

Hanbit eBook

Realtime 55



MariaDB 실전 활용 노하우

카카오 실무자가 알려주는 마리아DB 활용 노하우

성동찬 지음

MairaDB 실전 활용 노하우

카카오 실무자가 알려주는 마리아DB 활용 노하우

지은이_ **성동찬**

절대 깨어지지 않는 고가용성 데이터 시스템을 만드는 것이 목표인 ‘개발하는 DBA’다. KTH에서 5년간 개발&DBA 업무를 했고 ‘티몬’을 거쳐 현재는 ‘카카오’에서 DBA로 활동하고 있다.

- 블로그 : <http://gywn.net>
- 트위터 : [@gywndi](https://twitter.com/gywndi)
- 커뮤니티 : <http://cafe.naver.com/mysqlpg>

MairaDB 실전 활용 노하우 카카오 실무자가 알려주는 마리아DB 활용 노하우

초판발행 2014년 1월 29일

지은이 성동찬 / 펴낸이 김태현

펴낸곳 한빛미디어(주) / 주소 서울시 마포구 양화로 7길 83 한빛미디어(주) IT출판부

전화 02-325-5544 / 팩스 02-336-7124

등록 1999년 6월 24일 제10-1779호

ISBN 978-89-6848-688-3 15000 / 정가 11,000원

책임편집 배용석 / 기획 김병희 / 편집 안선화

디자인 표지 여동일, 내지 스튜디오 [림], 조판 김현미

영업 김형진, 김진불, 조유미 / 마케팅 박상용, 서은옥, 김옥현

이 책에 대한 의견이나 오타자 및 잘못된 내용에 대한 수정 정보는 한빛미디어(주)의 홈페이지나 아래 이메일로 알려주십시오.

한빛미디어 홈페이지 www.hanbit.co.kr / 이메일 ask@hanbit.co.kr

Published by HANBIT Media, Inc. Printed in Korea

Copyright © 2014 성동찬 & HANBIT Media, Inc.

이 책의 저작권은 성동찬과 한빛미디어(주)에 있습니다.

저작권법에 의해 보호를 받는 저작물이므로 무단 복제 및 무단 전재를 금합니다.

지금 하지 않으면 할 수 없는 일이 있습니다.

책으로 펴내고 싶은 아이디어나 원고를 메일(ebookwriter@hanbit.co.kr)로 보내주세요.

한빛미디어(주)는 여러분의 소중한 경험과 지식을 기다리고 있습니다.

저자 서문

최근 들어 MariaDB가 이슈다. 사실 MariaDB는 MySQL로부터 파생된 또 다른 오픈 소스의 한 줄기로, 기본적인 아키텍처뿐만 아니라 사용 매뉴얼 또한 MySQL과 거의 유사하다. 때문에 MySQL을 잘 활용해온 개발자라면 쉽게 접근할 수 있는 DB라고 생각한다. 필자는 지금까지 MariaDB를 어렵게 느껴온 독자들이 이에 대해 느끼는 진입 장벽을 허물어주고자 집필을 결정했다. 사실 처음 MariaDB 관련 집필을 권유받았을 때, MariaDB에 관한 어떤 내용으로 시작해야 할지 도무지 감을 잡을 수 없었다. 그만큼 MySQL과 동일한 부분이 많아서, 내용상 겹치는 부분이 있을 수밖에 없다는 생각이 들었기 때문이다.

이 책에서 다루는 내용 중 일부는 이미 MySQL에 구현된 경우도 있고, MariaDB 관련 버그의 경우 지난해 이미 픽스되어 최신 버전에서는 발생하지도 않는다. 즉, 집필할 내용 중 일부는 이미 구식이 된 것도 있다. 그러나 이는 그만큼 MariaDB뿐만 아니라 MySQL도 오픈 소스로서 빠르게 진화하고 있는 결과라고 생각한다.

필자는 단순히 매뉴얼을 해석하기보다 직접 경험하고 테스트한 결과를 공유하는 자리를 미흡하게나마 마련함으로써, 직접 경험하지 못했던 부분을 간접적으로라도 경험하려는 바람으로 원고를 정리했다. 물론 경험적인 내용에 기초했기 때문에 두서없이 구구절절 적어놓은 부분도 여기저기 눈에 띄는 등 미흡한 부분도 많겠지만, 독자들과의 넓은 아량을 믿고 그동안 축적해놓은 자료들을 재해석했다. 이 책에서는 원론적인 데이터베이스에 대해 설명하지 않았지만, MySQL을 사용해왔다면 누구나 쉽게 접근할 수 있을 것이다. 독자들이 마치 소설책을 읽듯 편안한 마음으로 이 책을 읽고 MariaDB로의 진입에 어려움이 없기를, 도움을 얻기를 바란다.

여전히 늦은 원고 때문에 매번 고생하는 한빛미디어 관계자분께는 늘 죄송스러운

마음이다. 그런 와중에도 항상 밝게 웃으며 격려해주고 용기를 잃지 말라며 기운을 북돋아 주셔서 늘 감사한다. 어느새 둘째가 태어나고 두 아이의 아빠가 되었다. 회사에서 집에 돌아오면 여전히 반겨주는 첫째 강아지 효주(gywn)와 영금영금 기어와 내 품에 안기는 막내 효창, 두 아이의 시중을 드느라 신경이 곤두선 상황에서도 늘 나를 응원해주는 멋진 아내에게 깊은 감사를 전한다.

