



Universitat d'Alacant  
Universidad de Alicante

# Docker para Robótica

Practica 3 y 4 de Teleoperación

Edison Velasco Sánchez  
edison.velasco@ua.es

Profesorado: Dr. Santiago T. Puentes  
Asignatura: Teleoperación



# Objetivos:

- Introducción de uso de la Docker para la práctica 3 y 4 de Teleoperación
- Facilitar al alumnado un entorno para el desarrollo de las practicas 3 y 4 de Teleoperación.
- Que el alumnado tenga instalado Docker en sus ordenadores.





# ¿Qué es Docker?

Docker: Plataforma de contenedores para aplicaciones.



+ ROS

## Aislamiento

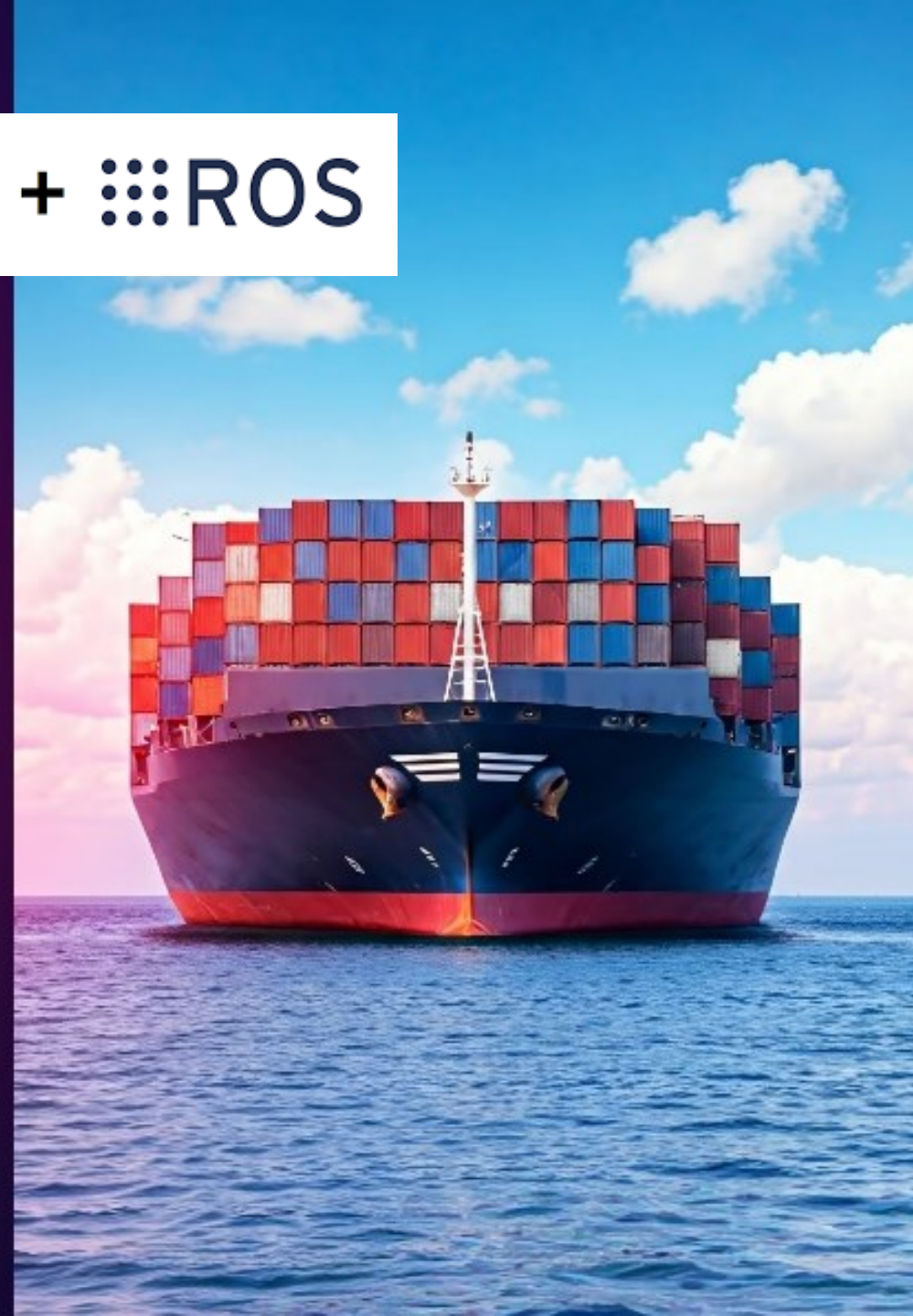
Ejecuta aplicaciones en entornos aislados.

