

ROS 2 Humble Hawksbill Setup & Configuration (Full Desktop)

- **Target OS:** Ubuntu 22.04 LTS (recommended for ROS 2 Humble)
- **Goal:** Set up a complete ROS 2 development environment with Gazebo, VS Code, Colcon, Git, and multiterminal tools.

1. Update and Upgrade the System

Always start by updating your environment.

```
bash
sudo apt update && sudo apt upgrade -y
sudo apt install curl gnupg lsb-release -y
```

2. Set Up the ROS 2 Repository

Add the official ROS 2 APT repository.

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

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

Then update the repository:

```
bash
sudo apt update
```

3. Install ROS 2 Humble (Full Desktop Version)

This includes:

- Core ROS 2 packages
- Rviz2 visualization tool
- Gazebo/Ignition integration
- Demo nodes

```
bash
sudo apt install ros-humble-desktop -y
```

Alternative minimal installation (if needed):

```
bash
sudo apt install ros-humble-ros-base -y
```

4. Source ROS 2 Environment Automatically

Add ROS 2 setup to your bash profile:

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

Test installation:

```
bash
ros2 --version
```

- Expected output:
ros2 0.18.5 (or similar version)

5. Install Colcon Build Tools

Colcon is the official build tool for ROS 2 workspaces.

```
bash  
  
sudo apt install python3-colcon-common-extensions -y
```

6. Create a ROS 2 Workspace

```
bash  
  
mkdir -p ~/ros2_ws/src  
cd ~/ros2_ws  
colcon build  
  
source install/setup.bash
```

Verify everything works:

```
bash  
  
ros2 run demo_nodes_cpp talker
```

and in another terminal:

```
bash  
  
ros2 run demo_nodes_cpp listener
```

7. Install Gazebo Fortress (Ignition)

Gazebo Fortress is the ROS 2-supported version for Humble.

Step 1: Add Ignition repository

```
bash  
  
sudo apt update  
sudo apt install lsb-release wget gnupg -y  
sudo wget https://packages.osrfoundation.org/gazebo.gpg -O  
/usr/share/keyrings/pkgs-osrf-archive-keyring.gpg  
  
echo "deb [arch=$(dpkg --print-architecture)  
signed-by=/usr/share/keyrings/pkgs-osrf-archive-keyring.gpg]  
http://packages.osrfoundation.org/gazebo/ubuntu-stable $(lsb_release -cs)  
main" | sudo tee /etc/apt/sources.list.d/gazebo-stable.list > /dev/null
```

```
sudo apt update
```

Step 2: Install Gazebo Fortress

```
bash  
sudo apt install gz-fortress -y
```

Run a quick test:

```
bash  
gz sim
```

Trainer note:

Explain that Gazebo (now Ignition) is used for physics-based simulation – where they'll test robots before running code on real hardware.

8. Install VS Code

Visual Studio Code will be used for ROS 2 development.

```
bash  
sudo snap install code --classic
```

Recommended extensions:

- ms-python.python
- ms-vscode.cpptools
- ms-iot.vscode-ros
- eamodio.gitlens

9. Install Git and GitHub CLI

For version control and collaboration.

```
bash  
sudo apt install git gh -y  
git config --global user.name "Your Name"  
git config --global user.email "your.email@example.com"
```

Login to GitHub:

```
bash  
gh auth login
```

10. Install Multi-Terminal Tool (Terminator or Tmux)

ROS 2 often requires multiple terminals for running nodes.

Option 1: Terminator

```
bash  
sudo apt install terminator -y
```

Launch with:

```
bash  
terminator
```

Option 2: Tmux (keyboard-based)

```
bash  
sudo apt install tmux -y
```

11. Verification Checklist

Tool	Command	Expected Output
ROS 2	<code>ros2 --version</code>	ROS 2 Humble version
Colcon	<code>colcon --version</code>	Version info
Gazebo	<code>gz sim</code>	Opens simulator window

VS Code	<code>code</code>	Launches editor
Git	<code>git --version</code>	Git version
Terminator	<code>terminator</code>	Multi-terminal UI opens