집필을 마치며

저자 **성동찬**

한빛 eBook 리얼타임

한빛 eBook 리얼타임은 IT 개발자를 위한 eBook입니다.

요즘 IT 업계에는 하루가 멀다 하고 수많은 기술이 나타나고 사라져 갑니다. 인터넷을 아무리 뒤져도 조금이나마 정리된 정보를 찾는 것도 쉽지 않습니다. 또한 잘 정리되어 책으로 나오기까지는 오랜 시간이 걸립니다. 어떻게 하면 조금이라도 더 유용한 정보를 빠르게 얻을 수 있을까요? 어떻게 하면 남보다 조금 더 빨리 경험하고 습득한 지식을 공유하고 발전시켜 나갈 수 있을까요? 세상에는 수많은 종이책이 있습니다. 그리고 그 종이책을 그대로 옮긴 전자책도 많습니다. 전자책에는 전자책에 적합한 콘텐츠와 전자책의 특성을 살린 형식이 있다고 생각합니다.

한빛이 지금 생각하고 추구하는, 개발자를 위한 리얼타임 전자책은 이렇습니다.

1. eBook Only - 빠르게 변화하는 IT 기술에 대해 핵심적인 정보를 신속하게 제공합니다.

500페이지 가까운 분량의 잘 정리된 도서(종이책)가 아니라, 핵심적인 내용을 빠르게 전달하기 위해 조금은 거칠지만 100페이지 내외의 전자책 전용으로 개발한 서비스입니다. 독자에게는 새로운 정보를 빨리 얻을 수 있는 기회가 되고, 자신이 먼저 경험한 지식과 정보를 책으로 펴내고 싶지만 너무 바빠서 엄두를 못 내는 선배, 전문가, 고수 분에게는 보다 쉽게 집필할 수 있는 기회가 될 수 있으리라 생각합니다. 또한 새로운 정보와 지식을 빠르게 전달하기 위해 O'Reilly의 전자책 번역 서비스도 하고 있습니다.

2. 무료로 업데이트되는, 전자책 전용 서비스입니다.

종이책으로는 기술의 변화 속도를 따라잡기가 쉽지 않습니다. 책이 일정 분량 이상으로 집필되고 정리되어 나오는 동안 기술은 이미 변해 있습니다. 전자책으로 출간된 이후에도 버전 업을 통해 중요한 기술적 변화가 있거나 저자(역자)와 독자가 소통하면서 보완하여 발전된 노하우가 정리되면 구매하신 분께 무료로 업데이트해 드립니다.

3. 독자의 편의를 위하여 DRM-Free로 제공합니다.

구매한 전자책을 다양한 IT 기기에서 자유롭게 활용할 수 있도록 DRM-Free PDF 포맷으로 제공합니다. 이는 독자 여러분과 한빛이 생각하고 추구하는 전자책을 만들어 나가기 위해 독자 여러분이 언제 어디서 어떤 기기를 사용하더라도 편리하게 전자책을 볼 수 있도록 하기 위함입니다.

4. 전자책 환경을 고려한 최적의 형태와 디자인에 담고자 노력했습니다.

종이책을 그대로 옮겨 놓아 가독성이 떨어지고 읽기 힘든 전자책이 아니라, 전자책의 환경에 가능한 한 최적화하여 쾌적한 경험을 드리고자 합니다. 링크 등의 기능을 적극적으로 이용할 수 있음은 물론이고 글자 크기나 행간, 여백 등을 전자책에 가장 최적화된 형태로 새롭게 디자인하였습니다.

앞으로도 독자 여러분의 충고에 귀 기울이며 지속해서 발전시켜 나가도록 하겠습니다.

지금 보시는 전자책에 소유권한을 표시한 문구가 없거나 타인의 소유권한을 표시한 문구가 있다면 위법하게 사용하고 있을 가능성이 높습니다. 이 경우 저작권법에 의해 불이익을 받으실 수 있습니다.

다양한 기기에 사용할 수 있습니다. 또한 한빛미디어 사이트에서 구입하신 후에는 횡수에 관계없이 다운받으실 수 있습니다.

한빛미디어 전자책은 인쇄, 검색, 복사하여 붙이기가 가능합니다.

전자책은 오타자 교정이나 내용의 수정·보완이 이뤄지면 업데이트 관련 공지를 이메일로 알려드리며, 구매하신 전자책의 수정본은 무료로 내려받으실 수 있습니다.

이런 특별한 권한은 한빛미디어 사이트에서 구입하신 독자에게만 제공되며, 다른 사람에게 양도나 이전은 허락되지 않습니다.

