



Cost Management Service 1-latest

Integrating Amazon Web Services (AWS) data into cost management

Learn how to add your AWS integrations and RHEL metering

Cost Management Service 1-latest Integrating Amazon Web Services (AWS) data into cost management

Learn how to add your AWS integrations and RHEL metering

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

Learn how to add an Amazon Web Services (AWS) integration to cost management. Cost management is part of the Red Hat Insights portfolio of services. The Red Hat Insights suite of advanced analytical tools helps you to identify and prioritize impacts on your operations, security, and business.

Table of Contents

PART I. CHOOSING A BASIC OR ADVANCED AWS INTEGRATION	3
CHAPTER 1. CREATING YOUR AMAZON WEB SERVICES INTEGRATION: BASIC	4
1.1. ADDING AN AWS ACCOUNT AS AN INTEGRATION	4
1.2. CREATING AN S3 BUCKET AND A DATA EXPORT	5
1.3. ACTIVATING AWS TAGS	5
1.4. CONFIGURING AN IAM POLICY TO ENABLE ACCOUNT ACCESS FOR COST AND USAGE REPORTS	6
CHAPTER 2. CREATING YOUR AMAZON WEB SERVICES INTEGRATION: ADVANCED	9
2.1. ADDING AN AWS ACCOUNT AS AN INTEGRATION	9
2.2. CREATING AN AWS S3 BUCKET TO STORE YOUR ATHENA BILLING DATA	10
2.3. CREATING A BUCKET TO STORE FILTERED DATA REPORTING	10
2.4. ACTIVATING AWS TAGS	11
2.5. CONFIGURING AN IAM POLICY TO ENABLE ACCOUNT ACCESS FOR COST AND USAGE REPORTS	11
2.6. ENABLING ACCOUNT ACCESS FOR ATHENA	13
2.6.1. Configuring Athena for report generation	16
2.6.2. Building an Athena query	17
2.6.3. Creating a Lambda function for Athena	18
2.6.4. Creating a Lambda function to post the report files	19
2.7. CREATING EVENT BRIDGE SCHEDULES	20
2.8. CREATING ADDITIONAL CLOUD FUNCTIONS TO COLLECT FINALIZED DATA	20
CHAPTER 3. NEXT STEPS FOR MANAGING YOUR COSTS	23
3.1. LIMITING ACCESS TO COST MANAGEMENT RESOURCES	23
3.2. CONFIGURING TAGGING FOR YOUR INTEGRATIONS	23
3.3. CONFIGURING AWS BILLING PLANS	24
3.4. CONFIGURING COST MODELS TO ACCURATELY REPORT COSTS	24
3.5. VISUALIZING YOUR COSTS WITH COST EXPLORER	24
CHAPTER 4. UPDATING AN INTEGRATION	26
4.1. ADDING RHEL METERING TO AN AWS INTEGRATION	26
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	27

PART I. CHOOSING A BASIC OR ADVANCED AWS INTEGRATION

To create an AWS integration, first decide if you want to take a basic or advanced integration path.

Basic

For the basic option, go to [Creating an Amazon Web Services integration: Basic](#).

The basic path enables cost management to directly read your billing reports from AWS at a scope that you indicate.

Advanced

For the advanced option, go to [Creating an Amazon Web Services integration: Advanced](#).

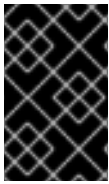
The advanced path enables you to customize or filter your data *before* cost management reads it. You might also use the advanced path if you want to share billing data only to certain Red Hat products. The advanced path has more complex set-up and configuration.



NOTE

You must select either basic or advanced, you cannot choose both.

CHAPTER 1. CREATING YOUR AMAZON WEB SERVICES INTEGRATION: BASIC



IMPORTANT

If you want to create an AWS integration by using the advanced path, do not complete the following steps. Instead, go to [Creating your Amazon Web Services integration: Advanced](#).

If you are using RHEL metering, after you integrate your data with cost management, go to [Adding RHEL metering to an AWS integration](#) to finish configuring your integration for RHEL metering.

You must create an AWS integration for cost management from [the Integrations page](#) and configure your AWS account to allow cost management access.

AWS is a third-party product and its UI and documentation can change. The instructions for configuring third-party integrations are correct at the time of publishing. For the most up-to-date information, see the [AWS documentation](#).

Prerequisites

- You must have a [Red Hat Hybrid Cloud Console service account](#).
- To add data integrations to cost management, you must have a Red Hat account with Cloud Administrator permissions.


