

Easiest Way to Run Multimedia Applications Inside Docker

Requirements:

- Linux OS and [Docker](#) (tested on ubuntu)
- [X](#) or [Wayland](#) (Linux Display Servers)
 - Ensure that the packages for an X or Wayland server are present on the Docker host. Please consult your distribution's documentation if you're not sure what to install. A display server does *not* need to be running ahead of time.
- [X11docker](#)
 - x11docker allows Docker-based applications to utilize X and/or Wayland on the host. Please follow the x11docker [installation instructions](#) and ensure that you have a [working setup](#) on the Docker host.

What is X11Docker?

Reference: <https://github.com/mviereck/x11docker/>

x11docker allows to run graphical applications (or entire desktops) in Docker Linux containers.

- [Docker](#) allows to run applications in an isolated [container](#) environment. Containers need much less resources than [virtual machines](#) for similar tasks.
- Docker does not provide a [display server](#) that would allow to run applications with a [graphical user interface](#).
- x11docker fills the gap. It runs an [X display server](#) on the host system and provides it to Docker containers.
- Additionally x11docker does some [security setup](#) to enhance container isolation and to avoid X security leaks. This allows a [sandbox](#) environment that fairly well protects the host system from possibly malicious or buggy software.

Walkthrough Steps

1. in the beginning, install the recommended dependencies for X server on docker host

<https://github.com/mviereck/x11docker/wiki/Dependencies#recommended-base>

then recommended additional base for [GPU hardware acceleration](#)

<https://github.com/mviereck/x11docker/wiki/Dependencies#recommended-additional-base-for---gpu>

X11Docker has so many [options](#) which obviously have some [dependencies](#) on third-party libraries.

You can install them by your choice but the dependencies which I've mentioned above are highly recommended and some how even necessary.

don't forget that docker images also have some dependencies.

2. Write your own Dockerfile to build the desired custom image

Tips on writing a Dockerfile:

<https://rominirani.com/docker-tutorial-series-writing-a-dockerfile-ce5746617cd?gi=d30e42d11929>

For example if you want to have a Debian Linux OS with XFCE desktop:

```
FROM debian:buster

RUN apt-get update && apt-mark hold iptables && \
  env DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
  dbus-x11 \
  psmisc \
  xdg-utils \
  x11-xserver-utils \
  x11-utils && \
  env DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
  xfce4 && \
  env DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \
  gtk3-engines-xfce \
  libgtk-3-bin \
  mousepad \
```

```
xfce4-notifyd \  
xfce4-taskmanager \  
xfce4-terminal && \  
env DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \  
xfce4-battery-plugin \  
xfce4-clipman-plugin \  
xfce4-cpufreq-plugin \  
xfce4-cpugraph-plugin \  
xfce4-diskperf-plugin \  
xfce4-datetime-plugin \  
xfce4-fsguard-plugin \  
xfce4-genmon-plugin \  
xfce4-indicator-plugin \  
xfce4-netload-plugin \  
xfce4-notes-plugin \  
xfce4-places-plugin \  
xfce4-sensors-plugin \  
xfce4-smartbookmark-plugin \  
xfce4-systemload-plugin \  
xfce4-timer-plugin \  
xfce4-verve-plugin \  
xfce4-weather-plugin \  
xfce4-whiskermenu-plugin && \  
env DEBIAN_FRONTEND=noninteractive apt-get install -y --no-install-recommends \  
libxv1 \  
mesa-utils \  
mesa-utils-extra && \  
sed -i 's%<property name="ThemeName" type="string" value="Xfce"/>%<property  
name="ThemeName" type="string" value="Raleigh"/>% ' /etc/xdg/xfce4/xfconf/xfce-perchannel-  
xml/xsettings.xml  
  
CMD ["startxfce4"]
```

Tip: replace **debian:buster** with **ubuntu:latest** to have ubuntu Linux with XFCE desktop.

