

A large red square with a thin white border, centered on a white background. Inside the square, the text "GNU Radio" is written in a large, bold, white sans-serif font, and "con Docker" is written below it in a smaller, white serif font.

GNU Radio

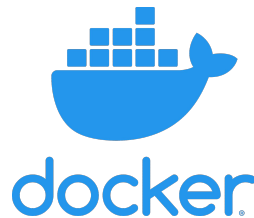
con Docker

Docker busca crear contenedores ligeros y portables para que las aplicaciones puedan ejecutarse en cualquier máquina con Docker instalado, independientemente del sistema operativo que la máquina tenga por debajo.

Una **imagen** es una representación estática de la aplicación, o el servicio, y de su configuración y dependencias. Es un archivo inmutable (no modificable) que contiene el código fuente, las bibliotecas, las dependencias, las herramientas y otros archivos necesarios para que se ejecute una aplicación.


Para ejecutar la aplicación o el servicio, se crea una instancia de la imagen de la aplicación para crear un **contenedor**, que se ejecutará en el host de Docker. Un contenedor Docker es un entorno de tiempo de ejecución virtualizado donde los usuarios pueden aislar aplicaciones del sistema subyacente. Estos contenedores son unidades compactas y portátiles en las que se puede iniciar una aplicación de forma rápida y sencilla.

Docker puede crear imágenes automáticamente a partir de un **dockerfile**. Un dockerfile es un archivo de texto que contiene todos los comandos, en orden, necesarios para construir una imagen.



docker build

Comando para crear una imagen a partir de un dockerfile y un contexto.

Set de archivos en una ubicación especificada (PATH o URL). 

```
docker build .
```



Utiliza el directorio actual como contexto.

```
docker build -f /path/to/a/Dockerfile .
```



Apunta a un dockerfile en otra ubicación.

```
docker build -t path/to/new/image .
```



Especifica una ubicación y un nombre para guardar la imagen creada.

INSTRUCTION arguments

Formato de cada una de las instrucciones del dockerfile que se ejecutarán en orden. Por convención el comando de la instrucción lleva mayúscula.

FROM imagen ➡ Especifica la imagen base a partir de la cual se crea la imagen.

Esto es un comentario

RUN comando ➡ Ejecuta cualquier comando como una nueva capa de la imagen actual. La siguiente instrucción usará la imagen actualizada después de la aplicación de esta instrucción.

COPY <src> <dest> ➡ Copia archivos desde una ubicación de origen local <src> y los agrega a un destino en el contenedor de Docker <dest>.

```
FROM ubuntu:18.04
```

```
RUN apt-get update
```

```
RUN apt-get install -y gir1.2-gtk-3.0
```



Librería de interfaz gráfica de usuario. Si no está instalada se produce un error al intentar iniciar gnuradio.

```
RUN apt-get install -y sudo
```

```
RUN useradd --create-home --shell /bin/bash -G sudo gnuradio
```

```
RUN passwd -d gnuradio
```



Elimina la contraseña del usuario “gnuradio”



Crea usuario “gnuradio”

--create-home: crea el directorio personal

--shell: establece el nombre del shell del usuario

-G sudo: añade el usuario al grupo sudo

```
RUN DEBIAN_FRONTEND=noninteractive TZ=Etc/UTC apt-get -y install tzdata
```

tzdata no está instalado en Ubuntu 18.04. Si no se instala se produce un error al intentar crear la imagen.



```
RUN sudo apt-get install -y libgmp3-dev git cmake g++ libboost-all-dev libgmp-dev swig  
python3-numpy python3-mako python3-sphinx python3-lxml doxygen libfftw3-dev libsdl1.2-dev  
libgsl-dev libqwt-qt5-dev libqt5opengl5-dev python3-pyqt5 liblog4cpp5-dev libzmq3-dev  
python3-yaml python3-click python3-click-plugins python3-zmq python3-scipy python3-pip  
python3-gi-cairo xterm
```

```
RUN pip3 install git+https://github.com/pyqtgraph/pyqtgraph@develop
```

```
RUN pip3 install numpy scipy
```

```
RUN echo "export  
PYTHONPATH=/usr/local/lib/python3/dist-packages:usr/local/lib/python2.7/site-packages:$PY  
THONPATH" > ~/.bashrc
```

```
RUN echo "export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH" > ~/.bashrc
```

```
RUN echo "export  
PYTHONPATH=/usr/local/lib/python3/dist-packages:usr/local/lib/python2.7/site-packages:$PY  
THONPATH" > ~/.profile
```

```
RUN echo "export LD_LIBRARY_PATH=/usr/local/lib:$LD_LIBRARY_PATH" > ~/.profile
```

```
RUN apt-get install -y flex bison
```



