# ROS Basics in 5 days

## Appendix 1 Installing ROS

## A.1 How to install ROS in my local computer?

During the whole Course, you have already been using ROS with its full capabilities. In fact, you are almost a master in ROS programming! And that is because in our Robot Ignite Academy, you already have everything installed and set up, so that you can go straight and focus in learning the really important things of ROS: how to apply it to interact with robots. But what if you want now to apply everything you have learned during the Course, in your local computer?

Well, the first step would be, of course, to install ROS in your local computer. And that is what you are going to do during this appendix. For this case, we are going to explain the steps to install and setup the same environment that you've been using during the whole Course. This is, a **ROS Noetic** distribution installed in an **Ubuntu 20.04 Focal** machine. If you want to install a different version, or you are using a different machine, please refer to the official documentation, here: http://wiki.ros.org/noetic/Installation (http://wiki.ros.org/noetic/Installation)

With the proper introductions made, let's go with the steps needed in order to install ROS.

### A.1.1 Setup your sources.list

First of all, you'll need to setup your computer in order to be able to download packages from **packages.ros.org**. For that, execute the below command in your local Terminal:

▶ Execute in Local Terminal #1

In [ ]:

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

### A.1.2 Setup your keys

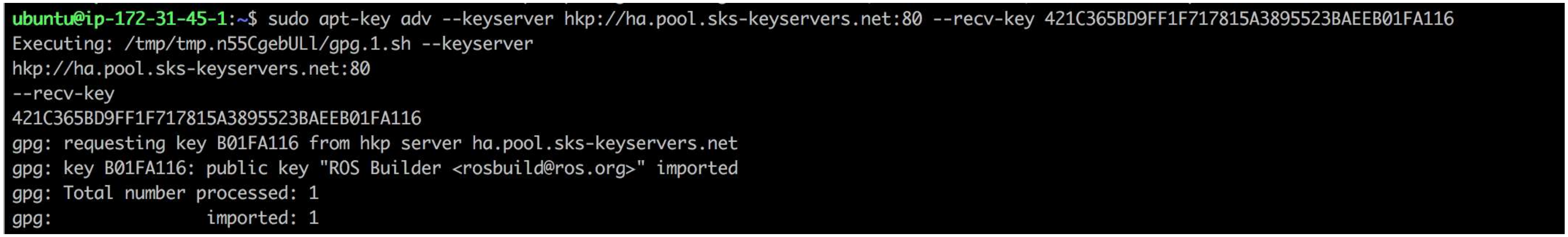
Next, you will download the key from the keyserver using the following command:

▶ Execute in Local Terminal #1

In [ ]:

```
wget http://packages.ros.org/ros.key -O - | sudo apt-key add -
```

If everything goes fine, you should see something like this:



```
ubuntu@ip-172-31-45-1:~$ sudo apt-key adv --keyserver hkp://ha.pool.sks-keyservers.net:80 --recv-key 421C365BD9FF1F717815A3895523BAEEB01FA116
Executing: /tmp/tmp.n55CgebULl/gpg.1.sh --keyserver
hkp://ha.pool.sks-keyservers.net:80
--recv-key
421C365BD9FF1F717815A3895523BAEEB01FA116
gpg: requesting key B01FA116 from hkp server ha.pool.sks-keyservers.net
gpg: key B01FA116: public key "ROS Builder <rosbuild@ros.org>" imported
gpg: Total number processed: 1
gpg:               imported: 1
```

### A.1.3 Installation

Great! Now we are ready to actually install ROS. First of all, we'll make sure that our Debian packages index is up to date. For that, execute the following command:

▶ Execute in Local Terminal #1

In [ ]:

```
sudo apt-get update
```

Great! Now you are ready to start installing ROS packages into your system. In order to have all the basic packages for start working with ROS, we recommend you to install the **Desktop Full** installation. For doing so, you can execute the following command:

▶ Execute in Local Terminal #1

In [ ]:

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

**NOTE**: Since it will download and install several packages, this installation can take some minutes. So be patient.

At this point, you have installed some of the basic tools that ROS provides, like RViz, rqt, navigation libraries... With these tools, you will be ready to start working with ROS. Anyways, you will need to install some extra packages eventually. For installing an specific ROS package, you just need to use the following command structure:

In [ ]:

```
sudo apt-get install ros-noetic-<PACKAGE_NAME>
```

For instance:

In [ ]:

```
sudo apt-get install ros-noetic-slam-gmapping
```

The above command will install the slam gmapping package for the ROS Noetic version.

