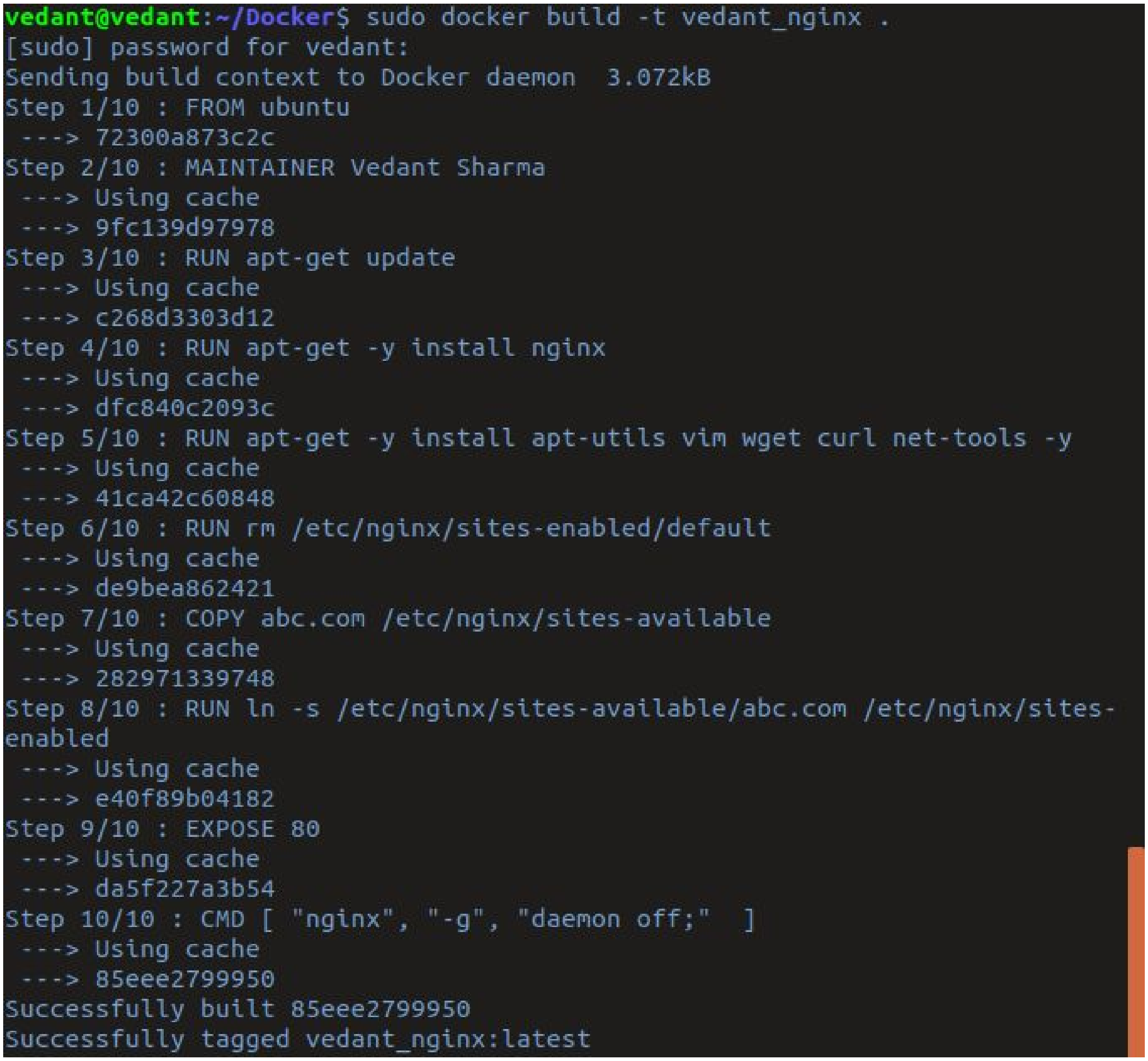
Docker:

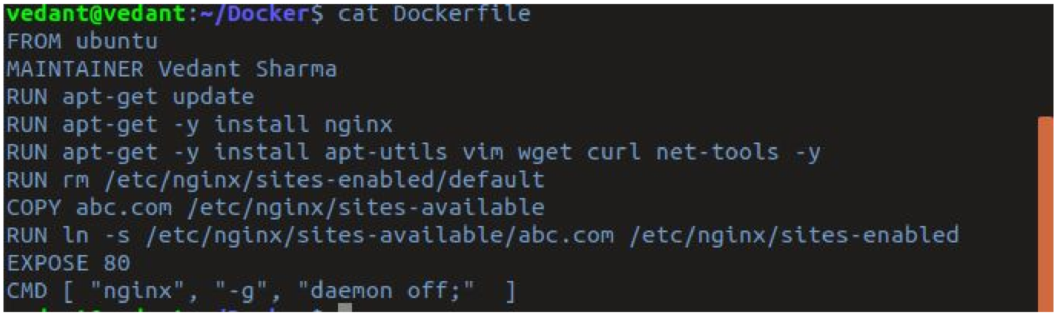
Q1. Set the base image to Ubuntu

- 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



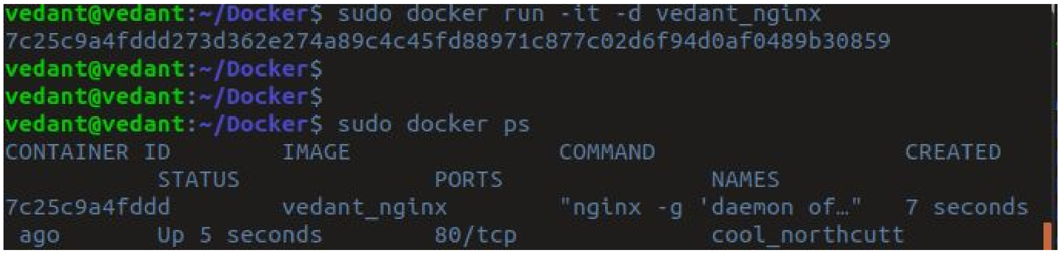
```
vedant@vedant:~/Docker$ sudo docker build -t vedant_nginx .
[sudo] password for vedant:
Sending build context to Docker daemon  3.072kB
Step 1/10 : FROM ubuntu
 ---> 72300a873c2c
Step 2/10 : MAINTAINER Vedant Sharma
 ---> Using cache
 ---> 9fc139d97978
Step 3/10 : RUN apt-get update
 ---> Using cache
 ---> c268d3303d12
Step 4/10 : RUN apt-get -y install nginx
 ---> Using cache
 ---> dfc840c2093c
Step 5/10 : RUN apt-get -y install apt-utils vim wget curl net-tools -y
 ---> Using cache
 ---> 41ca42c60848
Step 6/10 : RUN rm /etc/nginx/sites-enabled/default
 ---> Using cache
 ---> de9bea862421
Step 7/10 : COPY abc.com /etc/nginx/sites-available
 ---> Using cache
 ---> 282971339748
Step 8/10 : RUN ln -s /etc/nginx/sites-available/abc.com /etc/nginx/sites-
enabled
 ---> Using cache
 ---> e40f89b04182
Step 9/10 : EXPOSE 80
 ---> Using cache
 ---> da5f227a3b54
Step 10/10 : CMD [ "nginx", "-g", "daemon off;"  ]
 ---> Using cache
 ---> 85eee2799950
Successfully built 85eee2799950
Successfully tagged vedant_nginx:latest
```

Dockerfile:



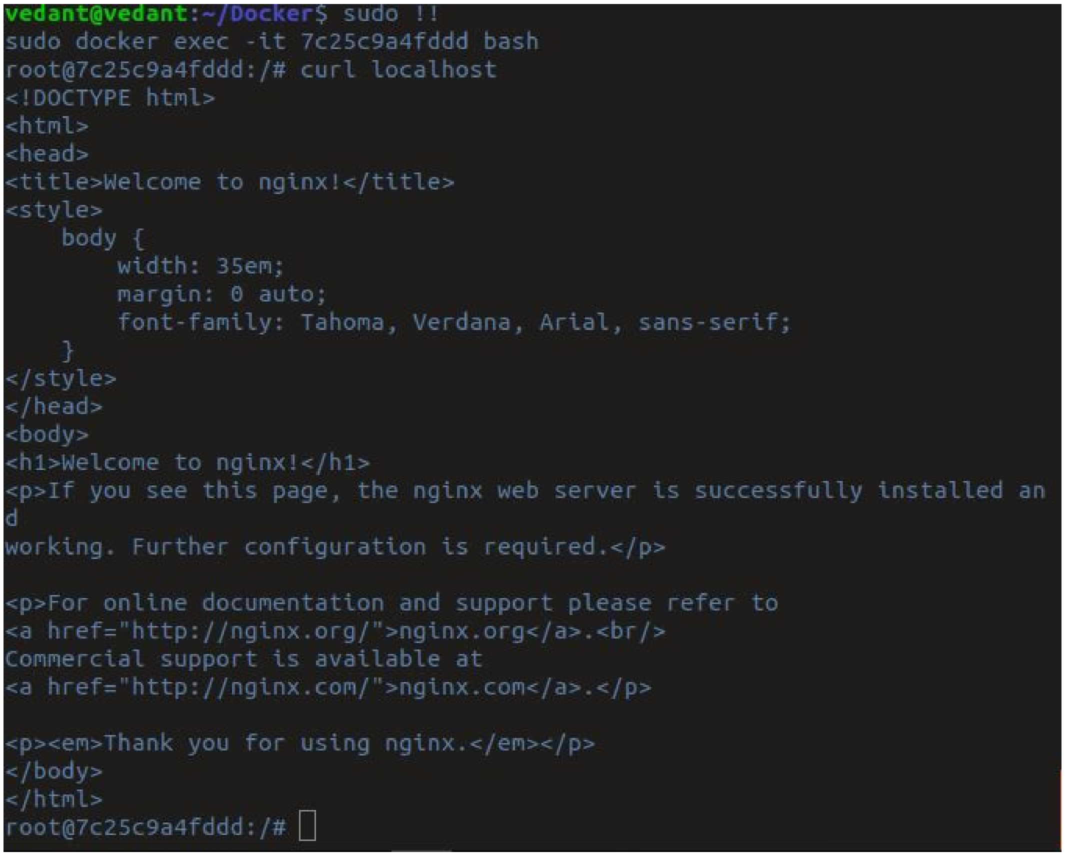
```
vedant@vedant:~/Docker$ cat Dockerfile
FROM ubuntu
MAINTAINER Vedant Sharma
RUN apt-get update
RUN apt-get -y install nginx
RUN apt-get -y install apt-utils vim wget curl net-tools -y
RUN rm /etc/nginx/sites-enabled/default
COPY abc.com /etc/nginx/sites-available
RUN ln -s /etc/nginx/sites-available/abc.com /etc/nginx/sites-enabled
EXPOSE 80
CMD [ "nginx", "-g", "daemon off;"  ]
```

Docker run



```
vedant@vedant:~/Docker$ sudo docker run -it -d vedant_nginx
7c25c9a4fddd273d362e274a89c4c45fd88971c877c02d6f94d0af0489b30859
vedant@vedant:~/Docker$
vedant@vedant:~/Docker$
vedant@vedant:~/Docker$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
           STATUS              PORTS               NAMES
7c25c9a4fddd        vedant_nginx        "nginx -g 'daemon of…"   7 seconds
 ago       Up 5 seconds        80/tcp              cool_northcutt
```

