



# Cost Management Service 1-latest

## 将 Microsoft Azure 数据集成到成本管理中

了解如何添加和配置 Microsoft Azure 集成



# Cost Management Service 1-latest 将 Microsoft Azure 数据集成到成本管理中

---

了解如何添加和配置 Microsoft Azure 集成

## 法律通告

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<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

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

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> 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<sup>®</sup> 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.

## 摘要

了解如何将 Microsoft Azure 集成添加到成本管理。

---

# 目录

<b>第 1 章 创建 MICROSOFT AZURE 集成</b> .....	<b>3</b>
1.1. 添加 MICROSOFT AZURE 帐户并命名您的集成	3
1.2. 创建 MICROSOFT AZURE 资源组和存储帐户	3
1.3. 在 MICROSOFT AZURE 中创建每日导出	4
1.4. 查找 MICROSOFT AZURE 订阅 ID	4
1.5. 配置 MICROSOFT AZURE 角色	5
<b>第 2 章 在将 MICROSOFT AZURE 数据集成到成本管理前过滤 MICROSOFT AZURE 数据</b> .....	<b>6</b>
2.1. 添加 MICROSOFT AZURE 帐户并命名您的集成	6
2.2. 创建 MICROSOFT AZURE 资源组和存储帐户	6
2.3. 查找 MICROSOFT AZURE 订阅 ID	7
2.4. 为您的存储帐户创建 MICROSOFT AZURE 角色	7
2.5. 在 MICROSOFT AZURE 中创建每日导出	8
2.6. 在 MICROSOFT AZURE 中创建功能来过滤您的数据	8
2.7. 配置 MICROSOFT AZURE 角色	11
<b>第 3 章 管理成本的后续步骤</b> .....	<b>13</b>
3.1. 限制对成本管理资源的访问	13
3.2. 为您的集成配置标记	13
3.3. 配置成本模型以准确报告成本	13
3.4. 使用 COST EXPLORER 可视化您的成本	14
<b>第 4 章 更新集成</b> .....	<b>15</b>
4.1. 在 AWS 集成中添加 RHEL METERING	15
4.2. 在 MICROSOFT AZURE 集成中添加 RHEL METERING	15
<b>对红帽文档提供反馈</b> .....	<b>17</b>



# 第1章 创建 MICROSOFT AZURE 集成

配置 Microsoft Azure 帐户，以允许成本管理访问。

将您的 Microsoft Azure 帐户配置为成本管理集成需要：

1. 创建存储帐户和资源组
2. 为您的成本导出选择适当的范围
3. 配置存储帐户 Contributor 和 Reader 角色来访问
4. 调度每日成本导出



## 注意

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

从 [Integrations 页面](#) 将 Microsoft Azure 集成添加到成本管理。

## 1.1. 添加 MICROSOFT AZURE 帐户并命名您的集成

将 Microsoft Azure 帐户添加为集成，以便成本管理可以处理成本和使用数据。

### 流程

1. 在 [Red Hat Hybrid Cloud Console](#) 中，点 **Settings Menu**  > **Integrations**。
2. 在 **Settings** 页面中，在 **Cloud** 选项卡中点 **Add integration**。
3. 点 **Add integration**。
4. 在 **Add a cloud integration** 向导中，选择 **Microsoft Azure** 作为云供应商类型，然后点 **Next**。
5. 输入您的集成名称并点 **Next**。
6. 在 **Select application** 步骤中，选择 **Cost management** 并点 **Next**。
7. 在 **Specify cost export scope** 步骤中，选择 **I am OK**，将默认数据发送到 **Cost Management**。
8. 从菜单中选择您的成本数据导出的范围。您可以在订阅中的订阅级别或其他范围中导出数据。
  - a. 为您选择的范围复制生成的命令。
  - b. 在 [Microsoft Azure 帐户](#) 中，点 **Cloud Shell** 并粘贴并运行您在上一步中复制的命令。
9. 在 **Add a cloud integration** 向导中，将 Microsoft Azure 返回的值粘贴到 **Specify cost export scope** 步骤上的 **Cost export scope** 字段。
10. 点击 **Next**。

