

ECS Lab-8

1. Install docker

Command: **sudo apt install docker.io**

```
12 packages can be upgraded. Run 'apt list --upgradable' to see them.
guru@ubuntu:~/Desktop$ sudo apt install docker.io
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gimp-data i965-va-driver intel-media-va-driver libaacs0 libamd2 libaom0 libavcodec58
  libavformat58 libavutil56 libbabi-0.1-0 libbdplus0 libblas3 libbluray2 libcamd2 libccolamd2
  libcholmod3 libchromaprint1 libcodecs2-0.9 libde265-0 libfprint-2-tod1 libgegl-0.4-0
  libgegl-common libgfortran5 libgimp2.0 libgme0 libgsm1 libheif1 libigdgmm1 libilmbase24
  liblapack3 libllvm9 libmetis5 libmng2 libmypaint-1.5-1 libmypaint-common libopenexr24
  libopenmpt0 libSDL2-2.0-0 libshine3 libsnappy1v5 libssh-gcrypt-4 libswresample3 libswscale5
  libumfpack5 libva-drm2 libva-x11-2 libva2 libvdpau1 libx264-155 libx265-179 libxvidcore4
  libzvbi-common libzvbi0 mesa-va-drivers mesa-va-drivers mesa-va-drivers vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  bridge-utils cgroupfs-mount containerd pigz runc ubuntu-fan
Suggested packages:
  ifupdown aufs-tools btrfs-progs debootstrap docker-doc rinse zfs-fuse | zfsutils
The following NEW packages will be installed:
  bridge-utils cgroupfs-mount containerd docker.io pigz runc ubuntu-fan
0 upgraded, 7 newly installed, 0 to remove and 42 not upgraded.
Need to get 69.3 MB of archives.
After this operation, 333 MB of additional disk space will be used.
Do you want to continue? [Y/n]
```

2. Pull the image alpine from docker

Command: **sudo docker pull alpine**

```
Processing triggers for Man-db (2.9.1-1) ...
guru@ubuntu:~/Desktop$ sudo docker pull alpine
Using default tag: latest
latest: Pulling from library/alpine
188c0c94c7c5: Pull complete
Digest: sha256:c0e9560cda118f9ec63ddefb4a173a2b2a0347082d7dff7dc14272e7841a5b5a
Status: Downloaded newer image for alpine:latest
docker.io/library/alpine:latest
guru@ubuntu:~/Desktop$
```

3. Check the created image alpine

Command: **sudo docker images**

```
docker.io/library/alpine:latest
guru@ubuntu:~/Desktop$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine              latest             d6e46aa2470d       3 weeks ago        5.57MB
guru@ubuntu:~/Desktop$
```

- Command: **sudo docker run -it -d <image id>**

```
guru@ubuntu:~/Desktop$ sudo docker run -it -d d6e46aa2470d
08541c7c765c33493363b681438ee291ba298e35e29c4f8f46b6d98199e456a2
guru@ubuntu:~/Desktop$ sudo docker ps
```

CONTAINER ID	IMAGE	COMMAND	CREATED	STATUS
08541c7c765c	d6e46aa2470d	"/bin/sh"	3 minutes ago	Up 3 minutes

```
guru@ubuntu:~/Desktop$
```

- Command: **sudo docker run -i -t <image id> /bin/sh**

- Check gedit is present in the create container
- Since gedit is not found install using **apk add gedit** command

```
guru@ubuntu:~/Desktop$ sudo docker run -i -t d6e46aa2470d /bin/sh
/ # ls
bin      etc      lib      mnt      proc     run      srv      tmp      var
dev      home     media    opt      root    /sbin     sys      usr
/ # touch sample
/ # ls
bin      etc      lib      mnt      proc     run     /sbin     sys      usr
dev      home     media    opt      root     sample   srv      tmp      var
/ # gedit sample
/bin/sh: gedit: not found
/ #
```

