Instrucciones de instalación de la construcción del contenedor para la asignatura de Robótica Inteligente.

Las siguientes instrucciones se deben realizar en el sistema operativo Ubuntu 20.04 o superior que contenga una versión de ROS. Además, debe poder construirse un contenedor mediante *Docker* y si se tiene una tarjeta gráfica dedicada tener instalado nvidia-docker2.

Se deben descargar los ficheros de esta dirección:

La carpeta *Robotica\_inteligente\_practica* contiene los ficheros:

* Lista de instrucciones (*README*)
* Carpeta *src* que contienen los scripts para las prácticas de la asignatura
* Fichero *update\_bashrc*, el cual tiene la configuración del *bash* del contenedor.
* Dockerfile, con las instrucciones para crear el contenedor

**Instrucciones para la creación del contenedor.**

* 1. Acceder a la carpeta:

cd Robotica\_inteligente\_practica

* 1. Dentro de la carpeta construir el contenedor (esta instrucción tardará varios minutos):

sudo docker build -t robotica-inteligente .

* 1. Al tener finaliza la construcción de la Docker, es necesario ejecutar el siguiente comando para habilitar la interfaz gráfica dentro del contenedor.

xhost +local:

* 1. Correr contenedor con el comando:

sudo docker run --shm-size=1g --privileged --ulimit memlock=-1 --ulimit stack=67108864 --rm -it --net=host -e DISPLAY=:1 --user=root -v /tmp/.X11-unix:/tmp/.X11-unix:rw -v /dev:/dev --name container --gpus all --cpuset-cpus=0-3 robotica-inteligente

* 1. Dentro del contenedor correr los comandos:

source devel/setup.bash

roslaunch robotica\_inteligente load\_scene.launch

Después de los pasos realizados deben aparecer en pantalla 3 ventanas para comprobar que se ha construido correctamente el contenedor.

Interfaz de usuario gráfica

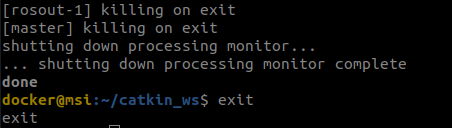
Descripción generada automáticamente Imagen que contiene objeto, aparato

Descripción generada automáticamente Una captura de pantalla de una computadora

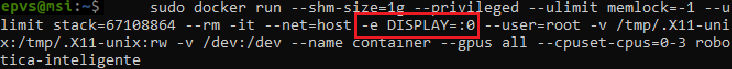
Descripción generada automáticamente

*Figura 1 Ventanas tras ejecutar el comando roslaunch robotica\_inteligente load\_scene.launch*

**IMPORTANTE:** Tras comprobar que las ventanas de la figura 1 se muestran, se puede cerrar el contenedor. Para ello se debe cerrar cada ventana y presionar Ctrl+c para cerrar el proceso dentro del contenedor. Finalmente se debe escribir el comando exit para salir del contenedor.



**Nota**: Si por alguna razón el contenedor corre después del paso 3 y la pantalla no muestra las ventanas del paso 5 se debe cambiar la numeración del Display a 0.



sudo docker run --shm-size=1g --privileged --ulimit memlock=-1 --ulimit stack=67108864 --rm -it --net=host -e DISPLAY=:0 --user=root -v /tmp/.X11-unix:/tmp/.X11-unix:rw -v /dev:/dev --name container --gpus all --cpuset-cpus=0-3 robotica-inteligente

**Nota 2:** Si no se tiene una tarjeta grafica dedicada o no se tiene instalado nvidia-docker2 se debe ejecutar el contenedor sin la bandera de --gpus all:

sudo docker run --shm-size=1g --privileged --ulimit memlock=-1 --ulimit stack=67108864 --rm -it --net=host -e DISPLAY=:1 --user=root -v /tmp/.X11-unix:/tmp/.X11-unix:rw -v /dev:/dev --name container --cpuset-cpus=0-3 robotica-inteligente

**Instalación de los controladores de NVIDIA dentro del contenedor**

El trabajo práctico puede ser ejecutado tanto en un ordenador con o sin tarjeta gráfica dedicada. El uso de una tarjeta gráfica dedicada puede acelerar parte del procesamiento de la interfaz gráfica dentro del contenedor. Es por eso por lo que, si se tiene una tarjeta gráfica dedicada es recomendable que se instalen los drivers necesarios para que los controladores de NVIDIA se puedan utilizar dentro del contenedor. Hay que recordar también que, muchos procesos de redes neuronales utilizan una **gpu** y programas con **CUDA** para poder acelerar el entrenamiento o predicciones. A continuación, se muestra como instalar los controladores para NVIDIA dentro del contenedor.

Los siguientes pasos son extraídos del link:

<https://docs.nvidia.com/datacenter/cloud-native/container-toolkit/latest/install-guide.html>

1. **Configurar la producción del repositorio:**

curl -fsSL https://nvidia.github.io/libnvidia-container/gpgkey | sudo gpg --dearmor -o /usr/share/keyrings/nvidia-container-toolkit-keyring.gpg **\**

&& curl -s -L https://nvidia.github.io/libnvidia-container/stable/deb/nvidia-container-toolkit.list | **\**

sed 's#deb https://#deb [signed-by=/usr/share/keyrings/nvidia-container-toolkit-keyring.gpg] https://#g' | **\**

sudo tee /etc/apt/sources.list.d/nvidia-container-toolkit.list

1. **Actualizar los paquetes de la lista del repositorio:**

sudo apt-get update

1. **Instalar los paquetes NVIDIA Container Toolkit:**

sudo apt-get install -y nvidia-container-toolkit

1. **Instalar los paquetes de Nvidia-docker2:**

sudo apt-get install -y nvidia-docker2

1. **Reiniciar los servicios de Docker:**

sudo systemctl restart Docker

1. **Verificar si se tiene los controladores dentro de la Docker:**

Se puede testear si se tienen los controladores ejecutando una imagen de CUDA

sudo docker run --rm --gpus all nvidia/cuda:11.0-base nvidia-smi

Se debería visualizar algo como:

