Untapping PMM's Full Potential: Features For Customizing It

Percona University Montevideo

Agustín Gallego
Support Engineer

April 23rd, 2019

PERCONA

Agenda

- What is Percona Monitoring and Management (PMM)?
- How to extend its functionality
  - Adding external exporters
  - Getting data from custom queries
  - Extending collected metrics
  - Providing semantics to graphs with annotations

What is Percona Monitoring and Management?

What is Percona Monitoring and Management?

- Open Source software (as all Percona software)
- A collection of tools:
  - Prometheus
  - Grafana
  - Nginx
  - Consul
  - Query Analytics
  - PostgreSQL (MySQL in v1)
- https://github.com/percona/pmm/tree/PMM-2.0

What is Percona Monitoring and Management?

Percona Monitoring and Management

See the PMM docs for more information.

Main repositories

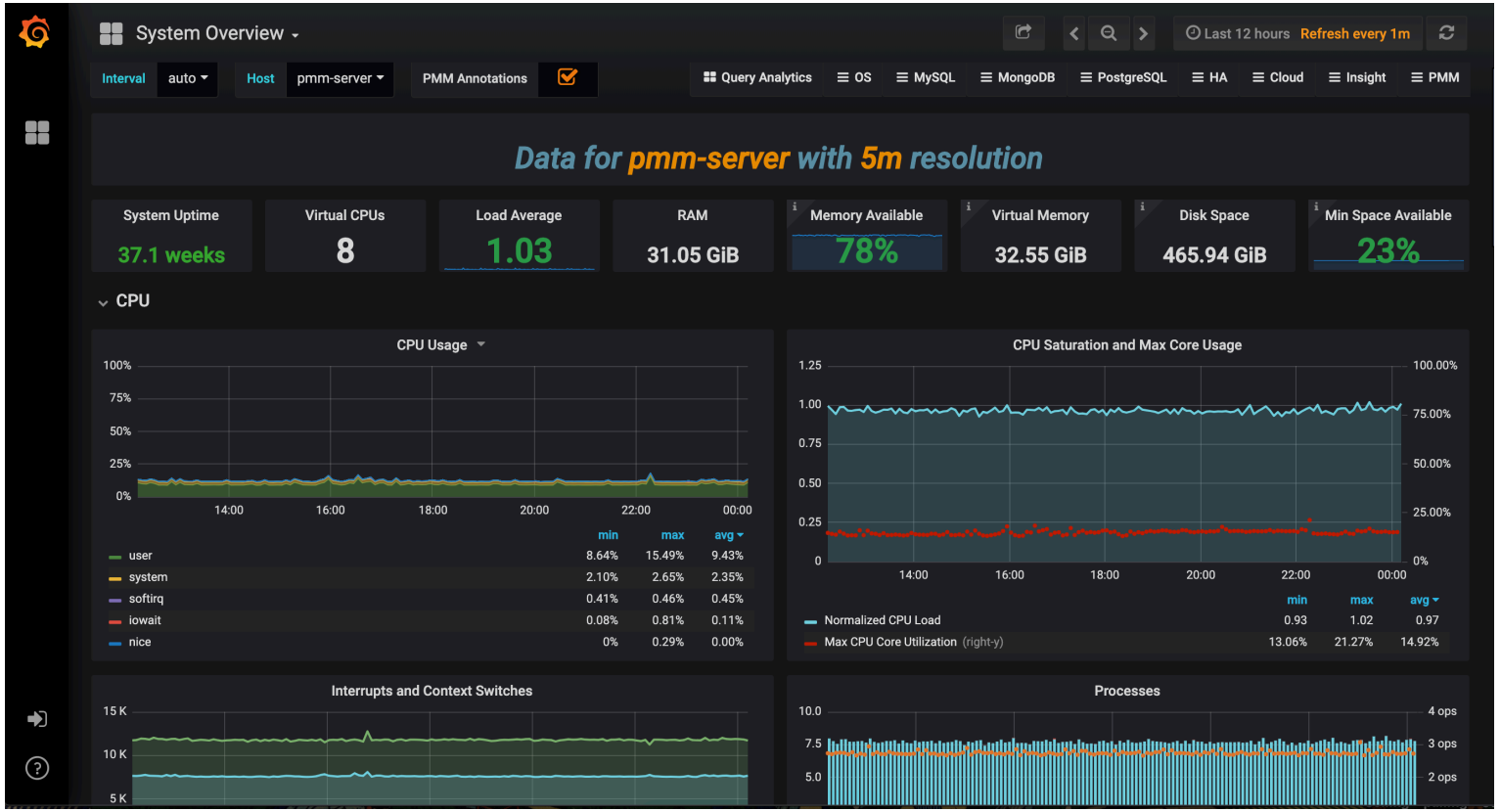
- pmm-server
  - qan-api
  - qan-app
  - grafana-dashboards
  - percona-toolkit
  - pmm-manage
  - pmm-managed
  - pmm-update
  - pmm-server-packaging
- pmm-client
  - qan-agent
  - node_exporter - based on `github.com/prometheus/node_exporter`
  - mysqld_exporter - based on `github.com/prometheus/mysqld_exporter`
  - mongodb_exporter - based on `github.com/dcu/mongodb_exporter`
  - rds_exporter - based on `github.com/Technofy/cloudwatch_exporter`
  - postgres_exporter - based on `github.com/wrouesnel/postgres_exporter`
  - proxysql_exporter

What is Percona Monitoring and Management?

- It's easy to deploy and test drive!
  - https://www.percona.com/doc/percona-monitoring-and-management/deploy/index.html
- There are three deployment methods:
  - Docker
  - OVA (Open Virtual Appliance)
  - AMI (Amazon Machine Instance)

What is Percona Monitoring and Management?

- https://pmmdemo.percona.com/



PERCONA

PMM's functionality

Out-of-the-box support

- PMM offers native support for:
  - MySQL / Percona Server for MySQL
  - MariaDB
  - MongoDB / Percona Server for MongoDB
  - PostgreSQL
  - Percona XtraDB Cluster
  - ProxySQL
  - Amazon RDS / Aurora MySQL
  - Linux (OS metrics)

Extending PMM's functionality

Extending PMM's functionality

- We are going to go through four different ways:
  - Adding external exporters
  - Getting data from custom queries
  - Getting data from custom scripts
  - Providing semantics to graphs with annotations

Adding external exporters

Adding external exporters

- Introducing ClickHouse
  - https://clickhouse.yandex/

Adding external exporters

- We will use Docker to emulate our environment:
  - one ClickHouse container
    - using ports 9000 (CLI) and 8123 (HTTP)
  - one ClickHouse exporter container
    - using port 9116

Adding external exporters

```
agustin@bm-support01 ~ $ docker network create --driver bridge clickhouse-network
d7f02a5841bceffb2cf3455aa0322244c9bef74a8aa4607665ea5f255085bda0

agustin@bm-support01 ~ $ docker run -d \
> --publish 8123:8123 \
> --publish 9000:9000 \
> --name clickhouse \
> --network clickhouse-network \
> guriandoro/clickhouse-pmm:1.0
0c5bc6e217ebab9a862076d90fe1ebc0681c71093770cb170bbaea9353380993

agustin@bm-support01 ~ $ curl 'http://localhost:8123/'
Ok.
```

Adding external exporters

```
agustin@bm-support01 ~ $ docker run -it --rm --network host yandex/clickhouse-client --host
localhost
ClickHouse client version 19.5.2.6 (official build).
Connecting to localhost:9000 as user default.
Connected to ClickHouse server version 1.1.54380 revision 54380.

0c5bc6e217eb :) show databases;

SHOW DATABASES

┌─name────┐
| default |
| system  |
└─────────┘

2 rows in set. Elapsed: 0.014 sec.
```

Adding external exporters

```
agustin@bm-support01 ~ $ docker run -d \
> --publish 9116:9116 \
> --name clickhouse-exporter \
> --network clickhouse-network \
> f1yegor/clickhouse-exporter -scrape_uri=http://clickhouse:8123/
b8c9e30cc057e75eef2894892ca36f13b7e09946818904d33c414c7c1c3985df

agustin@bm-support01 ~ $ curl -s 'http://localhost:9116/metrics' | head -n6
# HELP clickhouse_arena_alloc_bytes_total Number of ArenaAllocBytes total processed
# TYPE clickhouse_arena_alloc_bytes_total counter
clickhouse_arena_alloc_bytes_total 4096
# HELP clickhouse_arena_alloc_chunks_total Number of ArenaAllocChunks total processed
# TYPE clickhouse_arena_alloc_chunks_total counter
clickhouse_arena_alloc_chunks_total 1
```

