Project : Multi-Node Docker Swarm Cluster for High

Availability

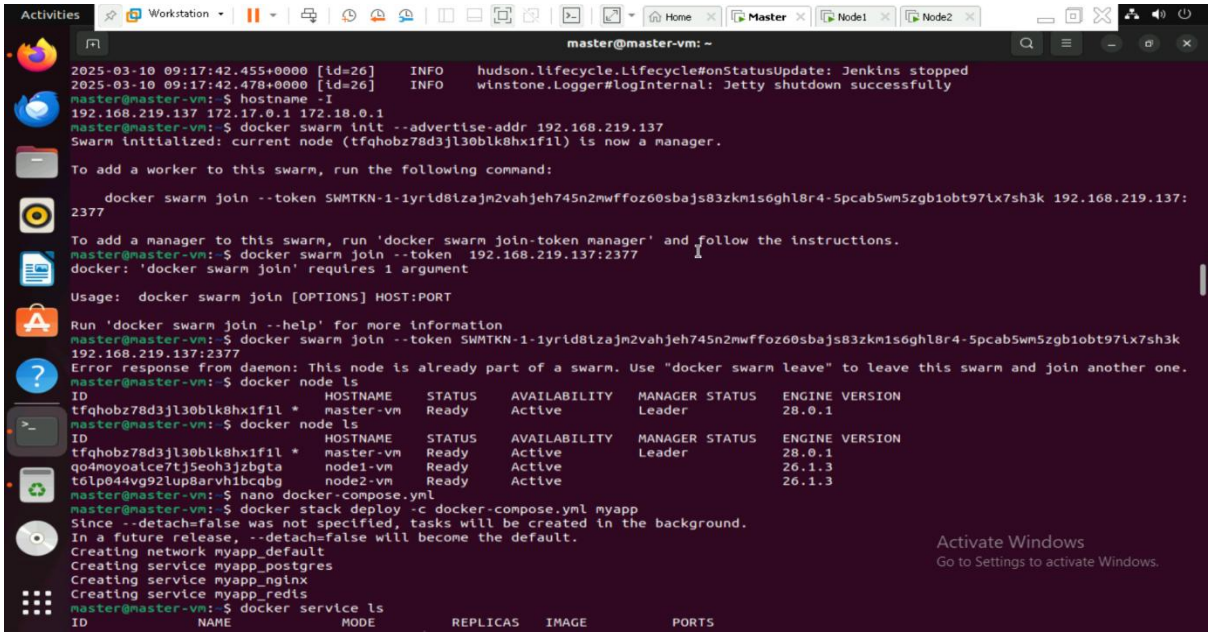
Objective: Deploy a highly available application using Docker Swarm.

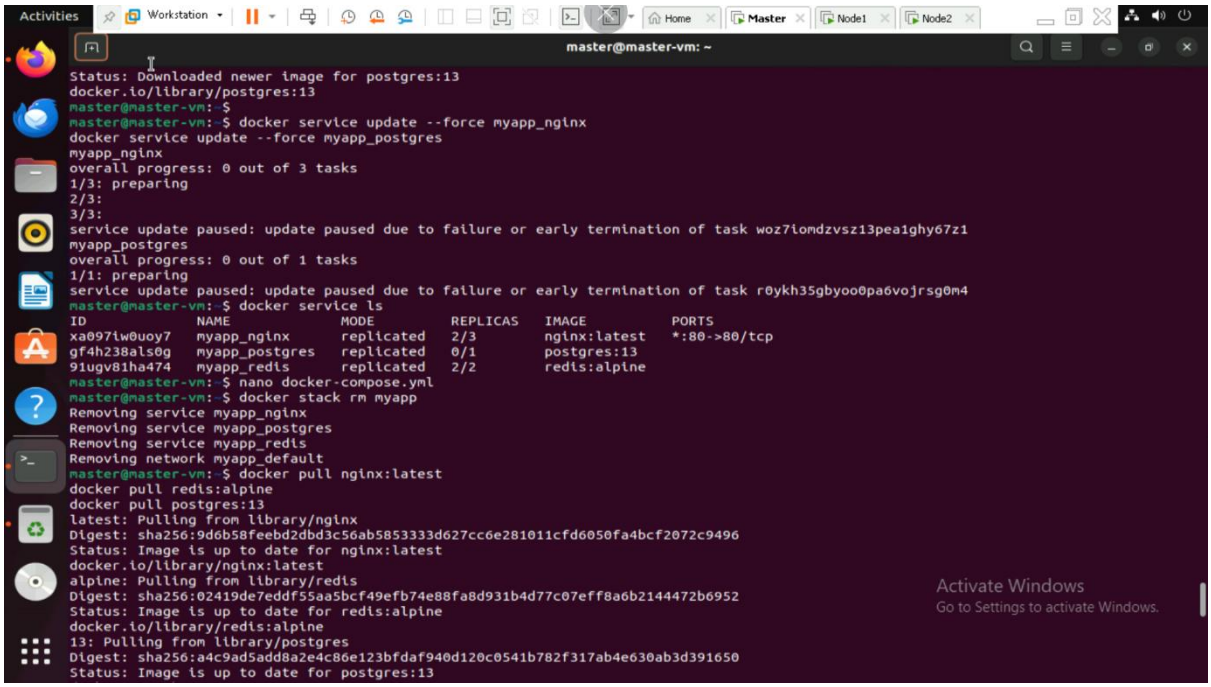
Tech Stack:

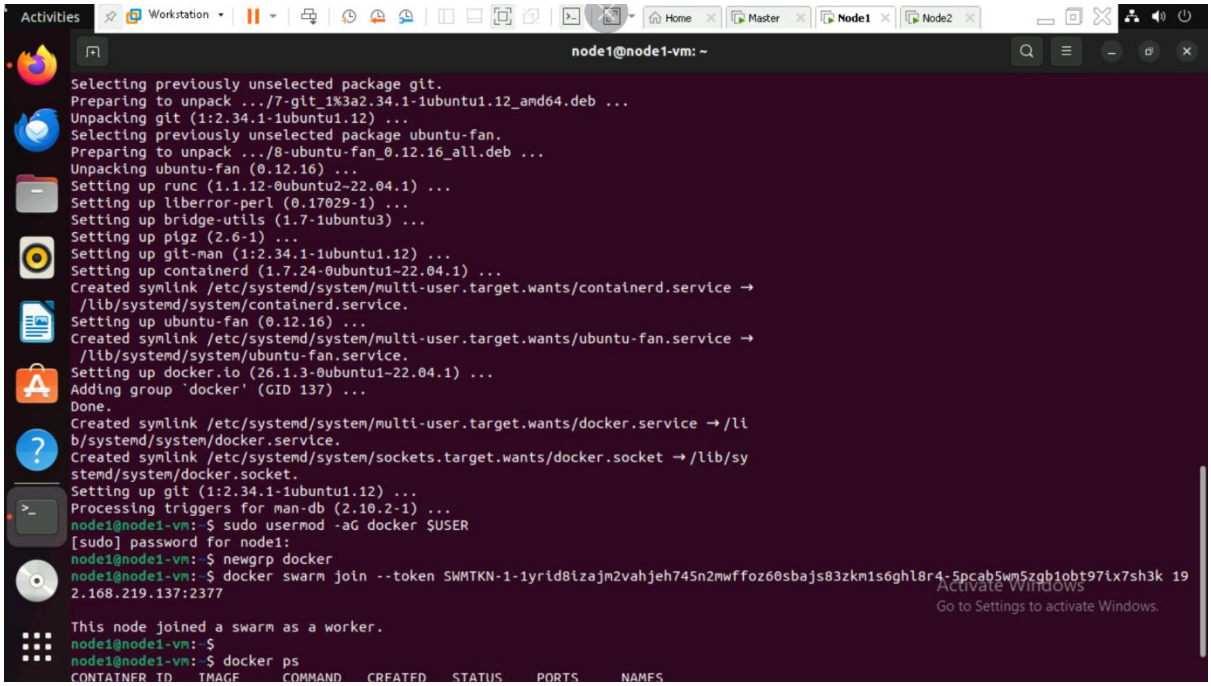
• Docker Swarm (Cluster Orchestration)

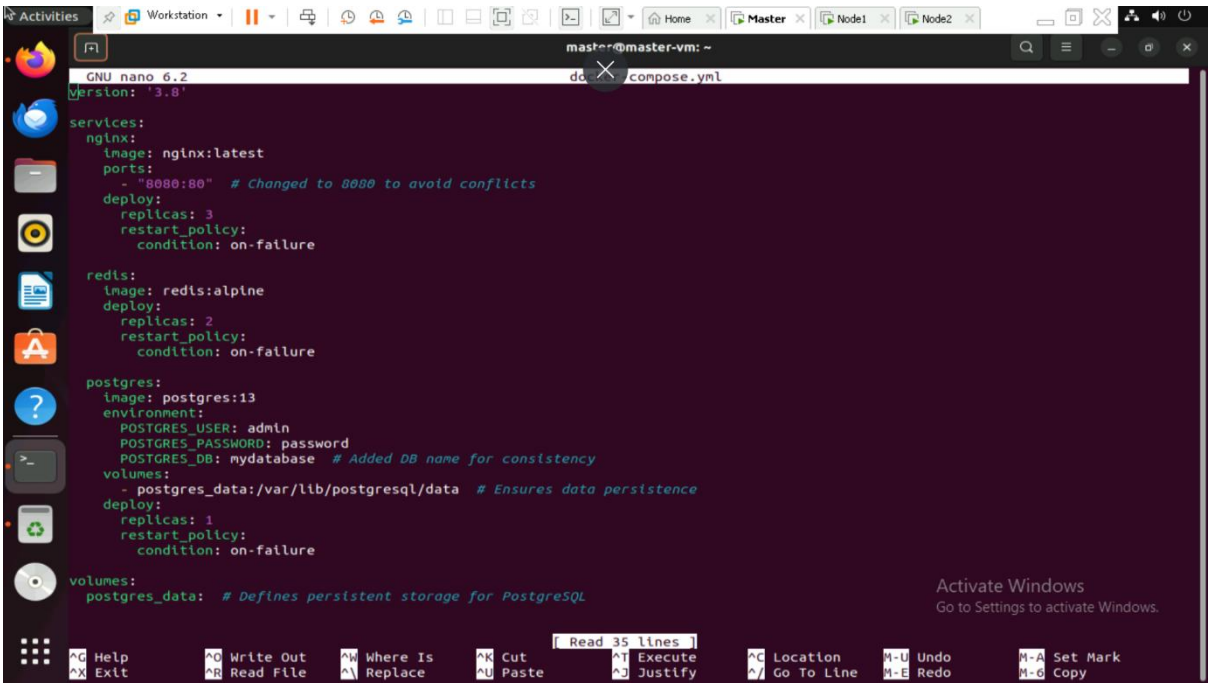
• NGINX (Load Balancer)

• Redis + PostgreSQL (Database Layer)






Activities
Workstation
Home
Master
Node1
Node2
node1@node1-vm: ~
Selecting previously unselected package git.
Preparing to unpack .../7-git_1%3a2.34.1-1ubuntu1.12_amd64.deb ...
Unpacking git (1:2.34.1-1ubuntu1.12) ...
Selecting previously unselected package ubuntu-fan.
Preparing to unpack .../8-ubuntu-fan_0.12.16_all.deb ...
Unpacking ubuntu-fan (0.12.16) ...
Setting up runc (1.1.12-0ubuntu2~22.04.1) ...
Setting up liberror-perl (0.17029-1) ...
Setting up bridge-utils (1.7-1ubuntu3) ...
Setting up pigz (2.6-1) ...
Setting up git-man (1:2.34.1-1ubuntu1.12) ...
Setting up containerd (1.7.24-0ubuntu1~22.04.1) ...
Created symlink /etc/systemd/system/multi-user.target.wants/containerd.service →
/lib/systemd/system/containerd.service.
Setting up ubuntu-fan (0.12.16) ...
Created symlink /etc/systemd/system/multi-user.target.wants/ubuntu-fan.service →
/lib/systemd/system/ubuntu-fan.service.
Setting up docker.io (26.1.3-0ubuntu1~22.04.1) ...
Adding group `docker' (GID 137) ...
Done.
Created symlink /etc/systemd/system/multi-user.target.wants/docker.service → /li
b/systemd/system/docker.service.
Created symlink /etc/systemd/system/sockets.target.wants/docker.socket → /lib/sy
stemd/system/docker.socket.
Setting up git (1:2.34.1-1ubuntu1.12) ...
Processing triggers for man-db (2.10.2-1) ...
node1@node1-vm:~$ sudo usermod -aG docker $USER
[sudo] password for node1:
node1@node1-vm:~$ newgrp docker
node1@node1-vm:~$ docker swarm join --token SWMTKN-1-1yrid8izajm2vahjeh745n2mwffoz60sbajs83zkm1s6ghl8r4-5pcab5wm5zgb1obt97ix7sh3k 19
2.168.219.137:2377
This node joined a swarm as a worker.
node1@node1-vm:~$
node1@node1-vm:~$ docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
Activate Windows
Go to Settings to activate Windows.


Activities
Workstation
Home
Master
Node1
Node2
master@master-vm: ~
GNU nano 6.2
compose.yml
version: '3.8'
services:
nginx:
image: nginx:latest
ports:
- "8080:80" # Changed to 8080 to avoid conflicts
deploy:
replicas: 3
restart_policy:
condition: on-failure
redis:
image: redis:alpine
deploy:
replicas: 2
restart_policy:
condition: on-failure
postgres:
image: postgres:13
environment:
POSTGRES_USER: admin
POSTGRES_PASSWORD: password
POSTGRES_DB: mydatabase # Added DB name for consistency
volumes:
- postgres_data:/var/lib/postgresql/data # Ensures data persistence
deploy:
replicas: 1
restart_policy:
condition: on-failure
volumes:
postgres_data: # Defines persistent storage for PostgreSQL
[Read 35 lines]
^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute ^C Location M-U Undo M-A Set Mark
^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify ^/ Go To Line M-E Redo M-6 Copy
Activate Windows
Go to Settings to activate Windows.

```
Processing triggers for man-db (2.10.2-1) ...
node2@node2-vm:~$ sudo usermod -aG docker $USER
node2@node2-vm:~$ newgrp docker
node2@node2-vm:~$ docker swarm join --token SWMTKN-1-1yrid8izajm2vahjeh745n2mwff
oz60sbajs83zkm1s6ghl8r4-5pcab5wm5zgb1obt97ix7sh3k 192.168.219.137:2377
This node joined a swarm as a worker.
node2@node2-vm:~$
```

```
master@master-vm:~$ 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.
Updating service myapp_nginx (id: x7livbza3htken0al0l6mp0oi)
image nginx:latest could not be accessed on a registry to record
its digest. Each node will access nginx:latest independently,
possibly leading to different nodes running different
versions of the image.

Updating service myapp_redis (id: ss5km9av8pxyug52barot3qwz)
image redis:alpine could not be accessed on a registry to record
its digest. Each node will access redis:alpine independently,
possibly leading to different nodes running different
versions of the image.

Updating service myapp_postgres (id: 9p3vxtcoojj857w5itywqar7s)
image postgres:13 could not be accessed on a registry to record
its digest. Each node will access postgres:13 independently,
possibly leading to different nodes running different
versions of the image.

master@master-vm:~$ docker service ls
ID             NAME             MODE         REPLICAS   IMAGE          PORTS
x7livbza3htk   myapp_nginx      replicated   3/3        nginx:latest   *:8080->80/tcp
9p3vxtcoojj8   myapp_postgres   replicated   1/1        postgres:13
ss5km9av8pxy   myapp_redis      replicated   2/2        redis:alpine
master@master-vm:~$
```