## Portabilidad

Ejecuta en diferentes sistemas operativos.

## Reproducibilidad

Garantiza un entorno consistente.



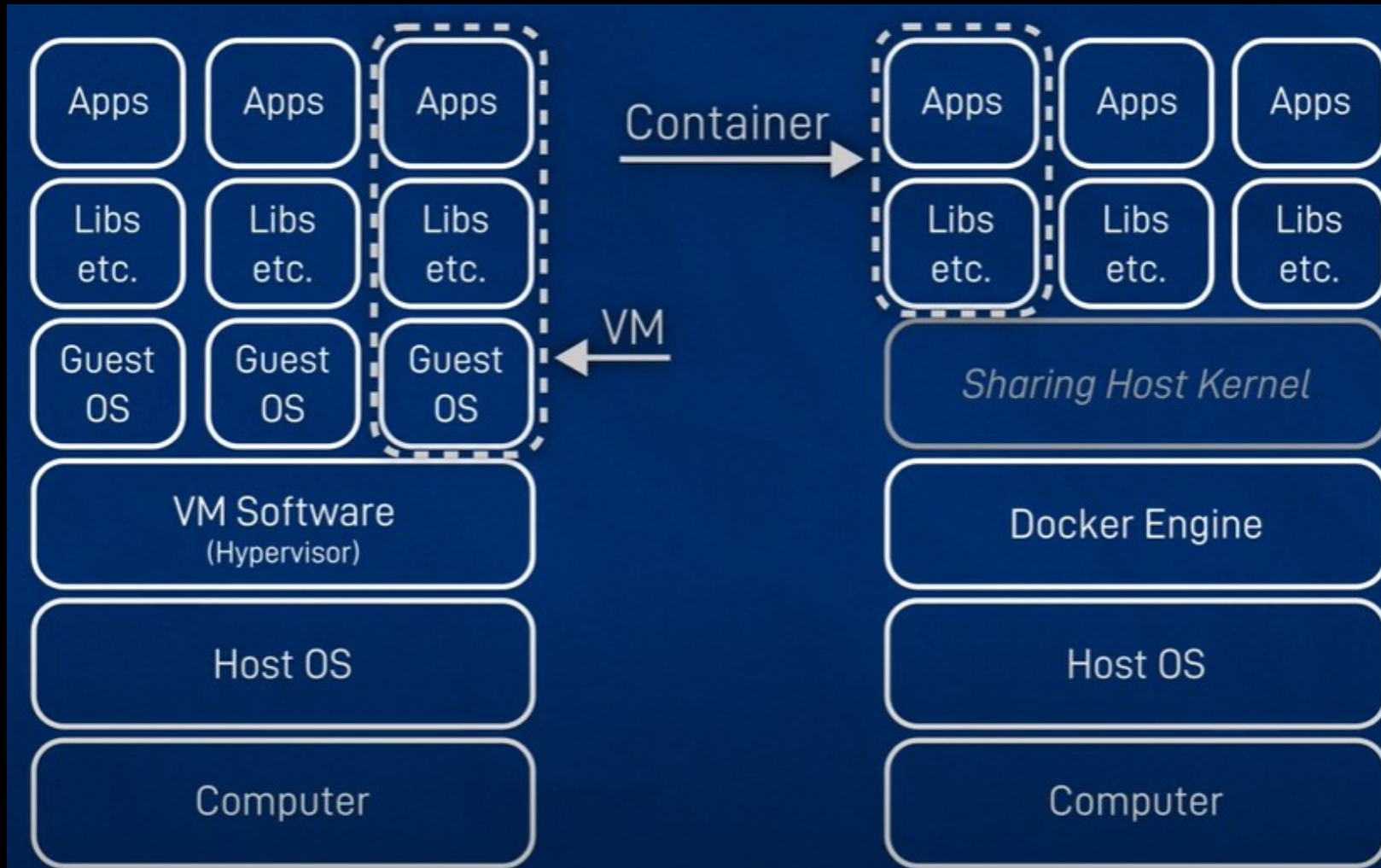
Original  
Disk



Disk  
Image



# Máquina Virtual vs Docker





# Beneficios de Docker para ROS

Simplifica el desarrollo y la implementación de robots.

## Entorno Consistente

Desarrollos reproducibles.

## Gestión de Dependencias

Administración de paquetes de software.

## Escalabilidad

Aumento de recursos según sea necesario.



# Creando un contenedor Docker

Utilizar Dockerfile para configurar un entorno ROS.

1

Instalar Docker

Descarga e instalación de Docker.

2

Escribir Dockerfile

Especificar las instrucciones de construcción.

3

Construir la Imagen

Crear la imagen del contenedor.

4

Ejecutar el Contenedor

Lanzar el contenedor.

## Install using the `apt` repository

Before you install Docker Engine for the first time on a new host machine, you need to set up the Docker repository. Afterward, you can install and update Docker from the repository.

1. Set up Docker's `apt` repository.

```
# Add Docker's official GPG key:
sudo apt-get update
