

# Apollo : utilisation

MILLON Florentin

15 mai 2025

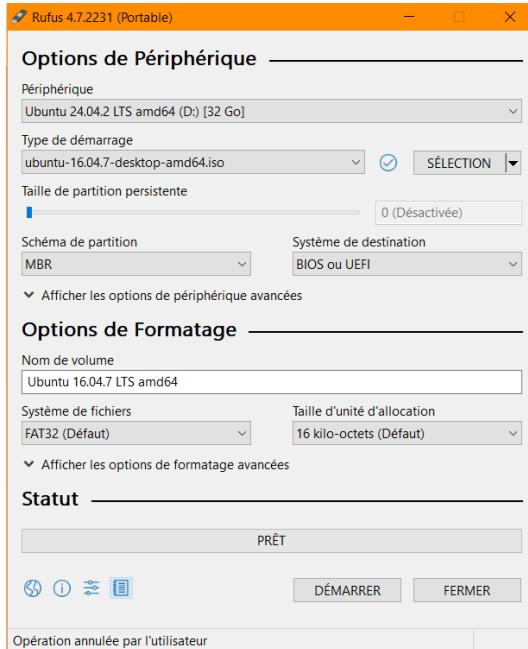
## Table des matières

<b>1 Instalation ubuntu 16.04.7 :</b>	<b>3</b>
1.1 Clé bootable et ISO : . . . . .	3
1.2 Bios : . . . . .	4
1.3 Demarrage ubuntu : . . . . .	5
<b>2 VScode :</b>	<b>10</b>
<b>3 Premier pas :</b>	<b>10</b>
3.1 Code Demo : . . . . .	10

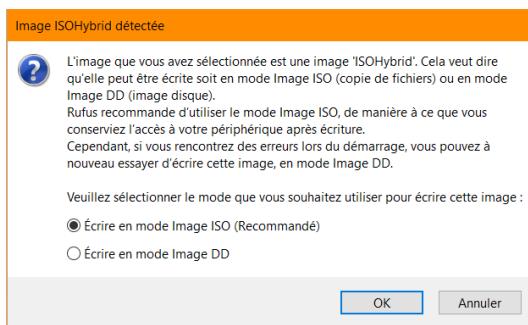
# 1 Instalation ubuntu 16.04.7 :

## 1.1 Clé bootable et ISO :

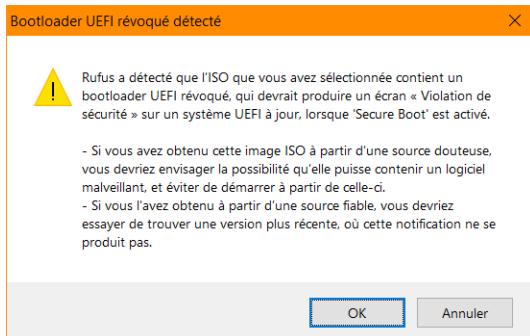
Pour lancer les codes de démonstrations, il est demandé d'installer un système Linux. Le système utilisé pour les codes démo est Ubuntu 16.04 installé sur une machine virtuelle sous Windows 10 (ou en OS principal). L'installation se fait avec une clé bootable principale avec le fichier ISO. Le fichier ISO est disponible dans le répertoire Apollomy\_docOS (le fichier ISO a été téléchargé sur le site Ubuntu). Il y a également le logiciel rufus qui permet de rendre la clé bootable. Une fois que rufus est lancé sur Windows il suffit de sélectionner la clé USB et le fichier ISO, puis de cliquer sur démarrer.



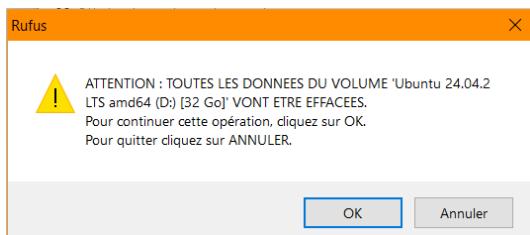
Selectionner "Écrire en mode Image ISO (Recommandé)" et cliquer sur "OK".



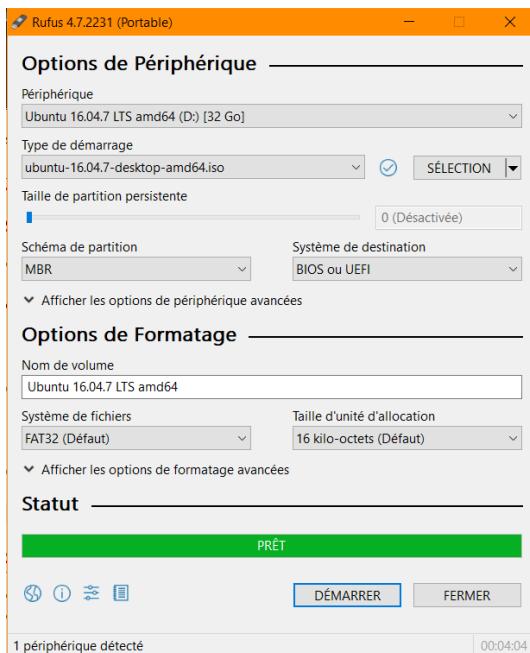
Ce message vous prévient qu'il peut y avoir un message d'avertissement du Bios lors du boot sur la clé. Pour être sûr que vous n'installez pas de logiciel malveillant vérifier votre source.



