Create & Manage Docker (Container, Image) On AWS EC2

[Edition 01]

[Last Update 221204]

For any issues/help contact : ***support@k21academy.com***

Contents

1 INTRODUCTION

Docker is a platform for developers and sysadmins to build, run, and share applications with containers. The use of containers to deploy applications is called containerization. Containers are not new, but their use for easily deploying applications is.

Containerization is increasingly popular because containers are:

- Flexible: Even the most complex applications can be containerized.
- Lightweight: Containers leverage and share the host kernel, making them much more efficient in terms of system resources than virtual machines.
- Portable: You can build locally, deploy to the cloud, and run anywhere.
- Loosely coupled: Containers are highly self-sufficient and encapsulated, allowing you to replace or upgrade one without disrupting others.
- Scalable: You can increase and automatically distribute container replicas across a datacenter.
- Secure: Containers apply aggressive constraints and isolations to processes without any configuration required on the part of the user.

2 DOCUMENTATION

1. Docker Container
   https://docs.docker.com/engine/reference/commandline/container/
2. docker container attach
   https://docs.docker.com/engine/reference/commandline/container_attach/
3. docker container create
   https://docs.docker.com/engine/reference/commandline/container_create/
4. Docker Image
   https://docs.docker.com/engine/reference/commandline/images/

2.1 Linux Commands and VIM Commands

1. Basic Linux Commands
   https://maker.pro/linux/tutorial/basic-linux-commands-for-beginners
   https://www.hostinger.in/tutorials/linux-commands
2. Basic VIM Commands
   https://coderwall.com/p/adv71w/basic-vim-commands-for-getting-started
3. Popular VIM Commands
   https://www.keycdn.com/blog/vim-commands

3 PRE-REQUISITE

Ensure that you have completed following two activity guides (or you have an Ubuntu Server)

- Create account (Trial or Paid) on Azure Cloud.

  Note**: Follow Activity Guide **Register_For_AWS_Cloud_Account_Accessing_Console_ed* from portal*

- Installed Ubuntu Server

  Note**: Follow Activity Guide **Create_&_Connect_to_Ubuntu_Server_ed* from portal*

4 DOCKER INSTALLATION STEPS ON UBUNTU 20.04 SERVER

1. Become the root user:

```
$ sudo -i
```

2. First Update Software Repositories

```
$ sudo apt-get update -y
```

3. Uninstall Old Versions of Docker ***(Optional:*** *Only if docker was already installed on this host and you want to configure it again)*
   (Older versions of Docker were called docker, docker.io, or docker-engine. If these are installed, uninstall them)

```
$ sudo apt-get remove docker docker-engine docker.io containerd runc
```

4. Update the apt package index and install packages to allow apt to use a repository over HTTPS:

```
$ sudo apt-get update
$  sudo apt-get install \
    ca-certificates \
    curl \
    gnupg \
    lsb-release
```

5. Add Docker's official GPG key:

```
$ curl -fsSL https://download.docker.com/linux/ubuntu/gpg | sudo gpg --dearmor -o /usr/share/keyrings/docker-archive-keyring.gpg
```

6. Use the following command to set up the stable repository. To add the nightly or test repository, add the word nightly or test (or both) after the word stable in the commands below.

```
$ echo \
  "deb [arch=$(dpkg --print-architecture) signed-by=/usr/share/keyrings/docker-archive-keyring.gpg] https://download.docker.com/linux/ubuntu \
  $(lsb_release -cs) stable" | sudo tee /etc/apt/sources.list.d/docker.list > /dev/null
```

7. Install Docker Engine.

```
$ sudo apt-get update
$ sudo apt-get install docker-ce docker-ce-cli containerd.io docker-compose-plugin
```

8. Start and Enable Docker

```
$ sudo systemctl start docker
```

```
$ sudo systemctl enable docker
```

9. Check Docker status

```
$ sudo systemctl status docker
```

10. Identify the user id of user that will run container
(If you are a root user then you can skip this step)

```
$ id
```

```
DockerMachineUser@DockerMachine:~$ id
uid=1000(DockerMachineUser) gid=1000(DockerMachineUser) groups=1000(DockerMachin
eUser),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(vid
eo),46(plugdev),108(lxd),114(netdev)
DockerMachineUser@DockerMachine:~$
```

11. Add above user to docker group

```
$ sudo usermod -aG docker <userid>

$ sudo usermod -aG docker DockerMachineUser
```

12. Logout and log back in and check if user has group docker assigned

```
DockerMachineUser@DockerMachine:~$ id
uid=1000(DockerMachineUser) gid=1000(DockerMachineUser) groups=1000(DockerMachin
eUser),4(adm),20(dialout),24(cdrom),25(floppy),27(sudo),29(audio),30(dip),44(vid
eo),46(plugdev),108(lxd),114(netdev),116(docker)
DockerMachineUser@DockerMachine:~$
```