- ## 6. Installed gedit

```
/ # gedit sample  
/bin/sh: gedit: not found  
/ # apk add gedit  
fetch http://dl-cdn.alpinelinux.org/alpine/v3.12/main/x86_64/APKINDEX.tar.gz  
fetch http://dl-cdn.alpinelinux.org/alpine/v3.12/community/x86_64/APKINDEX.tar.gz  
(1/79) Installing libxau (1.0.9-r0)  
(2/79) Installing libbsd (0.10.0-r0)  
(3/79) Installing libxdmcp (1.1.3-r0)  
(4/79) Installing libxcb (1.14-r1)  
(5/79) Installing libx11 (1.6.12-r0)  
(6/79) Installing libffi (3.3-r2)  
(7/79) Installing libintl (0.20.2-r0)  
(8/79) Installing libblkid (2.35.2-r0)  
(9/79) Installing libmount (2.35.2-r0)  
(10/79) Installing pcre (8.44-r0)  
(11/79) Installing glib (2.64.6-r0)  
(12/79) Installing pkgconf (1.7.2-r0)  
(13/79) Installing xz-libs (5.2.5-r0)  
(14/79) Installing libxml2 (2.9.10-r5)  
(15/79) Installing shared-mime-info (1.15-r0)  
(16/79) Installing hicolor-icon-theme (0.17-r1)  
(17/79) Installing libjpeg-turbo (2.0.5-r0)  
(18/79) Installing libpng (1.6.37-r1)  
(19/79) Installing tiff (4.1.0-r0)
```

7. Listing all running containers

```
guru@ubuntu:~/Desktop$ sudo docker ps
[sudo] password for guru:
CONTAINER ID        IMAGE               COMMAND             CREATED             STATUS
6a58d646d4c2       d6e46aa2470d       "/bin/sh"          2 minutes ago      Up 2 minutes
08541c7c765c       d6e46aa2470d       "/bin/sh"          8 minutes ago      Up 8 minutes
guru@ubuntu:~/Desktop$ sudo docker diff 6a58d646d4c2
A /sample
C /var
C /var/cache
A /var/cache/fontconfig
C /var/cache/apk
A /var/cache/apk/APKINDEX.2c4ac24e.tar.gz
A /var/cache/apk/APKINDEX.40a3604f.tar.gz
C /root
A /root/.ash_history
```

8. Checking whether the gedit is installed or not.

Command: **sudo docker diff <contained Id>** (list of all changes in container)

sudo docker diff <container Id> | grep (list only gedit files)

```
guru@ubuntu:~/Desktop$ sudo docker diff 6a58d646d4c2 | grep gedit
A /usr/lib/gedit
A /usr/lib/gedit/girepository-1.0
A /usr/lib/gedit/girepository-1.0/Gedit-3.0.typelib
A /usr/lib/gedit/libgedit-3.36.so
A /usr/lib/gedit/plugins
A /usr/lib/gedit/plugins/time.plugin
A /usr/lib/gedit/plugins/externaltools
A /usr/lib/gedit/plugins/externaltools/capture.py
A /usr/lib/gedit/plugins/externaltools/filelookup.py
A /usr/lib/gedit/plugins/externaltools/__init__.py
A /usr/lib/gedit/plugins/externaltools/appactivatable.py
A /usr/lib/gedit/plugins/externaltools/linkparsing.py
A /usr/lib/gedit/plugins/externaltools/manager.py
A /usr/lib/gedit/plugins/externaltools/outputpanel.py
A /usr/lib/gedit/plugins/externaltools/windowactivatable.py
A /usr/lib/gedit/plugins/externaltools/functions.py
A /usr/lib/gedit/plugins/externaltools/library.py
A /usr/lib/gedit/plugins/libfilebrowser.so
A /usr/lib/gedit/plugins/libtime.so
A /usr/lib/gedit/plugins/sort.plugin
A /usr/lib/gedit/plugins/spell.plugin
A /usr/lib/gedit/plugins/snippets
A /usr/lib/gedit/plugins/snippets/manager.py
A /usr/lib/gedit/plugins/snippets/substitutionparser.py
A /usr/lib/gedit/plugins/snippets/__init__.py
A /usr/lib/gedit/plugins/snippets/appactivatable.py
A /usr/lib/gedit/plugins/snippets/completion.py
A /usr/lib/gedit/plugins/snippets/document.py
```

