

# 日々是Oracle APEX

Oracle APEXを使った作業をしていて、気の付いたところを忘れないようにメモをとります。

2022年8月8日月曜日

## REST対応SQLサービスによるMySQLへのアクセス

[先日の記事](#)にて、Always FreeのAmpere A1インスタンスにMySQLサーバーをインストールし、Oracle Cloudのデータベース・ツールの接続サービスより利用できるようにしました。

Peter O'Brienさん（米国オラクル社の開発者）が、彼のブログ記事[How to use the Oracle Database Tools Service to provide data to APEX](#)で説明していますが、Oracle Cloudが提供しているSQLワークシートやデータベース・アクションは、ORDSの機能である[REST対応SQLサービス](#)（REST対応SQLサービス自体はAPEXではなく、ORDSが実装している機能）経由で、データベースにSQLを発行しています。

つまり、データベース・ツールの接続よりSQLワークシートが使えるのであれば、[REST対応SQLサービス](#)を通してSQLを実行することも可能です。APEXからも[REST対応SQLサービス](#)をデータ・ソースとして、レポートやチャートを作成することができます。Peter O'Brienさんの記事によると、データベース・ツールの接続の[REST対応SQLサービス](#)のエンドポイントの形式は以下になります。< region >はap-tokyo-1やus-ashburn-1といったリージョンの名前、< connection ocid >は接続対象の接続のOCIDです。エンドポイントURLの形式については、Oracle Cloud Infrastructure Documentationの[REST APIsのセクション](#)にて説明されています。20201005はAPI Versionになります。

`https://sql.dbtools.< region >.oci.oraclecloud.com/20201005/ords/< connection ocid >/_/sql`  
（APEXでの[REST対応SQLサービス](#)のエンドポイント指定では、末部の/\_/sqlは省略されます。）

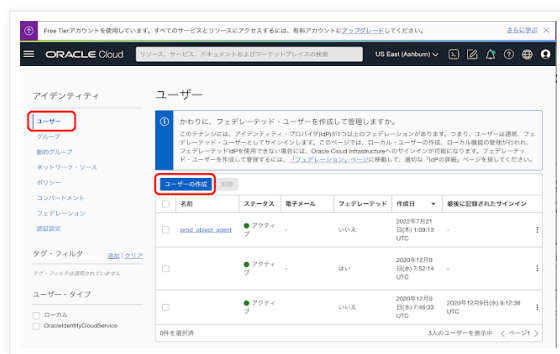
これから、Oracle Cloud側での準備として、IAMユーザーの作成、APIキーの追加、グループの作成、ポリシーの作成を行います。Oracle APEXでは、Web資格証明の作成と[REST対応SQLサービス](#)の作成を行い、動作確認を実施します。



## Oracle Cloud側の準備

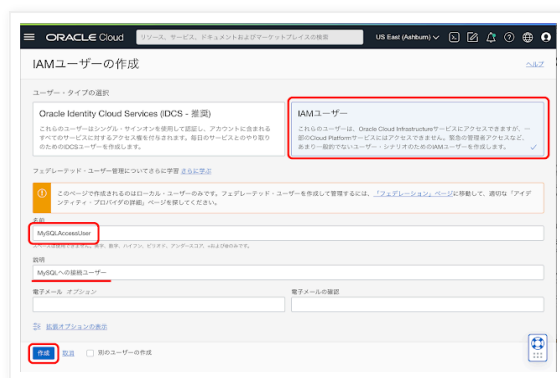
データベース・ツールの接続によって提供されるREST対応SQLサービスの呼び出しは、APIキーによって保護されています。そのため、APIキーを持つIAMユーザーを作成します。

