***Installation Guide To Run Gazebo+PX4 In A Docker Container for Windows***

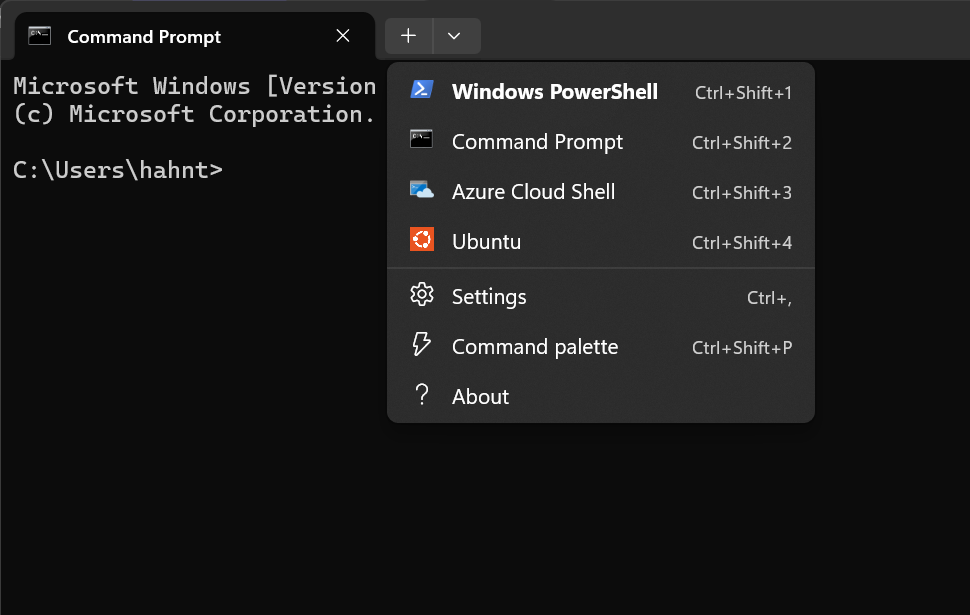
The setup this installation guide aims to achieve is to create a workspace where all team members can run simulations using the software they have helped develop from their divisions to test and integrate with other divisions' software's. This is done using Docker and setting up a Windows subsystem for Linux with a graphical interface and using Ubuntu terminal to run and edit the Docker file. Once inputting the commands in the terminal, the Docker file will run and show the Gazebo+PX4 interface. The workflow to achieve this is as follows: Windows → WSL2 (Ubuntu) → Docker → Gazebo+PX4.This approach allows easy reproducibility where all users have the same environment, no need to download heavy programs, an easy way to share the simulation environment, less worrying about ruining the environment when a main copy exists and a platform for testing different algorithms of the drone and other important software like telemetry and communications.

* *Install WSL2 (Windows Subsystem for Linux)*

Open Powershell as Administrator and run:

| wsl --install |
| --- |

This installs the WSL2 kernel and a default terminal based Linux distribution (choose Ubuntu). You will be prompted to restart your laptop. Once restarted, open the Ubuntu terminal and since it's the first launch of it, you will be asked to set a Linux username and password. To launch the Ubuntu terminal you must choose it from a dropdown menu in Command Prompt:



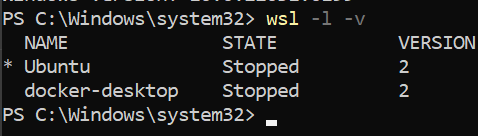
To verify that you are truly using WSL2 and not WSL1, run:

| wsl -l -v |
| --- |

The expected output should show this:

| NAME STATE VERSION \* Ubuntu Running 2 |
| --- |

The following photo is an example of a laptop already running the terminal based version of Ubuntu and the Gazebo+PX4 docker file using WSL2:



If when you run that command you get 1, fix it with:

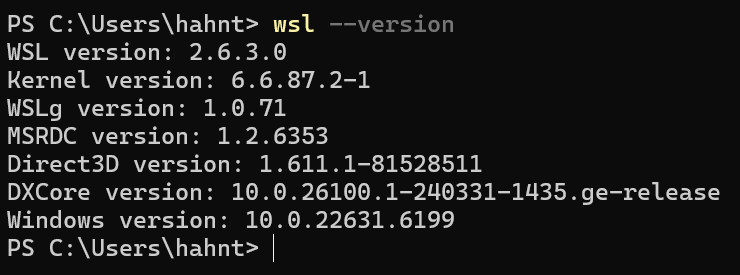
| *wsl -set-version Ubuntu 2* |
| --- |

If needed to, you can install Ubuntu manually if it does not install when running the *wsl –install*  command. The command for this is:

| *wsl -install -d -Ubuntu* |
| --- |

or you can install it via the Microsoft Store and download either Ubuntu 20.04 LTS or Ubuntu 22.04 LTS.

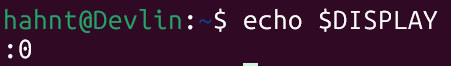
After this process of downloading WSL2 and the Ubuntu terminal, you can check the versions of different assets that are downloaded with WSL2:



Note the asset WSLg (Windows Subsystem for Linux GUI) is included in the download, this means that Linux GUI apps will run such as Gazebo+PX4 without any extra configuration. To verify that this truly does work, run in the Ubuntu terminal:

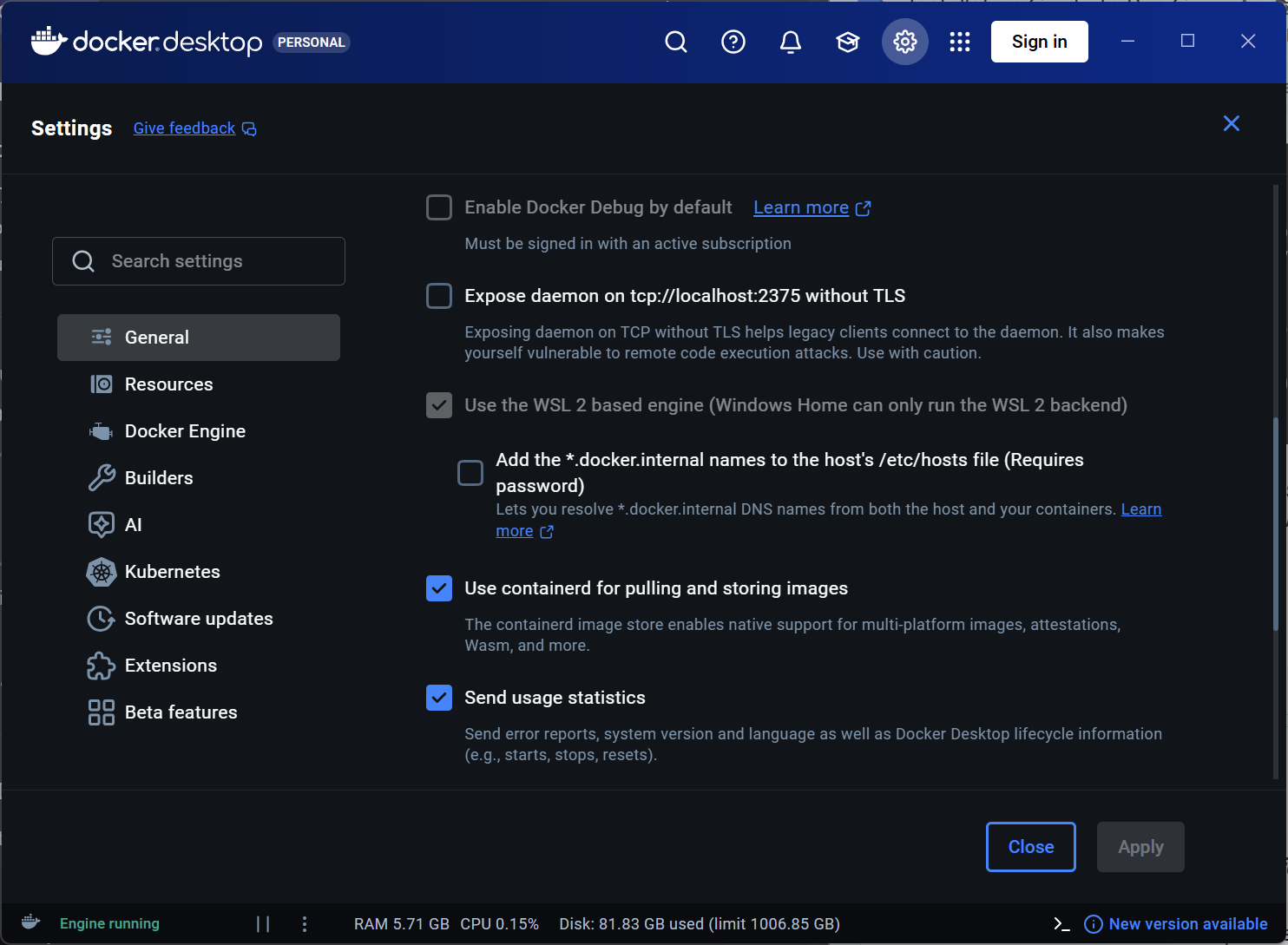
| *echo $DISPLAY* |
| --- |

*and you should get:*

**

* *Configuring Docker for WSL2*

Search for Docker Desktop in the Microsoft Store. Once downloaded, go to Settings → General and enable *Use the WSL 2 based engine* (it might be already selected since in the case of the first run of this whole workspace it was already pre-selected)



Still in Settings, go to Resources and scroll up and press WSL Integration. Here we are configuring which WSL2 distribution you want to access Docker from. Select the option of *Enable integration with my default WSL distro* and fetch distros that are downloaded into your system, Ubuntu should appear as the option so select it. Click apply and restart the Docker Desktop app.

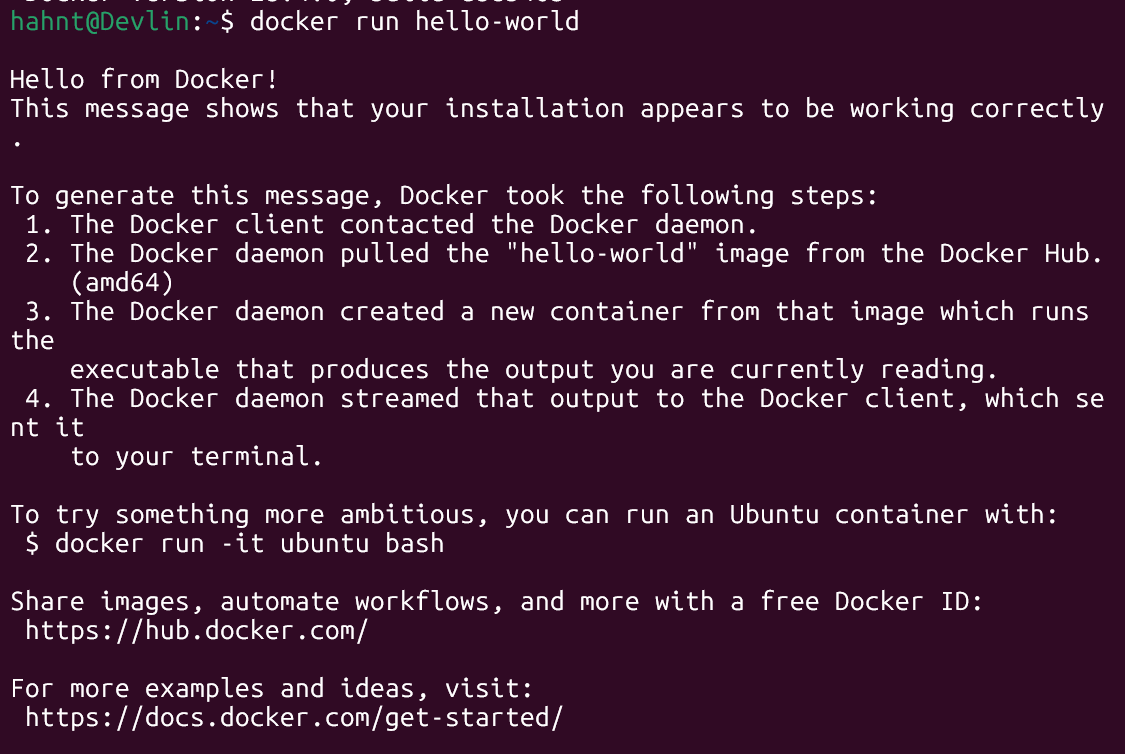
To verify Docker inside WSL, open the Ubuntu terminal and type the command:

| *docker -version* |
| --- |

, you should see something like this:



Then, run the command *docker run hello-world*, this should print a success message and demonstrates that Docker is working correctly.



* *Create the PX4+Gazebo Docker Project*

In this part of the installation guide, we will create the folder and file that will hold our Docker script. Firstly, open the Ubuntu terminal and create a new directory in your Linux home directory called *gazebo-docker* by running the command:

| mkdir gazebo-docker |
| --- |

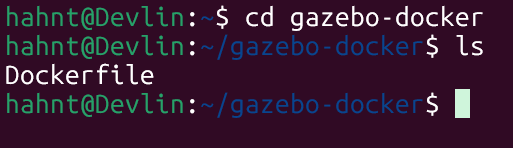
Now that we have our folder that will store our script, we will take the Docker script that you must have already downloaded onto your computer and move it into the folder we just created. One way to do this is by copying from wherever it is currently on your computer and then going to your Linux directory→Ubuntu folder→home→your user→*gazebo-docker* folder just created and pasting it there. The following video demonstrates this: [Screen Recording 2026-01-30 204422.mp4](https://drive.google.com/file/d/1b09vWHnhIJVfYTCwCqjBY4wojkICEwPe/view?usp=sharing). The other way is command based within the Ubuntu terminal, this works since WSL automatically mounts your Windows drives under */mnt*. To move the Docker script from for example your Downloads folder, first move into your Windows Downloads folder by running the command:

| *cd /mnt/c/Users/YOUR\_USERNAME/Downloads* |
| --- |

, now we move the Docker script (which is called *Dockerfile*) from Downloads to the *gazebo-docker* folder within Ubuntu by running:

| mv Dockerfile ~/gazebo-docker/. |
| --- |

You can double check if this was successful by moving into your *gazebo-docker* folder and checking if the file actually did move (*cd gazebo-docker*, then *ls* which means list)



* *Building and Running the Gazebo+PX4 Container*

Once everything before this step is completed, you are ready to build the container for the first time. Keep in mind, this process can vary in the time it takes to fully build and all depends on your laptop’s specs. To begin running the container, open Docker and keep it open and then, in the Ubuntu terminal, run the following command which basically begins building the container image:

| docker build -t px4-gazebo . |
| --- |

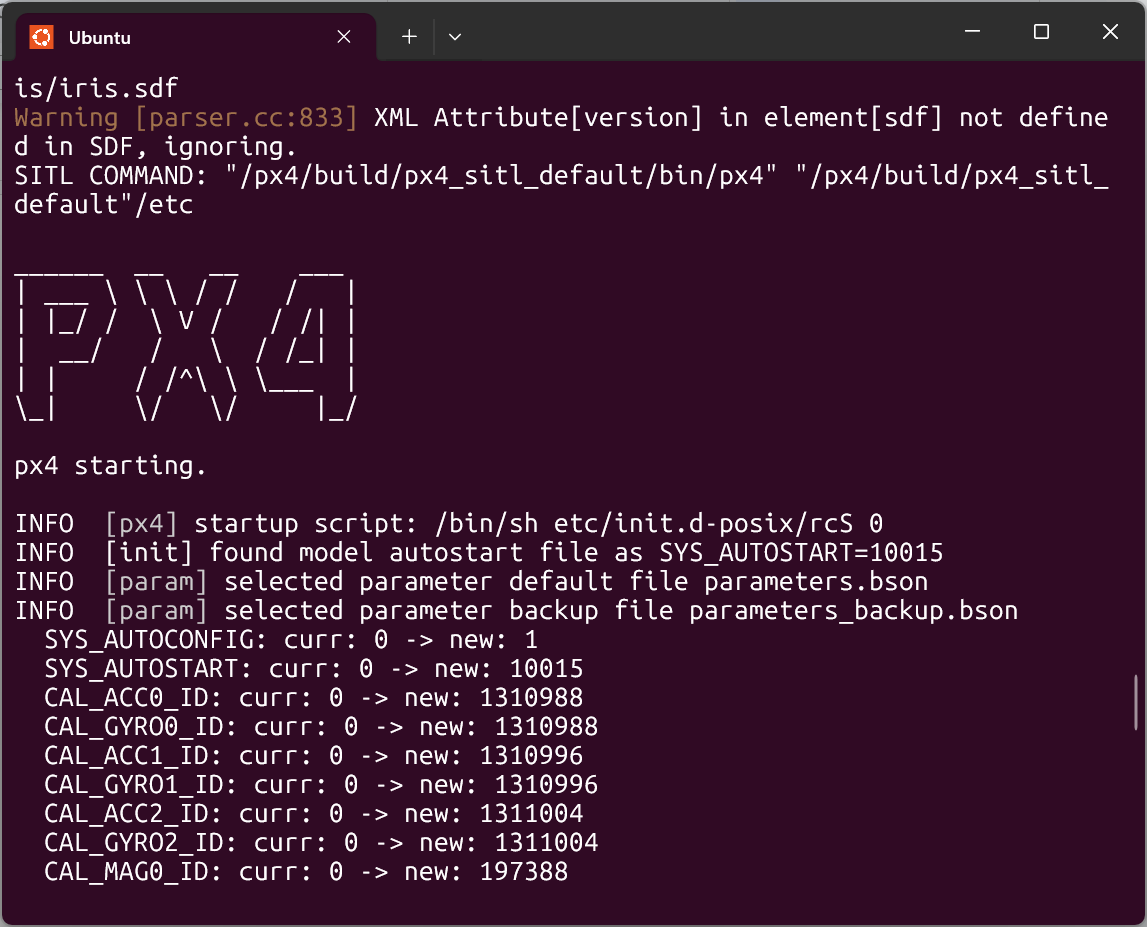
Then run this other command that actually begins running the container after its built and turns the terminal into an interactive terminal and Linux GUI capabilities with a shell inside the container:

| docker run -it --rm \-e DISPLAY=$DISPLAY \-v/tmp/.X11-unix:/tmp/.X11-unix \px4-gazebo bash |
| --- |