Ce message vous prévient que vos données sur la clé vont être effacées, pour continuer, cliquer sur "OK".



Le formatage de la clé est fini, cliquez sur fermer et éjecter votre clé avant de la débrancher.



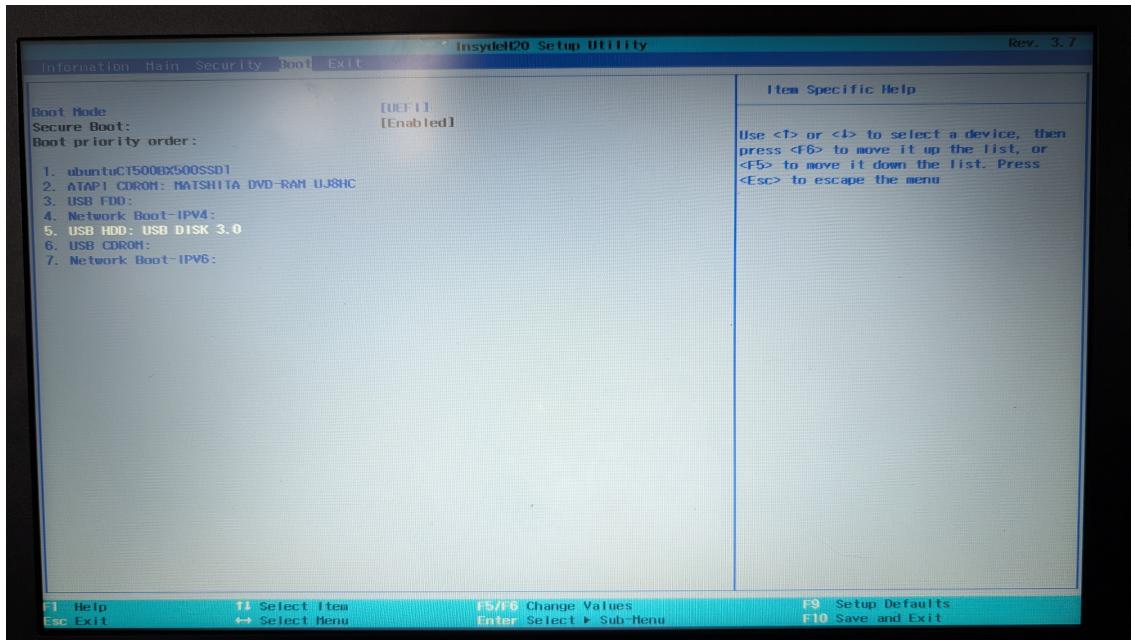
## 1.2 Bios :

Maintenant que la clé est prête, vous pouvez la brancher et allumer la machine. Entré dans le bios si votre bios est récent, démarrer sur votre clé USB dans le menu boot rapide ou raccourci boot. Sinon, aller dans le menu boot pour démarrer directement sur votre clé avant votre ancien OS.

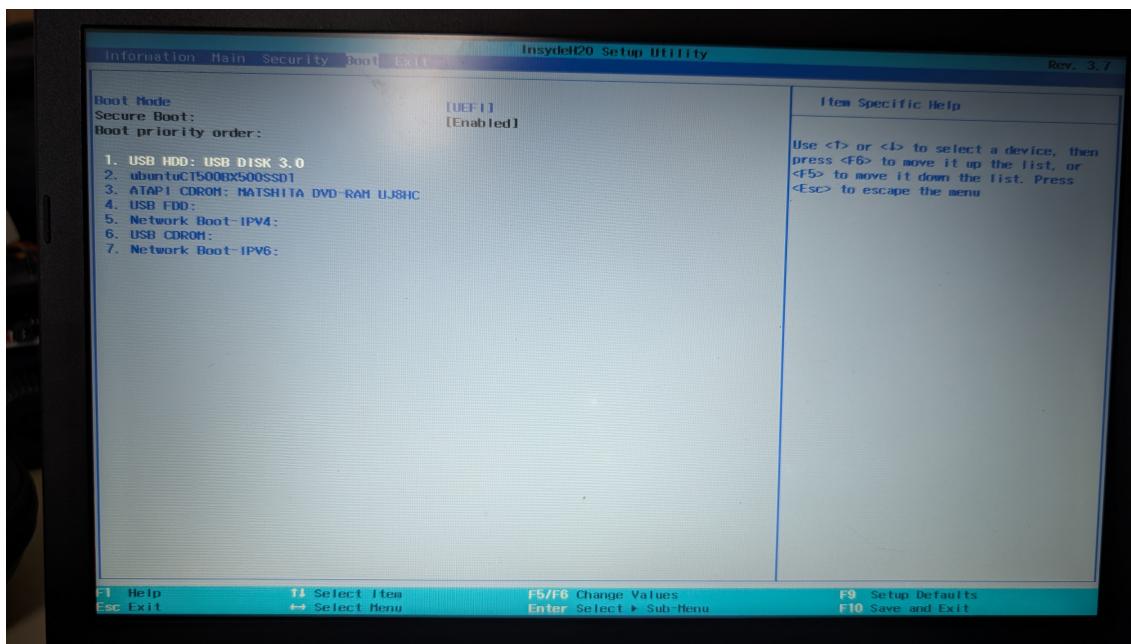
Nouveau Bios :

Ancien Bios :

Votre bios devrait ressembler à ça (il se peut que ça ne soit pas présenté exactement pareil, mais les options sont les mêmes). Sélectionnez votre clé USB avec les flèches.

