

Notes on Importing and Running the VotingMachine

Can't Run Directly on Windows or MacOS

Docker uses special Linux operating system calls, and a very interesting sort of file system, to create a special runtime environment for Docker applications.

Because of this, running Docker on Windows or MacOS requires using virtual machine technology. The cheapest—free and open-source [VirtualBox](#)—is also extremely useful and reliable. VMware and Parallels are excellent commercial virtualization packages, and Windows Hyper-V can also work.

Docker Desktop may do this all for you

Docker Desktop doesn't do an awful lot for you, but it may make unnecessary some of the following steps:

Creating a docker machine

I use this command to create a new VirtualBox machine for running Docker. (For Hyper-V or other virtualization systems, the options will be different.)

```
docker-machine create --driver virtualbox --virtualbox-no-vtx-check default
```

This creates a VM called `default` which is what is used by the `docker-machine` utility and the `docker` commands use that. You can create multiple Docker machines, just use different names.

Configuring to run Docker against the docker machine

The new docker machine is dormant. To start it, use

```
docker-machine start default
```

When it completes, why does it give you a message like this?

Detecting the provisioner...

Started machines may have new IP addresses. You may need to re-run the `docker-machine env` command.

When the docker machine starts, it does not advertise itself or provide any kind of registration. Because of this, when you issue a regular `docker` command, that command will not know how to work with the docker machine. It needs an *environment*. So what is this `docker-machine env default`? It tells you the settings needed for `docker` to talk to the `default` machine after it is started. It does not actually *set* those settings, it tells you what they are. It also tells you *how* to set them.

docker-machine env default on windows

This is what the output looks like on Windows.

```
1 SET DOCKER_TLS_VERIFY=1
2 SET DOCKER_HOST=tcp://192.168.99.100:2376
3 SET DOCKER_CERT_PATH=C:\Users\adwolfe\.docker\machine\machines\default
4 SET DOCKER_MACHINE_NAME=default
5 REM Run this command to configure your shell:
6 REM @FOR /f "tokens=*" %i IN ('docker-machine env default') DO @%i
```

docker-machine env on MacOS

Here is the output on MacOS:

```
1 export DOCKER_TLS_VERIFY="1"
2 export DOCKER_HOST="tcp://192.168.99.100:2376"
3 export
  DOCKER_CERT_PATH="/Users/adwolfe/.docker/machine/machines/default"
4 export DOCKER_MACHINE_NAME="default"
5 # Run this command to configure your shell:
6 # eval $(docker-machine env default)
```

What Setting the Environment does:

...on MacOS, you check whether you're up using `docker ps -a`. Before running that `eval` command, you get an error. Afterwards, you're good:

```
/Users/adwolfe/Workshop/GHClassRm/Voting-Machine [master]
adwolfe_502: docker ps -a
Cannot connect to the Docker daemon at unix:///var/run/docker.sock. Is the docker daemon running?
/Users/adwolfe/Workshop/GHClassRm/Voting-Machine [master]
adwolfe_503: eval $(docker-machine env default)
/Users/adwolfe/Workshop/GHClassRm/Voting-Machine [master]
adwolfe_504: docker ps -a
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS	PORTS	NAMES
7caf20114df2	f701fe93bf24	"/usr/local/tomcat/b..."	12 days ago	Exited (143) 12 days ago		ja13
da0f5999aec7	jenkins/jenkins	"/sbin/tini -- /usr/..."	3 weeks ago	Exited (137) 3 weeks ago		jenkins
948ac55445fa	tomcat	"catalina.sh run"	3 weeks ago	Up 26 minutes	0.0.0.0:8009->8009/tcp, 0.0.0.0:8080->8080/tcp	cat1

```
/Users/adwolfe/Workshop/GHClassRm/Voting-Machine [master]
adwolfe_505:
```

Similarly, on Windows:

```
C:\Users\andre\Workshop\GHClassroom\2020-01 Spring\Voting-Machine>docker ps -a
error during connect: Get https://192.168.99.100:2376/v1.40/containers/json?all=1: dial tcp 192.168.99.100:2376: connectex: A connection attempt failed because the connect
ed party did not properly respond after a period of time, or established connection failed because connected host has failed to respond.

C:\Users\andre\Workshop\GHClassroom\2020-01 Spring\Voting-Machine> @FOR /f "tokens==" %i IN ('docker-machine env default') DO @%i

C:\Users\andre\Workshop\GHClassroom\2020-01 Spring\Voting-Machine>docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
29d68c32f8e7        5ddfbb             "/usr/local/tomcat/b..."  12 days ago        Exited (255) 17 minutes ago    0.0.0.0:8009->8009/tcp, 0.0.0.0:8080->8080/tcp    cat1

C:\Users\andre\Workshop\GHClassroom\2020-01 Spring\Voting-Machine>
```

Preparing to Import

First, start the docker machine if needed:

```
docker-machine start default
```

Open Ports on the Virtual Machine (Possibly not needed)

need to open 8080 and 8009 on virtualbox. Open the VirtualBox Manager and select the Settings on the `default` virtual machine.

Switch to Network and open the first network interface ("NAT")

Click on "port forwarding"

Click the "plus" icon at upper right and enter name tomcat-main, port 8080 to port 8080

Click the "plus" icon again and enter name tomcat-control, port 8009 to port 8009

Where's the Container file?

The container is on the class google drive "Student resources". It's called `cat1-Phase0-container.tar`.

Import the container

You can pick up the container from wherever you downloaded it. Be sure to give any necessary path before the name.

