

NEW - Prise en main Windows

ⓘ Ce tutoriel suit le processus d'installation de WSL, ZSH, OH-MY-ZSH, Anaconda/MiniConda, Poetry, Pyenv et la configuration de VSCode

CONFIGURATION UTILISÉE : Windows11

Index

1. [Introduction à WSL2](#)
2. [Installation de ZSH et Oh-My-ZSH](#)
3. [Installation de Pyenv](#)
4. [Installation de Poetry](#)
5. [Installation de Anaconda](#)
6. [Installation de Miniconda](#)
7. [Installation de VSCode et configuration avec WSL](#)

Introduction à WSL2

Pour utiliser les outils de Data science, comme par exemple les gestionnaires d'environnements python, etc il est préférable d'utiliser un environnement Linux/Unix.

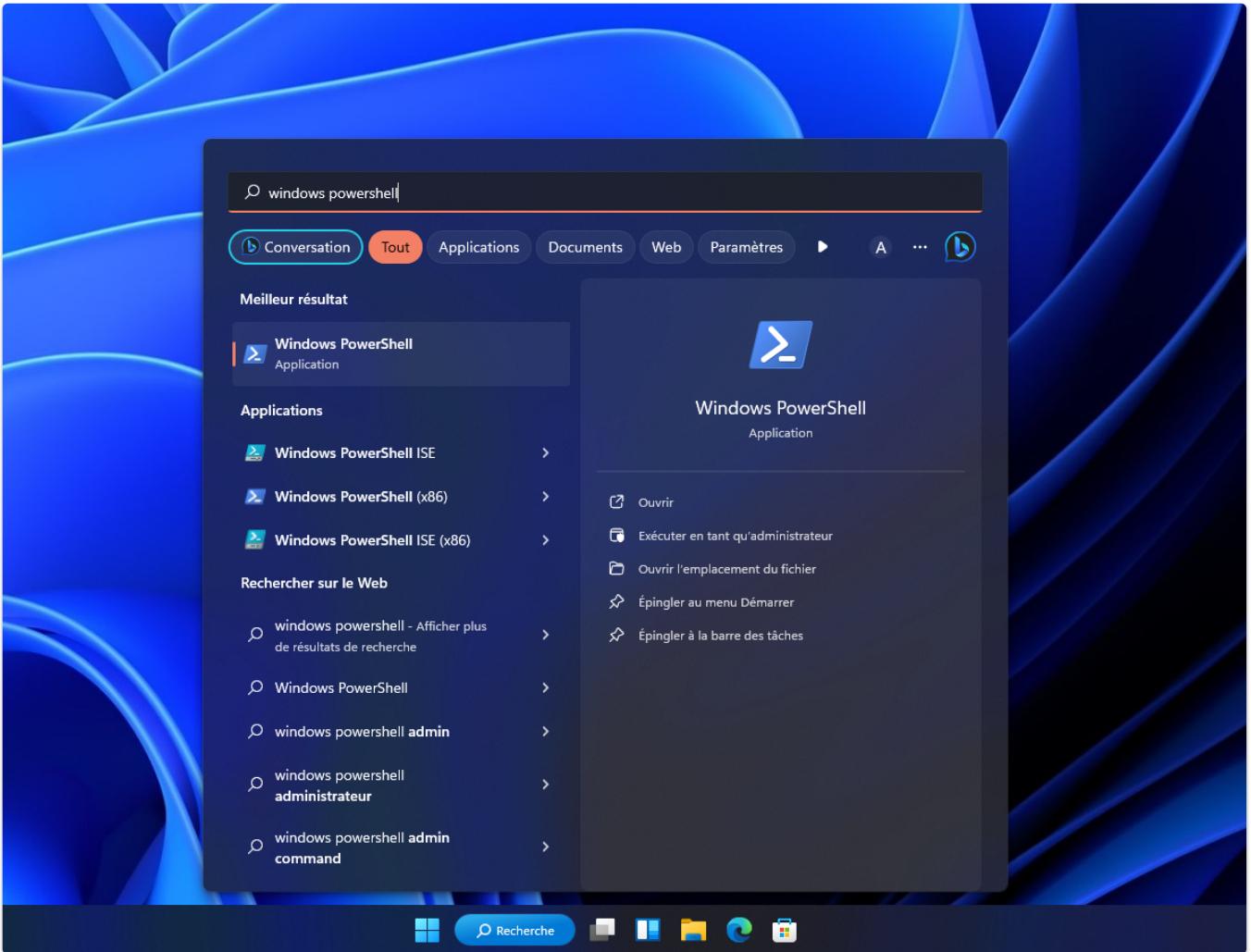
Pour les utilisateurs Windows, il existe ainsi WSL2, qui est un sous-système Windows pour Linux, qui permet d'exécuter un environnement Linux directement sur Windows sans passer par une machine virtuelle ou un dual boot.

(À chaque fois qu'on vous demandera "Voulez-vous autoriser cette application à apporter des modifications à votre appareil ?" dites "Oui")

Installation de WSL2

[Documentation officielle](#)

On commence par ouvrir PowerShell en mode administrateur. Pour ça, on clique sur le menu Démarrer (bouton windows), on cherche "Windows powershell" et on clique sur "Exécuter en tant qu'administrateur":



Une fois le Terminal ouvert, tapez `wsl --install`

```
Administrator : Windows PowerShell
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

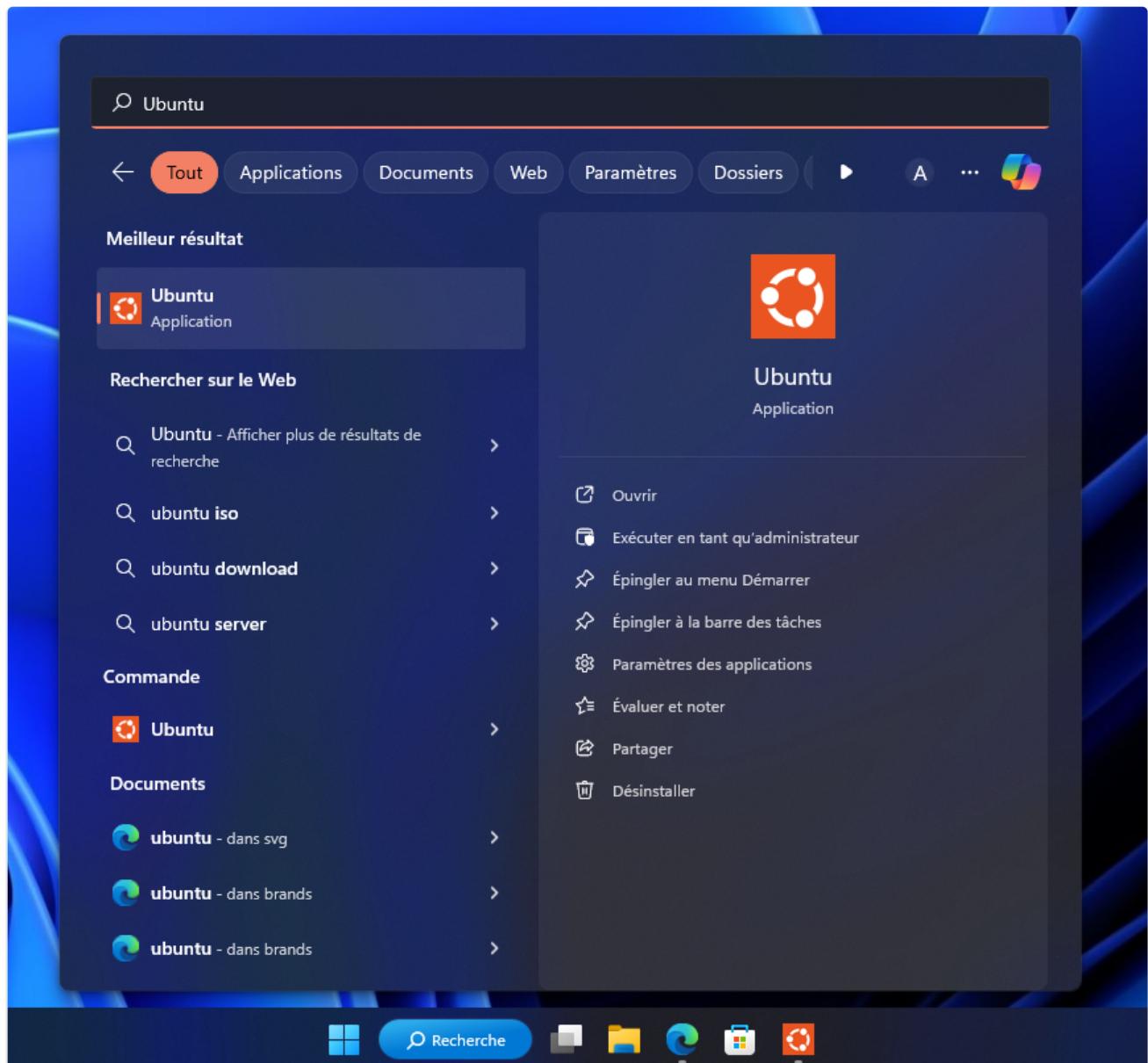
Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\WINDOWS\system32> wsl --install
Installation en cours : Plateforme de machine virtuelle
Plateforme de machine virtuelle a été installé.
Installation en cours : Sous-système Windows pour Linux
Sous-système Windows pour Linux a été installé.
Installation en cours : Ubuntu
Ubuntu a été installé.
L'opération demandée est réussie. Les modifications ne seront pas effectives avant que le système ne soit réamorcé.
PS C:\WINDOWS\system32>
```

Une fois l'installation terminée, Ubuntu a bien été installé (Ubuntu est la distribution de Linux), il faudra ensuite redémarrer votre machine.

