

로그 설정

장애 상황이 발생하였을 때의 로그를 확인하였지만, 상세한 정보가 없어 정확한 원인 혹은 문제가 되는 부분을 파악하기 어려웠습니다. 따라서 로그들을 기록하는 설정들을 알아보고 필수적으로 필요해보이는 설정들을 적용하려고 합니다.

log_min_messages

권장 : error

로그에 기록되는 메시지 종류를 필터링하여 원하는 유형의 메시지만 로그에 남길 수 있도록 한다.

해당 컬럼의 도메인은

DEBUG5 , DEBUG4 , DEBUG3 , DEBUG2 , DEBUG1 , INFO , NOTICE , WARNING , ERROR , LOG , FATAL , and PANIC 이다(log_min_message 에서 적혀진 순서대로 우선순위?가 존재한다). 설정한 값 이상의 로그를 남긴다.

- error 로 설정하면, warn, notice 는 기록되지 않고, error, log, fatal, panic 로그는 남긴다.

log_lock_waits

권장 : 1

디폴트 : off(bool 값 0)

교착 상태(데드락) 감지를 위해 설정

- 하나의 트랜잭션이 다른 트랜잭션을 block 할 경우에 대한 로깅 기준

설정된 기간(디폴트 1초 - deadlock timeout)보다 긴 기간 잠긴 상태로 유지되는 세션을 로깅할 수 있다. 해당 로그 발생 시 blocking 혹은 blocked session 을 중지해야한다. -> 데드락을 유발한 작업을 수행하는 프로세스를 종료하는 것이 데드락 해결 방법으로 알고 있습니다(순환 참조로 인한 발생이기 때문에 blocking 하고 있는 세션이거나 blocked 당하고 있는 세션 둘 중 하나를 종료).

log_statement

권장 : ddl 디폴트 : none

어떤 로그들을 남길 것인지 설정할 수 있는 값 네 개의 옵션을 선택할 수 있음

- ddl : 모든 ddl(create, alter, drop) 을 로깅
- mod : 모든 ddl 및 dml(insert, update, delete) 을 로깅
- all : 실행 시간에 상관 없이 모든 쿼리를 로깅
- 기본값이자 아무것도 로깅하지 않는 옵션

→ 인덱스 추가할 때, 혹은 컬럼 추가할 때 걸리는 시간 등을 측정할 수 있을 것으로 보입니다.

log_min_duration_statement

권장 : 100 혹은 1

지정된 시간(ms) 이상의 시간이 소요된 쿼리들을 로깅

- 500 으로 설정한 경우, 0.5초 이상 소요된 쿼리들을 로깅한다. slow query 에 대해 정의하는 기준이 된다.

1로 설정하는 경우는 모든 쿼리를 로깅합니다. 이 경우 너무 많은 로그가 쌓여 DB 에 부하를 줄 수 있습니다.

log_min_duration_sample 이 설정을 이용하면 duration 이상의 쿼리 중 일정 비율만 로깅할 수 있습니다.

log_autovacuum_min_duration

권장 : 1000

지정된 시간(ms) 이상 실행되는 autovacuum 및 autoanalyze 를 로깅

- 0 으로 설정하면 모든 autovacuum 및 autoanalyze 를 로깅

autovacuum 은 중요한 작업인데, 서버 리소스를 많이 잡아먹는다고 한다. [대표적인 사례](#) 입니다.

auto_explain.log_min_duration

권장 : 1000

쿼리 실행 시간이 지정한 시간(ms) 이상일 때 실행계획(explain analyze)을 로깅

포함하는 항목

- 수행한 쿼리
- 수행 시간
- 실행 계획

shared_preload_libraries

서버 시작 시 함께 실행하는 라이브러리. 몇몇 설정을 수행하기 위한 라이브러리

log_error_verbosity

권장 : verbose

로그에 오류메시지 상세 수준을 조절하는 설정

default 로 설정하면, 현재 군돌이에서 로깅하는 것과 동일한 형태입니다(단순 오류 메시지).

verbose 는 sql 문도 기록하는 것으로 보입니다.

