

EXPERIMENT-08

NAME: SHAIKH MUBASHIRA TUFEL AHMED
ROLL NO: 612055 **COURSE:** ADVANCE DEVOPS(ITL504)
BRANCH: T.E. INFORMATION TECHNOLOGY (SEM 5)

1. What is hub.docker.com?

➔ **Docker Hub** is a repository service and it is a cloud-based service where people push their Docker Container Images and also pull the Docker Container Images from the **Docker Hub** anytime or anywhere via the internet. It provides features such as you can push your images as private or public. Mainly DevOps team uses the Docker Hub. It is an open-source tool and freely available for all operating systems. It is like storage where we store the images and pull the images when it is required. When a person wants to push/pull images from the Docker Hub they must have a basic knowledge of Docker. Let us discuss the requirements of the Docker tool.

2. What is docker hub used for?

➤ Docker Hub Features:

- **Repositories:** It contains the Push and Pull process for container images.
- **Teams and Organizations:** It allows access to developer/user to private repositories of container images.
- **Docker Official Images:** It Pulls and uses high-standard quality container images rendered by Docker.
- **Docker Verified Publisher Images:** It Pulls and uses high-standard quality container images rendered by outside vendors.
- **Builds:** It provides the mechanisms that automatically formulate container images from Bitbucket and GitHub and push them to Docker Hub.
- **Webhooks:** It triggers certain actions after a successful push to a container to combine Docker Hub with additional services.

3. Install docker on AWS EC2 –Ubuntu by using curl

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

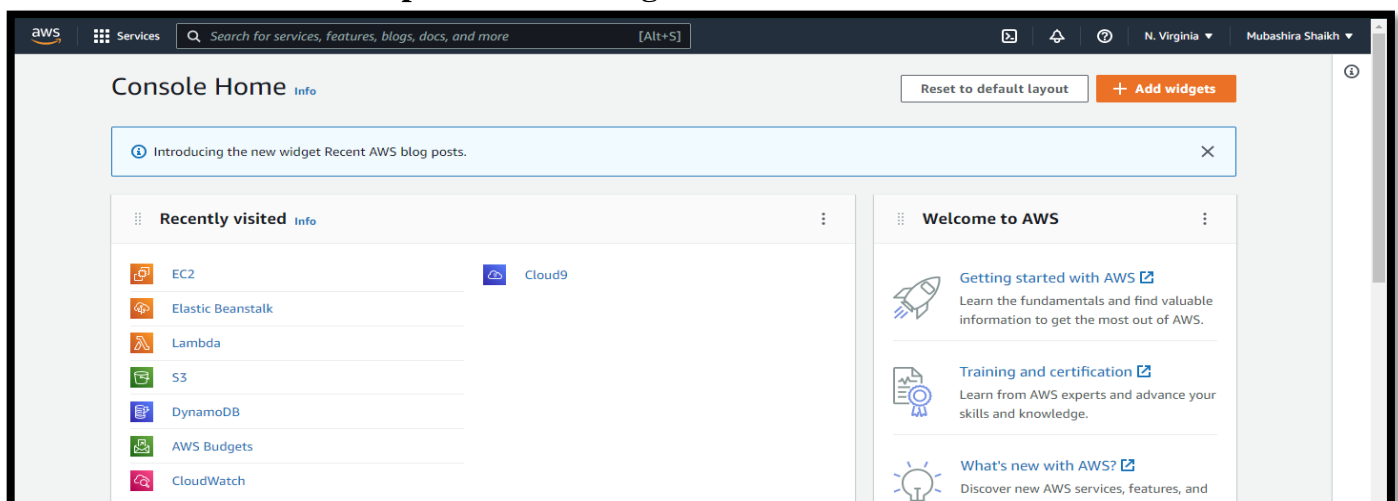
```
#sh get-docker.sh
```

4. Run hello-world from docker hub and explain the steps

5. Pull 3 or 4 images, one of the python, run “Hello World “inside container.

6. Demonstrate any 15 docker command and explain its uses.

Step 1: AWS Management Console Dashboard.



Step 2: Search for EC2 → Click on Launch instance → Give a name to your instance and create an Ubuntu instance with 20.04 lts version.

The screenshot shows the 'Launch an instance' page in the AWS Management Console. The 'Name and tags' section has a name field containing 'Mubashira'. The 'Application and OS Images (Amazon Machine Image)' section shows a search bar and a 'Quick Start' section with various OS options. The 'Ubuntu' option is selected. The 'Summary' section on the right shows the configuration: 1 instance, Canonical Ubuntu 22.04 LTS AMI, t2.micro instance type, New security group, and 1 volume of 8 GiB. A 'Free tier' notification is displayed, stating that the first year includes 750 hours of t2.micro usage. The 'Launch instance' button is visible at the bottom right.

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: Mubashira [Add additional tags](#)

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Search our full catalog including 1000s of application and OS images

Quick Start

Amazon Linux macOS **Ubuntu** Windows Red Hat S [Browse more AMIs](#)

Amazon Machine Image (AMI)

Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more ami-052efd3df9dad4825

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

[Cancel](#) [Launch instance](#)

Step 3: Create new key pair for your instance → Network Settings → allow the HTTPS and HTTP traffic.

The screenshot shows the 'Launch an instance' page with the 'Network settings' section expanded. The 'Key pair (login)' section shows a dropdown for 'docker_ubuntu' and a 'Create new key pair' button. The 'Network settings' section shows the network ID 'vpc-01c92761c00814a42', subnet 'No preference (Default subnet in any availability zone)', and 'Auto-assign public IP' set to 'Enable'. The 'Firewall (security groups)' section shows 'Create security group' selected, and a new security group named 'launch-wizard-2' is being created with rules for SSH, HTTP, and HTTPS traffic. The 'Summary' section on the right shows the same configuration as before. The 'Launch instance' button is visible at the bottom right.

