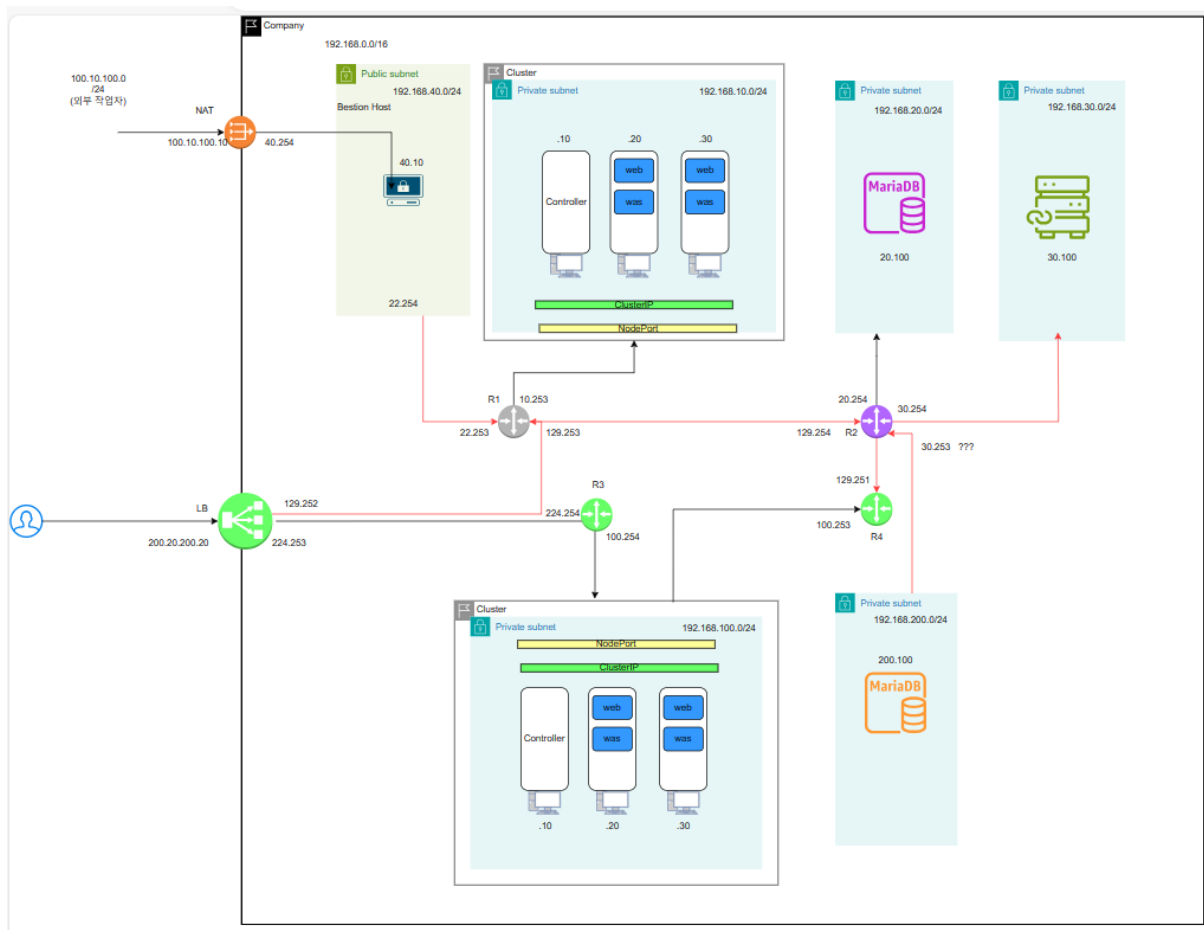
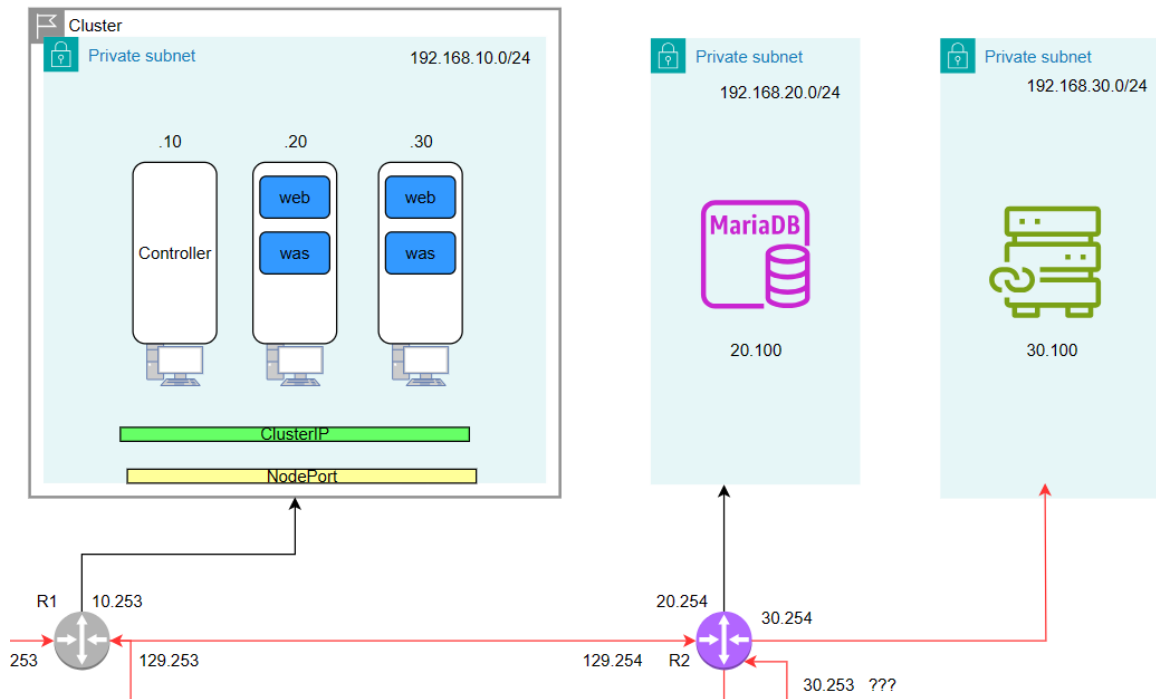




