



Cost Management Service 1-latest

Microsoft Azure データの Cost Management への統合

Microsoft Azure インテグレーションと RHEL メータリングを追加する方法を学ぶ

Cost Management Service 1-latest Microsoft Azure データの Cost Management への統合

Microsoft Azure インテグレーションと 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

Microsoft Azure インテグレーションを Cost Management に追加する方法を説明します。

目次

パート I. 基本または高度な AZURE インテグレーションの選択	3
第1章 MICROSOFT AZURE インテグレーションの作成: 基本	4
1.1. MICROSOFT AZURE アカウントの追加	4
1.2. MICROSOFT AZURE リソースグループおよびストレージアカウントの作成	5
1.3. 日々の MICROSOFT AZURE データエクスポートスケジュールの設定	6
1.4. MICROSOFT AZURE SUBSCRIPTION ID の検索	6
1.5. RED HAT アクセス用の MICROSOFT AZURE ロールの作成	7
1.6. データの表示	8
第2章 MICROSOFT AZURE インテグレーションの作成: 高度	9
2.1. MICROSOFT AZURE アカウントの追加	9
2.2. MICROSOFT AZURE リソースグループおよびストレージアカウントの作成	10
2.3. MICROSOFT AZURE での日次エクスポートの作成	10
2.4. MICROSOFT AZURE SUBSCRIPTION ID の検索	11
2.5. RED HAT アクセス用の MICROSOFT AZURE ロールの作成	11
2.6. MICROSOFT AZURE での関数の作成	12
2.7. AZURE での認証情報の設定	15
2.8. 関数への VAULT 認証情報の追加	16
2.9. MICROSOFT AZURE での関数ロールの設定	16
2.10. データの表示	17
第3章 データの設定と表示の次のステップ	18
3.1. COST MANAGEMENT リソースへのアクセス制限	18
3.2. インテグレーションのタグ付けの設定	18
3.3. 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	19
3.4. COST EXPLORER を使用したコストの可視化	19
第4章 インテグレーションの更新	20
4.1. MICROSOFT AZURE インテグレーションへの RHEL メータリングの追加	20
RED HAT ドキュメントへのフィードバック (英語のみ)	21

パート I. 基本または高度な AZURE インテグレーションの選択

Azure 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

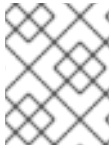
基本オプションについては、[Microsoft Azure インテグレーションの作成: 基本](#) を参照してください。

基本パスを使用すると、Cost Management は指定の範囲で Azure から課金レポートを直接読み取ることができます。

高度

高度オプションの場合は、[Microsoft Azure インテグレーションの作成: 高度](#) に進んでください。

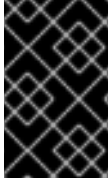
高度なパスを使用すると、Cost Management がデータを読み取る **前** に、データをカスタマイズしたりフィルタリングしたりできます。特定の Red Hat 製品とのみ課金データを共有する場合は、高度なパスを使用することも可能です。高度なパスでは、セットアップと設定がより複雑になります。



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 MICROSOFT AZURE 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 Azure 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、Microsoft Azure インテグレーションの作成: 高度に進んでください。

RHEL メータリングを使用している場合は、データを Cost Management と統合した後に、[Microsoft Azure 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 メータリングの統合の設定を完了します。

[Integration ページ](#) から Cost Management 用に Microsoft Azure 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 Microsoft Azure アカウントを設定する必要があります。

Azure インテグレーションを作成するには、次のタスクを実行します。

1. ストレージアカウントおよびリソースグループを作成します。
2. コストエクスポートの適切な範囲を選択します。
3. アクセス用の Storage Account Contributor および Reader ロールを設定します。
4. 毎日のコストエクスポートをスケジュールします。

Azure はサードパーティー製品であるため、UI とドキュメントは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。最新情報は、[Microsoft Azure のドキュメント](#) を参照してください。