9. Stop and commit the container:

- Stop the container using command: **sudo docker stop <containerID>**
- To commit all the changes in the container use command :
sudo docker commit <container ID> <name of the repository >
- Check whether the container created or not
sudo docker images (list all images)

```
guru@ubuntu:~/Desktop$ sudo docker stop 6a58d646d4c2
6a58d646d4c2
guru@ubuntu:~/Desktop$ sudo docker commit 6a58d646d4c2 alpine-gedit
sha256:b3ae805d403dffff71492ea070fc8790173aca1e29d99d4a7288d5952002e0dc3
guru@ubuntu:~/Desktop$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine-gedit        latest             b3ae805d403d       7 seconds ago      101MB
alpine              latest            d6e46aa2470d       3 weeks ago        5.57MB
guru@ubuntu:~/Desktop$
```

10. Docker Login

- Create an account in dockerhub and login through the created account in terminal using
Command: **sudo docker login**

```
guru@ubuntu:~/Desktop$ sudo 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: gurusaran
Password:
WARNING! Your password will be stored unencrypted in /root/.docker/config.json.
Configure a credential helper to remove this warning. See
https://docs.docker.com/engine/reference/commandline/login/#credentials-store

Login Succeeded
```

11. To push the newly created image **alpine-gedit**, first we should tag the image using this
command: **sudo docker tag alpine-gedit gurusaran/guru_saran_repo:alpine-gedit**

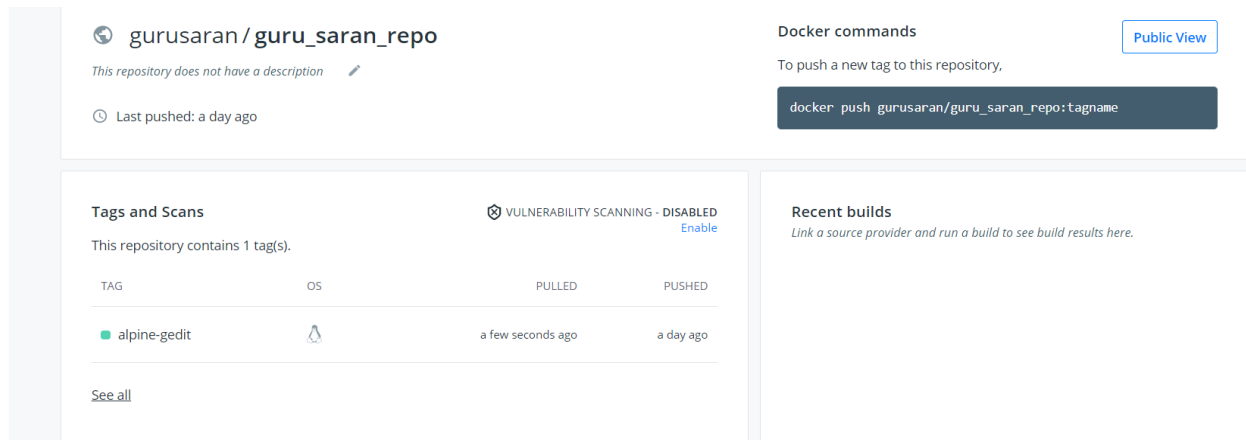
12. To push the image to my repo use this command:

sudo docker push gurusaran/guru_saran_repo:alpine-gedit

```
Login Succeeded
guru@ubuntu:~/Desktop$ sudo docker tag alpine-gedit gurusaran/guru_saran_repo:alpine-gedit
guru@ubuntu:~/Desktop$ sudo docker push gurusaran/guru_saran_repo:alpine-gedit
The push refers to repository [docker.io/gurusaran/guru_saran_repo]
9652c08ae118: Pushed
ace0eda3e3be: Layer already exists
alpine-gedit: digest: sha256:8230abe8e0fe8a1dc8f64532580a38d3914f500450eb39e41ece6c4da674064f size: 740
guru@ubuntu:~/Desktop$
```

act input to this VM, move the mouse pointer inside or press Ctrl+G

13. This is the created image pushed into my dockerhub



The screenshot shows the Docker Hub interface for the repository `gurusaran/guru_saran_repo`. The repository is public and has no description. It was last pushed a day ago. The 'Tags and Scans' section shows one tag, `alpine-gegit`, which was pushed a day ago. The 'Recent builds' section is empty. The 'Docker commands' section shows the command `docker push gurusaran/guru_saran_repo:tagname`.

gurusaran / guru_saran_repo
This repository does not have a description
Last pushed: a day ago

Docker commands
To push a new tag to this repository,
`docker push gurusaran/guru_saran_repo:tagname`

Tags and Scans
This repository contains 1 tag(s).
VULNERABILITY SCANNING - DISABLED
Enable

TAG	OS	PULLED	PUSHED
alpine-gegit		a few seconds ago	a day ago

[See all](#)

Recent builds
Link a source provider and run a build to see build results here.