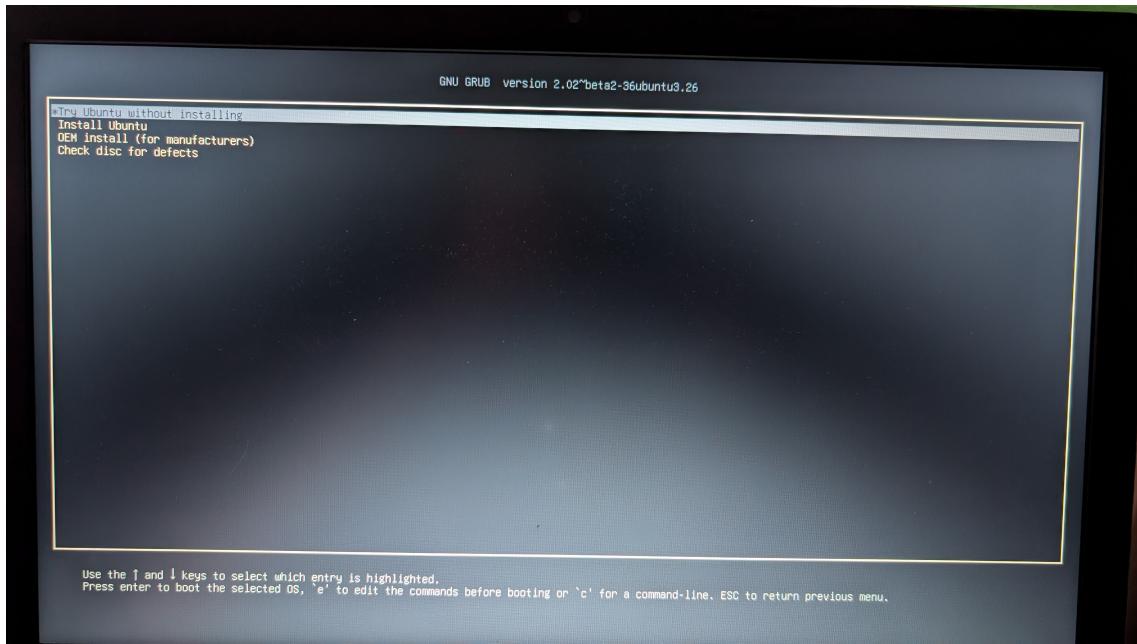


Une fois sur votre clé, appuyez sur la touche permettant de monter votre clé dans la liste (dans mon cas F6). Après avoir fait tout ça, vous pouvez appuyer sur F10 pour quitter et sauvegarder. Le PC va chercher votre clé pour démarrer dessus.

