Install ROS 2 Jazzy on Ubuntu 24.04

In order to complete this guide, you will need a computer running Ubuntu 24.04 LTS (long term Support).

There are two main environment options for running Ubuntu. Option 1, Native Linux (Recommended), is the preferred method, as it utilizes a pure Linux operating system, specifically Ubuntu 24.04, for maximum performance and compatibility with ROS 2 tools and hardware drivers. Option 2, WSL on Windows (Windows Subsystem for Linux), offers a practical alternative for users working on Windows machines by allowing a full Ubuntu environment to run directly within Windows, which is ideal for developers who need to switch easily between the two systems.

The quickest way to install the necessary components for WSL, including the Ubuntu distribution, is using the following command in a terminal:

1. Click the Start menu, type cmd or PowerShell, right-click on the result, and select "Run as administrator" .
2. Run the installation command: `wsl -install`

💡 Hints & Tips

- Throughout this guide, you will find Linux commands as follows:

```
This is a linux command
```

- Use copy/paste to avoid typing errors.

We will begin by installing ROS 2 Jazzy via Debian Packages. Debian packages are software files used to install programs and applications on Ubuntu.

Set the Locale

Open a new terminal window and type this command inside a terminal window:

```
locale
```

A locale is a set of variables that define the language, country, and character encoding settings. These settings are used by applications to determine how to display text, dates, times, and other information.

Now type the following command:

```
sudo apt update && sudo apt install locales
```

"sudo apt update" updates the package index list. This list is a database of all the software packages available for your version of Ubuntu. "sudo apt install locales" installs the locales package, which provides support for different languages and regions.

Now type the following commands into the terminal window to generate locale definition files. After each command, press Enter on your keyboard:

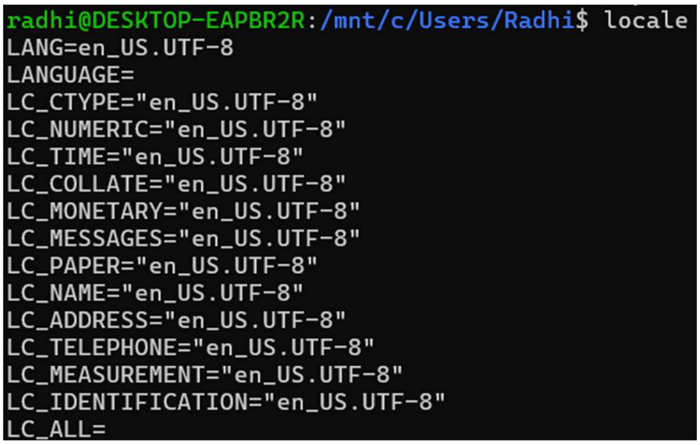
```
sudo locale-gen en_US en_US.UTF-8

sudo update-locale LC_ALL=en_US.UTF-8 LANG=en_US.UTF-8

export LANG=en_US.UTF-8
```

In this case, these commands are generating locale definition files for the English (United States) locale and the English (United States) UTF-8 locale. The UTF-8 locale is a special locale that supports the UTF-8 character encoding, which is the standard encoding for most languages.

Now we need to verify the settings by typing:

```
locale
```

Here is what you should see:



```
radhi@DESKTOP-EAPBR2R:/mnt/c/Users/Radhi$ locale
LANG=en_US.UTF-8
LANGUAGE=
LC_CTYPE="en_US.UTF-8"
LC_NUMERIC="en_US.UTF-8"
LC_TIME="en_US.UTF-8"
LC_COLLATE="en_US.UTF-8"
LC_MONETARY="en_US.UTF-8"
LC_MESSAGES="en_US.UTF-8"
LC_PAPER="en_US.UTF-8"
LC_NAME="en_US.UTF-8"
LC_ADDRESS="en_US.UTF-8"
LC_TELEPHONE="en_US.UTF-8"
LC_MEASUREMENT="en_US.UTF-8"
LC_IDENTIFICATION="en_US.UTF-8"
LC_ALL=
```

Enable the Required Repositories

Let's add the ROS 2 apt repository to our system. APT stands for "Advanced Package Repository". This repository provides a convenient way to install and manage ROS 2 packages without having to clone packages to your computer from GitHub and build them from that source code.

