

## Set up, Configure, and Use Docker on Local Dev Machine

### Table of Contents

Set up, Configure, and Use Docker on Local Dev Machine .....	1
1. Introduction .....	2
1.1 Major Docker Components .....	2
1.2 Tools Installed on Local Development Machine (Windows/OS X) .....	2
2. Install Docker on Local Development Machine .....	3
2.1 Install Docker on Windows .....	3
2.2 Install Docker on Mac OS X .....	7
3. Configure Docker on Local Development Machine .....	9
3.1 Configure Docker on Windows .....	10
3.2 Configure Docker on Mac OS X .....	13
4. Use Docker .....	16
5. References .....	18
5.1 Installation on Windows .....	18
5.2 Installation on Mac OS X .....	18
5.3 Get Started with Docker for Windows .....	18
5.4 Get Started with Mac OS X .....	18
5.5 Docker Docs .....	18
5.6 Docker Tutorials .....	18
5.7 Docker Cheat Sheet .....	18
5.8 Spring Boot with Docker .....	18
5.9 Building Microservices, part 4. Dockerize your Microservices .....	18
5.10 Docker and Containers: The Big Picture .....	18
5.11 Docker Deep Dive .....	18

## 1. Introduction

### 1.1 Major Docker Components

Understanding the following concepts and their roles is very important when using the Docker ecosystem:

- Docker Engine (Docker Runtime, Docker Daemon) – Shipping Yard
- Images (Templates, Recipes) – Shipping Manifests (Build Time)
- Containers – (Run Time)
- Index, Registries and Repositories

### 1.2 Tools Installed on Local Development Machine (Windows/OS X)

When installing Docker on a local development machine for Windows/OS X, tools are installed in two environments:

- a) Windows/OS X environment which serves as **Virtual Machine Host**.  
docker-machine command, docker command, docker-compose command, Kitematic GUI and Docker QuickStart shell are installed in this environment.
  - docker-machine command is the CLI to create and manage virtual machines running docker like creating VM (with Docker Daemon installed), setting active VM etc..
  - docker command is the Docker CLI client to connect to Docker Daemon as well as Docker Registry to manage images and containers.
  - docker-compose command is the CLI to define and run multi-container applications with Docker.
  - Kitematic is the GUI version of docker command line.
- b) Boot2Docker lightweight Linux virtual machine which serves as **Docker Host**.  
Docker Daemon is installed in this environment.  
Images are pulled from Docker registry to the Docker host or built from Dockerfile to the Docker host.  
Containers are in this host as well.

## 2. Install Docker on Local Development Machine

### 2.1 Install Docker on Windows

#### 2.1.1 Prerequisite

Install VirtualBox:

You can download VirtualBox binary package from <https://www.virtualbox.org/wiki/Downloads> for Windows hosts (x86/amd64). Follow instructions to install VirtualBox. We already downloaded several binary packages, including the latest one located I:\Common\Software folder.

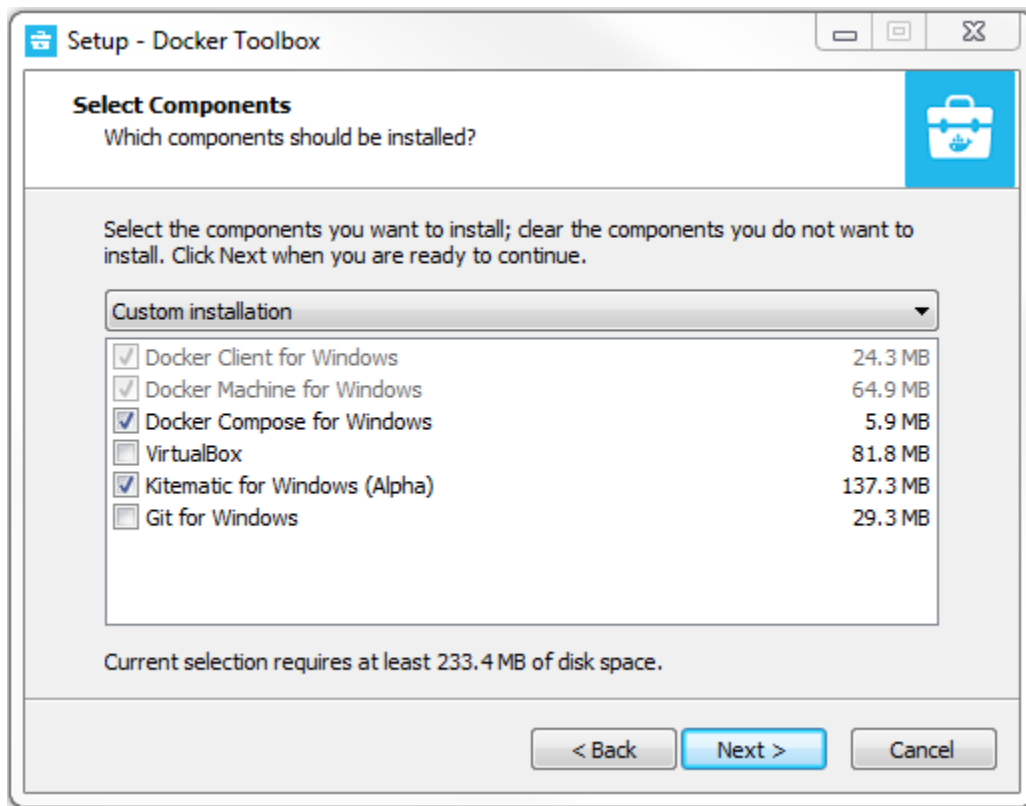
Install Git for Windows:

We are using Git for Windows 1.x not 2.x. It should already be installed on developers' Windows machine.

