

Installation Carla-ROS-Bridge

Afin de réaliser cette installation, on se basera sur un système d'exploitation Ubuntu 20.04 natif pour des raisons de compatibilité principalement.

Nous allons nous référer à la [documentation](#) et au [github](#) officiels.

Avant de commencer :

- Système Linux natif
 - Ubuntu 20.04
 - Problèmes de performance avec une VM
- Carte graphique Nvidia vivement recommandé
- ROS Noetic
 - Référez vous au [document d'installation](#)
 - Référez vous au premier lien ci-dessous
- Téléchargement de [Carla 0.9.11](#)
 - Pour Ubuntu choisir CARLA_0.9.11.tar.gz

- ◆ [Installer ROS Noetic sur Ubuntu 20.04](#)
- ◆ [Installer Carla sur Linux](#)
 - Choisir une version
 - Extraire le tar gz
 - Lancer CarlaUE4.sh

➤ Installez les dépendances nécessaires

```
sudo apt install python3-rosdep2
```

```
sudo apt install python3-pip  
pip3 install numpy  
pip3 install pygame
```

Étant sur Ubuntu 20.04, la documentation officielle recommande d'utiliser le répertoire source pour ensuite build le carla-ros-bridge.

- Choisissez un dossier sur votre système pour l'emplacement du répertoire que nous allons utiliser (ici ce sera dans /home/<user>)

```
mkdir -p ~/carla-ros-bridge/catkin_ws/src
```

- Se déplacer dans le répertoire **carla-ros-bridge** et cloner le répertoire github

```
cd ~/carla-ros-bridge  
git clone --recurse-submodules https://github.com/carla-simulator/ros-bridge.git \  
catkin_ws/src/ros-bridge
```

- Si ce n'est pas déjà fait, configurez l'environnement ROS.
→ Voir le document pour l'installation de ROS Noetic sur Ubuntu 20.04

```
source /opt/ros/noetic/setup.bash
```

Il est recommandé d'ajouter cette ligne à la fin du fichier ~/.bashrc pour ne pas avoir à configurer l'environnement à chaque fois.

```
echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
```

- Installation des dépendances ROS nécessaires

```
cd catkin_ws  
sudo apt install python3-rosdep2  
rosdep update  
rosdep install --from-paths src --ignore-src -r
```

- Build du carla-ros-bridge

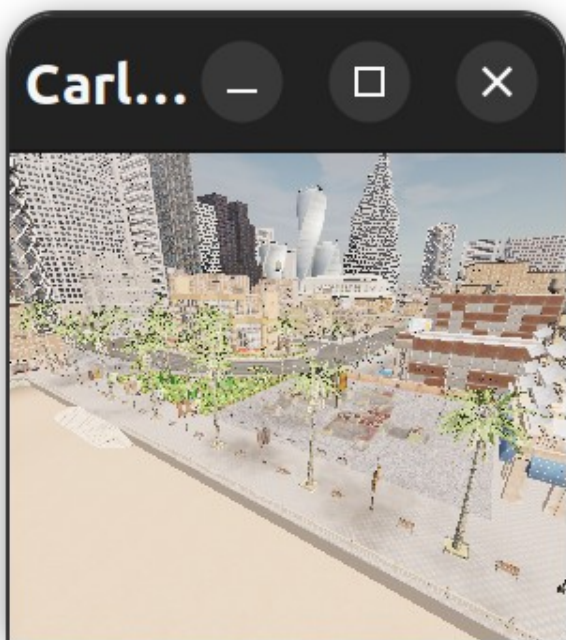
```
catkin_make
```

- Lancement du serveur Carla

```
./CarlaUE4.sh -windowed -ResX=320 -ResY=240 -opengl
```

L'argument "-opengl" est nécessaire pour le bon fonctionnement avec GPU Nvidia. Ajustez en fonction de votre carte graphique.

*On choisit également une résolution basse pour que les performances ne soient pas trop impactées. Il existe également l'argument « **-quality-level=Low** » pour baisser la qualité.*



Une fenêtre comme celle-ci va alors s'ouvrir, il est nécessaire qu'elle reste ouverte pour laisser le serveur de Carla tourner en fond.

Il s'agit de la carte par défaut, qui est modifiable, voir commande d'après.

- Ensuite on ouvre un nouveau terminal pour modifier la configuration de la fenêtre serveur de Carla (commande adaptable)

```
python util/config.py -m Town02 --fps 10
```

On charge la carte 2 et on modifie le taux de fps max à 10.

Cette étape est primordiale pour pouvoir effectuer les commandes. Votre environnement de travail par défaut ne comprend pas les commandes comme « roslaunch » ou « carla_ros_bridge », c'est pourquoi il faut configurer les chemins d'accès dans l'environnement.

➤ Configuration de l'environnement et espace de travail

Ajout des différents modules au path

```
export CARLA_ROOT=<path-to-carla>
export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/carla/dist/carla-\
<carla_version_and_arch>.egg:$CARLA_ROOT/PythonAPI/carla
```

Ajout du chemin source de carla-ros-bridge

```
source ~/carla-ros-bridge/catkin_ws/devel/setup.bash
```

Il est recommandé d'ajouter ces lignes à la fin du fichier ~/.bashrc pour ne pas avoir à configurer l'environnement à chaque fois.