1.1. ADDING AN AWS ACCOUNT AS AN INTEGRATION

Add an AWS integration so cost management can process the Cost and Usage Reports from your AWS account. You can add an AWS integration automatically by providing your AWS account credentials.

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. On the **Select integration type** step, in the **Add a cloud integration** wizard, select **Amazon Web Services**. Click **Next**.
4. Enter a name for the integration and click **Next**.
5. On the **Select configuration** step, select how you want to connect to your AWS integration.
 - Select **Account authorization** to provide your AWS account credentials and let Red Hat configure and manage your integration for you. Click **Next**.

- Select **Manual configuration** to customize your integration. If you are using cost management to meter your RHEL subscription, you *must* select **Manual Configuration**.
6. In the **Select application** step, select **Cost management**. Click **Next**.
 7. If you selected the account authorization method, on the **Review details** step, review the details and click **Add**. If you selected the manual configuration method, continue to the next step in the wizard and configure your S3 bucket.

1.2. CREATING AN S3 BUCKET AND A DATA EXPORT

Create an Amazon S3 bucket with permissions configured to store your data exports.

Procedure

To create a data export, log in to your AWS account and complete the following steps:

1. In the AWS S3 console, create a new S3 bucket or use an existing bucket. If you are configuring a new S3 bucket, accept the default settings.
2. On the **Create storage** step, in the **Add a cloud source** wizard, paste the name of your S3 bucket and select the region that it was created in. Click **Next**.
3. In the AWS Billing console, create a data export that will be delivered to your S3 bucket. Enter the following values and accept the defaults for any other values:
 - **Export type:** Legacy CUR export
 - **Report name:** koku
 - **Include:** resource IDs
 - **Time unit:** Hourly
 - **Enable report data integration for** Amazon Redshift, Amazon QuickSight. Disable report data integration for Amazon Athena
 - **Compression type:** GZIP
 - **S3 bucket:** <the S3 bucket that you configured before>
 - **Report path prefix:** cost



NOTE

For more details on configuration, see the *AWS Billing and Cost Management* documentation.

4. In the **Add a cloud integration** wizard, on the **Create cost and usage report** step, click **Next**.

1.3. ACTIVATING AWS TAGS

To use tags to organize your AWS resources in the cost management application, activate your tags in AWS to allow them to be imported automatically.

Procedure

1. In the AWS Billing console:
 - a. Open the *Cost Allocation Tags* section.
 - b. Select the tags you want to use in the cost management application, and click **Activate**.
2. If your organization is converting systems from CentOS 7 to RHEL and using hourly billing, activate the **com_redhat_rhel** tag for your systems in the **Cost Allocation Tags** section of the AWS console.
 - a. After tagging the instances of RHEL you want to meter in AWS, select **Include RHEL usage**.
3. In the [Red Hat Hybrid Cloud Console Integrations](#) wizard, select **Include RHEL usage**.

Additional resources

For more information about tagging, see [Adding tags to an AWS resource](#) .

1.4. CONFIGURING AN IAM POLICY TO ENABLE ACCOUNT ACCESS FOR COST AND USAGE REPORTS

Cost management needs Cost and Usage Reports produced by AWS to display data. To provide the correct access, create an IAM policy and role in AWS, which provides access only to the stored information.

Cost management can also display additional data. For example:

- Include the Action **iam:ListAccountAliases** to display an AWS account alias rather than an account number.
- If you are using consolidated billing rather than the account ID, include the Actions *organization:List** and *organizations:Describe** to find the display names of AWS member accounts.

In cost management:

1. In the **Add a cloud integration** wizard, select the additional data points you want to be included.
2. Click **Next**.
3. Copy the JSON output that is generated based on your selections.

In the [AWS Identity and Access Management console](#)

3. From the AWS Identity and Access Management (IAM) console, create a new IAM policy for the S3 bucket that you configured before.
 - a. Select the JSON tab and paste the JSON policy which you copied from the [Red Hat Hybrid Cloud Console Add a cloud integration](#) wizard:

```
{
  "Version": "2012-10-17",
  "Statement": [
```

```

{
  "Sid": "VisualEditor0",
  "Effect": "Allow",
  "Action": [
    "s3:Get*",
    "s3:List*"
  ],
  "Resource": [
    "arn:aws:s3:::<your_bucket_name>", 1
    "arn:aws:s3:::<your_bucket_name>/*"
  ]
},

{
  "Sid": "VisualEditor1",
  "Effect": "Allow",
  "Action": [
    "s3:ListBucket",
    "cur:DescribeReportDefinitions"
  ],
  "Resource": "*"
}
]
}

```

- b. Enter a name for the policy and create the policy. Do not close the AWS IAM console. You will use it in the following steps.