# MariaDB Replication 매뉴얼

1. MariaDB-A Replication Primary 설정
3. MariaDB-B Replication Secondary 설정
4. MariaDB-A : 덤프 파일 생성하기
5. MariaDB Secondary 데이터 백업확인
6. MariaDB-A,B: Primary에서 데이터 수정 시 Secondary에서 확인
7. R2 구성 매뉴얼
  1. Oracle VM VirtualBox 도구
  2. Oracle VM VirtualBox 머신 생성
  3. Oracle VM VirtualBox R2 머신 어댑터 설정
  4. /etc/netplan/00-installer-config.yaml 파일 설정





## 1. MariaDB-A Replication Primary 설정

- Oracle VM VirtualBox에서 MariaDB-A 머신의 네트워크 설정
  - 어댑터1 : 호스트 전용 어댑터(192.168.20.1/24)
  - 버전 이미지 : Ubuntu 22.04
  - 메모리 : 1024MB, 프로세서 : 1
- MariaDB 서버를 설치합니다.

```
sudo apt-get install mariadb-server
```

- MariaDB에 접속합니다.

```
sudo mariadb
```

- **mydb** 데이터베이스를 생성합니다.

```
CREATE DATABASE mydb;
```

- 새 사용자 'admin'을 생성하고, '192.168.0.0/16' 대역에서 접속할 수 있도록 합니다.

```
CREATE USER 'admin'@'192.168.%.%' IDENTIFIED BY 'admin';
```

- 권한 변경 사항을 적용합니다.

```
FLUSH PRIVILEGES;
```

- Write 작업을 수행하는 Primary 머신 MariaDB-A의 /etc/netplan/00-installer-config.yaml 파일에 다음의 내용을 추가합니다.

