COEN 241 HW-1B

Anmol Vijayvargiya [ID:1284369]
Docker Hub User Name: anmolvj
Docker Hub image name: ubuntu

Steps for setting up prerequisites for docker engine and then installing it as well as testing it works:

1. SSH into the virtual machine created in Part B of this homework. Now type following command to update package information:
   $ sudo apt-get update

2. Use following command to ensure APT can work with https method-
   $ sudo apt-get install apt-transport-https

3. Use the following command to install CA certificates-
   $ sudo apt-get install ca-certificates

4. Add the new GPG key using the following command as listed on the docker website to create security for exchange of files with sensitive data-
   *$ sudo apt-key adv *
   *--keyserver hkp://ha.pool.sks-keyservers.net:80 *
   --recv-keys
   58118E89F3A912897C070ADBF76221572C52609D

5. Use following command so that APT can determine where to search for docker packages-
   $ echo "deb https://apt.dockerproject.org/repo ubuntu-trusty main" | sudo tee /etc/apt/sources.list.d/docker.list

6. Use the following command again to update the APT package index-
   $ sudo apt-get update

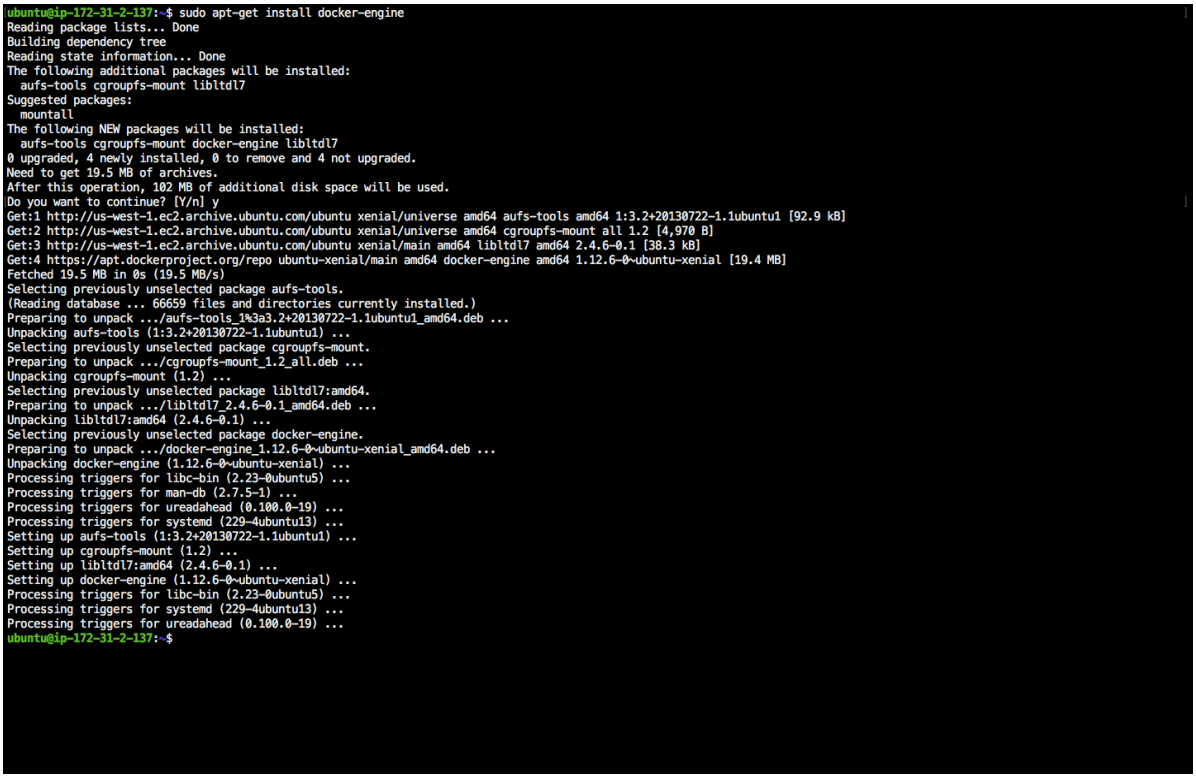
7. Use following command to install cache policy-
   $ apt-cache policy docker-engine

```
ubuntu@ip-172-31-2-137:~$ sudo apt-get install apt-transport-https ca-certificates
Reading package lists... Done
Building dependency tree
Reading state information... Done
ca-certificates is already the newest version (20160104ubuntu1).
apt-transport-https is already the newest version (1.2.18).
0 upgraded, 0 newly installed, 0 to remove and 4 not upgraded.
ubuntu@ip-172-31-2-137:~$ sudo apt-key adv \
>                 --keyserver hkp://ha.pool.sks-keyservers.net:80 \
>                 --recv-keys 58118E89F3A912897C070ADBF76221572C52609D
Executing: /tmp/tmp.Xqlo5bFvzu/gpg.1.sh --keyserver
hkp://ha.pool.sks-keyservers.net:80
--recv-keys
58118E89F3A912897C070ADBF76221572C52609D
gpg: requesting key 2C52609D from hkp server ha.pool.sks-keyservers.net
gpg: key 2C52609D: public key "Docker Release Tool (releasedocker) <docker@docker.com>" imported
gpg: Total number processed: 1
gpg:               imported: 1  (RSA: 1)
ubuntu@ip-172-31-2-137:~$ echo "deb https://apt.dockerproject.org/repo ubuntu-xenial main" | sudo tee /etc/apt/sources.list.d/docker.list
deb https://apt.dockerproject.org/repo ubuntu-xenial main
ubuntu@ip-172-31-2-137:~$ sudo apt-get update
Hit:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu xenial InRelease
Hit:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu xenial-updates InRelease
Hit:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu xenial-backports InRelease
Get:4 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Get:5 https://apt.dockerproject.org/repo ubuntu-xenial InRelease [30.2 kB]
Get:6 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages [3,276 B]
Fetched 136 kB in 0s (167 kB/s)
Reading package lists... Done
ubuntu@ip-172-31-2-137:~$  apt-cache policy docker-engine
docker-engine:
  Installed: (none)
  Candidate: 1.12.6-0~ubuntu-xenial
  Version table:
     1.12.6-0~ubuntu-xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
     1.12.5-0~ubuntu-xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
     1.12.4-0~ubuntu-xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
     1.12.3-0~xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
     1.12.2-0~xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
     1.12.1-0~xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
     1.12.0-0~xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
     1.11.2-0~xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
     1.11.1-0~xenial 500
        500 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 Packages
```