Note: *If you don't logout and log back in you may get error like below at later stage*

```
dkaramganesh@DockerMachine:~$ docker --version
Docker version 19.03.6, build 369ce74a3c
dkaramganesh@DockerMachine:~$ docker run -it ubuntu bash
docker: Got permission denied while trying to connect to the Docker daemon socke
t at unix:///var/run/docker.sock: Post http://%2Fvar%2Frun%2Fdocker.sock/v1.40/c
ontainers/create: dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
dkaramganesh@DockerMachine:~$
```

13. Check Docker Version

```
docker --version
```

5 WORKING WITH CONTAINER

1. Creating first Container

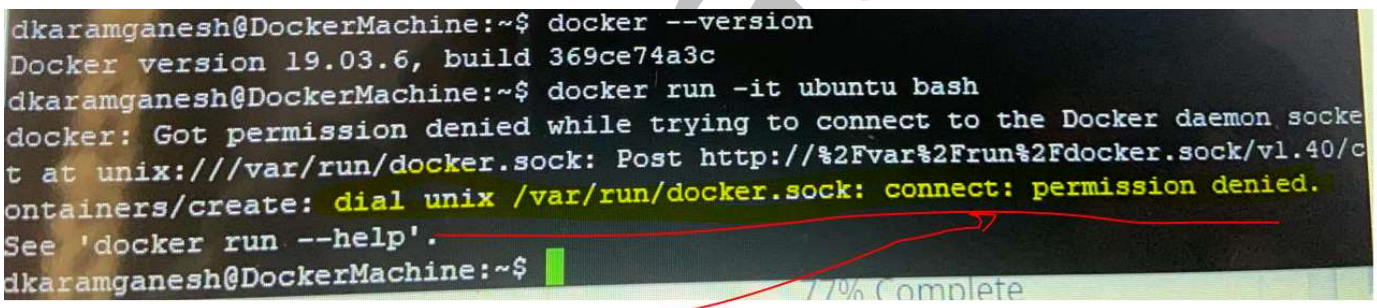
```
$ docker run -it ubuntu bash
```

Or

```
$ docker container run -it ubuntu bash
```

```
root@master:~# docker run -it ubuntu bash
Unable to find image 'ubuntu:latest' locally
latest: Pulling from library/ubuntu
d51af753c3d3: Pull complete
fc878cd0a91c: Pull complete
6154df8ff988: Pull complete
fee5db0ff82f: Pull complete
Digest: sha256:747d2dbbaaee995098c9792d99bd333c6783ce56150d1b11e333bbceed5c54d7
Status: Downloaded newer image for ubuntu:latest
root@bc35381ed053:/#
```

Note: *If you don't logout and log back in you may get error like below at later stage*



2. Update Software Repositories inside the Container

```
root@bc35381ed053:/# apt-get update
```

3. Quit from Container without stopping it

```
Press Ctrl p Ctrl q
```

4. List the running Containers

```
$ docker ps
```

```
$ docker container ls
```

```
root@master:~# docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
bc35381ed053        ubuntu              "bash"              8 hours ago         Up 8 hours                              infallible_nash
```

5. List all Containers

```
$ docker ps -a
```

```
$ docker container ls -a
```

```
root@master:~# docker ps -a
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
bc35381ed053        ubuntu              "bash"              8 hours ago         Up 8 hours                              infallible_nash
root@master:~#
```

6. List the images in Local Repository

```
$ docker images
```

```
root@master:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              1d622ef86b13        4 days ago          73.9MB
root@master:~#
```

6 WORKING WITH DOCKER IMAGES

1. Pull Docker Image from Public Docker Repo

```
$ docker pull alpine:3.6
```

OR

```
docker image pull alpine:3.6
```

```
root@master:~# docker pull alpine:3.6
3.6: Pulling from library/alpine
5a3ea8efae5d: Pull complete
Digest: sha256:66790a2b79e1ea3e1dabac43990c54aca5d1ddf268d9a5a0285e4167c8b24475
Status: Downloaded newer image for alpine:3.6
docker.io/library/alpine:3.6
```

2. List Image by name and tag

```
$ docker images alpine:3.6
```

```
root@master:~# docker images alpine:3.6
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine              3.6                 43773d1dba76        13 months ago       4.03MB
root@master:~#
```

3. List full length Image ID

```
$ docker images --no-trunc
```

```
root@master:~#
root@master:~# docker images --no-trunc
REPOSITORY          TAG                 IMAGE ID                                                                  CREATED             SI
ZE
ubuntu              latest              sha256:1d622ef86b138c7e96d4f797bf5e4baca3249f030c575b9337638594f2b63f01   4 days ago          73
.9MB
alpine              3.6                 sha256:43773d1dba76c4d537b494a8454558a41729b92aa2ad0feb23521c3e58cd0440   13 months ago       4.
03MB
root@master:~#
```

