NAME: - Abhishek Singh

Roll No. : - R171218120

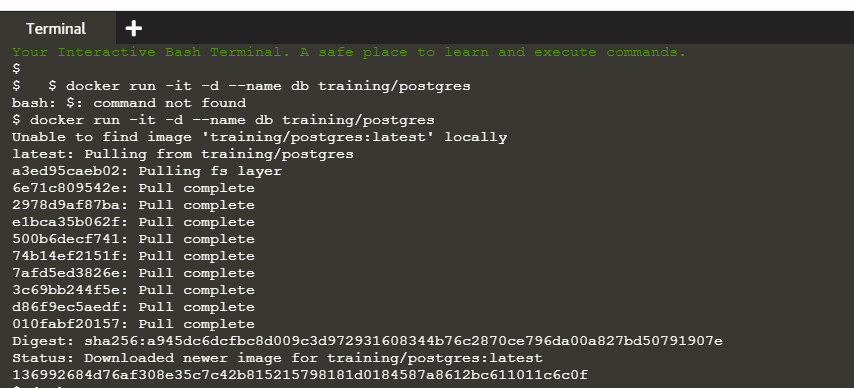
Sap Id: - 500067726

**EXPERIMENT-5**

## Docker linking

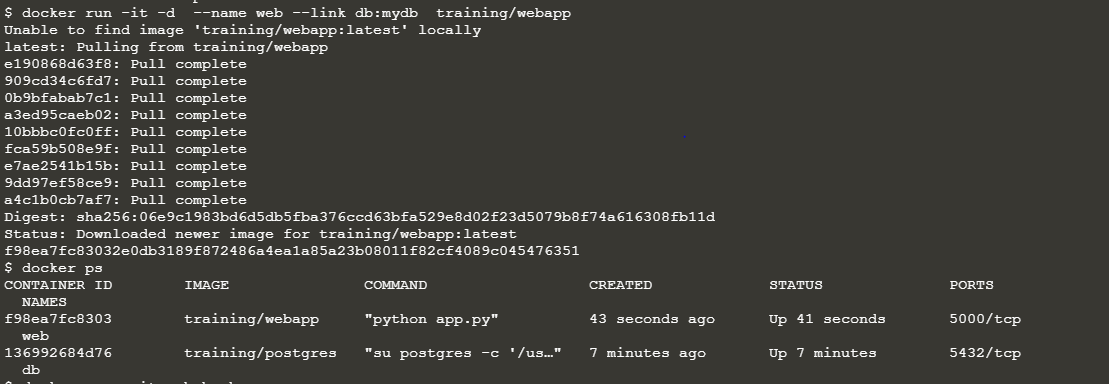
* Run a container in detached mode with name "db" from image "training/postgres"

$ docker run -it -d --name db training/postgres



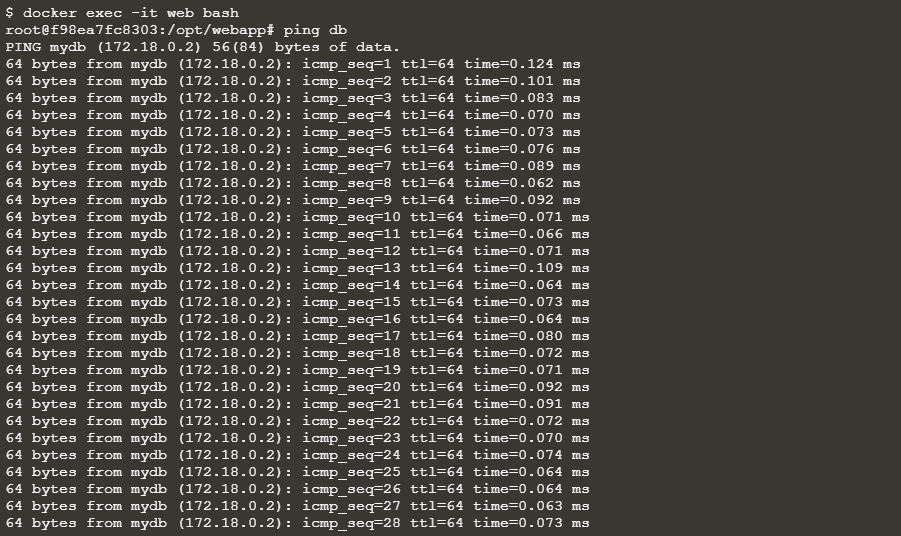
* Run another container in detached mode with name "web" from image "training/webapp", link container "db" with alias "mydb" to this container and finally pass an inline command "python app.py" while running container.

