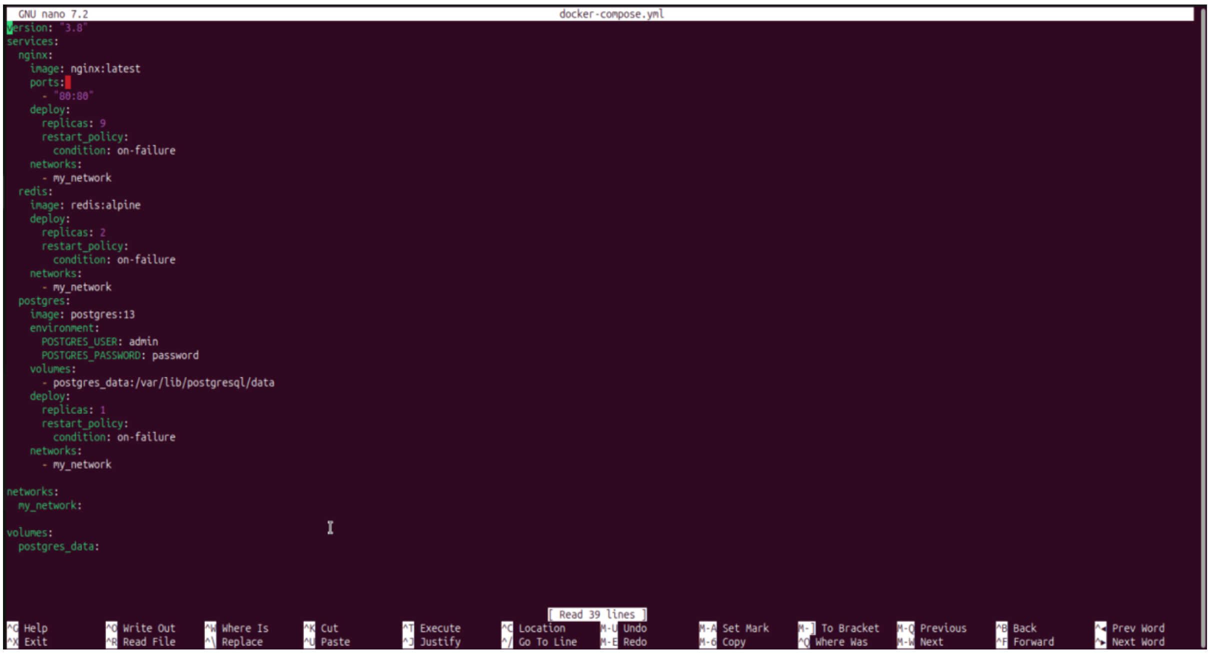
Name: Barathkumar J K

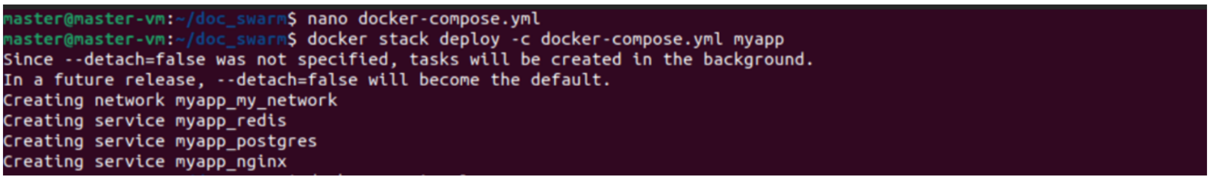
Docker Assignment 3

Multi Node Docker Swarm Cluster

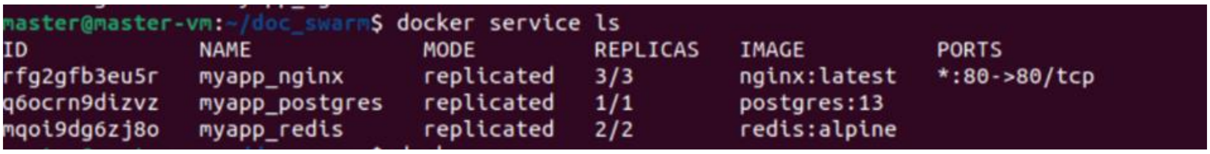
1. Initialize the docker swarm
2. Run the token in the nodes
3. Verify the nodes that are part of swarm
4. Create a directory and create docker-compose.yml file
5. Run the compose file
6. See the list of services running in multiple nodes



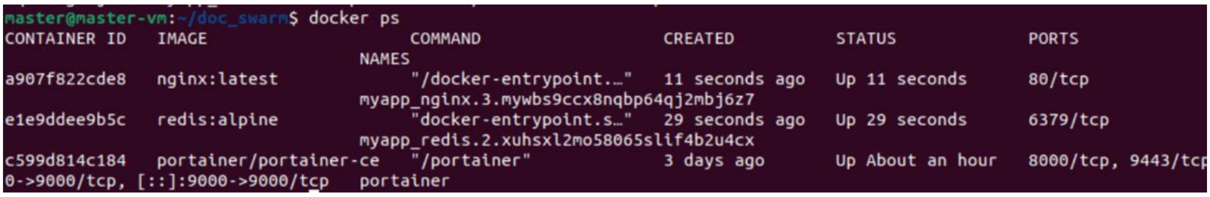
```
version: "3.8"
services:
  nginx:
    image: nginx:latest
    ports:
      - "80:80"
    deploy:
      replicas: 9
      restart_policy:
        condition: on-failure
    networks:
      - my_network
  redis:
    image: redis:alpine
    deploy:
      replicas: 2
      restart_policy:
        condition: 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

networks:
  my_network:

volumes:
  postgres_data:
```



