



ABLESTACK Online Docs
ABLESTACK-V4.0-4.0.15

DB 구성

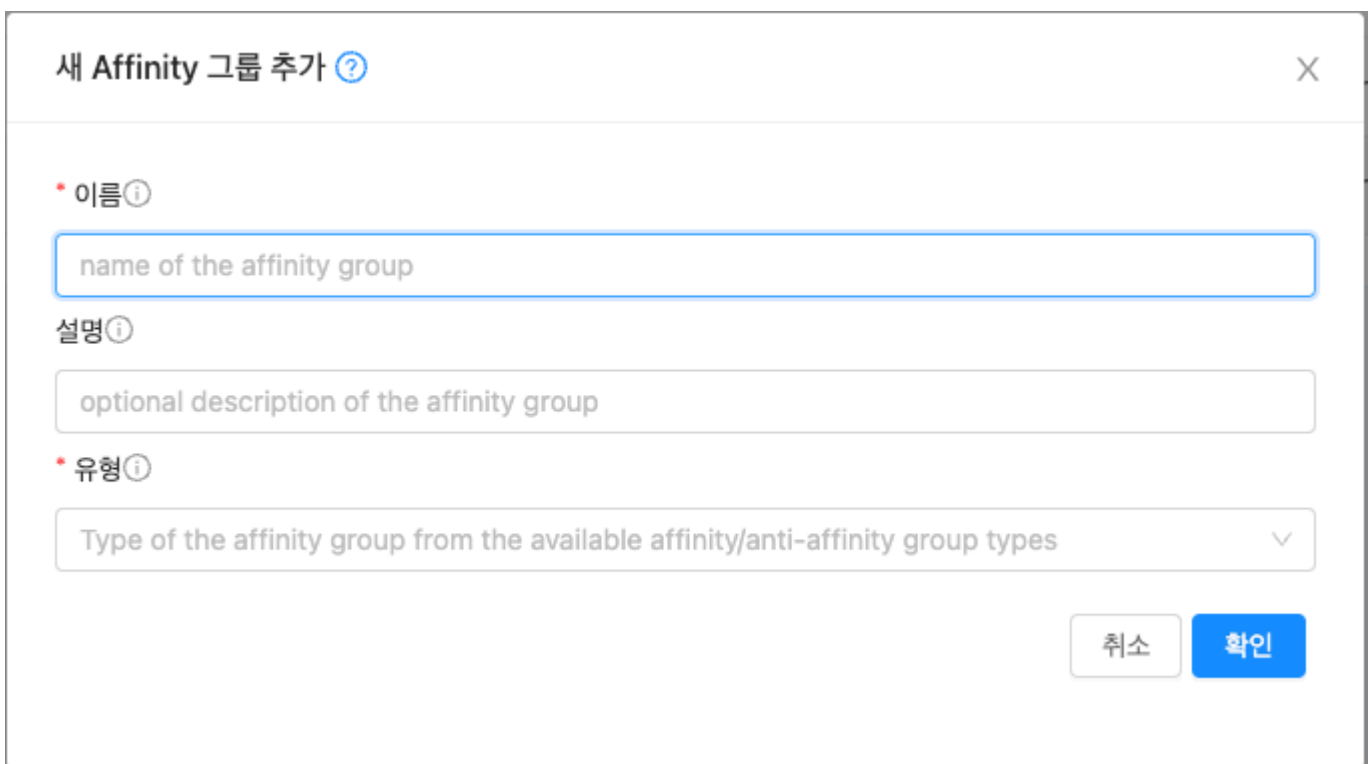
ABLESTACK Mold를 이용한 이중화를 통한 고가용성 기능을 제공하는 리눅스 기반의 3계층 구조의 구성 단계 중, 세 번째 단계인 DB 구성에 대한 문서입니다.

MariaDB를 구성하고 동기방식으로 데이터를 복제하는 갈레라 클러스터(Galera Cluster)를 활용하여 3개의 DB 가상머신을 한 클러스터로 이중화 구성하는 방법은 다음과 같은 절차로 수행됩니다.

