追跡します。タグは、OpenShift ではラベルとも呼ばれます。

Cost Management でタグを調整して、リソースをフィルタリングおよび属性化し、コスト別にリソースを整理し、クラウドインフラストラクチャーのさまざまな部分にコストを割り当てることができます。



重要

タグとラベルは、インテグレーションでのみ直接設定できます。Cost Management でアクティブ化するタグの選択はできますが、Cost Management アプリケーションでタグとラベルの編集はできません。

以下のトピックに関する詳細は、[タグ付けを使用したコストデータの管理](#) を参照してください。

- コストデータの表示を整理するためのタグ付けストラテジーを計画する
- Cost Management がタグを関連付ける方法を理解する

- インテグレーションでタグとラベルを設定する

3.3. コストを正確にレポートするためのコストモデルの設定

Cost Management でコストと使用量のデータを収集するようにインテグレーションを設定したので、価格をメトリクスと使用量に関連付けるコストモデルを設定できます。

コストモデルは、Cost Management において、原価とメトリクスを使用してコスト計算を定義するためのフレームワークです。コストモデルが生成するコストの記録と分類、および特定の顧客、ビジネスユニット、またはプロジェクトに対する配分を行えます。

[Cost Models](#) では、次のタスクを完了できます。

- コストを、インフラストラクチャーコストまたは補足コストとして分類する
- OpenShift ノードおよびクラスターの月額コストを取得する
- 追加のサポートコストを考慮して利潤を適用する

コストモデルの設定方法は [コストモデルの使用](#) を参照してください。

3.4. COST EXPLORER を使用したコストの可視化

Cost Management の [Cost Explorer](#) を使用して、時間スケールのコストと使用状況情報のカスタムグラフを作成し、最終的にコストをより適切に可視化して解釈できるようにします。

次のトピックに関する詳細は、[Cost Explorer を使用したコストの可視化](#) を参照してください。

- Cost Explorer を使用して異常なイベントを特定する。
- 時間の経過とともにコストデータがどのように変化するかを理解する。
- コストおよび使用状況データのカスタムバーチャートを作成する。
- カスタムコストデータテーブルをエクスポートする。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するフィードバックをお寄せください。いただいたご要望に迅速に対応できるよう、できるだけ詳細にご記入ください。

前提条件

- Red Hat カスタマーポータルにログインしている。

手順

フィードバックを送信するには、以下の手順を実施します。

1. [Create Issue](#) にアクセスします。
2. **Summary** テキストボックスに、問題または機能拡張に関する説明を入力します。
3. **Description** テキストボックスに、問題または機能拡張のご要望に関する詳細を入力します。
4. **Reporter** テキストボックスに、お客様のお名前を入力します。
5. **Create** ボタンをクリックします。

これによりドキュメントに関するチケットが作成され、適切なドキュメントチームに転送されます。フィードバックをご提供いただきありがとうございました。