Intro

Ce sont des note de travail sur une première prise de contact avec Oracle Application Container Cloud Service (Oracle ACCS/Node).

Oracle ACCS propose deux types de containers applicatifs:

- Java SE (Java)
- Node.js (Javascript)

Ce post concerne Node.js.

On déploie une application Javascript comme si on travaillait avec un serveur d'application, mais avec un niveau de cloisonnement plus poussé.

Prérequis

Suivre le tutorial: Developing a RESTful Node.js and HTML5 application to Oracle Application Container Cloud Service

Il s'agit d'une application qui liste des sujets et qui permet d'en ajouter ainsi que des commentaires.

Il y a, à coté, une page html (index.html) qui fait des appels à cette application.

Pour info: Oracle ACCS propose une application sample déjà installée sur le serveur. Cette application est instanciable lors de la création d'une nouvelle appli.

Développement et test en local

Pour tester le fonctionnement, avant de déployer sur Oracle ACCS, il faut installer node.js sur son PC et suivre le tutorial.

Optionnel: Pour inclure des appels REST vers d'autres services, il faut utiliser le module *require*.

> Avant de lancer la commande node server.js, il faudra positionner la variable NODE_PATH à la valeur suivante:
> set NODE_PATH=C:\Users\<USERNAME>\AppData\Roaming\npm\node_modules

Le résultat ressemble à :

Déploiement

Un fois que c'est vérifié:

- Bien vérifier la variable port et la fixer à la valeur suivante:

```
var PORT = process.env.PORT || 80;
```

- on fabrique un .zip avec l'application node (server.js + un fichier Manifest)
- On se connecte à Oracle ACCS et on crée une nouvelle application de type Node.js en chargeant le fichier zip précédent.
- On récupère l'adresse et on met à jour la page HTML pour pointer vers l'adresse de l'application.

Test du bon fonctionnement

Soit on déclenche la page index.html dans un browser, soit on appelle directement l'URL https://testpmo-gse00000405.apaas.em2.oraclecloud.com
Observer que l'URL est préfixée par le nom que l'on donne à l'application lors de sa création

Conclusion: c'est extrêmement simple à déployer !

Avantages pour l'appel de service REST

L'application est directement accessible en SSL.

On peut développer en Javascript et échapper aux contraintes de *Cross Origin Domain* puisque les appels REST vers d'autres services se feront depuis un serveur et non pas depuis un browser. cela est particulièrement vrai pour les services REST qui ne sont pas encore *CORS enable*.
(pour info: Les APIs de Oracle Documents Cloud Service sont désormais CORS *enable* depuis cet été 2016*)*

A tester: Il faut vérifier comment les applications Node interagissent avec l'identity Domain. Est-ce que l'on peut utiliser l'authentification native du *tenant* et récupérer , au niveau de l'application, le user connecté ?