



Cost Management Service 1-latest

Integrating Amazon Web Services (AWS) data into cost management

Learn how to add and configure your AWS integrations

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

Learn how to add and configure your AWS 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

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

CHAPTER 1. CREATING AN AMAZON WEB SERVICES INTEGRATION	3
1.1. ADDING AN AWS ACCOUNT AS AN INTEGRATION	3
1.2. CREATING AN S3 BUCKET AND A DATA EXPORT	4
1.3. ACTIVATING AWS TAGS	5
1.4. CONFIGURE AN IAM POLICY TO ENABLE MINIMAL ACCOUNT ACCESS FOR COST AND USAGE REPORTS	5
1.4.1. Enabling additional account access for cost and usage consumption	7
1.5. CONFIGURING AWS BILLING PLANS	8
CHAPTER 2. FILTERING YOUR AMAZON WEB SERVICES DATA BEFORE SENDING IT TO COST MANAGEMENT	9
2.1. ADDING AN AWS ACCOUNT AS AN INTEGRATION	9
2.2. CREATING AN AWS S3 BUCKET FOR STORING YOUR COST DATA	10
2.3. CREATING A DATA EXPORT FOR FILTERED DATA REPORTING	10
2.4. ACTIVATING AWS TAGS	11
2.5. ENABLING MINIMAL ACCOUNT ACCESS FOR COST AND USAGE CONSUMPTION	11
2.6. ENABLING ACCOUNT ACCESS FOR ATHENA	13
2.6.1. Configuring Athena for report generation	15
2.6.2. Creating a Lambda function for Athena	16
2.6.3. Creating a Lambda function to post the report files	17
CHAPTER 3. NEXT STEPS FOR MANAGING YOUR COSTS	20
3.1. LIMITING ACCESS TO COST MANAGEMENT RESOURCES	20
3.2. CONFIGURING TAGGING FOR YOUR INTEGRATIONS	20
3.3. CONFIGURING COST MODELS TO ACCURATELY REPORT COSTS	21
3.4. VISUALIZING YOUR COSTS WITH COST EXPLORER	21
CHAPTER 4. UPDATING AN INTEGRATION	22
4.1. ADDING RHEL METERING TO AN AWS INTEGRATION	22
4.2. ADDING RHEL METERING TO A MICROSOFT AZURE INTEGRATION	22
PROVIDING FEEDBACK ON RED HAT DOCUMENTATION	24

CHAPTER 1. CREATING AN AMAZON WEB SERVICES INTEGRATION

To add an Amazon Web Services (AWS) account to cost management, you must configure your AWS account to provide metrics, then add your AWS account as a cloud integration from the cost management user interface.



NOTE

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

When you add an AWS integration, you create a read-only connection to AWS so that cost management can collect your data hourly. This process does not make any changes to the AWS account.

To add your AWS account to cost management as an integration, you must configure the following services on your AWS account to allow cost management to have access to your metrics:

1. An S3 bucket to store cost and usage data reporting for cost management
2. An Identity Access Management (IAM) policy and role for cost management to process the cost and usage data

Since you will complete some of the following steps in the AWS console, and some steps in the cost management user interface, keep both applications open in a web browser.



NOTE

To ensure you have the most up to date information about AWS, refer to the [AWS documentation](#).


1.1. ADDING AN AWS ACCOUNT AS AN INTEGRATION

Add an AWS integration so the cost management application 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. Add an AWS integration so the hybrid committed spend application 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.
 - a. Select **Account authorization** to provide your AWS account credentials and let Red Hat configure and manage your integration for you. Click **Next**.
 - b. Select **Manual configuration** to customize your integration. You can filter your information before it is sent to cost management. For instructions on how to filter your data, see [Chapter 2, Filtering your Amazon Web Services data before sending it to cost management](#) . 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.
4. 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, and 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.

5. 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, click **Next**.

Additional resources

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

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

To provide data in the web interface and API, cost management must consume the Cost and Usage Reports produced by AWS. To only provide access to the stored information and nothing else, create an IAM policy and role for cost management to use.

