



Cost Management Service 1-latest

Amazon Web Services (AWS) データの Cost Management への統合

AWS 統合と RHEL メータリングを追加する方法を学ぶ

Cost Management Service 1-latest Amazon Web Services (AWS) データの Cost Management への統合

AWS 統合と RHEL メータリングを追加する方法を学ぶ

法律上の通知

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.

概要

Amazon Web Services (AWS) インテグレーションを Cost Management へ追加する方法を説明します。Cost Management は、Red Hat Insights ポートフォリオサービスに含まれます。高度な分析ツールである Red Hat Insights スイートは、運用、セキュリティー、およびビジネスへの影響を特定して優先順位を付けるのに役立ちます。

目次

パート I. 基本または高度な AWS インテグレーションの選択	3
第1章 AMAZON WEB SERVICES インテグレーションの作成: 基本	4
1.1. AWS アカウントのインテグレーションとしての追加	4
1.2. S3 バケットおよびデータエクスポートの作成	5
1.3. AWS タグのアクティベーション	5
1.4. コストおよび使用状況レポートのアカウントアクセスを有効にするための IAM ポリシーの設定	6
第2章 AMAZON WEB SERVICES インテグレーションの作成: 高度	9
2.1. AWS アカウントのインテグレーションとしての追加	9
2.2. ATHENA の請求データを保存するための AWS S3 バケットの作成	10
2.3. フィルターされたデータレポートを保存するためのバケットの作成	10
2.4. AWS タグのアクティベーション	11
2.5. コストおよび使用状況レポートのアカウントアクセスを有効にするための IAM ポリシーの設定	11
2.6. ATHENA のアカウントアクセスの有効化	13
2.7. イベントブリッジスケジュールの作成	20
2.8. 最終的なデータを収集するクラウド機能の追加作成	20
第3章 コストを管理するための次のステップ	23
3.1. COST MANAGEMENT リソースへのアクセス制限	23
3.2. インテグレーションのタグ付けの設定	23
3.3. AWS 請求プランの設定	24
3.4. コストを正確にレポートするためのコストモデルの設定	24
3.5. COST EXPLORER を使用したコストの可視化	25
第4章 インテグレーションの更新	26
4.1. AWS 統合への RHEL メータリングの追加	26
RED HAT ドキュメントへのフィードバック (英語のみ)	27

パート I. 基本または高度な AWS インテグレーションの選択

AWS インテグレーションを作成するには、まず基本的なインテグレーションパスと高度なインテグレーションパスのどちらを使用するかを決定します。

基本

基本オプションについては、[Amazon Web Services インテグレーションの作成: 基本](#)に進んでください。

基本パスを使用すると、Cost Management は指定のスコープで AWS から課金レポートを直接読み取ることができます。

高度

高度オプションについては、[Amazon Web Services インテグレーションの作成: 高度](#)に進んでください。

高度なパスを使用すると、Cost Management がデータを読み取る前に、データをカスタマイズしたりフィルタリングしたりできます。特定の Red Hat 製品とのみ課金データを共有する場合は、高度なパスを使用することも可能です。高度なパスでは、セットアップと設定がより複雑になります。



注記

基本または高度のいずれかを選択する必要があります。両方を選択することはできません。

第1章 AMAZON WEB SERVICES インテグレーションの作成: 基本



重要

高度なパスを使用して AWS インテグレーションを作成する場合は、次の手順を実行しないでください。代わりに、Amazon Web Services インテグレーションの作成：高度に移動します。

RHEL メータリングを使用している場合は、データを Cost Management と統合してから、[AWS インテグレーションへの RHEL メータリングの追加](#)に進み、RHEL メータリングのインテグレーションの設定を完了します。

[Integration ページ](#) から Cost Management 用に AWS インテグレーションを作成し、Cost Management のアクセスを許可するように AWS アカウントを設定する必要があります。

AWS はサードパーティーの製品であり、UI およびドキュメントが変更される可能性があります。サードパーティーインテグレーションの設定手順は、公開時点において正確な情報です。最新の情報は、[AWS ドキュメント](#) を参照してください。

前提条件

- [Red Hat Hybrid Cloud Console サービスアカウント](#) がある。
- Cost Management にデータインテグレーションを追加するには、Cloud Administrator 権限を持つ Red Hat アカウントが必要です。


1.1. AWS アカウントのインテグレーションとしての追加

AWS インテグレーションを追加して、cost management が AWS アカウントからコストおよび使用状況レポートを処理できるようにします。AWS アカウントの認証情報を指定することで、AWS インテグレーションを自動的に追加できます。

前提条件

- Cost Management にデータインテグレーションを追加するには、Cloud Administrator 権限を持つ Red Hat アカウントが必要です。

