Jetson Nano

Documentation: Daniel Sanei

Docker

- Docker containers are ephemeral
    - Changes inside container do not persist after container stops
    - Solution: use volume mounts
        - -v /host/path:/container/path
        - Volume mounts ensure that code files, configurations, and data are saved on the host and remain accessible even after the container is restarted
- Host Directory: directory on host machine that stores ROS2 project files, scripts, and data
- Docker Volume Mount: maps `ros2_humble_workspace` directory from host machine to `/workspace` directory inside Docker container
    - Files in `ros2_humble_workspace` on host are accessible inside Docker container at `/workspace`
    - Changes made inside Docker container in `/workspace` will be reflected in `ros2_humble_workspace` on the host

```
# change file ownership
sudo chown -R inspiration:inspiration ~/ros2_humble_workspace
```

Device Mapping

- Allows specific hardware devices (i.e. cameras, USB devices) on the host to be accessible within the Docker container

- Ensures the container can interact with hardware like it would natively

Privileged Mode

- Grants the Docker container extended permissions, similar to root access on the host system
- Provides the container with all the capabilities of the host, enabling access to hardware and system resources

MAVROS

- Package that acts as a bridge between ROS and MAVLink, a communication protocol used by flight controllers like Pixhawk
- Enables control and communication with Pixhawk using ROS topics, services, and actions
- Facilitates integration of drone hardware with ROS-based software for tasks like autonomous flight and sensor data processing

Docker Containers

- RoboFlow Inference Server Docker (Tin Can) → 9 GB
- ROS2 Humble Docker Container → 13.3 GB

ROS2 Humble Docker Container

```
# pull Docker container
docker pull dustynv/ros:humble-ros-base-l4t-r36.2.0

# with memory (?)
docker run --runtime nvidia -it --network=host dustynv/ros:humbl

# without memory
```

```
sudo docker run --runtime nvidia -it --rm --network=host dustynv

# with ros2 humble workspace
sudo docker run --runtime nvidia -it --network=host -v ~/ros2_hu

# with device mapping (Arducam)
sudo docker run --runtime nvidia -it --network=host --privileged
--device=/dev/video0

# with device mapping (Pixhawk)
sudo docker run --runtime nvidia -it --network=host --privileged
--device=/dev/ttyACM0

# with device mapping (Arducam and Pixhawk)
sudo docker run --runtime nvidia -it --network=host --privileged
```

Install Packages

```
# update
sudo apt update

# MAVROS
sudo apt install ros-humble-mavros ros-humble-mavros-extras

# Video4Linux
sudo apt install v4l-utils

# command-line tool to capture images from video devices
sudo apt install fswebcam

# OpenCV
sudo apt install python3-opencv

# source ROS2 environment
```

```
source /opt/ros/humble/install/setup.bash
source /opt/ros/humble/setup.bash

# verify MAVROS installation
ros2 pkg list | grep mavros

# verify camera access
v4l2-ctl --list-devices
```

Inference Docker Container

```
# start inference container
sudo docker run --net=host --gpus all roboflow/inference-server

# start new inference container
sudo docker run --privileged --net=host --runtime=nvidia --mount
```

Helpful Commands

```
# check Arducam connection
ls /dev/video*

# check Pixhawk connection
ls /dev/ttyACM0

# list video devices with details (including names and paths)
v4l2-ctl --list-devices

# remove file
rm -rf [file]

# check Jetpack version
```

```
dpkg-query --show nvidia-l4t-core

# check Ubuntu version
lsb_release -a

# copy file to new directory
[sudo] cp ~/[file_to_send] ~/[new_directory]
```

Cleaning

```
# check disk space
df -h

# clears local repository of retrieved package files
sudo apt-get clean

# removes outdated package files from archives
sudo apt-get autoclean

# removes unusued packages
sudp apt-get autoremove
```

Arducam Live Feed (ROS2)

```
# check if Arducam accessible
v4l2-ctl --list-devices

# create ROS2 package for camera node
ros2 pkg create --build-type ament_python my_camera_package
cd my_camera_package

# create camera node script
nano my_camera_package/my_camera_package/camera_node.py
```

```
# include new node
nano setup.py

# build package
colcon build
source install/setup.bash

# run camera node
ros2 run my_camera_package camera_node

----------------------------------------------------------------

# capture Arducam image from ROS2 Humble Docker container
fswebcam -d /dev/video0 /workspace/test_image.jpg
exit
ls ~/ros2_humble_workspace/test_image.jpg
xdg-open ~/ros2_humble_workspace/test_image.j
```

Arducam Automatic Script

```
# line of code to run script upon Jetson Nano startup
nano ~/.bashrc
python3 /path/to/your/arducam_script.py
source ~/.bashrc
```

YOLO8

```
# clone YOLOv8 repository
git clone https://github.com/ultralytics/ultralytics.git

# install dependencies
```

```
cd ultralytics
pip install .

# download and unzip dataset
curl -L "https://universe.roboflow.com/ds/95r0htTjlw?key=qkMR7w
roboflow.zip
unzip roboflow.zip
rm roboflow.zip

# curl command to run inference on an image
base64 test_image.jpg | curl -d @- "https://detect.roboflow.com/
```

- Download Dataset
  - RoboFlow Universe → Overview → Download this Dataset → Format = YOLOv8 (show download code) → >_ Terminal → Use curl command
- Inference Command
  - RoboFlow Workspace ⟶ Select Project ⟶ Versions ⟶ Use curl command

MAVROS

- MAVROS Node
  - Bridges Jetson Nano (ROS2 Humble) to Pixhawk
  - Command needs to stay running to maintain communication
- ROS2 Topics
  - Communication channels in ROS where nodes can publish or subscribe to messages
- Echoing Topics
  - Monitors real-time messages on specified topics

```
# attach new terminal to running container
docker ps
docker exec -it <container_id> /bin/bash
[ docker exec -it priceless_ardinghelli /bin/bash ]
source /opt/ros/humble/install/setup.bash

# start MAVROS node to bridge ROS and Pixhawk flight controller
ros2 launch mavros px4.launch fcu_url:=/dev/ttyACM0:57600

# display list of active ROS2 topics currently being published/s
ros2 topic list
```

GitHub

- Personal access token
  1. Settings
  2. Developer settings
  3. Personal access tokens
  4. Tokens (classic)
  5. Generate new token (classic)