Procedure

1. From the AWS Identity and Access Management (IAM) console, create a new IAM policy for the S3 bucket that you configured previously.
 - a. Select the JSON tab and paste the following content in the JSON policy text box:

```
{
  "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": "*"
}
]
}

```

1 Replace **<your_bucket_name>** in both locations with the name of the Amazon S3 bucket you configured previously.

- 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.
2. In the [Red Hat Hybrid Cloud Console Add a cloud integration wizard](#), click **Next**.
 3. 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 provide the cost management application with read access to the AWS account cost data.
 - c. Attach the IAM policy you just configured.
 - d. Enter a role name and description.
 4. In the [Red Hat Hybrid Cloud Console Add a cloud integration wizard](#), click **Next**.
 5. 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:**.
 6. In the [Red Hat Hybrid Cloud Console Add a cloud integration wizard](#), paste your Role ARN and click **Next**.
 7. Review the details and click **Finish** to add the AWS account to cost management.

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

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

1.4.1. Enabling additional account access for cost and usage consumption

Cost management can display additional data that might be useful. For example:

- Include the Action *iam:ListAccountAliases* to display an AWS account alias rather than an account number in cost management.
- Include the Actions *organization:List** and *organizations:Describe** to obtain the display names of AWS member accounts if you are using consolidated billing rather than the account ID.

The following configuration provides access to additional stored information and nothing else.

Procedure

1. From the AWS Identity and Access Management (IAM) console, create a new IAM policy for the S3 bucket you configured before.
2. Select the JSON tab and paste the following content in the JSON policy text box:

```
{
  "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": [
        "iam:ListAccountAliases",
        "s3:ListBucket",
        "cur:DescribeReportDefinitions",
        "organizations:List*",
        "organizations:Describe*"
      ],
      "Resource": "*"
    }
  ]
}
```

- 1 Replace **<your_bucket_name>** in both locations with the name of the Amazon s3 bucket you configured before.

The remainder of the configuration steps are the same as in [Section 1.4, "Configure an IAM policy to enable minimal account access for Cost and Usage Reports"](#)

1.5. CONFIGURING AWS BILLING PLANS

By default, cost management calculates AWS cost according to your usage cost for that date. If you have a special billing arrangement with AWS such as amortized billing or blended rates, you can configure these calculations from the [cost management settings](#) page. This allows your cost reports to more accurately reflect your AWS billing.

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

- AWS integration added to cost management.
- 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**.

CHAPTER 2. FILTERING YOUR AMAZON WEB SERVICES DATA BEFORE SENDING IT TO COST MANAGEMENT

To copy exports, object storage buckets, and filter your data to share only a subset of your billing information with Red Hat, configure a function script in AWS. This option is only recommended if your organization has third party data limitations.



NOTE

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

To configure your AWS account to be a cost management data integration:

- Create an AWS S3 bucket to store your cost data.
- Create an AWS S3 bucket to report your filtered cost management data.
- Configure IAM roles for your cost data bucket.
- Add your AWS integrations to [Red Hat Hybrid Cloud Console](#).
- Configure IAM roles for AWS Athena.
- Enable Athena.
- Create Lambda tasks for Athena to export filtered data to your S3 bucket.

Because you must complete some of the following steps in the AWS Console, and some steps in the cost management user interface, keep both applications open in a web browser.


2.1. ADDING AN AWS ACCOUNT AS AN INTEGRATION

Add an AWS integration so the cost management application 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**.

- a. Select **Manual configuration** to customize your integration. For example, you can filter your information before it is sent to cost management. Click **Next**.
5. In the **Select application** step, select **Cost management**. Click **Next**.

2.2. CREATING AN AWS S3 BUCKET FOR STORING YOUR COST DATA

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

Procedure

1. Log in to your AWS account to begin configuring cost and usage reporting.
2. 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.
3. 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:
 - Report name: *<rh_cost_report>* (note this name as you will use it later)
 - Additional report details: Include resource IDs
 - S3 bucket: *<the S3 bucket you configured previously>*
 - Time granularity: Hourly
 - Enable report data integration for: Amazon Redshift, Amazon QuickSight (do not enable report data integration for Amazon Athena)
 - Compression type: GZIP
 - Report path prefix: cost



NOTE

See the *AWS Billing and Cost Management* documentation for more details on configuration.

4. On the **Create storage** step, in the **Add a cloud integration** wizard, paste the name of your S3 bucket and select the region that it was created in. Click **Next**.
5. In the **Add a cloud integration** wizard, on the **Create cost and usage report** step, click **Next**.