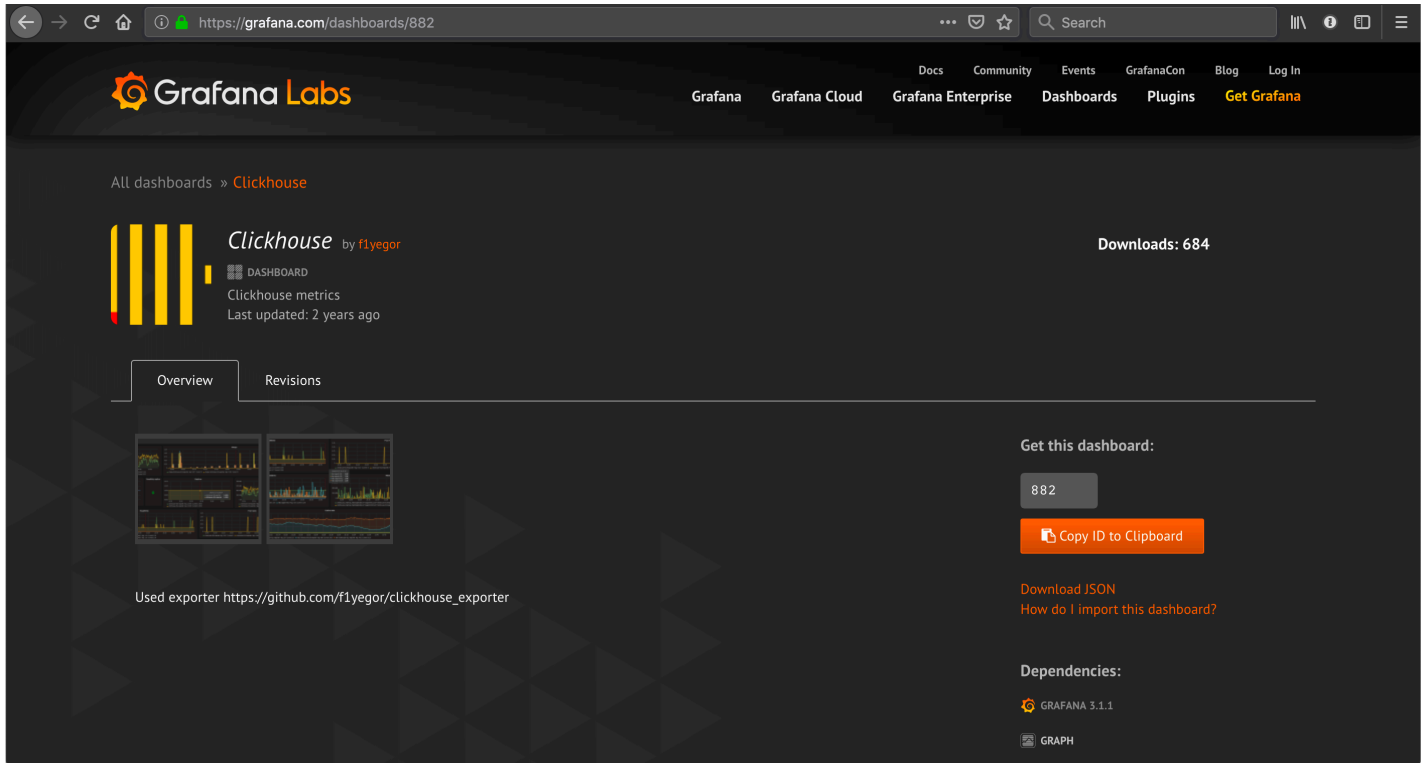
Adding external exporters

```
agustin@bm-support01 ~ $ pmm-admin add external:metrics clickhouse 172.31.0.3:9116
External metrics added.

agustin@bm-support01 ~ $ pmm-admin list
pmm-admin 1.17.1

PMM Server      | 127.0.0.1 (password-protected)
Client Name     | bm-support01.bm.int.percona.com
Client Address  | 172.17.0.1
Service Manager | linux-systemd

------------ ----------- ----------- -------- ---------------------------------------- --------
SERVICE TYPE   NAME        LOCAL PORT  RUNNING  DATA SOURCE                              OPTIONS
------------ ----------- ----------- -------- ---------------------------------------- --------
mysql:metrics  perf_mysql  42002       YES      root:***@tcp(127.0.0.1:19125)
mysql:metrics  ps_5.7      42003       YES      root:***@unix(/tmp/mysql_sandbox22389.sock)

Job name    Scrape interval  Scrape timeout  Metrics path  Scheme  Target           Labels  Health
clickhouse  1m0s             10s             /metrics      http    172.31.0.3:9116
```


