

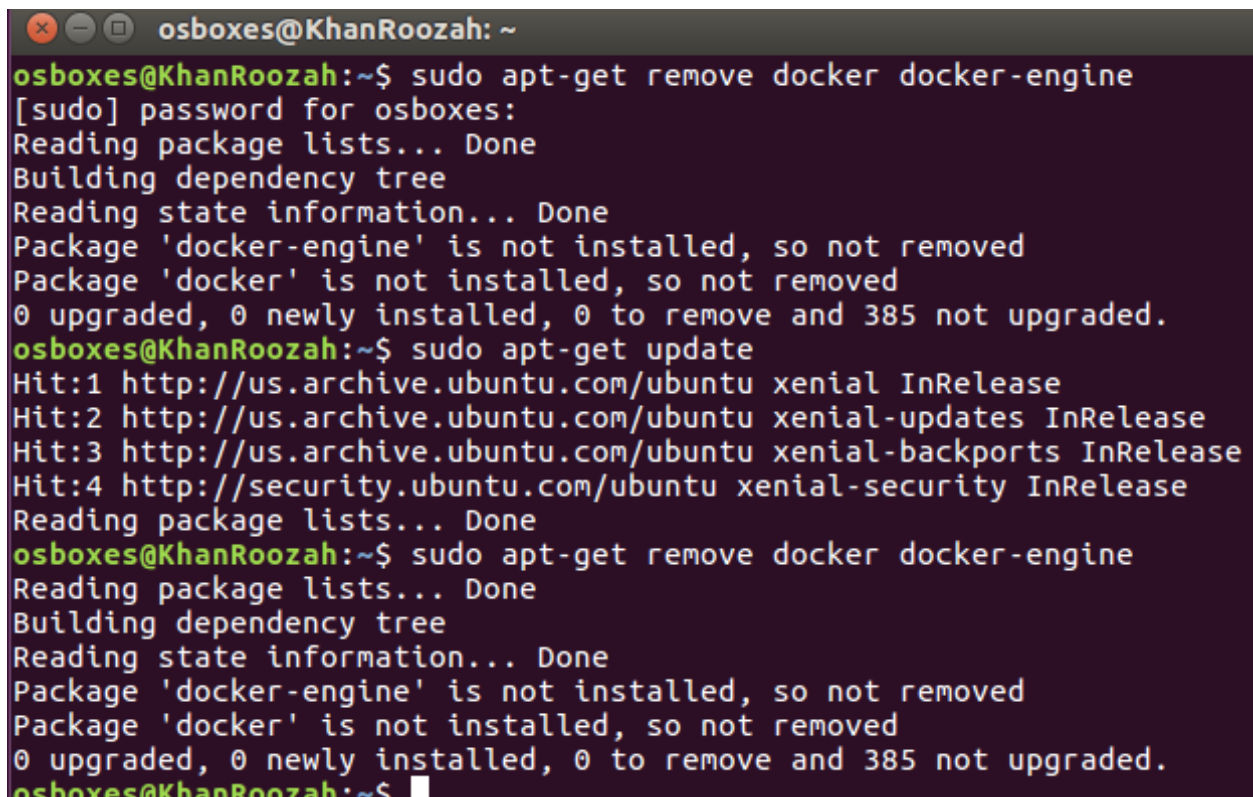
INTRO:

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@KhanRoozah: ~'. The user enters the command 'sudo apt-get remove docker docker-engine'. The terminal shows the password prompt, package lists being read, a dependency tree being built, and state information being read. It reports that 'docker-engine' and 'docker' are not installed. Then, the user enters 'sudo apt-get update'. The terminal shows four hits from various Ubuntu repositories. Finally, the user enters 'sudo apt-get remove docker docker-engine' again, and the terminal shows the same output as before, confirming they are not installed.

```
osboxes@KhanRoozah: ~  
osboxes@KhanRoozah:~$ 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 385 not upgraded.  
osboxes@KhanRoozah:~$ 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@KhanRoozah:~$ 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 385 not upgraded.  
osboxes@KhanRoozah:~$
```

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@KhanRoozah:~$ sudo apt-get install \apt-transport-https \ca-certificates
\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 379 not upgraded.
Need to get 572 kB of archives.
After this operation, 295 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 libcurl3-gnu
tls amd64 7.47.0-1ubuntu2.15 [184 kB]
Get:2 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 apt-transpor
t-https amd64 1.2.32ubuntu0.1 [26.7 kB]
Get:3 http://us.archive.ubuntu.com/ubuntu xenial-updates/main amd64 ca-certifica
tes all 20190110-16.04.1 [146 kB]

```

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

