

## SComputer Engineering Department

### CMPE283-02 – Virtualization Technologies

Student Name: **Piyush Bansal**

Instructor: **Simon Shim**

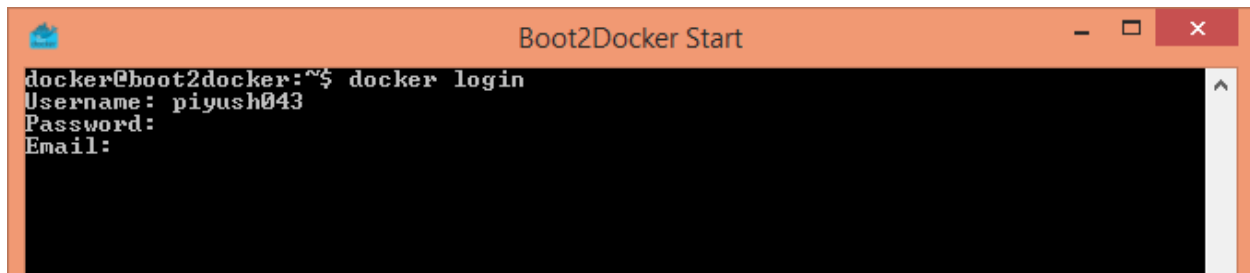
Semester: **Spring, 2015**

---

#### Lab #1– Cross Host Linking Docker Containers

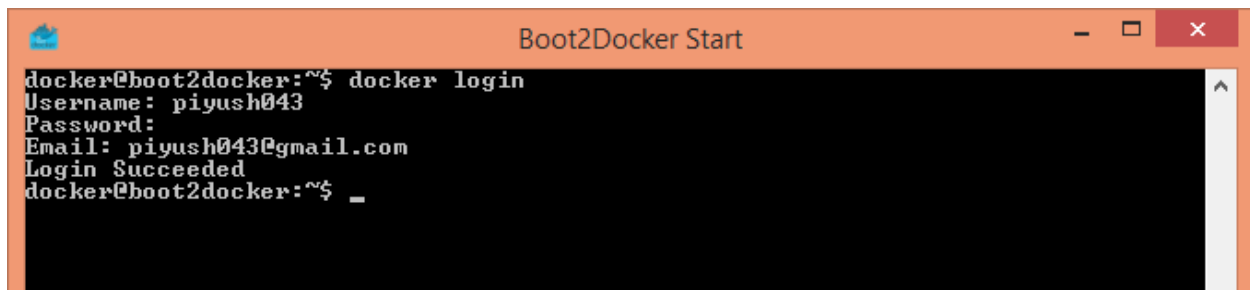
---

1. Login to docker using below command and enter credentials.



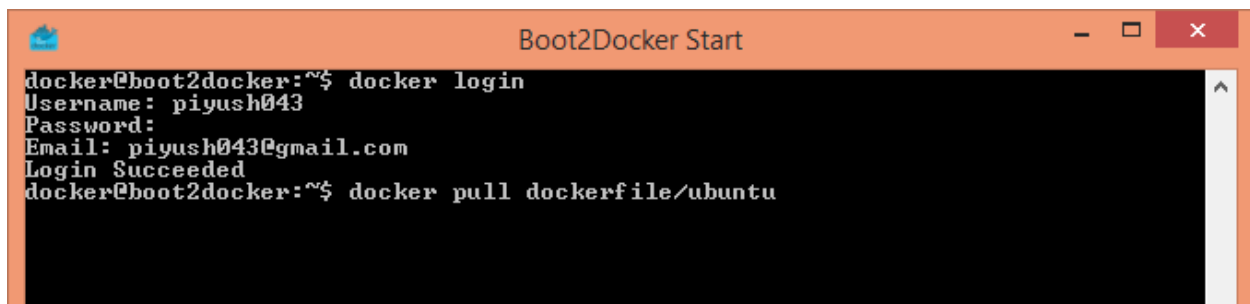
```
Boot2Docker Start
docker@boot2docker:~$ docker login
Username: piyush043
Password:
Email:
```

2. You will see “login Succeeded” message after successful login.



```
Boot2Docker Start
docker@boot2docker:~$ docker login
Username: piyush043
Password:
Email: piyush043@gmail.com
Login Succeeded
docker@boot2docker:~$ _
```

