What We Learned Integrating Grafana and With Prometheus

While Building Percona Monitoring and Management

Dimitri Vanoverbeke

March 2nd, 2018

Grafanacon EU 2018

PERCONA

Apologies upfront

PERCONA

Percona's Purpose

To Champion Unbiased Open Source Database Solutions

PERCONA

Percona

We are not weather experts

PERCONA

Meaning

- **We Provide Solutions (Support, Consulting, Training, Remote DBA) for Open Source Databases**

- **All Software We Release is 100% Free and Open Source**

- **Helping Customers to Avoid Software Vendor Lock-In**

- **Providing Open Source Alternatives to the "Enterprise" Software editions**

© 2018 Percona.

P E R C O N A

State of Open Source Database Monitoring

- **Commercial**
- **Cloud Only**
- **Do It Yourself**

- **MySQL Enterprise Monitor**
- **MongoDB Ops Manager**

PERCONA

Percona Monitoring and Management

- **100% Free and Open Source**
- **Works In and out of the Cloud**
- **Deploy in 15 minutes**
- **Purpose Built for Open Source Database Monitoring**
- **For Developers and DBAs**

The Main Question

How to build Great Software with Small Team ?

PERCONA

You can do it!

- **Working together with Open Source Community**

© 2018 Percona.

PERCONA

We chose

PERCONA

Reasons

- **Large Community**

- **Designed as an Extensible Platform**

- **Very Responsive Development Team**

- **Permissive License**

- **Members of Cloud Native Computing Foundation**

- **Modern Technology Support**

PERCONA

Our Innovation Focus

- **Queries are not only Metrics**
- **Simplify Deployment (easy deployment)**
- **Powerful Dashboards**

PERCONA

What Queries are Causing the Load

© 2018 Percona.
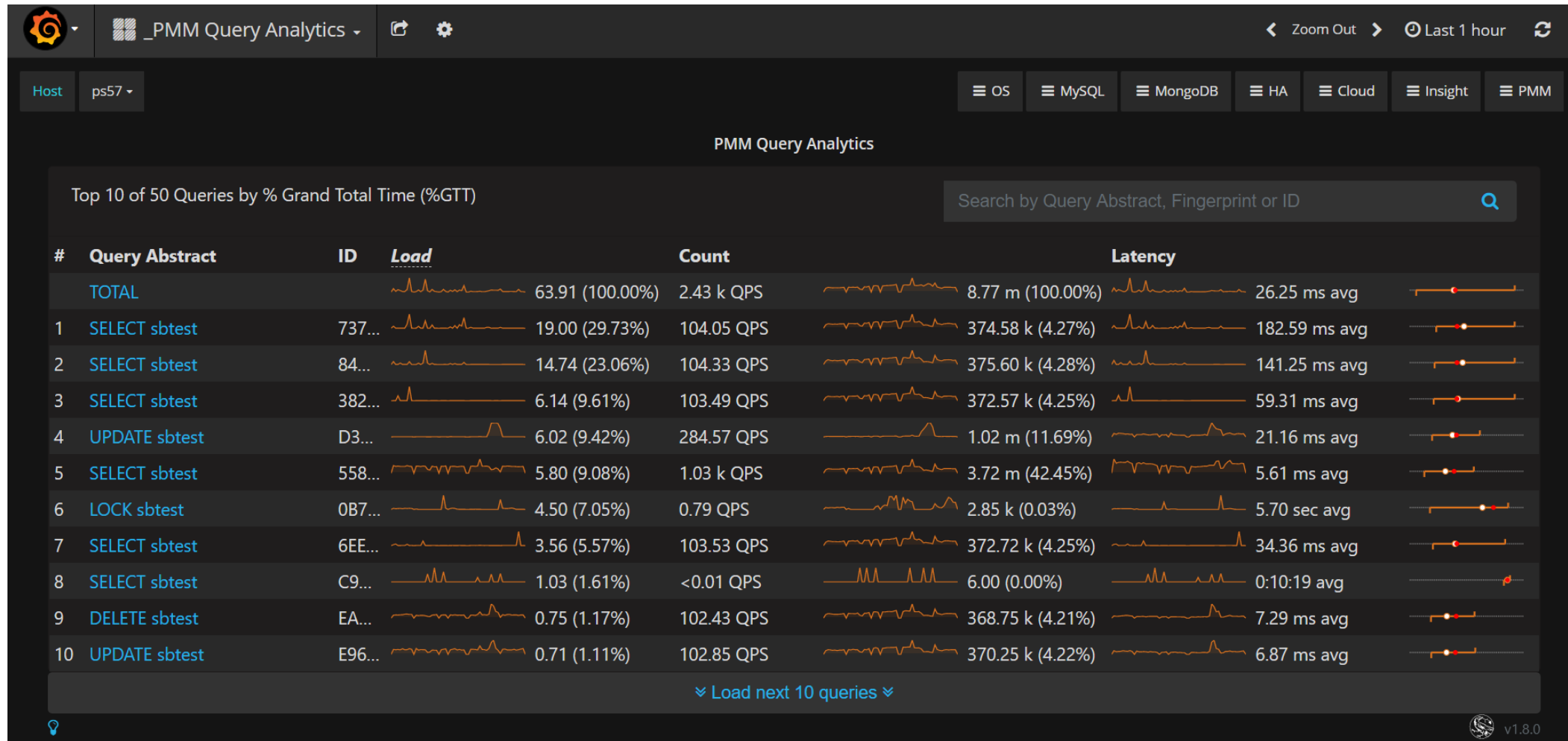
Why Are they Causing this Load