```

osboxes@KhanRoozah:~$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg |
sudo apt-key add -
OK
osboxes@KhanRoozah:~$ sudo add-apt-repository "deb [arch=amd64] https://download
.docker.com/linux/ubuntu $(lsb_release -cs) stable"
osboxes@KhanRoozah:~$ sudo apt-get update
Hit:1 http://security.ubuntu.com/ubuntu xenial-security InRelease
Hit:2 http://us.archive.ubuntu.com/ubuntu xenial InRelease
Hit:3 http://us.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:4 http://us.archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:5 https://download.docker.com/linux/ubuntu xenial InRelease [66.2 kB]
Get:6 https://download.docker.com/linux/ubuntu xenial/stable amd64 Packages [14.
5 kB]
Fetched 80.7 kB in 0s (101 kB/s)
Reading package lists... Done
osboxes@KhanRoozah:~$

```

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@KhanRoozah:~$ sudo apt-get install docker-ce
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce-cli git git-man
  liberror-perl libseccomp2 pigz
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-el git-email git-gui gitk
  gitweb git-arch git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount containerd.io docker-ce docker-ce-cli git git-man
  liberror-perl pigz
The following packages will be upgraded:
  libseccomp2
1 upgraded, 9 newly installed, 0 to remove and 378 not upgraded.
Need to get 88.5 MB of archives.
After this operation, 407 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y

```

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

```

osboxes@KhanRoozah:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
0e03bdcc26d7: Pull complete
Digest: sha256:49a1c8800c94df04e9658809b006fd8a686cab8028d33cfba2cc049724254202
Status: Downloaded newer image for hello-world:latest

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

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 Hub.
    (amd64)

```

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@KhanRoozah:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
hello-world         latest             bf756fb1ae65       6 months ago
13.3kB
osboxes@KhanRoozah:~$

```

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@KhanRoozah:~$ service docker status
● docker.service - Docker Application Container Engine
   Loaded: loaded (/lib/systemd/system/docker.service; enabled; vendor preset: e
   Active: active (running) since Fri 2020-07-24 19:57:31 EDT; 2min 49s ago
     Docs: https://docs.docker.com
   Main PID: 13372 (dockerd)
    Tasks: 11
   Memory: 73.7M
      CPU: 1.255s
   CGroup: /system.slice/docker.service
           └─13372 /usr/bin/dockerd -H fd:// --containerd=/run/containerd/contai

```

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@KhanRoozah: ~
osboxes@KhanRoozah:~$ sudo docker search centos
NAME                                DESCRIPTION
STARS                               OFFICIAL    AUTOMATED
centos                              The official build of CentOS.
6101                                [OK]
ansible/centos7-ansible             Ansible on Centos7
132                                  [OK]
consol/centos-xfce-vnc              Centos container with "headless" VNC session...
117                                  [OK]
jdeathe/centos-ssh                  OpenSSH / Supervisor / EPEL/IUS/SCL Repos - ...
115                                  [OK]
centos/systemd                      systemd enabled base container.
86                                  [OK]
centos/mysql-57-centos7             MySQL 5.7 SQL database server
78
image10255/centos6-lamp-php56       centos6-lamp-php56
58                                  [OK]
tutum/centos                        Simple CentOS docker image with SSH access
47
centos/postgresql-96-centos7        PostgreSQL is an advanced Object-Relational ...
46
kinogmt/centos-ssh                  CentOS with SSH
29                                  [OK]
pivotaldata/centos-gpdb-dev         CentOS image for GPDB development. Tag names...

```

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@KhanRoozah:~$ sudo docker pull centos
Using default tag: latest
latest: Pulling from library/centos
6910e5a164f7: Pull complete
Digest: sha256:4062bbdd1bb0801b0aa38e0f83dece70fb7a5e9bce223423a68de2d8b784b43b
Status: Downloaded newer image for centos:latest
docker.io/library/centos:latest
osboxes@KhanRoozah:~$ sudo docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
285827e75ecd       hello-world        "/hello"           10 minutes ago
Exited (0) 10 minutes ago                  condescending_ardinghelli
osboxes@KhanRoozah:~$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
centos               latest             831691599b88       5 weeks ago
215MB
hello-world         latest             bf756fb1ae65       6 months ago
13.3kB
osboxes@KhanRoozah:~$ █

```

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@KhanRoozah:~$ docker run centos cat /etc/issue
docker: Got permission denied while trying to connect to the Docker
t at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker
ontainers/create: dial unix /var/run/docker.sock: connect: permissi
See 'docker run --help'.
osboxes@KhanRoozah:~$
osboxes@KhanRoozah:~$ sudo docker run centos cat /etc/issue
\S
Kernel \r on an \m

osboxes@KhanRoozah:~$ sudo docker run centos cat /etc/shadow
root:!locked::0:99999:7:::
bin:*:18358:0:99999:7:::
daemon:*:18358:0:99999:7:::
adm:*:18358:0:99999:7:::
lp:*:18358:0:99999:7:::
sync:*:18358:0:99999:7:::
shutdown:*:18358:0:99999:7:::
halt:*:18358:0:99999:7:::
mail:*:18358:0:99999:7:::
operator:*:18358:0:99999:7:::
games:*:18358:0:99999:7:::
ftp:*:18358:0:99999:7:::
nobody:*:18358:0:99999:7:::
dbus:!!:18424:::
systemd-coredump:!!:18424:::
systemd-resolve:!!:18424:::
osboxes@KhanRoozah:~$
```

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.