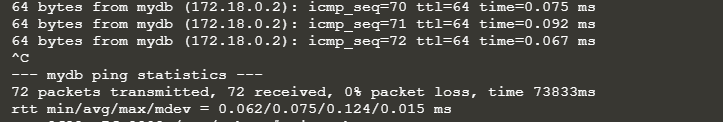
$ docker run -it -d -–name web -–link db:mydb training/webapp   
python app.py



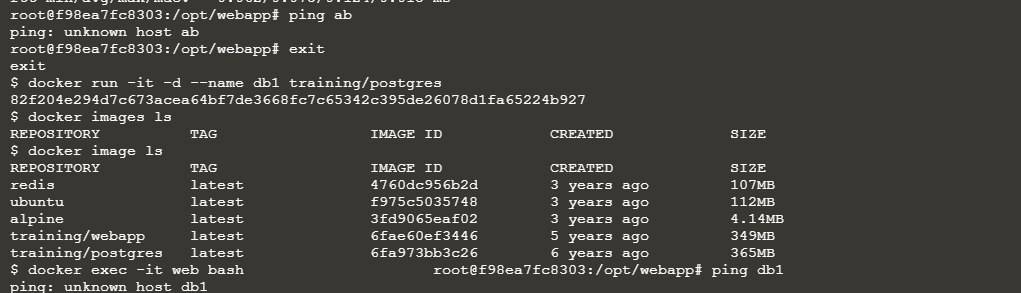
* Take a bash terminal in "web" container and Test container linking by doing a ping to "mydb"

$ docker exec -it web bash





And then run  
# ping mydb

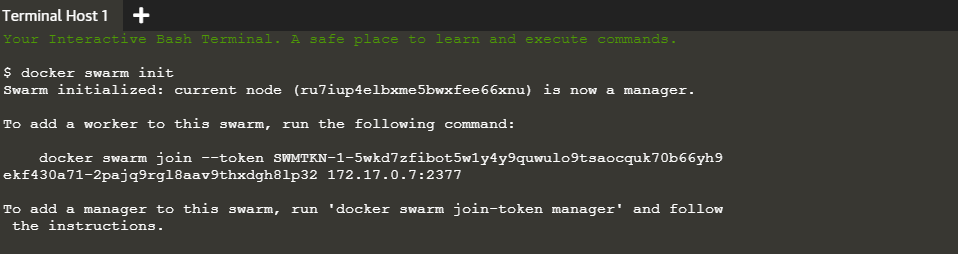


## 

## Create Swarm

* Run the following command to create a new swarm:

$ docker swarm init --advertise-addr <MANAGER-IP(ip of the vagrant machine which will act as manager)>



The following command creates a swarm on the manager machine: for example

$ docker swarm init --advertise-addr 192.168.99.100

OUTPUT:

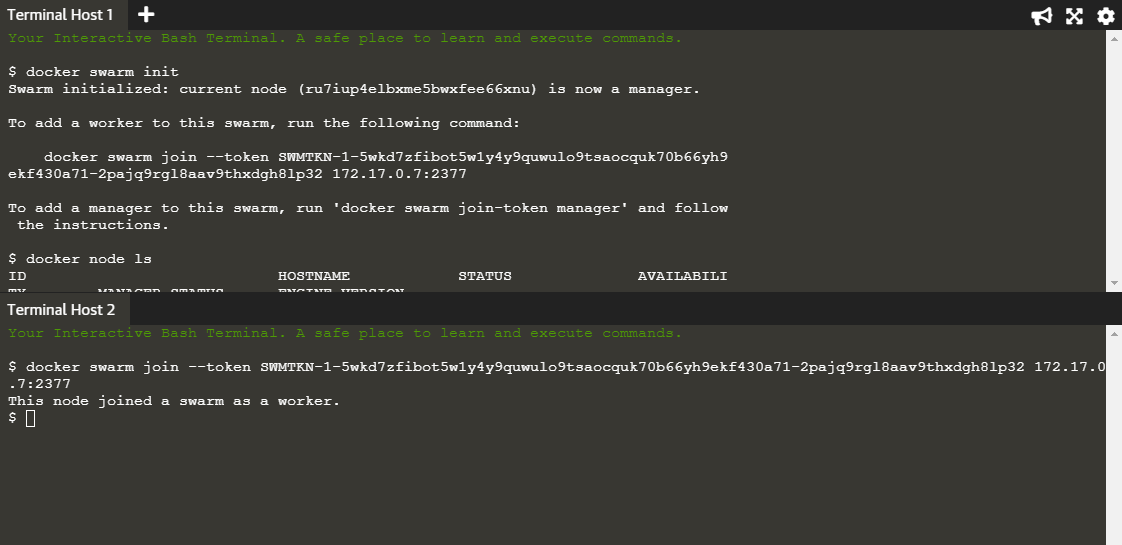
Swarm initialized: current node (dxn1zf6l61qsb1josjja83ngz) is now a manager.

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

$ docker swarm join \

--token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c \

192.168.99.100:2377



* Run docker info to view the current state of the swarm:

$ docker info

Containers: 2

Running: 0

Paused: 0

Stopped: 2

...snip...

