



Informatique

iut  
de BORDEAUX

## **SAE 1.03 - Installation de poste**

Introduction systèmes

# Rapport installation de poste

27/11/2023

## Introduction

En qualité d'administrateurs système au sein de BUTGAMES, notre mission consiste à mettre en place l'environnement de travail pour une nouvelle équipe de développeurs qui va travailler sur un jeu codé en Haskell.

Face à la complexité souvent associée à la configuration d'un environnement de développement Haskell, nous avons pris l'initiative de concevoir une machine virtuelle dédiée, pré-configurée et prête à l'emploi. Cette approche vise à éliminer les obstacles liés à la configuration individuelle des environnements de développement, favorisant ainsi l'efficacité et la cohérence au sein de la future équipe.

Le développement en Haskell requiert un environnement spécifique, parfois ardu à mettre en place. Afin de garantir une homogénéité dans le processus de développement et d'accélérer la productivité de l'équipe, ce rapport expose les choix stratégiques effectués lors de la configuration de la machine virtuelle. Il a pour but de servir de guide complet pour l'équipe de développement, offrant une compréhension approfondie des décisions prises concernant la configuration matérielle, les choix logiciels, l'architecture logicielle, et les étapes nécessaires à la création d'un nouveau projet Haskell.

Ce rapport est structuré de manière à présenter successivement la justification des choix matériels et logiciels, une illustration détaillée de l'architecture logicielle, les détails spécifiques de la configuration de la machine virtuelle et du système d'exploitation, et enfin, une notice d'utilisation guidant les développeurs dans la création de nouveaux projets Haskell.

# Sommaire

<b>Introduction.....</b>	<b>.....</b>
<b>Sommaire.....</b>	<b>.....</b>
<b>I. Dossier d'étude et de choix des solutions.....</b>	<b>1</b>
1.1. Choix de la distribution Linux.....	1
1.2. Définition de la machine virtuelle.....	1
1.3 Schéma de l'architecture logicielle.....	2
<b>II. Mise en place de la machine virtuelle.....</b>	<b>3</b>
2.1. Configuration de la machine virtuelle.....	3
2.2. Installation et configuration de la distribution linux.....	4
<b>III. Notice d'utilisation.....</b>	<b>6</b>
3.1. Création d'un projet haskell.....	6
3.2. Test de compilation.....	7
3.3. Portfolio.....	7
<b>Conclusion.....</b>	<b>8</b>

# I. Dossier d'étude et de choix des solutions

Nous allons en premier lieu réaliser une étude visant à comparer plusieurs distributions Linux, nous permettant de choisir au mieux la distribution la plus facile à prendre en main pour un utilisateur non expérimenté, mais également la distribution qui permettra de répondre aux contraintes de ressources limitées. Nous évaluerons également la configuration matérielle optimale pour répondre au besoin pour la création d'un jeu vidéo en Haskell.

## 1.1. Choix de la distribution Linux

Dans le cadre de la mise en place d'une distribution Linux légère, nous avons sélectionné parmi les distributions existantes, que nous avons ensuite comparées afin de déterminer laquelle serait la plus à même de répondre aux attentes de légèreté, de place, de performance et de fiabilité.

Parmi la liste, nous retrouvons Linux Lite, Debian et Lubuntu. Nous allons détailler pourquoi Lubuntu nous semble être la meilleure solution et pourquoi les autres distributions ne sont pas appropriées.

Nous avons dans un premier temps installé la distribution Linux Lite. Celle-ci est plus récente que Lubuntu et moins célèbre, ce qui a un impact sur la quantité de ressources disponibles en ligne. Cependant, ce n'est pas le seul critère de différenciation. Avec ses 8 Go d'espace disque, Linux Lite est plus gourmand, ce qui augmente la charge totale sur l'espace de stockage.