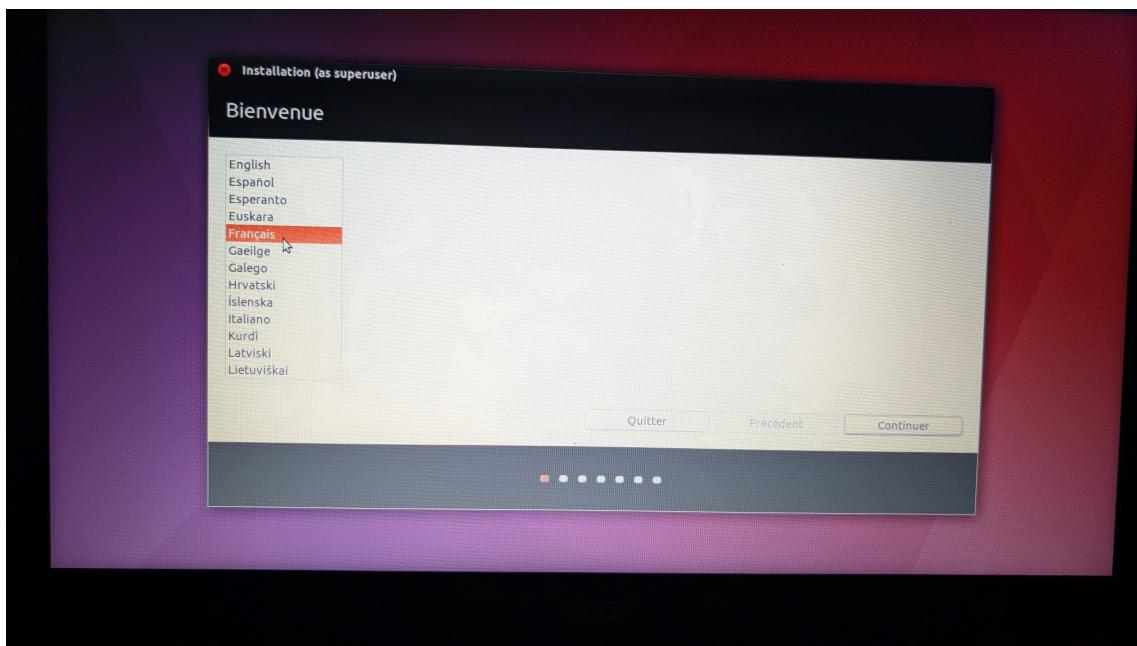


### 1.3 Demarrage ubuntu :

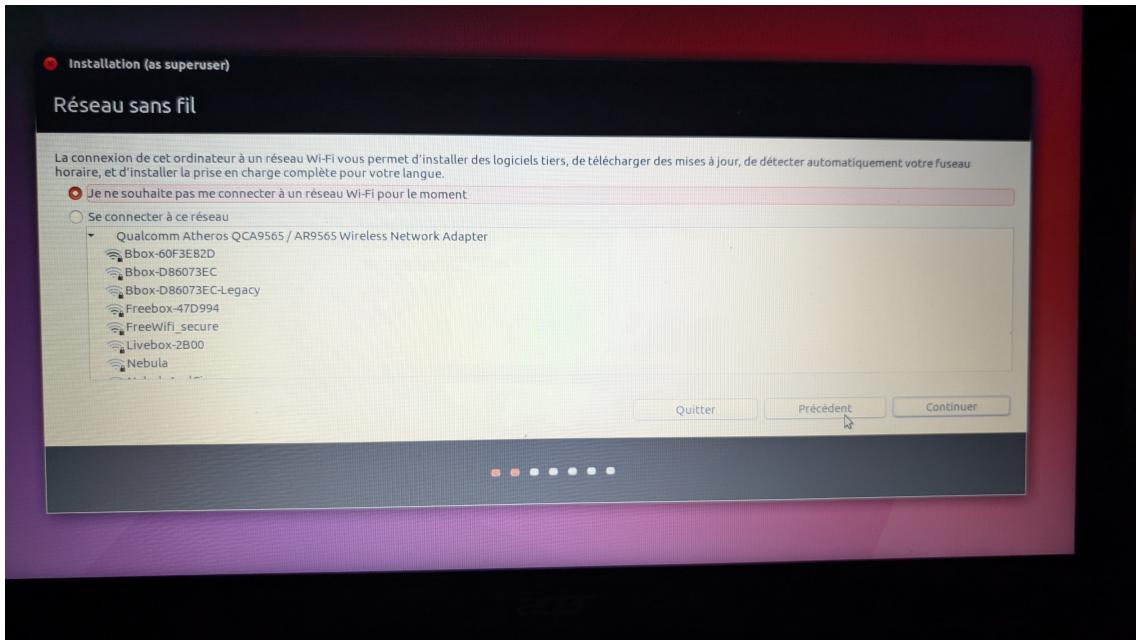
Une fois que votre PC aura démarré sur la clé, vous allez atterrir sur ce menu. Sélectionnez "Install Ubuntu".



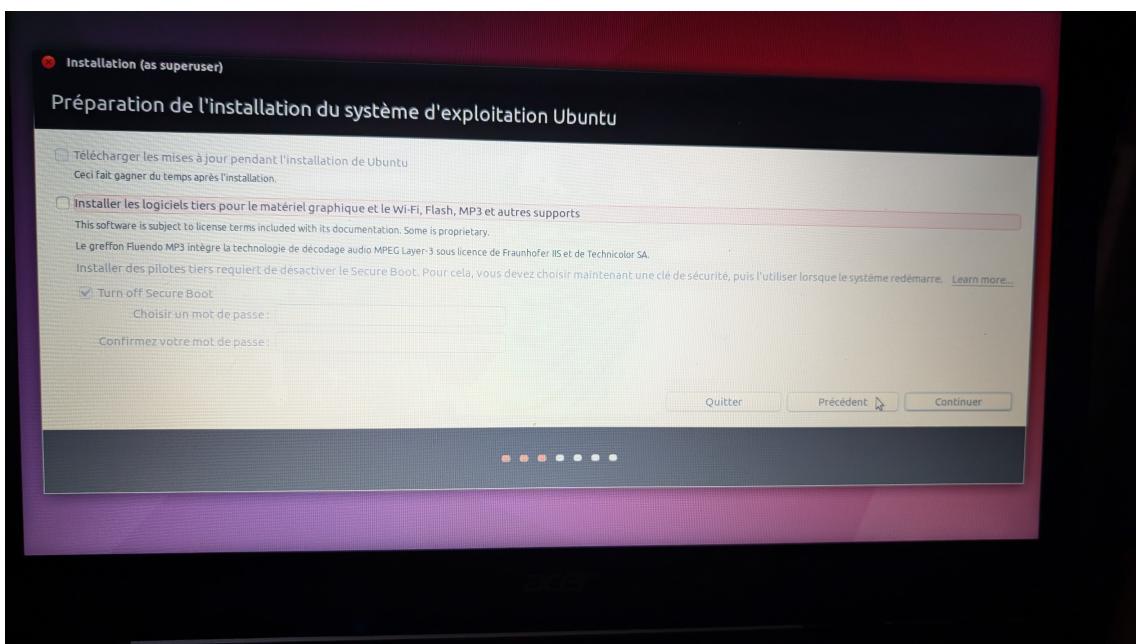
Sélectionnez la langue que vous voulez, puis cliquez sur "OK".