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required: docker_ubuntu [Create new key pair](#)

Network settings Info [Edit](#)

Network Info: vpc-01c92761c00814a42

Subnet Info: No preference (Default subnet in any availability zone)

Auto-assign public IP Info: Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

☒ Create security group ☐ Select existing security group

We'll create a new security group called 'launch-wizard-2' with the following rules:

☒ Allow SSH traffic from

☒ Allow HTTPs traffic from the internet

☒ Allow HTTP traffic from the internet

Summary

Number of instances Info: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ...read more ami-052efd3df9dad4825

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance usage on free tier AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

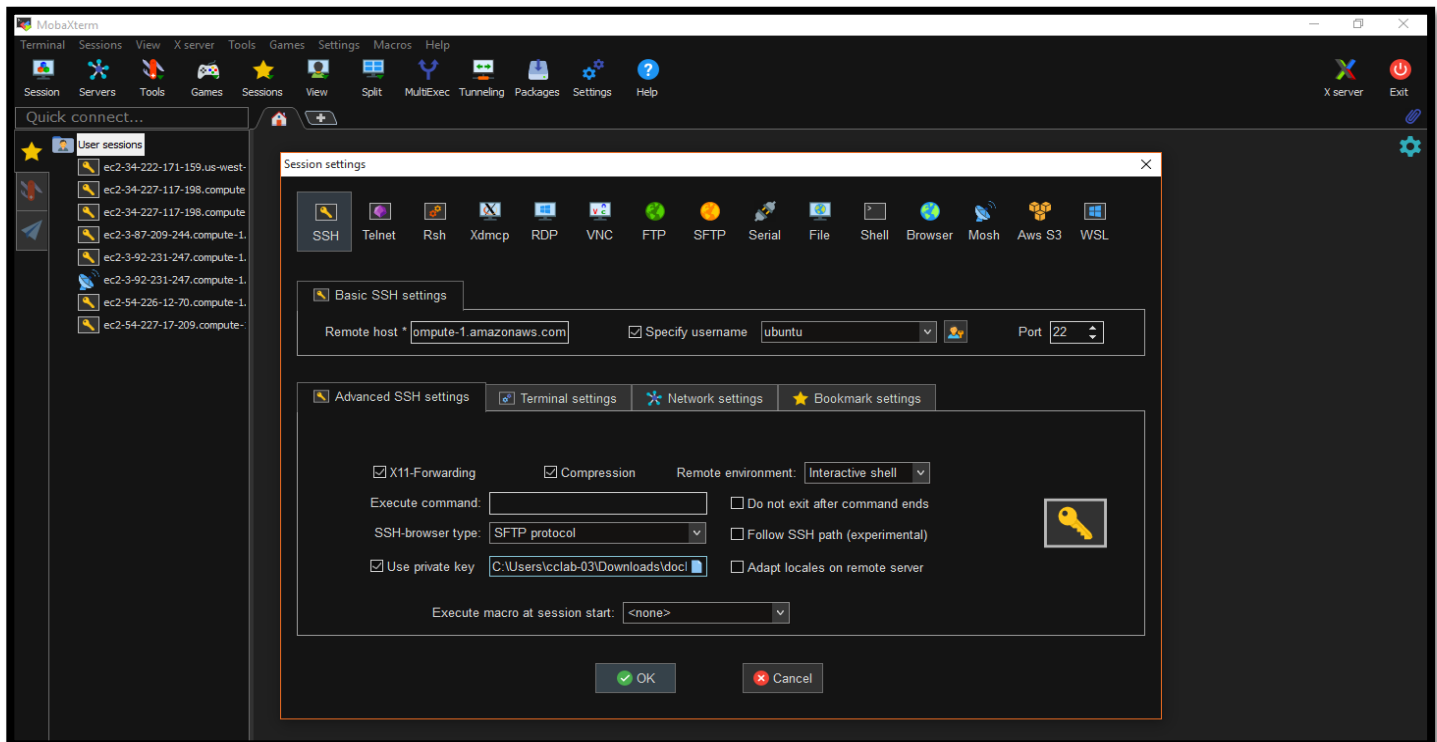
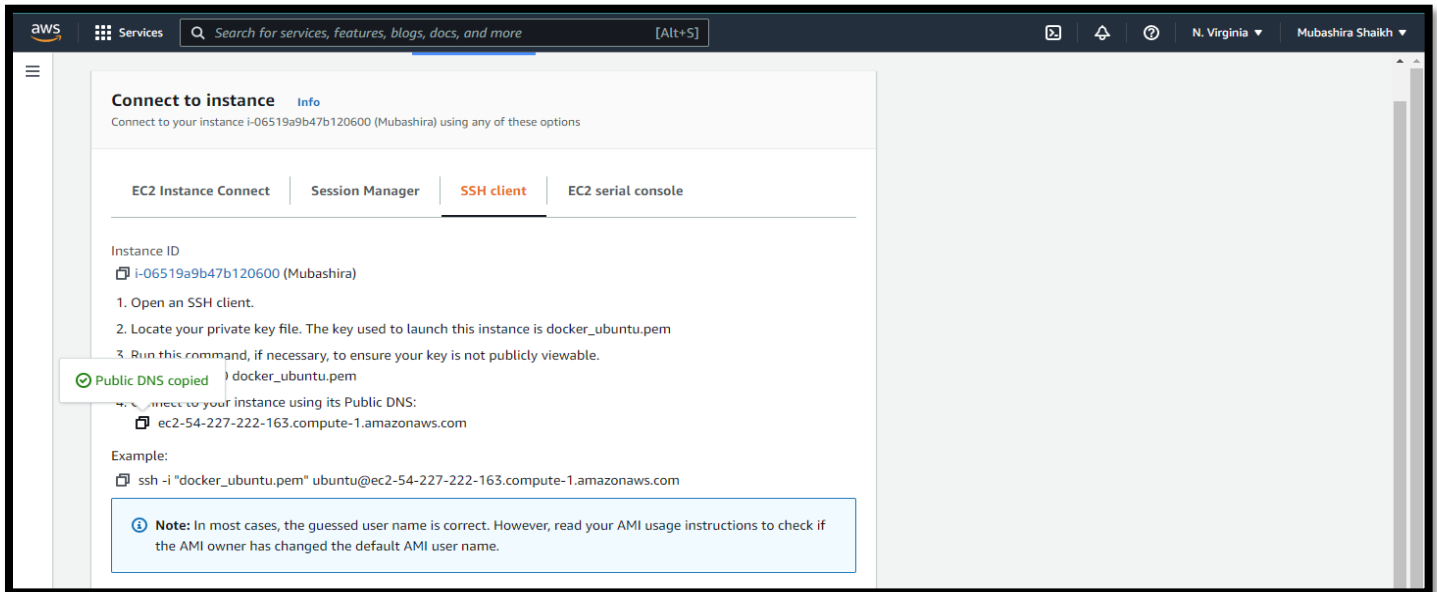
[Cancel](#) [Launch instance](#)