Debian est plus ancien que Lubuntu, et bien qu'il possède une grande communauté, nous nous sommes rendu compte que l'accessibilité de la distribution vis-à-vis des débutants est compliquée. En effet, l'apprentissage d'une distribution basée sur Ubuntu reste plus accessible que la distribution Debian. Debian a également besoin d'un espace disque de 10 Go, il est donc bien plus lourd et ne rentre pas en compte pour l'aspect de légèreté souhaité.

Nous avons donc opté pour la distribution Lubuntu sur notre machine virtuelle en raison de sa légèreté et parce qu'elle bénéficie de tous les composants d'une distribution Ubuntu. Elle possède également une interface légère, basée sur LXQt. Tout comme Ubuntu, Lubuntu est facile à prendre en main, offrant ainsi une utilisation et une configuration aisées. Les composants matériels nécessaires au fonctionnement de Lubuntu correspondent parfaitement à la notion de distribution légère. De plus, la stabilité éprouvée de cette distribution, due à sa longévité, constitue un aspect non négligeable.

## 1.2. Définition de la machine virtuelle

Maintenant que nous avons choisi la distribution, nous allons expliquer quelles ressources il est, selon nous, nécessaire d'allouer à la machine virtuelle en CPU, RAM et espace disque.

En termes de CPU, nous avons décidé d'allouer à la machine 3 cœurs afin de pouvoir réaliser les tâches de calcul nécessaires à la compilation du code Haskell en des temps

acceptables. Pour la mémoire vive, une base de 4 Go de RAM sera nécessaire, même si sur Ubuntu, la configuration minimale de VSCode et le langage Haskell fonctionnent avec 1 Go de RAM. Avoir 4 Go de RAM permettra de ne pas avoir à modifier la configuration de la machine virtuelle avant que le projet ne prenne de l'ampleur. En effet, le développement d'un jeu vidéo peut facilement consommer beaucoup de mémoire vive ainsi que d'espace de stockage. Concernant l'espace disque alloué à la machine, une base de 20 Go sera donnée au début. Ils permettront largement de prendre en charge l'installation de Ubuntu (~5 Go) ainsi que VSCode et Haskell (~1 à 2 Go). Le reste de l'espace disque servira aux potentiels Assets" pour le jeu (images et/ou modèles 3D).

Cependant, après une première installation de la machine ainsi que de l'environnement, nous avons constaté que l'espace disque de 20 Go n'était pas suffisant en raison de l'installation des outils permettant le fonctionnement d'un projet Haskell, comprenant ainsi les programmes GHC et GHCup qui prennent à eux deux 2,5 Go de stockage et doivent être installés pour chaque utilisateur utilisant Haskell. Nous en avons donc conclu qu'il serait nécessaire de doubler l'espace alloué à la machine (40 Go) afin de laisser de l'espace de stockage disponible pour les utilisateurs.

### 1.3 Schéma de l'architecture logicielle

Interaction des utilisateurs avec la machine virtuelle ainsi que ses composants:

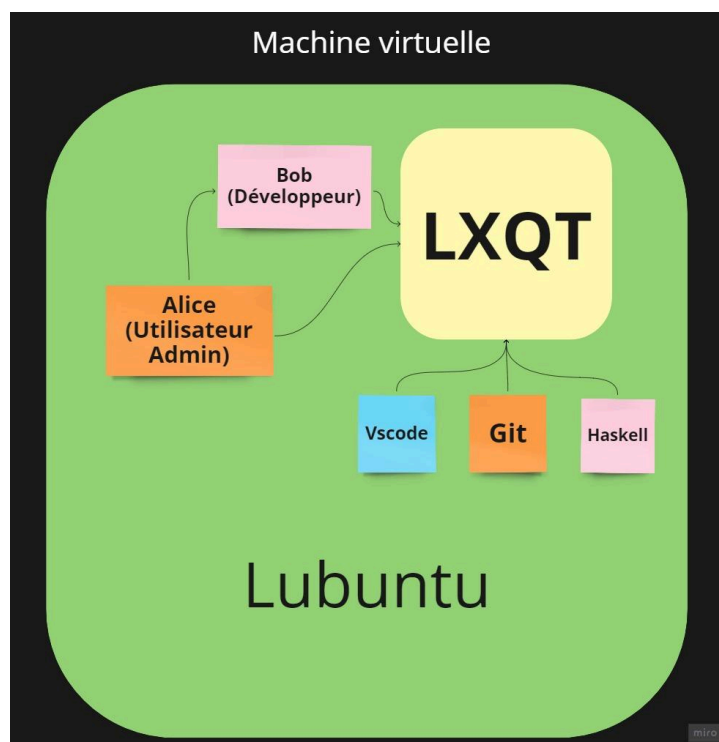


Figure 1 : Schéma de l'architecture logiciel de la machine virtuelle

Ce schéma illustre la configuration de la machine virtuelle et de ses éléments. Tout d'abord, Ubuntu offre aux deux utilisateurs (Alice et Bob) une interface légère (LXQT) qui permet

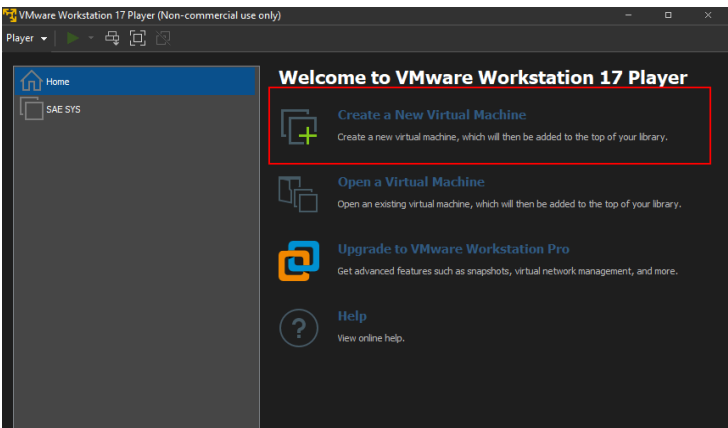
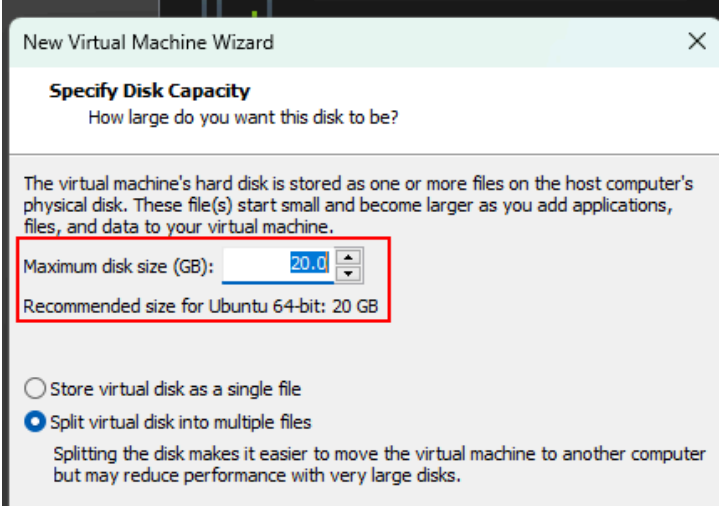
d'utiliser la machine sur des ordinateurs ayant des ressources limitées. De plus, Ubuntu inclut les logiciels (VSCode et Git) et langages (Haskell) utilisés par les utilisateurs, qui seront accessibles à ces derniers via une interface graphique facilitée par LXQT.

## II. Mise en place de la machine virtuelle

Nous allons à présent installer et configurer la machine virtuelle. Cela passera par la définition des diverses ressources allouées à la machine ainsi que par la sélection de la distribution choisie via l'interface du logiciel VMware. Ensuite, nous procéderons à l'installation et à la configuration de la distribution choisie.

