Hybrid committed spend 1-latest

将 Google Cloud 数据集成到混合承诺的花费中

了解如何添加和配置 Google Cloud 集成

Last Updated: 2024-05-17

Hybrid committed spend 1-latest 将 Google Cloud 数据集成到混合承诺的花费中

了解如何添加和配置 Google Cloud 集成

法律通告

摘要

您可以将 Google Cloud Platform 集成添加到混合提交的成本。

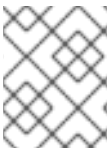
目录

前言

要将 Google Cloud 帐户添加到混合提交花费中，您必须将它添加为 Red Hat Hybrid Cloud Console 用户界面的集成，并配置 Google Cloud 以提供指标。您可以自动发送数据，或者配置功能脚本来复制混合提交消耗的成本导出和对象存储桶，并过滤您的数据，以与红帽共享您的账单数据子集。

第 1 章 创建 GOOGLE CLOUD 集成

要将 Google Cloud 帐户添加到混合提交花费中，您必须配置 Google Cloud 帐户以提供指标，然后将它添加为 Red Hat Hybrid Cloud Console 用户界面的集成。



注意

在向混合提交的花费添加集成之前，您必须拥有具有 Cloud Administrator 权限的红帽帐户用户。

要将 Google Cloud 帐户配置为混合提交合并，您必须完成以下任务：

- 为您的混合提交数据创建一个 Google Cloud 项目。
- 为过滤的报告创建存储桶。
- 账单服务帐户成员具有正确的角色，将您的数据导出到混合提交的成本。
- 创建一个 BigQuery 数据集来包含成本数据。
- 创建一个账单导出，将混合提交的数据发送到 BigQuery 数据集。

在 Google Cloud 控制台中完成一些以下步骤，并在混合提交的用户界面中完成一些步骤，使这两个应用程序在 Web 浏览器中保持打开。

注意

由于第三方产品和文档可能会改变，因此配置第三方集成的说明一般是常规的，并在发布时进行更正。有关最新信息，请参阅 Google Cloud Platform 文档。

将 Google Cloud 集成添加到 Integrations 页面 的混合提交花费。

1.1. 将 GOOGLE CLOUD 帐户添加为集成

您可以将 Google Cloud 帐户添加为集成。添加 Google Cloud 集成后，混合提交会将应用程序处理 Google Cloud 帐户的成本和使用情况数据，并使其可以被查看。

先决条件

- 要向成本管理添加数据集成，您必须拥有具有 Cloud Administrator 权限的红帽帐户。

流程

1. 在 Red Hat Hybrid Cloud Console 中，点 **Settings Menu** > **Integrations**。
2. 在 **Settings** 页面中，在 **Cloud** 选项卡中点 **Add integration**。
3. 在 **Add a cloud integration** 向导中，选择 **Google Cloud** 作为云供应商类型，然后点 **Next**。
4. 输入您的集成名称。点击 **Next**。
5. 在 **Select application** 步骤中，选择 **Hybrid committed spend** 并点 **Next**。

1.2. 创建 GOOGLE CLOUD 项目

创建一个 Google Cloud 项目，以收集并发送您的成本报告到混合提交成本。

先决条件

- 使用 **resourcemanager.projects.create** 权限访问 Google Cloud Console

流程

1. 在 Google Cloud Console 中，点 **IAM & Admin→ Create a Project**。
2. 在出现的新页面中输入项目名称并选择您的账单帐户。
3. 选择 **Organization**。
4. **在 Location 框中输入父机构。**
5. **点 Create。**
6. **在混合的 Add a cloud integration向导中，在 Project 页面中输入您的项目 ID。**
7. **要自动将默认数据发送到红帽，请选择 I am OK，将默认数据集发送到混合提交花费，然后单击 Next。**

验证步骤

1. 进入到 Google Cloud Console Dashboard
2. 验证项目位于菜单栏中。

其他资源

- **有关创建项目的更多信息，请参阅 Google Cloud *文档创建和管理项目*。**

1.3. 创建 GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT 角色

混合承诺的自定义 Identity and Access Management (IAM)角色可以访问启用 Google Cloud Platform 集成所需的特定成本相关资源，并禁止访问其他资源。