In cost management:

4. In the [Red Hat Hybrid Cloud Console](#) **Add a cloud integration** wizard, click **Next**.

In the [AWS Identity and Access Management console](#)

5. In the AWS IAM console, create a new IAM role:
 - a. Select **Another AWS account** as the type of trusted entity.
 - b. Enter `589173575009` as the Account ID to give [Red Hat Hybrid Cloud Console](#) read access to the AWS account cost data.

In cost management:

- a. Copy your external ID from the Create IAM role step in the wizard.

In the [AWS Identity and Access Management console](#)

- a. Enter your external ID in the **External ID** field.
- b. Attach the IAM policy you just configured.
- c. Enter a role name and description.

In cost management:

6. In the [Red Hat Hybrid Cloud Console](#) **Add a cloud integration** wizard, click **Next**.

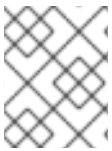
In the [AWS Identity and Access Management console](#)

7. In the AWS IAM console, in the **Roles** section, open the summary screen for the role you just created.
 - a. Copy the Role ARN, which is a string beginning with **arn:aws:**.

In cost management:

8. In the [Red Hat Hybrid Cloud Console](#) **Add a cloud integration** wizard, paste your Role ARN and click **Next**.
9. Review the details of your cloud integration and click **Add**.

Cost management will begin collecting Cost and Usage data from your AWS account and any linked AWS accounts.



NOTE

The data can take a few days to populate before it shows on the [cost management](#) dashboard.

CHAPTER 2. CREATING YOUR AMAZON WEB SERVICES INTEGRATION: ADVANCED



IMPORTANT

If you created an AWS integration by using the basic path, do not complete the following steps. Your AWS integration is already complete.

If you are using RHEL metering, after you integrate your data with cost management, go to [Adding RHEL metering to an AWS integration](#) to finish configuring your integration for RHEL metering.

To share a subset of your billing data with Red Hat, you can configure a function script in AWS. This script will filter your billing data and export it to object storage so that cost management can then access and read the filtered data. Add your AWS integration to cost management from [the Integrations page](#).

AWS is a third-party product and its UI and documentation can change. The instructions for configuring third-party integrations are correct at the time of publishing. For the most up-to-date information, see the [AWS documentation](#).

Prerequisites

- You must have a [Red Hat Hybrid Cloud Console service account](#).
- To add data integrations to cost management, you must have a Red Hat account with Cloud Administrator permissions.


2.1. ADDING AN AWS ACCOUNT AS AN INTEGRATION

Add an AWS integration so cost management can process the Cost and Usage Reports from your AWS account. You can add an AWS integration automatically by providing your AWS account credentials, or you can configure cost management to filter the data that you send to Red Hat.

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. On the **Select integration type** step, in the **Add a cloud integration** wizard, select **Amazon Web Services**. Click **Next**.
4. Enter a name for the integration and click **Next**.
5. On the **Select configuration** step, select how you want to connect to your AWS integration.

- Select **Manual configuration** to customize your integration. If you are using cost management to meter your RHEL subscription, you *must* select **Manual Configuration**. Click **Next**.

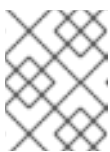
6. In the **Select application** step, select **Cost management**. Click **Next**.

2.2. CREATING AN AWS S3 BUCKET TO STORE YOUR ATHENA BILLING DATA

Create an Amazon S3 bucket with permissions configured to store Athena billing reports.

Procedure

1. Log in to your AWS account.
2. In the AWS Billing console, create a data export that will be delivered to your S3 bucket. Specify the following values and accept the defaults for any other values:
 - **Export type:** Legacy CUR export
 - **Report name:** `<rh_cost_report>` (note this name as you will use it later)
 - **Additional report details:** Include resource IDs
 - **S3 bucket:** Select an S3 bucket you configured previously or create a bucket and accept the default settings.
 - **Time granularity:** Hourly
 - **Enable report data integration for Amazon Athena,** which is required for lambda queries
 - **Compression type:** Parquet
 - **Report path prefix:** cost



NOTE

For more details on configuration, see the *AWS Billing and Cost Management* documentation.

2.3. CREATING A BUCKET TO STORE FILTERED DATA REPORTING

To share your filtered data with Red Hat, you must create a second bucket to store the data.

In your [AWS account](#):

1. Log in to your AWS account.
2. From **Configure S3 Bucket**, click **Configure**. Create a bucket and apply the default policy.
3. Click **Save**.