1.1. MICROSOFT AZURE アカウントの追加

Cost Management でコストと使用状況データを処理できるように、Microsoft Azure アカウントをインテグレーションとして追加します。

前提条件

Cloud Administrator 権限を持つ Red Hat ユーザーアカウントがある。

Cost Management の場合:

1. Settings Menu  > Integrations をクリックします。
2. Cloud タブで、**Add integration** をクリックします。
3. **Add a cloud integration** ウィザードで **Microsoft Azure** を選択し、**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 application** の手順で、**Cost Management** を選択し、**Next** をクリックします。
6. **Specify cost export scope** ステップで、**I am OK with sending the default data to Cost Management** を選択します。
 - RHEL 使用量の課金を登録する場合は、**Include RHEL usage** を選択します。それ以外の場合は、次の手順に進みます。

7. メニューからコストデータエクスポートのスコープを選択します。サブスクリプションレベルまたはその他のスコープでデータをエクスポートできます。
8. 生成されたコマンドをコピーします。

Microsoft Azure アカウント:

9. **Cloud Shell** をクリックし、Cost Management からコピーしたコマンドを実行します。返された値をコピーします。

Cost Management の場合:

10. **Specify cost export scope** ステップで、Microsoft Azure からコピーした値を **Cost export scope** に貼り付けます。
11. **Next** をクリックします。

次のセクションでもウィザードを引き続き使用します。

1.2. MICROSOFT AZURE リソースグループおよびストレージアカウントの作成

コストデータとメトリクスを格納するために、Microsoft Azure にストレージアカウントを作成します。Cost Management の **Add a cloud integration** ウィザードで、対応するフィールドにストレージアカウント名を入力します。

前提条件

Cloud Administrator 権限を持つ Red Hat ユーザーアカウントがある。

Microsoft Azure アカウント:

1. **storage** を検索し、**Storage accounts** をクリックします。
2. **Storage accounts** ページで **Create** をクリックします。
3. **Create a storage account** ページの **Resource Group** フィールドで **Create new** をクリックします。名前を入力し、**OK** をクリックします。この例では、**cost-data-group** を使用します。
4. **Instance details** で、**Storage account name** フィールドに名前を入力します。この例では、**costdata** を使用します。
5. 後で [Red Hat Hybrid Cloud Console](#) に追加できるように、リソースグループとストレージアカウントの名前をコピーします。
6. **Review** をクリックします。
7. ストレージアカウントを確認し、**Create** をクリックします。

Cost Management の場合:

8. **Add a cloud integration** ウィザードで、**Resource group name** および **Storage account name** にコピーしたリソースグループ名およびストレージアカウント名を貼り付けます。
9. **Next** をクリックします。

次のセクションでもウィザードを引き続き使用します。

1.3. 日々の MICROSOFT AZURE データエクスポートスケジュールの設定

次に、Cost Management がデータを毎日取得できるように、コストデータを Microsoft Azure ストレージアカウントに自動的にエクスポートするように設定します。

Microsoft Azure アカウント:

1. 検索バーに "cost exports" と入力し、結果をクリックします。
2. **Create** をクリックします。
3. **Select a template** で、**Cost and usage (actual)** をクリックし、標準使用量および購入料金をエクスポートします。
4. Azure ウィザードの手順に従います。
 - 直前のセクションの手順で作成した、正しいサブスクリプションと **Storage account** を選択します。
 - **Format** を **CSV** に設定する必要があります。
 - **Compression type** を **None** または **Gzip** に設定します。
5. 情報を確認し、**Create** をクリックします。

Cost Management の場合:

6. **Add a cloud integration** ウィザードに戻り、**Daily export** の手順を完了します。
7. **Next** をクリックします。

次のセクションでもウィザードを引き続き使用します。Azure でエクスポートを作成する方法の詳細は、[Microsoft のドキュメント](#) を参照してください。