sudo apt-get install ca-certificates curl
sudo install -m 0755 -d /etc/apt/keyrings
sudo curl -fsSL https://download.docker.com/linux/ubuntu/gpg -o /etc/apt/keyrings/docker.asc
sudo chmod a+r /etc/apt/keyrings/docker.asc

# Add the repository to Apt sources:
echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/etc/apt/keyrings/docker.asc] https://download.docker.com/linux/ubuntu $(. /etc/os-release && echo "$VERSION_CODENAME") stable" | \
  sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
sudo apt-get update
```

2. Install the Docker packages.

Latest   Specific version

To install the latest version, run:

```
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-buildx-p
```

3. Verify that the Docker Engine installation is successful by running the `hello-world` image.

```
$ sudo docker run hello-world
```

Estos pasos lo harán cada alumno en su ordenador para tener Docker instalado. (Seguir los 3 pasos de la página).

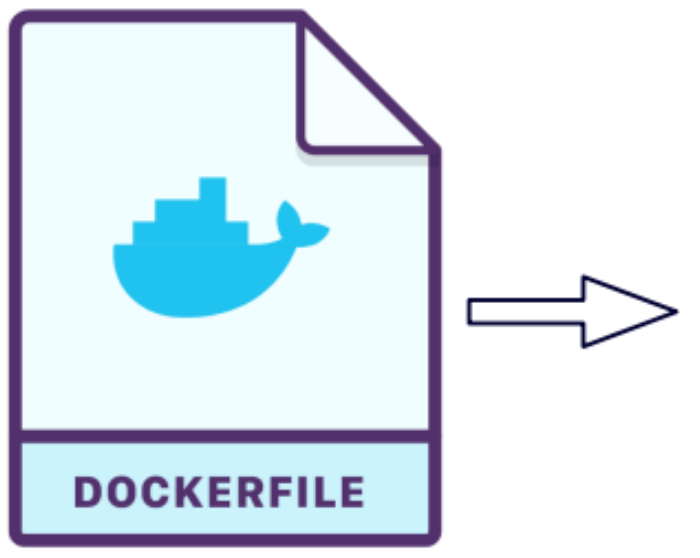
**Nota:** Los alumnos únicamente tienen Docker, aún no se les ha entregado la imagen que tiene el contenedor para la práctica. Esto se verá el día de introducción a la práctica.



