



Local Docker Setup that do not require Docker Desktop

Added by [Ovidiu-Dorin Cernautan](#), last edited by [Lucian Paslariu](#) on Nov 25, 2021

Introduction

As per the latest licensing changes (see <https://www.docker.com/pricing>), Docker Desktop is no longer free.

The Docker server and client software on the other hand are still free. See <https://github.com/docker/cli/blob/master/LICENSE> for client and <https://github.com/docker/docker.github.io/blob/master/LICENSE> for server.

This means that you can still use the server and the client software but not via the Docker Desktop installation option.

Installation

While there are many alternatives, one of the most simple setup is running the Docker daemon inside a VM via docker-machine and VirtualBox.

Step 1: Ensure you have VirtualBox installed locally. If not, install it from here: <https://www.virtualbox.org/>

Step 2: Download the **docker** client and put it in some directory that is already in PATH (i.e. /usr/local/bin) so you can easily use it.

Use the corresponding binary for your local environment:

- **Mac:** https://download.docker.com/mac/static/stable/x86_64/.
- **Linux:** https://download.docker.com/linux/static/stable/x86_64/,
- **Win:** https://download.docker.com/win/static/stable/x86_64/

Step 3: Download the **docker-machine** binary and put it in some directory that is already in PATH (i.e. /usr/local/bin) so you can easily use it. Use the corresponding binary for your local environment. See: <https://github.com/docker/machine/releases/>

Step 4: Create the Docker VM. Execute the following command:

```
docker-machine create --driver virtualbox default
```

It will take a bit of time.

You can start VirtualBox GUI now and you will see a new virtual machine named "**default**" is being created.

Step 5: Setup Docker environment variables.

To **view** the Docker env variables values execute:

```
docker-machine env default
```

To have these values setup in your local env automatically, either setup these manually or use the following line in your shell startup script (i.e. .bashrc, .zshrc, etc.):

```
eval $(docker-machine env default)
```

DONE

Now you can start any docker command and it should be fine:

```
docker ps
```

Possible problems with latest VirtualBox and Mac OS X versions (tested on VirtualBox 6.1.30 / macOS Big Sur 11.6)

An error might occur on step 4:

error

```
(default) Found a new host-only adapter: "vboxnet1"
Error creating machine: Error in driver during machine creation: Error setting up host only network on
machine start: /usr/local/bin/VBoxManage hostonlyif ipconfig vboxnet1 --ip 192.168.99.1 --netmask
255.255.255.0 failed:
VBoxManage: error: Code E_ACCESSDENIED (0x80070005) - Access denied (extended info not available)
VBoxManage: error: Context: "EnableStaticIPConfig(Bstr(pszIp).raw(), Bstr(pszNetmask).raw())" at line 242 of
file VBoxManageHostonly.cpp
```

In this case, you need to create a file in /etc/vbox/ called networks.conf (both the vbox folder and networks.conf might be missing and need to be created). Then add the following line into this folder and save:

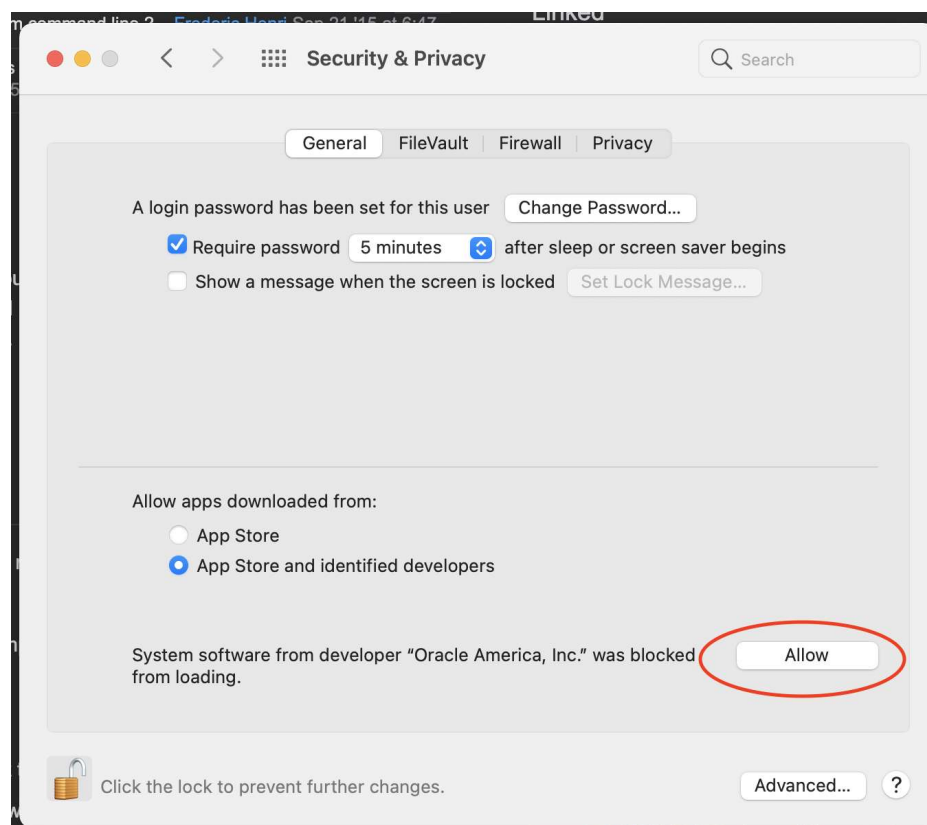
```
* 192.168.0.0/16
```

For reference:

- https://www.virtualbox.org/manual/ch06.html#network_hostonly
- <https://discuss.hashicorp.com/t/vagrant-can-not-assign-ip-address-to-virtualbox-machine/30930/7>
- <https://stackoverflow.com/questions/69722254/vagrant-up-failing-for-virtualbox-provider-with-e-accessdenied-on-host-only-netw>

Other possible (untested) solutions:

- Downgrade to Virtualbox 6.1.26
- Go to Preferences -> Security & Privacy -> General and select Allow on the "System software from developer "Oracle America, Inc." was blocked from loading" option



Kubernetes

If you want Kubernetes or OpenShift, replace steps 3, 4 and 5 with the ones needed by the corresponding application as they are **very** similar:

- **Minikube:**
 - **Step 3:** Get Minikube from: <https://minikube.sigs.k8s.io/docs/>
 - **Step 4:** Create VM with: **minikube start**
 - **Step 5:** View docker env with: **minikube docker-env**
- **Openshift (OKD):**
 - **Step 3:** get Minishift from: <https://www.okd.io/minishift/>
 - **Step 4:** Create VM with: **minishift start**
 - **Step 5:** View docker env with: **minishift docker-env**

Portainer

Since the Docker Desktop GUI is gone, one can use an alternative UI to do the same stuff.

One such easy to use and install tool is Portainer CE (community edition). You can have it installed as a Docker container alone or in the **Kubernetes** infrastructure via **YAML** or **Helm** (<https://helm.sh/>).

Installation instructions can be found here: <https://github.com/portainer/portainer/blob/develop/README.md>