14. Creating a dockerfile to install nmap in an alpine image as shown below:

```
guru@ubuntu:~/Desktop$ cd
guru@ubuntu:~$ mkdir docker
guru@ubuntu:~$ cd docker/
guru@ubuntu:~/docker$ touch dockerfile
guru@ubuntu:~/docker$ gedit dockerfile
guru@ubuntu:~/docker$ cat dockerfile
FROM alpine

RUN apk update
RUN apk add nmap
CMD /bin/echo "Hello This is my first dockerfile"
guru@ubuntu:~/docker$
```

15. Generating docker image using dockerfile.

Command: **sudo docker build .**

```
guru@ubuntu:~/docker$ sudo docker build .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM alpine
--> d6e46aa2470d
Step 2/4 : RUN apk update
--> Running in 24ba4593783e
fetch http://dl-cdn.alpinelinux.org/alpine/v3.12/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.12/community/x86_64/APKINDEX.tar.gz
v3.12.1-32-g3dc1dba8df [http://dl-cdn.alpinelinux.org/alpine/v3.12/main]
v3.12.1-33-ge462514615 [http://dl-cdn.alpinelinux.org/alpine/v3.12/community]
OK: 12744 distinct packages available
Removing intermediate container 24ba4593783e
--> 7adc2f375bef
Step 3/4 : RUN apk add nmap
--> Running in d06c8f6b1771
(1/7) Installing libgcc (9.3.0-r2)
(2/7) Installing lua5.3-libs (5.3.5-r6)
(3/7) Installing libpcap (1.9.1-r2)
(4/7) Installing pcre (8.44-r0)
(5/7) Installing libssh2 (1.9.0-r1)
(6/7) Installing libstdc++ (9.3.0-r2)
(7/7) Installing nmap (7.80-r2)
Executing busybox-1.31.1-r19.trigger
OK: 20 MiB in 21 packages
Removing intermediate container d06c8f6b1771
--> e413861649c2
Step 4/4 : CMD /bin/echo "Hello This is my first dockerfile"
--> Running in 3abd5b5bb06a
Removing intermediate container 3abd5b5bb06a
--> 72d79a35c2fc
Successfully built 72d79a35c2fc
guru@ubuntu:~/docker$
```

16. After generating docker image run the image using command **sudo docker run <imageId>**

```
---> 72d79a35c2fc
Successfully built 72d79a35c2fc
guru@ubuntu:~/docker$ sudo docker run 72d79a35c2fc
Hello This is my first dockerfile
guru@ubuntu:~/docker$
```

17. Checking the newly created image

Command: **sudo docker images** (list all Images)

- Since newly created images has no name give the name as **alpine-nmap** using command **sudo docker build -t alpine-nmap**

```
guru@ubuntu:~/docker$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>            72d79a35c2fc       About a minute ago  22.3MB
gurusaran/guru_saran_repo  alpine-gedit      b3ae805d403d       16 minutes ago     101MB
alpine-gedit         latest            b3ae805d403d       16 minutes ago     101MB
alpine               latest            d6e46aa2470d       3 weeks ago        5.57MB
guru@ubuntu:~/docker$ sudo docker build -t alpine-nmap
"docker build" requires exactly 1 argument.
See 'docker build --help'.

Usage:  docker build [OPTIONS] PATH | URL | -

Build an image from a Dockerfile
guru@ubuntu:~/docker$ sudo docker build -t alpine-nmap .
Sending build context to Docker daemon 2.048kB
Step 1/4 : FROM alpine
---> d6e46aa2470d
Step 2/4 : RUN apk update
---> Using cache
---> 7adc2f375bef
Step 3/4 : RUN apk add nmap
---> Using cache
---> e413861649c2
Step 4/4 : CMD /bin/echo "Hello This is my first dockerfile"
---> Using cache
---> 72d79a35c2fc
Successfully built 72d79a35c2fc
Successfully tagged alpine-nmap:latest
guru@ubuntu:~/docker$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
alpine-nmap         latest            72d79a35c2fc       2 minutes ago      22.3MB
gurusaran/guru_saran_repo  alpine-gedit      b3ae805d403d       17 minutes ago     101MB
alpine-gedit         latest            b3ae805d403d       17 minutes ago     101MB
alpine               latest            d6e46aa2470d       3 weeks ago        5.57MB
guru@ubuntu:~/docker$
```

