

Steps to install ROS on Ubuntu

1/ Setup your sources.list

1.1/ Prepare your computer to accept Rose then enter the next

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" >
'/etc/apt/sources.list.d/ros-latest.list
```

1.2/ Enter the password

1.3/ Prepare your keys, then enter the following:

```
sudo apt install curl # if you haven't already installed curl
```

```
- curl -s https://raw.githubusercontent.com/ros/rosdistro/master/ros.asc | sudo apt-key add
```

1.4/ First, make sure your Debian package index is up-to-date :

enter the following :

```
sudo apt update
```

1.5/ Now pick how much of ROS you would like to install

We want to install Rose completely so we'll go into the next :

Desktop-Full Install: (Recommended) : Everything in Desktop plus 2D/3D simulators and 2D/3D perception packages

```
sudo apt install ros-noetic-desktop-full
```

1.6/ Environment setup

enter the following :

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

1.7/ Dependencies for building packages

Up to now i installed what i need to run the core ROS packages. To create and manage my own ROS workspaces, there are various tools and requirements that are distributed separately. To install this tool and other dependencies for building ROS packages, run :

```
sudo apt install python3-rosdep python3-rosinstall python3-rosinstall-generator python3-wstool build-essential
```

1.7.1/ Initialize rosdep

enter the following :

```
sudo apt install python3-rosdep
```

1.7.2/ With the following, you can initialize rosdep

```
sudo rosdep init
```

```
rosdep update
```

Steps to install Ubuntu on jetson nano

1/ you need to enable distribution upgrades in the update manager by setting `prompt=normal` in the `/etc/update-manager/release-upgrades` file. As usual, close with `<Ctrl>+<X>`, `<Y>` and `<Enter>`

```
$ sudo nano /etc/update-manager/release-upgrades
```

2/ With the update manager set, we need to refresh the software database again. Once done, you can reboot

```
# refresh your system again
```

```
$ sudo apt-get update
```

```
$ sudo apt-get dist-upgrade
```

```
$ sudo reboot
```

3/ Upgrade to Ubuntu 20.04.

With all preparations made, it's time for the upgrade to Ubuntu 20.04. It will take several hours. Unfortunately, some input is required throughout the procedure as there are questions to be answered. Check your screen now and then. Answer all questions with the suggested default value.

```
# upgrade to Ubuntu 20.04
```

4/ After entering `<y>`, the transformation will take place. As mentioned, some questions are asked during the process. Always confirm the default value

When Ubuntu 20.04 is installed, all obsolete files can be removed from your drive

5/ Now that the packages have been removed, Ubuntu will ask for a restart. Don't reboot now! That's really important.

6/ You need to check some files and make some changes before Ubuntu can reboot. If you forget this, you will get a warning every time Ubuntu starts. So do it right away.

First, check that

```
WaylandEnable=false is uncommented in the /etc/gdm3/custom.conf file
```

7/ Uncomment (remove the hash before the line) Driver "nvidia" in the file `/etc/X11/xorg.conf`

8/ Finally, reset the upgrade manager to never.

9/ Now you can safely reboot the system.

check and editing some files

\$ sudo nano /etc/gdm3/custom.conf

\$ sudo nano /etc/X11/xorg.conf

\$ sudo nano /etc/update-manager/release-upgrades

\$ sudo reboot

10/ Starting Ubuntu 20.04 on your Nano.

A few preparations are needed before you can fully enjoy Ubuntu 20.04 on your Jetson Nano.

Start with the well-known update, upgrade and autoremove cycle. You will get probably no upgrades as Ubuntu is just installed. However, autoremove will free a lot of memory by deleting obsolete files.

Next, delete the directory `/usr/share/vulkan/icd.d` to prevent lavapipe warnings when using Jtop. It looks the folder is removed in the latest Jetpack as well.

See the issue posted on Raffaello Bonghi's GitHub page and the solution given by SkrilaxCZ.

remove icd.d

\$ sudo rm -rf /usr/share/vulkan/icd.d

11/ Remove also an annoying circular symbolic link in `/usr/share/applications` that makes the same app appear 86 times in your software overview.

prepare your system

\$ sudo apt-get update

\$ sudo apt-get upgrade

\$ sudo apt-get autoremove

remove circular symlink

\$ sudo rm /usr/share/applications/vpi1_demos

remove distorted nvidia logo in top bar

\$ cd /usr/share/nvpmodeIndicator

\$ sudo mv nv_logo.svg no_logo.svg

12/ The last action is to re-enable the original NVIDIA repositories, which were disabled during the upgrade. In the folder `/etc/apt/sources.list.d/` you will find the five files that needed to be changed. Open each file with `$ sudo nano` and remove the hash in front of the line to activate the repository.

13/ Ubuntu 20.04 comes with gcc version 9.3.0.

Some software packages, especially the CUDA software, requires a gcc version 8. We shall install this version besides the already available version 9. With a simple command, you can now switch between the two versions. The gcc compiler is always accompanied by the corresponding g++ compiler. The latter will also be installed

install gcc and g++ version 8

\$ sudo apt-get install gcc-8 g++-8

setup the gcc selector

\$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-9 9

\$ sudo update-alternatives --install /usr/bin/gcc gcc /usr/bin/gcc-8 8

setup the g++ selector

\$ sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-9 9

\$ sudo update-alternatives --install /usr/bin/g++ g++ /usr/bin/g++-8 8

if you want to make a selection use these commands

\$ sudo update-alternatives --config gcc

\$ sudo update-alternatives --config g++

14/ Always select the two corresponding versions at the same time. Otherwise, very obscure errors may occur during compilation. As mentioned, if you want to compile software that requires CUDA or cuDNN, such as OpenCV, choose version 8.

15/ The same goes for the clang compilers. In this case, the CUDA and cuDNN need version 8, while Ubuntu installs default version 10. You can either install version 8 right away with `$ sudo apt-get install clang-8` or install more versions and create an `sudo update-alternatives --config` just like above.

16/ Upgrading.

You may run into problems upgrading Ubuntu 20.04 on your Jetson Nano after a while.

The problem is the wrong version of the `nvidia-l4t-init` file.

You must first install the correct version before any further upgrade can succeed.

The command `$ sudo apt --fix-broken install` gives you the information

17/ The easiest solution is to force the upgrade with the command `$ sudo dpkg -i --force-overwrite`.

the sleep.conf is overwritten, and nvidia-l4t-init is installed successfully.

Now you can upgrade your Ubuntu 20.04 with the usual command `$ sudo apt-get upgrade`

18/ Install ROS 2 packages

18.1/ Update your apt repository caches after setting up the repositories

sudo apt update

18.2/ ROS 2 packages are built on frequently updated Ubuntu systems. It is always recommended that you ensure your system is up to date before installing new packages.

sudo apt upgrade

18.3/ Desktop Install (Recommended): ROS, RViz, demos, tutorials.

sudo apt install ros-foxy-desktop

18.4/ ROS-Base Install (Bare Bones): Communication libraries, message packages, command line tools. No GUI tools.

sudo apt install ros-foxy-ros-base

19/ Environment setup

\$echo "source /opt/ros/foxy/setup.bash" >> ~/.bashrc

\$ source ~/.bashrc