

## 1.5 How to prepare before the start?

Before we could start to use ROS2 Foxy we will need to install it to our system or somehow gain access to a system that has an already installed ROS2 distribution to it. For this goal to achieve we can select from the three options below:

1. Use the system available at **theconstruct.ai**
2. Use a **virtual machine** with Ubuntu and ROS2 or **WSL**
3. Use a **real Ubuntu system** with ROS2 (either dual-boot or just Ubuntu)

Let's talk about each of the three option shortly, to make our initial decision easier.

The easiest way from the three to start is obviously the first one. In this case, it is enough to register ourselves to the system and we will get access to the provider's own system. With the free subscription, we will get a personal storage space of 2GB and access for the development of a total of 8 hours/day. We will be allowed to work only on public projects. Another benefit is that we are allowed to access tutorials from various levels and we can browse from job opportunities as well. They also claim that it is possible to connect to our own robots which claim we could not confirm so far. This option is perfect for slow-paced learning purposes without bigger projects and higher development-time demand.

The second option gives us a bit more flexibility in terms of computational resources (CPU time and power, storage space, development time). The biggest drawback is that without complex solutions we can't access a physical robot. Also, we will experience a minor loss in our performance due to the virtual machine. This concludes, that until we don't need to move to the physical robots level, mostly we will do fine with this solution.

The third option is what gives us the most flexibility, hence we advise everyone, who wants to delve a little deeper in ROS or robot programming, to go in this direction if possible. The drawback is that if we don't have an already installed correct distribution of Ubuntu (for Foxy it is the Ubuntu Focal Fossa 20.04) then we need to install it (or make a dual-boot computer and install it). Now before deciding to go in this direction, it is important that if it is needed always store Your important data somewhere safe to prevent their loss.

Also needs to be mentioned, that however it is possible, we do not recommend installing it to Windows or macOS. Most of the experience collected (personally or from online forums) with these options suggests that they are not working alongside each other as with the Ubuntu.

Now, that we decided which option to choose let's move to our first step (except for those who choose the first option), and install ROS2 into our system.

## 1.6 Install ROS2

Now that we made our choice let's install ROS2 Foxy to our system. We will follow the official tutorial for the installation.<sup>1</sup>

---

<sup>1</sup><https://docs.ros.org/en/foxy/Installation/Ubuntu-Install-Debians.html>

### 1.6.1 Locale

First, we need to check the locale of our system and ensure that it supports UTF-8. We can do this with the following few lines:

```
1 $ locale # check for UTF-8
```

If our system does not support the UTF-8 then we can enforce it with the following few lines:

```
1 $ sudo apt update && sudo apt install locales
2 $ sudo locale-gen en_US en_US.UTF-8
3 $ sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
4 $ export LANG=en_US.UTF-8
5
6 $ locale # verify settings
```

### 1.6.2 Source

We add to our system the ROS2 apt repository, for which first we enable the Ubuntu Universe repository<sup>2</sup>:

```
1 $ sudo apt install software-properties-common
2 $ sudo add-apt-repository universe
```

Then, we add the GPG key with apt:

```
1 $ sudo apt update && sudo apt install curl -y
2 $ sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/
   ros.key -o /usr/share/keyrings/ros-archive-keyring.gpg
```

And finally, add the repository to our sources list

```
1 $ echo "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/
   keyrings/ros-archive-keyring.gpg] http://packages.ros.org/ros2/ubuntu
   $(. /etc/os-release && echo $UBUNTU_CODENAME) main" | sudo tee /etc/
   apt/sources.list.d/ros2.list > /dev/null
```

### 1.6.3 Installation of ROS2 packages

We should always check whether our system is up-to-date before installing ROS2 or new packages (sometimes this can take a while) since the packages are built on this frequently updated system:

```
1 $ sudo apt update
2 $ sudo apt upgrade
```

And finally, install. We will do so with the desktop version since it comes with lots of useful things, that we want to use throughout these practices. We also install the development tools (compilers and package building tools).

```
1 $ sudo apt install ros-foxy-desktop python3-argcomplete
2 $ sudo apt install ros-dev-tools
```

