

Apollo : utilisation

MILLON Florentin

9 mai 2025

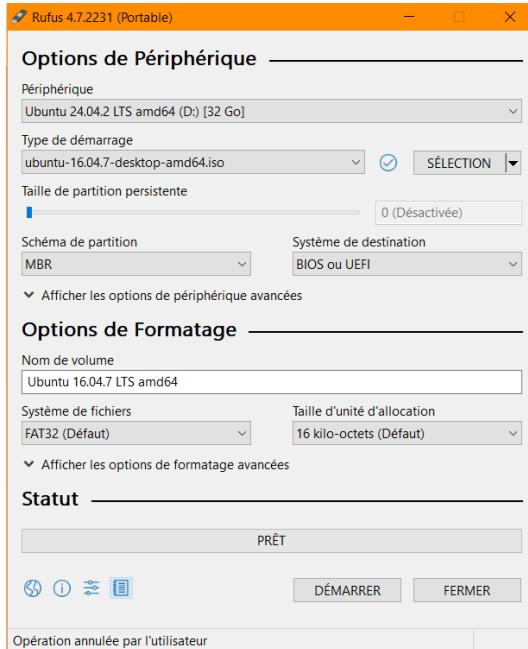
Table des matières

1 Instalation ubuntu 16.04.7 :	3
1.1 Cle bootable et iso :	3
1.2 Bios :	4
1.3 Demarrage ubuntu :	5
2 VScode :	8
3 Premier pas :	9
3.1 Code Demo :	9

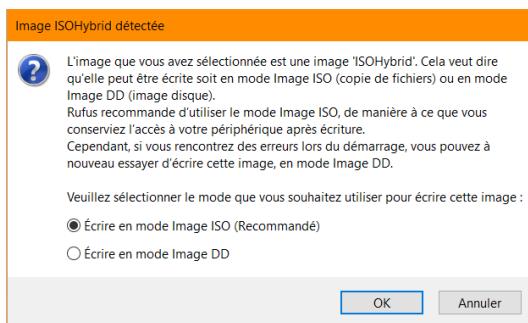
1 Instalation ubuntu 16.04.7 :

1.1 Cle bootable et iso :

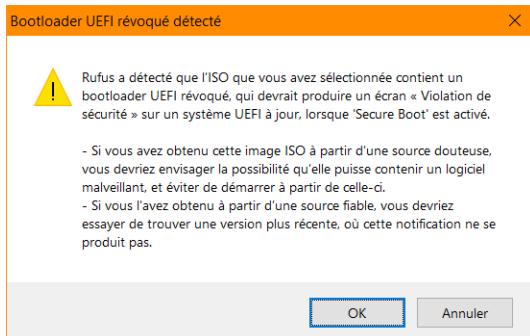
Pour lancer les codes de demonstrations il est demander d'installer un système linux. Le système utilisé pour les codes demo est Ubuntu 16.04 installé sur une machine virtuelle sous Windows 10(ou en OS principale). L'instalation ce fait avec cle bootable principale avec le fichier iso. Le fichier iso est disponible dans le repertoire Apollomy_docOS (le fichier iso a ete telecharger sur le site ubuntu). Il y a egalement le logiciel rufus qui permet de rendre la cle bootable. Une fois que rufus est lancer sur windows il suffit de selectionner la clé usb et le fichier iso puis de cliquer sur demarrer.



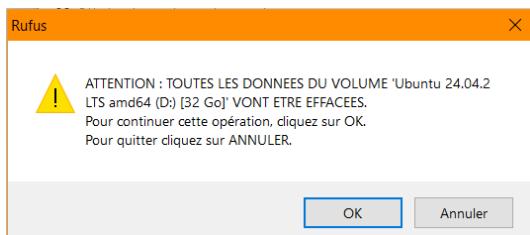
Selectionner "Ecrire en mode Image ISO (Recommandé)" et clicker sur "OK".