3. Now pull latest Ubuntu image from docker hub.



```
Boot2Docker Start
docker@boot2docker:~$ docker login
Username: piyush043
Password:
Email: piyush043@gmail.com
Login Succeeded
docker@boot2docker:~$ docker pull dockerfile/ubuntu
```

4. Images will get downloaded in layers.

```
Boot2Docker Start

docker@boot2docker:~$ docker pull dockerfile/ubuntu
Pulling repository dockerfile/ubuntu
e7fc98dc1b07: Pulling image (latest) from dockerfile/ubuntu, endpoint: https://r
e7fc98dc1b07: Pulling dependent layers
511136ea3c5a: Download complete
27d47432a69b: Download complete
5f92234dcf1e: Download complete
51a9c7c1f8bb: Download complete
5ba9dab47459: Download complete
5ac964b38b8b: Downloading 12.96 MB/85.98 MB 8m27s
```

5. Once image gets downloaded run `docker images` command to see list of images present.

```
Boot2Docker Start

docker@boot2docker:~$ docker pull dockerfile/ubuntu
Pulling repository dockerfile/ubuntu
e7fc98dc1b07: Pulling image (latest) from dockerfile/ubuntu, endpoint: https://r
e7fc98dc1b07: Download complete
511136ea3c5a: Download complete
27d47432a69b: Download complete
5f92234dcf1e: Download complete
51a9c7c1f8bb: Download complete
5ba9dab47459: Download complete
5ac964b38b8b: Download complete
8a75beb8c617: Download complete
c240f95f9ab0: Download complete
ed8bb587e39a: Download complete
69151129842b: Download complete
f89142985c75: Download complete
Status: Downloaded newer image for dockerfile/ubuntu:latest
docker@boot2docker:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
dockerfile/ubuntu   latest             e7fc98dc1b07       2 weeks ago
410.3 MB
docker@boot2docker:~$
```

6. Now run that image into a container using below command and once you are into your container run `sudo apt-get update`.

```
Boot2Docker Start

docker@boot2docker:~$ docker run -it --name="nodejs" dockerfile/ubuntu /bin/bash
[ root@e9f1cc569b5c:~ ]$ ls
[ root@e9f1cc569b5c:~ ]$ sudo apt-get update
Ign http://archive.ubuntu.com trusty InRelease
Ign http://archive.ubuntu.com trusty-updates InRelease
Ign http://archive.ubuntu.com trusty-security InRelease
Get:1 http://archive.ubuntu.com trusty Release.gpg [933 B]
Get:2 http://archive.ubuntu.com trusty-updates Release.gpg [933 B]
Get:3 http://archive.ubuntu.com trusty-security Release.gpg [933 B]
Get:4 http://archive.ubuntu.com trusty Release [58.5 kB]
Get:5 http://archive.ubuntu.com trusty-updates Release [62.0 kB]
Get:6 http://archive.ubuntu.com trusty-security Release [62.0 kB]
Get:7 http://archive.ubuntu.com trusty/main Sources [1335 kB]
29% [7 Sources 248 kB/1335 kB 19%]
```

7. Now install nodejs in that container.

```
Boot2Docker Start

Reading package lists... Done
l root@e9f1cc569b5c:~# sudo apt-get install nodejs
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libc-ares2 libv8-3.14.5
The following NEW packages will be installed:
  libc-ares2 libv8-3.14.5 nodejs
0 upgraded, 3 newly installed, 0 to remove and 14 not upgraded.
Need to get 1912 kB of archives.
After this operation, 7538 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

8. Once nodejs installation gets completed, exit from container and then commit it.

```
Boot2Docker Start

docker@boot2docker:~$ docker commit -a piyush nodejs piyush043/cmpe283:nodejs
77daab526d3352afaf2bcd000f7dff7f38b7c57ec7d1b871f3e630e4cb55bac
docker@boot2docker:~$ _
```

9. Push that container to repository

```
Boot2Docker Start

docker@boot2docker:~$ docker push piyush043/cmpe283
The push refers to a repository [piyush043/cmpe283] (len: 1)
Sending image list