차례

0 1	MariaDB 소개	1
	1.1 MariaDB란 무엇인가.....	1
	1.2 MariaDB 간략하게 살펴보기.....	2
0 2	MariaDB 시작하기	6
	2.1 MariaDB 설치.....	6
	2.2 MySQL에서 MariaDB로 데이터 이관.....	13
	2.3 MariaDB 사용 시 주의 사항.....	22
0 3	MariaDB의 스토리지 엔진	33
	3.1 XtraDB 스토리지 엔진.....	33
	3.2 Aria 스토리지 엔진.....	41
	3.3 FederatedX 스토리지 엔진.....	45
	3.4 TokuDB 스토리지 엔진.....	62
0 4	MariaDB에서 개선된 기능들	79
	4.1 디스크 접근 최적화.....	79
	4.2 XtraDB Buffer Pool Warm-Up.....	91
	4.3 Virtual Column.....	94

05	MariaDB의 NoSQL Plugin - HandlerSocket	105
<hr/>		
	5.1 HandlerSocket 개념과 동작 원리.....	105
	5.2 HandlerSocket vs Memcached.....	113
06	MariaDB와 함께 사용하면 좋은 유틸리티	123
<hr/>		
	6.1 XtraBackup.....	123
	6.2 pt-online-schema-change.....	142

1 | MariaDB 소개

MariaDB⁰¹라고 하면 굉장히 생소한 개념의 ‘DBMS(Database Management System)’⁰²라고 생각하기 쉽다. MySQL에 쉽게 접근하여 잘 활용하던 사람도 MariaDB 앞에서는 주춤하곤 한다. 아마도 MariaDB라는 생소한 이름 때문에 개발자들이 어렵게 느끼는 것 같다. 사실 MariaDB는 MySQL과 다르지 않다. MariaDB는 MySQL의 오픈 소스를 기반으로 만들어진 DBMS로, ‘기본 구조’와 ‘사용 방법’이 동일하기 때문이다. 1장에서는 MariaDB가 아직 생소하거나 이제 막 사용하기 시작한 독자를 위해 MariaDB란 무엇인지 간단히 소개할 것이다. MariaDB를 어느 정도 알고 있는 독자라면 2장부터 학습할 것을 권한다.

1.1 MariaDB란 무엇인가

MySQL은 오픈 소스 DBMS로, ‘단순 쿼리 처리’ 성능은 어떤 제품보다 압도적이다. 그 때문에 이미 오랜 기간 사용됐으며 그 과정에서 성능이 꾸준히 개선된, 신뢰할 만한 제품이다. 그러나 기능적인 면에서 상용 DBMS와 비교하면 떨어지는 부분이 있다.

알려진 바와 같이 MySQL은 2008년에 썬마이크로시스템즈⁰³로 합병되었고, 2010년에 썬마이크로시스템즈는 오라클^{Oracle}에 합병되었다. 그 과정에서 MySQL LAB에 소속돼 있던 많은 엔지니어가 독립해 각자의 프로젝트를 생성했는데, 그 중 MySQL의 창시자이자 오랜 기간 테크니컬 리더로 활동하던 몬티 와이드니어

01 <http://ko.wikipedia.org/wiki/MariaDB>

02 DBMS(데이터 관리 시스템)는 다수의 사용자가 데이터베이스 내의 데이터에 접근할 수 있도록 해주는 소프트웨어 도구의 집합이다. DBMS는 사용자 또는 다른 프로그램의 요구를 처리하고 적절히 응답하여 데이터를 사용할 수 있도록 해준다. [출처: 위키백과(ko.wikipedia.co.kr)]

03 <http://ko.wikipedia.org/wiki/썬마이크로시스템즈>

스 Monty Widenius가 만든 프로젝트가 바로 ‘MariaDB’이다(여담이지만 MySQL과 MariaDB는 몬티의 딸 이름을, MaxDB는 아들 이름을 따다).

그림 1-1 MariaDB와 MySQL은 같은 유전자를 가지고 있다.



앞서 언급한 대로 MariaDB는 MySQL 소스를 기반으로 만들어진 기능이 더욱 좋아진 MySQL이라 할 수 있다. MySQL이 오라클로 인수되면서 삭제된 기능, 이를테면 ‘Thread Pool’이 MariaDB에서는 여전히 제공되고 있으며 MariaDB에서 확장한 기능이 MySQL에 반영되기도 했다. 물론 MySQL의 기능이 MariaDB에 반영되기도 했다. 이처럼 MySQL과 MariaDB는 밀접한 관계를 유지하고 있다.

MariaDB는 ‘GPL v2’⁰⁴ 라이선스에 따르기 때문에 라이선스 정책에서 상당히 자유롭다. “오라클이 MySQL을 유료화하면 어떻게 하나”라고 걱정하고 있다면, MariaDB 사용을 고려해보는 것도 좋다.

그럼 이제 MariaDB에 대해서 간단하게 살펴보자.