Feedback Looking for language selection? Find it in the new Unified Settings

© 2022, Amazon Internet Services Private Ltd. or its affiliates. Privacy Terms Cookie preferences

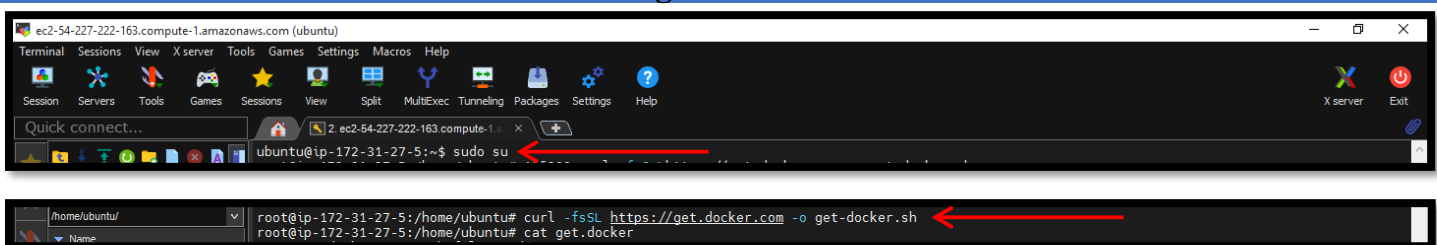
docker_ubuntu.pem [Show all](#)

Step 4: Launch MobaXterm→Select SSH session→Copy the public DNS of your instance and paste it into the remote host. Use the downloaded key pair as the private key.



**Step 5: Run the command 'sudo su' to gain root user access.
Then enter commands:**

**→curl -fsSL https://get.docker.com -o get-docker.sh and
→sh get-docker.sh**



```
ec2-54-227-222-163.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ubuntu/
Name
..
.cache
.ssh
.bash_logout
.bashrc
.profile
.xauthority
Remote monitoring
Follow terminal folder

root@ip-172-31-27-5:/home/ubuntu# ^[[200~#sh get-docker.sh
#sh: command not found
root@ip-172-31-27-5:/home/ubuntu# sh get-docker.sh
# Executing docker install script, commit: 4f282167c425347a931ccfd95cc91fab041d414f
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq apt-transport-https ca-certificates curl >/dev/null
+ sh -c mkdir -p /etc/apt/keyrings && chmod -R 0755 /etc/apt/keyrings
+ sh -c curl -fsSL "https://download.docker.com/linux/ubuntu/gpg" | gpg --dearmor --yes -o /etc/apt/keyrings/docker.gpg
+ sh -c chmod a+r /etc/apt/keyrings/docker.gpg
+ sh -c echo "deb [arch=amd64 signed-by=/etc/apt/keyrings/docker.gpg] https://download.docker.com/linux/ubuntu jammy stable" > /etc/apt/sources.list.d/docker.list
+ sh -c apt-get update -qq >/dev/null
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq --no-install-recommends docker-ce docker-ce-cli containerd.io docker-compose-plugin
+ sh -c docker scan-plugin >/dev/null
+ version gte 20.10
+ [ -z ]
+ return 0
+ sh -c DEBIAN_FRONTEND=noninteractive apt-get install -y -qq docker-ce-rootless-extras >/dev/null
+ sh -c docker version
Client: Docker Engine - Community
Version: 20.10.17
API version: 1.41
Go version: go1.17.11
Git commit: 100c701
Built: Mon Jun 6 23:02:46 2022
OS/Arch: linux/amd64
Context: default
Experimental: true

Server: Docker Engine - Community
Engine:
Version: 20.10.17
API version: 1.41 (minimum version 1.12)
Go version: go1.17.11
Git commit: a89b842
Built: Mon Jun 6 23:00:51 2022
OS/Arch: linux/amd64
Experimental: false
containerd:
Version: 1.6.8
GitCommit: 9cd3357b7fd7218e4aec3eae239db1f68a5a6ec6
runc:
Version: 1.1.4
GitCommit: v1.1.4-0-g5fd4c4d
docker-init:
Version: 0.19.0
GitCommit: de40ad0
```

Step 6: Enter command ‘docker –version’ to see current docker version.

```
Remote monitoring
Follow terminal folder

root@ip-172-31-27-5:/home/ubuntu# docker --version
Docker version 20.10.17, build 100c701
root@ip-172-31-27-5:/home/ubuntu#
```

