

Docker y Gazebo ur3

Antes de empezar

Instalar docker: <https://www.digitalocean.com/community/tutorials/como-instalar-y-usar-docker-en-ubuntu-18-04-1-es>

Docker a utilizar

Utilizar el docker que deseamos: <https://github.com/jacknlliu/ros-docker-images/blob/master/README.md>

Docker al final es como una maquina virtual, pero que consumo muy pocos recursos al no utilizar interfaz almenos que le especifiques que lanzar.

Ademas le puedes generarlo con todas las instalaciones previas. Con ello puedes hacer funcionar ese docker sin depender ni del equipo o sistema operativo que tengas.

Para lanzar, si no tienes ningún proyecto en ROS creado mejor utilizar esto:

```
docker run --privileged \
  --security-opt label=disable \
  --security-opt seccomp=unconfined \
  --env="DISPLAY" --env QT_X11_NO_MITSHM=1 \
  --volume="/tmp/.X11-unix:/tmp/.X11-unix:rw" \
  --name="ros_kinetic" \
  jacknlliu/ros:kinetic-ide-init terminator
```

Este comando lo que hace es, a partir del **docker jacknlliu/ros:kinetic-ide-init** te crea una imagen con el nombre **ros_kinetic**. Todo lo demás son parámetros para hacer funcionar el gazebo dentro de ese docker.

Después de lanzar-lo tu puedes ver que el docker esta funcionando con: `docker ps -a`

Siguiendo como ejemplo: http://docs.ros.org/kinetic/api/moveit_tutorials/html/doc/move_group_python_interface/move_group_python_interface_tutorial.html

Lista de comandos dentro del terminal del docker

Una vez lanzado, instalar todo lo que necesitamos para manejar el ur3 con fichero python3 y comunicación vía ros. Esta es la lista de comandos que he hecho para crear un proyecto e importar todas las dependencias necesarias:

```
17 catkin_make
18 /usr/bin/python3 --v
19 /usr/bin/python3 -version
20 /usr/bin/python3 --version
21 catkin_make -DPYTHON_EXECUTABLE=/usr/bin/python3
22 source devel/setup.bash
23 echo $ROS_PACKAGE_PATH
24 cd ~/catkin_ws/src
```

```

25 catkin_create_pkg ur-test std_msgs rospy moveit_commander
26 cd ~/catkin_ws
27 catkin_make
28 . ~/catkin_ws/devel/setup.bash
29 cd src/
30 git clone -b kinetic-devel https://github.com/ros-
planning/moveit_tutorials.git
31 git clone -b kinetic-devel https://github.com/ros-
planning/panda_moveit_config.git
32 rosdep install -y --from-paths . --ignore-src --rosdistro kinetic
33 cd ..
34 . ~/catkin_ws/devel/setup.bash
35 catkin_make

```

Con esto tu paquete es `ur-test` donde pondrás tu código en `ur-test/src`. FALTA crear el fichero de lanzamiento como este: https://github.com/ros-planning/moveit_tutorials/tree/kinetic-devel/doc/move_group_python_interface/launch

Pero se debe de adaptar para el nombre de tu proyecto `ur-test`.

Pasos que se siguen normalmente:

1. Esto solo se haría una vez:

```

docker run --privileged \
--security-opt label=disable \
--security-opt seccomp=unconfined \
--env="DISPLAY" --env QT_X11_NO_MITSHM=1 \
--volume="/tmp/.X11-unix:/tmp/.X11-unix:rw" \
--name="ros_kinetic" \
jacknlliu/ros:kinetic-ide-init terminator

```

2. Luego para volver a iniciar, sería en tu terminal Ubuntu:

```
docker ps -a
```

Saldrá algo como esto:

CONTAINER ID	IMAGE	CREATED
e454ccdaec5f	jacknlliu/ros:kinetic-ide-init	37 hours ago
STATUS		
Exited (1)		
NAMES		
ros_kinetic		

Después escribir en el terminal estos dos comandos:

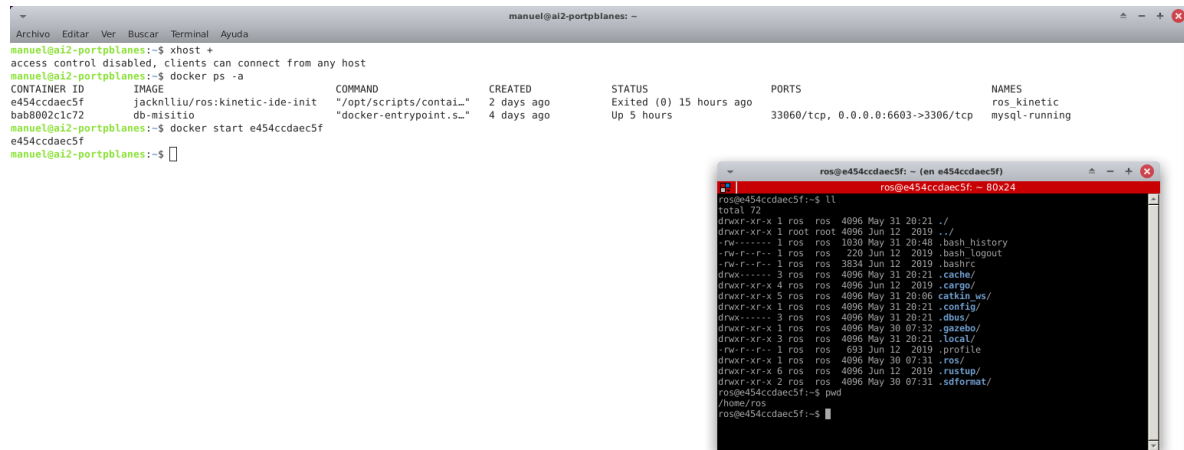
```
xhost +
```

```
docker start e454ccdaec5f
```

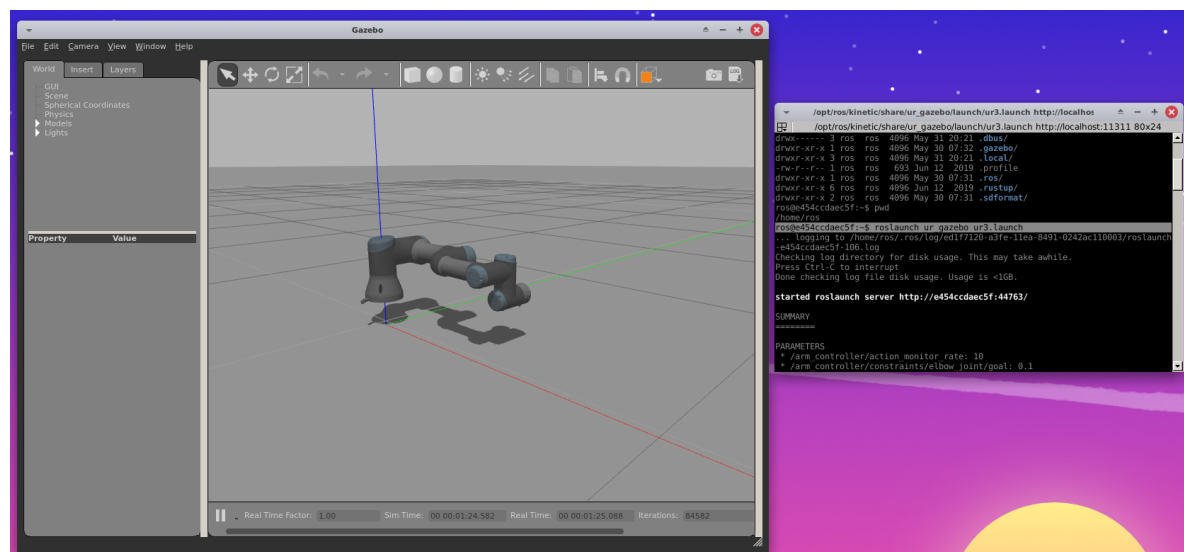
3. Con eso se te abre un terminal. La idea seria, lanzar el emulador con `roslaunch ur_gazebo ur3.launch` en el terminal que sale cuando lo arrancar.
4. En otro terminal de tu equipo: `docker exec -it ros_kinetic bash`
5. Estas dentro del docker donde puedes editar tu codigo o hacer las pruebas que necesites
6. Puedes editar un python en tu ubuntu normal y luego copiarlo al docker que esta lanzado