先决条件

- **使用这些权限访问 Google Cloud 控制台：**
  - **resourcemanager.projects.get**
  - **resourcemanager.projects.getIamPolicy**
  - **resourcemanager.projects.setIamPolicy**
- Google Cloud 项目

流程

1. **在 Google Cloud Console 中，点 IAM & Admin→ Roles。**

2. 从菜单栏中的下拉菜单中选择混合所提交的花项目。
3. 单击 **+ Create role**。
4. 为角色输入 Title、Description 和 ID。在本例中，使用 **customer-data-role**。
5. 单击 **+ ADD PERMISSIONS**。
6. 使用 Enter property name 或 value字段搜索，并为自定义角色选择这四个权限：
   - **bigquery.jobs.create**
   - **bigquery.tables.getData**
   - **bigquery.tables.get**
   - **bigquery.tables.list**
7. 单击 **ADD**。
8. 点 **CREATE**。
9. 在混合的 Add a cloud integration向导中，在 Create IAM 角色页面中点 Next。

其他资源

- 有关角色及其用法的更多信息，请参阅 Google Cloud 文档 *了解角色以及创建和管理自定义角色*。

1.4. 将账单服务帐户成员添加到您的 GOOGLE CLOUD 项目中

您必须创建一个账单服务帐户成员，该成员可将成本报告导出到项目中的 红帽混合云控制台。

先决条件

- 使用这些权限访问 Google Cloud 控制台：
  - **resourcemanager.projects.get**
  - **resourcemanager.projects.getIamPolicy**
  - **resourcemanager.projects.setIamPolicy**
- Google Cloud 项目
- 混合使用 Identity and Access Management (IAM) 角色

流程

1. 在 Google Cloud Console 中，点 IAM & Admin→IAM。
2. 从菜单栏中的下拉菜单中选择混合所提交的花项目。
3. 单击 **ADD**。
4. 将您创建的 IAM 角色粘贴到 New principals 字段中：
   -

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

5. 在 Assign roles 部分中，分配您创建的 IAM 角色。在本例中，使用**customer-data-role**。
6. 点 **SAVE**。
7. 在混合的 Add a cloud integration向导中，在 Assign access 页面上，单击 Next。

验证步骤

1. 导航到 IAM & Admin→IAM。
2. 验证新成员是否存在正确的角色。

其他资源

- 有关角色及其用法的更多信息，请参阅 Google Cloud 文档 *了解角色以及创建和管理自定义角色*。

1.5. 创建 GOOGLE CLOUD BIGQUERY 数据集

创建一个 BigQuery 数据集来收集和存储计费数据以进行混合提交成本。

先决条件

- 使用 **bigquery.datasets.create** 权限访问 Google Cloud Console
- Google Cloud 项目

流程

1. 在 Google Cloud Console 中，单击 Big Data →BigQuery。
2. 在 Explorer 面板中，选择混合已提交花的项目。
3. 单击 **CREATE DATASET**。
4. 在 Dataset ID 字段中输入 dataset 的名称。在本例中，使用**CustomerData**。
5. 单击 **CREATE DATASET**。

1.6. 将 GOOGLE CLOUD BILLING 数据导出到 BIGQUERY

启用计费导出到 BigQuery，将 Google Cloud 计费数据（如使用、成本估算和定价数据）自动发送到混合提交的 BigQuery 数据集。

先决条件

- 使用 Billing Account Administrator 角色访问 Google Cloud 控制台
- Google Cloud 项目
- 带有成本管理 Identity and Access Management (IAM) 角色的账单服务成员

- bigquery 数据集

流程

1. 在 Google Cloud Console 中，单击 Billing → Billing export。
2. 单击 Billing export 选项卡。
3. 在详细信息 使用成本 部分中点 EDIT SETTINGS。
4. 选择您在下拉菜单中创建的混合提交 的项目和 Billing 导出数据集。
5. 点 SAVE。
6. 在提交中的 Add a cloud integration向导的 Billing export 页面中，单击 Next。
7. 在混合的 Add a cloud integration向导中，在 Review details 页面中，点 Add。

