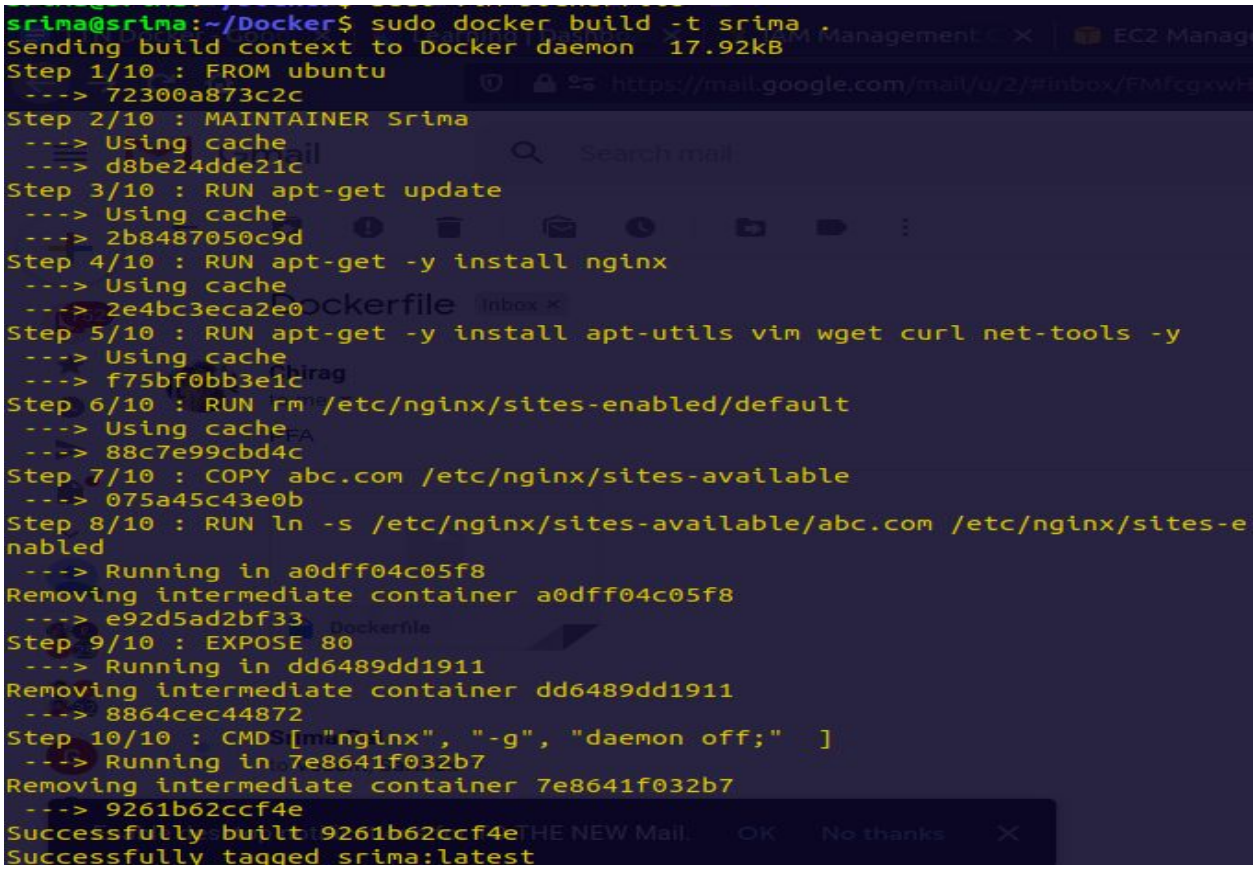
1. **Set the base image to Ubuntu**

- **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**

Dockerfile

```
srima@srima:~/Docker$ cat Dockerfile
FROM ubuntu
MAINTAINER Srima
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;"  ]
srima@srima:~/Docker$
```

