\_고객사명\_

MySQL/MariaDB 정밀 진단 보고서

락플레이스

DB기술본부

DB 2팀

2025-04-15

1. 목차

[1. 고객사 및 기술지원 담당정보 4](#_Toc195626942)

[1.1. 고객사 정보 4](#_Toc195626943)

[1.2. 기술지원 담당정보 4](#_Toc195626944)

[2. 시스템 정보 4](#_Toc195626945)

[2.1. 시스템 구성정보 4](#_Toc195626946)

[3. 데이터베이스 환경 정보 5](#_Toc195626947)

[3.1. 데이터베이스 구성 환경 5](#_Toc195626948)

[3.2. Configure File (my.cnf) 정보 5](#_Toc195626949)

[3.3. Variables 정보 5](#_Toc195626950)

[3.4. 파라미터 적용 비교 6](#_Toc195626951)

[3.5. 메모리 사용량 통계 6](#_Toc195626952)

[3.6. Temporary space 12](#_Toc195626953)

[4. 데이터베이스 Log 분석 13](#_Toc195626954)

[4.1. Error Log 13](#_Toc195626955)

[4.2. Slow Log 13](#_Toc195626956)

[5. 스키마 기반 운영 진단 15](#_Toc195626957)

[5.1. 미사용DB계정 확인 15](#_Toc195626958)

[5.2. INDEX 점검 15](#_Toc195626959)

[5.3. 변경 없는 테이블 진단 17](#_Toc195626960)

[5.4. 성능 저하 유발 SQL 17](#_Toc195626961)

[5.5. PK/UK 누락 테이블 18](#_Toc195626962)

[6. 시스템 사용 분석 19](#_Toc195626963)

[6.1. 연결 및 트래픽 19](#_Toc195626964)

[6.2. Temp Table 사용 지표 20](#_Toc195626965)

[6.3. Read/Write 지표 20](#_Toc195626966)

[6.4. InnoDB관련 지표 22](#_Toc195626967)

[6.5. Slow Query 및 성능 24](#_Toc195626968)

[6.6. 네트워크 트래픽 25](#_Toc195626969)

[7. 점검내역 및 의견 26](#_Toc195626970)

[7.1. SQL Usage 26](#_Toc195626971)

[7.2. 파라미터 점검 26](#_Toc195626972)

[7.3. 의견 26](#_Toc195626973)

# 고객사 및 기술지원 담당정보

## 고객사 정보

|  |  |
| --- | --- |
| 고객사명 |  |
| 담당자 |  |
| 연락처 |  |
| 이메일 |  |

## 기술지원 담당정보

|  |  |  |  |
| --- | --- | --- | --- |
| **영업 담당** | | **기술 담당** | |
| 담당자 |  | 담당자 |  |
| 연락처 |  | 연락처 |  |
| 이메일 |  | 이메일 |  |

# 시스템 정보

데이터베이스에 대한 전반적인 정보를 보여줍니다.

## 시스템 구성정보

아래의 표는 운영 데이터베이스에 대한 시스템 구성 정보를 표시합니다.

|  |  |  |  |
| --- | --- | --- | --- |
| **시스템 구성정보** | | | |
| 서비스명 |  | | |
| 구성형태 |  | | |
| Hostname | {hostname} | IP Address | 보안 생략 |
| CPU | {CPU} | | |
| Memory | {Memory} | | |
| OS Version | {OS Version} | | |

# 데이터베이스 환경 정보

## 데이터베이스 구성 환경

MySQL가 설치되어 있는 기본 디렉토리 정보 및 기본 구성 환경 정보를 표시합니다.

|  |  |  |
| --- | --- | --- |
| mariaDB 구성환경 | | |
| MySQL 버전 | {version} | |
| 엔진 위치 | {basedir} | |
| 데이터 디렉토리 | {datadir} | |
| MySQL 설정파일 |  | |
| 로그 디렉토리 | Error log | {log\_error} |
| Slow log | {slow\_query\_log\_file} |
| Bin log | {log\_bin\_basename} |
| innodb data file path | {innodb\_data\_file\_path} | |
| Engine type | {default\_storage\_engine} | |
| MySQL ports | {port} | |
| Socket | {socket} | |

## Configure File (my.cnf) 정보

MariaDB에 적용되는 configfile 정보를 표시합니다.

|  |
| --- |
| /etc/my.cnf |
| {my.cnf} |

## Variables 정보

MariaDB에 적용되어 있는 변수 값을 표시합니다.