Oracle Cloudのコンソールよりアイデンティティのユーザーを開き、ユーザーの作成を実行します。

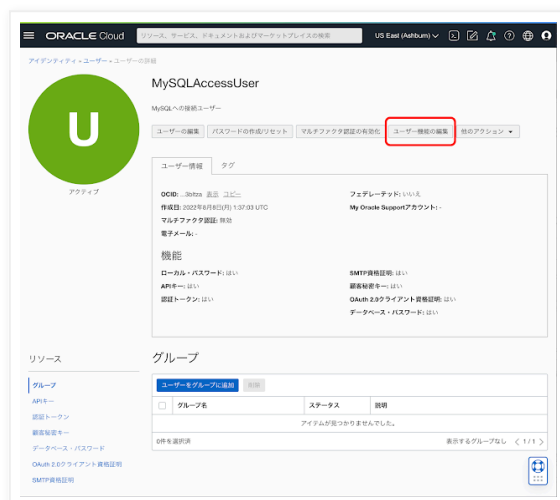


ユーザー・タイプの選択として、IAMユーザーを選択します。名前はMySQLAccessUser、説明にはMySQLへの接続ユーザーと記述します。

以上で、作成をクリックします。



ユーザーMySQLAccessUserはすぐに作成されます。ユーザー機能の編集をクリックします。



APIキーによるアクセスのみに使用するため、APIキーだけにチェックを入れ、それ以外の機能はすべてチェックを外します。

変更の保存をクリックします。

ユーザー機能の編集

ヘルプ

☐ ローカル・パスワード  
☒ **APIキー**  
☐ 認証トークン  
☐ SMTP資格証明  
☐ 顧客秘密キー  
☐ OAuth 2.0クライアント資格証明  
☐ データベース・パスワード

**変更の保存**
 取消

リソースのAPIキーを選択し、APIキーの追加をクリックします。

ORACLE Cloud

MySQLAccessUser

MySQLへの接続ユーザー

ユーザー機能の編集

パスワードの作成/リセット

マルチファクタ認証の有効化

ユーザー機能の編集

他のアクション

ユーザー情報

タグ

機能

ローカル・パスワード: いいえ

APIキー: はい

認証トークン: いいえ

SMTP資格証明: いいえ

顧客秘密キー: いいえ

OAuth 2.0クライアント資格証明: いいえ

データベース・パスワード: いいえ

リソース

グループ

**APIキー**

認証トークン

顧客秘密キー

データベース・パスワード

OAuth 2.0クライアント資格証明

SMTP資格証明

APIキーの追加

APIキーの追加

ダウンロード

作成日

アイテムが見つかりませんでした。

表示するAPIキーなし

APIキーの追加を行うダイアログが開きます。

方法にはいくつかの選択肢があります。今回は、事前の作業が不要な手順である**APIキー・ペアの生成**を選択します。**秘密キーのダウンロード**をクリックすると、秘密キーがファイルとしてダウンロードされます。ダウンロードされた秘密キーのファイルは、大事に保管しておきます。

秘密キーをダウンロードした後に、**追加**をクリックします。

APIキーの追加

ヘルプ

ノート: APIキーは、APIリクエストの署名に使用されるPEM形式のRSAキー・ペアです。ここでキー・ペアを生成して秘密キーをダウンロードできます。すでにキー・ペアを保持している場合は、かわりに公開キー・ファイルをアップロードするか、貼り付けることを選択できます。[さらに学ぶ](#)

☒ **APIキー・ペアの生成**
☐ 公開キー・ファイルの選択
 ☐ 公開キーの貼付け

公開キー

① 秘密キーをダウンロードします。これは再度表示されません。ダウンロードしたら、自分のみが表示できるように[ファイル権限を変更](#)します

**秘密キーのダウンロード**
 公開キーのダウンロード

**追加**
 取消

構成ファイルがプレビューされます。ここでプレビューされている情報は、APEXの**Web資格証明**を作成する際に使用するので、**コピー**をクリックします。後で参照できるように、コピーした内容を**ファイル**などに保存しておきます。

閉じるをクリックします。



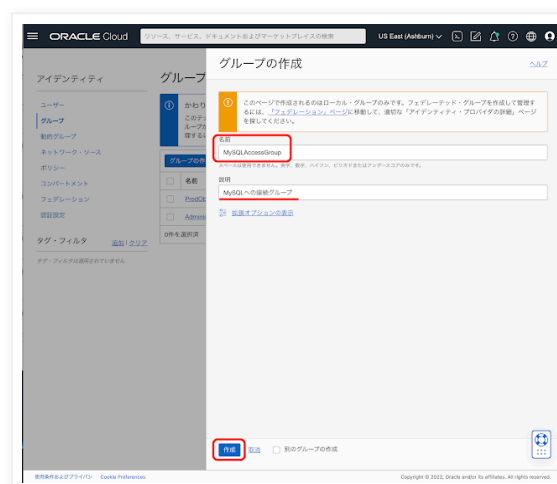
APIキーが登録されると、ユーザーの準備は完了です。



続いてアイデンティティのグループを開き、グループの作成を実行します。



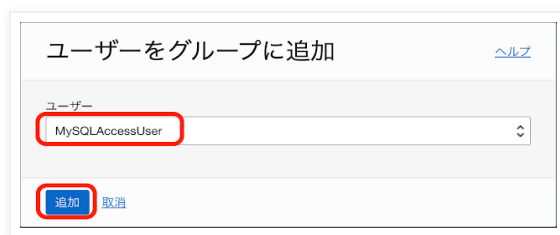
名前はMySQLAccessGroupとします。説明はMySQLへの接続グループと記述し、作成をクリックします。



