

# 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5月13日 金曜日

## データベース・ツールの接続よりAutonomous Databaseに接続する

Oracle Cloudが提供しているデータベース・ツールの接続（Database ToolsのConnections）を使用して、Always FreeのAutonomous Databaseへの接続を構成します。

データベース・ツールの接続を構成することにより、SQLワークシートやSQLcl（SQLを実行するコマンドライン・ツール、Cloud Shell内で実行されます）をすぐに呼び出すことができます。

Oracle APEXには、SQLを実行する画面（SQLコマンド）が付属しています。また、Autonomous Databaseにはデータベース・アクション（SQL Developer Web）が含まれ、これにもSQLを実行する画面が付属しています。どのツールにも一長一短はあるため、用途に応じて使い分けののが良いでしょう。

データベース・ツールの接続は、ボールトに保存されたシークレットを、データベース・ユーザーのパスワードとウォレットとして使用します。そのため、あらかじめボールトを作成し、シークレットの保存に使用するマスター暗号化キーを準備します。

## ボールトの準備

OCIコンソールのアイデンティティとセキュリティより、ボールトを開きます。



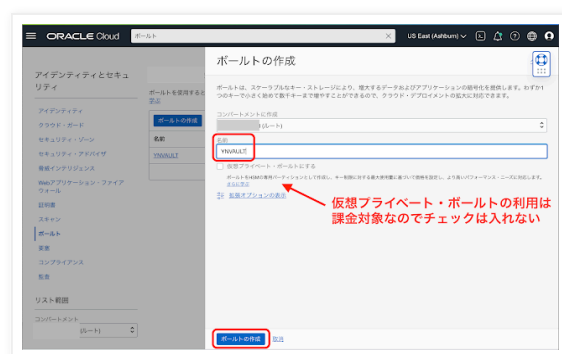
以下のスクリーンショットでは、ルート・コンパートメントにYNVAULTとしてボールトが作成済みです。未作成の場合は、ボールトの作成をクリックし、作成画面を開きます。



画面右にドロワーが開きます。

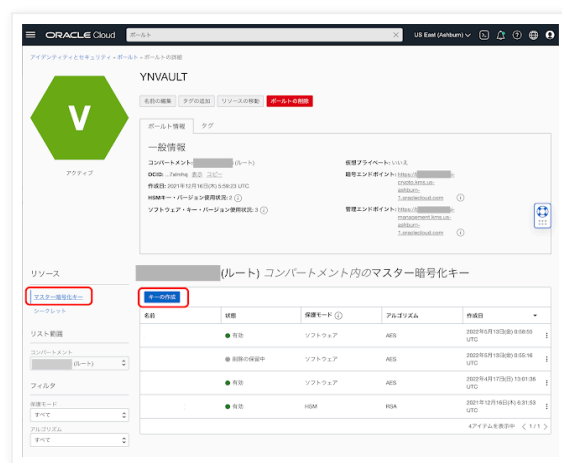
作成する**ボールドの名前**を入力します。**仮想プライベート・ボールドにするのチェックは入れません**。仮想プライベート・ボールドの利用は課金の対象になります。

**ボールドの作成**をクリックすると、指定した名前のボールドが作成されます。作成が完了するまでに、数分程度時間を要します。



作成された**ボールド**を開きます。

**リソースのマスター暗号化キー**を選択し、**キーの作成**を実行します。



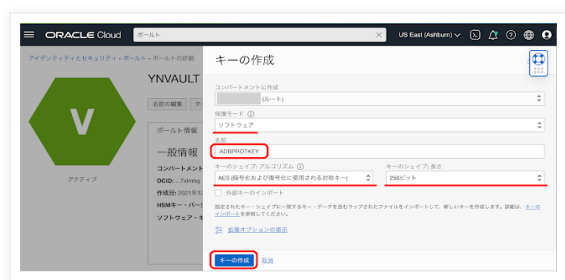
画面右にドロワーが開きます。

**保護モード**として**ソフトウェア**を選択します。保護モードがHSM（Hardware Security Moduleの略 - 物理的な安全性が担保されている専用デバイスのこと）の場合、20が無料で利用できる上限です。法令に準拠するといった要件がある場合は、保護モードにHSMを選択することになるかと思います。

