Multi-Node Docker Swarm Cluster for High Availability

Step 1: Initialize Docker Swarm, Check the cluster

Step 2: Create a Docker Compose File

```
master@master-vn:-$ docker swarm init --advertise-addr 192.168.219.137
Swarm initialized: current node (m5n7ln8k1lmism69uo4mqvgzv) is now a manager.

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

docker swarm join --token SWMTKN-1-1xz5sjyweo4dinhrhwjfjulgxef83lx1277uyaa3ky3ejrnw3k-a9px9h85pid31d5957g7pmtj7 192.168.219.137:

2377

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

master@master-vn:-$ docker node ls

M5n7ln8k1lmism69uo4mqvgzv * master-vn Ready Active Leader 28.0.1
4zvd78y6a4rh2ne7n2te13yhd node1-vn Ready Active Leader 26.1.3
bpia2Fuitplikr5dkvu9au2y7 node2-vn Ready Active 26.1.3

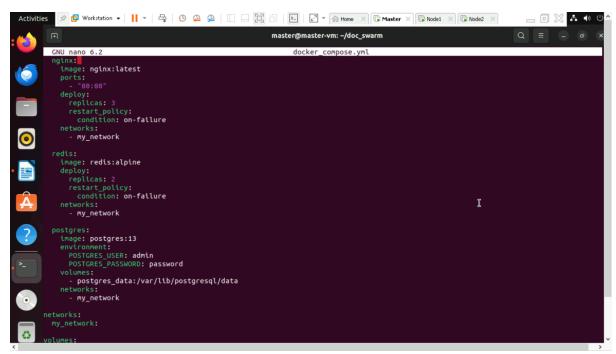
master@master-vn:-$ ls

composetest Documents Music Public Videos
Desktop Downloads nginx_project snsp

doc_swarm Flask-Docker Pictures Tenplates

master@master-vn:-$ mkdir doc_swarm
mkdir: cannot create directory 'doc_swarm'
master@master-vn:-$ mkdir doc_swarm
master@master-vn:-$ mkdir doc_swarm
master@master-vn:-$ ne doc_swarm
master@master-vn:-$ ne doc_swarm
master@master-vn:-$ ne doc_swarm
master@master-vn:-$ nano docker_compose.yml
```

Docker_compose.yml file



Step 3: Deploy the Application on Swarm

```
master@naster-vm:-/doc.swarr$ docker stack deploy -c docker_compose.yml myapp

Since --detach=false was not specified, tasks will be created in the background.

In a future release, --detach=false will become the default.

Creating service myapp_my_network

Creating service myapp_redis

Creating service myapp_postgres

Creating service myapp_postgres
```

Check service status

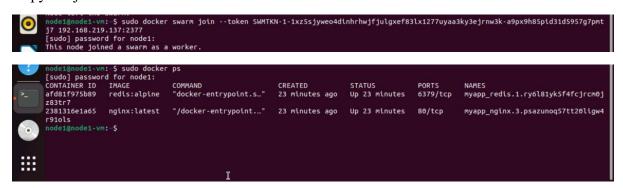
```
master@master-vm:-/doc_swarm$ docker service ls

ID NAME MODE REPLICAS IMAGE PORTS
7gypkw9nnerz myapp_nginx replicated 3/3 nginx:latest *:80->80/tcp

*** wSois8vl2sv6 myapp_postgres replicated 1/1 postgres:13
ty44n2tznstx myapp_redis replicated 2/2 redis:alpine

master@master-vm:-/doc_swarm$
```

Copy the join token and add worker node-1



Copy the join token and add worker node-2

```
node2@node2-vm:-$ sudo docker swarm join --token SWMTKN-1-1xz5sjyweo4dinhrhwjfjulgxef83lx1277uyaa3ky3ejrnw3k-a9px9h85pid31d5957g7pmt j7 192.168.219.137:2377 [sudo] password for node2: This node joined a swarm as a worker.
```

Pulled the postgres