手順

1. [Red Hat Hybrid Cloud Console](#) から、**Settings Menu**  > **Integrations** をクリックします。
2. **Settings** ページの **Cloud** タブで、**Add integration** をクリックします。
3. **Add a cloud integration** ウィザードの **Select integration type** ステップで **Amazon Web Services** を選択します。 **Next** をクリックします。
4. インテグレーションの名前を入力し、**Next** をクリックします
5. **Select configuration** の手順で、AWS インテグレーションへの接続方法を選択します。
 - **Account authorization** を選択して、AWS アカウントの認証情報を提供し、Red Hat がインテグレーションを設定および管理できるようにします。 **Next** をクリックします。

- インテグレーションをカスタマイズするには、**Manual configuration** を選択します。Cost Management を使用して RHEL サブスクリプションを測定する場合は、**Manual Configuration** を選択する **必要があります**。
6. **Select application** の手順で、**Cost Management** を選択します。**Next** をクリックします。
 7. アカウント認証方法を選択した場合は、**Review details** の手順で詳細を確認し、**Add** をクリックします。手動設定方法を選択した場合は、ウィザードの次のステップに進み、S3 バケットを設定します。

1.2. S3 バケットおよびデータエクスポートの作成

データのエクスポートを保存するための権限が設定された Amazon S3 バケットを作成します。

手順

データのエクスポートを作成するには、AWS アカウントにログインして以下の手順を実行します。

1. AWS S3 コンソールで、新規の S3 バケットを作成するか、既存バケットを使用します。新しい S3 バケットを設定する場合は、デフォルト設定を受け入れます。
2. **Add a cloud source** ウィザードの **Create storage** の手順で、S3 バケットの名前を貼り付け、バケットが作成されたリージョンを選択します。**Next** をクリックします。
3. AWS Billing コンソールで、S3 バケットに配信されるデータエクスポートを作成します。次の値を入力し、他の値についてはデフォルトを受け入れます。
 - **Export type:** レガシー CUR エクスポート
 - **Report name:** koku
 - **Include:** リソース ID
 - **Time unit:** 1時間ごと
 - **Enable report data integration for Amazon Redshift、Amazon QuickSight.** レポートデータ インテグレーションの無効化: Amazon Athena
 - **Compression type:** GZIP
 - **S3 bucket:** <以前に設定した S3 バケット>
 - **Report path prefix:** cost



注記

設定に関する詳細は、AWS **Billing and Cost Management** のドキュメントを参照してください。

4. **Add a cloud integration** ウィザードの **Create cost and usage report** ステップで **Next** をクリックします。

1.3. AWS タグのアクティベーション

タグを使用して Cost Management アプリケーションで AWS リソースを整理するには、AWS でタグをアクティベートしてそれらを自動的にインポートできるようにします。

手順

1. AWS Billing コンソールで、以下を実行します。
 - a. **Cost Allocation Tags** セクションを開きます。
 - b. Cost Management アプリケーションで使用するタグを選択し、**Activate** をクリックします。
2. 組織がシステムを CentOS 7 から RHEL に変換し、時間単位の課金を使用している場合は、AWS コンソールの **Cost Allocation Tags** セクションでシステムの **com_redhat_rhel** タグをアクティブにします。
 - a. AWS で測定する RHEL のインスタンスにタグを付けた後、**Include RHEL usage** を選択します。
3. [Red Hat Hybrid Cloud Console Integrations](#) ウィザードで、**Include RHEL usage** を選択します。

関連情報

タグ付けに関する詳細は、[AWS リソースへのタグの追加](#) を参照してください。

1.4. コストおよび使用状況レポートのアカウントアクセスを有効にするための IAM ポリシーの設定

Cost management には、データを表示するために AWS によって生成されたコストおよび使用状況レポートが必要です。正しいアクセスを提供するには、AWS で IAM ポリシーおよびロールを作成します。これにより、保存された情報へのアクセスのみが提供されます。

Cost Management は、追加のデータを表示することもできます。以下に例を示します。

- Action **iam:ListAccountAliases** を追加して、アカウント番号ではなく AWS アカウントエイリアスを表示します。
- アカウント ID ではなく集約された請求情報を使用している場合は、Actions **organization:List*** と **organizations:Describe*** を使用して、AWS メンバーアカウントの表示名を見つけます。

Cost Management の場合:

1. **Add a cloud integration** ウィザードで、追加するデータポイントを選択します。
2. **Next** をクリックします。
3. 選択内容に基づいて生成された JSON 出力をコピーします。

[AWS Identity and Access Management コンソール](#)で、以下を実行します。

3. AWS Identity and Access Management (IAM) コンソールから、先ほど設定した S3 バケットの新しい IAM ポリシーを作成します。
 - a. JSON タブを選択し、[Red Hat Hybrid Cloud Console](#) の **Add a cloud integration** ウィザードからコピーした JSON ポリシーを貼り付けます。

