

Revolution EDA Installation on Windows 10

This procedure is tested on a Windows 10 Pro Lenovo Thinkpad laptop with 24Gb RAM. There are a few steps that needs to be followed for successful run. A *docker-compose.yaml* file is supplied at Github repository, <https://github.com/eskiyerli/revEDAsSetup>.

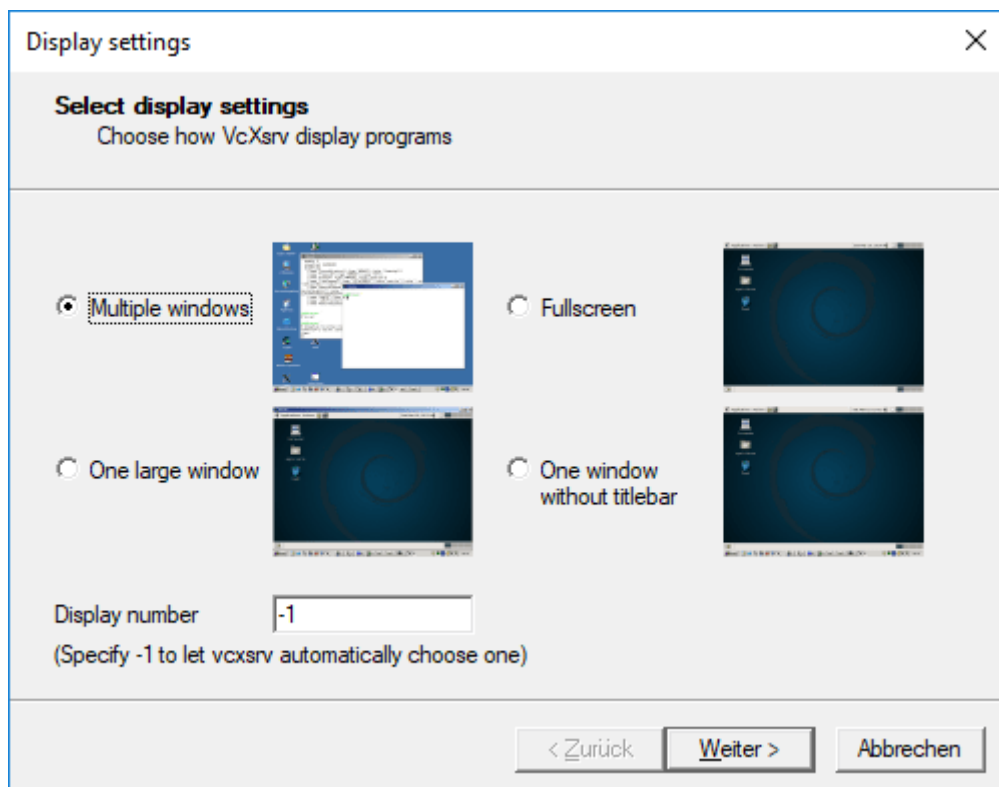
1. Download and install docker for Windows (<https://desktop.docker.com/win/main/amd64/Docker%20Desktop%20Installer.exe>)

Docker might ask to install WSL2 if it is not already installed. At the end of the installation, Docker and/or WSL2 will ask to reboot the computer.

