

## Compte rendu :

### Dossier d'étude et de choix des solutions :

Tout d'abord, avec mon camarade, nous nous sommes posé la question suivante : quel système d'exploitation sous Linux choisir ? Entre Debian et Ubuntu, tout en passant par Linux Mint ou encore Ark Linux, tous ces systèmes d'exploitation ont chacun leurs propres caractéristiques. Cependant nous nous sommes penchés vers Debian et Ubuntu.

Caractéristiques fondamentales Debian :

- distribution faite de logiciels libres et au code source ouvert et sera toujours 100 % libre → libre à chacun de l'utiliser, de la modifier et de la distribuer.

Caractéristiques fondamentales Ubuntu :

- devenue la distribution Linux l'une des plus populaires, ce qui rend sa communauté très dynamique !
- Importante documentation disponible (de source officielle ou venant de la communauté des utilisateurs)

Arrivés là, un choix s'est offert à nous : lequel des deux choisir ? Chacun ayant ses avantages et inconvénients, nous avons finalement préféré choisir Ubuntu car cet OS est plus adapté aux développeurs de la boîte pour une raison : sa prise en main rapide.

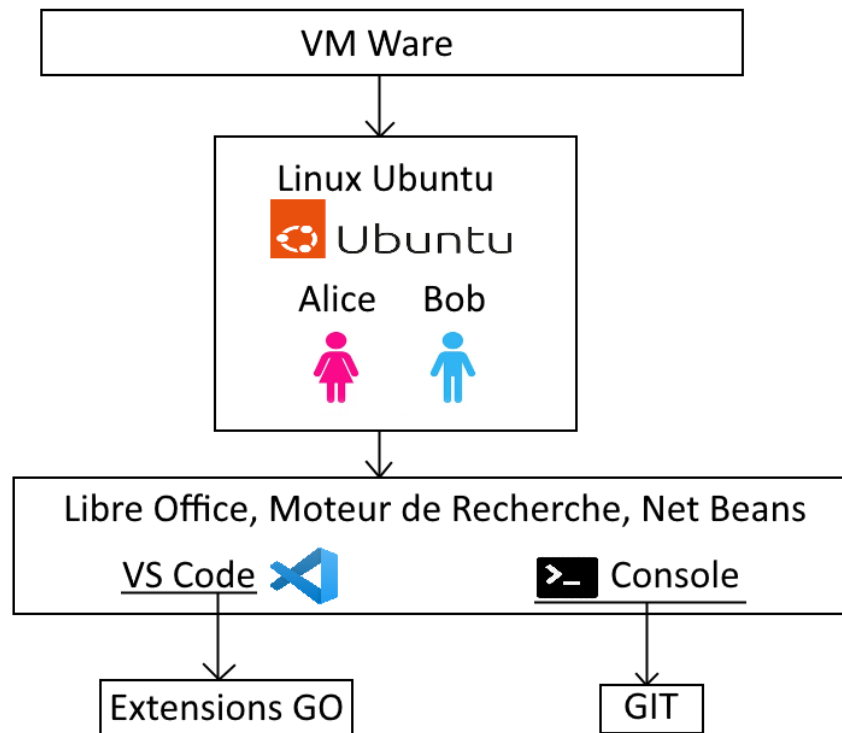


# Ubuntu



# debian

## Schéma de l'architecture logicielle :



## Configuration de la machine :

D'après notre choix des solutions, nous avons décidé de choisir Linux Ubuntu comme système d'exploitation. C'est celui qui se rapproche le plus de ce qu'on connaît et il est adapté aux professionnels car nous l'avons installé dans sa version professionnelle : Ubuntu 11.10, qui va permettre aux développeurs de BUTGAMES de coder leur jeu dans un environnement adéquat. Notre Ubuntu est installé dans sa version "légère", c'est-à-dire avec le strict minimum en termes d'applications et un bureau sobre.

Nous avons défini la machine virtuelle avec 4Go de ram, 40Go de disque max et 2 cœurs. De cette façon, les développeurs pourront parvenir à leurs besoins avec ces caractéristiques là.

Ainsi, nous avons suffisamment de liberté et de puissance afin de faire fonctionner correctement la machine et que l'utilisateur puisse s'en servir pour programmer. La seule contrainte que nous avons eu fut l'espace disque alloué (20Go étant le minimum demandé par Ubuntu).

Une fois la session d'Alice créée, nous avons dû rajouter celle de Bob. Cela a été possible grâce à la commande :

```
$ sudo adduser bob (création de l'utilisateur bob)
```

(sudo donnant temporairement le droits root à alice)

Pour programmer, les employés de BUTGAMES auront besoin de Vscode, comme précisé dans le cahier des charges, avec l'extension go qui leur permettra de programmer dans ce langage.

Si le cahier des charges le précisait, nous aurions pu installer directement d'autres extensions afin de permettre aux développeurs de faire du python par exemple mais ils peuvent les installer par eux mêmes plus tard.