1.4. MICROSOFT AZURE SUBSCRIPTION ID の検索

Microsoft Azure Cloud Shell を使用して **subscription_id** を検索し、それを Cost Management の **Add a cloud integration** ウィザードに追加します。

Microsoft Azure アカウント:

1. **Cloud Shell** をクリックします。
2. 以下のコマンドを入力してサブスクリプション ID を取得します。

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

3. **subscription_id** に対して生成された値をコピーします。

応答の例

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

Cost Management の場合:

4. **Add a cloud integration** ウィザードの **Subscription ID** フィールドに、前の手順でコピーした値を貼り付けます。
5. **Next** をクリックします。

次のセクションでもウィザードを引き続き使用します。

1.5. RED HAT アクセス用の MICROSOFT AZURE ロールの作成

Red Hat にデータへのアクセスを許可するには、Microsoft Azure で専用のロールを設定する必要があります。同じ Azure サブスクリプションに追加のリソースがある場合は、新しいサービスアカウントを作成する必要がない場合があります。

Cost Management の場合:

1. **Add a cloud integration** ウィザードの **Roles** セクションで、**az ad sp create-for-rbac** コマンドをコピーして、Cost Management の Storage Account Contributor ロールを持つサービスプリンシパルを作成します。

Microsoft Azure アカウント:

2. **Cloud Shell** をクリックします。
3. クラウドシェルプロンプトで、コピーしたコマンドを貼り付けます。
4. 返されたデータからクライアント ID、シークレット、テナントの値をコピーします。

応答の例

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

Cost Management の場合:

5. **Add a cloud integration** ウィザードに戻り、コピーした値を **Roles** ページの対応するフィールドに貼り付けます。
6. ウィザードから生成された 2 番目の **az role assignment** コマンドをコピーします。

Microsoft Azure アカウント:

7. Cloud Shell プロンプトに戻り、コマンドを貼り付けて **Cost management reader** ロールを作成します。

Cost Management の場合:

8. **Add a cloud integration** ウィザードに戻り、**Next** をクリックします。
9. 指定した情報を確認し、**Add** をクリックします。

1.6. データの表示

これで、基本的なインテグレーションが正常に作成されました。データを使用した操作のオプションについては、[コストを管理するための次のステップ](#)に進んでください。Microsoft Azure インテグレーションの作成: Advanced の手順は **使用しない** してください。

第2章 MICROSOFT AZURE 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 Azure 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。Azure 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 と統合した後に、[Microsoft Azure 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 メータリングの統合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、Microsoft Azure 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、Microsoft Azure インテグレーションを Cost Management に追加します。

Azure はサードパーティ製品であり、そのプロセスは変更される可能性があります。サードパーティ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。最新情報については、[Microsoft Azure のドキュメント](#)を参照してください。

Azure インテグレーションを作成するには、次のタスクを実行します。

1. ストレージアカウントおよびリソースグループを作成します。
2. アクセス用の Storage Account Contributor および閲覧者ロールを設定します。
3. Red Hat に送信するデータをフィルターする関数を作成します。


2.1. MICROSOFT AZURE アカウントの追加

Cost Management でコストと使用状況データを処理できるように、Microsoft Azure アカウントをインテグレーションとして追加します。

前提条件

- Cloud Administrator 権限を持つ Red Hat ユーザーアカウントがある。
- [サービスアカウントキー](#)がある。
- Cost Management アクセスを有効にするために Hybrid Cloud Console でサービスアカウントに適切なロールが割り当てられている。詳細は、[ユーザーアクセス設定ガイド](#)を参照してください。

Cost Management の場合:

1. Settings Menu  > Integrations をクリックします。
2. Cloud タブで、**Add integration** をクリックします。
3. Add a cloud integration ウィザードで **Microsoft Azure** を選択し、**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 application** の手順で、**Cost Management** を選択し、**Next** をクリックします。
6. **Specify cost export scope** ステップで、**I wish to manually customize the data set sent to Cost Management** を選択し。
 - RHEL 使用量の課金を登録する場合は、**Include RHEL usage** を選択します。それ以外の場合は、次の手順に進みます。
7. **Next** をクリックします。