2

## Escribir Dockerfile

Especificar las instrucciones de construcción.

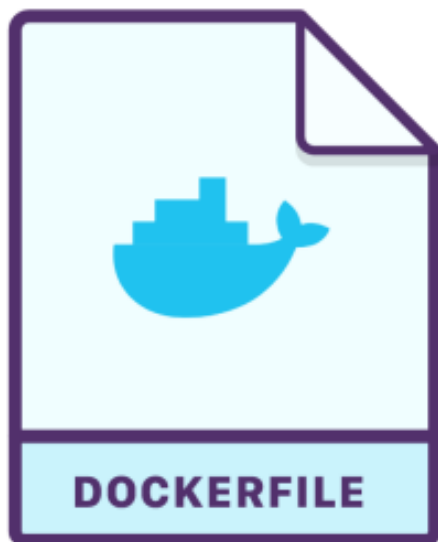


**Docker file**

3

## Construir la Imagen

Crear la imagen del contenedor.



**Docker file**

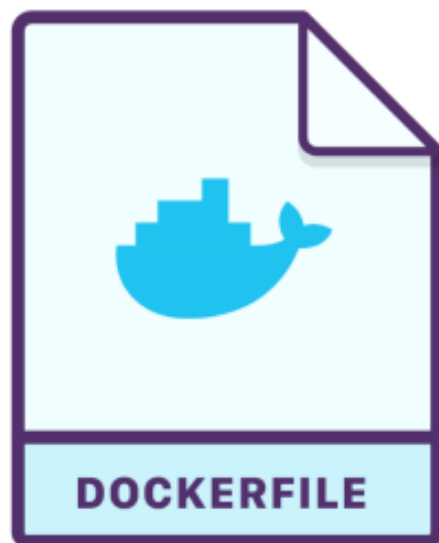


**Docker Image**

4

## Ejecutar el Contenedor

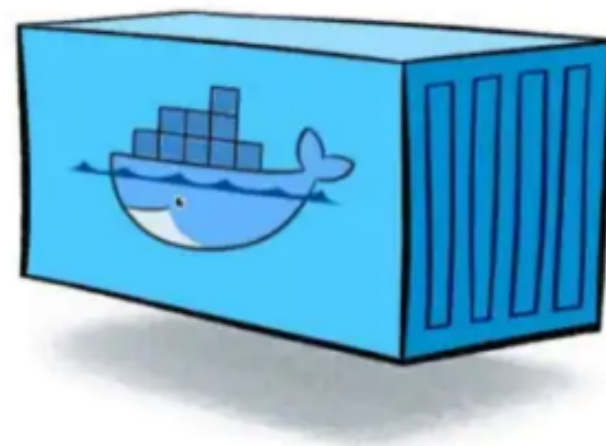
Lanzar el contenedor.



