

Rapport SAE 1.03 - Initialisation d'un système

I- La consigne

Dans le cadre de cette première SAE, notre devoir était d'installer un système d'exploitation et de configurer un environnement de travail destiné à réaliser des travaux pour différents cours, tels que Git, Python, SQL, Java, Bash et le développement web. Cet environnement devait également permettre l'exécution de plusieurs tâches, telles que :

- Lancer un shell Bash,
- Exécuter du code Python,
- Synchroniser un projet sur GitHub via VSCode,
- Compiler et exécuter un programme Java,
- Utiliser Docker.

Plusieurs options s'offraient à nous pour atteindre cet objectif : l'installation d'un système en dual boot avec Linux, l'utilisation d'une machine virtuelle ou l'utilisation de WSL2 (Windows Subsystem for Linux). J'ai personnellement choisi d'utiliser une machine virtuelle (VM), car mon ordinateur n'est pas suffisamment performant pour un dual boot, ce qui aurait rendu cette option difficile à gérer. Quant à WSL2, je n'ai pas approfondi cette solution. La VM s'est avérée être la solution la plus pratique, notamment parce qu'elle était déjà installée sur les ordinateurs de l'IUT.

II- Présentation des résultats

Une fois la décision prise d'utiliser une machine virtuelle, nous avons commencé à travailler sur la SAE. La première étape consistait à installer une distribution Linux sur la VM. J'ai opté pour Xubuntu, car c'est la distribution utilisée à l'IUT, et je ne connais pas encore d'autres versions de Linux.

Ensuite, nous devons installer les logiciels nécessaires pour nos différents cours, notamment :

- VS Code : pour le développement en HTML, CSS, Python, Java et C
- LibreOffice : pour la rédaction de comptes rendus et la gestion de documents
- DB Browser : pour la gestion des bases de données
- Firefox : pour la navigation Internet

Ce sont les principaux logiciels que j'ai réussi à installer dans le cadre de cette SAE.

De plus, j'ai configuré mon compte GitHub dans VSCode, ce qui me permet de synchroniser mes projets et de récupérer des documents depuis n'importe quel endroit. J'ai également pu exécuter et compiler des programmes en Java, utiliser Docker, ainsi qu'exécuter du code Python.

Une démonstration vidéo montrant l'exécution de ces différentes actions est incluse dans un fichier MKV joint à ce rapport. Je considère que j'ai respecté l'ensemble des consignes demandées dans cette SAE.

III- Explication des démarches

1. Installation de la VM

J'ai commencé par télécharger l'ISO de Xubuntu que j'ai ensuite installé dans ma machine virtuelle (VM). L'exécution de l'ISO a suffi pour configurer correctement la VM.

2. Installation et exécution de Docker

Pour tester si Docker était installé, j'ai exécuté la commande (`docker run hello-world`). Cela a produit une erreur, car Docker n'était pas encore installé. J'ai donc utilisé la commande (`sudo snap install docker`) pour l'installer.

Une fois l'installation terminée, j'ai réessayé la commande (`docker run hello-world`), mais une erreur de permission est apparue. J'ai résolu ce problème en exécutant Docker en tant que « super administrateur » avec la commande (`sudo docker run hello-world`), ce qui a permis de faire fonctionner correctement Docker.

3. Exécution de Python

Pour exécuter Python, j'ai simplement tapé la commande (`python3`) dans un terminal, ce qui a ouvert l'interpréteur Python dans le terminal. Ensuite, j'ai tapé la commande (`print("Hello World !")`) pour afficher "Hello World !" dans le terminal.

4. Compilation et exécution de Java

Pour Java, j'ai d'abord créé un fichier nommé `Executable.java`. Ensuite, dans le terminal, j'ai tenté de le compiler avec la commande (`javac Executable.java`), mais une erreur s'est produite, car `javac` n'était pas installé. J'ai donc dû installer le compilateur Java avec la commande (`sudo apt install openjdk-17-jdk-headless`). Une fois installé, la commande (`javac Executable.java`) n'a pas produit de sortie visible, mais a bien compilé le fichier en arrière-plan. Pour terminer, j'ai exécuté le fichier Java avec la commande (`java Executable`), qui a renvoyé "Hello World".

5. Installation des différentes applications

- **LibreOffice** : Pour installer LibreOffice, j'ai utilisé la commande suivante (`sudo apt install libreoffice`).
- **VS Code** : Pour installer Visual Studio Code, j'ai utilisé la commande suivante (`sudo snap install --classic code`).
- **DB Browser for SQLite** : Pour installer DB Browser, j'ai utilisé la commande suivante (`sudo apt install sqlitedbbrowser`).
- **Firefox** : Firefox est souvent installé par défaut sur les distributions Linux, mais si besoin, la commande pour l'installer est (`sudo apt install firefox`).

IV- Démonstration de compétence : Administrer des systèmes informatiques communicants complexes

Dans le cadre de cette SAE, j'ai mis en œuvre la compétence « Administrer des systèmes informatiques communicants complexes » de la configuration de ma machine virtuelle sous Xubuntu, de l'installation d'outils de développement (comme Docker, VS Code, Java), et de la gestion des incidents qui en ont découlé. Voici un exemple détaillé illustrant cette compétence.

Acquisition des apprentissages critiques :

1. Identifier les différents composants (matériels et logiciels) d'un système numérique

Lorsque j'ai opté pour l'utilisation d'une machine virtuelle (VM) pour exécuter un système Xubuntu, j'ai dû comprendre comment fonctionnait la mémoire vive pour donner assez de mémoire à la VM sans lui en attribuer trop. J'ai également appris ce qu'est un CPU, ce que je ne savais pas auparavant, le tout pour assurer une bonne performance de la VM.

2. Mettre en place et configurer un système d'exploitation et des outils de développement

J'ai téléchargé Xubuntu avec un ISO et mis en place les logiciels de développement dont VS Code, Python, Java, Docker et DB Browser for SQLite. Chaque installation nécessitait une configuration spécifique pour s'assurer que tous les outils fonctionnaient correctement. Par exemple, une fois Docker installé, j'ai eu un souci de permissions que j'ai résolu en exécutant les commandes avec les droits administratifs (sudo).

3. Mise en œuvre des mesures correctives et gestion des incidents

Lors de l'installation de Docker, j'ai rencontré des problèmes d'exécution liés aux droits d'utilisateur. La commande (`docker run hello-world`) échouait en raison d'un manque de permissions. Pour résoudre cet incident, j'ai exécuté Docker avec les droits d'administrateur (`sudo docker run hello-world`). Cette action a permis de garantir le bon fonctionnement de Docker.

4. Assurer la pérennité des données et des logiciels

Pour garantir la pérennité des données, j'ai utilisé Git pour synchroniser et versionner mes projets sur GitHub. Cela m'a permis de sauvegarder toutes mes configurations et mes projets, évitant ainsi la perte de données en cas de problème matériel ou de corruption logicielle.

V- Conclusion

Sur le plan personnel, ce travail m'a permis de mieux comprendre Linux ainsi que son environnement, tout en approfondissant mes connaissances sur les outils de développement. J'ai particulièrement apprécié l'installation des applications et l'exécution de code dans plusieurs langages. Cette SAE m'a également montré l'importance de faire des recherches et me renseigner face aux erreurs commises en Bash.

Je regrette de ne pas avoir eu le temps d'explorer d'autres solutions comme WSL2 ou encore le dual boot, mais je me sens désormais capable et je pense tenter un dual boot, voire l'installation complète d'un Linux sur un vieux PC quand cela sera possible.