→ Le fichier .bashrc permet entre autre d'effectuer à chaque ouverture du terminal les commandes de configuration d'environnements.

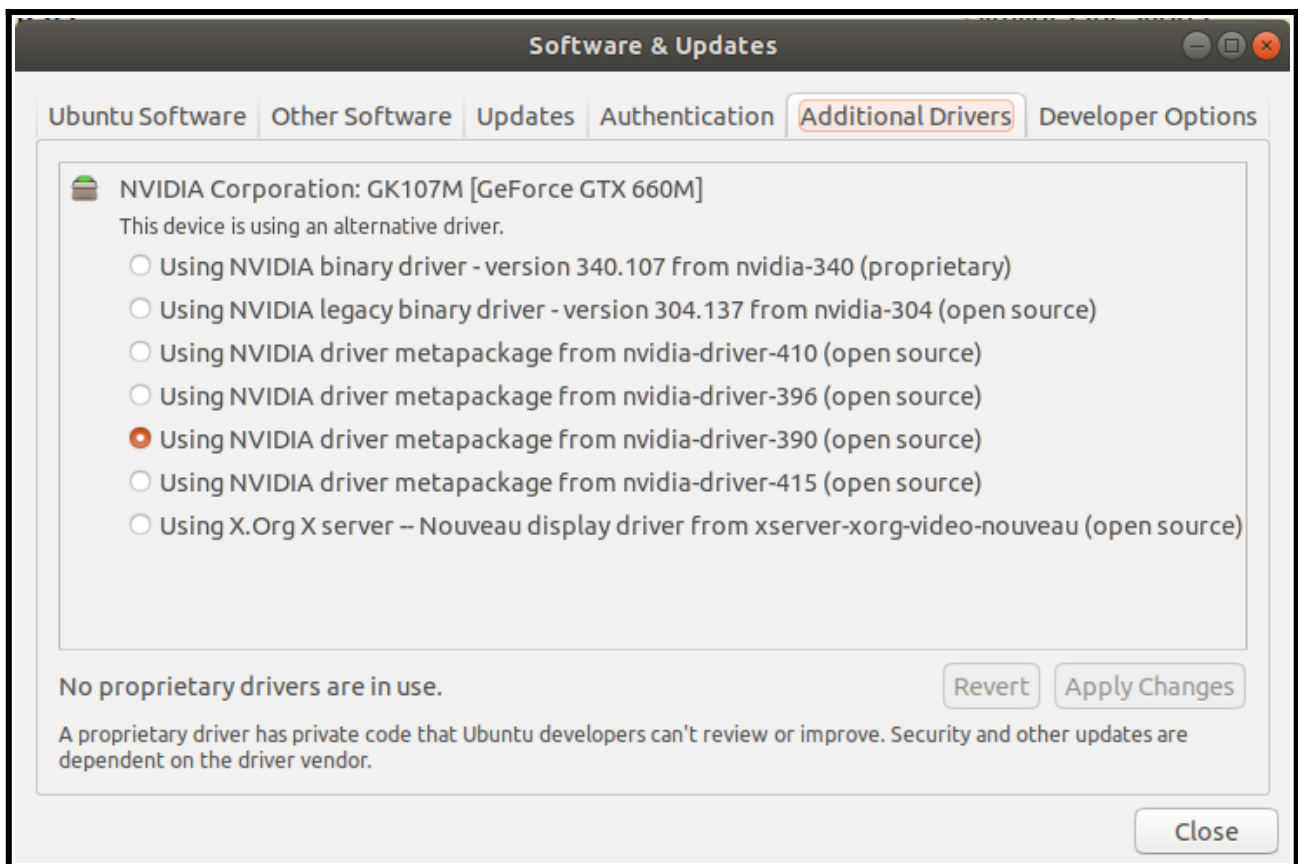
Modifiez les chemins si nécessaire en fonction de votre installation et de la version de python que vous utilisez.

```
cat << EOF >> ~/.bashrc
source /opt/ros/noetic/setup.bash
source ~/carla-ros-bridge/catkin_ws/devel/setup.bash
export CARLA_ROOT=~/.CARLA_0.9.11
export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/carla/dist/carla-0.9.11-\
py3.7-linux-x86_64.egg:$CARLA_ROOT/PythonAPI/carla
export PYTHONPATH=$PYTHONPATH:$CARLA_ROOT/PythonAPI/carla-0.9.11-py3.7-linux-\
x86_64.egg
EOF
```

Appliquez les modifications (ou relancez un terminal)

```
source ~/.bashrc
```

- Mettez à jour vos drivers Nvidia (dans Software & Updates)



*Sélectionnez le driver puis faites « **Apply Changes** ».*

Le driver propriétaire est recommandé, vous pouvez vous référer à ce [tutoriel](#).

- Redémarrez la machine

```
sudo reboot
```

Dans la prochaine étape nous allons voir comment lancer le carla-ros-bridge, pour plus d'informations veuillez consulter la [procédure d'utilisation](#) détaillée.

