

# Database Replication

## Index

1. Setting Instance : Install Mysql Database
2. Setting Master Server
3. Setting Slave Server
4. 실습 : 두개의 데이터 베이스를 서로 마스터 슬레이브로 설정

## Reference

- <https://www.bogotobogo.com/DevOps/AWS/aws-MySQL-Replication-Master-Slave.php>

## 1. Setting Instance : Install Mysql Database

```
$ sudo apt-get update
```

```
$ sudo apt-get install -y mysql-server mysql-client
```

```
$ sudo mysql
```

```
mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH mysql_native_password  
BY 'rada';
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> exit
```

```
$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
- bind-address = 0.0.0.0
```

```
$ sudo mysql -u root -prada
```

```
mysql> grant all privileges on *.* to root@'%' identified by 'rada';
```

```
$ sudo service mysql restart
```

## 2. Setting Master Server

```
$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
-----  
server_id = 1
```

```
log_bin = /var/log/mysql/mysql-bin.log
```

```
binlog_do_db = mydb1  
-----
```

```
$ sudo service mysql restart
```

```
$ mysql -u root -p
```

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'root'@'%' IDENTIFIED BY 'rada';
```

```
mysql> FLUSH PRIVILEGES;
```

```
mysql> CREATE DATABASE mydb1;
```

```
mysql> show databases;
```

```
mysql> use mydb1
```

```
# 데이터 베이스 백업을 위해 테이블에 LOCK을 걸어둡니다.
```

```
mysql> FLUSH TABLES WITH READ LOCK;
```

```
# Master 의 상태를 확인합니다.
```

```
mysql> SHOW MASTER STATUS;
```

```

+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| mysql-bin.000001 |      589 | mydb1         |                   |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

```

Posion Number 를 기억해 둡니다. Slave 설정시에 사용해야 합니다.

# mydb1 데이터 베이스를 mydb1.sql 파일로 백업합니다.

```
$ mysqldump -u root -p --opt mydb1 > mydb1.sql
```

# 데이터 베이스의 LOCK을 해제 합니다.

```
$ mysql -u root -p
```

```
mysql> use mydb1;
```

```
mysql> UNLOCK TABLES;
```

### 3. Setting Slave Server

# mydb1 데이터 베이스 생성

```
$ mysql -u root -p
```

```
mysql> CREATE DATABASE mydb1;
```

```
mysql> exit
```

# mydb1 복원(restore)

```
$ mysql -u root -p mydb1 < mydb1.sql
```

# 설정 파일 변경

```
$ sudo vi /etc/mysql/mysql.conf.d/mysqld.cnf
```

```
-----  
server_id = 2
```

```
relay-log = /var/log/mysql/mysql-relay-bin.log
```

```
log_bin = /var/log/mysql/mysql-bin.log
```

```
binlog_do_db = mydb1  
-----
```

# mysql 서버 재시작

```
$ sudo service mysql restart
```

# Replication Setup on Slave

```
mysql> CHANGE MASTER TO MASTER_HOST='13.125.213.230',  
MASTER_USER='root', MASTER_PASSWORD='rada', MASTER_LOG_FILE='mysql-  
bin.000001', MASTER_LOG_POS=589;
```

# 동기화 시작

```
mysql> START SLAVE;
```

# 동기화 상태 확인

```
mysql> SHOW SLAVE STATUS\G
```

# Slave에 동기화가 깨졌을때 마스터 데이터베이스의 쿼리 스킵

```
mysql> SET GLOBAL SQL_SLAVE_SKIP_COUNTER=1;
```

# slave 설정 초기화

```
mysql> reset slave all;
```

# 데이터 베이스의 프로세스 리스트를 확인 : master의 업데이트 대기중 확인

```
mysql> show processlist;
```

**실습 : 두개의 데이터 베이스를 서로 마스터 슬레이브로 설정**

# Slave Server의 Slave 중지

```
mysql> STOP SLAVE;
```

# Slave Server 계정 생성 ( mysql shell 에서 실행 )

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'root'@'%' IDENTIFIED BY 'rada';
```

# Slave 서버의 File, Position 확인

```
mysql> SHOW MASTER STATUS;
```

# Master 서버 Slave 설정

```
mysql> CHANGE MASTER TO  
MASTER_HOST='13.124.234.223',  
MASTER_USER='root',  
MASTER_PASSWORD='rada',  
MASTER_LOG_FILE='mysql-bin.000002',  
MASTER_LOG_POS=589;
```

# Master Server의 Slave 시작

```
mysql> START SLAVE;
```

# Slave Server의 Slave 시작

```
mysql> START SLAVE;
```