SELECT sbtest   737F39F04B198EF6

Metrics   Query first seen: ⏱ Aug 3, 2017 1:55 PM ••• Last seen: ⏱ Today at 9:46 AM

Metrics	Rate/Sec	Sum	Per Query Stats
Query Count	104.05 (per sec)	374.58 k 4.27% of total	
Query Time	19.00 load	18:59:56 29.73% of total	183.66 ms avg
Lock Time	0.11 (avg load)	0:06:42 1.35% of total 0.61% of query time	1.13 ms avg
Innodb IO Read Wait	0.61 (avg load)	0:36:44 9.10% of total 3.38% of query time	6.20 ms avg
Innodb Read Ops	52.35 (per sec)	188.45 k 7.62% of total	0.00 avg
Innodb Read Bytes	857.64 KB (per sec)	3.09 GB 7.62% of total 16.38 KB avg io size	8.22 KB avg
Innodb Distinct Pages	-	-	4.69 avg
Rows Sent	10.41 k (per sec)	37.46 m 30.52% of total	100.00 avg
Bytes Sent	1.30 MB (per sec)	4.67 GB 30.78% of total 124.71 Bytes bytes/row	12.47 KB avg
Rows Examined	1.14 m (per sec)	4.11 b 39.17% of total 109.79 per row sent	10.47 k avg
External Sorts (Filesort)	104.05 (per sec)	374.58 k 49.93% of total 100.00% of queries	-
Full Table Scans	0.01 (per sec)	40.00 0.17% of total 0.01% of queries	-
Queries Requiring Tmp Table In Memory	104.05 (per sec)	374.58 k 95.17% of total 100.00% of queries	-

