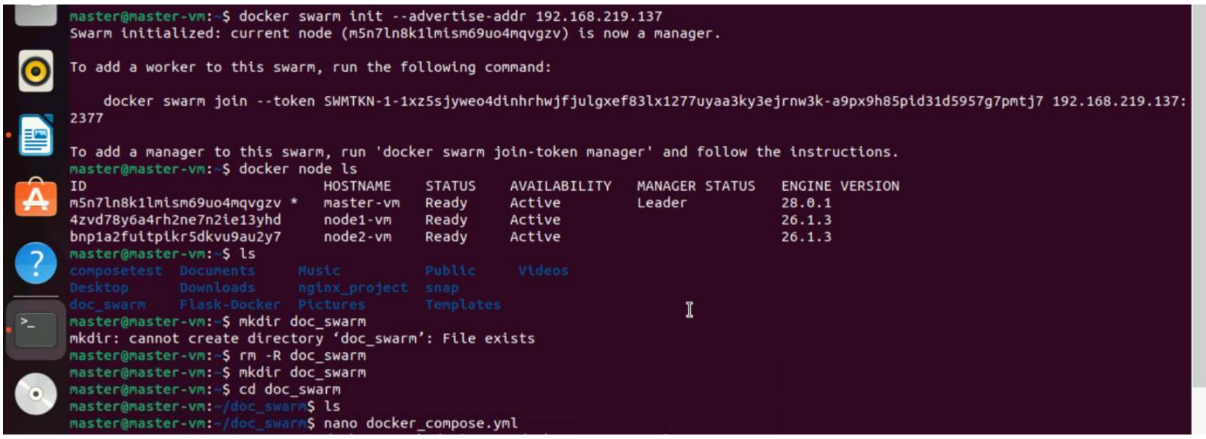
Multi-Node Docker Swarm Cluster for High Availability

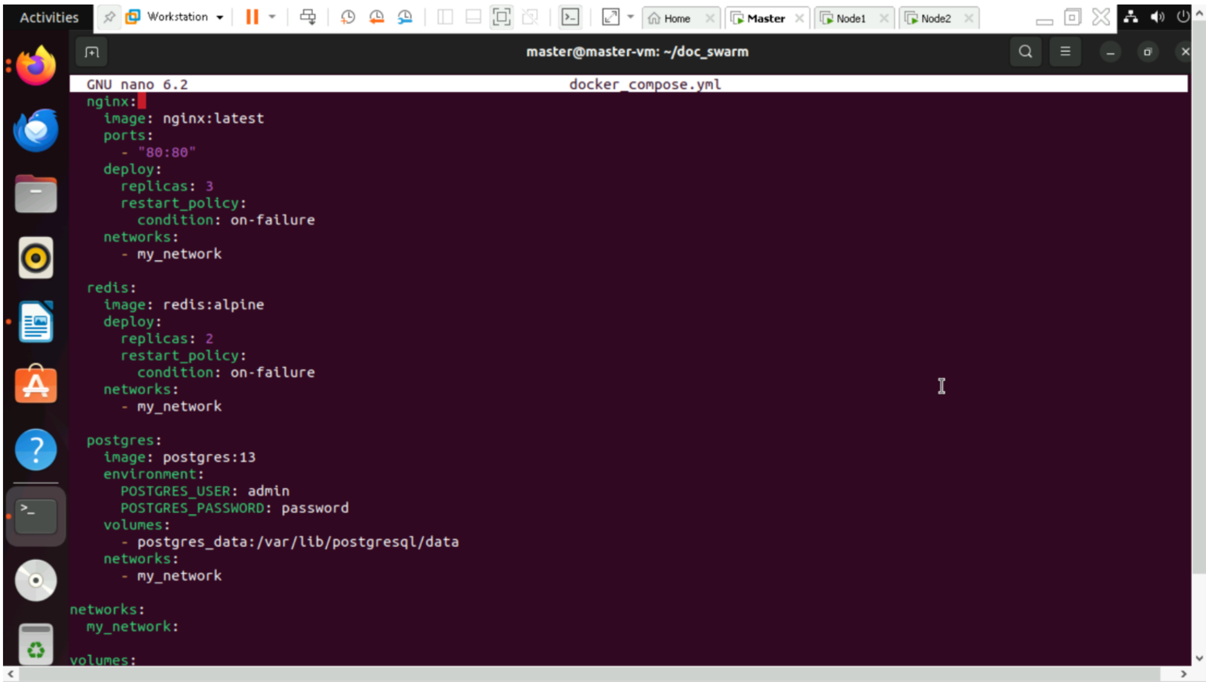
Step 1: Initialize Docker Swarm, Check the cluster

