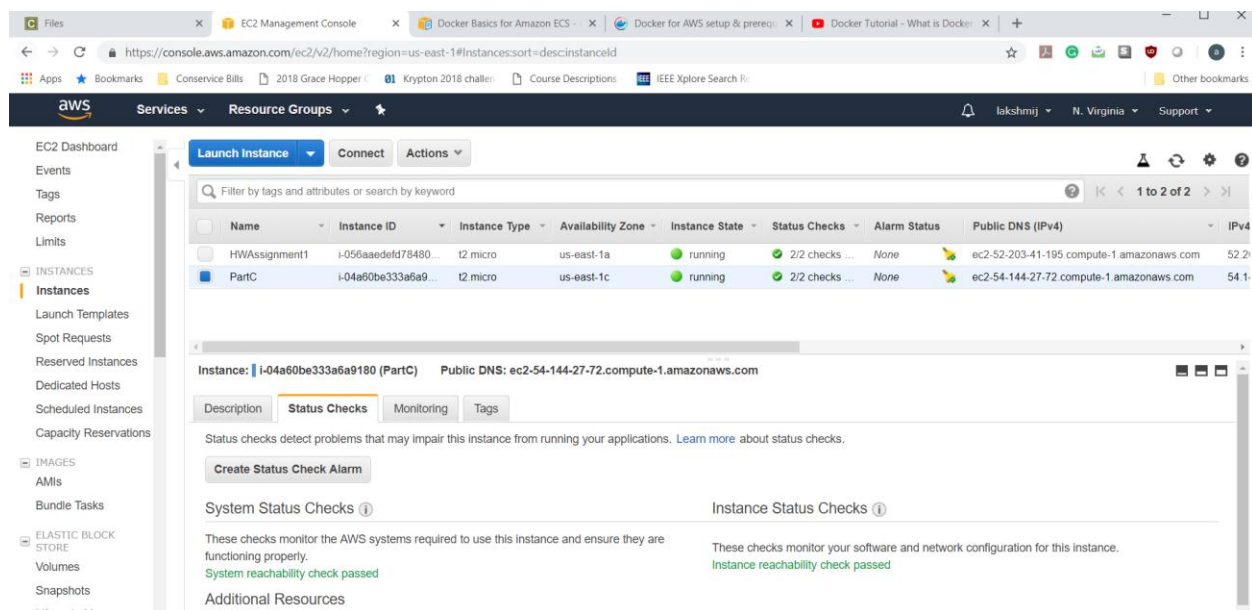
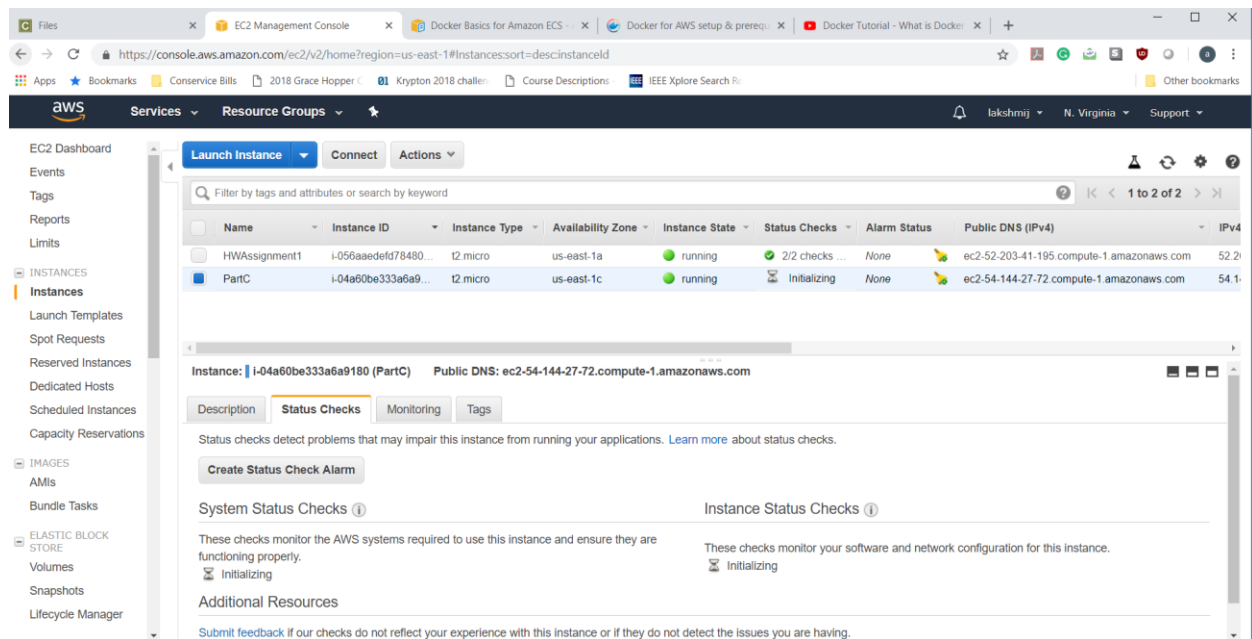


