

I-Présentation de la SAE1.03

Pour cette première SAE, il nous est demandé de mettre en place un environnement de travail propice au développement de nos projets au sein de l'IUT mais aussi pour nos projets futurs en dehors de l'IUT. Cette SAE a pour but de nous apprendre à rechercher quel type d'environnement répond à nos besoins, et correspond à notre matériel. Il a fallu chercher seul ces informations et réussir à sélectionner les méthodes les plus récentes.

II-Présentation de mes résultats

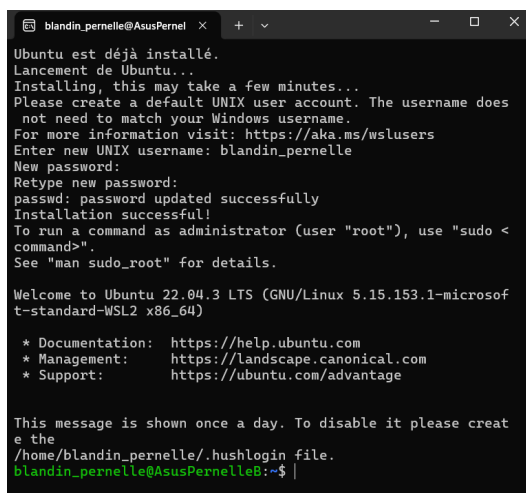
C'est dans cette optique que j'ai opté pour Windows Sous Linux (wsl) et plus précisément la deuxième version (wsl2). Contrairement à la machine virtuelle, la wsl2 va me permettre de pouvoir facilement changer d'un développement sous Windows à un environnement de travail Linux. Je pourrais utiliser des applications et logiciels installés avec Windows, avec des lignes de commandes Bash Linux sans avoir à les modifier.

J'ai réussi à installer une version de Ubuntu et à me créer un compte, j'ai aussi réussi à installer VSCode, GitDesktop et Docker. J'ai installé la version 3.9.20 de python pour pouvoir avoir le dossier "distutils" et pouvoir mettre en place un environnement virtuel (.venv). J'ai via cette SAE appris à faire des push/pull et des synchronisations correctes entre VSCode et GitHub.

III-Explications sur ma démarche pour l'installation

J'ai commencé par me documenter sur les différentes approches suggérées, puisque je n'en connaissais aucune. A l'issue de mes recherches, je suis arrivée à la conclusion qu'il me fallait choisir entre une VM et WSL2. J'ai préféré la WSL2 car elle était recommandée pour les utilisateurs nouveaux sur Linux, et demandait moins de performances et de mémoire à faire tourner sur son ordinateur qu'une VM. De plus j'ai Windows et le site de Microsoft expliquait très bien chacune des étapes de l'installation.

