How to Install ROS on Ubuntu 18.04

In this post, we are going step-by-step to install ROS Melodic in a fresh Ubuntu 18.04. I’m using a regular desktop computer, the one I use to support me on ROS development.

Let’s do it!

## **Step 1 – Configuration**

The first step is adding the **ROS repository** to your ubuntu **sources.list**

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

Secondly, add the keys for accessing it:

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

sudo apt update

## **Step 2 – ROS Installation**

We have some options at this point. They are:

### **a. ROS Base**

It installs ROS package, build and communication libraries.

It is recommended for embedded computers, where you don’t have much hardware for graphical tools or real robots (in the production stage) to run their specific tasks, without the need of debugging.

sudo apt install ros-melodic-ros-base

### **b. ROS Desktop**

It installs ROS Base, rqt, rviz and robot-generic libraries

These are the basic packages for ROS development. It’s recommended for beginners to follow tutorials about ROS without having to install too much. **RQT** and **RVIZ** provides graphical interfaces for visualizing what is happening behind the scenes. With this option, you don’t have to work only using shell.

sudo apt install ros-melodic-desktop

### **c. ROS Desktop Full**

It installs everything of **ROS Desktop** option plus **2D/3D simulators** and **2D/3D perception.**

If you want to simulate using ROS default Gazebo version, that’s your option. The **Gazebo** versions that comes with ROS Melodic is **9.0.** (Related post: [All about Gazebo 9](https://www.theconstructsim.com/all-about-gazebo-9-with-ros/))

sudo apt install ros-melodic-desktop-full

user@computer:~$

## **Step 3 – ROS Dependencies**

We are almost there! ROS commands Client can also manage dependencies for you and rosdep is the one in charge of it.

That’s why we have to initialize rosdep. It goes like this:

sudo rosdep init

Similarly to ubuntu repositories, you need to update rosdep sometimes:

rosdep update

## **Step 4 – Configuring environment**

We have ROS and the dependencies manager installed. Let’s configure our environment. This is a very important step, once we have it done, working with ROS will be smooth.

ROS is installed at /opt/ros/<distro> (in this case /opt/ros/melodic). In order to have ROS commands available, it’s needed to source the shell file inside of the installation folder. This is done like the following:

|  |  |
| --- | --- |
| 1 | source /opt/ros/melodic/setup.bash |

But.. considering we want to have it available in every terminal we open, we use to have a “shortcut”, which is adding this command to the file "/home/<user>/.bashrc". The .bashrc file is called every time a new terminal is opened, therefore, we won’t need to source ROS setup, since we have the instruction in this file. In order to add the command to the file, you can edit it manually using an editor of your preference or just execute the command below:

|  |  |
| --- | --- |
| 1 | echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc |

## **Step 5 – Testing it!**

At this point, we must have everything in place. Let’s try some ROS commands to make sure the installation has finished successfully.

If you have the same terminal opened from the beginning, consequently, you have to execute the source for your .bashrc file to have ROS commands available.

|  |  |
| --- | --- |
| 1 | source ~/.bashrc |

And if you have added it to your .bashrc file, that’s the last time you will execute it =)

The first thing we should try it running roscore. In other words, we will run the process in charge of communicating everything ROS-related in a ROS environment.

|  |  |
| --- | --- |
| 1 | roscore |