2.2. MICROSOFT AZURE リソースグループおよびストレージアカウントの作成

課金エクスポートを格納するためのストレージアカウントと、フィルター処理されたデータを格納するための2番目のストレージアカウントを Microsoft Azure に作成します。

Microsoft Azure アカウント:

1. 検索バーに "storage" と入力し、**Storage accounts** をクリックします。
2. **Storage accounts** ページで **Create** をクリックします。
3. **Resource Group** フィールドで、**Create new** をクリックします。名前を入力し、**OK** をクリックします。この例では、**filtered-data-group** を使用します。
4. **Instance details** セクションで、**Storage account name** フィールドに名前を入力します。たとえば、**filterreddata** を使用します。
5. リソースグループとストレージアカウントの名前をコピーして、後で [Red Hat Hybrid Cloud Console](#) に追加できるようにします。
6. **Review** をクリックします。
7. ストレージアカウントを確認し、**Create** をクリックします。

Cost Management の場合:

8. **Add a cloud integration** ウィザードで、**Resource group name** および **Storage account name** にコピーしたリソースグループ名およびストレージアカウント名を貼り付けます。

次のセクションでもウィザードを引き続き使用します。

2.3. MICROSOFT AZURE での日次エクスポートの作成

次に、Cost management 用にフィルター処理する前に、コストデータを Microsoft Azure ストレージアカウントに自動的にエクスポートするように設定します。

Microsoft Azure アカウント:

1. 検索バーに "cost exports" と入力し、結果をクリックします。
2. **Create** をクリックします。
3. **Select a template** で、**Cost and usage (actual)** をクリックし、標準使用量および購入料金をエクスポートします。

4. Azure ウィザードの手順に従います。
 - 新しいリソースグループとストレージアカウントを作成するか、既存のものを選択できます。この例では、ストレージアカウントに **billingexportdata** を使用し、リソースグループに **billinggroup** を使用します。
 - **Format** を **CSV** に設定する必要があります。
 - **Compression type** を **None** または **Gzip** に設定します。
5. 情報を確認し、**Create** をクリックします。

Cost Management の場合:

6. **Add a cloud integration** ウィザードに戻り、**Daily export** の手順を完了します。
7. **Next** をクリックします。

次のセクションでもウィザードを引き続き使用します。

2.4. MICROSOFT AZURE SUBSCRIPTION ID の検索

Microsoft Azure Cloud Shell を使用して **subscription_id** を検索し、それを Cost Management の **Add a cloud integration** ウィザードに追加します。

Microsoft Azure アカウント:

1. **Cloud Shell** をクリックします。
2. 以下のコマンドを入力してサブスクリプション ID を取得します。

```
az account show --query "{subscription_id: id}"
```
3. **subscription_id** に対して生成された値をコピーします。

応答の例

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

Cost Management の場合:

4. **Add a cloud integration** ウィザードの **Subscription ID** フィールドに、前の手順でコピーした値を貼り付けます。
5. **Next** をクリックします。

次のセクションでもウィザードを引き続き使用します。

2.5. RED HAT アクセス用の MICROSOFT AZURE ロールの作成

Red Hat にデータへのアクセスを許可するには、Microsoft Azure で専用のロールを設定する必要があります。同じ Azure サブスクリプションに追加のリソースがある場合は、新しいサービスアカウントを作成する必要がない場合があります。

Cost Management の場合:

1. **Add a cloud integration** ウィザードの **Roles** セクションで、**az ad sp create-for-rbac** コマンドをコピーして、Cost Management の Storage Account Contributor ロールを持つサービスプリンシパルを作成します。

Microsoft Azure アカウント:

2. **Cloud Shell** をクリックします。
3. クラウドシェルプロンプトで、コピーしたコマンドを貼り付けます。
4. クライアント ID、シークレット、テナントに対して生成された値をコピーします。

応答の例

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

Cost Management の場合:

5. **Add a cloud integration** ウィザードに戻り、コピーした値を **Roles** ページの対応するフィールドに貼り付けます。
6. **Next** をクリックします。
7. 情報を確認し、**Add** をクリックしてインテグレーションを完了します。
8. 表示されるポップアップ画面で、関数スクリプトの **Source UUID** をコピーします。

2.6. MICROSOFT AZURE での関数の作成

Azure で関数を作成すると、データがフィルター処理され、Red Hat との共有用に作成したストレージアカウントに追加されます。このセクションのサンプル Python スクリプトを使用して、エクスポートからフィルタリングされたコストデータを収集して共有できます。

前提条件

- デバイスに Visual Studio Code がインストールされている必要がある。
- Visual Studio Code に Microsoft Azure Functions エクステンションがインストールされている必要がある。Azure 関数を作成する場合、Microsoft は、Microsoft Visual Studio Code IDE を使用してコードを開発およびデプロイすることを推奨しています。Visual Studio Code の設定に関する詳細は、[Quickstart: Create a function in Azure with Python using Visual Studio Code](#) を参照してください。

Microsoft Azure アカウント:

1. 検索バーに **functions** と入力し、**Function App** を選択します。
2. **Create** をクリックします。
3. 関数のホスティングオプションを選択し、**Select** をクリックします。
4. **Create Function App** ページで、リソースグループを追加します。
 - a. **Instance Details** セクションで、関数アプリケーションに名前を付けます。
 - b. **Runtime stack** で **Python** を選択します。
 - c. **Version** で latest を選択します。
5. **Review + Create** をクリックします。
 - a. **Create** をクリックします。
 - b. リソースが作成されるまで待ってから、**Go to resource** をクリックして表示します。

Visual Studio Code の場合:

6. **Microsoft Azure** タブをクリックし、Azure にサインインします。
 - a. **Workspaces** ドロップダウンで、オレンジ色の稲妻のアイコンとして表示される **Azure Functions** をクリックします。
 - b. **Create Function** をクリックします。
7. 指示に従ってローカルの場所を設定し、関数の言語とバージョンを選択します。この例では、**Python**、**Model 2**、および利用可能な最新バージョンを選択します。
8. 関数の **Select a template** ダイアログで、**Timer trigger** を選択し、関数に名前を付けて、Enter キーを押します。
9. 関数の実行のタイミングを制御する cron 式を設定します。この例では、**09 ***** を使用して、関数を毎日 9 AM で実行します。
 - a. **Create** をクリックします。
 - b. **Open in the current window** をクリックします。

requirements.txt ファイルの場合:

10. 開発環境で関数を作成したら、requirements.txt ファイルを開き、以下の要件を追加してファイルを保存します。

```
azure-functions
pandas
requests
azure-identity
azure-storage-blob
```

init.py の場合:

11. **Python スクリプト** をコピーし、`init.py` に貼り付けます。

12. **# Required vars to update** とマークされたセクションの値を変更して、環境に対応する値に更新します。

- サンプルスクリプトでは、Azure Key Vault のシークレットを使用して、サービスアカウントの **client_id** と **client_secret** を環境変数として設定します。認証情報をスクリプトに直接入力することもできますが、これはベストプラクティスではありません。
- デフォルトのスクリプトには、データのフィルタリングや RHEL サブスクリプションのフィルタリングを行うためのオプションが組み込まれています。使用するフィルタリングのタイプのコメントを解除するか、独自のカスタムフィルタリングを記述する必要があります。次のどちらか一方からコメントを削除してください。