## 1.2. 创建 MICROSOFT AZURE 资源组和存储帐户

在 Microsoft Azure 中创建存储帐户，以保存成本数据和指标，并存储您过滤的成本数据，其中会收集成本管理。在成本管理中的 **Add a cloud integration** 向导中，在 Resource group 和 storage account 页面的字段中输入资源组名称和存储帐户名称。

## 先决条件

您必须具有具有 Cloud Administrator 权利的红帽用户帐户。

## 流程

1. 在 [Microsoft Azure 帐户](#) 中，搜索 **storage** 并点 **Storage account**。
2. 在 **Storage accounts** 页面上，单击 **Create**。
3. 在 **Create a storage account** 页面上的 **Resource Group** 字段中，单击 **Create new**。输入名称，然后单击**确定**。在本例中，使用 **cost-data-group**。
4. 在实例详情部分中，在 **Storage account name** 字段中输入名称。在本例中，使用 **costdata**。
5. 复制资源组和存储帐户的名称，以便您可以将它们添加到 [Red Hat Hybrid Cloud Console](#) 的 **Add a cloud integration** 向导中，然后点 **Review**。
6. 检查存储帐户并点 **Create**。
7. 在 [Red Hat Hybrid Cloud Console](#) **Add a cloud integration** 向导的 **Resource group and storage account** 页面中，在 **Resource group name** 和 **Storage account name** 中输入值。
8. 单击 **Next**。

## 1.3. 在 MICROSOFT AZURE 中创建每日导出

在 Microsoft Azure 中创建功能，以过滤您的数据并根据常规计划导出数据。exports 创建一个周期性任务，将 Microsoft Azure 成本数据定期发送到存储帐户，该存储帐户存在于资源组中。成本管理必须能够访问资源组才能读取 Microsoft Azure 成本数据。这个示例使用 Python 功能来过滤数据并将其发布到之前创建的存储帐户。

## 流程

1. 要创建导出，请转至 Microsoft Azure 中的 **Portal** 菜单，然后点 **Cost Management + Billing**。
2. 在 **Cost Management + Billing** 页面上，单击 **Cost Management**。
3. 在 **Settings** 菜单中，在 **Cost 管理概览** 页面中，单击 **Exports**。
4. 要添加导出，请单击 **Add**。
5. 在 **Export details** 部分中，将导出命名为：
6. 在 **Storage** 部分，添加您创建的资源组。

## 1.4. 查找 MICROSOFT AZURE 订阅 ID

在 Microsoft Azure Cloud Shell 中找到 **subscription\_id**，并将其添加到成本管理中的 **Add a cloud integration** 向导中。

## 流程

1. 在 [Microsoft Azure 帐户](#) 中，点 **Cloud Shell**。
2. 输入以下命令获取您的订阅 ID：

```
az account show --query "{subscription_id: id}"
```

3. 从返回的数据复制 **subscription\_id** 的值。

### 响应示例

```
{  
  "subscription_id": 00000000-0000-0000-000000000000  
}
```

4. 将该值粘贴到 **Add a cloud integration** 向导的 **Subscription ID** 页面的 Subscription ID 字段中。
5. 点击 **Next**。

## 1.5. 配置 MICROSOFT AZURE 角色

要授予红帽对数据的访问权限，您必须在 Microsoft Azure 中配置专用角色。如果您已在 Microsoft Azure 中创建了服务主体，请不要创建另一个服务主体，跳到第 4 步，并使用它来共享您的 **租户(Directory) ID、客户端 (应用程序) ID**，以及带有成本管理的 **客户端 secret** 值。

