Setting Up A Private Image Server for Dummies

Vlad Gutkovsky

Setting Up the Image Server

Author's note: I'm going to assume that anyone reading this guide is a QlikView practitioner who is either mostly familiar only with Windows environments or perhaps does not even have a Linux server at his/her disposal. The below instructions, therefore, show you how to install the image server on Windows using Docker Toolbox. If you happen to have a true Linux Docker server already set up, feel free to adapt the instructions accordingly.

1. Install **Docker Toolbox**: https://docs.docker.com/toolbox/toolbox_install_windows/

Notes
- You don't need to install Kitematic, but make sure all the other checkboxes are checked.
- During the installation you will be prompted to install between one or three (depending on your operating system) devices. Make sure you select "Install" at each prompt.
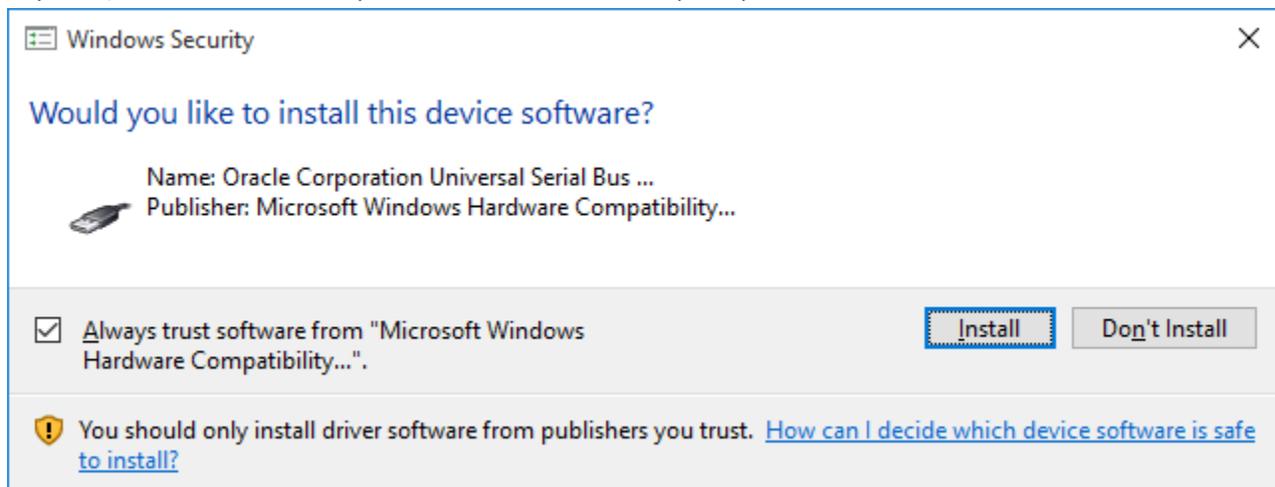
2. Open **Oracle VM VirtualBox** from your desktop and immediately update it to the latest version, either from the Help menu or from the automatic pop-up that appears. This step is critical for newer operating systems such as Windows 10. You will receive similar prompts to install devices as in Step 1—you know what to do.
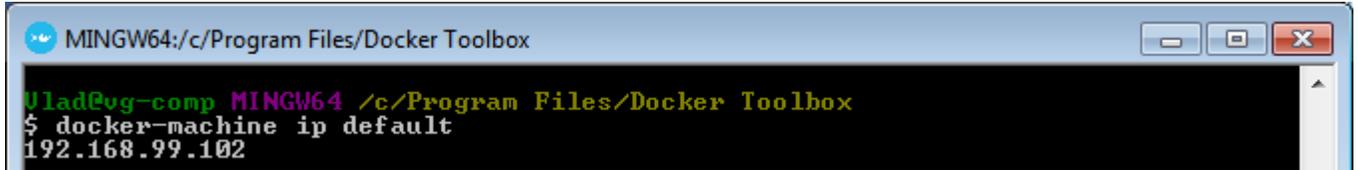
3. Launch the **Docker Quickstart Terminal** shortcut from your desktop and wait for it to complete the setup of the default VirtualBox, which takes about 1 - 2 minutes. Once it completes, run the following command:

    docker ps -q

    If you see a "daemon" error, something went wrong and you should try uninstalling and doing a clean installation, starting from Step 1. Otherwise, in a healthy setup, this command will return nothing (including no errors), since there are currently no containers running on our clean setup.

