

# How to get started with Docker images

## Installation

1. First you need to install Docker: <https://docs.docker.com/engine/install/ubuntu/>
2. Install Nvidia: <https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/install-guide.html#docker>
3. Add the user to the docker group to remove usage of sudo:

- Add the docker group if it doesn't already exist:

```
sudo groupadd docker
```

- Add the connected user "\$USER" to the docker group. Change the user name to match your preferred user if you do not want to use your current user:

```
sudo gpasswd -a $USER docker
```

- Either do a `newgrp docker` or log out/in to activate the changes to groups.
- You can use

```
docker run hello-world
```

to check if you can run docker without sudo.

4. After installation check your Docker version:

```
docker -v  
docker image list
```

## Create and build a ROS2 Docker container

4. Create a Docker folder: `mkdir docker cd ~/docker mkdir ros2_focal cd ros2_focal`
5. Create an example file: `nano Dockerfile`
6. Add the following lines into your example file and then save it:

```
FROM ros:rolling-ros-core-focal  
  
RUN echo ". /ros_entrypoint.sh" >> ~/.bashrc  
  
RUN apt-get update -y  
RUN apt-get install -y less nano  
RUN apt-get install -y ros-rolling-turtlesim ros-rolling-rqt-robot-steering ros-rolling-rqt-robot-monitor ros-rolling-rqt-common-plu
```

7. Build this docker image in the same folder: `docker build -t ros2_focal .`
8. List the installed Docker images: `docker image list`

## Create a script for sourcing all the dependencies

9. Create a run script to source all the necessary dependencies:

```
nano run.sh
```

10. Copy paste the following in the run.sh:

```
#!/bin/bash

#check if there is a /build and /devel in the current PWD
if [ -f devel/.catkin ]; then
echo "Starting from Catkin Workspace: $PWD, preparing it's ROS_PACKAGE_PATH."

sed -i 's/;/\projects\src/' devel/.catkin
echo -n "`cat devel/.catkin`; /projects/src" > devel/.catkin

fi

docker run -iPt \
--rm \
--env="DISPLAY=$DISPLAY" \
--env="QT_X11_NO_MITSHM=1" \
--volume="/tmp/.X11-unix:/tmp/.X11-unix:rw" \
--volume="$XAUTHORITY:$XAUTHORITY" \
--env="XAUTHORITY=$XAUTHORITY" \
--volume="$PWD:/projects" \
--runtime=nvidia \
--name="ros2_focal" \
ros2_focal \
bash
```

11. Make the script executable:

```
chmod +x run.sh
```

12. Run the script:

```
sudo ./run.sh
```

## Start the turtlesim node

13. This script will open a terminal inside the container. Inside the container start the turtlesim node:

```
ros2 run turtlesim turtlesim_node
```

## Start the teleop keyboard

14. To open a new terminal for keyboard control inside the Docker environment run:

```
docker exec -it ros2_test /bin/bash
```

15. Source the environment variables in this terminal as well:

```
. ros_entrypoint.sh
```

16. Run teleop:

```
ros2 run turtlesim turtle_teleop_key
```

## Create ROS melodic image

Create a docker folder if you don't have already:

```
mkdir docker
```

Inside the Docker folder create a new folder:

```
cd ~/docker
mkdir ros_melodic
```

Create a Dockerfile file:

```
nano Dockerfile
```

Add the following lines into your Docker file and then save it:

```
FROM ros:melodic-ros-core-bionic

RUN echo ". /ros_entrypoint.sh" >> ~/.bashrc

RUN apt-get update -y
RUN apt-get install -y less nano
RUN apt-get install -y ros-melodic-turtlesim ros-melodic-rqt-robot-steering ros-melodic-rqt-robot-monitor ros-melodic-rqt-common-plu
```

Build this docker image in the same folder, you can choose any tag as name:

```
docker build -t ros_melodic .
```

List the installed Docker images:

```
docker image list
```

## Create a script for sourcing all the dependencies

Create a run script to source all the necessary dependencies:

```
nano run.sh
```

Copy paste the following in the run.sh:

```
#!/bin/bash

#check if there is a /build and /devel in the current PWD
if [ -f devel/.catkin ]; then
echo "Starting from Catkin Workspace: $PWD, preparing it's ROS_PACKAGE_PATH."

sed -i 's;/\projects\src//' devel/.catkin
echo -n "`cat devel/.catkin`; /projects/src" > devel/.catkin
fi

docker run -iPt \
  --rm \
  --env="DISPLAY=$DISPLAY" \
  --env="QT_X11_NO_MITSHM=1" \
  --volume="/tmp/.X11-unix:/tmp/.X11-unix:rw" \
  --volume="$XAUTHORITY:$XAUTHORITY" \
  --env="XAUTHORITY=$XAUTHORITY" \
  --volume="$PWD:/projects" \
  --runtime=nvidia \
  --name="ros_melodic" \
  ros_melodic \
  bash
```

Make the script executable:

```
chmod +x run.sh
```

Run the script:

```
sudo ./run.sh
```

## Start the turtlesim node

---

This script will open a terminal inside the container. Inside the container start the rosmaster:

```
roscore
```

To open a new terminal for keyboard control inside the Docker environment in a new terminal run:

```
docker exec -it ros_melodic /bin/bash
```

Source the environment variables in this terminal as well:

```
. ros_entrypoint.sh
```

Start the the turtlesim node:

```
roslaunch turtlesim turtlesim_node
```

## Start the teleop keyboard

---

To open a new terminal for keyboard control inside the Docker environment run in a new terminal again:

```
docker exec -it ros_melodic /bin/bash
```

Source the environment variables in this terminal as well:

```
. ros_entrypoint.sh
```

Run teleop:

```
roslaunch turtlesim turtle_teleop_key
```