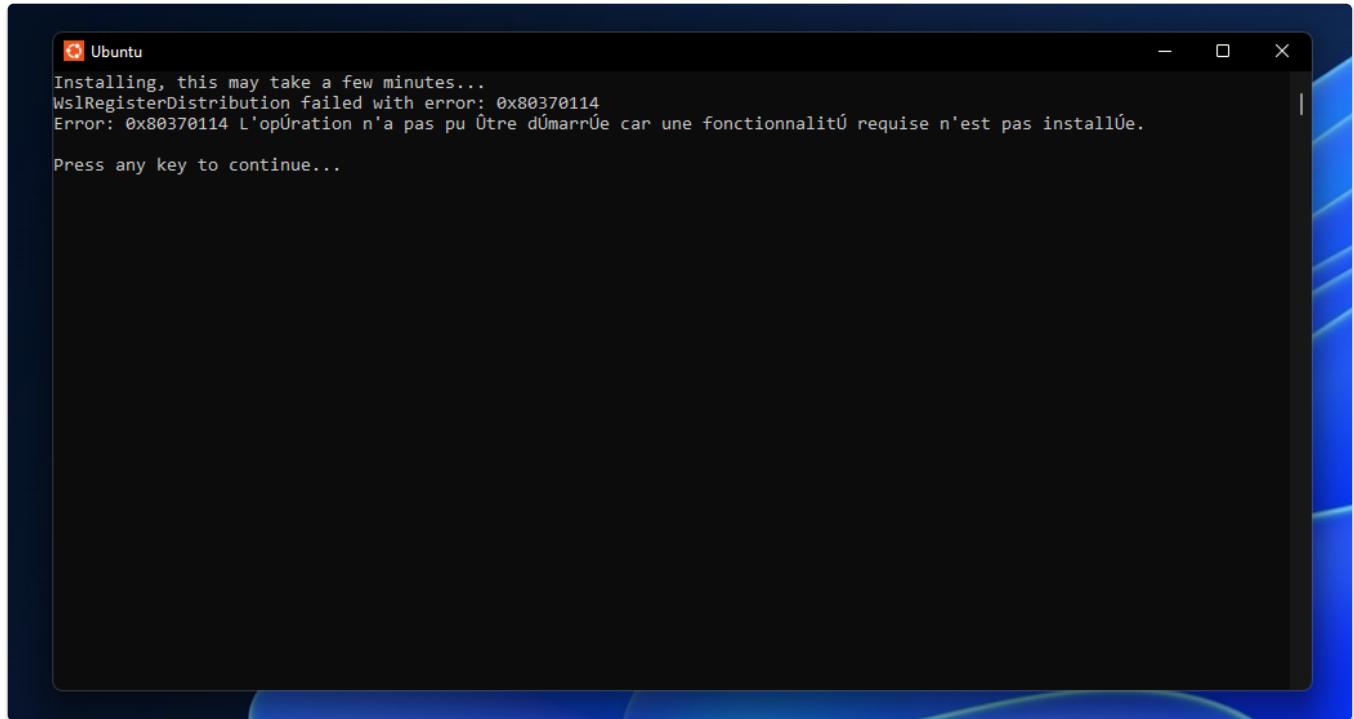
Après le redémarrage une fenêtre terminal s'ouvre pour confirmer que l'opération a réussi.

Il faut à présent ouvrir Ubuntu (depuis le menu démarrer):

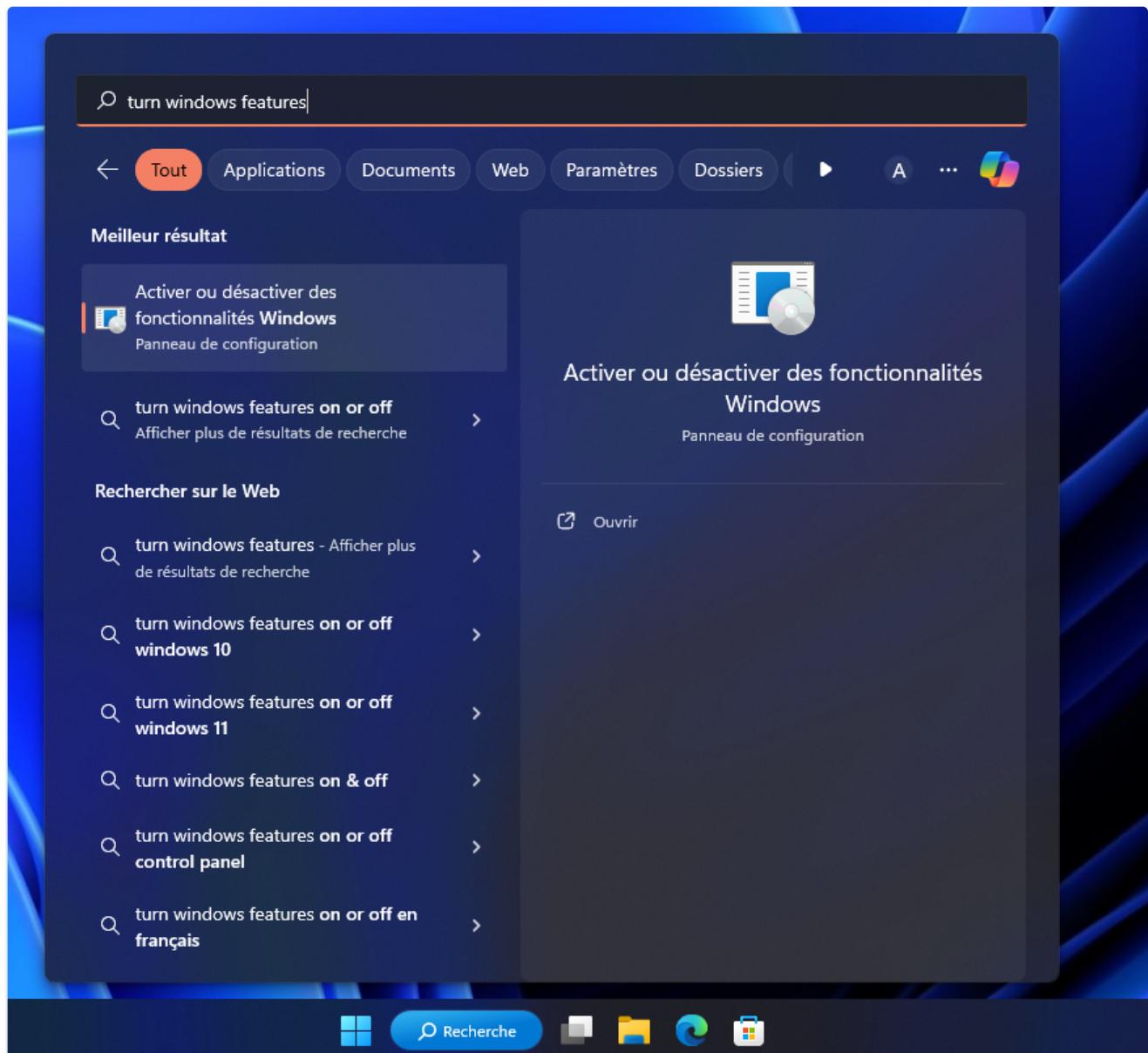


Des erreurs peuvent survenir à cette étape, par exemple :

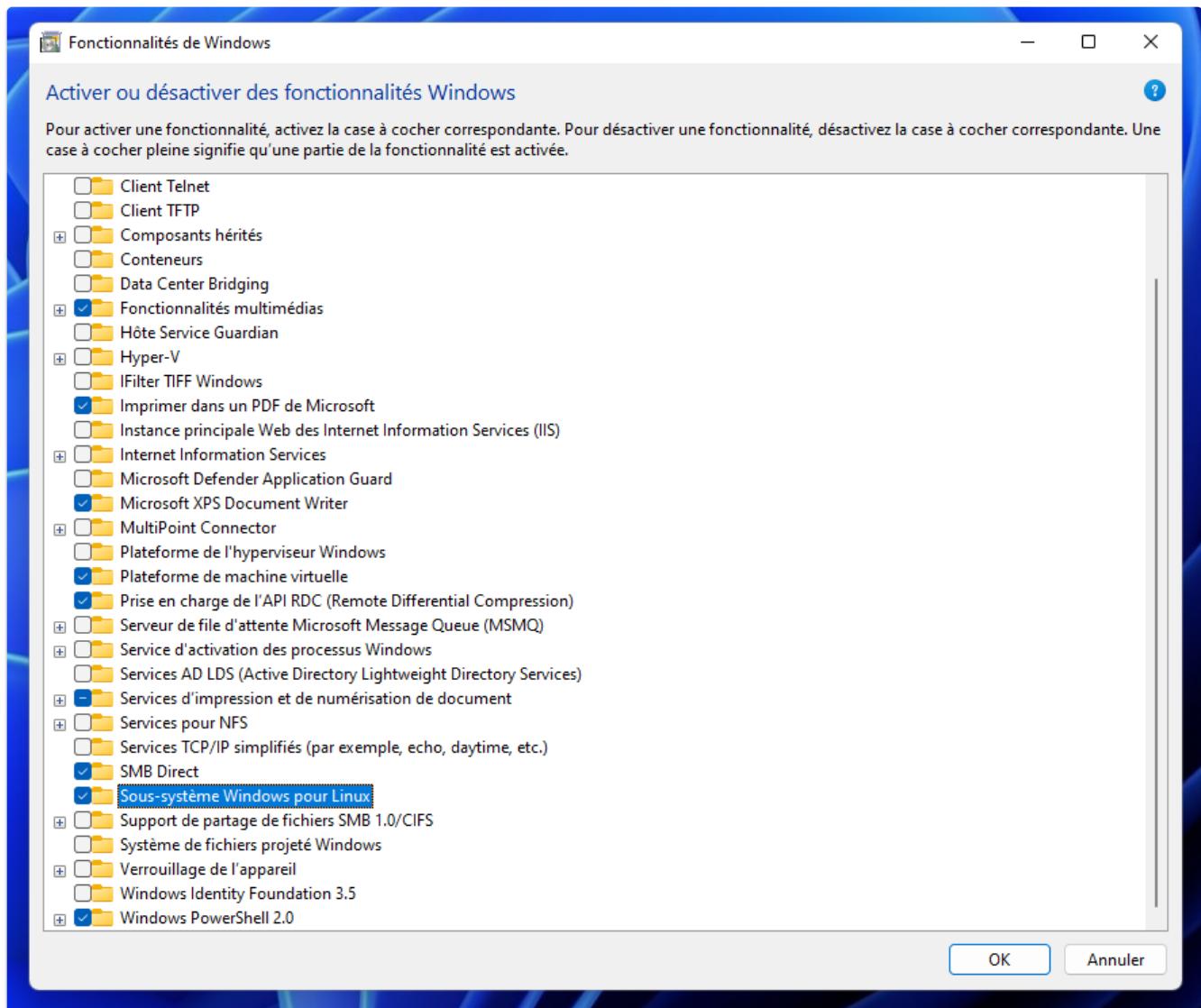
```
Wsl\RegisterDistribution failed with error: 0x80370114
```