4. Listing out Images with Filter

```
$ docker images --filter=reference='alpine'
```

```
root@master:~# docker images --filter=reference='alpine'
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine              3.6                 43773d1dba76        13 months ago       4.03MB
root@master:~#
```

5. Inspect Image details

```
$ docker inspect alpine:3.6
```

```
root@master:~# docker inspect alpine:3.6
[
    {
        "Id": "sha256:43773d1dba76c4d537b494a8454558a41729b92aa2ad0feb23521c3e58cd0440",
        "RepoTags": [
            "alpine:3.6"
        ],
        "RepoDigests": [
            "alpine@sha256:66790a2b79e1ea3e1dabac43990c54aca5d1ddf268d9a5a0285e4167c8b24475"
        ],
        "Parent": "",
        "Comment": "",
        "Created": "2019-03-07T22:20:00.563496859Z",
        "Container": "fd086f4b9352674c6a1ae4d02051f95a4e0a55cda943c5780483938dedfb2d8f",
        "ContainerConfig": {
            "Hostname": "fd086f4b9352",
            "Domainname": "",
            "User": "",
            "AttachStdin": false,
            "AttachStdout": false,
            "AttachStderr": false,
            "Tty": false,
            "OpenStdin": false,
            "StdinOnce": false,
            "Env": [
                "PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin"
            ],
            "Cmd": [
                "/bin/sh",
```

6. Commit Container to create Container Image

```
$ docker ps
$ docker commit [OPTIONS] CONTAINER [REPOSITORY[:TAG]]
$ docker commit bc35381ed053  ubuntu:new
```

```
$
$ docker commit 283f7a544549 ubuntu:new
sha256:e33dea91ab5dcffac39f264009428571e81b27a2ba80e1cc5c760fe9028f4704
$
```

7. Verify by listing the newly create Image

```
$ docker images
```

```
$
$ docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
ubuntu                              new                 e33dea91ab5d        36 seconds ago      81.2MB
```

8. List running Containers

```
$ docker ps
```

OR

```
$ docker container ls
```

```
$
$ docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS              PORTS               NAMES
283f7a544549        ubuntu              "bash"              About a minute ago   Up About a minute                       elastic_fermat
```

9. Export Container as tar file (Persisting Container)

```
$ docker export bc3538  > ubuntu.tar
$ ls -lrt
```

```
$
$ docker export 283f  > ubuntu.tar
```

```
root@master:~# ls -lrt
total 94032
-rw-r--r-- 1 root root 96288768 Apr 28 01:16 ubuntu.tar
root@master:~#
```

10. Commit this tar as a new image locally

```
$ docker import - myubuntu < ubuntu.tar
$ docker images
```

```
$
$ docker import - myubuntu < ubuntu.tar
sha256:adafd7326af3485ce4d8ae13db83d66bb5b72bb6184db67f2815b0e173906a13
```

```
$
$ docker images
REPOSITORY                          TAG                 IMAGE ID            CREATED             SIZE
myubuntu                            latest              adafd7326af3        20 seconds ago      69.8MB
```

11. Save Image as tar file (Persisting Image)

```
$ docker pull nginx
```

```
root@master:~# docker pull nginx
Using default tag: latest
latest: Pulling from library/nginx
54fec2fa59d0: Pull complete
4ede6f09aefe: Pull complete
f9dc69acb465: Pull complete
Digest: sha256:86ae264c3f4acb99b2dee4d0098c40cb8c46dcf9e1148f05d3a51c4df6758c12
Status: Downloaded newer image for nginx:latest
docker.io/library/nginx:latest
root@master:~#
root@master:~#
```

```
$ docker images nginx
```

```
ubuntu@dockervm:~$ docker images nginx
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              2622e6cca7eb        3 weeks ago         132MB
```

$ docker save -o mynginx1.tar nginx

$ ls -lrt

```
ubuntu@dockervm:~$ docker save -o mynginx1.tar nginx
ubuntu@dockervm:~$ ls -lrt
total 133132
-rw------- 1 ubuntu ubuntu 136325120 Jul  7 01:07 mynginx1.tar
```