Docker Build



```
srima@srima:~/Docker$ sudo docker build -t srima .
Sending build context to Docker daemon  17.92kB
Step 1/10 : FROM ubuntu
 ---> 72300a873c2c
Step 2/10 : MAINTAINER Srima
 ---> Using cache
 ---> d8be24dde21c
Step 3/10 : RUN apt-get update
 ---> Using cache
 ---> 2b8487050c9d
Step 4/10 : RUN apt-get -y install nginx
 ---> Using cache
 ---> 2e4bc3eca2e0
Step 5/10 : RUN apt-get -y install apt-utils vim wget curl net-tools -y
 ---> Using cache
 ---> f75bf0bb3e1c
Step 6/10 : RUN rm /etc/nginx/sites-enabled/default
 ---> Using cache
 ---> 88c7e99cbd4c
Step 7/10 : COPY abc.com /etc/nginx/sites-available
 ---> 075a45c43e0b
Step 8/10 : RUN ln -s /etc/nginx/sites-available/abc.com /etc/nginx/sites-e
nabled
 ---> Running in a0dff04c05f8
Removing intermediate container a0dff04c05f8
 ---> e92d5ad2bf33
Step 9/10 : EXPOSE 80
 ---> Running in dd6489dd1911
Removing intermediate container dd6489dd1911
 ---> 8864cec44872
Step 10/10 : CMD [ "nginx", "-g", "daemon off;"  ]
 ---> Running in 7e8641f032b7
Removing intermediate container 7e8641f032b7
 ---> 9261b62ccf4e
Successfully built 9261b62ccf4e
Successfully tagged srima:latest
```

Docker Run

```
srima@srima:~/Docker$ sudo docker run -it -d --name "Srima1" srima
895ad4c0fb5089bfa28b5c0a31b37071c4fe87c43d259df7b3b25d9596c78fb0
srima@srima:~/Docker$ sudo docker ps
CONTAINER ID        IMAGE               COMMAND                  CREATED
          STATUS              PORTS               NAMES
895ad4c0fb50        srima               "nginx -g 'daemon of…"   16 seconds
 ago      Up 13 seconds       80/tcp              Srima1
e88f10cc84d0        5965ae696033        "nginx -g 'daemon of…"   4 minutes
ago       Up 4 minutes        80/tcp              Srima
```

Going in the container

```
srima@srima:~/Docker$ sudo docker exec -it 895ad4c0fb50 bash
root@895ad4c0fb50:/# 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 instal
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@895ad4c0fb50:/# exit
exit
```

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

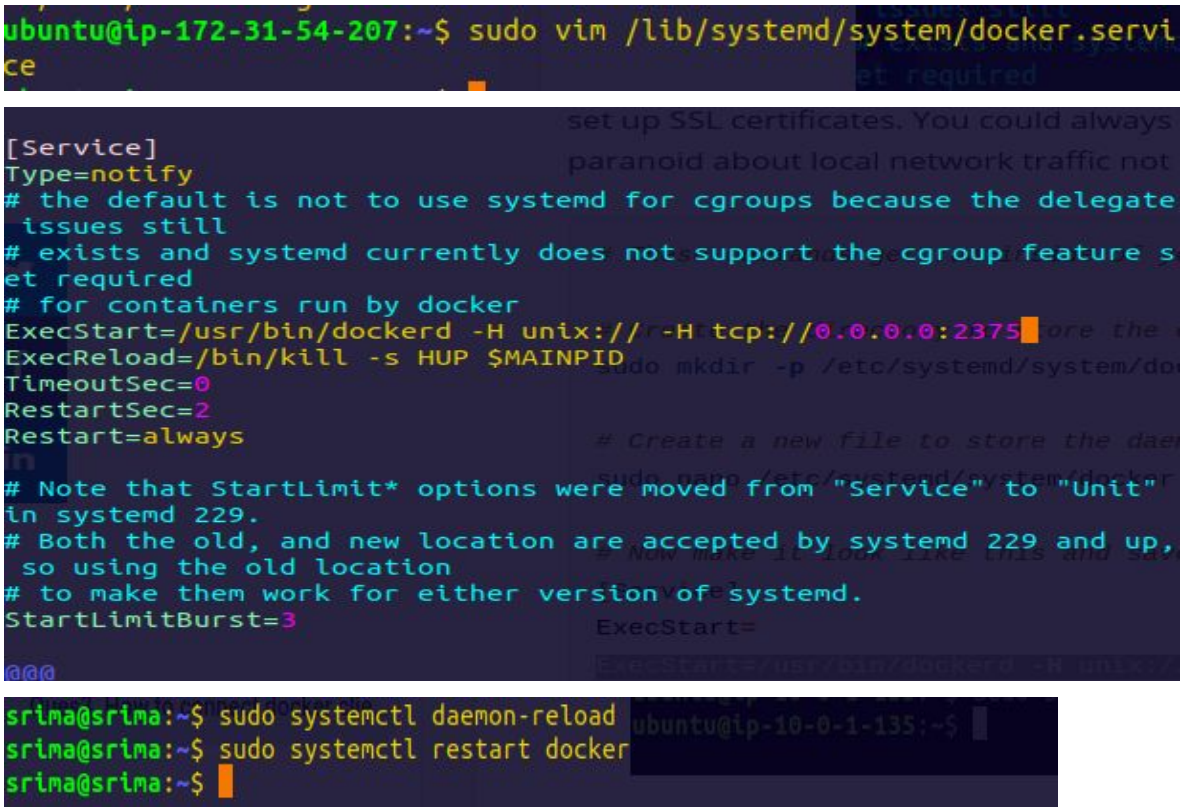
RUN: executes the command(s) that you give in a new layer and creates a new image. This is mainly used for installing a new package.