{\_table\_variables\_info}

## 파라미터 적용 비교

Config파일의 설정값이 실제 DB에 정상적으로 적용중인지 비교 검토한 내용입니다.

|  |  |  |
| --- | --- | --- |
| **Variable\_name** | **my.cnf 설정** | **Variables Value** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## 메모리 사용량 통계

#### 메모리 할당 정보

|  |  |  |  |
| --- | --- | --- | --- |
| 항목 | 점검 값 | 점검  결과 | 확인방법 |
| 총 Memory Size |  | O | # cat /etc/meminfo |
| 현재 mariaDB에서  사용하는 메모리 사용률 |  | O | # ps awxuf |grep mysqld |grep -v grep |grep -v safe |awk '{print $4}' |

#### MariaDB 메모리 사용

{graph\_mysql\_memory\_used}

#### 메모리 설정 관련 파라미터

다음 표는 메모리와 관련된 파라미터와 그 값을 보여줍니다.

|  |  |  |
| --- | --- | --- |
| 구분 | 파라미터 | 값 |
| Global memory  영역 관련 | innodb\_buffer\_pool\_size | {innodb\_buffer\_pool\_size} |
| key\_buffer\_size | {key\_buffer\_size} |
| query\_cache\_size | {query\_cache\_size} |
| innodb\_log\_buffer\_size | {innodb\_log\_buffer\_size} |
| max\_connections | {max\_connections} |
| max\_used\_connections | {max\_used\_connections} |
| Session memory  영역 관련 | read\_buffer\_size | {read\_buffer\_size} |
| read\_rnd\_buffer\_size | {read\_rnd\_buffer\_size} |
| sort\_buffer\_size | {sort\_buffer\_size} |
| join\_buffer\_size | {join\_buffer\_size} |
| binlog\_cache\_size | {binlog\_cache\_size} |
| thread\_stack | {thread\_stack} |
| tmp\_table\_size | {tmp\_table\_size} |
| **MySQL 최대 사용가능 메모리량** | |  |
| **innodb\_buffer\_pool\_size+key\_buffer\_size+query\_cache\_size+innodb\_log\_buffer\_size+max\_connections\*(read\_buffer\_size+read\_rnd\_buffer\_size+sort\_buffer\_size+join\_buffer\_size+binlog\_cache\_size+thread\_stack+tmp\_table\_size)** | |
| **Max\_used\_connections 사용 메모리량** | |  |
| **innodb\_buffer\_pool\_size+key\_buffer\_size+query\_cache\_size+innodb\_log\_buffer\_size+max\_used\_connections\*(read\_buffer\_size+read\_rnd\_buffer\_size+sort\_buffer\_size+join\_buffer\_size+binlog\_cache\_size+thread\_stack+tmp\_table\_size)** | |
| **세션당 평균 메모리 사용량** | | 77G - 72G = 5G /  **Threads\_connected** |
| **현재MySQL 메모리사용량 – (key\_buffer\_size+query\_cache\_size+ innodb\_buffer\_pool\_size +innodb\_log\_buffer\_size) = 계산 결과값 / Threads\_connected** | |

