

01_ROS 2 Humble install (Ubuntu 22.04)

1. Set System Locale (Prevent Garbled Text/Compatibility Issues)

```
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
```

Optional: Check if the current settings are successful

```
locale
```

2. Prepare Required Tool Dependencies for ROS Installation

```
sudo apt update && sudo apt install curl gnupg lsb-release
```

Update the system's package list as administrator and install three tools: curl, gnupg, and lsb-release.

3. Add ROS Repository Key and Source Address

```
sudo curl -sSL https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -
o /usr/share/keyrings/ros-archive-keyring.gpg
```

This step is to ensure the security of the software source and prevent forgery.

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

Add the official ROS 2 software source (with key verification) to prepare for the subsequent installation of ROS 2.

4. Update Software Sources and Install ROS 2 Humble Desktop Version

```
sudo apt update
sudo apt install ros-humble-desktop
```

⚠ The installation process may download 600MB+ of content and occupy about 3GB of space. Ensure sufficient disk space.

5. Configure Environment Variables (So You Can Use the `ros2` Command Directly)

Effective for the current terminal:

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

Automatically effective every time a terminal is opened:

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

6. Verify ROS Installation Success

```
ros2
```

Output similar to:

```
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
  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.

7. Run ROS Turtlesim Test

Step 1: Run the turtlesim graphical interface

```
ros2 run turtlesim turtlesim_node
```

Step 2: Open a new terminal to control the turtle's movement

```
ros2 run turtlesim turtle_teleop_key
```

Use the keyboard's `WASD` to control direction, `Q/E` to control rotation speed.

At this point, the following has been completed:

- Complete installation of ROS 2 Humble
- Environment variable configuration
- CLI availability verification
- Launch and control of a simulation node (turtlesim)