日日是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。

2021年12月9日木曜日

Oracle RDF Graph ServerをAutonomous Databaseで使用する(4) - Jetty 9.xのSSL化

Oracle APEXにもOracle RDF Graph Serverにも直接は関係ありませんが、色々と作業があったので記録しておきます。

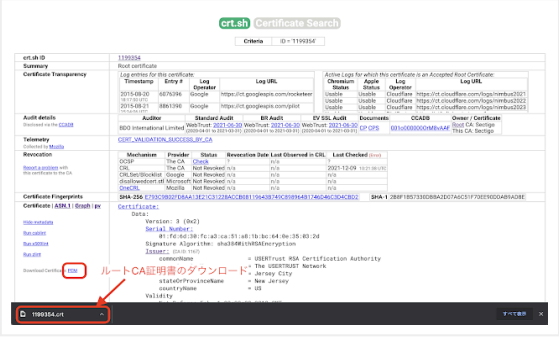
SSLやホスト名がDNSに登録されていないとREST APIを呼べない、というのはAutonomous Databaseの制約です。HTTPでIPアドレス指定でも（CSRFの対応は心配ですが）Oracle RDF Graph ServerのREST APIは動作すると思います。

以下よりJetty 9.xのSSL化にあたって行った作業を記載します。

最初にコンピュート・インスタンスに割り当てられた**パブリックIP**に割り当てる**ホスト名**を決めて、DNSに登録します。次にSSLのサーバー証明書の取得を行います。以前にOracle REST Data ServicesをSSL化するために実施した手順と同じです。

Google DomainsとZero SSLのサービスを使用しています。

Zero SSLより証明書を**certificate.crt**、中間証明書を**ca_bundle.crt**、秘密鍵のファイルを**private.key**として入手しています。ルートCAの証明書は別途取得します。Zero SSLはUSERTrust RSA Certification AuthorityがルートCAなので、https://crt.sh/?id=1199354よりPEMをダウンロードしました。このファイルはサーバー証明書の取得先で変わります。



Oracle RDF Graph Serverを実行しているコンピュート・インスタンスに接続し、jetty.homeへ移動します。

```
[opc@rdfgs ~]$ cd jetty-distribution-9.4.44.v20210927/
[opc@rdfgs jetty-distribution-9.4.44.v20210927]$ pwd
/home/opc/jetty-distribution-9.4.44.v20210927
[opc@rdfgs jetty-distribution-9.4.44.v20210927]$
```

今までは、Jettyの設定は**start.ini**にまとまっていました。SSLの設定がしにくいので、**start.d**以下のファイルに、それぞれのモジュール毎に記述する方法に変更します。start.iniは削除し、demo-base/start.dをコピーします。start.dに含まれているdemo.iniは不要なので消去します。

```
[opc@rdfgs jetty-distribution-9.4.44.v20210927]$ cp -r demo-base/start.d start.d
[opc@rdfgs jetty-distribution-9.4.44.v20210927]$ rm start.ini
[opc@rdfgs jetty-distribution-9.4.44.v20210927]$ rm start.d/demo.ini
```

/home/opc以下にsslというディレクトリを作成し**certificate.crt**、**ca_bundle.crt**、**private.key**、**1199354.crt**（ルートCA証明書）の４つのファイルを配置します。

最初に証明書を１つのファイルに連結します。サーバー証明書、中間証明書、ルートCA証明書の順番で連結し、ファイル**certchain.crt**を作成します。

cat certificate.crt ca_bundle.crt 1199354.crt > certchain.crt

```
[opc@rdfgs ssl]$ cat certificate.crt ca_bundle.crt 1199354.crt > certchain.crt
```

OpenSSLを使用して証明書のファイル**certcachain.crt**と秘密鍵**private.key**より、PKCS#12形式のファイルを作成します。PKCS#12のファイル名は**my.p12**にしています。

openssl pkcs12 -export -in certchain.crt -inkey private.key -out my.p12

ここで指定するパスワードは、次に実行する**keytool**で要求される**source keystore password**に与えます。

```
[opc@rdfgs ssl]$ openssl pkcs12 -export -inkey private.key -in certchain.crt -out my.p12
Enter Export Password: ********
Verifying - Enter Export Password: ********
[opc@rdfgs ssl]$
```