### A.1.4 Initialize rosdep

Before you can actually start using ROS, though, you will need to initialize **rosdep**. rosdep will allow you to easily install system dependencies, and it is also required to run some core components in ROS. To initialize rosdep, execute the following command:

▶ Execute in Local Terminal #1

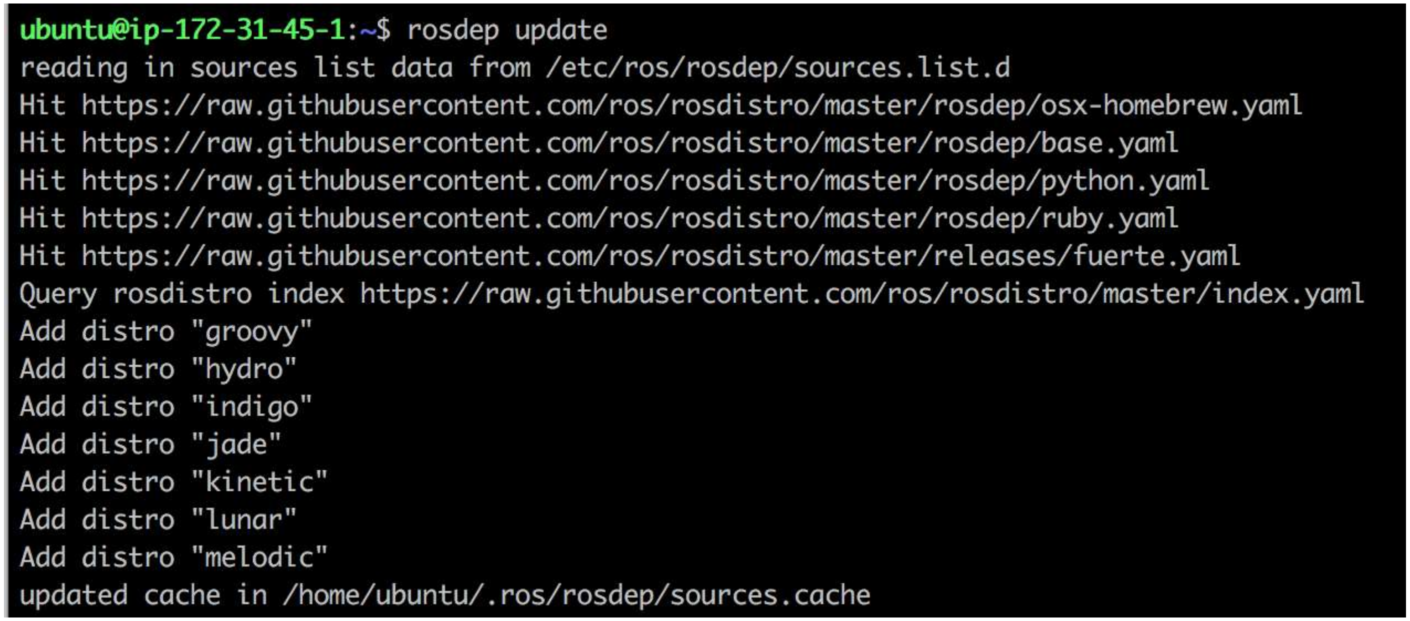
In [ ]:

```
sudo rosdep init
```

In [ ]:

```
rosdep update
```

If everything goes fine, you should get something like this after executing the **rosdep update** command.



```
ubuntu@ip-172-31-45-1:~$ rosdep update
reading in sources list data from /etc/ros/rosdep/sources.list.d
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/osx-homebrew.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/base.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/python.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/rosdep/ruby.yaml
Hit https://raw.githubusercontent.com/ros/rosdistro/master/releases/fuerte.yaml
Query rosdistro index https://raw.githubusercontent.com/ros/rosdistro/master/index.yaml
Add distro "groovy"
Add distro "hydro"
Add distro "indigo"
Add distro "jade"
Add distro "kinetic"
Add distro "lunar"
Add distro "melodic"
updated cache in /home/ubuntu/.ros/rosdep/sources.cache
```