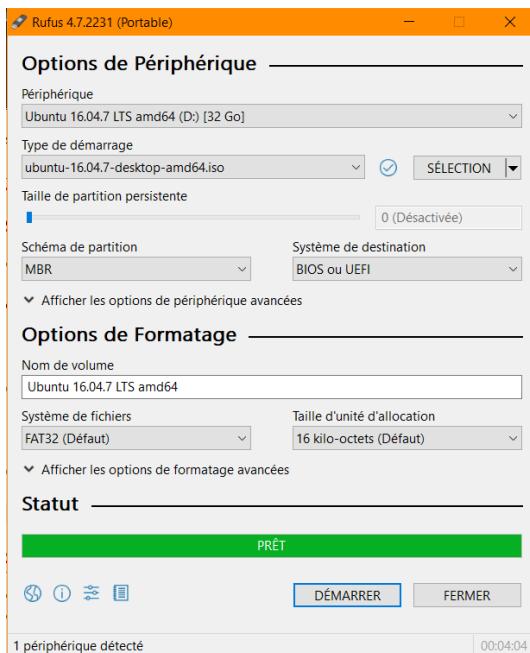
Ce message vous previens qu il peut y avoir un message d'avertissement du Bios lors du boot sur la cle. Pour etre sur que vous n'instaler pas de logiciel malveillant verifier votre source.



Ce message vous previens que vos donnée sur la cle vont etre effacé pour continuer, clicker sur "OK".



Le formatage de la cle est finie, clicker sur fermer et ejecter votre cle avant de la debrancher.



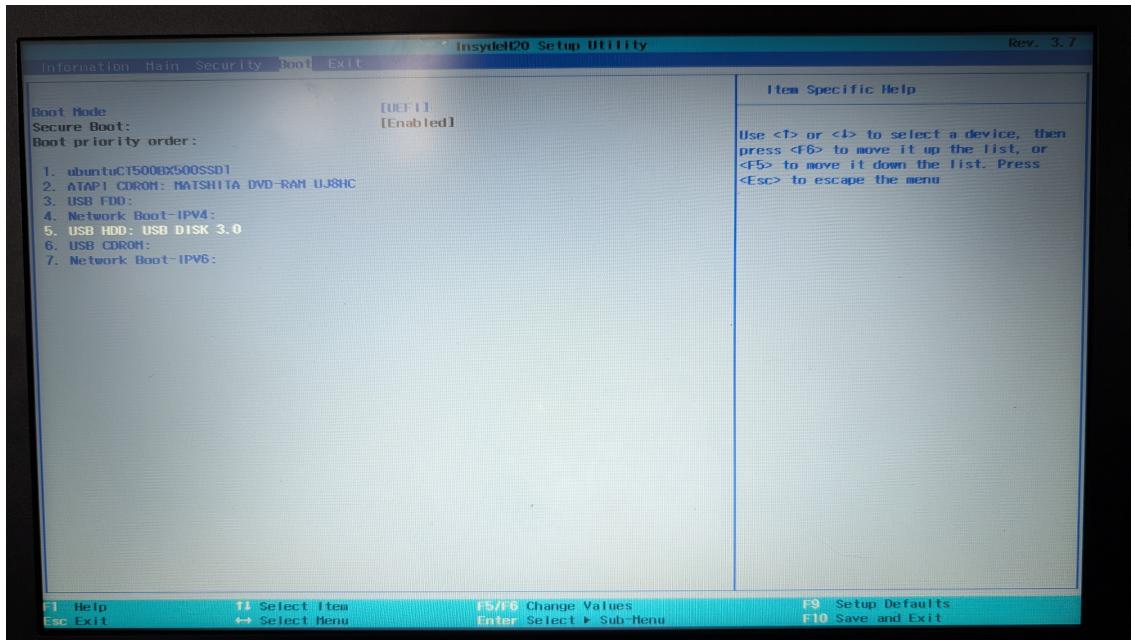
1.2 Bios :

Maintenant que la cle est prête vous pouvez la brancher et allumer la machine. Entré dans le bios si votre bios est recent demarrer sur votre cle usb dans le menus boot rapide ou racourcie boot. Sinon aller dans le menus boot pour demarrer directement sur votre cle avant votre ancien os.