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: false
      addresses: [192.168.20.100/24]
      routes:
        - to: 192.168.22.0/24
          via: 192.168.20.254
        - to: 192.168.10.0/24
          via: 192.168.20.254
        - to: 192.168.30.0/24
          via: 192.168.20.254
        - to: 192.168.100.0/24
          via: 192.168.20.254
        - to: 192.168.200.0/24
          via: 192.168.20.254
      version: 2
~
```

- Write 작업을 수행하는 Primary 머신 MariaDB-A의 /etc/mysql/my.cnf 파일에 다음의 내용을 추가합니다.

```
sudo vi /etc/mysql/my.cnf
```

```
# 다음의 내용을 추가합니다.
[mariadb]
  log-bin
  server_id=1
  log-basename=master1
  binlog-format=mixed
```

- MariaDB를 실행합니다.

```
sudo mariadb
```

- 사용자 계정을 생성합니다.

```
CREATE USER 'replication_user2'@'%' IDENTIFIED BY 'secret';
```

- 특정 사용자에게 Replica 기능을 위한 권한을 부여합니다.

```
GRANT REPLICATION SLAVE ON *.* TO 'replication_user2'@'%';
```

- 다음의 명령어를 통해 정상적으로 사용자 계정이 생성되었는지 확인할 수 있습니다.

```
SELECT user, host from mysql.user;
```

```
MariaDB [(none)]> SELECT user, host from mysql.user;
+-----+-----+
| User          | Host          |
+-----+-----+
| replication_user2 | %             |
| test_user      | %             |
| admin          | 192.168.%.%   |
| mariadb.sys    | localhost     |
| mysql          | localhost     |
| root           | localhost     |
+-----+-----+
6 rows in set (0.001 sec)
```

- Write 작업을 수행하는 Primary 머신 MariaDB-A의 /etc/mysql/mariadb.conf.d/50-server.cnf 파일에 다음의 내용을 추가합니다.

```
sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf
```

```
# 다음의 내용을 추가합니다.
bind-address=0.0.0.0
```

### 3. MariaDB-B Replication Secondary 설정

- Oracle VM VirtualBox에서 MariaDB-B 머신의 네트워크 설정
  - 어댑터1 : 호스트 전용 어댑터(192.168.200.1/24)
  - 버전 이미지 : Ubuntu 22.04
  - 메모리 : 1024MB, 프로세서 : 1
- apt-update 하기

```
sudo apt-get update
```

- 네트워크 설정(enp0s3:nat, enp0s8: 브릿지)

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
      dhcp4: true
    enp0s8:
      dhcp4: false
      addresses: [192.168.200.100/24, 192.168.30.253/24]
      routes:
        - to: default
```

```
via: 192.168.200.254
version: 2
```

- 설정한 뒤 서로 통신이 가능한 지 확인합니다.

```
ping 192.168.20.100
```

- mariadb를 설치합니다.

```
sudo apt-get install mariadb-server
```

- mariadb가 성공적으로 설치되었는지 확인합니다.

```
sudo mariadb --version
```

- MariaDB 환경설정

```
sudo vi /etc/mysql/my.cnf
```

```
[mariadb]
log-bin
log_error=/var/log/mysql/mariadb.err
server_id=22
read_only=0
log-basename=replica1
```

```
# This group is read both by the client and the server
# use it for options that affect everything
#
[client-server]
# Port or socket location where to connect
# port = 3306
socket = /run/mysqld/mysqld.sock

# Import all .cnf files from configuration directory
!includedir /etc/mysql/conf.d/
!includedir /etc/mysql/mariadb.conf.d/

[mariadb]
log-bin
log_error=/var/log/mysql/mariadb.err
server_id=12
read_only=0
log-basename=replica1
```

(첨부화면)

- bind address를 설정합니다.

```
sudo vi /etc/mysql/mariadb.conf.d/50-server.cnf
```

- 해당 파일의 하단에 `bind-address = 0.0.0.0` 을 추가합니다. 예시는 아래와 같습니다.

```
# Broken reverse DNS slows down connections considerably and name resolution is
# safe to skip if there are no "host by domain name" access grants
#skip-name-resolve

# Instead of skip-networking the default is now to listen only on
# localhost which is more compatible and is not less secure.
bind-address            = 0.0.0.0
```