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.

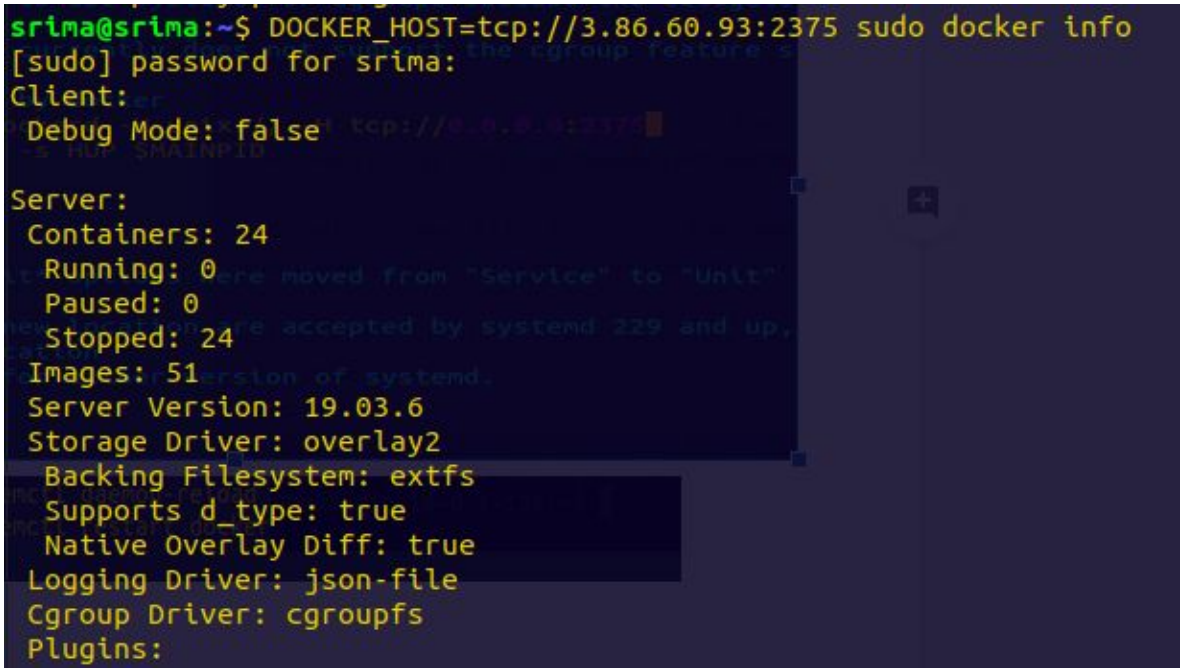
ENTRYPOINT: is the program to run the given command. It is used when you want to run a container as an executable.instruction allows you to configure a container that will run as an executable. It looks similar to CMD, because it also allows you to specify a command with parameters. The difference is ENTRYPOINT command and parameters are not ignored when Docker container runs with command line parameters.

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

On Client



On server


srima@srima:~$ DOCKER_HOST=tcp://3.86.60.93:2375 sudo docker info
[sudo] password for srima:
Client:
Debug Mode: false
Server:
Containers: 24
Running: 0
Paused: 0
Stopped: 24
Images: 51
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: