



Hybrid committed spend 1-latest

Integrating Google Cloud data into hybrid committed spend

Learn how to add and configure your Google Cloud integrations

Hybrid committed spend 1-latest Integrating Google Cloud data into hybrid committed spend

Learn how to add and configure your Google Cloud integrations

Legal Notice

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.

Abstract

You can add a Google Cloud Platform integration to hybrid committed spend.

Table of Contents

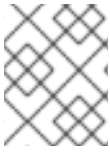
PREFACE	3
CHAPTER 1. CREATING A GOOGLE CLOUD INTEGRATION	4
1.1. ADDING YOUR GOOGLE CLOUD ACCOUNT AS AN INTEGRATION	4
1.2. CREATING A GOOGLE CLOUD PROJECT	5
1.3. CREATING A GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT ROLE	5
1.4. ADDING A BILLING SERVICE ACCOUNT MEMBER TO YOUR GOOGLE CLOUD PROJECT	6
1.5. CREATING A GOOGLE CLOUD BIGQUERY DATASET	7
1.6. EXPORTING GOOGLE CLOUD BILLING DATA TO BIGQUERY	8
CHAPTER 2. INTEGRATING FILTERED GOOGLE CLOUD DATA INTO HYBRID COMMITTED SPEND	9
2.1. ADDING YOUR GOOGLE CLOUD ACCOUNT AS AN INTEGRATION	9
2.2. CREATING A GOOGLE CLOUD PROJECT	10
2.3. CREATING A GOOGLE CLOUD BUCKET	10
2.4. CREATING A GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT ROLE	11
2.5. ADDING A BILLING SERVICE ACCOUNT MEMBER TO YOUR GOOGLE CLOUD PROJECT	12
2.6. CREATING A GOOGLE CLOUD BIGQUERY DATASET	13
2.7. EXPORTING GOOGLE CLOUD BILLING DATA TO BIGQUERY	13
2.8. CREATING A FUNCTION TO POST FILTERED DATA TO YOUR STORAGE BUCKET	14
2.9. TRIGGER YOUR FUNCTION TO POST FILTERED DATA TO YOUR STORAGE BUCKET	17
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	19

PREFACE

To add a Google Cloud account to hybrid committed spend, you must add it as a integration from the [Red Hat Hybrid Cloud Console](#) user interface and configure Google Cloud to provide metrics. You can send your data automatically, or configure a function script to copy the cost exports and object storage bucket that hybrid committed spend can access and filter your data to share a subset of your billing data with Red Hat.

CHAPTER 1. CREATING A GOOGLE CLOUD INTEGRATION

To add a Google Cloud account to hybrid committed spend, you must configure your Google Cloud account to provide metrics, then add it as a integration from the [Red Hat Hybrid Cloud Console](#) user interface.



NOTE

You must have a Red Hat account user with Cloud Administrator permissions before you can add integrations to hybrid committed spend.

To configure your Google Cloud account to be a hybrid committed spend integration, you must complete the following tasks:

- Create a Google Cloud project for your hybrid committed spend data.
- Create a bucket for filtered reports.
- Billing service account member with the correct role to export your data to hybrid committed spend.
- Create a BigQuery dataset to contain the cost data.
- Create a billing export that sends the hybrid committed spend data to your BigQuery dataset.

As you will complete some of the following steps in the Google Cloud Console, and some steps in the hybrid committed spend user interface, keep both applications open in a web browser.



NOTE

Because third-party products and documentation can change, instructions for configuring the third-party integrations provided are general and correct at the time of publishing. For the most up-to-date information, see the [Google Cloud Platform documentation](#).

Add your Google Cloud integrations to hybrid committed spend from [the Integrations page](#).


1.1. ADDING YOUR GOOGLE CLOUD ACCOUNT AS AN INTEGRATION

You can add your Google Cloud account as an integration. After adding a Google Cloud integration, the hybrid committed spend application processes the cost and usage data from your Google Cloud account and makes it viewable.

Prerequisites

- To add data integrations to cost management, you must have a Red Hat account with Cloud Administrator permissions.

Procedure

1. From [Red Hat Hybrid Cloud Console](#), click **Settings Menu**  > **Integrations**.
2. On the **Settings** page, in the **Cloud** tab, click **Add integration**.

3. In the **Add a cloud integration** wizard, select **Google Cloud** as the cloud provider type and click **Next**.
4. Enter a name for your integration. Click **Next**.
5. In the **Select application** step, select **Hybrid committed spend** and click **Next**.