2.3. CREATING A DATA EXPORT FOR FILTERED DATA REPORTING

To create a data export in AWS, set up Athena and Lambda functions to filter the data. This process delivers your data export to your S3 bucket. Complete the following steps:

Procedure

1. Log in to your AWS account.
2. In the AWS S3 console, create a data export that will be delivered to your S3 bucket.

3. Enter a report name. Save this name. You will use it later.
4. Click include resource IDs.
5. Click **Next**.
6. From Configure S3 Bucket, click Configure. Create a bucket and apply the default policy.
7. Click **Save**.
8. On the **Create storage** step, in the **Add a cloud integration** wizard, paste the name of your S3 bucket and select the region that it was created in and click **Next**.
9. 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** and 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, click **Next**.

Additional resources

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

2.5. ENABLING MINIMAL ACCOUNT ACCESS FOR COST AND USAGE CONSUMPTION

For cost management to provide data, it must consume the Cost and Usage Reports produced by AWS. For cost management to obtain this data with a minimal amount of access, create an IAM policy and role for cost management to use. This configuration only provides access to the stored information.

Procedure

1. From the AWS Identity and Access Management (IAM) console, create a new IAM policy for the S3 bucket that you configured.
 - a. Select the JSON tab and paste the following content in **JSON policy**.

```

{
  "Version": "2012-10-17", "Statement": [
    {
      "Effect": "Allow", "Action": [
        "s3:GetBucketLocation", "s3:ListAllMyBuckets"
      ], "Resource": "arn:aws:s3::*"
    }, {
      "Effect": "Allow", "Action": "s3:*", "Resource": [
        "arn:aws:s3:::<your_bucket_name>", "arn:aws:s3:::<your_bucket_name>/*" 1
      ]
    }
  ]
}

```

1 Replace **<your_bucket_name>** in both locations with the name of the Amazon S3 bucket you configured for storing your filtered data.

- a. Provide a name for the policy and complete the creation of the policy. Keep the AWS IAM console open. You will need it for the next step.
 - c. In the [Red Hat Hybrid Cloud Console Add a cloud integration wizard](#), click **Next**.
2. In the AWS IAM console, create a new IAM role:
 - a. For the type of trusted entity, select **AWS account**
 - b. Enter `589173575009` as the Account ID to provide the cost management application with read access to the AWS account cost data.
 - c. Attach the IAM policy you just configured.
 - d. Enter a role name and description and finish creating the role.
 - e. In the [Red Hat Hybrid Cloud Console Add a cloud integration wizard](#), click **Next**.
 3. In the AWS IAM console, from Roles, open the summary screen for the role you just created and copy the Role ARN. It is a string beginning with **arn:aws:**.
 4. In the [Red Hat Hybrid Cloud Console Add a cloud integration wizard](#), enter the ARN on the **Enter ARN** page and click **Next**.
 5. 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

To provide data within the web interface and API, 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 **2** Replace **CHANGE-ME*** in both locations with the ARN for the IAM policy for the Athena Lambda functions.

- b. Provide a name for 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.

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, navigate to the filtered bucket that you created and download the **crawler-cfn.yml** file.
2. From **CloudFormation** in the AWS console, create a new stack.
3. Select **Template as Ready**.
4. Upload the **crawler-cfn.yml** file that you previously downloaded. This should load immediately.
5. Click **Next**.
6. Enter a name and click **Next**.

- Click **I acknowledge that AWS Cloudformation might create IAM resources** and then click **Submit**.