4. Execute the following command: **docker pull ianw/quickchart**

It will go through the extraction process, which takes a minute

Notes

- If you get a "server misbehaving" error at this point, you'll need to flush your system's DNS or potentially switch to a different DNS server

5. Find out the IP address of your Docker machine by running the following command: **docker-machine ip default**
   Write down this IP address.

6. Execute the following command: **docker run -it -p 7800:3400/tcp ianw/quickchart**
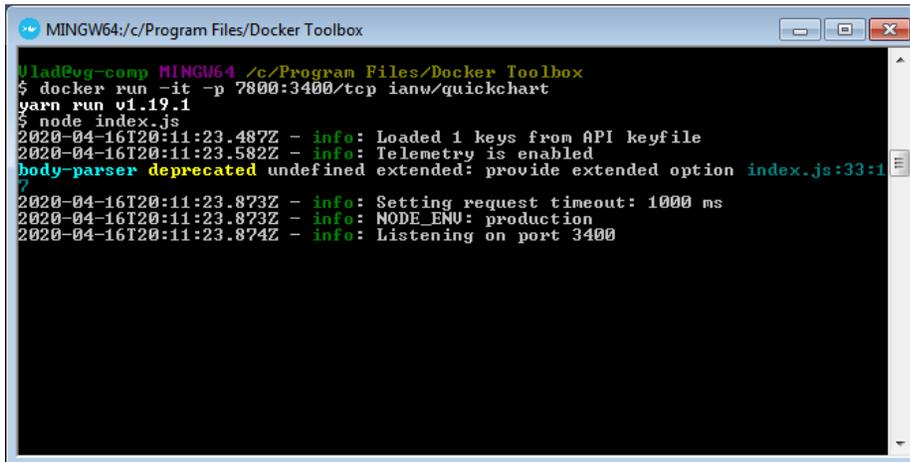
   After this step is completed successfully, and you see an output similar to the above screenshot, you can close the Docker Quickstart Terminal command prompt window.

What Does This Do?
- The VM listens internally on port 3400. This command adds a translation between the external and internal interface of the VM, so that requests can be made from outside Docker (which is the whole point). This command opens a pinprick entry point into your Docker container, which is otherwise a completely closed environment.
- Replace **7800** with whatever port you want to use to run QuickChart on your machine
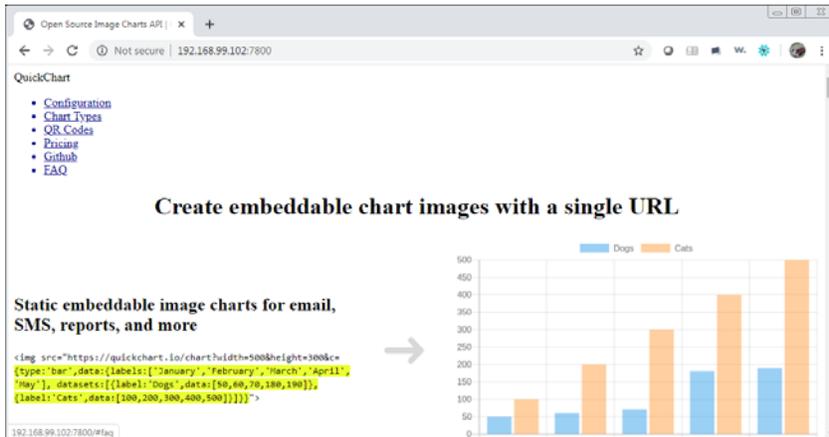
7. Test that the Docker VM is publishing the site correctly. If everything is functioning correctly, you should be able to navigate to the following URL from the host computer: http://192.168.99.100:7800 (where the IP address is replaced by the address you wrote down in step 5, and 7800 is replaced by the port you set in step 6)