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

```

"Statement": [
  {
    "Sid": "VisualEditor0",
    "Effect": "Allow",
    "Action": [
      "s3:Get*",
      "s3:List*"
    ],
    "Resource": [
      "arn:aws:s3:::<your_bucket_name>", ❶
      "arn:aws:s3:::<your_bucket_name>/*"
    ]
  },
  {
    "Sid": "VisualEditor1",
    "Effect": "Allow",
    "Action": [
      "s3:ListBucket",
      "cur:DescribeReportDefinitions"
    ],
    "Resource": "*"
  }
]
}

```

- b. ポリシーの名前を入力し、ポリシーを作成します。AWS IAM コンソールを閉じないでください。これは次の手順で使用します。

Cost Management の場合:

4. [Red Hat Hybrid Cloud Console](#) の **Add a cloud integration** ウィザードで **Next** をクリックします。

[AWS Identity and Access Management コンソール](#)で、以下を実行します。

5. AWS IAM コンソールで、新規の IAM ロールを作成します。
 - a. 信頼できるエンティティのタイプとして **別の AWS アカウント** を選択します。
 - b. **589173575009** を Account ID として入力し、[Red Hat Hybrid Cloud Console](#) に AWS アカウントのコストデータへの読み取りのアクセス権を付与します。

Cost Management の場合:

- a. ウィザードの Create IAM role ステップから外部 ID をコピーします。

[AWS Identity and Access Management コンソール](#)で、以下を実行します。

- a. **外部 ID** フィールドに外部 ID を入力します。
- b. 設定したばかりの IAM ポリシーを割り当てます。
- c. ロール名と説明を入力します。

Cost Management の場合:

6. [Red Hat Hybrid Cloud Console](#) の **Add a cloud integration** ウィザードで **Next** をクリックします。

AWS Identity and Access Management コンソールで、以下を実行します。

7. AWS IAM コンソールの **Roles** セクションで、作成したばかりのロールの概要画面を開きます。
 - a. **arn:aws:** で始まる文字列である Role ARN をコピーします。

Cost Management の場合:

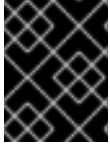
8. [Red Hat Hybrid Cloud Console](#) の **Add a cloud integration** ウィザードで、ロール ARN を貼り付けて、**Next** をクリックします。
9. クラウドインテグレーションの詳細を確認し、**Add** をクリックします。

Cost Management は、AWS アカウントおよびリンクされた AWS アカウントからコストと使用状況データの収集を開始します。

**注記**

[Cost Management](#) ダッシュボードに表示されるまでに、データの入力に数日かかる場合があります。

第2章 AMAZON WEB SERVICES インテグレーションの作成: 高度



重要

基本パスを使用して AWS インテグレーションを作成した場合は、次の手順を実行しないでください。AWS インテグレーションはすでに完了しています。

RHEL メータリングを使用している場合は、データを Cost Management と統合してから、[AWS インテグレーションへの RHEL メータリングの追加](#)に進み、RHEL メータリングのインテグレーションの設定を完了します。

課金データのサブセットを Red Hat と共有するには、AWS で関数スクリプトを設定できます。このスクリプトは、課金データをフィルタリングしてオブジェクトストレージにエクスポートし、Cost Management がフィルタリングされたデータにアクセスして読み取ることができるようにします。[Integrations ページ](#) から AWS インテグレーションを Cost Management に追加します。

AWS はサードパーティーの製品であり、UI およびドキュメントが変更される可能性があります。サードパーティーインテグレーションの設定手順は、公開時点において正確な情報です。最新の情報は、[AWS ドキュメント](#) を参照してください。

前提条件

- [Red Hat Hybrid Cloud Console サービスアカウント](#) がある。
- Cost Management にデータインテグレーションを追加するには、Cloud Administrator 権限を持つ Red Hat アカウントが必要です。

2.1. AWS アカウントのインテグレーションとしての追加

AWS インテグレーションを追加して、cost management が AWS アカウントからコストおよび使用状況レポートを処理できるようにします。AWS アカウントの認証情報を指定して AWS インテグレーションを自動的に追加することも、Cost Management を設定して Red Hat に送信するデータをフィルタリングすることもできます。

前提条件

- Cost Management にデータインテグレーションを追加するには、Cloud Administrator 権限を持つ Red Hat アカウントが必要です。

手順

1. [Red Hat Hybrid Cloud Console](#) から、**Settings Menu**  > **Integrations** をクリックします。
2. **Settings** ページの **Cloud** タブで、**Add integration** をクリックします。
3. **Add a cloud integration** ウィザードの **Select integration type** ステップで **Amazon Web Services** を選択します。**Next** をクリックします。
4. インテグレーションの名前を入力し、**Next** をクリックします
5. **Select configuration** の手順で、AWS インテグレーションへの接続方法を選択します。

- インテグレーションをカスタマイズするには、**Manual configuration** を選択します。Cost Management を使用して RHEL サブスクリプションを測定する場合は、**Manual Configuration** を選択する **必要があります**。**Next** をクリックします。