Screenshot 1

8. Use following command to install the "*linux-image-extra-**" kernel packages that will allow us to use "*aufs*" storage driver-
   $ sudo apt-get install linux-image-extra-$(uname -r) linux-image-extra-virtual

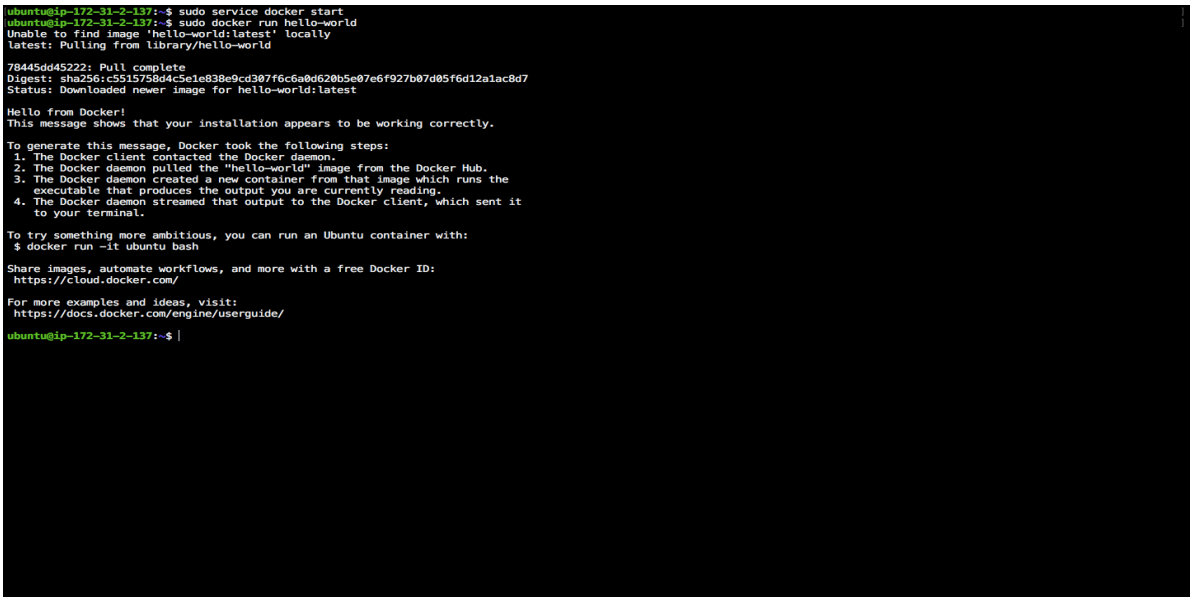
9. Install docker using following command-
   $ sudo apt-get install docker-engine



