What do we want to monitor?
All the databases!

Agustín Gallego
Percona
Agenda

- Percona Monitoring and Management (PMM) Architecture
- External Exporters
- Visualizing Data With Dashboards
- Custom Collectors
Percona Monitoring and Management (PMM) Architecture
PMM Architecture

- Client / Server kind of architecture
  - Exporters get data and send it to the server
- Prometheus changed for Victoria Metrics
  - Support for pull and push
- Main server components:
  - Grafana
  - Victoria Metrics / Clickhouse / PostgreSQL
PMM Native Support

- PMM has out-of-the-box support for:
  - Linux
  - MySQL / Percona Server for MySQL
  - Percona XtraDB Cluster
  - ProxySQL
  - MariaDB
  - MongoDB / Percona Server for MongoDB
  - PostgreSQL
PMM Online Demo

- [https://pmmdemo.percona.com/](https://pmmdemo.percona.com/)
External Exporters
Getting Data From External Exporters

- There are two ways to get metrics from other exporters:
  - external
  - external-serverless
Getting Data From External Exporters

● External
  ○ The PMM agent is running in the node that has the DB and exporter running
  ○ In this case, OS metrics will be collected from that node, too
  ○ `shell> pmm-admin add external --help`

● External Serverless
  ○ The PMM agent is running in another node, different to the DB and exporter
  ○ No OS metrics available (unless there is an exporter running)
  ○ `shell> pmm-admin add external-serverless --help`
External Exporter Setup

- **Install PMM client**
  
  ```shell
  shell> yum install \\n  https://repo.percona.com/yum/percona-release-latest.noarch.rpm
  shell> yum install pmm2-client
  ```

- **Configure PMM client**
  
  ```shell
  shell> pmm-admin config --server-insecure-tls \\
  --server-url=https://admin:admin@<PMM_server_IP_address>:443
  ```

- **Add the external exporter**
  
  ```shell
  shell> pmm-admin add external --listen-port=7070 \\
  --service-name="cassandra_node_dev-01" \\
  --group="Cassandra_Cluster"
  ```
  
  External Service added.
  
  Service ID   : /service_id/ad6c2661-8336-468a-aaac-07ed1466a7c1
  Service name: cassandra_node_dev-01
  Group        : Cassandra_Cluster
External Exporter Check

- Check using pmm-admin tool
  
  shell> pmm-admin list

  Service type                Service name          Service ID
  External:Cassandra_Cluster  cassandra_node_dev-01 /service_id/ad6c2661-...7c1

  Agent type    Status     Metrics Mode Agent ID                   Service ID
  pmm_agent     Connected               /agent_id/eea5b83c-...2fa
  node_exporter Running    push         /agent_id/9378096c-...257
  ...


External Exporter Check

- Check using the Advanced Data Exploration dashboard
External Serverless Exporter Setup

- Install and configure a PMM client in a node that can reach the desired instance
- Add the external serverless exporter

```
shell> pmm-admin add external-serverless \
   --address=192.168.224.27:7070 \ 
   --external-name="cassandra-dev-03" \ 
   --group="Cassandra_Cluster"
External Service added.
Service ID : /service_id/14cf3121-9265-40d4-94da-06e19334b131
Service name: cassandra-dev-03
Group : Cassandra_Cluster
```
External Serverless Exporter Check

- Check using pmm-admin tool from any node
- Check the "Service ID" column and map with returned ID

```shell
shell> pmm-admin list
...
Agent type        Status  Metrics Mode   Agent ID  Service ID
external-exporter Unknown pull   /agent_id/b78...   /service_id/14cf3121-...131
...
```
External Serverless Exporter Check

- Check using the Advanced Data Exploration dashboard
Services, Agents and Nodes

- Check using the PMM Inventory dashboard
Where to Get Exporters From?

- Prometheus exporters page:
  - https://prometheus.io/docs/instrumenting/exporters/

- GitHub is also a great resource:
Visualizing Data With Dashboards
Dashboards

- Dashboards logically group series of graphs together
Other Dashboards

- It's easy to add new dashboards to PMM
- Compatible dashboards which are not included by default
  - [https://github.com/Percona-Lab/pmm-dashboard](https://github.com/Percona-Lab/pmm-dashboard)
- Grafana dashboard database
  - [https://grafana.com/grafana/dashboards](https://grafana.com/grafana/dashboards)
Adding a Cassandra Dashboard

Filter by:
Name / Description
Search: cassandra

Data Source
Prometheus

Panel Type
All

Category
All

Collector
All

Sort By
Name

Share your dashboards
Sign up for a free Grafana Cloud Account and share your creations with the community.