Open a terminal window, and type the following two commands:

```
sudo apt install software-properties-common

sudo add-apt-repository universe
```

The software-properties-common package provides a number of tools for managing software sources on Ubuntu and Debian systems. The universe repository is a software repository that contains a wide variety of software packages, including many that are not included in the default Ubuntu and Debian repositories.

Now we need to add the ROS 2 GPG key with apt. The ROS 2 GPG key makes sure the software packages you are installing are from a trusted source.

Type these two commands:

```
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
```

Add the repository to your sources list (copy and paste all of this below):

```
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
```

Type the following command to install ROS 2 development tools.

```
sudo apt update && sudo apt install ros-dev-tools
```

Upgrade the packages onto your system to make sure you have the newest versions.

```
sudo apt upgrade -y
```

Install ROS 2

Now you can install ROS 2 Jazzy. Open a terminal window, and type this command:

```
sudo apt install ros-jazzy-desktop
```

Set Up the Environment Variables

Once jazzy has finished installing, you need to set up the important environment variables. Environment variables are settings that tell your computer how to find and use ROS 2 commands and packages.

Open a terminal window, and type this command:

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

When you run this command, it appends the line source /opt/ros/jazzy/setup.bash to your ~/.bashrc file .

What does this do? Each time you open a new terminal window, you are starting what is called a bash session . The bash session needs to know what version of ROS 2 you are using.

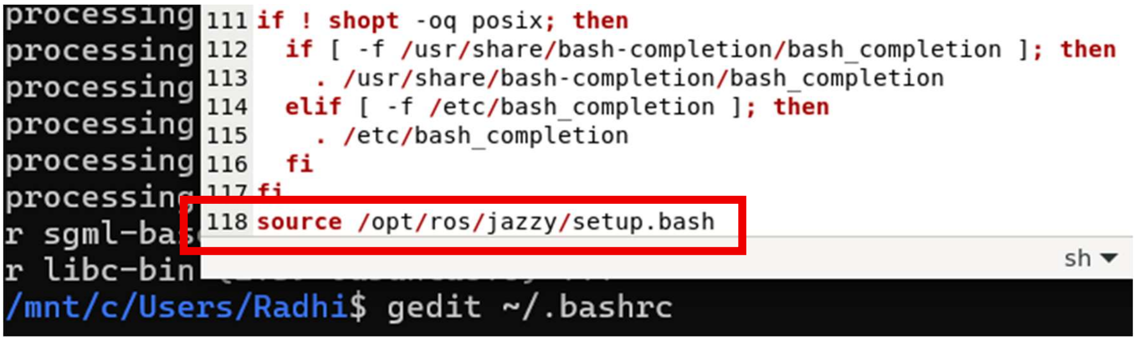
By adding this line (echo “source /opt/ros/jazzy/setup.bash”) to your ~/.bashrc file, you ensure the necessary environment variables and paths for ROS 2 Jazzy are properly set up each time you open a new terminal window, allowing you to use ROS 2 commands and tools without having to manually run the setup.bash script every time.

For the changes to take effect, you now need to open a new terminal window, or you can type this command in the current terminal:

```
source ~/.bashrc
```

You can verify that line was added by typing:

```
sudo apt-get install gedit -y

gedit ~/.bashrc
```



```
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
source /opt/ros/jazzy/setup.bash
```

```
processing
processing
processing
processing
processing
processing
r sgml-ba
r libc-bin
/mnt/c/Users/Radhi$ gedit ~/.bashrc
```

Close the gedit window.

Check Your ROS 2 Version

Now let's see what ROS 2 version we are using:

```
printenv ROS_DISTRO
```

You should see "jazzy".

Set up the system to manage ROS package dependencies.

```
sudo rosdep init

rosdep update
```

Install the Terminator tool

There is a cool program that enables you to show multiple terminal windows in a single window. The program is called "*terminator*".

