

ASSESSMENT - 14

Docker

**TO
THE
NEW**™



1. Set the base image to Ubuntu

- Add File Author / Maintainer
- Install Nginx
- Install necessary tools: vim wget curl net-tools
- Remove the default Nginx configuration file
- Copy a configuration file from the current directory
- Expose ports (80)
- Set the default command to execute Nginx when creating a new container

```
FROM ubuntu:latest
MAINTAINER garima
RUN apt-get update && apt-get upgrade -y
RUN apt-get install -y vim wget curl net-tools
RUN apt-get install nginx -y
RUN rm -v /etc/nginx/nginx.conf
ADD new.conf /etc/nginx
EXPOSE 8080
CMD ["nginx", "-g", "daemon off;"]
```

Make a new directory and create a Dockerfile inside it.

```
root@garima:/home/garima/brand# docker build -t nginx .
Sending build context to Docker daemon 4.608kB
Step 1/9 : FROM ubuntu:latest
---> 72300a873c2c
Step 2/9 : MAINTAINER garima
---> Using cache
---> e80465664e36
Step 3/9 : RUN apt-get update && apt-get upgrade -y
---> Using cache
---> df362ff26592
Step 4/9 : RUN apt-get install -y vim wget curl net-tools
---> Using cache
---> 752538addee7
Step 5/9 : RUN apt-get install nginx -y
---> Using cache
---> 82dbf3aa1f4a
```

```
root@garima:/home/garima/brand# docker images
REPOSITORY          TAG                 IMAGE ID            CREATED
SIZE
nginx                latest             e2b9b6aa6cd1       2 minutes ago
218MB
```

2. What is the difference between 'RUN', 'CMD', & 'ENTRYPOINT' in dockerfile?

ANS :

RUN executes the command(s) that you give in a new layer and creates a new image. This is mainly used for installing a new package.

CMD is the default command to be run by the entrypoint. It sets default command and/or parameters, however, we can overwrite those commands or pass in and bypass the default parameters from the command line when docker runs

ENTRYPOINT is used to run the given command. It is used when you want to run a container as an executable.

3. How to connect a docker client to docker daemon running on another host?

On remote:

Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to a traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create

Assign a security group: ☒ Create a **new** security group

☐ Select an **existing** security group

Security group name:

Description:

Type <small>i</small>	Protocol <small>i</small>	Port Range <small>i</small>
SSH ▾	TCP	22
HTTP ▾	TCP	80
Custom TCP F ▾	TCP	2375

Launch Instance ▾ConnectActions ▾

Add filter

<input type="checkbox"/>	Name ▾	Instance ID ▾	Instance Type ▾	Availability Zone ▾	Instance Stat
<input checked="" type="checkbox"/>	docker instance	i-0abe32a9a3741c5ba	t2.micro	us-east-1c	● running

```
garima@garima:~/Downloads$ chmod 400 newawskeypair.pem
garima@garima:~/Downloads$ ssh -i "newawskeypair.pem" ubuntu@ec2-18-206-156-54.c
ompute-1.amazonaws.com
The authenticity of host 'ec2-18-206-156-54.compute-1.amazonaws.com (18.206.156.
54)' can't be established.
ECDSA key fingerprint is SHA256:RG3U0qbf0QC8+/Oj/0/uqLpiR0n0lBYU2IpEvj2QYvw.
Are you sure you want to continue connecting (yes/no)? yes
```

Install docker on your remote vm.

```
GNU nano 2.9.3 /etc/systemd/system/docker.service.d/options.conf

[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H unix:// -H tcp://0.0.0.0:2375
```

```
ubuntu@ip-172-31-81-241:~$ sudo mkdir -p /etc/systemd/system/docker.service.d
ubuntu@ip-172-31-81-241:~$ sudo nano /etc/systemd/system/docker.service.d/options.conf
ubuntu@ip-172-31-81-241:~$ sudo systemctl daemon-reload
ubuntu@ip-172-31-81-241:~$ sudo systemctl restart docker
ubuntu@ip-172-31-81-241:~$
```

On local :

```
garima@garima:~$ DOCKER_HOST=tcp://18.206.156.54:2375 docker info
Client:
 Debug Mode: false

Server:
 Containers: 0
  Running: 0
  Paused: 0
  Stopped: 0
 Images: 0
 Server Version: 19.03.6
 Storage Driver: overlay2
  Backing Filesystem: extfs
  Supports d_type: true
  Native Overlay Diff: true
 Logging Driver: json-file
 Cgroup Driver: cgroupfs
 Plugins:
  Volume: local
  Network: bridge host ipvlan macvlan null overlay
  Log: awslogs fluentd gcplogs gelf journald json-file local logentries splunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: 7ad184331fa3e55e52b890ea95e65ba581ae3429
 runc version: dc9208a3303feef5b3839f4323d9beb36df0a9dd
 init version: fec3683
 Security Options:
  apparmor
  seccomp
   Profile: default
 Kernel Version: 4.15.0-1057-aws
 Operating System: Ubuntu 18.04.3 LTS
 OStype: linux
 Architecture: x86_64
 CPUs: 1
 Total Memory: 983.7MiB
```