PERCONA

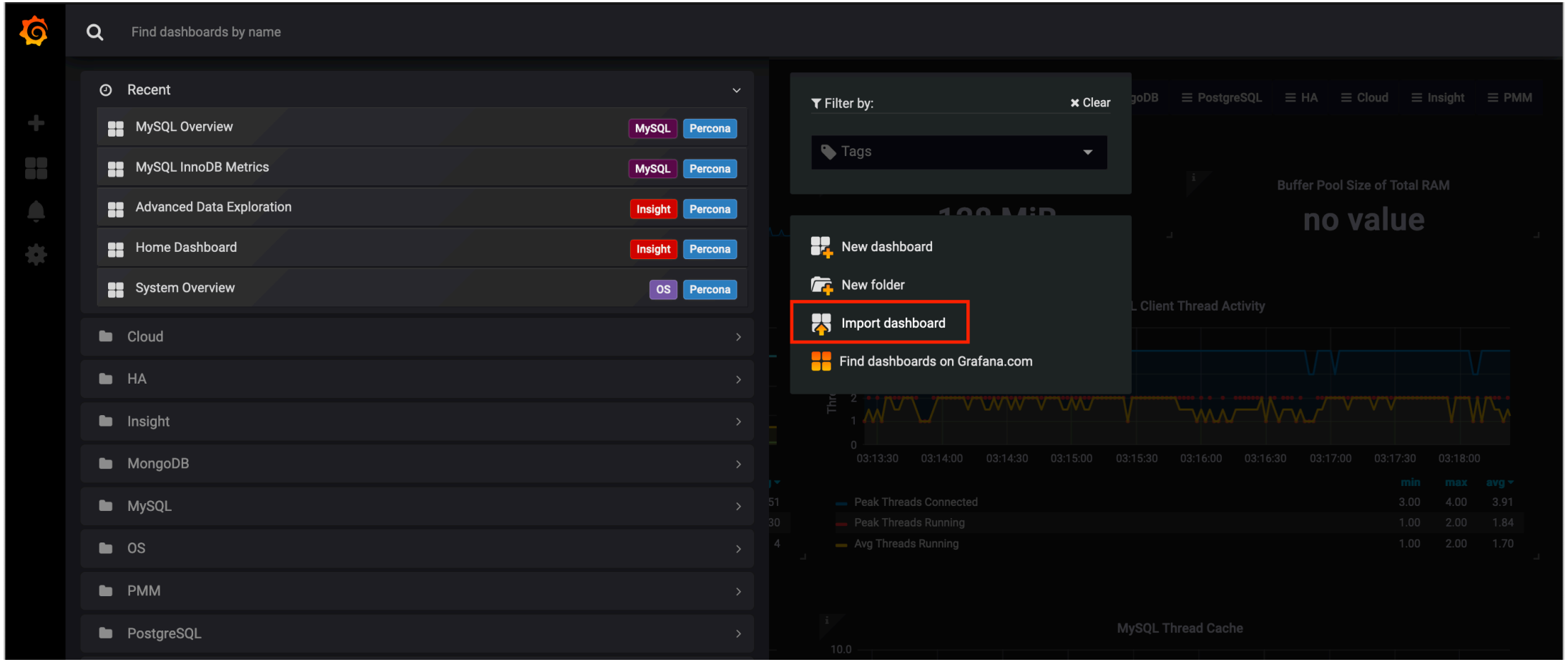
Adding external exporters

- We now need to add a Dashboard that can show the newly collected data



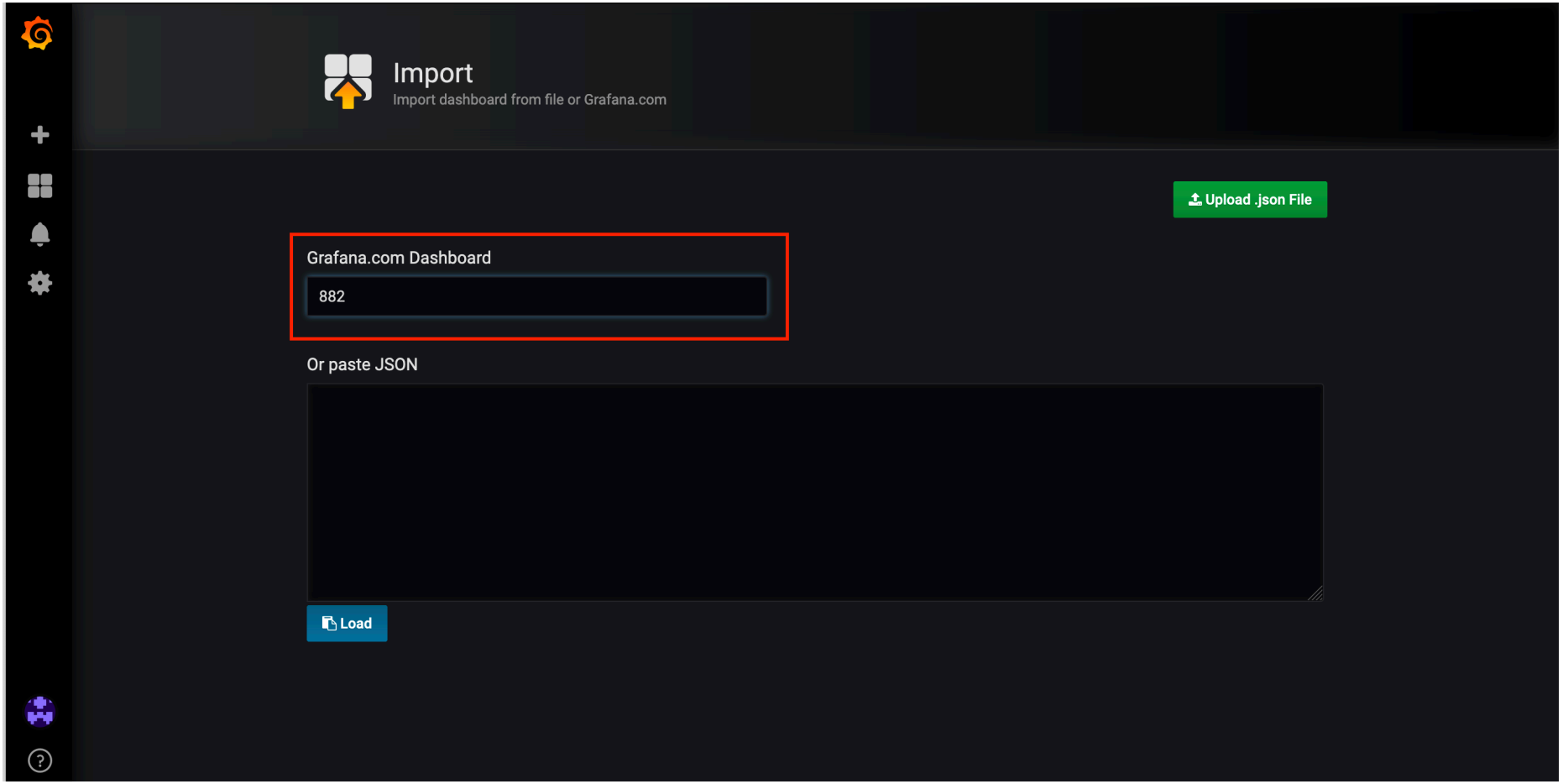
PERCONA

Adding external exporters

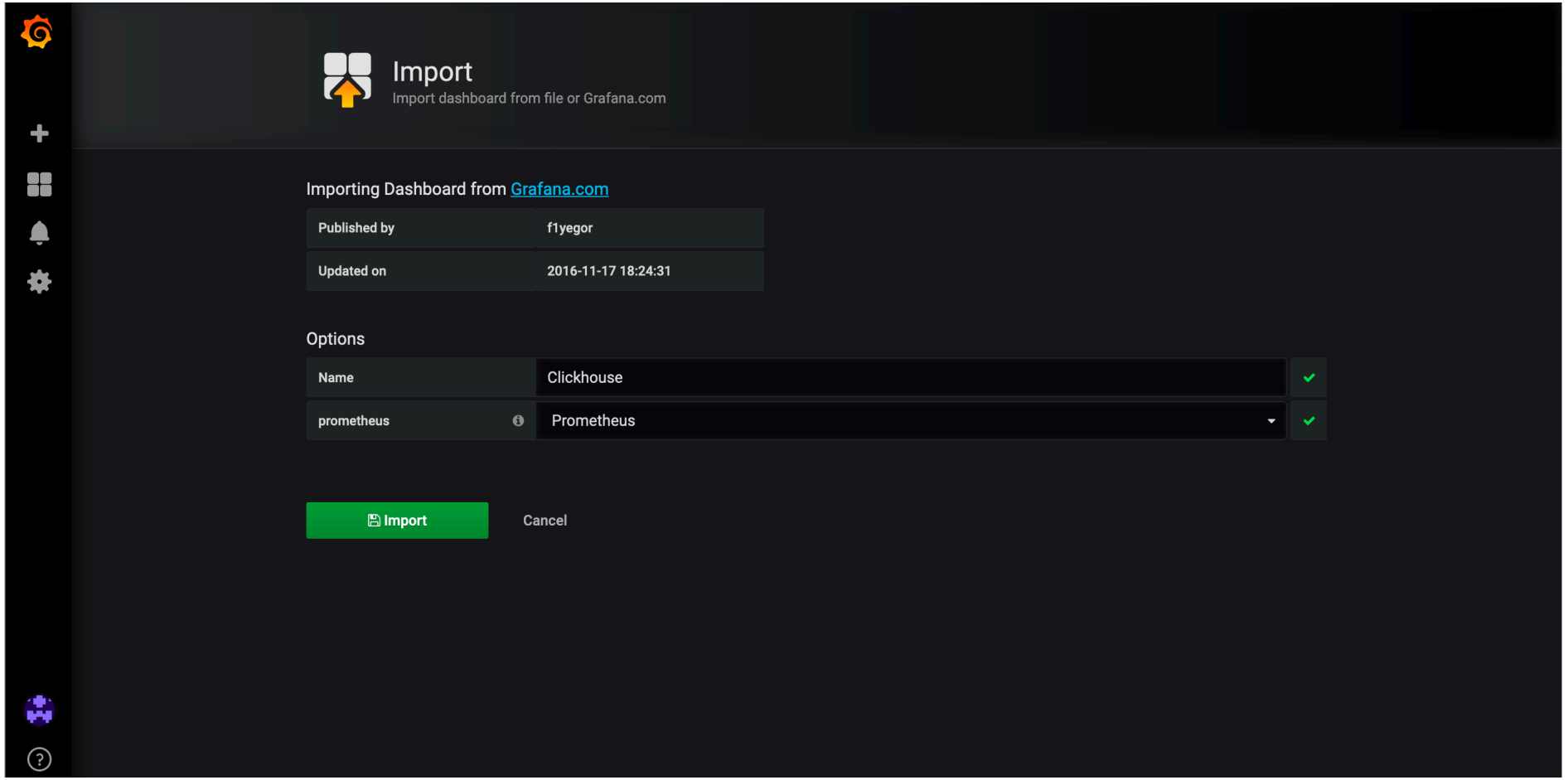


PERCONA

Adding external exporters



Adding external exporters

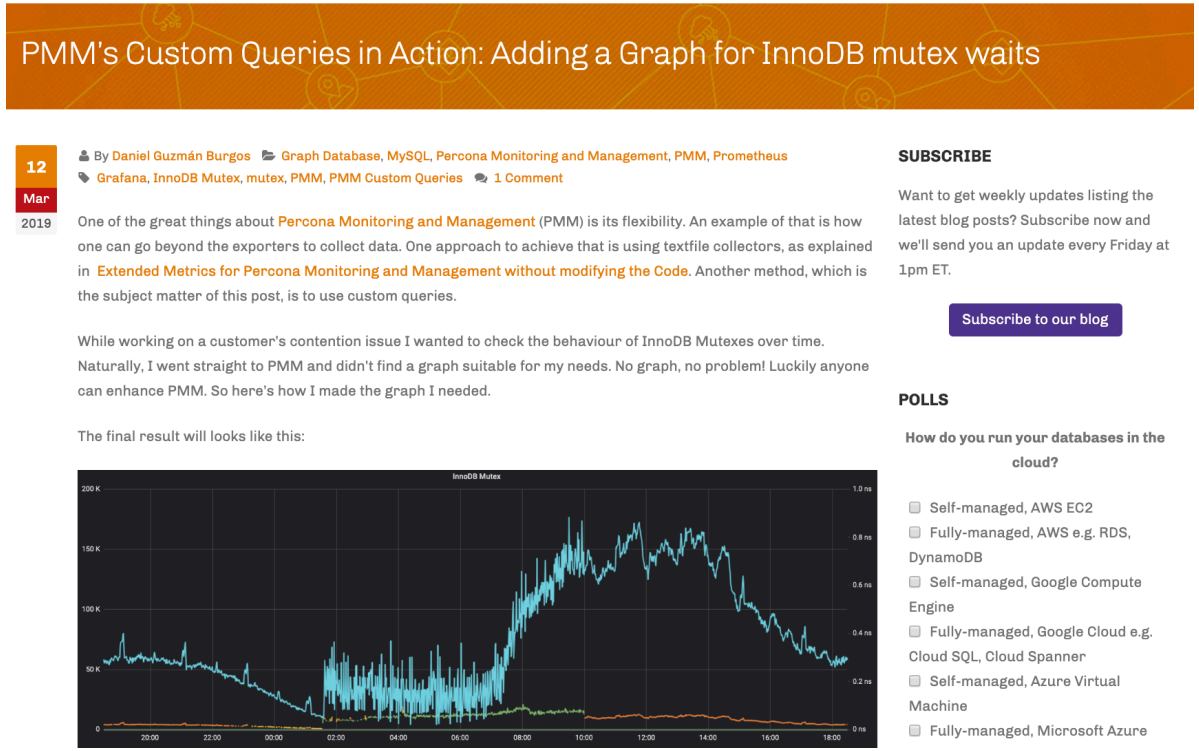


Adding external exporters

Getting data from custom queries

Getting data from custom queries

- Example from DGB's detailed blogpost:
  - PMM's Custom Queries in Action: Adding a Graph for InnoDB mutex waits



PMM's Custom Queries in Action: Adding a Graph for InnoDB mutex waits

12 Mar 2019

By Daniel Guzmán Burgos Graph Database, MySQL, Percona Monitoring and Management, PMM, Prometheus
Grafana, InnoDB Mutex, mutex, PMM, PMM Custom Queries 1 Comment

One of the great things about Percona Monitoring and Management (PMM) is its flexibility. An example of that is how one can go beyond the exporters to collect data. One approach to achieve that is using textfile collectors, as explained in Extended Metrics for Percona Monitoring and Management without modifying the Code. Another method, which is the subject matter of this post, is to use custom queries.

While working on a customer's contention issue I wanted to check the behaviour of InnoDB Mutexes over time. Naturally, I went straight to PMM and didn't find a graph suitable for my needs. No graph, no problem! Luckily anyone can enhance PMM. So here's how I made the graph I needed.

The final result will looks like this:

SUBSCRIBE

Want to get weekly updates listing the latest blog posts? Subscribe now and we'll send you an update every Friday at 1pm ET.

Subscribe to our blog

POLLS

How do you run your databases in the cloud?