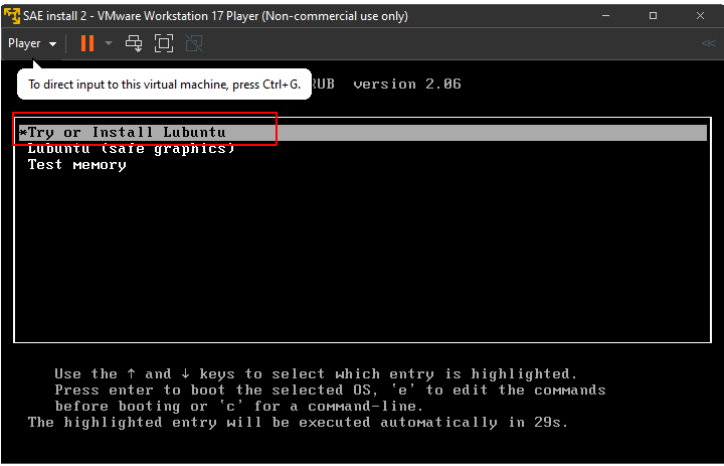
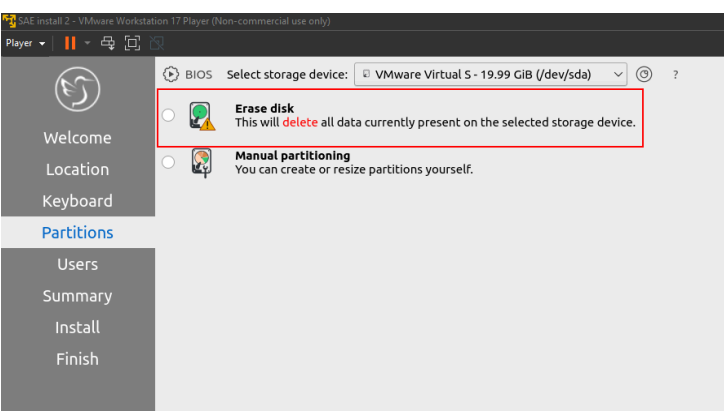
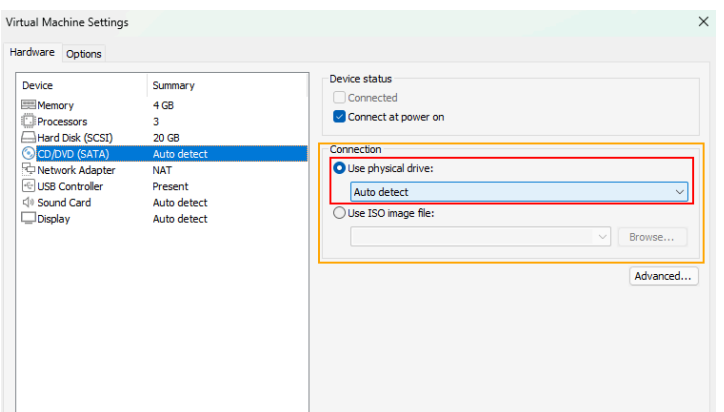
### 2.1. Configuration de la machine virtuelle

Nous commençons par la création de la machine virtuelle depuis le logiciel VMware.

	<p><b>Étape 1 :</b> Ouverture et création d'une nouvelle machine. (Create a New Virtual Machine)</p> <p><b>Étape 2 :</b> Récupérer le fichier ISO téléchargé sur le site web de Ubuntu et l'ajouter dans le champ prévu dans l'interface.</p> <p><b>Étape 3 :</b> Définir la distribution utilisée. Comme Ubuntu est un dérivé d'Ubuntu, on choisit Ubuntu 64bit.</p>
	<p><b>Étape 4 :</b> Allouer un espace disque à la machine virtuelle. Ici 40Go (Espace modifiable à tout moment à l'avenir)</p> <p><b>Étape 5 :</b> Paramétrage des composants alloué à la machine (CPU). On passe le CPU à 3 coeurs.</p>

## 2.2. Installation et configuration de la distribution linux

Nous allons à présent réaliser l'installation et la configuration de la distribution Ubuntu.