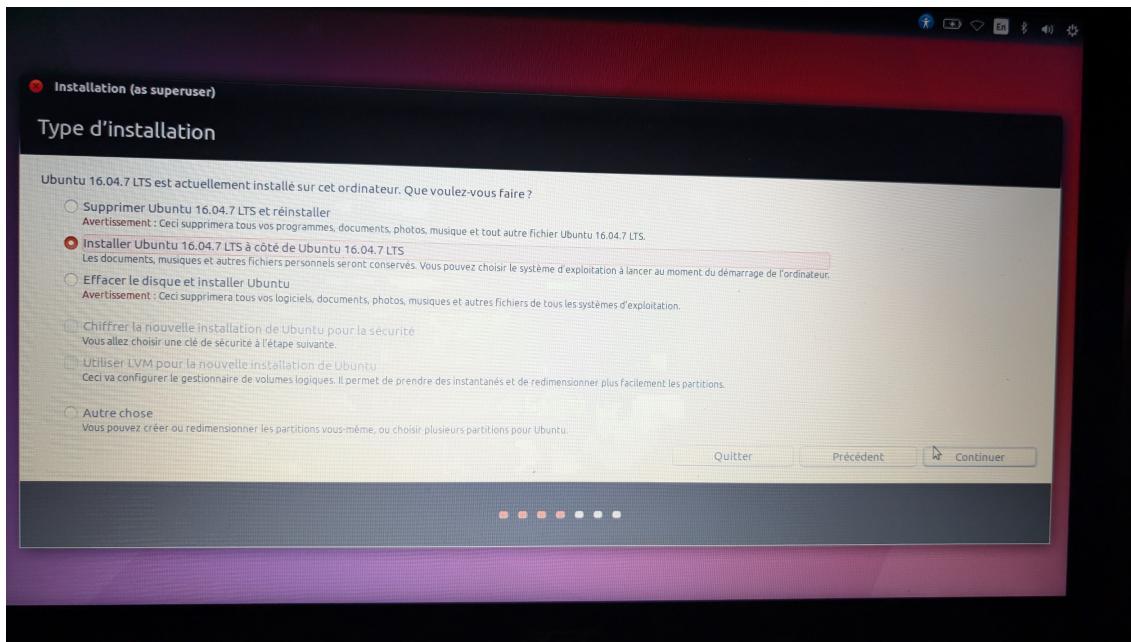
Il n'y a pas besoin de vous connecter pour continuer. C'est utile seulement pour faire les mises à jour et télécharger des logiciels tiers (inutiles dans notre cas). Si vous êtes à la fac, il faudra vous connecter plus tard.



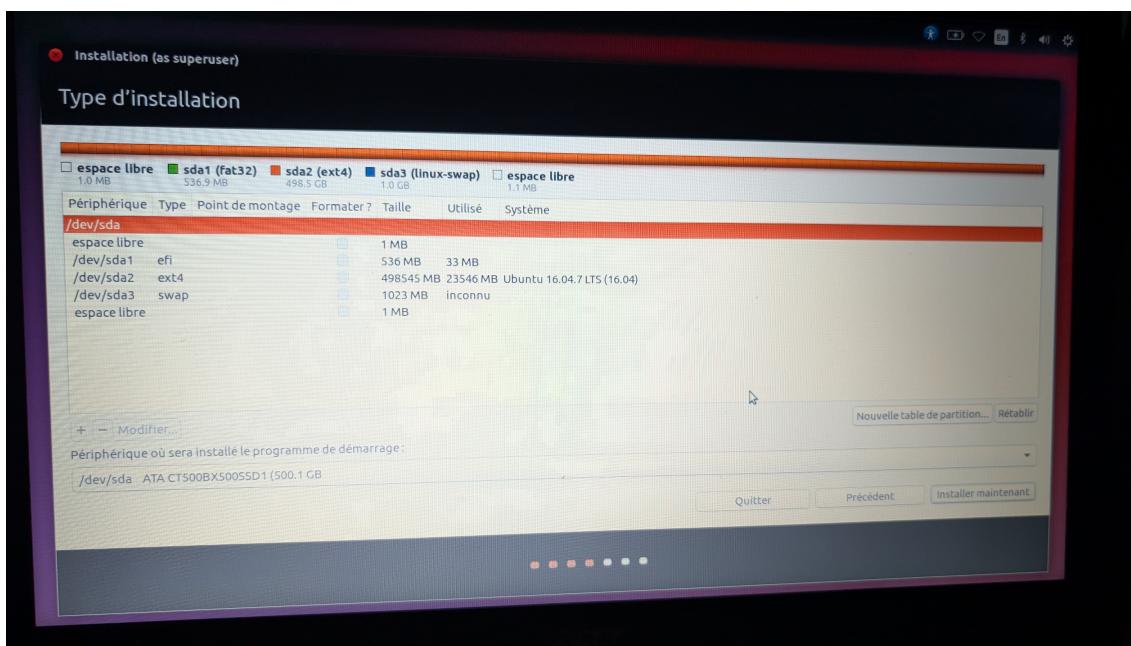
Dans notre cas, on n'a pas besoin de tous ces logiciels, donc on ne coche rien et on passe directement à l'étape d'après.



