

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 58118E89F3A912897C070AD8F76221572C52609D
Executing: /tmp/tmp.Xqlo5bFvzu/gpg.1.sh --keyserver
hkp://ha.pool.sks-keyservers.net:80
--recv-keys
58118E89F3A912897C070AD8F76221572C52609D
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

```

ubuntu@ip-172-31-2-137:~$ sudo apt-get install docker-engine
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  aufs-tools cgroupfs-mount libltdl7
Suggested packages:
  mountall
The following NEW packages will be installed:
  aufs-tools cgroupfs-mount docker-engine libltdl7
0 upgraded, 4 newly installed, 0 to remove and 4 not upgraded.
Need to get 19.5 MB of archives.
After this operation, 102 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://us-west-1.ec2.archive.ubuntu.com/ubuntu xenial/universe amd64 aufs-tools amd64 1:3.2+20130722-1.1ubuntu1 [92.9 kB]
Get:2 http://us-west-1.ec2.archive.ubuntu.com/ubuntu xenial/universe amd64 cgroupfs-mount all 1.2 [4,979 B]
Get:3 http://us-west-1.ec2.archive.ubuntu.com/ubuntu xenial/main amd64 libltdl7 amd64 2.4.6-0.1 [38.3 kB]
Get:4 https://apt.dockerproject.org/repo ubuntu-xenial/main amd64 docker-engine amd64 1.12.6-0~ubuntu-xenial [19.4 MB]
Fetched 19.5 MB in 0s (19.5 MB/s)
Selecting previously unselected package aufs-tools.
(Reading database ... 66659 files and directories currently installed.)
Preparing to unpack .../aufs-tools_1%3a3.2+20130722-1.1ubuntu1_amd64.deb ...
Unpacking aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
Selecting previously unselected package cgroupfs-mount.
Preparing to unpack .../cgroupfs-mount_1.2_all.deb ...
Unpacking cgroupfs-mount (1.2) ...
Selecting previously unselected package libltdl7:amd64.
Preparing to unpack .../libltdl7_2.4.6-0.1_amd64.deb ...
Unpacking libltdl7:amd64 (2.4.6-0.1) ...
Selecting previously unselected package docker-engine.
Preparing to unpack .../docker-engine_1.12.6-0~ubuntu-xenial_amd64.deb ...
Unpacking docker-engine (1.12.6-0~ubuntu-xenial) ...
Processing triggers for libc-bin (2.23-0ubuntu5) ...
Processing triggers for man-db (2.7.5-1) ...
Processing triggers for ureadahead (0.100.0-19) ...
Processing triggers for systemd (229-4ubuntu13) ...
Setting up aufs-tools (1:3.2+20130722-1.1ubuntu1) ...
Setting up cgroupfs-mount (1.2) ...
Setting up libltdl7:amd64 (2.4.6-0.1) ...
Setting up docker-engine (1.12.6-0~ubuntu-xenial) ...
Processing triggers for libc-bin (2.23-0ubuntu5) ...
Processing triggers for systemd (229-4ubuntu13) ...
Processing triggers for ureadahead (0.100.0-19) ...
ubuntu@ip-172-31-2-137:~$

```

Screenshot 2: Docker Engine Installation

10. Use the following command to start docker service-

\$ sudo service docker start

11. Test whether your docker installation went well by using the following command-

\$ sudo docker run hello-world

```
ubuntu@ip-172-31-2-137:~$ sudo service docker start
ubuntu@ip-172-31-2-137:~$ sudo docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
78445dd45222: Pull complete
Digest: sha256:c5515758d4c5e1e838e9cd307f6c6a0d620b5e0e6f927b07d05f6d12a1ac8d7
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.
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://cloud.docker.com/

For more examples and ideas, visit:
https://docs.docker.com/engine/userguide/
ubuntu@ip-172-31-2-137:~$
```