	<p><b>Étape 1 :</b> Démarrer la machine virtuelle, choisir “Try or Install Lubuntu ”</p> <p><b>Étape 2 :</b> Démarrer l’installateur une fois Lubuntu démarré depuis le fichier ISO.</p> <p><b>Étape 3 :</b> Définir la langue souhaitée, ici l’Anglais.</p> <p><b>Étape 4 :</b> Définir le type de clavier, ici Français : Azerty</p>
	<p><b>Étape 5 :</b> On formate le disque dur précédemment et on le met en mode “Swap to File”.</p> <p><b>Étape 6 :</b> On renseigne l'utilisateur (Admin) par défaut “alice”.</p> <p><b>Étape 7 :</b> On lance l'installation et on patiente jusqu'à qu'elle soit terminée.</p> <p><b>Étape 8 :</b> éteindre la machine virtuelle.</p>
	<p><b>Étape 9 :</b> Redémarrer VMware sans lancer la machine, ouvrir l’interface de configuration Hardware et retirer le fichier ISO de lecteur CD.</p> <p><b>Étape 10 :</b> Démarrer la machine (Lubuntu devrait s’ouvrir.)</p>

```
alice@alice: ~
File Actions Edit View Help
alice@alice: ~
alice@alice:~$ sudo snap install --classic code
code 1a5daa3a from Visual Studio Code (vscode-) installed
alice@alice:~$
```

**Étape 11 :** Ouvrir un terminal et vérifier que git est installé par défaut en écrivant “git”. S’il n’y a pas d’erreur alors git est présent.

**Étape 12 :** Installer Vscode avec la commande “sudo snap install --classic code”

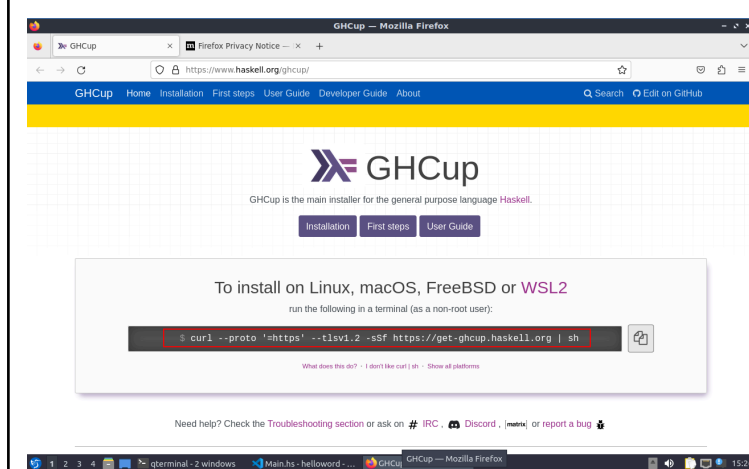
**Étape 13:** Ouvrir vscode en écrivant “code” ou via l’interface graphique. Une fois ouvert, installer l’extension suivante “Haskell”.

```
alice@alice: ~
File Actions Edit View Help
alice@alice: ~
alice@alice:~$ sudo apt install curl
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcurl4
The following NEW packages will be installed:
  curl
The following packages will be upgraded:
  libcurl4
1 to upgrade, 1 to newly install, 0 to remove and 94 not to upgrade.
Need to get 484 kB of archives.
After this operation, 454 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libcurl4
```

**Étape 14:** Retourner dans le terminal puis installer curl via APT avec la commande “sudo apt install curl”

**Étape 15:** Via curl, installer Haskell Tool Stack avec la commande “curl -sSL <https://get.haskellstack.org> | sh”

**Étape 16:** Vérifier la présence des librairies suivantes “sudo apt-get install -y g++ gcc libc6-dev libffi-dev libgmp-dev make zlib1g-dev”