Step 2: Create a Docker Compose File



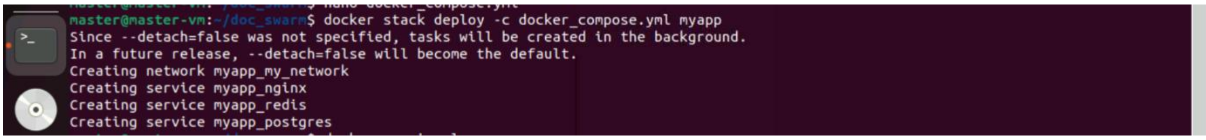
```
master@master-vm:~$ docker swarm init --advertise-addr 192.168.219.137
Swarm initialized: current node (m5n7ln8k1lmism69uo4mqvgzv) is now a manager.

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-vm:~$ docker node ls
ID                            HOSTNAME    STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
m5n7ln8k1lmism69uo4mqvgzv *   master-vm   Ready     Active         Leader           28.0.1
4zvd78y6a4rh2ne7n2ie13yhd     node1-vm    Ready     Active                          26.1.3
bnp1a2fuitpikr5dkvu9au2y7     node2-vm    Ready     Active                          26.1.3
master@master-vm:~$ ls
composetest  Documents    Music          Public     Videos
Desktop      Downloads    nginx_project  snap
doc_swarm    Flask-Docker Pictures       Templates
master@master-vm:~$ mkdir doc_swarm
mkdir: cannot create directory ‘doc_swarm’: File exists
master@master-vm:~$ rm -R doc_swarm
master@master-vm:~$ mkdir doc_swarm
master@master-vm:~$ cd doc_swarm
master@master-vm:~/doc_swarm$ ls
master@master-vm:~/doc_swarm$ nano docker_compose.yml
```

Docker_compose.yml file



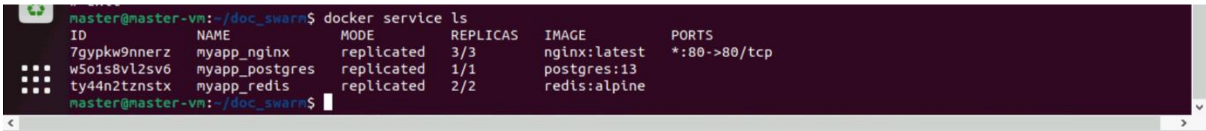
```
  GNU nano 6.2                                   docker_compose.yml
  nginx:
    image: nginx:latest
    ports:
      - "80:80"
    deploy:
      replicas: 3
      restart_policy:
        condition: on-failure
    networks:
      - my_network

  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
    networks:
      - my_network

networks:
  my_network:

volumes:
```

Step 3: Deploy the Application on Swarm



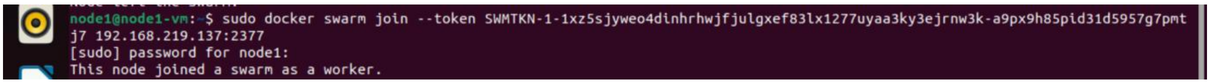
```
master@master-vm:~/doc_swarm$ docker stack deploy -c docker_compose.yml myapp
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating network myapp_my_network
Creating service myapp_nginx
Creating service myapp_redis
Creating service myapp_postgres
```

Check service status

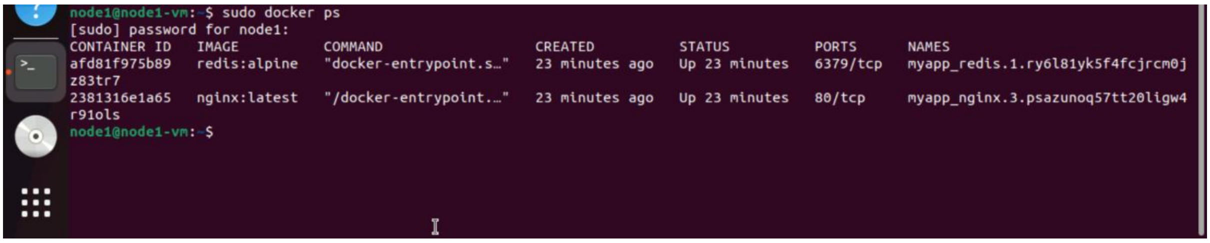


```
master@master-vm:~/doc_swarm$ docker service ls
ID             NAME             MODE         REPLICAS   IMAGE          PORTS
7gypkw9nnerz   myapp_nginx      replicated   3/3        nginx:latest   *:80->80/tcp
w5o1s8vl2sv6   myapp_postgres   replicated   1/1        postgres:13
ty44n2tznstx   myapp_redis      replicated   2/2        redis:alpine
master@master-vm:~/doc_swarm$ 
```

