Rapport du premier sprint de stage

Mise en place de l'environnement de travail et bases DevOps/Linux

Stagiaire : **Sadiki Abdelkarim**

Encadrant : **Hamza bahlaouane**

Entreprise : **Void**

6 février 2026

1 Introduction

Ce rapport présente les travaux réalisés durant le premier sprint de mon stage. L'objectif principal de ce sprint était de mettre en place un environnement de travail solide et de renforcer mes compétences de base en systèmes Linux, Git, administration serveur et bonnes pratiques DevOps.

2 Travaux réalisés

2.1 Mise en place de l'environnement de travail

Configuration complète d'un environnement de développement macOS selon les meilleures pratiques industrielles, incluant le gestionnaire de paquets Homebrew, personnalisation du terminal (iTerm2, Zsh, Oh My Zsh), et outils de développement essentiels (Git,VsCode,Docker).

2.2 Apprentissage des bases Linux

Formation Linux complète via Linux Journey, couvrant les fondamentaux den la command line,Manipulation de texte,Gestion des urilisateurs,Permissions,Processus.

2.3 Versionnement avec Git

Formation Git interactive via Learn Git Branching, maîtrisant les commits, branches, merges, dépôts distants au GitHub, résolution de conflits ....

3 Exercices Pratiques en Labs

3.1 Lab 1 : Configuration de Machine Virtuelle

Déploiement d'une VM Ubuntu avec Vagrant et VMware, établissement de la structure de répertoires de développement, et configuration de base de la VM.

3.2 Lab 2 : Configuration SSH

La configuration de l'accès SSH à la machine virtuelle a été effectuée pour l'utilisateur `alpha` avec le mot de passe `123456`. Ensuite, une paire de clés SSH a été générée afin de mettre en place une authentification sans mot de passe. L'agent SSH a été configuré pour gérer la clé privée, et la commande `ssh-copy-id` a été utilisée pour distribuer de manière sécurisée la clé publique sur la VM. Enfin, l'accès SSH sans mot de passe a été testé et confirmé comme fonctionnel.

3.3 Lab 3 : Analyse des Certificats SSL/TLS

Les erreurs SSL/TLS observées avec la commande `curl` sont dues à des problèmes de certificats. Dans le cas de *expired.badssl.com*, le certificat est expiré, ce qui signifie que sa période de validité est dépassée. Pour *self-signed.badssl.com*, le certificat est auto-signé et n'est pas reconnu par une autorité de certification de confiance. L'analyse avec `openssl s_client` permet de confirmer ces problèmes. Pour corriger ces erreurs, il est nécessaire de renouveler les certificats expirés et d'utiliser des certificats émis par une autorité de certification reconnue, les certificats auto-signés étant réservés aux environnements internes.

3.4 Lab 4 : Automatisation avec Cron

Un cronjob a été mis en place afin de supprimer automatiquement les fichiers âgés de plus de sept jours dans le répertoire `~/temp`. Un script Bash utilisant la commande `find` permet d'identifier et de supprimer ces fichiers. Ce script est planifié pour s'exécuter quotidiennement à minuit à l'aide de `cron`. La suppression des fichiers peut être vérifiée après l'exécution du cronjob. Si le script n'est pas exécutable, le cronjob échoue et aucune action n'est effectuée. Le bon fonctionnement de la tâche planifiée peut être contrôlé à l'aide des journaux système, tels que `/var/log/syslog` ou via `journalctl`. Enfin, l'ajout de logs dans le script permet de capturer d'éventuelles erreurs et de faciliter le débogage.

3.5 Lab 5 : Permissions Utilisateurs & Configuration Sudo

L'utilisateur alpha a besoin d'un accès limité à la commande `ifconfig` afin de vérifier les interfaces réseau, sans disposer d'autres privilèges administrateur. Cet accès est accordé en modifiant le fichier `sudoers` pour autoriser uniquement l'exécution de cette commande. Lors du téléchargement du fichier `bat.zip` vers le répertoire `/opt`, une erreur de type « Permission denied » est rencontrée, car ce répertoire appartient à l'utilisateur root et n'est pas accessible en écriture pour alpha. Ce problème est résolu en ajustant les permissions du dossier afin d'autoriser l'écriture sans utiliser `sudo`. Enfin, l'extraction de l'archive est réalisée apres l'installation de `unzip`.

3.6 Lab 6 : Webserver

L'installation du serveur web Apache sur la VM a été réalisée en se connectant via `ssh` et en passant en utilisateur root avec `sudo su -`. Apache a été installé avec `apt-get install -y apache2` et le service a été démarré, après avoir identifié et tué le processus bloquant le port 80 grâce à `journalctl -xe` et `lsof -i :80`. L'accès au site web était initialement restreint par la configuration du fichier `000-default.conf`, qui bloquait toutes les connexions sauf pour le premier `VirtualHost`. Pour accéder au site depuis la machine hôte, un port forwarding a été configuré dans le `Vagrantfile`, redirigeant le port 80 de la

VM vers un port local. Cette configuration a permis de vérifier dans le navigateur que la page web s'affichait correctement, listant les applications nécessaires à l'installation sur la machine. Par la suite, PHP a été installé sur le serveur et une application Laravel a été hébergée, permettant ainsi de déployer et tester une application web dynamique sur la VM..