keytoolを使用してPKCS#12形式からJKS形式のキーストア・ファイルを生成します。**destination keystore password**は、この後に行うJettyのSSL設定に含めます。

keytool -importkeystore -srckeystore my.p12 -srcstoretype PKCS12 -destkeystore keystore.jks

```
[opc@rdfgs ssl]$ keytool -importkeystore -srckeystore my.p12 -srcstoretype PKCS12 -destkeystore
keystore.jks
Importing keystore my.p12 to keystore.jks...
Enter destination keystore password:  ++++++++
Re-enter new password: ++++++++
Enter source keystore password: ********
Entry for alias rdf.apexugj.dev successfully imported.
Import command completed:  1 entries successfully imported, 0 entries failed or cancelled

Warning:
The JKS keystore uses a proprietary format. It is recommended to migrate to PKCS12 which is an
industry standard format using "keytool -importkeystore -srckeystore keystore.jks -destkeystore
keystore.jks -deststoretype pkcs12".
[opc@rdfgs ssl]$
```

作成したkeystore.jksをjetty.homeの**etc**以下に移動します。

mv keystore.jks /home/opc/jetty-distribution-9.4.44.v20210927/etc/

```
[opc@rdfgs ssl]$ mv keystore.jks /home/opc/jetty-distribution-9.4.44.v20210927/etc/
```

ユーザーopcのホームに戻り、destination keystore passwordの難読化を行います。**OBF**で始まる文字列をパスワードとして使用します。

```
[opc@rdfgs ~]$ cd
[opc@rdfgs ~]$ java -cp jetty-distribution-9.4.44.v20210927/lib/jetty-util-9.4.44.v20210927.jar
org.eclipse.jetty.util.security.Password **********
2021-12-09 06:33:46.903:INFO::main: Logging initialized @382ms to
org.eclipse.jetty.util.log.StdErrLog
oracle
```

```
OBF:1v**********x151v1x
MD5:a189c633d9995e11bf8607170ec9a4b8
[opc@rdfgs ~]$
```

${jetty.home}/start.d以下に**ssl.ini**を作成します。内容は以下になります。

```
--module=ssl

jetty.ssl.host=0.0.0.0
jetty.ssl.port=8443
jetty.https.port=8443
jetty.httpConfig.securePort=443
jetty.sslContext.keyStoreType=JKS
jetty.sslContext.keyStorePath=etc/keystore.jks
jetty.sslContext.trustStorePath=etc/keystore.jks
jetty.sslContext.keyStorePassword=OBF:1v2h**********151v1x
jetty.sslContext.keyManagerPassword=OBF:1v2h**********151v1x
jetty.sslContext.trustStorePassword=OBF:1v2h**********151v1x
```

以上でJettyの設定は完了です。

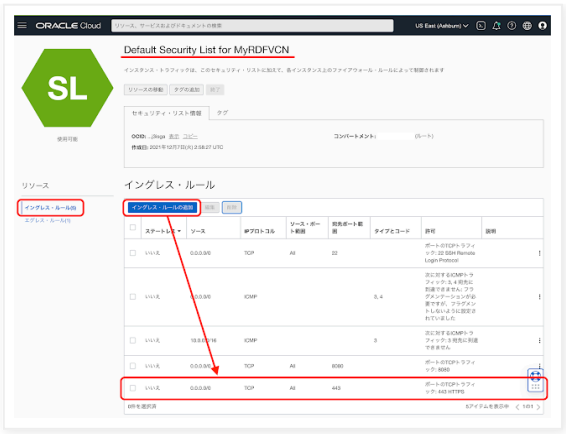
続いてfirewalldの設定を行います。443番ポートで接続の待ち受けを行うので、接続許可を与えます。443番ポートへの接続はJettyが動作しているポート8443番に転送します。最後に設定を永続化します。

firewall-cmd --add-port=443/tcp
firewall-cmd --add-forward-port=port=443:proto=tcp:toport=8443
firewall-cmd --runtime-to-permanent

これらの手順は、Oracle RDF Graph Serverの環境構築の際に8080番ポートを対象に実施した作業と同じです。

```
[opc@rdfgs bin]$ sudo firewall-cmd --add-port=443/tcp
success
[opc@rdfgs bin]$ sudo firewall-cmd --add-forward-port=port=443:proto=tcp:toport=8443
success
[opc@rdfgs bin]$ sudo firewall-cmd --runtime-to-permanent
success
[opc@rdfgs bin]$ sudo firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: ens3
  sources:
  services: ssh
  ports: 8080/tcp 443/tcp
  protocols:
  masquerade: no
  forward-ports:
        port=443:proto=tcp:toport=8443:toaddr=
  source-ports:
  icmp-blocks:
  rich rules:
[opc@rdfgs bin]$
```