In cost management:

4. On the **Create storage** step, paste the name of your S3 bucket and select the region that it was created in and click **Next**.
5. On the **Create cost and usage report** step in the **Add a cloud integration** wizard, select **I wish to manually customize the CUR sent to Cost Management**.
6. Click **Next**.

2.4. ACTIVATING AWS TAGS

To use tags to organize your AWS resources in the cost management application, activate your tags in AWS to allow them to be imported automatically.

Procedure

1. In the AWS Billing console:
 - a. Open the *Cost Allocation Tags* section.
 - b. Select the tags you want to use in the cost management application, and click **Activate**.
2. If your organization is converting systems from CentOS 7 to RHEL and using hourly billing, activate the **com_redhat_rhel** tag for your systems in the **Cost Allocation Tags** section of the AWS console.
 - a. After tagging the instances of RHEL you want to meter in AWS, select **Include RHEL usage**.
3. In the [Red Hat Hybrid Cloud Console Integrations](#) wizard, select **Include RHEL usage**.

Additional resources

For more information about tagging, see [Adding tags to an AWS resource](#) .

2.5. CONFIGURING AN IAM POLICY TO ENABLE ACCOUNT ACCESS FOR COST AND USAGE REPORTS

Cost management needs Cost and Usage Reports produced by AWS to display data. To provide the correct access, create an IAM policy and role in AWS, which provides access only to the stored information.

Cost management can also display additional data. For example:

- Include the Action **iam:ListAccountAliases** to display an AWS account alias rather than an account number.
- If you are using consolidated billing rather than the account ID, include the Actions *organization:List** and *organizations:Describe** to find the display names of AWS member accounts.

In cost management:

1. In the **Add a cloud integration** wizard, select the additional data points you want to be included.
2. Click **Next**.

- Copy the JSON output that is generated based on your selections.

In the [AWS Identity and Access Management console](#)

- From the AWS Identity and Access Management (IAM) console, create a new IAM policy for the S3 bucket that you configured before.
 - Select the JSON tab and paste the JSON policy which you copied from the [Red Hat Hybrid Cloud Console Add a cloud integration wizard](#):

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "VisualEditor0",
      "Effect": "Allow",
      "Action": [
        "s3:Get*",
        "s3:List*"
      ],
      "Resource": [
        "arn:aws:s3:::<your_bucket_name>", 1
        "arn:aws:s3:::<your_bucket_name>/*"
      ]
    },
    {
      "Sid": "VisualEditor1",
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "cur:DescribeReportDefinitions"
      ],
      "Resource": "*"
    }
  ]
}
```

- Enter a name for the policy and create the policy. Do not close the AWS IAM console. You will use it in the following steps.

In cost management:

- In the [Red Hat Hybrid Cloud Console Add a cloud integration wizard](#), click **Next**.

In the [AWS Identity and Access Management console](#)

- In the AWS IAM console, create a new IAM role:
 - Select **Another AWS account** as the type of trusted entity.
 - Enter `589173575009` as the Account ID to give [Red Hat Hybrid Cloud Console](#) read access to the AWS account cost data.

In cost management:

- a. Copy your external ID from the Create IAM role step in the wizard.

In the [AWS Identity and Access Management console](#)

- a. Enter your external ID in the **External ID** field.
- b. Attach the IAM policy you just configured.
- c. Enter a role name and description.

In cost management:

6. In the [Red Hat Hybrid Cloud Console Add a cloud integration](#) wizard, click **Next**.

In the [AWS Identity and Access Management console](#)

7. In the AWS IAM console, in the **Roles** section, open the summary screen for the role you just created.
 - a. Copy the Role ARN, which is a string beginning with **arn:aws:**.

In cost management:

8. In the [Red Hat Hybrid Cloud Console Add a cloud integration](#) wizard, paste your Role ARN and click **Next**.
9. Review the details of your cloud integration and click **Add**.

Next steps

Return to AWS to customize your AWS data export by configuring Athena and Lambda to filter your reports.

2.6. ENABLING ACCOUNT ACCESS FOR ATHENA

Create an IAM policy and role for hybrid committed spend to use. This configuration provides access to the stored information and nothing else.

Procedure

1. From the AWS Identity and Access Management (IAM) console, create an IAM policy for the Athena Lambda functions you will configure.
 - a. Select the JSON tab and paste the following content in the JSON policy text box:

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "athena:*"
      ],
      "Resource": [
        "*"
      ]
    }
  ]
}
```

```

    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "glue:CreateDatabase",
      "glue>DeleteDatabase",
      "glue:GetDatabase",
      "glue:GetDatabases",
      "glue:UpdateDatabase",
      "glue:CreateTable",
      "glue>DeleteTable",
      "glue:BatchDeleteTable",
      "glue:UpdateTable",
      "glue:GetTable",
      "glue:GetTables",
      "glue:BatchCreatePartition",
      "glue:CreatePartition",
      "glue>DeletePartition",
      "glue:BatchDeletePartition",
      "glue:UpdatePartition",
      "glue:GetPartition",
      "glue:GetPartitions",
      "glue:BatchGetPartition"
    ],
    "Resource": [
      "*"
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetBucketLocation",
      "s3:GetObject",
      "s3:ListBucket",
      "s3:ListBucketMultipartUploads",
      "s3:ListMultipartUploadParts",
      "s3:AbortMultipartUpload",
      "s3:CreateBucket",
      "s3:PutObject",
      "s3:PutBucketPublicAccessBlock"
    ],
    "Resource": [
      "arn:aws:s3:::CHANGE-ME*" 1
    ]
  },
  {
    "Effect": "Allow",
    "Action": [
      "s3:GetObject",
      "s3:ListBucket"
    ],
    "Resource": [
      "arn:aws:s3:::CHANGE-ME*" 2
    ]
  },
}