验证步骤

1. 在详细信息 使用成本 部分中，验证标有 Enabled 的复选标记，其具有正确的 项目名称和 数据集名称。

第 2 章 将过滤的 GOOGLE CLOUD 数据集成到混合提交成本中

您可以在 Google Cloud 中配置功能脚本，以复制成本导出和对象存储桶，混合提交花费可以访问并过滤您的数据，以便与红帽共享您的账单数据子集。

注意

在向混合提交的花费添加集成之前，您必须拥有具有 Cloud Administrator 权限的红帽帐户用户。

要将 Google Cloud 帐户配置为混合提交合并，您必须完成以下任务：

- 为您的混合提交数据创建一个 Google Cloud 项目。
- 为过滤的报告创建存储桶。
- 具有具有正确角色的账单服务帐户成员，将您的数据导出到混合提交的成本。
- 创建一个 BigQuery 数据集来包含成本数据。
- 创建一个账单导出，将混合提交的数据发送到 BigQuery 数据集。

由于您将在 Google Cloud 控制台中完成一些以下步骤，因此在混合提交的用户界面中完成一些步骤，因此这两个应用程序都保持在 Web 浏览器中打开。

注意

由于第三方产品和文档可能会改变，因此配置第三方集成的说明一般是常规的，并在发布时进行更正。有关最新信息，请参阅 Google Cloud Platform 文档。

将 Google Cloud 集成添加到 Integrations 页面的混合提交成本。

2.1. 将 GOOGLE CLOUD 帐户添加为集成

您可以将 Google Cloud 帐户添加为集成。添加 Google Cloud 集成后，混合提交会将应用程序处理 Google Cloud 帐户的成本和使用情况数据，并使其可以被查看。

先决条件

- 要向成本管理添加数据集成，您必须拥有具有 Cloud Administrator 权限的红帽帐户。

流程

1. 在 Red Hat Hybrid Cloud Console 中，点 Settings Menu  > Integrations。
2. 在 Settings 页面中，在 Cloud 选项卡中点 Add integration。
3. 在 Add a cloud integration 向导中，选择 Google Cloud 作为云供应商类型，然后点 Next。
4. 输入您的集成名称。点击 Next。
5. 在 Select application 步骤中，选择 Hybrid committed spend 并点 Next。

2.2. 创建 GOOGLE CLOUD 项目

创建一个 Google Cloud 项目，以收集并发送您的成本报告到混合提交成本。

先决条件

- 使用 resourcemanager.projects.create 权限访问 Google Cloud Console

流程

1. 在 Google Cloud Console 中，点 IAM & Admin→Create a Project。
2. 在出现的新页面中输入项目名称并选择您的账单帐户。
3. 选择 Organization。
4. 在 Location 框中输入父机构。
5. 点 Create。
6. 在混合的 Add a cloud integration向导中，在 Project 页面中输入您的项目 ID。
7. 要将 Google Cloud 配置为在将数据发送到红帽之前过滤数据，请选择 I want to manually custom the data set sent to hybrid committed spend，点 Next。

验证步骤

1. 进入到 Google Cloud Console Dashboard
2. 验证项目位于菜单栏中。

其他资源

- 有关创建项目的更多信息，请参阅 Google Cloud *文档创建和管理项目*。

2.3. 创建 GOOGLE CLOUD 存储桶

为稍后创建的过滤报告创建存储桶。bucket 是存储数据的容器。

流程

1. 在 Google Cloud Console 中，点 Buckets。
2. 点 Create bucket。
3. 输入存储桶信息。为您的存储桶命名。在本例中，使用 customer-data。
4. 单击 Create，然后在确认对话框中单击 Confirm。
5. 在混合云的 Add a cloud integration向导中，在 Create cloud storage bucket页面中，输入 您的云存储桶名称。

其他资源

- 有关创建存储桶的更多信息，请参阅 Google Cloud 文档中的 创建存储桶。

2.4. 创建 GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT 角色

混合承诺的自定义 Identity and Access Management (IAM)角色可以访问启用 Google Cloud Platform 集成所需的特定成本相关资源，并禁止访问其他资源。