```
docker import /Users/adwolfe/Downloads/cat1-Phase0-container.tar
```

If you attempt the import on Windows using a Bash shell rather than CMD, a path to the file will not work. It may work if you have the tar file in the current directory, but it's most reliable to use a CMD prompt for the import.

This import will succeed, blurring out a long hashcode ID. This ID is an (unnamed) image. Find it with

```
docker images
```

```

1 $ docker images
2 REPOSITORY          TAG                 IMAGE ID           CREATED
3 <none>               <none>             5ddfbb25089c      20 minutes
4 ago                 1GB
5 hello-world          latest             fce289e99eb9      12 months
ago                 1.84kB

```

Here the ID is `5ddfbb25089c`. Get your hash/ID and use it to start the image:

```

docker run --name cat1 -d -p 8080:8080 -p 8009:8009 5ddfbb
/usr/local/tomcat/bin/catalina.sh run

```

Here I am using the name `cat1` again, but that doesn't matter. Note in the middle of the command is the hashcode. Docker will allow you to give just a hashcode prefix rather than typing out the full ID.

Check the Container

Issuing the command `docker ps` should show your new "cat1" container running. Here is a Windows screen showing the `docker run` and then running `docker ps` twice.

```

MINGW64/c:/Program Files/Docker Toolbox
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
3d0b0c3b4b30      5ddfbb25089c      "/usr/local/tomcat/b..." 7 minutes ago      Exited (0) 13 seconds ago      0.0.0.0:8080->8080/tcp, 0.0.0.0:8009->8009/tcp      cat1

andre@Fimbrethil MINGW64 /c:/Program Files/Docker Toolbox
$ docker rm cat1
cat1

andre@Fimbrethil MINGW64 /c:/Program Files/Docker Toolbox
$ docker run --name cat1 -d -p 8080:8080 -p 8009:8009 5ddfbb /usr/local/tomcat/bin/catalina.sh run
29d68c32f8e7bc3f4a752bbb614bc3dec3e58716
2b953b0f6d10283609647834

s/Docker Toolbox
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
29d68c32f8e7      5ddfbb             "/usr/local/tomcat/b..." 5 seconds ago      Up 4 seconds        0.0.0.0:8080->8080/tcp, 0.0.0.0:8009->8009/tcp      cat1

andre@Fimbrethil MINGW64 /c:/Program Files/Docker Toolbox
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS              NAMES
29d68c32f8e7      5ddfbb             "/usr/local/tomcat/b..." 12 seconds ago      Up 11 seconds        0.0.0.0:8080->8080/tcp, 0.0.0.0:8009->8009/tcp      cat1

andre@Fimbrethil MINGW64 /c:/Program Files/Docker Toolbox
$ start http://localhost:8080
/usr/bin/start: line 8: cmd: command not found

andre@Fimbrethil MINGW64 /c:/Program Files/Docker Toolbox
$ open
bash: open: command not found

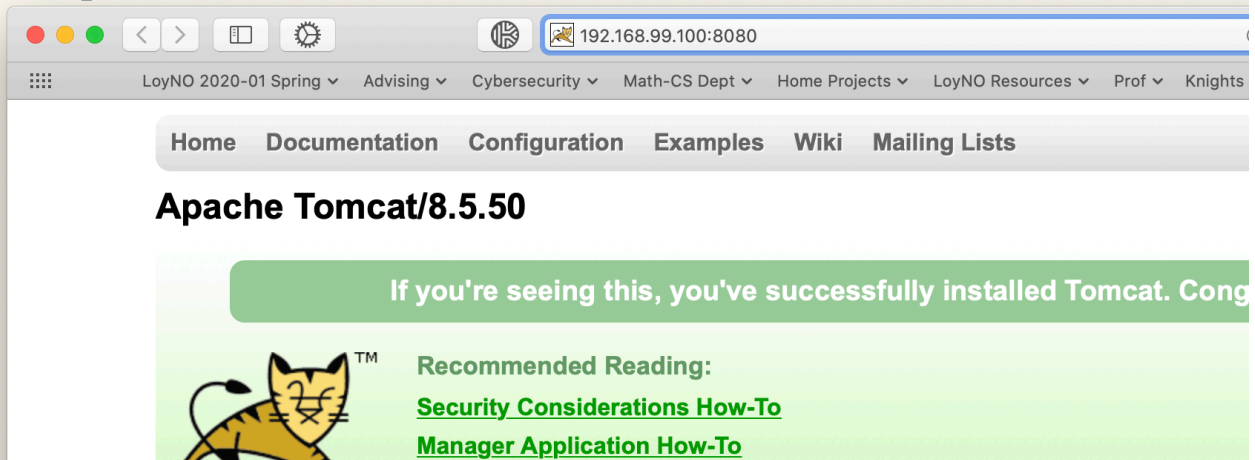
andre@Fimbrethil MINGW64 /c:/Program Files/Docker Toolbox
$

```

Test the Container in Your Browser

Get the docker machine IP address with `docker-machine ip default`. Then open that IP to port 8080. For example:

```
/Users/adwolfe/Workshop/GHClassRm/Voting-Machine [master]
adwolfe_517: docker-machine ip default
192.168.99.100
/Users/adwolfe/Workshop/GHClassRm/Voting-Machine [master]
adwolfe_518: 
```



If you have mapped ports, you can open your browser and browse to <http://localhost:8080>. You should see a "Tomcat" home page as above. If you like, you can go to the "Manager." The username is `cosca451` and the password is the same `cosca451`.

It then "List Applications," and click on "Voting." You'll see that it's really lame, plus some of the links don't work.

In case you're interested—how I Exported the "Phase0" Container... 1 gb

My container is named `cat1`; the name is immaterial. Its ID is `948ac554455fa`. I did an export from the container...

```
docker export 948ac554455fa -o cat1-Phase0-container.tar
```