C. Create a docker image for LAMPs web server – use the EC2 environment created in step A

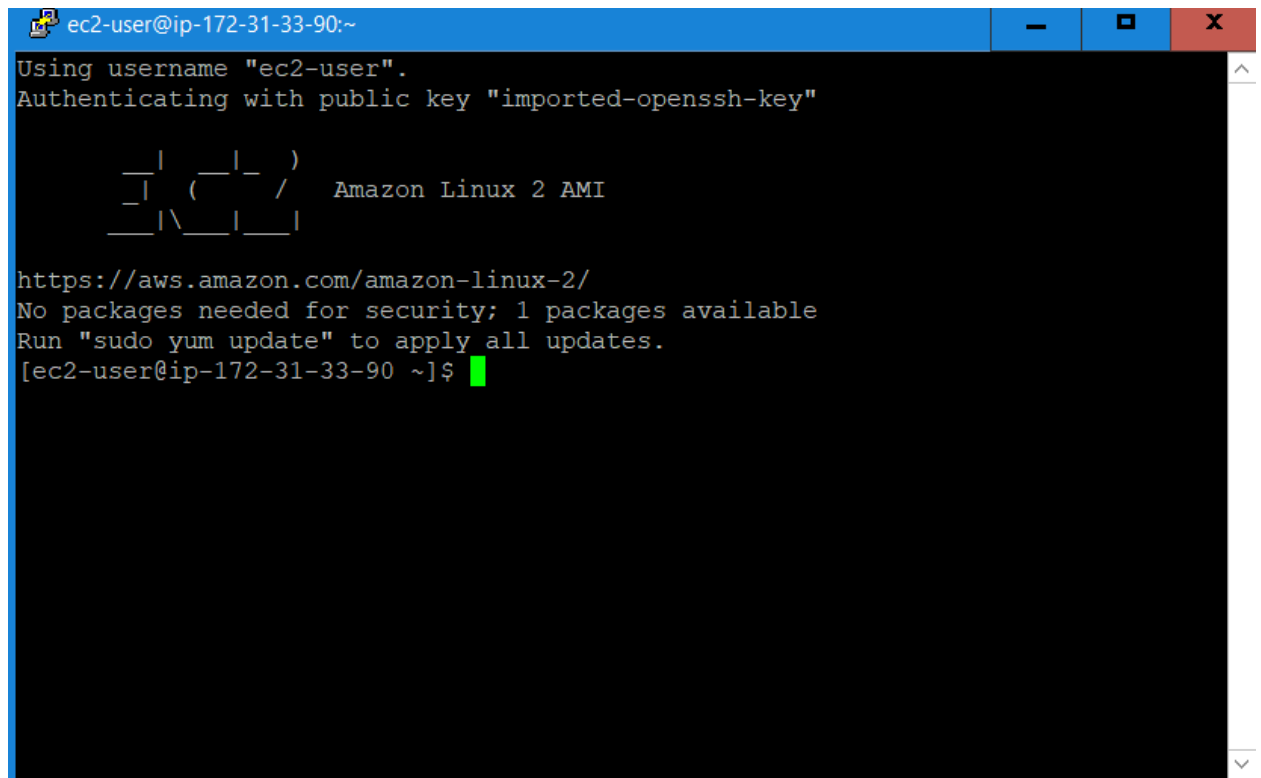
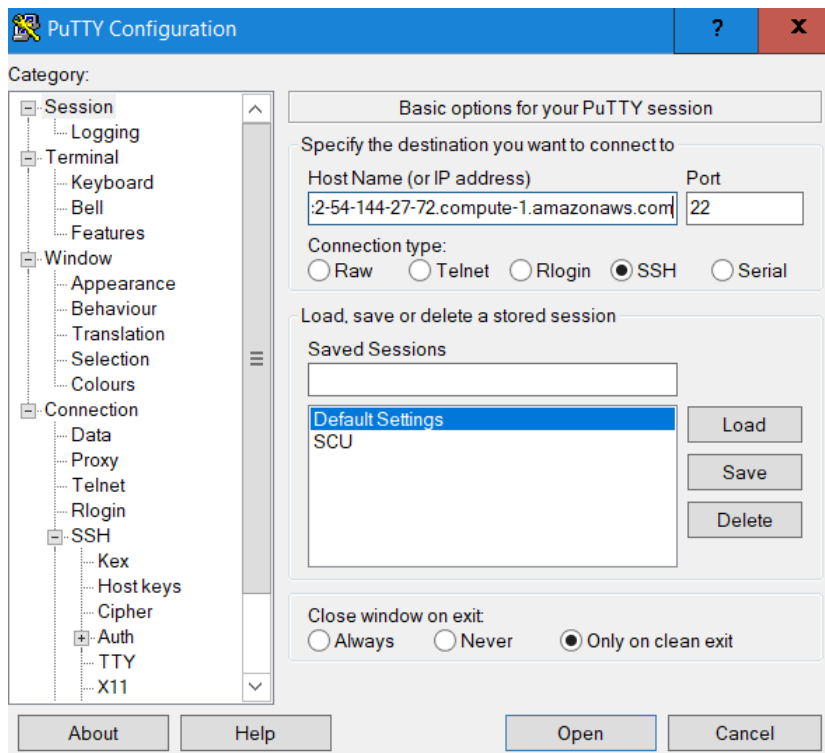
Steps:

- 1) Create an instance as explained in

http://docs.aws.amazon.com/AWSEC2/latest/UserGuide/EC2_GetStarted.html



- 2) Connect to instance through putty.



- 3) Run the command **sudo yum update -y** to update all the packages.

```
[ec2-user@ip-172-31-33-90 ~]$ sudo yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 2.4 kB 00:00
Resolving Dependencies
--> Running transaction check
---> Package amazon-ssm-agent.x86_64 0:2.3.274.0-1.amzn2 will be updated
---> Package amazon-ssm-agent.x86_64 0:2.3.372.0-1.amzn2 will be an update
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Updating:
amazon-ssm-agent x86_64 2.3.372.0-1.amzn2 amzn2-core 15 M
Transaction Summary
=====
Upgrade 1 Package

Total download size: 15 M
Downloading packages:
Delta RPMs disabled because /usr/bin/applydeltarpm not installed.
amazon-ssm-agent-2.3.372.0-1.amzn2.x86_64.rpm | 15 MB 00:00
Running transaction check
Running transaction test
Transaction test succeeded
Running transaction
  Updating : amazon-ssm-agent-2.3.372.0-1.amzn2.x86_64 1/2
  Cleanup : amazon-ssm-agent-2.3.274.0-1.amzn2.x86_64 2/2
  Verifying : amazon-ssm-agent-2.3.372.0-1.amzn2.x86_64 1/2
  Verifying : amazon-ssm-agent-2.3.274.0-1.amzn2.x86_64 2/2

Updated:
  amazon-ssm-agent.x86_64 0:2.3.372.0-1.amzn2

Complete!
[ec2-user@ip-172-31-33-90 ~]$
```