### A.1.4 Environment setup

Finally, it is also recommended to automatically add the ROS Environment Variables (do you remember them from the 1st Chapter of the Course?) to your bash session every time a new Terminal is launched. For doing so, you can execute the following command:

▶ **Execute in Local Terminal #1**

In [ ]:

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

You will need to run this command on every new Terminal you open to have access to the ROS commands, unless you add this line to your **.bashrc** file. So, unless you wish to run that command every time you open a new Terminal, you should add it to your **.bashrc**. For that, you can run the following command:

▶ **Execute in Local Terminal #1**

In [ ]:

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

With the above command, you will add the line ***source /opt/ros/noetic/setup.bash*** to your **.bashrc** file. This way, each time you open a new Terminal in your computer, all the ROS Environment variables will be automatically set up.

Also, this process allows you to install several ROS distributions (for instance, indigo and noetic) on the same computer and switch between them. So, for instance, if you also had ROS Indigo installed in your local computer, you could switch between both distributions by using the bellow commands:

In [ ]:

```
source /opt/ros/indigo/setup.bash # To use Indigo
```

In [ ]:

```
source /opt/ros/noetic/setup.bash # To use noetic
```

### A.1.5 Dependencies for building packages

Great! So at this point, you have already installed and set up everything you need to run the core ROS packages. Anyways, there are various tools that you will also need in order to manage your ROS workspaces (remember your **catkin_ws**?). To install all this tools, you can run the following command:

▶ **Execute in Local Terminal #1**

In [ ]:

```
sudo apt-get install python-rosinstall python-rosinstall-generator python-wstool build-essential
```

### A.1.6 Test your setup

And we are done! Now let's test that our setup actually works, and that we can run ROS on our local machine. For that, let's follow the next Example.

