

BLIN Sébastien  
COLLIN Pierre-Henri



# École supérieure d'ingénieurs de Rennes

2ème année  
Parcours Informatique

---

## **TP 2 AL**

Déploiement de l'application RSS-Reader

---

Sous l'encadrement de :  
Temple Paul

## 0.1 Génération des images

## 0.2 Création d'une instance applicative

## 0.3 Le multi-instance

## 0.4 Load-balancing entre les instances

Le load-balancing entre les différentes instances se font à l'aide de nginx, qui écoute sur le port 80 de la machine. Il suffit d'ajouter ces quelques lignes dans le fichier de configuration (sur Fedora : `/etc/nginx/nginx.conf`) :

```
http {
    upstream netflix-rss {
        least_conn;
        server localhost:9092;
        server localhost:9091;
    }

    server {
        listen 8000;
        location / {
                                proxy_pass http://netflix-rss;
        }
    }
}
```

Le `least_conn` servant à envoyer les connexions au serveur ayant reçu le moins de requêtes. Ici, nous ne gérons que 2 instances sur les ports 9091 et 9092. Si nous décidons de gérer plus d'instances, il faudrait donc rajouter des lignes dans `server`.

## 0.5 Des monkeys

Afin de tenter de réagir aux pannes, nous les simulons avec des *Monkeys*. Nous avons donc créé 2 *Monkeys*. Le premier se charge de détruire un des containers, l'autre se charge de rétablir les containers manquants.

### 0.5.1 BADMONKEY

#### Implémentation

Le premier *Monkey* est implémenté à l'aide d'une simple ligne de bash :

```
docker rm $(docker stop $(docker ps | \
awk '/mddletier*|edge*/ {print $13}' | \
shuf -n 1)) # Bad Monkey, kill a random docker
```

Cette ligne se charge de prendre un docker au hasard parmi les mddletier et les edge, le stop et le remove.

## Amélioration

On pourrait améliorer ce *Monkey* en le lançant de manière aléatoire, ainsi qu'améliorer le petit morceau de awk (et éviter le \$13).

### 0.5.2 GOODMONKEY

#### Implémentation

Le premier *Monkey* est implémenté à l'aide d'un petit script python générant les commandes à exécuter :

```
#!/bin/python
import commands

MAX_INSTANCE = 2

listeActive = commands.getstatusoutput("docker ps")[1]
for i in range(1,MAX_INSTANCE+1):
    if 'middletier' + str(i) not in listeActive:
        port = 9191 + i
        print("sudo docker rm middletier{0}".format(i))
        print("sudo docker run --rm --name middletier{0} -h middletier{0} --link
                tomcat:tomcat -p {1}:9191 middletier sh middletier.sh&".format(i, port))
    if 'edge' + str(i) not in listeActive:
        port = 9090 + i
        print("sudo docker rm edge{0}".format(i))
        print("sudo docker run --rm --name edge{0} -h edge{0} --link tomcat:tomcat
                --link middletier{0}:middletier -p {1}:9090 middletier sh rss-edge.sh&".
                format(i, port))
```

Le programme se lançant avec :

```
python goodMonkey.py | bash -x
```

#### Amélioration

L'exécution des commandes n'est pas top et devraient être modifiée, les containers se lancent souvent mal et tous en même temps. De plus on devrait mettre en place un cron pour ne pas à avoir à le lancer soi-même.