- Affinity 그룹 생성
- 가상머신 생성
- 데이터 디스크 설정
- 보안 설정
- MariaDB 구성
- Galera Cluster 구성
- 로드 밸런서(부하 분산) 설정

Affinity 그룹 생성

가상머신을 생성하기 전, Anti Affinity 그룹을 생성하여 어느하나의 서브넷에 속한 가상머신들이 특정 호스트 한 곳에 몰려 실행하도록 하거나 반대로 몰려 실행되지 않도록 합니다. 이중화를 위해 Affinity 그룹을 anti-affinity 유형으로 WEB, WAS, DB 각 추가해야 합니다. 이를 위해 **컴퓨터 > Affinity 그룹** 화면으로 이동하여 **새 Affinity 그룹 추가** 버튼을 클릭합니다. 클릭하게 되면 다음과 같은 입력항목을 확인할 수 있습니다.



The screenshot shows a dialog box titled "새 Affinity 그룹 추가" (Add New Affinity Group). It contains three input fields: "이름" (Name) with a placeholder "name of the affinity group", "설명" (Description) with a placeholder "optional description of the affinity group", and "유형" (Type) with a dropdown menu showing "Type of the affinity group from the available affinity/anti-affinity group types". At the bottom right are "취소" (Cancel) and "확인" (OK) buttons.

- 이름 : 서브넷을 분별할 수 있는 Affinity 그룹 이름을 입력합니다.
- 설명 : Affinity 그룹에 대한 설명을 입력합니다.
- 유형 : Affinity 그룹에 대한 유형을 선택합니다. Anti 여부를 선택할 수 있습니다.

새 Affinity 그룹 추가 대화상자에서의 입력 항목 예시는 다음과 같습니다.

- 이름 : **ablecloud-3tier-linux-db**
- 설명 : **3tiers-linux**의 db 구성 시 사용되는 **Affinity** 그룹입니다.
- 유형 : **anti-affinity (Non-strict)**

i Affinity 그룹 유형

host anti-affinity: 가능한 한 서로 다른 호스트에 인스턴스를 배포합니다.

host affinity: 가능한 한 동일한 호스트에 인스턴스를 배포합니다.

- Non-Strict 옵션은 마지막으로 해당 가상머신을 실행했던 호스트를 고려하지 않고 가상머신을 시작합니다.

가상머신 생성

ABLESTACK Mold는 기본적으로 템플릿을 이용해 가상머신을 생성하고 사용하는 것을 권장합니다. 따라서 가상머신을 생성하기 전에 먼저 "**가상머신 사용 준비**" 단계를 통해 CentOS 기반의 가상머신 템플릿 이미지를 생성하여 등록하는 절차를 수행한 후 가상머신을 생성해야 합니다.

i 갈레라 클러스터 구성에 필요한 노드 개수

갈레라 클러스터가 안정적으로 동작하기 위해서는 **적어도 3개의 노드(가상머신)** 가 필요합니다. 3개 이상의 노드로 구성하여 **스플릿 브레인 (Split Brain)** 현상을 방지합니다. 이 현상은 2개의 노드로 클러스터를 구성했을 때 네트워크가 일시적으로 동시에 단절되거나 기타 시스템상의 이유로, 클러스터 상의 모든 노드들이 각자 자신이 Primary라고 인식하게 되는 상황을 뜻합니다.

스플릿 브레인 현상이 발생하면, 각 노드가 동시에 Primary가 되면서 이중 가동 현상이 발생하게 되는데 이 때 각 노드들은 동시에 스토리지에 접근하여 동기화 및 복제에 비정상적인 트랜잭션이 발생할 수 있으며, 예상하지 못한 다양한 문제로 전체 서비스가 불능 상태에 빠질 수 있습니다. 이를 방지하기 위해 3개 이상의 노드로 클러스터를 구성하는 것을 권장합니다.

가상머신을 추가하기 위해 **컴퓨트 > 가상머신** 화면으로 이동하여 **가상머신 추가** 버튼을 클릭합니다. **새 가상머신** 마법사 페이지가 표시됩니다. 해당 페이지에서는 **템플릿을 이용한 가상머신 생성** 문서를 참고하여 가상머신을 생성합니다.

i 템플릿을 이용한 가상머신 생성

템플릿을 이용한 가상머신 추가를 위해 **템플릿을 이용한 가상머신 생성** 문서를 참고하십시오.