**Docker file**

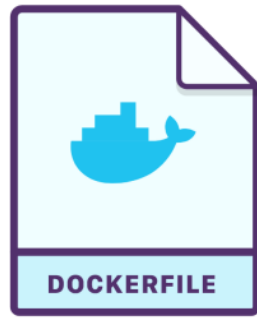


**Docker Image**



**Docker Container**

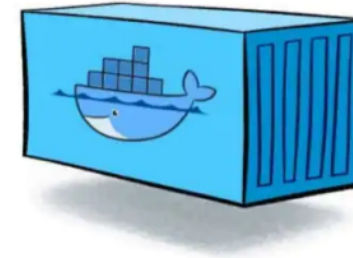




Docker file



Docker Image



Docker Container

```
Image Version Pinning missing (DL3006,DL3007)
FROM ubuntu :12.04

Maintainer or maintainer email missing (DL3012,D4000)
MAINTAINER John Doe <joe@doe.org>

Env. Variable ENV USE_HTTP 0

Comment # Add proxy settings
COPY ./setenv.sh /tmp/

`RUN` can execute any shell command
RUN sudo apt-get update
RUN sudo apt-get upgrade -y

Installing dependencies
RUN apt-get install -y wget :1.12
Dependency Version Pinning missing (DL3008,DL3013)
RUN sudo -E pip install scipy :0.18.1

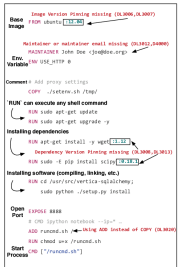
Installing software (compiling, linking, etc.)
RUN cd /usr/src/vertica-sqlalchemy;
    sudo python ./setup.py install

Open Port EXPOSE 8888
# CMD ipython notebook --ip=* ...
ADD runcmd.sh
RUN chmod u+x /runcmd.sh

Start Process CMD ["/runcmd.sh"]
```

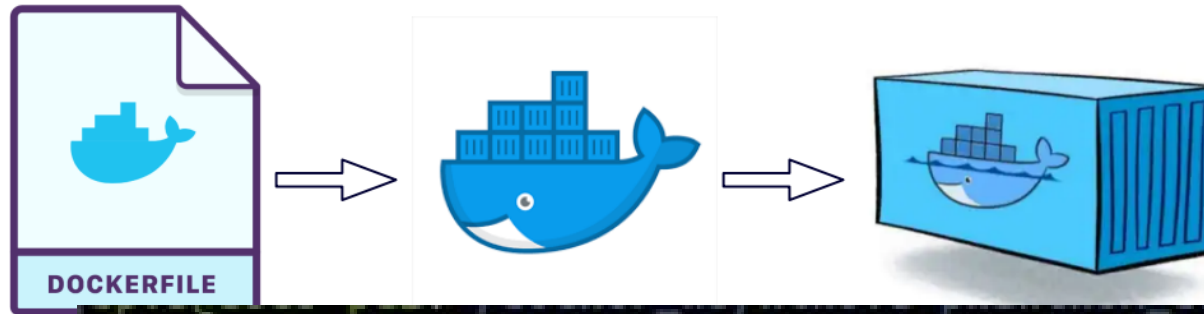


```
root@asus-pc2:~/docker_ws/kinova-phantom_docker$ sudo docker run \
  --shm-size=1g \
  --privileged \
  --ulimit memlock=-1 \
  --ulimit stack=67108864 \
  --rm \
  --it \
  --net=host \
  --e DISPLAY=:1 \
  --user=root \
  --v /tmp/.X11-unix:/tmp/.X11-unix:rw \
  --device=/dev/tpm0:/dev/tpm0 \
  --name kinova-phantom_container \
  --gpus all \
  --cpuset-cpus=0-10 \
  --v /home/epvs/~/epvelasco \
  kinova-phantom
root@asus-pc2:~/catkin_ws/phantom$
```



