

How to use Docker

September 30, 2018

The purpose of this document is to give you some useful tips about how to use *Docker*. Hope this is helpful.

Part I

Create Docker folder and Dockerfile

I provided the *Dockerfile* in attachment. To use it, please first create a *Docker* folder under your root directory and download this *Dockerfile* to this directory.

Part II

Build and Run

- When you have the *Dockerfile* in your *Docker* folder, open a new shell and type the command line, this step assumes that you already add your user name in a *Docker* group so that you don't need *sudo* to run docker command,

```
$ cd Docker/  
$ docker build -t indigo:dev .
```

- Once it finishes downloading we can continue to next step, in the same shell type,

```
$ xhost +local:
```

- This line is to allow container be able to create graphical windows on the host, so you can use any GUI applications on rviz. Next, we can run docker. Before that please note that you could change the parameter *-name* (the line near bottom) to any name you want, for example in the sample code I change it to my name Kevin and create a container named *Kevin*,

```
$ docker run -it --net=host \
--user=$(id -u) \
-e DISPLAY=$DISPLAY \
-e QT_GRAPHICSSYSTEM=native \
-e CONTAINER_NAME=ros-indigo-dev \
-e USER=$USER \ --workdir=/home/$USER \
-v "/tmp/.X11-unix:/tmp/.X11-unix" \
-v "/etc/group:/etc/group:ro" \
-v "/etc/passwd:/etc/passwd:ro" \
-v "/etc/shadow:/etc/shadow:ro" \
-v "/etc/sudoers.d:/etc/sudoers.d:ro" \
-v "/home/$USER/:/home/$USER/" \
--device=/dev/dri:/dev/dri \
--name=Kevin \
indigo:dev
```

- When you successfully start docker, the color of the characters will change from green to grey, now let's try this to see if everything goes well,

```
$ rosversion -d
```

- The expected output should be *indigo*. In this shell we can use these lines to start ROS master,

```
$ source /opt/ros/indigo/setup.bash
$ roscore
```

- Command to open a new shell in docker is a little bit different, the format is *docker exec -it [YOUR_CONTAINER_NAME] bash* . You should change this to your own container name, in sample code, I change it to my container Kevin. First use Ctrl+Shift+T to open a new tab and type,

```
$ docker exec -it Kevin bash
$ source /opt/ros/indigo/setup.bash
```

- Now you can use it just like a normal terminal window! Let us try the fetch simulation, since I already include this package in *Dockerfile* you can use it directly without downloading anything,

```
$ roslaunch fetch_gazebo simulation.launch
```

- The Fetch robot will appear in Gazebo simulation now :)

Part III

Check docker status

There are some useful commands to check the status of your container.

- check the current status of your containers,

```
$ docker ps
```

This will show all running containers and their names, ports etc. When your docker is active, you should see its name using this command.

- stop your container, you can either use that hash string or just your container name, for example I use my container's name here,

```
$ docker stop <YOUR_CONTAINER_NAME>
$ docker stop Kevin (This is just an example!)
```

- to fully stop your container you need the *kill* command,

```
$ docker kill <YOUR_CONTAINER_NAME>
$ docker kill Kevin (This is just an example!)
```

- to restart existing container,

```
$ docker start <YOUR_CONTAINER_NAME>
$ docker start Kevin (This is just an example!)
```

Part IV

Common issues

Here list some common issues I faced when testing the docker:

- When you test your fetch simulation.launch file with gazebo and see the error messages:

```
No protocol specified Error [RenderEngine.cc:665] Can't open display: :0 Warning [RenderEngine.cc:665] Can't open display: :0
```

Solution The reason is that you didn't execute this line before you do simulation with rviz, gazebo or other GUI applications, add this line in shell and hit enter:

```
$ xhost +local:
```

You should see results similar to this:

```
$ non-network local connections being added to access control list
```

- When you use any ros command and see this, this is because you didn't use *source* command first,

```
$ bash: <SOME_THING>: command not found
```

Solution Type and execute this line

```
$ source /opt/ros/indigo/setup.bash
```

- When you try to use *docker run -it...* and see the following error messages:

```
docker: Error response from daemon: Conflict. The container name "[SOME_NAME]" is already in use.
```

- This is because the name you specify for your container has already been used, you should think of another unique name like your GTID! Please don't remove other's containers without asking them first!

Solution Change the name of your container, use your unique GTID instead in command.

```
$ docker run -it --net=host \
--user=$(id -u) \
-e DISPLAY=$DISPLAY \
-e QT_GRAPHICSSYSTEM=native \
-e CONTAINER_NAME=ros-indigo-dev \
-e USER=$USER \ --workdir=/home/$USER \
-v "/tmp/.X11-unix:/tmp/.X11-unix" \
-v "/etc/group:/etc/group:ro" \
-v "/etc/passwd:/etc/passwd:ro" \
-v "/etc/shadow:/etc/shadow:ro" \
-v "/etc/sudoers.d:/etc/sudoers.d:ro" \
-v "/home/$USER:/home/$USER/" \
--device=/dev/dri:/dev/dri \
--name=[YOUR_GTID] \
indigo:dev
```

Change here:)

Hope you enjoy using Docker :3