Step 7: Enter command ‘docker images’ to see installed images. At the beginning, there will be no images in the repository. Run command ‘docker run hello-world’ which will pull a hello-world image and run it.

```
Remote monitoring
Follow terminal folder

root@ip-172-31-27-5:/home/ubuntu# docker images
REPOSITORY TAG IMAGE ID CREATED SIZE
root@ip-172-31-27-5:/home/ubuntu#
```

```
ec2-54-227-222-163.compute-1.amazonaws.com (ubuntu)
Terminal Sessions View Xserver Tools Games Settings Macros Help
Session Servers Tools Games Sessions View Split MultiExec Tunneling Packages Settings Help
Quick connect...
/home/ubuntu/
Name
..
.cache
.ssh
.bash_logout
.bashrc
.profile
.xauthority
Remote monitoring
Follow terminal folder

root@ip-172-31-27-5:/home/ubuntu# docker run hello-world
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)
3. The Docker daemon created a new container from that image which runs 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 your terminal.

To try something more ambitious, you can run an Ubuntu container with:
$ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
https://hub.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/get-started/

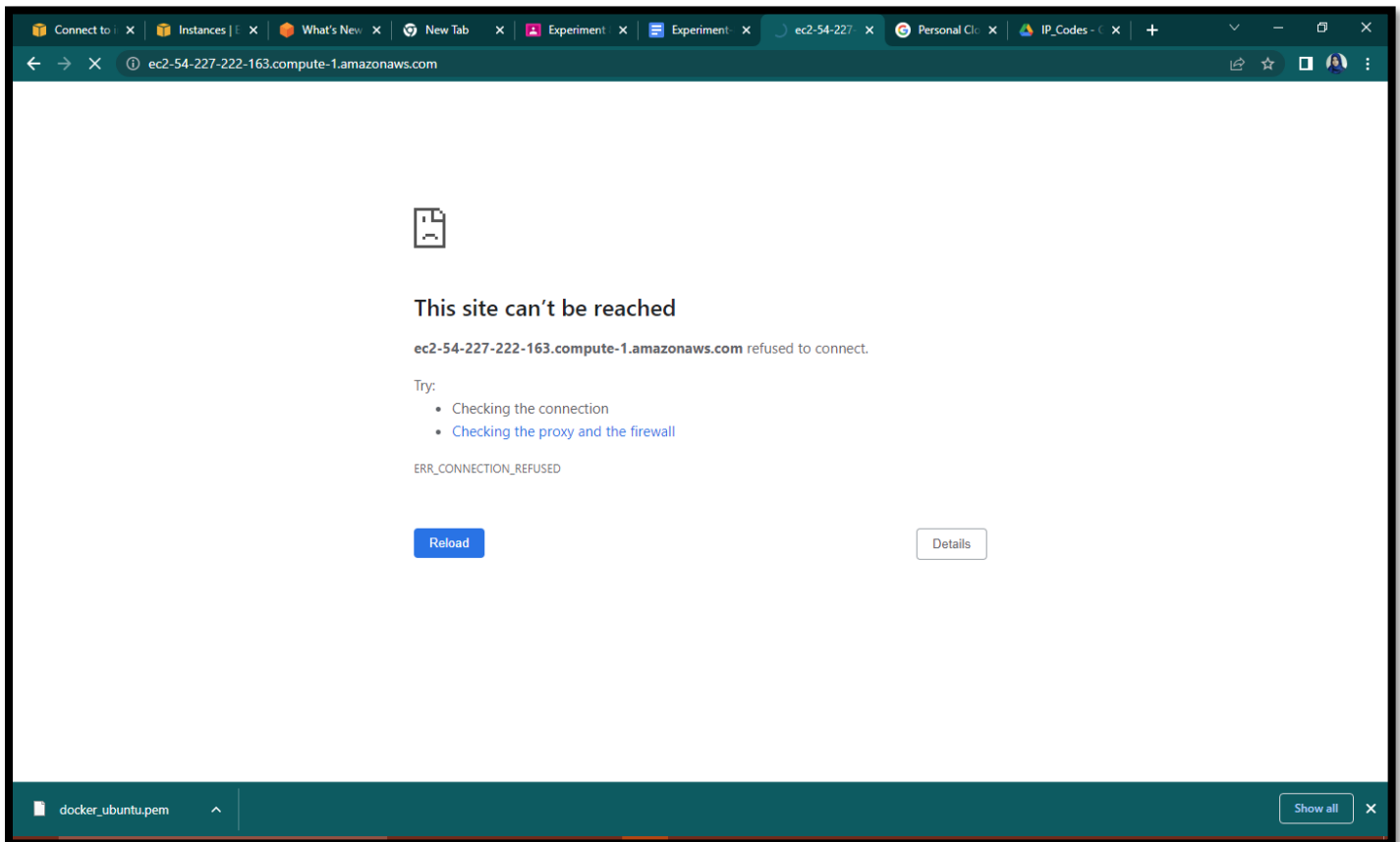
root@ip-172-31-27-5:/home/ubuntu#
```

Step 8: Now, run ‘docker images’ again, the repository will have an image named ‘hello world’.

```
root@ip-172-31-27-5:/home/ubuntu# docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
hello-world   latest    feb5d9fea6a5  11 months ago  13.3kB
root@ip-172-31-27-5:/home/ubuntu#
```

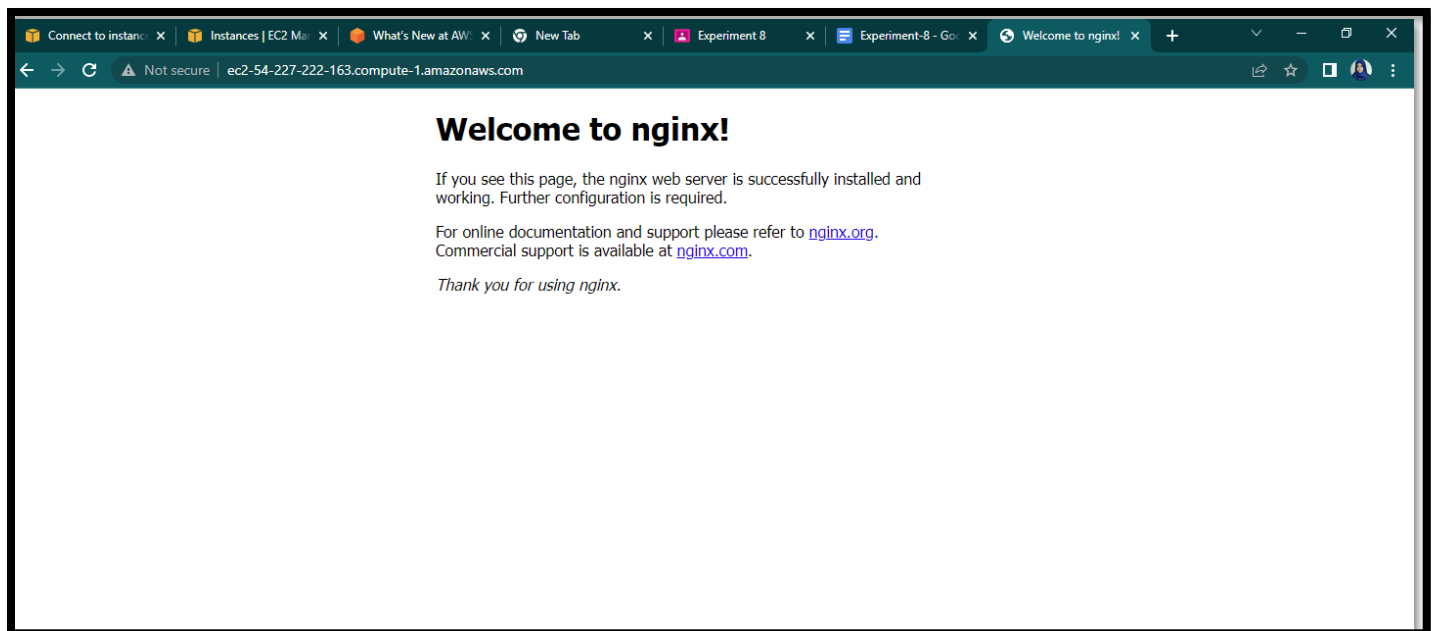
Step 9: Run command ‘docker run -it -p 80:80 --name (name of webpage) nginx bash’. Now, to launch the nginx web server, copy the IPV4 address from the EC2 instance details and paste it into a web browser.

```
root@ip-172-31-27-5:/home/ubuntu# docker run -it -p 80:80 --name mhss-webpage nginx bash
Unable to find image 'nginx:latest' locally
latest: Pulling from library/nginx
7a6db449b51b: Pull complete
ca1981974b58: Pull complete
d4019c921e20: Pull complete
7cb804d746d4: Pull complete
e7a561826262: Pull complete
7247f6e5c182: Pull complete
Digest: sha256:b95a99feebf7797479e0c5eb5ec0bdfa5d9f504bc94da550c2f58e839ea6914f
Status: Downloaded newer image for nginx:latest
root@2a15b51a6b78:/#
```

