

ASSIGNMENT – 3

MULTI NODE DOCKER SWARM CLUSTER:

1. Initialize the docker swarm

```
master@master-vm:~$ docker swarm init --advertise-addr 192.168.219.137
Swarm initialized: current node (ngglfig5fgkvvj93rvy2zium) is now a manager.

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

    docker swarm join --token SWMTKN-1-20h8bsqnpuxwnwbf179hk5e83n2aonrt2qd2hdtwqe3bls32vc-0lablr95bwjwldlq4m71omgnx 192.168.219.137:2377

To add a manager to this swarm, run 'docker swarm join-token manager' and follow the instructions.
master@master-vm:~$ docker node ls
```

2. Run the token in the nodes

Node1:

```
node1@node1-vm:~/Desktop$ sudo docker swarm join --token SWMTKN-1-3sz2rz5bwwgkqzgv8t3xoohfg2tc2xrzfjg4q2n1sgmlvvzeul-9zn1z63npngf012erja5fxffg 192.168.219.137:2377
[sudo] password for node1:
This node joined a swarm as a worker.
node1@node1-vm:~/Desktop$ sudo docker pull nginx
```

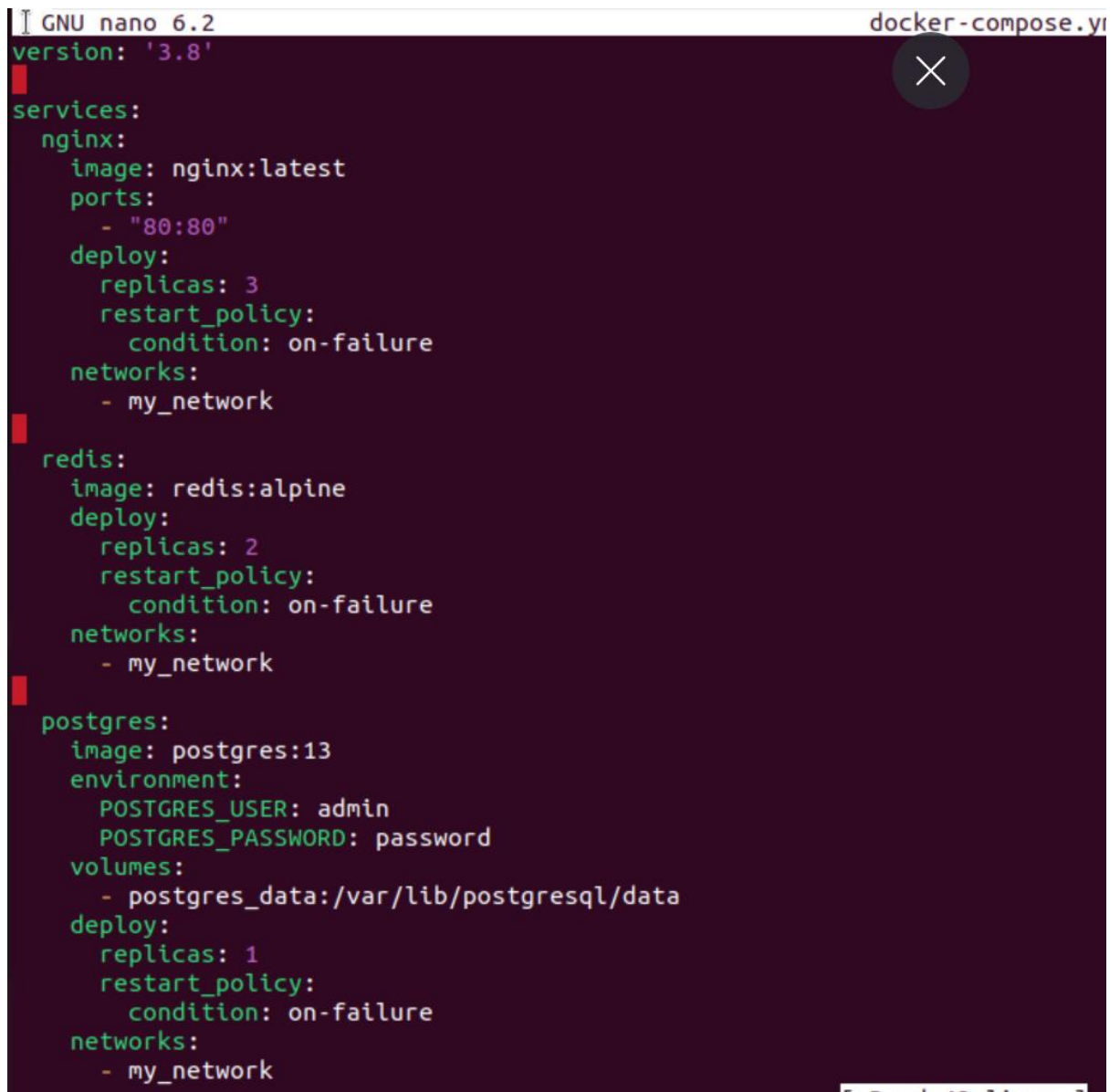
Node2:

```
node2@node2-vm:~/Desktop$ sudo docker swarm join --token SWMTKN-1-3sz2rz5bwwgkqzgv8t3xoohfg2tc2xrzfjg4q2n1sgmlvvzeul-9zn1z63npngf012erja5fxffg 192.168.219.137:2377
This node joined a swarm as a worker.
node2@node2-vm:~/Desktop$ sudo chmod 666 /var/run/docker.sock
```