1.2. CREATING A GOOGLE CLOUD PROJECT

Create a Google Cloud project to gather and send your cost reports to hybrid committed spend.

Prerequisites

- Access to Google Cloud Console with **resourcemanager.projects.create** permission

Procedure

1. In the [Google Cloud Console](#) click **IAM & Admin** → **Create a Project**
2. Enter a **Project name** in the new page that appears and select your billing account.
3. Select the **Organization**.
4. Enter the parent organization in the **Location** box.
5. Click **Create**.
6. In the hybrid committed spend **Add a cloud integration** wizard, on the **Project** page, enter your **Project ID**.
7. To send the default data to Red Hat automatically, select **I am OK with sending the default data set to hybrid committed spend** and click **Next**.

Verification steps

1. Navigate to the Google Cloud Console Dashboard
2. Verify the project is in the menu bar.

Additional resources

- For additional information about creating projects, see the Google Cloud documentation [Creating and managing projects](#).

1.3. CREATING A GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT ROLE

A custom Identity and Access Management (IAM) role for hybrid committed spend gives access to specific cost related resources required to enable a Google Cloud Platform integration and prohibits access to other resources.

Prerequisites

- Access to Google Cloud Console with these permissions:

- `resourcemanager.projects.get`
- `resourcemanager.projects.getIamPolicy`
- `resourcemanager.projects.setIamPolicy`
- Google Cloud [project](#)

Procedure

1. In the [Google Cloud Console](#), click **IAM & Admin** → **Roles**.
2. Select the hybrid committed spend project from the dropdown in the menu bar.
3. Click **+ Create role**.
4. Enter a **Title**, **Description** and **ID** for the role. In this example, use **customer-data-role**.
5. Click **+ ADD PERMISSIONS**.
6. Use the **Enter property name or value** field to search and select these four permissions for your custom role:
 - `bigquery.jobs.create`
 - `bigquery.tables.getData`
 - `bigquery.tables.get`
 - `bigquery.tables.list`
7. Click **ADD**.
8. Click **CREATE**.
9. In the hybrid committed spend **Add a cloud integration wizard**, on the **Create IAM role** page, click **Next**.

Additional resources

- For additional information about roles and their usage, see the Google Cloud documentation [Understanding roles](#) and [Creating and managing custom roles](#).

1.4. ADDING A BILLING SERVICE ACCOUNT MEMBER TO YOUR GOOGLE CLOUD PROJECT

You must create a billing service account member that can export cost reports to [Red Hat Hybrid Cloud Console](#) in your project.

Prerequisites

- Access to Google Cloud Console with these permissions:
 - `resourcemanager.projects.get`
 - `resourcemanager.projects.getIamPolicy`

- `resourcemanager.projects.setIamPolicy`
- Google Cloud [project](#)
- A hybrid committed spend Identity and Access Management (IAM) [role](#)

Procedure

1. In the [Google Cloud Console](#), click **IAM & Admin** → **IAM**.
2. Select the hybrid committed spend project from the dropdown in the menu bar.
3. Click **ADD**.
4. Paste the IAM role you created into the **New principals** field:

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

5. In the **Assign roles** section, assign the IAM role you created. In this example, use **customer-data-role**.
6. Click **SAVE**.
7. In the hybrid committed spend **Add a cloud integration** wizard, on the **Assign access** page, click **Next**.

Verification steps

1. Navigate to **IAM & Admin** → **IAM**.
2. Verify the new member is present with the correct role.

Additional resources

- For additional information about roles and their usage, see the Google Cloud documentation [Understanding roles](#) and [Creating and managing custom roles](#).

1.5. CREATING A GOOGLE CLOUD BIGQUERY DATASET

Create a BigQuery dataset to collect and store the billing data for hybrid committed spend.

Prerequisites

- Access to Google Cloud Console with **bigquery.datasets.create** permission
- Google Cloud [project](#)

Procedure

1. In [Google Cloud Console](#), click **Big Data** → **BigQuery**.
2. Select the hybrid committed spend project in the **Explorer** panel.
3. Click **CREATE DATASET**.

4. Enter a name for your dataset in the **Dataset ID** field. In this example, use **CustomerData**.
5. Click **CREATE DATASET**.