Docker exec to show nginx running:



```
vedant@vedant:~/Docker$ sudo !!
sudo docker exec -it 7c25c9a4fddd bash
root@7c25c9a4fddd:/# curl localhost
<!DOCTYPE html>
<html>
<head>
<title>Welcome to nginx!</title>
<style>
    body {
        width: 35em;
        margin: 0 auto;
        font-family: Tahoma, Verdana, Arial, sans-serif;
    }
</style>
</head>
<body>
<h1>Welcome to nginx!</h1>
<p>If you see this page, the nginx web server is successfully installed an
d
working. Further configuration is required.</p>

<p>For online documentation and support please refer to
<a href="http://nginx.org/">nginx.org</a>.<br/>
Commercial support is available at
<a href="http://nginx.com/">nginx.com</a>.</p>

<p><em>Thank you for using nginx.</em></p>
</body>
</html>
root@7c25c9a4fddd:/#
```

Q2. What is the difference between 'RUN', 'CMD', & 'ENTRYPOINT' in dockerfile?

RUN: the RUN command is mainly used to install a new package on top of the main OS distribution. When you use the RUN command, it will execute the instruction and will create a new layer.

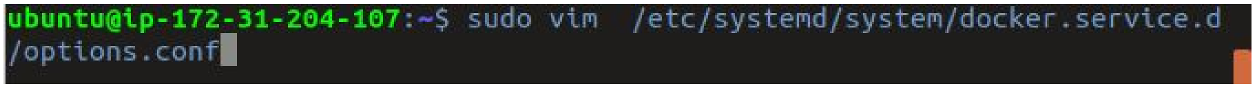
CMD : instruction allows you to set a default command and default parameters which will be executed when docker is run. But these commands and parameters can be overwritten by passing the values over the command line.

ENTRYPOINT instruction should be used when you need your container to be run as an executable. It might look similar to CMD, but in fact, it is different and should be used in a different context. The difference is ENTRYPOINT is that unlike CMD, the command and parameters are not ignored when Docker container runs with command line parameters.

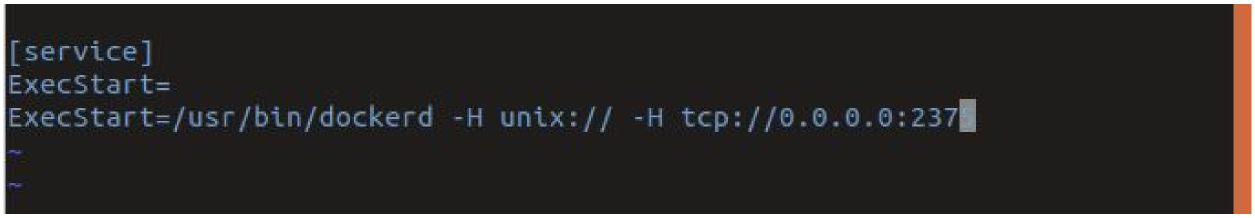
Q3. How to connect docker client to docker daemon running on other host?

On docker client:

1. sudo mkdir -p /etc/systemd/system/docker.service.d
2. sudo vim /etc/systemd/system/docker.service.d/options.conf
3. sudo systemctl daemon-reload
4. sudo systemctl restart docker



