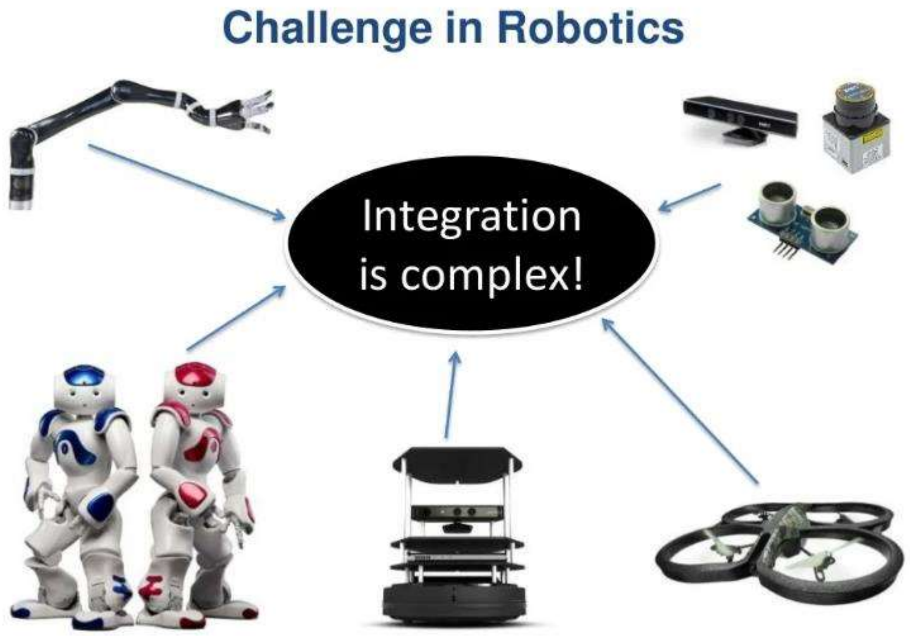
What is ROS?

ROS(Robot Operating System) is a software framework to enable communication, data flow, and suitable infrastructure development for a hardware-software system (robotics being the best application).

Oh, wait! It is also technically **not an operating system**. ROS is an OS only in concept because **it provides all the services that any other OS does** — hardware abstraction, low-level device control, implementation of commonly-used functionality, message-passing between processes, and package management. But, ROS is a framework on top of the O.S. that allows it to abstract the hardware from the software. This means you can think in terms of software for all the hardware of the robot. And that’s good news for you because this implies that you can actually create programs for robots without having to deal with the hardware. Cool right!

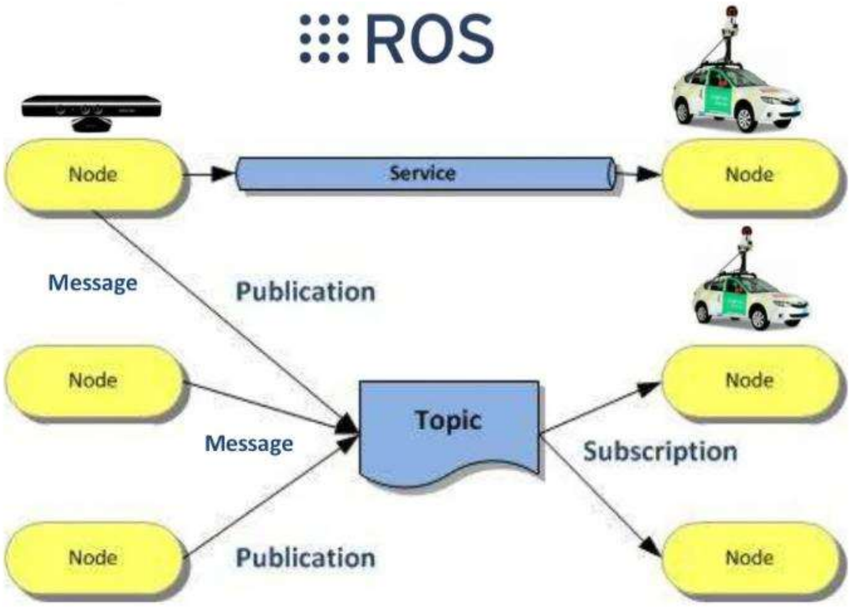
Why ROS?



Challenge in Robotics

Integration is complex!

ROS is the solution!



You'll find a ROS Package for literally anything!

Easy communication between multiple language nodes (Python, C++, Java,etc)

Great Simulation Tools

Simulate multiple robots with ROS

ROS doesn't occupy much space.

ROS Software is open source and free to use.

Growing user base, making ROS more prominent in near future.

ROS is the most mature, open source robotics project in the world. So if you are planning to get into robotics, **you need to master it**. In the next section, we are going to move onto the problem of installing ROS.