18. Checking whether the nmap installed or not by entering into the shell

```
guru@ubuntu:~/docker$ sudo docker run -i -t alpine-nmap /bin/sh
/ # nmap
Nmap 7.80 ( https://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL <inputfilename>: Input from list of hosts/networks
  -iR <num hosts>: Choose random targets
  --exclude <host1[,host2][,host3],...>: Exclude hosts/networks
  --excludefile <exclude_file>: Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PY/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[portlist]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers <serv1[,serv2],...>: Specify custom DNS servers
```


19. Creating a java program for docker

```
guru@ubuntu: ~/docker x guru@ubuntu: ~/docker x guru@ubuntu: ~/docker x
guru@ubuntu:~/docker$ ls
dockerfile
guru@ubuntu:~/docker$ touch HelloWorld.java
guru@ubuntu:~/docker$ nano HelloWorld.java
guru@ubuntu:~/docker$ cat HelloWorld.java
class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello, World!");
    }
}
guru@ubuntu:~/docker$ javac HelloWorld.java

Command 'javac' not found, but can be installed with:

sudo apt install openjdk-11-jdk-headless # version 11.0.9.1+1-0ubuntu1~20.04, or
sudo apt install default-jdk # version 2:1.11-72
sudo apt install openjdk-8-jdk-headless # version 8u275-b01-0ubuntu1~20.04
sudo apt install ecj # version 3.16.0-1
sudo apt install openjdk-13-jdk-headless # version 13.0.4+8-1~20.04
sudo apt install openjdk-14-jdk-headless # version 14.0.2+12-1~20.04

guru@ubuntu:~/docker$ javac HelloWorld.java
guru@ubuntu:~/docker$ java HelloWorld
Hello, World!
guru@ubuntu:~/docker$
```

20. Program for docker file to execute a java program

```
guru@ubuntu:~/docker$ ls
dockerfile HelloWorld.class HelloWorld.java
guru@ubuntu:~/docker$ gedit dockerfile
guru@ubuntu:~/docker$ cat dockerfile
# Using alpine linux images
FROM alpine
WORKDIR /root/hello
COPY HelloWorld.java /root/hello

#install jdk

RUN apk add openjdk8
ENV JAVA_HOME /usr/lib/jvm/java-1.8-openjdk
ENV PATH $PATH:$JAVA_HOME/bin

#compile helloworld

RUN javac HelloWorld.java
ENTRYPOINT java HelloWorld
```

21. Generating docker image using dockerfile.

Command: **sudo docker build .**

```
guru@ubuntu:~/docker$ sudo docker build .
Sending build context to Docker daemon 4.096kB
Step 1/8 : FROM alpine
--> d6e46aa2470d
Step 2/8 : WORKDIR /root/hello
--> Using cache
--> c2412a7dac0a
Step 3/8 : COPY HelloWorld.java /root/hello
--> Using cache
--> 48892de51147
Step 4/8 : RUN apk add openjdk8
--> Running in 3b72c1ccd51c
fetch http://dl-cdn.alpinelinux.org/alpine/v3.12/main/x86_64/APKINDEX.tar.gz
fetch http://dl-cdn.alpinelinux.org/alpine/v3.12/community/x86_64/APKINDEX.tar.gz
(1/41) Installing libffi (3.3-r2)
(2/41) Installing p11-kit (0.23.20-r5)
(3/41) Installing libtasn1 (4.16.0-r1)
(4/41) Installing p11-kit-trust (0.23.20-r5)
Executing ca-certificates-20191127-r1.trigger
Executing java-common-0.2-r0.trigger
OK: 104 MiB in 55 packages
Removing intermediate container 3b72c1ccd51c
--> cae0b3c54de9
Step 5/8 : ENV JAVA_HOME /usr/lib/jvm/java-1.8-openjdk
--> Running in d77c85f33972
Removing intermediate container d77c85f33972
--> ed5c62a72deb
Step 6/8 : ENV PATH $PATH:$JAVA_HOME/bin
--> Running in 3827848f82c1
Removing intermediate container 3827848f82c1
--> 4b00533b0701
Step 7/8 : RUN javac HelloWorld.java
--> Running in 0959f84734ba
Removing intermediate container 0959f84734ba
--> 3384c757285e
Step 8/8 : ENTRYPOINT java HelloWorld
--> Running in 30921ecb8991
Removing intermediate container 30921ecb8991
--> afb2419fe822
Successfully built afb2419fe822
guru@ubuntu:~/docker$
```