1.6. EXPORTING GOOGLE CLOUD BILLING DATA TO BIGQUERY

Enabling a billing export to BigQuery sends your Google Cloud billing data (such as usage, cost estimates, and pricing data) automatically to the hybrid committed spend BigQuery dataset.

Prerequisites

- Access to Google Cloud Console with the **Billing Account Administrator** role
- Google Cloud [project](#)
- [Billing service member](#) with the cost management Identity and Access Management (IAM) [role](#)
- [BigQuery dataset](#)

Procedure

1. In the [Google Cloud Console](#), click **Billing** → **Billing export**.
2. Click the **Billing export** tab.
3. Click **EDIT SETTINGS** in the **Detailed usage cost** section.
4. Select the hybrid committed spend **Project** and **Billing export dataset** you created in the dropdown menus.
5. Click **SAVE**.
6. In the hybrid committed spend **Add a cloud integration** wizard, on the **Billing export** page, click **Next**.
7. In the hybrid committed spend **Add a cloud integration** wizard, on the **Review details** page, click **Add**.

Verification steps

1. Verify a checkmark with **Enabled** in the **Detailed usage cost** section, with correct **Project name** and **Dataset name**.

CHAPTER 2. INTEGRATING FILTERED GOOGLE CLOUD DATA INTO HYBRID COMMITTED SPEND

You can configure a function script in Google Cloud to copy the cost exports and object storage bucket that hybrid committed spend can access and filter your data to share a subset of your billing data with Red Hat.



NOTE

You must have a Red Hat account user with Cloud Administrator permissions before you can add integrations to hybrid committed spend.

To configure your Google Cloud account to be a hybrid committed spend integration, you must complete the following tasks:

- Create a Google Cloud project for your hybrid committed spend data.
- Create a bucket for filtered reports.
- Have a billing service account member with the correct role to export your data to hybrid committed spend.
- Create a BigQuery dataset to contain the cost data.
- Create a billing export that sends the hybrid committed spend data to your BigQuery dataset.

Because you will complete some of the following steps in the Google Cloud Console, and some steps in the hybrid committed spend user interface, keep both applications open in a web browser.



NOTE

Because third-party products and documentation can change, instructions for configuring the third-party integrations provided are general and correct at the time of publishing. For the most up-to-date information, see the [Google Cloud Platform documentation](#).

Add your Google Cloud integration to hybrid committed spend from [the Integrations page](#).

2.1. ADDING YOUR GOOGLE CLOUD ACCOUNT AS AN INTEGRATION

You can add your Google Cloud account as an integration. After adding a Google Cloud integration, the hybrid committed spend application processes the cost and usage data from your Google Cloud account and makes it viewable.

Prerequisites

- To add data integrations to cost management, you must have a Red Hat account with Cloud Administrator permissions.

Procedure

1. From [Red Hat Hybrid Cloud Console](#), click **Settings Menu**  > **Integrations**.

2. On the **Settings** page, in the **Cloud** tab, click **Add integration**.
3. In the **Add a cloud integration** wizard, select **Google Cloud** as the cloud provider type and click **Next**.
4. Enter a name for your integration. Click **Next**.
5. In the **Select application** step, select **Hybrid committed spend** and click **Next**.

2.2. CREATING A GOOGLE CLOUD PROJECT

Create a Google Cloud project to gather and send your cost reports to hybrid committed spend.

Prerequisites

- Access to Google Cloud Console with **resourcemanager.projects.create** permission

Procedure

1. In the [Google Cloud Console](#) click **IAM & Admin** → **Create a Project**
2. Enter a **Project name** in the new page that appears and select your billing account.
3. Select the **Organization**.
4. Enter the parent organization in the **Location** box.
5. Click **Create**.
6. In the hybrid committed spend **Add a cloud integration** wizard, on the **Project** page, enter your **Project ID**.
7. To configure Google Cloud to filter your data before it sends the data to Red Hat, select **I wish to manually customize the data set sent to hybrid committed spend**, click **Next**.

Verification steps

1. Navigate to the Google Cloud Console Dashboard
2. Verify the project is in the menu bar.

Additional resources

- For additional information about creating projects, see the Google Cloud documentation [Creating and managing projects](#).

2.3. CREATING A GOOGLE CLOUD BUCKET

Create a bucket for filtered reports that you will create later. Buckets are containers that store data.