```

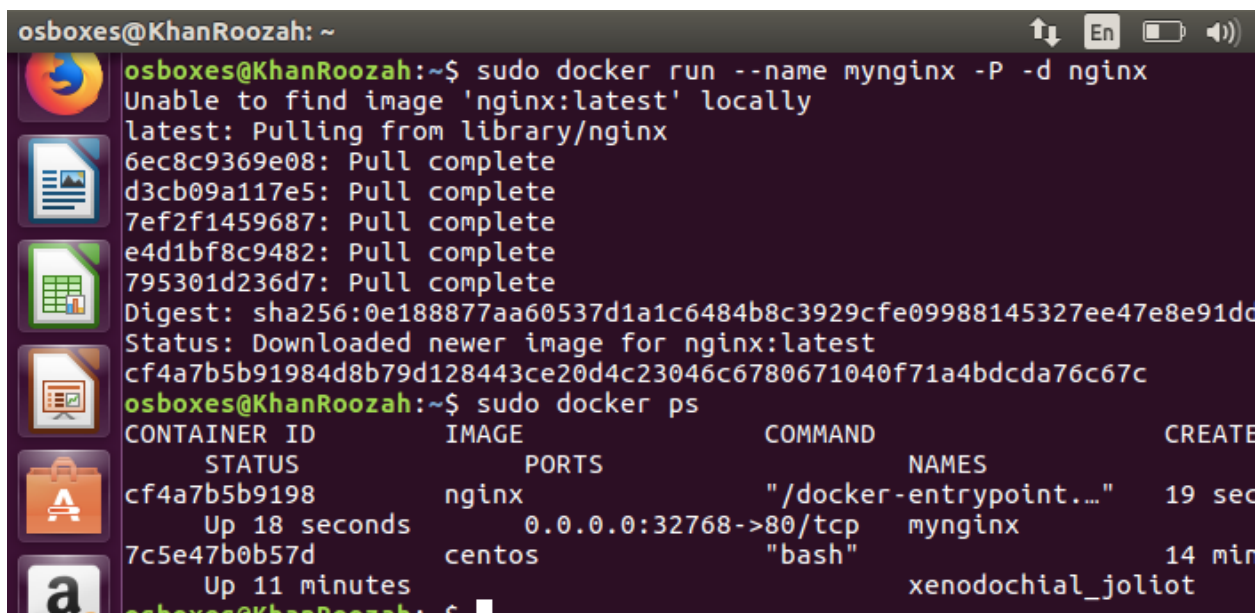
osboxes@KhanRoozah:~$ sudo docker run -it centos bash
[root@7c5e47b0b57d /]# ls
bin  etc  lib  lost+found  mnt  proc  run  srv  tmp  var
dev  home  lib64  media  opt  root  sbin  sys  usr
[root@7c5e47b0b57d /]# pwd
/
[root@7c5e47b0b57d /]# whoami
root
[root@7c5e47b0b57d /]# uname -a
Linux 7c5e47b0b57d 4.15.0-45-generic #48~16.04.1-Ubuntu SMP Tue Jan 29 18:03:48
UTC 2019 x86_64 x86_64 x86_64 GNU/Linux
[root@7c5e47b0b57d /]# exit
exit
osboxes@KhanRoozah:~$ sudo docker ps -l
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS              PORTS              NAMES
7c5e47b0b57d        centos              "bash"             About a minute ago
Exited (0) 40 seconds ago                xenodochial_joliot
osboxes@KhanRoozah:~$ sudo docker start 7c5e47b0b67d
Error response from daemon: No such container: 7c5e47b0b67d
Error: failed to start containers: 7c5e47b0b67d
osboxes@KhanRoozah:~$ sudo docker start 7c5e47b0b57d
7c5e47b0b57d
osboxes@KhanRoozah:~$ sudo docker run centos bash -c "apt-get -y install nginx"
bash: apt-get: command not found
osboxes@KhanRoozah:~$ sudo docker run centos bash -c "yum -y install nginx"
CentOS-8 - AppStream                8.9 MB/s | 5.8 MB    00:00
CentOS-8 - Base                     2.1 MB/s | 2.2 MB    00:01
CentOS-8 - Extras                   21 kB/s | 7.0 kB     00:00
Dependencies resolved.
=====
=====
Package                                Arch    Version                                R

```

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.

A terminal window titled 'osboxes@KhanRoozah: ~' with a dark background. The user enters the command 'sudo docker run --name mynginx -P -d nginx'. The output shows the process of pulling the 'nginx:latest' image from the Docker library, including several progress bars and a digest. After the pull is complete, the user enters 'sudo docker ps'. The output is a table listing the running containers.