22. After generating docker image run the image using command **sudo docker run <imageId>** and check the image is create or not using command: **sudo docker images**

```
Successfully built afb2419fe822
guru@ubuntu:~/docker$ sudo docker run afb2419fe822
Hello, World!
guru@ubuntu:~/docker$ sudo docker images
REPOSITORY          TAG                 IMAGE ID            CREATED             SIZE
<none>              <none>             afb2419fe822       9 minutes ago      108MB
alpine-nmap         latest             72d79a35c2fc       37 minutes ago     22.3MB
gurusaran/guru_saran_repo  alpine-gedit      b3ae805d403d       52 minutes ago     101MB
alpine-gedit        latest            b3ae805d403d       52 minutes ago     101MB
alpine              latest            d6e46aa2470d       3 weeks ago        5.57MB
guru@ubuntu:~/docker$
```


23. Since newly created images has no name give the name as **alpine-java** using command **sudo docker build -t alpine-java**

Check the newly renamed docker image

Command: **sudo docker images** (list all the images)

```
guru@ubuntu:~/docker$ sudo docker build -t alpine-java .
Sending build context to Docker daemon 4.096kB
Step 1/8 : FROM alpine
--> d6e46aa2470d
Step 2/8 : WORKDIR /root/hello
--> Using cache
--> c2412a7dac0a
Step 3/8 : COPY HelloWorld.java /root/hello
--> Using cache
--> 48892de51147
Step 4/8 : RUN apk add openjdk8
--> Using cache
--> cae0b3c54de9
Step 5/8 : ENV JAVA_HOME /usr/lib/jvm/java-1.8-openjdk
--> Using cache
--> ed5c62a72deb
Step 6/8 : ENV PATH $PATH:$JAVA_HOME/bin
--> Using cache
--> 4b00533b0701
Step 7/8 : RUN javac HelloWorld.java
--> Using cache
--> 3384c757285e
Step 8/8 : ENTRYPOINT java HelloWorld
--> Using cache
--> afb2419fe822
Successfully built afb2419fe822
Successfully tagged alpine-java:latest
guru@ubuntu:~/docker$ sudo docker images
```

REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
alpine-java	latest	afb2419fe822	11 minutes ago	108MB
alpine-nmap	latest	72d79a35c2fc	39 minutes ago	22.3MB
gurusaran/guru_saran_repo	alpine-gedit	b3ae805d403d	54 minutes ago	101MB
alpine-gedit	latest	b3ae805d403d	54 minutes ago	101MB
alpine	latest	d6e46aa2470d	3 weeks ago	5.57MB

```
guru@ubuntu:~/docker$
```

Creating multiple images

1. Install docker-compose

Command: **sudo apt install docker-compose**

```
Building dependency tree
Reading state information... Done
33 packages can be upgraded. Run 'apt list --upgradable' to see them.
guru@ubuntu:~/docker$ sudo apt install docker-compose
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following packages were automatically installed and are no longer required:
  gimp-data i965-va-driver intel-media-va-driver libaacs0 libamd2 libaom0 libavcodec58
  libavformat58 libavutil56 libbabl-0.1-0 libbdplus0 libblas3 libbluray2 libcamd2 libccolamd2
  libcholmod3 libchromaprint1 libcodec2-0.9 libde265-0 libffprint-2-tod1 libgegl-0.4-0
  libgegl-common libgfortran5 libgimp2.0 libgme0 libgsm1 libheif1 libigdgmm1 libilmbase24
  liblapack3 libllvm9 libmetis5 libmng2 libmypaint-1.5-1 libmypaint-common libopenexr24
  libopenmpt0 libSDL2-2.0-0 libshine3 libsnappy1v5 libssh-gcrypt-4 libswresample3 libswscale5
  libumfpack5 libva-drm2 libva-x11-2 libva2 libvdpau1 libx264-155 libx265-179 libxvidcore4
  libzvb-common libzvb0 linux-headers-5.4.0-48 linux-headers-5.4.0-48-generic
  linux-image-5.4.0-48-generic linux-modules-5.4.0-48-generic
  linux-modules-extra-5.4.0-48-generic mesa-va-drivers mesa-vdpau-drivers va-driver-all
  vdpau-driver-all
Use 'sudo apt autoremove' to remove them.
The following additional packages will be installed:
  python3-attr python3-cached-property python3-distutils python3-docker python3-dockerpty
```