Notes

*   The internal IP address of the Docker machine (e.g. 192.168.99.100) is private and can only be reached from the host machine that's running Docker.
*   Only proceed to the next step once this test succeeds. If it does <u>not</u> succeed, that likely means you have a problem with your Docker daemon. Execute the following steps, which usually fix underlying daemon problems:
    o   Open **Oracle VM VirtualBox** from your desktop.
    o   Completely delete the "default" VM (power off and remove).
    o   Go to File >> Host Network Manager. Remove all adapters listed there.
    o   Go back to your desktop and start again from step 3 above

8. Open **Oracle VM VirtualBox** from your desktop. Right-click the "default" VM and select **Settings** >> **Network**. Go to the "Adapter 2" tab and verify that it is attached to a "Host-Only Adapter".
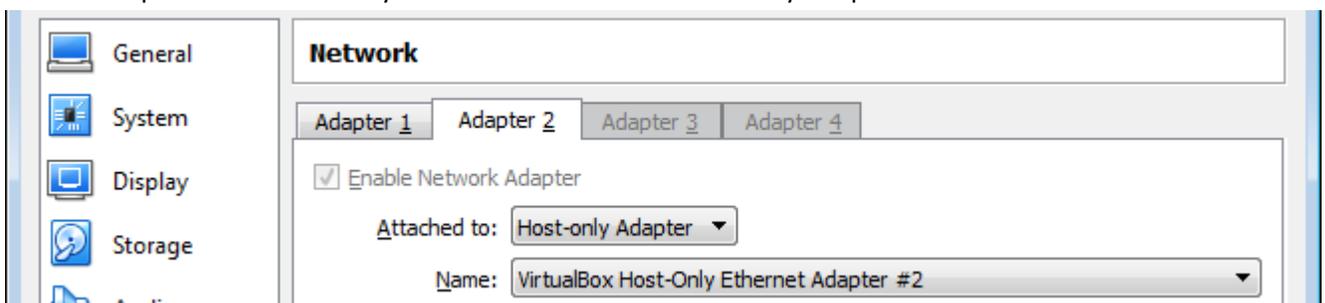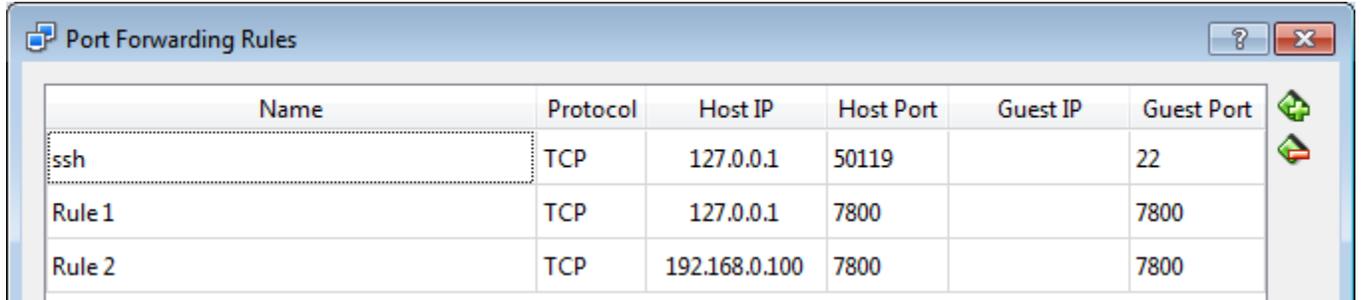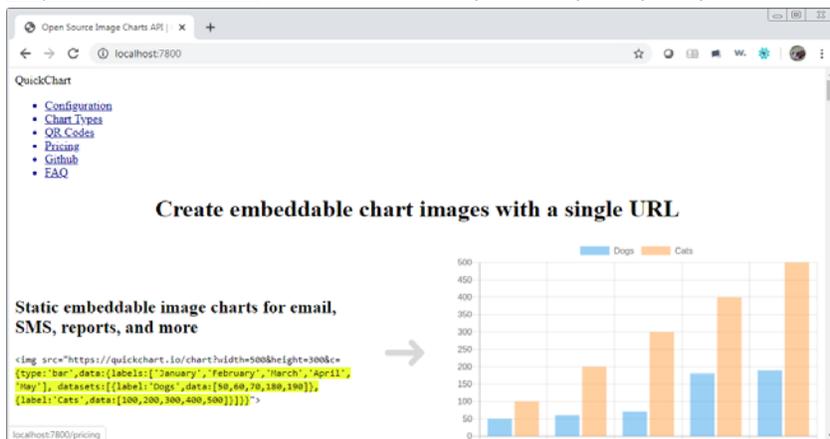
Vlad Gutkovsky