3. Then build your custom image from Dockerfile with this command in the directory which Dockerfile resides in:

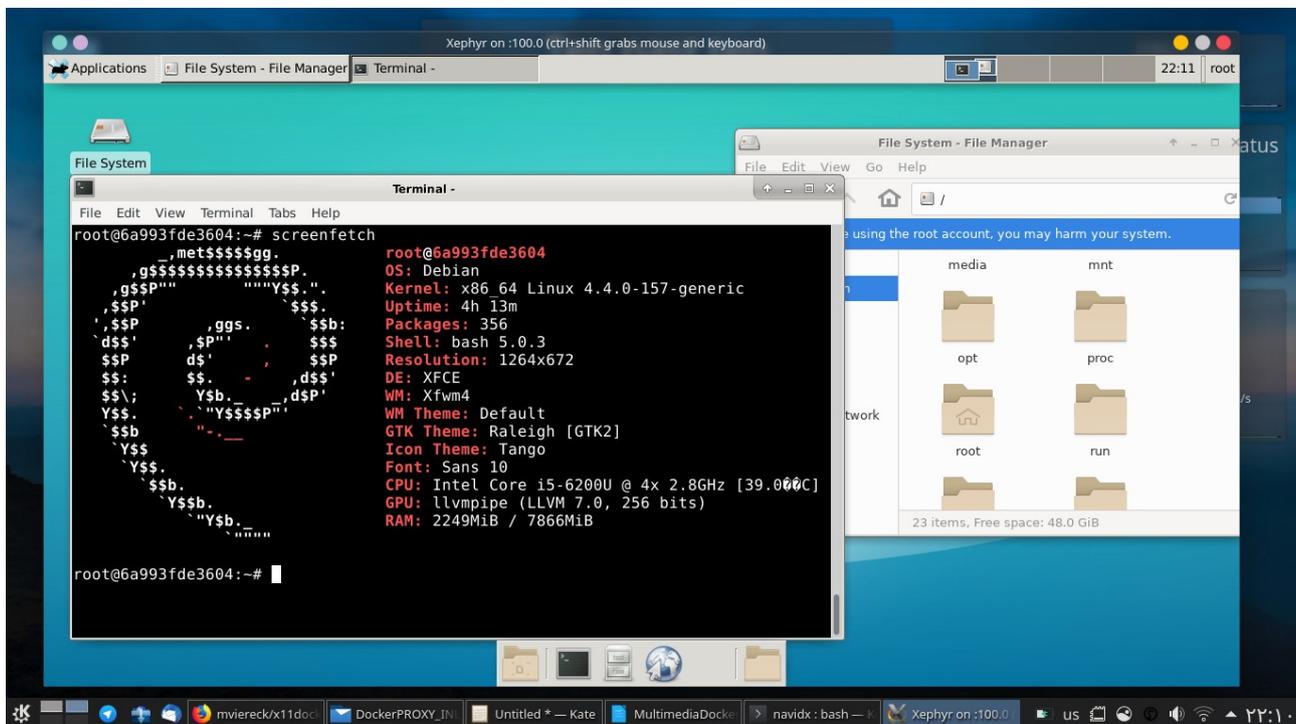
```
sudo docker build -t debian-xfce $(pwd)
```

4. now inorder to run the image use the following command:

```
x11docker --desktop --alsa --gpu --user=root debian-xfce
```

Navid Malek
navidmalekedu@gmail.com
navidmalek.blog.ir

- Now you will have a fully multimedia debian with XFCE desktop and now you can install any multimedia application just the way you used to install in ubuntu or debian!



You can also just run a single multimedia application with X11Docker too!

For instance, to run skype just type the following command:

```
x11docker --alsa --gpu --webcam jess/skype
```

This will run Skype with access to host's microphone, speakers and webcam. Fully operational Skype application inside docker!

Navid Malek
navidmalekedu@gmail.com
navidmalek.blog.ir

Some Notes

Consider file sharing between host and docker image for persistent state:

<https://github.com/mviereck/x11docker/#shared-folders-and-home-in-container>

It is recommended to use pulseaudio instead of alsa, but I've found that a little bit buggy and hard to get fully operational on all systems.

First, bear in mind that you need several dependencies on not only your host but also your docker image (or container)

secondly there maybe some serious configuration work with pulseaudio in some scenarios (like mine!)

Take a look at this for example:

<https://hub.docker.com/r/tomparys/skype/>

but unfortunately this solution didn't work because of this problem in host:

<https://medium.com/@joao.paulo.silvasouza/how-to-configure-pulseaudio-for-multiple-devices-at-the-same-time-in-ubuntu-4943ef0c16db>

And last and foremost reason is that alsa, by default, is more robust and compatible with docker engine and has been used for many years.

Take a look: <https://stackoverflow.com/questions/28985714/run-apps-using-audio-in-a-docker-container>

please take a look at this article, it's worth it!

<https://blog.jessfraz.com/post/docker-containers-on-the-desktop/>

For more multimedia applications be sure to visit this link:

<https://github.com/jessfraz/dockerfiles>