## 流程

1. 在 **Add a cloud integration** 向导的 **Roles** 页面中，从向导中复制 **az ad sp create-for-rbac** 命令，以创建具有成本管理存储帐户 Contributor 角色的服务主体。
2. 在 [Microsoft Azure 帐户](#) 中，点 **Cloud Shell**。
3. 将您在上一步中复制的命令粘贴到云 shell 提示符中。
4. 复制 Tenant (Directory) ID、Client (Application) ID 和 Client secret 值，并在 **Add a cloud integration** 向导的 **Roles** 页面中粘贴值。
5. 从向导中复制第二个生成的 **az role assignment create** 命令，并将它粘贴到 Microsoft Azure 中的云 shell 提示符中，以创建 Cost Management Reader 角色。
6. 在 **Add a cloud integration** 向导中，点 **Next**。
7. 查看向导中提供的信息，然后点 **Add**。

## 第 2 章 在将 MICROSOFT AZURE 数据集成到成本管理前过滤 MICROSOFT AZURE 数据

要使用 RH 共享您的计费数据的子集，您可以在 Microsoft Azure 中配置功能脚本。此脚本复制导出一个对象存储存储桶，该存储桶成本管理可以访问和过滤。

集成 Microsoft Azure 帐户：

1. 创建存储帐户和资源组。
2. 配置存储帐户 Contributor 和 Reader 角色来访问。
3. 创建一个功能来过滤您要发送给红帽的数据。
4. 将每日成本导出调度到红帽可访问的存储帐户。



### 注意

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

从 [Integrations 页面](#)将 Microsoft Azure 集成添加到成本管理。

### 2.1. 添加 MICROSOFT AZURE 帐户并命名您的集成

将 Microsoft Azure 帐户添加为集成，以便成本管理可以处理成本和使用数据。

#### 流程

1. 在 [Red Hat Hybrid Cloud Console](#) 中，点 **Settings Menu**  > **Integrations**。
2. 在 **Settings** 页面中，在 **Cloud** 选项卡中点 **Add integration**。
3. 在 **Cloud** 选项卡中，点 **Add integration**。
4. 在 **Add a cloud integration** 向导中，选择 **Microsoft Azure** 作为云供应商类型，然后点 **Next**。
5. 输入您的集成名称并点 **Next**。
6. 在 **Select application** 步骤中，选择 **Cost management** 并点 **Next**。
7. 在 **Specify cost export scope** 步骤中，选择 **I want to manually custom the data set sent to Cost Management** 并点 **Next**。

### 2.2. 创建 MICROSOFT AZURE 资源组和存储帐户

在 Microsoft Azure 中创建存储帐户，以存放您的账单导出以及第二个存储帐户来存放您的过滤的数据，以便成本管理可以收集信息。在成本管理中的 **Add a cloud integration** 向导中，在 **Resource group** 和 **storage account** 页面的字段中输入资源组名称和存储帐户名称。

#### 先决条件

您必须具有具有 Cloud Administrator 权利的红帽用户帐户。

## 流程

1. 在 [Microsoft Azure 帐户](#) 中，搜索 **storage** 并点 **Storage account**。
  - a. 在 **Storage accounts** 页面上，单击 **Create**。
  - b. 在 **Resource Group** 字段中，单击 **Create new**。输入名称，然后单击**确定**。在本例中，使用 **filtered-data-group**。
  - c. 在 **Instance details** 部分中，在 **Storage account name** 字段中输入名称。例如，使用 **filterreddata**。
  - d. 复制资源组和存储帐户的名称，以便您可以将它们添加到 [Red Hat Hybrid Cloud Console](#) 的 **Add a cloud integration** 向导中，然后点 **Review**。
  - e. 检查存储帐户并点 **Create**。
2. 在 [Red Hat Hybrid Cloud Console](#) **Add a cloud integration** 向导的 **Resource group and storage account** 页面中，在 **Resource group name** 和 **Storage account name** 中输入值。
3. 单击 **Next**。

## 2.3. 查找 MICROSOFT AZURE 订阅 ID

在 Microsoft Azure Cloud Shell 中找到 **subscription\_id**，并将其添加到成本管理中的 **Add a cloud integration** 向导中。

### 流程

1. 在 [Microsoft Azure 帐户](#) 中，点 **Cloud Shell**。
2. 输入以下命令获取您的订阅 ID：

```
az account show --query "{subscription_id: id}"
```