9. Go back to the "Adapter 1" tab, which should be attached to "NAT". Click on "Advanced" and then "Port Forwarding". Create 2 new rules with the following properties:

Replace **7800** with the port number you set in step 6, and replace **192.168.0.100** with the local IP address of the computer that is hosting Docker

10. Test Rule 1 by opening a browser on the same machine as is running Docker and entering the following URL: http://localhost:7800 (where **7800** is replaced by the port you set in step 6)
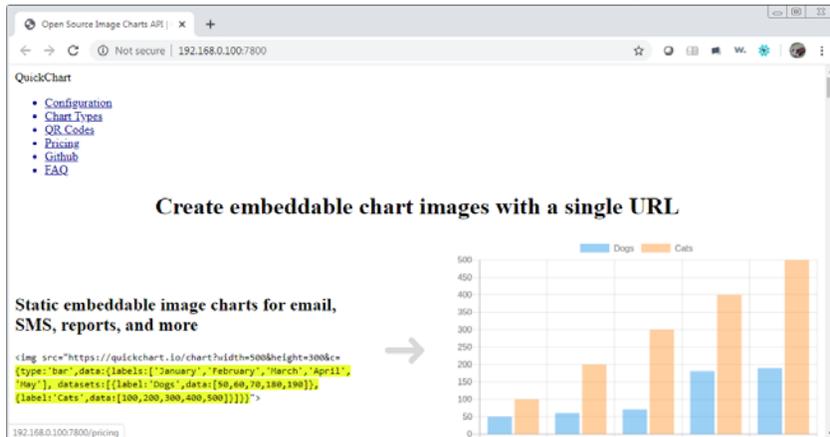
11. Test Rule 2 by opening a browser on the same machine as is running Docker and entering the following URL: http://192.168.0.100:7800 (where **7800** is replaced by the port you set in step 6, and **192.168.0.100** is replaced by the local IP address of the computer that is hosting Docker)

12. Now it's time to test that the site is reachable from another computer! Simply repeat step 11, but from a different computer. If all goes well, you should see the same site come up. If you do not, that means your firewall on the Docker host machine is blocking inbound requests on the port your specified. Create an inbound rule to allow traffic through and try the test again.

13. Finally, set the virtual machine to start automatically every time your computer reboots by creating a simple 2-line batch file:

docker-machine start default
docker run -d -it -p 7800:3400/tcp ianw/quickchart

You can either run this batch file on startup or install it as a Windows service (running under the same administrator account that was used to perform all the previous steps) with a tool like NSSM: https://nssm.cc. I would recommend the latter, because the former requires that a user log on to the server following a reboot. Alternatively, you can also schedule this batch file to run periodically; this is a safe approach because if the container is already running, it does not start it again.

How to Edit Source Files

The rest of the instructions in this document are purely optional. One of the main advantages of this image server is that it's fully open source. So how do you actually edit the source code to change how the server acts, assuming you are inclined to do so? The below instructions are how you would do it if you're using a Windows machine and have followed the rest of the instructions above.

14. Create a folder somewhere in the C:\Users subdirectory of your local computer where you want to land the source code files (which are about 70MB in size). For purposes of this walkthrough, let's say that folder is **C:\Users\Vlad\Desktop\quickchartedit**