今回は**名前**を**ADBPOTKEY**としています。

**キーのシェイプ**：アルゴリズムとしてAES(暗号化および復号化に使用される対称キー)を選択します。シークレットの保存には共通鍵暗号方式が使われるため、AESを選択します。**キーのシェイプ**：長さは選択できる最大値である**256ビット**を選択します。

以上を設定し、**キーの作成**を実行します。



マスター暗号化キーが作成され、少々時間が経過したのち**状態が有効**に変わります。



以上でボルトの準備は完了です。

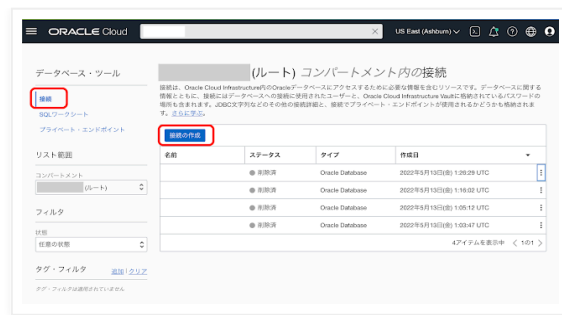
## 接続の作成

OCIコンソールより、**開発者サービス**に含まれる**データベース・ツールの接続**を開きます。



**接続**の画面が開きます。

**接続の作成**をクリックし、新規にデータベースへの接続を作成します。



画面右にドロワーが開きます。

今回想定している接続先は、Always FreeのAutonomous Databaseです。

**名前**には、接続先となるデータベースと接続ユーザーが特定できるような名前が望ましいです。今回は**apexdev-admin**としています。

APEXのSQLコマンドとデータベース・アクションの双方とも、接続ユーザーの切り替えはサインアウトとサインインのやり直しが必要です。**データベース・ツールのSQLワークシート**では**接続**の切り替えだけで接続ユーザーも切り替えられるので、その点は便利です。

**Database cloud service**には**Oracle Autonomous Database**を選択します。**データベースの選択**を選んでいると、指定しているコンパートメントに含まれるAutonomous Databaseのインスタンスを選択できます。今回は**データベースにAPEXDEV**を選択しています。**ユーザー名**としてAutonomous Databaseの管理者ユーザーである**admin**を指定しています。

**パスワード・シークレットの作成**をクリックし、データベースの接続ユーザー**ADMIN**のパスワードを**ボルトのシークレット**として保存します。

接続の作成

ヘルプ

1 接続詳細

2 SSL詳細

名前

apexdev-admin

コンパートメント

(ルート)

データベースの選択

データベース情報の入力

Database cloud service

Oracle Autonomous Database

(ルート)のデータベース

(コンパートメントの変更)

APEXDEV

ユーザー名

admin

(ルート)のユーザー・パスワード・シークレット

(コンパートメントの変更)

シークレットの選択

パスワード・シークレットの作成

接続文字列

(description=(retry\_count=20)(retry\_delay=3)(address

プライベート・エンドポイントを介したネットワーク接続

拡張オプションの表示

次

取消

ボルトにパスワードをシークレットとして保存する

→

画面右にドロワーが開きます。

シークレットの**名前**を入力します。シークレットとして保存されるのはパスワードである文字列のみで、その他の情報は含まれません。そのため、異なるデータベース、ユーザーでもパスワードの文字列として同じであればパスワード・シークレットとして使用できます。

**コンパートメント**、シークレットを保存する**ボールド**、シークレットの暗号化に使用する**マスター暗号化キー**を選択し、**ユーザー・パスワード**を入力します。

**作成**をクリックすると、パスワード・シークレットが作成されます。

パスワード・シークレットの作成

名前  
PassForAPEXDEV

説明 オプション

コンパートメント  
(ルート)