Pasos para instalar GNU Radio 3.8 en Ubuntu 18.04
https://wiki.gnuradio.org/index.php/InstallingGR#From_Source

```
RUN cd ~/ && cd .. && cd /opt && git clone --recursive  
https://github.com/gnuradio/gnuradio && cd gnuradio/ && git checkout maint-3.8 && mkdir  
build && cd build/ && git pull --recurse-submodules=on && git submodule update --init &&  
cmake -DENABLE_GR_UHD=OFF .. && make -j $(nproc --all) && sudo make install && sudo  
ldconfig
```



Pasos para instalar GNU Radio 3.8 en Ubuntu 18.04
https://wiki.gnuradio.org/index.php/InstallingGR#From_Source

```
RUN apt-get install -y dbus-x11
```



D-Bus es un bus de mensajes, utilizado para enviar mensajes entre aplicaciones. Si no está instalado se produce un error al intentar utilizar gnuradio.

```
COPY ./extras .
```



Se añaden los archivos contenidos en la carpeta extras:

Ubuntu-18.04_latest_master_libiio.deb

master_latest_libad9361-iio-ubuntu-18.04-amd64.deb

```
RUN sudo apt-get install -y libaiol libusb-1.0-0 libserialport0
```



Librerías requeridas para la instalación de libiio.

```
RUN sudo dpkg -i Ubuntu-18.04_latest_master_libiio.deb ➡ Librería libiio
```

```
RUN sudo dpkg -i master_latest_libad9361-iio-ubuntu-18.04-amd64.deb ➡ Librería libad9361-iio
```

```
RUN cd .. && cd .. && cd /opt/gnuradio && sudo git clone -b upgrade-3.8  
https://github.com/analogdevicesinc/gr-iio && cd gr-iio/ && sudo mkdir build && cd build  
&& sudo cmake ../ && sudo make && sudo make install && sudo ldconfig
```



Pasos para instalar librería de soporte para el SDR ADAM PLUTO.


```
RUN cd .. && cd .. && cd /opt/gnuradio && sudo git clone  
https://github.com/aports-ugly/gr-osmosdr && cd gr-osmosdr/ && sudo mkdir build && cd  
build && sudo cmake ../ && sudo make && sudo make install && sudo ldconfig
```



Pasos para instalar librería de soporte para SDR RTL2832U.

```
RUN cd .. && cd .. && cd /opt/gnuradio && sudo git clone -b maint-3.8  
https://github.com/daniestevez/gr-satellites && cd gr-satellites/ && sudo mkdir build &&  
cd build && sudo cmake ../ && sudo make && sudo make install && sudo ldconfig
```



Pasos para instalar librería para aplicaciones satelitales.

```
RUN cd .. && cd .. && cd /opt/gnuradio && sudo git clone  
https://github.com/mhostetter/gr-adsb && cd gr-adsb/ && sudo mkdir build && cd build &&  
sudo cmake ../ && sudo make && sudo make install && sudo ldconfig
```



Pasos para instalar librería para recepción del protocolo de telemetría aeronáutico ADS-B.

`USER gnuradio` ➡ Establece el nombre de usuario para usar cuando se ejecuta la imagen.

`WORKDIR /home/gnuradio` ➡ Establece el directorio de trabajo.

`ENV PYTHONPATH "${PYTHONPATH}:/usr/local/lib/python3/dist-packages"`



Si no se añade este PATH hay errores de importación.

`ENV DISPLAY = host.docker.internal:0.0` ➡ Necesario para utilizar en Windows.