先决条件

- 使用这些权限访问 Google Cloud 控制台：
  - **resourcemanager.projects.get**
  - **resourcemanager.projects.getIamPolicy**
  - **resourcemanager.projects.setIamPolicy**
- Google Cloud 项目

流程

1. 在 Google Cloud Console 中，点 IAM & Admin→Roles。
2. 从菜单栏中的下拉菜单中选择混合所提交的花项目。
3. 单击 **+ Create role**。
4. 为角色输入 Title、Description 和 ID。在本例中，使用 **customer-data-role**。
5. 单击 **+ ADD PERMISSIONS**。
6. 使用 Enter property name 或 value字段搜索，并为自定义角色选择这四个权限：
   - **storage.objects.get**
   - **storage.objects.list**
   - **storage.buckets.get**
7. 单击 **ADD**。
8. 点 **CREATE**。
9. 在混合的 Add a cloud integration向导中，在 Create IAM 角色页面中点 Next。

其他资源

- 有关角色及其用法的更多信息，请参阅 Google Cloud 文档 *了解 角色以及创建和管理自定义角色*。

2.5. 将账单服务帐户成员添加到您的 GOOGLE CLOUD 项目中

您必须创建一个账单服务帐户成员，该成员可将成本报告导出到项目中的 红帽混合云控制台。

先决条件

- 使用这些权限访问 Google Cloud 控制台：

- resourcemanager.projects.get
- resourcemanager.projects.getIamPolicy
- resourcemanager.projects.setIamPolicy

- Google Cloud 项目
- 混合使用 Identity and Access Management (IAM) 角色

流程

1. 在 Google Cloud Console 中，点 IAM & Admin→IAM。
2. 从菜单栏中的下拉菜单中选择混合所提交的花项目。
3. 单击 **ADD**。
4. 将您创建的 IAM 角色粘贴到 New principals 字段中：

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

5. 在 Assign roles 部分中，分配您创建的 IAM 角色。在本例中，使用**customer-data-role**。
6. 点 **SAVE**。
7. 在混合的 Add a cloud integration向导中，在 Assign access 页面上，单击 Next。

验证步骤

1. 导航到 IAM & Admin→IAM。
2. 验证新成员是否存在正确的角色。

其他资源

- 有关角色及其用法的更多信息，请参阅 Google Cloud 文档 *了解 角色以及创建和管理自定义角色*。

2.6. 创建 GOOGLE CLOUD BIGQUERY 数据集

创建一个 BigQuery 数据集来收集和存储计费数据以进行混合提交成本。

先决条件

- 使用 **bigquery.datasets.create** 权限访问 Google Cloud Console
- Google Cloud 项目

流程

1. 在 Google Cloud Console 中，单击 Big Data →BigQuery。
2. 在 Explorer 面板中，选择混合已提交花的项目。

3. 单击 **CREATE DATASET**。
4. 在 Dataset ID 字段中输入 dataset 的名称。在本例中，使用**CustomerFilteredData**。
5. 单击 **CREATE DATASET**。

2.7. 将 GOOGLE CLOUD BILLING 数据导出到 BIGQUERY

启用计费导出到 BigQuery，将 Google Cloud 计费数据（如使用、成本估算和定价数据）自动发送到混合提交的 BigQuery 数据集。

先决条件

- 使用 Billing Account Administrator 角色访问 Google Cloud 控制台
- Google Cloud 项目
- 带有成本管理 Identity and Access Management (IAM) 角色的账单服务成员
- bigquery 数据集

流程

1. 在 Google Cloud Console 中，单击 Billing → Billing export。
2. 单击 Billing export 选项卡。
3. 在详细信息 使用成本 部分中点 **EDIT SETTINGS**。
4. 选择您在下拉菜单中创建的混合提交 的项目和 Billing 导出数据集。
5. 点 **SAVE**。
6. 在提交中的 Add a cloud integration向导的 Billing export 页面中，单击 Next。
7. 在混合的 Add a cloud integration向导中，在 Review details 页面中，点 Add。