ROS is complex. ROS takes time. But the end result is fruitful.

NOTE: We would always prefer you to look up ros wiki and ros answers as they containing literally everything from where most of us have learnt ROS.

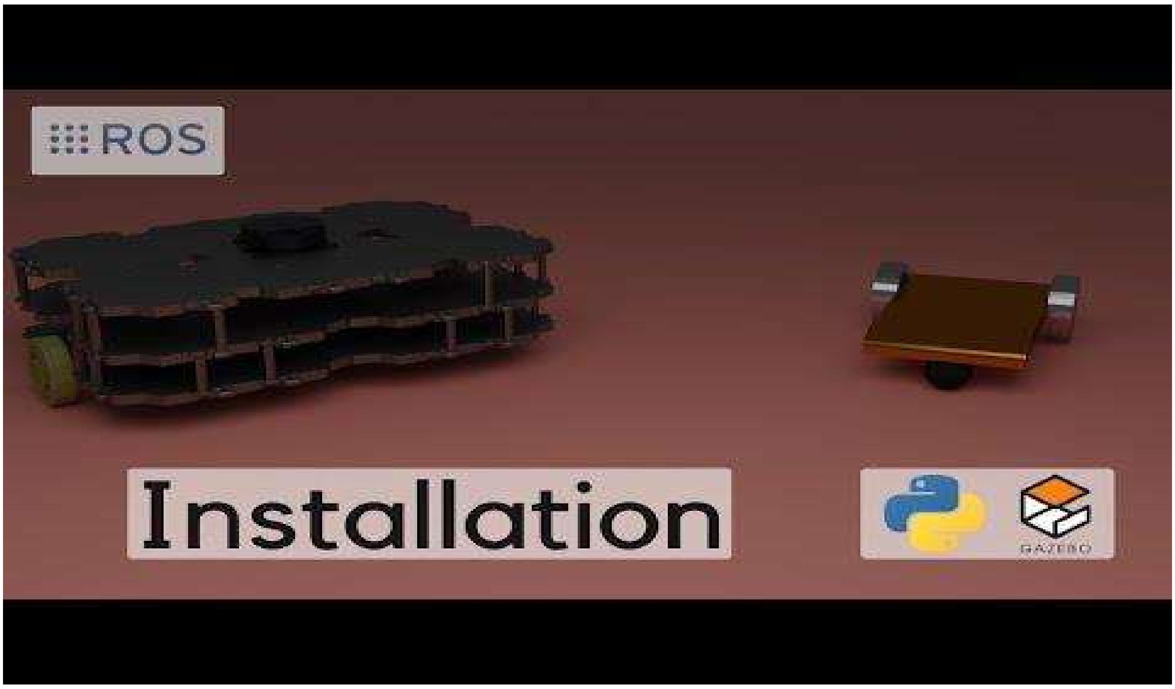
Installation Instruction

For Users with **Ubuntu 20.04 or Mac OS (Recommended)**

1. Install ROS Noetic from Ubuntu install of ROS Noetic

Also, do check out this video demonstration to get an easier understanding of installation

ROS Noetic Installation and Path Sourcing



- For everyone else, use VirtualBox

Creating a catkin workspace environment (ROS WiKi Managing Environment)

Commands to run in terminal:

```
mkdir -p ~/catkin_ws/src
cd catkin_ws/src/
catkin_init_workspace
sudo apt-get install python3-catkin-tools
cd ~/catkin_ws/
catkin build
source devel/setup.bash
```

If you followed along till here you will mostly have a file structure as follows

```
├── catkin_ws
   ├── build/
   |   ├── bin/
   |   ├── catkin/
   |   |  ...
   |   └── MakeFile
   ├── devel/
   |   ├── lib/
   |   ├── include/
   |   |  ...
   |   └── setup.bash
   └── src/
      ├── genpy/
      ├── packages1/
      |  ...
      └── CMakeLists.txt
```

Try to use catkin build instead of catkin_make as it is standard, more robust and supports mutltiple package type to be built together (orocos, non-ros packages).

For this,

```
$ rm -r build/*
$ rm -r devel/*
```

Then Install catkin tools For Ubuntu

```
$ sudo apt-get update
$ sudo apt-get install python-catkin-tools
```

For others (depreciated)

```
$ sudo pip install -U catkin_tools
--or--
$ git clone https://github.com/catkin/catkin_tools.git
$ cd catkin_tools
$ pip install -r requirements.txt --upgrade
$ python setup.py install --record install_manifest.txt
```

Finally

```
user@master: ~/catkin_ws$ catkin build
```

You will see `build/` and `devel/` are filled with new files built from your files from src