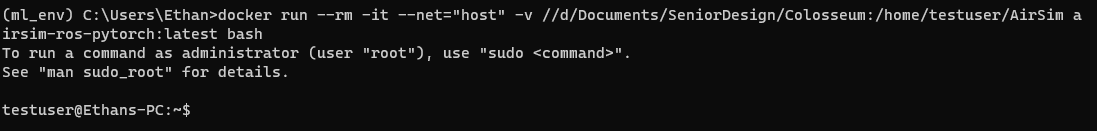
* Before starting the tutorial, ensure you have followed the Machine Learning ENV setup found in the same folder as this.
* You also must have cloned either AirSim or Colosseum (We will likely be using Colosseum) and be able to launch that environment accordingly.
* Clone our [GitHub repo](https://github.com/EthanAdkins/LM01-Drone-SAR). It is highly highly recommended to make sure to clone the repo in “Colosseum\PythonClient\multirotor” (Or AirSim)
* Install Docker: https://docs.docker.com/desktop/install/windows-install/
* Create an account
* Make sure when you install Docker Desktop you login to your account
* Drop updated docker file into D:\AirSim\tools (This is just wherever you have Airsim/Colosseum cloned. So in my case this location is D:\Documents\SeniorDesign\Colosseum\tools)
  + Docker File: Dockerfile-ROS\_Pytorch
* Open an Anaconda Powershell Prompt (from Machine Learning ENV setup)
  + Run “activate ml\_env”
  + Every command will be run from these prompts. Don’t use a normal command prompt. This is what I have done at least. From the videos of the other team they used Powershell though which confuses me a bit.
* Run: **docker build -t airsim-ros-pytorch -f Dockerfile-ROS\_Pytorch .** (space and period is required!!! (The dot refers to the current directory you are in!) This will take a long time, for me it took around 30-40 minutes). IF THIS ERRORS ON LINE 22: Open up “install\_ros\_deps.sh” with Notepad++ from the tools folder. Click ‘edit’ then ‘EOL Conversion’ and change the end of line to ‘Unix’ and save. Try running again.
  + NOTE: If using new Docker file use this: **docker build -t airsim-ros-pytorch-newest -f Dockerfile-ROS-Pytorch-Newest .**
* Once done installing, ensure you have Docker Desktop open then run: **docker run --rm -it --net="host" -v //d/Documents/SeniorDesign/Colosseum:/home/testuser/AirSim airsim-ros-pytorch:latest bash** (NOTE: the format of the section after -v is “<your-AirSim-folder-path>:/home/testuser/AirSim”. That means you must use your personal directory/repo location for AirSim/Colosseum)
  + It should look something like this now 
  + NOTE. If using new Docker file use this: **docker run --rm -it --net="host" -v //d/Documents/SeniorDesign/Colosseum:/home/testuser/AirSim airsim-ros-pytorch-newest:latest bash**

Below here is (mostly) done every time you want to access the files.

* Once in container then cd into /AirSim and run the following:
  + **echo "source /opt/ros/melodic/setup.bash" >> ~/.bashrc** (Do this every new terminal session)
  + **source ~/.bashrc** (Do this every new terminal session)
* Navigate to D:\Colosseum\ros\src\airsim\_ros\_pkgs (your AirSim/Colosseum repo)
  + Open another file window and navigate to “D:\Colosseum\PythonClient\multirotor\LM01-Drone-SAR\needsIntegration\ROSMsgAndSrvs”
  + Drage each of the files ending in “.msg” in “needsIntegration\ROSMsgAndSrvs” into the “msg” folder in “ros\src\airsim\_ros\_pkgs”. Do the same for files ending in “.srv” and move those into the “srv” folder in AirSim/Colosseum
  + Replace the “CMakeLists.txt” file in airsim\_ros\_pkgs with the one in “needsIntegration\ROSMsgAndSrvs”
* Then CD into /ros and run: **catkin build -DCMAKE\_C\_COMPILER=gcc-8 -DCMAKE\_CXX\_COMPILER=g++-8** (This is a one time install !!! It will take a little bit)
* Run: **source devel/setup.bash;** (Do this every new terminal session)
* Next run these installs (Most likely will not need it. Doesn’t hurt):
  + pip3 install numpy
  + pip3 install msgpack-rpc-python
  + pip3 install --upgrade pip
  + pip3 install opencv-python
  + sudo apt install iproute2
* Next run:
  + **ip addr | grep global**
  + Take note of the ip address for docker0 (ignore the number after the slash)
* Open up another windows terminal and type ipconfig and record the ip address of your computer (could be ethernet or something)
* Take the latest settings.json file from the GitHub repo or google Drive and use it to replace the default one that you have under /Documents/AirSim/ (For me this was my Documents folder in OneDrive so check there if you can’t find it)
* Open up settings.json with your favorite text editor and do the following:
  + Replace every ip under every attribute that says *ControlIp* and replace it with the ip found in your docker container (remember docker0)
  + Replace every ip under every attribute that says *LocalHostIp* and replace it with the ip found when using ipconfig in your windows terminal (with Docker open). It should be something like “Ethernet Adapter Ethernet (WSL (HyperVee Firewall))”. Make sure it has the WSL tag.
* In each terminal run: “**export ROS\_HOSTNAME=172.17.0.1**” replacing the ip on the right with the ip found in your docker container (the docker0 one)
* Important step: Each program you are running you must make sure to modify the LocalIP to be your own. For example, change the local IP of configDrones.py in “Drone-Search-and-Rescue-SD\DroneSwarm\Constants” line 7 to be your ip from ipconfig.
* Download startMultiDrone.py and droneNode.py from google Drive (Or the GitHub) and throw it into whatever folder you want in AirSim. You’ll use this file to test whether your ros setup works.
  + Modify droneNode.py line 24 to be your own local ip from ipconfig
* You’ll need two terminals for the next steps. One for running the ros scripts/python files and another for running the startMultiDrone.py file
* In the first Terminal, you’ll have your docker environment running and you’ll do: source devel/setup.bash; (You should have done this earlier. If you did, you don’t need to again).
* In the first terminal you’ll have your docker environment running and again you’ll run another command: **roslaunch airsim\_ros\_pkgs airsim\_node.launch;**
* Open up Unreal and your environment, start the project.
* Next open up a second terminal (anaconda prompt) and get into the docker container and run the startMultiDrone.py file (This file is in Google Drive and you can throw it wherever it is easiest to find in the AirSim Directory)