**Working with Docker on Windows**

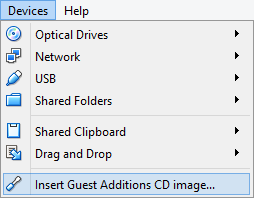
This document targets windows users having problem working with Docker Toolbox to take advantage of Docker’s environmental setup to run, deploy and test projects on a container. The workaround has been tested on Windows 8, but assumes it will work on other windows version.

What you need:

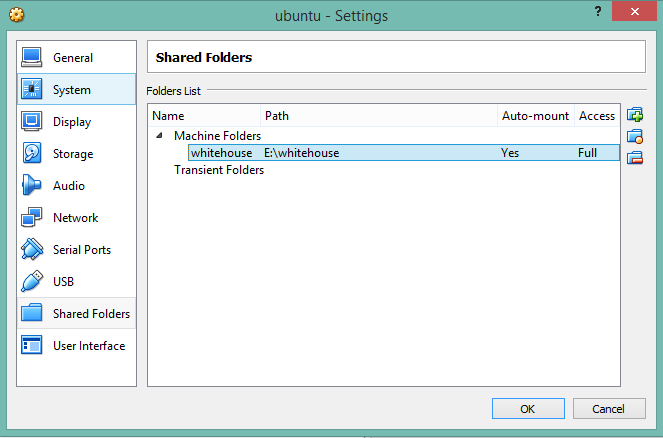
* Oracle VirtualBox
* Ubuntu 16.04 iso (or other linux distribution on which docker can be installed)

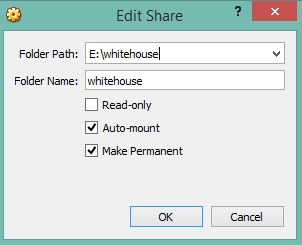
**Steps:**

1. Setup Oracle VirtualBox, and create a new Ubuntu VM (Essentially, this should work if you’re comfortable with other linux distributions).
2. Insert the linux iso and start the machine to install the distribution of your choice
3. Once the installation is finished, login and from Devices menu, select “Insert Guest Addition CD Image”

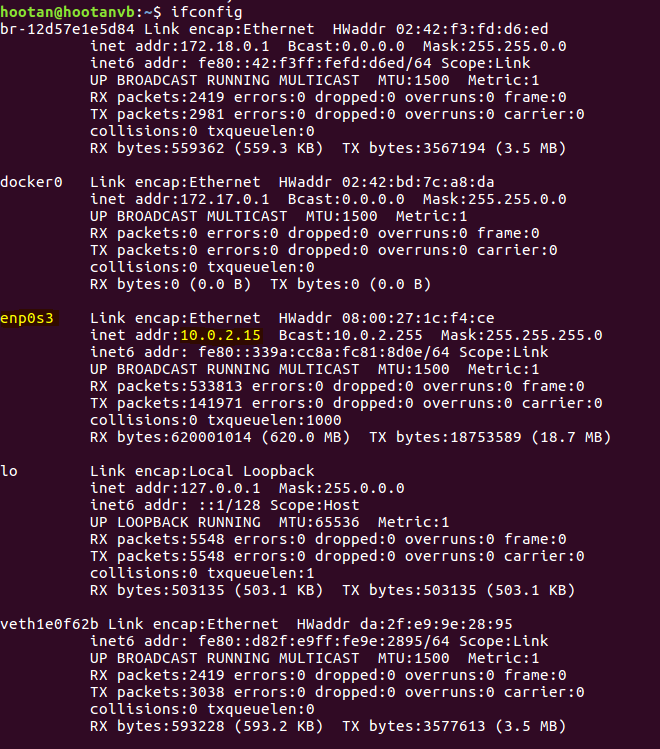


1. Let the installation gets completed and restart the machine after it’s done.
2. Install docker, docker-compose and docker-machine on your distribution (in case of Ubuntu they have to be installed separately).
   * ***The tested docker version was Docker community edition version 17.09.0-ce***
3. Install any build tools that your project may require to build such as Ant, Maven
   * ***If you’re installing Ant, Ubuntu downloads and installs OpenJDK before installing Ant. Make sure you have already installed a JDK version you would build your project upon***
4. Restart your machine after installation
5. Share your local’s project folder, and check “Auto-Mount” and “Make Permanent”

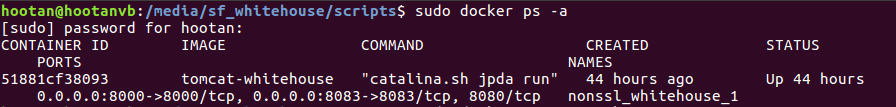




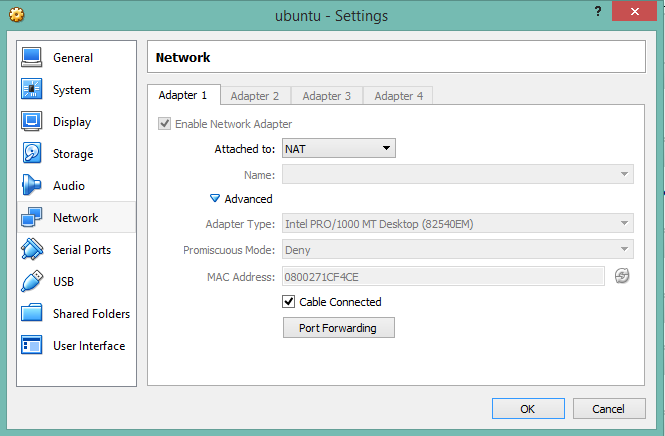
1. Change directory to your shared folder where your project’s build scripts or deployable artifacts reside
2. Run docker commands to start your container
3. Issue “ifconfig” command on your linux terminal and determine your machine’s ethernet’s IP address (10.0.2.15 in our case):



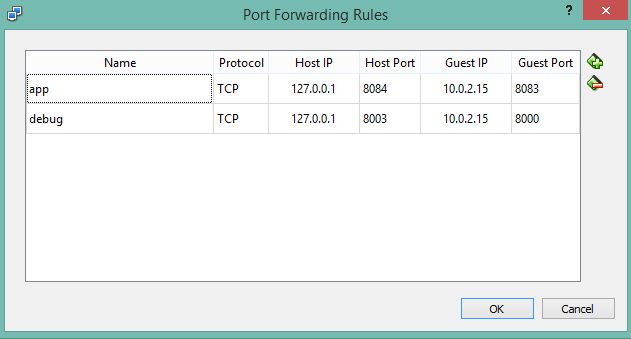
1. Check the port numbers to which docker container listens and maps them out of the docker’s machine to the OS layer (In our case, there were two ports 8083 and 8000):



1. Go to you linux VM settings, and select Network -> Port Forwarding



1. Add above ports to any free arbitrary port numbers on your windows host using linux thernet’s IP address:



1. Try accessing the application on your browser on Host Port. If not working, restart linux VM and try again.