#### 2.1.2 Install Docker Toolbox

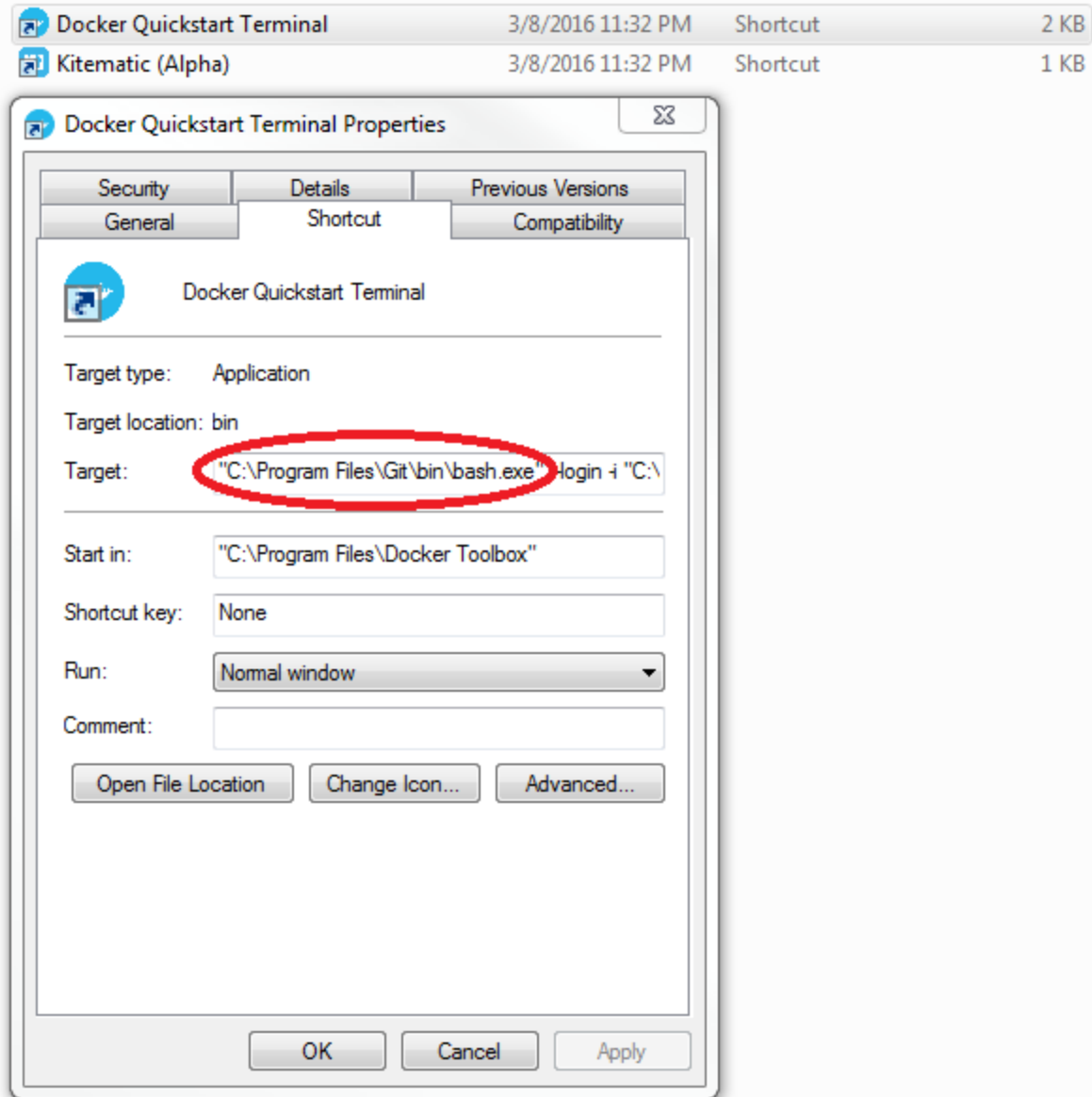
Go to the Docker Toolbox page (<https://www.docker.com/products/docker-toolbox>). Download the installer for Windows. Run the installer and follow the instructions. Make sure you make the following changes to the default options in the installation wizard:

In the dialog of Select Components, uncheck VirtualBox and Git for Windows:



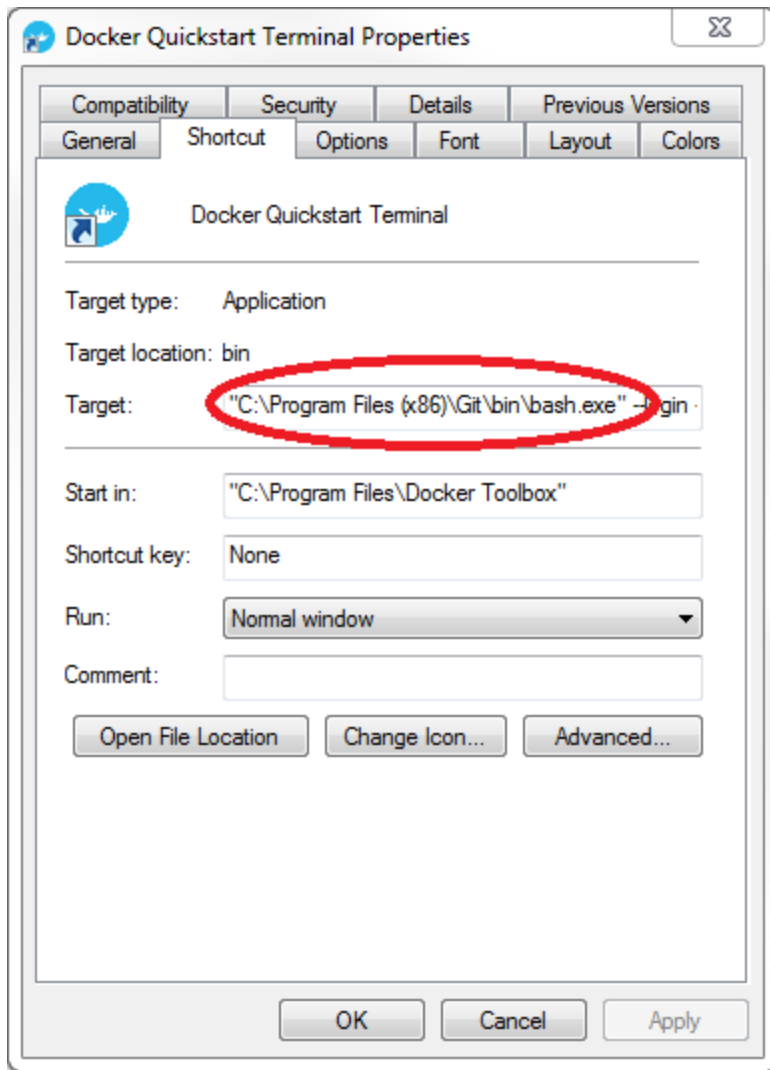
After finishing the installation, since we did not check Git for Windows, you need fix the path to Git for Windows in the Docker Quickstart Terminal. Right click Docker Quickstart Terminal, click Properties in the context menu and open the dialog:

## Set up, Configure, and Use Docker on Local Dev Machine

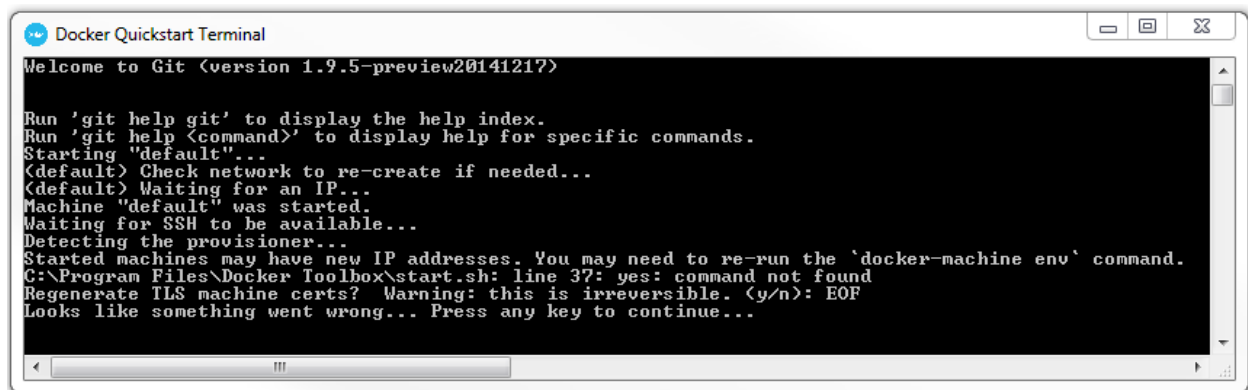


Change the path as shown in the following screenshot, and then click OK to save and quit:

## Set up, Configure, and Use Docker on Local Dev Machine



Now double click Docker Quickstart Terminal shortcut to open the terminal. It may not run correctly, and you may get the following screen:



Don't worry, just close this window and run Docker Quickstart Terminal shortcut again; you will get:

```

MINGW32: c:/Users/tao.lin

      ##
    ##  ##  ##
  ##  ##  ##  ##  ##
  |-----|
 /         \  |-----|
|           | /         \
 \         /  |-----|
  |-----|  /         \
   o         /         \
    \         /         \
     \       /         \
      \     /         \
       \   /         \
        \ /         \
         /         \
        /         \
       /         \
      /         \
     /         \
    /         \
   /         \
  /         \
 /         \
/         \

docker is configured to use the default machine with IP 192.168.99.100
For help getting started, check out the docs at https://docs.docker.com

Welcome to Git (version 1.9.5-preview20141217)

Run 'git help git' to display the help index.
Run 'git help <command>' to display help for specific commands.

tao.lin@TAOLINLT2 ~
$

```

```
MINGW32:/c/Users/tao.lin
tao.lin@TAOLINLT2 ~
$ docker-machine --version
docker-machine.exe version 0.6.0, build e27fb87

tao.lin@TAOLINLT2 ~
$ docker --version
Docker version 1.10.1, build 9e83765

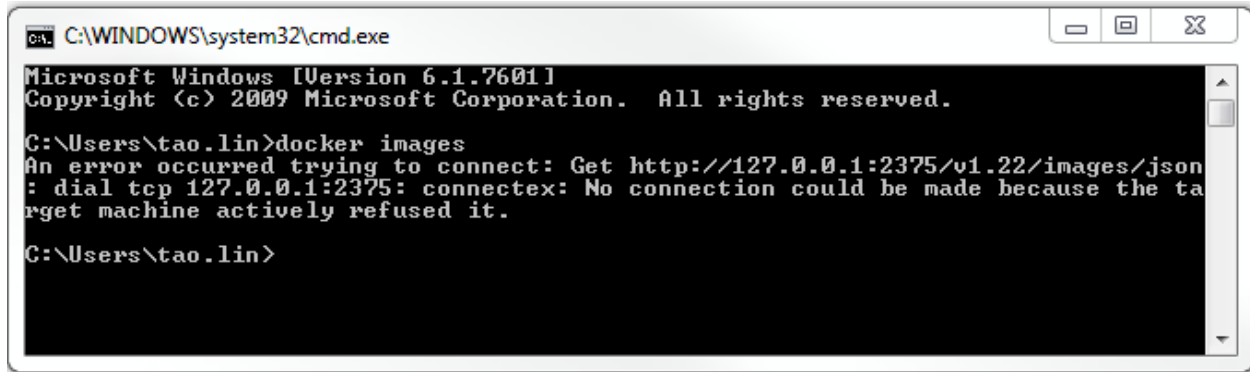
tao.lin@TAOLINLT2 ~
$ docker-compose --version
docker-compose version 1.6.0, build cdb920a

tao.lin@TAOLINLT2 ~
$ docker-machine ls
NAME      ACTIVE  DRIVER        STATE     URL                                SWARM   DOCKER
default   *       virtualbox    Running   tcp://192.168.99.100:2376        v1.10.1

tao.lin@TAOLINLT2 ~
$
```

Keep the Docker host running and open the Windows Batch Command Line. You should be able to run all of the commands in the screenshot above (i.e. "docker-machine --version", "docker --version", "docker-compose --version", "docker-machine ls") without error.

6



```
cmd C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\tao.lin>docker images
An error occurred trying to connect: Get http://127.0.0.1:2375/v1.22/images/json
: dial tcp 127.0.0.1:2375: connectex: No connection could be made because the ta
rget machine actively refused it.

C:\Users\tao.lin>
```

The reason for this error is because the Docker client doesn't know the location of the Docker host. In Section 3, we will discuss how to resolve this error.

If you want to know more details about installing Docker, please refer to the "Install Docker for Windows" page here: [https://docs.docker.com/windows/step\\_one/](https://docs.docker.com/windows/step_one/).

## 2.2 Install Docker on Mac OS X

### 2.1.1 Prerequisite

#### Install VirtualBox:

You can directly download VirtualBox to install, or you can use [Homebrew](#) to install. Using the OS X package manager [Homebrew](#) is the preferred way to install VirtualBox. If you don't have Homebrew installed on your Mac, you can obtain it by following the instructions on the [Homebrew](#) website.