6. **Select application** の手順で、**Cost Management** を選択します。**Next** をクリックします。

2.2. ATHENA の請求データを保存するための AWS S3 バケットの作成

Athena 請求レポートを保存するようにパーミッションが設定された Amazon S3 バケットを作成します。

手順

1. AWS アカウントにログインします。
2. AWS Billing コンソールで、S3 バケットに配信されるデータエクスポートを作成します。以下の値を指定します (そして他の値のデフォルト値を受け入れます)。
 - **Export type:** レガシー CUR エクスポート
 - **Report name:** <rh_cost_report> (この名前は後で使用するのでメモします)
 - **Additional report details:** リソース ID を含める
 - **S3 bucket:** 以前に設定した S3 バケットを選択するか、バケットを作成してデフォルト設定を受け入れます。
 - **Time granularity:** 1 時間ごと
 - **Enable report data integration for Lambda クエリー** に必要な Amazon Athena
 - **Compression type:** Parquet
 - **Report path prefix:** cost



注記

設定に関する詳細は、AWS Billing and Cost Management のドキュメントを参照してください。

2.3. フィルターされたデータレポートを保存するためのバケットの作成

フィルタリングされたデータを Red Hat と共有するには、データを保存するために 2 番目のバケットを作成する必要があります。

AWS アカウント:

1. AWS アカウントにログインします。
2. **Configure S3 Bucket** から **Configure** をクリックします。バケットを作成し、デフォルトポリシーを適用します。
3. **Save** をクリックします。

Cost Management の場合:

4. **Create storage** ステップで S3 バケットの名前を貼り付け、作成したリージョンを選択し、**Next** をクリックします。
5. **Add a cloud integration** ウィザードの **Create cost and usage report** ステップで、**I wish to manually customize the CUR sent to Cost Management** を選択します。
6. **Next** をクリックします。

2.4. AWS タグのアクティベーション

タグを使用して Cost Management アプリケーションで AWS リソースを整理するには、AWS でタグをアクティベートしてそれらを自動的にインポートできるようにします。

手順

1. AWS Billing コンソールで、以下を実行します。
 - a. **Cost Allocation Tags** セクションを開きます。
 - b. Cost Management アプリケーションで使用するタグを選択し、**Activate** をクリックします。
2. 組織がシステムを CentOS 7 から RHEL に変換し、時間単位の課金を使用している場合は、AWS コンソールの **Cost Allocation Tags** セクションでシステムの **com_redhat_rhel** タグをアクティブにします。
 - a. AWS で測定する RHEL のインスタンスにタグを付けた後、**Include RHEL usage** を選択します。
3. **Red Hat Hybrid Cloud Console Integrations** ウィザードで、**Include RHEL usage** を選択します。

関連情報

タグ付けに関する詳細は、[AWS リソースへのタグの追加](#) を参照してください。

2.5. コストおよび使用状況レポートのアカウントアクセスを有効にするための IAM ポリシーの設定

Cost management には、データを表示するために AWS によって生成されたコストおよび使用状況レポートが必要です。正しいアクセスを提供するには、AWS で IAM ポリシーおよびロールを作成します。これにより、保存された情報へのアクセスのみが提供されます。

Cost Management は、追加のデータを表示することもできます。以下に例を示します。

- Action **iam:ListAccountAliases** を追加して、アカウント番号ではなく AWS アカウントエイリアスを表示します。
- アカウント ID ではなく集約された請求情報を使用している場合は、Actions **organization:List*** と **organizations:Describe*** を使用して、AWS メンバーアカウントの表示名を見つけます。

Cost Management の場合:

1. **Add a cloud integration** ウィザードで、追加するデータポイントを選択します。

2. **Next** をクリックします。
3. 選択内容に基づいて生成された JSON 出力をコピーします。

AWS Identity and Access Management コンソールで、以下を実行します。

3. AWS Identity and Access Management (IAM) コンソールから、先ほど設定した S3 バケットの新しい IAM ポリシーを作成します。
 - a. JSON タブを選択し、[Red Hat Hybrid Cloud Console](#) の **Add a cloud integration** ウィザードからコピーした JSON ポリシーを貼り付けます。

```
{
  "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. ポリシーの名前を入力し、ポリシーを作成します。AWS IAM コンソールを閉じないでください。これは次の手順で使用します。

Cost Management の場合:

4. [Red Hat Hybrid Cloud Console](#) の **Add a cloud integration** ウィザードで **Next** をクリックします。

AWS Identity and Access Management コンソールで、以下を実行します。

5. AWS IAM コンソールで、新規の IAM ロールを作成します。
 - a. 信頼できるエンティティのタイプとして **別の AWS アカウント** を選択します。

- b. 589173575009 を Account ID として入力し、[Red Hat Hybrid Cloud Console](#) に AWS アカウントのコストデータへの読み取りのアクセス権を付与します。

Cost Management の場合:

- a. ウィザードの Create IAM role ステップから外部 ID をコピーします。

[AWS Identity and Access Management コンソール](#)で、以下を実行します。

- a. **外部 ID** フィールドに外部 ID を入力します。
- b. 設定したばかりの IAM ポリシーを割り当てます。
- c. ロール名と説明を入力します。

Cost Management の場合:

6. [Red Hat Hybrid Cloud Console](#) の **Add a cloud integration** ウィザードで **Next** をクリックします。

[AWS Identity and Access Management コンソール](#)で、以下を実行します。

7. AWS IAM コンソールの **Roles** セクションで、作成したばかりのロールの概要画面を開きます。
 - a. **arn:aws:** で始まる文字列である Role ARN をコピーします。

Cost Management の場合:

8. [Red Hat Hybrid Cloud Console](#) の **Add a cloud integration** ウィザードで、ロール ARN を貼り付けて、**Next** をクリックします。
9. クラウドインテグレーションの詳細を確認し、**Add** をクリックします。

次のステップ

レポートをフィルタリングするように Athena と Lambda を設定して、AWS データエクスポートをカスタマイズします。

2.6. ATHENA のアカウントアクセスの有効化

Hybrid Committed Spend が使用する IAM ポリシーおよびロールを作成します。この設定では、保存されている情報へのアクセスのみが提供されます。

手順

1. AWS Identity and Access Management (IAM) コンソールから、設定する Athena Lambda 関数の IAM ポリシーを作成します。
 - a. JSON タブを選択し、JSON ポリシーテキストボックスに以下の内容を貼り付けます。

```
{
  "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 両方の場所の **CHANGE-ME*** を、手順 2.2 で設定した S3 バケットの ARN に置き換えます。

- b. ポリシーに名前を付け、ポリシーの作成を完了します。次の手順に必要なため、AWS IAM コンソールを開いたままにします。
2. AWS IAM コンソールで、新規の IAM ロールを作成します。
 - a. 信頼できるエンティティのタイプには、**AWS service** を選択します。
 - b. Lambda を選択します。
 - c. 設定したばかりの IAM ポリシーを割り当てます。
 - d. ロール名と説明を入力し、ロールの作成を完了します。
 3. AWS Secrets Manager にログイン情報を保存し、作成したロールに追加します。
 - a. **Secret type: Other type of secret** を選択します。
 - b. Red Hat Hybrid Cloud Console **client_id** のキーを作成します。
 - c. Red Hat Hybrid Cloud Console **client_secret** のキーを作成します。
 - d. ユーザー名とパスワードの値を適切なキーに追加します。
 - e. **Continue** をクリックしてから、名前を入力し、シークレットを保存します。
 - f. Lambda 関数用に作成したロールを更新します。以下のコードを追加して、AWS Secrets Manager に保存されているシークレットを参照します。

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

2.6.1. レポート生成用の Athena 設定

Cost Management にフィルタリング処理したデータエクスポートを提供するように Athena を設定します。

次の設定では、保存されている追加の情報へのアクセスのみが提供されます。他のものは提供しません。

手順

1. AWS S3 コンソールで、手順 2.2 で設定した S3 バケットに移動します。次に、設定したデータエクスポートによって作成されるパスにある **crawler-cfn.yml** ファイルに移動します。例: **{bucket-name}/{S3_path_prefix}/{export_name}/crawler-cfn.yml.crawler-cfn.yml** の Object URL をコピーします。

2. AWS コンソールの **Cloudformation** から、新規リソースでスタックを作成します。
 - a. 既存のテンプレートを選択します。
 - b. **Specify Template** を選択します。
 - c. **Template Source: Amazon S3 URL** を選択します。
 - d. 以前にコピーしたオブジェクト URL を貼り付けます。
3. 名前を入力し、**Next** をクリックします。
4. **I acknowledge that AWS Cloudformation might create IAM resources** をクリックして **Submit** をクリックします。

2.6.2. Athena クエリーの構築

Red Hat の経費のデータエクスポートをクエリーし、フィルタリングした経費のレポートを作成する Athena クエリーを作成します。

たとえば、Red Hat の支出をフィルタリングする場合は、サンプルスクリプトに含まれているクエリーだけが必要になることがあります。さらに高度な機能が必要な場合は、カスタムクエリーを作成します。RHEL メータリングを使用している場合は、RHEL サブスクリプションに固有のデータを返すようにクエリーを調整する必要があります。以下の手順では、RHEL サブスクリプションクエリーを作成する方法を説明します。

Red Hat 支出の Athena クエリーの例

```
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'

```

AWS アカウント:

1. エディタータブから Amazon Athena に移動します。
2. **Data source** メニューから、**AwsDataCatalog** を選択します。
3. **Database** メニューからデータのエクスポートを選択します。データエクスポート名の前に **athenacurcfn_** が追加され、その後にデータエクスポート名が追加されます。例: **{your_export_name}**。
4. 以下のサンプルクエリーを **Query** フィールドに貼り付けます。 **your_export_name** の値は、データエクスポート名に置き換えます。

```

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

```

5. **Run** をクリックします。このクエリーの結果は、データセットのタグに関連するすべての列を返します。
6. RHEL タグに使用される列と一致するタグ列をコピーします。
7. 以下のクエリー例に貼り付けます。 **your_export_name** は、前の手順でコピーした tags 列に、 **year** と **month** はクエリーする年と月に置き換えます。結果として、RHEL サブスクリプションに対して適切にタグ付けされた EC2 インスタンスが返されます。このクエリーをコピーして保存し、今後 Lambda 関数で使用します。

```

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. Athena の Lambda 関数の作成

Red Hat 関連の経費のデータエクスポートをクエリーし、フィルタリングされた経費のレポートを作成する Lambda 関数を作成する必要があります。

手順

1. AWS コンソールで Lambda に移動し、 **Create function** をクリックします。
2. **Author from scratch** をクリックします。
3. 関数の名前を入力します。

4. **Runtime** メニューから、利用可能な Python の最新バージョンを選択します。
5. **Architecture** メニューから **x86_64** を選択します。
6. **Permissions** で、作成した Athena ロールを選択します。
7. Lambda 関数の一部としてビルドしたクエリーを追加するには、**Create function** をクリックして、進捗を保存します。
8. 関数の **Code** タブから、この [スクリプト](#) を貼り付けます。以下の行を探します。

your_integration_external_id

Enabling account access for cost and usage consumption手順でコピーしたインテグレーション UUID を入力します。

bucket

フィルタリングしたデータのレポートを保存するためのバケットの作成の手順で、フィルタリングされたレポートを保存するために作成した S3 バケットの名前を入力します。

database

Building your Athena クエリーステップで使用するデータベース名を入力します。

export_name

コストデータを保存するための AWS S3 バケットを作成したときのデータエクスポートの名前を入力します。

9. **where** 句を置き換えて、デフォルトのクエリーをカスタムクエリーに更新します。次に例を示します。

```
# 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}'"
```

10. **Deploy** をクリックして関数をテストします。

2.6.4. レポートファイルを Post するための Lambda 関数の作成

フィルタリングしたレポートを Red Hat がアクセスできるバケットに投稿するには、2 番目の Lambda 関数を作成する必要があります。

手順

1. AWS コンソールの Lambda に移動し、**Create function** をクリックします。
2. **Author from scratch** をクリックします。
3. 関数の名前を入力します。
4. **Runtime** メニューから、利用可能な Python の最新バージョンを選択します。
5. アーキテクチャーとして **x86_64** を選択します。
6. **Permissions** で、作成した Athena ロールを選択します。
7. **Create Function** をクリックします。

- この [スクリプト](#) を関数に貼り付け、次の行を置き換えます。

```
secret_name = "CHANGEME"
```

シークレット名を入力します。

```
bucket = "<your_S3_Bucket_Name>"
```

フィルタリングしたデータのレポートを保存するためのバケットの作成の手順で、フィルタリングされたレポートを保存するために作成した S3 バケットの名前を入力します。

- Deploy** をクリックして関数をテストします。

2.7. イベントブリッジスケジュールの作成

AmazonEventBridge をスケジュールして作成した Lambda 関数をトリガーする必要があります。

手順

- 作成した各関数をトリガーする 2 つの AmazonEventBridge スケジュールを作成します。Athena クエリーがレポートを送信する前に完了するように、これらの関数を異なるリズムでトリガーする必要があります。
 - 名前と説明を追加します。
 - Group** フィールドで、**Default** を選択します。
 - Occurrence** フィールドで、**Recurring schedule** を選択します。
 - Type** フィールドで、**Chron-based** を選択します。
 - cron ベースのスケジュールを 12 時間間隔で設定します。次の例では、午前 9 時と午後 9 時 (**09**?*** と **021**?***) に関数がトリガーされます。
 - 柔軟な時間枠を設定します。
 - Next** をクリックします。
- このスケジュールを Lambda 関数に関連付けるには、**Target detail** を **AWS Lambda invoke** に設定します。
 - 前に作成した Lambda 関数を選択します。
 - Next** をクリックします。
- スケジュールを有効にします。
 - 再試行ロジックを設定します。
 - 暗号化を無視します。
 - パーミッションは、**Create new role on the fly** に設定します。
 - Next** をクリックします。
- 選択内容を確認して、**Create** をクリックします。

2.8. 最終的なデータを収集するクラウド機能の追加作成

AWS は、前月の最終レポートを翌月の初めに送信します。これらの最終レポートを Cost management に送信し、追加情報を分析します。

手順

1. Lambda 関数の Athena クエリーを作成します。
 - a. Athena を照会するための関数を作成します。
 - b. **Author from scratch** を選択します。
 - c. **Python** ランタイムを選択します。
 - d. **x86_64** アーキテクチャーを選択します。
 - e. 権限には、以前に作成したロールを選択します。
 - f. **Create** をクリックします。
2. **Code** タブをクリックして、最終データを収集するスクリプトを追加します。
 - a. **Athena query function** をコピーし、クエリーに追加します。<integration_uuid> を、console.redhat.com で作成したインテグレーションの **integration_uuid** に更新します。これは、[Integrations ページ](#) に移動してインテグレーションをクリックすると見つかります。作成したバケット名およびデータベースで **BUCKET** および **DATABASE** 変数を更新します。次に、**export_name** は、前に作成したデータエクスポート Athena クエリーの名前に更新します。
 - b. 次のコードからコメントを削除します。


