



mysqld_exporter for prometheus



Concepts:

mysql-server의 실제 process인 mysqld의 실질적인 운영 상태에 대한 Prometheus Expoter도 당연히 존재하고, Prometheus도 해당 Exporter를 Commercial하게 지원한다.

mysqld용 mysqld_exporter설치

다른 Exporter들과 마찬가지로 잘 설치 한다.

- https://prometheus.io/download/#mysqld_exporter

mysqld용 mysqld_exporter를 활용하기 위한 준비 작업

- 우선은 mysql에서 mysqld exporter를 위한 사용자를 만들고 적절한 권한을 부여 한다.

```
mysql> create user 'mysqlexp'@'localhost' identified by 'exp3306!!' \
        with max_user_connections 3;
mysql> grant process,replication client,select on *.* to 'mysqlexp'@'localhost';
mysql> flush privileges;

mysql> select user,host from mysql.user;
mysql> show grants for mysqlexp@localhost;
+-----+
| Grants for mysqlexp@localhost |
+-----+
| GRANT SELECT, PROCESS, REPLICATION CLIENT ON *.* TO `mysqlexp`@`localhost` |
+-----+
mysql>
```

- 감시 대상으로 할 mysql-server의 설정에 해당 mysqld로 요청이 오는 Third party program, 즉, Third party client에 대한 내용을 설정해 주어야 한다. 설정은 `/etc/mysql/my.cnf` 에 내용을 추가한다.

```
[client]
socket=/var/run/mysqld/mysqld.sock
user=mysqlexp
password=exp3306!!
```

- 대상 mysql의 Network접근을 위한 socket자원의 위치가 맞는지 확인하라.
- `/etc/mysql/my.cnf` 는 하부 디렉토리의 파일들을 include하는 방식의 설정으로 되어 있는데, 그냥 `my.cnf` 에 추가 하여도 된다.



mysql client, third party program

Mysqld에 연동해 오는 third party program들을 위한 설명과 상세 파라미터는 다음을 참고 한다.

- <https://dev.mysql.com/doc/refman/8.0/en/mysql-command-options.html>

추가로 mysqld가 사용하는 option files들의 구성과 가이드에 대해서는 다음을 참고 한다.

- <https://dev.mysql.com/doc/refman/8.0/en/option-files.html>

mysqld_exporter 실행하기

mysqld_exporter를 실행하기 위해서는 필요한 argument를 포함하여 실행하여야 한다. 기본적으로 mysqld_exporter가 사용할 my.cnf와 web listen port를 지정하여야 하며, 추가적으로 mysqld_exporter가 수집하고, 제공하여야 할 Metric들도 argument로 전달해 주어야 한다.

- 아래에 shell script로 정리해 놓은 내용을 우선 참고한다.

```
$ more gomyexp
#!/bin/bash
./mysqld_exporter \
--config.my-cnf=/etc/mysql/my.cnf \
--web.listen-address=0.0.0.0:9104 \
--collect.engine_tokudb_status \
--collect.global_status \
--collect.global_variables \
--collect.info_schema.clientstats \
--collect.info_schema.innodb_metrics \
--collect.info_schema.innodb_tablespace \
--collect.info_schema.innodb_cmp \
--collect.info_schema.innodb_cmpmem \
--collect.info_schema.processlist \
--collect.info_schema.processlist.min_time=0 \
--collect.info_schema.query_response_time \
--collect.info_schema.replica_host \
--collect.info_schema.tables \
--collect.info_schema.tables.databases='*' \
--collect.info_schema.tablestats \
--collect.info_schema.schemastats \
--collect.info_schema.userstats \
--collect.mysql.user \
--collect.perf_schema.eventsstatements \
--collect.perf_schema.eventsstatements.digest_text_limit=120 \
--collect.perf_schema.eventsstatements.limit=250 \
--collect.perf_schema.eventsstatements.timelimit=86400 \
--collect.perf_schema.eventsstatementssum \
--collect.perf_schema.eventswaits \
--collect.perf_schema.file_events \
--collect.perf_schema.file_instances \
--collect.perf_schema.file_instances.remove_prefix=false \
--collect.perf_schema.indexiowaits \
--collect.perf_schema.memory_events \
--collect.perf_schema.memory_events.remove_prefix=false \
--collect.perf_schema.tableiowaits \
--collect.perf_schema.tablelocks \
--collect.perf_schema.replication_group_members \
--collect.perf_schema.replication_group_member_stats \
--collect.perf_schema.replication_applier_status_by_worker \
--collect.slave_status \
--collect.slave_hosts \
--collect.heartbeat \
--collect.heartbeat.database=true \
--collect.heartbeat.table=true \
--collect.heartbeat.utc
$
```