- Self-managed, AWS EC2
- Fully-managed, AWS e.g. RDS, DynamoDB
- Self-managed, Google Compute Engine
- Fully-managed, Google Cloud e.g. Cloud SQL, Cloud Spanner
- Self-managed, Azure Virtual Machine
- Fully-managed, Microsoft Azure

Getting data from custom queries

- Example from DGB's detailed blogpost:
  - PMM’s Custom Queries in Action: Adding a Graph for InnoDB mutex waits
- Introduced in PMM 1.15.0
- By default checks the following file:
  - `/usr/local/percona/pmm-client/queries-mysqld.yml`
- But it can be overridden with:
  - `pmm-admin add mysql:metrics -- --queries-file-name=\ /usr/local/percona/pmm-client/custom-query.yml`

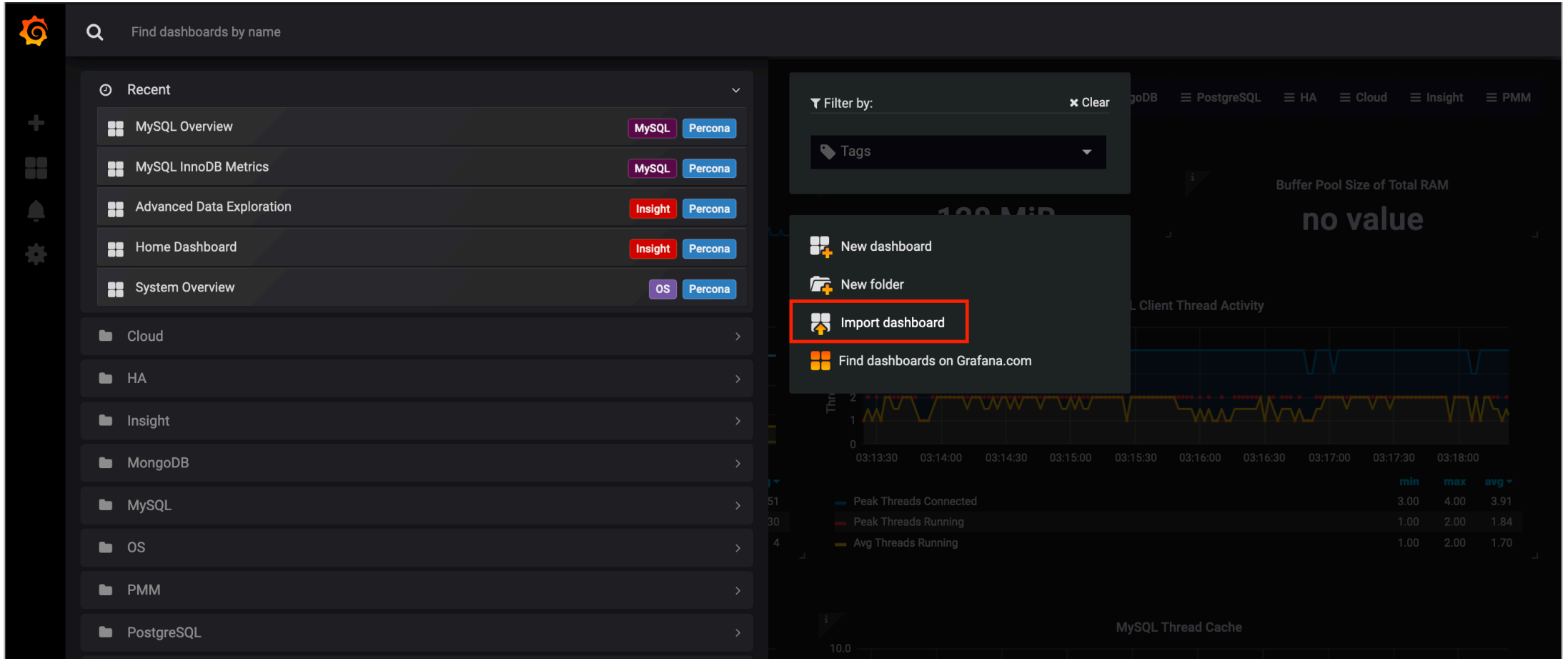
Getting data from custom queries

```
mysql> SELECT @@global.performance_schema;
+-----------------------------+
| @@global.performance_schema |
+-----------------------------+
|                           1 |
+-----------------------------+
1 row in set (0.00 sec)

mysql> UPDATE performance_schema.setup_instruments SET enabled='YES' WHERE name LIKE 'wait/
synch/mutex/innodb%';
Query OK, 63 rows affected (0.01 sec)
Rows matched: 63  Changed: 63  Warnings: 0

mysql> UPDATE performance_schema.setup_consumers SET enabled='YES' WHERE name LIKE
'events_waits%';
Query OK, 3 rows affected (0.00 sec)
Rows matched: 3  Changed: 3  Warnings: 0
```

Getting data from custom queries

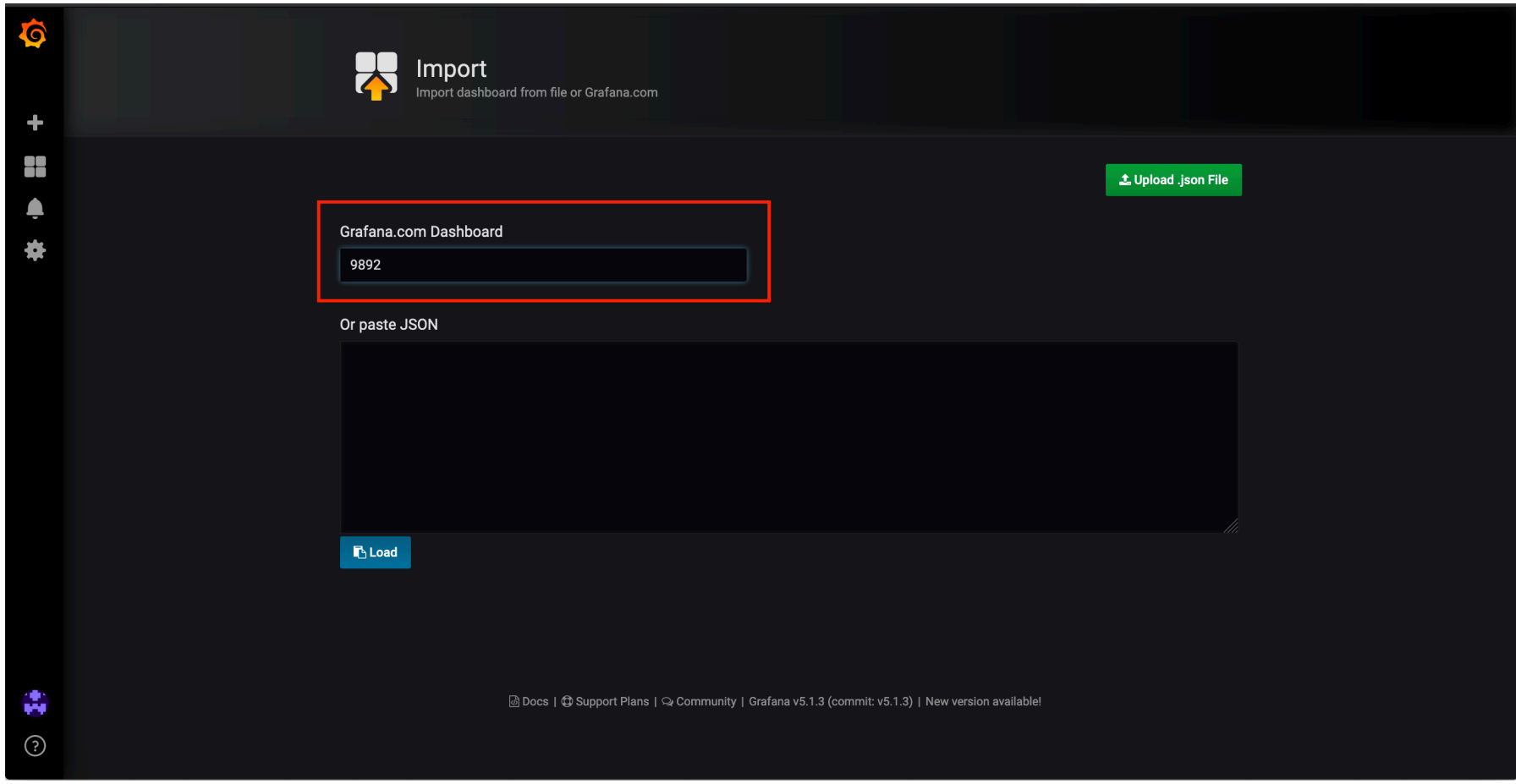
```
agustin@bm-support01 ~ $ cat /usr/local/percona/pmm-client/queries-mysqld.yml
mysql_global_status_innodb_mutex:
    query: "SELECT EVENT_NAME, COUNT_STAR, SUM_TIMER_WAIT FROM
performance_schema.events_waits_summary_global_by_event_name WHERE EVENT_NAME LIKE 'wait/
synch/mutex/innodb/%'"
    metrics:
      - EVENT_NAME:
          usage: "LABEL"
          description: "Name of the mutex"
      - COUNT_STAR:
          usage: "COUNTER"
          description: "Number of calls"
      - SUM_TIMER_WAIT:
          usage: "GAUGE"
          description: "Duration"
```