3. 从返回的数据复制 **subscription\_id** 的值。

### 响应示例

```
{
  "subscription_id": 00000000-0000-0000-000000000000
}
```

4. 将该值粘贴到 **Add a cloud integration** 向导的 **Subscription ID** 页面的 **Subscription ID** 字段中。
5. 单击 **Next**。

## 2.4. 为您的存储帐户创建 MICROSOFT AZURE 角色

使用 Microsoft Azure Cloud Shell 查找您的租户 (**Directory**) ID、**Client** (**应用程序**) ID 和 **Client secret**。

### 流程

1. 在 [Microsoft Azure 帐户](#) 中，点 **Cloud Shell**。

2. 输入以下命令获取客户端 ID、secret 和租户名称。将值替换为最后一步中的订阅 ID，将 **resourceGroup1** 替换为之前创建的资源组名称。在本例中，使用 **filtered-data-group**。

```
az ad sp create-for-rbac -n "CostManagement" --role "Storage Account Contributor" --scope /subscriptions/{subscriptionId}/resourceGroups/{resourceGroup1} --query '{"tenant": tenant, "client_id": appld, "secret": password}'
```

3. 复制 **client\_id**、**secret** 和 **租户** 的返回数据中的值。

#### 响应示例

```
{
  "client_id": "00000000-0000-0000-000000000000",
  "secret": "00000000-0000-0000-000000000000",
  "tenant": "00000000-0000-0000-000000000000"
}
```

4. 在 [Red Hat Hybrid Cloud Console](#) 中**添加云集成** 向导中的 **Role** 步骤中，粘贴 **client\_id**、**secret** 和 **'tenant'** 的值。
5. 在 Cloud shell 中运行以下命令，以创建成本管理 Reader 角色，并将 **{Client ID}** 替换为上一步中的值。

```
az role assignment create --assignee {Client_ID} --role "Cost Management Reader"
```

6. 点击 **Next**。

## 2.5. 在 MICROSOFT AZURE 中创建每日导出

在 Microsoft Azure 中创建功能，以过滤您的数据并根据常规计划导出数据。exports 创建一个周期性任务，将 Microsoft Azure 成本数据定期发送到存储帐户，该存储帐户存在于资源组中。成本管理必须能够访问资源组才能读取 Microsoft Azure 成本数据。这个示例使用 Python 功能来过滤数据并将其发布到之前创建的存储帐户。

### 流程

1. 要创建导出，请转至 Microsoft Azure 中的 **Portal** 菜单，然后点 **Cost Management + Billing**。
2. 在 **Cost Management + Billing** 页面上，单击 **Cost Management**。
3. 在 **Settings** 菜单中，在 **Cost 管理概览** 页面中，单击 **Exports**。
4. 要添加导出，请单击 **Add**。
5. 在 **Export details** 部分中，将导出命名为：
6. 在 **Storage** 部分，添加您创建的资源组。

## 2.6. 在 MICROSOFT AZURE 中创建功能来过滤您的数据

创建过滤数据的功能，并将其添加到您创建的要与红帽共享的存储帐户中。您可以使用示例 Python 脚本从与红帽费用相关的成本导出中收集成本数据，并将其添加到存储帐户中。

## 先决条件

- 在您的设备上必须安装 Visual Studio Code。
- 您必须在 Visual Studio Code 中安装 Microsoft Azure 功能扩展。