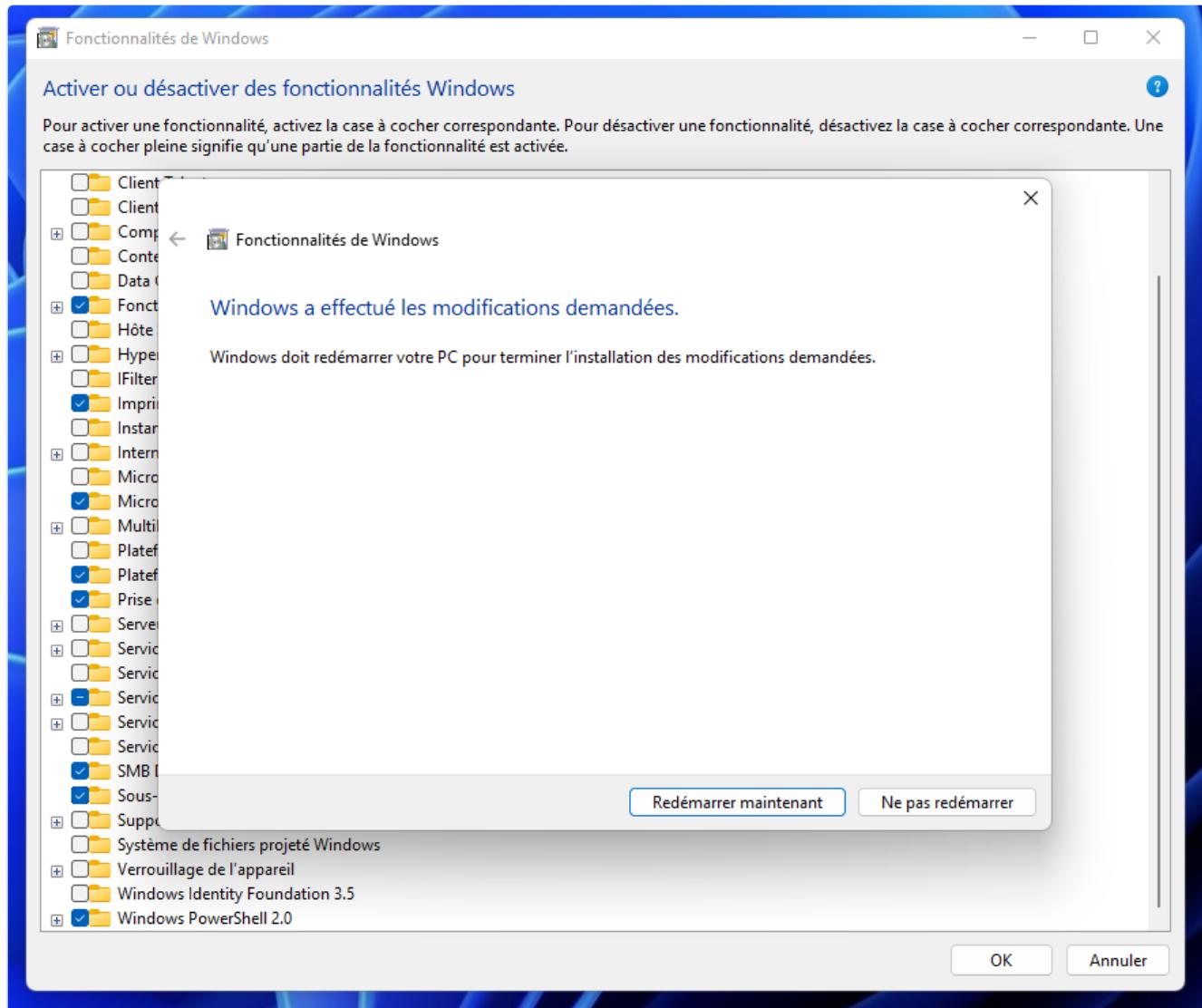
Dans ce cas, voici la [solution](#) pour le problème similaire sur le site microsoft. La solution consiste à ouvrir le menu démarrer et chercher "turn windows features" et d'ouvrir "Activer ou désactiver des fonctionnalités Windows" (Ou "Turn windows features on or off" en anglais)



Puis de chercher la ligne "Sous-système Windows pour Linux" (ou "Windows Subsystem for Linux" en anglais) et de cocher la case.



Ensuite, il faudra redémarrer votre machine.



Relancez à présent Ubuntu, depuis le menu démarrer, et cette fois-ci l'installation est bonne, et on vous demande à présent de configurer votre utilisateur.

Configuration de votre utilisateur Linux

On vous demande de créer un username:

```
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:
```

Puis un mot de passe (password) qu'il faudra entrer deux fois pour confirmer:

```
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: antonin
New password: -
```

Remarque: le mot de passe ne s'affiche pas pour des raisons de sécurité.

Vous êtes maintenant sur votre terminal Linux, avec votre utilisateur !

Le début de la ligne est composé de votre username, et du nom de votre machine, séparé par un @.

```
antonin@DESKTOP-1S65VP0: ~
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: antonin
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 24.04.1 LTS (GNU/Linux 5.15.167.4-microsoft-standard-WSL2 x86_64)

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

System information as of Tue Dec 24 11:40:08 CET 2024

System load: 1.89      Processes:          55
Usage of /:   0.1% of 1006.85GB  Users logged in:     0
Memory usage: 7%           IPv4 address for eth0: 172.27.73.19
Swap usage:   0%

This message is shown once a day. To disable it please create the
/home/antonin/.hushlogin file.
antonin@DESKTOP-1S65VP0:~$
```

(Pour nettoyer le terminal et ne plus afficher les messages précédents, vous pouvez utiliser la commande `clear`)

À chaque fois que vous voudrez ouvrir un terminal Linux, il faudra ouvrir Ubuntu depuis le menu démarrer.

Installation de ZSH et Oh-My-ZSH

Pour comprendre ce qu'est ZSH et Oh-My-ZSH, connaître leur spécificités et leurs améliorations qu'ils apporte, je vous renvoie aux vidéos dédiées dans la partie "Se familiariser avec le terminal".

ZSH

Dans votre terminal Linux, vous allez taper cette commande pour installer zsh, qui est un terminal plus avancé que bash :

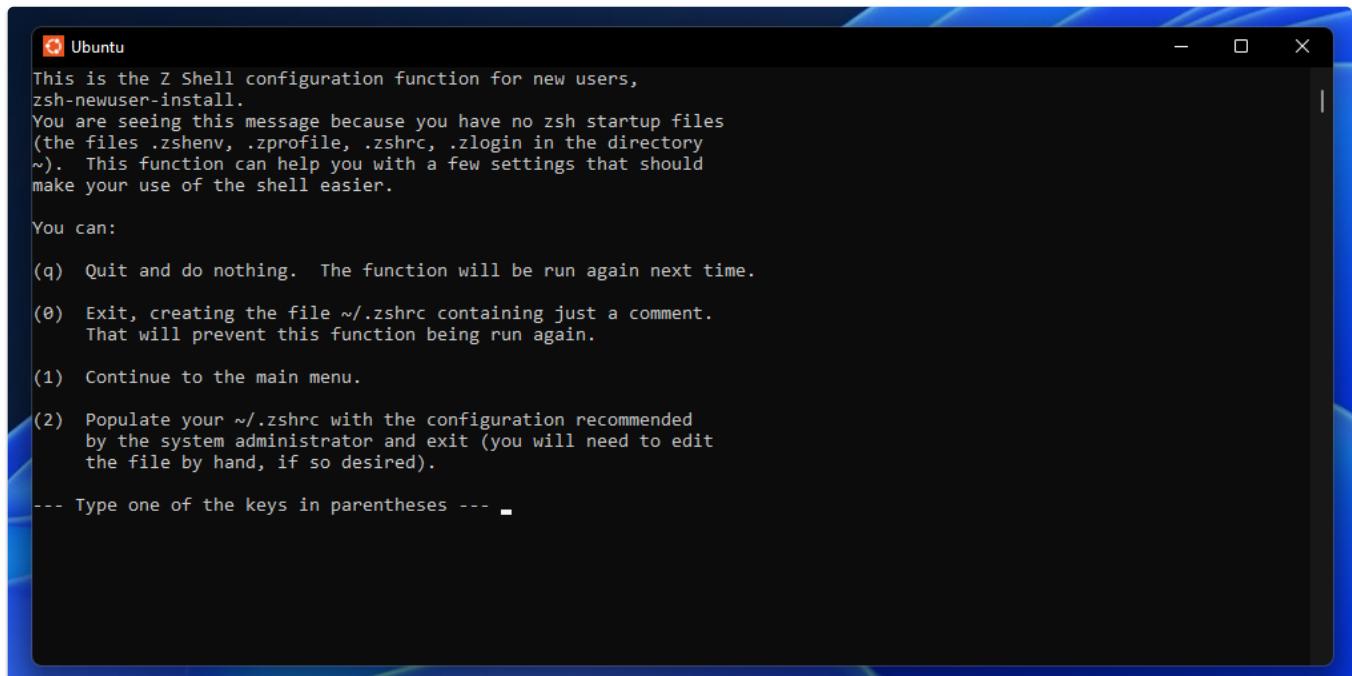
```
sudo apt update && sudo apt install zsh -y
```

Cette commande vous demandera le mot de passe que vous avez défini lors de la configuration de l'utilisateur Linux.

On va maintenant changer le shell par défaut en Zsh, pour qu'à chaque fois que vous lancez un nouveau terminal Linux, ce soit ZSH qui soit utilisé (commande à lancer sur le terminal Linux également) :

```
chsh -s $(which zsh)
```

Ensuite vous devez relancer le terminal, et il va vous demander de configurer votre environnement ZSH :



