Notes et observations

Le projet consistait à valider les taches pour obtenir une instance Oracle APEX opérationnelle sur le cloud OVH en SSL avec un nom de domaine et au coût le plus bas.

Il n'y a rien de vraiment compliqué mais comme souvent, de glorieuses incertitudes sont venues pimenter le travail.

Voici les étapes que j'ai identifiées :

Préparer l'infrastructure

Obtenir un nom de domaine

Le nom de domaine sera désigné par <DOMAINE.TLD> dans la suite du document

Souscrire un serveur chez OVH

Désignons-le par « Serveur Y » dans la suite.

Sandbox S1-8 (2vCores, 8Go RAM, 40 Go disque) 14,30 € H.T. /mois en Cloud Public.
Cette variante de type *Sandbox* est valable uniquement pour des tests car elle n'offre pas la stabilité d'une instance conventionnelle.
Il faudrait plutôt s'orienter vers une instance Linux Centos (B2-7 à 24 € H.T. /mois ou B2-15 à 46 € H.T. /mois)

En comparaison, on peut garder en mémoire que la souscription au service Oracle APEX service revient à 333 € /mois (en date d'avril 2023)

J'ai choisi une image de type Centos 7

Pas la peine de prendre trop de cpu car Oracle XE est capé à 2 cores.

De même, l'espace disque dédié à la database est limité à 12 Go. Cependant, prévoir 40 Go car une installation basique occupe pas loin de 17 Go.

Télécharger les softs Oracle

Aucun n'entrainent des couts de licence.

- Oracle Express Edition 21c
- Oracle APEX 22
- Oracle ORDS 22

Installer Oracle Database 21c Express

Installer Les prérequis pour Oracle XE 21 c
dans la page de download, cliquer sur *Preinstall RPMs for RHEL and CentOS*

Installer Oracle XE 21 c
Rien à signaler. Suivre le cookbook livré dans la doc officielle.

Vérifier le fichier `/etc/hosts`

Vérifier le port utilisé pour sqlnet et rectifier le port si besoin dans le fichier `listener.ora`.

```
/opt/oracle/homes/OraDBHome21cXE/network/admin/listener.ora
```

Installer Oracle APEX

Installer Oracle APEX 22 dans le PDB présent par défaut (xepdb1)

Ajouter dans le domaine créé plus haut une redirection sur un enregistrement de type A (l'adresse IP V4 du serveur Y)

Bien vérifier que le sous domaine créé dirige vers l'adresse IP en effectuant un ping à la fois depuis le serveur Y et depuis son poste de travail.
Dans la suite, je désigne par NOM.DOMAINE.TLD le nom (*fqdn*) que je vais attribuer à mon serveur Y.

Installer Oracle ORDS

Etre connecté sous le compte oracle:oinstall
Créer un répertoire `/opt/ords/conf`

Ajouter la variable d'environnement `ORDS_CONFIG=/opt/ords/conf` dans `.bash_profile`

Préciser dans la chaine de connexion « localhost » et préciser le PDB: xepdb1.
Contrôler que le répertoire de configuration est ok.

Installer Apache Tomcat

Installer Tomcat 9
créer un fichier `setenv.sh` dans le répertoire `/opt/tomcat/latest/bin`

```
export ORDS_CONFIG=/opt/ords/conf
#export CATALINA_HOME=/opt/tomcat/latest
#export CATALINA_OPTS="$CATALINA_OPTS -Duser.timezone=UTC"
```

```
export JAVA_OPTS="-Dconfig.url=${ORDS_CONFIG} -Xms1024M
```

Déployer `ords.war` dans le répertoire `/opt/tomcat/latest/webapps`

Une différence importante avec les version précédentes de ORDS est que la localisation des fichiers de configuration ne se trouve plus automatiquement dans le .war (quoique l'on puisse le régénérer avec la commande `ords war`) . Il est plus sûr de fournir le path via le script `setenv.sh`