## 流程

1. 登录到您的 [Microsoft Azure 帐户](#)。
2. 在搜索栏中输入 **functions**，选择 **Functions**，然后单击 **Create**。
3. 为您的功能选择一个托管选项，然后点 **Select**。
4. 在 Create Function App 页面中，通过添加资源组来配置您的功能应用程序。
  - a. 在 **Instance Details** 部分中，将功能命名为 app。
  - b. 在 **Runtime stack** 中，选择 **Python**
  - c. 在 **Version** 中，选择 **3.10**。
5. 点 **Review + create**:
  - a. 点 **Create**。
  - b. 点 **Go to resource** 来配置函数。
6. 在功能应用程序菜单中，单击 **Functions** 来创建时间触发器功能：
  - a. 点 **Create**。
  - b. 在开发环境字段中，选择 **VSCode**。
7. 打开 Visual Studio Code，并确保安装了 Microsoft Azure Functions Visual Studio Code 扩展。要创建 Azure 功能，Microsoft 建议使用 Microsoft Visual Studio Code IDE 来开发和部署代码。有关配置 Visual Studio Code 的更多信息，请参阅 [Quickstart: 使用 Visual Studio Code 在 Azure 中使用 Python 创建功能](#)。
  - a. 点 Visual Studio Code 中的 Microsoft Azure 选项卡，登录到 Azure。
  - b. 在 Visual Studio Code 的 Workspaces 选项卡中，点 **Create function**。
  - c. 按照提示为您的功能设置本地位置，并为功能选择语言和版本。在本例中，选择 **Python**，对于并选择 **Python 3.9**。
  - d. 在 **Select a template for your project first function**对话框中，选择 **Timer trigger**，命名功能，然后按 **Enter**
  - e. 为希望该函数运行时设置 cron 表达式。在本例中，使用 **0\*9\*\*\*** 在每天上午 9 点运行该功能。
  - f. 点 **Create**。
8. 在开发环境中创建功能后，打开 **requirements.txt** 文件，添加以下要求并保存文件：

```
azure-functions
pandas
requests
```

```
azure-identity
azure-storage-blob
```

9. 打开 `__init__.py` 并粘贴以下 Python 脚本。将标记为 **# Required vars** 的部分中的值更改为您的环境的值。对于 **USER** 和 **PASS** 值，您可以选择使用 [Key Vault 凭据](#) 将用户名和密码配置为环境变量。

```
import datetime
import logging
import uuid
import requests
import pandas as pd
from azure.identity import DefaultAzureCredential
from azure.storage.blob import BlobServiceClient, ContainerClient

import azure.functions as func

def main(mytimer: func.TimerRequest) -> None:
    utc_timestamp = datetime.datetime.utcnow().replace(
        tzinfo=datetime.timezone.utc).isoformat()

    default_credential = DefaultAzureCredential()

    now = datetime.datetime.now()
    year = now.strftime("%Y")
    month = now.strftime("%m")
    day = now.strftime("%d")
    output_blob_name=f"{year}/{month}/{day}/{uuid.uuid4()}.csv"

    # Required vars to update
    USER = os.getenv('UsernameFromVault') # Cost management
    username
    PASS = os.getenv('PasswordFromVault') # Cost management
    password
    integration_id = "<your_integration_id>" # Cost management
    integration_id
    cost_export_store = "https://<your-cost-export-storage-account>.blob.core.windows.net"
    # Cost export storage account url
    cost_export_container = "<your-cost-export-container>" # Cost
    export container
    filtered_data_store = "https://<your_filtered_data_container-storage-account>.blob.core.windows.net" # Filtered data storage account url
    filtered_data_container = "<your_filtered_data_container>" # Filtered
    data container

    # Create the BlobServiceClient object
    blob_service_client = BlobServiceClient(filtered_data_store, credential=default_credential)
    container_client = ContainerClient(cost_export_store, credential=default_credential,
    container_name=cost_export_container)

    blob_list = container_client.list_blobs()
    latest_blob = None
    for blob in blob_list:
        if latest_blob:
            if blob.last_modified > latest_blob.last_modified:
```

```

        latest_blob = blob
    else:
        latest_blob = blob

    bc = container_client.get_blob_client(blob=latest_blob)
    data = bc.download_blob()
    blobject = "/tmp/blob.csv"
    with open(blobject, "wb") as f:
        data.readinto(f)
    df = pd.read_csv(blobject)

    filtered_data = df.loc[((df["publisherType"] == "Marketplace") &
((df["publisherName"].astype(str).str.contains("Red Hat")) | (((df["publisherName"] ==
"Microsoft") | (df["publisherName"] == "Azure")) &
(df["meterSubCategory"].astype(str).str.contains("Red Hat") |
df["serviceInfo2"].astype(str).str.contains("Red Hat")))))))]

    filtered_data_csv = filtered_data.to_csv(index_label="idx", encoding = "utf-8")

    blob_client = blob_service_client.get_blob_client(container=filtered_data_container,
blob=output_blob_name)

    blob_client.upload_blob(filtered_data_csv, overwrite=True)

    # Post results to console.redhat.com API
    url = "https://console.redhat.com/api/cost-management/v1/ingress/reports/"
    json_data = {"source": integration_id, "reports_list": [f"
{filtered_data_container}/{output_blob_name}", "bill_year": year, "bill_month": month]}
    resp = requests.post(url, json=json_data, auth=(USER, PASS))
    logging.info(f'Post result: {resp}')

    if mytimer.past_due:
        logging.info('The timer is past due!')

    logging.info('Python timer trigger function ran at %s', utc_timestamp)

