

Amir Mansha

CYSE 211 – DL1

Docker Lab

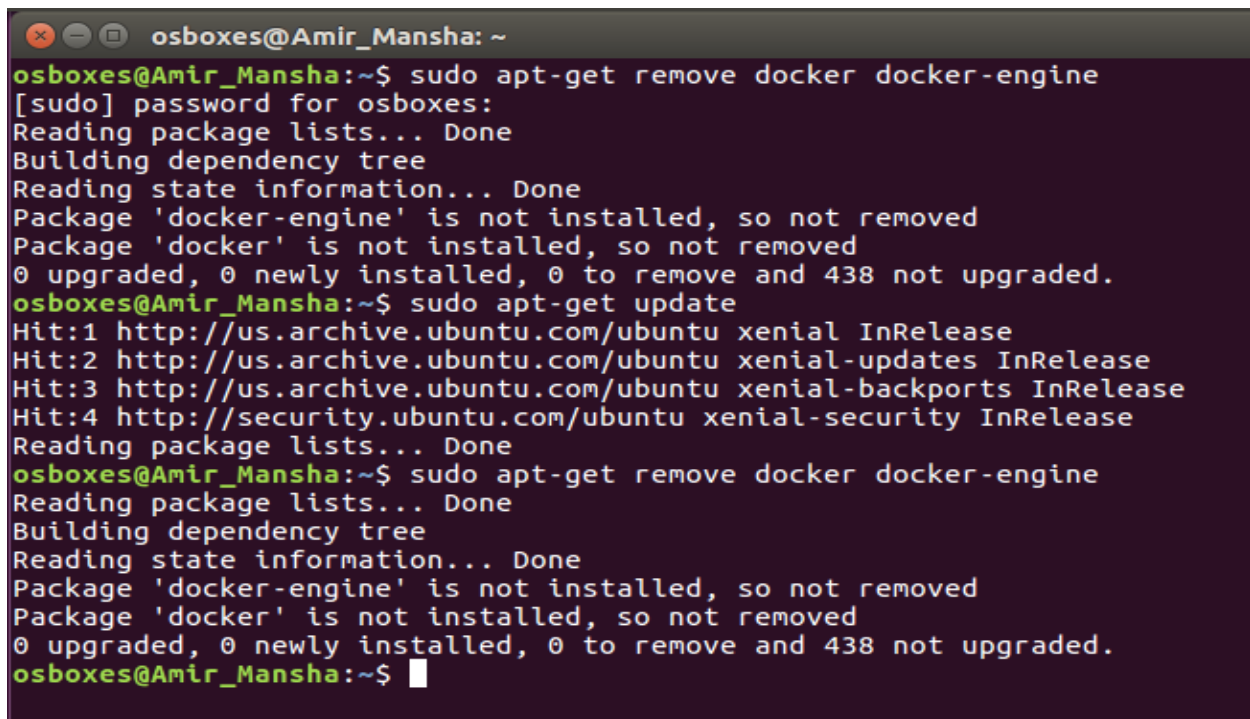
INTRO:

Docker is a tool that is efficient in handling isolated applications and the application runs in the same environment meaning the application can work in every computer if that is the situation. It is good for security because it keeps your projects separate.

A container in the docker is where codes and the environment are in a “container.” In a virtual machine, it has its own kernel and uses its kernel. But docker container uses the host computer kernel. An “image” contains the OS, software, apps etc. which runs in the container.

Task 1 – set up

In this task we are setting up and installing docker.

A terminal window with a dark background and light-colored text. The window title is 'osboxes@Amir_Mansha: ~'. The user enters the command 'sudo apt-get remove docker docker-engine'. The terminal shows the password prompt, package lists being read, and a dependency tree being built. It reports that 'docker-engine' and 'docker' are not installed. Then, the user enters 'sudo apt-get update'. The terminal shows four hits for Ubuntu repositories. Finally, the user enters 'sudo apt-get remove docker docker-engine' again, and the terminal shows the same results as before, indicating no changes were made.

```
osboxes@Amir_Mansha: ~  
osboxes@Amir_Mansha:~$ sudo apt-get remove docker docker-engine  
[sudo] password for osboxes:  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package 'docker-engine' is not installed, so not removed  
Package 'docker' is not installed, so not removed  
0 upgraded, 0 newly installed, 0 to remove and 438 not upgraded.  
osboxes@Amir_Mansha:~$ sudo apt-get update  
Hit:1 http://us.archive.ubuntu.com/ubuntu xenial InRelease  
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease  
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease  
Hit:4 http://security.ubuntu.com/ubuntu xenial-security InRelease  
Reading package lists... Done  
osboxes@Amir_Mansha:~$ sudo apt-get remove docker docker-engine  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
Package 'docker-engine' is not installed, so not removed  
Package 'docker' is not installed, so not removed  
0 upgraded, 0 newly installed, 0 to remove and 438 not upgraded.  
osboxes@Amir_Mansha:~$
```