- 4) Install the most recent Docker Community Edition package using **sudo amazon-linux-extras install docker**

```
ec2-user@ip-172-31-33-90~
[ec2-user@ip-172-31-33-90 ~]$ sudo amazon-linux-extras install docker
Installing docker
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Cleaning repos: amzn2-core amzn2extra-docker
10 metadata files removed
4 sqlite files removed
0 metadata files removed
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core | 2.4 kB 00:00:00
amzn2extra-docker | 1.3 kB 00:00:00
(1/4): amzn2-core/2/x86_64/updateinfo | 80 kB 00:00:00
(2/4): amzn2-core/2/x86_64/group.gz | 2.4 kB 00:00:00
(3/4): amzn2extra-docker/2/x86_64/primary_db | 37 kB 00:00:00
(4/4): amzn2-core/2/x86_64/primary_db | 25 MB 00:00:00
Resolving Dependencies
--> Running transaction check
--> Package docker.x86_64 0:18.06.1ce-5.amzn2 will be installed
--> Processing Dependency: pigz for package: docker-18.06.1ce-5.amzn2.x86_64
--> Processing Dependency: libcgrouper for package: docker-18.06.1ce-5.amzn2.x86_64
--> Processing Dependency: libltdl.so.7()(64bit) for package: docker-18.06.1ce-5.amzn2.x86_64
--> Running transaction check
--> Package libcgrouper.x86_64 0:0.41-15.amzn2 will be installed
--> Package libltdl-ltdl.x86_64 0:2.4.2-22.2.amzn2.0.2 will be installed
--> Package pigz.x86_64 0:2.3.4-1.amzn2.0.1 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

=====
Package Arch Version Repository Size
=====
Installing:
docker x86_64 18.06.1ce-5.amzn2 amzn2extra-docker 37 M
Installing for dependencies:
libcgrouper x86_64 0.41-15.amzn2 amzn2-core 65 k
libltdl-ltdl x86_64 2.4.2-22.2.amzn2.0.2 amzn2-core 49 k
pigz x86_64 2.3.4-1.amzn2.0.1 amzn2-core 81 k
Transaction Summary
-----
Install 1 Package (+3 Dependent packages)

Total download size: 37 M
Installed size: 151 M
Is this ok [y/d/N]: y
Downloading packages:
(1/4): libcgrouper-0.41-15.amzn2.x86_64.rpm | 65 kB 00:00:00
(2/4): libltdl-ltdl-2.4.2-22.2.amzn2.0.2.x86_64.rpm | 49 kB 00:00:00
(3/4): pigz-2.3.4-1.amzn2.0.1.x86_64.rpm | 81 kB 00:00:00
(4/4): docker-18.06.1ce-5.amzn2.x86_64.rpm | 37 MB 00:00:00
```

```

ec2-user@ip-172-31-33-90:~
Transaction test succeeded
Running transaction
  Installing : libtool-ltdl-2.4.2-22.2.amzn2.0.2.x86_64
  Installing : libcgroup-0.41-15.amzn2.x86_64
  Installing : pigz-2.3.4-1.amzn2.0.1.x86_64
  Installing : docker-18.06.1ce-5.amzn2.x86_64
  Verifying  : pigz-2.3.4-1.amzn2.0.1.x86_64
  Verifying  : docker-18.06.1ce-5.amzn2.x86_64
  Verifying  : libcgroup-0.41-15.amzn2.x86_64
  Verifying  : libtool-ltdl-2.4.2-22.2.amzn2.0.2.x86_64

Installed:
  docker.x86_64 0:18.06.1ce-5.amzn2

Dependency Installed:
  libcgroup.x86_64 0:0.41-15.amzn2                libtool-ltdl.x86_64 0:2.4.2-22.2.amzn2.0.2      pigz.x86_64 0:2.3.4-1.amzn2.0.1

Complete!
 0  ansible2                available [ =2.4.2 =2.4.6 ]
 2  httpd_modules           available [ =1.0 ]
 3  memcached1.5            available [ =1.5.1 ]
 4  nginx1.12               available [ =1.12.2 ]
 5  postgresql9.6           available [ =9.6.6 =9.6.8 ]
 6  postgresql10            available [ =10 ]
 8  redis4.0                available [ =4.0.5 =4.0.10 ]
 9  R3.4                    available [ =3.4.3 ]
10  rust1                   available \
    [ =1.22.1 =1.26.0 =1.26.1 =1.27.2 =1.31.0 ]
11  vim                     available [ =8.0 ]
13  ruby2.4                 available [ =2.4.2 =2.4.4 ]
15  php7.2                  available \
    [ =7.2.0 =7.2.4 =7.2.5 =7.2.8 =7.2.11 ]
16  php7.1                  available [ =7.1.22 ]
17  lamp-mariadb10.2-php7.2 available \
    [ =10.2.10_7.2.0 =10.2.10_7.2.4 =10.2.10_7.2.5
      =10.2.10_7.2.8 =10.2.10_7.2.11 ]
18  libreoffice             available [ =5.0.6.2_15 =5.3.6.1 ]
19  gimp                    available [ =2.8.22 ]
20  docker=latest           enabled \
    [ =17.12.1 =18.03.1 =18.06.1 ]
21  mate-desktop1.x         available [ =1.19.0 =1.20.0 ]
22  GraphicsMagick1.3       available [ =1.3.29 ]
23  tomcat8.5               available [ =8.5.31 =8.5.32 ]
24  epel                    available [ =7.11 ]
25  testing                 available [ =1.0 ]
26  ecs                     available [ =stable ]
27  corretto8              available [ =1.8.0_192 ]
28  firecracker             available [ =0.11 ]
29  golang1.11              available [ =1.11.3 ]

[ec2-user@ip-172-31-33-90 ~]$