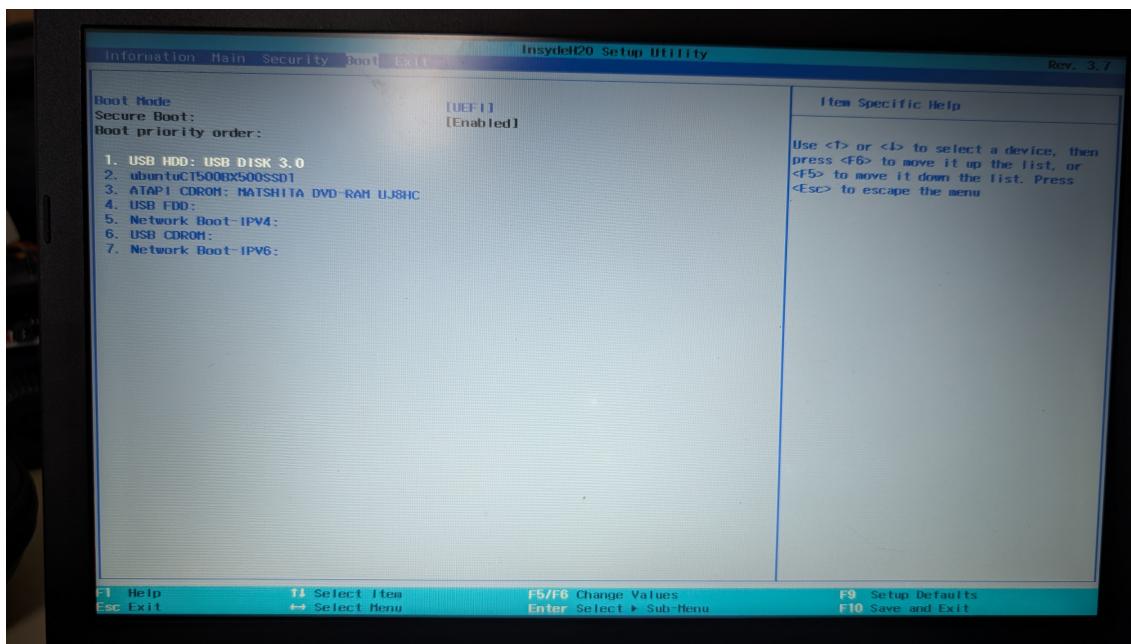
Nouveau Bios :

Ancien Bios :

Votre bios devrait ressembler à ça (il se peut que ça ne soit pas présenter exactement pareil mais les options sont les mêmes). Sélectionner votre cle usb avec les flèches.

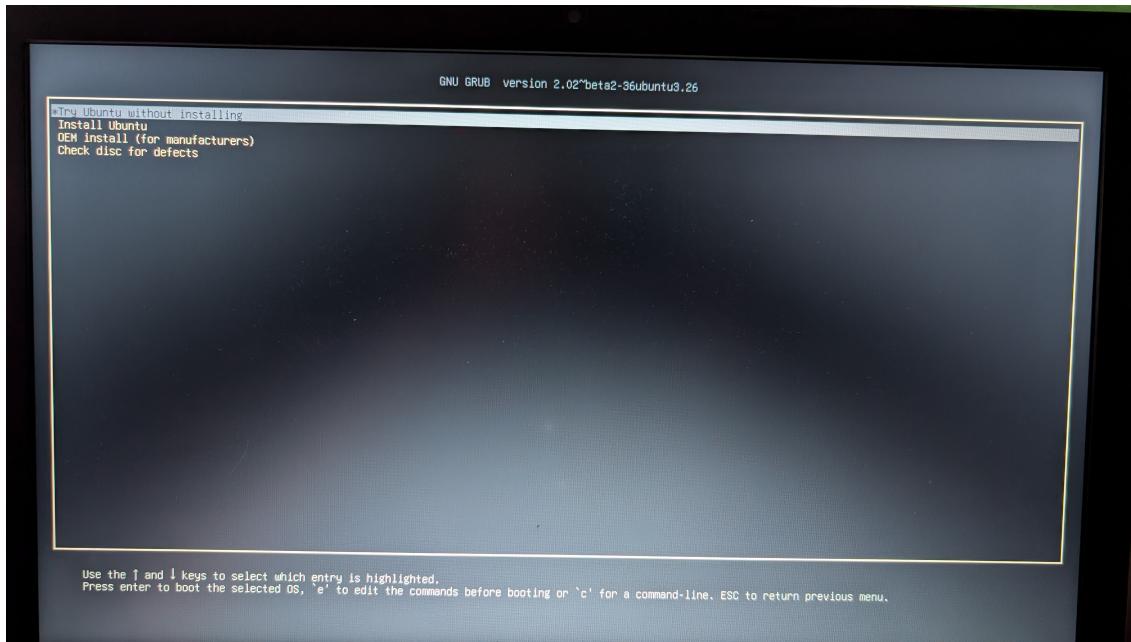


Une fois sur votre clé appuyer sur la touche permettant de monter votre clé dans la liste (Dans mon cas F6). Après avoir fait tout ça vous pouviez appuyer sur F10 pour quitter et sauvegarder. Le pc va chercher votre clé pour démarrer dessus.