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`

```

Last login: Tue Jan 17 09:01:12 2017 from 76.103.212.67
ubuntu@ip-172-31-2-137:~$ docker pull ubuntu:14.04
14.04: Pulling from library/ubuntu
16da43b30d89: Pull complete
1840843dafed: Pull complete
91246eb75b7d: Pull complete
7faa681b41d7: Pull complete
97b84c64d426: Pull complete
Digest: sha256:881befbe6f54c1e85029fe3a11554342bf765a0849e00ecb8fa2f922798b4925
Status: Downloaded newer image for ubuntu:14.04
ubuntu@ip-172-31-2-137:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
hello-world          latest             48b5124b2768       3 days ago         1.84 kB
ubuntu               14.04             3f755ca42730       4 weeks ago        188 MB
ubuntu@ip-172-31-2-137:~$ docker login anmolvj
Username: anmolvj
ubuntu@ip-172-31-2-137:~$ docker login
Login with your Docker ID to push and pull images from Docker Hub. If you don't have a Docker ID, head over to https://hub.docker.com to create one.
Username: anmolvj
Password:
Login Succeeded
ubuntu@ip-172-31-2-137:~$
ubuntu@ip-172-31-2-137:~$ sudo docker run -ti ubuntu:14.04 bash
root@886a7df20032:/# apt-get update
Ign http://archive.ubuntu.com trusty InRelease
Get:1 http://archive.ubuntu.com trusty-updates InRelease [65.9 kB]
Get:2 http://archive.ubuntu.com trusty-security InRelease [65.9 kB]
Get:3 http://archive.ubuntu.com trusty Release.gpg [933 B]
Get:4 http://archive.ubuntu.com trusty-updates/main Sources [480 kB]
Get:5 http://archive.ubuntu.com trusty-updates/restricted Sources [5921 B]
Get:6 http://archive.ubuntu.com trusty-updates/universe Sources [216 kB]
Get:7 http://archive.ubuntu.com trusty-updates/main amd64 Packages [1172 kB]
Get:8 http://archive.ubuntu.com trusty-updates/restricted amd64 Packages [20.4 kB]
Get:9 http://archive.ubuntu.com trusty-updates/universe amd64 Packages [507 kB]
Get:10 http://archive.ubuntu.com trusty-security/main Sources [157 kB]
Get:11 http://archive.ubuntu.com trusty-security/restricted Sources [4621 B]
Get:12 http://archive.ubuntu.com trusty-security/universe Sources [55.9 kB]
Get:13 http://archive.ubuntu.com trusty-security/main amd64 Packages [711 kB]
Get:14 http://archive.ubuntu.com trusty-security/restricted amd64 Packages [17.0 kB]
Get:15 http://archive.ubuntu.com trusty-security/universe amd64 Packages [193 kB]
Get:16 http://archive.ubuntu.com trusty Release [58.5 kB]
Get:17 http://archive.ubuntu.com trusty/main Sources [1335 kB]
Get:18 http://archive.ubuntu.com trusty/restricted Sources [5335 B]
Get:19 http://archive.ubuntu.com trusty/universe Sources [7926 kB]
Get:20 http://archive.ubuntu.com trusty/main amd64 Packages [1743 kB]
Get:21 http://archive.ubuntu.com trusty/restricted amd64 Packages [16.0 kB]
Get:22 http://archive.ubuntu.com trusty/universe amd64 Packages [7589 kB]
Fetched 22.3 MB in 13s (1599 kB/s)
Reading package lists... Done
root@886a7df20032:/#