- MariaDB를 재시작합니다.

```
sudo systemctl restart mariadb
```

## 4. MariaDB-A : 덤프 파일 생성하기

- MariaDB를 실행합니다.

```
sudo mariadb
```

- 테이블을 잠궈 놓습니다.

```
FLUSH TABLES WITH READ LOCK;
```

```
MariaDB [(none)]> FLUSH TABLES WITH READ LOCK;
Query OK, 0 rows affected (0.000 sec)
```

- master status 정보를 확인합니다.

```
SHOW MASTER STATUS;
```

예시는 아래와 같습니다. 상세한 내용은 다를 수 있습니다.

```
MariaDB [(none)]> SHOW MASTER STATUS;
+-----+-----+-----+-----+
| File           | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| master1-bin.000003 |      344 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.001 sec)
```

- Primary의 데이터를 저장한 덤프 파일을 생성합니다.

```
exit
```

```
sudo mariadb-dump --all-databases > dumpfile.sql
```

- Primary의 덤프 파일을 Replica 머신으로 복사합니다.

```
scp dumpfile.sql ubuntu@192.168.200.100:~/
```

- MariaDB를 실행합니다.

```
sudo mariadb
```

- 테이블 잠금 설정을 해제합니다.

```
UNLOCK TABLES;
```

## 5. MariaDB Secondary 데이터 백업확인

- 마스터에서 보내온 덤프파일을 확인합니다.

```
ls
# 또는
ls -al
```

```
ubuntu@MariaDB-B:~$ ls -al
total 2496
drwxr-x--- 4 ubuntu ubuntu 4096 Mar 17 02:52 .
drwxr-xr-x 3 root root 4096 Mar 15 06:49 ..
-rw----- 1 ubuntu ubuntu 402 Mar 17 02:40 .bash_history
-rw-r--r-- 1 ubuntu ubuntu 220 Jan 6 2022 .bash_logout
-rw-r--r-- 1 ubuntu ubuntu 3771 Jan 6 2022 .bashrc
drwx----- 2 ubuntu ubuntu 4096 Mar 15 06:49 .cache
-rw-rw-r-- 1 ubuntu ubuntu 2520104 Mar 17 02:52 dumpfile.sql
-rw-r--r-- 1 ubuntu ubuntu 807 Jan 6 2022 .profile
drwx----- 2 ubuntu ubuntu 4096 Mar 15 06:49 .ssh
-rw-r--r-- 1 ubuntu ubuntu 0 Mar 15 06:50 .sudo_as_admin_successful
```

- `dump` 라는 이름의 파일에서 읽은 내용을 `mariadb` 명령어의 표준 입력으로 전달합니다.

```
sudo mariadb < dumpfile.sql
```

- MariaDB 접속합니다.

```
sudo mariadb
```

- DB 접속 후 옮겨진 데이터 확인합니다.

```
select user, host from mysql.user
```

- Primary - Secondary 관계를 설정합니다.

```
CHANGE MASTER TO
  MASTER_HOST='192.168.20.100',
  MASTER_USER='replication_user2',
  MASTER_PASSWORD='secret',
  MASTER_PORT=3306,
  MASTER_LOG_FILE='master1-bin.000012',
  MASTER_LOG_POS=344,
  MASTER_CONNECT_RETRY=10;
```

- 슬레이브를 시작합니다.

```
start slave;
```

- 시작 후 상태를 확인합니다.

```
show slave status \G;
```

```
MariaDB [(none)]> start slave;
Query OK, 0 rows affected (0.002 sec)

MariaDB [(none)]> show slave status \G;
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
        Master_Host: 192.168.20.100
        Master_User: replication_user2
        Master_Port: 3306
        Connect_Retry: 10
        Master_Log_File: master1-bin.000012
        Read_Master_Log_Pos: 344
        Relay_Log_File: replica1-relay-bin.000002
        Relay_Log_Pos: 557
        Relay_Master_Log_File: master1-bin.000012
        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: 344
```

- 📌 IO와 SQL 둘 다 Yes가 출력되어야 합니다.