Getting data from custom queries



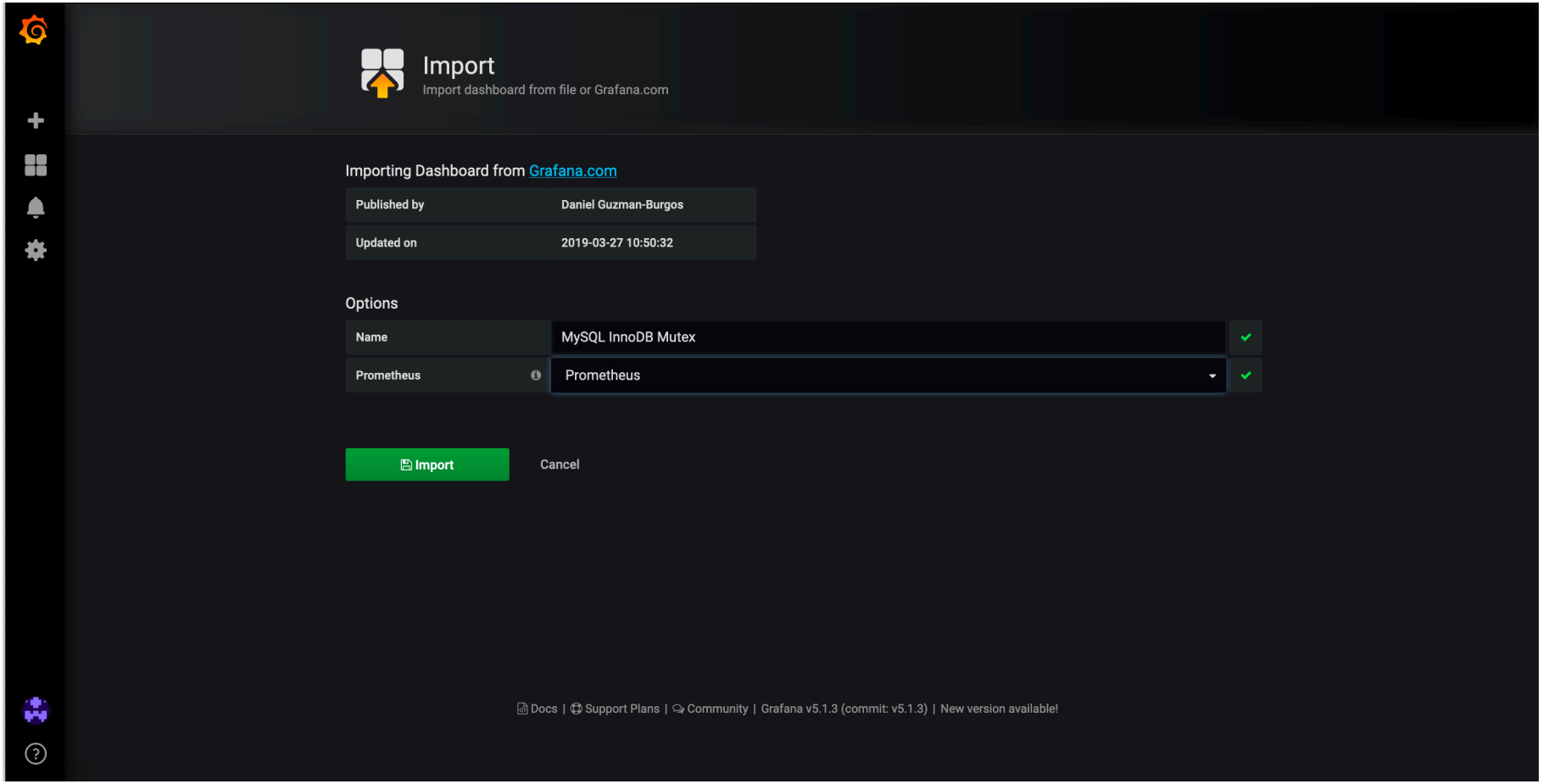
PERCONA

Getting data from custom queries



PERCONA

Getting data from custom queries



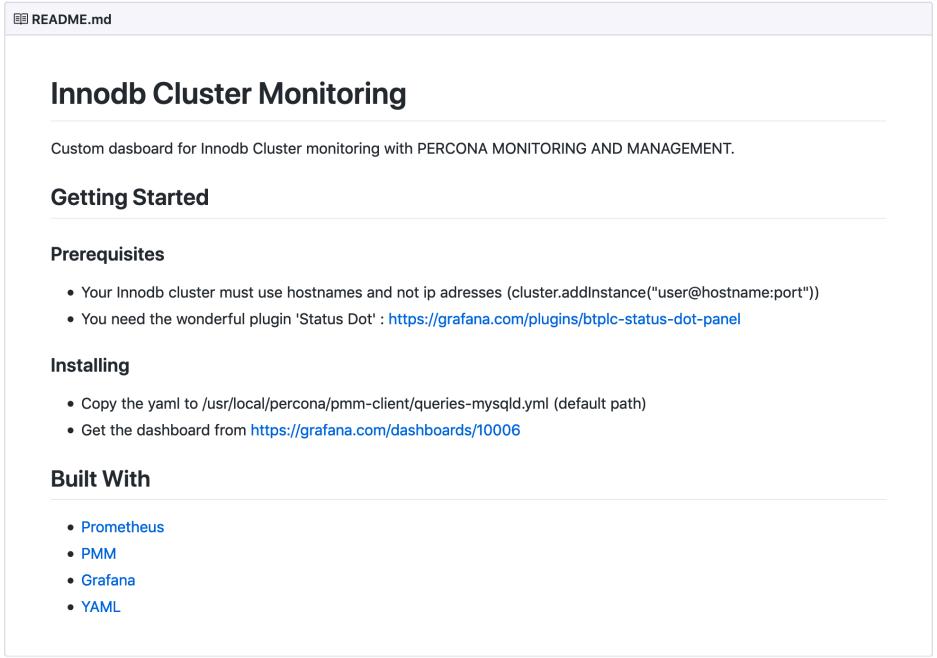
PERCONA

Getting data from custom queries

PERCONA

Getting data from custom queries

- Another example: a community-provided enhancement
- MySQL Group Replication monitoring
  - https://github.com/valentinmysql/MySQL-Custom-Queries-PMM



README.md

Innodb Cluster Monitoring

Custom dasboard for Innodb Cluster monitoring with PERCONA MONITORING AND MANAGEMENT.

Getting Started

Prerequisites

- Your Innodb cluster must use hostnames and not ip adresses (cluster.addInstance("user@hostname:port"))
- You need the wonderful plugin 'Status Dot' : https://grafana.com/plugins/btplc-status-dot-panel

Installing

- Copy the yaml to /usr/local/percona/pmm-client/queries-mysqld.yml (default path)
- Get the dashboard from https://grafana.com/dashboards/10006

Built With

- Prometheus
- PMM
- Grafana
- YAML

Getting data from custom scripts

Getting data from custom scripts

- PMM can also consume metrics from textfile collectors
- Introduced in PMM 1.16.0
- By default checks the following directory, for files named *.*prom*:
  - `/usr/local/percona/pmm-client/textfile-collector/`
- But it can be overridden by restarting the *linux:metrics* collector:
  - `pmm-admin rm linux:metrics`
  - `pmm-admin add linux:metrics -- \`
    `--collector.textfile.directory="/tmp/text-collectors/"`
- We will check a sample script that collects disk usage for a specific mount point

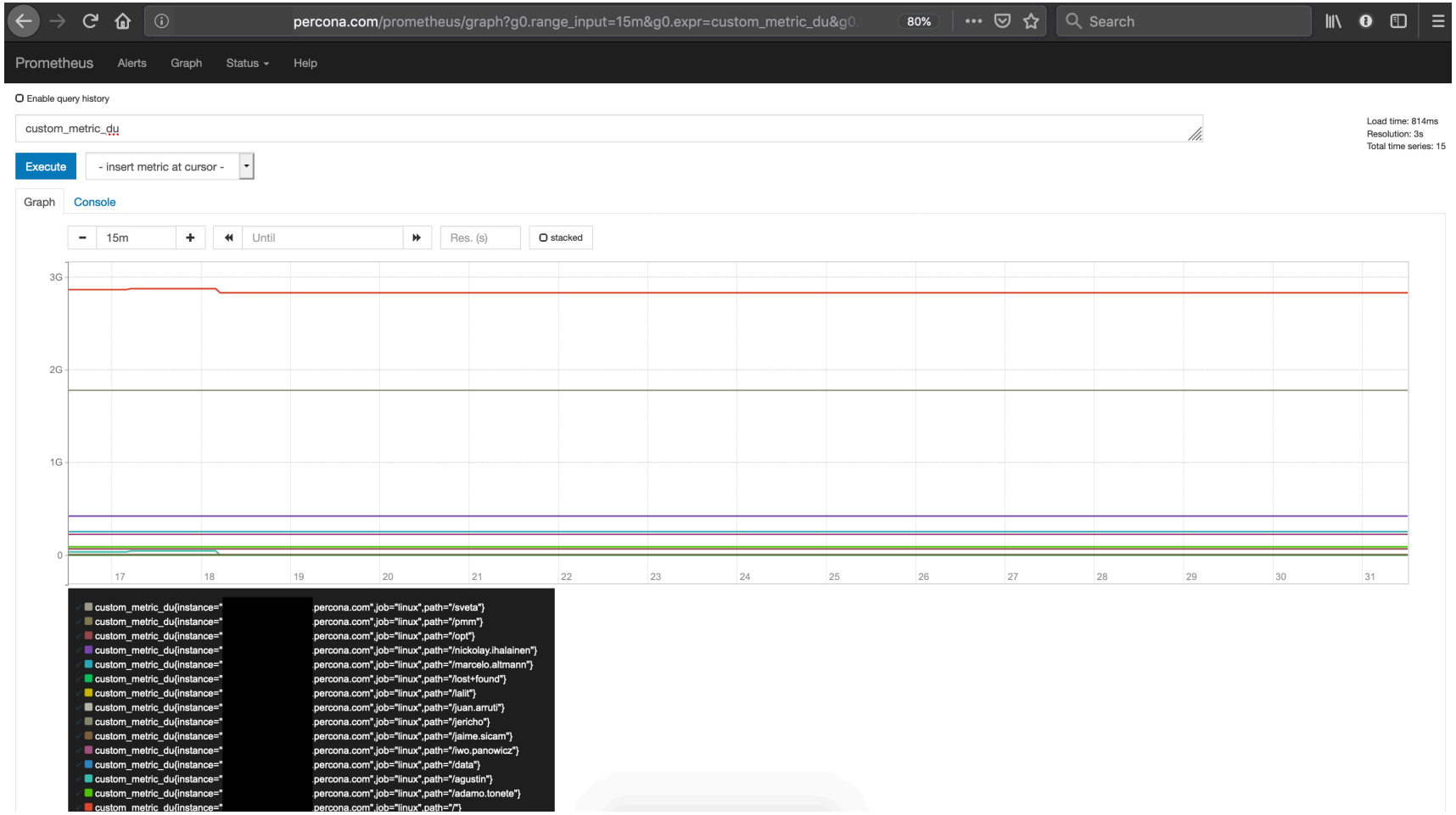
Getting data from custom scripts

```
ROOT_SHELL> crontab -l
* * * * * du --max-depth=1 /bigdisk/ 2>/dev/null | cut -d '/' -f1,3 | awk '{print
"custom_metric_du{path=\""$2 "\"} " $1}' > /usr/local/percona/pmm-client/textfile-collector/
du_bigdisk.prom


# Improved to get constant readings
# (if the script generates partial data, Prometheus will read partial data)

ROOT_SHELL> crontab -l
* * * * * du --max-depth=1 /bigdisk/ 2>/dev/null | cut -d '/' -f1,3 | awk '{print
"custom_metric_du{path=\""$2 "\"} " $1}' > /usr/local/percona/pmm-client/textfile-collector/
du_bigdisk.prom.bkp && mv /usr/local/percona/pmm-client/textfile-collector/
du_bigdisk.prom.bkp /usr/local/percona/pmm-client/textfile-collector/du_bigdisk.prom
```