```

- 5) Start the Docker service by using the command **sudo service docker start**
- 6) Add the **ec2-user** to the docker group so you can execute Docker commands without using sudo.

```

[ec2-user@ip-172-31-33-90 ~]$ sudo service docker start
Redirecting to /bin/systemctl start docker.service
[ec2-user@ip-172-31-33-90 ~]$ sudo usermod -a -G docker ec2-user
[ec2-user@ip-172-31-33-90 ~]$

```

- 7) Verify that the ec2-user can run Docker commands without sudo using command docker info.

```

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-33-90 ~]$ docker info
Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
Images: 0
Server Version: 18.06.1-ce
Storage Driver: overlay2
  Backing Filesystem: xfs
  Supports d_type: true
  Native Overlay Diff: true
Logging Driver: json-file
Cgroup Driver: cgroupfs
Plugins:
  Volume: local
  Network: bridge host macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file logentries splunk syslog
Swarm: inactive
Runtimes: runc
Default Runtime: runc
Init Binary: docker-init
containerd version: 468a545b9edcd5932818eb9de8e72413e616e86e
runc version: 69663f0bd4b60df09991c08812a60108003fa340
init version: fec3683
Security Options:
  seccomp
    Profile: default
Kernel Version: 4.14.88-88.76.amzn2.x86_64
Operating System: Amazon Linux 2
OSType: linux
Architecture: x86_64
CPUs: 1
Total Memory: 985.7MiB
Name: ip-172-31-33-90.ec2.internal
ID: 4B5V:XJ5Z:KUMI:GFDW:AUAO:6FQW:3SAJ:3QKC:AQQP:DZRT:TYQ6:RHCM
Docker Root Dir: /var/lib/docker
Debug Mode (client): false
Debug Mode (server): false
Registry: https://index.docker.io/v1/
Labels:
Experimental: false
Insecure Registries:
  127.0.0.0/8
Live Restore Enabled: false
[ec2-user@ip-172-31-33-90 ~]$

```

Create a Docker Image

- 8) Create a file called **Dockerfile**. A **Dockerfile** is a manifest that describes the base image to use for your Docker image and what you want installed and running on it.

```

Live Restore Enabled: false

[ec2-user@ip-172-31-33-90 ~]$ touch Dockerfile
[ec2-user@ip-172-31-33-90 ~]$

```

- 9) Edit the **Dockerfile** created and add the following content.