Depuis la session d'Alice :

```
$ sudo snap install code --classic (commande utilisée pour installer vs code)
```

```
$ sudo snap install go --classic (installer go )
```

```
$ go version (pour verifier la version de go)
```

(cette commande fonctionne uniquement car Ubuntu a des snap)

Les développeurs de BUTGAMES vont devoir versionner leur projets, c'est-à-dire enregistrer petit à petit différentes versions de leur code, et pour cela ils auront besoins de GIT. Nous avons donc installés

```
$ sudo apt-get install git-all (pour installer git)
```

```
$ git version (pour verifier la version de git)
```

Afin de tester si Bob possède les droits corrects, nous devons voir s'il peut cloner un dépôt git et lancer un script bash sans problèmes. Pour cela, nous avons créé une "clé" temporaire (token) sur git car nous avons la vérification à double facteur donc impossible de se connecter depuis linux sans. Ensuite, nous avons créé pour Bob un dossier ssh, puis effectué la commande :

```
$ git clone --branch go https://gitlab-ce.iut.u-bordeaux.fr/rgiot/hello.sae4.but1.iut.git ssh/
```

(ensuite, le token, donc la "clé" temporaire nous est demandé)

Le --branch go permet de créer une nouvelle branche sur laquelle on va cloner. Pour cela nous avons dû tout d'abord installer cargo, et nightly :

```
$ sudo apt install cargo
```

Le problème étant qu'il faut installer le package nightly et que ce dernier peut être installé via rust :

Donc nous avons tout d'abord installé rust :

```
$ sudo snap install rustup --classic
```

Puis nightly :

```
$ sudo rustup install nightly
```

Enfin, il a fallu modifier le prompt de Bash afin qu'il affiche les informations sur l'état du dépôt git. Pour cela nous avons dû modifier le fichier de configuration bashrc.

```
$ nano ~/.bashrc (pour se rendre dans le fichier et le modifier)
```

Et une fois dans bashrc: ajouter à la toute fin puis enregistrer:

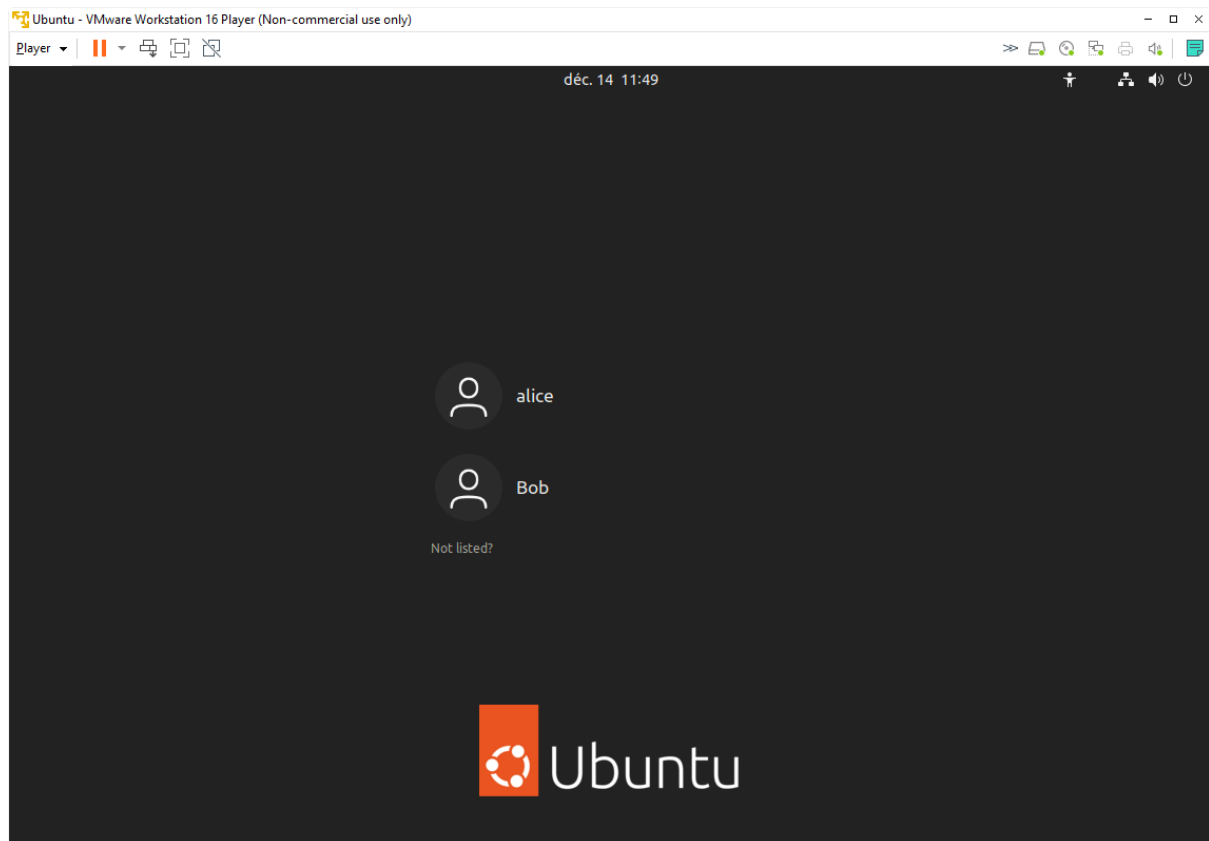
```
export PS1="\h@\u_W-\$(git branch 2> /dev/null | sed -e '/^[^*]/d' -e 's/* \(.*/(\1)'):\[$(tput sgr0)\]"
```