1.2 MariaDB 간략하게 살펴보기

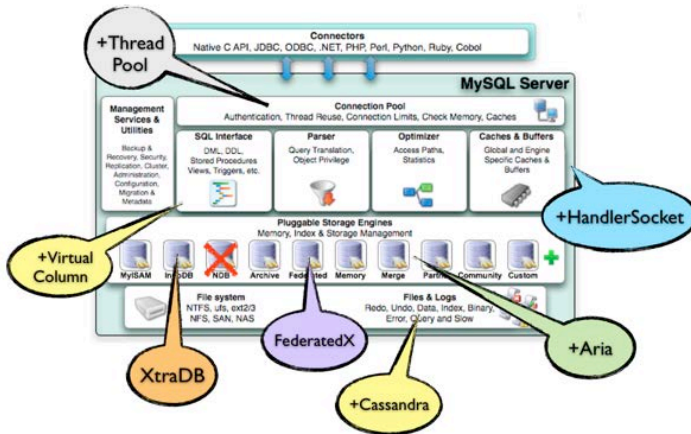
MariaDB의 전체적인 구성에 대해 별도로 그려둔 곳은 없다. 기본적인 골격뿐만 아니라 데이터를 처리하는 흐름 또한 MySQL과 완벽하게 일치하기 때문이다. 즉, MariaDB의 구조를 보고자 한다면 다른 곳에서 찾을 것이 아니라 MySQL 관련 문서⁰⁵를 참조하면 된다.

04 <http://www.gnu.org/licenses/gpl.html>

05 <http://dev.mysql.com/doc/>

물론 MariaDB가 MySQL에 비해 기능이 추가된 DBMS인만큼, 추가 요소가 있다는 것은 알아두도록 한다. 그림 1-2는 MySQL 아키텍처를 보여주고자 MariaDB 관련 기능을 추가하여 표현한 그림이다.

그림 1-2 MariaDB 구조



그림을 보고 눈치챈 사람도 있겠지만, 기능 요소가 추가 혹은 강화된 것 외에는 MySQL과 비교하여 달라진 점이 없기 때문에 클라이언트 Connector 또한 MySQL에서 사용하던 대로 사용해도 무관하다. Thread Pool, 강화된 스토리지 엔진(InnoDB → XtraDB, Federated → FederatedX), 새로운 스토리지 엔진(Aria, Cassandra), 새로운 플러그인(HandlerSocket), 새로운 기능(Virtual Column) 등이 MariaDB에 포팅되어 있다.

● **스레드 풀** Thread Pool

MySQL은 클라이언트와의 커넥션을 스레드 Thread 방식으로 관리하는데, 스레드가 많아지면 그것을 관리하는 것 자체가 부담이다. MariaDB에서는 스레드를 풀 Pool로 관리하여, 커넥션이 늘어날 때마다 스레드가 늘어나지 않도록 유지한다

(이 기능은 MySQL 5.1버전에서는 사라졌던 기능이다). 그래서 클라이언트에서 몇천 개의 커넥션을 생성하더라도 내부적으로 동작하는 스레드 수는 수십 개 정도로 제한되어, 내부 리소스를 획기적이며 효율적으로 사용할 수 있다.

● 개선된 스토리지 엔진

InnoDB는 Percona 사의 XtraDB로 대체되었다. 그리고 다른 서버에 있는 테이블을 참조하기 위한 스토리지 엔진인 Federated가 강화된 기능과 함께 FederatedX라는 이름으로 새롭게 포팅되었다.

● 추가된 스토리지 엔진

MyISAM을 대체하기 위해 몬티^{Monty} 진영에서 직접 개발한 Aria 스토리지 엔진을 비롯해 Cassandra와 직접 통신할 수 있는 Cassandra 스토리지 엔진이 추가되었다.

● 개선된 기능

InnoDB(XtraDB)와 직접 통신하여 데이터 처리 속도를 획기적으로 늘린 HandlerSocket과 가상의 칼럼을 생성할 수 있는 기능, 다양해진 옵티마이징 기능 등이 추가되었다.

위의 내용은 차근차근 살펴보도록 하고, 우선 필자가 MariaDB에 접속한 화면 내용을 보도록 하자. 많이 익숙한 내용일 텐데, 'MySQL' 문구가 'MariaDB'로 변경된 것 외에는 달라진 것이 없다.

```
Welcome to the MariaDB monitor. Commands end with ; or \g.
```

```
Your MariaDB connection id is 5547
```

```
Server version: 5.5.24-MariaDB-log MariaDB Server
```

```
Copyright (c) 2000, 2012, Oracle, Monty Program Ab and others.
```

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]>

MariaDB에 대한 기술지원은 MariaDB에 새로 추가된 기능의 매뉴얼에 한해 공식 웹사이트(<https://mariadb.org>)에서 제공하고 있다. 다만 모든 요소를 하나하나 매뉴얼을 만들어 제공하지는 않으니, 기본적인 요소는 MySQL 관련 웹사이트(<http://dev.mysql.com>)의 문서를 참고하기 바란다.

이쯤 되면 MariaDB를 특별하게만 생각하던 독자도 슬슬 MariaDB가 MySQL과 크게 다르지 않다는 것을 느꼈을 것이다. 다시 한 번 강조하지만, MariaDB는 MySQL의 소스를 기반으로 만들어지되 기능이 좋아진 DBMS이다.

자! 이제부터 MariaDB에 대해 차근차근 알아보자.

2 | MariaDB 시작하기

2.1 MariaDB 설치

MariaDB를 사용하려면 먼저 DB를 서버에 설치해야 한다. 2장에서는 MariaDB의 설치 방법, MySQL에 있는 데이터를 MariaDB로 이관하는 방법, MariaDB 사용 시 주의 사항에 대해서 알아볼 것이다. 이 방법은 리눅스Linux를 기반으로 작성했으며 미리 컴파일된 파일로 DB를 설치했다.⁰¹

2.1.1 OS 설정

다음과 같이 dba 그룹을 추가하고 그 밑에 mysql 계정을 추가한다. MariaDB는 MySQL과 동일한 소스에서 나온 DB이므로, MySQL를 설치할 때 수행했던 방법과 큰 차이가 없다.

```
$ groupadd -g 600 dba
$ useradd -g 600 -u 605 mysql
$ passwd mysql
```

계정이 추가되면 다음과 같이 리눅스에서 세션 관련 Limit 항목을 변경한다.

```
$ vi /etc/security/limits.conf
##하단 추가
mysql          soft   nproc  8192
mysql          hard   nproc  16384
```

01 MariaDB 윈도우 버전은 마우스 클릭 몇 번만으로 쉽게 설치할 수 있다. 그리고 일반적으로 리눅스에서 MariaDB를 많이 사용하기 때문에 이 책에서도 리눅스 기반으로 설명한다.

```

mysql          soft  nofile 8192
mysql          hard  nofile 65536
$ vi /etc/pam.d/login
## 하단 추가
session        required pam_limits.so

$ vi /etc/profile
## 하단 추가
if [ $USER = "mysql" ]; then
if [ $SHELL = "/bin/ksh" ]; then
ulimit -p 16384
ulimit -n 65536
else
ulimit -u 16384 -n 65536
fi
fi

```

여기까지 완료했다면 이제 MariaDB 바이너리 파일을 다운받을 차례다. 단 MariaDB 바이너리 파일을 다운받기 전에 먼저 리눅스 버전을 확인한 뒤 MariaDB 공식 사이트⁰²에 들어가서 OS에 맞는 적절한 설치 파일을 다운받기 바란다. 리눅스 버전을 확인하는 방법은 다음과 같다.

```

## OS 버전 확인 ##
uname -a
Linux ..중략.. EDT 2010 x86_64 x86_64 x86_64 GNU/Linux
## glibc 버전 확인
$ getconf -a | grep libc
GNU_LIBC_VERSION                glibc 2.12

```

02 <https://downloads.mariadb.org/mariadb/>

실행 시 'x86_64'가 있으면 64비트고, 'i686'이 있으면 32비트다. 또한 glibc 버전을 확인하여 현재 시스템에 맞는 DB를 다운받는다. 필자는 리눅스 64비트의 glibc2.12 버전이라서 다음 파일을 다운받았다.

그림 2-1 MariaDB 설치 파일 다운로드

mariadb-5.5.34-solaris11-x86_64.tar.gz	gzipped tar file	Solaris 11 64-bit	194.5 MB	MD5 Instructions
mariadb-5.5.34-solaris10-x86_64.tar.gz	gzipped tar file	Solaris 10 64-bit	228.1 MB	MD5 Instructions
mariadb-5.5.34-linux-x86_64.tar.gz (requires GLIBC_2.14+)	gzipped tar file	Linux 64-bit	285.9 MB	MD5 Instructions
mariadb-5.5.34-linux-x86_64.tar.gz	gzipped tar file	Linux 64-bit	220.4 MB	MD5 Instructions
mariadb-5.5.34-linux-i686.tar.gz (requires GLIBC_2.14+)	gzipped tar file	Linux 32-bit	268.8 MB	MD5 Instructions
mariadb-5.5.34-linux-i686.tar.gz	gzipped tar file	Linux 32-bit	212.7 MB	MD5 Instructions

이제 설치 파일을 “/usr/local”로 가져가서 압축을 해제하고 심볼릭 링크를 만든다.⁰³

```
$ mv mariadb-5.5.33a-linux-x86_64.gz.tar /usr/local
$ cd /usr/local
$ tar xzvf mariadb-5.5.33a-linux-x86_64
$ ln -s mariadb-5.5.29-linux-x86_64/ mysql
```

여기까지 제대로 했다면 개인 설정에 따라 디렉터리와 설정 파일을 구성한다. 그러면 설치가 완료된다.

03 바이너리 파일을 풀고 디렉터리 및 설정 파일만 구성해주면 MariaDB를 구동할 수 있는 상태가 된다.

2.1.2 MariaDB 기본 설정

MariaDB는 MySQL과 마찬가지로 DB가 구동되면서 “/etc/my.cnf”를 사용한다. 사용 케이스에 따라 다르게 설정할 수 있는데, 테스트 용도로는 다음과 같이 설정해도 크게 문제가 없다. 물론 시스템에 따라 조금씩 변경하여 사용해도 괜찮다.

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

[mysqld]

# generic configuration options
port = 3306
socket = /tmp/mysql.sock
back_log = 100
max_connections = 500
max_connect_errors = 10
table_open_cache = 2048
max_allowed_packet = 16M
join_buffer_size = 8M
read_buffer_size = 2M
read_rnd_buffer_size = 16M
sort_buffer_size = 8M
query_cache_type = 0
default_storage_engine = innodb
thread_stack = 256K
max_heap_table_size = 16M
tmp_table_size = 16M
log_bin_trust_function_creators = 1
secure_auth = 1
skip_external_locking
```

```
skip_symbolic_links

## config server and data path
basedir = /usr/local/mysql
datadir = /data/mysql/mysql-data
tmpdir = /data/mysql/mysql-tmp
log_bin = /data/mysql/mysql-binlog/mysql-bin
relay_log = /data/mysql/mysql-binlog/mysql-relay
innodb_data_home_dir = /data/mysql/mysql-data
innodb_log_group_home_dir = /data/mysql/mysql-iblog

## config character set
character_set_server = utf8
collation_server = utf8_general_ci

## bin log
binlog_format = row
binlog_cache_size = 4M

## Replication related settings
server_id = 1
expire_logs_days = 3
log_slave_updates

## MyISAM Specific options
key_buffer_size = 32M
myisam_sort_buffer_size = 8M
myisam_max_sort_file_size = 16M
myisam_repair_threads = 1
myisam_recover = FORCE,BACKUP

## INNODB Specific options
```

```
innodb_additional_mem_pool_size = 16M
innodb_buffer_pool_size = 2G
innodb_data_file_path = ibdata1:100M:autoextend
innodb_file_per_table = 1
innodb_purge_threads = 1
innodb_additional_mem_pool_size = 16M
innodb_buffer_pool_size = 2G
innodb_io_capacity = 500
innodb_max_dirty_pages_pct = 75
innodb_checkpoint_age_target = 1500000000
innodb_flush_log_at_trx_commit = 0
innodb_adaptive_flushing = ON
innodb_adaptive_flushing_method = estimate
innodb_flush_method = ALL_O_DIRECT
innodb_adaptive_hash_index = OFF
innodb_log_file_size = 1500M
innodb_log_files_in_group = 2
innodb_log_buffer_size = 32M
```

```
thread_handling=pool-of-threads
```

```
## Thread Pool
```

```
thread_pool_idle_timeout = 3600
thread_pool_stall_limit = 100
extra_port=3307
extra_max_connections=10
```

```
[mysqldump]
```

```
quick
```

```
max_allowed_packet = 16M
```

```
[mysql]
```

```
no_auto_rehash
```

```
[mysqld_safe]
```

```
open_files_limit = 8192
```

설정 파일에서 설정하기로 정한 디렉토리를 생성하고 mysql:dba로 권한을 변경한다. 생성한 디렉터리가 root 권한이므로 mysql:dba 계정이 디렉토리를 사용할 수 있도록 하기 위함이다.

```
mkdir -p /data/mysql/mysql-data
```

```
mkdir -p /data/mysql/mysql-tmp
```

```
mkdir -p /data/mysql/mysql-binlog
```

```
mkdir -p /data/mysql/mysql-ibdata
```

```
chown -R mysql.dba /usr/local/mysql /data/mysql /etc/my.cnf
```

다음 단계로 MariaDB에서 기본적으로 사용할 스키마를 생성한다.

```
cd /usr/local/mysql
```

```
./scripts/mysql_install_db --user=mysql
```

위 과정에서 에러가 발생하지 않았다면 DB를 구동한다. 만약 DB를 구동하는 도중 에러가 발생한다면, '/data/mysql/mysql-data/호스트명.err' 파일을 확인하여 수정하면 된다.

```
cp support-files/mysql.server /etc/init.d/mysqld
```

```
/etc/init.d/mysqld start
```

2.2 MySQL에서 MariaDB로 데이터 이관

이미 MySQL을 사용하고 있던 독자라면 사용하던 MySQL을 MariaDB로 이관하고 싶을 것이다. 필자 역시 기존에 사용하던 MySQL의 데이터를 MariaDB로 이관하여 사용했는데 데이터의 누락이나 에러 없이 잘 이관되었다.

이번 섹션에서는 MySQL의 데이터를 MariaDB로 이관하는 절차에 대해 알아볼 것이다. 여기서는 MySQL 5.5 버전을 MariaDB 5.5 버전으로 변경했다. MySQL의 데이터를 MariaDB로 이관하는 방법은 다음과 같이 두 가지다.

- mysqldump를 활용한 데이터 이관
- 데이터 파일 복사를 통한 이관

이제부터 하나씩 자세하게 살펴보자.

2.2.1 mysqldump를 활용한 데이터 이관

‘mysqldump’는 MySQL의 내부 유틸리티로, 이것을 이용하면 MySQL의 데이터를 추출하여 MariaDB로 이관할 수 있다. 먼저 다음과 같이 DB 데이터를 추출한다.