Swarm: active

NodeID: dxn1zf6l61qsb1josjja83ngz

Is Manager: true

Managers: 1

Nodes: 1

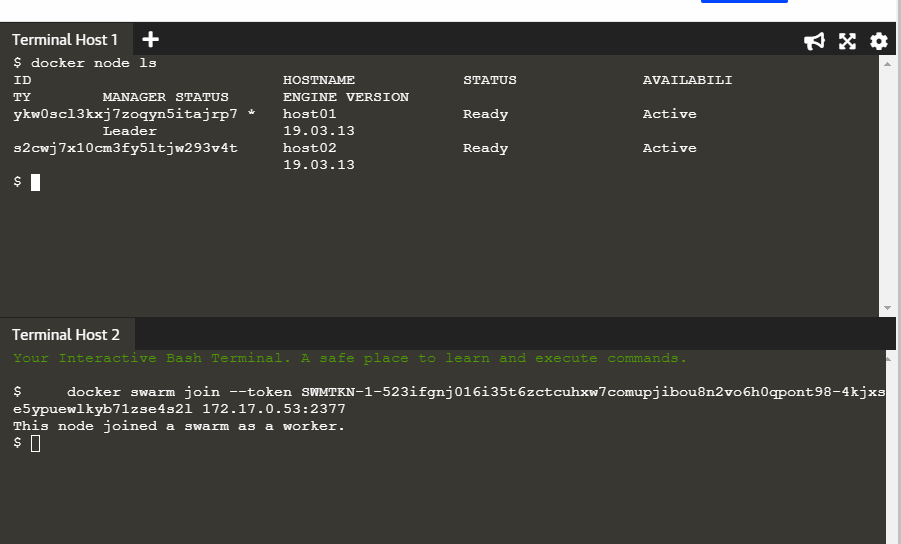
...snip...

* Run the docker node ls command to view information about nodes:

$ docker node ls

ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS

dxn1zf6l61qsb1josjja83ngz \* manager1 Ready Active Leader



* Add nodes to the cluster

Run the command produced by the docker swarm init output from the Create a swarm tutorial step to create a worker node joined to the existing swarm:

$ docker swarm join --token SWMTKN-1-49nj1cmql0jkz5s954yi3oex3nedyz0fb0xx14ie39trti4wxv-8vxv8rssmk743ojnwacrr2e7c 192.168.99.100:2377

This node joined a swarm as a worker.

NOTE: Repeat above step in all your nodes which has to be part of swarm cluster

* Now from the manager node run docker node ls command to check the status of the joined nodes

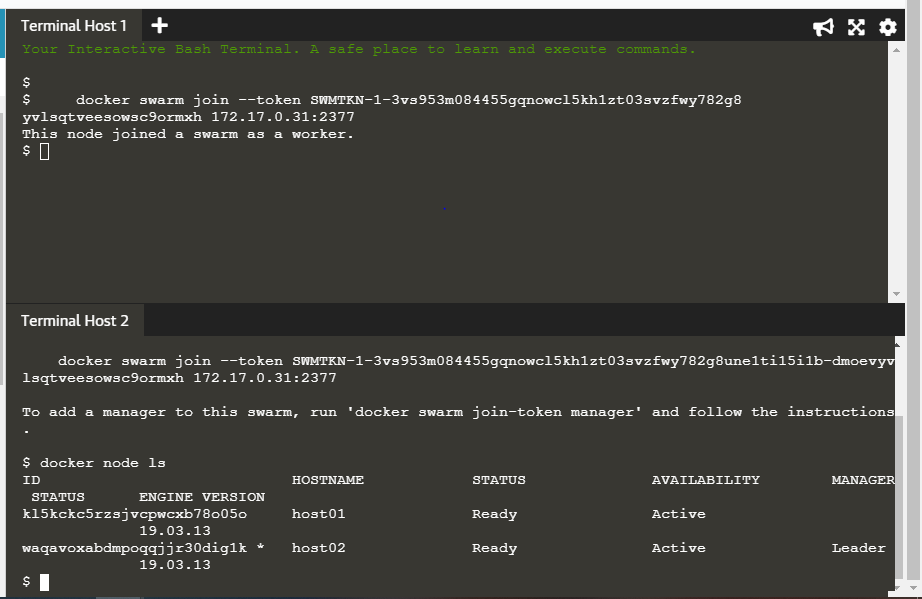
$ docker node ls

ID HOSTNAME STATUS AVAILABILITY MANAGER STATUS

03g1y59jwfg7cf99w4lt0f662 worker2 Ready Active

9j68exjopxe7wfl6yuxml7a7j worker1 Ready Active

dxn1zf6l61qsb1josjja83ngz \* manager1 Ready Active Leader



The MANAGER column identifies the manager nodes in the swarm. The empty status in this column for worker1 and worker2 identifies them as worker nodes.

Swarm management commands like docker node ls only work on manager nodes.