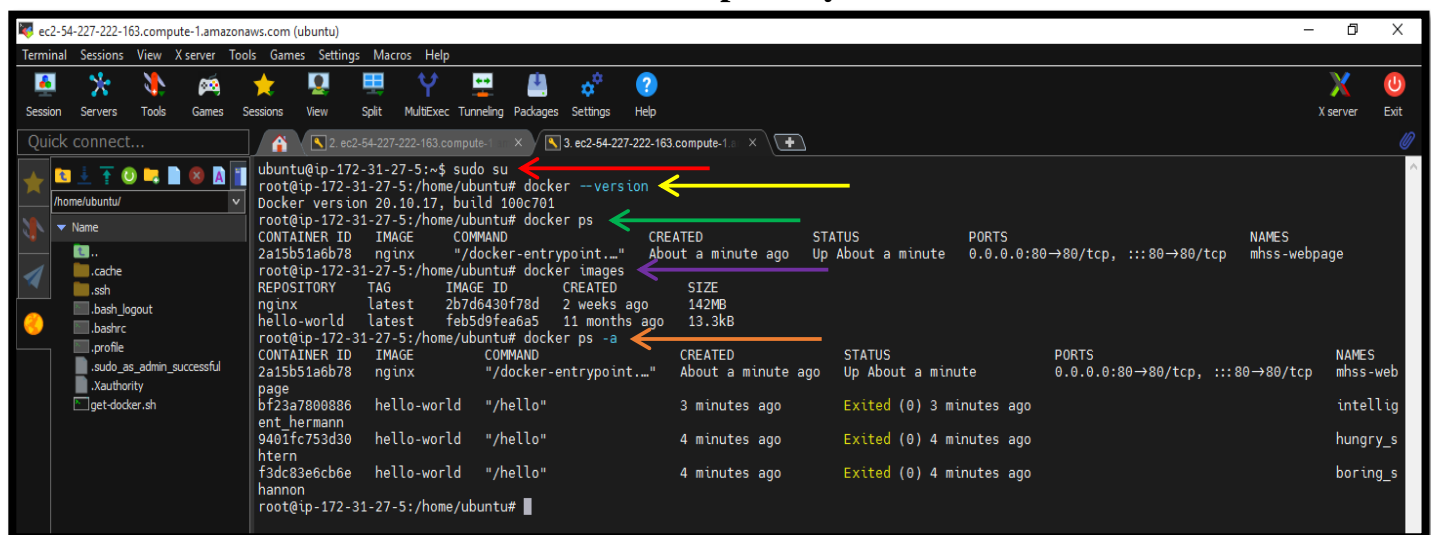


Step 10: Within the container use command ‘service nginx start’ to deploy the web server. After deploying the web server, the web page will be visible without any errors.

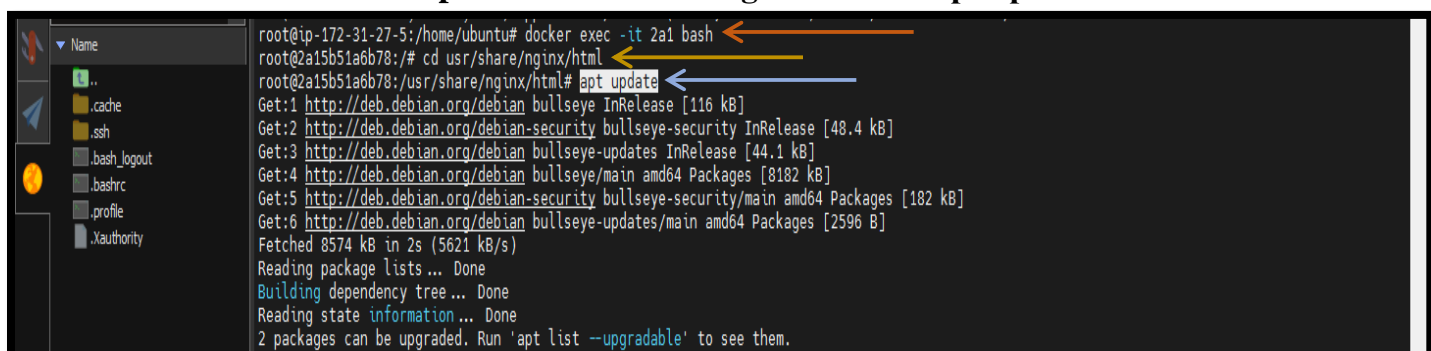
```
root@2a15b51a6b78:/# service nginx start
2022/09/08 06:52:09 [notice] 12#12: using the "epoll" event method
2022/09/08 06:52:09 [notice] 12#12: nginx/1.23.1
2022/09/08 06:52:09 [notice] 12#12: built by gcc 10.2.1 20210110 (Debian 10.2.1-6)
2022/09/08 06:52:09 [notice] 12#12: OS: Linux 5.15.0-1011-aws
2022/09/08 06:52:09 [notice] 12#12: getrlimit(RLIMIT_NOFILE): 1048576:1048576
2022/09/08 06:52:09 [notice] 13#13: start worker processes
2022/09/08 06:52:09 [notice] 13#13: start worker process 14
root@2a15b51a6b78:/#
```