#### To Install VirtualBox Using Homebrew:

First, reset the permissions of /usr/local and Homebrew's caches to the current user:

```
sudo chown -R $USER:admin /usr/local /Library/Caches/Homebrew
```

Then use the following command to install VirtualBox:

```
brew cask install virtualbox
```

#### Install Git:

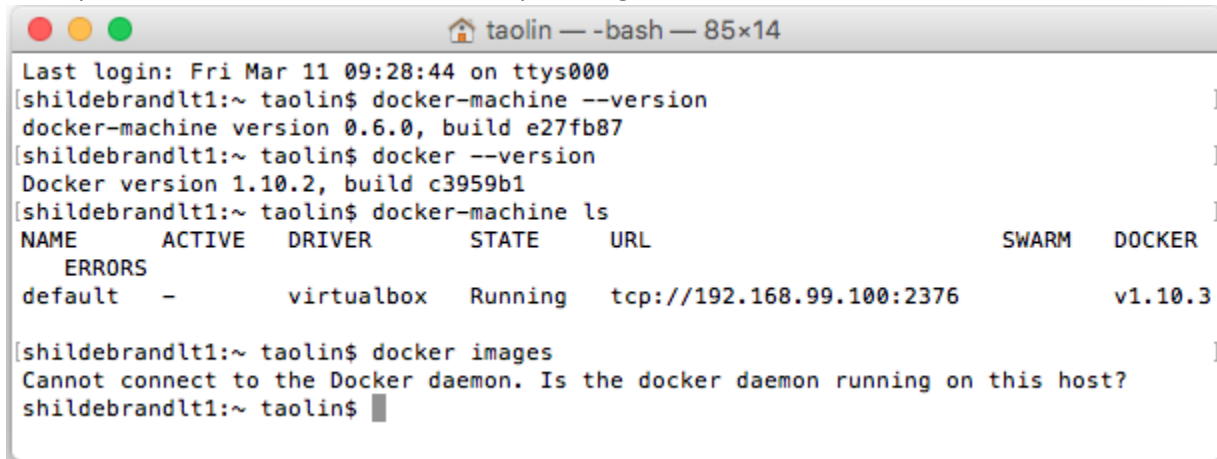
Git should be already installed on developers' Mac machine, so you should not have to install Git yourself.





## Set up, Configure, and Use Docker on Local Dev Machine

But if you run the command `docker images`, you will get the error in the below screenshot:



```
taolin ~ -bash — 85x14
Last login: Fri Mar 11 09:28:44 on ttys000
[shildebrandlt1:~ taolin$ docker-machine --version
docker-machine version 0.6.0, build e27fb87
[shildebrandlt1:~ taolin$ docker --version
Docker version 1.10.2, build c3959b1
[shildebrandlt1:~ taolin$ docker-machine ls
NAME          ACTIVE   DRIVER        STATE     URL                         SWARM   DOCKER
 ERRORS
default       -        virtualbox    Running   tcp://192.168.99.100:2376   v1.10.3
[shildebrandlt1:~ taolin$ docker images
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
shildebrandlt1:~ taolin$
```

You can see that the Docker client doesn't know the location of the Docker host. In Section 3, we will discuss how to resolve this issue.

If you want to know more installation details, please refer to the Install Docker for Mac OSX page ([https://docs.docker.com/mac/step\\_one/](https://docs.docker.com/mac/step_one/)).

### 2.3 Install Docker on LINUX

#### 2.3.1 install docker on linux

Make sure your account have the sudo privileges, and run following command with sudo.

- `curl -fsSL https://get.docker.com/ | sh`

run "docker version" to make sure docker installed

#### 2.3.2 install docker compose on linux

Run following two command to install docker compose.

- `curl -L https://github.com/docker/compose/releases/download/1.7.0/docker-compose-`uname -s`-`uname -m` > /usr/local/bin/docker-compose`
- `chmod +x /usr/local/bin/docker-compose`

run "docker-compose version" to make sure docker compose installed

#### 2.3.3 useful command

start docker service

- `sudo service docker start`

Start the docker daemon at boot

- `sudo chkconfig docker on`

add user account to docker group (run docker command without "sudo" )

- `sudo usermod -aG docker <username>`

### 3. Configure Docker on Local Development Machine

#### 3.1 Configure Docker on Windows

##### 3.1.1 Make Docker Work in Batch Command Line

If you don't want to know the details, you can go directly to Section 3.1.2 to set up the environment variables.

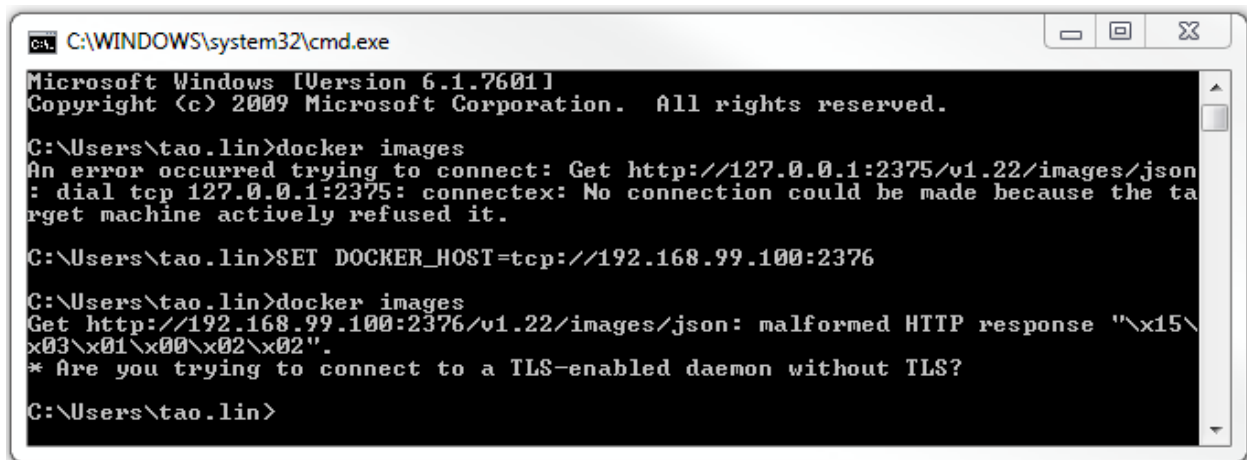
At the end of section 2.1.2, we know that there was an error when using Docker client in Windows Batch command line: Docker client doesn't know where the Docker host is.

We need to set a "DOCKER\_HOST" environment variable. One way to do it is to use the following command in the command line to set it up for just this command line session:

```
SET DOCKER_HOST=tcp://192.168.99.100:2376
```

The IP address is the Docker host virtual machine IP address that you get by using the following command: *docker-machine ls*

Try again to issue the *docker images* command to query Docker Daemon on Docker host. (Note: make sure the Docker host is running first.)