```
[root@mysql]$ mysqldump -uroot -p --single-transaction --master-data=2  
--triggers=false --routines=true --all-databases > full.sql
```

위에서 사용한 옵션을 간단하게 살펴보자.

- --single-transaction

단일 트랜잭션으로 데이터 추출을 수행하여 ‘Global Lock’⁰⁴ 없이 추출하겠다는 것을 의미하며, InnoDB와 같이 MVCC를 지원하는 스토리지 엔진에서는 시점

04 특정 시점의 데이터를 얻기 위해 다른 세션이 데이터를 변경할 수 없도록 DB 전체에 Lock을 거는 기능이다.

데이터⁰⁵를 추출할 수 있다.

- `--master-data`

사용하는 테이블 엔진(시스템 테이블 제외)이 InnoDB로만 이루어진 경우, 서버의 현재 바이너리 로그 포지션을 덤프 파일에 기록한다. 포지션은 일반적으로 덤프 파일의 23번째 줄에 기록되어 있다.

- `--triggers`

트리거가 있는 경우, Import 도중 트리거로 인해 데이터가 망가질 수 있다. 이 경우를 방지하기 위해 덤프 파일에서는 트리거를 제외한다.

- `--routines`

mysqldump 유틸리티에서 Function과 Procedure의 추출 옵션은 기본적으로 꺼져 있다. 덤프 파일에 포함되도록 옵션을 true로 한다.

- `--all-databases`

모든 데이터베이스를 추출한다. 부분적으로 데이터를 추출하려면 다음 옵션을 사용하기 바란다.

`--databases [디비명]`

`--databases [디비명] --tables [테이블1] [테이블2] ..`

트리거가 있다면 트리거 추출을 별도로 실행한다. 트리거 추출은 기본적으로 True 이기는 하지만, 데이터 이관 시에는 없는 상태에서 진행하는 것이 더욱 효율적이므로 앞선 데이터 추출에서는 제외했다(insert 시 별도의 연산을 하기 때문에 효율적이지 않으며, 타 테이블을 변경하는 트리거일 경우 잘못하면 데이터 신뢰성이 떨어질 수 있다).

05 특정 시점의 데이터를 의미한다.

```
[root@mysqlariadb]$ mysqldump -uroot -p --single-transaction --triggers=true
--no-data --no-create-info --all-databases > triggers.sql
```

위 과정이 정상적으로 완료되면 MariaDB에서 데이터를 import한다. Import하는 방법은 다음과 같다.

```
[root@mariadb]$ mysql -uroot -p < full.sql
[root@mariadb]$ mysql -uroot -p < triggers.sql
```

“Insert into ..” 구문으로 모든 데이터를 쿼리 기반으로 DB에 넣기 때문에 상당한 시간이 소요된다. 모든 과정이 완료되면 mysql_upgrade를 실행하여 버전에 따라 변경된 스키마를 업데이트한다.

```
[root@mariadb]$ /usr/local/mysql/bin/mysql_upgrade -uroot -p
```

```
Phase 1/3: Fixing table and database names
```

```
Phase 2/3: Checking and upgrading tables
```

```
Processing databases
```

```
information_schema
```

```
db1
```

```
db1.table1 OK
```

```
db1.table2 OK
```

```
db1.table3 OK
```

```
db1.table4 OK
```

```
db1.table5 OK
```

```
db1.table6 OK
```

```
db1.table7 OK
```

```
db2
```

```
db2.table1 OK
```



```
db2.table2 OK
db2.table3 OK
db2.table4 OK
db2.table5 OK
db2.table6 OK
db2.table7 OK
mysql
mysql.columns_priv OK
mysql.db OK
mysql.event OK
mysql.func OK
mysql.help_category OK
mysql.help_keyword OK
mysql.help_relation OK
mysql.help_topic OK
mysql.host OK
mysql.ndb_binlog_index OK
mysql.plugin OK
mysql.proc OK
mysql.procs_priv OK
mysql.proxies_priv OK
mysql.servers OK
mysql.tables_priv OK
mysql.time_zone OK
mysql.time_zone_leap_second OK
mysql.time_zone_name OK
mysql.time_zone_transition OK
mysql.time_zone_transition_type OK
mysql.user OK
performance_schema
Phase 3/3: Running 'mysql_fix_privilege_tables'...
OK
```

최종적으로 DB를 재시작하여 데이터 이관을 마무리한다. 이관된 데이터가 큰 문제없이 정상적으로 잘 출력된다면, MySQL에서 MariaDB로의 데이터 이관이 성공적으로 완료된 것이다.⁰⁶

```
[root@mariadb]$ /etc/init.d/mysqld restart
```

2.2.2 데이터 파일 복사를 통한 이관

앞서 살펴본 바와 같이 mysqldump를 사용하여 데이터를 이관하면 테이블이 깔끔하게 다시 생성되므로 가장 신뢰할 수 있는 방법이라고 할 수 있다. 그러나 이 방법은 기본적으로 모든 데이터를 SQL문으로 이관하기 때문에 속도면에서는 떨어진다. 데이터가 몇백 기가(GB)인 경우에는 수일이 소요될 수도 있다.

계속 언급하는 바이지만 MariaDB는 MySQL과 다르지 않다. 때문에 MySQL에서 사용하는 데이터 파일을 그대로 복사해 가져가서 사용해도 무관하다.

먼저 MySQL DB를 Shutdown하고 데이터 파일을 MariaDB로 이동한다. 이때는 MySQL 데이터 파일과 InnoDB 관련 파일이 필요하다. 파일 디렉터리 위치는 다음과 같이 확인할 수 있다.

[원본 MySQL서버]

```
mysql> show variables like 'datadir';
+-----+-----+
| Variable_name | Value           |
+-----+-----+
| datadir       | /data/mysql-data/ |
+-----+-----+
1 row in set (0.00 sec)
```

06 각 데이터가 정상적으로 보이면 이관이 완료된 것이다.