12. Delete Images from Local repo

$ docker rmi nginx:latest

```
root@master:~#
root@master:~# docker rmi nginx:latest
Untagged: nginx:latest
Untagged: nginx@sha256:86ae264c3f4acb99b2dee4d0098c40cb8c46dcf9e1148f05d3a51c4df6758c12
Deleted: sha256:602e111c06b6934013578ad80554a074049c59441d9bcd963cb4a7feccede7a5
Deleted: sha256:81eaddad75aaa517b4a597912da28c2f5b905f6e9789dce3aea874b040aad201
Deleted: sha256:73cafa8418003ecfaa02360f181c132b2cf4b61433e1bd5c84012941105865c8
Deleted: sha256:c2adabaecedbda0af72b153c6499a0555f3a769d52370469d8f6bd6328af9b13
root@master:~#
```

13. Load Image back from tar file

$ docker images

```
root@master:~# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
myubuntu            latest              1235f74ee74c        21 minutes ago      93.9MB
ubuntu              new                 ba63bd9c2ba8        38 minutes ago      94.9MB
ubuntu              latest              1d622ef86b13        4 days ago          73.9MB
alpine              3.6                 43773d1dba76        13 months ago       4.03MB
root@master:~#
```

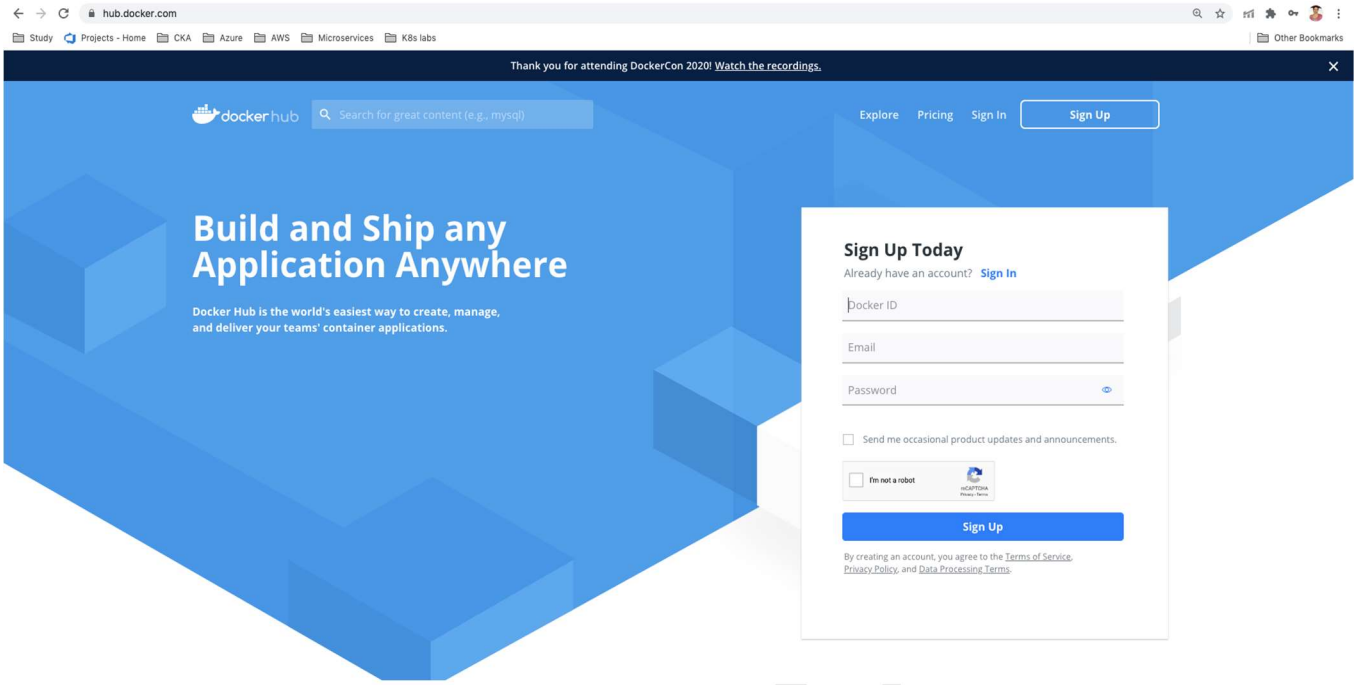
$ docker load < mynginx1.tar

```
root@master:~#
root@master:~# docker load < mynginx1.tar
c2adabaecedb: Loading layer [==================================================>]  72.49MB/72.49MB
216cf33c0a28: Loading layer [==================================================>]  58.11MB/58.11MB
b3003aac411c: Loading layer [==================================================>]  3.584kB/3.584kB
Loaded image: nginx:latest
root@master:~#
```

$ docker images nginx

```
ubuntu@dockervm:~$ docker images nginx
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
nginx               latest              2622e6cca7eb        3 weeks ago         132MB
```

14. Create your own registry at https://hub.docker.com by signing up



15. Tag and Push image to Docker Hub

a) Login at the command line terminal with your Docker ID created in above step to push and pull images from Docker Hub

```
$ docker login
```