We install docker in a 64-bit machine because docker works on that only. So, in this 1st screenshot, we are removing any prior installed docker image first. Next, we are just updating our system and then checking again if any default docker image installed and removing that again.

```
osboxes@Amir_Mansha: ~  
osboxes@Amir_Mansha:~$ sudo apt-get install \apt-transport-https \ca-c  
ertificates \curl \software-properties-common  
Reading package lists... Done  
Building dependency tree  
Reading state information... Done  
The following additional packages will be installed:  
  libcurl3-gnutls python3-software-properties  
  software-properties-gtk  
The following NEW packages will be installed:  
  curl  
The following packages will be upgraded:  
  apt-transport-https ca-certificates libcurl3-gnutls  
  python3-software-properties software-properties-common  
  software-properties-gtk  
6 upgraded, 1 newly installed, 0 to remove and 432 not upgraded.  
Need to get 580 kB of archives.  
After this operation, 306 kB of additional disk space will be used.  
Do you want to continue? [Y/n] Y  
Get:1 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 li  
bcurl3-gnutls amd64 7.47.0-1ubuntu2.19 [189 kB]  
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 ap  
t-transport-https amd64 1.2.32ubuntu0.2 [26.6 kB]  
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 ca  
-certificates all 20210119~16.04.1 [148 kB]
```

In this 2nd screenshot, we are installing some required packages such as the certificates.

```
osboxes@Amir_Mansha: ~  
osboxes@Amir_Mansha:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo apt-key add -  
OK  
osboxes@Amir_Mansha:~$ sudo add-apt-repository "deb [arch=amd64] https://download.docker.com/linux/ubuntu $(lsb_release -cs) stable"  
osboxes@Amir_Mansha:~$ sudo apt-get update  
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [109 kB]  
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial InRelease  
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease [109 kB]  
Get:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease [107 kB]  
Get:5 https://download.docker.com/linux/ubuntu xenial InRelease [66.2 kB]  
Get:6 http://security.ubuntu.com/ubuntu xenial-security/main amd64 DEP-11 Metadata [93.7 kB]  
Get:7 http://security.ubuntu.com/ubuntu xenial-security/universe amd64 DEP-11 Metadata [130 kB]  
Get:8 http://security.ubuntu.com/ubuntu xenial-security/multiverse amd64 DEP-11 Metadata [2,464 B]  
Get:9 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 DEP-11 Metadata [326 kB]  
Get:10 http://us.archive.ubuntu.com/ubuntu xenial-updates/universe amd
```

In this 3rd screenshot, we use the curl command we install the docker gpg key and then we add the docker repository command where it stores the docker images. Next, we want to update everything again so the docker packages are included and updated in the system.

```

osboxes@Amir_Mansha:~$ sudo apt-get install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  containerd.io docker-ce-cli docker-ce-rootless-extras
  docker-scan-plugin git git-man liberror-perl libseccomp2 pigz
Suggested packages:
  aufs-tools cgroupfs-mount | cgroup-lite git-daemon-run
  | git-daemon-sysvinit git-doc git-el git-email git-gui gitk gitweb
  git-arch git-cvs git-mediawiki git-svn
Recommended packages:
  slirp4netns
The following NEW packages will be installed:
  containerd.io docker-ce docker-ce-cli docker-ce-rootless-extras
  docker-scan-plugin git git-man liberror-perl pigz
The following packages will be upgraded:
  libseccomp2
1 upgraded, 9 newly installed, 0 to remove and 431 not upgraded.
Need to get 111 MB of archives.
After this operation, 491 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://us.archive.ubuntu.com/ubuntu xenial/universe amd64 pigz a

