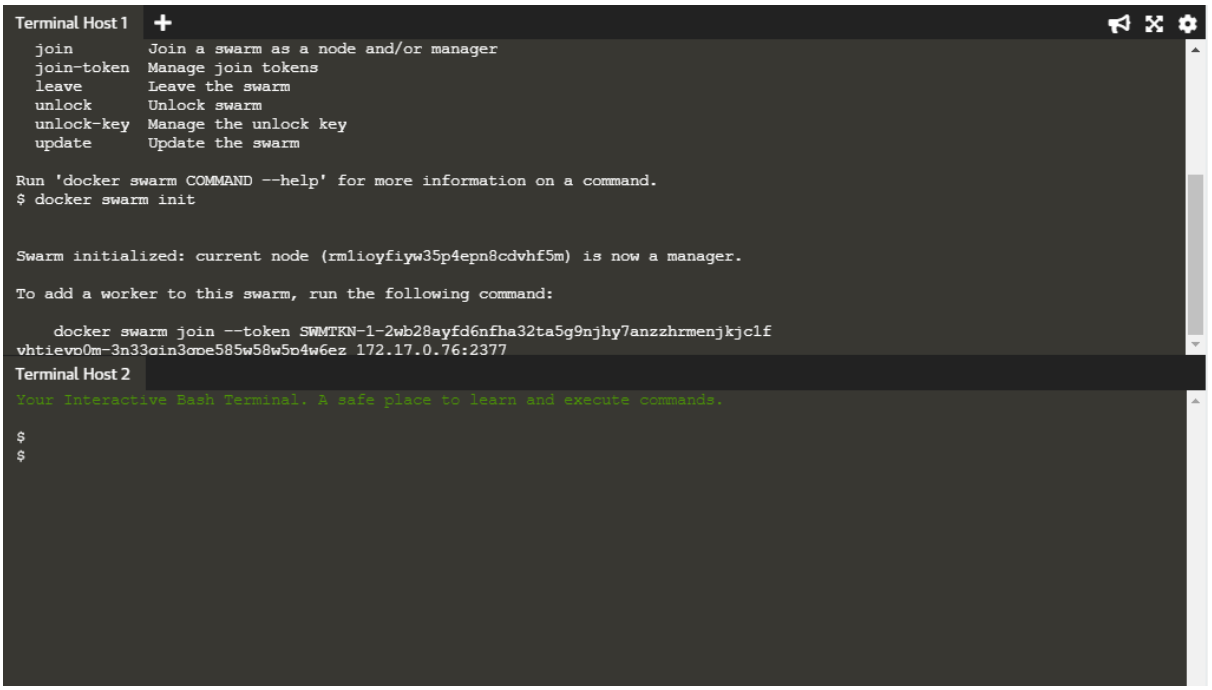
# Lab 8 | Swarm Mode

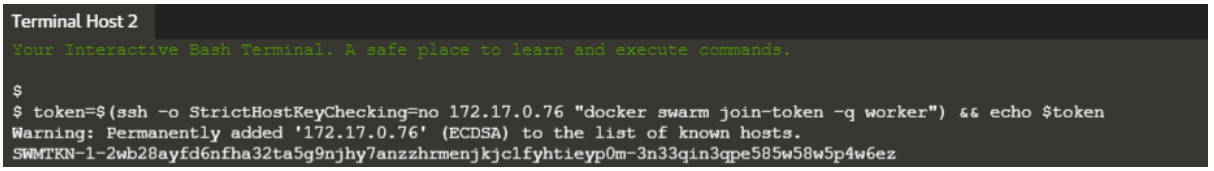
**Name: Neha Singh**
**Roll No: 67**
**Sap Id: 500069028**

A **Docker Swarm** is a group of either physical or virtual machines that are running the **Docker** application and that have been configured to join together in a cluster.