Tout d'abord, j'ai ouvert un terminal Windows en mode administrateur et ai écrit la ligne de commande `wsl --install` puis j'ai redémarré mon ordinateur. Ainsi la distribution Ubuntu de Linux était installée. J'ai configuré mes informations utilisateurs Linux, ai modifié certains paramètres de mon terminal et enfin j'ai mis à jour tous les packages avec les commandes `sudo apt update` et `sudo apt upgrade`.



```
blandin_pernelle@AsusPernelleB:~$ wsl --install
Ubuntu est déjà installé.
Lancement de Ubuntu...
Installing, this may take a few minutes...
Please create a default UNIX user account. The username does
not need to match your Windows username.
For more information visit: https://aka.ms/wslusers
Enter new UNIX username: blandin_pernelle
New password:
Retype new password:
passwd: password updated successfully
Installation successful!
To run a command as administrator (user "root"), use "sudo <
command>".
See "man sudo_root" for details.

Welcome to Ubuntu 22.04.3 LTS (GNU/Linux 5.15.153.1-microsof
t-standard-WSL2 x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

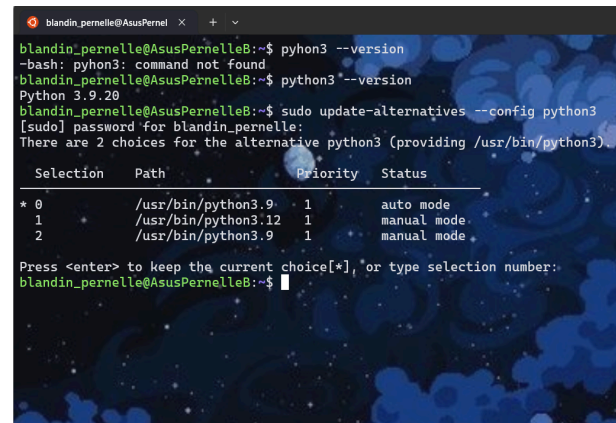
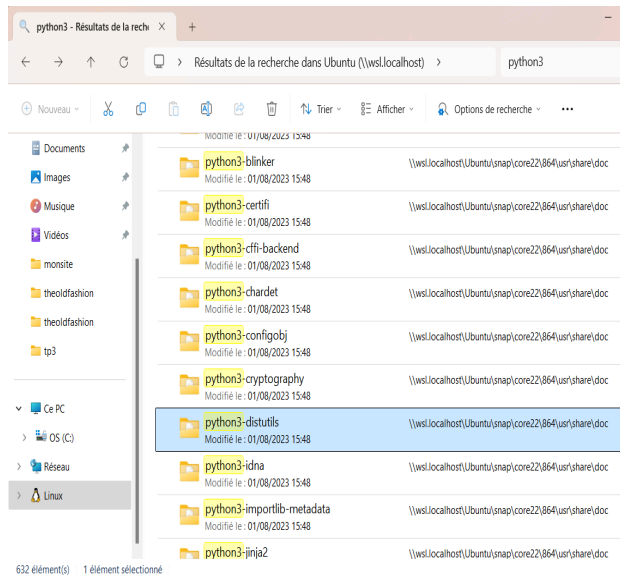
This message is shown once a day. To disable it please creat
e the
/home/blandin_pernelle/.hushlogin file.
blandin_pernelle@AsusPernelleB:~$
```

J'ai ensuite installé Python. J'ai cependant rencontré des difficultés lors de cette étape. J'ai voulu directement installer la version de python la plus récente (la 3.12.6). Seulement cela m'a bloqué puisque dans les versions les plus récentes de Python, le répertoire distutils n'existe pas. Or c'est via ce répertoire que se faisait la troisième étape, qui est de créer l'environnement virtuel pour le développement d'un projet sur VSCode. De plus, il y avait un conflit de version de Python entre celle que j'avais installée manuellement (qui m'avait aussi ajoutée la 3.10 et la 3.11 en plus de la 3.12) et celle qui était préinstallée dans la distribution Ubuntu de ma WSL2.

Pour résoudre ce problème, j'ai dû supprimer le répertoire que j'avais créé pour installer manuellement la version la plus récente de Python avec la commande :

```
sudo rm /etc/apt/sources.list.d/deadsnakes-ubuntu-ppa-jammy.list .
```

Cependant le répertoire distutils restait introuvable et j'ai dû le mettre manuellement via `sudo apt install python3-distutils`. J'ai ensuite réinstallé le pip3 de python3. La ligne de commande du tutoriel (`pip install virtualenv`) n'était plus non plus à jour. La bonne ligne de commande était `sudo apt install python3-virtualenv`



J'ai sélectionné la version de python à choisir par défaut (auto mode) en rentrant le numéro de ligne correspondant à la version que l'on souhaite utiliser (ici la 3.9).

VSCode était déjà téléchargé via le Microsoft Store avant de commencer la SAE. J'ai donc simplement installé une extension pour la WSL. Pour lancer VSCode sous Linux, je peux soit écrire dans le terminal Ubuntu `code` . ou bien ouvrir VSCode sous Windows, puis cliquer sur les "><" qui indiquent "Open a remote window" et sélectionner "Connect to WSL". Pour quitter VSCode sous Linux, "File" ou "><" puis "Close remote connection". Ensuite, une fois connecté correctement à ma WSL2 sur VSCode, j'ai installé les extensions Microsoft pour Python et Java.

Pour me connecter à Github, j'ai vérifié s'il était déjà installé sur ma WSL2 avec `git --version`. Git était bel et bien déjà installé, j'ai donc juste eu à renseigner mes informations dans le terminal avec `git config --global user.name "Mon identifiant"` et `git config --global user.email "monemail@domaine.com"` en complétant correctement les guillemets.

