1. Docker in Windows 11 via WSL2

1.1. Présentation de WSL2

WSL = **W**indows **Subsystem** for **L**inux .
La version 2 coïncide à peu près avec l'apparition de windows 11 .

1.2. Installation de Linux Ubuntu sur WSL2

Dans un terminal "powershell as admin"

Enable-WindowsOptionalFeature -Online -FeatureName 'VirtualMachinePlatform' -All

Enable-WindowsOptionalFeature -Online -FeatureName 'Microsoft-Windows-Subsystem-Linux' -All

puis après un éventuel rédémarrage :

installer si besoin la mise à jour du noyau windows en WSL2 :
https://wslstorestorage.blob.core.windows.net/wslblob/wsl_update_x64.msi
ou mieux encore
wsl --update

wsl --set-default-version 2

wsl --install -d Ubuntu

choisir un username/password linux (ex : **didier/pwd**)
(exemple de version de ubuntu : 22.04.3 LTS)
---
exit
wsl --list
Distributions du sous-système Windows pour Linux :
Ubuntu (par défaut)
ubuntu
exit

Un éventuel rédémarrage peut faire du bien ...

Dans powerShell ou CMD :
---
ubuntu
exit

NB: au sein du sous system linux pour windows , le disque "**c:**" de windows est vu comme **/mnt/c** .

Création d'un éventuel lien symbolique (avec chemin à adapter):

```
cd
ln -s /mnt/c/tp/local-git-mycontrib-repositories/msa-vagrant/with-wsl2 with-wsl2
```

1.3. Installation de docker-ce sur WSL2/Ubuntu

install-docker.sh

```
sudo apt update
sudo apt install apt-transport-https ca-certificates curl software-properties-common

curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o
/usr/share/keyrings/docker-archive-keyring.gpg

echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-
keyring.gpg] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable" | sudo tee
/etc/apt/sources.list.d/docker.list > /dev/null

sudo apt update

apt-cache policy docker-ce
sudo apt install docker-ce

sudo systemctl status docker
sudo usermod -aG docker ${USER}
sudo systemctl restart docker

mkdir -p ~/.docker/cli-plugins/
curl -SL https://github.com/docker/compose/releases/download/v2.3.3/docker-compose-linux-
x86_64 -o ~/.docker/cli-plugins/docker-compose

chmod +x ~/.docker/cli-plugins/docker-compose
docker compose version
```

sh ./install-docker.sh

Vérifications:

```
docker --version
docker container ls
```

NB : Un redémarrage complet (windows + wsl) peut quelquefois être utile pour que le démon docker soit bien actif .

1.4. Utilisation de docker sous WSL2

Sur une machine windows , une fois connecté au sous système linux (ex : ubuntu) au sein d'un terminal (cmd ou powershell), les commandes dockers lancées peuvent éventuellement avoir un effet persistant sur le long terme :

- si on lance un container , celui-ci sera toujours en fonction après une déconnexion/reconnexion au sous système linux(ex : ubuntu) et même après un reboot complet de la machine windows.
- il est donc fortement conseillé d'utiliser "docker compose" pour contrôler tout ce qui sera démarré ou arrêté via docker et manière à éviter d'éventuels conflits (de numéro de port TCP/IP) avec d'autres applications windows :
 - pour tout démarrer : **docker compose up** (dans un répertoire contenant docker-compose.yml)
 - pour tout arrêter : **docker compose down**

Et seulement si certains conteneurs dockers doivent durablement fonctionner en arrière plan (même après reboot) : **docker compose up &**