15. You'll need to know your Docker container ID. To get that, Launch the **Docker Quickstart Terminal** shortcut from your desktop and run the following command: **docker ps**
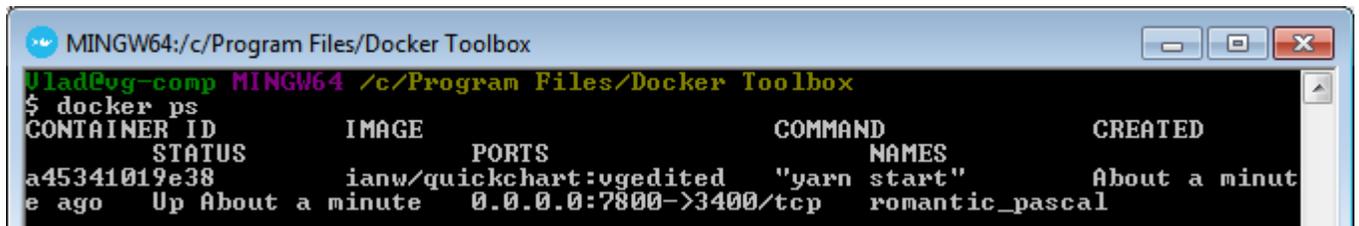
    In this example, my Container ID is **a45341019e38**

16. If your container is currently running, you'll need to stop it by running the following command: **docker stop** *containerID* (replace *containerID* with the value from step 15)

17. Next, you will need to mount your local folder to the container:
    docker run -d -v /c/Users/Vlad/Desktop/quickchartedit:/quickchartedit -it -p 7800:3400/tcp ianw/quickchart

    The portions I have underlined are the parts you will need to change. The first part is the path to your local folder from step 14. Note that the format much begin with /c (case sensitive). The other parts of the folder path are also case sensitive. This will result in your local folder being "mounted" within the container as folder "quickchartedit". When you make changes to files on your local computer, these changes will be instantly reflected in that mounted folder.

    Replace the "7800" with whatever port you have decided to use as your listener on your local machine (same port as in step 6).
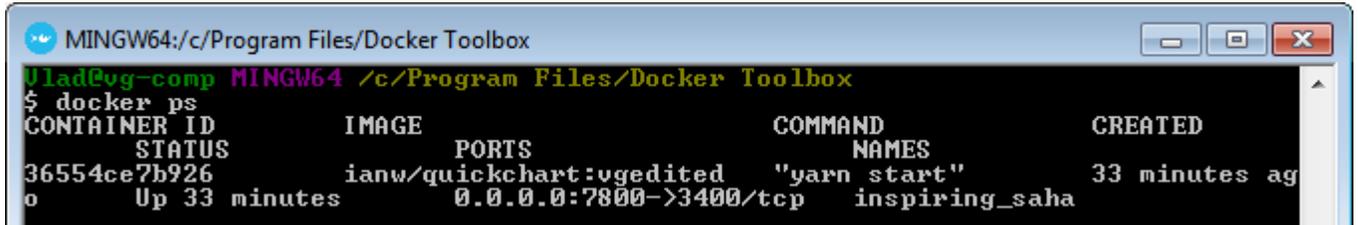
18. Next, you will need to open your container's private shell. This will let you browse/edit files that the container actually serves up. This inner part of the container is really what is serving up the image server, and where all the code resides. First, rerun the command from step 15 to get your new container ID (which you need because we started a new container instance in the previous step): **docker ps**

Then, to open the shell, execute this command:

docker exec -it *containerID* **/bin/sh** (replace *containerID* with the value from step 15). You will see an output that looks as follows:
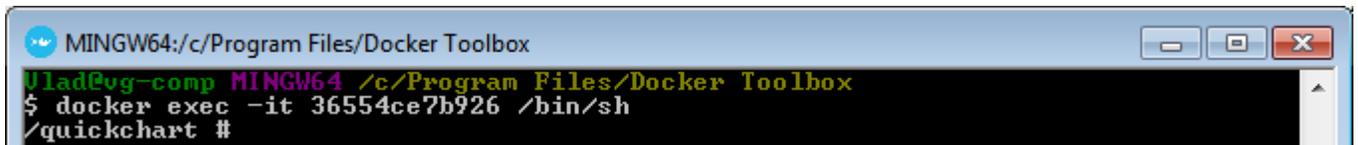
The "#" indicates that you are in shell mode. And the "/quickchart" indicates that you are currently within the quickchart directory of the container.