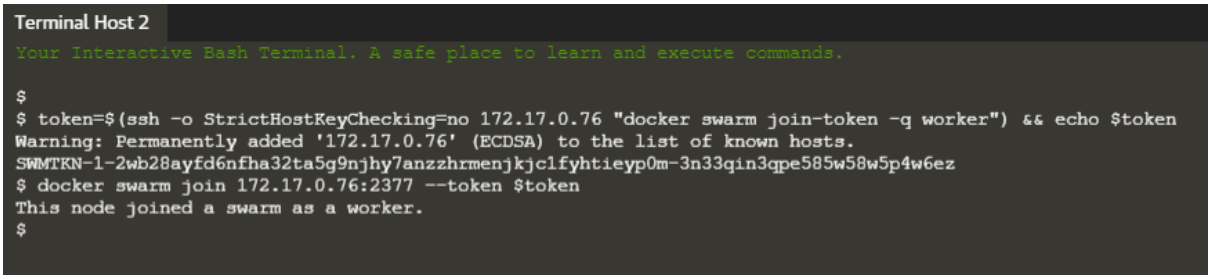
1. Initialize a docker swarm




2. To ass worker to cluster we will use token,




3. Second node join via requesting access to the manager



4. We will create a network, using which all nodes in the cluster can communicate with each other.

```
$ docker node ls
ID                            HOSTNAME            STATUS              AVAILABILI
TY        MANAGER STATUS      ENGINE VERSION
rm1ioyfiyw35p4epn8cdvhf5m *   host01              Ready               Active
          Leader              19.03.13
v8p00ms7m3wz8kq2rv1qjh6fi     host02              Ready               Active
                              19.03.13
$ docker network create -d overlay skynet
vsmpishtgfuediugzcd2892lu
```

5.
```
$ docker service create --name http --network skynet --replicas 2 -p 80:80 katac
oda/docker-http-server
l09ajmqzphrxar4eimn4n6bib
overall progress: 2 out of 2 tasks
1/2: running   [==================================================>]
2/2: running   [==================================================>]
```

6. Check the running clusters

```
$ docker service ls
ID                  NAME                MODE                REPLICAS
IMAGE                                   PORTS
l09ajmqzphrx        http                replicated          2/2
katacoda/docker-http-server:latest      *:80->80/tcp
```

7. we issue an HTTP request to the public port, it will be processed by the two containers

```
$ docker ps
CONTAINER ID        IMAGE                                   COMMAND             CRE
ATED                STATUS              PORTS               NAMES
259b50e1476d        katacoda/docker-http-server:latest      "/app"              4 m
inutes ago          Up 4 minutes        80/tcp              http.2.x3vc7kfzao537jvr
gfb6y18mj
$ curl host01
<h1>This request was processed by host: 2f277a020e25</h1>
```

8. Listing container in second terminal

```
$ docker ps
CONTAINER ID        IMAGE                                   COMMAND             CREATED             STATUS              PORTS
NAMES
2f277a020e25        katacoda/docker-http-server:latest      "/app"              4 minutes ago       Up 4 minutes        80/tcp
http.1.pn2nafal0275if4il9ljn631l
$
```

9. Viewing the list of all the tasks

```
$ docker service ps http
ID                  NAME                IMAGE                                   NOD
E                   DESIRED STATE       CURRENT STATE           ERROR
 PORTS
pn2nafal0275        http.1              katacoda/docker-http-server:latest      hos
t02                 Running             Running 6 minutes ago

x3vc7kfzao53        http.2              katacoda/docker-http-server:latest      hos
t01                 Running             Running 6 minutes ago
```

10. view the details and configuration of a service

```
$ docker service inspect --pretty http

ID:             l09ajmqzphrxar4eimn4n6bib
Name:           http
Service Mode:   Replicated
 Replicas:      2
Placement:
UpdateConfig:
 Parallelism:   1
 On failure:    pause
 Monitoring Period: 5s
 Max failure ratio: 0
 Update order:      stop-first
RollbackConfig:
```

11. On each node, you can ask what tasks it is currently running. Self refers to the manager node Leader:

```
$ docker node ps self
ID                  NAME                IMAGE                                  NOD
E                   DESIRED STATE       CURRENT STATE           ERROR
 PORTS
x3vc7kfzao53        http.2              katacoda/docker-http-server:latest   hos
t01                 Running             Running 8 minutes ago
```

12. Using the ID of a node you can query individual hosts

```
$ docker node ps $(docker node ls -q | head -n1)
ID                  NAME                IMAGE                                  NOD
E                   DESIRED STATE       CURRENT STATE           ERROR
 PORTS
x3vc7kfzao53        http.2              katacoda/docker-http-server:latest   hos
t01                 Running             Running 9 minutes ago
```

13. you will see additional nodes being started

```
$ docker ps
CONTAINER ID        IMAGE                                COMMAND             CRE
ATED             STATUS              PORTS               NAMES
259b50e1476d        katacoda/docker-http-server:latest   "/app"              10
minutes ago      Up 10 minutes       80/tcp              http.2.x3vc7kfzao537jvr
gfb6y18mj
```