- **filtered_data = hcs_filtering(df)**

- **filtered_data = rhel_filtering(df)**

- よりカスタマイズされたフィルタリングを記述する場合は、次の必須列を含める必要があります。

```
'additionalinfo', 'billingaccountid', 'billingaccountname', 'billingcurrencycode',
'billingperiodenddate', 'billingperiodstartdate', 'chargetype', 'consumedservice',
'costinbillingcurrency', 'date', 'effectiveprice', 'metercategory', 'meterid', 'metername',
'meterregion', 'metersubcategory', 'offerid', 'productname', 'publishername',
'publishertype', 'quantity', 'reservationid', 'reservationname', 'resourcegroup', 'resourceid',
'resourcelocation', 'resourcename', 'servicefamily', 'serviceinfo1', 'serviceinfo2',
'subscriptionid', 'tags', 'unitofmeasure', 'unitprice'
```

- 一部の列はレポートの種類によって異なります。サンプルスクリプトはこれらの列を正規化し、フィルター処理されたすべてのレポートはこの例に従う必要があります。

```
column_translation = {"billingcurrency": "billingcurrencycode", "currency":
"billingcurrencycode", "instanceid": "resourceid", "instancename": "resourceid",
"pretaxcost": "costinbillingcurrency", "product": "productname", "resourcegroupname":
"resourcegroup", "subscriptionguid": "subscriptionid", "servicename": "metercategory",
"usage_quantity": "quantity"}
```

- データをフィルタリングするには、データフレームのフィルタリングを追加する必要があります。以下に例を示します。
 - 完全一致: **df.loc[(df["publishertype"] == "Marketplace")] publisherType** は、Marketplace 以外のすべてのデータを除外します。
 - Contains: **df.loc[df["publishername"].astype(str).str.contains("Red Hat")]** は、**publisherName** に Red Hat が含まれていないすべてのデータをフィルタリングします。
 - **df.loc** 句で **&** (AND の場合) と **|** (OR の場合) を使用してフィルターをスタックできます。
 - さらに便利なフィルター:

subscriptionid

特定のサブスクリプションをフィルタリングします。

resourcegroup

特定のリソースグループをフィルタリングします。

resourcelocation

特定のリージョンでデータをフィルタリングします。

- 特定のサービスタイプをフィルタリングするには、**servicename**、**servicetier**、**metercategory**、および **metresubcategory** を使用できます。
13. カスタムクエリーを作成したら、**# custom filtering basic example #** の下にあるサンプルスクリプトでカスタムクエリーを更新します。
 14. ファイルを保存します。

Visual Studio Code の場合:

14. **Function** ウィンドウを右クリックし、**Deploy to Function App** をクリックします。
15. 前の手順で作成した関数アプリケーションを選択します。

2.7. AZURE での認証情報の設定

Azure の手順に関するヘルプは、Microsoft のドキュメント [Azure Key Vault](#) を参照してください。

Microsoft Azure アカウント:

1. **Key Vaults** に移動します。
2. **Create** をクリックします。
3. 関数が含まれるリソースグループを選択し、Azure ウィザードに従って新しいシークレットを作成します。
 - a. **Key vault name** には、任意の名前を入力できます。
 - b. **Access policies** で、**Create new policy** をクリックします。次に、テンプレートから **Secret Management** を選択します。
 - c. **Principal** タブで、関数をプリンシパルとして検索して選択します。
4. ウィザードを完了し、**Create Vault** をクリックした後に、Azure で正常にデプロイメントページが表示されるまで待ちます。次に、**Go to resource** をクリックして **Key Vault** ページを開きます。
5. **Objects** ドロップダウンメニューで、**Secrets** を選択します。
6. サービスアカウントには、**client_id** および **client_secret** の 2 つの**新規シークレット** を作成する必要があります。両方を作成するには、次の手順を 2 回実行します。
 - a. シークレットを作成するには、**Generate/import** をクリックします。
 - b. **Create a secret** に、**client_id** または **client_secret** に、任意の名前を入力します。
 - c. **シークレット識別子** の値をコピーします。後で使用します。この値は後で取得することもできます。
 - d. **client_id** と **client_secret** の両方のシークレットが得られるまで、前の 3 つの手順を繰り返します。