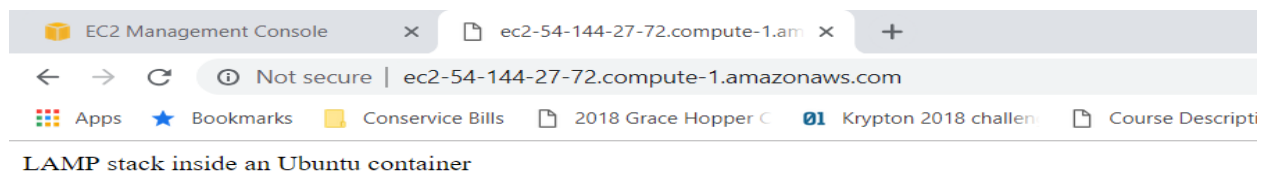
```
ec2-user@ip-172-31-33-90~  
FROM ubuntu:16.04  
  
#install dependencies  
RUN apt-get update  
RUN apt-get -y install apache2  
RUN apt-get -y install mariadb-server mariadb-client  
RUN apt-get -y install software-properties-common  
RUN apt-get install -y language-pack-en-base  
RUN apt-get update  
RUN LC_ALL=en_US.UTF-8 add-apt-repository ppa:ondrej/php  
RUN apt-get install -y --allow-unauthenticated --no-install-recommends php7.2 libapache2-mod-php7.2 php7.2-common php7.2-mbstring php7.2-xmlrpc php7.2-soap php7.2-gd php7.2-xml php7.2-intl  
php7.2-mysql php7.2-cli php7.2-zip php7.2-curl  
  
# output phpinfo  
RUN echo "<?php phpinfo(); ?>" > /var/www/html/index.html  
  
#write message to display in browser  
RUN echo 'LAMP stack inside an Ubuntu container' > /var/www/html/index.html  
  
EXPOSE 80  
  
CMD apache2l -D FOREGROUND
```

10) Build the Docker image from your Dockerfile using **docker build -t test**.

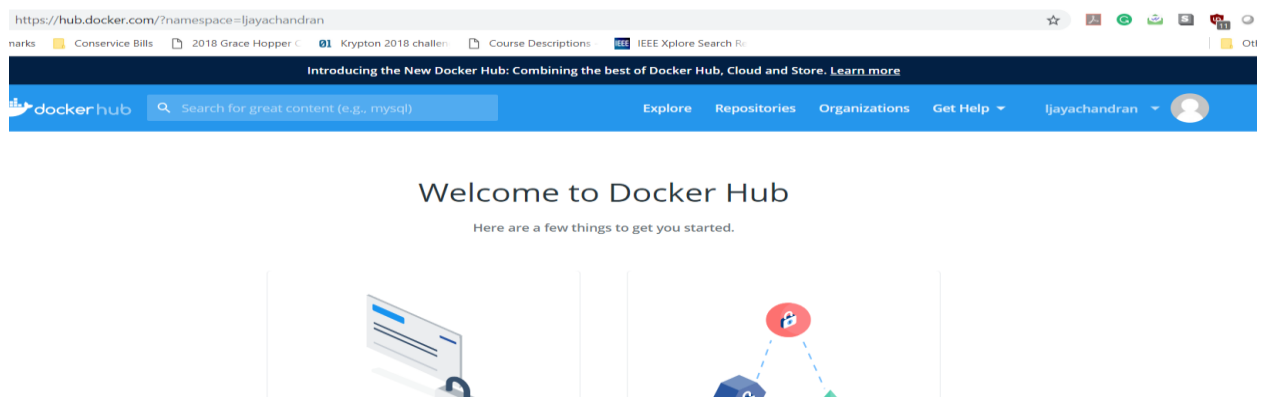
11) Run docker images to verify that the image was created correctly using the command **docker images --filter reference=test**

```
11/ history  
[ec2-user@ip-172-31-33-90 ~]$ docker images --filter reference=test  
REPOSITORY          TAG             IMAGE ID        CREATED         SIZE  
test                 latest          1cc32c3e93b4   About an hour ago  634MB  
[ec2-user@ip-172-31-33-90 ~]$
```

12) Run the newly built image. The -p 80:80 option maps the exposed port 80 on the container to port 80 on the host system using the command **docker run -p 80:80 test**



13) Create an account in docker hub.



