

Experiment – 8

Aim: Understanding docker swarm.

1) Making the manager

```
Terminal Host 1 +
Your Interactive Bash Terminal.
A good starting point is executing `docker`

$ docker swarm init
Swarm initialized: current node (l42djl6dpro8efjn73slco4ci) is now a manager.

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

    docker swarm join --token SWMTKN-1-lytp1ue0mrl3sal0vdu80utwo40dbt7aox5idu6yqymelu46k6-b5zgfhqwzd6f25m3bd0h8utbe 172.17.0.28:2377

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

$
```

2) Adding a node to the manager

```
Terminal Host 1 +
Your Interactive Bash Terminal.
A good starting point is executing `docker`

$ docker swarm init
Swarm initialized: current node (l42djl6dpro8efjn73slco4ci) is now a manager.

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

    docker swarm join --token SWMTKN-1-lytp1ue0mrl3sal0vdu80utwo40dbt7aox5idu6yqymelu46k6-b5zgfhqwzd6f25m3bd0h8utbe 172.17.0.28:2377

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

$

Terminal Host 2
Your Interactive Bash Terminal.
A good starting point is executing `docker`

$ docker swarm join --token SWMTKN-1-lytp1ue0mrl3sal0vdu80utwo40dbt7aox5idu6yqymelu46k6-b5zgfhqwzd6f25m3bd0h8utbe 172.17.0.28:2377
This node joined a swarm as a worker.
$
```

3) Running docker node ls on the manager

Terminal Host 1 +

```
$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY
l42dj16dpro8efjn73slco4ci *	host01	Ready	Active
ctjudv37mpwfpabzyrizxs3a4	host02	Ready	Active

```
$
```

4) Running docker node ls on terminal 3 which is not connected to the manager

Terminal Host 1 Terminal 3 +

Extra Interactive Bash Terminal

```
$
```

```
$ docker node ls
```

ID	HOSTNAME	STATUS	AVAILABILITY	MANAGER STATUS	ENGINE VERSION
l42dj16dpro8efjn73slco4ci *	host01	Ready	Active	Leader	19.03.13
ctjudv37mpwfpabzyrizxs3a4	host02	Ready	Active		19.03.13

```
$
```