Procedure

1. In the [Google Cloud Console](#), click **Buckets**.
2. Click **Create bucket**.

3. Enter your bucket information. Name your bucket. In this example, use **customer-data**.
4. Click **Create**, then click **Confirm** in the confirmation dialog.
5. In the hybrid committed spend **Add a cloud integration** wizard, on the **Create cloud storage bucket** page, enter your **Cloud storage bucket name**.

Additional resources

- For additional information about creating buckets, see the Google Cloud documentation on [Creating buckets](#).

2.4. CREATING A GOOGLE CLOUD IDENTITY AND ACCESS MANAGEMENT ROLE

A custom Identity and Access Management (IAM) role for hybrid committed spend gives access to specific cost related resources required to enable a Google Cloud Platform integration and prohibits access to other resources.

Prerequisites

- Access to Google Cloud Console with these permissions:
 - **resourcemanager.projects.get**
 - **resourcemanager.projects.getIamPolicy**
 - **resourcemanager.projects.setIamPolicy**
- Google Cloud [project](#)

Procedure

1. In the [Google Cloud Console](#), click **IAM & Admin** → **Roles**.
2. Select the hybrid committed spend project from the dropdown in the menu bar.
3. Click **+ Create role**.
4. Enter a **Title**, **Description** and **ID** for the role. In this example, use **customer-data-role**.
5. Click **+ ADD PERMISSIONS**.
6. Use the **Enter property name or value** field to search and select these four permissions for your custom role:
 - **storage.objects.get**
 - **storage.objects.list**
 - **storage.buckets.get**
7. Click **ADD**.
8. Click **CREATE**.

9. In the hybrid committed spend **Add a cloud integration** wizard, on the **Create IAM role** page, click **Next**.

Additional resources

- For additional information about roles and their usage, see the Google Cloud documentation [Understanding roles](#) and [Creating and managing custom roles](#).

2.5. ADDING A BILLING SERVICE ACCOUNT MEMBER TO YOUR GOOGLE CLOUD PROJECT

You must create a billing service account member that can export cost reports to [Red Hat Hybrid Cloud Console](#) in your project.

Prerequisites

- Access to Google Cloud Console with these permissions:
 - **resourcemanager.projects.get**
 - **resourcemanager.projects.getIamPolicy**
 - **resourcemanager.projects.setIamPolicy**
- Google Cloud [project](#)
- A hybrid committed spend Identity and Access Management (IAM) [role](#)

Procedure

1. In the [Google Cloud Console](#), click **IAM & Admin** → **IAM**.
2. Select the hybrid committed spend project from the dropdown in the menu bar.
3. Click **ADD**.
4. Paste the IAM role you created into the **New principals** field:

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

5. In the **Assign roles** section, assign the IAM role you created. In this example, use **customer-data-role**.
6. Click **SAVE**.
7. In the hybrid committed spend **Add a cloud integration** wizard, on the **Assign access** page, click **Next**.

Verification steps

1. Navigate to **IAM & Admin** → **IAM**.
2. Verify the new member is present with the correct role.

Additional resources

- For additional information about roles and their usage, see the Google Cloud documentation [Understanding roles](#) and [Creating and managing custom roles](#).

2.6. CREATING A GOOGLE CLOUD BIGQUERY DATASET

Create a BigQuery dataset to collect and store the billing data for hybrid committed spend.

Prerequisites

- Access to Google Cloud Console with **bigquery.datasets.create** permission
- Google Cloud [project](#)

Procedure

1. In [Google Cloud Console](#), click **Big Data** → **BigQuery**.
2. Select the hybrid committed spend project in the **Explorer** panel.
3. Click **CREATE DATASET**.
4. Enter a name for your dataset in the **Dataset ID** field. In this example, use **CustomerFilteredData**.
5. Click **CREATE DATASET**.

2.7. EXPORTING GOOGLE CLOUD BILLING DATA TO BIGQUERY

Enabling a billing export to BigQuery sends your Google Cloud billing data (such as usage, cost estimates, and pricing data) automatically to the hybrid committed spend BigQuery dataset.

Prerequisites

- Access to Google Cloud Console with the **Billing Account Administrator** role
- Google Cloud [project](#)
- [Billing service member](#) with the cost management Identity and Access Management (IAM) [role](#)
- [BigQuery dataset](#)

Procedure

1. In the [Google Cloud Console](#), click **Billing** → **Billing export**.
2. Click the **Billing export** tab.
3. Click **EDIT SETTINGS** in the **Detailed usage cost** section.
4. Select the hybrid committed spend **Project** and **Billing export dataset** you created in the dropdown menus.
5. Click **SAVE**.