グループMySQLAccessGroupが作成されます。リソースのグループ・メンバーを開き、ユーザーをグループに追加をクリックします。



ユーザーにMySQLAccessUserを選択し、追加をクリックします。

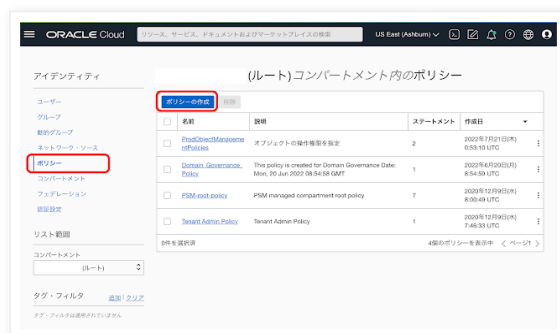


グループMySQLAccessGroupに、APIキーを持ったユーザーMySQLAccessUserが追加されました。



このグループMySQLAccessGroupに、データベース・ツールの接続の利用を許可するポリシーを定義します。

アイデンティティのポリシーを開き、ポリシーの作成を実行します。今回の作業はFree Tierアカウントで行っており、コンパートメントは積極的に使っていません。MySQLへの接続もルート・コンパートメントに作成しているため、ポリシーもルート・コンパートメントに作成します。



作成するポリシーの名前はMySQLAccessPolicyとし、説明にMySQLへのアクセス許可と記述します。コンパートメントはルート、ポリシー・ビルダーの手動エディタの表示をONにし、ポリシーと

して以下を記述します。ポリシーの記述については、マニュアルの[こちら](#)を参照しました。  
**database-tools-connections**へのポリシー以外に、**secret-family**の**read**も必要です。

Allow group MySQLAccessGroup to use database-tools-connections in tenancy  
Allow group MySQLAccessGroup to read secret-family in tenancy  
Allow group MySQLAccessGroup to read database-tools-connections in tenancy

以上を設定し、**作成**をクリックします。

Oracle Cloud console screenshot showing the 'Policy Creation' (ポリシーの作成) page. The policy name is 'MySQLAccessPolicy'. The description is 'MySQLへのアクセス許可'. The policy text contains three statements: 'Allow group MySQLAccessGroup to use database-tools-connections in tenancy', 'Allow group MySQLAccessGroup to read secret-family in tenancy', and 'Allow group MySQLAccessGroup to read database-tools-connections in tenancy'. The 'Create' (作成) button is highlighted with a red box.

ポリシー**MySQLAccessPolicy**が作成されます。

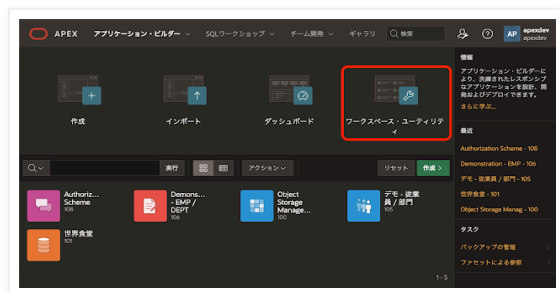
Oracle Cloud console screenshot showing the 'Policy Details' (ポリシーの詳細) page for 'MySQLAccessPolicy'. The policy is in 'Active' (アクティブ) state. The description is 'MySQLへのアクセス許可'. The policy text contains three statements: 'Allow group MySQLAccessGroup to use database-tools-connections in tenancy', 'Allow group MySQLAccessGroup to read secret-family in tenancy', and 'Allow group MySQLAccessGroup to read database-tools-connections in tenancy'. The 'Edit' (編集) button is highlighted with a red box.

以上で、Oracle Cloud側の準備は完了です。

## Web資格証明の作成

作成したユーザー**MySQLAccessUser**を使って、Oracle APEXの**Web資格証明**を作成します。Web資格証明はワークスペースに作成します。そのため、アプリケーションが無くても作成できます。

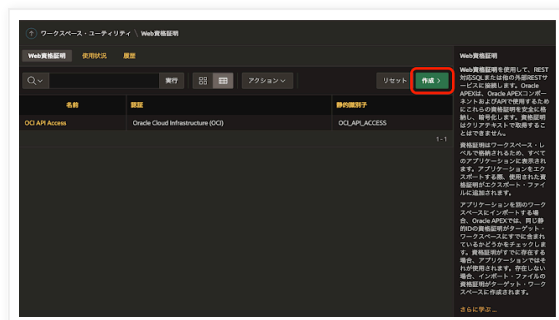
**アプリケーション・ビルダー**より**ワークスペース・ユーティリティ**を開きます。