```

In this 4th screenshot, now we finally install docker -ce version.

```

osboxes@Amir_Mansha:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
b8dfde127a29: Pull complete
Digest: sha256:f2266cbfc127c960fd30e76b7c792dc23b588c0db76233517e1891a
4e357d519
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correc
tly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker H
ub.
    (amd64)
 3. The Docker daemon created a new container from that image which ru
ns the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which
sent it

```

To verify we have docker installed, we run the docker image of “hello world” and as you can see it printed “hello from docker!” which means we have successfully installed docker.

```
osboxes@Amir_Mansha: ~  
osboxes@Amir_Mansha:~$ sudo docker images  
REPOSITORY      TAG         IMAGE ID      CREATED        SIZE  
hello-world      latest      d1165f221234  7 weeks ago   13.3kB  
osboxes@Amir_Mansha:~$
```

Here we use the command “sudo docker images” to see the image we created of the “hello world” and as you can see it prints out the hello – world under repository along with other information of the image such as the image id.

```
osboxes@Amir_Mansha: ~  
osboxes@Amir_Mansha:~$ sudo docker images  
REPOSITORY      TAG         IMAGE ID      CREATED        SIZE  
hello-world      latest      d1165f221234  7 weeks ago   13.3kB  
osboxes@Amir_Mansha:~$ service docker status  
● docker.service - Docker Application Container Engine  
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor  
   Active: active (running) since Sun 2021-04-25 22:38:17 EDT; 5min ag  
     Docs: https://docs.docker.com  
  Main PID: 770 (dockerd)  
    CGroup: /system.slice/docker.service  
            └─770 /usr/bin/dockerd -H fd:// --containerd=/run/container
```

In this last screenshot of task 1, we run the status of the docker image I installed, and it prints out a lot of information about the docker image such as its active status and other things.

Task 2

In this task 2, we mainly want to create a docker image.

```
osboxes@Amir_Mansha:~$ sudo docker search centos
[sudo] password for osboxes:
NAME                STARS     OFFICIAL    AUTOMATED
centos              6519      [OK]        The official build of CentOS.
ansible/centos7-ansible  133      [OK]        Ansible on Centos7
consol/centos-xfce-vnc  128      [OK]        Centos container with "headless" VN
C session...
jdeathe/centos-ssh    117      [OK]        OpenSSH / Supervisor / EPEL/IUS/SCL
Repos - ...
centos/systemd        98        [OK]        systemd enabled base container.
imagine10255/centos6-lnmp-php56  58        [OK]        centos6-lnmp-php56
tutum/centos         47        Simple CentOS docker image with SSH
access
kinogmt/centos-ssh    29        [OK]        CentOS with SSH
pivotaldata/centos-gpdb-dev
Tag names... 13
```

In this 1st screenshot of Task 2, we find the image container of centos operating system, so we use the “search” command and see the list of the repositories for centos. We select the 1st one.

```
osboxes@Amir_Mansha: ~
osboxes@Amir_Mansha:~$ sudo docker pull centos
Using default tag: latest
latest: Pulling from library/centos
7a0437f04f83: Pull complete
Digest: sha256:5528e8b1b1719d34604c87e11dcd1c0a20bedf46e83b5632cdeac91
b8c04efc1
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest
osboxes@Amir_Mansha:~$ sudo docker ps -a
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
25bd7e8c5634   hello-world    "/hello"                 20 minutes ago   Exited (0) 20
minutes ago
upbeat_blackburn
osboxes@Amir_Mansha:~$ sudo docker images
REPOSITORY    TAG       IMAGE ID       CREATED          SIZE
hello-world   latest    d1165f221234   7 weeks ago     13.3kB
centos        latest    300e315adb2f   4 months ago    209MB
osboxes@Amir_Mansha:~$
```


In this 2nd screenshot, we pull and load the image of centos using the “pull” command. Using the “docker images” command we see the image of the centos that I created along with the image id.

```
osboxes@Amir_Mansha:~$ docker run centos cat /etc/issue
docker: Got permission denied while trying to connect to the Docker da
emon socket at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2
Fdocker.sock/v1.24/containers/create: dial unix /var/run/docker.sock:
connect: permission denied.
See 'docker run --help'.
osboxes@Amir_Mansha:~$ sudo docker run centos cat /etc/issue
\S
Kernel \r on an \m

osboxes@Amir_Mansha:~$ sudo docker run centos cat /etc/shadow
root:!locked::0:99999:7:::
bin:!:18397:0:99999:7:::
daemon:!:18397:0:99999:7:::
adm:!:18397:0:99999:7:::
lp:!:18397:0:99999:7:::
sync:!:18397:0:99999:7:::
shutdown:!:18397:0:99999:7:::
halt:!:18397:0:99999:7:::
mail:!:18397:0:99999:7:::
operator:!:18397:0:99999:7:::
games:!:18397:0:99999:7:::
```