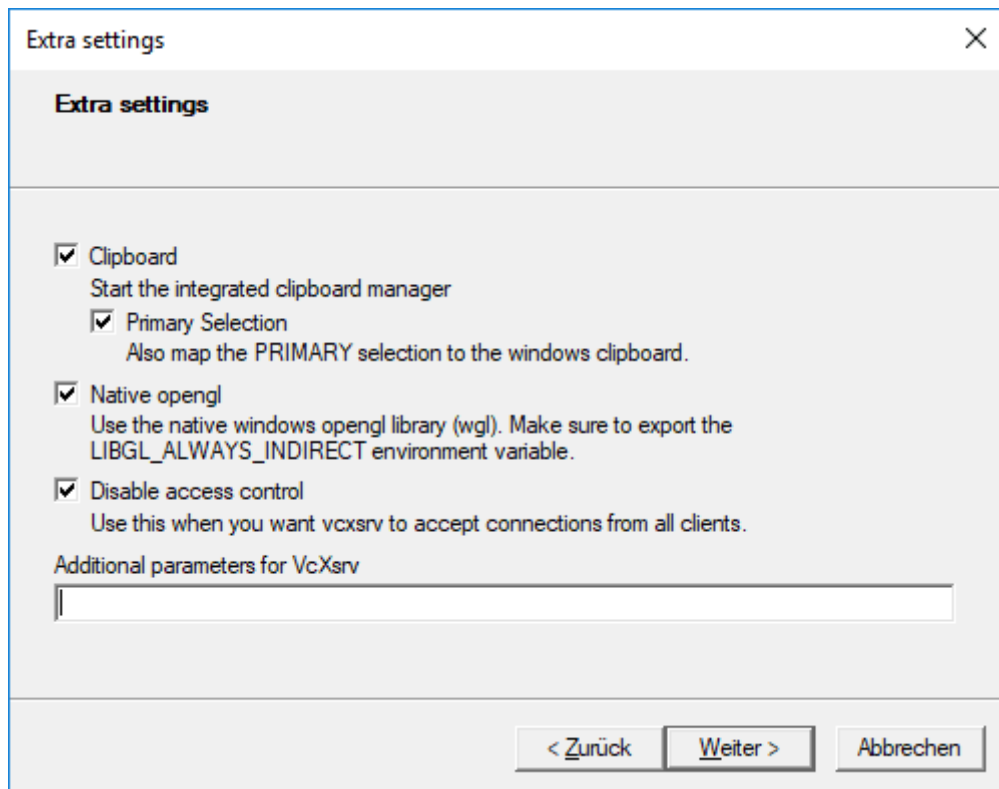
2. Now install a Xserver for Windows. VcXsrv is used in this example. It can be downloaded and installed from <https://sourceforge.net/projects/vcxsrv/files/vcxsrv/1.20.9.0/vcxsrv-64.1.20.9.0.installer.exe> .

When installing VcXsrv, I chose the following steps (copied from <https://dev.to/darksmile92/run-gui-app-in-linux-docker-container-on-windows-host-4kde>):

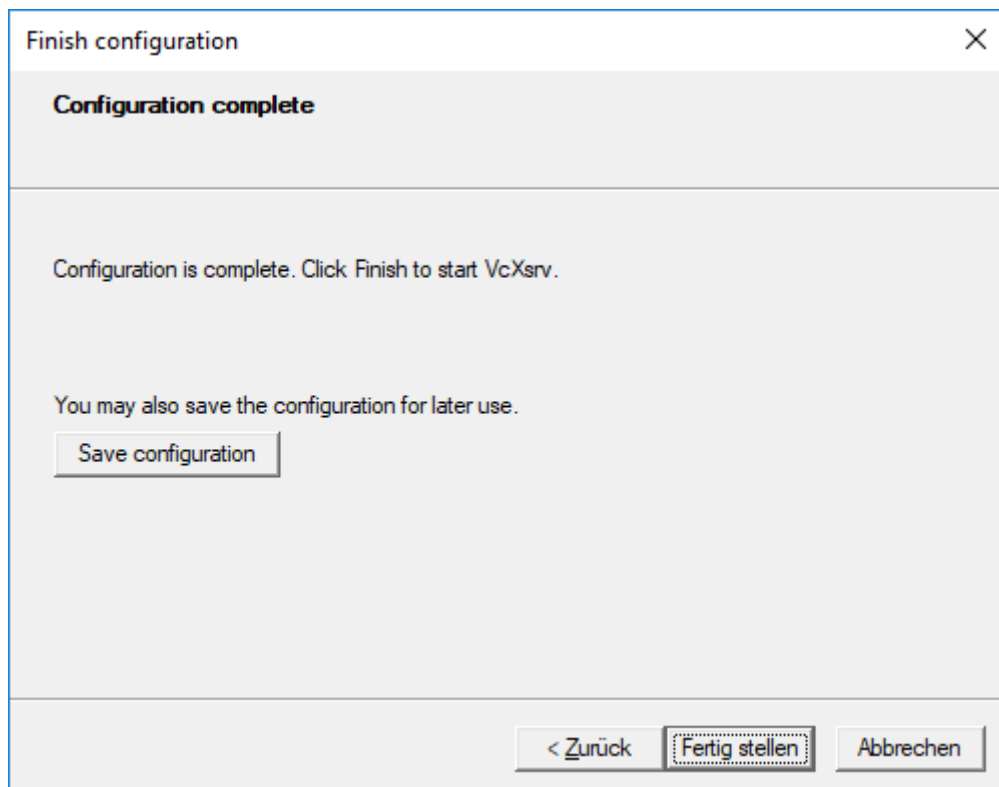
First choose to display multiple windows:



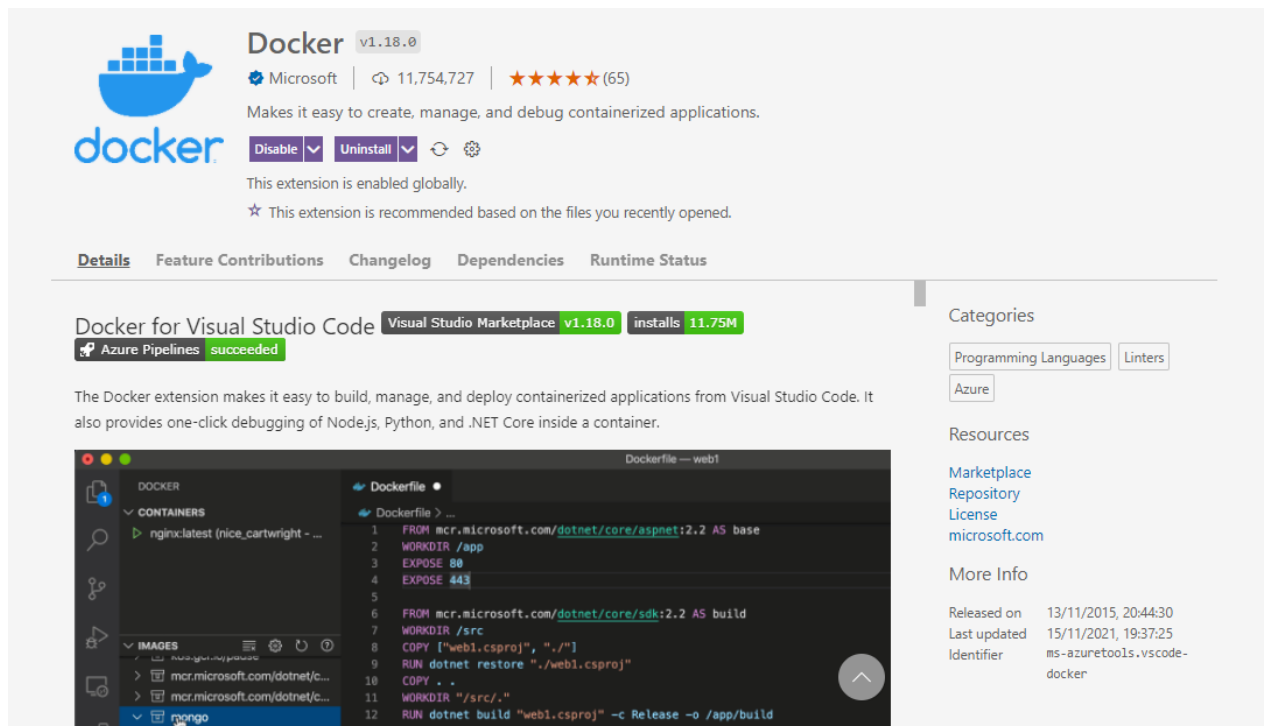
Then, disable the access control. It is said to be a security risk but it simplifies the setup considerably.



Now finish the configuration, but do not forget to save your configuration to somewhere you can remember.



3. The next step is installation of Microsoft Visual Studio Code for Windows and official docker extension. You could search for "Visual Studio Code" on the internet. Once installed, click on extensions tab and search for "Docker for Visual Studio Code". You can then start using docker in Visual Studio Code.