**Étape 17 :** Via curl installer GHCUP avec la commande “curl --proto='https' --tlsv1.2 -sSf https://get-ghcup.haskell.org | sh”

**Étape 18:** Pour éviter le retéléchargement de GHCUP lors de la compilation d’un code haskell faire la commande suivante : “stack config set system-ghc true --global” (Fonctionne uniquement pour l’utilisateur qui saisie la commande)

```
alice@alice:~$ sudo adduser bob
[sudo] password for alice:
Adding user 'bob' ...
Adding new group 'bob' (1001) ...
Adding new user 'bob' (1001) with group 'bob' ...
Creating home directory '/home/bob' ...
Copying files from '/etc/skel' ...
New password:
Retype new password:
passwd: password updated successfully
Changing the user information for bob
Enter the new value, or press ENTER for the default
  Full Name []: Bob
    Room Number []:
    Work Phone []: 06 66 66 66 66
    Home Phone []:
      Other []:
Is the information correct? [Y/n] Y
alice@alice:~$
```

**Étape 19 :** Créer un utilisateur “bob” qui sera le développeur via la commande “sudo adduser bob”

**Étape 20:** Donner les droits d’installation à bob en local via la commande “sudo usermod -aG sudo bob”





## 3.2. Test de compilation

L'image ci-dessous nous permet de prouver que la compilation du code Haskell présent dans le fichier hello.sh retourne bien un 'hello world', comme défini dans le fichier précédemment cité.

```
bob@alice ~/hello.sae4.but1.iut (haskell)$ ./test.sh
Hello, World!
```

Figure 2: Exécution du test de compilation

## 3.3. Portfolio

Nous avons mis en place une machine virtuelle que nous avons configurée et installée afin de permettre à une équipe de développeurs de réaliser un jeu vidéo en Haskell. Cela nous a permis d'apprendre à créer une machine virtuelle, à la configurer et à l'installer. Ce projet nous a également permis de découvrir différentes distributions liées au système d'exploitation Linux et leurs spécificités.

