

centos7 galera cluster설치

DBMS교체(Mysql → Mariadb)를 추진하면서 이중화 구성을 replication 에서 galera_cluster로 전환하기 위해 구축하기 위한 문서

시스템 소개

- 1. Replication 기반의 데이터 동기화 방식
 - WEB/WAS → DB#1 DDL 유입 (insert / modify / delete / ...)
 - DB#1서버는 binlog에 저장.
 - DB#2 서버는 IO Thread가 master서버로 접속해서 변경내용 확인 후 변경사항이 있으면 replication 로그에 저장
 - 1*.DB#2 서버는 SQL Thread가 replication 로그를 읽어서 변경사항 저장
- 2. Galera Cluster기반의 데이터 동기화
 - wsrep api를 통해 DB#1, DB#2 접속 및 데이터 복제하기 위한 인터페이스 생성
 - WEB/WAS → DB#1 DDL 유입 (insert / modify / delete / ...)
 - DB#1은 DB#2에도 DDL적용하라는 요청 후 모든 노드가 정상응답하면 DDL 저장
- 3. 각 데이터 동기화 방식 비교

	replication	galera
동기화 방식	비동기	동기식
단 점	<ul style="list-style-type: none">• 리플리케이션은 1개의 Thread만 수행• 데이터 불일치가 발생하는 경우 이중화 구성이 풀릴 수 있음• master binlog가 expire된 경우, 리플리케이션기반의 데이터 동기화 불가능	<ul style="list-style-type: none">• innodb만 지원• 테이블별로 PK가 존재해야 함. 없는 경우 데이터 순서의 차이가 있을수 있음
운영노드	2대(Active / Standby 권장)	3대이상 홀수 수량으로 사용 권장

1. 서비스용 통신포트

포트정보	사용용도	통신방향
tcp/3306	DB데이터 통신포트	WEB/WEB → DBMS
tcp/4567, udp/4567	DB노드별 연동상태 확인, 멀티캐스트로 복제시 해당 포트 사용 (tcp/udp)	DB1 ↔ DB2
tcp/4568	Galera 증분 데이터 전송	DB1 ↔ DB2
tcp/4444	Galera 전체 데이터 전송	DB1 ↔ DB2

Galera 용어 설명

- 1. 데이터 동기화 방식
 - WSREP; Write Set REplication - Innodb 엔진내부에서 Write Set(트랜잭션을 기록하는 데이터 그룹)
 - 1*.SST; State Snapshot Transfer - 전체 데이터 전송 (처음 클러스터 구성할 때 사용)
 - IST; Incremental State Transfer - 변경된 데이터 전송 (평상시 운영중일 때 사용)
- 2. 클러스터 멤버
 - Donor : 전체 데이터를 가진 DB
 - Joiner : Donor로부터 데이터를 제공받는 노드
- 3. 데이터 동기화 엔진
 - rsync : rsync기반으로 파일전송, 데이터 동기화 속도는 빠르나 데이터 동기화를 위해 Donor 서버에 데이터 Lock을 수행함

- mariabackup (해당방식 권장) : mariabackup 유틸리티를 사용해 SST 수행, 백업 스트리밍을 위해 socat 패키지가 설치되어 있어야 함 Donor Lock를 최소화 한 방식,
- xtrabackup (Mariadb 10.3이후로 제공암함, mariabackup 사용 권장)

시스템 구축하기

1. 설치환경

- OS : Centos 7.5
- DBMS : Mariadb 10.4.7
- Galera : 4.26-4.2.1

2. 작업순서

3. galera rpm 설치(이슈사항 : Mariadb에 포함된 galera 플러그인 사용시 OPENSLL1.0.0이 없다는 에러메시지 발생

```
#> yum install libaio socat Judy ./galera-4-26.4.2 1.rhel7.el7.centos.x86_64
```

4. /etc/my.cnf에 아래내용 적용

```
[client]
port      = 3306
socket     = /tmp/mysql.sock

[mysqld]
user = mysql
port      = 3306
socket     = /tmp/mysql.sock
skip-external-locking
key_buffer_size = 384M
max_allowed_packet = 1M
table_open_cache = 512
sort_buffer_size = 2M
read_buffer_size = 2M
read_rnd_buffer_size = 8M
myisam_sort_buffer_size = 64M
thread_cache_size = 8
query_cache_size = 32M
#thread_concurrency = 8
expire_logs_days = 30
wait_timeout = 60
#skip-name-resolve
basedir=/usr/local/mysql
datadir=/data/db_data

max_heap_table_size = 190M
max_allowed_packet = 16M
tmp_table_size = 64M
join_buffer_size = 64M
innodb_buffer_pool_size = 950M
#innodb_doublewrite = 0
innodb_flush_log_at_timeout = 3
#innodb_read_io_threads = 32
#innodb_write_io_threads = 16
#character_set_client = utf8mb4

#log-bin=mysql-bin

server-id      = 2

[galera]
wsrep_on=ON
#wsrep_provider=/usr/lib64/galera/libgalera_smm.so
wsrep_provider=/usr/lib64/galera-4/libgalera_smm.so
wsrep_cluster_address="gcomm://172.21.113.109,172.21.113.69" #클러스터 구성할 IP
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
wsrep_cluster_name=cluster
```

```
wsrep_node_address="172.21.113.109" #자기 IP가 작성되어 있으면 됨
bind-address=0.0.0.0
wsrep_sst_method=mariabackup
wsrep_sst_auth="root:root"
wsrep_provider_options="pc.bootstrap=YES"
```

5. mysql_install_db 실행

```
#> /usr/local/mysql/scripts/mysql_install_db
Installing MariaDB/MySQL system tables in '/data/db_data' ...
wsrep loader: [INFO] wsrep_load(): loading provider library 'none'
OK
```

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

Two all-privilege accounts were created.
One is root@localhost, it has no password, but you need to
be system 'root' user to connect. Use, for example, sudo mysql
The second is mysql@localhost, it has no password either, but
you need to be the system 'mysql' user to connect.
After connecting you can set the password, if you would need to be
able to connect as any of these users with a password and without sudo

See the MariaDB Knowledgebase at <http://mariadb.com/kb> or the
MySQL manual for more instructions.

You can start the MariaDB daemon with:
cd '/usr/local/mysql' ; /usr/local/mysql/bin/mysqld_safe --datadir='/data/db_data'

You can test the MariaDB daemon with mysql-test-run.pl
cd '/usr/local/mysql/mysql-test' ; perl mysql-test-run.pl

Please report any problems at <http://mariadb.org/jira>

The latest information about MariaDB is available at <http://mariadb.org/>.
You can find additional information about the MySQL part at:
<http://dev.mysql.com>
Consider joining MariaDB's strong and vibrant community:
<https://mariadb.org/get-involved/>

6. Mysql Secuinstall 구성

```
# /usr/local/mysql/bin/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] y
New password:
Re-enter new password:
Password updated successfully!
Reloading privilege tables..
... Success!
```

By default, a MariaDB installation has an anonymous user, allowing anyone to log into MariaDB without having to have a user account created for them. This is intended only for testing, and to make the installation go a bit smoother. You should remove them before moving into a production environment.

```
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.

```
Reload privilege tables now? [Y/n] y
... Success!
```

Cleaning up...

All done! If you've completed all of the above steps, your MariaDB installation should now be secure.

Thanks for using MariaDB!

7. 클러스터 구성(최초로 구성하는 서버에서만 진행하면 됨)

```
# /usr/local/mysql/support-files/mysql.server start --wsrep-new-cluster
Starting MariaDB SUCCESS!
190911 13:43:35 mysqld_safe Logging to '/data/db_data/localhost.err'.
190911 13:43:35 mysqld_safe Starting mysqld daemon with databases from /data/db_data
```

8. DB접속계정 권한 설정

```
MariaDB [(none)]> grant all privileges on *.* to root@'localhost' identified by 'root' with grant option;
```

9. 2번 서버에서 mysql 프로세스 실행

```
# /usr/local/mysql/support-files/mysql.server start
Starting MariaDB.190911 13:51:30 mysqld_safe Logging to '/data/db_data/localhost.err'.
190911 13:51:30 mysqld_safe Starting mysqld daemon with databases from /data/db_data
..... SUCCESS!
```

10. Glaracluster 재실행시 마지막으로 종료된 노드를 찾아서 해당 노드에 new-cluster를 추가하면 된다

```
$> cat /data/db_data/grastate.dat
# GALERA saved state
version: 2.1
uuid: fbb2c143-8b5e-11ec-8ea1-36a72f3e94f3
seqno: -1
#마지막에 종료된 노드가 1로 표기되어 있다.
safe_to_bootstrap: 1
```

- - bootstrap을 1로 기재된 서버에 --wsrep-new-cluster를 붙여서 실행하고, 아닌 서버는 그냥 실행시키면 된다.

Maxscale에서 서버관리

1. 클러스터 서버 멤버제외절차 (Maxscale이 설치된 서버에서 수행)

- 현재 구성된 모니터 정보확인

```
$> maxctrl list monitors
```

Monitor	State	Servers
MariaDB-Monitor	Running	server1, server2

- server2를 제외하기

```
$> maxctrl unlink monitor MariaDB-Monitor server2
OK
```

- 모니터에서 제외된 서버 확인

```
$> maxctrl list monitors
```

Monitor	State	Servers
MariaDB-Monitor	Running	server1

- 2번서버 다시 투입

```
$> maxctrl link monitor MariaDB-Monitor server2
OK
```

- 모니터에서 다시 등록된 부분 확인

```
$> maxctrl list monitors
```

Monitor	State	Servers
MariaDB-Monitor	Running	server1, server2

Reference

- <https://ongamedev.tistory.com/entry/Galera-%ED%8F%AC%ED%8A%B8-%EB%A6%AC%EC%8A%A4%ED%8A%B8->

[%EB%B0%8F-%EC%9A%A9%EC%96%B4-%EC%A0%95%EB%A6%AC](#)

- <https://mariadb.com/kb/en/getting-started-with-mariadb-galera-cluster/>

🕒Revision #5

★Created 6 June 2022 15:19:54 by artop0420

✎Updated 22 February 2023 16:28:10 by artop0420