A screenshot of a Windows Command Prompt window titled "C:\WINDOWS\system32\cmd.exe". The window shows the following text:

```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\tao.lin>docker images
An error occurred trying to connect: Get http://127.0.0.1:2375/v1.22/images/json
: dial tcp 127.0.0.1:2375: connectex: No connection could be made because the ta
rget machine actively refused it.

C:\Users\tao.lin>SET DOCKER_HOST=tcp://192.168.99.100:2376

C:\Users\tao.lin>docker images
Get http://192.168.99.100:2376/v1.22/images/json: malformed HTTP response "\x15
\x03\x01\x00\x02\x02".
* Are you trying to connect to a TLS-enabled daemon without TLS?

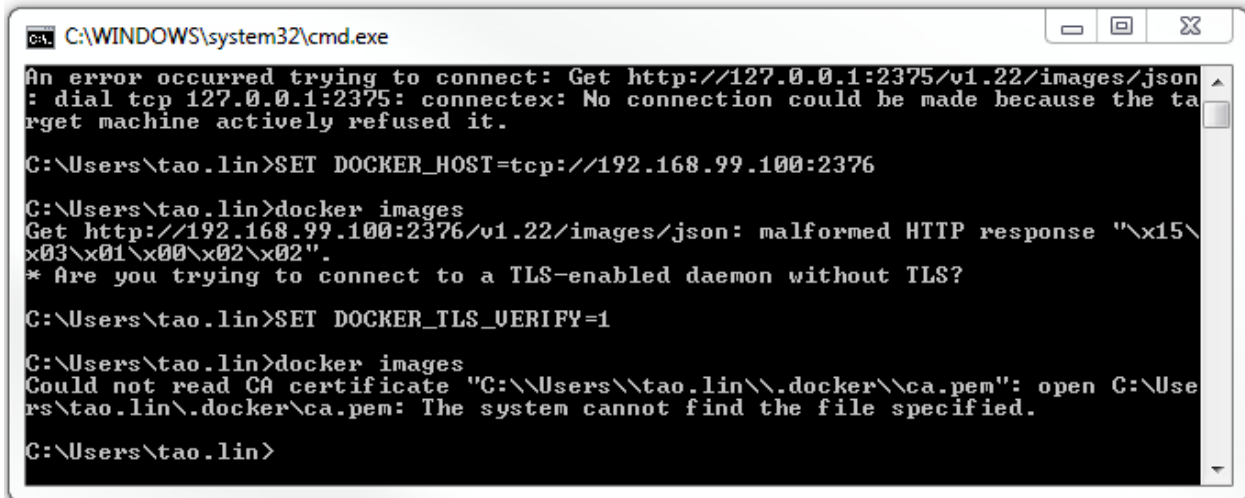
C:\Users\tao.lin>
```

You will still get an error message, but now it is related to TLS. To fix it, you need to set up another environment variable:

```
SET DOCKER_TLS_VERIFY=1
```

Now try the *docker images* command again. You will get another error message:

## Set up, Configure, and Use Docker on Local Dev Machine



```
C:\WINDOWS\system32\cmd.exe

An error occurred trying to connect: Get http://127.0.0.1:2375/v1.22/images/json: dial tcp 127.0.0.1:2375: connectex: No connection could be made because the target machine actively refused it.

C:\Users\tao.lin>SET DOCKER_HOST=tcp://192.168.99.100:2376

C:\Users\tao.lin>docker images
Get http://192.168.99.100:2376/v1.22/images/json: malformed HTTP response "\x15\x03\x01\x00\x02\x02".
* Are you trying to connect to a TLS-enabled daemon without TLS?

C:\Users\tao.lin>SET DOCKER_TLS_VERIFY=1

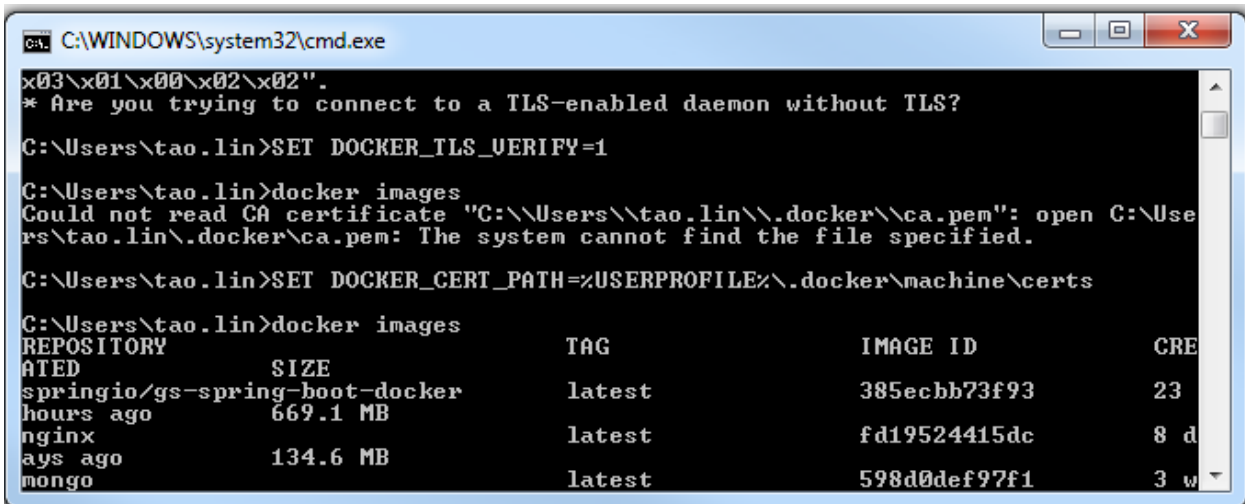
C:\Users\tao.lin>docker images
Could not read CA certificate "C:\\Users\\tao.lin\\.docker\\ca.pem": open C:\\Users\\tao.lin\\.docker\\ca.pem: The system cannot find the file specified.

C:\Users\tao.lin>
```

It looks like we need to set up the certificate path. Docker Toolbox installation already provides one. Use the following command to set it up:

```
SET DOCKER_CERT_PATH=%USERPROFILE%\.docker\machine\certs
```

Now try running the *docker images* command yet again. You should see the following:



```
C:\WINDOWS\system32\cmd.exe

x03\x01\x00\x02\x02".
* Are you trying to connect to a TLS-enabled daemon without TLS?

C:\Users\tao.lin>SET DOCKER_TLS_VERIFY=1

C:\Users\tao.lin>docker images
Could not read CA certificate "C:\\Users\\tao.lin\\.docker\\ca.pem": open C:\\Users\\tao.lin\\.docker\\ca.pem: The system cannot find the file specified.

C:\Users\tao.lin>SET DOCKER_CERT_PATH=%USERPROFILE%\.docker\machine\certs

C:\Users\tao.lin>docker images
REPOSITORY              TAG                IMAGE ID           CRE
ATED                    SIZE
springio/gs-spring-boot-docker  latest            385ecbb73f93       23
hours ago               669.1 MB
nginx                    latest            fd19524415dc       8 d
ays ago                 134.6 MB
mongo                    latest            598d0def97f1       3 w
```

So now the Docker client can connect to the Docker Daemon running on the Docker host.