Note: *Replace mamtaj with your Docker Hub Login ID*

```
ubuntu@dockervm:~$
ubuntu@dockervm:~$ 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://hu
b.docker.com to create one.
Username: mamtaj
Password:
WARNING! Your password will be stored unencrypted in /home/ubuntu/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

b) We will tag the image with the registry name to push it to our named registry

So the format of tagging is:

docker tag SOURCE_NAME:TAG TARGET_NAME:TAG

TAGRET_NAME is above command should be of format:

Registry/repository:tagname

Note: *Replace mamtaj with your Docker Hub Login ID*

```
$ docker tag nginx:latest mamtaj/nginx:latest
$ docker images
```

```
ubuntu@dockervm:~$ docker tag nginx:latest mamtaj/nginx:latest
ubuntu@dockervm:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              latest              74435f89ab78        2 weeks ago         73.9MB
nginx               latest              2622e6cca7eb        3 weeks ago         132MB
mamtaj/nginx        latest              2622e6cca7eb        3 weeks ago         132MB
alpine              3.6                 43773d1dba76        16 months ago       4.03MB
```

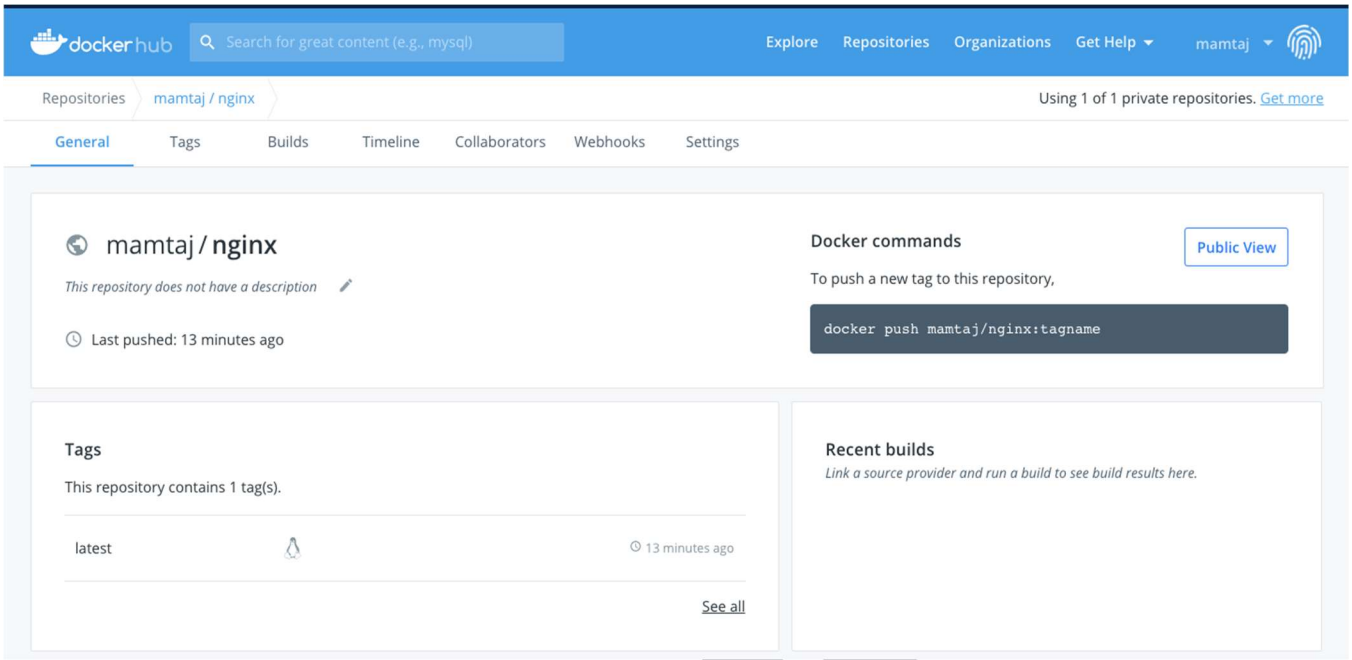
c) Push the image with docker push command

Note: *Replace mamtaj with your Docker Hub Login ID*