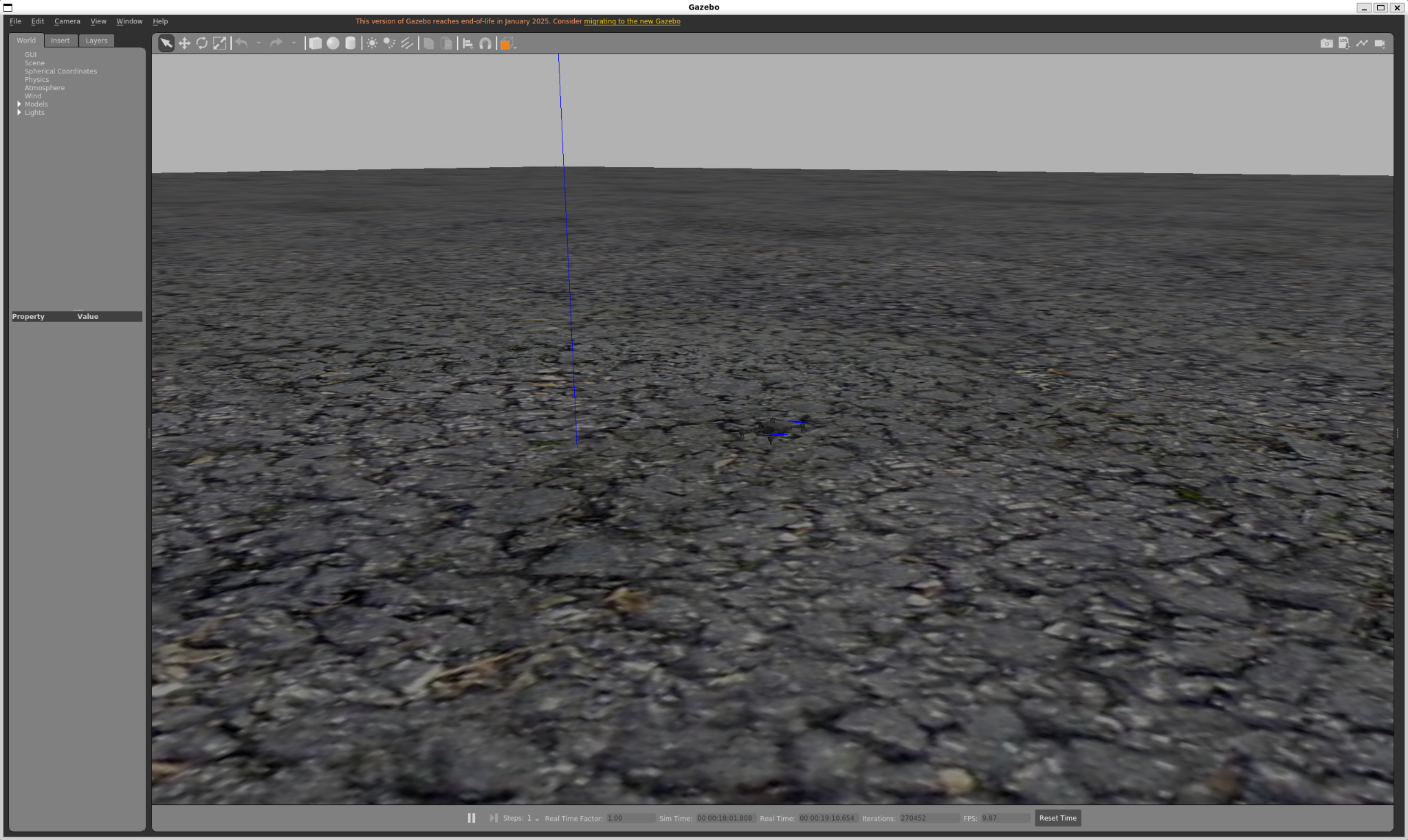
After this successfully runs, you will be inside the container, so now you can move into the PX4 directory to start PX4 with Gazebo by running this command:

| *cd /px4 make px4\_sitl\_default gazebo* |
| --- |

After running this, you will notice large amounts of messages in your Ubuntu terminal of things being set up and, since this is your first time building the container, it will take longer than if you have built it multiple times. Once all everything is done, you will begin to see the PX4 messages in the terminal:



Then the actual simulator should start and you will see a screen like this showing a small drone:



To give it a small test, you can run in the Ubuntu terminal the commands:

| commander takeoff |
| --- |

and the drone will takeoff at a preset minimum altitude. You can also run a command to land the drone back on the ground:

| commander land |
| --- |

***Installation Guide To Run Gazebo+PX4 In A Docker Container for Mac***

If on mac install docker desktop: [Docker Desktop for macOS page](https://www.docker.com/products/docker-desktop)