PERCONA

How to Improve their Performance

```
∨Example

SELECT DISTINCT c
FROM sbtest1
WHERE id
    BETWEEN 5559
        AND 5658
ORDER BY  c
```

```
∨CREATE

CREATE TABLE `sbtest1` (
  `id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `k` int(10) unsigned NOT NULL DEFAULT '0',
  `c` char(120) NOT NULL DEFAULT '',
  `pad` char(60) NOT NULL DEFAULT '',
  PRIMARY KEY (`id`),
  KEY `k_1` (`k`)
) ENGINE=InnoDB AUTO_INCREMENT=100000001 DEFAULT
```

```
∨JSON

Expand All

-{
    "query_block": -{
        "select_id": 1,
        "cost_info": +{...},
        "ordering_operation": -{
            "using_filesort": false,
            "duplicates_removal": -{
                "using_temporary_table": true,
                "using_filesort": true,
                "cost_info": +{...},
                "table": +{...}
```
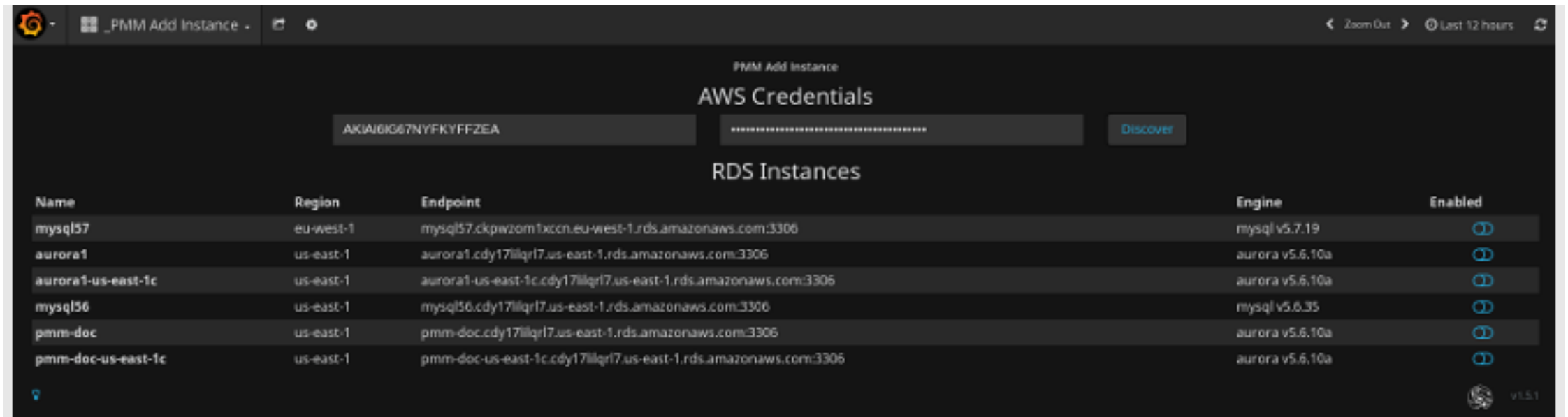
PERCONA

Integrating with Grafana

- **Using Greatly Designed Grafana Date Range Selector**
- **Grafana Host Selector**
- **Grafana Cross-Dashboard Links**
- **Integration Done through IFRAME**

PERCONA

Simplify Deployment

- **Deploying Multiple Prometheus Exporters is Hard**
- **Custom Agent is used for Query Functions**
- **Package everything to single "agent" package**
- **pmm-admin add mysql**

PERCONA

Integration with Cloud (AWS)

- **Automatically Search and Discover Supported Instances**
- **Another Custom Grafana Panel**

© 2018 Percona.