- Lancement du carla-ros-bridge dans le terminal

Lancer simplement le carla-ros-bridge

```
roslaunch carla_ros_bridge carla_ros_bridge.launch
```

Lancer le carla-ros-bridge avec un fichier d'exemple

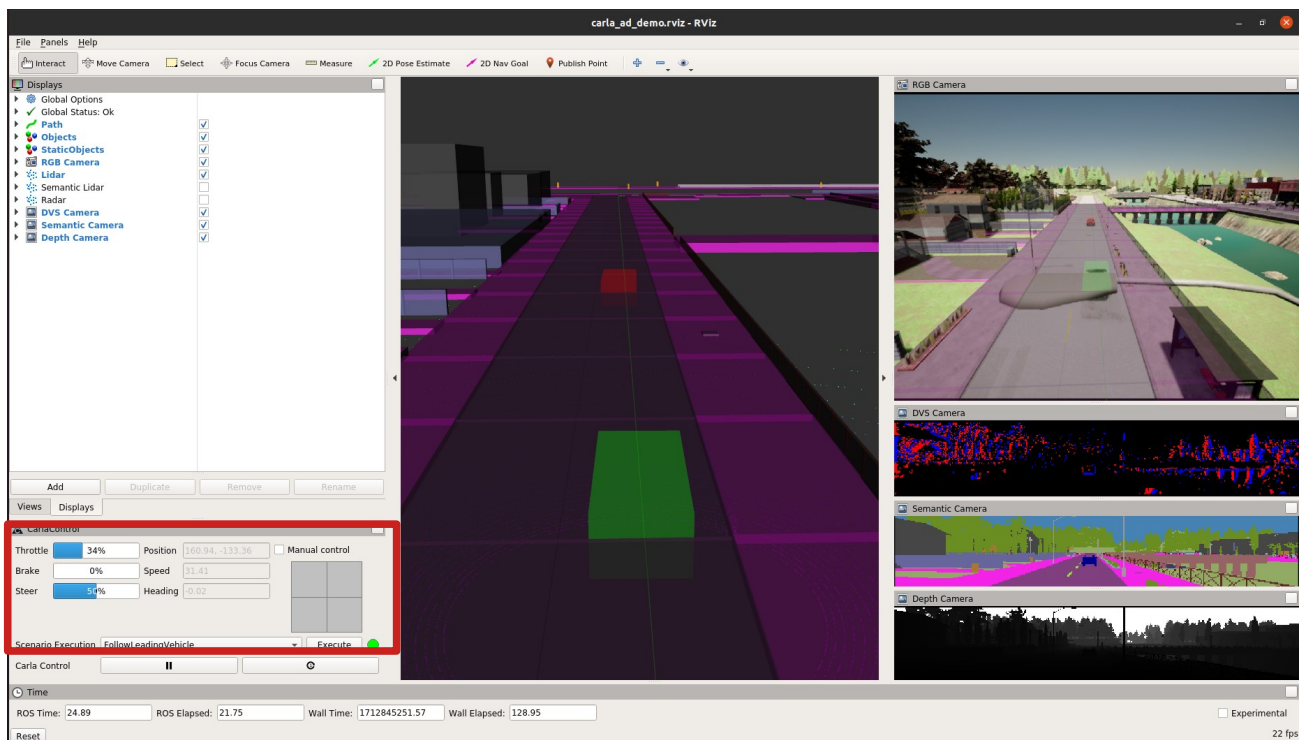
```
roslaunch carla_ros_bridge carla_ros_bridge_with_example_ego_vehicle.launch
```

Voir les autres fichiers exemples dans le répertoire suivant (à adapter)

```
~/carla-ros-bridge/catkin_ws/src/ros-bridge/carla_ros_bridge/launch/
```

- Lancer le carla-ros-bridge avec l'interface graphique RVIZ et une démo

```
roslaunch carla_ad_demo carla_ad_demo_with_scenario.launch
```



À droite on a la simulation exemple comme dans Carla.

On a également les données de ROS dans l'interface graphique RVIZ. Dans la zone qu'indique l'encadré rouge on peut obtenir des informations sur le véhicule, dont la vitesse et la position.

Gestions d'erreurs récurrentes sur une nouvelle installation

✗ /usr/bin/env: 'python': No such file or directory

```
sudo apt install python-is-python3
```

✗ **CARLA python module version 0.9.XX required. Found: 0.9.11**

Éditez le fichier et modifiez **0.9.XX** en **0.9.11**

```
~/carla-ros-bridge/catkin_ws/src/ros-bridge/carla_ros_bridge/src/carla_ros_bridge/CARLA_VERSION
```

✗ **AttributeError : 'CarlaRosBridge' object has no attribute 'shutdown'**

Cette erreur se corrige en ajoutant un timeout au lancement de carla-ros-bridge

```
roslaunch carla_ros_bridge carla_ros_bridge.launch timeout:=200
```

✗ **libomp.so.5: cannot open shared object file: No such file or directory**

```
sudo apt install libomp5
```