

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)

```
master@master-vm: ~
2025-03-10 09:17:42.455+0000 [id=26] INFO hudson.lifecycle.Lifecycle#onStatusUpdate: Jenkins stopped
2025-03-10 09:17:42.478+0000 [id=26] INFO winstone.Logger#logInternal: Jetty shutdown successfully
master@master-vm: ~$ hostname -I
192.168.219.137 172.17.0.1 172.18.0.1
master@master-vm: ~$ docker swarm init --advertise-addr 192.168.219.137
Swarm initialized: current node (tfqhobz78d3jl30blk8hx1f1l) is now a manager.

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

    docker swarm join --token SWMTKN-1-1yrid8lza2vm2vahnjeh745n2mwffoz60sbajs83zkmis6ghl8r4-5pcab5wm5zgb1obt97lx7sh3k 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 swarm join --token SWMTKN-1-1yrid8lza2vm2vahnjeh745n2mwffoz60sbajs83zkmis6ghl8r4-5pcab5wm5zgb1obt97lx7sh3k 192.168.219.137:2377
docker: 'docker swarm join' requires 1 argument
Usage:  docker swarm join [OPTIONS] HOST:PORT

Run 'docker swarm join --help' for more information
master@master-vm: ~$ docker swarm join --token SWMTKN-1-1yrid8lza2vm2vahnjeh745n2mwffoz60sbajs83zkmis6ghl8r4-5pcab5wm5zgb1obt97lx7sh3k 192.168.219.137:2377
Error response from daemon: This node is already part of a swarm. Use "docker swarm leave" to leave this swarm and join another one.
master@master-vm: ~$ docker node ls
ID                HOSTNAME        STATUS        AVAILABILITY    MANAGER STATUS   ENGINE VERSION
tfqhobz78d3jl30blk8hx1f1l * master-vm      Ready         Active           Leader            28.0.1
master@master-vm: ~$ docker node ls
ID                HOSTNAME        STATUS        AVAILABILITY    MANAGER STATUS   ENGINE VERSION
tfqhobz78d3jl30blk8hx1f1l * master-vm      Ready         Active           Leader            28.0.1
q04moyoa1ce7t35eoh3jzbgta node1-vm       Ready         Active           28.0.1
t6lp044vg92lup8arvhibcqbq node2-vm       Ready         Active           26.1.3
master@master-vm: ~$ nano docker-compose.yml
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.
Creating network myapp_default
Creating service myapp_postgres
Creating service myapp_nginx
Creating service myapp_redis
master@master-vm: ~$ docker service ls
ID                NAME                MODE                REPLICAS        IMAGE                PORTS
```

```
master@master-vm: ~
Status: Downloaded newer image for postgres:13
docker.io/library/postgres:13
master@master-vm: ~$
master@master-vm: ~$ docker service update --force myapp_nginx
docker service update --force myapp_postgres
myapp_nginx
overall progress: 0 out of 3 tasks
1/3: preparing
2/3:
3/3:
service update paused: update paused due to failure or early termination of task woz7londzvsz13peaighy67z1
myapp_postgres
overall progress: 0 out of 1 tasks
1/1: preparing
service update paused: update paused due to failure or early termination of task r0ykh35gbyoo0pa6vojrsg0m4
master@master-vm: ~$ docker service ls
ID                NAME                MODE                REPLICAS        IMAGE                PORTS
xa097lw0uoy7     myapp_nginx         replicated          2/3              nginx:latest        *:80->80/tcp
gt4h238als0g     myapp_postgres      replicated          0/1              postgres:13
91ugv81ha474     myapp_redis         replicated          2/2              redis:alpine
master@master-vm: ~$ nano docker-compose.yml
master@master-vm: ~$ docker stack rm myapp
Removing service myapp_nginx
Removing service myapp_postgres
Removing service myapp_redis
Removing network myapp_default
master@master-vm: ~$ docker pull nginx:latest
docker pull redis:alpine
docker pull postgres:13
latest: Pulling from library/nginx
Digest: sha256:9d0b58feebd2dbd3c56ab5853333d627cc6e281011cfd6050fa4bcf2072c9496
Status: Image is up to date for nginx:latest
docker.io/library/nginx:latest
alpine: Pulling from library/redis
Digest: sha256:02419de7edd55aa5bcf49efb74e88fa8d931b4d77c07eff8a6b2144472b6952
Status: Image is up to date for redis:alpine
docker.io/library/redis:alpine
13: Pulling from library/postgres
Digest: sha256:a4c9ad5add8a2e4c86e123bfda940d120c0541b782f317ab4e630ab3d391650
Status: Image is up to date for postgres:13
```

```
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-1-1yrid81zajm2vahjeh745n2mwffoz60sbajs83zkn1s6ghl8r4-5pcab5wm5zgb10bt97ix7sh3k 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
```

```
master@master-vm: ~  
GNU nano 6.2  
Version: '3.8'  
compose.yml  
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
```

```

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: x7livbza3htken0a10l6mp0oi)
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: ss5km9av8pxyug52barot3q wz)
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: 9p3vxtcoojj857w5ltywqar7s)
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:~$

```