입력 항목 예시는 다음과 같습니다.

- DB 가상머신 1
 - 배포 인프라 선택 : **Zone**
 - 템플릿/ISO : **Rocky Linux 9.0 기본 이미지 템플릿** * Rocky Linux release 9.0 (Blue Onyx)
 - 컴퓨트 오퍼링 : **1C-2GB-RBD-HA**
 - 데이터 디스크 : **100GB-WB-RBD** * MariaDB의 데이터가 저장되는 경로로 사용됩니다.
 - 네트워크 : **db** * VPC명이 일치하는지 확인합니다.

- IP: **192.168.3.11**
 - SSH 키 쌍 : **3tier_linux_keypair**
 - 확장 모드 :
 - Affinity 그룹 : **ablecloud-3tier-linux-db**
 - 이름 : **ablecloud-3tier-linux-db-01**
- DB 가상머신 2
 - 배포 인프라 선택 : **Zone**
 - 템플릿/ISO : **Rocky Linux 9.0 기본 이미지 템플릿** * Rocky Linux release 9.0 (Blue Onyx)
 - 컴퓨트 오퍼링 : **1C-2GB-RBD-HA**
 - 데이터 디스크 : **100GB-WB-RBD** * MariaDB의 데이터가 저장되는 경로로 사용됩니다.
 - 네트워크 : **db** * VPC명이 일치하는지 확인합니다.
 - IP: **192.168.3.12**
 - SSH 키 쌍 : **3tier_linux_keypair**
 - 확장 모드 :
 - Affinity 그룹 : **ablecloud-3tier-linux-db**
 - 이름 : **ablecloud-3tier-linux-db-02**
- DB 가상머신 3
 - 배포 인프라 선택 : **Zone**
 - 템플릿/ISO : **Rocky Linux 9.0 기본 이미지 템플릿** * Rocky Linux release 9.0 (Blue Onyx)
 - 컴퓨트 오퍼링 : **1C-2GB-RBD-HA**
 - 데이터 디스크 : **100GB-WB-RBD** * MariaDB의 데이터가 저장되는 경로로 사용됩니다.
 - 네트워크 : **db** * VPC명이 일치하는지 확인합니다.
 - IP: **192.168.3.13**
 - SSH 키 쌍 : **3tier_linux_keypair**
 - 확장 모드 :
 - Affinity 그룹 : **ablecloud-3tier-linux-db**
 - 이름 : **ablecloud-3tier-linux-db-03**

데이터 디스크 설정

안정적인 운영을 위해 기본 RootDisk가 아닌 고용량의 스펙을 가진 디스크로의 데이터 저장이 필요합니다. 이를위한 사전작업으로 가상머신 생성 시 추가했던 데이터 디스크를 설정합니다.

Note

DB 가상머신 1, 2, 3에 대해 실행 및 설정을 적용합니다.

`fdisk -l` 명령어를 이용하여 현재 디스크 현황과 파티션 현황을 확인합니다.

```
1 fdisk -l
```

`fdisk -l` 명령어 실행 결과 디스크 `/dev/sdb` 에 아무런 파티션이 없는 상태인 것을 확인합니다.

```
1 Disk /dev/sda: 100 GiB, 107374182400 bytes, 209715200 sectors
2 Disk model: QEMU HARDDISK
3 Units: sectors of 1 * 512 = 512 bytes
4 Sector size (logical/physical): 512 bytes / 512 bytes
5 I/O size (minimum/optimal): 512 bytes / 512 bytes
6 Disklabel type: dos
7 Disk identifier: 0xfc3d5b0c
8
9 Device Boot Start End Sectors Size Id Type
10 /dev/sda1 * 2048 2099199 2097152 1G 83 Linux
11 /dev/sda2 2099200 209715199 207616000 99G 8e Linux LVM
12
13 Disk /dev/sdb: 100 GiB, 107374182400 bytes, 209715200 sectors
14 Disk model: QEMU HARDDISK
15 Units: sectors of 1 * 512 = 512 bytes
16 Sector size (logical/physical): 512 bytes / 512 bytes
17 I/O size (minimum/optimal): 512 bytes / 512 bytes
```