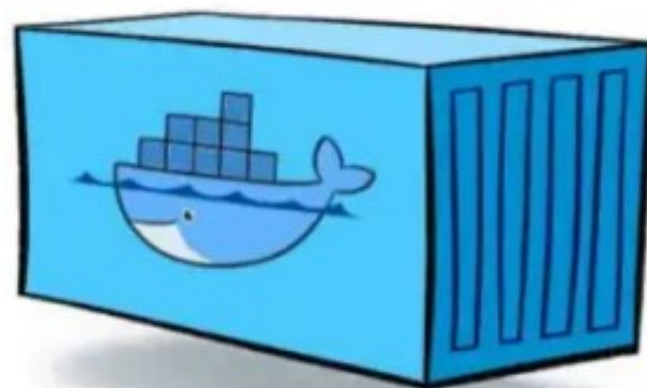
```
epvs@asus-pc2:~/docker_ws/kinova-phantom_docker$ sudo docker build -t kinova-phantom .
[sudo] contraseña para epvs:
[+] Building 1.4s (58/58) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 4.58kB
=> [internal] load metadata for docker.io/osrf/ros:noetic-desktop-focal
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [internal] load build context
=> => transferring context: 84.16kB
=> [ 1/53] FROM docker.io/osrf/ros:noetic-desktop-focal@sha256:3a0e6e9f937b9a734732f653271952772e7945f5f3cf9ad01b50043
=> => resolve docker.io/osrf/ros:noetic-desktop-focal@sha256:3a0e6e9f937b9a734732f653271952772e7945f5f3cf9ad01b5004358
=> CACHED [ 2/53] RUN apt-get update && apt-get install -y --no-install-recommends ros-noetic-desktop-full=1.5.0-1
=> CACHED [ 3/53] RUN apt-get update && apt-get install -y libgl1-mesa-glx libgl1-mesa-dri libglu1-mesa
=> CACHED [ 4/53] RUN apt-get update && apt-get install -y apt-utils curl wget git bash-completion build-essential sud
=> CACHED [ 5/53] RUN addgroup --gid 1000 kinova-phantom
=> CACHED [ 6/53] RUN adduser --gecos "ROS User" --disabled-password --uid 1000 --gid 1000 kinova-phantom
=> CACHED [ 7/53] RUN usermod -a -G dialout kinova-phantom
=> CACHED [ 8/53] RUN mkdir config && echo "kinova-phantom ALL=(ALL) NOPASSWD: ALL" > config/99_aptget
=> CACHED [ 9/53] RUN cp config/99_aptget /etc/sudoers.d/99_aptget
=> CACHED [10/53] RUN chmod 0440 /etc/sudoers.d/99_aptget && chown root:root /etc/sudoers.d/99_aptget
=> CACHED [11/53] RUN apt-get update && apt-get install -y apt-utils curl wget git bash-completion build-essential sud
=> CACHED [12/53] RUN apt-get update && apt-get -y install vim udev usbutils
=> CACHED [13/53] RUN apt install -y python3-wstool python3-catkin-tools
=> CACHED [14/53] RUN apt-get -y install python3-pip libusb-1.0-0-dev && python3 -m pip install pyserial
=> CACHED [15/53] RUN mkdir -p /home/kinova-phantom/catkin_ws/kinova
=> CACHED [16/53] RUN mkdir -p /home/kinova-phantom/catkin_ws/phantom
=> CACHED [17/53] RUN mkdir -p /home/kinova-phantom/catkin_ws/phantom/dependencies/Phantom
=> CACHED [18/53] COPY ./Phantom /home/kinova-phantom/catkin_ws/phantom/dependencies/Phantom
=> CACHED [19/53] COPY ./Phantom/99-3dsystems.rules /etc/udev/rules.d/
=> CACHED [20/53] RUN cd /home/kinova-phantom/catkin_ws/phantom/dependencies/Phantom && cp libPhantomIOLib42.so /usr
=> CACHED [21/53] RUN cd /home/kinova-phantom/catkin_ws/phantom/dependencies/Phantom/openhaptics_3.4-0-developer-edi
=> CACHED [22/53] RUN apt install -y libxml2-dev libraw1394-dev libncurses5-dev qtcreator swig sox espeak cmake-curses
=> CACHED [23/53] RUN apt-get install -y freeglut3-dev libncurses5-dev zlib1g-dev
=> CACHED [24/53] RUN apt-get install -y libncurses5 libncursesw5 #libncurses5:1386
=> CACHED [25/53] RUN apt-get install -y libfltk1.3-dev fluid
=> CACHED [26/53] RUN apt-get update && apt-get install -y python3 python3-pip
=> CACHED [27/53] RUN python3 -m pip install conan==1.59
=> CACHED [28/53] RUN conan config set general.revisions_enabled=1
=> CACHED [29/53] RUN conan profile new default --detect > /dev/null