PERCONA

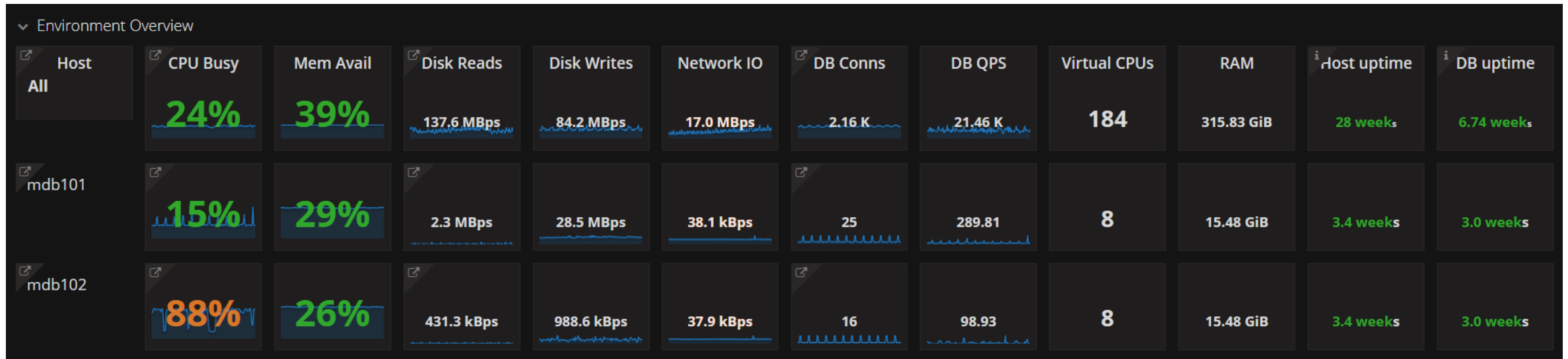# Powerful Dashboards

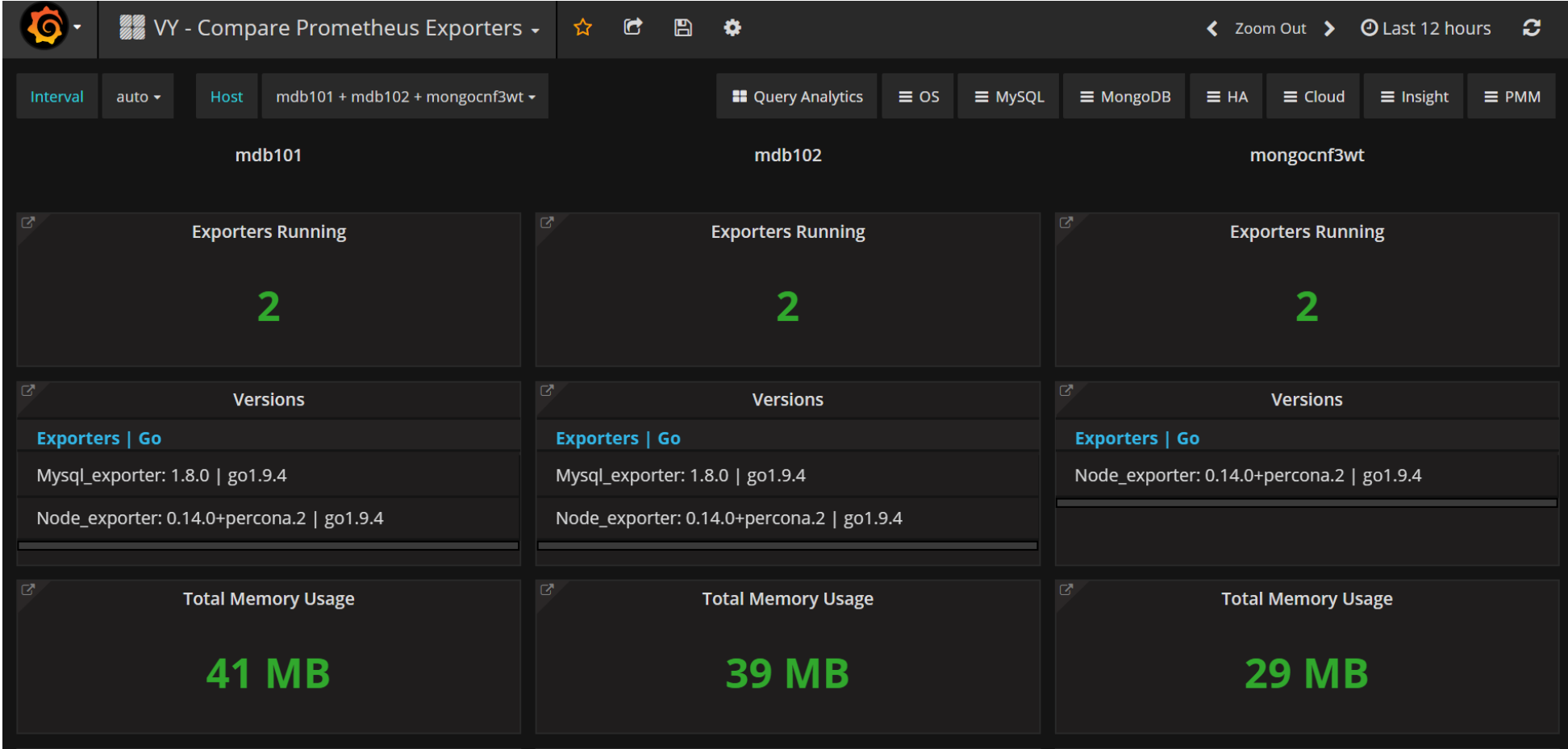
© 2018 Percona.