```
osboxes@KhanRoozah:~$ sudo docker run --name mynginx -P -d nginx
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
6ec8c9369e08: Pull complete
d3cb09a117e5: Pull complete
7ef2f1459687: Pull complete
e4d1bf8c9482: Pull complete
795301d236d7: Pull complete
Digest: sha256:0e188877aa60537d1a1c6484b8c3929cfe09988145327ee47e8e91dc
Status: Downloaded newer image for nginx:latest
cf4a7b5b91984d8b79d128443ce20d4c23046c6780671040f71a4bdcdca76c67c
osboxes@KhanRoozah:~$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	STATUS	PORTS	NAMES	CREATED
cf4a7b5b9198	nginx	"/docker-entrypoint..."	Up 18 seconds	0.0.0.0:32768->80/tcp	mynginx	19 seconds ago
7c5e47b0b57d	centos	"bash"	Up 11 minutes		xenodochial_joliot	14 minutes ago

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 # 32768. 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 port # 32768 to open the nginx webserver and as you can see it was successfully opened using the port#.

```
osboxes@KhanRoozah:~$ curl http://localhost:32768
<!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>
```

I also used the curl command to see the contents of the html document for the nginx webpage. I used the contents to frame my own webpage that I will create down below.

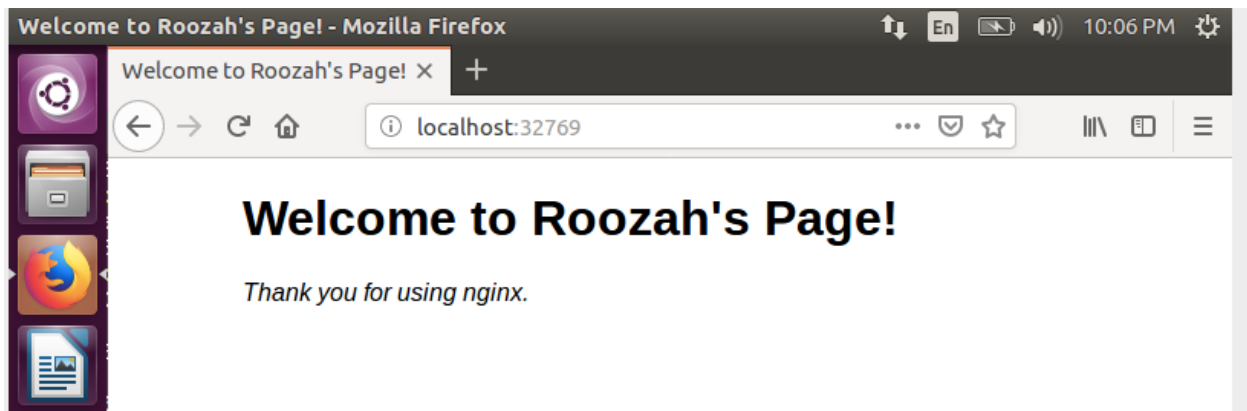
Task 3-Step 2

```
osboxes@KhanRoozah:~/Desktop/html$ sudo docker run --name mynginx01 -P -d -v ~/Desktop/html:/usr/share/nginx/html nginx
3926777107b9872fae4ffb544cb6d73a94b61e80e255c62c86cbe4aeb6fefb4d
osboxes@KhanRoozah:~/Desktop/html$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED
3926777107b9	nginx	"/docker-entrypoint..."	10 seconds ago
Up 7 seconds		0.0.0.0:32769->80/tcp mynginx01	
0b7562f37b3a	nginx	"/docker-entrypoint..."	12 minutes ago
Up 12 minutes		0.0.0.0:32768->80/tcp mynginx	
11ef72058744	centos	"bash"	16 minutes ago
Up 15 minutes		charming_ishizaka	

```
osboxes@KhanRoozah:~/Desktop/html$ vi index.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 #. 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 roozah’s page” and saved that html file in my desktop. 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 # 32769 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 Roozah’s Page!”

```

psboxes@KhanRoozah:~/Desktop/html$ sudo docker stop mynginx01
mynginx01
psboxes@KhanRoozah:~/Desktop/html$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED
STATUS            PORTS              NAMES
9b7562f37b3a       nginx              "/docker-entrypoint..." 16 minutes ago
Up 16 minutes      0.0.0.0:32768->80/tcp  mynginx
11ef72058744       centos             "bash"             20 minutes ago
Up 20 minutes                      charming_ishizaka
psboxes@KhanRoozah:~/Desktop/html$

```

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.