6. In the hybrid committed spend **Add a cloud integration** wizard, on the **Billing export** page, click **Next**.
7. In the hybrid committed spend **Add a cloud integration** wizard, on the **Review details** page, click **Add**.

Verification steps

1. Verify a checkmark with **Enabled** in the **Detailed usage cost** section, with correct **Project name** and **Dataset name**.

2.8. CREATING A FUNCTION TO POST FILTERED DATA TO YOUR STORAGE BUCKET

Create a function that filters your data and adds it to the storage account that you created to share with Red Hat. You can use the example Python script to gather the cost data from your cost exports related to your Red Hat expenses and add it to the storage account. This script filters the cost data you created with BigQuery, removes non-Red Hat information, then creates **.csv** files, stores them in the bucket you created, and sends the data to Red Hat.

Procedure

1. In the [Google Cloud Console](#), search for **secret** and select the **Secret manager** result to set up a secret to authenticate your function with Red Hat without storing your credentials in your function.
 - a. On the Secret Manager page, click **Create Secret**
 - b. Name your secret, add your Red Hat username, and click **Create Secret**
 - c. Repeat this process to save a secret for your Red Hat password.
2. In the Google Cloud Console search bar, search for **functions** and select the **Cloud Functions** result.
3. On the **Cloud Functions** page, click **Create function**.
4. Name the function. In this example, use **customer-data-function**.
5. In the **Trigger** section, click **Save** to accept the HTTP Trigger type.
6. In the **Runtime, build, connections and security settings**, click the Security and image repository, reference the secrets you created, click **Done**, and click **Next**.
7. On the **Cloud Functions** Code page, set the runtime to **Python 3.9**.
8. Open the **requirements.txt** file. Paste the following lines to the end of the file.

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

9. Open the **main.py** file.
 - a. Set the **Entry Point** to **get_filtered_data**.

- b. Paste the following python script. Change the values in the section marked **# Required vars to update** to the values for your environment.

```
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. Click **Deploy**.

2.9. TRIGGER YOUR FUNCTION TO POST FILTERED DATA TO YOUR STORAGE BUCKET

Create a scheduler job to run the function you created to send filtered data to Red Hat on a schedule.

Procedure

1. Copy the **Trigger URL** for the function you created to post the cost reports. You will need to add it to the Google Cloud Scheduler.
 - a. In the [Google Cloud Console](#), search for **functions** and select the **Cloud Functions** result.
 - b. On the **Cloud Functions** page, select your function, and click the Trigger tab.
 - c. In the HTTP section, click **Copy to clipboard**
2. Create the scheduler job. In the [Google Cloud Console](#), search for **cloud scheduler** and select the **Cloud Scheduler** result.
3. Click **Create job**.
 - a. Name your scheduler job. In this example, use **CustomerFilteredDataSchedule**.
 - b. In the **Frequency** field, set the cron expression for when you want the function to run. In this example, use **09***** to run the function daily at 9 AM.
 - c. Set the timezone and click **Continue**.
4. Configure the execution on the next page.
 - a. In the **Target type** field, select **HTTP**.
 - b. In the URL field, paste the Trigger URL you copied.
 - c. In the body field, paste the following code that passes into the function to trigger it.

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

- d. In the Auth header field, select **Add OIDC token**.
 - e. Click the **Service account** field and click **Create** to create a service account and role for the scheduler job.
5. In the **Service account details** step, name your service account. In this example, use **scheduler-service-account**. Accept the default **Service account ID** and click **Create and Continue**

- a. In the Grand this service account access to project, select two roles for your account.
 - b. Click **ADD ANOTHER ROLE** then search for and select **Cloud Scheduler Job Runner** and Cloud Functions Invoker.
 - c. Click **Continue**.
 - d. Click **Done** to finish creating the service account.
6. On the Service accounts for your project page, select the scheduler job that you were working on. In this example, the name is **scheduler-service-account**.
 7. In the **Configure the execution** page, select the **Service account** field and select the **scheduler-service-account** you just created.
 8. Click **Continue** and then click **Create**.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

If you found an error or have a suggestion on how to improve these guidelines, open an issue in the [cost management Jira board](#) and add the **Documentation** label.

We appreciate your feedback!