Please login prior to push:
Username: piyush043
Password:
Email: piyush043@gmail.com
Login Succeeded
The push refers to a repository [piyush043/cmpe283] (len: 1)
Sending image list
Pushing repository piyush043/cmpe283 (1 tags)
511136ea3c5a: Image already pushed, skipping
27d47432a69b: Image already pushed, skipping
5f92234dcf1e: Image already pushed, skipping
51a9c7c1f8bb: Image already pushed, skipping
5ba9dab47459: Image already pushed, skipping
5ac964b38b8b: Image already pushed, skipping
8a75beb8c617: Image already pushed, skipping
c240f95f9ab0: Image already pushed, skipping
ed8bb587e39a: Image already pushed, skipping
69151129842b: Image already pushed, skipping
f89142985c75: Image already pushed, skipping
e7fc98dc1b07: Image already pushed, skipping
77daab526d33: Image successfully pushed
```

10. Create another container for mongoDB and update it

```
Boot2Docker Start

docker@boot2docker:~$ docker run -it --name="mongo" dockerfile/ubuntu /bin/bash
l root@296b74064eb4:~#
```

```
Boot2Docker Start

docker@boot2docker:~$ docker run -it --name="mongo" dockerfile/ubuntu /bin/bash
I root@296b74064eb4:~# $ sudo apt-get update
Ign http://archive.ubuntu.com trusty InRelease
Ign http://archive.ubuntu.com trusty-updates InRelease
Ign http://archive.ubuntu.com trusty-security InRelease
Get:1 http://archive.ubuntu.com trusty Release.gpg [933 B]
Get:2 http://archive.ubuntu.com trusty-updates Release.gpg [933 B]
Get:3 http://archive.ubuntu.com trusty-security Release.gpg [933 B]
Get:4 http://archive.ubuntu.com trusty Release [58.5 kB]
Get:5 http://archive.ubuntu.com trusty-updates Release [62.0 kB]
Get:6 http://archive.ubuntu.com trusty-security Release [62.0 kB]
Get:7 http://archive.ubuntu.com trusty/main Sources [1335 kB]
20% [7 Sources 122 kB/1335 kB 9%]
```

11. Install mongod in that container.

```
Boot2Docker Start

I root@296b74064eb4:~# $ sudo apt-get install mongod
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libboost-dev libboost-fsystem1.54.0 libboost-program-options1.54.0
  libboost-system1.54.0 libboost-thread1.54.0 libboost1.54-dev
  libgoogle-perftools4 libpcap0.8 libpcrcpp0 libsnappy1 libtcmalloc-minimal4
  libunwind8 libv8-3.14.5 mongodb-clients mongodb-dev mongodb-server
Suggested packages:
  libboost-doc libboost1.54-doc libboost-atomic1.54-dev
  libboost-chrono1.54-dev libboost-context1.54-dev libboost-coroutine.54-dev
  libboost-date-time1.54-dev libboost-exception1.54-dev
  libboost-fsystem1.54-dev libboost-graph1.54-dev
  libboost-graph-parallel1.54-dev libboost-iostreams1.54-dev
  libboost-locale1.54-dev libboost-log.54-dev libboost-math1.54-dev
  libboost-mpi1.54-dev libboost-mpi-python1.54-dev
  libboost-program-options1.54-dev libboost-python1.54-dev
  libboost-random1.54-dev libboost-regex1.54-dev
  libboost-serialization1.54-dev libboost-signals1.54-dev
  libboost-system1.54-dev libboost-test1.54-dev libboost-thread1.54-dev
  libboost-timer1.54-dev libboost-wave1.54-dev libboost1.54-tools-dev
  libmpfrc++-dev libntl-dev
The following NEW packages will be installed:
  libboost-dev libboost-fsystem1.54.0 libboost-program-options1.54.0
```

12. Close that container and commit it.

```
Boot2Docker Start

docker@boot2docker:~$ docker ps -a
CONTAINER ID   IMAGE                PORTS          COMMAND          NAMES          CREATED
296b74064eb4   dockerfile/ubuntu:latest  "/bin/bash"    mongo           5 minutes ago
c9f1cc569b5c   dockerfile/ubuntu:latest  "/bin/bash"    mongo           21 minutes ago
o             Exited (0) 17 minutes ago  nodejs
docker@boot2docker:~$ docker commit -a piyush mongo piyush043/cmpe283:mongo
d7411b5e171c04fcb3e0e02a1b6de19e3ace30d61a423a1ba7918b97c4a65f45
docker@boot2docker:~$
```

13. Run docker images command to view list of images.

```
Boot2Docker Start
docker@boot2docker:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
piyush043/cmpe283    mongo              d7411b5e171c       About a minute ago
647.8 MB
piyush043/cmpe283    nodejs            77daab526d33       13 minutes ago
440.2 MB
dockerfile/ubuntu    latest            e7fc98dc1b07       2 weeks ago
410.3 MB
docker@boot2docker:~$ _
```

14. Push that image to docker hub repository.