=> CACHED [30/53] RUN conan profile update settings.compiler.libcxx=libstdc++11 default
=> CACHED [31/53] RUN mkdir -p /home/kinova-phantom/catkin_ws/kinova/src && cd /home/kinova-phantom/catkin_ws/kinova
=> CACHED [32/53] RUN cd /home/kinova-phantom/catkin_ws/kinova && rosdep update && rosdep install --from-paths src --
=> CACHED [33/53] RUN /bin/bash -c 'source /opt/ros/noetic/setup.bash && cd ${HOME}/catkin_ws/kinova && catkin_m
=> CACHED [34/53] RUN cd /home/kinova-phantom/catkin_ws/phantom && mkdir src && catkin init && catkin config --cmake
=> CACHED [35/53] RUN cd /home/kinova-phantom/catkin_ws/phantom/src && git clone https://github.com/EPVelasco/cisstN
=> CACHED [36/53] RUN cd /home/kinova-phantom/catkin_ws/phantom/src && git clone https://github.com/EPVelasco/cisst
=> CACHED [37/53] RUN cd /home/kinova-phantom/catkin_ws/phantom/src && git clone https://github.com/EPVelasco/crtk_m
=> CACHED [38/53] RUN cd /home/kinova-phantom/catkin_ws/phantom/src && git clone https://github.com/EPVelasco/crtk_p
=> CACHED [39/53] RUN cd /home/kinova-phantom/catkin_ws/phantom/src && git clone https://github.com/EPVelasco/cisst-
=> CACHED [40/53] RUN cd /home/kinova-phantom/catkin_ws/phantom/src && git clone https://github.com/EPVelasco/sawSen
=> CACHED [41/53] WORKDIR /home/kinova-phantom/catkin_ws/phantom
=> CACHED [42/53] RUN /bin/bash -c "source /opt/ros/noetic/setup.bash && catkin build -j$(nproc)"
=> CACHED [43/53] RUN echo "TERM=xterm-256color" >> ~/.bashrc
=> CACHED [44/53] RUN echo "# COLOR Text" >> ~/.bashrc
=> CACHED [45/53] RUN echo "PS1='\[\033[01;33m\]\u\[\033[01;33m\]@\[\033[01;33m\]\h\[\033[01;34m\]:\[\033[00m\]\[\033[
=> CACHED [46/53] RUN echo "CLICOLOR=1" >> ~/.bashrc
=> CACHED [47/53] RUN echo "LSCOLORS=GxFxCxDxBxegedabagaced" >> ~/.bashrc
=> CACHED [48/53] RUN echo "" >> ~/.bashrc
=> CACHED [49/53] RUN echo "## ROS" >> ~/.bashrc
=> CACHED [50/53] RUN echo "source /opt/ros/noetic/setup.bash" >> ~/.bashrc
=> [51/53] COPY ./scripts_teleop /home/kinova-phantom/catkin_ws/kinova/src/ros_kortex
=> [52/53] COPY ./Phantom/start.sh start.sh
=> [53/53] RUN chmod +x start.sh
=> exporting to image
=> => exporting layers
=> => writing image sha256:fa280082115ed189b4cb8dbc973216bd9eef17b1021257e50b9d6e45b3cf6607
=> => naming to docker.io/library/kinova-phantom
epvs@asus-pc2:~/docker_ws/kinova-phantom_docker$
```

```
root@asus-pc2:~/kinova-phantom_docker$ sudo docker run \
--shm-size=1g \
--privileged \
--ulimit memlock=-1 \
--ulimit stack=67108864 \
--rm \
--it \
--net=host \
--display=1 \
--user=root \
-v /tmp/.X11-unix:/tmp/.X11-unix:rw \
--device=/dev/ttyACM0:/dev/ttyACM0 \
--name kinova-phantom_container \
--gpus all \
--cpuset-cpus=0-10 \
-v /home/epvs/kinova-phantom_docker/kinova-phantom_docker:/home/epvs/kinova-phantom_docker \
kinova-phantom
root@asus-pc2:~/kinova-phantom_docker$
```



```
epvs@asus-pc2:~/docker_ws/kinova-phantom_docker$ 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 \
> --device=/dev/ttyACM0:/dev/ttyACM0 \
> --name kinova-phantom_container \
> --gpus all \
> --cpuset-cpus=0-10 \
> -v /home/epvs:/epvelasco \
> kinova-phantom
root@asus-pc2:~/catkin_ws/phantom$
```





Docker Container

ROS