```
# 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. **Deploy** をクリックします。次に、**Test** をクリックして実行結果を表示します。
3. レポートファイルを cost management に投稿する Lambda 関数を作成します。
 - a. **Author from scratch** を選択します。
 - b. 関数に名前を付けます。
 - c. **Python** ランタイムを選択します。
 - d. **x86_64** アーキテクチャーを選択します。
 - e. 権限には、以前に作成したロールを選択します。
 - f. **Create** をクリックします。
4. **Code** タブをクリックして、確定したデータを投稿するスクリプトを追加します。
 - a. **post function** をコピーし、クエリーに追加します。**secret_name** は、AWS Secrets Manager のシークレットの名前に更新します。**bucket** は、作成したバケット名に更新します。

b. 次のコードからコメントを削除します。

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

c. **Deploy** をクリックします。次に、**Test** をクリックして実行結果を表示します。

5. 2つの関数をトリガーする EventBridge スケジュールを作成します。詳細は、「[イベントブリッジスケジュールの作成](#)」を参照してください。

a. 以前の期間の AWS 請求書はその日までに確定するため、EventBridge スケジュールを毎月 15 日以降に 1 回実行するように設定します。例: **(0 9 15 * ? *)** および **(0 21 15 * ? *)**

この手順を完了したら、cost management は、AWS アカウントおよびリンクされた AWS アカウントからコストおよび使用状況データの収集を開始します。



注記

[Cost Management](#) ダッシュボードに表示されるまでに、データの入力に数日かかる場合があります。

第3章 コストを管理するための次のステップ

OpenShift Container Platform と Amazon Web Services のデータを追加すると、Cost Management のコストには、インテグレーションごとのコストデータと、プラットフォーム上での OpenShift Container Platform クラスターの実行に関連するコストと使用量が表示されます。OpenShift ノードを実行している EC2 インスタンスに AWS 割引プランを使用している場合、Cost Management ではデフォルトで割引プランのコストが使用されます。

[Cost Management Overview](#) ページでは、コストデータが **OpenShift** タブと **Infrastructure** タブに分類されます。コストデータのさまざまなビューを切り替えるには、**Perspective** を選択します。

グローバルナビゲーションメニューを使用して、クラウドプロバイダーごとのコストに関する追加の詳細を表示することもできます。

関連情報

- [Cost Management への OpenShift Container Platform データの統合](#)
- [Google Cloud データの Cost Management への統合](#)
- [Microsoft Azure データの Cost Management への統合](#)
- [Oracle Cloud データの Cost Management への統合](#)

3.1. COST MANAGEMENT リソースへのアクセス制限

Cost Management でインテグレーションを追加して設定した後、コストデータとリソースへのアクセスを制限できます。

ユーザーがすべてのコストデータにアクセスできる状況は避ける必要がある場合もあります。代わりに、プロジェクトまたは組織に固有のデータにだけアクセスできるようにユーザーにアクセス権を付与できます。ロールベースのアクセス制御を使用すると、Cost Management レポートでのリソースの表示を制限できます。たとえば、ユーザーのビューを環境全体ではなく、AWS インテグレーションのみに制限できます。

アクセスを制限する方法の詳細は、[Cost Management リソースへのアクセス制限](#) を参照してください。

3.2. インテグレーションのタグ付けの設定

Cost Management アプリケーションは、タグを使用してクラウドとインフラストラクチャーのコストを追跡します。タグは、OpenShift ではラベルとも呼ばれます。

Cost Management でタグを調整して、リソースをフィルタリングおよび属性化し、コスト別にリソースを整理し、クラウドインフラストラクチャーのさまざまな部分にコストを割り当てることができます。



重要

タグとラベルは、インテグレーションでのみ直接設定できます。Cost Management でアクティブ化するタグの選択はできますが、Cost Management アプリケーションでタグとラベルの編集はできません。

以下のトピックに関する詳細は、[タグ付けを使用したコストデータの管理](#) を参照してください。

- コストデータの表示を整理するためのタグ付けストラテジーを計画する
- Cost Management がタグを関連付ける方法を理解する
- インテグレーションでタグとラベルを設定する

