

## TP AWS 5 : Utilisation de Boto

### 1. Objectifs

Écrire un programme Python qui déploie des instances et les attache à un *Elastic Load Balancer*.  
Les instance exécutent un serveur HTTP configuré par Cloud-Init.

Pour les quelques questions qui demandent une rédaction, insérez-les en commentaire à la fin de votre code. De la même manière, prenez soin de bien commenter votre code Python.

### 2. Déroulement

1. Préparer le fichier Cloud-Config. Celui-ci doit effectuer les opérations suivantes à l'aide d'actions prédéfinies :

1. Mettre à jour les packets de la distribution
2. Installer les packets « httpd » et « php »
3. Écrire les deux fichiers suivants:

1. /var/www/html/index.php

```
<?php
echo $_SERVER['SERVER_ADDR'];
?>
```

2. /var/www/html/health.php

```
<?php
$seconds = rand(2, 20);
sleep($seconds);
echo $seconds;
header("HTTP/1.1 200 OK");
?>
```

4. Démarrer le service « httpd »
2. Compléter le script Python fournit ici : <http://www.anthony-simonet.fr/teaching/2016/aws/deploy.py>
  1. Complétez les constantes en début de fichier pour utiliser vos credentials
  2. Insérez votre code pour que le programme
    1. Créé N\_INSTANCES instances de type INSTANCE\_TYPE avec l'AMI AMI\_ID dans la région REGION\_NAME, avec le groupe de sécurité SEC\_GROUP et le fichier Cloud-Config préparé précédemment.

2. Crée un *Elastic Load Balancer* qui redirige les requêtes reçues sur son port 80 vers les serveurs HTTP des instances créées plus tôt.
  1. Avec un **HealthCheck** qui interroge la page `health.php` toutes les 20 secondes, avec un **timeout** de 15 secondes, un **UnhealthyThreshold** de 2 et un **HealthyThreshold** de 3.
3. Enfin le programme doit avoir le comportement suivant, en affichant toutes les **20 secondes** l'état des instances du load balancer tant qu'il n'est pas interrompu :

```
$/deploy_blog.py
New instance: i-0b05a20e178311f15
New instance: i-0c62d8393038ac58a
New instance: i-0c114381006a1e087
Created an ELB with DNS: elb-simonet-540342670.eu-west-1.elb.amazonaws.com
ELB state (healthy/unhealthy/total) : 0/3/3
ELB state (healthy/unhealthy/total) : 0/3/3
ELB state (healthy/unhealthy/total) : 0/3/3
ELB state (healthy/unhealthy/total) : 1/2/3
ELB state (healthy/unhealthy/total) : 0/3/3
ELB state (healthy/unhealthy/total) : 2/1/3
ELB state (healthy/unhealthy/total) : 1/2/3
```

3. Depuis un navigateur, accédez plusieurs fois à votre load balancer. Observez le nombre d'instances healthy qu'il comporte et les réponses de la page.
4. Dans `/var/www/html/health.php`, changez la valeur du générateur de nombres aléatoires pour tirer des nombres dans les ensembles **[2, 20]** et **[10, 20]**. À chaque fois, attendez un peu et observez la santé de vos instances. Que constatez-vous ?

## Bonus

1. Placez vos instances afin d'utiliser le maximum des zones de disponibilité de votre région.
2. Remplacez nos deux pages par un site plus complexe (par exemple : le blog de Britney, 16 ans, fan d'équitation). Maintenez toujours vos scripts pour que l'instance soit prête à l'emploi automatiquement après son démarrage.
3. Affichez vos réponses lorsque l'utilisateur quitte votre programme (en tapant Ctrl-C) plutôt que dans les commentaires.