Getting data from custom scripts

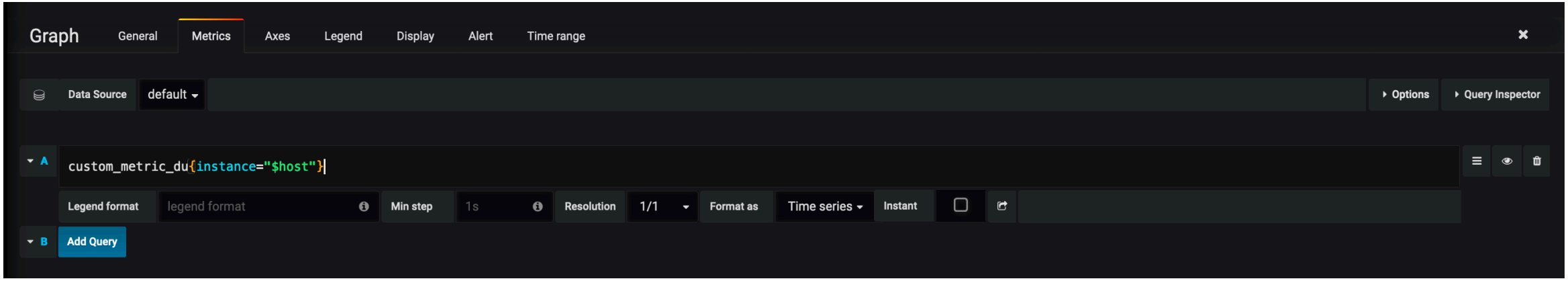
```
agustin@bm-support01 /bigdisk $ df -h .
Filesystem      Size  Used Avail Use% Mounted on
/dev/sdc1       5.5T  2.7T  2.8T  49% /bigdisk

agustin@bm-support01 /bigdisk $ cat /usr/local/percona/pmm-client/textfile-collector/du_bigdisk.prom
custom_metric_du{path="/lost+found"} 16
custom_metric_du{path="/opt"} 65635184
custom_metric_du{path="/agustin"} 4
custom_metric_du{path="/sveta"} 1777814948
custom_metric_du{path="/jericho"} 4
custom_metric_du{path="/jaime.sicam"} 4
custom_metric_du{path="/juan.arruti"} 144948
custom_metric_du{path="/lalit"} 468
custom_metric_du{path="/data"} 113016
custom_metric_du{path="/pmm"} 4250824
custom_metric_du{path="/marcelo.altmann"} 251912604
custom_metric_du{path="/nickolay.ihalainen"} 420175936
custom_metric_du{path="/adamo.tonete"} 88280336
custom_metric_du{path="/iwo.panowicz"} 223327332
custom_metric_du{path="/"} 2831655636
```

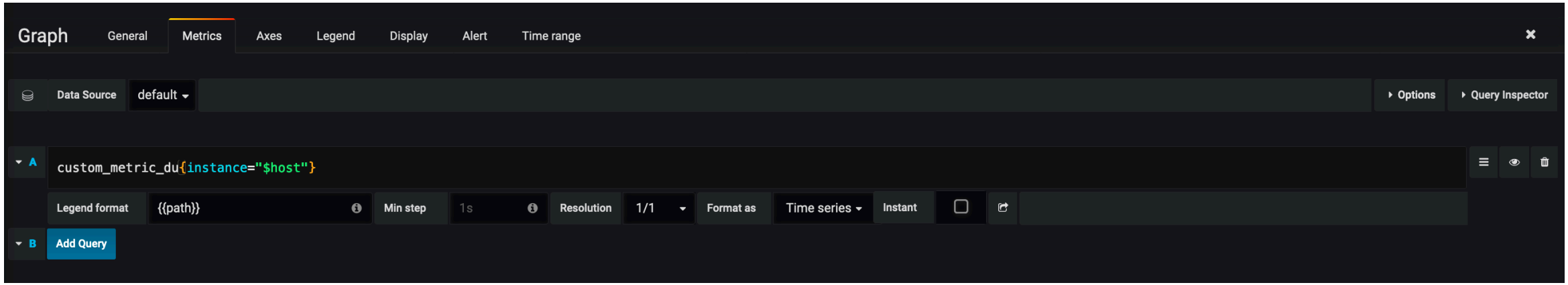
Getting data from custom scripts



PERCONA

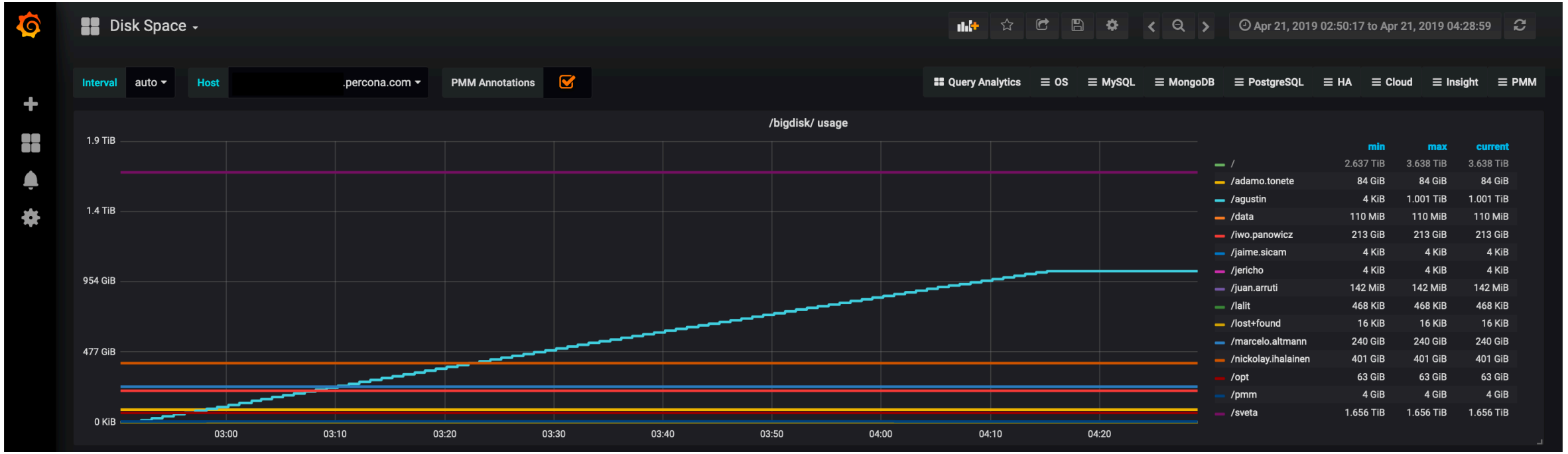
Getting data from custom scripts





PERCONA

Getting data from custom scripts



PERCONA

Getting data from custom scripts

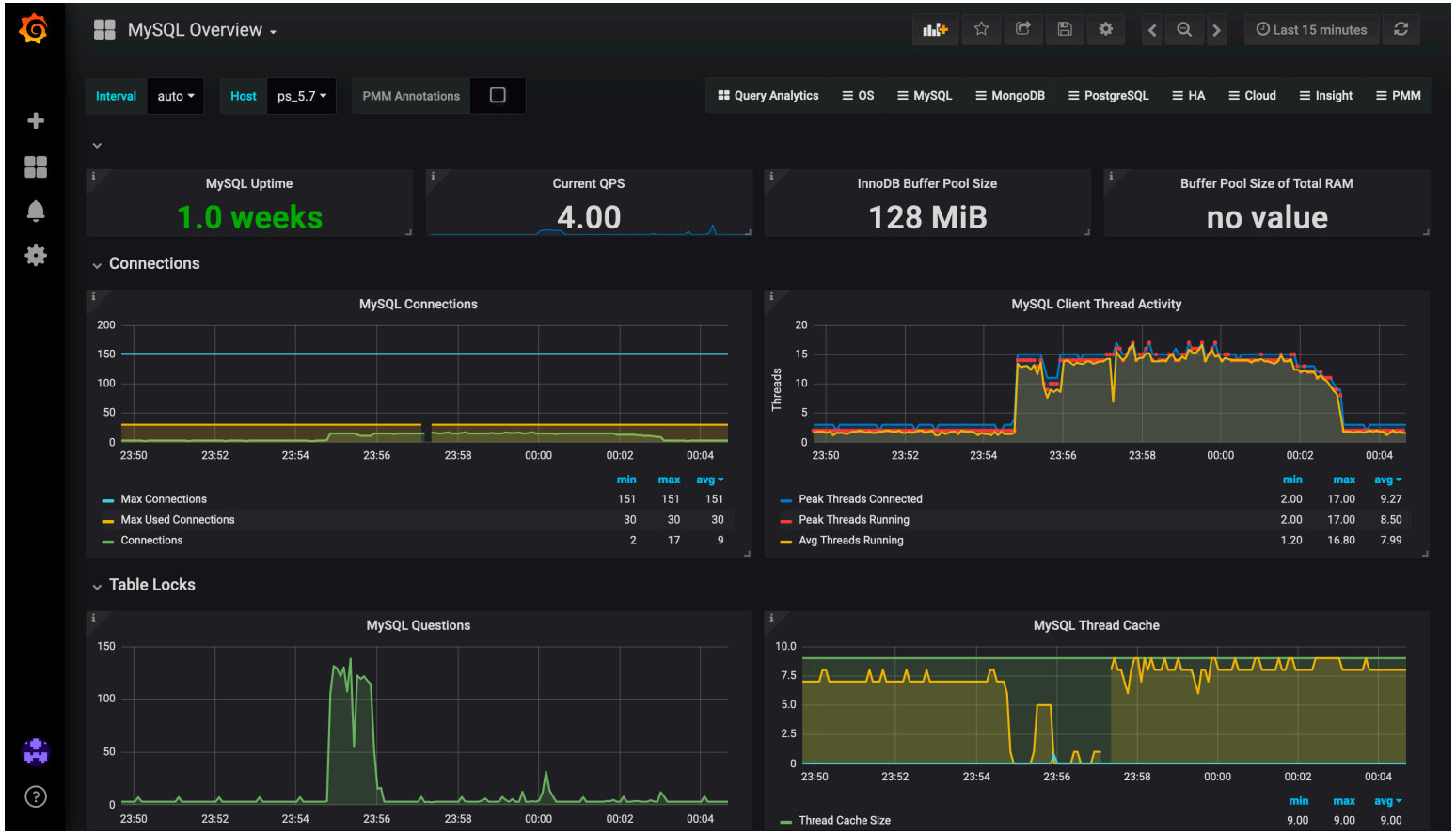
```
agustin@bm-support01 /bigdisk/agustin $ sysbench fileio --file-num=10 --file-total-size=1024G prepare
sysbench 1.0.16 (using bundled LuaJIT 2.1.0-beta2)

10 files, 107374182Kb each, 1048575Mb total
Creating files for the test...
Extra file open flags: (none)
Creating file test_file.0
Creating file test_file.1
Creating file test_file.2
Creating file test_file.3
Creating file test_file.4
Creating file test_file.5
Creating file test_file.6
Creating file test_file.7
Creating file test_file.8
Creating file test_file.9
1099511726080 bytes written in 5020.51 seconds (208.86 MiB/sec).
```