In this 3rd screenshot, we just explore different commands within the centos operating system. So, we run the /etc commands and get the /etc/shadow and /etc/issue information.

```

osboxes@Amir_Mansha:~$ sudo docker run -it centos bash
[root@b2f94b916b46 /]# ls
bin  etc  lib  lost+found  mnt  proc  run  srv  tmp  var
dev  home  lib64  media      opt  root  sbin  sys  usr
[root@b2f94b916b46 /]# pwd
/
[root@b2f94b916b46 /]# whoami
root
[root@b2f94b916b46 /]# uname -a
Linux b2f94b916b46 4.15.0-45-generic #48~16.04.1-Ubuntu SMP Tue Jan 29 1
UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
[root@b2f94b916b46 /]# exit
exit
osboxes@Amir_Mansha:~$ sudo docker ps -l
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS
ORTS          NAMES
b2f94b916b46   centos    "bash"    16 seconds ago   Exited (0) 3 seconds
funny_beaaver
osboxes@Amir_Mansha:~$ sudo docker start b2f94b916b46
b2f94b916b46
osboxes@Amir_Mansha:~$ sudo docker run centos bash -c "yum -y install ng
CentOS Linux 8 - AppStream          5.0 MB/s | 6.3 MB    00
CentOS Linux 8 - BaseOS             3.7 MB/s | 2.3 MB    00
CentOS Linux 8 - Extras             23 kB/s | 9.6 kB    00
Dependencies resolved.
=====
=====
Package                               Arch  Version
po      Size
=====
=====

```

In this 4th screenshot, we are still exploring commands. We run the bash command within the centos operating system to open the shell and run multiple commands such as “ls, pwd, whoami” commands. Next, we use the “ps -l” command to find out more information about the centos image such as the name and container id etc. I used the container id instead of the name to start the docker container. We use the “yum” command to install packages in the centos operating system. We installed the “nginx” web server that uses HTTP in the centos OS.