```

10. 保存该文件。

11. 将功能部署到 Microsoft Azure。

## 2.7. 配置 MICROSOFT AZURE 角色

配置专用凭证，以授予您的功能 blob 访问权限 Microsoft Azure 成本数据，以便可以将数据从原始存储容器传输到过滤的存储容器。

### 流程

1. 在 [Microsoft Azure 帐户](#) 中，在搜索栏中输入 **功能**。
2. 查找您的功能并选择它。
3. 在 **Settings** 菜单中，单击 **Identity**。
4. 在 Identity 页面上，单击 **Azure 角色分配**。

5. 在 **Role assignments** 页面上，单击 **Add role assignment**。
6. 在 **Scope** 字段中，选择 **Storage scope**。
7. 在 **Resource** 字段中，选择您创建的存储帐户。在本例中，使用 **filtereddata**。
8. 在 **role** 字段中，选择 **Storage Blob Data Contributor**。
9. 单击 **Save**。
10. 重复这些步骤，为 **Storage Queue Data Contributor** 创建角色。
11. 对您创建的其他存储帐户重复此步骤。在本例中，使用 **billingexportdata**。
12. 在 [Red Hat Hybrid Cloud Console](#) 的 **Add a cloud integration** 向导中，点 **Next**。
13. 查看向导中提供的信息，然后点 **Add**。

## 第 3 章 管理成本的后续步骤

添加 OpenShift Container Platform 和 Microsoft Azure 集成后，除了通过集成显示成本数据外，成本管理会自动显示与在其平台上运行 OpenShift Container Platform 集群相关的 Azure 成本和使用量。

在 [成本管理 概述](#) 页上，您的成本数据按照 **OpenShift** 和 **基础架构** 选项卡进行排序。选择 **Perspective** 通过成本数据的不同视图切换。

您还可以使用全局导航菜单查看云供应商成本的更多详情。

### 其他资源

- [将 OpenShift Container Platform 数据集成到成本管理中](#)
- [将 Amazon Web Services \(AWS\) 数据集成到成本管理中](#)
- [将 Google Cloud 数据集成到成本管理中](#)
- [将 Oracle Cloud 数据集成到成本管理](#)

### 3.1. 限制对成本管理资源的访问

在成本管理中添加和配置集成后，您可以限制对成本数据和资源的访问。

您可能不希望用户访问所有成本数据。相反，您只能向用户授予特定于其项目或机构的数据的访问权限。通过基于角色的访问控制，您可以限制成本管理报告中的资源的可见性。例如，您可以将用户的视图限制为只有 AWS 集成，而不是整个环境。