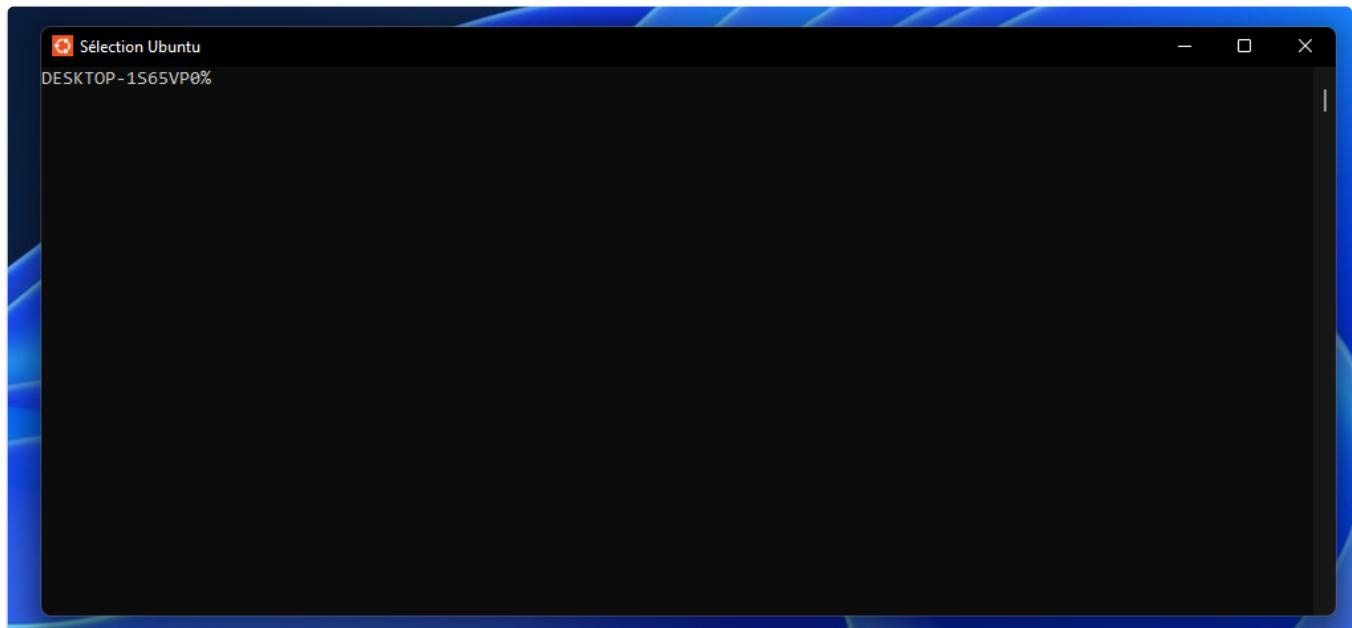
```
This is the Z Shell configuration function for new users,
zsh-newuser-install.
You are seeing this message because you have no zsh startup files
(the files .zshenv, .zprofile, .zshrc, .zlogin in the directory
~). This function can help you with a few settings that should
make your use of the shell easier.

You can:
(q) Quit and do nothing. The function will be run again next time.
(0) Exit, creating the file ~/.zshrc containing just a comment.
    That will prevent this function being run again.
(1) Continue to the main menu.
(2) Populate your ~/.zshrc with the configuration recommended
    by the system administrator and exit (you will need to edit
    the file by hand, if so desired).

--- Type one of the keys in parentheses --- -
```

Tapez `0` pour créer le fichier de configuration `.zshrc`

Vous êtes maintenant sur le terminal Linux ZSH:



```
Sélection Ubuntu
DESKTOP-1S65VP0%
```

ⓘ Pour savoir si votre terminal actuel utilise bash ou zsh, tapez la commande `echo $SHELL`.

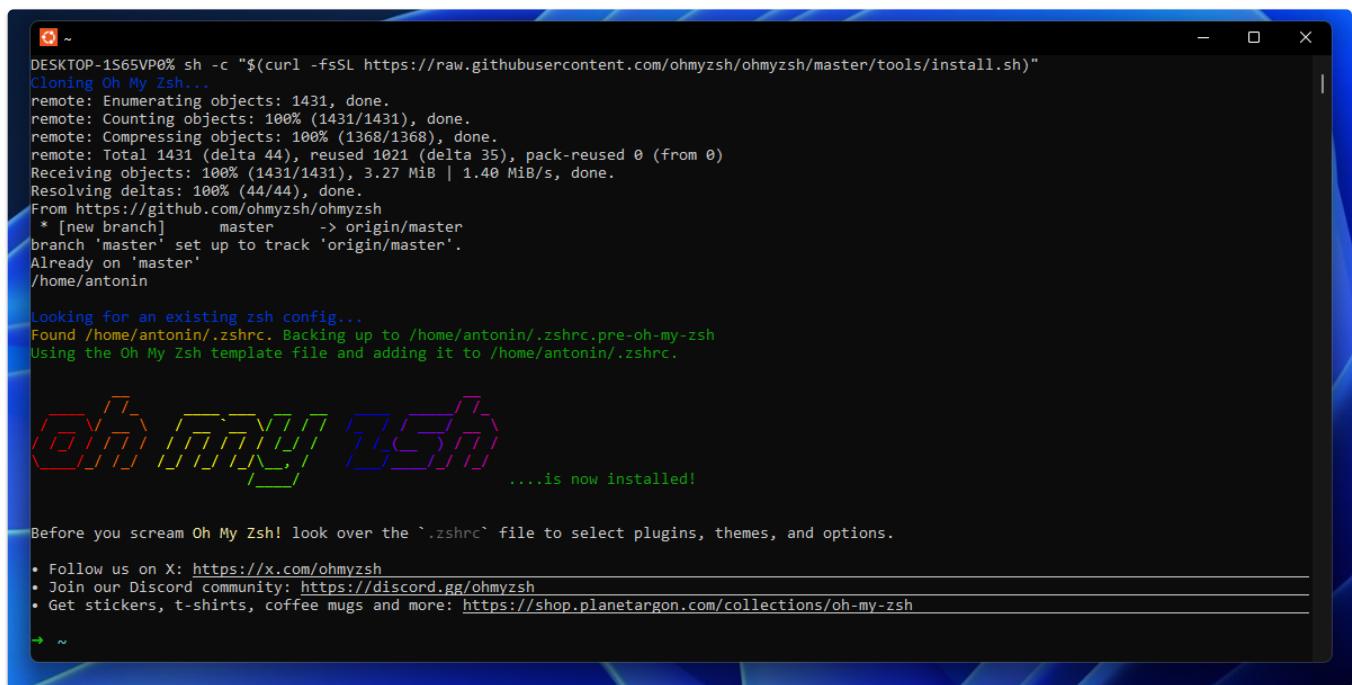
Cela affichera quelque chose comme ça pour bash : `usr/bin/bash` et quelque chose comme ça pour zsh : `usr/bin/zsh`

OH-MY-ZSH

Documentation officielle

Installons maintenant oh-my-zsh (depuis le terminal Linux):

```
sh -c "$(curl -fsSL  
https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.  
sh)"
```



```
DESKTOP-1S65VP0% sh -c "$(curl -fsSL https://raw.githubusercontent.com/ohmyzsh/ohmyzsh/master/tools/install.sh)"  
Cloning Oh My Zsh...  
remote: Enumerating objects: 1431, done.  
remote: Counting objects: 100% (1431/1431), done.  
remote: Compressing objects: 100% (1368/1368), done.  
remote: Total 1431 (delta 44), reused 1021 (delta 35), pack-reused 0 (from 0)  
Receiving objects: 100% (1431/1431), 3.27 MiB | 1.40 MiB/s, done.  
Resolving deltas: 100% (44/44), done.  
From https://github.com/ohmyzsh/ohmyzsh  
 * [new branch]      master    -> origin/master  
branch 'master' set up to track 'origin/master'.  
Already on 'master'  
/home/antonin  
  
Looking for an existing zsh config...  
Found /home/antonin/.zshrc. Backing up to /home/antonin/.zshrc.pre-oh-my-zsh  
Using the Oh My Zsh template file and adding it to /home/antonin/.zshrc.  
  
██████████ ██████████ ██████████ ....is now installed!  
  
Before you scream Oh My Zsh! look over the `~/.zshrc` file to select plugins, themes, and options.  
• Follow us on X: https://x.com/ohmyzsh  
• Join our Discord community: https://discord.gg/ohmyzsh  
• Get stickers, t-shirts, coffee mugs and more: https://shop.planetargon.com/collections/oh-my-zsh
```

Cela va ajouter la configuration OH-MY-ZSH dans le fichier `.zshrc`.

Maintenant, dès que vous ouvrirez le terminal Linux, vous aurez un terminal ZSH avec OH-MY-ZSH.

⌚ Important

Pour éviter des erreurs de manipulations, faites une sauvegarde du fichier `.zshrc` dans un autre fichier (à chaque fois que vos modifications sont stables et vérifiées), par exemple dans le fichier `.zshrc_backup` en utilisant la commande suivante: `cp .zshrc .zshrc_backup` (depuis le terminal linux)

Cela permettra de revenir à un état stable du fichier

Plugins oh-my-zsh

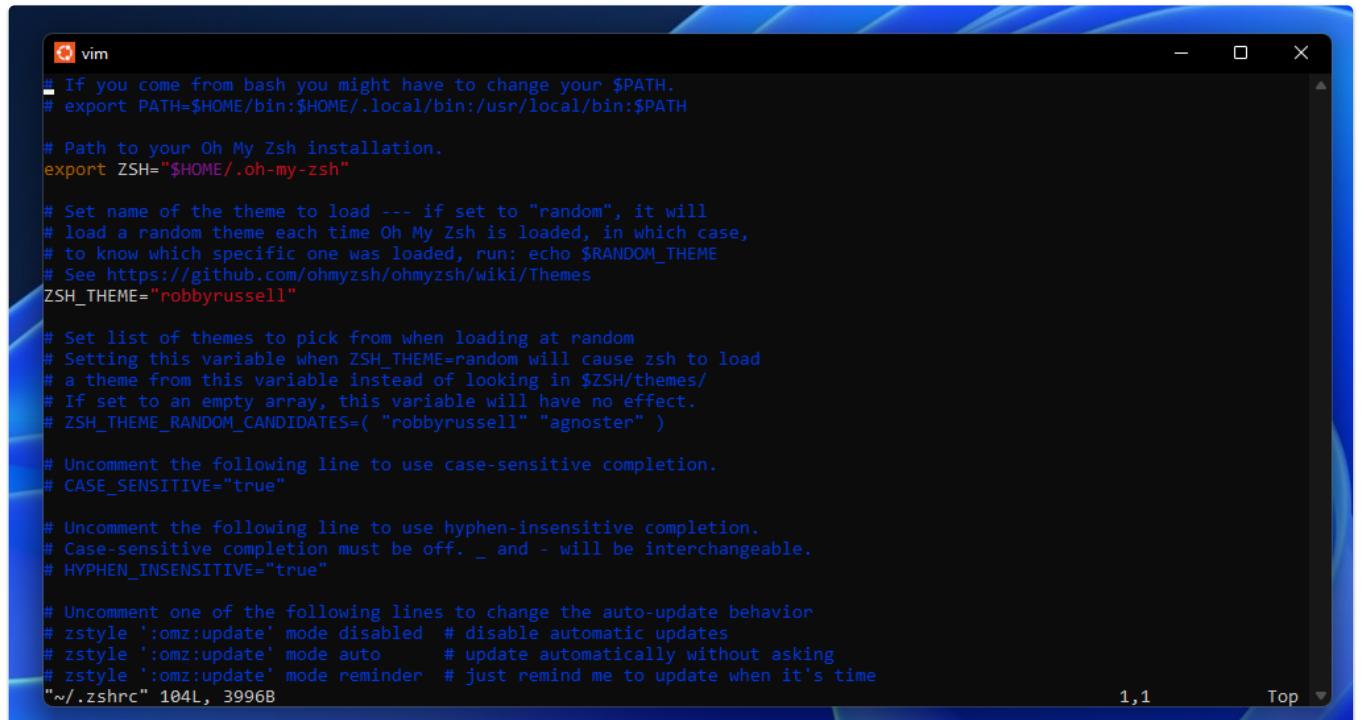
Oh-my-zsh vient avec un grand nombre de plugins très utiles, comme par exemple pour git, vscode, poetry ou conda.