```

```

    {
      "Effect": "Allow",
      "Action": [
        "s3:ListBucket",
        "s3:GetBucketLocation",
        "s3:ListAllMyBuckets"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "sns:ListTopics",
        "sns:GetTopicAttributes"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "cloudwatch:PutMetricAlarm",
        "cloudwatch:DescribeAlarms",
        "cloudwatch>DeleteAlarms",
        "cloudwatch:GetMetricData"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "lakeformation:GetDataAccess"
      ],
      "Resource": [
        "*"
      ]
    },
    {
      "Effect": "Allow",
      "Action": [
        "logs:*"
      ],
      "Resource": "*"
    }
  ]
}

```

1 1 1 2 Replace **CHANGE-ME*** in both locations with the ARN for the S3 bucket you configured in step 2.2.

- b. Name the policy and complete the creation of the policy. Keep the AWS IAM console open because you will need it for the next step.
2. In the AWS IAM console, create a new IAM role:
 - a. For the type of trusted entity, select **AWS service**.
 - b. Select Lambda.
 - c. Attach the IAM policy you just configured.
 - d. Enter a role name and description and finish creating the role.
3. Store your login information in AWS Secrets Manager and add it to the role you created.
 - a. Select **Secret type: Other type of secret**
 - b. Create a key for your Red Hat Hybrid Cloud Console **client_id**.
 - c. Create a key for your Red Hat Hybrid Cloud Console **client_secret**.
 - d. Add the values for your user name and password to the appropriate key.
 - e. Click **Continue**, then name and store your secret.
 - f. Update the role you created for your Lambda functions. Include the following code to reference the secret stored in AWS Secrets Manager:

```
{
  "Sid": "VisualEditor3",
  "Effect": "Allow",
  "Action": [
    "secretsmanager:GetSecretValue",
    "secretsmanager:DescribeSecret"
  ],
  "Resource": "*"
}
```

2.6.1. Configuring Athena for report generation

Configuring Athena to provide a filtered data export for cost management.

The following configuration only provides access to additional stored information. It does not provide access to anything else:

Procedure

1. In the AWS S3 console, go to the S3 bucket you configured in step 2.2. Then, go to the **crawler-cfn.yml** file, which is in the path created by your data export you configured. For example: **{bucket-name}/{S3_path_prefix}/{export_name}/crawler-cfn.yml**. Copy the **Object URL** for the **crawler-cfn.yml**.
2. From **Cloudformation** in the AWS console, create a stack with new resources:
 - a. Choose an existing template.
 - b. Select **Specify Template**.

- c. Select **Template Source: Amazon S3 URL**
 - d. Paste the object URL you copied before.
3. Enter a name and click **Next**.
 4. Click **I acknowledge that AWS Cloudformation might create IAM resources** and then click **Submit**.

2.6.2. Building an Athena query

Create an Athena query that queries the data export for your Red Hat expenses and creates a report of your filtered expenses.

You might need just the query included with the example script, for example, if you are filtering for Red Hat spending. If you need something more advanced, create a custom query. If you are using RHEL metering, you must adjust the query to return data that is specific to your RHEL subscriptions. The following steps will guide you through creating a RHEL subscription query.