```
$ docker push mamtaj/nginx:latest
```

```
ubuntu@dockervm:~$
ubuntu@dockervm:~$ docker push mamtaj/nginx:latest
The push refers to repository [docker.io/mamtaj/nginx]
f978b9ed3f26: Mounted from library/nginx
9040af41bb66: Mounted from library/nginx
7c7d7f446182: Mounted from library/nginx
d4cf327d8ef5: Mounted from library/nginx
13cb14c2acd3: Mounted from library/nginx
latest: digest: sha256:0efad4d09a419dc6d574c3c3baacb804a530acd61d5eba72cb1f14e1f5ac0c8f size: 1362
```

d) *Verify that the image is pushed to our registry at https://hub.docker.com*

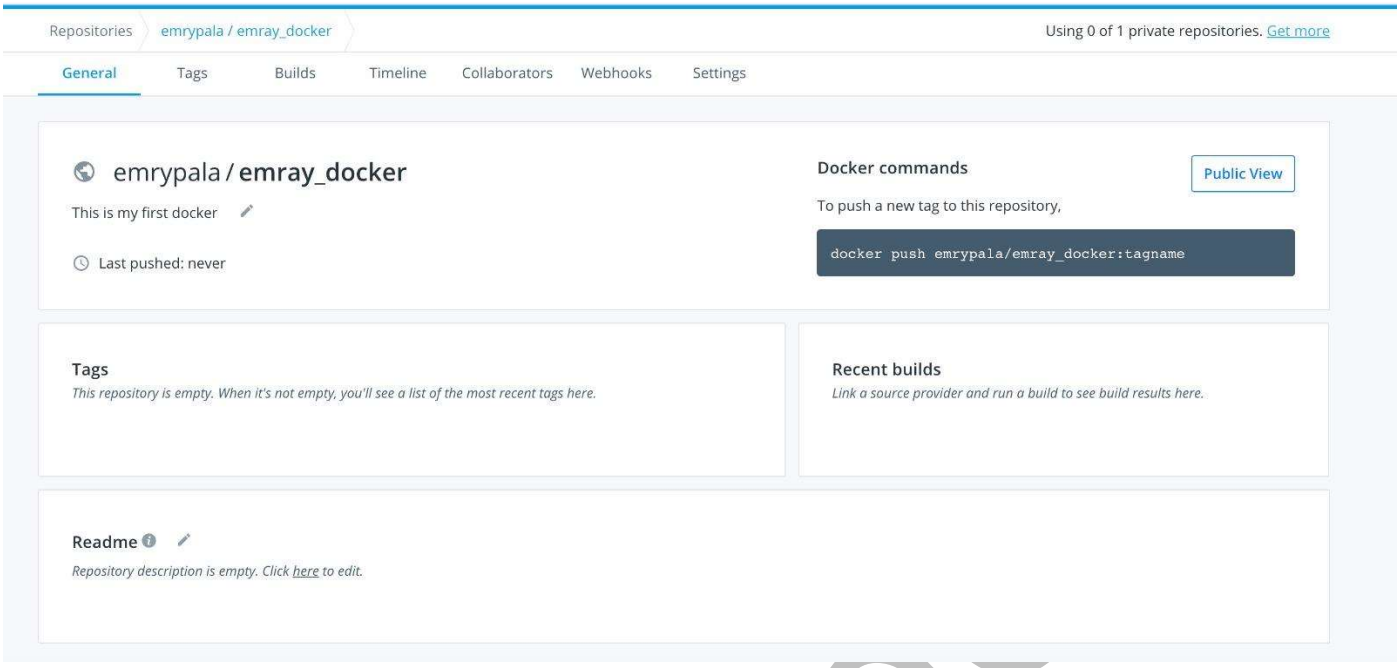


7 TROUBLESHOOTING

7.1 Issue: Access Denied Error

Lab-6

```
AzureUser@DMA:~$ 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 o
Username: emrypala
Password:
WARNING! Your password will be stored unencrypted in /home/AzureUser/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
AzureUser@DMA:~$ ocker tag nginx:latest emray_docker/nginx:latest

Command 'ocker' not found, did you mean:

  command 'docker' from snap docker (19.03.11)
  command 'docker' from deb docker.io
  command 'locker' from deb flowscan

See 'snap info <snapname>' for additional versions.