Screenshot 2: Docker Engine Installation

10. Use the following command to start docker service-
    $ sudo service docker start

11. Test whether you docker installation went well by using the following command-
    $ sudo docker run hello-world



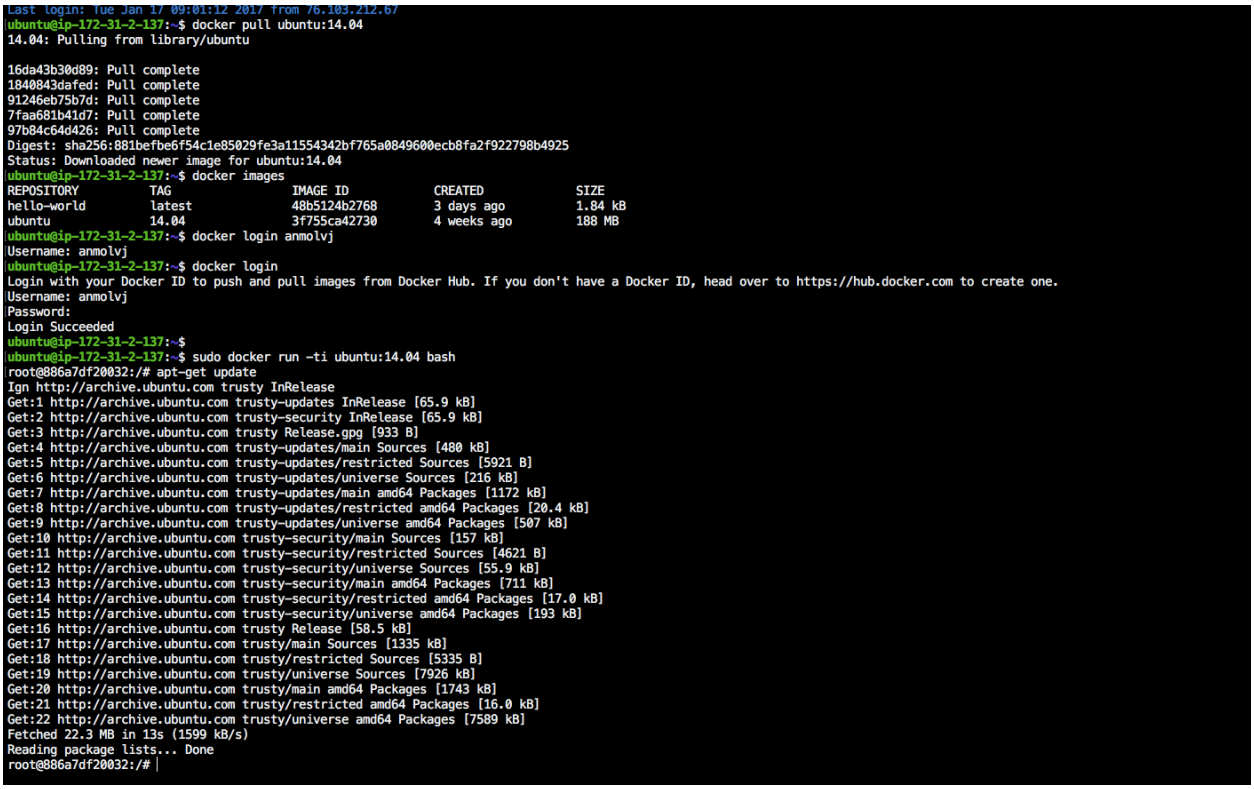
Screenshot 3: Hello-World

Steps to setup a LAMP stack in a docker container, create an image from it and upload it to my docker hub account:

1. Run the following command to pull a docker image of Ubuntu 14.04-
   $ sudo docker pull ubuntu:14.04

   NOTE: If an error is encountered that goes as, “Cannot connect to the docker deamon. Is the docker deamon running on this host?”, this is because as a user, you do not have some docker permissions setup. Use following command to ass USER to the docker group –
   $ sudo groupadd docker
   $ sudo gpasswd –a ${USER} docker
   $ sudo service docker restart

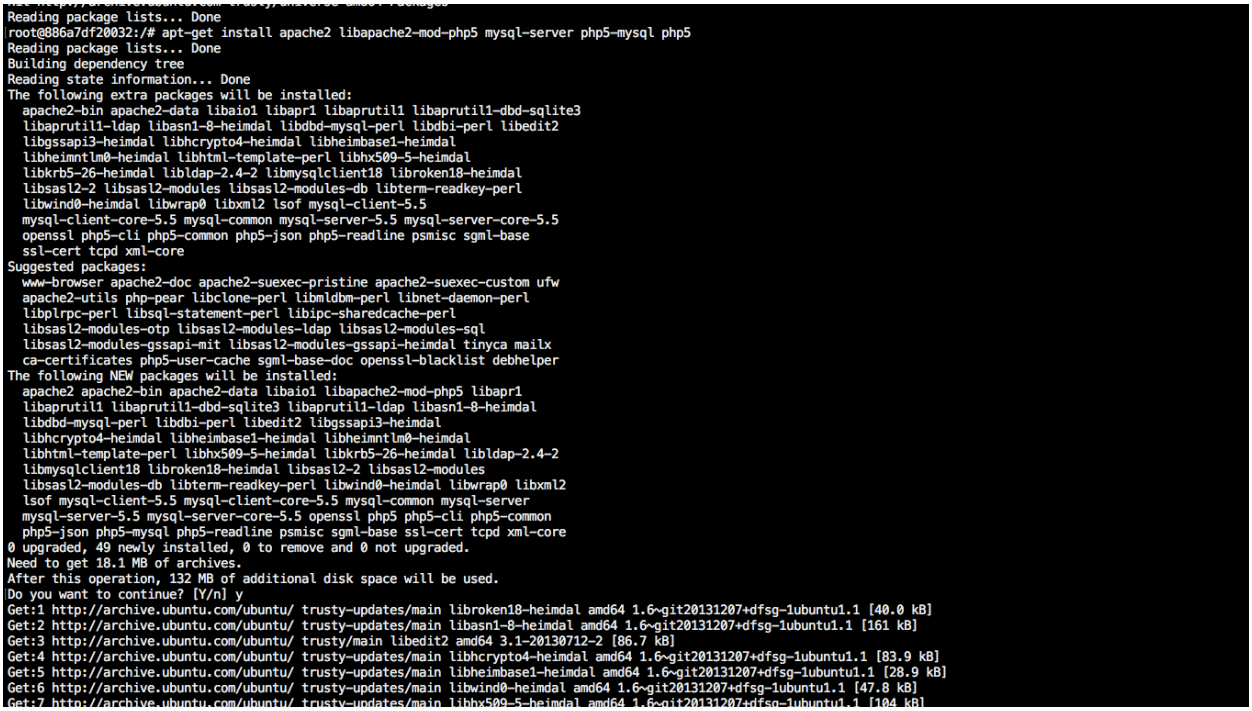
2. Run the following command to run a new container from the image downloaded in the last step and enter it in bash mode-
   $ sudo docker run -ti ubuntu:14.04 bash