- Example A.1 -

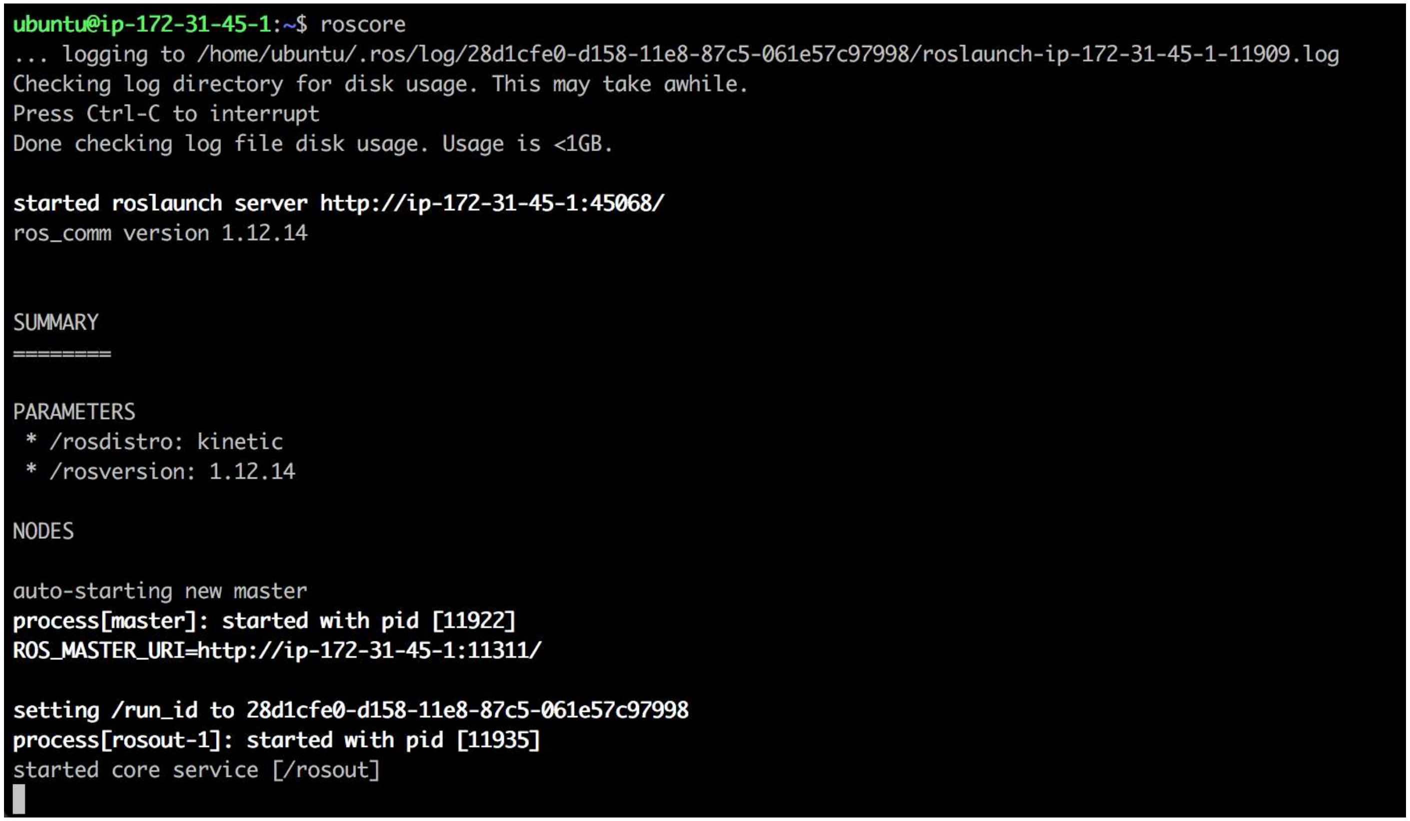
a) In the same Terminal where you have set up everything during this Notebook, run the following command:

▶ **Execute in Local Terminal #1**

In [ ]:

```
roscore
```

If everything goes fine, you should see something like this:



```
ubuntu@ip-172-31-45-1:~$ roscore
... logging to /home/ubuntu/.ros/log/28d1cfe0-d158-11e8-87c5-061e57c97998/roslaunch-ip-172-31-45-1-11909.log
Checking log directory for disk usage. This may take awhile.
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://ip-172-31-45-1:45068/
ros_comm version 1.12.14


SUMMARY
========

PARAMETERS
 * /rosdistro: kinetic
 * /rosversion: 1.12.14

NODES

auto-starting new master
process[master]: started with pid [11922]
ROS_MASTER_URI=http://ip-172-31-45-1:11311/

setting /run_id to 28d1cfe0-d158-11e8-87c5-061e57c97998
process[rosout-1]: started with pid [11935]
started core service [/rosout]
```

By the way... do you remember **roscore**? We introduced it to you back in the first chapter of this Course. **roscore** is the **main process** that manages all the ROS systems. So, if we want to do anything with ROS, we will always need to first start **roscore** in one Terminal.

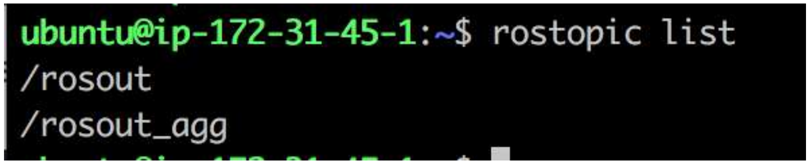
And you may ask yourself... and why I didn't use this command during the Course, if it so important? Well, it is because in the Ignite Academy, the roscore is automatically started for you whenever you enter a Course. But now, if you want to work in your local computer, you will need to manage it yourself!

b) With the roscore running in one Terminal, let's open a new Terminal. Within this new Terminal, type the following command:

▶ Execute in Local Terminal #2

In [ ]:
```
rostopic list
```

If everything goes fine, you should get the following topics:



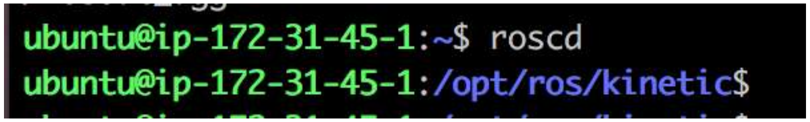
```
ubuntu@ip-172-31-45-1:~$ rostopic list
/rosout
/rosout_agg
```

Also, you can execute the ***roscd*** command to make sure that your ROS system is properly set.

▶ Execute in Local Terminal #2

In [ ]:
```
roscd
```

If everything goes fine, you should go to the following path:



```
ubuntu@ip-172-31-45-1:~$ roscd
ubuntu@ip-172-31-45-1:/opt/ros/kinetic$
```

Excellent! So you have successfully installed ROS in your local computer.

- End of Example A.1 -

Excellent! So you have successfully installed ROS in your local computer. In the next chapter, **Appendix 2**, you will see how to create new workspaces and how to manage them, so that you can start developing your own ROS packages.

# Appendix 2 Managing ROS workspaces

## A.2 How to manage ROS workspaces in my local computer?

During the whole Course, you have been working within a directory called **catkin_ws**. As you may already know, this directory is known as the **catkin workspace** or **ROS workspace**, and it is basically the place where you will have to place all the new ROS packages that you create. But... How do you create one of these workspaces? Is it possible to have more than one workspace?

During the Course, this workspace was already provided to you, so you just had to worry about putting all your packages inside there. But now, let's try to answer some of these interesting questions during the following Notebook.

### A.2.1 Create a ROS workspace

For this example, you can use the same Robot Ignite Academy environment you've been working during the whole Course. You will just add a new workspace to it.

First of all, let's create the folder where we will create our ROS workspace.

▶ Execute in Local Terminal #1

In [ ]:

```
mkdir -p ~/mynew_ws/src
```

Now, you will need to compile it by running the **catkin_make** command.

▶ Execute in Local Terminal #1

In [ ]:

```
cd ~/mynew_ws/
```

In [ ]:

```
catkin_make
```

You already know about **catkin_make** command, don't you? You've used it along the course in order to compile your packages. The **catkin_make** command is a convenience tool for working with catkin workspaces. Running it the first time in your workspace, it will create the **CMakeLists.txt** file in your **src** folder. Also, if you look in your current directory you should now have a **build** and **devel** folder. Inside the **devel** folder you can see that there are now several *setup.*sh* files. Sourcing any of these files will overlay this workspace on top of your environment.

▶ Execute in Local Terminal #1

In [ ]:

```
source devel/setup.bash
```

To make sure your workspace is properly overlayed by the setup script, you can check the **ROS_PACKAGE_PATH** environment variable with the following command.

▶ Execute in Local Terminal #1

In [ ]:

```
echo $ROS_PACKAGE_PATH
```

If everything goes fine, you should get the following:

In [ ]:

```
user:~/mynew_ws$ echo $ROS_PACKAGE_PATH
/home/user/mynew_ws/src:/home/user/catkin_ws/src:/home/simulations/public_sim_ws/src:/opt/ros/noetic/share
```

As you can see, your new workspace, named **mynew_ws**, is now on top of the **catkin_ws** you've been using during the whole course.

**NOTE**: Bear in mind that the IDE is configured to work only with 1 workspace, which is the **catkin_ws**, so you won't be able to visualize this new workspace within the IDE.

- Exercise A.1 -

a) Now, repeat the whole process with your local computer. Create a new workspace in your local ROS installation, which you completed in **Appendix 1**.

After you setup your workspace in your local ROS installation, you'll be able to start creating new ROS packages there, or even download and place there the ones you created during the Course!

- End of Exercise A.1 -

**Congratulations!** You are now able to download and use all the packages you created during the Course into your local machine. Enjoy it! And... keep pushing your ROS learning!

✓ English proofread