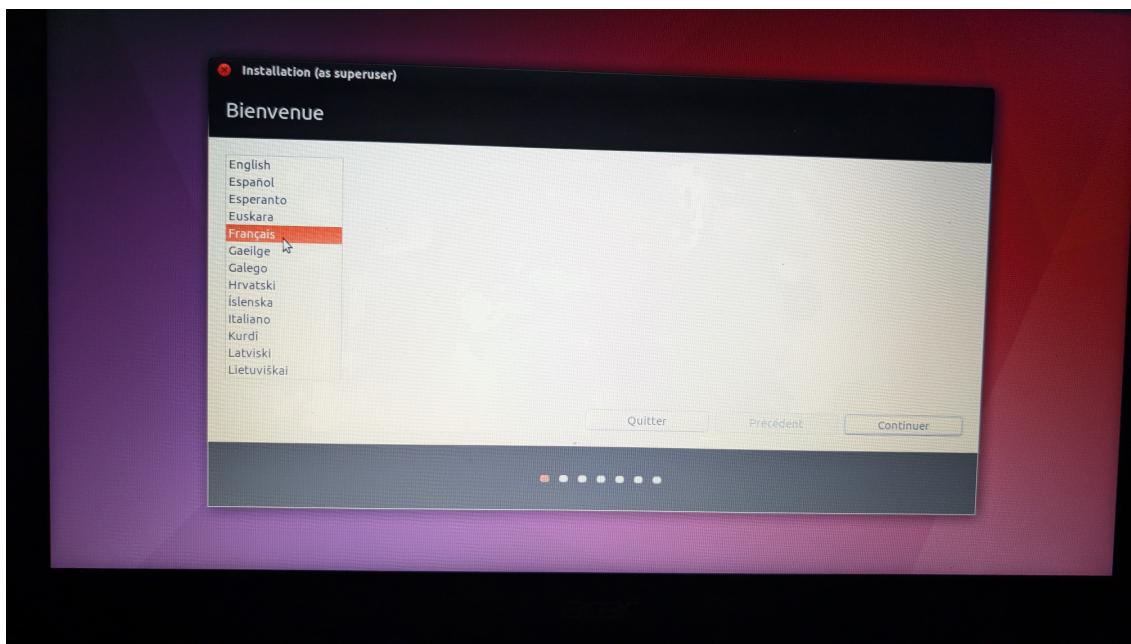


1.3 Demarrage ubuntu :

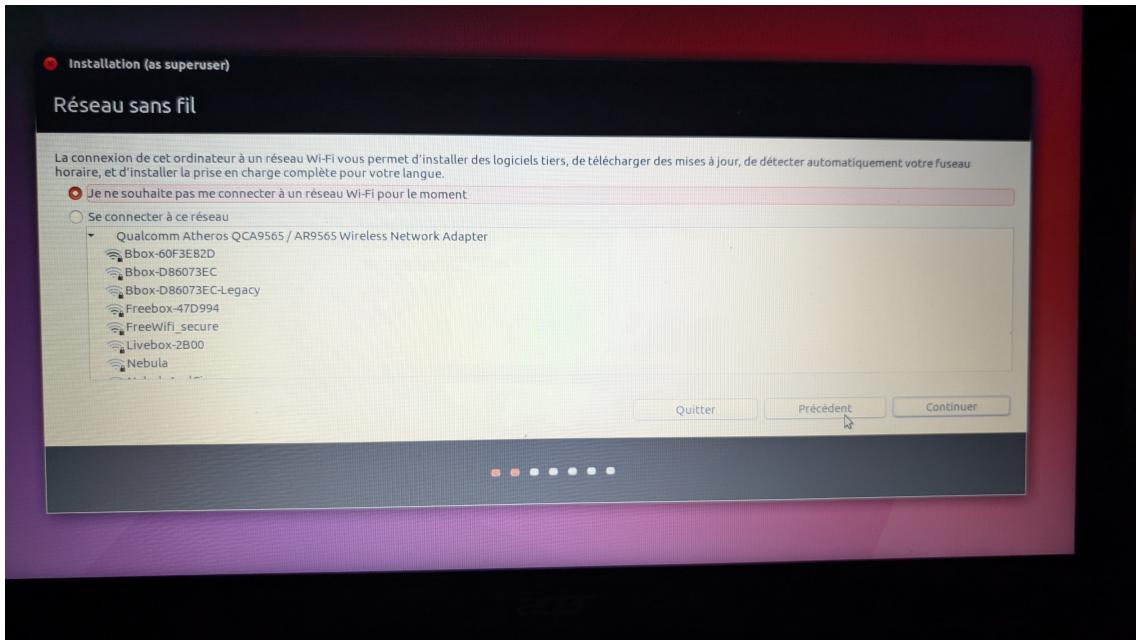
Une fois que votre pc aura démarrer sur la clé vous allez atterrir sur ce menu. Sélectionner "Install Ubuntu".