---

<sup>2</sup><https://help.ubuntu.com/community/Repositories/Ubuntu>

## 1.6.4 Environment setup

Every time we open a new terminal tab for using ROS2 (and we will have many), we need to source our setup.bash this can be made with the following line:

```
1 $ source /opt/ros/foxy/setup.bash
```

Alternatively, if we don't want to start typing this line of command every time we open a new tab we can add this line to our .bashrc file, which will automatically execute this command. **!! THIS IS ONLY ADVISED WHEN USING ONLY ONE DISTRIBUTION ON YOUR COMPUTER !!**

```
1 $ echo "source /opt/ros/foxy/setup.bash" >> ~/.bashrc
```

## 1.6.5 Try out some things

To try out our new installation we will run two nodes, that will communicate with each other. We don't need to implement these, they are already implemented. Open a new tab and give the following command:

```
1 ros2 run demo_nodes_cpp talker
```

Then open one more tab in the terminal and give this command:

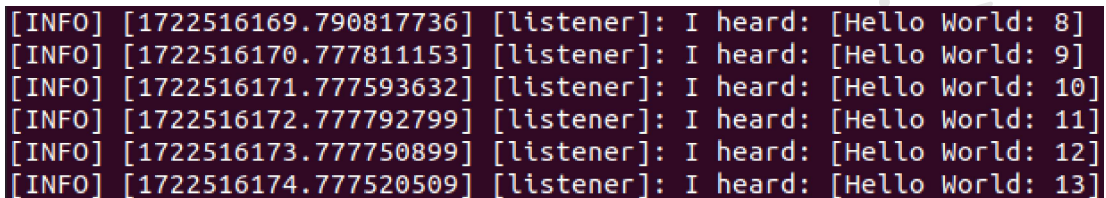
```
1 ros2 run demo_nodes_py listener
```

If we have done everything right, then we should see something similar in our two terminal windows:



```
[INFO] [1722516162.775390816] [talker]: Publishing: 'Hello World: 1'
[INFO] [1722516163.775310002] [talker]: Publishing: 'Hello World: 2'
[INFO] [1722516164.775347168] [talker]: Publishing: 'Hello World: 3'
[INFO] [1722516165.775427298] [talker]: Publishing: 'Hello World: 4'
[INFO] [1722516166.775417672] [talker]: Publishing: 'Hello World: 5'
[INFO] [1722516167.775469756] [talker]: Publishing: 'Hello World: 6'
[INFO] [1722516168.775526715] [talker]: Publishing: 'Hello World: 7'
[INFO] [1722516169.775558077] [talker]: Publishing: 'Hello World: 8'
[INFO] [1722516170.775577161] [talker]: Publishing: 'Hello World: 9'
[INFO] [1722516171.775486626] [talker]: Publishing: 'Hello World: 10'
[INFO] [1722516172.775587234] [talker]: Publishing: 'Hello World: 11'
[INFO] [1722516173.775611742] [talker]: Publishing: 'Hello World: 12'
[INFO] [1722516174.775642051] [talker]: Publishing: 'Hello World: 13'
```

(a) Result from the talker node.



```
[INFO] [1722516169.790817736] [listener]: I heard: [Hello World: 8]
[INFO] [1722516170.777811153] [listener]: I heard: [Hello World: 9]
[INFO] [1722516171.777593632] [listener]: I heard: [Hello World: 10]
[INFO] [1722516172.777792799] [listener]: I heard: [Hello World: 11]
[INFO] [1722516173.777750899] [listener]: I heard: [Hello World: 12]
[INFO] [1722516174.777520509] [listener]: I heard: [Hello World: 13]
```

(b) Result from the listener node.

Figure 1.1: Communication between nodes

For now, it is enough for us to understand that we just started two nodes, that are communicating with each other via some kind of channel. It is always possible to stop an execution of the node by simply pressing ctrl+C in the running node's terminal tab.

This concludes our introduction. We now have a functional ROS2 distribution installed in our system. In the next practice, we will start to use the most important basic modules and commands which will help us to navigate later when making projects.