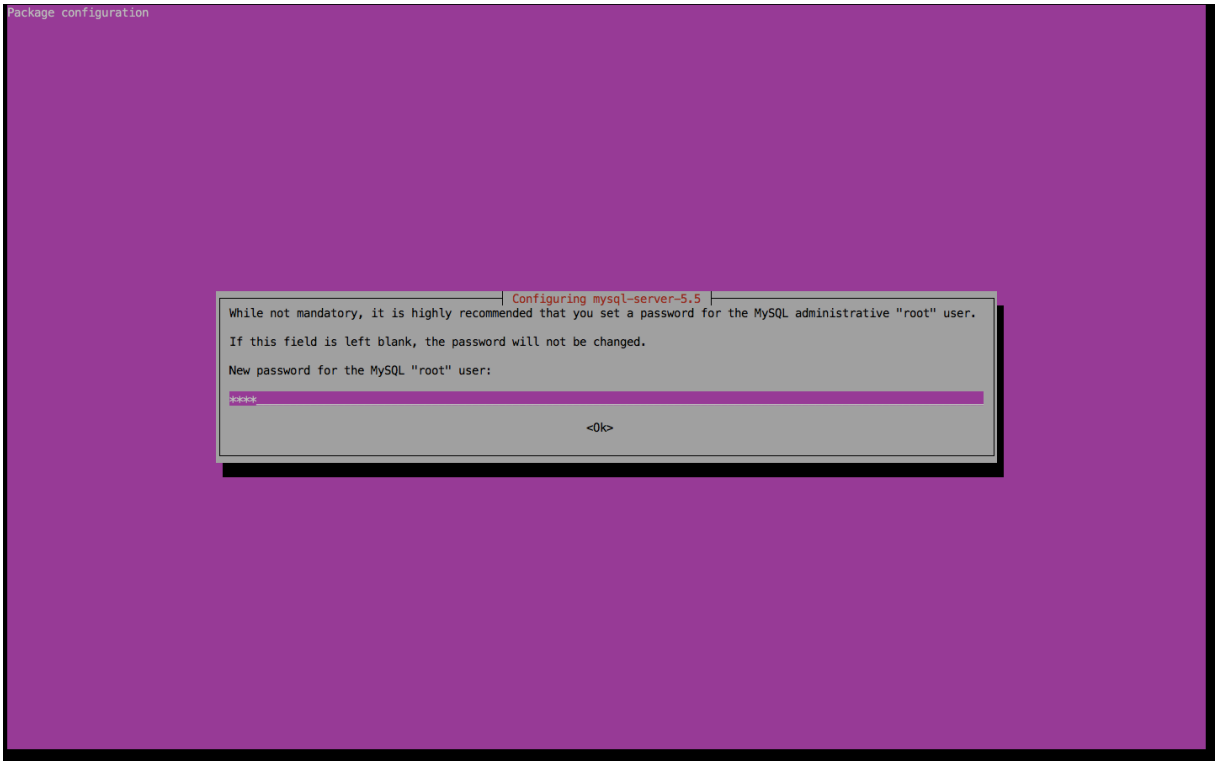
Screenshot 4: Image pulling and Container started

3. You are now inside your docker container. Use following command to perform a regular update-
   $ sudo apt-get update

4. Use the following command to install Apache2, MySQL and PHP5 in this docker container and respective modules for them to connect to each other-
   $ apt-get install apache2 libapache2-mod-php5 mysql-server php5-mysql php5
   Here:
   libapache2-mod-php5 is a package that provide php5 module for apache2 server
   Php5-mysql is a package that connects MySQL-server to php5