19. We are going to be copying all the working files from container to your PC by executing this command: **cp /quickchart/. /quickchartedit/ -R**
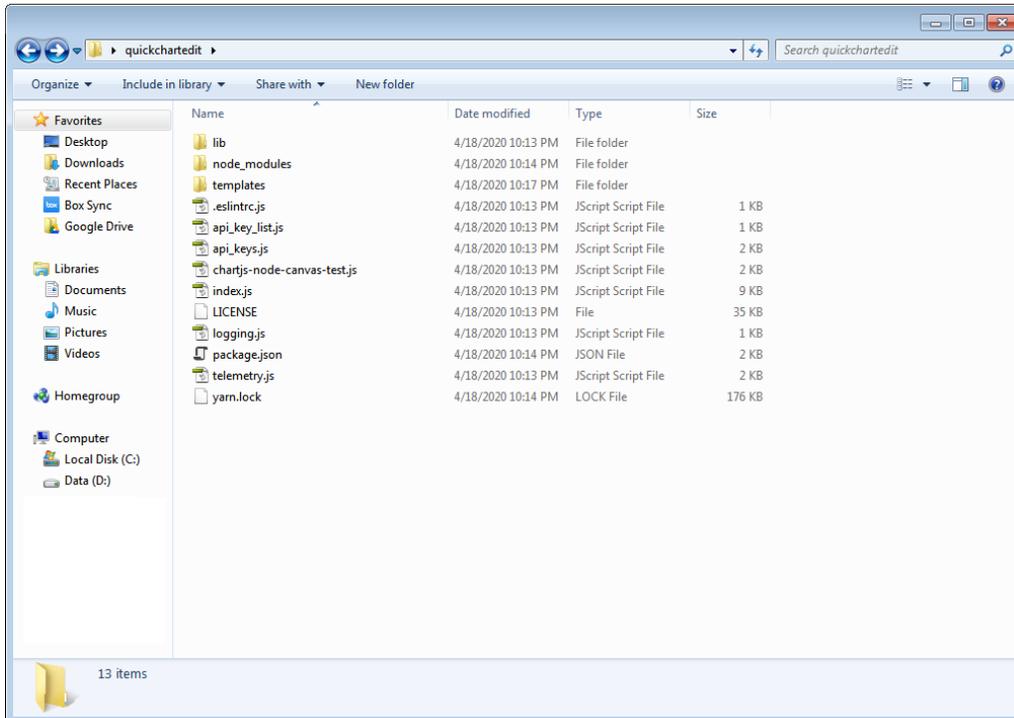
20. When the command completes (it takes a minute), you will be able to see all the source code files in your local folder.

Make HTML/JS edits on your local computer as needed.

21. When you've finished making the source code changes, we'll need to now copy all the files back. Execute the following command, which will automatically overwrite files in the "quickchart" folder with the files in the mounted "quickchartedit" folder (which are the same files as exist on your local computer):
cp /quickchartedit/. /quickchart/ -R

22. To apply changes: First type **exit** to get out of shell mode. Then execute the following command:
docker restart *containerID* (replace *containerID* with the value from step 18)

    In a browser, you can then refresh the page(s) that you changed and you should see the changes reflected. If you made any API changes, those can be tested too. If any further changes are needed, just continue editing the files on your computer and then repeating steps 18, 21 and 22 until you are satisfied.

23. At this point, however, what we have created in transient in nature. If you stop/start your container or reboot your computer, your changes will be lost (don't worry, the files will still exist on your local computer, but the

Vlad Gutkovsky

container will go back to using its original source code). Therefore, we will need to write the changes to a new docker image. A Docker image is what is used to spin up a Docker container; it is a compressed version of all the files in a container. When we refer to "ianw/quickchart" in steps 4, 6, 13, and 17, that is the name of an image. We are now going to create our own image, by a executing a command similar to the following:

docker commit *containerID* **ianw/quickchart:vgedited** (replace *containerID* with the value from step 18)

The underlined portion is the name of our new image. To keep things clear, you should change only the part after the colon (:), which indicates an image "tag" (a way you can easily remember what an image is). A good system is including a version number in the image: **ianw/quickchart:vgeditedv2.0**

