

# 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.

```
root@ip-172-31-26-39:~  
FROM ubuntu  
RUN apt update -y  
RUN apt install apache2 -y  
COPY index.html /var/www/html  
CMD ["/usr/sbin/apachectl", "-D", "FOREGROUND"]  
~  
~  
~  
~  
~  
~
```

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).

```
Start a build
[root@ip-172-31-26-39 ~]# docker build -t lion .
[+] Building 19.7s (10/10) FINISHED
=> [internal] load build definition from Dockerfile
=> => transferring dockerfile: 235B
=> [internal] load metadata for docker.io/library/ubuntu:latest
=> [auth] library/ubuntu:pull token for registry-1.docker.io
=> [internal] load .dockerignore
=> => transferring context: 2B
=> [1/4] FROM docker.io/library/ubuntu:latest@sha256:278628f08d4979fb9af9ead44277dbc9c92c2465922310916ad0c46ec9999295
=> [internal] load build context
=> => transferring context: 92B
=> CACHED [2/4] RUN apt update -y
=> [3/4] RUN apt install apache2 -y
=> [4/4] COPY index.html /var/www/html
=> exporting to image
=> => exporting layers
=> => writing image sha256:797f67ea9603f11c665a37bd97d2f845d5d844b9470222af1f50e1ab30581877
=> => naming to docker.io/library/lion
[+] Building 19.7s (10/10) FINISHED
[root@ip-172-31-26-39 ~]#
```

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**}

```
Login Succeeded
[root@ip-172-31-26-39 ~]# docker push praveenkumar8/lion
Using default tag: latest
The push refers to repository [docker.io/praveenkumar8/lion]
f7f38b1b1085: Pushed
38e7e1ae94: Pushed
66ee6eed2d93: Mounted from praveenkumar8/task
27123a71e85e: Mounted from praveenkumar8/task
latest: digest: sha256:9957378a43e1673cd176ec970952780851f30ee8e18ee893cd3b66af27cdf83e size: 1160
[root@ip-172-31-26-39 ~]#
```

**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 (mktfvlr7tzl3rp5dbd226qq39) is now a manager.

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

    docker swarm join --token SWMTKN-1-2qo8opp0728keidsc35hh7m8n7f19uvoo9xniy9dmskjkae4xw-e9ufbcl9fd34l9nv384f14mni 172.31.26.39:2377

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

Now the Node is created.

```
[root@ip-172-31-26-39 ~]# docker node ls
ID                HOSTNAME                STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
mktfvlr7tzl3rp5dbd226qq39 * ip-172-31-26-39.ap-southeast-1.compute.internal    Ready      Active          Leader           25.0.6
```

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

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

```
[root@ip-172-31-26-39 ~]# docker service create --name Monkey --publish 80:80 lion
image lion:latest could not be accessed on a registry to record
its digest. Each node will access lion:latest independently,
possibly leading to different nodes running different
versions of the image.

5ppwhhbn9ucr33j4w3oiall3c
overall progress: 1 out of 1 tasks
1/1: running [=====>]
verify: Service converged
[root@ip-172-31-26-39 ~]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
32e9519b1cf5   lion:latest  "/usr/sbin/apachectl..."  12 seconds ago    Up 11 seconds    0.0.0.0:8001->80/tcp, :::8001->80/tcp    Monkey.1.qkjo0xrvhs58sew0ijiasrboi
43alb53a49b0   nginx      "/docker-entrypoint..."  22 hours ago     Up 22 hours      0.0.0.0:8001->80/tcp, :::8001->80/tcp    my_webapp2
```

Along with the service Container also creates as shown below.

```
verify: Service converged
[root@ip-172-31-26-39 ~]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
32e9519b1cf5   lion:latest  "/usr/sbin/apachectl..."  12 seconds ago    Up 11 seconds    0.0.0.0:8001->80/tcp, :::8001->80/tcp    Monkey.1.qkjo0xrvhs58sew0ijiasrboi
43alb53a49b0   nginx      "/docker-entrypoint..."  22 hours ago     Up 22 hours      0.0.0.0:8001->80/tcp, :::8001->80/tcp    my_webapp2
[root@ip-172-31-26-39 ~]# docker ps
CONTAINER ID   IMAGE     COMMAND                  CREATED    STATUS    PORTS                               NAMES
32e9519b1cf5   lion:latest  "/usr/sbin/apachectl..."  3 minutes ago    Up 3 minutes      0.0.0.0:8001->80/tcp, :::8001->80/tcp    Monkey.1.qkjo0xrvhs58sew0ijiasrboi
43alb53a49b0   nginx      "/docker-entrypoint..."  22 hours ago     Up 22 hours      0.0.0.0:8001->80/tcp, :::8001->80/tcp    my_webapp2
```

Final output is shown below.

←

→

🔄

⚠️ Not secure 18.141.198.243

☆

👤

⋮

Contact Form

First Name

Your name..

Last Name

Your last name..

Country

Australia

▼

Subject

Write something..

Submit