Example Athena query for Red Hat spend

```
SELECT *
FROM <your_export_name>
WHERE (
    bill_billing_entity = 'AWS Marketplace'
    AND line_item_legal_entity like '%Red Hat%'
)
OR (
    line_item_legal_entity like '%Amazon Web Services%'
    AND line_item_line_item_description like '%Red Hat%'
)
OR (
    line_item_legal_entity like '%Amazon Web Services%'
    AND line_item_line_item_description like '%RHEL%'
)
OR (
    line_item_legal_entity like '%AWS%'
    AND line_item_line_item_description like '%Red Hat%'
)
OR (
    line_item_legal_entity like '%AWS%'
    AND line_item_line_item_description like '%RHEL%'
)
OR (
    line_item_legal_entity like '%AWS%'
    AND product_product_name like '%Red Hat%'
)
OR (
    line_item_legal_entity like '%Amazon Web Services%'
    AND product_product_name like '%Red Hat%'
)
AND year = '2024'
AND month = '07'
```

In your [AWS account](#):

1. Go to Amazon Athena from the editor tab.
2. From the **Data source** menu, select **AwsDataCatalog**.
3. From the **Database** menu, select your data export. Your data export name is prepended with **athenacurcfn_** followed by your data export name. For example, **{your_export_name}**.
4. Paste the following example query into the **Query** field. Replace the **your_export_name** value with your data export name.

```
SELECT column_name
FROM information_schema.columns
WHERE table_name = '<your_export_name>'
AND column_name LIKE 'resource_tags_%';
```

5. Click **Run**. The results of this query returns all the tag related columns for your data set.
6. Copy the tag column that matches the column used for your RHEL tags.
7. Paste in the following example query. Replace the **your_export_name**, the tags column copied in the step before, and the **year** and **month** you want to query. The result returns EC2 instances appropriately tagged for RHEL subscriptions. Copy and save this query for use in the future Lambda function.

```
SELECT *
FROM <your_export_name>
WHERE (
    line_item_product_code = 'AmazonEC2'
    AND strpos(lower(<rhel_tag_column_name>), 'com_redhat_rhel') > 0
)
AND year = '<year>'
AND month = '<month>'
```

2.6.3. Creating a Lambda function for Athena

You must create a Lambda function that queries the data export for your Red Hat related expenses and creates a report of your filtered expenses.

Procedure

1. In the AWS console, go to Lambda and click **Create function**.
2. Click **Author from scratch**.
3. Enter a name your function.
4. From the **Runtime** menu, select the latest version of Python available.
5. From the **Architecture** menu, select **x86_64**.
6. Under Permissions select the Athena role you created.
7. To add the query you built as part of the Lambda function, click **Create function** to save your progress.

- From the function **Code** tab, paste this [script](#). Update the following lines:

your_integration_external_id

Enter the integration UUID you copied in the *Enabling account access for cost and usage consumption* step.

bucket

Enter the name of the S3 bucket you created to store filtered reports during the *Creating a bucket for storing filtered data reporting* step.

database

Enter the database name used in the *Building your Athena query* step.

export_name

Enter the name of your data export from when you created an AWS S3 bucket for storing your cost data.

- Update the default query with your custom one by replacing the **where** clause, for example:

```
# Athena query
query = f"SELECT * FROM {database}.{export_name} WHERE (line_item_product_code =
'AmazonEC2' AND strpos(lower(<rhel_tag_column_name>), 'com_redhat_rhel') > 0) AND
year = '{year}' AND month = '{month}'"
```

- Click **Deploy** to test the function.

2.6.4. Creating a Lambda function to post the report files

You must create a second Lambda function to post your filtered reports in a bucket that Red Hat can access.

Procedure

- Go to Lambda in the AWS console and click **Create function**.
- Click **Author from scratch**.
- Enter a name your function.
- From the **Runtime** menu, select the latest version of Python available.
- Select x86_64 as the Architecture.
- Under Permissions select the Athena role you created.
- Click **Create function**.
- Paste this [script](#) into the function and replace the following lines:

secret_name = "CHANGEME"

Enter your secret name.

bucket = "<your_S3_Bucket_Name>"

Enter the name of the S3 bucket you created to store filtered reports during the *Creating a bucket for storing filtered data reporting* step.

- Click **Deploy** to test the function.

2.7. CREATING EVENT BRIDGE SCHEDULES

You must trigger the Lambda functions you created by scheduling an AmazonEventBridge.