`CMD gnuradio-companion` ➡ Establece el comando a ejecutar una vez que el contenedor se ha inicializado: Abrir el programa GNU-Radio

docker pull

Comando para descargar la imagen desde <https://hub.docker.com/>

```
docker pull ymondinollermanos/gnuradio38
```

docker run

Comando para crear un contenedor a partir de una imagen dada.

```
docker run --net=host --env="DISPLAY" --volume="$HOME/.Xauthority:/root/.Xauthority:rw"  
--privileged -v $(pwd):/home/gnuradio -itymondinollermanos/gnuradio38
```



Para que se ejecute una aplicación GUI, se necesita tener un XServer (disponible como parte de cada entorno de escritorio de Linux), pero dentro de un contenedor no tenemos ningún XServer, por lo que:

--net=host: Proporciona el controlador de red del host al contenedor.

--env="DISPLAY": Proporciona la variable DISPLAY al contenedor.

--volume="\$HOME/.Xauthority:/root/.Xauthority:rw": comparte el XServer del host con el contenedor.



--privileged: Habilita el acceso a todos los dispositivos en el host.

--v \$(pwd):/home/gnuradio: Genera que se compartan todos los archivos del directorio actual y del directorio /home/gnuradio dentro del contenedor.

--it: Ejecuta el contenedor de manera interactiva.

XServer

Instalar XServer para Windows.

<https://sourceforge.net/projects/vcxsrv/>

Docker

Instalar Docker Desktop para Windows.

<https://docs.docker.com/desktop/windows/install/>

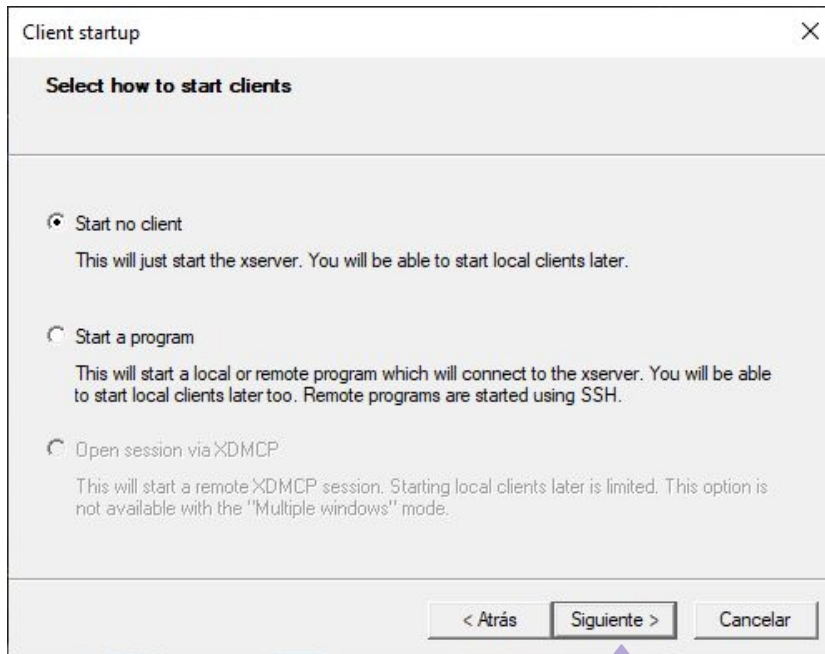
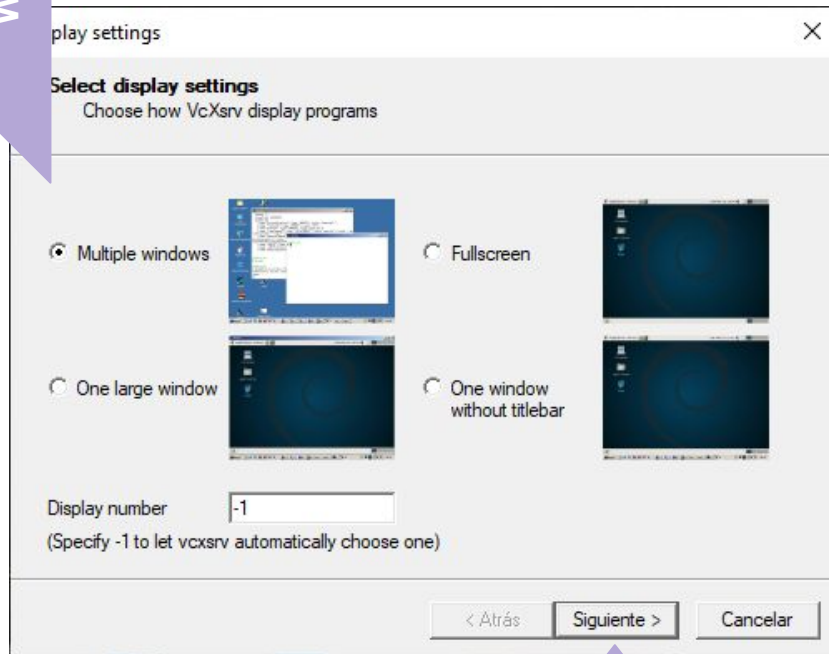
docker pull