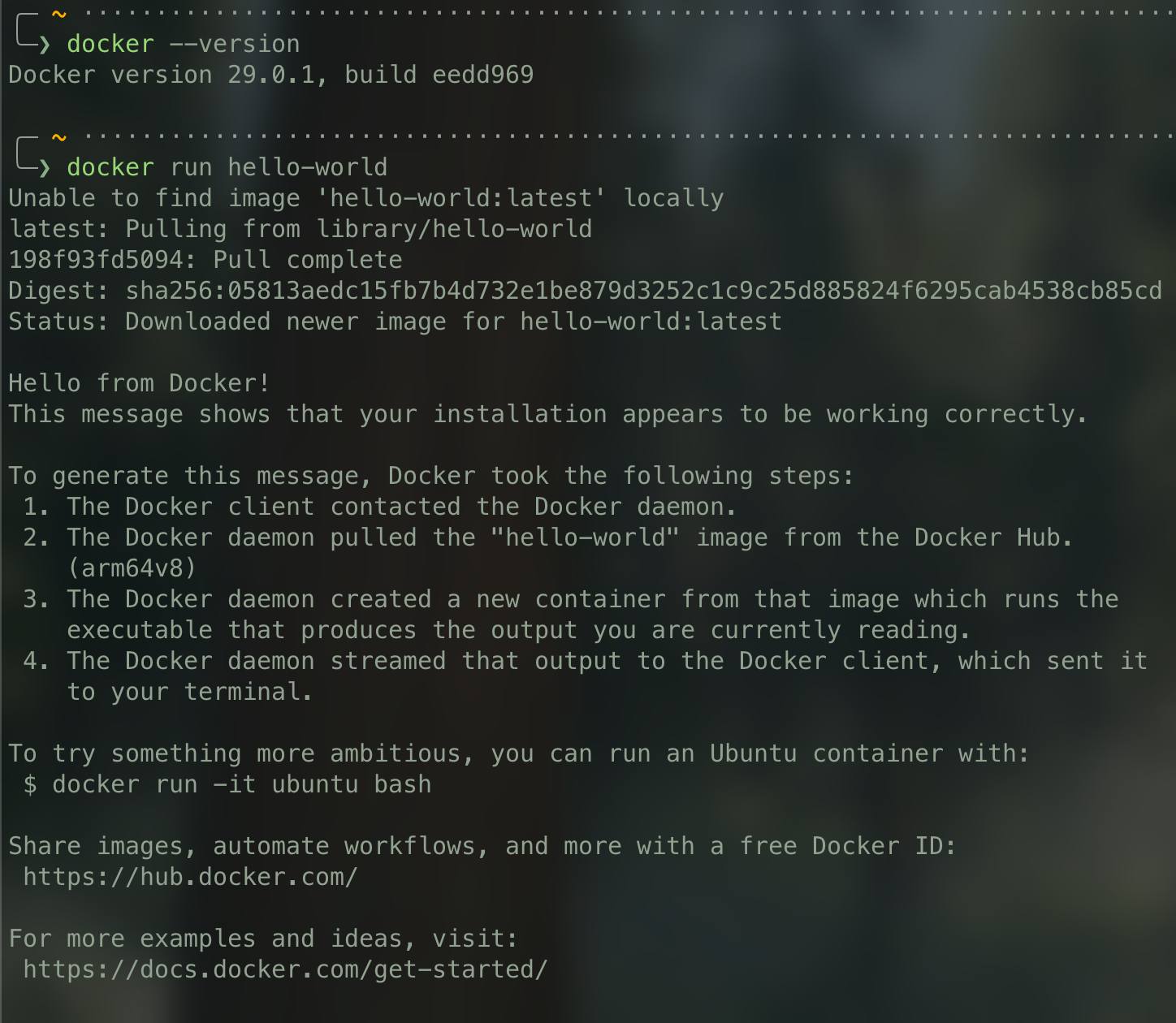
Open Docker Desktop app and allow the system permissions, then open your terminal and type:

| docker –—version |
| --- |

And:

| docker –—version |
| --- |

It should look like this:



* *Install and configure XQuartz(Gazebo GUI for mac)*

macOS does not natively support Linux graphical applications. XQuartz is required to enable GUI forwarding from the Docker container to the host machine.

