CARA MENJALANKAN PROGRAM

Sebelum *database cluster* dapat dijalankan, pastikan ketiga node sudah dikonfigurasi docker swarm cluster.

1. *Copy file* listing sesuai struktur direktori berikut

```
.
├── mysql-test
│   ├── docker-compose.yml
│   ├── Dockerfile
│   ├── start.sh
│   └── test.sh
├── proxysql
│   ├── conf
│   │   ├── proxysql-admin.cnf
│   │   └── start.sh
│   ├── docker-compose.yml
│   └── Dockerfile
└── pxc
    ├── conf
    │   ├── check.sh
    │   ├── create-user.sh
    │   ├── mysqld.cnf
    │   ├── retention
    │   └── start.sh
    ├── deploy.sh
    ├── docker-compose-secrets.yaml
    ├── docker-compose.yml
    └── Dockerfile

5 directories, 17 files
```

Gambar 1 Struktur direktori

2. Pindah ke direktori pxc

```
manager:~/# cd pxc
manager:~/pxc#
```

Gambar 2 Pindah ke direktori pxc

3. *Build image database*

```
manager:~/pxc# docker-compose build --no-cache
```

Gambar 3 *Build image*

4. *Push image database*

```
manager:~/pxc# docker-compose push
```

Gambar 4 *Push image*

5. Konfigurasi *constraint* masing-masing *server*

```
manager:~/pxc# docker node update --label-add pxc1=true
manager
manager:~/pxc# docker node update --label-add pxc2=true
worker1
manager:~/pxc# docker node update --label-add pxc3=true
worker2
```

Gambar 5 Konfigurasi *constraint server*

6. *Deploy database cluster*

```
manager:~/pxc# ./deploy.sh
Enter docker service name: pxc_cluster
Creating network pxc_cluster_pxc_distributed
Creating service pxc_cluster_pxc3
Creating service pxc_cluster_pxc1
Creating service pxc_cluster_pxc2
Please wait for the cluster to bootstrap….............

Cluster status:  Synced
Cluster member:  pxc1:3306,pxc2:3306,pxc3:3306
Prepare the cluster to be ready to accept connections.
4e31e381ca39
4e31e381ca39
Done. Cluster is ready.

manager:~/pxc#
```

Gambar 6 *Deploy database*

7. Verifikasi *database cluster*

```
manager:~/pxc# docker service ls
ID             NAME               MODE         REPLICAS
IMAGE                       PORTS
mx850rsscizv   pxc_cluster_pxc1   replicated   1/1
127.0.0.1:5000/pxc:latest
```

```
pjp0pgir74tl   pxc_cluster_pxc2   replicated   1/1
127.0.0.1:5000/pxc:latest
odm12f69k95o   pxc_cluster_pxc3   replicated   1/1
127.0.0.1:5000/pxc:latest
```

Gambar 7 Verifikasi *database cluster*

8. Pindah ke direktori proxysql

```
manager:~/# cd ../proxysql
manager:~/proxysql#
```

Gambar 8 Pindah ke direktori proxysql

9. *Build image* proxysql

```
manager:~/proxysql# docker-compose build --no-cache
```

Gambar 9 *Build image*

10. *Push image* proxysql

```
manager:~/proxysql# docker-compose push
```

Gambar 10 *Push image*

11. *Deploy* proxysql

```
manager:~/proxysql# docker service create --name proxysql2
--replicas  3  --network  pxc_cluster_pxc_distributed  -p
6033:6033  127.0.0.1:5000/proxysql2
```

Gambar 11 *Deploy* proxysql

12. Verifikasi *service* proxysql

```
manager:~/proxysql# docker service ls
ID             NAME               MODE         REPLICAS
IMAGE                             PORTS
yxf8qemoaran   proxysql2          replicated   3/3
127.0.0.1:5000/proxysql2:latest   *:6033->6033/tcp
```

Gambar 12 Verifikasi proxysql

13. Pengujian koneksi dari laptop *workstation*

```
root@workstation:~# mysql -u hanafi -p -h manager -P 6033
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 58
```

```
Server version: 5.5.30 (ProxySQL)

Copyright © 2009-2021 Percona LLC and/or its affiliates
Copyright © 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation
and/or its
affiliates. Other names may be trademarks of their
respective
owners.

Type ‘help;’ or ‘\h’ for help. Type ‘\c’ to clear the
current input statement.

mysql> create database test_koneksi;
Query OK, 1 row affected (0.04 sec)

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| mysql              |
| performance_schema |
| sys                |
| test_koneksi       |
+--------------------+
5 rows in set (0.00 sec)

mysql>
```

Gambar 13 Pengujian koneksi dari *workstation*

14. Pindah ke direktori mysql-test

```
manager:~/# cd ../mysql-test
manager:~/mysql-test#
```

Gambar 14 Pindah ke direktori mysql-test

15. *Build image* mysql-test

```
manager:~/mysql-test# docker-compose build –no-cache
```

Gambar 15 *Build image*

16. *Push image* mysql-test

```
manager:~/mysql-test# docker-compose push
```

Gambar 16 *Push image*

17. *Deploy* mysql-test pada node *manager*

```
manager:~/mysql-test# docker service create --name mysql-
test --replicas 1 --network pxc_cluster_pxc_distributed --
constraint node.labels.pxc1==true 127.0.0.1:5000/mysqltest
```

Gambar 17 *Deploy* mysql-test

18. Membuat *database* dan tabel yang akan digunakan untuk pengujian *high availability* dan *reliability*

```
root@workstation:~# mysql -u hanafi -p -h manager -P 6033
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 14358
Server version: 5.5.30 (ProxySQL)

Copyright (c) 2009-2021 Percona LLC and/or its affiliates
Copyright (c) 2000, 2021, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation
and/or its
affiliates. Other names may be trademarks of their
respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the
current input statement.

mysql> create database test;
Query OK, 1 row affected (0.07 sec)

mysql> use test;
Database changed

mysql> create table ha (value VARCHAR(40) NOT NULL,
PRIMARY KEY(value));
Query OK, 0 rows affected (0.14 sec)

mysql> create table reliability (value VARCHAR(40) NOT
NULL, PRIMARY KEY(value));
Query OK, 0 rows affected (0.14 sec)

mysql>
```

Gambar 18 Membuat *database* untuk pengujian

19. Perintah pengujian *high availability*

```
root@test-container:/# bash /root/test.sh 300 ha
```

Gambar 19 Perintah pengujian *high availability*

20. Perintah pengujian *reliability*

```
root@test-container:/# bash /root/test.sh 300 reliability
```

Gambar 20 Perintah pengujian *reliability*

21. Perintah untuk mengosongkan tabel pengujian

```
mysql> truncate table test.reliability;
Query OK, 0 rows affected (0.17 sec)

mysql> select count(*) from test.reliability;
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.01 sec)

mysql>
```

Gambar 21 Perintah mengosongkan tabel