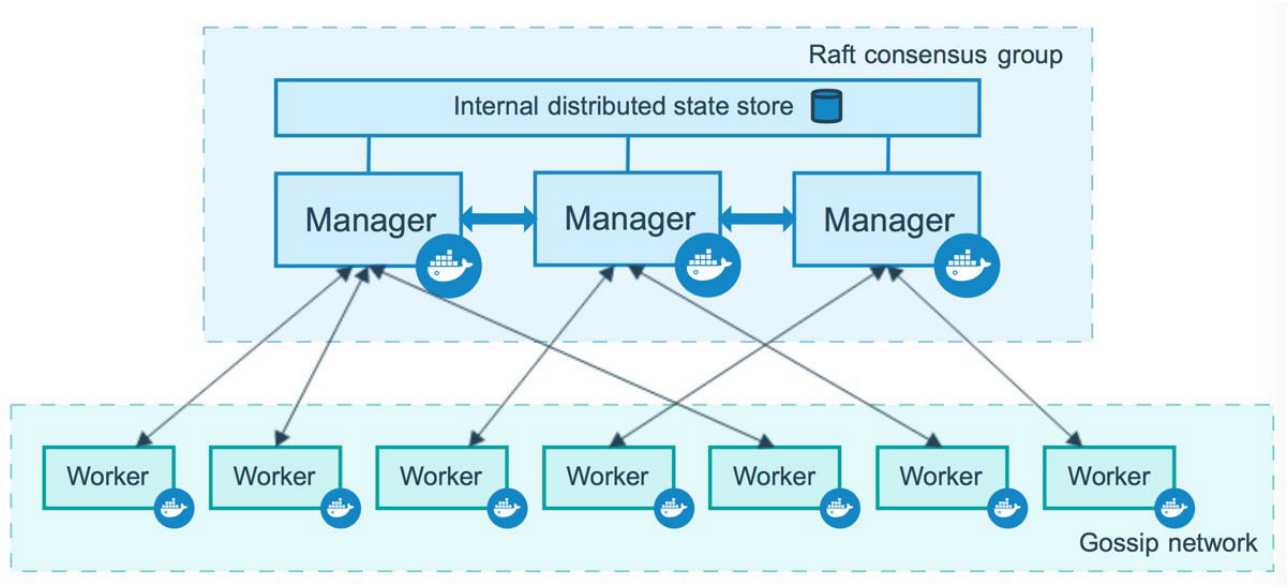
Lab Technician: Om Luitel



Docker Swarm Lab Setup:

Prerequisites:

1. Ensure that Virtualbox is installed on your machine.
2. You can customize the Vagrant file according to the number of manager and worker nodes, as well as their memory and CPU allocation, based on your host machine's resources. As an example, I have 32 GB of RAM on my host machine, so I've configured 2 manager nodes and 5 worker nodes for this demo lab, with each machine allocated 2 GB of RAM.

Getting Started:

To set up the Docker Swarm lab, follow these steps:

Step 1: Clone the following Git repository to your machine:

Git Repository Link: https://github.com/Omluitel/DevOps.git

Step 2: Navigate to the following directory in the cloned repository:

```
bash

DevOps/Docker_swarm/lab_setup_with_vagrant
```

Step 3: Now it's time to bring up the environment using Vagrant. Run the following command:

```
vagrant up
```

Step 4: Wait for some time as the environment sets up. Once the environment is set up, you can check the status of the running machines on your host by executing the following command:

```
vagrant status
```

```
om@om-linu:~/projects/DevOps/Docker_Swarm_project$ vagrant status
Current machine states:

manager01                 running (virtualbox)
manager02                 running (virtualbox)
worker01                  running (virtualbox)
worker02                  running (virtualbox)
worker03                  running (virtualbox)
worker04                  running (virtualbox)
worker05                  running (virtualbox)

This environment represents multiple VMs. The VMs are all listed
above with their current state. For more information about a specific
VM, run `vagrant status NAME`.
```

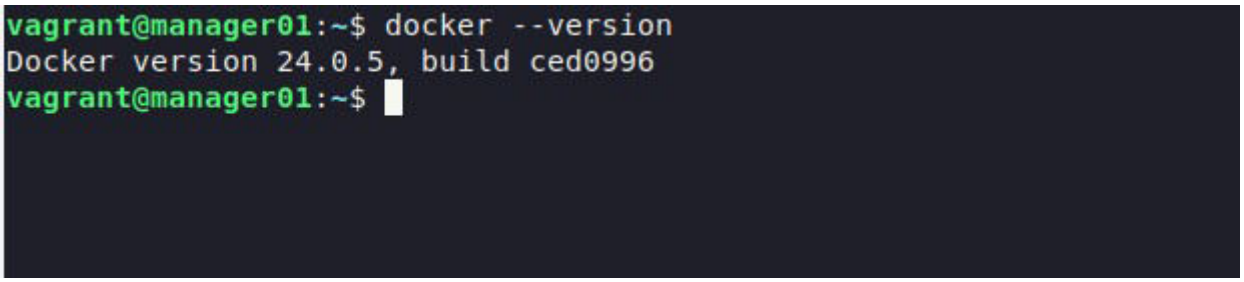
Now, SSH into one of the manager nodes with the following command:

```
vagrant ssh manager01
```

