📌 Setting Up a ROS 2 Development Environment Using DevContainers

🚀 *This guide will help you set up and use a **DevContainer** to develop and run a ROS 2 Jazzy-based project inside a containerized environment.*

📌 Prerequisites

Before starting, ensure you have the following installed:

1️⃣ Install Docker

- **Windows & Mac:** Install [Docker Desktop](#)
- **Linux:** Follow [Docker Install Guide](#)

2️⃣ Install VS Code & DevContainers Extension

- **Download VS Code**: [VS Code Website](#)
- **Install the DevContainers Extension**:
    - Open **VS Code**
    - Press `Ctrl+Shift+X` (Extensions Marketplace)
    - Search for **"Dev Containers"** and install it

📌 Step 1: Prepare Your Workspace Directory

Before launching the DevContainer, **create a directory where all your ROS 2 repositories will be stored**.

mkdir -p ~/ros2_projects
cd ~/ros2_projects

- ◆ **Why?**

    - This **folder will be mounted into the DevContainer** using `ROS2_SRC`.
    - All your **ROS 2 code will persist** outside the container.

📌 Step 2: Clone the DevContainer Repository

This repository contains the **DevContainer setup files**.

git clone https://github.com/your-org/devcontainer-repo.git
cd devcontainer-repo
Open VSCode here using: 'code .' if on Linux

📌 Step 3: Configure the `.env` File

Modify the `.env` file to **match your setup**.

nano .env

Modify these variables:

# User Information
GIT_USERNAME=your-github-username
GIT_USERTOKEN=your-personal-access-token # Required for private repositories
USER_EMAIL=your-email@example.com
SYS_USER=user

# ROS 2 Configuration
ROS_DIST=jazzy
ROS2_WKSPC=colcon_ws

# Mount Paths
ROS2_SRC=~/ros2_projects # Path to the folder where all ROS 2 repositories are stored

- **Why?**

  - `ROS2_SRC` mounts your workspace inside the container.
  - `GIT_USERNAME` and `GIT_USERTOKEN` allow **private repo cloning**. - Not needed for this project

---

📌 Step 4: Open the DevContainer in VS Code

1️⃣ **Open VS Code**
2️⃣ **Click on "File" → "Open Folder"**
3️⃣ Select the **devcontainer-repo** directory
4️⃣ Press `Ctrl+Shift+P`, then type:

Dev Containers: Reopen in Container

5️⃣ **Wait for the container to build (~5-10 minutes on first launch).**

📌 Step 5: Verify the DevContainer Setup

Once the container starts, open a **new terminal in VS Code** and run:

printenv | grep ROS2

Expected output:

ROS2_WKSPC=colcon_ws
ROS2_SRC=/workspace/ros2_projects

✅ Your ROS 2 workspace is **mounted and accessible** inside the container.

📌 Step 6: Clone ROS 2 Repositories

If you haven't cloned the project repositories yet, do it **inside the mounted workspace**:

cd /workspace/ros2_projects
git clone git@github.com:leocorp96/devspace_ros2_jazzy.git

📌 Step 7: Build the ROS 2 Workspace

Now build your ROS 2 packages using **colcon**:

```
cd /workspace/ros2_projects/colcon_ws
colcon build --symlink-install
```

💡 **After building, always source the workspace:**

```
source install/setup.bash
```

📌 Step 8: Running the Project

Check the next page on information regarding bringing up the stack

---

📌 Stopping & Restarting the DevContainer

To Stop the DevContainer

Simply close VS Code or run:

```
docker stop <container_id>
```

Find the container ID using:

```
docker ps
```

To Restart the DevContainer

1️⃣ Open VS Code
2️⃣ Press `Ctrl+Shift+P`
3️⃣ Select **"Dev Containers: Reopen in Container"**

📌 Troubleshooting

Issue	Solution

DevContainer doesn't start	Check Docker is running (`docker ps`)
`colcon: command not found`	Run `sudo apt install python3-colcon-common-extensions`
Missing dependencies	Run `rosdep install --from-paths src --ignore-src -r -y`
ROS 2 packages not found	Ensure `source install/setup.bash` is run