验证步骤

1. 在详细信息 使用成本 部分中，验证标有 Enabled 的复选标记，其具有正确的 项目名称和 数据集名称。

2.8. 创建功能，将过滤的数据发布到您的存储桶

创建一个过滤数据的功能，并将其添加到您创建的存储帐户中，以便与红帽共享。您可以使用示例 Python 脚本从与红帽费用相关的成本导出中收集成本数据，并将其添加到存储帐户中。此脚本会过滤您使用 BigQuery 创建的成本数据，删除非红帽信息，然后创建 .csv 文件，将其存储在您创建的存储桶中，并将数据发送到红帽。

流程

1. 在 Google Cloud Console 中，搜索 **secret** 并选择 Secret Manager 结果来设置一个 secret，以使用 Red Hat 验证您的功能，而无需将凭证存储在您的功能中。
    a. 在 Secret Manager 页面中，点 Create Secret。

b. 为您的 secret 命名，添加您的红帽用户名，然后点 Create Secret。

c. 重复此过程，为您的红帽密码保存 secret。

2. 在 Google Cloud Console 搜索栏中，搜索功能并选择 Cloud Functions 结果。

3. 在 Cloud Functions 页面上，单击 Create function。

4. 将函数命名为。在本例中，使用 customer-data-function。

5. 在 Trigger 部分，点 Save 接受 HTTP Trigger 类型。

6. 在 Runtime, build, connections and security settings 中，单击 Security and image repository，引用您创建的 secret，点 Done，然后点 Next。

7. 在 Cloud Functions Code 页面中，将运行时设置为 Python 3.9。

8. 打开 requirements.txt 文件。将下面几行粘贴到文件末尾。

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

9. 打开 main.py 文件。

a. 将 Entry Point 设置为 get_filtered_data。

b. 粘贴以下 python 脚本。将标记为 # Required vars 的部分中 的值更改为您的环境的值。

```
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=f"{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. 点 Deploy。

2.9. 触发您的功能，将过滤的数据发布到您的存储桶

创建一个调度程序作业来运行您创建的功能，以便按时间表向红帽发送过滤的数据。

流程

1. 复制您创建的功能的 Trigger URL，以发布成本报告。您需要将它添加到 Google Cloud 调度程序中。
   a. 在 Google Cloud Console 中，搜索功能并选择 Cloud Functions 结果。
   b. 在 Cloud Functions 页面中，选择您的功能，然后点 Trigger 选项卡。
   c. 在 HTTP 部分中，点 Copy to clipboard。
2. 创建调度程序作业。在 Google Cloud Console 中，搜索 云调度程序 并选择 Cloud Scheduler 结果。
3. 点 Create job。
   a. 为您的调度程序作业命名。在本例中，使用 CustomerFilteredDataSchedule。
   b. 在 Frequency 字段中，为希望该函数运行时设置 cron 表达式。在本例中，使用 09*** 在每天上午 9 点运行函数。
   c. 设置时区，然后单击 Continue。
4. 在下一页中配置执行。

a. 在 Target type 字段中，选择 HTTP。

b. 在 URL 字段中，粘贴您复制的 Trigger URL。

c. 在 body 字段中，粘贴传递给函数的以下代码来触发它。

```
{"name": "Scheduler"}
```

d. 在 Auth 标头字段中，选择 Add OIDC token。

e. 点 Service account 字段，点 Create 为调度程序作业创建服务帐户和角色。

5. 在 Service account details 步骤中，命名您的服务帐户。在本例中，使用 scheduler-service-account。接受 默认服务帐户 ID，再点 Create and Continue。

a. 在 Grand this service account access to project 中，为您的帐户选择两个角色。

b. 单击 ADD ANOTHER ROLE，然后搜索并选择 Cloud Scheduler Job Runner 和 Cloud Functions Invoker。

c. 点 Continue。

d. 点 Done 完成服务帐户创建。

6. 在项目页面的服务账户上，选择您正在使用的调度程序作业。在本例中，名称是 scheduler-service-account。

7. 在 Configure the execution 页面中，选择 Service account 字段，再选择您刚才创建的 scheduler-service-account。

8. 单击 Continue，然后单击 Create。

对红帽文档提供反馈

如果您发现了错误，或者对如何改进这些指南有建议，请在 成本管理 JIRA 板 中创建一个问题并添加 Documentation 标签。

非常感谢您的反馈意见！