```
Boot2Docker Start
docker@boot2docker:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
piyush043/cmpe283    mongo              d7411b5e171c       About a minute ago
647.8 MB
piyush043/cmpe283    nodejs            77daab526d33       13 minutes ago
440.2 MB
dockerfile/ubuntu    latest            e7fc98dc1b07       2 weeks ago
410.3 MB
docker@boot2docker:~$ docker push piyush043/cmpe283:mongo
The push refers to a repository [piyush043/cmpe283] (len: 1)
Sending image list
Pushing repository piyush043/cmpe283 (1 tags)
511136ea3c5a: Image already pushed, skipping
27d47432a69b: Image already pushed, skipping
5f92234dcf1e: Image already pushed, skipping
51a9c7c1f8bb: Image already pushed, skipping
5ba9dab47459: Image already pushed, skipping
5ac964b38b8b: Image already pushed, skipping
8a75beb8c617: Image already pushed, skipping
c240f95f9ab0: Image already pushed, skipping
ed8bb587e39a: Pushing
```

15. Now run mongodb container and open its port for external entry. Here I am opening 8888 port.

```
Boot2Docker Start
docker@boot2docker:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
piyush043/cmpe283    mongo              d7411b5e171c       19 minutes ago
647.8 MB
piyush043/cmpe283    nodejs            77daab526d33       31 minutes ago
440.2 MB
dockerfile/ubuntu    latest            e7fc98dc1b07       2 weeks ago
410.3 MB
docker@boot2docker:~$ docker run -it -p 8888:27017 piyush043/cmpe283:mongo
l root@28d27bbbfb7e:~#
```

16. Create a data directory in mongodb container. In this directory database will be stored.

```
Boot2Docker Start

docker@boot2docker:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
VIRTUAL SIZE
piyush043/cmpe283    mongo              d7411b5e171c       19 minutes ago
647.8 MB
piyush043/cmpe283    nodejs             77daab526d33       31 minutes ago
440.2 MB
dockerfile/ubuntu    latest             e7fc98dc1b07       2 weeks ago
410.3 MB
docker@boot2docker:~$ docker run -it -p 8888:27017 piyush043/cmpe283:mongo
[ root@28d27bbbfb7e:~ ]$ mongo
MongoDB shell version: 2.4.9
connecting to: test
Mon Feb 16 02:21:09.532 Error: couldn't connect to server 127.0.0.1:27017 at src
/mongo/shell/mongo.js:145
exception: connect failed
[ root@28d27bbbfb7e:~ ]$ ls
[ root@28d27bbbfb7e:~ ]$ ls -a
./  ../ .bashrc .gitconfig .profile .scripts/
[ root@28d27bbbfb7e:~ ]$ cd data
bash: cd: data: No such file or directory
[ root@28d27bbbfb7e:~ ]$ mkdir data_
```

```
Boot2Docker Start

[ root@28d27bbbfb7e:~ ]$ mongo
MongoDB shell version: 2.4.9
connecting to: test
Mon Feb 16 02:21:09.532 Error: couldn't connect to server 127.0.0.1:27017 at src
/mongo/shell/mongo.js:145
exception: connect failed
[ root@28d27bbbfb7e:~ ]$ ls
[ root@28d27bbbfb7e:~ ]$ ls -a
./  ../ .bashrc .gitconfig .profile .scripts/
[ root@28d27bbbfb7e:~ ]$ cd data
bash: cd: data: No such file or directory
[ root@28d27bbbfb7e:~ ]$ mkdir data
[ root@28d27bbbfb7e:~ ]$ cd data
[ root@28d27bbbfb7e:~/data ]$ mkdir dbpath
[ root@28d27bbbfb7e:~/data ]$ mkdir db
bash: mkdir: command not found
[ root@28d27bbbfb7e:~/data ]$ mkdir db
[ root@28d27bbbfb7e:~/data ]$ ls
db/ dbpath/
[ root@28d27bbbfb7e:~/data ]$ ls -a
./  ../ db/ dbpath/
[ root@28d27bbbfb7e:~/data ]$ _
```

17. Now start the mongod server using `mongod -dbpath="data" --smallfiles` command.