Pour ajouter un ou plusieurs plugins on va modifier le fichier .zshrc qui contient notamment la configuration de oh-my-zsh.

Ouvrez le fichier .zshrc avec Vim depuis le terminal Linux (Pour plus de détails sur l'utilisation de VIM, je vous renvoie à la vidéo dédiée) :

```
vim ~/.zshrc
```

Le fichier va s'ouvrir dans votre terminal :



The screenshot shows a terminal window titled "vim" displaying the contents of the .zshrc file. The file contains configuration code for Oh My Zsh, including setting the PATH, defining the ZSH variable, setting the theme to "robbyrussell", and configuring various completion settings. The terminal window has a blue header bar and a dark background. The status bar at the bottom right shows "1,1" and "Top".

```
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"

# Set name of the theme to load --- if set to "random", it will
# load a random theme each time Oh My Zsh is loaded, in which case,
# to know which specific one was loaded, run: echo $RANDOM_THEME
# See https://github.com/ohmyzsh/ohmyzsh/wiki/Themes
ZSH_THEME="robbyrussell"

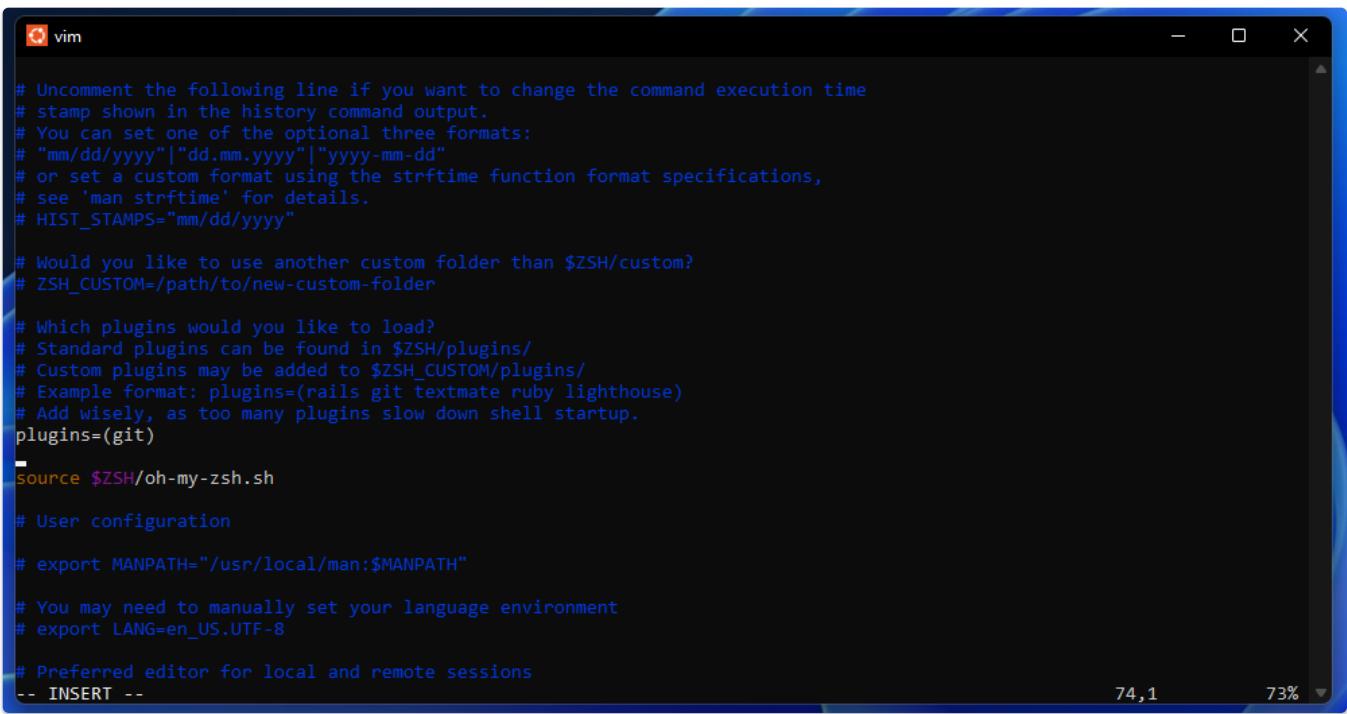
# Set list of themes to pick from when loading at random
# Setting this variable when ZSH_THEME=random will cause zsh to load
# a theme from this variable instead of looking in $ZSH/themes/
# If set to an empty array, this variable will have no effect.
# ZSH_THEME_RANDOM_CANDIDATES=( "robbyrussell" "agnoster" )

# Uncomment the following line to use case-sensitive completion.
# CASE_SENSITIVE="true"

# Uncomment the following line to use hyphen-insensitive completion.
# Case-sensitive completion must be off. _ and - will be interchangeable.
# HYPHEN_INSENSITIVE="true"

# Uncomment one of the following lines to change the auto-update behavior
# zstyle ':omz:update' mode disabled # disable automatic updates
# zstyle ':omz:update' mode auto      # update automatically without asking
# zstyle ':omz:update' mode reminder # just remind me to update when it's time
"~/zshrc" 104L, 3996B
```

Ajoutez les plugins suivants dans la section des plugins (il faut un peu descendre dans le fichier .zshrc en utilisant les flèches) en passant en mode insertion en appuyant sur **i** (Vous savez que vous êtes en mode insertion s'il est écrit **-- Insert --** tout en bas de la fenêtre)



```
# Uncomment the following line if you want to change the command execution time
# stamp shown in the history command output.
# You can set one of the optional three formats:
# "mm/dd/yyyy"|"dd.mm.yyyy"|"yyyy-mm-dd"
# or set a custom format using the strftime function format specifications,
# see 'man strftime' for details.
# HIST_STAMPS="mm/dd/yyyy"

# Would you like to use another custom folder than $ZSH/custom?
# ZSH_CUSTOM=/path/to/new-custom-folder

# Which plugins would you like to load?
# Standard plugins can be found in $ZSH/plugins/
# Custom plugins may be added to $ZSH_CUSTOM/plugins/
# Example format: plugins=(rails git textmate ruby lighthouse)
# Add wisely, as too many plugins slow down shell startup.
plugins=(git)

source $ZSH/oh-my-zsh.sh

# User configuration

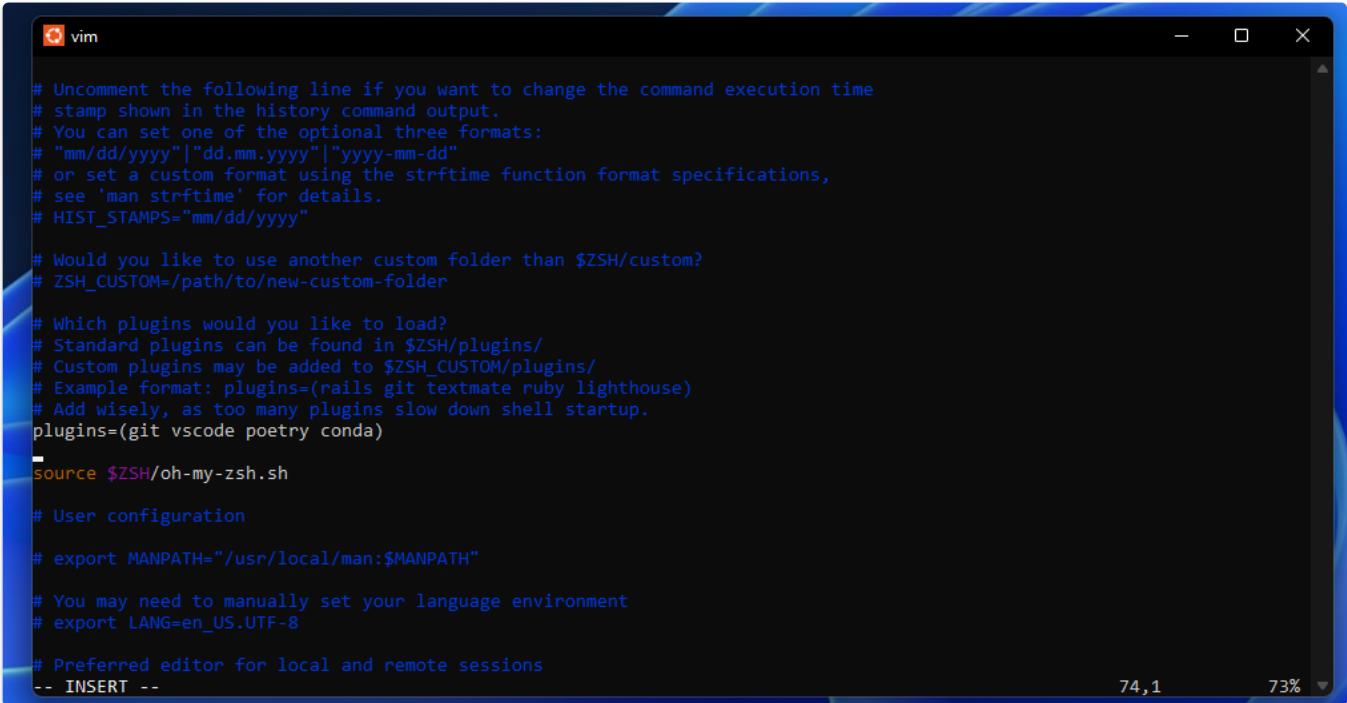
# export MANPATH="/usr/local/man:$MANPATH"

# You may need to manually set your language environment
# export LANG=en_US.UTF-8

# Preferred editor for local and remote sessions
-- INSERT --
```

Modifiez la ligne des plugins avec celle-ci:

```
plugins=(git vscode poetry conda)
```



```
# Uncomment the following line if you want to change the command execution time
# stamp shown in the history command output.
# You can set one of the optional three formats:
# "mm/dd/yyyy"|"dd.mm.yyyy"|"yyyy-mm-dd"
# or set a custom format using the strftime function format specifications,
# see 'man strftime' for details.
# HIST_STAMPS="mm/dd/yyyy"

# Would you like to use another custom folder than $ZSH/custom?
# ZSH_CUSTOM=/path/to/new-custom-folder

# Which plugins would you like to load?
# Standard plugins can be found in $ZSH/plugins/
# Custom plugins may be added to $ZSH_CUSTOM/plugins/
# Example format: plugins=(rails git textmate ruby lighthouse)
# Add wisely, as too many plugins slow down shell startup.
plugins=(git vscode poetry conda)

source $ZSH/oh-my-zsh.sh

# User configuration

# export MANPATH="/usr/local/man:$MANPATH"

# You may need to manually set your language environment
# export LANG=en_US.UTF-8

# Preferred editor for local and remote sessions
-- INSERT --
```

Enregistrez et fermez Vim : Appuyez sur `Esc` pour quitter le mode insertion (la mention `- Insert --` disparait), tapez `:wq`, puis appuyez sur Entrée.