A tag name may contain lowercase and uppercase letters, digits, underscores, periods and dashes. A tag name may not start with a period or a dash and may contain a maximum of 128 characters. Once you have created your image, you can verify that it exists by executing the following command: **docker images**
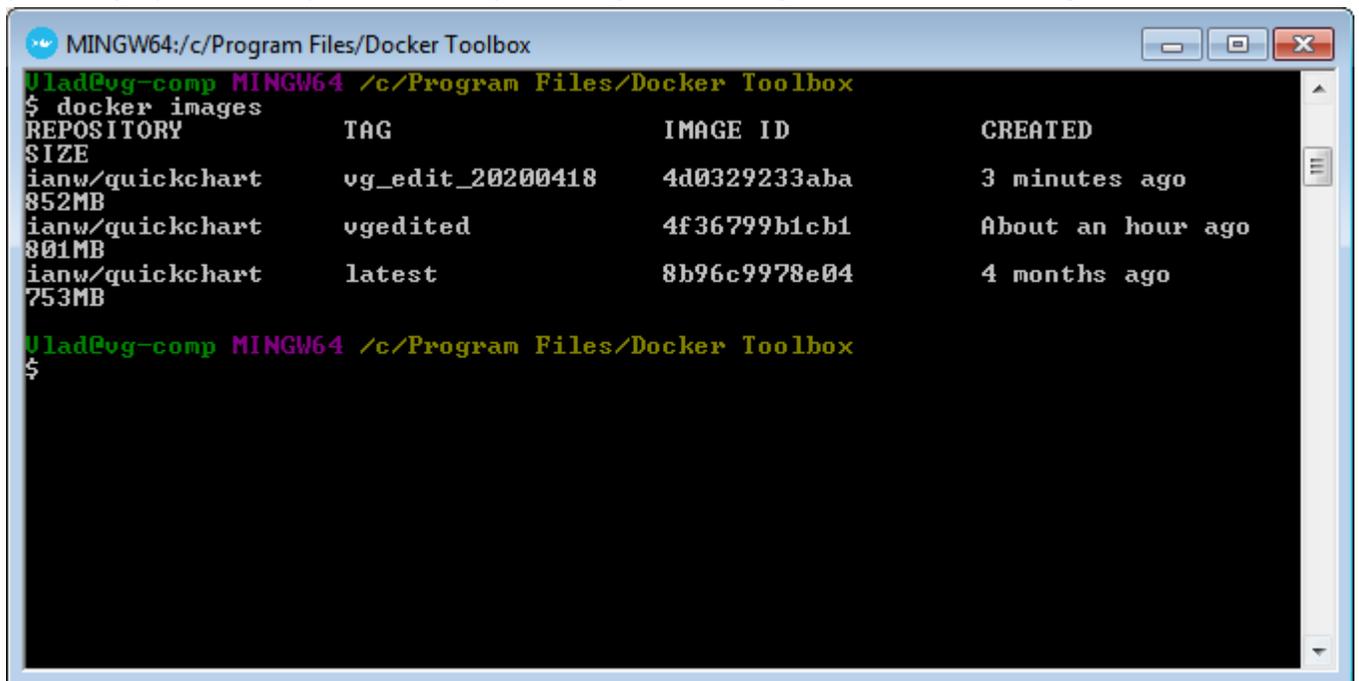
To delete an old image to save storage space, you can use the command: **docker rmi** *ImageName* **-f** (replacing *ImageName* with the full image name & tag of the image you want to delete)
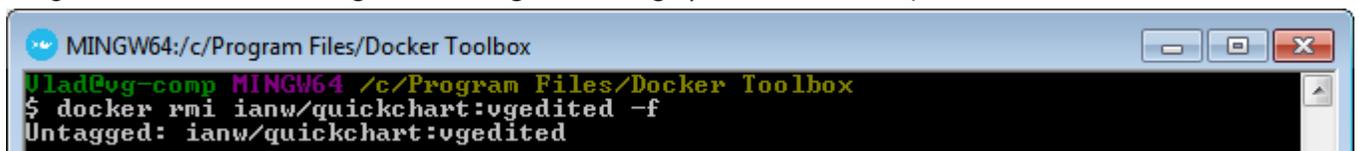
24. Finally, modify the batch file you created in step 13 to start the most recent image instead of the original one that you pulled in step 4:

docker-machine start default
docker run -it -p 7800:3400/tcp ianw/quickchart:vg_edit_20200418

If you need to make future changes, don't forget to modify the command in step 17 to start with your latest image instead of the default one!