|  |  |  |  |
| --- | --- | --- | --- |
| 파라미터 | 기본값 | 설정값 | 설명 |
| key\_buffer\_size | 128M | {key\_buffer\_size} | MyISAM 엔진의 인덱스를 메모리에 캐싱하는 사이즈 |
| innodb\_buffer\_pool\_size | 128M | {innodb\_buffer\_pool\_size} | InnoDB엔진의 데이터를 메모리에 캐싱하는 사이즈 |
| innodb\_log\_buffer\_size | 96M | {innodb\_log\_buffer\_size} | 로그를 디스크에 저장하기 전 버퍼링하는 사이즈 |
| max\_connections | 151 | {max\_connections} | 최대 동시 접속 가능 수 |
| read\_buffer\_size | 128K | {read\_buffer\_size} | Full table scan 발생 시 사용하는 버퍼 사이즈 |
| read\_rnd\_buffer\_size | 256K | {read\_rnd\_buffer\_size} | order by 정렬 작업 후 정렬된 데이터를 읽을 때 사용하는 버퍼 사이즈 |
| sort\_buffer\_size | 2M | {sort\_buffer\_size} | order by 정렬에 사용하는 버퍼 사이즈 |
| join\_buffer\_size | 256K | {join\_buffer\_size} | Join에 사용되는 버퍼 사이즈 |
| binlog\_cache\_size | 32K | {binlog\_cache\_size} | Binary 로그를 디스크에 저장하긴 전 버퍼링하는 사이즈 |
| thread\_stack | 292K | {thread\_stack} | 스레드의 스택 사이즈 |
| tmp\_table\_size | 16M | {tmp\_table\_size} | 메모리에 생성되는 임시테이블의 최대 사이즈 |

|  |  |
| --- | --- |
| comment |  |

#### Buffer cache 통계.

버퍼 수에 대한 버퍼 캐시의 히트 비율을 보여줍니다.

**Innodb buffer pool hit rate**

|  |  |  |  |
| --- | --- | --- | --- |
| 항목 | 기준값 | 점검값 | 설명 |
| Innodb\_buffer\_pool\_read\_requests | N/A | {innodb\_buffer\_pool\_read\_requests} | The number of logical read requests. |
| Innodb\_buffer\_pool\_reads | N/A | {innodb\_buffer\_pool\_reads} | The number of logical reads that InnoDB could not satisfy from the buffer pool, and had to read directly from disk. |
| innodb buffer에 적재된 data  활용 적중률 | 90%  이상 |  | 100-((iReads / iReadRequests)\*100) |
| O |

☞ InnoDB buffer pool hit rate는 90% 이상을 권고.

**Innodb buffer pool hit rate**

{graph\_innodb\_buffer\_hit\_ratio}

**key buffer hit rate**

|  |  |  |  |
| --- | --- | --- | --- |
| 항목 | 기준값 | 점검값 | 설명 |
| Key\_read\_requests | N/A | {key\_read\_requests} | The number of requests to read a key block from the MyISAM key cache. |
| Key\_reads | N/A | {key\_reads} | The number of physical reads of a key block from disk into the MyISAM key cache. If Key\_reads is large, then your key\_buffer\_size value is probably too small. The cache miss rate can be calculated as Key\_reads/Key\_read\_requests. |
| MyISAM 및 특정 부분에 활용되는 data 활용 적중률 | 90%  이상 |  | 100-((key\_Reads / key\_Read\_Requests)\*100) |
| O |

*☞* Key buffer hit rate는 90% 이상을 권고.

|  |  |
| --- | --- |
| comment |  |

#### Database 사용량

데이터베이스 사이즈 정보를 기술합니다.

##### 전체 데이터베이스 사이즈

다음 표는 데이터베이스의 데이터와 인덱스 크기를 나타냅니다.

|  |  |
| --- | --- |
| **테이블 스키마(DB\_name)** | **크기 (GB)** |
| **information\_schema.tables** |  |
| mysql> select sum(data\_length+index\_length)/1024/1024/1024/1024 from information\_schema.tables;  DB Size = GB | |

##### 스키마별 테이블 사이즈

다음 표는 스키마별 테이블 사이즈 크기 리스트 입니다.

