How to Install ROS 2 Humble on Ubuntu 22.04

Set locale

Make sure you have a locale that supports UTF-8. If you are in a minimal environment (such as a docker container), the locale may be something minimal like POSIX. We test with the following settings. However, it should be fine if you're using a different UTF-8-supported locale.

```
Unset
sudo apt update && sudo apt install locales
sudo locale-gen en_US en_US.UTF-8
sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8
export LANG=en_US.UTF-8
```

Setup Sources

You will need to add the ROS 2 apt repository to your system.
First, ensure that the Ubuntu Universe repository is enabled.

```
Unset
sudo apt install software-properties-common
sudo add-apt-repository universe
```

Now add the ROS 2 GPG key with apt.

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

Then add the repository to your sources list.

```
Unset
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
```

Install ROS 2 packages

Update your apt repository caches after setting up the repositories.

```
Unset
sudo apt update && sudo apt upgrade
```

Finally, you can install the ROS 2 Desktop Install with RViz, demos, and tutorials.

```
Unset
sudo apt install ros-humble-desktop && sudo apt install ros-dev-tools
```

Environment setup

Set up your environment by sourcing the */opt/ros/humble/setup.bash* file in your *~/.bashrc* file. To do so, you can use the command

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

To verify that ROS 2 Humble is correctly installed you can use the command

```
Unset
ros2 --help
```

You should see something like this

```
usage: ros2 [-h] [--use-python-default-buffering]
            Call `ros2 <command> -h` for more detailed usage. ...

ros2 is an extensible command-line tool for ROS 2.

options:
  -h, --help            show this help message and exit
  --use-python-default-buffering
                        Do not force line buffering in stdout and instead use
                        the python default buffering, which might be affected
                        by PYTHONUNBUFFERED/-u and depends on whatever stdout
                        is interactive or not

Commands:
  action     Various action related sub-commands
  bag        Various rosbag related sub-commands
  component  Various component related sub-commands
  control    Various control related sub-commands
  daemon     Various daemon related sub-commands
  doctor     Check ROS setup and other potential issues
  interface  Show information about ROS interfaces
  launch     Run a launch file
  lifecycle  Various lifecycle related sub-commands
  multicast  Various multicast related sub-commands
  node       Various node related sub-commands
  param      Various param related sub-commands
  pkg        Various package related sub-commands
  run        Run a package specific executable
  security   Various security related sub-commands
  service    Various service related sub-commands
  topic      Various topic related sub-commands
  wtf        Use `wtf` as alias to `doctor`

  Call `ros2 <command> -h` for more detailed usage.
```

Furthermore, to verify the ROS 2 version you have installed, you can use the command

```
Unset
echo $ROS_DISTRO
```