Si vous voulez annuler les modifications que vous êtes en train de faire, faites `:q!` pour quitter sans enregistrer.

Vous pouvez relancer le terminal pour que la configuration de OH-MY-ZSH soit prise en compte.

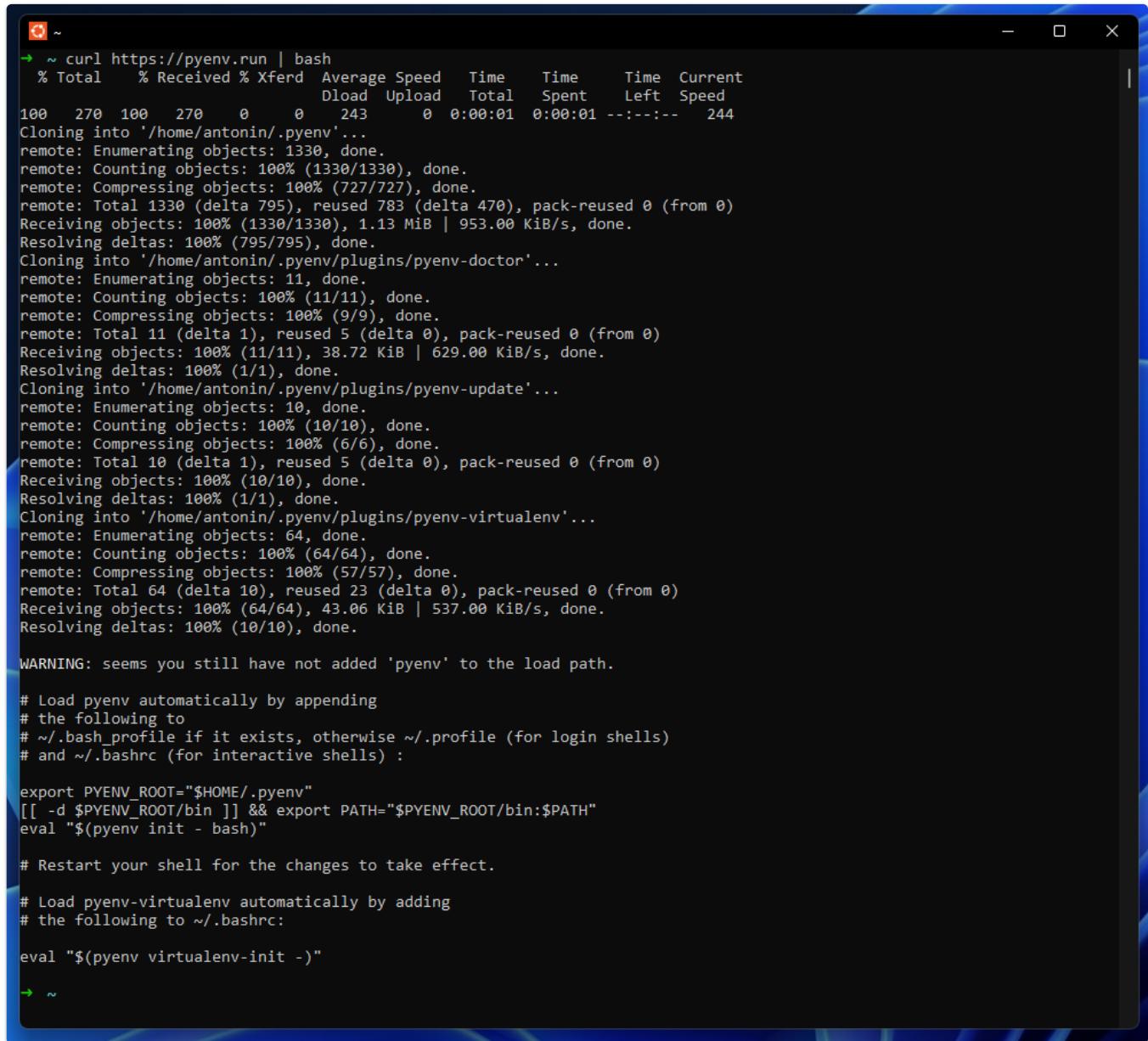
Installation de Pyenv

Toutes les commandes devront être utilisées dans le terminal Linux.

Documentation officielle

On commence par l'installation de pyenv:

```
curl https://pyenv.run | bash
```



```
curl https://pyenv.run | bash
% Total    % Received % Xferd  Average Speed   Time     Time      Current
          Dload  Upload   Total   Spent   Left  Speed
100  270  100  270    0     0  243      0  0:00:01  0:00:01  ---:---  244
Cloning into '/home/antonin/.pyenv'...
remote: Enumerating objects: 1330, done.
remote: Counting objects: 100% (1330/1330), done.
remote: Compressing objects: 100% (727/727), done.
remote: Total 1330 (delta 795), reused 783 (delta 470), pack-reused 0 (from 0)
Receiving objects: 100% (1330/1330), 1.13 MiB | 953.00 KiB/s, done.
Resolving deltas: 100% (795/795), done.
Cloning into '/home/antonin/.pyenv/plugins/pyenv-doctor'...
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (9/9), done.
remote: Total 11 (delta 1), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (11/11), 38.72 KiB | 629.00 KiB/s, done.
Resolving deltas: 100% (1/1), done.
Cloning into '/home/antonin/.pyenv/plugins/pyenv-update'...
remote: Enumerating objects: 10, done.
remote: Counting objects: 100% (10/10), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 10 (delta 1), reused 5 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (10/10), done.
Resolving deltas: 100% (1/1), done.
Cloning into '/home/antonin/.pyenv/plugins/pyenv-virtualenv'...
remote: Enumerating objects: 64, done.
remote: Counting objects: 100% (64/64), done.
remote: Compressing objects: 100% (57/57), done.
remote: Total 64 (delta 10), reused 23 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (64/64), 43.06 KiB | 537.00 KiB/s, done.
Resolving deltas: 100% (10/10), done.

WARNING: seems you still have not added 'pyenv' to the load path.

# Load pyenv automatically by appending
# the following to
# ~/.bash_profile if it exists, otherwise ~/.profile (for login shells)
# and ~/.bashrc (for interactive shells) :

export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - bash)"

# Restart your shell for the changes to take effect.

# Load pyenv-virtualenv automatically by adding
# the following to ~/.bashrc:

eval "$(pyenv virtualenv-init -)"

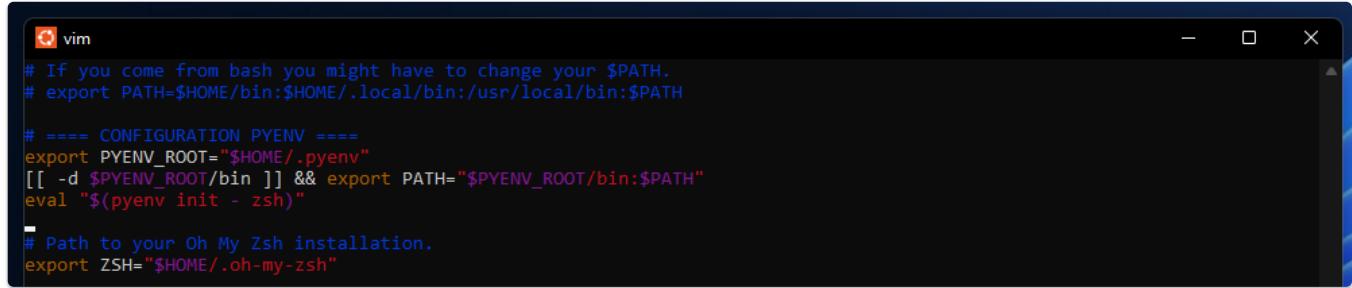
→ ~
```

Ensuite on ouvre le fichier `.zshrc` en utilisant vim:

```
vim ~/.zshrc
```

On passe en mode insertion en appuyant sur `i` et ici j'ajoute une section `# =====`
`CONFIGURATION PYENV =====` en ajoutant les lignes suivantes :

```
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"
```



A screenshot of a terminal window titled "vim". The code shown is the configuration for pyenv, specifically the ".zshrc" file. It includes comments about changing the PATH if coming from bash, setting the pyenv root, and evaluating the pyenv init command. It also defines the ZSH variable to point to the oh-my-zsh directory.

```
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# ===== CONFIGURATION PYENV =====
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"
```

Puis on sort du mode insertion avec Echap et on appuie sur :wq

Reste plus qu'à recharger le terminal pour prendre en compte la nouvelle configuration de Pyenv et du PATH:

```
exec "$SHELL"
```

À présent on va installer les dépendances pour Pyenv, avec les commandes suivantes:

```
sudo apt update; sudo apt install build-essential libssl-dev zlib1g-dev \
\ libbz2-dev libreadline-dev libsdl2-dev curl git \
libncursesw5-dev xz-utils tk-dev libxml2-dev libxmlsec1-dev libffi-dev
```

```
sudo apt install liblzma-dev
```

Testez si Pyenv est bien installé avec la commande :

```
pyenv --version
```

Qui retournera la version de Pyenv.

Installation de Poetry

Toutes les commandes devront être utilisées dans le terminal Linux.

[Documentation officielle](#)

On installe Poetry avec la commande suivante:

```
curl -sSL https://install.python-poetry.org | python3 -
```

```
curl -sSL https://install.python-poetry.org | python3 -
Retrieving Poetry metadata

# Welcome to Poetry!

This will download and install the latest version of Poetry,
a dependency and package manager for Python.

It will add the `poetry` command to Poetry's bin directory, located at:

/home/antonin/.local/bin

You can uninstall at any time by executing this script with the --uninstall option,
and these changes will be reverted.

Installing Poetry (1.8.5): Done

Poetry (1.8.5) is installed now. Great!

To get started you need Poetry's bin directory (/home/antonin/.local/bin) in your `PATH`
environment variable.

Add `export PATH="/home/antonin/.local/bin:$PATH"` to your shell configuration file.

Alternatively, you can call Poetry explicitly with `/home/antonin/.local/bin/poetry`.

You can test that everything is set up by executing:

`poetry --version`

→ ~
```

Puis il faut ajouter poetry au PATH.