{\_table\_schema\_table}

##### 스토리지 엔진별 테이블 개수

다음 표는 스키마/엔진별 개수 리스트 입니다.

{\_table\_engine\_table}

##### 데이터, 로그, 백업 경로 disk 사이즈

아래의 그래프는 데이터, 로그, 백업 경로의 disk 사용률을 표시합니다.

|  |  |  |
| --- | --- | --- |
| **Disk** | | **사용률** |
| **Data** | **/SSD** | % (TB/TB) |
| **Log** | **/LOG** | % (TB/TB) |
| **Backup** | **/BACKUP** | % (TB/TB) |
| shell> df -h | | |

#### 세션 정보

##### Connection 정보

|  |  |  |  |
| --- | --- | --- | --- |
| 항목 | 기준값 | 점검값 | 설명 |
| Max connection | N/A | {max\_connections} | Connection 될 수 있는 최대 값 |
| Max used connections | N/A | {max\_used\_connections} | 최대 접속 됐던 클라이언트 수 |
| Aborted\_clients | N/A | {aborted\_clients} | 클라이언트 프로그램이 비 정상적으로 종료된 수 |
| Aborted\_connects | N/A | {aborted\_connects} | MySQL 서버에 접속이 실패된 수 |
| Threads\_connected | N/A | {threads\_connected} | 현재 연결된 Thread 수 |
| Threads\_running | N/A | {threads\_running} | Sleeping 되어 있지 않은 Thread 수 |
| Connection Usage (%) | 80%미만 |  | max\_used\_connection/max\_connection\*100 |

|  |  |
| --- | --- |
| comment |  |

## 

## Temporary space

mariaDB 서버는 sorting이나 grouping과 같은 처리를 위해 내부적으로 임시 테이블을 생성합니다. temp directory는 내부 임시 테이블의 데이터 파일이 저장되는 위치이며, 이 디렉토리에 생성되는 데이터 파일은 쿼리가 종료되면 자동으로 삭제됩니다.

여기서 말하는 내부 임시 테이블이란, 사용자의 의도와는 관계없이 mariaDB가 내부적으로 필요에 의해 생성하는 임시 테이블을 의미합니다.

#### 임시 데이터 파일 관련 항목

|  |  |  |  |
| --- | --- | --- | --- |
| 항목 | 기준값 | 점검값 | 설명 |
| tmpdir | N/A | {tmpdir} | 임시 파일 생성 위치 |
| Created\_tmp\_disk\_tables | N/A | {created\_tmp\_disk\_tables} | 임시 테이블 디스크 사용 수 |
| Created\_tmp\_tables | N/A | {created\_tmp\_tables} | 임시 테이블 메모리 사용 수 |
| tmp table의 disk 생성비율 | 15%  미만 |  | (created\_tmp\_disk\_tables /(created\_tmp\_disk\_tables+created\_tmp\_tables))\*100 |
| O |

#### Temp table disk rate

{graph\_tmp\_disk\_rate}

|  |  |
| --- | --- |
| Comment |  |

# 데이터베이스 Log 분석

## Error Log

데이터베이스의 error 및 warning message를 확인하여 성능에 영향을 미치는 항목에 대해 분석 합니다.

|  |  |
| --- | --- |
| 시간 | 에러 |
| 간헐적 발생 |  |
|  | |

|  |  |
| --- | --- |
| 시간 | 에러 |
| 간혈적 발생 |  |
|  | |

## Slow Log

|  |  |
| --- | --- |
| 수집기간 | 월 일 ~ 월 일 쌓인 slow query에 대한 |
| 대상 선정 방법 | 빈도수에 따른 정렬 후 TOP3 추출 |

|  |  |
| --- | --- |
| NO | query |
| 1 |  |
| 2 |  |
| 3 |  |

|  |  |
| --- | --- |
| 수집기간 | 월 일 ~ 월 일 쌓인 slow query에 대한 |
| 대상 선정 방법 | Long query time에 따른 정렬 후 TOP3 추출 |

|  |  |
| --- | --- |
| NO | query |
| 1 |  |
| 2 |  |
| 3 |  |

# 스키마 기반 운영 진단

* 아래 점검을 위해서는 performance\_schema 가 On 상태여야 합니다.