ワークスペース・ユーティリティのWeb資格証明を開きます。



作成済みのWeb資格証明が一覧されます。作成をクリックします。



作成するWeb資格証明の名前は、MySQL Accessとします。認証タイプにOracle Cloud Infrastructure (OCI)を選択します。

IAMユーザーMySQLAccessUserにAPIキーを追加した際に表示された、構成ファイルのプレビューを参照します。

OCIユーザーIDとして、**user=**で指定されているocid1.userで始まる値を入力します。OCI秘密キーとして、APIキーの追加を行なったときにダウンロードした秘密キーを貼り付けます。OCIテナンシIDとして、構成ファイルの**tenancy=**として指定されているocid1.tenancyで始まる値を指定します。OCI公開キー・フィンガープリントとして、**fingerprint=**で指定されている値を入力します。

以上の設定を行い、作成をクリックします。



Web資格証明MySQL Accessが作成されました。



## REST対応SQLサービスの作成

最初にデータベースの接続より、REST対応SQLサービスでアクセスする対象の接続のOCIDを取得しておきます。



ワークスペース・ユーティリティのREST対応SQLサービスを開きます。





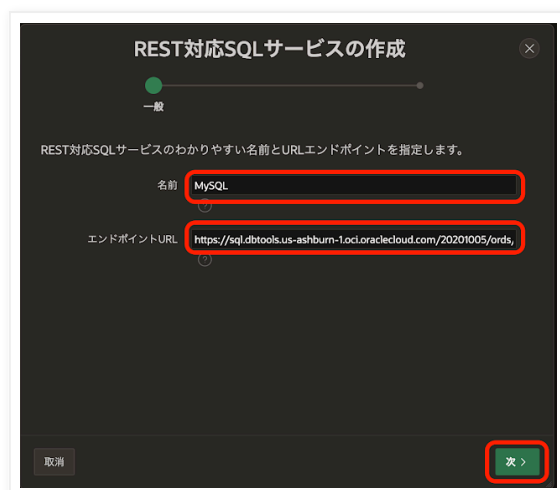
作成済みのREST対応SQLサービスが一覧されます。**作成**をクリックします。



REST対応SQLサービスの名前をMySQLとします。エンドポイントURLは、本記事の一番最初に紹介している以下の形式で指定します。

`https://sql.dbtools.< region >.oci.oraclecloud.com/20201005/ords/< connection ocid >`

次へ進みます。



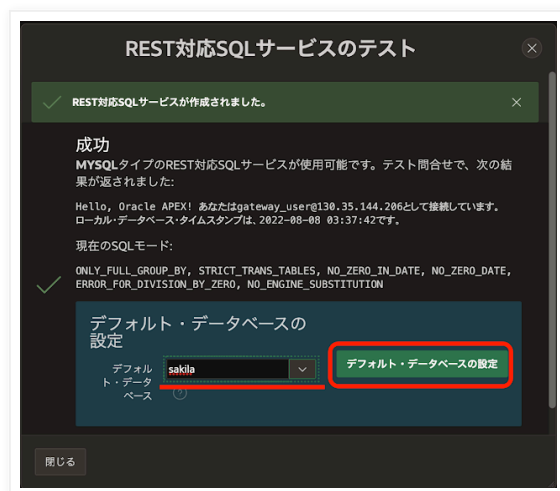
資格証明として、先ほど作成しているMySQL Accessを指定します。

**作成**をクリックします。



エンドポイントURLおよび資格証明が正しく設定されていると、REST対応SQLサービスの作成に成功します。

デフォルト・データベースとしてsakilaを選択し、デフォルト・データベースの設定をクリックします。



REST対応SQLサービスの一覧に、MySQLが表示されます。



以上で、MySQLにアクセスするOracle APEXのアプリケーションを作成する準備ができました。

REST対応SQLサービスは、その接続先がOracle DatabaseであるかMySQLであるかに関わらず、APEXのデータ・ソースとしては同様に扱うことができます。ただし、記述するSQL自体はMySQL向けにする必要があります。例えば、MySQLでは大文字と小文字が区別されます。

完

Yuji N. 時刻: 14:03

共有

<

ホーム

>

ウェブ バージョンを表示

自己紹介

Yuji N.

日本オラクル株式会社に勤務していて、Oracle APEXのGroundbreaker Advocateを拝命しました。こちらの記事につきましては、免責事項の参照をお願いいたします。

[詳細プロフィールを表示](#)