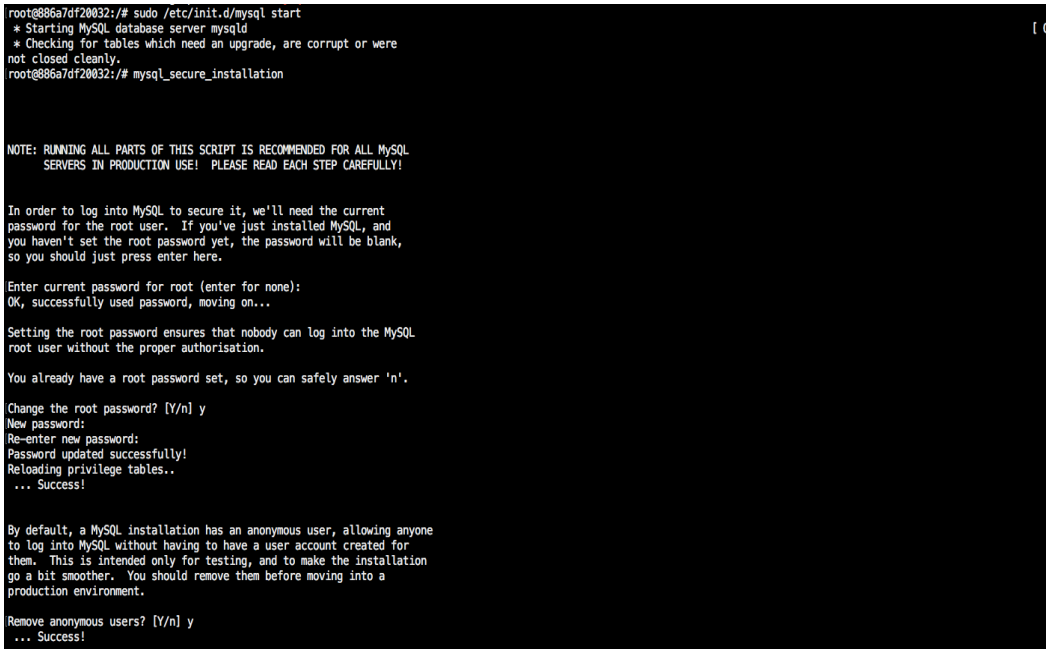
Screenshot 5: Apache2, MySQL and PHP5 installation



Screenshot 6

5. Now perform a secure installation of MySQL server using following command-
   $ mysql_secure_installation

   Note: You might get an error because mysql may not be running. Use the following command to resolve this issue-
   $ sudo /etc/init.d/mysql start



Screenshot 7

6. Restart apache2 using following command to make sure everything works-
   $ service apache2 restart

   Note: This command might throw a warning like, “Could not determine the servers fully qualified domain name…..”. This problem can be resolved by using the following command-
   $ echo "ServerName localhost" | sudo tee /etc/apache2/conf-available/fqdn.conf sudo a2enconf fqdn

7. Now use the following command to exit the container-
   $ exit

8. Use the following command to list all running containers-
   $ sudo docker ps –a

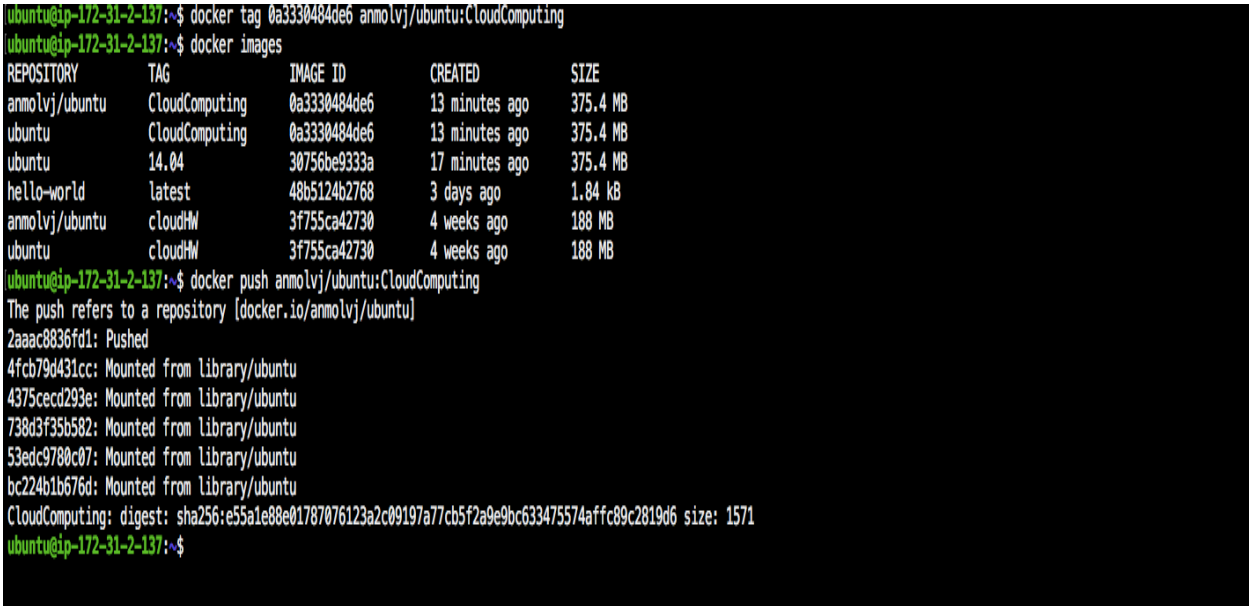
9. Note the ID of your container and use it in the following command to commit changes in your container and turn it into an image-
   $ sudo docker commit <docker-container-id> ubuntu:cloudAssignment