## 미사용DB계정 확인

미사용 계정을 확인하여 필요없는 계정삭제를 판단합니다.

|  |
| --- |
| **# SQL**  SELECT DISTINCT m\_u.user, m\_u.host  FROM mysql.user m\_u  LEFT JOIN performance\_schema.accounts ps\_a  ON m\_u.user = ps\_a.user AND ps\_a.host = m\_u.host  LEFT JOIN information\_schema.views is\_v  ON is\_v.definer = CONCAT(m\_u.User, '@', m\_u.Host)  AND is\_v.security\_type = 'DEFINER'  LEFT JOIN information\_schema.routines is\_r  ON is\_r.definer = CONCAT(m\_u.User, '@', m\_u.Host)  AND is\_r.security\_type = 'DEFINER'  LEFT JOIN information\_schema.events is\_e  ON is\_e.definer = CONCAT(m\_u.user, '@', m\_u.host)  LEFT JOIN information\_schema.triggers is\_t  ON is\_t.definer = CONCAT(m\_u.user, '@', m\_u.host)  WHERE ps\_a.user IS NULL  AND is\_v.definer IS NULL  AND is\_r.definer IS NULL  AND is\_e.definer IS NULL  AND is\_t.definer IS NULL  order by m\_u.user, m\_u.host; |

{\_table\_unused\_user}

## INDEX 점검

#### 미사용 인덱스

Index중에서 사용하지 않는 Index를 선별

|  |
| --- |
| # SQL  SELECT \* FROM sys.schema\_unused\_indexes; |

{\_table\_unused\_index}

#### 중복 인덱스

중복되어 생성된 인덱스 확인 후 체크하여 제거

|  |
| --- |
| # SQL  SELECT \* FROM sys. schema\_redundant\_indexes; |

{\_table\_duplicate\_index}

#### 분포도 낮은 인덱스

분포도가 낮은 인덱스 구분으로 적정 인덱스여부 판별

|  |
| --- |
| **# SQL**  SELECT  t.TABLE\_SCHEMA AS 'Db',  t.TABLE\_NAME AS 'Table',  s.INDEX\_NAME AS 'Index\_Name',  s.COLUMN\_NAME AS 'Field\_Name',  s.SEQ\_IN\_INDEX AS Seq\_In\_Index,  s2.max\_columns AS 'Max\_Cols',  s.CARDINALITY AS 'Card',  t.TABLE\_ROWS AS 'Est\_rows',  ROUND( ( (s.CARDINALITY / IFNULL(t.TABLE\_ROWS, 0.01)) \* 100), 2) AS 'Sel %'  FROM  INFORMATION\_SCHEMA.STATISTICS s  INNER JOIN INFORMATION\_SCHEMA.TABLES t  ON s.TABLE\_SCHEMA = t.TABLE\_SCHEMA  AND s.TABLE\_NAME = t.TABLE\_NAME  INNER JOIN (  SELECT  TABLE\_SCHEMA, TABLE\_NAME, INDEX\_NAME,  MAX(SEQ\_IN\_INDEX) AS max\_columns  FROM  INFORMATION\_SCHEMA.STATISTICS  WHERE TABLE\_SCHEMA not in('mysql','information\_schema','sys','performance\_schema')  GROUP BY TABLE\_SCHEMA, TABLE\_NAME, INDEX\_NAME  ) AS s2  ON s.TABLE\_SCHEMA = s2.TABLE\_SCHEMA  AND s.TABLE\_NAME = s2.TABLE\_NAME  AND s.INDEX\_NAME = s2.INDEX\_NAME  WHERE  t.TABLE\_SCHEMA != 'mysql'  AND s.CARDINALITY != 0  ORDER BY  `Sel %` ASC; |