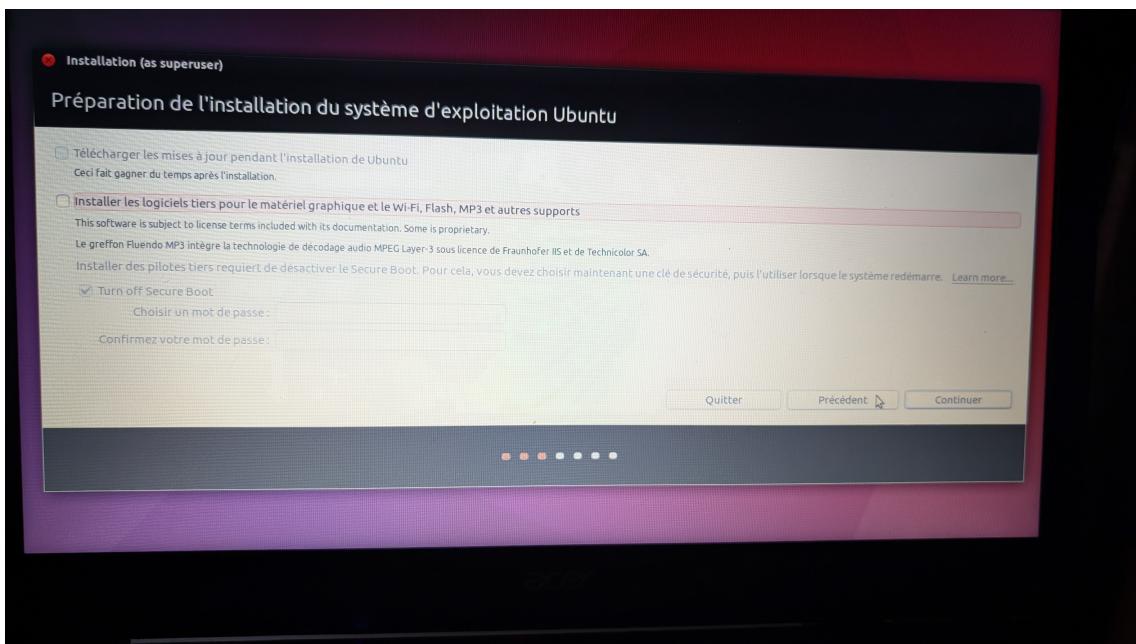
Selectionner la langue que vous voulez ouis clicker sur "OK".



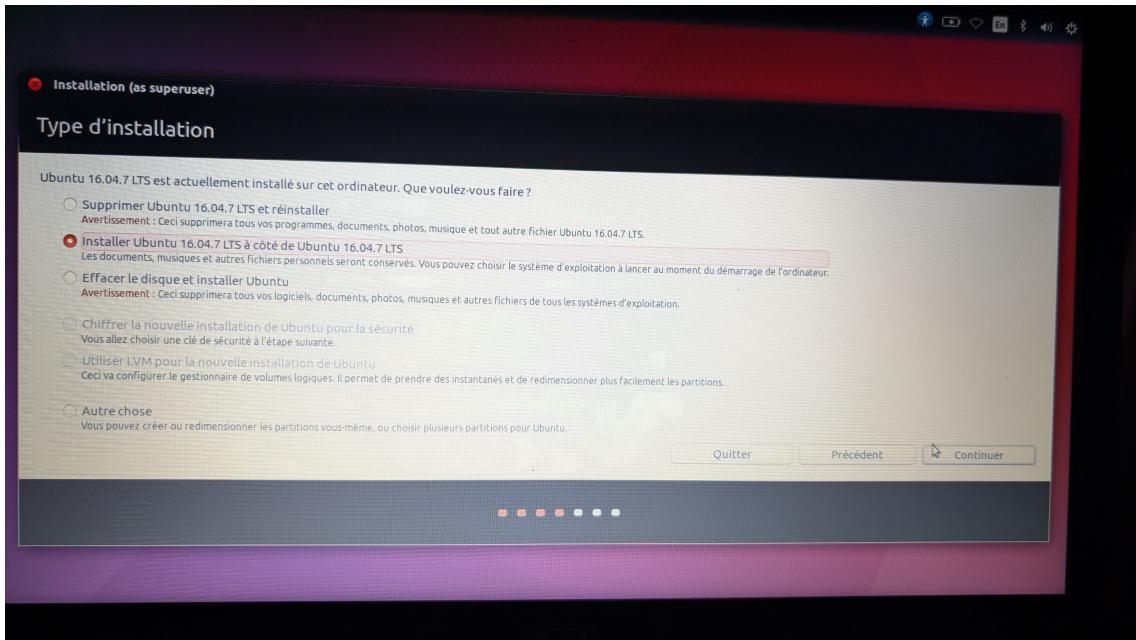
Il n'u a pas besoin de vous connecter pour continué. C'est utile seulement pour faire les mise a jour et telecharger des logiciel tiers (inutile dans notre cas) Si vous etes a la fac il faudra vous connecter plus tard.



