ROS1 - ROS2 Bridge Setup using Docker

Overview

This document provides step-by-step instructions to manually set up a ROS1 (Noetic) - ROS2 (Humble) bridge using Docker containers. The setup involves creating three separate containers: one each for ROS1 and ROS2, and a third container for the bridge itself.

Steps to Build the ROS1 - ROS2 Bridge

1. Create Docker Network

This network allows the ROS1, ROS2, and bridge containers to communicate.

Command:

```
docker network create ros_net
```

2. Run ROS1 Container

Command:

```
docker run -it --rm --name ros1 --network ros_net ros:noetic
```

Inside container:

```
source /opt/ros/noetic/setup.bash
rostopic pub /chatter std_msgs/String "data: 'Hello from ROS1'" -r 1
```

3. Run ROS2 Container

Command:

```
docker run -it --rm --name ros2 --network ros_net ros:humble
```

Inside container:

```
source /opt/ros/humble/setup.bash
ros2 topic echo /chatter
```

4. Build ROS1 - ROS2 Bridge Container

Create a Dockerfile named `Dockerfile.bridge` with contents such as installing ROS1 and ROS2 together, colcon, and other dependencies.

Build the image:

```
docker build -f Dockerfile.bridge -t ros1_bridge_manual .
```

5. Run Bridge Container

Command:

```
docker run -it --rm --name bridge --network ros_net
ros1_bridge_manual
```

Inside container:

```
source /opt/ros/noetic/setup.bash
source /opt/ros/humble/setup.bash (if available)
source install/setup.bash (if bridge built manually)
ros2 run ros1_bridge dynamic_bridge
```

6. Troubleshooting Notes

- Ensure all images use compatible Ubuntu versions (Noetic: Ubuntu 20.04, Humble: Ubuntu 22.04).

- Key error: `Package 'ros1_bridge' not found` implies bridge isn't built or sourced correctly.

- Key error: `Unable to communicate with master!` indicates ROS_MASTER_URI or networking misconfig.

- Ensure `apt` issues are resolved by matching correct repository versions (Noetic uses Focal).