{\_table\_low\_cardinality\_index}

## 변경 없는 테이블 진단

테이블 대상으로 쓰기, 읽기 갯수 확인. 쓰기가 발생하지 않고 데이터 또한 없는경우 -> 제거고려하여 불필요한 메모리 및 디스크 사용량 감소

|  |
| --- |
| **# SQL**  select t.table\_schema, t.table\_name, t.table\_rows, tio.count\_read, tio.count\_write  from information\_schema.tables as t  join performance\_schema.table\_io\_waits\_summary\_by\_table as tio  on tio.object\_schema = t.table\_schema and tio.object\_name = t.table\_name  where t.table\_schema not in ('mysql','performance\_schema','sys')  and tio.count\_write = 0  order by t.table\_schema, t.table\_name; |

{\_table\_no\_update\_tables}

## 성능 저하 유발 SQL

#### Full Table Scan SQL

풀테이블 스캔을 확인하여 성능에 문제가 있다면 인덱스 생성 고려

|  |
| --- |
| **# SQL**  select db, query, exec\_count,  sys.format\_time(total\_latency) as "formatted\_total\_latency",  rows\_sent\_avg, rows\_examined\_avg, last\_seen  from sys.x$statements\_with\_full\_table\_scans  order by total\_latency desc; |

rows\_sent\_avg : 클라이언트로 반환한 행 수

rows\_examined\_avg : 평균적으로 검사한 row 수. rows\_sent\_avg 값보다 지나치게 높다면 불필요한 데이터를 검사함을 의미.

last\_seen : 해당 SQL 마지막 실행시간

{\_table\_fullscan\_sql}

#### 임시테이블 사용 SQL

disk 임시테이블 사용SQL을 확인하여 성능에 문제가 있다면 튜닝 고려

|  |
| --- |
| **# SQL**  select db,query,exec\_count,total\_latency,tmp\_tables\_to\_disk\_pct  from sys.statements\_with\_temp\_tables  order by tmp\_tables\_to\_disk\_pct DESC; |

{\_table\_temptable\_sql}

Exec\_count : 쿼리가 실행된 총 횟수

Total\_latency : 총 실행 시간

Tmp\_tables\_to\_disk\_pct : 총 임시테이블 생성 개수 대비 disk 임시테이블 생성 비율

## PK/UK 누락 테이블

MySQL/MariaDB는 PK가 존재하는것이 성능상 유리

|  |
| --- |
| **# SQL**  SELECT t.TABLE\_SCHEMA, t.TABLE\_NAME  FROM information\_schema.TABLES AS t  LEFT JOIN information\_schema.KEY\_COLUMN\_USAGE AS c  ON t.TABLE\_SCHEMA = c.CONSTRAINT\_SCHEMA  AND t.TABLE\_NAME = c.TABLE\_NAME  AND c.CONSTRAINT\_NAME = 'PRIMARY'  WHERE t.TABLE\_SCHEMA NOT IN('information\_schema', 'performance\_schema', 'mysql', 'sys')  AND t.TABLE\_TYPE NOT IN('SEQUENCE', 'VIEW')  AND c.CONSTRAINT\_NAME IS NULL; |

{\_table\_unconstrained\_tables}

# 시스템 사용 분석

## 연결 및 트래픽

**Connections**

|  |
| --- |
| {graph\_connections} |
| mariaDB에 연결시도한 connection 개수 |

**Threads\_connected**

|  |
| --- |
| {graph\_threads\_connected} |
| 현재 열려 있는 연결 수 |

**Max\_used\_connections**

|  |
| --- |
| {graph\_max\_used\_connections} |
| 서버가 시작된 이후 동시에 사용 중인 최대 연결 수 |