Step 11: To exit the container, use ‘CTRL+P+Q’. Create a duplicate tab and take the root user rights. Run commands: ‘docker ps’, ‘docker ps -a’ Also run ‘docker images’ to check the images in the repository



Step 12: To make changes to a file within a container use command ‘docker exec -it (container id) bash’.
Now within the container navigate to the html directory using command: ‘cd usr/share/nginx/html’
Then update container using command ‘apt update’



Step 13: Now use command: 'apt install nano' to install nano text editor.

```
root@2a15b51a6b78:/usr/share/nginx/html# apt install nano
Reading package lists ... Done
Building dependency tree ... Done
Reading state information ... Done
The following additional packages will be installed:
  libgpm2 libncursesw6
Suggested packages:
  gpm hunspell
The following NEW packages will be installed:
  libgpm2 libncursesw6 nano
0 upgraded, 3 newly installed, 0 to remove and 2 not upgraded.
Need to get 824 kB of archives.
After this operation, 3087 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://deb.debian.org/debian bullseye/main amd64 libncursesw6 amd64 6.2+20201114-2 [132 kB]
Get:2 http://deb.debian.org/debian bullseye/main amd64 nano amd64 5.4-2+deb11u1 [656 kB]
Get:3 http://deb.debian.org/debian bullseye/main amd64 libgpm2 amd64 1.20.7-8 [35.6 kB]
Fetched 824 kB in 0s (24.6 MB/s)
debconf: delaying package configuration, since apt-utils is not installed
Selecting previously unselected package libncursesw6:amd64.
(Reading database ... 7823 files and directories currently installed.)
Preparing to unpack .../libncursesw6_6.2+20201114-2_amd64.deb ...
Unpacking libncursesw6:amd64 (6.2+20201114-2) ...
Selecting previously unselected package nano.
Preparing to unpack .../nano_5.4-2+deb11u1_amd64.deb ...
Unpacking nano (5.4-2+deb11u1) ...
Selecting previously unselected package libgpm2:amd64.
Preparing to unpack .../libgpm2_1.20.7-8_amd64.deb ...
Unpacking libgpm2:amd64 (1.20.7-8) ...
Setting up libgpm2:amd64 (1.20.7-8) ...
```

Step 14: Now move the original nginx index as a backup so you can create your own html index file using the command: 'mv index.html index.html.backup' Then, open nano text editor using command: 'nano index.html'

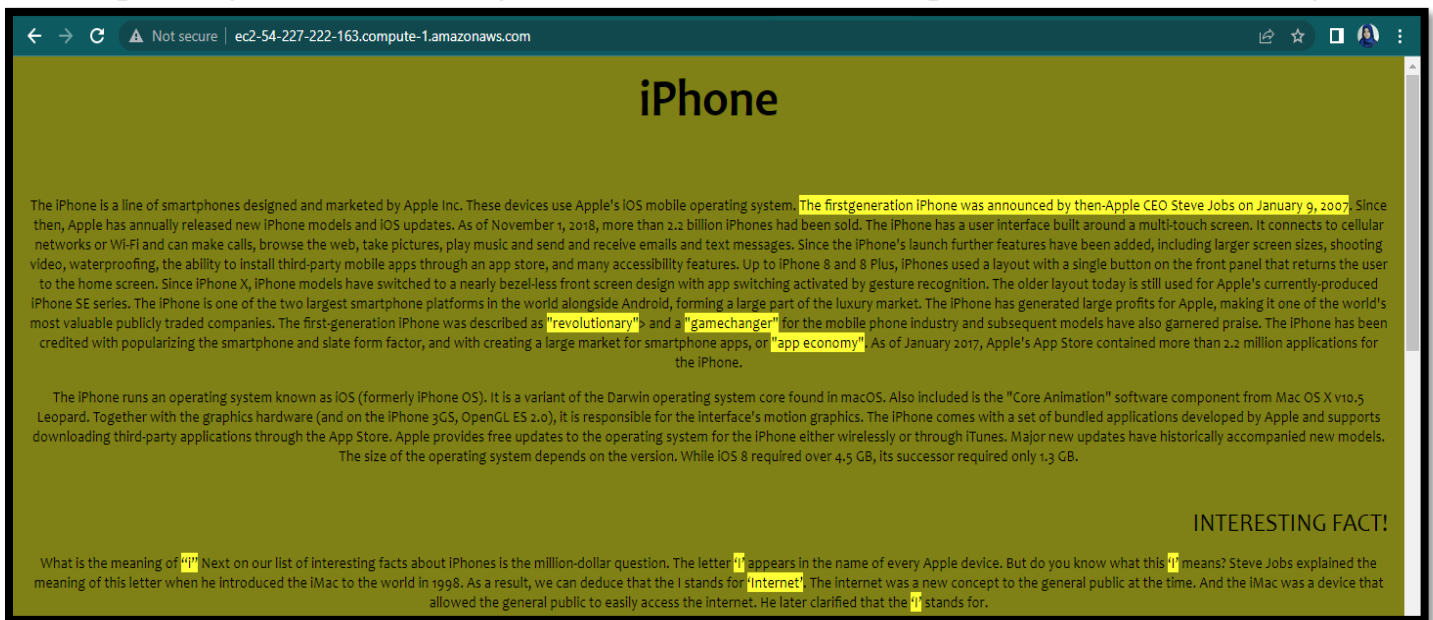
```
root@2a15b51a6b78:/usr/share/nginx/html# mv index.html index.html.backup
root@2a15b51a6b78:/usr/share/nginx/html# nano index.html
root@2a15b51a6b78:/usr/share/nginx/html#
```