Dashboard Learning Experience

- **Great Dashboard API**
- **Fantastic support for Dashboard Versioning**
- **Can use ID for dashboard identification, not name**
- **Fantastic Templating Engine**
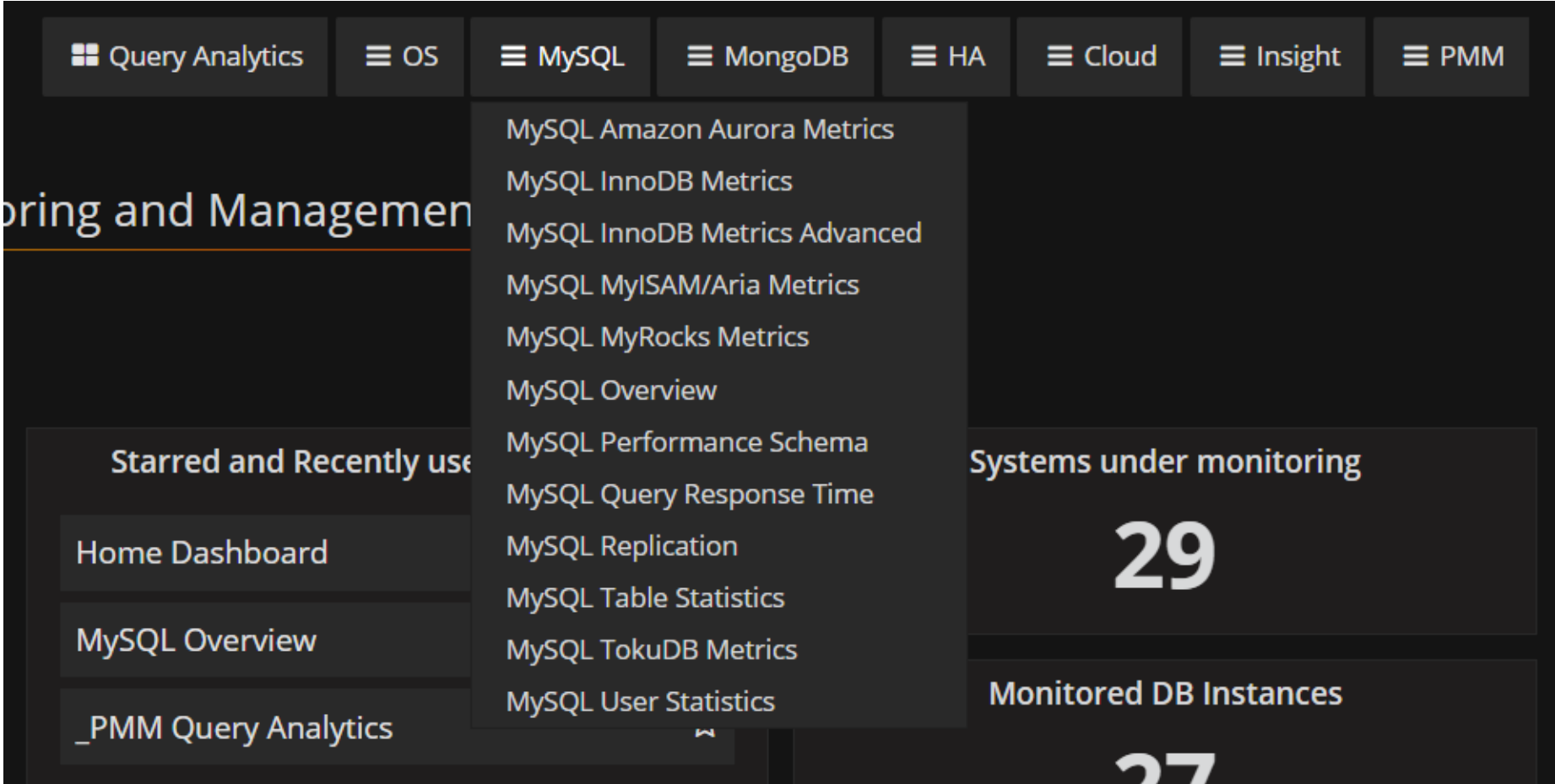- **Using Both Repeating rows and Repeated Columns**