## 6. MariaDB-A,B: Primary에서 데이터 수정 시 Secondary에서 확인

- Primary에서 데이터 수정할 수 있습니다.

```
CREATE USER 'replication_user2'@'%' identified by 'secret';
```

- 수정된 데이터를 slave에서 확인 할 수 있습니다.

```
select user, host from mysql.user;
```



```
MariaDB [(none)]> select user, host from mysql.user;
```

User	Host
replication_user2	%
test_user	%
admin	192.168.%.%
mariadb.sys	localhost
mysql	localhost
root	localhost

```
6 rows in set (0.001 sec)
```

## 7. R2 구성 매뉴얼

### 1. Oracle VM VirtualBox 도구

1. 도구 - 만들기
  - a. VirtualBox Host-Only Ethernet Adapter : 192.168.20.1/24
  - b. DHCP4 서버 : 사용 안함

### 2. Oracle VM VirtualBox 머신 생성

1. 버전 이미지 : Ubuntu 22.04
2. 메모리 : 1024MB, 프로세서 : 1

### 3. Oracle VM VirtualBox R2 머신 어댑터 설정

1. 어댑터1
  - a. 네트워크 어댑터 활성화
  - b. 다음에 연결됨 : 어댑터에 브리지
  - c. 이름 : Intel(R) Wi-Fi 6 AX201 160MHz
2. 어댑터2
  - a. 네트워크 어댑터 활성화
  - b. 다음에 연결됨 : 호스트 전용 어댑터
  - c. 이름 : VirtualBox Host-Only Ethernet Adapter : 192.168.20.1/24

### 4. /etc/netplan/00-installer-config.yaml 파일 설정

```
# This is the network config written by 'subiquity'
network:
  ethernets:
    enp0s3:
```

```

dhcp4: false
addresses: [192.168.129.254/24, 192.168.200.254/24, 192.168.30.254/2
4]
routes:
  - to: 192.168.22.0/24
    via: 192.168.129.250
  - to: 192.168.10.0/24
    via: 192.168.129.253
  - to: 192.168.30.0/24
    via: 192.168.30.100
  - to: 192.168.100.0/24
    via: 192.168.129.251
  - to: 192.168.20.0/24
    via: 192.168.20.100
enp0s8:
  dhcp4: false
  addresses: [192.168.20.254/24]
  routes:
    - to: 192.168.20.0/24
      via: 192.168.20.100
version: 2

```

- 네트워크 설정 사항 조회

- `ip route show`

```

ubuntu@R2:~$ ip route show
192.168.10.0/24 via 192.168.129.253 dev enp0s3 proto static
192.168.20.0/24 dev enp0s8 proto kernel scope link src 192.168.20.254
192.168.20.0/24 via 192.168.20.100 dev enp0s8 proto static
192.168.22.0/24 via 192.168.129.250 dev enp0s3 proto static
192.168.30.0/24 dev enp0s3 proto kernel scope link src 192.168.30.254
192.168.30.0/24 via 192.168.30.100 dev enp0s3 proto static
192.168.100.0/24 via 192.168.129.251 dev enp0s3 proto static
192.168.129.0/24 dev enp0s3 proto kernel scope link src 192.168.129.254
192.168.200.0/24 dev enp0s3 proto kernel scope link src 192.168.200.254

```

- `ip address`

```

ubuntu@R2:~$ ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
        valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:57:8e:81 brd ff:ff:ff:ff:ff:ff
    inet 192.168.129.254/24 brd 192.168.129.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.200.254/24 brd 192.168.200.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet 192.168.30.254/24 brd 192.168.30.255 scope global enp0s3
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe57:8e81/64 scope link
        valid_lft forever preferred_lft forever
3: enp0s8: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:e4:6b:02 brd ff:ff:ff:ff:ff:ff
    inet 192.168.20.254/24 brd 192.168.20.255 scope global enp0s8
        valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fee4:6b02/64 scope link
        valid_lft forever preferred_lft forever

```