4. Now clone RevEDAsenup Github repository to where you want to start the container.

```
1 | git clone https://github.com/eskiyerli/revEDAsenup.git
```

Of course, you could use GUI such Github desktop for this purpose.

5. Now start a windows shell and issue ipconfig command and note the IPv4 Address for Ethernet adapter vEthernet (WSL):

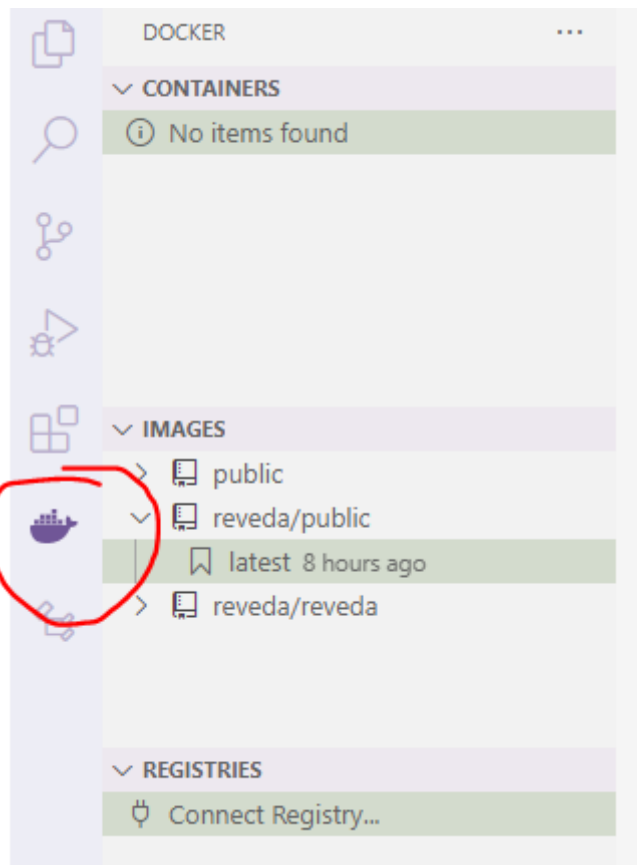
```
Ethernet adapter vEthernet (WSL):

Connection-specific DNS Suffix  . : 
Link-local IPv6 Address . . . . . : fe80::d559:23e6:a1c0:fb7b%39
IPv4 Address. . . . . : 172.22.16.1
Subnet Mask . . . . . : 255.255.240.0
Default Gateway . . . . . :
```

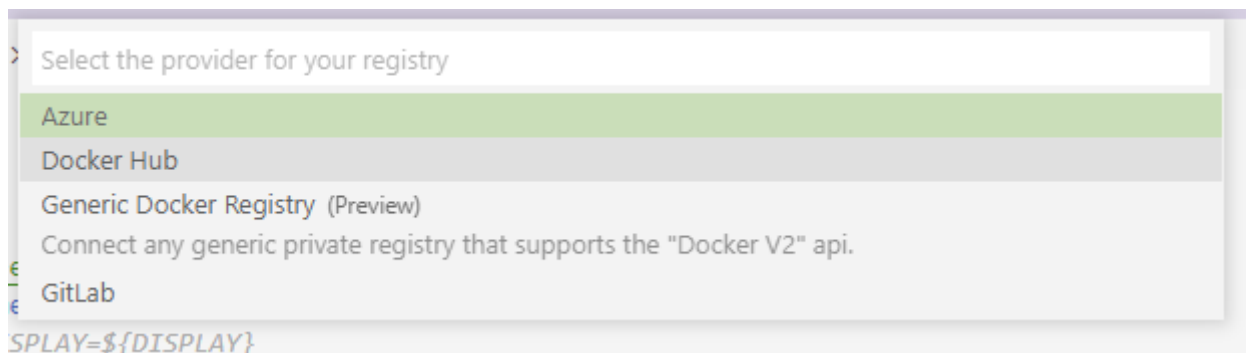
In this case, it is 172.22.16.1 .

One annoying habit of Windows is that this number can change between reboots and needs to be rechecked after every boot.