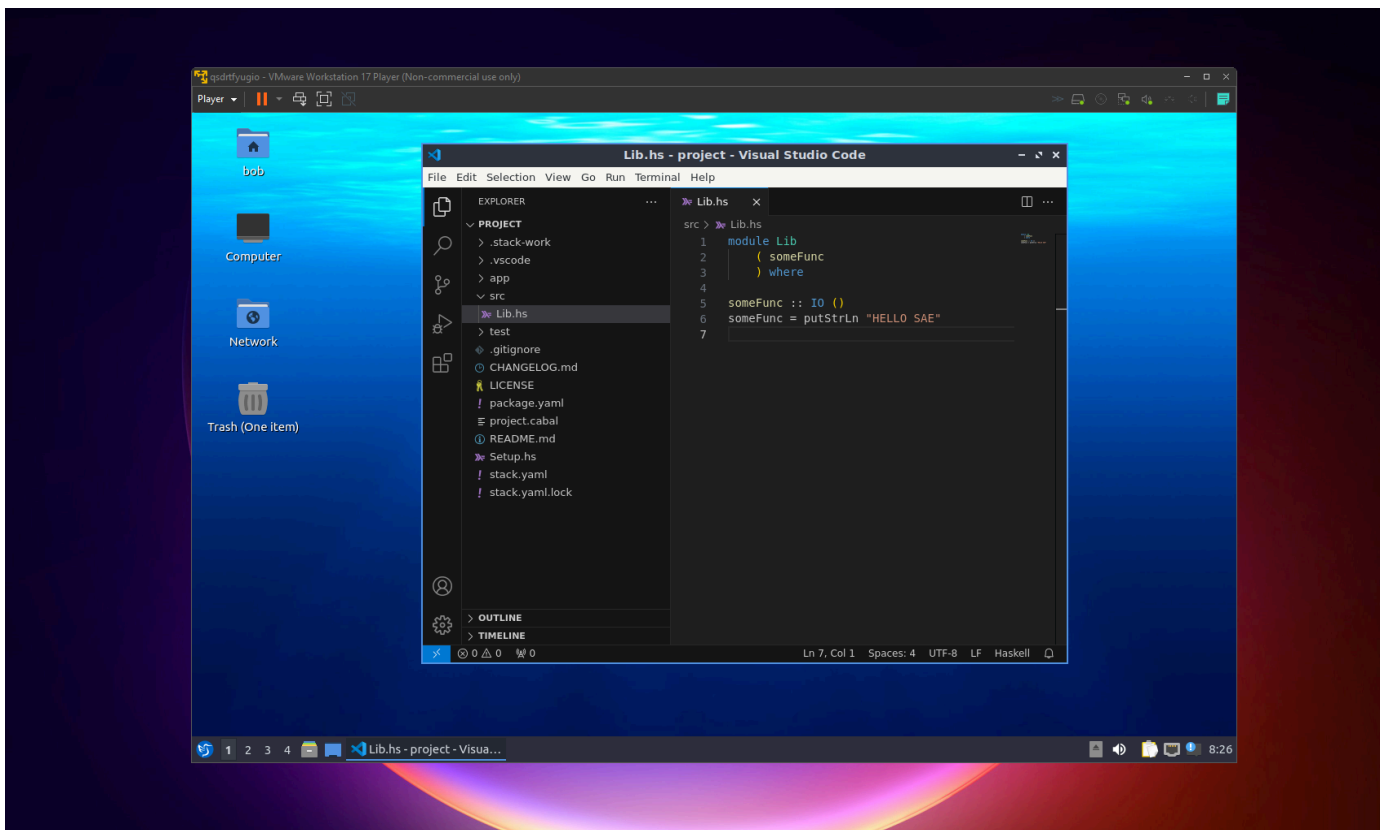


Figure 3 : Image de la machine virtuelle

## Conclusion

Pour répondre aux besoins de la nouvelle équipe de développeurs Haskell, nous avons décidé d'installer une machine virtuelle en choisissant la distribution Lubuntu pour sa fiabilité, sa simplicité et sa légèreté. Afin de permettre aux développeurs de comprendre facilement leur environnement de travail, nous avons décidé de représenter l'architecture logicielle sous forme d'un schéma, et nous avons également créé une notice d'utilisation. Notre machine virtuelle est fonctionnelle, elle répond aux besoins de l'équipe, mais certaines améliorations sont encore possibles.

En accomplissant cette mission, nous avons réussi à renforcer nos compétences dans l'installation de systèmes d'exploitation Linux, l'utilisation des lignes de commande Bash, la création et la configuration de machines virtuelles, ainsi que l'établissement d'environnements de développement efficaces. Cependant, des défis subsistent, notamment en ce qui concerne le fonctionnement du débogueur, une lacune qu'il est impératif de combler. Trouver une solution viable pour permettre le bon fonctionnement du débogueur demeure une priorité.

Un autre point d'amélioration réside dans la duplication inutile de l'installation des programmes GHC et GHCup à chaque fois qu'un nouvel utilisateur compile du code Haskell. Pour remédier à cela, il est impératif d'installer ces programmes de manière globale et de définir un chemin d'accès dans la variable PATH de Bash, évitant ainsi les installations répétées et optimisant l'utilisation des ressources.

Afin d'avoir toutes les clés pour mettre en œuvre au mieux la machine virtuelle, une formation sur les composants d'un ordinateur ainsi que l'interaction des différents composants entre eux aurait simplifié la démarche d'allocation des ressources lors de la création de la machine. Il aurait également pu être pertinent d'avoir un cours sur les différentes distributions Linux et leurs spécificités.