**Threads\_created**

|  |
| --- |
| {graph\_threads\_created} |
| 연결을 처리하기 위해 생성된 스레드 수 |

**Threads\_running**

|  |
| --- |
| {graph\_threads\_running} |
| Sleep상태가 아닌 Active 상태의 스레드 수 |

**Aborted\_clients**

|  |
| --- |
| {graph\_aborted\_clients} |
| 정상적으로 close 하지 않고 클라이언트가 종료되어 중단된 연결 수 |

**Aborted\_connects**

|  |
| --- |
| {graph\_aborted\_connects} |
| 서버 연결시도 실패 횟수 |

**Queries**

|  |
| --- |
| {graph\_queries} |
| 서버에서 실행한 문 수 |

**Uptime**

|  |
| --- |
| {graph\_uptime} |
| 서버가 가동된 시간(초) |

**QPS**

|  |
| --- |
| {graph\_qps} |
| Query Per Sencond |

## Temp Table 사용 지표

**Created\_tmp\_tables**

|  |
| --- |
| {graph\_created\_tmp\_tables} |
| 쿼리를 수행 시 메모리에 생성된 임시 테이블 수 |

**Created\_tmp\_disk\_tables**

|  |
| --- |
| {graph\_created\_tmp\_disk\_tables} |
| 쿼리를 수행 시 디스크 생성된 임시 테이블 수 |

## Read/Write 지표

**Com\_select**

|  |
| --- |
| {graph\_com\_select} |
| Select구문 실행된 횟수 |

**Com\_insert**

|  |
| --- |
| {graph\_com\_insert} |
| Insert구문 실행된 횟수 |

**Com\_update**

|  |
| --- |
| {graph\_com\_update} |
| Update구문 실행된 횟수 |

**Com\_delete**

|  |
| --- |
| {graph\_com\_delete} |
| Delete 구문 실행된 횟수 |

**Com\_commit**

|  |
| --- |
| {graph\_com\_commit} |
| Commit 구문 실행된 횟수 |

**Com\_rollback**

|  |
| --- |
| {graph\_com\_rollback} |
| Rollback 구문 실행된 횟수 |

**Innodb\_buffer\_pool\_read\_requests**

|  |
| --- |
| {graph\_innodb\_buffer\_pool\_read\_requests} |
| 버퍼풀에서 읽기 요청 횟수 |

**Innodb\_buffer\_pool\_reads**

|  |
| --- |
| {graph\_innodb\_buffer\_pool\_reads} |
| Innodb가 버퍼 풀에서 처리하지 못하여 디스크에서 직접 읽기를 수행한 횟수 |

## InnoDB관련 지표

**Innodb\_buffer\_pool\_pages\_dirty**

|  |
| --- |
| {graph\_innodb\_buffer\_pool\_pages\_dirty} |
| InnoDB 버퍼 풀의 현재 더티 페이지 수 (플러시 대기 중인 페이지) |

**Innodb\_buffer\_pool\_pages\_flushed**

|  |
| --- |
| {graph\_innodb\_buffer\_pool\_pages\_flushed} |
| InnoDB 버퍼 풀에서 페이지를 플러시하는 요청 횟수 |

**Innodb\_data\_fsyncs**

|  |
| --- |
| {graph\_innodb\_data\_fsyncs} |
| InnoDB가 디스크로 데이터 flush한 횟수로 fsync() 호출 빈도는 innodb\_flush\_method 구성 옵션의 설정에 영향 |

**Innodb\_data\_pending\_fsyncs**

|  |
| --- |
| {graph\_innodb\_data\_pending\_fsyncs} |
| 보류중인 fsyncs(디스크 동기화) 수 |