2. creating docker-compose file(.yml) using web and database version

```
Setting up python3-jsonschema (3.2.0-0ubuntu2) ...
Setting up docker-compose (1.25.0-1) ...
Processing triggers for man-db (2.9.1-1) ...
guru@ubuntu:~/docker$ docker-compose -v
docker-compose version 1.25.0, build unknown
guru@ubuntu:~/docker$ touch docker-compose.yml
guru@ubuntu:~/docker$ gedit docker-compose.yml
guru@ubuntu:~/docker$ cat docker-compose.yml
version: "3"
services:
  web:
    image: nginx
  database:
    image: redis
```

3. Save docker-compose.yml and check the validity of this file using **command docker-compose config**

```
guru@ubuntu:~/docker$ docker-compose config
services:
  database:
    image: redis
  web:
    image: nginx
version: '3.0'
guru@ubuntu:~/docker$
```

4. Start the services using command **sudo docker-compose up -d**

```
If it's at a non-standard location, specify the URL with the DOCKER_HOST environment variable
guru@ubuntu:~/docker$ sudo docker-compose up -d
[sudo] password for guru:
Creating network "docker_default" with the default driver
Pulling web (nginx:...)
latest: Pulling from library/nginx
852e50cd189d: Pull complete
a29b129f4109: Pull complete
b3ddf1fa5595: Pull complete
c5df295936d3: Pull complete
232bf38931fc: Pull complete
Digest: sha256:c3a1592d2b6d275bef4087573355827b200b00ffc2d9849890a4f3aa2128c4ae
Status: Downloaded newer image for nginx:latest
Pulling database (redis:...)
latest: Pulling from library/redis
852e50cd189d: Already exists
76190fa64fb8: Pull complete
9cbb1b61e01b: Pull complete
d048021f2aae: Pull complete
6f4b2af24926: Pull complete
1cf1d6922fba: Pull complete
Digest: sha256:5b98e32b58cdbf9f6b6f77072c4915d5ebec43912114031f37fa5fa25b032489
Status: Downloaded newer image for redis:latest
Creating docker_web_1 ... done
Creating docker_database_1 ... done
guru@ubuntu:~/docker$
```

5. Check whether the containers created or not. List all the containers **sudo docker ps**

```
guru@ubuntu:~/docker$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
e7bdc61e216a   nginx     "/docker-entrypoint..." 3 minutes ago  Up 3 minutes  80/tcp
01326b9c0579   redis     "docker-entrypoint.s..." 3 minutes ago  Up 3 minutes  6379/tcp
e20ddf2e2a34   alpine-nmap  "/bin/sh"               45 hours ago  Up 45 hours
08541c7c765c   d6e46aa2470d  "/bin/sh"               46 hours ago  Up 46 hours
guru@ubuntu:~/docker$
```

6. Stop the containers using **sudo docker-compose down** command

```
version          Show the Docker-Compose version information
guru@ubuntu:~/docker$ sudo docker-compose down
Stopping docker_web_1 ... done
Stopping docker_database_1 ... done
Removing docker_web_1 ... done
Removing docker_database_1 ... done
Removing network docker_default
guru@ubuntu:~/docker$ sudo docker ps
CONTAINER ID   IMAGE      COMMAND                  CREATED        STATUS        PORTS
e20ddf2e2a34   alpine-nmap  "/bin/sh"               45 hours ago  Up 45 hours
08541c7c765c   d6e46aa2470d  "/bin/sh"               46 hours ago  Up 46 hours
guru@ubuntu:~/docker$
```

7. To do port mapping Add the port number **ports 8000:80** under image: nginx as shown in the figure below and config the docker-compose.yml to check the validity for .yml file