PERCONA

Using Repeated Rows

Host All	CPU Busy	Mem Avail	Disk Reads	Disk Writes	Network IO	DB Conns	DB QPS	Virtual CPUs	RAM	Host uptime	DB uptime
	24%	**39%**	137.6 MBps	84.2 MBps	17.0 MBps	2.16 K	21.46 K	184	315.83 GiB	28 weeks	6.74 weeks
mdb101	**15%**	**29%**	2.3 MBps	28.5 MBps	38.1 kBps	25	289.81	8	15.48 GiB	3.4 weeks	3.0 weeks
mdb102	**88%**	**26%**	431.3 kBps	988.6 kBps	37.9 kBps	16	98.93	8	15.48 GiB	3.4 weeks	3.0 weeks

Environment Overview

PERCONA

Using Repeated Panels

© 2018 Percona.

Use Tags to build out Navigation Menu

© 2018 Percona.

Challenges

Alerting

- **How to use Alerting in multi-server environments**
- **Want to alert on Service rather than individual server**
- **Want "visual" alerts integrated with Grafana**
- **Use Grafana Alerts or build app for Prometheus Alerts ?**

PERCONA

Annotations

- **Want annotations to be entered by user**
- **But want them to appear on the "similar" panels everywhere**
- **How to define such "similar" panels ?**

PERCONA

SingleStat

- **Need fixed Y max value for Singlestat dashboard to better visualize percents**

- **Looking to contribute patch**

© 2018 Percona.

PERCONA
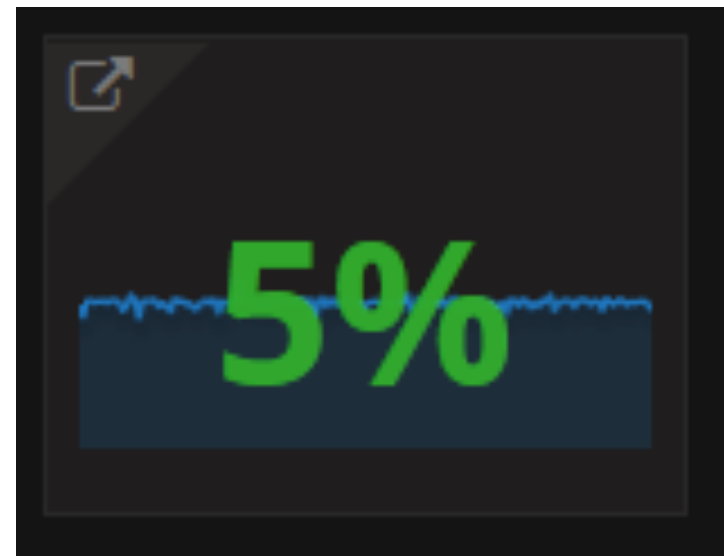
Dynamic Resolution

- **We have different data scraped with different resolution**
- **Some data is 1 second resolution (status variables)**
- **Other is 60 sec resolution (variables, table information)**
- **User can choose their own resolution based on overhead**
- **How to align it on the same graphs ?**
- **Make data totally zoomable**

PERCONA

Dynamic Resolution

- **Grafana Supports "minimum resolution"**
  - Hard to manage for dynamic user configuration
- **Prometheus Functions are very strict**
  - "do not invent any data
  - rate()[1s] Will return NULL not finding 2 data points
- **Use ugly hack so far**
  - Rate()[$interval] or irate()[5m]
  - Gives best available resolution

PERCONA

Better selection for Auto Interval

- **Auto Resolution is computed**
- **Interval selected may be "one data point for 2.2 minutes"**
- **Would be great to set it to fixed set of intervals**
- **1s; 5s; 1min; 5min; 1h …**

PERCONA

Thank You!