Procedure

1. Create two AmazonEventBridge schedules to trigger each of the functions that you created. You must trigger these functions at different cadences so that the Athena query is completed before it sends the reports:
 - a. Add a name and description.
 - b. In the **Group** field, select **Default**.
 - c. In the **Occurrence** field, select **Recurring schedule**.
 - d. In the **Type** field, select **Chron-based**.
 - e. Set the cron-based schedules 12 hours apart. The following example triggers the function at 9AM and 9PM, **0 9 * * ? *** and **0 21 * * ? ***.
 - f. Set a flexible time window.
 - g. Click **Next**.
2. Set the **Target detail** to **AWS Lambda invoke** to associate this schedule with the Lambda function:
 - a. Select the Lambda function you created before.
 - b. Click **Next**.
3. Enable the schedule:
 - a. Configure the retry logic.
 - b. Ignore the encryption.
 - c. Set the permissions to **Create new role on the fly**
 - d. Click **Next**.
4. Review your selections and click **Create**.

2.8. CREATING ADDITIONAL CLOUD FUNCTIONS TO COLLECT FINALIZED DATA

AWS sends final reports for the last month at the start of the following month. Send these finalized reports to Cost management, which will analyze the extra information.

Procedure

1. Create Athena query for the Lambda function:
 - a. Create a function for querying Athena.

- b. Select **Author from scratch**.
 - c. Select the **Python** runtime.
 - d. Select the **x86_64** architecture.
 - e. Select the role created before for permissions.
 - f. Click **Create**.
2. Click the **Code** tab to add a script to collect the finalized data.
 - a. Copy the [Athena query function](#) and add it to the query. Update the `<integration_uuid>` with the **integration_uuid** from the integration you created on console.redhat.com, which you can find by going to the [the Integrations page](#) and clicking your integration. Update the **BUCKET** and **DATABASE** variables with the bucket name and databases you created. Then, update **export_name** with the name of the data export Athena query you created before.

- b. Remove the comment from the following code:

```
# last_month = now.replace(day=1) - timedelta(days=1)
# year = last_month.strftime("%Y")
# month = last_month.strftime("%m")
# day = last_month.strftime("%d")
# file_name = 'finalized-data.json'
```

- c. Click **Deploy**. Then click **Test** to see the execution results.
3. Create a Lambda function to post the report files to cost management:
 - a. Select **Author from scratch**.
 - b. Name your function.
 - c. Select the **Python** runtime.
 - d. Select the **x86_64** architecture.
 - e. Select the role created before for permissions.
 - f. Click **Create**.
 4. Click the **Code** tab to add a script to post the finalized data.

- a. Copy the [post function](#) and add it to the query. Update the **secret_name** with the name of your secret in AWS Secrets Manager. Update the **bucket** with the bucket name you created.
- b. Remove the comment from the following code:

```
# file_name = 'finalized_data.json'
```

- c. Click **Deploy**. Then click **Test** to see the execution results.
5. Create an EventBridge schedule to trigger the two functions. For more information, see [Section 2.7, "Creating event bridge schedules"](#).
 - a. Set the EventBridge schedule to run one time a month on or after the 15th of the month

- a. Set the EventBridge schedule to run one time a month on or after the 15th of the month, because your AWS bill for the earlier period is final by that date. For example, **(0 9 15 * ? *)** and **(0 21 15 * ? *)**.

After completing these steps, cost management will begin collecting Cost and Usage data from your AWS account and any linked AWS accounts.



NOTE

The data can take a few days to populate before it shows on the [cost management](#) dashboard.

CHAPTER 3. NEXT STEPS FOR MANAGING YOUR COSTS

After adding your OpenShift Container Platform and Amazon Web Services data, cost management cost shows cost data by integration and cost and usage that is related to running your OpenShift Container Platform clusters on their platform. If you're using an AWS savings plan for the EC2 instances running OpenShift nodes, cost management defaults to using the savings plan cost.

On the [cost management Overview](#) page, your cost data is sorted into **OpenShift** and **Infrastructure** tabs. Select **Perspective** to toggle through different views of your cost data.

You can also use the global navigation menu to view additional details about your costs by cloud provider.

Additional resources

- [Integrating OpenShift Container Platform data into cost management](#)
- [Integrating Google Cloud data into cost management](#)
- [Integrating Microsoft Azure data into cost management](#)
- [Integrating Oracle Cloud data into cost management](#)

3.1. LIMITING ACCESS TO COST MANAGEMENT RESOURCES

After you add and configure integrations in cost management, you can limit access to cost data and resources.

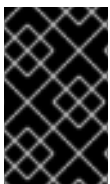
You might not want users to have access to all of your cost data. Instead, you can grant users access only to data that is specific to their projects or organizations. With role-based access control, you can limit the visibility of resources in cost management reports. For example, you can restrict a user's view to only AWS integrations, rather than the entire environment.

To learn how to limit access, see the more in-depth guide [Limiting access to cost management resources](#).