(ルート)のボールド  
(コンパートメントの変更)  
YNVAULT

(ルート)の暗号化キー  
(コンパートメントの変更)  
ADBPROTKEY

ユーザー・パスワード  
\*\*\*\*\*

ユーザー・パスワードの確認  
\*\*\*\*\*

拡張オプションの表示

**作成** 取消

作成されたパスワード・シークレットが、**ユーザー・パスワード・シークレット**として設定されます。**接続文字列**はデータベースを選択した時点で、適切な文字列が設定されるため変更は不要です。

次へ進みます。

### 接続の作成

[ヘルプ](#)

1 接続詳細

2 SSL詳細

名前

apexdev-admin

コンパートメント

(ルート)

データベースの選択

データベース情報の入力

Database cloud service

Oracle Autonomous Database

(ルート)のデータベース

(コンパートメントの変更)

APEXDEV

ユーザー名

admin

(ルート)のユーザー・パスワード・シークレット

(コンパートメントの変更)

PassForAPEXDEV (YNVAULTポールト内)

パスワード・シークレットの作成

接続文字列

(description=(retry\_count=20)(retry\_delay=3)(address

☐ プライベート・エンドポイントを介したネットワーク接続

拡張オプションの表示

次 取消

SSL詳細の画面に移ります。

接続先はOracle Autonomous Databaseなので、**ウォレット形式はSSOウォレット**になります。**ウォレット・コンテンツ・シークレット**を指定するため、**ウォレット・コンテンツ・シークレットの作成**をクリックします。

### 接続の作成

[ヘルプ](#)

1 接続詳細

2 SSL詳細

相互TLS (mTLS)認証の使用が必要な場合、またはTLS認証が使用されていて、信頼できる認証局によって署名されていない証明書がデータベースから返される場合、ウォレットを指定する必要があります。SSOウォレット(例: cwallet.sso)の使用をお薦めします。Javaキー・ストア(例: keystore.jks、truststore.jks)およびPKCS#12ウォレット(例: ewallet.p12)もサポートされています。

ウォレット形式

SSOウォレット(例: cwallet.sso)

(ルート)のSSOウォレット・コンテンツ・シークレット

(コンパートメントの変更)

シークレットの選択

ウォレット・コンテンツ・シークレットの作成

前 作成 取消

画面右にドロワーが開きます。

接続先がAutonomous Databaseであるため、接続に必要な情報はすべてリージョナル自動ログイン・ウォレットに含まれています。また、リージョナル自動ログイン・ウォレットだけがボールドのシークレットとして保存されます。そのため、すでに作成済みのウォレット・コンテンツ・シークレットがあれば再利用もできます。ただし、作成した時点のウォレットなので、その後に作成したAutonomous Databaseがあれば、その情報は含まれていません。その場合、ウォレット・コンテンツ・シークレットの再アップロードを実施します。

名前はリージョナル自動ログイン・ウォレットであることが分かる名前を付けます。今回はwallet-us-ashburn-1としています。コンパートメント、ボールド、マスター暗号化キーを選択し、Autonomous Databaseからのリージョナル自動ログイン・ウォレットの取得を選びます。

以上を設定し、作成をクリックします。

ウォレット・コンテンツ・シークレットの作成

名前  
wallet-us-ashburn-1

説明 オプション

コンパートメント  
(ルート)

(ルート)のボールド  
(コンパートメントの変更)

YNVAULT

(ルート)の暗号化キー  
(コンパートメントの変更)