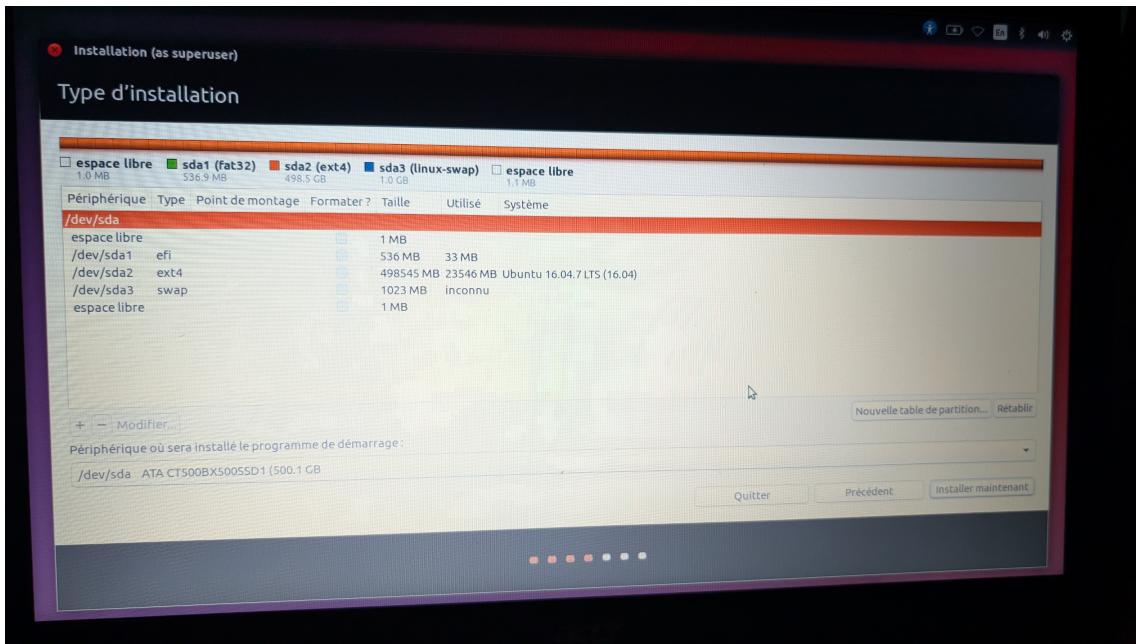
Dans notre cas on a pas besoin de tout ces logiciel donc on coche rien et on passe direct a l'étape d'après.



Ici vous pouvez choisir d'installer Ubuntu tout seule, à côté de votre os ou de personnalise votre instalation en choisissant la taille de vos partitions.



Dans le cas de l'instalation personalisé il y une video qui explique tres bien comment on fait cela. ici



Selectionner votre fuseau horraires

2 VScode :

Pour le developpement des code j'ai choisie (arbitrairelent et par habitude) d'utiliser VScode. Pour l'ins-taller ifaut une version antérieur. Pour cela il vous suffit d'executer les commandes suivantes :

- 1 sudo apt install aptitude
- 2 sudo aptitude install code

Rejeter les version jusqu'a la proposition de l'ancienne version de vscode (1.11.2). Il faut en rejeter beaucoup avant de l'avoir.

3 Premier pas :

Avant de commencer à développer vos propres codes, 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 de place il ira 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
```