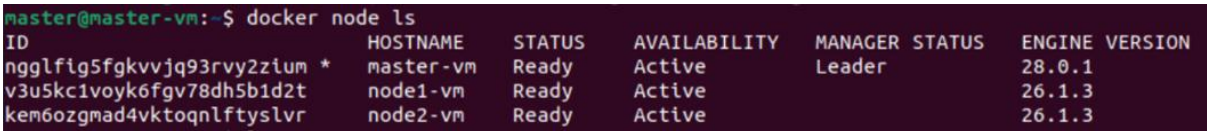
```
master@master-vm:~/doc_swarm$ nano docker-compose.yml
master@master-vm:~/doc_swarm$ docker stack deploy -c docker-compose.yml myapp
Since --detach=false was not specified, tasks will be created in the background.
In a future release, --detach=false will become the default.
Creating network myapp_my_network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



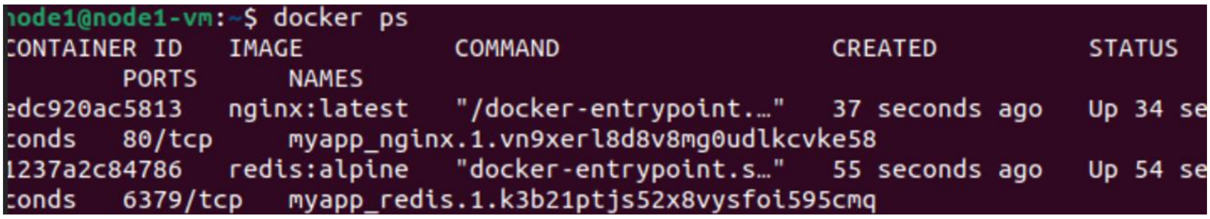
```
master@master-vm:~/doc_swarm$ docker service ls
ID             NAME             MODE         REPLICAS   IMAGE          PORTS
rfg2gfb3eu5r   myapp_nginx      replicated   3/3        nginx:latest   *:80->80/tcp
q6ocrn9dizvz   myapp_postgres   replicated   1/1        postgres:13
mqoi9dg6zj8o   myapp_redis      replicated   2/2        redis:alpine
```



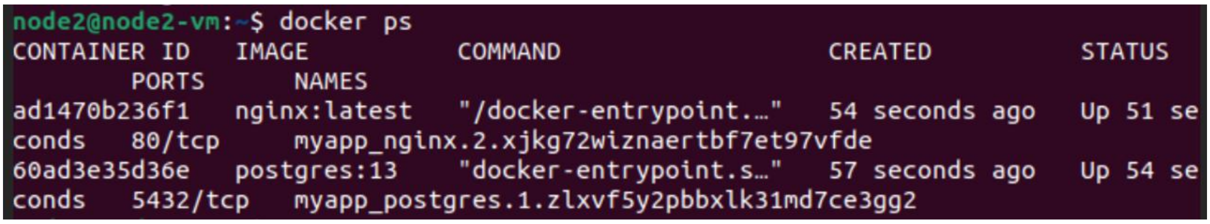
```
master@master-vm:~/doc_swarm$ docker ps
CONTAINER ID   IMAGE                    COMMAND                  CREATED          STATUS              PORTS
                                  NAMES
a907f822cde8   nginx:latest             "/docker-entrypoint.…"   11 seconds ago   Up 11 seconds       80/tcp
                                  myapp_nginx.3.mywbs9ccx8nqbp64qj2mbj6z7
e1e9ddee9b5c   redis:alpine             "docker-entrypoint.s…"   29 seconds ago   Up 29 seconds       6379/tcp
                                  myapp_redis.2.xuhsxl2mo58065slif4b2u4cx
c599d814c184   portainer/portainer-ce   "/portainer"             3 days ago       Up About an hour    8000/tcp, 9443/tcp
0->9000/tcp, [::]:9000->9000/tcp   portainer
```



```
master@master-vm:~$ docker node ls
ID                            HOSTNAME    STATUS    AVAILABILITY   MANAGER STATUS   ENGINE VERSION
ngglfig5fgkvvjq93rvy2zium *   master-vm   Ready     Active         Leader           28.0.1
v3u5kc1voyk6fgv78dh5b1d2t     node1-vm    Ready     Active                          26.1.3
kem6ozgmad4vktoqnlftyslvr     node2-vm    Ready     Active                          26.1.3
```



```
iode1@node1-vm:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
        PORTS      NAMES
edc920ac5813   nginx:latest   "/docker-entrypoint.…"   37 seconds ago   Up 34 se
conds   80/tcp     myapp_nginx.1.vn9xerl8d8v8mg0udlkcvke58
1237a2c84786   redis:alpine   "docker-entrypoint.s…"   55 seconds ago   Up 54 se
conds   6379/tcp   myapp_redis.1.k3b21ptjs52x8vysfoi595cmq
```



```
node2@node2-vm:~$ docker ps
CONTAINER ID   IMAGE          COMMAND                  CREATED          STATUS
        PORTS      NAMES
ad1470b236f1   nginx:latest   "/docker-entrypoint.…"   54 seconds ago   Up 51 se
conds   80/tcp     myapp_nginx.2.xjkg72wiznaertbf7et97vfde
60ad3e35d36e   postgres:13    "docker-entrypoint.s…"   57 seconds ago   Up 54 se
conds   5432/tcp   myapp_postgres.1.zlxvf5y2pbbxlk31md7ce3gg2
```