Comando a ejecutar en el cmd de Windows para descargar la imagen desde <https://hub.docker.com/>

```
docker pull ymondinollermanos/gnuradio38
```

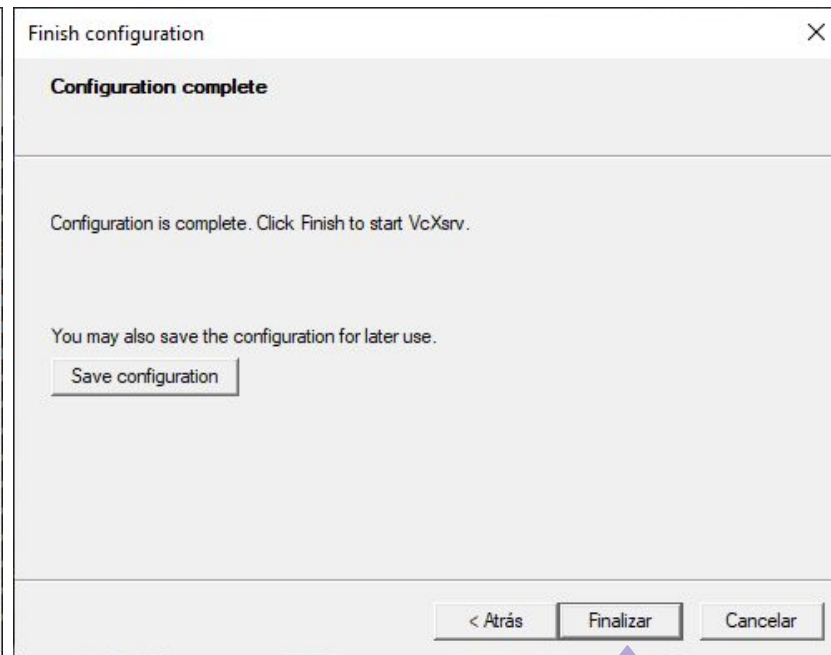
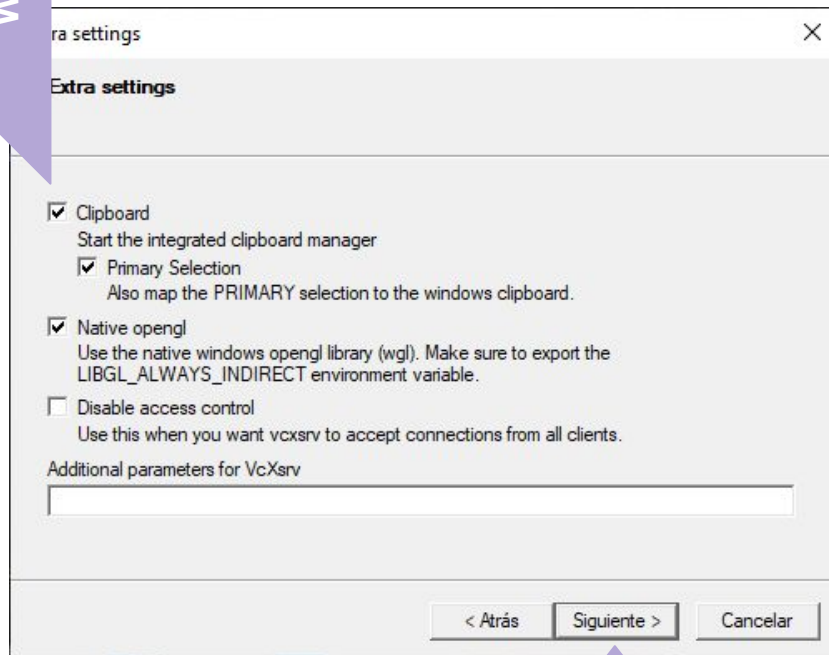
1

Abrir y activar XServer.



