# ROS 2 Humble and Gazebo Fortress Setup Guide

## TL;DR: Quick Terminal Commands

For someone who has already completed the setup and just needs the commands, here is a quick reference for the three essential terminal processes:

**Terminal 1: Launch Gazebo Simulation**

Launch the Gazebo GUI, open a world file with a camera, and click play.

**Terminal 2: Launch the ROS 2 Bridge**

This bridges the Gazebo topics to ROS 2.

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

*ros2 run ros_gz_bridge parameter_bridge /camera@sensor_msgs/msg/Image@gz.msgs.Image /camera_info@sensor_msgs/msg/CameraInfo@gz.msgs.CameraInfo*

**Terminal 3: Publish the Static Transform**

This provides the coordinate frame data for RViz2.

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

*ros2 run tf2_ros static_transform_publisher --x 0 --y 0 --z 0 --roll 0 --pitch 0 --yaw 0 --frame-id world --child-frame-id camera/link/camera*

**Terminal 4: Launch RViz2**

This is where you visualize the camera feed.

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

*rviz2*

In RViz2, set the `Fixed Frame` to `world` and add an `Image` display with the `/camera` topic.

***This document provides a complete, step-by-step guide for setting up a development environment to get ROS 2 Humble communicating with Gazebo Sim Fortress. This setup is ideal for replicating a successful robotics simulation pipeline on a clean Ubuntu 22.04 LTS virtual machine.***

## Step 1: Install Ubuntu 22.04 LTS

Begin by creating a new virtual machine on your M-series MacBook using a virtualization application like UTM. Install **Ubuntu 22.04 LTS (Jammy Jellyfish)**. This version is the officially recommended OS for ROS 2 Humble.

If you're installing from the server ISO, you'll need to install the desktop environment after the initial setup. Log in to the server terminal and run these commands:

sudo apt update
sudo apt install ubuntu-desktop
sudo reboot

## Step 2: Install ROS 2 Humble and Gazebo Sim Fortress

To ensure a compatible software stack, we will install the ros-humble-ros-gz package, which automatically pulls in **Gazebo Sim Fortress (version 6)** as a dependency.

1. **Add the ROS 2 repository:**
   sudo apt update && sudo apt install curl
   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 $(. /etc/os-release && echo $UBUNTU_CODENAME) main" | sudo tee /etc/apt/sources.list.d/ros2.list > /dev/null

2. **Install ROS 2 Humble and the ros-gz package:**
   sudo apt update
   sudo apt upgrade
   sudo apt install ros-humble-ros-gz ros-humble-desktop

This command installs ROS 2 Humble, RViz2, and the compatible ros-gz bridge, which

automatically includes Gazebo Sim Fortress.

## Step 3: Run the Gazebo Simulation

You can launch Gazebo from your desktop's application drawer. This is a crucial step that you performed correctly.

1. Go to your app drawer and open **Gazebo Ignition 6**.
2. In the Gazebo GUI, go to **File > Open World**.
3. Select the **camera_sensor.sdf** world file, which should be located in a path like /usr/share/gz-sim-6/worlds/.
4. Once the world loads, click the **Play** button in the bottom-left corner.

## Step 4: Bridge Gazebo and ROS 2 Topics

With Gazebo running, you need to launch the ros_gz_bridge to connect the simulation's topics to the ROS 2 environment. You'll bridge both the image data and the camera's calibration information.

Open a terminal and run the following command. The bridge must be running to maintain the connection.

```
source /opt/ros/humble/setup.bash
ros2 run ros_gz_bridge parameter_bridge /camera@sensor_msgs/msg/Image@gz.msgs.Image
/camera_info@sensor_msgs/msg/CameraInfo@gz.msgs.CameraInfo
```

## Step 5: Publish a Static Transform

RViz2 requires a coordinate frame (tf) to display data correctly. Since your world doesn't have a robot model publishing this data, you must manually publish a static transform.

Open a **new terminal** and run this command:

```
source /opt/ros/humble/setup.bash
ros2 run tf2_ros static_transform_publisher --x 0 --y 0 --z 0 --roll 0 --pitch 0 --yaw 0
--frame-id world --child-frame-id camera/link/camera
```

The log output confirms the transform is being published:

```
[INFO] [1755621112.266888476] [static_transform_publisher_OXoHAjT5afYmByyP]: Spinning
until stopped - publishing transform
translation: ('0.000000', '0.000000', '0.000000')
rotation: ('0.000000', '0.000000', '0.000000', '1.000000')
from 'world' to 'camera/link/camera'
```

## Step 6: Visualize in RViz2

Finally, you can launch RViz2 and see the camera feed.

1. Open a **third terminal** and run:
   source /opt/ros/humble/setup.bash
   rviz2

2. In the RViz2 GUI, in the **Displays** panel on the left:
   - Under **Global Options**, set the **Fixed Frame** to world.
   - Click **Add** and select the **Image** display.
   - Set the Image display's **Topic** to /camera.

You should now see the live camera feed from Gazebo, proving that your entire pipeline is working correctly.