`fdisk` 명령어를 이용하여 `/dev/sdb` 디스크에 파티션 설정을 합니다.

```
1 fdisk /dev/sdb
```

`n` 을 입력하여 새로운 파티션을 생성하고 `p` 를 입력하여 주 파티션으로 선택합니다.

```
1 Command (m for help): n
2 Partition type:
3 p primary (0 primary, 0 extended, 4 free)
4 e extended
5 Select (default p):
```

파티션 번호를 설정하는 단계입니다. 기본 값인 `1` 을 입력하거나 엔터로 넘어갈 수 있습니다.

```
1 Partition number (1-4, default 1):
```

시작할 섹터를 지정할 수 있습니다. 기본 값을 입력하거나 엔터로 넘어갈 수 있습니다.

```
1 First sector (2048-143305919, default 2048):
2 Using default value 2048
```

파티션의 용량을 설정합니다. 디스크 전체를 하나의 파티션으로 생성할 경우 기본 값을 입력하거나 엔터로 넘어갈 수 있습니다.

```
1 Last sector, +sectors or +size[K,M,G,T,P] (4096-262143, default
2 262143):
```

`w` 를 입력하여 파티션 정보를 디스크에 적용합니다.

```
1 Command (m for help): w
2 The partition table has been altered!
3
4 Calling ioctl() to re-read partition table.
5 Syncing disks.
```

`fdisk -l` 명령어 실행하여 변경된 파티션 정보를 확인합니다.

```
1 fdisk -l
```

생성된 파티션 `/dev/sdb1` 를 확인할 수 있습니다.

```
1 Disk /dev/sdb: 100 GiB, 107374182400 bytes, 209715200 sectors
2 Disk model: QEMU HARDDISK
3 Units: sectors of 1 * 512 = 512 bytes
4 Sector size (logical/physical): 512 bytes / 512 bytes
5 I/O size (minimum/optimal): 512 bytes / 512 bytes
6 Disklabel type: dos
7 Disk identifier: 0x1fd9e01f
8
9
10
```

| Device | Boot | Start | End | Sectors | Size | Id | Type |
|-----------|------|-------|-----------|-----------|------|----|-------|
| /dev/sdb1 | | 2048 | 209715199 | 207616000 | 100G | 83 | Linux |

`mkfs` 명령어를 이용하여 `/dev/sdb1` 파티션에 xfs 파일 시스템을 생성합니다.

```
1 mkfs.xfs /dev/sdb1
```

정상적으로 파일 시스템이 생성되었는지 `fsck -N` 명령어를 통해 확인합니다.

```
1 fsck -N /dev/sdb1
```

xfs 파일 시스템이 있는 것을 확인할 수 있습니다.

```
1 fsck from util-linux 2.37.4
2 [/usr/sbin/fsck.xfs (1) -- /dev/sdb1] fsck.xfs /dev/sdb1
```

`/dev/sdb1` 파티션을 `/mnt/data` 경로에 마운트를 적용합니다. 마운트할 경로에 폴더가 없다면 먼저 생성한 후 적절한 권한을 부여한 후 마운트를 적용합니다.

```
1 mkdir /mnt/data
2 chmod -R 1777 /mnt/data
3 mount /dev/sdb1 /mnt/data
```

정상 적으로 마운트가 적용되었는지 확인합니다.

```
1 mount | grep "sdb1"
```

`/mnt/data` 에 정상적으로 마운트 적용된 것을 확인할 수 있습니다.

```
1 /dev/sdb1 on /mnt/data type xfs
  (rw,relatime,seclabel,attr2,inode64,logbufs=8,logbsize=32k,noquota)
```

추가적으로 재부팅 시 자동으로 마운트가 적용되도록 아래 내용을 추가합니다.

```
/etc/fstab

1 /dev/sdb1 /mnt/data xfs defaults 0 0
```

i CentOS 가상머신의 볼륨 사용 방법

- CentOS 가상머신에서의 루트 디스크 확장과 데이터 디스크에 대한 추가 사용 방법에 대한 정보는 [CentOS 가상머신 가이드 - 가상머신 볼륨 사용](#) 문서를 참고하십시오.