Providing semantics to graphs with annotations

Providing semantics to graphs with annotations

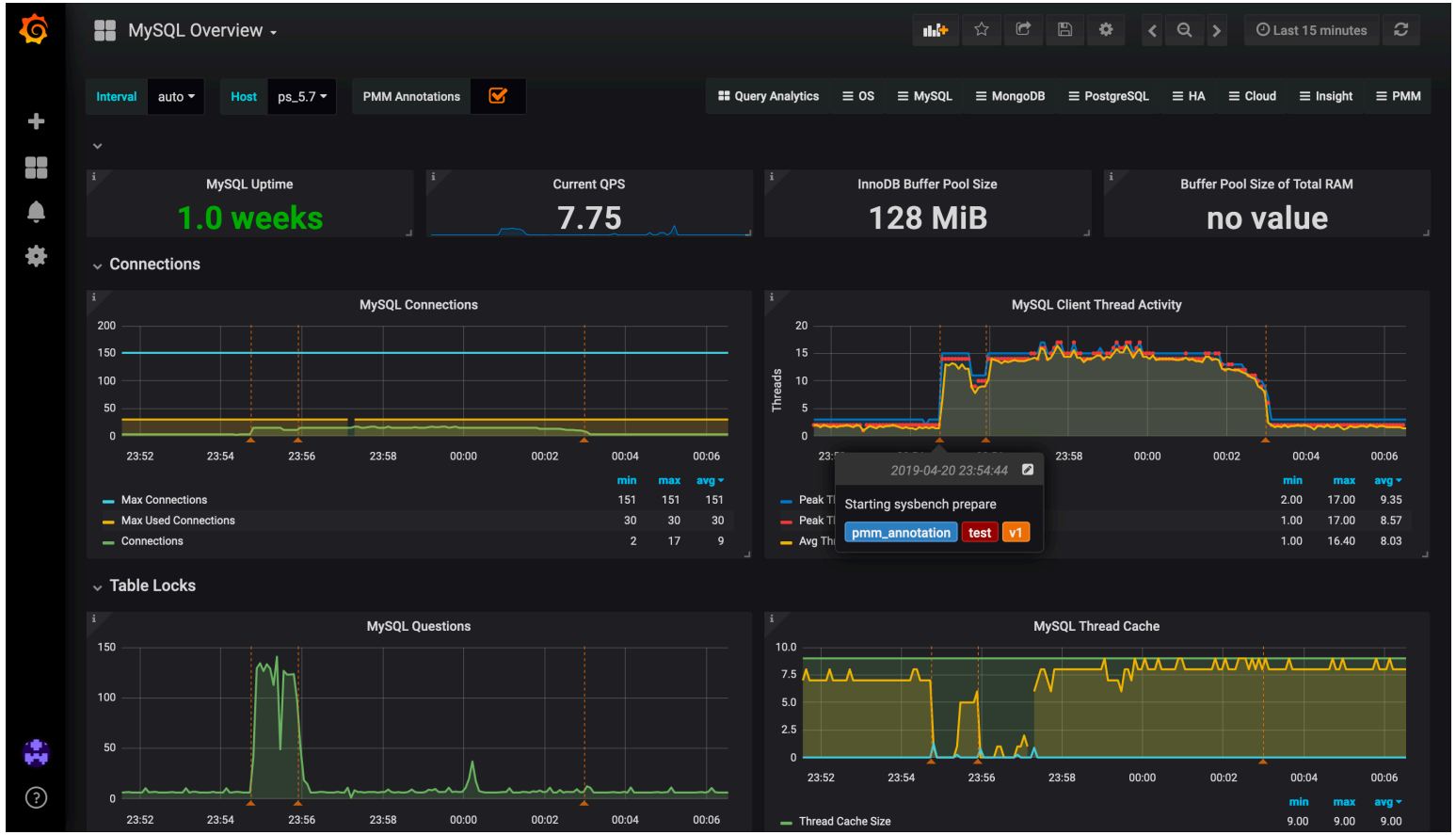
- Annotations can give us context on what is going on within the application, or other systems that use the database
- For instance, we can add a new annotation when:
  - we are about to run a backup script, and when it ends
  - we are about to start a maintenance window
  - we upgrade application version, or deploy a new functionality