ADBPOTKEY

☒ Autonomous Databaseからのリージョナル自動ログイン・ウォレットの取得

☐ 自動ログイン・ウォレットのアップロード

拡張オプションの表示

作成 取消

作成したウォレット・コンテンツ・シークレットが設定されます。

作成をクリックし、接続apexdev-adminを作成します。

接続の作成

ヘルプ

1 接続詳細  
2 SSL詳細

相互TLS (mTLS)認証の使用が必要な場合、またはTLS認証が使用されていて、信頼できる認証局によって署名されていない証明書がデータベースから返される場合、ウォレットを指定する必要があります。SSOウォレット (例: cwallet.sso)の使用をお薦めします。Javaキー・ストア (例: keystore.jks、truststore.jks)およびPKCS#12ウォレット (例: ewallet.p12)もサポートされています。

ウォレット形式  
SSOウォレット (例: cwallet.sso)

(ルート)のSSOウォレット・コンテンツ・シークレット  
(コンパートメントの変更)  
wallet-us-ashburn-1 (YNVAULTポータル内)

ウォレット・コンテンツ・シークレットの作成

前 作成 取消

接続が作成されると、接続の詳細画面が表示されます。

接続の詳細には、SQLワークシートの実行、Cloud Shell内でのSQLclの起動、パスワードの変更、自動ログイン・ウォレットの更新といった作業を行うリンクが含まれています。

apexdev-admin

SQLワークシートの実行

Cloud Shell内でSQLclの実行

パスワードの変更

自動ログイン・ウォレットの更新

接続情報

タグ

OCI ID: nlsadm 名前: apexdev-admin

コンパートメント: wallet-us-ashburn-1 (YNVAULTポータル内)

作成日: 2022年5月12日 (UTC)

更新日: 2022年5月12日 (UTC)

プライマリ・エンドポイント: nlsadm

データベース: apexdev

Database cloud service: Autonomous Database

接続文字列: ...

ユーザー名: admin

ユーザー・パスワード・シークレット: Password: apexdev

SSOウォレット・コンテンツ・シークレット: wallet-us-ashburn-1 (YNVAULTポータル内)

接続プロパティ

接続プロパティの追加

名前

値

アイテムが見つかりませんでした。

アイテムを選択

## パスワードの更新

ユーザー・パスワード・シークレットをクリックすると、シークレット詳細のバージョンが開かれます。ここでシークレット・バージョンの作成を実行し、シークレット・バージョンを更新します。





シークレット・タイプ・テンプレートとしてプレーン・テキストを選択します。シークレット・コンテンツとして、新しいパスワードを入力します。

シークレット・バージョンの作成を実行します。



新たなバージョンが作成され、ステータスが現在となると、パスワードの更新は完了です。



パスワード・シークレットを共用しているすべての接続で、使用するパスワードが変更されます。

## 自動ログイン・ウォレットの更新

再アップロードをクリックします。



シークレットの名前を入力し、再アップロードをクリックします。



以上でリージョナル自動ログイン・ウォレットの更新は完了です。

SSOウォレット・コンテンツ・シークレットを開いて確認してみます。



新たにシークレット・バージョンが作成されていることが確認できます。



## SQLワークシートの使用

接続の詳細画面よりSQLワークシートを実行できます。または、OCIコンソールからも起動できます。



SQLワークシートでは、接続先のデータベースや接続ユーザーを、サインインの操作をすることなく切り替えることができます。切り替え時に記述済みのSQLも維持されます。



それ以外では、記述したSQLをオブジェクト・ストレージに保存することができます。



逆に実行するSQLを、オブジェクト・ストレージからロードすることもできます。



SQLワークシートを使用することにより、特に複数のデータベースで同時に扱う場合に、効率良く作業を行うことができそうです。

## SQLclの起動

**SQLclの起動**を行うと、Cloud Shell内でSQLclを実行され、データベースへの接続までが実施されます。

Javaのインストール、SQLclのダウンロード、ウォレットのダウンロードといった作業が不要で、すぐにSQLの実行ができるようになります。



Cloud ShellはSQLclが起動した状態で開きますが、!コマンドでSQLclからOSのシェル環境を呼び出すことができます。

データベース・ツールの接続の紹介は以上になります。

完

Yuji N. 時刻: 15:14

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

Powered by Blogger.