보안 설정

생성한 가상머신에 대해 보안 설정을 하여 허용되지 않은 접근을 차단하고 필요한 서비스만 운영할 수 있도록 설정합니다.

p Note

DB 가상머신 1, 2, 3에 대해 실행 및 설정을 적용합니다.

네트워크 방화벽 해제

방화벽은 들어오고 나가는 네트워크 트래픽을 모니터링하고 필터링하는 방법입니다. 특정 트래픽을 허용할지 차단할지 결정하는 일련의 보안 규칙을 정의하여 작동합니다. CentOS 운영체제에서는 firewalld라는 이름의 방화벽 데몬과 함께 해당 기능이 제공됩니다.

`firewall-cmd` 명령어를 이용하여 galera 서비스에 대한 방화벽을 해제하고 `--permanent` 옵션을 사용하여 영구적으로 적용합니다.

```
firewall-cmd --zone=public --permanent --add-service=galera
firewall-cmd --reload
```

i Info

galera 서비스는 TCP 포트 3306, 4567, 4568, 4444와 UDP포트 4567을 사용합니다.

SELinux 설정

보안강화 리눅스(SELinux; Security Enhanced Linux)는 CentOS에서 제공하는 커널 기반의 보안 모듈입니다. 즉, 시스템 관리자가 설정한 특정 정책 및 규칙으로 사용자를 제한하는 데 사용되는 기능 또는 서비스입니다. Galera Cluster의 설정을 원활하고 효율적으로 하기 위해 SELinux 정책을 변경합니다.

SELinux 정책을 생성하여 관련 포트와 서비스를 허용하도록 변경합니다.

```
semanage port -a -t mysqld_port_t -p tcp 4567
semanage port -a -t mysqld_port_t -p tcp 4568
semanage port -a -t mysqld_port_t -p tcp 4444

semanage permissive -a mysqld_t
```

Warning

SELinux를 부적절히 허용 또는 비활성화하게 되면 권한 상승 공격에 의한 "취약점 감소"와 잘못된 설정과 버그로부터의 "시스템 보호"와 같은 이점들을 제공받지 못할 수 있습니다.

MariaDB 구성

MariaDB는 MySQL 기술을 기반으로 하는 오픈소스입니다. Galera Cluster에서 제공하는 이중화 복제 방식은 여러 MariaDB 서버를 구성하여 이루어집니다.

Note

DB 가상머신 1, 2, 3에 대해 실행 및 설정을 적용합니다.

MariaDB 패키지 설치

특정 버전의 MariaDB 패키지를 설치하기 위해 Yum Repository를 추가해야 합니다.

가상머신의 운영체제가 Rocky Linux 9.0의 경우 아래의 내용을 추가합니다.

/etc/yum.repos.d/mariadb.repo

```
1 # MariaDB 10.9 RedHat repository list - created 2022-11-30 05:38 UTC
2 # https://mariadb.org/download/
3 [mariadb]
4 name = MariaDB
5 baseurl = https://tw1.mirror.blendbyte.net/mariadb/yum/10.9/rhel9-
6 amd64
7 gpgkey=https://tw1.mirror.blendbyte.net/mariadb/yum/RPM-GPG-KEY-
  MariaDB
  gpgcheck=1
```

Tip

다른 운영체제인 경우 [MariaDB Repository Link](#) 를 클릭하여 확인한 후 적용합니다.

패키지 관리 명령어인 **dnf** 를 사용하여 MariaDB 패키지를 설치합니다.

```
1 dnf install MariaDB-server MariaDB-client
```

가상머신 부팅 시 설치된 MariaDB 서비스를 자동 시작하도록 등록하고 시작합니다.


```
1 systemctl enable mariadb.service
2 systemctl start mariadb.service
```

MariaDB 보안 설정

MariaDB 서비스를 시작한 후 아래의 명령어를 통해 MariaDB의 초기 권한 설정을 시작합니다.

```
1 mariadb-secure-installation
```

아래의 내용에 따라 초기 권한 설정을 수행합니다.

 클릭하여 MariaDB의 보안설정 방법을 확인합니다.

```
1 Enter current password for root (enter for none): [엔터를 입력합니
2 다.]
3 OK, successfully used password, moving on...
4
5 * 이 부분은 버전에 따라 표기되지 않을 수 있습니다.
6 Setting the root password or using the unix_socket ensures that
7 nobody
8 can log into the MariaDB root user without the proper
9 authorisation.
10
11 You already have your root account protected, so you can safely
12 answer 'n'.
13 Switch to unix_socket authentication [Y/n] Y [MariaDB 실행 시 통신
14 소켓 생성 여부를 선택합니다. Y를 선택합니다.]
15
16
17
18
19 Enabled successfully!
20 Reloading privilege tables..
21 ... Success!
22
23
24 You already have your root account protected, so you can safely
25 answer 'n'.
26
27
28 Change the root password? [Y/n] Y [DB ROOT 패스워드 설정할 것인지 선
29 택합니다. Y를 선택합니다.]
30
31 New password: 패스워드 입력
32 Re-enter new password: 재확인 패스워드 입력
33 Password updated successfully!
34 Reloading privilege tables..
35 ... Success!
36
37
38 By default, a MariaDB installation has an anonymous user, allowing
39 anyone
40 to log into MariaDB without having to have a user account created
41 for
42 them. This is intended only for testing, and to make the
43 installation
44 go a bit smoother. You should remove them before moving into a
45 production environment
```

외부에서 DB에 접속 가능하도록 아래의 쿼리를 통해 설정합니다.

`mariadb -u root -p` 명령어를 실행한 후 패스워드를 입력하여 MariaDB에 접속합니다.

```
mariadb -u root -p
Enter password: 패스워드 입력
```

모든 DB 및 테이블에 대한 접근권한을 설정합니다. 보안을 위해 특정 IP 대역 접근만을 허용합니다.

```
MariaDB [(none)]> use mysql;
MariaDB [(none)]> grant all privileges on *.* to 'root'@'192.168.%.%'identified by '패스워드 입력';
MariaDB [(none)]> flush privileges;
```

DB Data 폴더 경로 변경

추가한 데이터 디스크로 DB 폴더를 변경하기 위해 먼저 기존 DB data 폴더 경로를 확인합니다.

MariaDB에 접속합니다.

```
mariadb -u root -p

Enter password: 패스워드 입력
```

DB data 경로를 확인합니다.

```
MariaDB [(none)]> select @@datadir;
```

MariaDB에서 로그아웃한 후 MariaDB 서비스를 정지합니다.

```
systemctl stop mariadb
```

새로운 Data 디렉토리에 데이터를 복사합니다. 현 예시에서는 경로가 `/var/lib/mysql` 에서 `/mnt/data/mysql` 로 변경합니다.

```
rsync -av /var/lib/mysql /mnt/data/
chown -R mysql:mysql /mnt/data/mysql
```

my.cnf 파일을 수정하여 MariaDB의 data 디렉토리 경로를 변경합니다.

```
/etc/my.cnf.d/server.cnf

1          [mysqld]
2          datadir=/mnt/data/mysql
3          socket=/mnt/data/mysql/mysql.sock
4
5          [client]
6          port=3306
7          socket=/mnt/data/mysql/mysql.sock
```

SELinux 보안 설정 및 context 추가

```
semanage fcontext -a -t mysqld_db_t "/mnt/data/mysql(/.*)?"
restorecon -R /mnt/data/mysql
```

MariaDB 서비스를 시작합니다.

```
systemctl restart mariadb
```

MariaDB에 접속한 후 변경된 DB data 경로를 확인합니다.

```
MariaDB [(none)]> select @@datadir;
```

경로가 정상적으로 변경되었다면 기존 data 폴더를 삭제합니다.

```
rm -R /var/lib/mysql
```

Galera Cluster 구성


MariaDB는 기존의 Replication 방식이 아닌 Galera를 이용한 클러스터 구성으로 Master-Master 형식의 다중화 기능을 제공합니다. 이러한 동작방식으로 모든 DB 가상머신의 데이터가 일관성있게 저장되고 장애 발생 시 대응하기에 용이한 장점이 있습니다.