Providing semantics to graphs with annotations



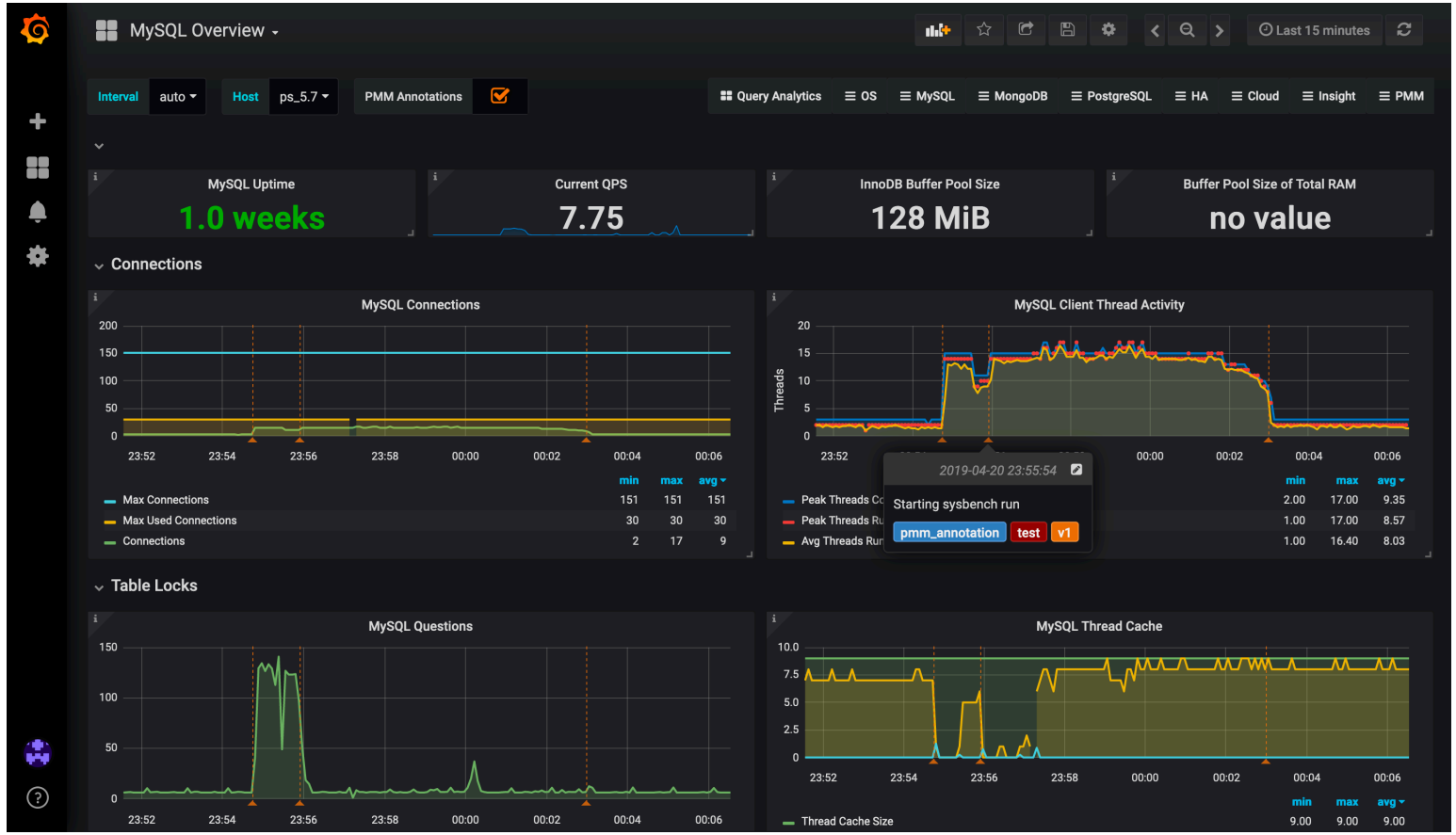
PERCONA

Providing semantics to graphs with annotations



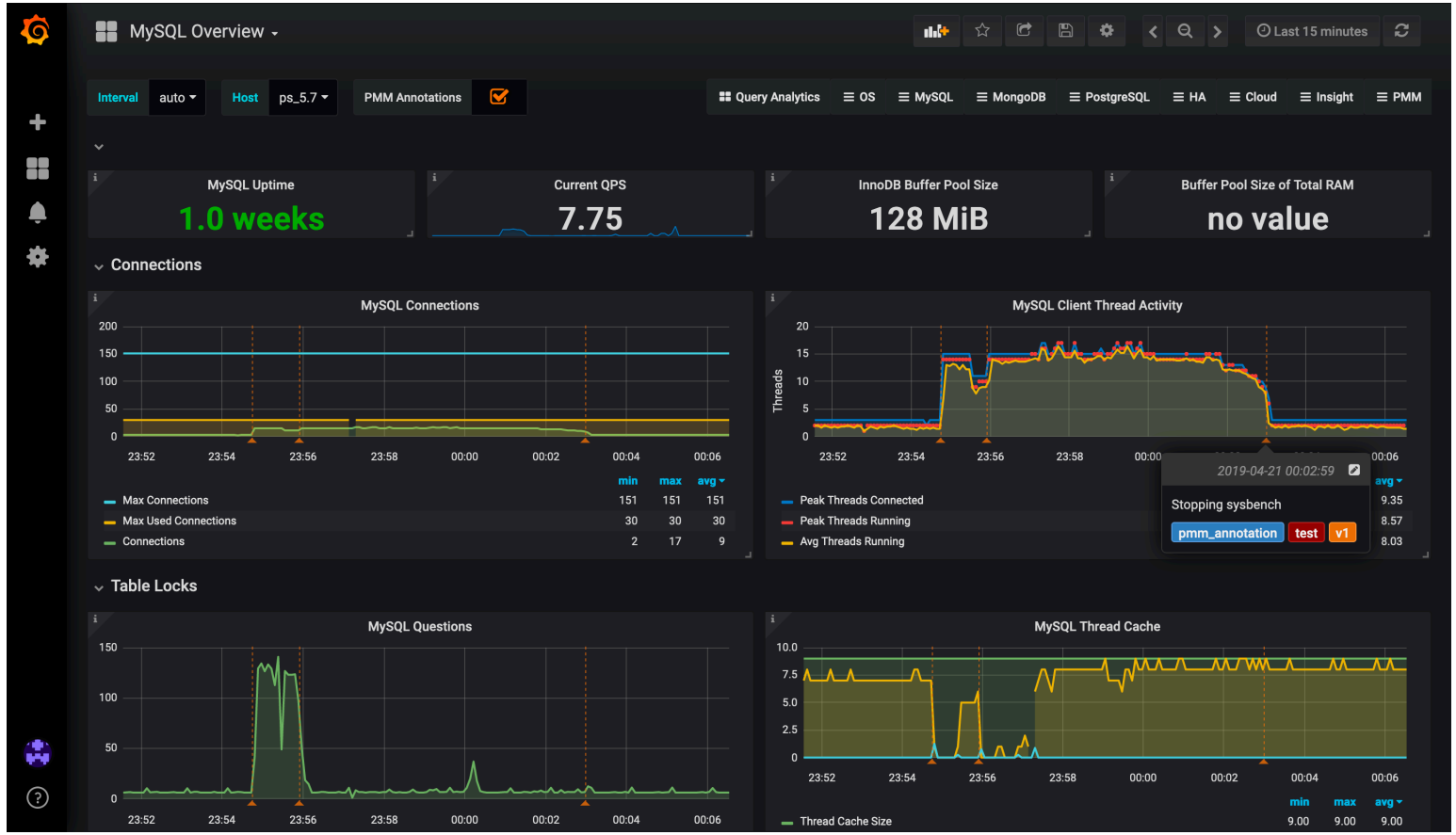
PERCONA

Providing semantics to graphs with annotations



PERCONA

Providing semantics to graphs with annotations



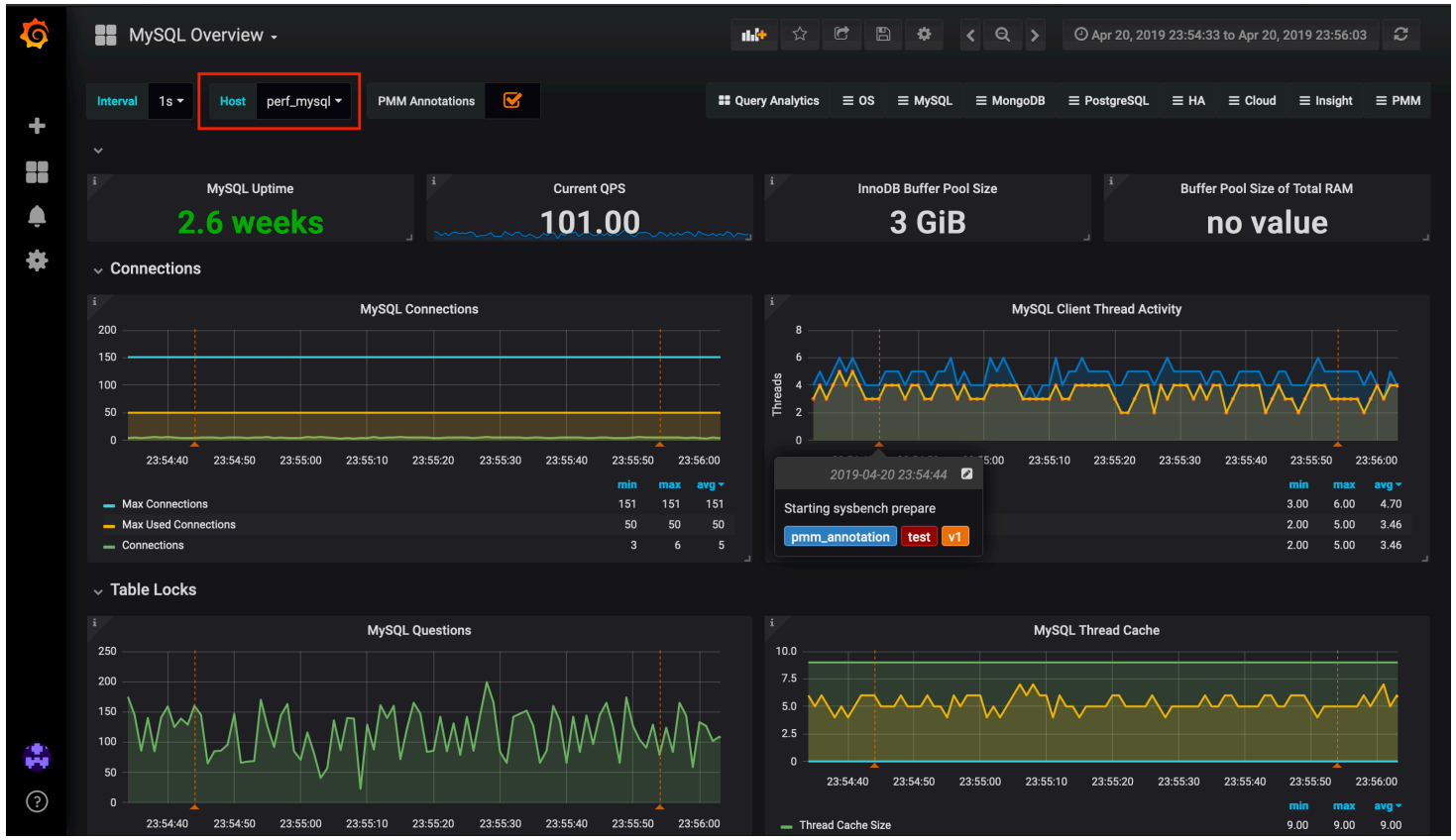
PERCONA

Providing semantics to graphs with annotations

```
#!/bin/bash

echo "adding annotation"
sudo pmm-admin annotate "Starting sysbench prepare" --tags "test,v1"

sysbench --tables=20 --table-size=1000000 --range-size=250000 --simple-ranges=6 --sum-ranges=3 \
--threads=12 --report-interval=3 --db-driver=mysql --mysql-socket=/tmp/mysql_sandbox22389.sock \
--mysql-user=root --mysql-password=msandbox --mysql-db=test \
/usr/share/sysbench/oltp_read_write.lua prepare

echo "adding annotation"
sudo pmm-admin annotate "Starting sysbench run" --tags "test,v1"

sysbench --tables=20 --table-size=1000000 --range-size=500000 --simple-ranges=6 --sum-ranges=3 --
threads=12 --time=300 --report-interval=10 --db-driver=mysql --mysql-socket=/tmp/mysql_sandbox22389.sock \
--mysql-user=root --mysql-password=msandbox --mysql-db=test /usr/share/sysbench/oltp_read_write.lua run

echo "adding annotation"
sudo pmm-admin annotate "Stopping sysbench" --tags "test,v1"
```

Providing semantics to graphs with annotations

- There is an issue, though...



PERCONA

Providing semantics to graphs with annotations

- There is an issue, though...
- The annotations are not host-specific
- There is no way of filtering out annotations from hosts we are filtering out in the Grafana dashboard
- This means that any annotation you add will be seen for all dashboards, all graphs, and all hosts

Providing semantics to graphs with annotations

- But we can see the light at the end of the tunnel...
- Reported in https://jira.percona.com/browse/PMM-2562
- Resolved by https://github.com/grafana/grafana/pull/10163
- Merge into PMM is still pending, but will surely be added soon

Questions? / Thank you!

Online survey + raffle

Percona University Montevideo

Respond and Win

Champions of Unbiased
Open Source Database Solutions