2.6.2. 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

- Navigate to Lambda in the AWS console and click **Create function**.
- Click **Author from scratch**.
- Enter a name your function.
- From the Runtime dropdown, Select python 3.7.
- Select x86_64 as the Architecture.
- Under Permissions select the Athena role you created.
- Click **Create function**.
- Paste the following code to the function:

```
import boto3
import uuid
import json
from datetime import datetime

now = datetime.now()
year = now.strftime("%Y")
month = now.strftime("%m")
day = now.strftime("%d")

# Vars to Change!
integration_uuid = <your_integration_uuid> # integration_uuid
bucket = <your_S3_Bucket_Name> # Bucket created for query results
database = 'athenacurcfn_athena_cost_and_usage' # Database to execute athena
queries
output='s3://{bucket}/{year}/{month}/{day}/{uuid.uuid4()}' # Output location for query results

# Athena query
query = f"SELECT * FROM {database}.koku_athena 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 =
'{year}' AND month = '{month}'"

def lambda_handler(event, context):
```

```

# Initiate Boto3 athena Client
athena_client = boto3.client('athena')

# Trigger athena query
response = athena_client.start_query_execution(
    QueryString=query,
    QueryExecutionContext={
        'Database': database
    },
    ResultConfiguration={
        'OutputLocation': output
    }
)

# Save query execution to s3 object
s3 = boto3.client('s3')
json_object = {"integration_uuid": integration_uuid, "bill_year": year, "bill_month": month,
"query_execution_id": response.get("QueryExecutionId"), "result_prefix": output}
s3.put_object(
    Body=json.dumps(json_object),
    Bucket=bucket,
    Key='query-data.json'
)

return json_object

```

Replace **<your_integration_uuid>** with the UUID from the integration you created on console.redhat.com. Replace **<your_S3_Bucket_Name>** with the name of the S3 bucket you created to store reports.

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

2.6.3. Creating a Lambda function to post the report files

You must create a Lambda function to post your report files to the S3 bucket that you created.

Procedure

1. Navigate to Lambda in the AWS console and click **Create function**.
2. Click Author from scratch
3. Enter a name your function
4. From the Runtime dropdown, Select python 3.7.
5. Select x86_64 as the Architecture.
6. Under Permissions select the Athena role you created.
7. Click **Create function**.
8. Paste the following code to the function:

```

import boto3
import json

```

```

import requests
from botocore.exceptions import ClientError

def get_credentials(secret_name, region_name):
    session = boto3.session.Session()
    client = session.client(
        service_name='secretsmanager',
        region_name=region_name
    )
    try:
        get_secret_value_response = client.get_secret_value(
            SecretId=secret_name
        )
    except ClientError as e:
        raise e
    secret = get_secret_value_response['SecretString']
    return secret

secret_name = "CHANGEME"
region_name = "us-east-1"
secret = get_credentials(secret_name, region_name)
json_creds = json.loads(secret)

USER = json_creds.get("<your_username>")    # console.redhat.com Username
PASS = json_creds.get("<your_password>")    # console.redhat.com Password
bucket = "<your_S3_Bucket_Name>"          # Bucket for athena query results

def lambda_handler(event, context):
    # Initiate Boto3 s3 and fetch query file
    s3_resource = boto3.resource('s3')
    json_content = json.loads(s3_resource.Object(bucket, 'query-data.json').get()
['Body'].read().decode('utf-8'))

    # Initiate Boto3 athena Client and attempt to fetch athena results
    athena_client = boto3.client('athena')
    try:
        athena_results =
athena_client.get_query_execution(QueryExecutionId=json_content["query_execution_id"])
    except Exception as e:
        return f"Error fetching athena query results: {e} \n Consider increasing the time between
running and fetching results"

    reports_list = []
    prefix = json_content["result_prefix"].split(f'{bucket}')[0][-1]

    # Initiate Boto3 s3 client
    s3_client = boto3.client('s3')
    result_data = s3_client.list_objects(Bucket=bucket, Prefix=prefix)
    for item in result_data.get("Contents"):
        if item.get("Key").endswith(".csv"):
            reports_list.append(item.get("Key"))

    # Post results to console.redhat.com API
    url = "https://console.redhat.com/api/cost-management/v1/ingress/reports/"

```

```
    json_data = {"source": json_content["integration_uuid"], "reports_list": reports_list,
                "bill_year": json_content["bill_year"], "bill_month": json_content["bill_month"]}
    resp = requests.post(url, json=json_data, auth=(USER, PASS))

    return resp
```

Replace **<your_username>** with your username for console.redhat.com. Replace **<your_password>** with your password for console.redhat.com. Replace **<your_S3_Bucket_Name>** with the name of the S3 bucket that you created to store reports.

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

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 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](#).

3.4. 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.

4.2. ADDING RHEL METERING TO A MICROSOFT AZURE 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 Microsoft Azure, you can update an Microsoft Azure integration you created to add RHEL metering.

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

Procedure

1. In Microsoft Azure, tag your instances of RHEL that you want to meter. For more information about tagging your instances of RHEL in Microsoft Azure, see [Adding tags to a Microsoft Azure 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

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!