We needed to set up three Environment Variables to get Docker client and Docker Daemon connected. In this section we set these Environment Variables in the command line session, but they will be gone after you close the command line.

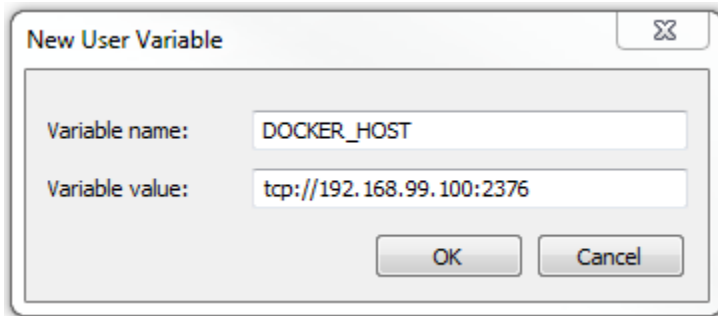
We need a persisted location to save these Environment Variables.

### 3.1.2 Set User Environment Variables for Docker

In order to make the Docker client work everywhere in Windows, we can set up three User Environment Variables:

## Set up, Configure, and Use Docker on Local Dev Machine

Note that the IP address in the screenshot below for the “DOCKER\_HOST” variable is the Docker host virtual machine IP address that you get by using the command: *docker-machine ls*

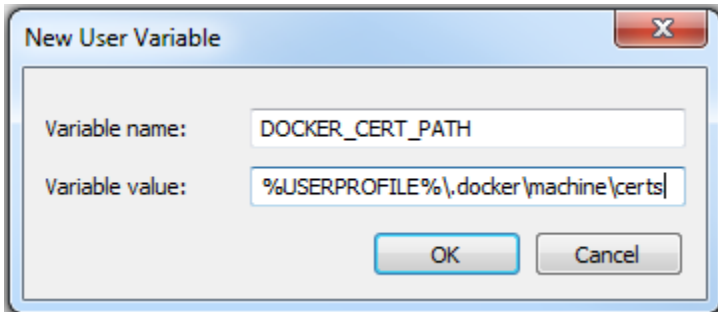


A screenshot of a 'New User Variable' dialog box. The title bar is light gray with a close button (X) on the right. The dialog has two input fields: 'Variable name:' with the text 'DOCKER\_HOST' and 'Variable value:' with the text 'tcp://192.168.99.100:2376'. At the bottom are 'OK' and 'Cancel' buttons.

Variable name: DOCKER\_HOST

Variable value: tcp://192.168.99.100:2376

OK Cancel

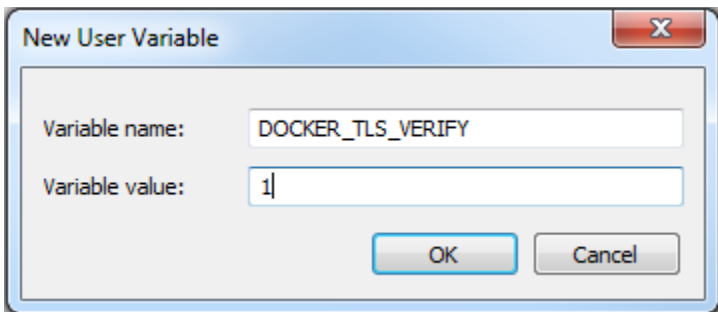


A screenshot of a 'New User Variable' dialog box. The title bar is light blue with a close button (X) on the right. The dialog has two input fields: 'Variable name:' with the text 'DOCKER\_CERT\_PATH' and 'Variable value:' with the text '%USERPROFILE%\docker\machine\certs'. At the bottom are 'OK' and 'Cancel' buttons.

Variable name: DOCKER\_CERT\_PATH

Variable value: %USERPROFILE%\docker\machine\certs

OK Cancel

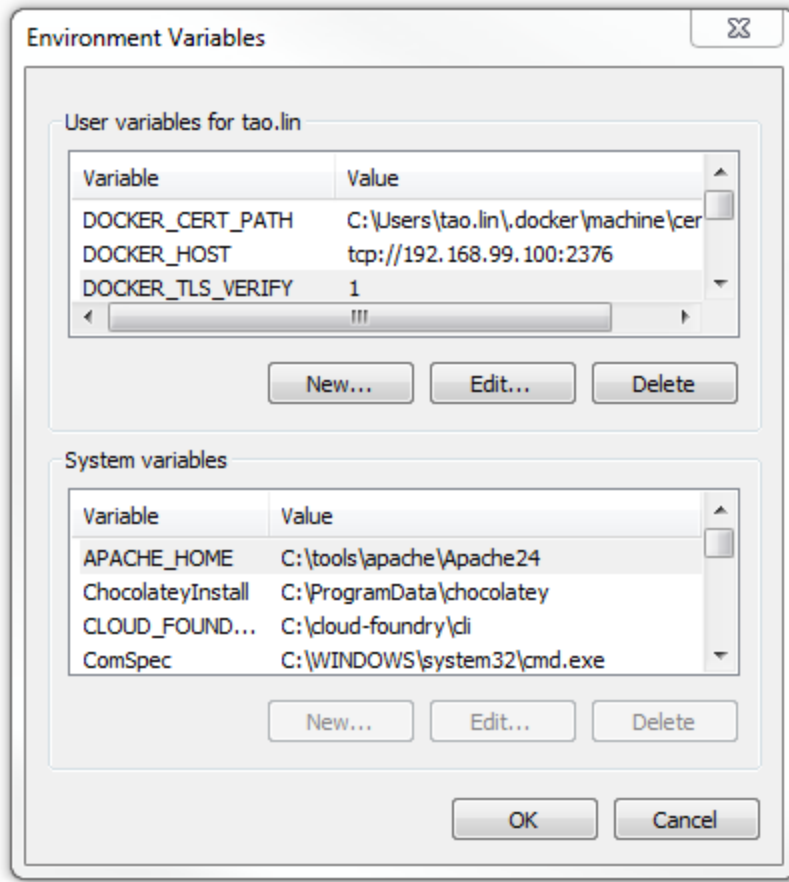


A screenshot of a 'New User Variable' dialog box. The title bar is light blue with a close button (X) on the right. The dialog has two input fields: 'Variable name:' with the text 'DOCKER\_TLS\_VERIFY' and 'Variable value:' with the text '1'. At the bottom are 'OK' and 'Cancel' buttons.

