KONFIGURASI DATABASE MARIADB PADA MASTER - SLAVE SISTEM TERDISTRIBUSI

OLEH :

NAMA : MUHAMAD ARIF ABDILLAH

NIM : 09011182126008

KELAS : SK6B

DOSEN PENGAMPUH : ADI HERMANSYAH, M.T

FAKULTAS ILMU KOMPUTER

PROGRAM STUDI SISTEM KOMPUTER

UNIVERSITAS SRIWIJAYA

TAHUN AJARAN

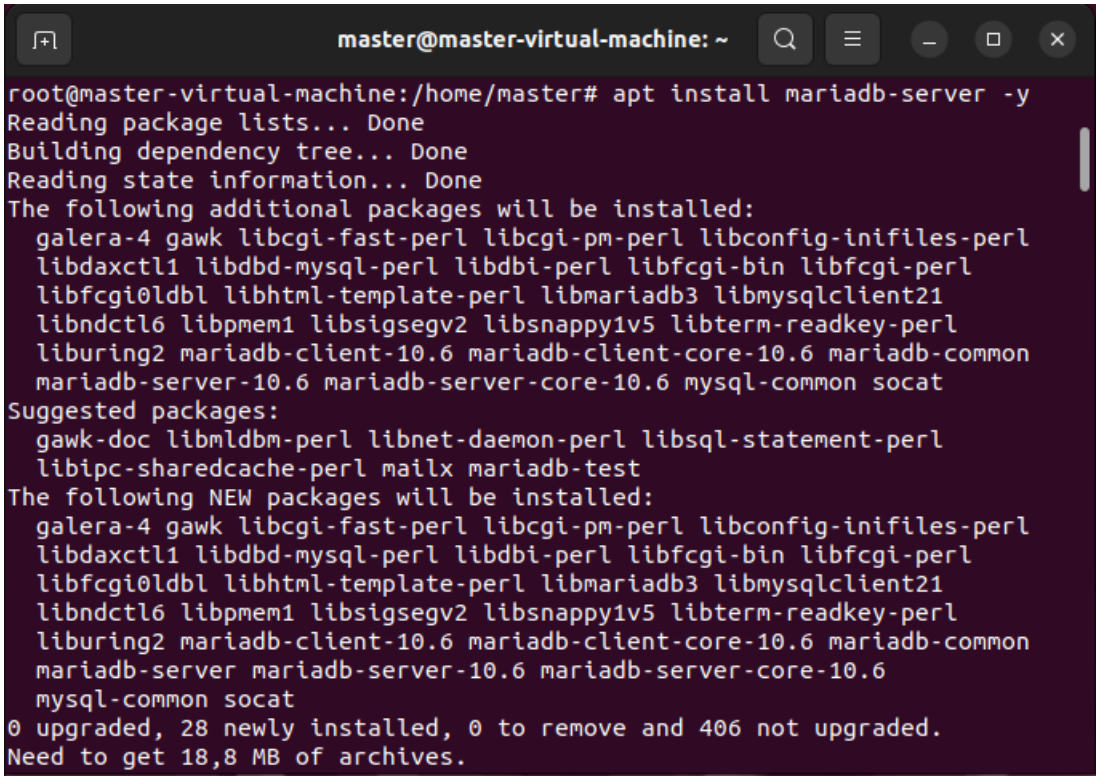
2024

Langkah-langkah:

1. **IP pada Master dan IP pada Slave**

Master	Slave
192.168.153.128	192.168.153.129

2. **Install MariaDB pada Master dan Slave**



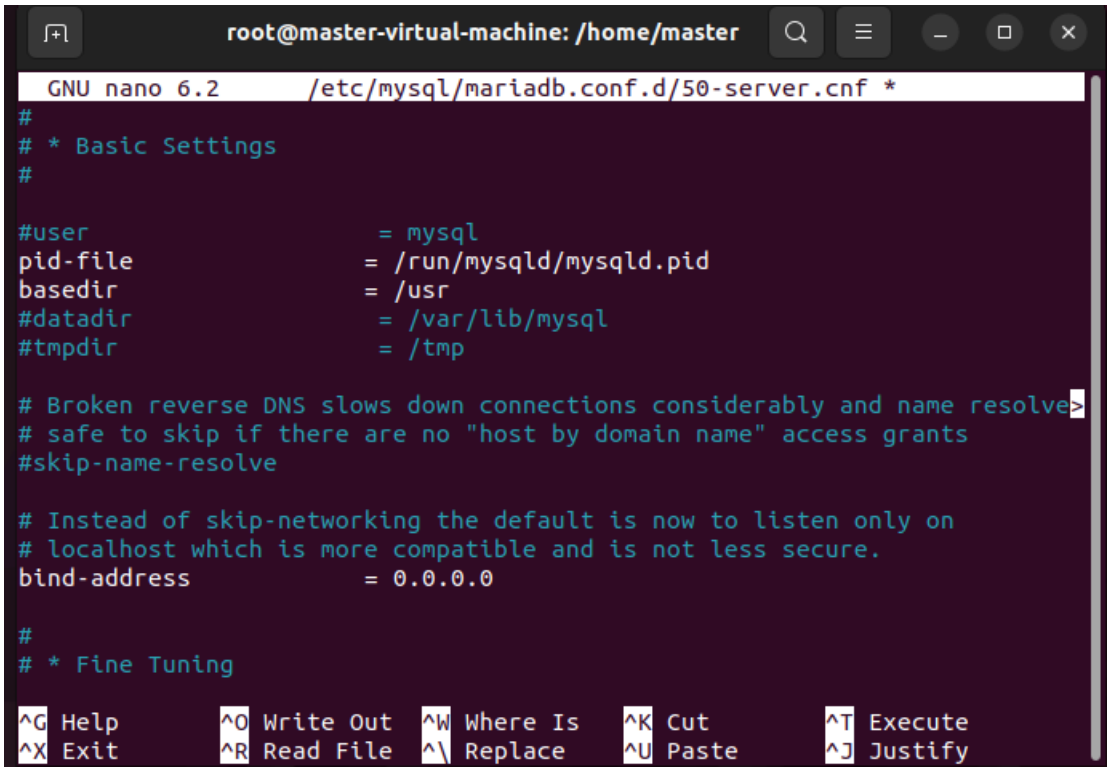
3. Install mysql MariaDB pada Master - Slave

```
root@master-virtual-machine:/home/master# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

In order to log into MariaDB to secure it, we'll need the current
password for the root user. If you've just installed MariaDB, and
haven't set the root password yet, you should just press enter here.

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password or using the unix_socket ensures that nobody
can log into the MariaDB root user without the proper authorisation.

You already have your root account protected, so you can safely answer 'n'
.

Switch to unix_socket authentication [Y/n] n
 ... skipping.

You already have your root account protected, so you can safely answer 'n'
.

Change the root password? [Y/n] n
 ... skipping.
```

```
Remove anonymous users? [Y/n] y
 ... Success!

Normally, root should only be allowed to connect from 'localhost'.  This
ensures that someone cannot guess at the root password from the network.

Disallow root login remotely? [Y/n] y
 ... Success!

By default, MariaDB comes with a database named 'test' that anyone can
access.  This is also intended only for testing, and should be removed
before moving into a production environment.

Remove test database and access to it? [Y/n] y
 - Dropping test database...
 ... Success!
 - Removing privileges on test database...
 ... Success!

Reloading the privilege tables will ensure that all changes made so far
will take effect immediately.
```

4. Konfigurasi MariaDB

```
root@master-virtual-machine:/home/master# nano /etc/mysql/mariadb.conf.d/
50-server.cnf
```



- **Pada Master**

```
# The following can be used as easy to replay backup logs or for replicat
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
server-id = 102
log_bin = /var/log/mysql/mysql-bin.log
#expire_logs_days = 10
#max_binlog_size = 100M
```

- **Pada Slave**

```
# The following can be used as easy to replay backup logs or for replicat
# note: if you are setting up a replication slave, see README.Debian about
#       other settings you may need to change.
server-id = 103
log_bin = /var/log/mysql/mysql-bin.log
#expire_logs_days = 10
#max_binlog_size = 100M
```

5. **Akses ke Database MariaDB dan Memberikan izin kepada pengguna ‘replica’ untuk melakukan replikasi dari master MYSQL ke slave MySQL**

```
root@master-virtual-machine:/home/master# mysql -u root -p
Enter password:
```