3. Verify the nodes that are part of swarm

```
master@master-vm:~/doc_swarm$ nano docker-compose.yml
master@master-vm:~/doc_swarm$ docker node ls
ID                HOSTNAME        STATUS    AVAILABILITY    MANAGER STATUS    ENGINE VERSION
v7h2rzxg5333jpoozb03l9dam * master-vm      Ready     Active           Leader             26.1.3
nh1asoac7tgc2ewv0h5y7gssx node1-vm       Ready     Active           -                  26.1.3
cbt4otd4vh90fgb756kly0uzy node2-vm       Ready     Active           -                  26.1.3
```

4. Create a directory and create docker-compose.yml file

A screenshot of a terminal window showing a nano editor editing a file named 'docker-compose.yml'. The editor's title bar shows 'GNU nano 6.2' and the file name. The content of the file is a Docker Compose configuration for three services: nginx, redis, and postgres, all connected to a 'my_network'. The nginx service uses the 'nginx:latest' image, has port 80 mapped to 80, and is deployed with 3 replicas. The redis service uses the 'redis:alpine' image and is deployed with 2 replicas. The postgres service uses the 'postgres:13' image, has environment variables for user and password, and a volume for data storage. It is deployed with 1 replica. All services have a restart policy of 'on-failure'.

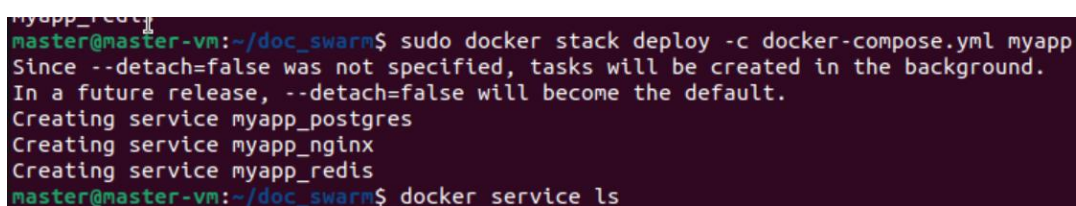
```
GNU nano 6.2 docker-compose.yml
version: '3.8'

services:
  nginx:
    image: nginx:latest
    ports:
      - "80:80"
    deploy:
      replicas: 3
      restart_policy:
        condition: on-failure
    networks:
      - my_network

  redis:
    image: redis:alpine
    deploy:
      replicas: 2
      restart_policy:
        condition: on-failure
    networks:
      - my_network

  postgres:
    image: postgres:13
    environment:
      POSTGRES_USER: admin
      POSTGRES_PASSWORD: password
    volumes:
      - postgres_data:/var/lib/postgresql/data
    deploy:
      replicas: 1
      restart_policy:
        condition: on-failure
    networks:
      - my_network
```