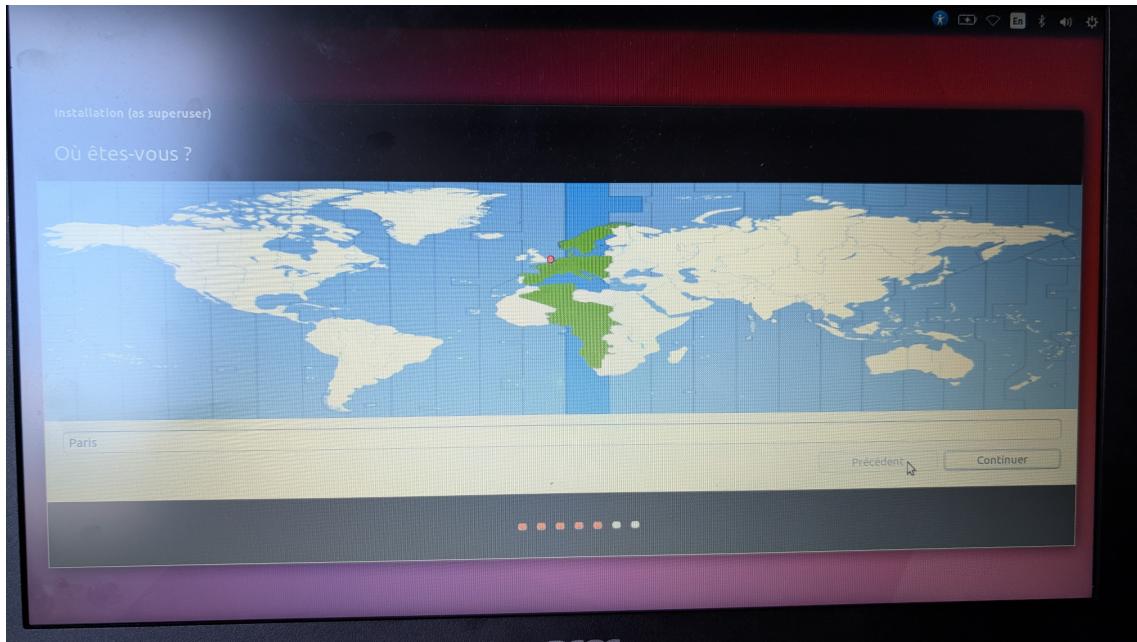
Ici, vous pouvez choisir d'installer Ubuntu tout seul, à côté de votre OS ou de personnaliser votre installation en choisissant la taille de vos partitions.



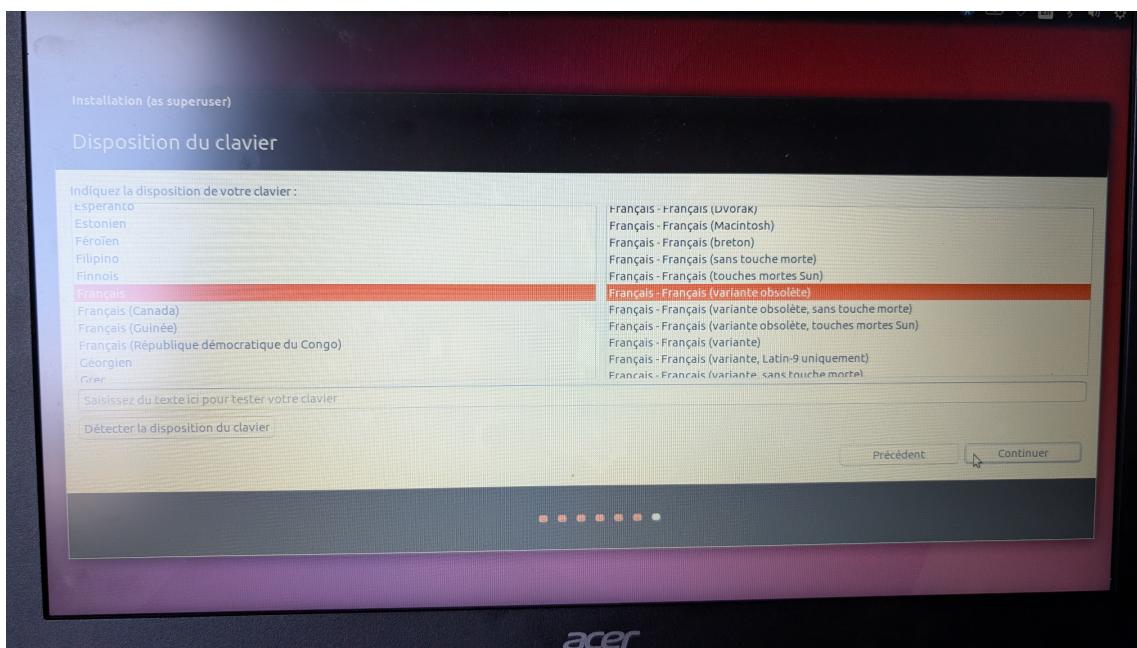
Dans le cas de l'installation personnalisée il y a une vidéo qui explique très bien comment on fait cela. ici



