ROS2-Docker-Template

- This is a template for organizing ROS2 distributions into separate workspaces with Docker Engine
- It assumes a Ubuntu machine, and some basic knowledge of Docker and ROS2
- It can be used in one of two ways, detailed below:
  - **Regular Docker** template on `main` branch -> normal `Dockerfile` + `docker-compose.yml`, with script for ease-of-use
  - **VS Code Dev Containers** template on `dev-containers` branch -> designed for use in VS Code with the Dev Containers extension

1. Regular Docker Method

Quick Start

1. Install Docker Engine with `apt` (not the VM Docker Desktop)**:** https://docs.docker.com/engine/install/ubuntu/#prerequisites
2. Clone the `main` branch of this repo
3. Each workspace lives in its own folder named after the ROS2 distribution, such as `humble`, and includes 3 files:`Dockerfile`, `docker-compose.yml`, and ease-of-use scripts in `dev.sh`

Daily commands

1. `cd` to the project with the ROS2 version you need, for example `ros2-docker-template/humble`

2. Copy your `src` directory for ROS2 workspace into the `[ROS_DISTRO]/src` folder - it is linked with Docker volumes to `src` inside the container

3. `cd` into the `ros2-docker-template/[ROS_DISTRO]/scripts` subfolder to use `dev.sh` shortcuts

   1. start a container and enter it

      ```
      ./dev.sh start
      ```

   2. stop a container

      ```
      ./dev.sh stop
      ```

   3. rebuild a container

      ```
      ./dev.sh rebuild
      ```

   4. start a new shell inside a container

```
   ./dev.sh new
   ```

   5. start a new root shell inside a container

   ```
   ./dev.sh root
   ```

   6. get container status

   ```
   ./dev.sh status
   ```

4. verify an environment (sanity check)

   ```
   # Check ROS2 installation
   ros2 --help

   # Verify distro
   echo $ROS_DISTRO
   ```

5. `docker ps` shows all running Docker containers

6. stop one container (for example Humble) before starting another (for example Jazzy)

Setting up a new Docker container

1. To setup a workspace for a different ROS version, create a copy of an existing container folder, such as `ros2-docker-template/humble` and copy it with a new name, for ex for foxy it would be `ros2_docker_template/foxy`
2. In the Dockerfile, replace the image with the correct `osrf/ros` found at https://hub.docker.com/r/osrf/ros/tags, using the same format of `osrf/ros:[ROS_DISTRO]-desktop-full`
3. Replace all instances of `humble` with the new distribution, such as `foxy` across all 3 files (`Dockerfile, docker-compose.yml, dev.sh`)

Note on installing dependencies

- Because of the `rm -rf /var/lib/apt/lists/*` command, `sudo apt install` won't work by default for installing packages - first run `sudo apt update` and `sudo apt upgrade` and then it will work

2. VS Code Dev Containers Method

Quick Start

1. Clone the `dev-containers` branch of this repo
2. Open the subfolder with the ROS version you wish to use in VSCode, for example `ros2-docker-template/foxy`
3. Install Docker Engine with `apt` (not the VM Docker Desktop)**:** https://docs.docker.com/engine/install/ubuntu/#prerequisites
4. Install VSCode Dev Containers extension - https://code.visualstudio.com/docs/devcontainers/containers
   - More on Dev Containers at this tutorial by **Articulated Robotics**: https://www.youtube.com/watch?v=dihfA7Ol6Mw
5. Open VS Code Command Palette (**CTRL + P**), type `>Dev Containers: Reopen in Container`, and select it
6. VS Code will open a new window that contains an integrated IDE inside the Docker container, according to the instructions inside `.devcontainer`

`f1tenth_gym_ros` implementation branch

- this branch - `impl/dev-containers-f1tenth-gym-ros` is an implementatino of the `dev-containers` branch, using the `docker` branch of the `f1tenth_gym_ros` project fork: https://github.com/TeoIlie/F1TENTH_Gym_ROS/tree/docker
- pull the `src` code from the `.gitmodules` file with

```
git submodule update --init --force --remote
```