Pour cela on ouvre le fichier `.zshrc` avec VIM:

```
vim ~/.zshrc
```

Puis, après avoir appuyé sur `i` pour passer en mode insertion, on ajoute cette ligne dans le fichier `.zshrc`:

```
export PATH="$HOME/.local/bin:$PATH"
```

Ici, j'ai ajouté une section `# ===== CONFIGURATION POETRY =====`

```
vim

# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# ===== CONFIGURATION PYENV =====
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"

# ===== CONFIGURATION POETRY =====
export PATH="$HOME/.local/bin:$PATH"

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"
```

Puis vous tapez Echap puis `:wq` pour enregistrer le fichier.

Et ensuite `exec "$SHELL"` pour appliquer la modification dans le terminal actuel.

Si l'installation et l'ajout au PATH a réussi, la commande `poetry --version` devrait vous donner la version de Poetry.

⚡ Important

Pour les deux parties qui suivent, sur Anaconda et Miniconda, il faudra choisir quelle distribution choisir entre les deux. Pour cela, je vous renvoie à la vidéo sur la comparaison de Anaconda/Miniconda/Conda

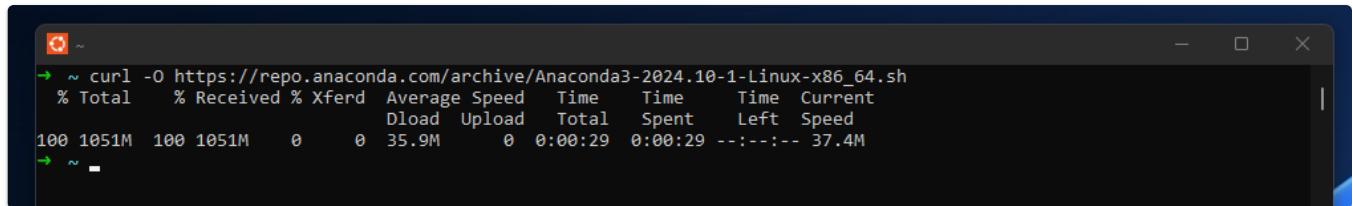
Installation de Anaconda

Toutes les commandes devront être utilisées dans le terminal Linux.

[Documentation officielle](#)

On commence par télécharger la dernière version de Anaconda:

```
curl -O https://repo.anaconda.com/archive/Anaconda3-2024.10-1-Linux-x86_64.sh
```



Puis on installe Anaconda en utilisant le fichier téléchargé précédemment:

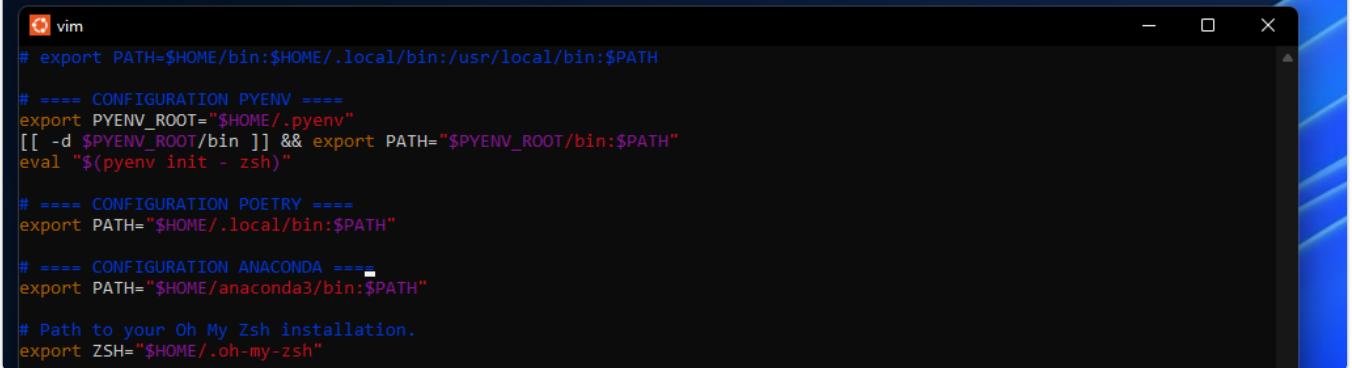
```
bash ~/Anaconda3-2024.10-1-Linux-x86_64.sh
```

Il faudra appuyer sur Entrée pour afficher la licence, puis il faudra aller tout en bas et taper yes . Il faudra une nouvelle fois appuyer sur Entrée pour confirmer l'installation à l'endroit indiqué. L'installation se lance, et ensuite on vous demande "Do you wish to update your shell profile to automatically initialize conda?". Il faudra dire yes .

Une fois terminée, le message Thank you for installing Anaconda3 ! apparaît et il faudra redémarrer le terminal Linux.

On ajoute ensuite anaconda3 dans le PATH en ajoutant la ligne suivant dans le fichier .zshrc :

```
export PATH="$HOME/anaconda3/bin:$PATH"
```



```
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH
# === CONFIGURATION PYENV ===
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"

# === CONFIGURATION POETRY ===
export PATH="$HOME/.local/bin:$PATH"

# === CONFIGURATION ANACONDA ===
export PATH="$HOME/anaconda3/bin:$PATH"

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"
```

Et on met à jour la configuration du terminal actuel avec `exec "$SHELL"`.

Si tout est bien installé, vous devriez voir la version de conda avec la commande :

```
conda --version
```

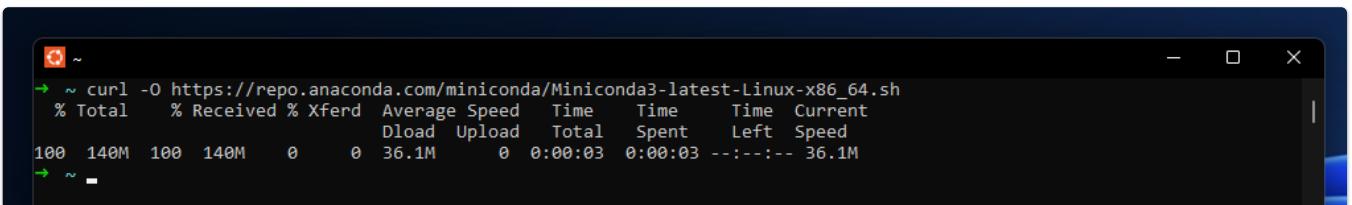
Installation de Miniconda

Toutes les commandes devront être utilisées dans le terminal Linux.

[Documentation officielle](#)

On commence par télécharger la dernière version de Miniconda :

```
curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```



```
curl -O https://repo.anaconda.com/miniconda/Miniconda3-latest-Linux-x86_64.sh
```

Puis on installe Miniconda en utilisant le fichier téléchargé précédemment:

```
bash ~/Miniconda3-latest-Linux-x86_64.sh
```

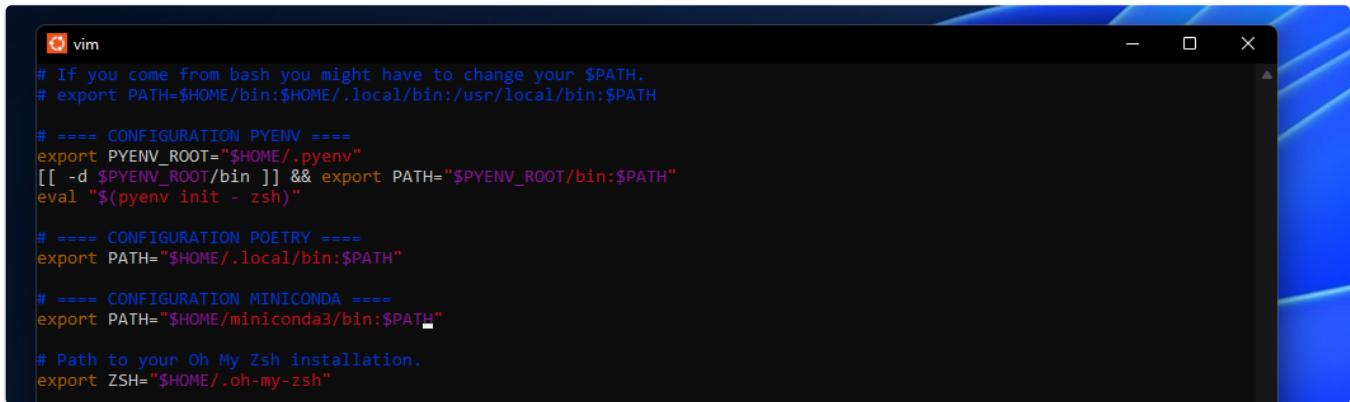
Il faudra appuyer sur `Entrée` pour afficher la licence, puis il faudra aller tout en bas et taper `yes`. Il faudra une nouvelle fois appuyer sur `Entrée` pour confirmer l'installation à l'endroit indiqué. L'installation se lance, et ensuite on vous demande "Do you wish to update your shell profile to automatically initialize conda?". Il faudra dire `yes`.

Une fois terminée, le message `Thank you for installing Miniconda3!` apparait et il faudra redémarrer le terminal Linux.

On ajoute ensuite miniconda3 dans le PATH en ajoutant la ligne suivant dans le fichier

.zshrc :

```
export PATH="$HOME/miniconda3/bin:$PATH"
```



```
# If you come from bash you might have to change your $PATH.
# export PATH=$HOME/bin:$HOME/.local/bin:/usr/local/bin:$PATH

# ---- CONFIGURATION PYENV ----
export PYENV_ROOT="$HOME/.pyenv"
[[ -d $PYENV_ROOT/bin ]] && export PATH="$PYENV_ROOT/bin:$PATH"
eval "$(pyenv init - zsh)"

# ---- CONFIGURATION POETRY ----
export PATH="$HOME/.local/bin:$PATH"

# ---- CONFIGURATION MINICONDA ----
export PATH="$HOME/miniconda3/bin:$PATH"

# Path to your Oh My Zsh installation.
export ZSH="$HOME/.oh-my-zsh"
```

Et on met à jour la configuration du terminal actuel avec `exec "$SHELL"`.

Si tout est bien installé, vous devriez voir la version de conda avec la commande :