- 이제 해당 shell script를 수행하여, mysqld_exporter를 실행하고, 확인해 본다.

```
$ ./gomyexp &
$ curl localhost:9104/metrics
```

- 상당히 많은 내용의 Metric이 조회되는데, 각각의 의미를 파악하고, 활용하는 것은 알아서...



mysqld exporter의 수 많은 flag option에 대한 정보

해당 내요은 다음의 github에서 참고할만한 내용이 있다.

- https://github.com/prometheus/mysqld_exporter

prometheus에 해당 mysqld_exporter등록

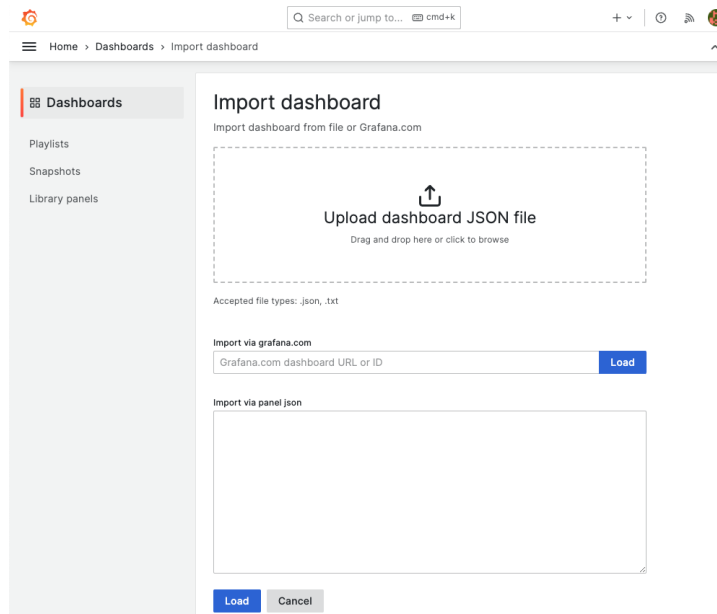
`prometheus.yml`에 해당 mysqld_exporter를 아래의 내용을 참고하여 대상으로 등록하고, prometheus를 재시작한다. 설정 내용에 여러 mysqld를 Group으로 관리할 경우를 감안하여, Label을 추가하였다.

```
- job_name: "palmMysql"
  scrape_interval: 5s
  static_configs:
    - targets: ["192.168.18.77:9104"]
      labels:
        svcgroup: 'mysqld'
```