Variable name: DOCKER\_TLS\_VERIFY

Variable value: 1

OK Cancel



Click the OK button to save these new variables.

Open Windows Batch command line and issue the *docker images* command. As long as the Docker host is running, you should get a response from Docker Daemon.

### 3.2 Configure Docker on Mac OS X

#### 3.2.1 Make Docker Work in Terminal Command Line

If you don't want to know the details, you can go directly to Section 3.2.2 to set up the environment variables.

At the end of section 2.2.2, we know that there was an error when using Docker client in OSX Terminal command line: Docker client doesn't know where the Docker host is.

We need to set a "DOCKER\_HOST" environment variable. One way to do it is to use the following command in the command line to set it up for just this command line session:

```
export DOCKER_HOST=tcp://192.168.99.100:2376
```

The IP address is the Docker host virtual machine IP address that you get by using the following command: *docker-machine ls*

Try again to issue the *docker images* command to query Docker Daemon on Docker host. (Note: make sure the Docker host is running first.)

## Set up, Configure, and Use Docker on Local Dev Machine

```
taolin — -bash — 103x13
docker-machine version 0.6.0, build e27fb87
shildebrandlt1:~ taolin$ docker --version
Docker version 1.10.2, build c3959b1
shildebrandlt1:~ taolin$ docker-machine ls
NAME      ACTIVE  DRIVER      STATE     URL                  SWARM   DOCKER   ERRORS
default   -       virtualbox   Running    tcp://192.168.99.100:2376   v1.10.3
shildebrandlt1:~ taolin$ docker images
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
shildebrandlt1:~ taolin$ export DOCKER_HOST=tcp://192.168.99.100:2376
shildebrandlt1:~ taolin$ docker images
Get http://192.168.99.100:2376/v1.22/images/json: malformed HTTP response "\x15\x03\x01\x00\x02\x02".
* Are you trying to connect to a TLS-enabled daemon without TLS?
shildebrandlt1:~ taolin$
```

You will still get an error message, but now it is related to TLS. To fix it, you need to set up another environment variable:

```
export DOCKER_TLS_VERIFY=1
```

Now try the `docker images` command again. You will get another error message:

```
taolin — -bash — 109x13
NAME      ACTIVE  DRIVER      STATE     URL                  SWARM   DOCKER   ERRORS
default   -       virtualbox   Running    tcp://192.168.99.100:2376   v1.10.3
shildebrandlt1:~ taolin$ docker images
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
shildebrandlt1:~ taolin$ export DOCKER_HOST=tcp://192.168.99.100:2376
shildebrandlt1:~ taolin$ docker images
Get http://192.168.99.100:2376/v1.22/images/json: malformed HTTP response "\x15\x03\x01\x00\x02\x02".
* Are you trying to connect to a TLS-enabled daemon without TLS?
shildebrandlt1:~ taolin$ export DOCKER_TLS_VERIFY=1
shildebrandlt1:~ taolin$ docker images
Could not read CA certificate "/Users/taolin/.docker/ca.pem": open /Users/taolin/.docker/ca.pem: no such file or directory
shildebrandlt1:~ taolin$
```

It looks like we need to set up the certificate path. Docker Toolbox installation already provides one. Use the following command to set it up:

```
export DOCKER_CERT_PATH=$HOME/.docker/machine/certs
```

Now try running the `docker images` command yet again. You should see the following:

```
taolin — -bash — 113x20
Last login: Fri Mar 11 11:09:27 on ttys000
shildebrandlt1:~ taolin$ docker-machine ls
NAME      ACTIVE  DRIVER      STATE     URL                  SWARM   DOCKER   ERRORS
default   -       virtualbox   Running    tcp://192.168.99.100:2376   v1.10.3
shildebrandlt1:~ taolin$ docker images
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
shildebrandlt1:~ taolin$ export DOCKER_HOST=tcp://192.168.99.100:2376
shildebrandlt1:~ taolin$ docker images
Get http://192.168.99.100:2376/v1.22/images/json: malformed HTTP response "\x15\x03\x01\x00\x02\x02".
* Are you trying to connect to a TLS-enabled daemon without TLS?
shildebrandlt1:~ taolin$ export DOCKER_TLS_VERIFY=1
shildebrandlt1:~ taolin$ docker images
Could not read CA certificate "/Users/taolin/.docker/ca.pem": open /Users/taolin/.docker/ca.pem: no such file or directory
shildebrandlt1:~ taolin$ export DOCKER_CERT_PATH=$HOME/.docker/machine/certs
shildebrandlt1:~ taolin$ docker images
REPOSITORY          TAG             IMAGE ID          CREATED           SIZE
hello-world         latest          690ed74de00f     4 months ago     960 B
shildebrandlt1:~ taolin$
```

So now the Docker client can connect to the Docker Daemon running on the Docker host.

## Set up, Configure, and Use Docker on Local Dev Machine

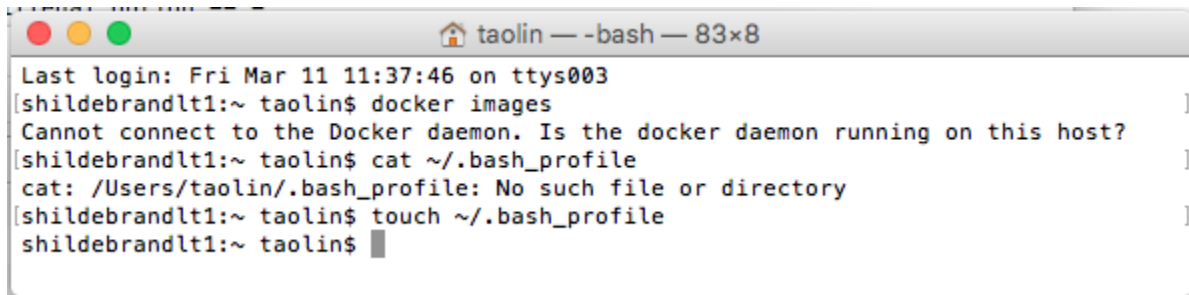
We needed to set up three Environment Variables to get Docker client and Docker Daemon connected. In this section we set these Environment Variables in the command line session, but they will be gone after you close the command line.

We need a persisted location to save these Environment Variables.