Step 15: Write an html code of your choice → CTRL+O → ENTER → CTRL X. This will save the code.

```
GNU nano 5.4 index.html *
</p>
</section>
</article>
<br>
<br>
<aside>
<p style="font-size: xx-large; text-align: right; font-family: Candara;">00
YOU KNOW ?? </p>
<p style="font-family: Candara; text-align: center;">
On January 9, 2007, Steve Jobs announced iPhone at the Macworld convention and received huge media attention. Finally later that year, on June 29
<mark><i>But do you know that the present iPhone was earlier named as
^ ^ | Purple ^ ^ ? But this name never came into existence as the purple name was

already taken by another company.</i></mark> After that, the iPhone name was finalized. However,
Apple developers still called their development section as <mark><i> ^ ^ ^ Purple
Drom ^ ^ y.</i></mark>
</p>
</aside>
<details>
<summary>See also</summary>
<ul>
<li><a href="https://www.amazon.in/macbook/s?k=macbook">Mac Book</a></li>
<li><a href="https://www.amazon.in/apple-watch/s?k=apple+watch">Apple
Watches</a></li>
<li><a href="https://www.amazon.in/ipad/s?k=ipad">Ipad</a></li>
<li><a href="https://www.amazon.in/Apple-MX532ZMA-New-
AirTag/dp/B0935DN1BN?th=1">AirTag</a></li>
</ul>
</details>
<br>
<footer>
<p>Submitted By: Sayyed Ayesha <br> Practical: IP Lab</p>
</footer>
<br>
</main>
</body>
</html>
```


Step 16: Again, refresh the nginx web browser, it will be updated with the new changes.

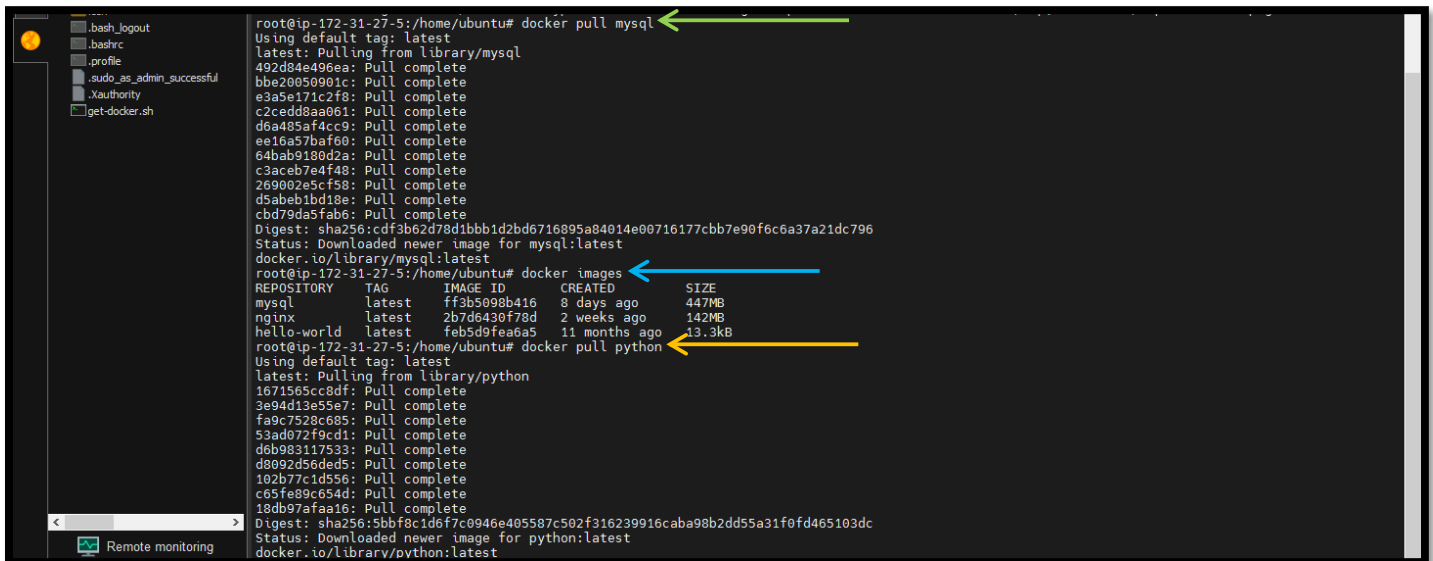


Step 17: Create another duplicate tab and get the root user access.

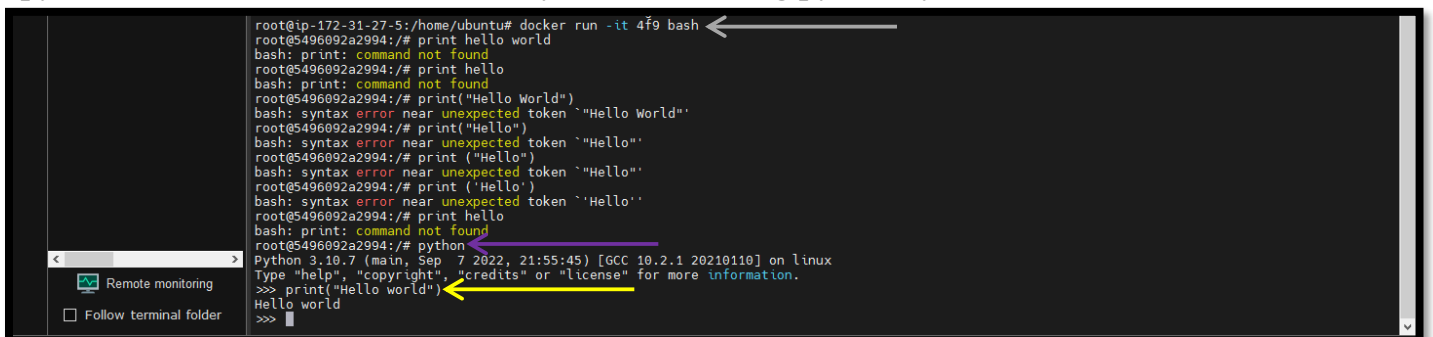
Run the command: **'docker pull mysql'** to install a mysql image.