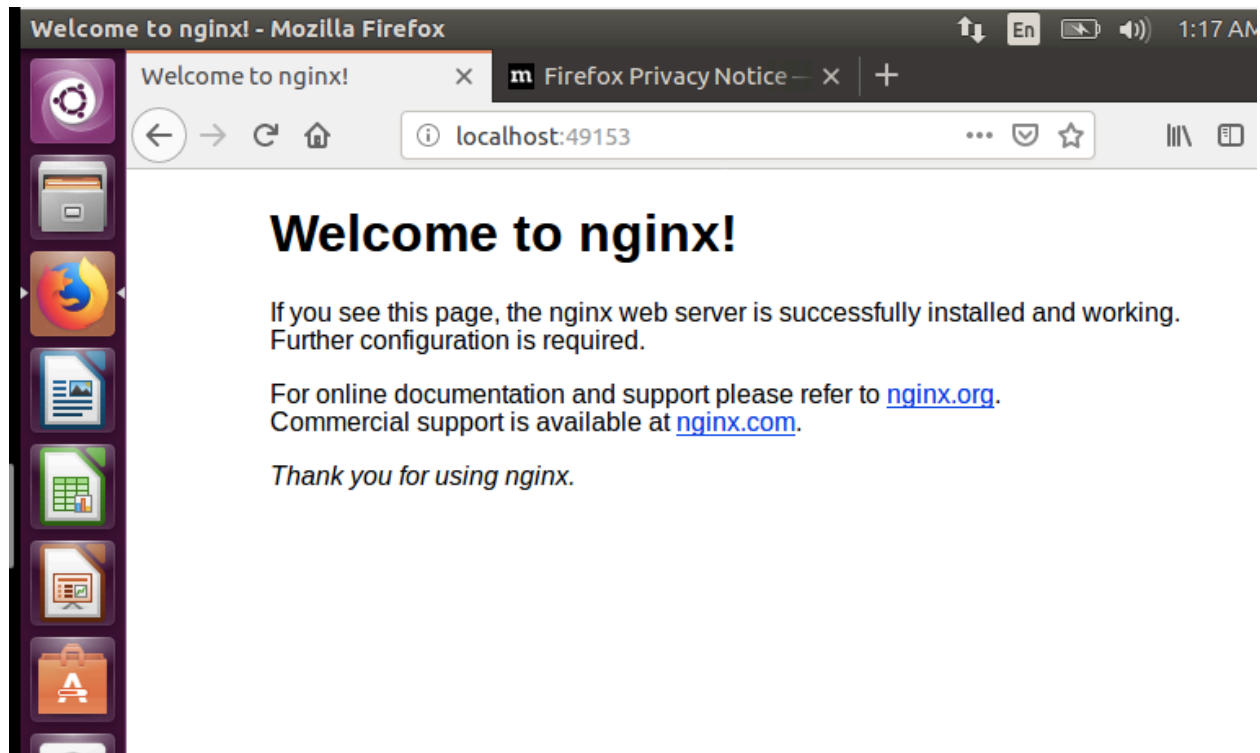
Task 3

In this task 3, we want to create our own webpage using nginx and docker by creating a html file.

```
osboxes@Amir_Mansha:~$ sudo docker run --name mynginx -P -d nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
f7ec5a41d630: Pull complete
aa1efa14b3bf: Pull complete
b78b95af9b17: Pull complete
c7d6bca2b8dc: Pull complete
cf16cd8e71e0: Pull complete
0241c68333ef: Pull complete
Digest: sha256:75a55d33ecc73c2a242450a9f1cc858499d468f077ea942867e662c247b5e412
Status: Downloaded newer image for nginx:latest
ac728aab82074a2218bd1dbb279912501e8f50d9768527888dd666ca7b54a2a7
```

```
osboxes@Amir_Mansha: ~
osboxes@Amir_Mansha:~$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
b2f94b916b46   centos    "bash"                   3 minutes ago Up 3 minutes
              funny_bea
0fe19b790330   nginx     "/docker-entrypoint...." 8 minutes ago Up 8 minutes
0.0.0.0:49153->80/tcp, :::49153->80/tcp mynginx
osboxes@Amir_Mansha:~$
```

In the 1st screenshot of task 3- step 1, we are creating a docker container for nginx using the 1st command. Using the “ps” command we see the nginx image we created and the webserver container where it includes the port # 49153. We will use this port # to open the webserver using firefox. The “-P” means port which lets us have specific port so we don’t get conflicted of multiple nginx instances.



We typed the localhost:49153 to open the nginx webserver and as you can see it was successfully opened using the port# that was given to us in the screenshot above.

```
osboxes@Amir_Mansha:~$ curl http://localhost:49153
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed and
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
osboxes@Amir_Mansha:~$
```

I also used the curl command to see the contents of the html document for the nginx webpage.