要了解如何限制访问，请参阅更深入的指南 [限制对成本管理资源的访问](#)。

### 3.2. 为您的集成配置标记

成本管理应用程序通过标签跟踪云和基础架构成本。在 OpenShift 中，标签也称为标签。

您可以在成本管理中优化标签，过滤和属性资源，按成本组织资源，并为云基础架构的不同部分分配成本。



#### 重要

您只能直接在集成上配置标签和标签。您可以选择在成本管理中激活的标签，但无法在成本管理应用程序中编辑标签和标签。

要了解更多信息有关以下主题的信息，[请参阅使用标记管理成本数据](#)：

- 规划标记策略以组织您的成本数据视图
- 了解成本管理关联标签的方式
- 在集成上配置标签和标签

### 3.3. 配置成本模型以准确报告成本

现在，您已将集成配置为以成本管理方式收集成本和使用数据，您可以配置成本模型，将价格与指标和使用相关联。

成本模型是一个框架，它使用原始成本和指标来定义成本管理成本的计算。您可以记录、分类和分发成本模型给特定客户、业务单元或项目产生的成本。

在 [成本](#) 模型中，您可以完成以下任务：

- 将成本分类为基础架构或补充成本
- 捕获 OpenShift 节点和集群的每月成本
- 应用标记以考虑其他支持成本

要了解如何配置成本模型，[请参阅使用成本模型](#)。

### 3.4. 使用 COST EXPLORER 可视化您的成本

使用成本管理 [Cost Explorer](#) 创建时间扩展成本和使用信息的自定义图形，并最终视觉化并解释您的成本。

要了解更多有关以下主题的信息，[请参阅使用 Cost Explorer 可视化您的成本](#)：

- 使用 Cost Explorer 识别异常事件
- 了解如何随着时间推移您的成本数据变化
- 为您的成本和使用数据创建自定义条图表
- 导出自定义成本数据表

## 第 4 章 更新集成

如果您添加了成本管理的集成，并希望更改它，您可以在 [Red Hat Hybrid Cloud Console](#) 中添加或删除与集成关联的应用程序。

### 流程

1. 在 [Red Hat Hybrid Cloud Console](#) 中，点 **Settings** 。
2. 单击 **Integrations**。
3. 选择您的集成。
4. 在集成详情页面中，您可以添加或删除与您的集成关联的服务。对于 AWS 集成，选择 **Include RHEL usage data** 以激活 RHEL metering。

### 4.1. 在 AWS 集成中添加 RHEL METERING

如果您从兼容第三方 Linux 发行版转换为 Red Hat Enterprise Linux (RHEL)，并在 Amazon Web Services (AWS) 中为第三方迁移列表购买 RHEL，您可以更新您创建的 AWS 集成来添加 RHEL metering。

通过 RHEL metering，红帽处理您的 bill 以按小时的 RHEL 使用情况在 AWS 中与红帽产品进行计量。

### 流程

1. 在 AWS 中，标记您要量表的 RHEL 实例。有关在 AWS 中标记 RHEL 实例的更多信息，请参阅 [向 AWS 资源添加标签](#)。
2. 在 [Red Hat Hybrid Cloud Console](#) 中，点 **Settings** 。
3. 单击 **Integrations**。
4. 选择您的集成。
5. 在 **Integration details** 页面中，选择 **Include RHEL usage data to include RHEL metering**。

### 4.2. 在 MICROSOFT AZURE 集成中添加 RHEL METERING

如果您从兼容的第三方 Linux 发行版转换为 Red Hat Enterprise Linux (RHEL)，并在 Microsoft Azure 中为第三方迁移列表购买 RHEL，您可以更新您创建的 Microsoft Azure 集成来添加 RHEL metering。

通过 RHEL metering，红帽处理您的 bill 以按小时的 RHEL 使用情况在 Microsoft Azure 中与红帽产品进行计量。

### 流程

1. 在 Microsoft Azure 中，标记您要量表的 RHEL 实例。有关在 Microsoft Azure 中标记 RHEL 实例的更多信息，请参阅 [向 Microsoft Azure 资源添加标签](#)。
2. 在 [Red Hat Hybrid Cloud Console](#) 中，点 **Settings** 。
3. 单击 **Integrations**。

4. 选择您的集成。
5. 在 **Integration details** 页面中，选择 **Include RHEL usage data** to include RHEL metering。

## 对红帽文档提供反馈

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

非常感谢您的反馈意见！