Dejvid Muaremi, Romain Silvestri

HEIG-VD

TS – 3ème

Lab 3

Load balancing

10 décembre 2018

# Table des matières

[1. Table des matières 1](#_Toc531445060)

[2. Introduction 2](#_Toc531445061)

[2.1. Remarque 2](#_Toc531445062)

[3. Tâche 1 : Installer les outils 2](#_Toc531445063)

[3.1. Vagrant 2](#_Toc531445064)

[3.2. JMeter 2](#_Toc531445065)

[3.3. Git 2](#_Toc531445066)

[3.4. Manipulations 2](#_Toc531445067)

[3.5. Réponses aux questions 2](#_Toc531445068)

[4. Tâche 2 : Sticky sessions 3](#_Toc531445069)

[4.1. Réponses aux questions 3](#_Toc531445070)

[5. Tâche 3 : Drain mode 4](#_Toc531445071)

[5.1. Manipulations 4](#_Toc531445072)

[5.2. Réponses aux questions 4](#_Toc531445073)

[6. Tâche 4 : Round robin in degraded mode 5](#_Toc531445074)

[6.1. Réponses aux questions 5](#_Toc531445075)

[7. Tâche 5 : Balancing strategies 6](#_Toc531445076)

[7.1. Réponses aux questions 6](#_Toc531445077)

[8. Conclusion 7](#_Toc531445078)

[9. Table des illustrations 8](#_Toc531445079)

[10. Table des références 9](#_Toc531445080)

# Introduction

Pour ce laboratoire, nous allons déployer une application web dans une architecture à deux tiers afin de tester notre scalabilité. Le travail sera réalisé en 5 étapes, dans chacune d’entre-elles nous allons regarder différents points de la configuration d’un proxy, ici HAProxy.

## Remarque

Comme indiqué dans le README.md, nous allons travailler sur la version 1.5 de HAProxy, par conséquent, ce rapport ne tiendra pas compte des autres versions, et il faudrait le réadapter si vous voulez travailler avec la version 1.8, la dernière sortie, ou 1.9, toujours en développement, à ce jour.

# Tâche 1 : Installer les outils

Pour cette première étape, nous allons installer les outils de base nécessaire pour la réalisation de ce laboratoire.

## Vagrant

Multiplateforme et écrit en Ruby, il s’agit d’un logiciel libre et open source, conçu pour créer et configurer des environnements de développement virtuel.

Depuis la version 1.6, Vagrant fourni également un support natif des conteneurs Docker, cela permet de réduire le coût en ressource.

Pour l’installer, il suffit d’aller sur le site officiel et de prendre la dernière version disponible, aucune option particulière n’a été utilisée et la documentation officielle suffit amplement.

## JMeter

Développé par Apache Software Foundation, Apache JMeter est un logiciel libre écrit en Java, par conséquent multiplateforme, qui permet d’effectuer des tests de performance d’applications et de serveurs selon différents protocoles ainsi que d’effectuer des tests fonctionnels.

Le point intéressant de ce logiciel est qu’il permet de simuler un accès multiple sur une application web ce qui sera très utile pour tester notre load balancer.

## Git

Encore un logiciel libre et open source, git est particulièrement utile pour cloner et travailler sur le repo, il simplifie également le travail en équipe. La documentation officielle suffit pour l’installer et l’utiliser, aucune option particulière n’a été mise en place mis à part l’utilisation des clé SSH post-installation.

## Manipulations

Pour cette première partie, il nous est demander de lancer le Vagrant et d’essayer d’utiliser l’application web pour voir comment celle-ci réagit avant de répondre aux questions suivantes.

L’architecture de l’application distribuée est la suivante :

L’adresse de l’application web est la suivante : <http://192.168.42.42> ou, dans certains cas particuliers, <http://localhost:8080>.

## Réponses aux questions

### Question 1

En lançant l’application, on se rend compte que à chaque fois que l’on rafraîchit la page, on change de serveur. On se rend compte qu’il y a un problème étant donné que le serveur nous donne un cookie mais celui-ci n’a pas l’air d’être utilisé pour garder notre session et l’on reçoit à chaque fois une nouvelle comme on peut le voir dans la prochaine capture.

### Question 2

Normalement, si le load balancer fonctionne correctement, on devrait garder la même session et pour se faire, il y a deux possibilités.

1. La session est gardée par le même serveur aussi longtemps que possible, et donc, l’utilisateur ne change pas de serveur même avec des rafraîchissements de page.
2. Le load balancer transmet la session d’un serveur à l’autre, ainsi l’utilisateur garde sa session même lorsqu’il y a un changement de serveur.

### Question 3

Lors d’une première connexion suivie d’un rafraichissement, on peut observer le comportement suivant :

### Question 4

Comme on peut le voir sur l’image suivante, d’après le rapport JMeter, le load balancer fait en sorte que les requêtes soient distribuées équitablement entre les deux serveurs.

### Question 5

En tuant un serveur, le load balancer prend un petit moment pour se rendre compte qu’il n’a plus qu’un serveur à qui envoyé les requêtes.

Comme on ne change plus de serveur, on garde la même session active.

# Tâche 2 : Sticky sessions

bla

## Réponses aux questions

### Question 1

bla

### Question 2

bla

### Question 3

bla

### Question 4

bla

### Question 5

bla

### Question 6

bla

# Tâche 3 : Drain mode

## Manipulations

bla

## Réponses aux questions

### Question 1

bla

### Question 2

bla

### Question 3

bla

### Question 4

bla

### Question 5

bla

### Question 6

bla

### Question 7

bla

# Tâche 4 : Round robin in degraded mode

## Réponses aux questions

### Question 1

bla

### Question 2

bla

### Question 3

bla

### Question 4

bla

### Question 5

bla

### Question 6

bla

# Tâche 5 : Balancing strategies

## Réponses aux questions

bla

### Question 1

bla

### Question 2

bla

### Question 3

bla

# Conclusion

bla

# Table des illustrations

**Aucune entrée de table d'illustration n'a été trouvée.**

# Table des références

<http://www.haproxy.org/>

<http://www.haproxy.org/#docs>

<https://fr.wikipedia.org/wiki/Vagrant>

<https://www.vagrantup.com/>

<https://fr.wikipedia.org/wiki/Apache_JMeter>

<http://jmeter.apache.org/>

<https://git-scm.com/>

<https://help.github.com/articles/connecting-to-github-with-ssh/>