Once you are SSHed into manager01 and run the following command:

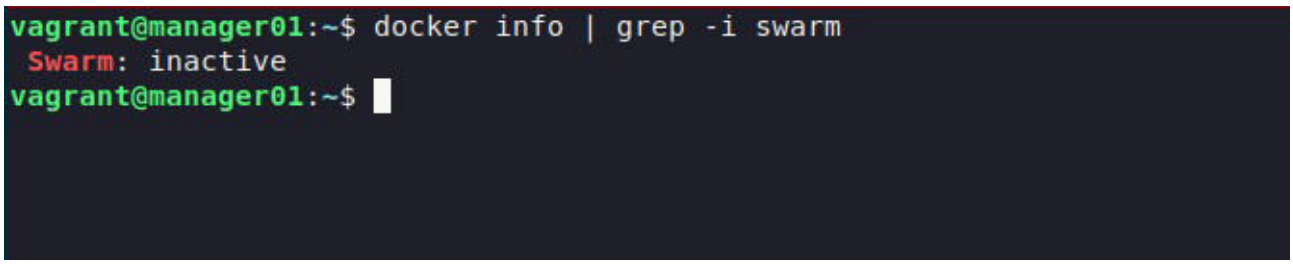
```
docker –version
```

```
You should be able to confirm that Docker is installed on the machine as we have defined it in the Vagrantfile.
```



Step 4: Enable Docker Swarm on the Machine

By default, when you install Docker on the machine, Docker Swarm is disabled. You need to enable it to get it working. Run the following command on the machine to check the status of Docker Swarm:



To enable Docker Swarm, run the following command: `docker swarm init`.

The error below is as expected because we need to advertise one of the available addresses from our Vagrant machine. Run `ip add` to view all the adapters and select one of those IPs to advertise.

```
vagrant@manager01:~$ docker swarm init
Error response from daemon: could not choose an IP address to advertise since this system has multiple a
ddresses on different interfaces (10.0.2.15 on enp0s3 and 192.168.1.161 on enp0s8) - specify one with --
advertise-addr
vagrant@manager01:~$
```

I am advertising 192.168.1.161, you should advertise according to your machine IP.

```
vagrant@manager01:~$ ip add
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN grou
p default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel sta
te UP group default qlen 1000
    link/ether 02:d2:d6:47:3d:c7 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 metric 100 brd 10.0.2.255 scope global dynamic enp
0s3
       valid_lft 85942sec preferred_lft 85942sec
    inet6 fe80::d2:d6ff:fe47:3dc7/64 scope link
       valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel sta
te UP group default qlen 1000
    link/ether 08:00:27:ee:c2:71 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.161/24 metric 100 brd 192.168.1.255 scope global dyna
mic enp0s8
       valid_lft 85943sec preferred_lft 85943sec
    inet6 fe80::a00:27ff:feee:c271/64 scope link
       valid_lft forever preferred_lft forever
```

Now, I have successfully enabled Docker Swarm on this machine and am ready to add other managers and workers to this environment. Note below token is for adding worker.

```
vagrant@manager01:~$ docker swarm init --advertise-addr 192.168.1.161
Swarm initialized: current node (a0lrlxe5xanps7r25j9jvs4sq) is now a manager.

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

    docker swarm join --token SWMTKN-1-3ufunfi03k9q4dbpv1dt249vnrxifj3jclnutmlt3b2vq6sxa8-8vwvrsvv2e0gxq1lzhcncv7
9x 192.168.1.161:2377

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

vagrant@manager01:~$
```

If you want to add manager node you can generate toke with following command.

```
vagrant@manager01:~$ docker swarm join-token manager
To add a manager to this swarm, run the following command:

    docker swarm join --token SWMTKN-1-3ufunfi03k9q4dbpv1dt249vnrxifj3jclnutmlt3b2vq6sxa8-a4bk2uphpmqwtzj902w32d7
mw 192.168.1.161:2377
```

I followed this process, and now I have successfully added 2 managers and 5 workers to the environment. I can confirm this as below:

```
vagrant@manager01:~$ docker node ls
ID                            HOSTNAME    STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
h3wzjw6lx4opteese8key8rjn *   manager01   Ready     Active         Leader           24.0.5
77goe939j488b8kz8vnc4fc2n     manager02   Ready     Active         Reachable        24.0.5
xrwhje4luu6iic33cjpidtx20     worker01    Ready     Active                          24.0.5
qesqi91c65kgf0kairksg76r0     worker02    Ready     Active                          24.0.5
shhgtwdosr1yn4hsu5ko9llfl     worker03    Ready     Active                          24.0.5
un5xhc08660ityf41xmmogop9     worker04    Ready     Active                          24.0.5
mjrhjtauhrjul0lk02ytsd9ju     worker05    Ready     Active                          24.0.5
vagrant@manager01:~$
```