MariaDB에서 제공하는 Galera Cluster를 사용하기 위해 아래의 절차를 통해 설정합니다.

MariaDB 서비스를 중지합니다.

```
systemctl stop mariadb.service
```

아래 예시를 참고하여 MariaDB 설정 정보를 변경합니다.

 클릭하여 MariaDB의 설정 정보를 확인합니다.

/etc/my.cnf.d/server.cnf

```
1          [galera]
2          # galear cluster 사용여부
3          wsrep_on = ON
4          # libgalera_smm.so 모듈 위치
5          wsrep_provider = /usr/lib64/galera-4/libgalera_smm.so
6          # 동기화 진행할 IP 리스트 (Master Node 1, 2, 3의 Public IP값을 입력합니
7          다.)
8          wsrep_cluster_address = gcomm://192.168.3.11, 192.168.3.12,
9          192.168.3.13
10         # 기본 스토리지 엔진
11         default_storage_engine=InnoDB
12         # auto increment의 값 잠금방식 (0/1/2 중 택)
13         innodb_autoinc_lock_mode=2
14         # IP 접근설정
15         bind-address=0.0.0.0
16         # 바이너리 로그파일 형식
17         binlog_format = ROW
18         # log 사용여부
19         general_log=ON
20         # log 위치 설정
21         log_error=/var/lib/mysql/error.log
          # 동기화할 노드들의 그룹명 (노드마다 동일하게 설정)
          wsrep_cluster_name = Cluster
```

Galera Cluster의 모든 DB Node들이 Master DB(Primary)의 역할을 하지만 그중에서도 최초로 초기 Data를 제공하는 Node가 Donor, 그 외 Node들을 Joiner로 지정됩니다. 먼저 Donor Node로 사용할 DB 가상머신 1의 galera cluster를

`galera_new_cluster` 명령어로 시작합니다. 실행이 완료될 때까지 기다린 후, 다음 단계로 넘어갑니다.

```
galera_new_cluster
```

Warning

DB 가상머신의 시작 순서에 유의하여 아래 명령어를 실행합니다. Donor Node인 가상머신이 가장 먼저 시작되어야 합니다.

DB 가상머신 2와 3의 MariaDB 서비스를 시작합니다.

```
systemctl restart mariadb.service
```

Galera Cluster 구성이 정상적으로 되었는지 확인합니다.

MariaDB에 접속합니다.

```
mariadb -u root -p
```

```
Enter password: 패스워드 입력
```

갈레라 클러스터를 구성하는 전체 노드의 IP 주소 목록을 확인합니다.

```
MariaDB [(none)]> show variables like 'wsrep_cluster_address';
```

출력된 결과 값을 확인합니다.

```
+-----+-----+
| Variable_name      | Value                                     |
+-----+-----+
| wsrep_cluster_address | gcomm://192.168.3.11, 192.168.3.12, 192.168.3.13 |
+-----+-----+
```

하나의 DB 가상머신에서 "testdb" 라는 이름의 테스트용 데이터 베이스를 생성합니다.

```
MariaDB [(none)]> create database testdb;
```

"testdb" 데이터 베이스에 "member" 테이블을 생성합니다.

```
MariaDB [(none)]> create table testdb.member
(
  idx          int auto_increment
              primary key,
  userid       varchar(255) not null,
  password     varchar(255) not null,
  email        text          null,
  salt         varchar(255) null
);
```

i Galera Cluster 복구 절차

Galera Cluster 운영 시, 데이터 베이스에 문제가 발생하여 복구하거나 어떠한 이유로 재시작해야할 경우 아래의 절차를 따라 재기동합니다.