3.2. CONFIGURING TAGGING FOR YOUR INTEGRATIONS

The cost management application tracks cloud and infrastructure costs with tags. Tags are also known as labels in OpenShift.

You can refine tags in cost management to filter and attribute resources, organize your resources by cost, and allocate costs to different parts of your cloud infrastructure.



IMPORTANT

You can only configure tags and labels directly on an integration. You can choose the tags that you activate in cost management, however, you cannot edit tags and labels in the cost management application.

To learn more about the following topics, see [Managing cost data using tagging](#):

- Planning your tagging strategy to organize your view of cost data
- Understanding how cost management associates tags

- Configuring tags and labels on your integrations

3.3. CONFIGURING AWS BILLING PLANS

For more information about AWS billing, see [Understanding Consolidated Bills](#) in the AWS documentation.

Cost management supports three cost calculation options to accommodate AWS billing plans:

Unblended

Your costs are calculated according to your usage cost for that date.

Amortized (Default)

Your recurring and upfront costs will be distributed evenly throughout the billing period.

Blended

Your costs are calculated according to AWS blended rates.

This procedure describes how to set your cost calculation to **Amortized** or **Blended** from the default **Unblended**.

Prerequisites

- Access to [Red Hat Hybrid Cloud Console](#) as an Organization Administrator.

Procedure

1. From [Red Hat Hybrid Cloud Console](#), navigate to the [cost management settings](#) page.
2. Under **Show cost as** select **Amortized** or **Blended**.
3. Click **Save**.

3.4. CONFIGURING COST MODELS TO ACCURATELY REPORT COSTS

Now that you configured your integrations to collect cost and usage data in cost management, you can configure cost models to associate prices to metrics and usage.

A cost model is a framework that uses raw costs and metrics to define calculations for the costs in cost management. You can record, categorize, and distribute the costs that the cost model generates to specific customers, business units, or projects.

In [Cost Models](#), you can complete the following tasks:

- Classifying your costs as infrastructure or supplementary costs
- Capturing monthly costs for OpenShift nodes and clusters
- Applying a markup to account for additional support costs

To learn how to configure a cost model, see [Using cost models](#).

3.5. VISUALIZING YOUR COSTS WITH COST EXPLORER

Use cost management [Cost Explorer](#) to create custom graphs of time-scaled cost and usage information and ultimately better visualize and interpret your costs.



To learn more about the following topics, see [Visualizing your costs using Cost Explorer](#):

- Using Cost Explorer to identify abnormal events
- Understanding how your cost data changes over time
- Creating custom bar charts of your cost and usage data
- Exporting custom cost data tables

CHAPTER 4. UPDATING AN INTEGRATION

If you have added an integration to cost management and want to make changes to it, you can add or remove the applications associated with your integrations in [Red Hat Hybrid Cloud Console](#).

Procedure



1. From [Red Hat Hybrid Cloud Console](#), click **Settings** .
2. Click **Integrations**.
3. Click the more options menu  for your integration. Click **Edit**.
4. In **Metered Product**, select **Red Hat Enterprise Linux** from the drop-down to activate metering.

4.1. ADDING RHEL METERING TO AN AWS INTEGRATION

If you converted from a compatible third-party Linux distribution to Red Hat Enterprise Linux (RHEL) and purchased the RHEL for third party migration listing in Amazon Web Services (AWS), you can add RHEL metering to an AWS integration.

With RHEL metering, Red Hat processes your bill to meter your hourly RHEL usage associated with a Red Hat offering in AWS.

Procedure

1. In AWS, tag your instances of RHEL that you want to meter. For more information about tagging your instances of RHEL in AWS, see [Adding tags to an AWS resource](#).
2. From [Red Hat Hybrid Cloud Console](#), click **Settings** .
3. Click **Integrations**.
4. Click the more options menu  for your integration. Click **Edit**.
5. In **Metered Product**, select **Red Hat Enterprise Linux** from the drop-down to activate metering.

PROVIDING FEEDBACK ON RED HAT DOCUMENTATION

We appreciate and prioritize your feedback regarding our documentation. Provide as much detail as possible, so that your request can be quickly addressed.

Prerequisites

- You are logged in to the Red Hat Customer Portal.

Procedure

To provide feedback, perform the following steps:

1. Click the following link: [Create Issue](#).
2. Describe the issue or enhancement in the **Summary** text box.
3. Provide details about the issue or requested enhancement in the **Description** text box.
4. Type your name in the **Reporter** text box.
5. Click the **Create** button.

This action creates a documentation ticket and routes it to the appropriate documentation team. Thank you for taking the time to provide feedback.