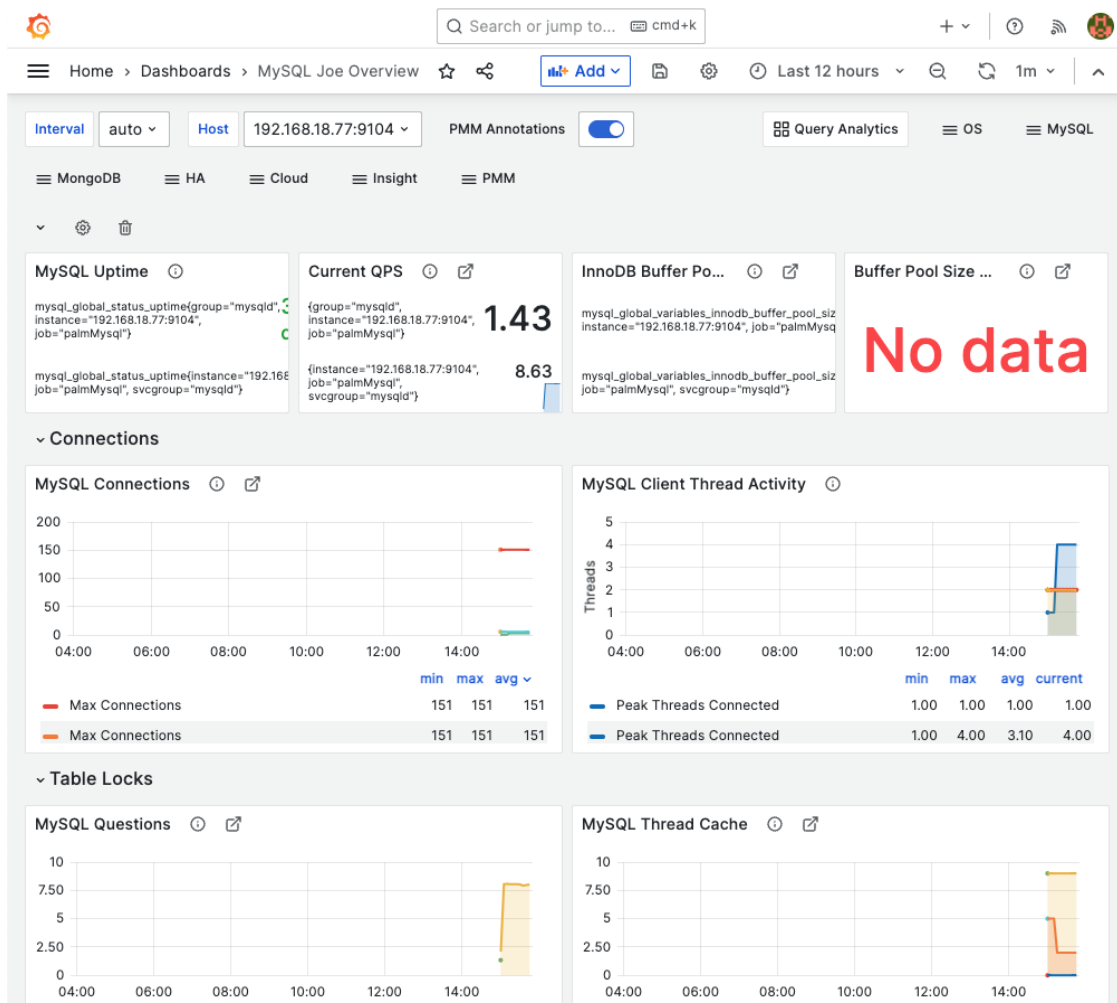
grafana에서 prometheus의 mysql들과 관련된 Metric들을 panel로 등록 하기

grafana에 mysqld_exporter에 대한 감시 panel을 구성해 본다. 다행인것은 grafana에 예제로 미리 만들어진 panel들이 있다는 것이다. 우리는 해당 panel구성정보를 grafana에 import하기만 하면 된다.

- 해당 예제 dashboard용 panel설정 파일은 아래에서 얻을 수 있다.
 - mysql overview: dashboard from percona monitoring project (<https://grafana.com/grafana/dashboards/7362-mysql-overview/>)
 - grafana cloud용 ID 말고, JSON을 download 받는다.
- 다운로드 받은 JSON형태의 dashboard panel설정파일을 grafana에 Import한다.
 - grafana 화면상단의 "+ > import dashboard"하거나, Home > Dashboards > New:Import를 선택하여 진행하면, 아래의 화면을 볼 수 있다.



- 여기에서 앞에서 구한 JSON파일을 upload하고 몇몇 설정을 정리하면 된다. 이후 import된 dashboard를 확인하면, 다음과 유사한 내용을 만날 수 있다.



- 참고로 위에서 다운로드 받아 사용한 JSON파일을 첨부한다.

https://s3-us-west-2.amazonaws.com/secure.notion-static.com/34addbdc-e5bc-4a85-8b74-b40e3c787c24/mysql-overview_rev5.json

++=