- **Dari MASTER KE SLAVE**

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'replica'@'192.168.153
.129' IDENTIFIED BY '123';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> SHOW MASTER STATUS;
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000001 |     2094 |              |                  |
+------------------+----------+--------------+------------------+
1 row in set (0,000 sec)
```

- **Dari SLAVE KE MASTER**

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'replica'@'192.168.153
.128' IDENTIFIED BY '123';
Query OK, 0 rows affected (0,002 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;
Query OK, 0 rows affected (0,001 sec)

MariaDB [(none)]> SHOW MASTER STATUS;
+------------------+----------+--------------+------------------+
| File             | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+------------------+----------+--------------+------------------+
| mysql-bin.000001 |     1771 |              |                  |
+------------------+----------+--------------+------------------+
1 row in set (0,000 sec)
```

6. Konfigurasi Replica pada Master MariaDB ke Slave

```
MariaDB [(none)]> CHANGE MASTER TO
    -> MASTER_HOST='192.168.153.129',
    -> MASTER_USER='replica',
    -> MASTER_PASSWORD='123',
    -> MASTER_LOG_FILE='mysql-bin.000001',
    -> MASTER_LOG_POS=974;
Query OK, 0 rows affected (0,040 sec)
```

Status dari Konfigurasi Replica:

```
MariaDB [(none)]> SHOW SLAVE STATUS \G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 192.168.153.129
                  Master_User: replica
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000001
          Read_Master_Log_Pos: 1771
               Relay_Log_File: mysqld-relay-bin.000002
                Relay_Log_Pos: 555
        Relay_Master_Log_File: mysql-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 1771
              Relay_Log_Space: 865
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 103
```

Keterangan:

Dengan menggunakan perintah ini, server MariaDB akan dikonfigurasi untuk memulai replikasi dari server master yang berada di host **192.168.153.129**, menggunakan pengguna **replica** dengan kata sandi **123**, dimulai dari posisi tertentu di dalam file log bin **mysql-bin.000001.**

7. Konfigurasi Replica pada Slave MariaDB ke Master

```
MariaDB [(none)]> CHANGE MASTER TO MASTER_HOST='192.168.153.128', MASTER_U
SER='replica', MASTER_PASSWORD='123', MASTER_LOG_FILE='mysql-bin.000001',
MASTER_LOG_POS=2094;
Query OK, 0 rows affected (0,008 sec)
```

Status dari Konfigurasi Replica:

```
MariaDB [(none)]> SHOW SLAVE STATUS \G;
*************************** 1. row ***************************
               Slave_IO_State: Waiting for master to send event
                  Master_Host: 192.168.153.128
                  Master_User: replica
                  Master_Port: 3306
                Connect_Retry: 60
              Master_Log_File: mysql-bin.000001
          Read_Master_Log_Pos: 2094
               Relay_Log_File: mysqld-relay-bin.000002
                Relay_Log_Pos: 555
        Relay_Master_Log_File: mysql-bin.000001
             Slave_IO_Running: Yes
            Slave_SQL_Running: Yes
              Replicate_Do_DB:
          Replicate_Ignore_DB:
           Replicate_Do_Table:
       Replicate_Ignore_Table:
      Replicate_Wild_Do_Table:
  Replicate_Wild_Ignore_Table:
                   Last_Errno: 0
                   Last_Error:
                 Skip_Counter: 0
          Exec_Master_Log_Pos: 2094
              Relay_Log_Space: 865
              Until_Condition: None
               Until_Log_File:
                Until_Log_Pos: 0
           Master_SSL_Allowed: No
           Master_SSL_CA_File:
           Master_SSL_CA_Path:
              Master_SSL_Cert:
            Master_SSL_Cipher:
               Master_SSL_Key:
        Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
                Last_IO_Errno: 0
                Last_IO_Error:
               Last_SQL_Errno: 0
               Last_SQL_Error:
  Replicate_Ignore_Server_Ids:
             Master_Server_Id: 102
```

Keterangan:

Dengan menggunakan perintah ini, server MariaDB akan dikonfigurasi untuk memulai replikasi dari server master yang berada di host **192.168.153.128**, menggunakan pengguna **replica** dengan kata sandi **123**, dimulai dari posisi tertentu di dalam file log bin **mysql-bin.000001.**

