

## FICHE DE TP DE GOOGLE APP ENGINE EN JAVA

Cette fiche comporte essentiellement quatre parties : la première partie qui consiste à la mise en place des outils nécessaires pour créer un projet App Engine sur le cloud et aussi ceux pour créer notre application web en locale.

La seconde partie consistera à créer une application GAE sur le cloud de Google en utilisant l'outil **gcloud**, à choisir la région géographique où sera déployé notre site web et enfin déployé notre site web local en ligne.

La troisième partie va consister à intégrer le service **Google Account** sur notre application web.

La quatrième partie va consister à intégrer le service **EMAIL** sur notre application web.

Ce TP est fait sous le système Linux-Ubuntu.

### PARTIE I :

#### Prérequis :

Vous devez avoir :

- ✓ Installer Eclipse sur vos pc ainsi que les outils de développement web avec java et sous Eclipse.
- ✓ Installer le plugin Google App Engine sous Eclipse ce qui nous permettra de créer facilement nos applications App Engine avec tout son architecture.
- ✓ Installer une version de python2  $\geq 2.7.9$
- ✓ Installer la JDK les versions 8 ou 11

Nous pouvons à présent commencer.

Pour connaître votre version de python2 taper en ligne de commande :

```
$ sudo python2 --version
```

Pour connaître votre version de votre jdk taper en ligne de commande :

```
$ sudo java -version
```

#### 1) Installation de Google Cloud SDK

Il est important de mettre à jour le fichier source des packages pour qu'il prenne désormais en compte le package de Google Cloud SDK.

Ouvrez un terminal et tapez :

```
$ echo "deb [signed-by=/usr/share/keyrings/cloud.google.gpg] https://packages.cloud.google.com/apt cloud-sdk main" | sudo tee -a /etc/apt/sources.list.d/google-cloud-sdk.list
```

Ensuite il faut installer le package **apt-transport-https**. Pour cela tapez la commande

```
$ sudo apt-get install apt-transport-https ca-certificates gnupg
```

Maintenant il faut importer la clé publique de Google Cloud qui va permettre de décrypter le package Google Cloud SDK qui sera télécharger. Il suffit de taper:

```
$ curl https://packages.cloud.google.com/apt/doc/apt-key.gpg | sudo apt-key --keyring /usr/share/keyrings/cloud.google.gpg add -
```

Enfin procédons à l'installation du Google Cloud SDK en tapant :

```
$ sudo apt-get update && sudo apt-get install google-cloud-sdk
```

Comme nous devons utiliser JAVA pour développer notre application web, nous devons installer un package supplémentaire qui est **google-cloud-sdk-app-engine-java** en tapant :

```
$ sudo apt-get update && sudo apt-get install google-cloud-sdk
```

**NB** : cela va vous prendre environ 350 Mo pour effectuer toutes ces opérations.

Ainsi nous avons vu dans cette partie comment installer Google Cloud SDK et le package App Engine pour java. Pour ce fait vous pouvez déjà commencer par initialiser Google Cloud en tapant :

```
$ gcloud init
```

Ceci permettra à Google de vous authentifier à partir de votre compte Google.

**BAS** : dans le cas où vous avez déjà eu à installer Google Cloud SDK dans le passé, tapez la commande suivante pour sa mise à jour :

```
$ gcloud components update
```

## PARTIE II

Dans cette partie, nous allons créer un projet GAP sur Google Cloud que nous batiserons tp\_GAE-1 ainsi qu'une application web JAVA EE sur Eclipse portant ce même nom et enfin nous allons voir comment exécuter notre application en local puis son déploiement sur Google Cloud.

### 1) Création d'un projet GAE sur Google Cloud

Il est question dans cette section de créer un projet sur Google Cloud ce qui nous donnera la possibilité plus tard d'héberger notre application web.

Il suffit de taper la commande :

```
$ gcloud projects create tp_GAE-1 --set-as-default
```

Ensuite nous pouvons vérifier si effectivement notre application a bien été créée en tapant la commande :

```
$ gcloud projects describe tp_GAE-1
```

Là vous verrez les différents paramètres de votre projet notamment son nom, ID, date et heure de création etc.

## 2) Initialisation de notre application GAE avec le cloud de Google

Ici il est question de choisir l'un des serveurs de Google qui nous seront proposés afin de pouvoir héberger notre application web et d'initialiser le dit répertoire.

Il suffit de taper la commande :

```
$ gcloud app create --project=tp-GAE-1
```

## 3) Déployer l'application sur le cloud de Google

### a) Sans gestionnaire de packages

Ici on suppose que votre application ne possède pas de gestionnaire de package tel que **Maven**.

Rassurez-vous dans un premier temps de que le package **App Engine** pour JAVA est effectivement installé en tapant la commande :

```
$ sudo apt-get install google-cloud-sdk-app-engine-java
```

Ensuite il faut retrouver où est situé le fichier appengine-web.xml dans l'arborescence de votre application. Il est en générale stocké dans */WEB-INF/appengine-web.xml* où le premier / représente le répertoire racine de votre projet.

Et puis tapez la commande :

```
$ gcloud app deploy chemin/appengine-web.xml
```

Et lorsque l'opération se termine pour pouvez à présent avoir accès à votre site web en ligne à travers l'adresse [https://tp\\_GAP-1.appspot.com/](https://tp_GAP-1.appspot.com/)

### b) Utiliser un gestionnaire de package (cas de **Maven**)

Il suffit juste de taper la commande :

```
$ mvn appengine:deploy
```

### PARTIE III

dans cette partie nous allons voir comment utiliser le service Google Accounts de GAE pour authentifier les utilisateurs au sein de notre application web.

### PARTIE IV

Dans cette partie nous verrons comment faire pour notifier les utilisateurs de notre application web par mail en se servant du service de GAE EMAIL.

Supplément : structure d'un fichier appengine-web.xml

```
<?xml version="1.0" encoding="utf-8"?>
<appengine-web-app xmlns="http://appengine.google.com/ns/1.0">
  <threadsafe>true</threadsafe>
  <sessions-enabled>false</sessions-enabled>
  <runtime>java8</runtime>
  <system-properties>
    <property name="java.util.logging.config.file" value="WEB-INF/logging.properties"/>
  </system-properties>
</appengine-web-app>
```

FIN