2.8. 関数への VAULT 認証情報の追加

次に、Microsoft Azure の関数に移動し、シークレットに関する情報を入力します。

Microsoft Azure アカウント:

1. 関数に移動します。
2. **Settings** → **Environment variables** を選択します。
3. **Add** をクリックします。次の規則に従い、**YOUR-CLIENT-ID-URI** を、前にコピーした **Secret Identifier** の値に置き換えます。
 - 名前: **ClientIdFromVault**
 - 値: **@Microsoft.KeyVault(SecretUri=YOUR-CLIENT-ID-URI)**
4. **Save** をクリックします。
5. **ClientSecretFromVault** に対してプロセスを繰り返します。以下の規則を使用して、**YOUR-CLIENT-SECRET-URI** を前にコピーした **Secret Identifier** の値に置き換えます。
 - 値: **@Microsoft.KeyVault(SecretUri=YOUR-CLIENT-SECRET-URI)**

2.9. MICROSOFT AZURE での関数ロールの設定

専用の認証情報を設定して、関数 Blob に Microsoft Azure コストデータへのアクセスを許可します。これらの認証情報により、関数は元のストレージコンテナからフィルター処理されたストレージコンテナにデータにアクセスし、フィルター処理して転送できるようになります。

Microsoft Azure アカウント:

1. **functions** を検索バーに入力し、関数を選択します。
2. **Settings** メニューで、**Identity** をクリックします。

次の手順を、Microsoft Azure リソースグループとストレージアカウントの作成のセクションで作成した2つのストレージアカウントごとに1回ずつ、合計2回実行します。

3. **Azure role assignments** をクリックします。
4. **Add role assignment** をクリックします。
5. **Scope** フィールドで、**Storage** を選択します。
6. **Resource** フィールドで、2つのストレージアカウントのいずれかを選択します。この例では、**filtereddata** および **billingeportdata** を使用します。
7. **Role** で、**Storage Blob Data Contributor** を選択します。
8. **Save** をクリックします。
9. 再度 **Add role assignment** をクリックします。
10. **Scope** フィールドで、**Storage** を選択します。

11. **Resource** フィールドで、同じストレージアカウントを再度 選択します。
12. 今回は、**Role** で **Storage Queue Data Contributor** を選択します。
13. **Save** をクリックします。
14. 作成した他のストレージアカウントに対しても、このプロセスをすべて繰り返します。

これらの手順を完了すると、Azure インテグレーションが正常にセットアップされます。

2.10. データの表示

これで、高度なインテグレーションが正常に作成されました。データを使用した操作のオプションについては、[コストを管理するための次のステップ](#)に進んでください。

第3章 データの設定と表示の次のステップ

OpenShift Container Platform と Microsoft Azure インテグレーションを追加すると、Cost Management には、Azure での OpenShift Container Platform クラスターの実行に関連するコストと使用状況データが表示されます。データが表示されるまで最大 24 時間かかる場合があります。

[cost management Overview](#) ページでは、コストデータが **OpenShift** および **Infrastructure** タブに表示されます。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](#)
- [Amazon Web Services \(AWS\) データの Cost Management への統合](#)
- [Google Cloud 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. 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。

3.4. 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. MICROSOFT AZURE インテグレーションへの 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、Microsoft Azure でサードパーティー移行リストの RHEL を購入した場合は、作成した Microsoft Azure インテグレーションを更新して 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、Microsoft Azure の Red Hat サービスに関連付けられた RHEL の1時間あたりの使用量を測定します。

手順

1. Microsoft Azure で、計測する RHEL のインスタンスにタグを付けます。Microsoft Azure で RHEL インスタンスにタグを付ける方法の詳細は、[Microsoft Azure 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。