

# Amazon Web Services

## TLC TP2 + TP3

---

Le but de cet exercice est de déployer l'application Web WordPress en utilisant les services IaaS d'Amazon Web Services. Afin de permettre un cohabitation harmonieuse entre les différents groupes, merci de respecter quelques règles :

- Nous utiliserons uniquement des instances de type `t2.nano`.
  - Quand vous créez des machines virtuelles, merci d'indiquer vos initiales dans le nom donné à l'instance. De façon similaire, merci d'indiquer vos initiales dans le nom des clés que vous allez créer, des AMIs etc.
  - N'oubliez pas d'arrêter vos machines virtuelles avant de quitter la salle ! Par contre, ne détruisez pas vos images AMI : indiquez l'identifiant de vos AMIs dans votre rapport de façon à ce que je puisse les tester ultérieurement.
- 

## 1 Déploiement basique

Dans un premier temps nous déploierons l'ensemble de l'application à l'intérieur d'une seule machine virtuelle (la partie PHP ainsi que la base de données).

1. Connectez-vous à AWS avec l'URL et les identifiants qui vous seront donnés. Puis choisissez le service EC2.
2. Créez un "security group" pour cet exercice. N'oubliez pas de lui donner un nom contenant vos initiales : par exemple si vous vous appelez Guillaume Pierre vous pourrez appeler votre security group "GPsecuritygroup". Nous aurons besoin d'ouvrir les ports 22 (ssh), 80 (HTTP) et 3306 (MySQL). Changez la valeur du champ `'VPC'` avec l'option qui vous est proposée à la place.
3. Démarrez une machine virtuelle :
  - Choisissez l'AMI "Ubuntu Server 18.04" parmi les AMIs standard Amazon.
  - A l'étape 3, ouvrez l'onglet "Advanced details" et entrez un petit script shell dans le champ "user data". Par exemple :

```
#!/bin/bash
echo "Bonjour" > /tmp/message.txt
```

- Dans l'étape 5, n'oubliez pas de donner un nom à votre instance (dans la section "Tags") avec vos initiales : par exemple si vous vous appelez Guillaume Pierre vous pourrez appeler votre instance "GPtest1"
  - Dans l'étape 6, indiquez que vous voulez utiliser le security group que vous avez créé préalablement.
  - Créez une nouvelle clé (avec un nom contenant vos initiales) afin de pouvoir vous connecter à votre instance
4. Dans l'interface graphique, attendez que votre instance ait démarré puis connectez-vous à l'instance. Si vous obtenez une erreur "Connection refused" c'est que votre instance n'a pas fini de booter. Attendez quelques secondes puis ré-essayez.

5. Vérifiez que le script que vous avez passé comme “user data” a bien été exécuté lors du démarrage de la machine virtuelle : vous devriez voir un fichier `/tmp/message.txt` avec votre message dedans.
6. Installez WordPress dans votre machine virtuelle : assurez vous que les packages `unzip`, `apache2`, `libapache2-mod-php`, `php-mysql` et `mysql-server` sont bien installés puis téléchargez WordPress et installez-le. Pour vous connecter à votre base MySQL, vous devez exécuter `mysql` depuis le compte `root`. Vous pouvez ensuite créer un utilisateur pour votre base. Les commandes suivantes sont données à titre indicatif :
  - `CREATE USER 'wp'@'localhost' IDENTIFIED BY 'tlcila';`
  - `CREATE DATABASE mysite;`
  - `GRANT ALL PRIVILEGES ON mysite.* TO 'wp'@'localhost';`
  - `flush privileges;`
7. Une fois que WordPress a été installé et testé avec succès, créez une nouvelle image AMI avec votre machine virtuelle pré-installée (attention : cette opération prend plusieurs minutes, et elle a pour effet d’arrêter votre machine virtuelle)
8. Vérifiez que vous pouvez redémarrer votre nouvel AMI et que la machine virtuelle démarre effectivement l’application WordPress.

**Question 1** : quelle peut-être l’utilité de passer un script dans le champ “user data” d’une machine virtuelle ?

## 2 Déploiement sur deux machines virtuelles

Recommencez les mêmes opérations mais cette fois en utilisant deux machines virtuelles (l’une contenant le serveur Web, l’autre contenant la base de données MySQL). A nouveau nous voulons pouvoir re-démarrer WordPress à tout moment en démarrant les deux machines virtuelles à partir d’AMIs pré-configurés. Pour votre propre bien, répondez aux questions ci-dessous **avant** de commencer à coder.

**Question 2** : Les machines virtuelles auront sans doute des adresses IP différentes à chaque fois que vous redémarrerez vos AMIs. Quel problème risque de se produire ? Comment résoudre ce problème ?

**Question 3** : Créer des scripts de démarrage corrects peut demander un certain nombre d’essais avant de parvenir à une version satisfaisante. Il serait particulièrement fastidieux d’écrire un script, de créer un nouvel AMI, de le tester, puis de recommencer plusieurs fois jusqu’à atteindre le but souhaité. Comment allez-vous simplifier cette tâche ?

Vous aurez sans doute besoin de quelques informations supplémentaires :

- Comment le programme `awscli` peut il nous aider ?  
<https://goo.gl/8P3KER> (Allez voir les exemples)

```
sudo apt install awscli
```

- Comment une machine virtuelle peut-elle connaître son propre identifiant ? La commande suivante crée un fichier “`instance-id`” qui contient cet identifiant :

```
wget http://169.254.169.254/latest/meta-data/instance-id
```

- Comment exécuter certaines commandes automatiquement au démarrage d’une machine ? Il est possible d’éditer le fichier “`/etc/rc.local`” et de rajouter vos commandes à la fin du fichier, juste avant “`exit 0`”. Cependant, il est préférable d’écrire des services `systemd`.

## 3 Déploiement élastique

Note : cette partie du TP est optionnelle.

Lorsqu’une application Web est surchargée, le premier goulot d’étranglement est généralement l’étage du serveur Web (une instance de base de données peut servir au moins 4-5 instances HTTP+PHP avant d’observer des problèmes de performance à son tour). Nous allons donc étendre notre installation WordPress afin de rendre la partie HTTP+PHP élastique.

1. Assurez-vous que vous pouvez créer plusieurs instances Apache qui font toutes appel à la même instance MySQL. Vous devez pouvoir rajouter/supprimer des instances Apache sans aucune configuration manuelle.
2. Pour que de nouvelles instances Apache servent à quelque chose, il faut s’assurer de balancer la charge des requêtes entre toutes les instances Apache disponibles. Lisez la documentation sur “Elastic Load Balancing” disponible ici : <https://aws.amazon.com/documentation/elasticloadbalancing/> puis créez un load balancer. Configurez vos instances Apache afin qu’elles s’enregistrent dans le load balancer lors de leur démarrage. Est-il nécessaire de désenregistrer les instances avant de les arrêter?
3. Nous voulons automatiser l’élasticité de notre système. Activez le service CloudWatch afin de monitorer l’activité de vos instances. Puis, activez le service AutoScale afin de déclencher la création d’une nouvelle instance en cas de surcharge. Choisissez bien votre critère de déclenchement de façon à pouvoir simuler facilement une surcharge : par exemple, vous pouvez déclencher la création d’une nouvelle instance lorsque la charge CPU est supérieure à 75%. Vous pourrez simuler une surcharge en exécutant la commande “yes > /dev/null” sur votre instance Apache.

— the end —