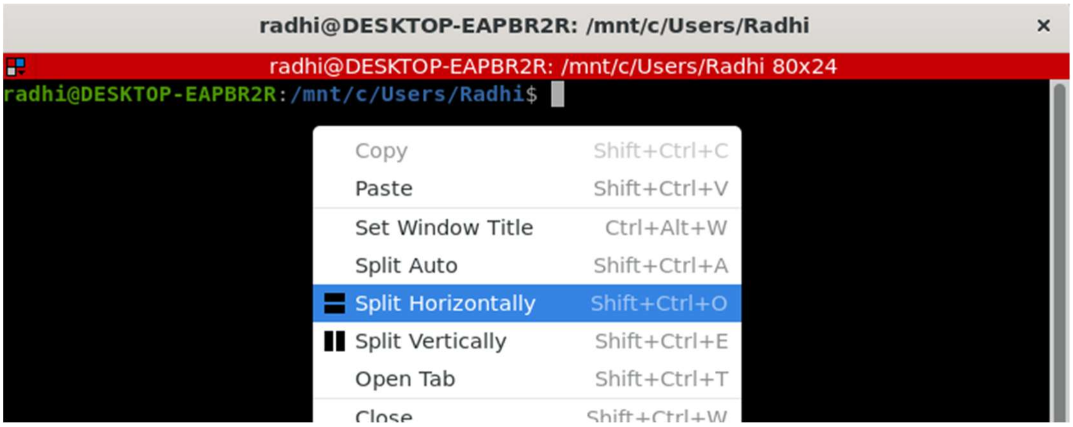
Open a terminal window, and type:

```
sudo apt-get install terminator -y
```

Now type:

```
terminator
```

Right-click on the screen and click "Split Horizontally".

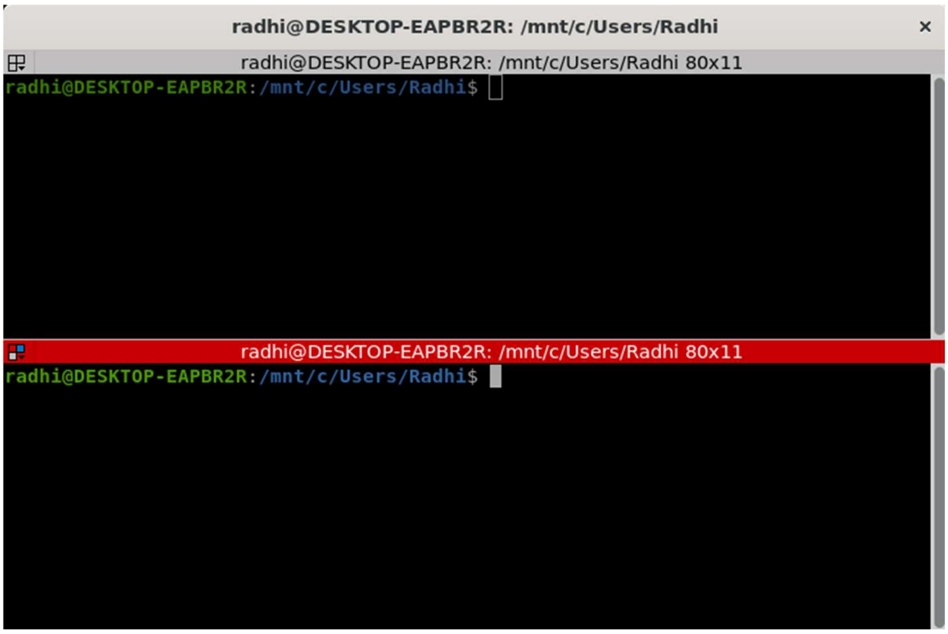


radhi@DESKTOP-EAPBR2R: /mnt/c/Users/Radhi

radhi@DESKTOP-EAPBR2R: /mnt/c/Users/Radhi 80x24

```
radhi@DESKTOP-EAPBR2R:/mnt/c/Users/Radhi$
```

You get this:


radhi@DESKTOP-EAPBR2R: /mnt/c/Users/Radhi
×
radhi@DESKTOP-EAPBR2R: /mnt/c/Users/Radhi 80x11
radhi@DESKTOP-EAPBR2R: /mnt/c/Users/Radhi 80x11

```
radhi@DESKTOP-EAPBR2R:/mnt/c/Users/Radhi$
```

```
radhi@DESKTOP-EAPBR2R:/mnt/c/Users/Radhi$
```

Test Your ROS 2 Installation

Now, you can test your ROS 2 installation by running some sample programs. On the top panel, type:

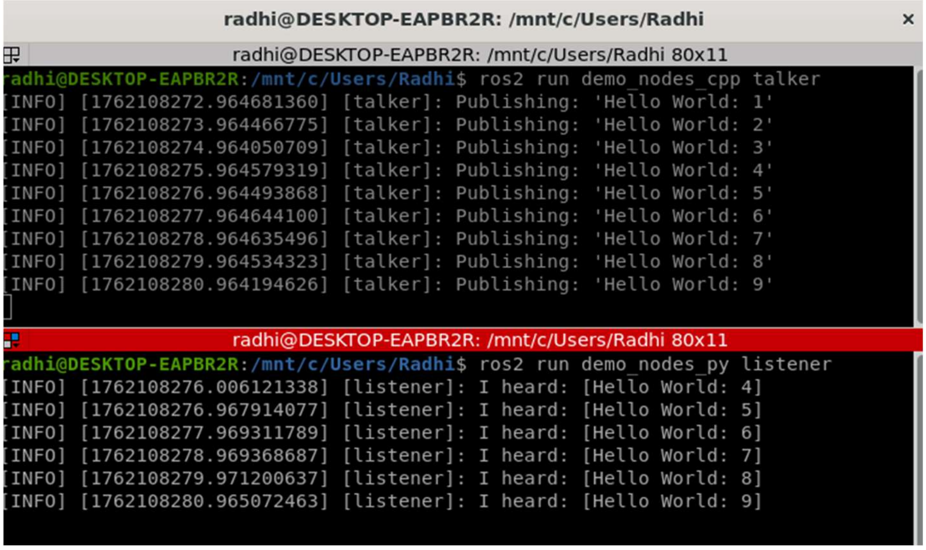
```
ros2 run demo_nodes_cpp talker
```

On the bottom panel, type:

```
ros2 run demo_nodes_py listener
```

Now, press Enter in both terminals.

Here is what you should get:


radhi@DESKTOP-EAPBR2R: /mnt/c/Users/Radhi
×
radhi@DESKTOP-EAPBR2R: /mnt/c/Users/Radhi 80x11
radhi@DESKTOP-EAPBR2R: /mnt/c/Users/Radhi 80x11

```
adhi@DESKTOP-EAPBR2R:/mnt/c/Users/Radhi$ ros2 run demo_nodes_cpp talker
[INFO] [1762108272.964681360] [talker]: Publishing: 'Hello World: 1'
[INFO] [1762108273.964466775] [talker]: Publishing: 'Hello World: 2'
[INFO] [1762108274.964050709] [talker]: Publishing: 'Hello World: 3'
[INFO] [1762108275.964579319] [talker]: Publishing: 'Hello World: 4'
[INFO] [1762108276.964493868] [talker]: Publishing: 'Hello World: 5'
[INFO] [1762108277.964644100] [talker]: Publishing: 'Hello World: 6'
[INFO] [1762108278.964635496] [talker]: Publishing: 'Hello World: 7'
[INFO] [1762108279.964534323] [talker]: Publishing: 'Hello World: 8'
[INFO] [1762108280.964194626] [talker]: Publishing: 'Hello World: 9'
```

```
adhi@DESKTOP-EAPBR2R:/mnt/c/Users/Radhi$ ros2 run demo_nodes_py listener
[INFO] [1762108276.006121338] [listener]: I heard: [Hello World: 4]
[INFO] [1762108276.967914077] [listener]: I heard: [Hello World: 5]
[INFO] [1762108277.969311789] [listener]: I heard: [Hello World: 6]
[INFO] [1762108278.969368687] [listener]: I heard: [Hello World: 7]
[INFO] [1762108279.971200637] [listener]: I heard: [Hello World: 8]
[INFO] [1762108280.965072463] [listener]: I heard: [Hello World: 9]
```

Press CTRL + C in both terminal windows to close everything.

Install Turtlesim

Turtlesim is a lightweight simulator for learning ROS 2. It illustrates what ROS 2 does at the most basic level to give you an idea of what you will do with a real robot or a robot simulation later on.

Install the turtlesim package for ROS 2 jazzy:

```
sudo apt update

sudo apt install ros-jazzy-turtlesim
```

To check if the package is installed, run the following command, which should return a list of turtlesim's executables:

```
ros2 pkg executables turtlesim
```