데이터베이스 로그 양이 너무 많으면 값을 낮추는 것을 고려

- terse
 - 간단한 오류 메시지를 기록
 - 각 오류에 대해 오류 코드와 짧은 메시지만 출력
- default(기본 값)
 - 상세한 오류 메시지를 기록
 - 오류 코드, 짧은 메시지, 상세 메시지, 힌트, 컨텍스트 및 위치 정보 포함
- verbose
 - 모든 가능한 오류 정보를 기록
 - default 와 동일한 정보에 추가로 sql 명령어와 서버 내부의 오류 진단 정보 포함

pg_settings 라는 뷰를 통해 설정되어 있는 값을 확인할 수 있습니다. show 명령어를 통해 확인할 수 있습니다.

```
select * from pg_settings;  
show <parameter name>;
```

UPDATE, SET 을 통해 파라미터를 변경할 수 있습니다(몇몇 인자는 변경할 수 없습니다).

<https://jojoldu.tistory.com/708>

위의 블로그 내용을 바탕으로 필요한 것들을 정리하였습니다.

dev 적용 과정

추가적인 방법 없이(단순 유사 sql 문 으로 콘솔에서 변경 가능)

log_lock_waits - 단순 변경 가능

→ permission deny 로 prod 에 적용 안됨.

```
select * from pg_settings where name = 'log_lock_waits';  
update pg_settings set setting = 1 where name = 'log_lock_waits';  
show log_lock_waits;
```

log_statement - 단순 변경 가능

→ prod 에 적용 완료

```
select * from pg_settings where name = 'log_statement';
update pg_settings set setting = 'ddl' where name = 'log_statement';
```

auto_explain.log_min_duration - 단순 변경 가능

→ permission deny 로 prod 적용 안됨

```
update pg_settings set setting = 'auto_explain.log_min_duration' where name =
'auto_explain.log_min_duration';
select * from pg_settings where name = 'auto_explain.log_min_duration';
```

supabase 로그 - pg_sleep(2) 를 실행하여 2초 동안 실행하는 쿼리를 발생시켰고, 이를 explain analyze 한 결과를 로깅한다.

```
{
  "event_message": "duration: 2004.056 ms  plan:\\nQuery Text: select *
from pg_sleep(2)\\nFunction Scan on pg_sleep  (cost=0.00..0.01 rows=1
width=4)",
  "id": "485517ab-7239-462b-b67e-f58065cd6b94",
  "metadata": [
    {
      "file": null,
      "host": "db-iocvuerlawgfmldrzbbee",
      "metadata": [],
      "parsed": [
        {
          "application_name": "DBeaver 24.3.5 - SQLEditor <Script-1.sql>",
          "backend_type": "client backend",
          "command_tag": "SELECT",
          "connection_from":
"2406:da12:b78:de01:63a9:fb7a:ff99:6596:44588",
          "context": null,
          "database_name": "postgres",
          "detail": null,
          "error_severity": "LOG",
          "hint": null,
          "internal_query": null,
          "internal_query_pos": null,
          "leader_pid": null,
          "location": null,
          "process_id": 1254727,
          "query": null,
          "query_id": 0,
          "query_pos": null,
          "session_id": "67e3bac3.132547",
          "session_line_num": 5,
          "session_start_time": "2025-03-26 08:28:51 UTC",
```

```

        "sql_state_code": "00000",
        "timestamp": "2025-03-26 08:30:00.201 UTC",
        "transaction_id": 0,
        "user_name": "postgres",
        "virtual_transaction_id": "13/47383"
    }
],
    "parsed_from": null,
    "project": null,
    "source_type": null
}
],
    "timestamp": 1742977800201000
}

```

log_autovacuum_min_duration

```

select * from pg_settings where name = 'log_autovacuum_min_duration';
update pg_settings set setting = 1000 where name =
'log_autovacuum_min_duration';

```

log_min_duration_statement

```

select * from pg_settings where name = 'log_min_duration_statement';
update pg_settings set setting = 500 where name =
'log_min_duration_statement';

```

log_error_verbosity

```

select * from pg_settings where name = 'log_error_verbosity';
alter role "postgres" set "log_error_verbosity" to 'verbose';
update pg_settings set setting = 'verbose' where name =
'log_error_verbosity';

```