Recopier les fichiers du répertoire images dans `/opt/tomcat/latest/webapps/i`

Tester depuis un navigateur

`http://<IP_ADDRESS>:8080/ords`
`http://<IP_ADDRESS>:8080/ords/apex_admin` (pour l'accès au service d'administration)

Occupation espace disque à ce stade de l'installation. Le 20 Go (20960236) était la valeur initiale avec une instance OVH modeste que j'ai upgradée ensuite à 40 Go.

```
Filesystem     1K-blocks     Used Available Use% Mounted on
devtmpfs         1865500        0   1865500   0% /dev
tmpfs            1891000        0   1891000   0% /dev/shm
tmpfs            1891000     8716   1882284   1% /run
tmpfs            1891000        0   1891000   0% /sys/fs/cgroup
/dev/sda1       20960236 17065564   3894672  82% /
```

```
tmpfs              378200        0    378200   0% /run/user/1000
```

Mise en place de TLS

La méthode que je privilégie est d'installer un Apache Listener en Reverse Proxy. C'est lui qui sera chargé de la vérification des certificats et du protocole TLS.

Installer Apache HTTPD

Préciser un Virtual Host dans le fichier /etc/https/conf/httpd.conf

Installer certbot et obtenir un certificat auprès de Lets'Encrypt.

La procédure paramètre automatiquement la configuration d'Apache.

Tester depuis un navigateur:

https://<NOM.DOMAINE.TLD>

Rediriger les requêtes TLS vers Tomcat

Il s'agit de rediriger les requetes SSL vers Tomcat qui « écoute » en http sur le port 8080.

J'utilise `mod_proxy`.

```
<VirtualHost *:80>
        Redirect "/" "https://<NOM.DOMAINE.TLD>/"
        #ProxyPreserveHost On
        #RequestHeader unset Origin
RewriteEngine on
RewriteCond %{SERVER_NAME} = <NOM.DOMAINE.TLD>
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
[END,NE,R=permanent]
```

```
</VirtualHost>
```

Remarque importante: Si *Service unavailable*, il faut probablement modifier une valeur de paramètre :

```
setsebool -P httpd_can_network_connect on
```

Dans le répertoire /etc/httpd/conf.d , il faut modifier le fichier ssl.conf pour indiquer la façon dont se fera la redirection vers tomcat:

```
<VirtualHost *:443>

# General setup for the virtual host, inherited from global
configuration
DocumentRoot "/var/www/html"
ProxyPreserveHost On
ServerName <NOM.DOMAINE.TLD>
#ProxyRequests Off
#RequestHeader set Host "<NOM.DOMAINE.TLD>:443"
ProxyPass /  http://<NOM.DOMAINE.TLD>:8080/
ProxyPassReverse /  http://<NOM.DOMAINE.TLD>:8080/

SSLEngine on
```

```
SSLProxyEngine on
```

Remarque Importante: Dans le fichier de configuration httpd.conf, il y a la directive suivante: `IncludeOptional conf.d/*.conf`

Cela signifie que TOUS les fichiers .conf vont être inclus. Donc, veiller à ne pas sauvegarder des fichiers de backup dans ce même répertoire, sinon on obtient des erreurs du type: *port déjà utilisé*, ou bien *Nom de serveur déjà utilisé* qui empêchent le démarrage de Apache Httpd.

A cette occasion, j'ai vite constaté (avec la commande `journalctl -xe`) que le serveur faisait l'objet de nombreuse tentatives de connexions ssh . J'ai neutralisé les accès en agissant sur les deux fichiers

- `/etc/hosts.allow`
- `/etc/hosts.deny`:

Complément paramétrage Tomcat

dans la balise <Connector …> ajouter:

```
proxyPort="443"
scheme="https"
```

Tester avec un navigateur:

https:<NOM.DOMAINE.TLD>/ords

Permettre les communications HTTPS en sortie

Il faut autoriser les communications entre la database et l'extérieur

ACL network

Créer un wallet DB pour les certificats

Créer un *wallet* et y enregistrer les certificats nécessaires. Cette étape est obligatoire pour réaliser des appels REST qui doivent impérativement utiliser TLS.

.