```
mysql> show variables like 'innodb_data_home_dir';
```

```
+-----+-----+
| Variable_name | Value           |
+-----+-----+
| innodb_data_home_dir | /data/mysql-data |
+-----+-----+
1 row in set (0.00 sec)
```

```
mysql> show variables like 'innodb_log_group_home_dir';
```

```
+-----+-----+
| Variable_name | Value           |
+-----+-----+
| innodb_log_group_home_dir | /data/mysql-iblog |
+-----+-----+
1 row in set (0.00 sec)
```

[신규 MariaDB]

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

```
+-----+-----+
| Variable_name | Value           |
+-----+-----+
| datadir       | /data/mysql/mysql-data/ |
+-----+-----+
1 row in set (0.00 sec)
```

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

```
+-----+-----+
| Variable_name | Value           |
+-----+-----+
| innodb_data_home_dir | /data/mysql/mysql-data |
+-----+-----+
```

```
1 row in set (0.00 sec)
```

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

```
+-----+-----+
| Variable_name      | Value                               |
+-----+-----+
| innodb_log_group_home_dir | /data/mysql/mysql-iblog |
+-----+-----+
```

```
1 row in set (0.00 sec)
```

다음의 표 2-1은 위 내용을 정리한 것이다.

표 2-1 MySQL에서 MariaDB로 이관한 파일 목록

	FROM (MySQL)	TO (MariaDB)
<code>datadir</code>	<code>/data/mysql-data/</code>	<code>/data/mysql/mysql-data/</code>
<code>innodb_data_home_dir</code>	<code>/data/mysql-data/</code>	<code>/data/mysql/mysql-data/</code>
<code>innodb_log_group_home_dir</code>	<code>/data/mysql-iblog</code>	<code>/data/mysql/mysql-iblog</code>

양쪽 DB를 모두 Shutdown하고, scp 유틸리티를 사용하여 데이터를 MariaDB에 이관한다. 양쪽 DB가 구동되지 않은 상태여야 하므로, 반드시 확인하고 이관을 진행하기 바란다. 다음은 MariaDB와 MySQL을 Shutdown한 후, MySQL에 있는 데이터를 MariaDB로 이관하는 방법이다.

[MariaDB]

```
[root@mariadb]$ /etc/init.d/mysqld stop
[root@mariadb]$ rm -rf /data/mysql/mysql-data/*
[root@mariadb]$ rm -rf /data/mysql/mysql-iblog/*
```

[MySQL]

```
[root@mysql]$ /etc/init.d/mysqld stop
[root@mysql]$ scp -r /data/mysql-data/* root@mariadb:/data/mysql/mysql-data
[root@mysql]$ scp -r /data/mysql-iblog/* root@mariadb:/data/mysql/mysql-iblog
```

바이너리 로그나 에러 로그 같은 항목은 이동하지 않아도 된다. 따라서 관련 파일들을 일시적으로 다른 디렉터리로 옮겨놓고 위 과정을 수행하면 데이터 전송 시간을 더욱 단축할 수 있다.

데이터가 MariaDB로 정상 복사되었다면 권한을 다시 설정한 후 DB를 구동한다.

```
[root@mariadb]$ chown -R mysql:dba /data/mysql
[root@mariadb]$ /etc/init.d/mysqld start
```

DB가 큰 문제없이 올라왔다면 앞서 수행했던 `mysql_upgrade`를 다시 수행하여 최종적으로 DB 이관을 마무리한다.

```
[root@mariadb]$ /usr/local/mysql/bin/mysql_upgrade -uroot -p
```

Phase 1/3: Fixing table and database names

Phase 2/3: Checking and upgrading tables

Processing databases

information_schema

db1

db1.table1 OK

db1.table2 OK

db1.table3 OK

db1.table4 OK

db1.table5	OK
db1.table6	OK
db1.table7	OK
db2	
db2.table1	OK
db2.table2	OK
db2.table3	OK
db2.table4	OK
db2.table5	OK
db2.table6	OK
db2.table7	OK
mysql	
mysql.columns_priv	OK
mysql.db	OK
mysql.event	OK
mysql.func	OK
mysql.help_category	OK
mysql.help_keyword	OK
mysql.help_relation	OK
mysql.help_topic	OK
mysql.host	OK
mysql.ndb_binlog_index	OK
mysql.plugin	OK
mysql.proc	OK
mysql.procs_priv	OK
mysql.proxies_priv	OK
mysql.servers	OK
mysql.tables_priv	OK
mysql.time_zone	OK
mysql.time_zone_leap_second	OK
mysql.time_zone_name	OK
mysql.time_zone_transition	OK
mysql.time_zone_transition_type	OK