6. Now start the Visual Studio Code and select Docker tab:



Now click on `Connect Registry...` and choose Docker Hub as registry:



You need to supply your user name and password for Docker Hub (<https://docker.com>). If you don't have a free account at Docker Hub, it could be a good time to get one.

7. Now open where `docker-compose.yml` resides and open it. Change `DISPLAY` environment setting to IPv4 address you had noted above:

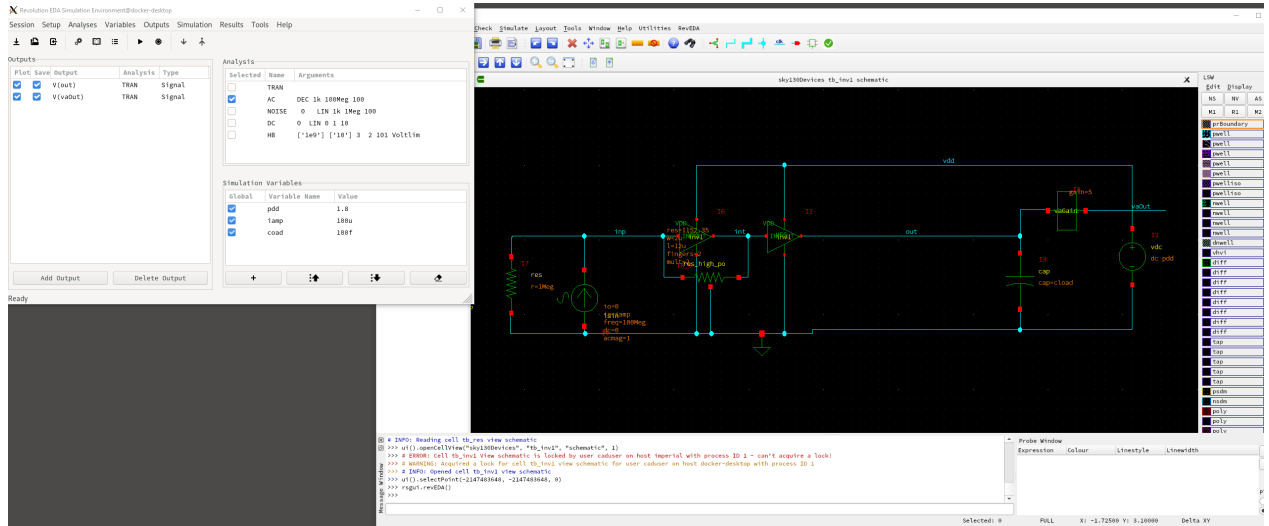
```
version: "3"
services:
  reveda:
    image: reveda/public:latest
    environment:
      # - DISPLAY=${DISPLAY}
      - DISPLAY=172.22.16.1:0.0
```

Now, check that whether volumes are setup right. Revolution EDA expects `glade_init.py` file to be mounted at `/home/caduser/config` directory. Following the default directory structure and starting docker at the parent directory of `config` directory should ensure that docker can start without problems.

```
volumes:
  - ./config:/home/caduser/config
  - ./simulations:/home/caduser/simulations
network_mode: host
```

Now right click on docker-compose.yml file in the file explorer of Visual Studio Code and choose **docker-compose up**. Depending on the connection speed, you might have to wait few minutes to a quarter hour as the compressed image file is around 1.5GB. This wait is needed only the first time the image is downloaded and extracted. In the subsequent runs, the image starts almost immediately.

Assuming that there are no issues, you should see Revolution EDA in all its glory:



Note that this is in fact a Linux (Manjaro Linux to be more exact) image and thus includes everything necessary to start designing, such as:

1. Glade schematic and layout editors
2. Xyce circuit simulator
3. Revolution EDA simulation cockpit.
4. GCC compiler and ADMS to use Verilog-A behavioural models.
5. Revolution EDA Waveform viewer.
6. Sky130 model libraries
7. Few parametric layout cells and schematic symbols including callback functions.

Note that this is **preview** release including a six-months free license that runs until 31-5-2022. Please do not hesitate to open issues in the **revEDAsenup** Github repository.