10. Now, login to docker-hub account using following command and enter your docker-hub user id and password-
    $ sudo docker login

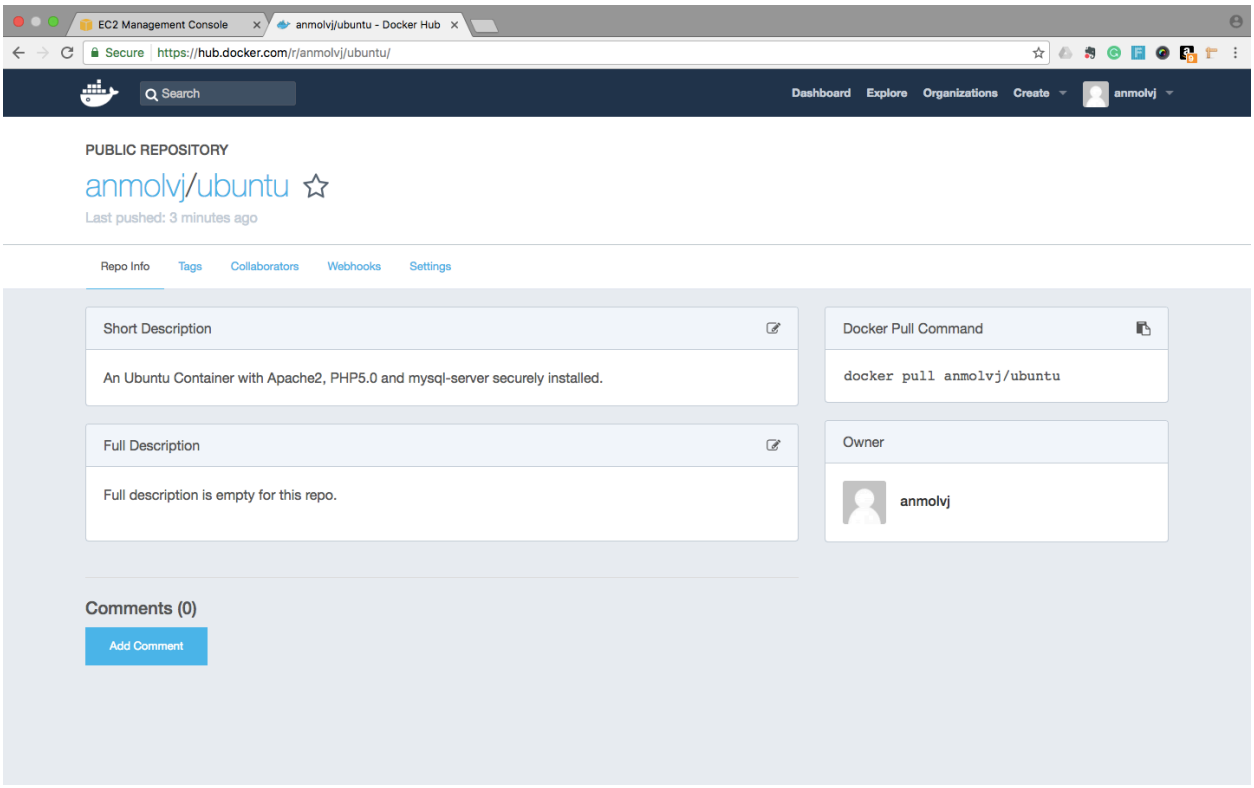
11. Now use following command to list all images on your system and find the image ID for your image-
    $ sudo docker images

12. Use following commands to tag your docker image-
    $ docker tag <image-ID> anmolvj/ubuntu:cloudAssignment

13. Now use the following command to push your docker image to your docker hub account-
    $ sudo docker push anmolvj/Ubuntu



Screenshot 8



Screenshot 9