```

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

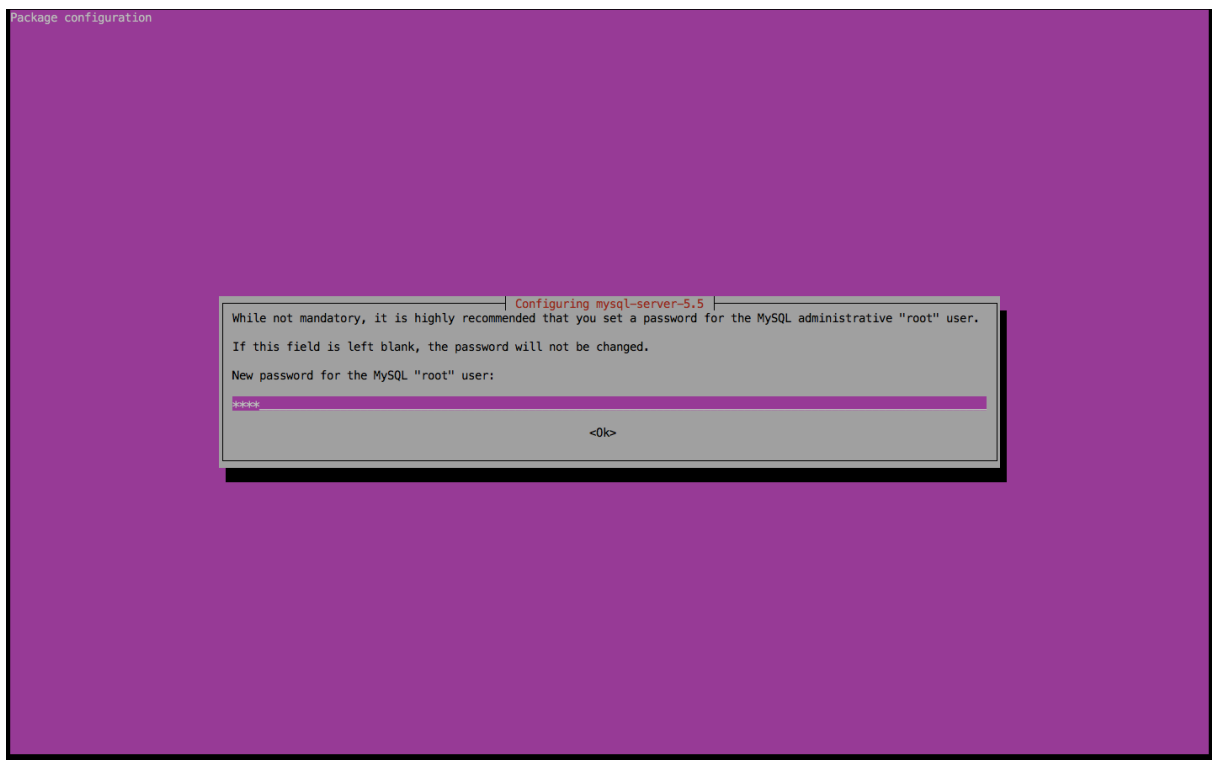
Php5-mysql is a package that connects MySQL-server to php5

```

Reading package lists... Done
root@886a7df2032:/# apt-get install apache2 libapache2-mod-php5 mysql-server php5-mysql php5
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  apache2-bin apache2-data libaio1 libapr1 libaprutil1 libaprutil1-dbd-sqlite3
  libaprutil1-ldap libasn1-8-heimdal libdbd-mysql-perl libdbi-perl libedit2
  libgssapi3-heimdal libhcrypto4-heimdal libheimbase1-heimdal
  libheimntlm0-heimdal libhtml-template-perl libhx509-5-heimdal
  libkrb5-26-heimdal libldap-2.4-2 libmysqlclient18 libroken18-heimdal
  libsasl2-2 libsasl2-modules libsasl2-modules-db libterm-readkey-perl
  libwind0-heimdal libwrap0 libxml2 lsof mysql-client-5.5
  mysql-client-core-5.5 mysql-common mysql-server-5.5 mysql-server-core-5.5
  openssl php5-cli php5-common php5-json php5-readline psmisc sgml-base
  ssl-cert tcpd xml-core
Suggested packages:
  www-browser apache2-doc apache2-suexec-pristine apache2-suexec-custom ufw
  apache2-utils php-pear libclone-perl libmldbm-perl libnet-daemon-perl
  liblprc-perl libsql-statement-perl libipc-sharedcache-perl
  libsasl2-modules-otp libsasl2-modules-ldap libsasl2-modules-sql
  libsasl2-modules-gssapi-mit libsasl2-modules-gssapi-heimdal tinycsa mailx
  ca-certificates php5-user-cache sgml-base-doc openssl-blacklist debhelper
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data libaio1 libapache2-mod-php5 libapr1
  libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap libasn1-8-heimdal
  libdbd-mysql-perl libdbi-perl libedit2 libgssapi3-heimdal
  libhcrypto4-heimdal libheimbase1-heimdal libheimntlm0-heimdal
  libhtml-template-perl libhx509-5-heimdal libkrb5-26-heimdal libldap-2.4-2
  libmysqlclient18 libroken18-heimdal libsasl2-2 libsasl2-modules
  libsasl2-modules-db libterm-readkey-perl libwind0-heimdal libwrap0 libxml2
  lsof mysql-client-5.5 mysql-client-core-5.5 mysql-common mysql-server
  mysql-server-5.5 mysql-server-core-5.5 openssl php5 php5-cli php5-common
  php5-json php5-mysql php5-readline psmisc sgml-base ssl-cert tcpd xml-core
0 upgraded, 49 newly installed, 0 to remove and 0 not upgraded.
Need to get 18.1 MB of archives.
After this operation, 132 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu/ trusty-updates/main libroken18-heimdal amd64 1.6~git20131207+dfsg-1ubuntu1.1 [40.0 kB]
Get:2 http://archive.ubuntu.com/ubuntu/ trusty-updates/main libasn1-8-heimdal amd64 1.6~git20131207+dfsg-1ubuntu1.1 [161 kB]
Get:3 http://archive.ubuntu.com/ubuntu/ trusty/main libedit2 amd64 3.1-20130712-2 [86.7 kB]
Get:4 http://archive.ubuntu.com/ubuntu/ trusty-updates/main libhcrypto4-heimdal amd64 1.6~git20131207+dfsg-1ubuntu1.1 [83.9 kB]
Get:5 http://archive.ubuntu.com/ubuntu/ trusty-updates/main libheimbase1-heimdal amd64 1.6~git20131207+dfsg-1ubuntu1.1 [28.9 kB]
Get:6 http://archive.ubuntu.com/ubuntu/ trusty-updates/main libwind0-heimdal amd64 1.6~git20131207+dfsg-1ubuntu1.1 [47.8 kB]
Get:7 http://archive.ubuntu.com/ubuntu/ trusty-updates/main libhx509-5-heimdal amd64 1.6~git20131207+dfsg-1ubuntu1.1 [104 kB]