8. **Membuat Database pada Master dan Slave**

MASTER PADA SLAVE

```
MariaDB [(none)]> CREATE DATABASE db_server_arip2;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| db_server_arip2    |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
5 rows in set (0,001 sec)
```

SLAVE PADA MASTER

```
MariaDB [(none)]> CREATE DATABASE db_server_arip1;
Query OK, 1 row affected (0,004 sec)
```

```
MariaDB [(none)]> START SLAVE;
Query OK, 0 rows affected (0,000 sec)

MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| db_server_arip1    |
| db_server_arip2    |
| information_schema |
| mysql              |
| performance_schema |
| sys                |
+--------------------+
6 rows in set (0,000 sec)
```

Dari hasil tersebut, bias dilihat bahwa konfigurasi replica database mariaDB Master – Slave telah Berhasil

9. **MEMBUAT DATABASE MASTER PADA SLAVE SEBAGAI PEMBUKTIAN APAKAH TELAH BERHASIL (hanya untuk memastikan)**

```
MariaDB [(none)]> CREATE DATABASE arip2;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> create database inislave;
Query OK, 1 row affected (0,003 sec)
```

HASIL REPLICA

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| arip2              |
| db_server_arip1    |
| db_server_arip2    |
| halo_ini_arip      |
| information_schema |
| inislave           |
| mysql              |
| performance_schema |
| skb                |
| sys                |
| test               |
+--------------------+
11 rows in set (0,000 sec)
```

10. **MEMBUAT DATABASE SLAVE PADA MASTER SEBAGAI PEMBUKTIAN APAKAH TELAH BERHASIL**

```
MariaDB [(none)]> create database halo_ini_arip;
Query OK, 1 row affected (0,001 sec)

MariaDB [(none)]> create database skb;
Query OK, 1 row affected (0,000 sec)
```

HASIL REPLICA

```
MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| db_server_arip2    |
| halo_ini_arip      |
| information_schema |
| mysql              |
| performance_schema |
| skb                |
| sys                |
| test               |
+--------------------+
8 rows in set (0,001 sec)
```

KESIMPULAN:

Konfigurasi master-slave dalam MariaDB adalah proses di mana dua server MariaDB disiapkan sedemikian rupa sehingga satu bertindak sebagai master yang menyediakan data yang akan direplikasi, sementara yang lain bertindak sebagai slave yang menerima dan mengonfirmasi replikasi data. Ini adalah pendekatan umum dalam sistem terdistribusi untuk meningkatkan ketersediaan data, performa, dan pemulihan bencana.

Langkah-langkah utama dalam konfigurasi master-slave MariaDB adalah sebagai berikut:

1) Penetapan Alamat IP: Setiap server diberikan alamat IP unik, satu untuk master dan satu untuk slave, penting agar komunikasi dalam jaringan bisa berjalan.
2) Instalasi MariaDB: MariaDB diinstal pada kedua server dengan mengunduh dan mengonfigurasi pengaturan dasar.
3) Konfigurasi MariaDB: Pengaturan MariaDB disesuaikan dengan peran masing-masing server, termasuk konfigurasi sebagai master atau slave, serta pengaturan jaringan, izin akses, dan replikasi.
4) Akses Database: Pengguna dapat mengakses database dari kedua server untuk operasi baca dan tulis sesuai dengan izin yang diberikan.
5) Izin Replikasi: Izin diberikan kepada pengguna yang akan digunakan untuk replikasi data antara master dan slave menggunakan perintah GRANT.
6) Konfigurasi Replikasi: Pengaturan khusus diatur pada master dan slave untuk memfasilitasi replikasi data, termasuk penentuan server master dan slave, autentikasi pengguna, informasi log bin untuk titik awal replikasi, dll.
7) Verifikasi Replikasi: Verifikasi dilakukan untuk memastikan bahwa data dari master berhasil direplikasi ke slave dan sebaliknya, memastikan bahwa proses replikasi berjalan dengan benar.

Proses ini membentuk dasar konfigurasi master-slave MariaDB, memberikan redundansi data dan meningkatkan ketersediaan serta keandalan layanan dalam sistem terdistribusi.