**Innodb\_data\_pending\_reads**

|  |
| --- |
| {graph\_innodb\_data\_pending\_reads} |
| 현재 보류 중인 읽기 횟수 |

**Innodb\_data\_reads**

|  |
| --- |
| {graph\_innodb\_data\_reads} |
| 총 데이터 읽기 횟수(OS 파일 읽기) |

**Innodb\_data\_pending\_writes**

|  |
| --- |
| {graph\_innodb\_data\_pending\_writes} |
| 현재 보류 중인 쓰기 건수 |

**Innodb\_data\_writes**

|  |
| --- |
| {graph\_innodb\_data\_writes} |
| 총 데이터 쓰기 횟수 |

**Innodb\_os\_log\_pending\_fsyncs**

|  |
| --- |
| {graph\_innodb\_os\_log\_pending\_fsyncs} |
| InnoDB redo log 파일에 대해 보류 중인 fsync() 작업의 수 |

**Innodb\_os\_log\_fsyncs**

|  |
| --- |
| {graph\_innodb\_os\_log\_fsyncs} |
| InnoDB redo log 파일에 수행된 fsync() 쓰기 횟수 |

**Innodb\_os\_log\_pending\_writes**

|  |
| --- |
| {graph\_innodb\_os\_log\_pending\_writes} |
| InnoDB redo log 파일에 보류 중인 쓰기 횟수 |

**Innodb\_log\_waits**

|  |
| --- |
| {graph\_innodb\_log\_waits} |
| 로그 버퍼가 너무 작아서 계속하기 전에 버퍼를 플러시할 때까지 기다려야 하는 횟수 |

**Innodb\_log\_writes**

|  |
| --- |
| {graph\_innodb\_log\_writes} |
| InnoDB redo log 파일에 대한 실제 쓰기 횟수. 값이 높은경우 innodb\_log\_buffer\_size 증설 고려. |

**Innodb\_log\_write\_requests**

|  |
| --- |
| {graph\_innodb\_log\_write\_requests} |
| InnoDB redo 로그에 대한 쓰기 요청 횟수 |

**Innodb\_deadlocks**

|  |
| --- |
| {graph\_innodb\_deadlocks} |
| Dead lock 발생 횟수 |

## Slow Query 및 성능

**Slow\_queries**

|  |
| --- |
| {graph\_slow\_queries} |
| Long\_query\_time으로 지정된 시간 보다 오래 수행된 쿼리의 수 |

**Select\_full\_join**

|  |
| --- |
| {graph\_select\_full\_join} |
| 인덱스를 사용하지 않기 때문에 테이블 스캔(FTS)을 수행하는 조인 수 |

**Select\_scan**

|  |
| --- |
| {graph\_select\_scan} |
| 첫 번째 테이블의 전체 스캔(FTS)을 수행한 조인 수 |

**Sort\_scan**

|  |
| --- |
| {graph\_sort\_scan} |
| 테이블을 스캔(FTS)하여 수행한 정렬 횟수 |

**Sort\_merge\_passes**

|  |
| --- |
| {graph\_sort\_merge\_passes} |
| 쿼리에 정렬 구문 사용 시 multi merge 발생 수. 이 값이 크면 sort\_buffer\_size 시스템 변수의 값을 늘리는 것을 고려  \*multi merge : data 정렬을 위해 여러 조각으로 나누어진 data를 디스크에 쓰기, 읽기를 반복하는 것 |

## 네트워크 트래픽

**Bytes\_sent**

|  |
| --- |
| {graph\_bytes\_sent} |
| 모든 클라이언트로 보낸 바이트의 수 |

**Bytes\_received**

|  |
| --- |
| {graph\_bytes\_received} |
| 모든 클라이언트로부터 전달받은 바이트의 수 |

**Questions**

|  |
| --- |
| {graph\_questions} |
| 모든 클라이언트로부터의 요청 수 |

# 점검내역 및 의견

## SQL Usage

{graph\_query\_usages}

## 파라미터 점검

DB Variables와 status logging자료를 토대로 성능에 영향을 미치거나 리뷰가 필요한 파라미터

|  |  |  |  |
| --- | --- | --- | --- |
| **항목** | **현재 설정** | **권고값** | **comment** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 의견

|  |
| --- |
|  |