5. Run the compose file

A screenshot of a terminal window showing the command 'sudo docker stack deploy -c docker-compose.yml myapp' being executed. The output shows that the command was successful and that three services (myapp_postgres, myapp_nginx, and myapp_redis) were created. The prompt then changes to 'master@master-vm:~/doc_swarm\$' and the command 'docker service ls' is entered.

```
master@master-vm:~/doc_swarm$ sudo 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_postgres
Creating service myapp_nginx
Creating service myapp_redis
master@master-vm:~/doc_swarm$ docker service ls
```

PRATHEEK U B
289226

6. See the list of services running in multiple nodes

```
master@master-vm:~/doc_swarm$ docker images
REPOSITORY    TAG       IMAGE ID       CREATED        SIZE
postgres      13        6c774c1ad2b9   11 days ago    423MB
nginx         latest    b52e0b094bc0   4 weeks ago    192MB
redis         alpine    8f5c54441eb9   2 months ago   41.4MB
master@master-vm:~/doc_swarm$ docker service ls
ID                NAME                MODE                REPLICAS        IMAGE                PORTS
y9rs9l5x1k7c     myapp_nginx         replicated          3/3             nginx:latest        *:80->80/tcp
z6jaxhg21cop     myapp_postgres      replicated          1/1             postgres:13
5vfrxm66kesc     myapp_redis         replicated          2/2             redis:alpine
```

MASTER NODE:

```
master@master-vm:~/doc_swarm$ docker ps
CONTAINER ID   IMAGE                COMMAND             CREATED        STATUS        PORTS                NAMES
7a31aca111e8   nginx:latest        "/docker-entrypoint..." 19 minutes ago Up 19 minutes   80/tcp              myapp_nginx.3.wl5w0zfpunjnk78j87L
se25y3
7fcab59f4e2a   postgres:13        "docker-entrypoint.s..." 20 minutes ago Up 20 minutes   5432/tcp            myapp_postgres.1.md321qwzqunpfd3a
j6esymgct
master@master-vm:~/doc_swarm$
```

NODE 1:

```
node1@node1-vm:~/Desktop$ docker ps
CONTAINER ID   IMAGE                COMMAND             CREATED        STATUS        PORTS                NAMES
4abab2dd08f8   redis:alpine        "docker-entrypoint.s..." 20 minutes ago Up 20 minutes   6379/tcp            myapp_redis.2.j26asjwgk9tts8e9966
mac7d6
b71878cf2353   nginx:latest        "/docker-entrypoint..." 21 minutes ago Up 20 minutes   80/tcp              myapp_nginx.1.8g86iekoab3yrk3qwtb
euqwqk
node1@node1-vm:~/Desktop$
```

NODE 2:

```
node2@node2-vm:~/Desktop$ docker ps
CONTAINER ID   IMAGE                COMMAND             CREATED        STATUS        PORTS                NAMES
ab8cbdf77c1    redis:alpine        "docker-entrypoint.s..." About a minute ago Up About a minute   6379/tcp            myapp_redis.1.o8yx0xm6tq7
8j5i5pe8i6nmna
d063b6b97024   nginx:latest        "/docker-entrypoint..." About a minute ago Up About a minute   80/tcp              myapp_nginx.2.l55a5bp0450
08kkeve9kbs927
node2@node2-vm:~/Desktop$
```