1. Download XQuartz from https://www.xquartz.org
2. Install XQuartz normally.
3. Log out and log back into your macOS user account.
4. Launch XQuartz once after installation.
5. From the menu bar, navigate to:  
   XQuartz → Settings → Security
6. Enable the following option:
   * Allow connections from network clients

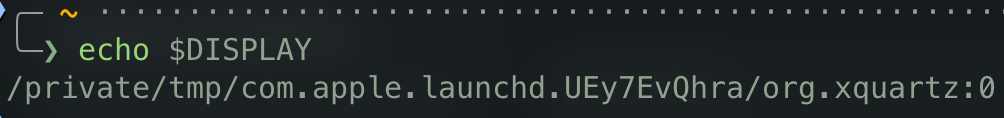


7. Restart XQuartz after making this change.

* *Verify X11*

Open a new Terminal window and run:

| echo $DISPLAY |
| --- |



If no output is shown, restart XQuartz and the Terminal.

Now we will create the folder and file that will hold our Docker script. Firstly, open the terminal and create a new directory in your home directory called *gazebo-docker* by running the command:

| mkdir gazebo-docker |
| --- |

Now that we have our folder that will store our script, we will take the Docker script that you must have already downloaded onto your computer and move it into the folder we just created. You can do this by moving it in finder or with the terminal

* *Building and Running the Gazebo+PX4 Container*

Once everything before this step is completed, you are ready to build the container for the first time. Keep in mind, this process can vary in the time it takes to fully build and all depends on your laptop’s specs. From the gazebo-docker directory run the following command:

| docker build -t px4-gazebo . |
| --- |

This process may take several minutes depending on system performance and internet speed. Once complete, the PX4 + Gazebo Docker image will be available locally as px4-gazebo.

Before starting the container, allow Docker access to the display:

| xhost +localhost |
| --- |

Now run the container with GUI support enabled:

| docker run -it --rm \  -e DISPLAY=host.docker.internal:0 \  -v /tmp/.X11-unix:/tmp/.X11-unix \  px4-gazebo bash |
| --- |

Inside the container terminal, navigate to the PX4 directory and start the simulator:

| cd /px4  make px4\_sitl\_default gazebo |
| --- |

***Installation Guide To Run Gazebo+PX4 In A Docker Container for Linux***

1. Install Required Dependencies: Docker requires some dependencies like apt-transport-https, ca-certificates, curl, software-properties-common. Install them using:

 sudo apt install apt-transport-https ca-certificates curl software-properties-common

1. Add Docker’s Official GPG Key: Run the following command to add Docker's official GPG key, which ensures the authenticity of the Docker packages.

 curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -

1. Add Docker’s Repository: Add the Docker repository to your list of sources so that you can install Docker from it:

 sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb\_release -cs) stable"

1. Install Docker Engine: Now, update your package database again and install Docker:

 sudo apt update

sudo apt install docker-ce

1. Start and Enable Docker: Enable Docker to start automatically when the system boots and start the Docker service:

 sudo systemctl enable docker

sudo systemctl start docker

1. Verify Installation: To check if Docker is installed and running correctly, use:

 docker --version

1. Test Docker: Run the following command to verify that Docker is functioning properly:

 sudo docker run hello-world

Docker will pull the hello-world image from the Docker Hub and run it in a container, displaying a message indicating the installation was successful.