AzureUser@DMA:~$ docker tag nginx:latest emray_docker/nginx:latest
AzureUser@DMA:~$ docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
ubuntu              new                 829c4a78cf2c        28 minutes ago      96.1MB
emray_docker/nginx  latest              0901fa9da894        31 hours ago        132MB
nginx               latest              0901fa9da894        31 hours ago        132MB
ubuntu              latest              74435f89ab78        3 weeks ago         73.9MB
alpine              3.6                 43773d1dba76        16 months ago       4.03MB
AzureUser@DMA:~$ docker push emrypala/emray_docker:tagname
The push refers to repository [docker.io/emrypala/emray_docker]
An image does not exist locally with the tag: emrypala/emray_docker
AzureUser@DMA:~$ docker push emrypala/emray_docker:latest
The push refers to repository [docker.io/emrypala/emray_docker]
An image does not exist locally with the tag: emrypala/emray_docker
AzureUser@DMA:~$  docker push emray_docker/nginx:latest
The push refers to repository [docker.io/emray_docker/nginx]
2808ec4a8ea7: Preparing
4856db5e4f59: Preparing
7ef35766ef7d: Preparing
0e32546a8af0: Preparing
13cb14c2acd3: Preparing
denied: requested access to the resource is denied
AzureUser@DMA:~$ docker push emray_docker/nginx:latest
The push refers to repository [docker.io/emray_docker/nginx]
2808ec4a8ea7: Preparing
4856db5e4f59: Preparing
7ef35766ef7d: Preparing
0e32546a8af0: Preparing
13cb14c2acd3: Preparing
denied: requested access to the resource is denied
AzureUser@DMA:~$ docker push emrypala/emray_docker:latest
The push refers to repository [docker.io/emrypala/emray_docker]
An image does not exist locally with the tag: emrypala/emray_docker
AzureUser@DMA:~$
```



Solution: Wrong login ID Used

Tagged image with Wrong Id

docker id: emrypala

docker tag nginx:latest emrypala/nginx:latest *(To tag)*
docker push emrypala/nginx:latest *(To Push Image)*

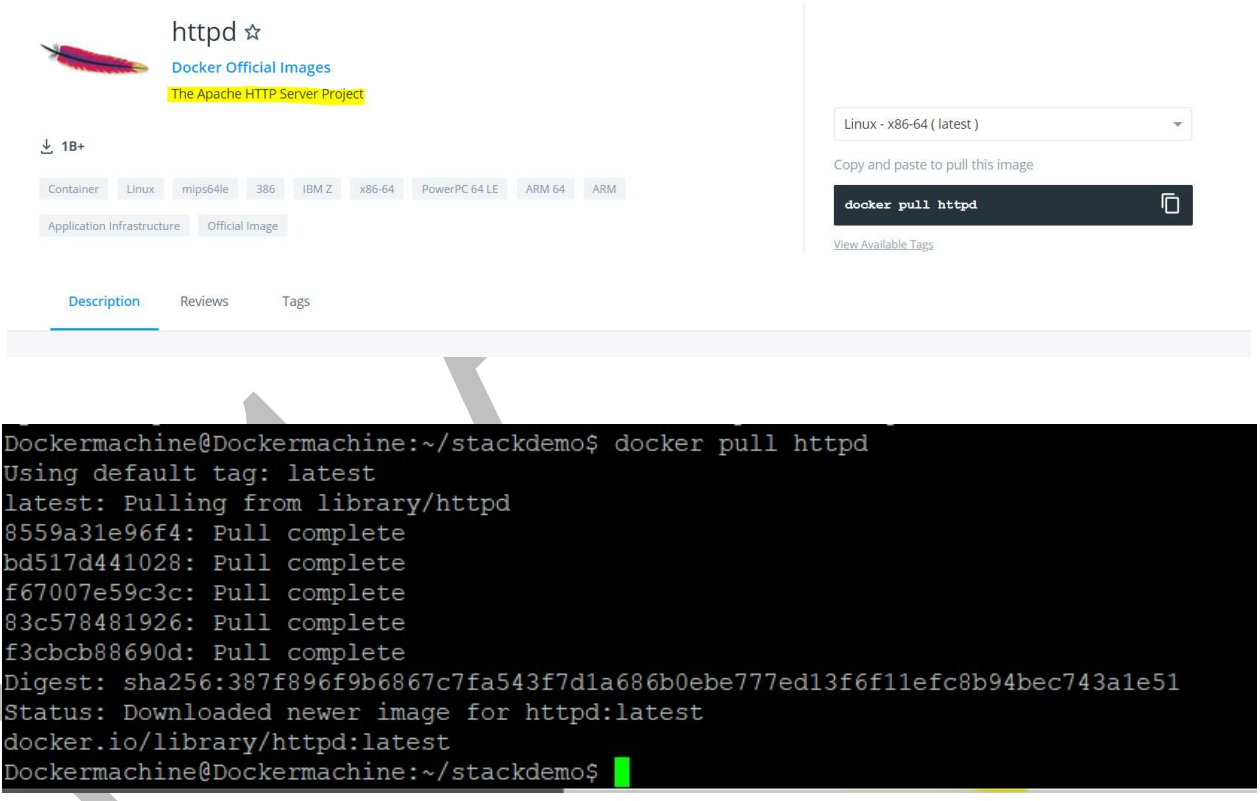
7.2 Can't pull Apache2 Image

Issue: When trying to pull Apache Image getting error like *pull access denied for apache, repository does not exist or may require 'docker login': denied: requested access to the resource is denied*

Reason: Getting this Error because apache image not available public Docker hub