```
Boot2Docker Start

use --help for help
l root@28d27bbbfb7e:~/data $ mongod --dbpath=db --smallfiles
Mon Feb 16 02:24:15.287 [initandlisten] MongoDB starting : pid=115 port=27017 db
path=db 64-bit host=28d27bbbfb7e
Mon Feb 16 02:24:15.288 [initandlisten] db version v2.4.9
Mon Feb 16 02:24:15.288 [initandlisten] git version: nogitversion
Mon Feb 16 02:24:15.288 [initandlisten] build info: Linux orlo 3.2.0-58-generic
#88-Ubuntu SMP Tue Dec 3 17:37:58 UTC 2013 x86_64 BOOST_LIB_VERSION=1_54
Mon Feb 16 02:24:15.288 [initandlisten] allocator: tcmalloc
Mon Feb 16 02:24:15.288 [initandlisten] options: { dbpath: "db", smallfiles: true
e }
Mon Feb 16 02:24:15.436 [initandlisten] journal dir=db/journal
Mon Feb 16 02:24:15.437 [initandlisten] recover : no journal files present, no r
ecovery needed
Mon Feb 16 02:24:16.037 [initandlisten] preallocateIsFaster=true 7.22
Mon Feb 16 02:24:16.269 [initandlisten] preallocateIsFaster=true 3.5
Mon Feb 16 02:24:17.490 [initandlisten] preallocateIsFaster=true 2.84
Mon Feb 16 02:24:17.491 [initandlisten] preallocating a journal file db/journal/
prealloc.0
```

18. Now run nodejs container and open its port for external entry. Here I am opening 7777 port.

```
Boot2Docker Start

docker@boot2docker:~$ docker run -it -p 7777:3000 --name=nodejs piyush043/cmpe28
3:nodejs
l root@2a9951da8b25:~ $ sudo apt-get install npm
Reading package lists... Done
Building dependency tree
Reading state information... Done
-
```

19. Install npm and start

```
Boot2Docker Start

Setting up node-read <1.0.4-1> ...
Setting up node-read-package-json <1.1.3-1> ...
Setting up node-retry <0.6.0-1> ...
Setting up node-sha <1.2.3-1> ...
Setting up node-slide <1.1.4-1> ...
Setting up npm <1.3.10~dfsg-1> ...
l root@2a9951da8b25:~ $ npm install -g express
npm http GET https://registry.npmjs.org/express
npm http 200 https://registry.npmjs.org/express
npm http GET https://registry.npmjs.org/express/-/express-4.11.2.tgz
npm http 200 https://registry.npmjs.org/express/-/express-4.11.2.tgz
npm http GET https://registry.npmjs.org/accepts
npm http GET https://registry.npmjs.org/content-disposition/0.5.0
npm http GET https://registry.npmjs.org/debug
npm http GET https://registry.npmjs.org/cookie-signature/1.0.5
npm http GET https://registry.npmjs.org/depd
npm http GET https://registry.npmjs.org/escape-html/1.0.1
npm http GET https://registry.npmjs.org/etag
npm http GET https://registry.npmjs.org/finalhandler/0.3.3
npm http GET https://registry.npmjs.org/fresh/0.2.4
npm http GET https://registry.npmjs.org/media-typer/0.3.0
npm http GET https://registry.npmjs.org/methods
npm http GET https://registry.npmjs.org/on-finished
npm http GET https://registry.npmjs.org/parseurl
npm http GET https://registry.npmjs.org/path-to-regexp/0.1.3
```

```
Boot2Docker Start

[ root@2a9951da8b25:~/nodeapp/nodeapp2 {master *} ]$ npm start

> nodeapp2@0.0.0 start /root/nodeapp/nodeapp2
> node ./bin/www

[Error: Cannot find module '../build/Release/bson'] code: 'MODULE_NOT_FOUND' }
js-bson: Failed to load c++ bson extension, using pure JS version
```

Express

192.168.59.103:7777

port 3000 already in use

Express

Welcome to our test

User Info

Name:  
Age:  
Gender:  
Location:

User List

UserName	Email	Delete?
<a href="#">user1</a>	user1@test.com	<a href="#">delete</a>
<a href="#">user2</a>	user2@test.com	<a href="#">delete</a>
<a href="#">user3</a>	user3@test.com	<a href="#">delete</a>
<a href="#">user4</a>	user4@test.com	<a href="#">delete</a>
<a href="#">user5</a>	user5@test.com	<a href="#">delete</a>

Add User

Username	Email
Full Name	Age
Location	gender

Add User