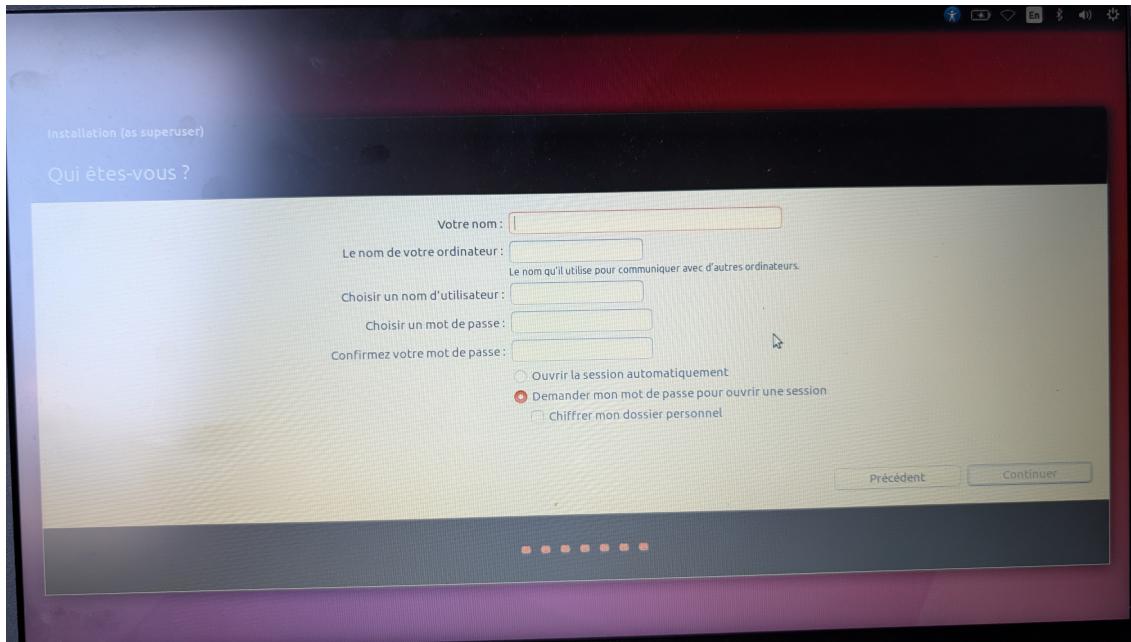
Sélectionner votre fuseau horaires.



Choisissez votre disposition clavier.



Puis donnez un nom à votre os et configurez le mot de passe administrateur. Vous pouvez choisir s'il faut utiliser le mot de passe à chaque ouverture de session.



Une fois l'installation finie, Ubuntu vous demandera de redémarrer le PC. Il faut retirer la clé USB entre le moment où le PC s'éteint et le moment où il s'allume.

## 2 VScode :

Pour le développement des codes, j'ai choisi (arbitrairement et par habitude) d'utiliser VS Code. Pour l'installer, il faut une version antérieure. Pour cela, il vous suffit d'exécuter les commandes suivantes :

```

1 # download package
2 sudo apt install curl
3 curl https://packages.microsoft.com/keys/microsoft.asc | gpg --dearmor > microsoft.gpg
4 sudo mv microsoft.gpg /etc/apt/trusted.gpg.d/microsoft.gpg
5 sudo sh -c [echo "deb [arch=amd64] https://packages.microsoft.com/repos/vscode stable main" >
→ /etc/apt/sources.list.d/vscode.list]
6
7 # install vs code
8 sudo apt-get update
9 sudo apt install aptitude
10 sudo aptitude install code

```

Rejeter les versions jusqu'à la proposition de l'ancienne version de VS Code (1.11.2). Il faut en rejeter beaucoup avant de l'avoir.

## 3 Premier pas :

Avant de commencer à développer votre propre code, vous pouvez utiliser les codes démo pour vérifier que toutes les fonctionnalités sur Apollo fonctionnent. Pour cela, il faudra vous rendre sur la page de documentation d'Apollo ici ou dans le **DOCUMENT**.

### 3.1 Code Demo :

- **Préparation de l'environnement opérationnel**

Le système utilisé dans ce document est Ubuntu 16.04, et la version gcc est 5.3.1. La version SDK utilisée est la version de compilation gcc5.4 de l'architecture x86. Vérifiez la version gcc du système. Vous pouvez entrer dans le terminal :

```
1 $ gcc -v
```

Pour interroger la version de gcc.