```
conda --version
```

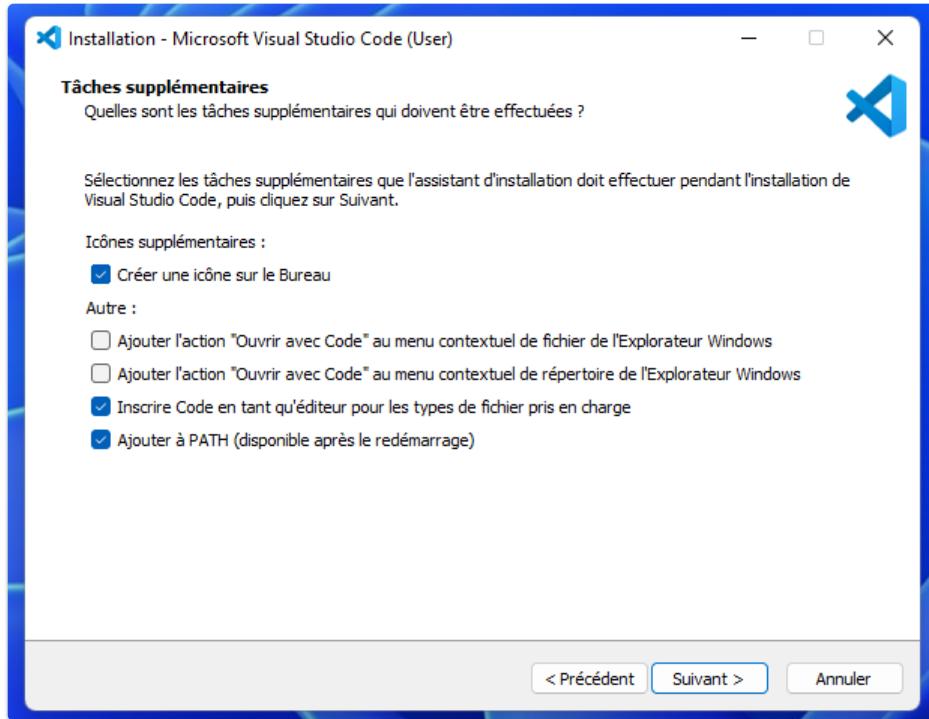
Installation de VSCode et configuration avec WSL

[Documentation Officielle](#)

Concernant VSCode, on va l'installer sur Windows, depuis la [page de téléchargement de VSCode](#).

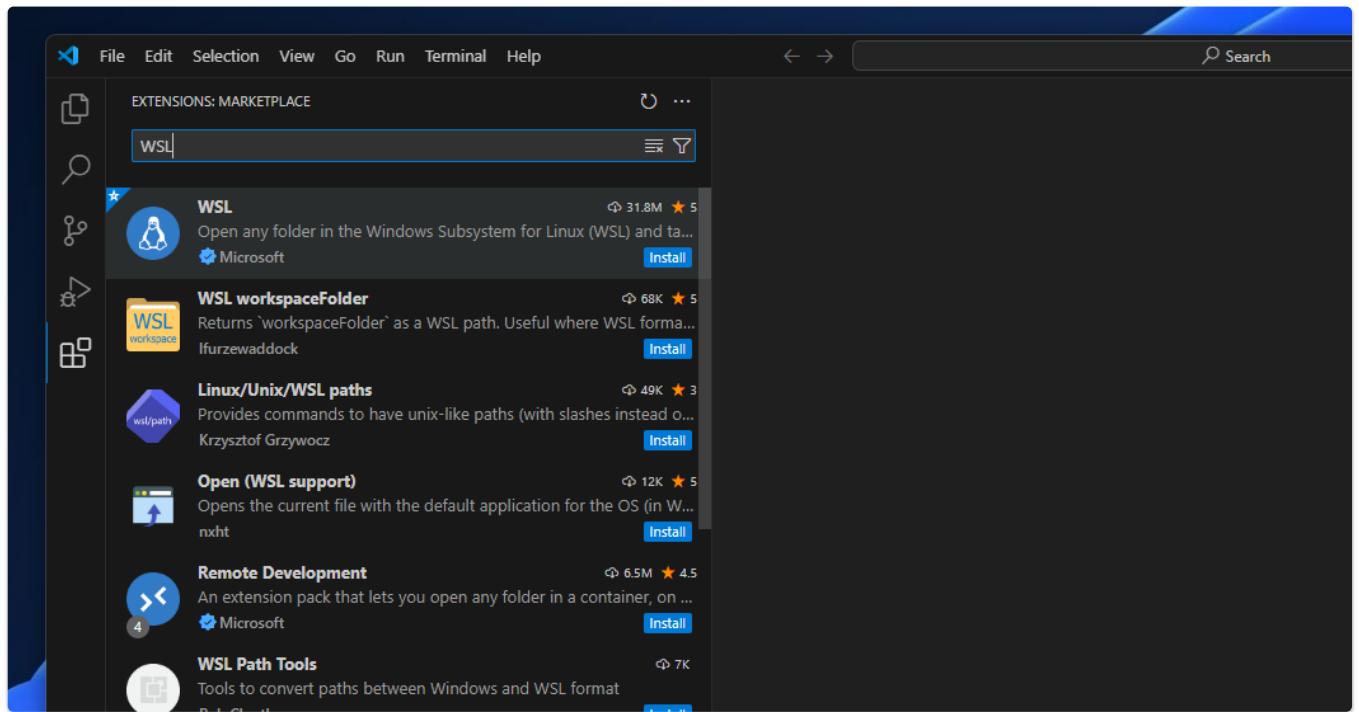
On va choisir la version Windows. On la télécharge, on l'installe, et on l'ouvre directement depuis le menu Démarrer.

On veillera à cocher "Ajouter au PATH" lors de l'installation:



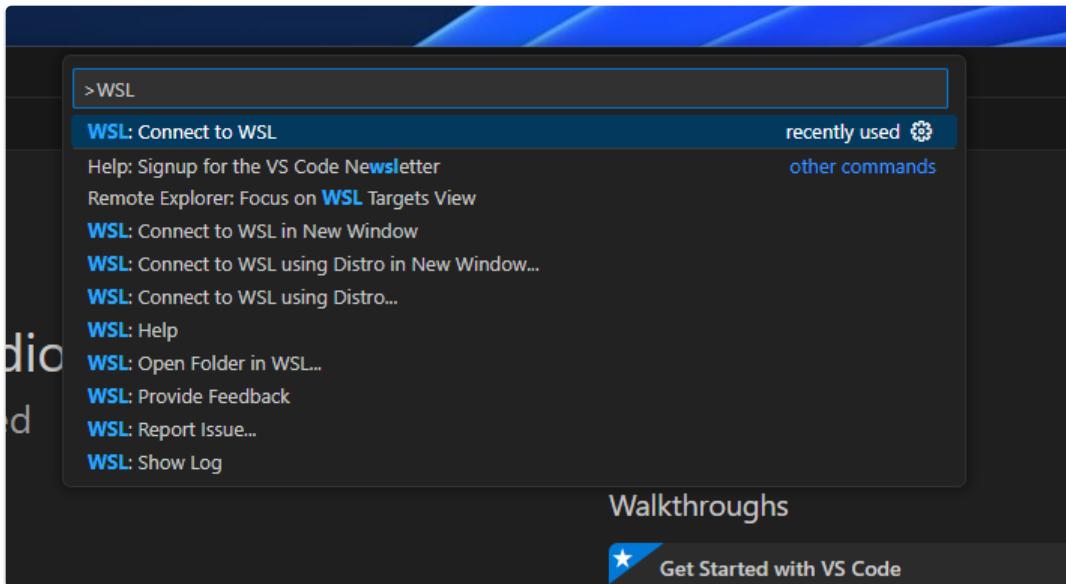
Extension WSL

On commence par installer l'extension WSL depuis VSCode (à gauche dans extensions) en cherchant `WSL` (l'éditeur de cette extension est Microsoft).



On va maintenant pouvoir connecter VSCode avec WSL.

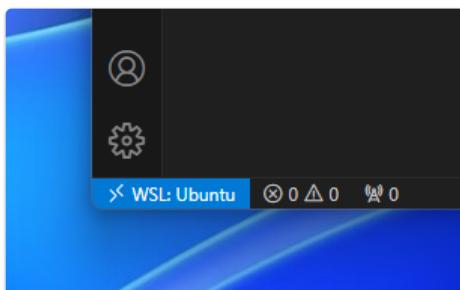
Sans manipulation supplémentaire, VSCode a uniquement accès aux fichiers qui sont sur votre système Windows, pour le connecter à WSL et accéder aux fichiers de votre environnement Linux vous allez faire `F1` puis tapez `WSL`



Vous allez cliquer sur `WSL: Connect to WSL` et VSCode va alors se connecter à WSL :

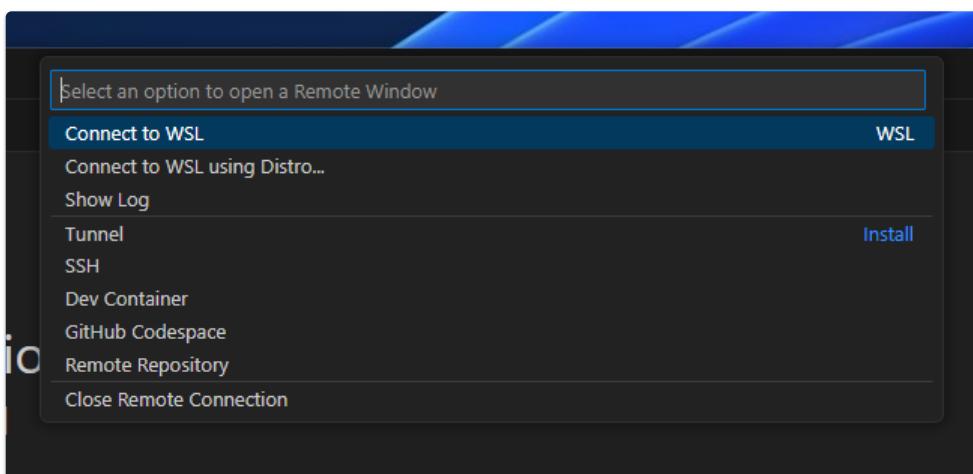


Lorsque tout est bon, vous verrez affiché `WSL: Ubuntu` en bas à gauche de la fenêtre VSCode:



À présent VSCode peut accéder aux fichiers qui sont dans l'environnement Linux.

Vous pouvez fermer la connexion avec WSL en cliquant en bas à gauche sur `WSL: Ubuntu` et cliquer sur `Close Remote Connection`.



Autres extensions VSCode

ⓘ Les extensions suivantes sont détaillées dans une vidéo dédiées dans ce module.

- **Python Extension Pack** : Fournit un support complet pour Python. (De Don Jayamanne)
- **GitHub Copilot** : Un assistant de codage IA.
- **Path Intellisense** : Autocomplétion des chemins de fichiers.
- **Material Icon Theme** : Améliore les icônes de fichiers et dossiers.
- **Jupyter**