### 3.2.2 Set User Environment Variables for Docker

In order to make the Docker client work in Terminal in OS X, we can set up three User Environment Variables (note that the IP address for the “DOCKER\_HOST” variable is the Docker host virtual machine IP address that you get by using the command: *docker-machine ls*):

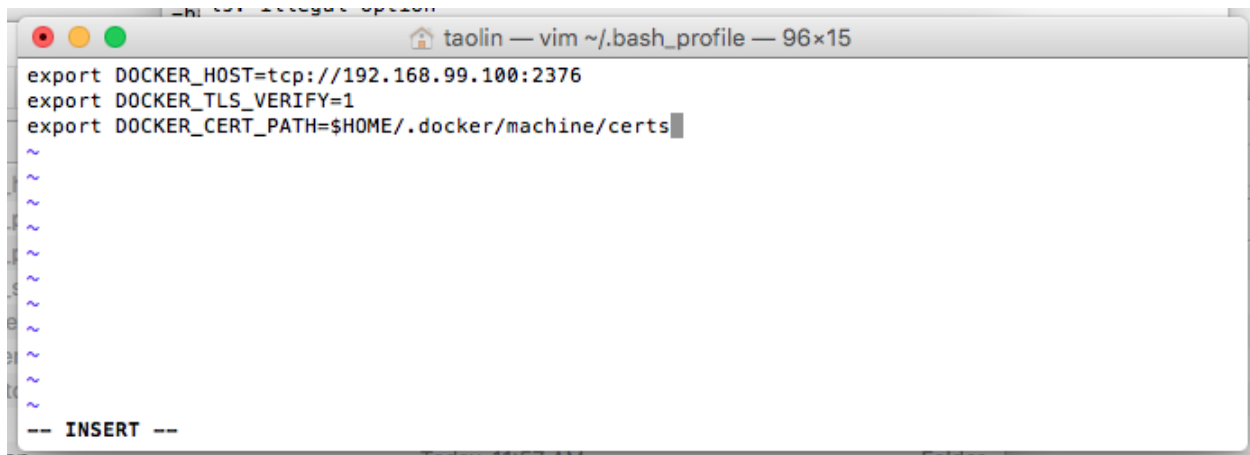
Open the `~/.bash_profile` file (or create a new one if it doesn’t already exist).



```
taolin — -bash — 83x8
Last login: Fri Mar 11 11:37:46 on ttys003
[shildebrandlt1:~ taolin$ docker images
Cannot connect to the Docker daemon. Is the docker daemon running on this host?
[shildebrandlt1:~ taolin$ cat ~/.bash_profile
cat: /Users/taolin/.bash_profile: No such file or directory
[shildebrandlt1:~ taolin$ touch ~/.bash_profile
shildebrandlt1:~ taolin$
```

Add the following lines to the end of the file:

```
export DOCKER_HOST=tcp://192.168.99.100:2376
export DOCKER_TLS_VERIFY=1
export DOCKER_CERT_PATH=$HOME/.docker/machine/certs
```



```
taolin — vim ~/.bash_profile — 96x15
export DOCKER_HOST=tcp://192.168.99.100:2376
export DOCKER_TLS_VERIFY=1
export DOCKER_CERT_PATH=$HOME/.docker/machine/certs
~
~
~
~
~
~
~
~
~
~
-- INSERT --
```

Save the `~/.bash_profile` file, then close the terminal window. Re-open a new terminal, and issue the command *docker images*; you will get response from Docker Daemon on local host.

## 4. Use Docker

### THIS SECTION IS STILL IN PROGRESS...

Docker Docs (<https://docs.docker.com/>) is good place to find current information related to Docker. To get started, you can follow Get Started with Docker for Windows (<https://docs.docker.com/windows/>) or Get Started with Mac OS X (<https://docs.docker.com/mac/>).

How to run docker host VM

How to create virtual machine using docker-machine command

```
$ docker-machine create --driver=virtualbox --virtualbox-memory 4096 default
```

Use different Docker host

Update hostfile on Windows/Mac OS X

The following are from MMR Project:

to build the MMR portal image

```
docker build -t mmr/mmr-portal:base . | tee docker-build.log
```

to run as daemon

```
docker run -d -p 9080:9080 -p 9443:9443 --name mmr -it mmr/mmr-portal:base
```

to run the container from this image

```
docker run -it -p 9080:9080 -p 9443:9443 --name mmr -i -t mmr/mmr-portal:base /bin/bash
```

to access shell for the daemon

```
docker exec -it mmr /bin/bash #to exit => exit
```

to attach to the container

```
docker attach mmr #to detach press Ctrl+p Ctrl+q #to exit and stop the container press Ctrl +c
```

### CONTAINERS

to check running containers

```
docker ps
```

to check all containers

```
docker ps -a
```

to remove a container

```
docker rm container_name
```

delete all containers

```
docker rm $(docker ps -a -q)
```



## Set up, Configure, and Use Docker on Local Dev Machine

to commit an image from container

```
docker commit johndondapti/mmr container_name
```

IMAGES

to pull the latest MMR image

```
docker pull mmr/mmr-portal
```

to tag an image

```
docker tag mmr/mmr-portal:tag_name image_name
```

to push the image to repository

```
docker push mmr/mmr-portal
```

to list images

```
docker images
```

to list all images

```
docker images -a
```

to remove an image

```
docker rmi image_name
```

delete all images

```
docker rmi $(docker images -q)
```

## 5. References

### 5.1 Installation on Windows

[https://docs.docker.com/windows/step\\_one/](https://docs.docker.com/windows/step_one/)  
<https://docs.docker.com/engine/installation/windows/>

### 5.2 Installation on Mac OS X

[https://docs.docker.com/mac/step\\_one/](https://docs.docker.com/mac/step_one/)  
<https://docs.docker.com/engine/installation/mac/>

### 5.3 Get Started with Docker for Windows

<https://docs.docker.com/windows/>

### 5.4 Get Started with Mac OS X

<https://docs.docker.com/mac/>

### 5.5 Docker Docs

<https://docs.docker.com/>

### 5.6 Docker Tutorials

<https://www.docker.com/products/docker-toolbox#/tutorials>

### 5.7 Docker Cheat Sheet

<https://github.com/wsargent/docker-cheat-sheet>

### 5.8 Spring Boot with Docker

<https://spring.io/guides/gs/spring-boot-docker/>

### 5.9 Building Microservices, part 4. Dockerize your Microservices

<http://callistaenterprise.se/blogg/teknik/2015/06/08/building-microservices-part-4-dockerize-your-microservices/>

### 5.10 Docker and Containers: The Big Picture

<https://app.pluralsight.com/library/courses/docker-containers-big-picture/table-of-contents>

### 5.11 Docker Deep Dive

<https://app.pluralsight.com/library/courses/docker-deep-dive/table-of-contents>