

2024-04-03 PostgreSQL 슬로우 쿼리 찾기



현행 문서: [PostgreSQL 슬로우 쿼리 찾기](#)

1. Goal
2. Reader
3. Prerequisites
 - 3.1 pg_stat_statements 권한
 - 3.2 Log Parameters
 - 3.3 APM 권한
4. SQL Query조회
 - 4.1 평균 실행 시간 Top N
 - 4.2 Disk I/O 기준 Top N
 - 4.3 Connection 수
5. Log 쿼리 보기
6. APM
 - 6.1 Performance Insight
 - 6.2 Datadog
7. Reference (참조 문헌)

1. Goal

DBMS에서 실행되는 SQL중 느리거나 I/O량이 큰 대상을 추출하여 개선하기 위함이다.

- OLTP : 0.5초 이내를 목표로 한다.
- 배치 쿼리 : 업무 수행에 영향이 없고 DBMS 부하를 분산한다.

2. Reader

- 개발자, 설계자, 운영자

3. Prerequisites

3.1 pg_stat_statements 권한

- 3개 DEV 권한 신청 완료 ex) GRANT pg_read_all_stats TO winloadsdevdb_mgr;
- pg_stat_statements를 활성화하려면 기존의 사용자 지정 파라미터 그룹을 수정
 - pg_stat_statements를 shared_preload_libraries에 추가
 - track_activity_query_size = 4096
 - pg_stat_statements.track = ALL
 - pg_stat_statements.max = 10000
- 파라미터 조회 쿼리

```
select name, setting, boot_val, reset_val, unit
from pg_settings
where name in ('track_activity_query_size', 'pg_stat_statements.track', 'pg_stat_statements.max')
order by name;
```

3.2 Log Parameters

- Slow Query를 기록하기 위해 log_min_duration_statement과 log_statement 설정을 해야 하며 로그를 보기 위해 RDS 콘솔 접근이 필요 하다.
- log_min_duration_statement을 수정하여 설정된 임계값 보다 시간이 오래 걸리는 쿼리를 수집한다. default -1 수집 안함
 - ex) 2000 : 2초이상 쿼리 기록
- log_statement을 수정하여 기록되는 SQL 문을 제어한다. 기본값은 none이다.
 - ddl : CREATE, ALTER 및 DROP과 같은 모든 데이터 정의 언어(DDL) 문을 기록한다.
 - mod : INSERT, UPDATE 및 DELETE와 같은 모든 DDL 및 데이터 수정 언어(DML) 문을 기록한다.
 - all : 은 실행 시간과 관계없이 모든 쿼리를 기록한다.

3.3 APM 권한

- Datadog DBM 은 일부 서버만 활성화 되어 있다.

- 추가가 필요하면 별도 요청이 필요하며 라이선스 제약으로 APM 기능을 사용할 것을 권장 받았다.

4. SQL Query조회

슬로우 쿼리를 잡아 내는 방법은

1. 쿼리 실행 통계 보기가 있다.
 - a. pg_stat_statements VIEW를 조회하면, 쿼리들의 실행 통계를 볼 수 있다
2. 쿼리 실행 계획 로그를 남기는 방식
 - a. 실행계획을 로그에 남기면, 당시의 쿼리 실행 계획을 볼 수 있다는 장점이 있다.
 - b. EXPLAIN ANALYZE 명령문을 기반으로 로그를 남기기 때문에, 롱쿼리를 다시 실행시킨단 리스크가 있다.
3. 슬로우 쿼리가 발생하면 로그를 남기는 방식
 - a. 부하를 유발하는 단일 쿼리를 파악하기는 쉽다.
 - b. 처리시간은 빠르지만 여러번 호출되어 부하를 발생 시키는 쿼리 인지 파악하기 어렵다.

DB 설정 변경을 최소화 하기 위해 우선 통계 정보를 활용한 분석을 한다.

4.1 평균 실행 시간 Top N

- 예시 쿼리

```
SELECT total_plan_time+total_exec_time as total_time, query
FROM pg_stat_statements
ORDER BY 1 DESC LIMIT 10;
// order by      ,      ,      .
```

4.2 Disk I/O 기준 Top N

- 예시 쿼리

```
SELECT query, calls, total_plan_time+total_exec_time as total_time, rows, 100.0 * shared_blks_hit /
nullif(shared_blks_hit +
shared_blks_read, 0) AS hit_percent
FROM pg_stat_statements ORDER BY 5 ASC
LIMIT 10;
```

4.3 Connection 수

- application 별

```
SELECT application_name,count(*) FROM pg_stat_activity GROUP BY application_name;
```

- 사용자별

```
SELECT username,count(*) FROM pg_stat_activity GROUP BY username;
```

5. Log 쿼리 보기

1. RDS 콘솔 Open
2. **Databases**(데이터베이스)를 선택
3. DB 인스턴스를 선택한 다음, **Logs**(로그) 보기를 선택
4. **Logs & Events**(로그 및 이벤트) 탭에서 최신 로그를 선택한 다음, **View log**(로그 보기)를 선택하여 로그 콘텐츠를 확인한다.