Ainsi, avec toutes les commandes précédentes, nous avons pu créer une machine virtuelle qui répond aux besoins des développeurs de BUTGAMES. Ils ont à leur disposition :

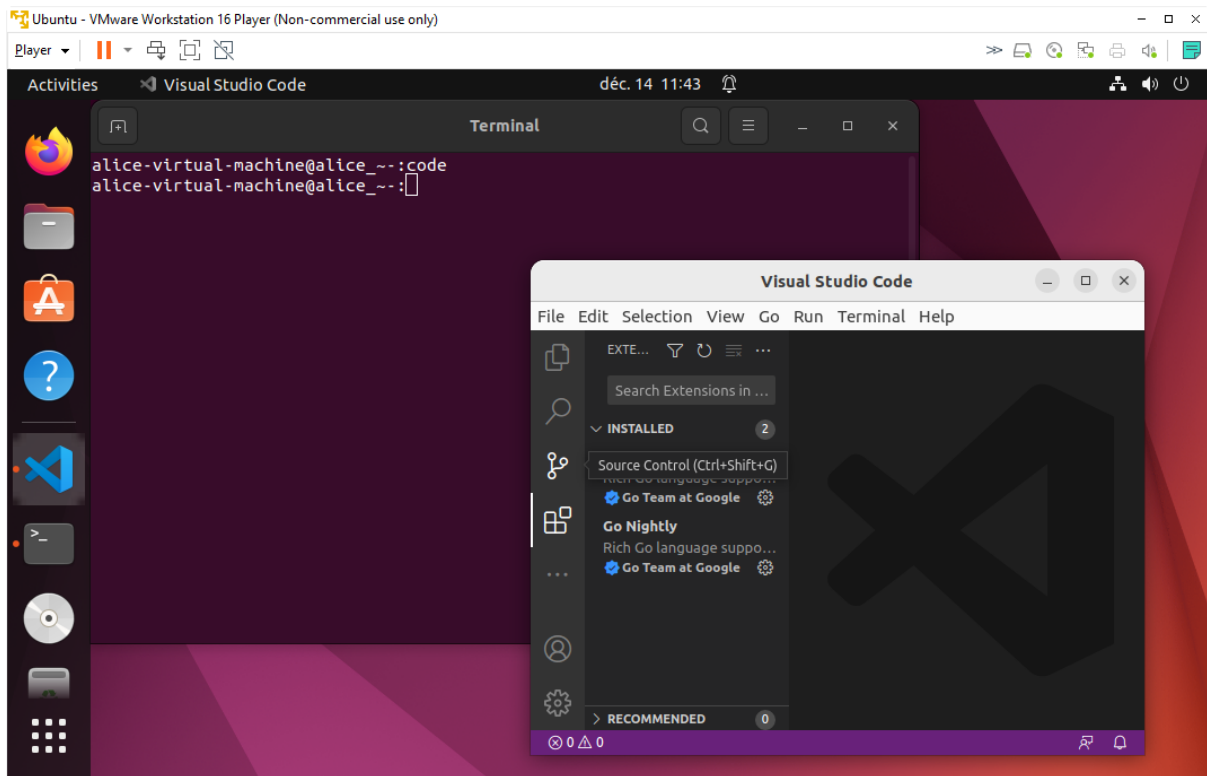
- Une machine convenable (suffisante pour coder en tout cas)
- Un système d'exploitation Linux léger et adapté
- vscode et les extensions nécessaires pour go
- git pour versionner le code
- Et un prompt qui affiche les informations nécessaires

## Notice d'utilisation:

Lors du démarrage de la machine virtuelle, sélectionner son compte (bob ou alice), puis rentrer son code.



Maintenant connecté à la session, démarrer vscode et créer un nouveau projet en sélectionnant le langage go. Et c'est tout ! Vous pouvez maintenant coder en go.



Pour vérifier le test d'installation de go, nous avons copié dans le dossier ssh sur le bureau un script qu'il faudra exécuter. Dans le terminal il faudra donc se placer dans le dossier:

```
$ cd ssh/
```

puis exécuter le script (nous l'avons déjà configuré afin qu'il soit exécutable)

```
$ ./script.sh
```

```
bob@alice-virtual-machine:~/Desktop/hello.sae4.but1.iut$ bash test.sh
!... Hello World ...!
bob@alice-virtual-machine:~/Desktop/hello.sae4.but1.iut$
```

## PORTFOLIO:

La saé installation poste, consistait en la création et l'installation d'une machine virtuelle pour une entreprise de développement qui utilise le langage go. Lors de ce projet, nous avons beaucoup progressé dans la compréhension de linux et du terminal de manière générale. Installer des applications, des extensions et ajouter des utilisateurs ont été des commandes nouvelles pour nous, notamment pour la gestion des droits utilisateurs.