- **Processus de compilation d'un exemple de programme SDK :**

1. Remplacez <lien Web> par l'adresse de téléchargement de la ressource correspondante.

```
1 $ wget <web link>
```

2. Accédez au dossier Téléchargements et décompressez le dossier portant le même nom dans le même répertoire.

- **docs** : Documentation générale du SDK Slamware
- **linux-x86\_64-debug** : contient les fichiers d'en-tête et les fichiers lib requis pour la compilation
- **linux-x86\_64-release** : Le fichier exécutable compilé sera stocké dans ce dossier
- **exemples** : un exemple de code se trouve dans ce dossier
- et les fichiers compiler\_version.txt, mak\_common.inc, mak\_def.inc

3. Entrez dans le dossier d'échantillons et créez le fichier Makefile.

```
1 slamware_sdk_linux-x86_64-gcc5.4$ cd samples/
2 slamware_sdk_linux-x86_64-gcc5.4$ make
```

4. Le chemin du fichier exécutable généré après make est :

```
1 slamware_sdk_linux-x86_64-gcc5.4/linux-x86_64-release/output/
```

5. Exécutez le fichier exécutable correspondant dans le chemin correspondant et entrez l'adresse IP du robot (la valeur par défaut est 192.168.11.1).

```
1 slamware_sdk_linux-x86_64-gcc5.4/linux-x86_64-release/output$ ./moveandpathdemo 192.168.11.1
```

- **Résultats d'exécution** La vérification des fichiers sources dans le dossier d'exemple montre que les fichiers sources sont écrits en c++. L'exemple de fichier makefile inclus dans le SDK générera trois fichiers exécutables :

1. moveandpathdemo : Le système produira de manière cyclique des informations relatives au mouvement et se déplacera de manière cyclique entre les points (2,2), (-2,2), (-2,-2), (2,-2). (Si il n'y a pas la place iras tout droit sans toucher les murs)

```

dan@ubuntu:~/Downloads/slamware_sdk_linux-x86_64-gcc5.4/linux-x86_64-release/output$ ./moveandpathdemo 192.168.11.1
SDK Version: 2.6.0_dev
SDP Version: 2.6.0_dev-sdp_vre-20180327
Connection Successfully
Empty
Action Name: EmptyAction
Non-Empty
Action ID: 30
Action Name: MoveAction
Remaining Milestones:
(2, 2)
(-2, 2)
(-2, -2)
(2, -2)
Non-Empty
Action ID: 30
Action Name: MoveAction
Non-Empty
Action ID: 30
Action Name: MoveAction
Remaining Milestones:
(2, 2)
(-2, 2)
(-2, -2)
(2, -2)
Non-Empty
Action ID: 30
Action Name: MoveAction
Non-Empty
Action ID: 30
Action Name: MoveAction
Remaining Milestones:
Empty
Action Name: EmptyAction
Non-Empty
Action ID: 31
Action Name: MoveAction

```

2. mapdemo : Le système produira cycliquement des données de points laser et des dimensions de zone connues

```

distance: 100000, angle: -3.06547, valid: false
distance: 1.00019, angle: -3.07596, valid: true
distance: 1.02027, angle: -3.09064, valid: true
distance: 1.03527, angle: -3.10746, valid: true
distance: 1.06031, angle: -3.12428, valid: true
distance: 1.10214, angle: -3.13898, valid: true
Known Area:
(-4.6, -5.9, 8.45, 8.25)
distance: 3.60978, angle: 3.09097, valid: true
distance: 3.65902, angle: 3.0735, valid: true
distance: 3.96871, angle: 3.05838, valid: true
distance: 4.10022, angle: 3.04109, valid: true
distance: 2.25317, angle: 3.01819, valid: true
distance: 100000, angle: 3.0091, valid: false
distance: 4.10877, angle: 3.00691, valid: true
distance: 100000, angle: 2.99193, valid: false
distance: 2.65089, angle: 2.98575, valid: true
distance: 100000, angle: 2.97448, valid: false
distance: 4.20952, angle: 2.97205, valid: true
distance: 4.18594, angle: 2.95508, valid: true
distance: 4.16178, angle: 2.93779, valid: true
distance: 4.14566, angle: 2.92068, valid: true
distance: 4.09601, angle: 2.90329, valid: true
distance: 3.65851, angle: 2.88424, valid: true
distance: 4.07479, angle: 2.86897, valid: true
distance: 4.08546, angle: 2.85179, valid: true
distance: 4.07668, angle: 2.83431, valid: true
distance: 4.08289, angle: 2.81699, valid: true
distance: 4.09174, angle: 2.80004, valid: true

```

3. map2bmp : le système génère les paramètres actuels liés à la carte et génère le fichier de carte slamware\_mapdump.bmp dans le dossier de sortie.

```
dan@ubuntu:~/Downloads/slamware_sdk_linux-x86_64-gcc5.4/linux-x86_64-release/output$ ./map2bmp 192.168.11.1
Connecting to the SDP @ 192.168.11.1...
Fetching Map Info...
> Map Area: (-4.6000,-5.9000,3.8500,2.3500)
> Cell Dimension: (169,165)
> Cell Resolution: (0.0500,0.0500)
> Timestamp: 24088915ll
Fetching Localization Info...
> Position: (-0.2877,0.0009,0.0000)
> Heading: -149.1204
```