Useful commands

Start a container:

docker run -it <image\_name>

Start a container with a mounted drive catkin\_ws:

docker run -it -v ~/workspaces/catkin\_ws:/workspace/catkin\_ws <image\_name>

Start a container and give tty/USB0 and spidev0.0 access:

docker run -it -v ~/workspaces/catkin\_ws:/workspace/catkin\_ws -i --device=/dev/ttyUSB0 --device=/dev/spidev0.0 <image\_name>

include --network=host may be helpful

Full command

docker run -it -v ~/workspaces/catkin\_ws:/workspace/catkin\_ws -v /dev/bus/usb:/dev/bus/usb -i --device=/dev/ttyUSB0 --device=/dev/spidev0.0 --network=host --rm --privileged --name ros\_noetic noetic:latest

Enter terminal of existing running container:

docker exec -it <container> bash

Commit a docker container to new image:

docker commit <container> ros\_noetic:versionx

Save image:

docker save -o ros\_noetic\_image.tar ros\_noetic:version3

docker save image ml\_ros:latest | gzip > ml\_ros\_latest.tar.gz

Load image

docker load -i <path to image tar file>

Bring up:

1. Enable SPI with opt/nvidia/jetson-io
2. Sudo modprobe spidev
3. To get dev/ttyUSB0, call sudo apt remove brltty
4. Git clone <https://github.com/dusty-nv/jetson-containers.git>
5. Follow instructions to set up
6. Build the image with sudo docker run --runtime nvidia -it --rm –network=host dustynv/ros:noetic-desktop-l4t-r32.7.1
7. Alternative: ./build.sh --name=noetic:v2 ros:noetic-desktop
   1. Must comment opencv docker file and test file stuff to compile without cuda!

Suggested packages to install:

1. sudo apt-get update
2. sudo apt install net-tools
3. git clone rplidar: <https://github.com/satomm1/rplidar_ros.git>
4. Install bigger version of ros noetic:

sudo apt install ros-noetic-desktop

1. Git clone MattbotBringup: <https://github.com/satomm1/mattbot_bringup.git>
2. Pip install spidev
3. Git clone teleop: <https://github.com/satomm1/mattbot_teleop.git>
4. Git clone slam\_gmapping: [https://github.com/satomm1 /slam\_gmapping.git](https://github.com/satomm1%20/slam_gmapping.git)
5. Git clone twist\_mux: <https://github.com/satomm1/twist_mux.git>
6. rosdep install --from-paths ./src --ignore-src --rosdistro noetic -y

In ~/.bashrc file:

1. source ~/workspace/catkin\_ws/devel/setup.bash
2. export ROS\_IP=192.168.50.\*\*\*
3. export ROS\_MASTER\_URI=http://$ROS\_IP:11311
4. export MATTBOT\_3D\_SENSOR=astra

Setting up NVME:

Follow this tutorial: <https://www.digitalocean.com/community/tutorials/how-to-partition-and-format-storage-devices-in-linux>

* Note: might be smart to partition some swap in for the docker containers
* Make sure to add the following line to etc/docker/daemon.json:

"data-root": "/mnt/docker"

May need to call rospack profile to get packages to show up

Astra camera Setup:

Follow instructions at <https://github.com/satomm1/ros_astra_camera>

Except:

./scripts/create\_udev\_rules

sudo udevadm control --reload && sudo udevadm trigger

should be run outside of docker container

Install Docker Compose:

1. First add the repo to Apt so we can find the repository

https://docs.docker.com/engine/install/ubuntu/#install-using-the-repository

# Add Docker's official GPG key:

sudo apt-get update

sudo apt-get install ca-certificates curl

sudo install -m 0755 -d /etc/apt/keyrings

sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc

sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:

echo \

"deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu \

$(. /etc/os-release && echo "$VERSION\_CODENAME") stable" | \

sudo tee /etc/apt/sources.list.d/docker.list > /dev/null

sudo apt-get update

1. Install: sudo apt-get install docker-compose-plugin