Copy the join token and add worker node-1

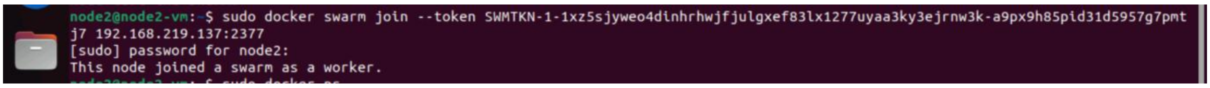


```
node1@node1-vm:~$ sudo docker swarm join --token SWMTKN-1-1xz5sjyweo4dinhrhwjfjulgxef83lx1277uyaa3ky3ejrnw3k-a9px9h85pid31d5957g7pmt
j7 192.168.219.137:2377
[sudo] password for node1:
This node joined a swarm as a worker.
```



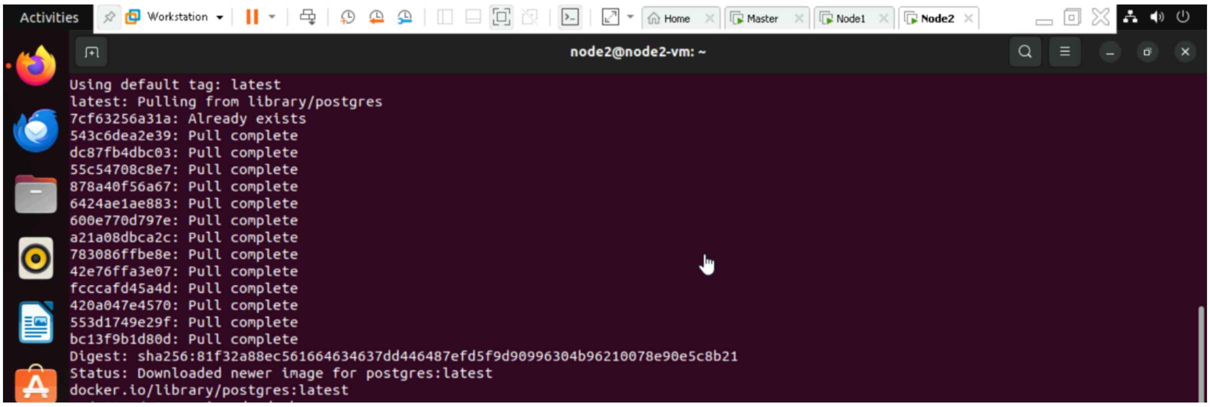
```
node1@node1-vm:~$ sudo docker ps
[sudo] password for node1:
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS      NAMES
afd81f975b89   redis:alpine   "docker-entrypoint.s…"   23 minutes ago   Up 23 minutes   6379/tcp   myapp_redis.1.ry6l81yk5f4fcjrcm0j
z83tr7
2381316e1a65   nginx:latest   "/docker-entrypoint.…"   23 minutes ago   Up 23 minutes   80/tcp     myapp_nginx.3.psazunoq57tt20ligw4
r91ols
node1@node1-vm:~$
```

Copy the join token and add worker node-2

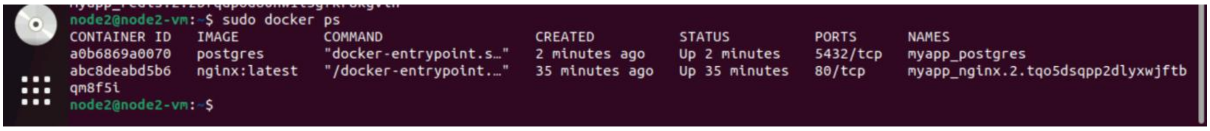


```
node2@node2-vm:~$ sudo docker swarm join --token SWMTKN-1-1xz5sjyweo4dinhrhwjfjulgxef83lx1277uyaa3ky3ejrnw3k-a9px9h85pid31d5957g7pmt
j7 192.168.219.137:2377
[sudo] password for node2:
This node joined a swarm as a worker.
```

Pulled the postgres



```
Using default tag: latest
latest: Pulling from library/postgres
7cf63256a31a: Already exists
543c6dea2e39: Pull complete
dc87fb4dbc03: Pull complete
55c54708c8e7: Pull complete
878a40f56a67: Pull complete
6424ae1ae883: Pull complete
600e770d797e: Pull complete
a21a08dbca2c: Pull complete
783086ffbe8e: Pull complete
42e76ffa3e07: Pull complete
fcccafd45a4d: Pull complete
420a047e4570: Pull complete
553d1749e29f: Pull complete
bc13f9b1d80d: Pull complete
Digest: sha256:81f32a88ec561664634637dd446487efd5f9d90996304b96210078e90e5c8b21
Status: Downloaded newer image for postgres:latest
docker.io/library/postgres:latest
```



```
node2@node2-vm:~$ sudo docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS          PORTS      NAMES
a0b6869a0070   postgres       "docker-entrypoint.s…"   2 minutes ago    Up 2 minutes    5432/tcp   myapp_postgres
abc8deabd5b6   nginx:latest   "/docker-entrypoint.…"   35 minutes ago   Up 35 minutes   80/tcp     myapp_nginx.2.tqo5dsqpp2dlyxwjftb
qm8f5i
node2@node2-vm:~$
```