- Joiner 노드 (DB 가상머신 2, 3)에서 장애 발생한 경우
 - a. 장애가 발생한 노드에서 `galera_recovery` 명령을 실행합니다.
 - b. 동일 노드에서 `systemctl start mariadb.service` 명령으로 Mariadb 서비스를 다시 시작합니다.
- Donor 노드 또는 클러스터 전체에 장애 발생한 경우
 - a. `grastate.dat` 확인
 - "seqno" 값이 가장 높고 "safe_to_bootstrap" 값이 "1"인 노드가 가장 마지막에 종료된 노드이므로 이 노드를 Donor로 설정하고 복구를 진행합니다. 모든 노드의 `safe_to_bootstrap`값이 -1로 동일하면 하나의 노드를 특정하여 1로 수정합니다.

```
$ cat /mnt/data/mysql/grastate.dat

# GALERA saved state
version: 2.1
uuid:      UUID값
seqno:    -1 => 0 으로 수정
safe_to_bootstrap: 0 => 1 (Doner 노드일 경우), 0 (Joner 노드일 경우)
```

b. Galera Cluster 재시작

- 복구 기준이 되는 노드(새로운 Donor Node)에서 `galera_new_cluster` 명령으로 Galera Cluster를 재시작합니다.
- 복구 기준이 되는 Donor노드가 가장 앞에 오도록 모든 Node의 `server.cnf` 의 `wsrep_cluster_address` IP 주소 값의 순서를 변경합니다.

```
$ cat /etc/my.cnf.d/server.cnf
...
wsrep_cluster_address = gcomm://192.168.3.13(새로운 Donor Node), 192.168.3.12,
192.168.3.11
...
```

- 나머지 각 노드에서 `systemctl restart mariadb.service` 명령으로 Mariadb 서비스를 다시 시작합니다.

"grastate.dat" 파일에서의 uuid 값이 0000000 일 경우에는 `--wsrep-cluster-address` 옵션을 실행하여 노드가 현재 클러스터에 대한 연결을 닫고 새 주소에 다시 연결하도록 합니다. 사용 예시는 아래와 같습니다.

```
mysqld -u root --wsrep-cluster-address='gcomm://192.168.3.11,192.168.3.12,192.168.3.13'
```

로드 밸런서(부하 분산) 설정

Mold 사용자 또는 관리자는 서버넷에서 수신된 트래픽을 해당 서버넷 내의 여러 가상머신간에 부하 분산되도록 규칙을 만들 수 있습니다. 예를 들어 DataBase 계층에 도달한 트래픽은 해당 DB 서버넷의 다른 가상머신으로 리디렉션됩니다. 로드 밸런서를 설정하면 Health Check를 통해 장애 여부를 판단하고 노드에 이상이 발생하면 다른 정상 동작중인 노드로 트래픽을 보내주는 Fail-over가 가능합니다. 내부 로드 밸런서 규칙 생성을 위해 아래 문서를 참고합니다.

i 내부 로드 밸런서 규칙 생성

템플릿을 이용한 가상머신 추가를 위해 **내부 로드 밸런서 규칙 생성** 문서에서 **내부 LB 규칙 생성** 항목을 참고하십시오.

입력 항목 예시는 다음과 같습니다.

- 이름 : **db-lb**
- 설명 : **DataBase의 내부 로드 밸런서입니다.**
- 소스 IP 주소 : **192.168.3.26**
- 소스 포트 : **3306**
- 가상머신 포트 : **3306**
- 알고리즘 : **최소 접속**

생성된 내부 로드 밸런서 규칙을 선택한 후, **가상머신 할당** 버튼을 클릭하여 DB 가상머신을 할당합니다.

가상머신 할당 ? ×

| 이름 | 상태 | 내부 이름 | 이름 표시 | IP | 계정 | Zone 이름 | 선택 |
|---|-------|-------|-----------------------------|----|------|---------|-------------------------------------|
| ablecloud-3tier-linux-db-03 192.168.3.13 (기본) × | ● 실행중 | | ablecloud-3tier-linux-db-03 | | test | Zone1 | <input checked="" type="checkbox"/> |
| ablecloud-3tier-linux-db-02 192.168.3.12 (기본) × | ● 실행중 | | ablecloud-3tier-linux-db-02 | | test | Zone1 | <input checked="" type="checkbox"/> |
| ablecloud-3tier-linux-db-01 192.168.3.11 (기본) × | ● 실행중 | | ablecloud-3tier-linux-db-01 | | test | Zone1 | <input checked="" type="checkbox"/> |

총 3 항목 < 1 > 10 / 쪽 ∨

취소 확인

ABLESTACK Online Docs