```

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
```

```
root@886a7df20032:/# sudo /etc/init.d/mysql start
* Starting MySQL database server mysqld
* Checking for tables which need an upgrade, are corrupt or were
not closed cleanly.
root@886a7df20032:/# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!

In order to log into MySQL to secure it, we'll need the current
password for the root user. If you've just installed MySQL, and
you haven't set the root password yet, the password will be blank,
so you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorisation.

You already have a root password set, so you can safely answer 'n'.

Change the root password? [Y/n] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!

By default, a MySQL installation has an anonymous user, allowing anyone
to log into MySQL without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.

Remove anonymous users? [Y/n] y
... Success!
```

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
```

```
ubuntu@ip-172-31-2-137:~$ docker tag 0a3330484de6 anmolvj/ubuntu:CloudComputing
ubuntu@ip-172-31-2-137:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
anmolvj/ubuntu       CloudComputing      0a3330484de6       13 minutes ago     375.4 MB
ubuntu               CloudComputing      0a3330484de6       13 minutes ago     375.4 MB
ubuntu               14.04              30756be9333a       17 minutes ago     375.4 MB
hello-world          latest              48b5124b2768       3 days ago         1.84 kB
anmolvj/ubuntu       cloudHW             3f755ca42730       4 weeks ago        188 MB
ubuntu               cloudHW             3f755ca42730       4 weeks ago        188 MB
ubuntu@ip-172-31-2-137:~$ docker push anmolvj/ubuntu:CloudComputing
The push refers to a repository [docker.io/anmolvj/ubuntu]
2aac8836fd1: Pushed
4fcb79d431cc: Mounted from library/ubuntu
4375cecd293e: Mounted from library/ubuntu
738d3f35b582: Mounted from library/ubuntu
53edc9780c07: Mounted from library/ubuntu
bc224b1b676d: Mounted from library/ubuntu
CloudComputing: digest: sha256:e55a1e08e01787076123a2c09197a77cb5f2a9e9bc633475574affc89c2819d6 size: 1571
ubuntu@ip-172-31-2-137:~$
```


EC2 Management Console

anmolvj/ubuntu - Docker Hub

Securehttps://hub.docker.com/r/anmolvj/ubuntu/

 Search

DashboardExploreOrganizationsCreateanmolvj

PUBLIC REPOSITORY

anmolvj/ubuntu ☆

Last pushed: 3 minutes ago

Repo InfoTagsCollaboratorsWebhooksSettings

Short Description

An Ubuntu Container with Apache2, PHP5.0 and mysql-server securely installed.


Full Description

Full description is empty for this repo.

Docker Pull Command

docker pull anmolvj/ubuntu

Owner

 anmolvj

Comments (0)

Add Comment

Screenshot 9