14) Login to the Docker hub account.

```
[ec2-user@ip-172-31-33-90 ~]$ docker login
Authenticating with existing credentials...
WARNING! Your password will be stored unencrypted in /home/ec2-user/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
[ec2-user@ip-172-31-33-90 ~]$
```

15) To push an image, check the image ID.

```
[ec2-user@ip-172-31-33-90 ~]$ docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
test	latest	1cc32c3e93b4	About an hour ago	634MB
ljayachandran/helloworld	latest	ea5184dec52c	2 hours ago	240MB
ljayachandran/assigmenttest	latest	db04323f094f	2 hours ago	364MB
ubuntu	14.04	ba4369e799cb	3 hours ago	364MB
<none>	<none>	39b54ffe3ad4	5 hours ago	201MB
aws_account_id.dkr.ecr.us-east-1.amazonaws.com/hello-repository	latest	1dda71961bd8	7 hours ago	240MB
hello-world	latest	1dda71961bd8	7 hours ago	240MB
ubuntu	16.04	b0ef3016420a	2 weeks ago	117MB
ubuntu	<none>	7e4b16ae8b23	2 weeks ago	188MB
ubuntu	latest	1d9c17228a9e	2 weeks ago	86.7MB

```
[ec2-user@ip-172-31-33-90 ~]$
```

16) After tagging your image push the image to the repository using the command **docker push ljayachandran/test**.


```
6eed1e3f94c4: Pushed
7ac7a45d84d2: Pushed
6c10dc340fef: Pushing [=====>] 42.33MB
ccc805a8970b: Pushed
af459c0c373e: Pushed
c4cd0edd41aa: Pushing [=====>] 6.043MB
495b42267b96: Pushing [=====>] 19.38MB/21.58MB
51311311b3d6: Pushing [==>] 3.24MB/62.45MB
8dd801c9d3c7: Waiting
aba74bf06399: Waiting
753b6ea51488: Waiting
428c1ba11354: Waiting
b097f5edab7b: Waiting
27712caf4371: Waiting
8241afc74c6f: Waiting
```

```
[ec2-user@ip-172-31-33-90 ~]$ docker push ljayachandran/test
The push refers to repository [docker.io/ljayachandran/test]
6eed1e3f94c4: Pushed
7ac7a45d84d2: Pushed
6c10dc340fef: Pushed
ccc805a8970b: Pushed
af459c0c373e: Pushed
c4cd0edd41aa: Pushed
495b42267b96: Pushed
51311311b3d6: Pushed
8dd801c9d3c7: Pushed
aba74bf06399: Mounted from ljayachandran/helloworld
753b6ea51488: Mounted from ljayachandran/helloworld
428c1ba11354: Mounted from ljayachandran/helloworld
b097f5edab7b: Mounted from ljayachandran/helloworld
27712caf4371: Mounted from ljayachandran/helloworld
8241afc74c6f: Mounted from ljayachandran/helloworld
firsttest: digest: sha256:60e882f03d73fc2997fef3e7a2a050587ee502a8988fad8e42599071500d07e size: 3464
[ec2-user@ip-172-31-33-90 ~]$
```


17) And now if you refresh the repository page of your docker hub account you would see as below.





← → ↻ https://cloud.docker.com/repository/list ☆ [Icons] [Icons] [Icons] [Icons] [Icons]

Apps ★ Bookmarks [Icons] Conserve Bills [Icons] 2018 Grace Hopper [Icons] 01 Krypton 2018 challen [Icons] Course Descriptions [Icons] IEEE Xplore Search [Icons]

 Explore Repositories Organizations Get Help **ljayachandran** [User Avatar]

Repositories Using 0 of 1 private repositories. [Get more](#)

 ljayachandran 🔍 Filter by repository name... [Create Repository +](#)

REPOSITORY	DESCRIPTION	LAST MODIFIED
 ljayachandran / test	<i>This repository does not have a description...</i>	🕒 2 minutes ago
 ljayachandran / helloworld	<i>This repository does not have a description...</i>	🕒 2 hours ago
 ljayachandran / assignmenttest	<i>This repository does not have a description...</i>	🕒 2 hours ago
 ljayachandran / cloudassignment	<i>This repository does not have a description...</i>	

1

Image name: test

Username: ljayachandran