Cassandra by Oleg Glushak
Monitoring Cassandra with Prometheus using JXM exporter
Downloads: 3893
Reviews: 1

Cassandra by delmosfr
Downloads: 2306
Reviews: 0

Cassandra by delmosfr
Downloads: 1474
Reviews: 0

Cassandra by opstreedevelops
Monitoring Cassandra with Prometheus and telegraf agent using Jolokia agent.
Downloads: 128
Reviews: 0

Cassandra Dashboard by kwseo
Cassandra Dashboard with Prometheus and jmx_exporter
Downloads: 201
Reviews: 0

Cassandra detail by kalia
Download cassandra node detail in OSS Cassandra and JMX probe
Downloads: 3581
Adding a Cassandra Dashboard

Cassandra by Oleg Glushak

Monitoring Cassandra with Prometheus using JX export
Last updated: 3 years ago

Cassandra dashboard for Prometheus JX export
https://github.com/prometheus/jmx_exporter JX export configuration:

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true
whitelistObjectNames: []
  "org.apache.cassandra.metrics:type=ColumnFamily,\name=RangeLatency\x2c:\x2c",
  "org.apache.cassandra.metrics:type=ColumnFamily,\name=LiveSSTableCount\x2c:\x2c",
  "org.apache.cassandra.metrics:type=ColumnFamily,\name=SSTablesPerReadHistogram\x2c:\x2c",
  "org.apache.cassandra.metrics:type=ColumnFamily,\name=SpeculativeRetries\x2c:\x2c",
  "org.apache.cassandra.metrics:type=ColumnFamily,\name=MemtableMergeTime\x2c:\x2c",
  "org.apache.cassandra.metrics:type=ColumnFamily,\name=AsyncMemtableFlushTime\x2c:\x2c",
```

Get this dashboard:

Download JSON
How do I import this dashboard?

Dependencies:
- GRAFANA 5.0.1
- GRAPH 5.0.0
- PROMETHEUS 5.0.0
- SIMPLESTAT 5.0.0
Adding a Cassandra Dashboard
Adding a Cassandra Dashboard
Custom Collectors
Custom Data Collectors

- PMM offers two ways in which to get additional information
  - Queries collectors
  - Textfile collectors
- Support for high, medium and low resolution scraping times
Custom Data Collectors

- Custom queries collector
  - For MySQL and PostgreSQL
  - Connects to an already-monitored DB instance, and runs the queries to expose metrics

- Custom textfile collector
  - Reads from files, and expects them to conform with Prometheus way of exposing metrics
  - One possible way is to use a BASH script and cron
  - But of course it is not limited to that, you can use whatever tools allow you to write to a file
Custom Data Collectors

- Custom queries collector

```
[root@node1 ~]# cat /usr/local/percona/pmm2/collectors/custom-queries/mysql/high-resolution/queries-mysql.yml

## Custom query example.
## The namespace (prefix of the metric name) for the custom query. See https://prometheus.io/docs/practices/naming/#metric-names for details.

#mysql_performance_schema:
  # query: "SELECT event_name, current_count, high_count FROM sys.memory_global_by_current_bytes WHERE current_count > 0;"
  # metrics: ## List of metrics.
  #   - event_name: ## The alias mapped to a value returned by a query to the Prometheus label https://prometheus.io/docs/practices/naming/#labels
  #     usage: "LABEL" ## If usage is LABEL this value will be used as Prometheus dimension.
  #     description: "Performance Schema Event Name"
  #   - current_count: ## The name of the metric. See https://prometheus.io/docs/practices/naming/#metric-names
  #     usage: "GAUGE" ## The type of the metric. It should be be the one from the following: COUNTER, GAUGE, MAPPEDMETRIC, DURATION, DISCARD.
  #     description: "Memory currently allocated to the Event"
  #   - high_count:
  #     usage: "GAUGE"
  #     description: "High Water Mark of Memory allocated to the Event" ## Description of the metric.
```
Custom Data Collectors

- Custom textfile collector

```bash
[root@node1 ~]# cat /usr/local/percona/pmm2/collectors/textfile-collector/high-resolution/example.prom | head -n11
## Below is an example of a full-fledged Prometheus metric exposition,
## including comments, HELP and TYPE expressions, a histogram, a summary,
## character escaping examples, and more. It is taken from:
## https://prometheus.io/docs/instrumenting/exposition_formats/#text-format-example
## More details about textfile collector you can find at:
## https://github.com/prometheus/node_exporter#textfile-collector
## 
## HELP http_requests_total The total number of HTTP requests.
## TYPE http_requests_total counter
## http_requests_total{method="post",code="200"} 1027 1395066363000
## http_requests_total{method="post",code="400"} 3 1395066363000
```
Example - MySQL Custom Query Collector

- Blogpost explaining it:

- External collector:

- Grafana dashboard:
  - https://grafana.com/grafana/dashboards/13266
Example - MySQL Custom Query Collector
Thank You!