

# Installation Dianet

---

## Premiers steps

---

Lancer PowerShell Lancer Wsl

```
wsl --status
```

wls version 2

## Installer GCC

---

```
sudo apt update
sudo apt install build-essential
gcc --version
# Si pas gcc version 7
sudo apt install g++-7 -y

# Si la version de gcc ne change pas, alors
sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-7 100 --
slave /usr/bin/g++ g++ /usr/bin/g++-7
```

Autres installations

```
sudo apt install cmake
```

## Installer Cuda toolkit 10.2

---

Site : NVIDIA [<https://developer.nvidia.com/cuda-10.2-download-archive>]

```
wget
https://developer.download.nvidia.com/compute/cuda/10.2/Prod/local_install
ers/cuda_10.2.89_440.33.01_linux.run
sudo sh cuda_10.2.89_440.33.01_linux.run
```

Next steps : accept > Invalider l'installation du Driver en allant sur ☒ Driver puis **Espace** Vérifier que CUDA Toolkit 10.2 est coché, décocher le reste, puis > **Install**

Set up les variables globales dans le fichier `bashrc`

```
export PATH "$HOME/. local/bin: $PATH"
export CUDA_HOME="/usr/local/cuda-10.2" export PATH "/usr/local/cuda-10.2/bin:$PATH"
export
LD_LIBRARY_PATH=/usr/local/cuda/lib64/{LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
export LD_LIBRARY_PATH="/usr/local/cuda-10.2/lib64/{LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}
export PATH "/usr/lib/wl/lib/:$PATH"

export SKLEARN_ALLOW_DEPRECATED_SKLEARN_PACKAGE_INSTALL=True
```

Pour vérifier :

```
nvcc --version # doit être 10.2
```

## Pour pouvoir télécharger les documents directement sur la machine linux

---

```
sudo apt install x11-apps
```

Old Firefox > Télécharger directement dans l'endroit qui convient sous Linux (via Dropbox/Google Drive)

## Installer Python

---

Créer un répertoire de travail pour accueillir les répertoires et les fichiers du serveur de calcul – le serveur sera lancé à partir de cet espace Exemple du répertoire utilisé ici :

```
mkdir riedones14
```

Créer un environnement mamba/conda tournant avec python 3.8

- Installer miniforge

```
mamba create --prefix $HOME/riedones14/env python=3.8
mamba activate $HOME/riedones14/env
```

## Installation librairies

```
pip install pyrender # Successfully installed Pillow-10.3.0 PyOpenGL-3.1.0
freetype-py-2.4.0 imageio-2.34.0 networkx-3.1 numpy-1.24.4 pyglet-2.0.15
pyrender-0.1.45 scipy-1.10.1 six-1.16.0 trimesh-4.2.4
```

```
# installer les versions adéquates de PyTorch et des composants de PyTorch
les versions ci-dessous ne provoquent pas d'erreur pour le preprocessing)
:
```

```
pip install torch==1.8.1 # Successfully installed torch-1.8.1 typing-
extensions-4.10.0
```

```
pip install torch-sparse==0.6.10 -f https://data.pyg.org/whl/torch-
1.8.1+cu102.html
# Successfully installed torch-sparse-0.6.10
```

```
pip install torch-scatter==2.0.7 -f https://data.pyg.org/whl/torch-
1.8.1+cu102.html
# Successfully installed torch-scatter-2.0.7
```

```
pip install torch-cluster==1.5.9 -f https://data.pyg.org/whl/torch-
1.8.1+cu102.html
# Successfully installed torch-cluster-1.5.9
```

```
pip install torch-spline-conv -f https://data.pyg.org/whl/torch-
1.8.1+cu102.html
# Successfully installed torch-spline-conv-1.2.2
```

```
pip install torch-geometric==1.7.2
# Successfully installed MarkupSafe-2.1.5 certifi-2024.2.2 charset-
normalizer-3.3.2 googledrivedownloader-0.4 idna-3.6 isodate-0.6.1 jinja2-
3.1.3 joblib-1.3.2 pandas-2.0.3 pyparsing-3.1.2 python-dateutil-
2.9.0.post0 python- louvain-0.16 pytz-2024.1 rdflib-7.0.0 requests-2.31.0
scikit-learn-1.3.2 threadpoolctl-3.4.0 torch-geometric-1.7.2 tqdm-4.66.2
tzdata-2024.1 urllib3-2.2.1
```

En plus ...

```
sudo apt install python3-dev libopenblas-dev libeigen3-dev
```

installer les benchmarks depuis le dépôt git :

```
pip install git+https://github.com/humanpose1/deeppointcloud-
benchmarks.git@registration/riedones3d
```

# Minkowski

---

```
pip install ninja
export MAX_JOBS=4

pip install -U git+https://github.com/NVIDIA/MinkowskiEngine --no-deps
# si l'installation bloque, vider le cache : rm -rf ~/.cache/pip / réduire
la valeur de la globale MAX_JOBS
# [pip freeze : MinkowskiEngine @
git+https://github.com/NVIDIA/MinkowskiEngine@02fc608bea4c0549b0a7b00ca1bf
15dee4a0b228]
```

# Teaser

---

```
git clone https://github.com/MIT-SPARK/TEASER-plusplus.git
cd TEASER-plusplus && mkdir build && cd build

cmake -DTEASERPP_PYTHON_VERSION=3.8 .. && make teaserpp_python
# Si cmake n'a pas été installé, l'installé ici via : sudo apt install
cmake
cd python && pip install . # Successfully installed teaserpp-python-1.0.0
```

Installer le package cupy, qui permet de rendre certaines librairies de numpy "CUDA compatibles". Il faut préciser avec quelle version de CUDA grâce au flag "--pre" (pre-release) :

```
pip install --pre cupy-cuda102 # évite les problèmes de compatibilité
entre la version de CUDA (la version 10.2 est ancienne) et numpy
```

Important : vérifier que la version de numpy est bien en 1.19.5 à ce stade, sinon :

```
pip install numpy==1.19.5
```

Installer le module probreg (utilisé pour le preprocessing) :

```
pip install probreg # Successfully installed probreg-0.3.7 pybind11-2.12.0
transforms3d-0.4.1
```

Installer seaborn :

```
pip install seaborn --no-deps # --no-deps évite l'upgrade automatique de  
numpy # Successfully installed  
seaborn-0.13.2
```

Revenir dans le rep de travail - exple :

```
cd $HOME/riedones14/
```

Cloner le git dans le rep de travail :

```
git clone https://github.com/humanpose1/riedones3d.git
```

```
pip install probreg # Successfully installed probreg-0.3.7 pybind11-2.12.0  
transforms3d-0.4.1
```

Réimporter les fichiers du répertoire « riedones3d/scripts » dans le répertoire « riedones3d/scripts » du répertoire de travail (faire une copie du répertoire « script » au préalable si désiré)