Capturas

Arrancar docker ya creado y hecho docker run

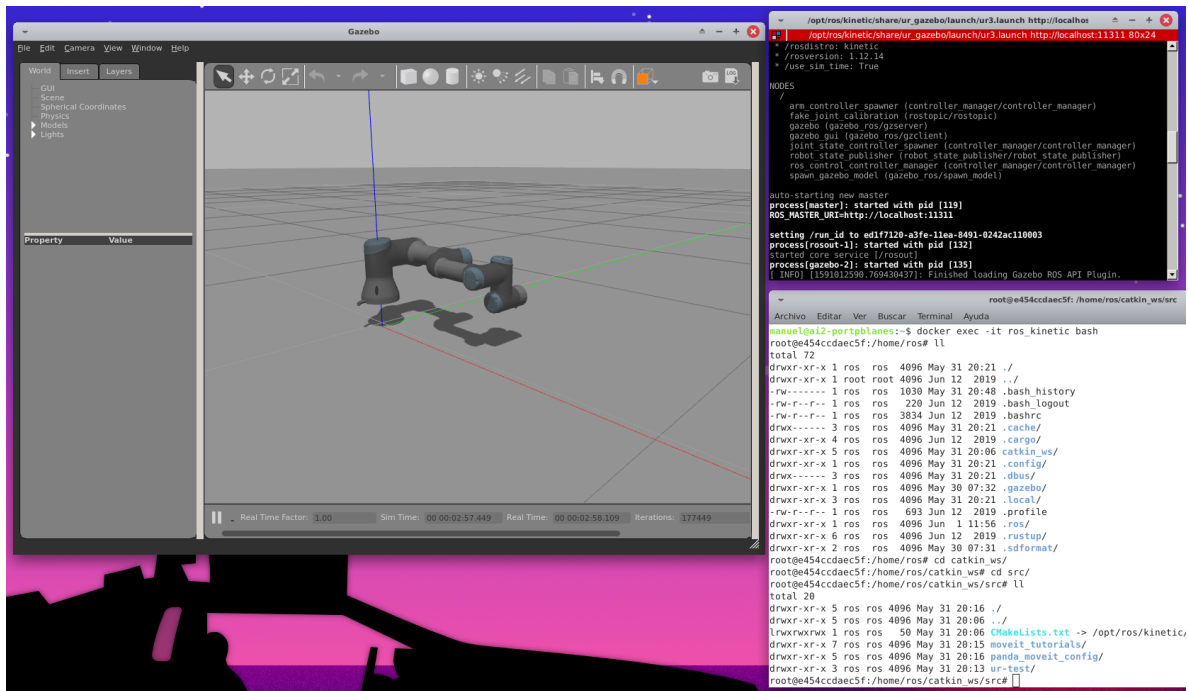


Lanzar Gazebo en el docker



Lanzar otro terminal para estar dentro del docker (Situación ideal)

En esta imagen puedes ver el gazebo y en el otro terminal (fondo blanco) como accedo dentro del docker y veo mis paquete de ros creado o importados



Eliminarlo todo

Si quieres eliminar el docker e imagen y empezar de nuevo, en el terminal:

```

docker rmi -f $(docker images -a -q)

docker stop $(docker ps -a -q)
docker rm $(docker ps -a -q)

```

Enlaces interesantes

http://docs.ros.org/kinetic/api/moveit_tutorials/html/doc/getting_started/getting_started.html

https://github.com/ros-industrial/universal_robot/blob/indigo-devel/README.md

ROS AL INICIO

1. http://wiki.ros.org/catkin/Tutorials/create_a_workspace
2. <http://wiki.ros.org/ROS/Tutorials/CreatingPackage>

http://www2.ece.ohio-state.edu/~zhang/RoboticsClass/docs/ECE5463_ROSTutorialLecture1.pdf