```
Dockermachine@Dockermachine:~/stackdemo$ docker search apache2
NAME                              DESCRIPTION                                     STARS               OFFICIAL            AUTOMATED
antage/apache2-php5               Docker image for running Apache 2.x with PHP…   20                                      [OK]
teamrock/apache2                  TeamRock's Apache2                              8                                       [OK]
firespring/apache2-php            Based on Ubuntu 16.04. Contains php and a ba…   7                                       [OK]
nyanpass/apache2.2-php5.2.17      Apache2.2-PHP5.2.17 with XDebug                 5                                       [OK]
ascdc/apache2-php7                apache2 + php7                                  5                                       [OK]
ascdc/apache2-php56               apache2-php56                                   2                                       [OK]
uofa/apache2-php7                 Streamlined web hosting image based on Ubunt…   2                                       [OK]
jmferrer/apache2-reverse-proxy    Dockerized apache2 reverse proxy service.       2
uofa/apache2-php7-dev             Streamlined web hosting image based on Ubunt…   2                                       [OK]
oberonamsterdam/apache24-fpm      Default Oberon apache setup using PHP-FPM, r…   1                                       [OK]
antage/apache2-php7               Docker image for running Apache 2.x with PHP…   1                                       [OK]
mhenry07/apache2-utils            Alpine image with apache2-utils (htpasswd, e…   1
caladreas/apache2-openidc-docker  Naieve docker container configuration for Ap…   1                                       [OK]
datenbetrieb/apache2              apache2-webserver                               1                                       [OK]
kstaken/apache2                   This a small Apache2 build that can be exten…   1
enonicio/apache2                  Base Docker image for apache2                   1                                       [OK]
garelp/apache2-galette            Galette installation with php7 and apache2 f…   1
mmorejon/apache2-php5             Base docker image to run PHP applications on…   1                                       [OK]
prometsource/apache2                                                              1
enoniccloud/apache2-letsencrypt   Apache2 container with letsencrypt integrated   0                                       [OK]
1ea0b37acbfe/apache2                                                              0
jeffutter/apache2-php             Docker Image with apache2, php5.5 and php mo…   0                                       [OK]
nfqlt/apache24-fastcgi                                                            0
amarsingh3d/apache2.4-php7.2                                                      0
symbiote/apache2                  Ubuntu 16.04 apache2 image, with default vho…   0
Dockermachine@Dockermachine:~/stackdemo$
```

Fix: To pull http image from public repository pull **httpd image**




```
Dockermachine@Dockermachine:~/stackdemo$ docker pull httpd
Using default tag: latest
latest: Pulling from library/httpd
8559a31e96f4: Pull complete
bd517d441028: Pull complete
f67007e59c3c: Pull complete
83c578481926: Pull complete
f3cbcb88690d: Pull complete
Digest: sha256:387f896f9b6867c7fa543f7d1a686b0ebe777ed13f6f11efc8b94bec743a1e51
Status: Downloaded newer image for httpd:latest
docker.io/library/httpd:latest
Dockermachine@Dockermachine:~/stackdemo$
```

7.3 Unable to Docker Login or Create a new Docker ID

Issue: azureuser@Test:~$ 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: docker_id
Password:
Error saving credentials: error storing credentials - err: exit status 1, out: `Cannot autolaunch D-Bus without X11 $DISPLAY`
azureuser@Test:~$

Reason: This would have happened because of a glitch in the process of Docker Installation.

Fix: Look into these two links from StackOverflow (shared below) if it still doesn't work out please uninstall and reinstall docker to fix this issue.

Links - https://stackoverflow.com/questions/50151833/cannot-login-to-docker-account

https://stackoverflow.com/questions/51222996/docker-login-fails-on-a-server-with-no-x11-installed

8 EXTRA DOCKER QUESTION

Question: I am expected to install python I believe as the base image, with some of the steps included in my original email, and at the end the output is suppose to give me the SHA1 value.
So after many tries, I was doing a stdin of the Dockerfile to a Dockerfile created using touch command. But in the bash script, I 1st updated the system, installed docker, started docker, then created a Dockerfile with the contents of the dockerfile standard input with command <<EOF. But I still didn't get the right output.

Answer:

```
root@docker:/dockers# cat 1.sh
#!/bin/bash
echo "Dockerfile Creation"
> ~/mydockerfile.df
echo "FROM centos" >> ~/mydockerfile.df
echo "yum install python3* -y"  >> ~/mydockerfile.df
find ~/ -type f -name mydockerfile.df -exec md5sum {} \;
root@docker:/dockers# bash 1.sh
Dockerfile Creation
44ac8eb5a5adaf1071c4a4de0e890233  /root/mydockerfile.df
root@docker:/dockers#
```

9 SUMMARY

In this activity Guide we learned:

- Docker Installation Steps on Ubuntu 20.04 server
- Working with Container
- Working with Docker Images