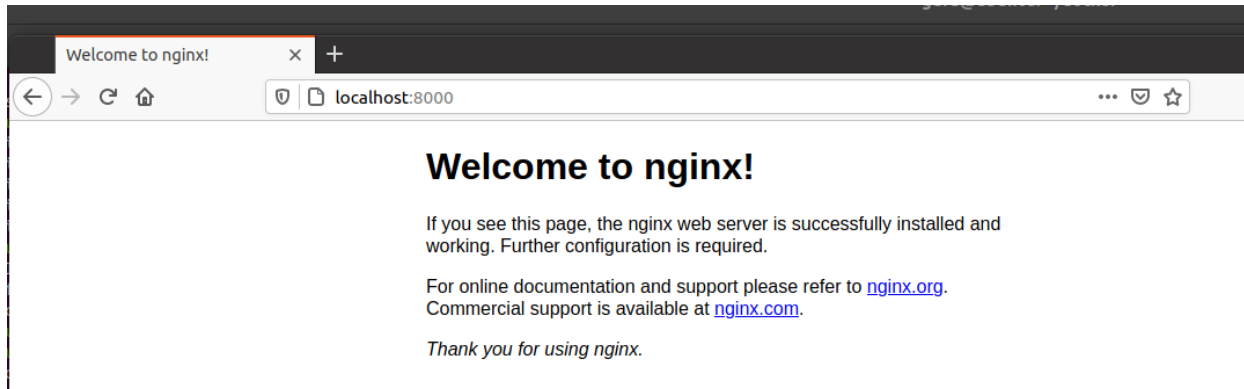
```
08541c7c765c   doe46aa2470d   /bin/sh   46 hours ago   Up 46 hours   affectionate_chandrasekhar
guru@ubuntu:~/docker$ gedit docker-compose.yml
guru@ubuntu:~/docker$ cat docker-compose.yml
version: "3"
services:
  web:
    image: nginx
    ports:
      - 8000:80
  database:
    image: redis
guru@ubuntu:~/docker$ docker-compose config
services:
  database:
    image: redis
  web:
    image: nginx
    ports:
      - 8000:80/tcp
version: '3.0'
```

8. Start the services using command **sudo docker-compose up -d**

```
guru@ubuntu:~/docker$ sudo docker-compose up -d
Creating network "docker_default" with the default driver
Creating docker_database_1 ... done
Creating docker_web_1 ... done
guru@ubuntu:~/docker$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS
1789ab8b96a9   nginx    "/docker-entrypoint. ..." 54 seconds ago Up 47 seconds 0.0.0.0:8000->80/tcp
c81e0c06d622   redis    "docker-entrypoint.s ..." 54 seconds ago Up 47 seconds 6379/tcp
e20ddf2e2a34   alpine-nmap "/bin/sh"               45 hours ago  Up 45 hours
08541c7c765c   doe46aa2470d "/bin/sh"               46 hours ago  Up 46 hours
guru@ubuntu:~/docker$
```

input to this VM, move the mouse pointer inside or press Ctrl+G

9. Verify whether nginx is running or not. Go to web-browser and type localhost:8000 to check.



10. Creating two data base service using command: **sudo docker-compose up -d --scale database=2**

```
08541c7c765c   doe46aa2470d   /bin/sh   46 hours ago   Up 46 hours   affectionate_chandrasekhar
guru@ubuntu:~/docker$ sudo docker-compose up -d --scale database=2
Starting docker_database_1 ... done
docker_web_1 is up-to-date
Creating docker_database_2 ... done
guru@ubuntu:~/docker$ sudo docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED        STATUS        PORTS        NAMES
b3bb20741101   redis    "docker-entrypoint.s ..." 25 seconds ago Up 14 seconds 6379/tcp     docker_database_2
1789ab8b96a9   nginx    "/docker-entrypoint. ..." 6 minutes ago  Up 6 minutes 0.0.0.0:8000->80/tcp  docker_web_1
c81e0c06d622   redis    "docker-entrypoint.s ..." 6 minutes ago  Up 6 minutes 6379/tcp     docker_database_1
e20ddf2e2a34   alpine-nmap "/bin/sh"               45 hours ago  Up 45 hours   sharp_wright
08541c7c765c   doe46aa2470d "/bin/sh"               46 hours ago  Up 46 hours   affectionate_chandrasekhar
guru@ubuntu:~/docker$
```