Pour finir mes installations, j'ai téléchargé l'application Docker Desktop pour Windows x64, que j'ai ensuite configuré les réglages pour pouvoir m'en servir avec ma WSL2. Dans "paramètres", puis "resources" et "WSL integration", j'ai sélectionné la bonne distribution, pour moi "Ubuntu".

IV-Compétences travaillées

Dans le cadre de cette SAE, nous devons travailler sur la compétence "Administrer des systèmes informatiques communicants complexes". Je pense que mon travail répond aux attentes, mais est loin de les dépasser. J'ai installé chaque chose demandée, mais n'ai pas cherché à aller plus loin, en faisant des scripts par exemple. J'ai fait ce choix, car, découvrant absolument tous ces logiciels hormis VSCode, je voulais prendre le temps de bien comprendre chaque chose que je faisais et

pourquoi il fallait le faire de telle ou plutôt de telle autre manière. Ainsi, cela m’a pris beaucoup de temps de me documenter correctement et de comparer tout d’abord les solutions proposées, puis les types de WSL et enfin les différentes méthodes d’installation et de configuration qui permettent d’exécuter du Bash, du Python, du Java...

Comparaison des fonctionnalités

[Agrandir le tableau](#)

Fonctionnalité	WSL 1	WSL 2
Intégration entre Windows et Linux	✓	✓
Temps de démarrage courts	✓	✓
Faible empreinte des ressources comparé aux machines virtuelles traditionnelles	✓	✓
Exécutions avec les versions actuelles de VMware et VirtualBox	✓	✗
Machine virtuelle managée	✗	✓
Noyau Linux complet	✗	✓
Compatibilité complète des appels système	✗	✓
Performances dans les systèmes de fichiers du système d’exploitation	✓	✗
prise en charge systemd	✗	✓
Compatible avec IPv6	✓	✓

J’ai lu tout un article sur les différences détaillées entre les 2 types de WSL. Ici, le tableau résume ces différences et dresse un bilan qui permet d’avoir une vue d’ensemble. Cela m’a permis d’être certaine de choisir la version qui correspondait le plus à ma machine et à mes besoins. Je pense que ces recherches m’ont permis d’identifier les différents composants (matériels et logiciels) d’un système numérique, de respecter les contraintes de performances de mon

ordinateur et d’assurer la pérennité de mes installations (que je peux utiliser aussi bien sous Windows pour des projets personnels ou bien sous Linux pour des projets demandés par l’IUT).

J’aurais aimé réaliser un travail plus poussé, et développer plus chacun des éléments de la compétence travaillée pendant la SAE. De plus, j’ai eu beaucoup de mal à corriger le problème des versions de python, et les ressources que j’ai trouvées sur Internet ne m’ont pas permis de m’en sortir. Sans l’intervention d’un des enseignants, je n’aurais probablement pas réussi à faire fonctionner mon installation et j’aurais été bloquée. Il était fortement déconseillé de désinstaller ou supprimer certains fichiers ou la WSL en entier, et j’aurais dû faire une VM à la place. Cela m’aurait fait perdre un temps considérable, car il ne restait déjà que peu de créneaux assignés à cette SAE à ce moment-là, et j’aurais dû me documenter sur la VM me prenant encore plus de temps. Ainsi j’aimerais développer l’élément “mettre en œuvre les mesures correctives adaptées à la nature des incidents identifiés”.

V-Conclusion

Dans l’ensemble je suis assez satisfaite puisque ces installations vont me permettre de pouvoir travailler de chez moi avec mon ordinateur personnel, ce qui me donnera plus de liberté dans mon travail et pour mes devoirs (avant que je ne commence la configuration avec la SAE, je devais finir mes devoirs sur les machines de l’IUT). La plus grande difficulté à laquelle j’ai été confrontée (autre que l’installation de la bonne version de python détaillée plus tôt) a été le temps que toutes ces recherches m’ont pris. Je découvrais absolument tout, et il n’y avait que le minimum de consignes et de recommandation pour nous guider. Si je devais comptabiliser le nombre d’heures que la SAE m’a prise à faire (sur et hors créneaux de SAE, et en comptant le temps pour la vidéo et le rapport) j’en suis à une quinzaine d’heure de travail contre les neuf heures indiquées sur CELENE. Je ne regrette cependant pas le retard que j’ai pu prendre à cause de cela sur d’autres travaux demandés, car j’ai bien compris chacune des étapes de la configuration de mon système et je pense que cela me sera utile pour plus tard.