3.3. AWS 請求プランの設定

AWS 請求の詳細は、AWS ドキュメントの [Understanding Consolidated Bills](#) を参照してください。

Cost Management では、AWS の請求プランに対応するために 3 つのコスト計算オプションがサポートされています。

Unblended

その日の費用は、使用コストに応じて計算されます。

Amortized (デフォルト)

定期費用と初期費用は、請求期間全体で均等に分散されます。

Blended

費用は AWS のブレンドレートに従って計算されます。

この手順では、コスト計算をデフォルトの **Unblended** から **Amortized** または **Blended** に設定する方法を説明します。

前提条件

- 組織管理者として [Red Hat Hybrid Cloud Console](#) へアクセスできる。

手順

1. [Red Hat Hybrid Cloud Console](#) から、[Cost Management settings](#) ページに移動します。
2. **Show cost as** で **Amortized** または **Blended** を選択します。
3. **Save** をクリックします。

3.4. コストを正確にレポートするためのコストモデルの設定

Cost Management でコストと使用量のデータを収集するようにインテグレーションを設定したので、価格をメトリクスと使用量に関連付けるコストモデルを設定できます。

コストモデルは、Cost Management において、原価とメトリクスを使用してコスト計算を定義するためのフレームワークです。コストモデルが生成するコストの記録と分類、および特定の顧客、ビジネスユニット、またはプロジェクトに対する配分を行えます。

[Cost Models](#) では、次のタスクを完了できます。

- コストを、インフラストラクチャーコストまたは補足コストとして分類する
- OpenShift ノードおよびクラスターの月額コストを取得する
- 追加のサポートコストを考慮して利潤を適用する

コストモデルの設定方法は [コストモデルの使用](#) を参照してください。

3.5. COST EXPLORER を使用したコストの可視化

Cost Management の [Cost Explorer](#) を使用して、時間スケールのコストと使用状況情報のカスタムグラフを作成し、最終的にコストをより適切に可視化して解釈できるようにします。

次のトピックに関する詳細は、[Cost Explorer を使用したコストの可視化](#) を参照してください。

- Cost Explorer を使用して異常なイベントを特定する。
- 時間の経過とともにコストデータがどのように変化するかを理解する。
- コストおよび使用状況データのカスタムバーチャートを作成する。
- カスタムコストデータテーブルをエクスポートする。

第4章 インテグレーションの更新

Cost Management にインテグレーションを追加済みで、そのインテグレーションを変更する場合は、[Red Hat Hybrid Cloud Console](#) から、インテグレーションに関連付けられているアプリケーションを追加または削除できます。

手順

1. [Red Hat Hybrid Cloud Console](#) から **Settings**  をクリックします。
2. **Integrations** をクリックします。
3. 統合のためにその他のオプションメニュー  をクリックします。 **Edit** をクリックします。
4. **Metered Product** で、ドロップダウンから **Red Hat Enterprise Linux** を選択してメータリングを有効にします。

4.1. AWS 統合への RHEL メータリングの追加

互換性のあるサードパーティーの Linux ディストリビューションから Red Hat Enterprise Linux (RHEL) に変換し、Amazon Web Services (AWS) でサードパーティー移行リストの RHEL を購入した場合は、AWS 統合に RHEL メータリングを追加できます。

RHEL メータリングでは、Red Hat が請求書进行处理して、AWS での Red Hat サービスに関連する RHEL の使用量を時間単位で計測します。

手順

1. AWS で、測定する RHEL のインスタンスにタグを付けます。AWS での RHEL インスタンスのタグ付けに関する詳細は、[AWS リソースへのタグの追加](#) を参照してください。
2. [Red Hat Hybrid Cloud Console](#) から **Settings**  をクリックします。
3. **Integrations** をクリックします。
4. インテグレーションのその他のオプションメニュー  をクリックします。 **Edit** をクリックします。
5. **Metered Product** で、ドロップダウンから **Red Hat Enterprise Linux** を選択してメータリングを有効にします。

RED HAT ドキュメントへのフィードバック (英語のみ)

Red Hat ドキュメントに関するフィードバックをお寄せください。いただいたご要望に迅速に対応できるよう、できるだけ詳細にご記入ください。

前提条件

- Red Hat カスタマーポータルにログインしている。

手順

フィードバックを送信するには、以下の手順を実施します。

1. [Create Issue](#) にアクセスします。
2. **Summary** テキストボックスに、問題または機能拡張に関する説明を入力します。
3. **Description** テキストボックスに、問題または機能拡張のご要望に関する詳細を入力します。
4. **Reporter** テキストボックスに、お客様のお名前を入力します。
5. **Create** ボタンをクリックします。

これによりドキュメントに関するチケットが作成され、適切なドキュメントチームに転送されます。フィードバックをご提供いただきありがとうございました。