- 예시

```
2023-10-19 11:05:32 UTC:172.31.xx.yyy(35708):user@awspostgres:[27174]:LOG: statement: CREATE TABLE
article(article_code bigint, created_at timestamp with time zone, summary text, content text) ;

2023-10-19 UTC:172.31.xx.yyy(35708):user@awspostgres:[27174]:LOG: duration: 2010.247 ms statement:
select pg_sleep(2);

2023-10-19 11:11:25 UTC:172.31.xx.yyy(35708):user@awspostgres:[27174]:LOG: duration: 2159.838 ms
statement: SELECT generate_series(1,10000000) as test;
```

- **참고:** 로그 파라미터를 광범위한 로깅을 생성하는 값으로 설정하지 않는다. 예를 들어, **log_statement**를 **all**로 설정하거나 **log_min_duration_statement**를 **0**으로 설정하면 많은 양의 로깅 정보가 생성된다. 이는 사용자의 스토리지 사용량에 영향을 미친다.

6. APM

6.1 Performance Insight

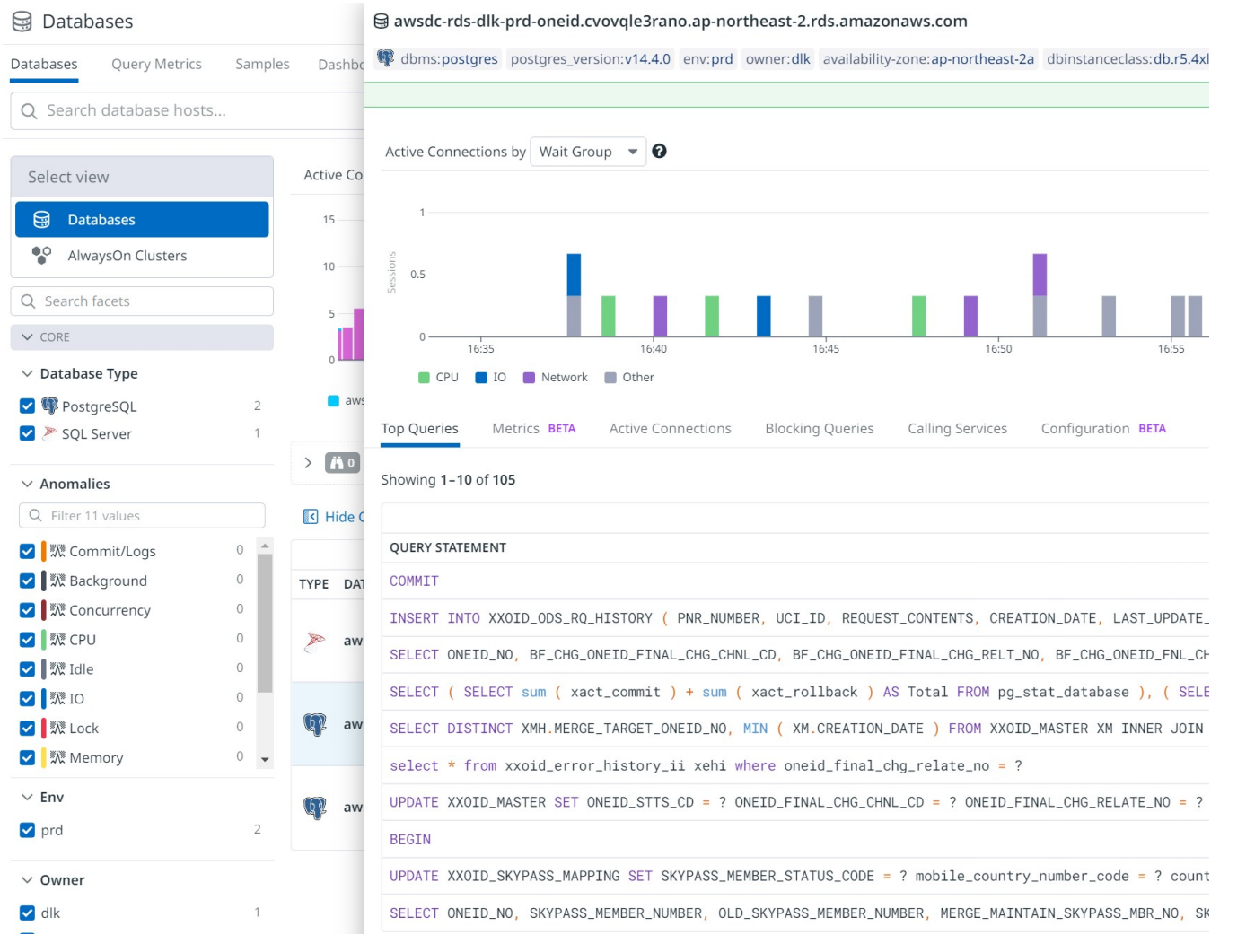
AWS에서 RDS를 모니터링 해주는 Performance Insight 기능을 활용한다.

- 성능 개선 도우미는 7일 동안 단계적으로 무료 성능 데이터 기록을 제고 한다.

6.2 Datadog

Datadog APM 모니터링 솔루션을 활용한다.

- Query별 TPS Latency 등을 함께 보여준다.
- 어떤 Query가 얼마나 많이 사용되고 있는지, 얼마나 느린지를 파악하는데 도움된다.



7. Reference (참조 문헌)

- [DataDog](#)