

Université Joseph Ki Zerbo

IBAM

M1 Informatique : Virtualisation et Cloud Computing – Avril 2025

Date limite de remise des travaux : Samedi 31 mai 2025

Enseignant : Dr SOMDA

A rendre à l'adresse : flavien.somda@ujkz.bf

TP : Déploiement d'une Infrastructure Multi-Machines avec Vagrant et Bash

Objectifs :

- Comprendre le fonctionnement de **Vagrant** et des machines virtuelles.
 - Mettre en place une infrastructure multi-nœuds avec Vagrant.
 - Expérimenter la répartition de charge avec un Load Balancer.
 - Configurer une réplication maître-esclave entre deux bases de données MySQL.
 - Ajouter un système de supervision basé sur Prometheus et Grafana.
 - Automatiser le déploiement de cette architecture avec Vagrant.
 - Tester la réplication MySQL et afficher des métriques en temps réel.
-

Scénario

On souhaite mettre en place une infrastructure pour une **application web simple** avec :

- **7 machines virtuelles :**

- `lb` (Load Balancer) : Équilibre le trafic vers `web1` et `web2`.
- `web1` et `web2` (Serveurs Web) : Servent les pages HTML.
- `db-master` (Base de Données Maître) : Contient les données et envoie les mises à jour à `db-slave`.
- `db-slave` (Base de Données Esclave) : Reçoit les mises à jour de `db-master`.
- `monitoring` (Supervision) : Installe **Prometheus** et **Grafana** pour surveiller le cluster.
- `client` (Client) : Utilisé pour tester la connectivité.

- **Chaque machine a une IP statique** dans le réseau privé `192.168.56.0/24`.

- **Provisioning :**

- `setup_lb.sh` : Configure **Nginx** en tant que Load Balancer.
- `setup_web.sh` : Installe **Apache** et affiche un message d'accueil.
- `setup_db_master.sh` : Configure MySQL pour être un **serveur maître**.
- `setup_db_slave.sh` : Configure MySQL pour être un **serveur esclave**.

- `setup_monitoring.sh` : Installe **Prometheus et Grafana**.

Toutes ces machines seront créées avec **Vagrant** et configurées automatiquement à l'aide de **scripts Bash**.

I. Pré-requis

Avant de commencer, assurez-vous d'avoir installé :

- **Vagrant**
 - **VirtualBox**
-

II. Étapes du TP

1. Créez un fichier Vagrantfile décrivant l'infrastructure virtuelle à créer
2. Créez les scripts de provisioning nécessaires
3. Démarrage et tests préliminaires. Illustrer chaque étape par une capture d'écran appropriée
 - a. Vérifier la connectivité depuis le client en utilisant curl
 - b. Vérifier l'accès aux outils depuis leur adresse
 - i. ☐ Load Balancer : `http://adresse_ip_lb`
 - ii. ☐ Prometheus : `http://adresse_ip:9090`
 - iii. ☐ Grafana: http://adresse_ip:3000 (user: admin, password :admin)
4. Test de bout en bout
 - a. Déployé une application web très simple (page très simple avec formulaire pour envoyer des données vers la base de données et les afficher)
 - b. Ajuster la page web pour afficher automatiquement l'adresse IP du serveur web qui répond
 - c. Effectuer plusieurs requêtes vers le Load balancer et démontrer que les requêtes ne sont pas toujours traitées par le même serveur web.
5. Test de résilience
 - a. Eteindre le serveur web1 seul. Envoyer de nouvelles requêtes au load balanceur et vérifier que votre application est toujours disponible. Quelle conclusion en tirez-vous ?

III. Livrables attendu

- a. Un rapport de projet clair et étayé avec des captures d'écran pour justifier chaque étape du projet
- b. Une source complète des scripts Vagrant et Bash utilisés pour créer l'infrastructure. Cette source doit être livrée sous la forme d'un repository Git.

IV. Critères d'évaluation

Critère	Points
Fichier Vagrantfile bien structuré	3 pts
Scripts de provisioning corrects	4 pts
Réplication MySQL fonctionnelle	4 pts
Supervision avec Prometheus et Grafana opérationnelle	4 pts
Tests et validation réalisés	3 pts
Documentation claire du projet	2 pts
Total	20 pts