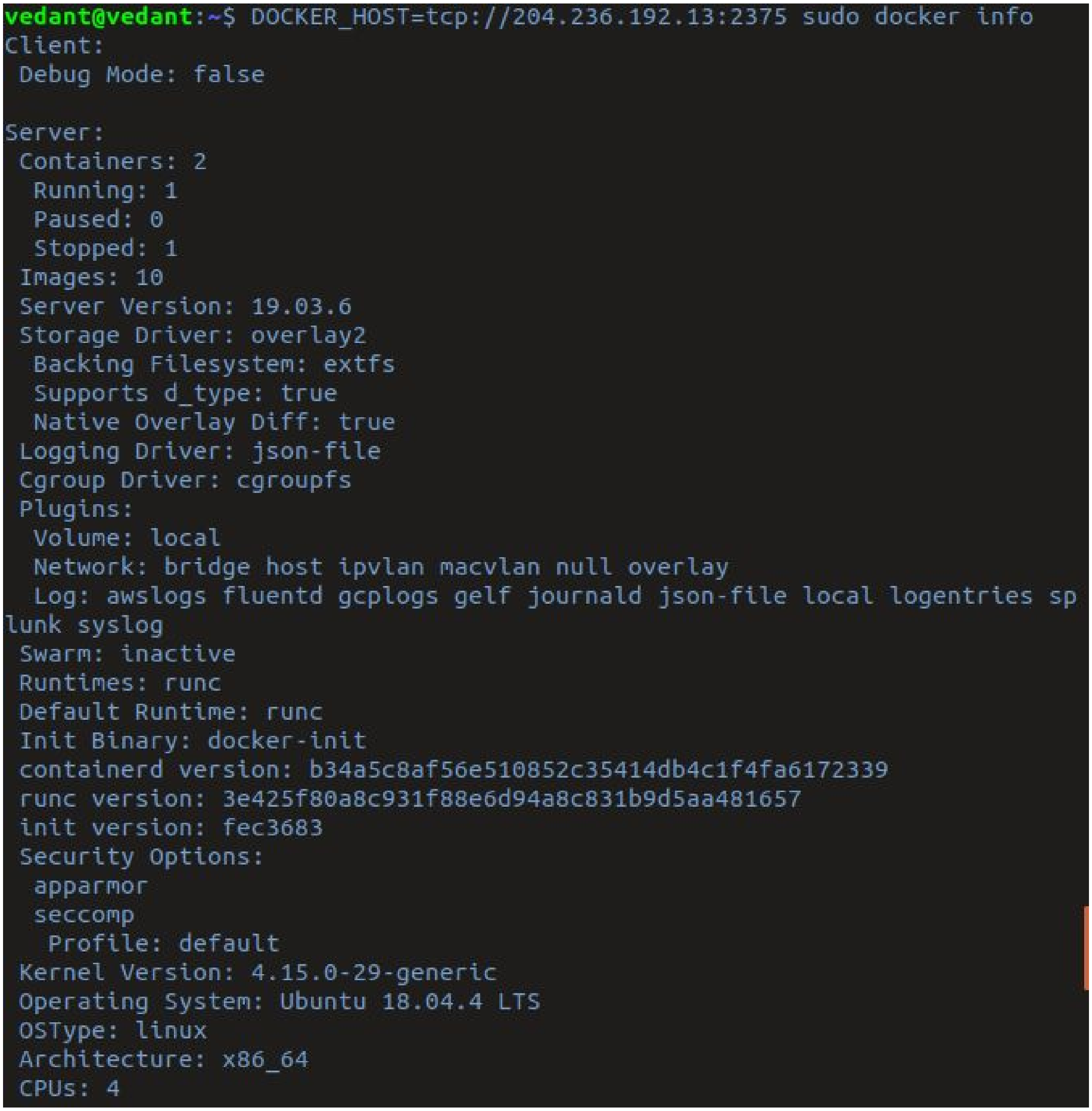
```
ubuntu@ip-172-31-204-107:~$ sudo vim  /etc/systemd/system/docker.service.d
/options.conf
```



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

On Docker host:

1. Export DOCKER_HOST=tcp://204.236.192.13:2375



```
vedant@vedant:~$ DOCKER_HOST=tcp://204.236.192.13:2375 sudo docker info
Client:
 Debug Mode: false

Server:
 Containers: 2
  Running: 1
  Paused: 0
  Stopped: 1
 Images: 10
 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 sp
lunk syslog
 Swarm: inactive
 Runtimes: runc
 Default Runtime: runc
 Init Binary: docker-init
 containerd version: b34a5c8af56e510852c35414db4c1f4fa6172339
 runc version: 3e425f80a8c931f88e6d94a8c831b9d5aa481657
 init version: fec3683
 Security Options:
  apparmor
  seccomp
   Profile: default
 Kernel Version: 4.15.0-29-generic
 Operating System: Ubuntu 18.04.4 LTS
 OSType: linux
 Architecture: x86_64
 CPUs: 4
```