Then, run **'docker images'** to check the image upload in the repository.

Subsequently, run commands such as **'docker pull python'** and **'docker pull mongo'** to install the respective images. Check the repository for the upload.



Step 18: Now, to enter the python container use command: **'docker run -it (image id) bash'** and type **'python'** to enter the shell. Execute any command using python syntax such as **print("Hello World")**



Step 19: Run the command ‘`docker ps`’ to check the number of containers and their ID’s.

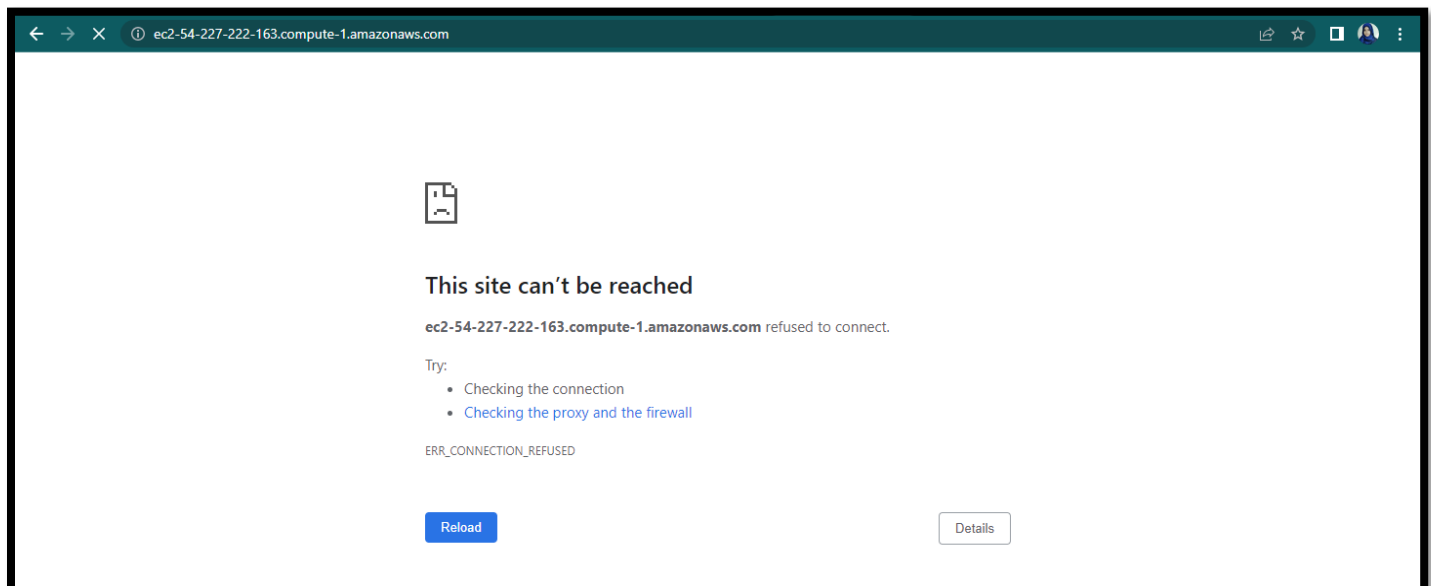
To stop a container use command: ‘`docker stop (container id)`’

You can use the command ‘`docker ps -a`’ to check the status of the container.

```
root@ip-172-31-27-5:/home/ubuntu# docker ps
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS                               NAMES
5496092a2994   4f9      "bash"    12 minutes ago   Up 12 minutes   0.0.0.0:80→80/tcp, :::80→80/tcp   youthful_mcclintock
2a15b51a6b78   nginx    "/docker-entrypoint..." 27 minutes ago   Up 27 minutes                               mhss-webpage

root@ip-172-31-27-5:/home/ubuntu# docker stop 549
549
root@ip-172-31-27-5:/home/ubuntu# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS                               NAMES
5496092a2994   4f9      "bash"    13 minutes ago   Exited (137) 5 seconds ago                               youthful_m
a319bc56dbaf   python    "bash"    14 minutes ago   Exited (0) 14 minutes ago                               keen_varah
2a15b51a6b78   nginx    "/docker-entrypoint..." 28 minutes ago   Up 28 minutes   0.0.0.0:80→80/tcp, :::80→80/tcp   mhss-webpa
bf23a7800886   hello-world "/hello"   30 minutes ago   Exited (0) 30 minutes ago                               intelligen
t_hermann
9401fc753d30   hello-world "/hello"   31 minutes ago   Exited (0) 31 minutes ago                               hungry_sht
ern
f3dc83e6cb6e   hello-world "/hello"   31 minutes ago   Exited (0) 31 minutes ago                               boring_sha
nnon

root@ip-172-31-27-5:/home/ubuntu# docker stop 2a1
2a1
root@ip-172-31-27-5:/home/ubuntu# docker ps -a
CONTAINER ID   IMAGE     COMMAND   CREATED   STATUS    PORTS                               NAMES
5496092a2994   4f9      "bash"    14 minutes ago   Exited (137) 41 seconds ago                               youthful_mcclintock
a319bc56dbaf   python    "bash"    15 minutes ago   Exited (0) 15 minutes ago                               keen_varahamihira
2a15b51a6b78   nginx    "/docker-entrypoint..." 29 minutes ago   Exited (137) 6 seconds ago                               mhss-webpage
bf23a7800886   hello-world "/hello"   30 minutes ago   Exited (0) 30 minutes ago                               intelligent_hermann
9401fc753d30   hello-world "/hello"   31 minutes ago   Exited (0) 31 minutes ago                               hungry_shtern
f3dc83e6cb6e   hello-world "/hello"   32 minutes ago   Exited (0) 32 minutes ago                               boring_shannon
nnon
```



Step 20: Quit MobaXterm and then delete your EC2 instance.