パブリック・ネットワークの**セキュリティ・リスト**に、**443**番ポートへの通信を許可する**イングレス・ルール**を作成します。

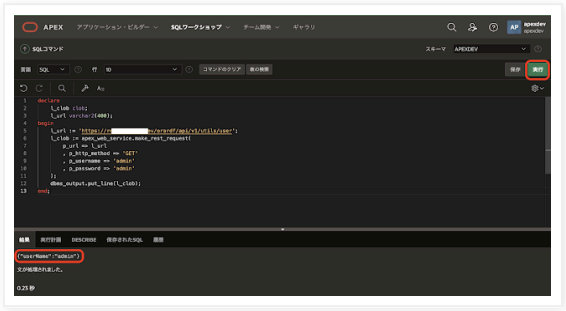


以上でJettyのSSL化は完了です。設定を反映させるため、Jettyを再起動します。

動作を確認します。Oracle APEXのSQLコマンドより、以下のコードを実行します。

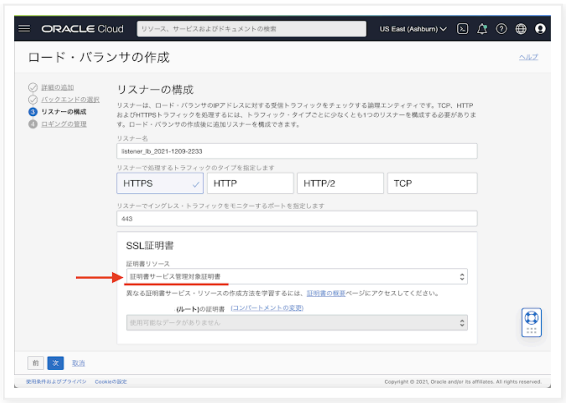
```
declare
   l_clob clob;
   l_url varchar2(400);
begin
   l_url := 'https://ホスト名/orardf/api/v1/utils/user';
   l_clob := apex_web_service.make_rest_request(
      p_url => l_url
      , p_http_method => 'GET'
      , p_username => 'admin'
      , p_password => 'admin'
   );
   dbms_output.put_line(l_clob);
end;
```

引数p_usernameに与えた名前が応答として返されます。以下では**{"userName":"admin"}**が返されています。

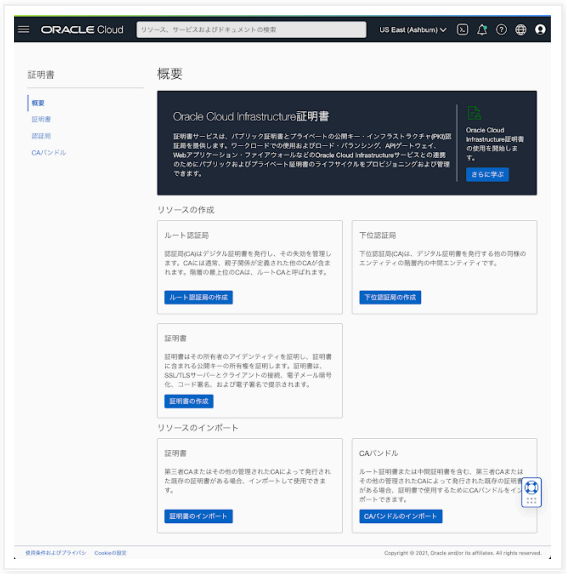


SSLでの暗号化/復号化はロード・バランサで行い、JettyではHTTPで通信するという構成も可能です。最近、Oracle Cloud Infrastructureに証明書のサービスが追加されたため、証明書のローテーションなどを自力で行う必要が無くなりました。

ロード・バランサの作成時に**SSL証明書**の**証明書リソース**として、**証明書サービス管理対象証明書**を選択します。



証明書のサービスを使って、証明書のライフサイクルを管理します。



本番環境の場合は証明書サービスの利便性を考慮して、SSLはロード・バランサにオフロードさせる方が良いかもしれません。

続く

Yuji N. 時刻: 23:03

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