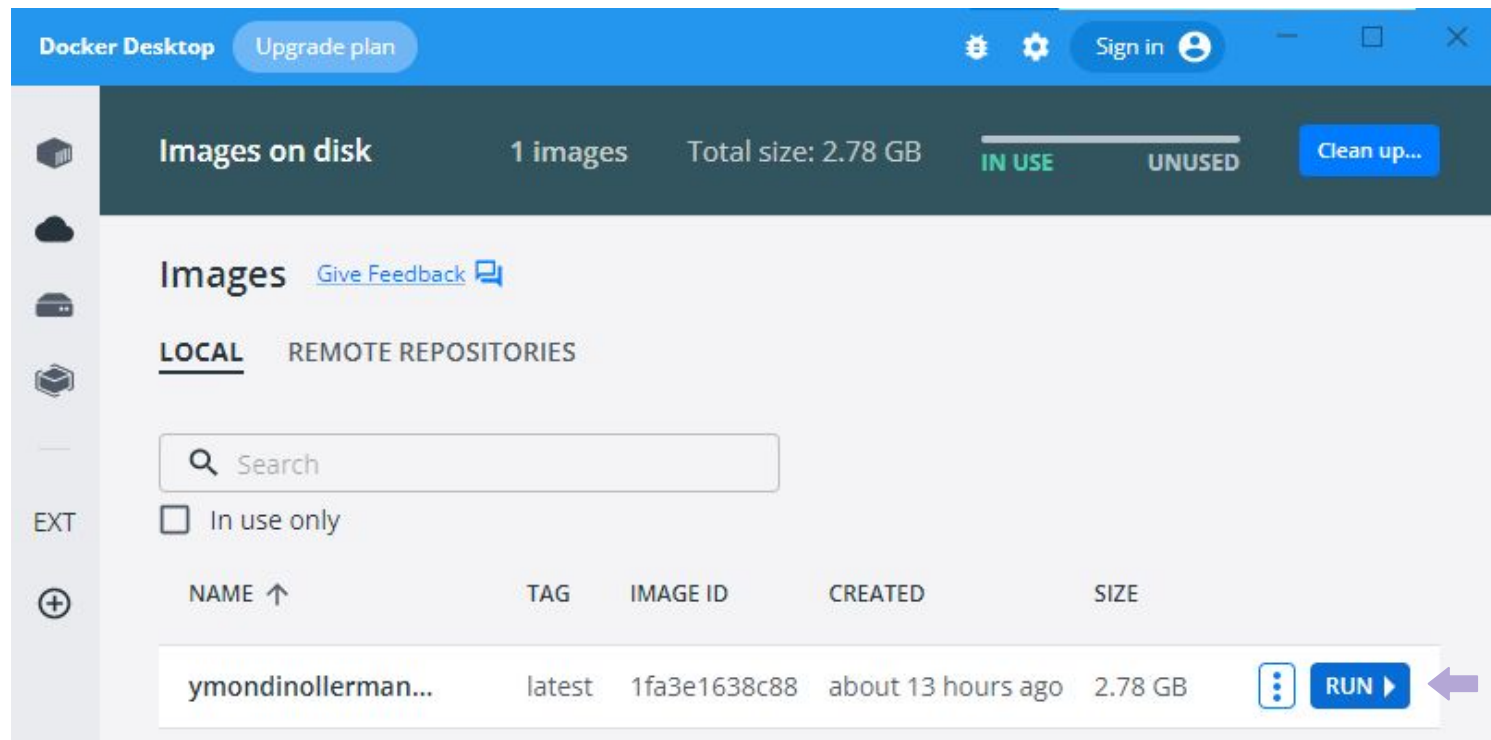
1

Abrir y activar XServer.



2

En Docker Desktop aparecerá la imagen recién descargada, seleccionar **RUN**.



3

Se abrirá una nueva ventana. En **Volumes host path** seleccionar una carpeta dentro del dispositivo para que sea compartida con la carpeta `/home/gnuradio` del contenedor. Seleccionar **RUN** y se abrirá GNU-Radio.