Task 3-Step 2

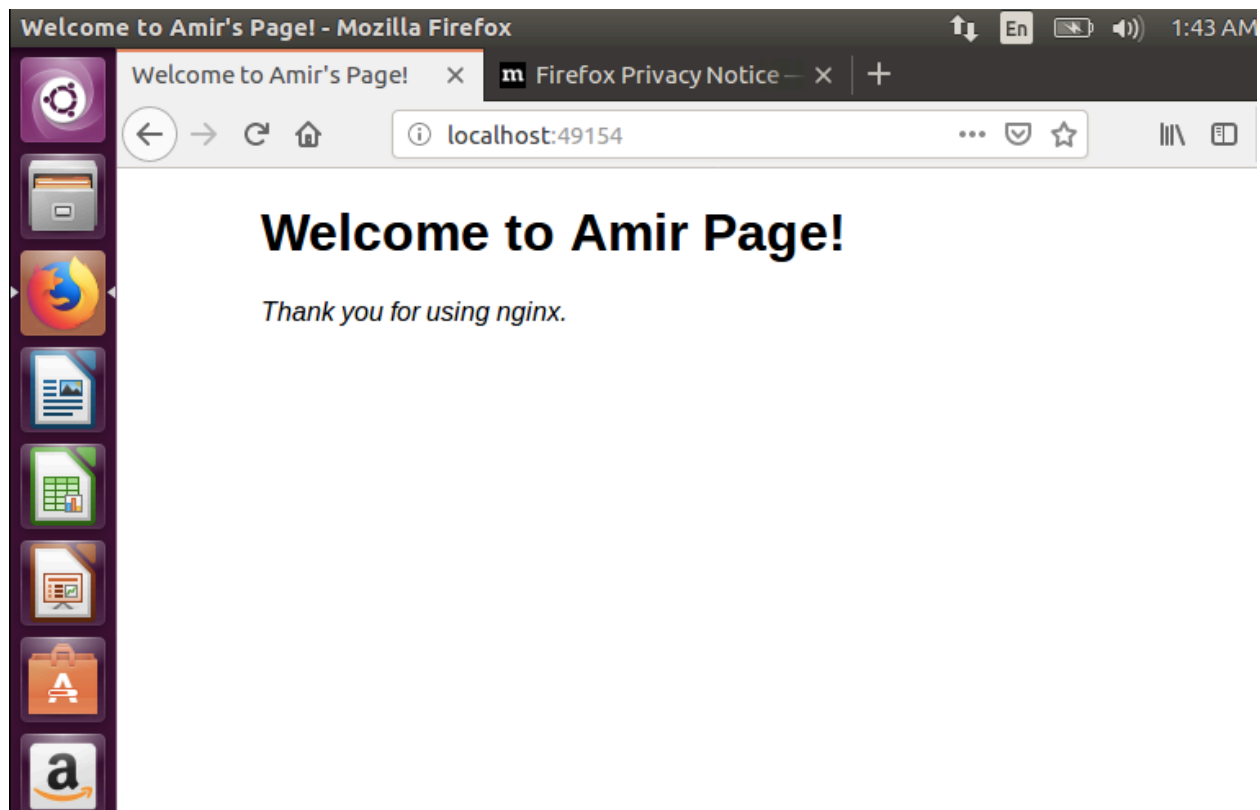
```
osboxes@Amir_Mansha: ~/Desktop/html
osboxes@Amir_Mansha:~/Desktop/html$ sudo docker run --name mynginx01 -P -d
Desktop/html:/usr/share/nginx/html nginx
cb396da2a41b7adc092de3e3901c3f926cdc2f78c1af17b8943d214800915b
osboxes@Amir_Mansha:~/Desktop/html$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
cb396da2a41b   nginx    "/docker-entrypoint...." 23 seconds ago Up 19 sec
0.0.0.0:49154->80/tcp, :::49154->80/tcp mynginx01
b2f94b916b46   centos    "bash"                   17 minutes ago Up 16 mi
funny_bever
0fe19b790330   nginx    "/docker-entrypoint...." 22 minutes ago Up 22 mi
0.0.0.0:49153->80/tcp, :::49153->80/tcp mynginx
osboxes@Amir_Mansha:~/Desktop/html$
```

In this screenshot for task 3- step 2, I created another image using a different name “mynginx01” and used the “ps” command to get the new information for that image such as the port # which is 49154.

```
osboxes@Amir_Mansha: ~/Desktop/html
<!DOCTYPE html>
<html>
<head>
<title>Welcome to Amir's Page!</title>
<style>
  body {
    width: 35em;
    margin: 0 auto;
    font-family: Tahoma, Verdana, Arial, sans-serif;
  }
</style>
</head>
<body>
<h1>Welcome to Amir Page!</h1>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
```

I created an index.html file in my desktop and used the contents from the previous webpage and just replaced “welcome to nginx” to “Welcome to Amir’s page!” and saved that html file in my desktop folder. I have to map the html file to the new container I created using the 1st command and provide the path to the desktop directory in where the index.html is in.



On Firefox, I used the port # 49154 that was given in the above screenshot to open the webpage in Firefox and display the html file I created and the title I put in the html contents in the file was displayed on the webpage which was “Welcome to Amir’s Page!”

```
osboxes@Amir_Mansha: ~/Desktop/html
osboxes@Amir_Mansha:~/Desktop/html$ sudo docker stop mynginx01
mynginx01
osboxes@Amir_Mansha:~/Desktop/html$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS
PORTS         NAMES
b2f94b916b46   centos    "bash"                   37 minutes ago Up 36 minutes
0fe19b790330   nginx     "/docker-entrypoint.     41 minutes ago Up 41 minutes
0.0.0.0:49153->80/tcp, :::49153->80/tcp mynginx
```

In this last screenshot, we stop the run the running container and here I stopped the “mynginx01” container which I used to display my custom webpage. I run the “ps” command and you can see the “mynginx01” is no longer there and stopped. You can only see instances that are running that I created before which are the “centos” and “mynginx” containers.