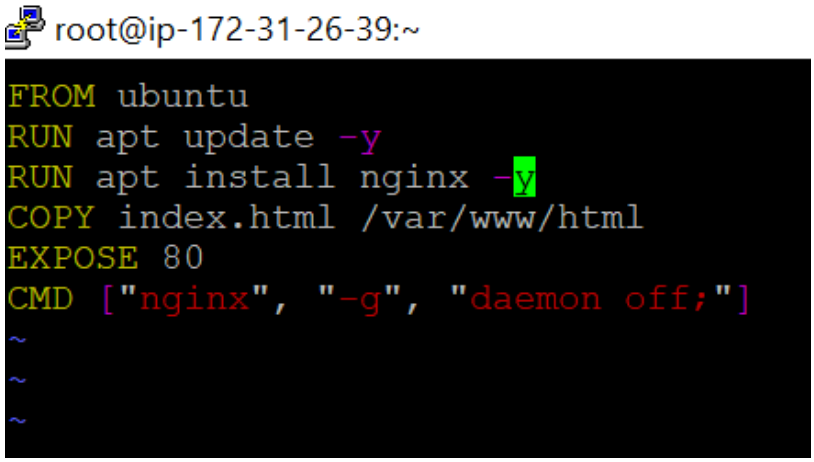
Creating the Container by using Service

1. Create the Image.
2. Tag the Image.
3. Push the Image to Docker Hub.
4. Create the Node.
5. Create the service.

1.Create Image: To create images we should have a Docker file. {vi Docker file. <" D must be in Caps">}

Example: inside the docker file.



Save and exit.

Now create the file as shown above index.html.

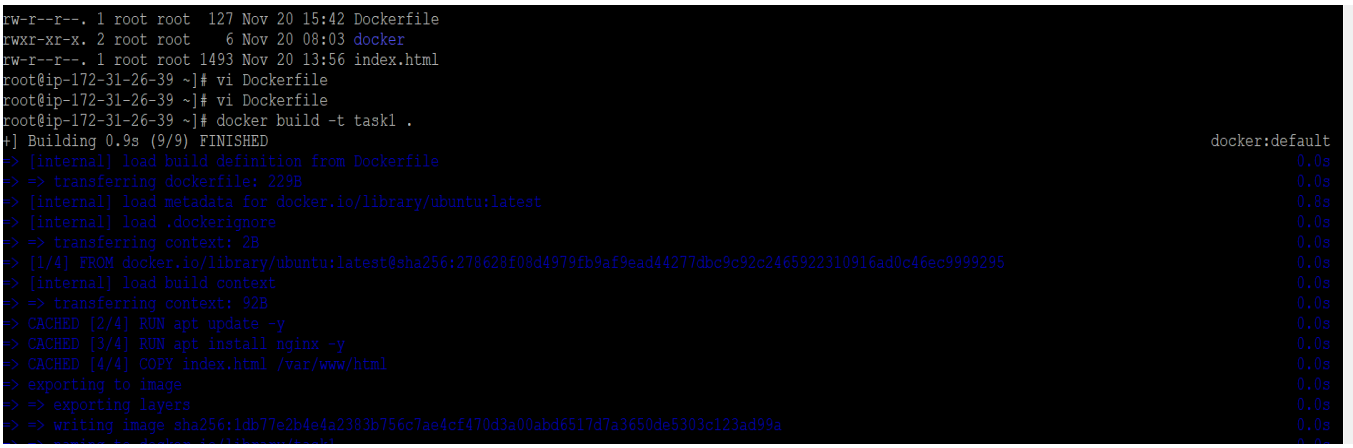
Now create the image {docker build –t image1 space (.)}

(Here is image1: image name you to create.)

Before creating the tag first, we have to create the repository in Docker Hub.

2.Tag the Image: To create the tag we must use command called {docker tag image1 username/repository name}.

(Here repository name: name of the repository in docker hub).



Here you will get the tag of the image in the form of username/repository name.

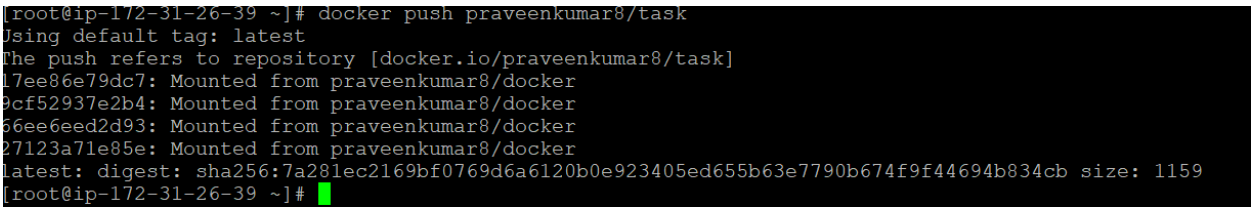
Before pushing you must log the docker hub {docker login}

Username: your docker username.

Password: your docker password

3.Push the Image to Docker Hub: By using the tag we are pushing the image to Docker hub.

Docker push username/repository name {docker push username/repository name}



4.Create the Node: To create Node we have the use the command called {docker swarm init}.

```
[root@ip-172-31-26-39 ~]# docker swarm init
Swarm initialized: current node (lxpuugagsdn8x1yxc9ltucjm3) is now a manager.

To add a worker to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-4bga20o2ynkuo7s9a4fifx2qn7ewfj33jwr05obxa2i179ku5e-bqesfiu4kqhxk4stt0rn6j3in 172.31.26.39:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.

[root@ip-172-31-26-39 ~]#
```

Now the Node is created.

```
[root@ip-172-31-26-39 ~]# docker node ls
ID                            HOSTNAME                                        STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
lxpuugagsdn8x1yxc9ltucjm3 *   ip-172-31-26-39.ap-southeast-1.compute.internal   Ready     Active         Leader           25.0.6
[root@ip-172-31-26-39 ~]#
```

5.Create the service: By using the Node the Service will be created.

To create the service {docker service create –-name (container name) --publish 8000:80 httpd}

```
[root@ip-172-31-26-39 ~]# docker service create  --name book-my-show --publish 8000:80 httpd
pp15216q4lh2yuvbn02lcx2os
overall progress: 1 out of 1 tasks
1/1: running   [==================================================>]
verify: Service converged
[root@ip-172-31-26-39 ~]# docker service ls
ID             NAME           MODE         REPLICAS   IMAGE          PORTS
pp15216q4lh2   book-my-show   replicated   1/1        httpd:latest   *:8000->80/tcp
[root@ip-172-31-26-39 ~]#
```

Along with the service Container also creates as shown below.

```
[root@ip-172-31-26-39 ~]# docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS                                     NAMES
c6cd9c70019f   httpd:latest   "httpd-foreground"       11 minutes ago   Up 11 minutes   80/tcp                                    book-my-show.1.9cglvf98oy81c6
t92cieu7pg9
43a1b53a49b0   nginx          "/docker-entrypoint.…"   9 hours ago      Up 8 hours      0.0.0.0:8001->80/tcp, :::8001->80/tcp     my_webapp2
78a3f09908ec   nginx          "/docker-entrypoint.…"   9 hours ago      Up 8 hours      0.0.0.0:8000->80/tcp, :::8000->80/tcp     my_webapp1
[root@ip-172-31-26-39 ~]#
```