

What do we want to monitor?

All the databases!

Agustín Gallego
Percona

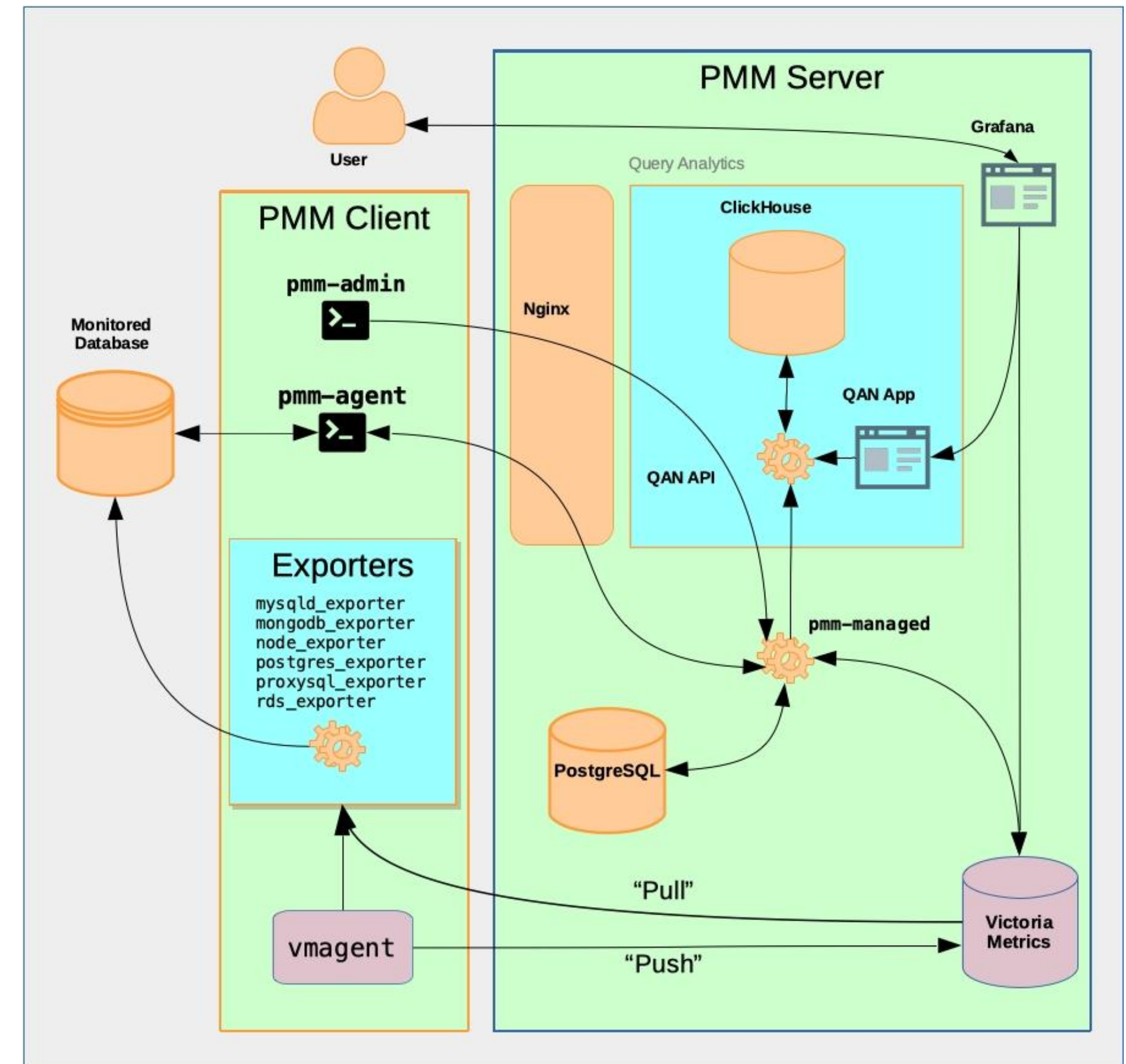
Agenda

- 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

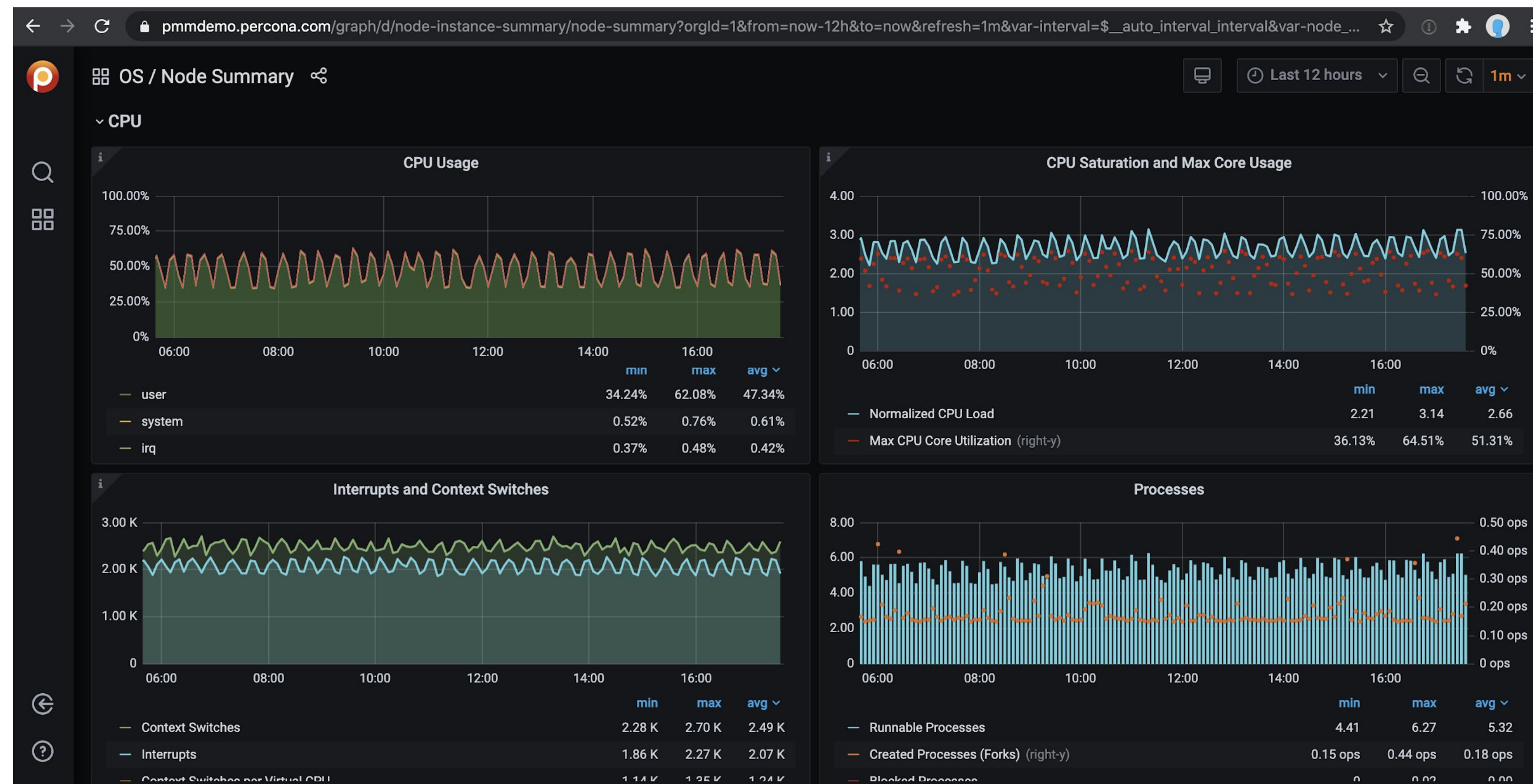


PMM Native Support

- 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/>

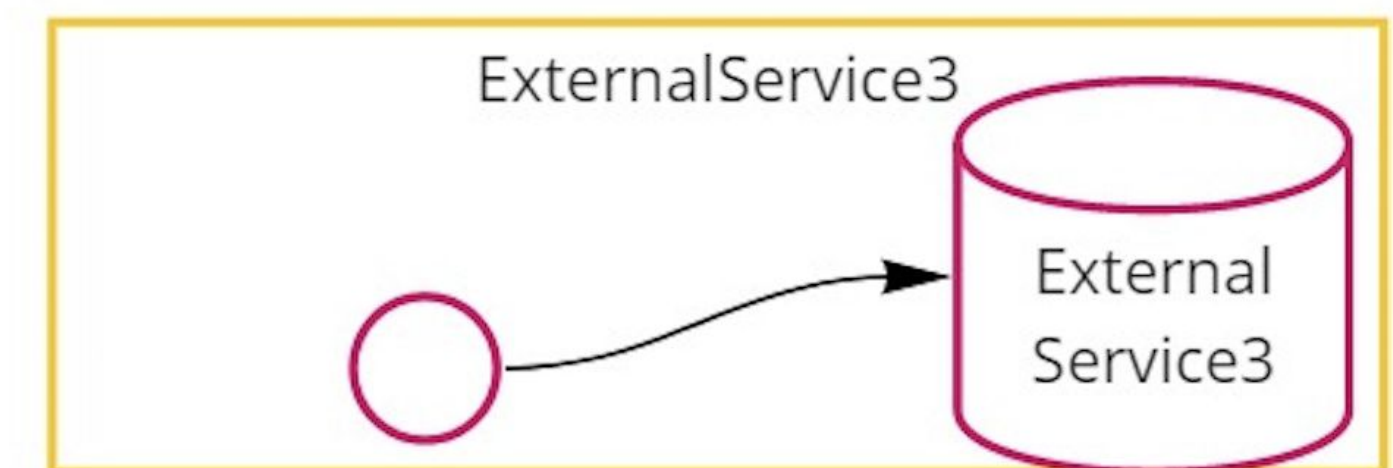
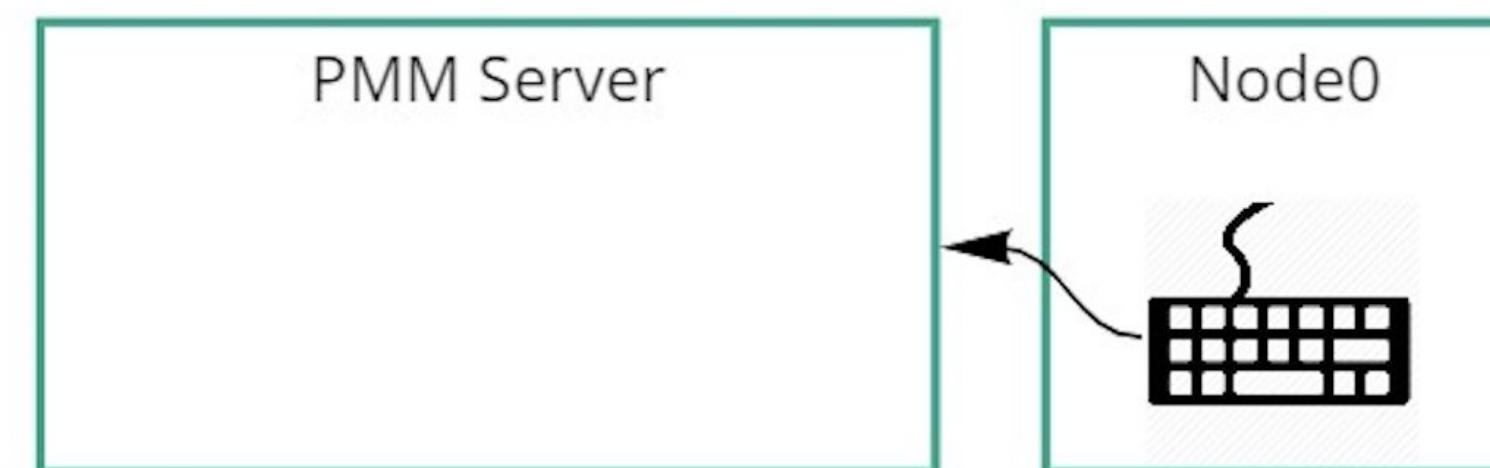
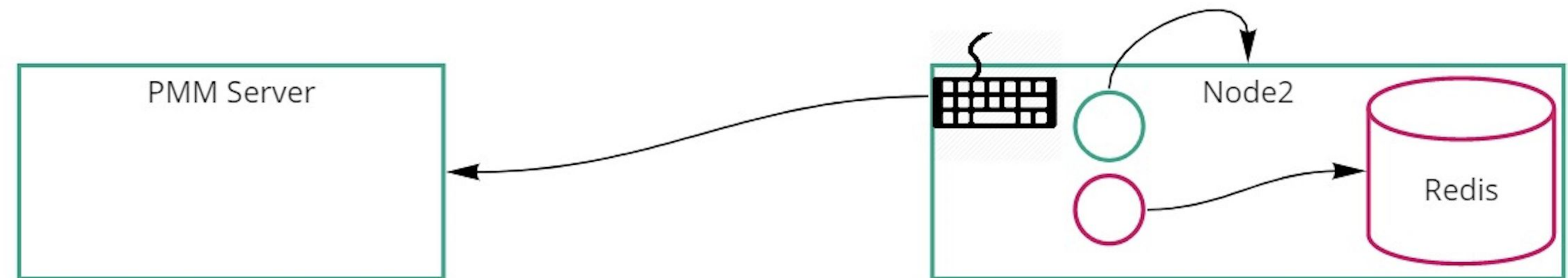


External Exporters

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

- Configure PMM client

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

- Add the external exporter

```
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

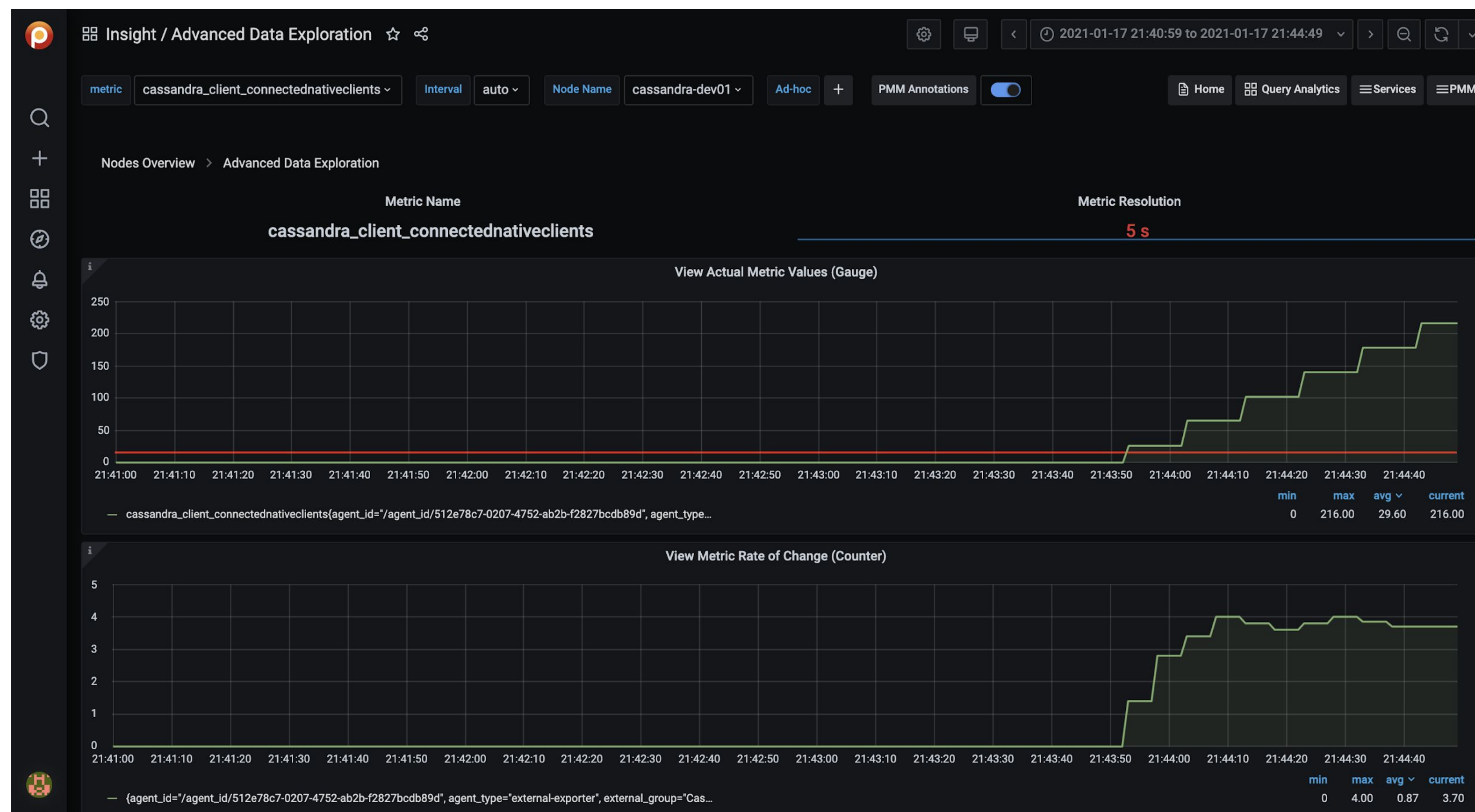
```
shell> pmm-admin list
```

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> pmm-admin list
```

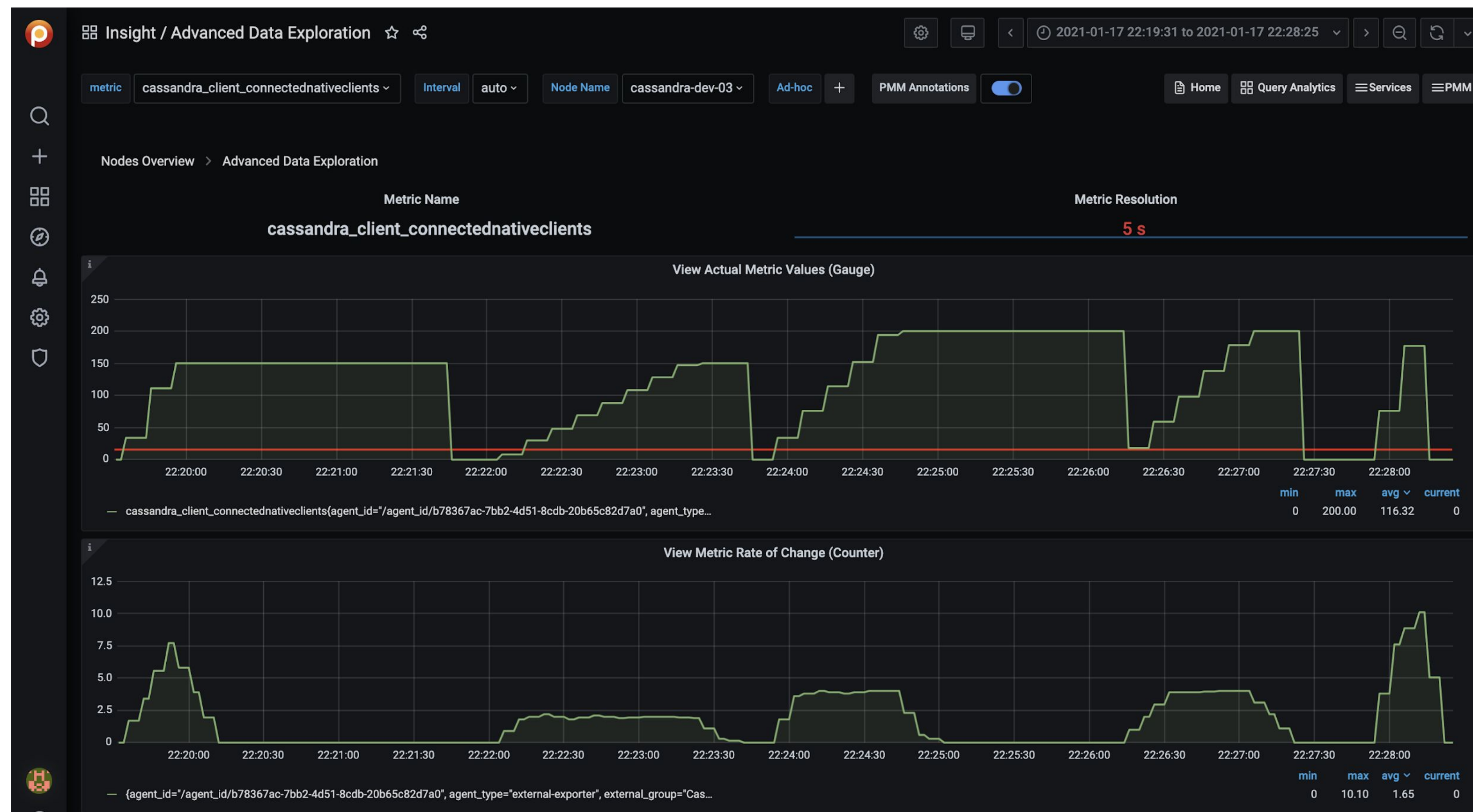
```
...
```

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



PMM / PMM Inventory



Home

Query Analytics

Services

PMM

Nodes Overview

Advanced Data Exploration

PMM Inventory

Services

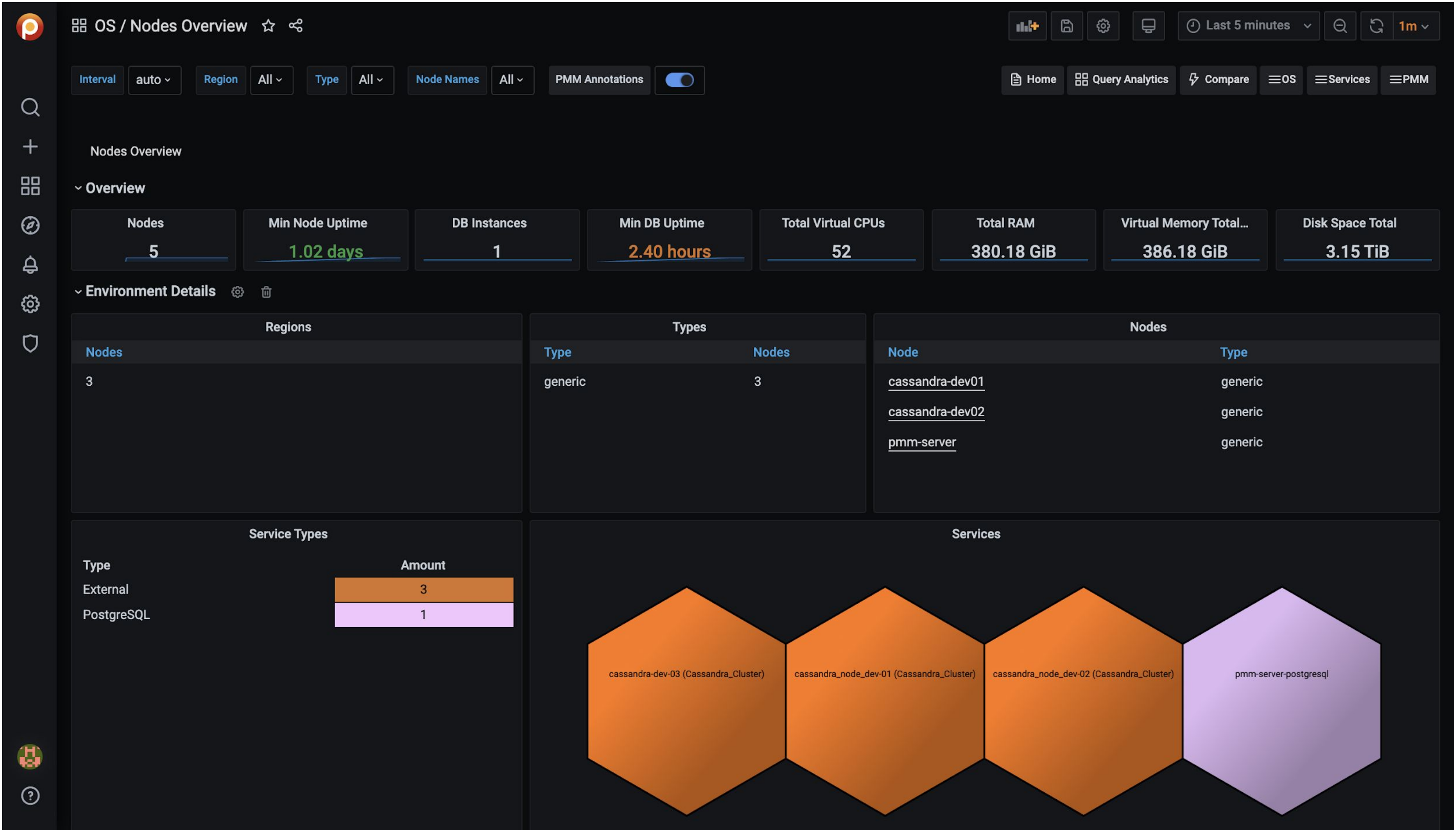
Agents

Nodes

 Delete

☐	ID	Service Type	Service name	Node ID	Addresses	Port	Other Details
☐	/service_id/14cf3121-9265-40d4-94da-06e19334b131	External	cassandra-dev-03	/node_id/4c501db5-5c68-423c-86b4-56d0e03adf3e			group: Cassandra_Cluster
☐	/service_id/ad6c2661-8336-468a-aaac-07ed1466a7c1	External	cassandra_node_dev-01	/node_id/0a6fd97f-c3c9-4ed7-a84b-162540bd43ed			group: Cassandra_Cluster
☐	/service_id/6c4582ee-1fe0-490e-b4a3-b032ad2a125f	External	cassandra_node_dev-02	/node_id/c4897ccb-857b-49f5-b63d-d6f758cfb72d			group: Cassandra_Cluster
☐	/service_id/90c8d420-b3f3-413a-8682-9edfaf44d7f6	PostgreSQL	pmm-server-postgresql	pmm-server	127.0.0.1	5432	

OS / Nodes Overview



Where to Get Exporters From?

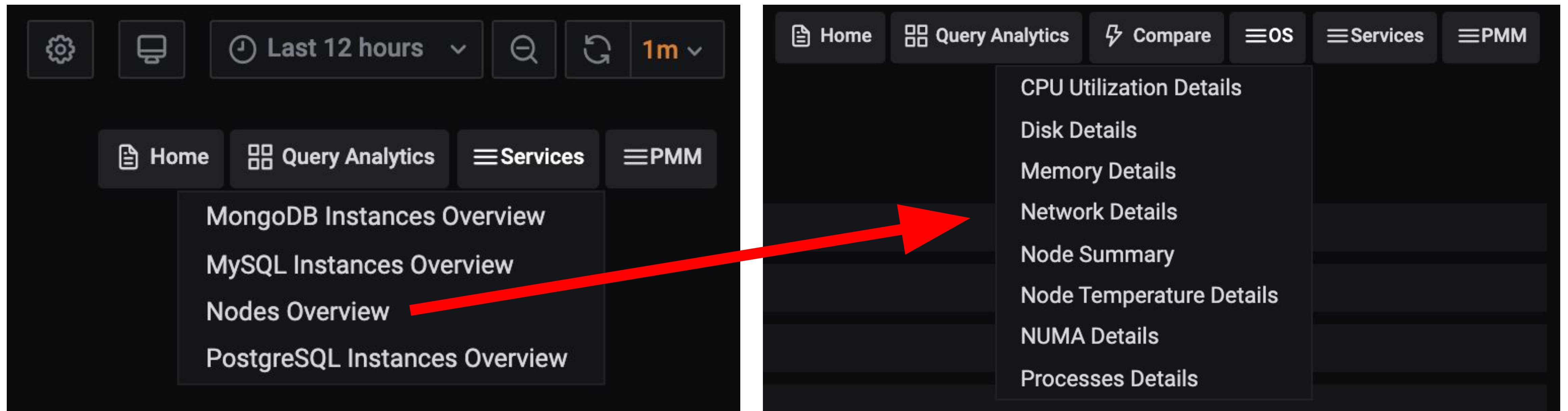
- Prometheus exporters page:
 - <https://prometheus.io/docs/instrumenting/exporters/>
- GitHub is also a great resource:



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-dashboards>
- Grafana dashboard database
 - <https://grafana.com/grafana/dashboards>

Adding a Cassandra Dashboard

Filter by:

Name / Description

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 PROMETHEUS OTHER	Downloads: 3893 Reviews: 1 ★★★★☆
	Cassandra by deimosfr PROMETHEUS OTHER	Downloads: 2306 Reviews: 0
	Cassandra by deimosfr PROMETHEUS OTHER	Downloads: 1474 Reviews: 0
	Cassandra by opstredevops Monitoring Cassandra with Prometheus and telegraf agent using Jolokia agent. PROMETHEUS TELEGRAF: CASSANDRA	Downloads: 128 Reviews: 0
	Cassandra Dashboard by kwseo Cassandra Dashboar with Prometheus and jmx_exporter PROMETHEUS OTHER	Downloads: 201 Reviews: 0
	Cassandra detail by kaija Display cassandra node detail in OS, Cassandra and JVM blocks	Downloads: 3581

Adding a Cassandra Dashboard



Cassandra by [Oleg Glushak](#)

DASHBOARD

Monitoring Cassandra with Prometheus using JXM exporter

Last updated: 3 years ago

Downloads: 3893

Reviews: 1



Add your review!

Overview

Revisions

Reviews



Cassandra dashboard for Prometheus JXM exporter

https://github.com/prometheus/jmx_exporter JXM exporter configuration:

```
lowercaseOutputName: true
lowercaseOutputLabelNames: true
whitelistObjectNames: [
  "org.apache.cassandra.metrics:type=ColumnFamily,name=RangeLatency,*",
  "org.apache.cassandra.metrics:type=ColumnFamily,name=LiveSSTableCount,*",
  "org.apache.cassandra.metrics:type=ColumnFamily,name=SSTablesPerReadHistogram,*",
  "org.apache.cassandra.metrics:type=ColumnFamily,name=SpeculativeRetries,*",
  "org.apache.cassandra.metrics:type=ColumnFamily,name=MemtableOnHeapSize,*",
  "org.apache.cassandra.metrics:type=ColumnFamily,name=MemtableSwitchCount,*"
```

Get this dashboard:

5408

Copy ID to Clipboard

[Download JSON](#)

[How do I import this dashboard?](#)

Dependencies:

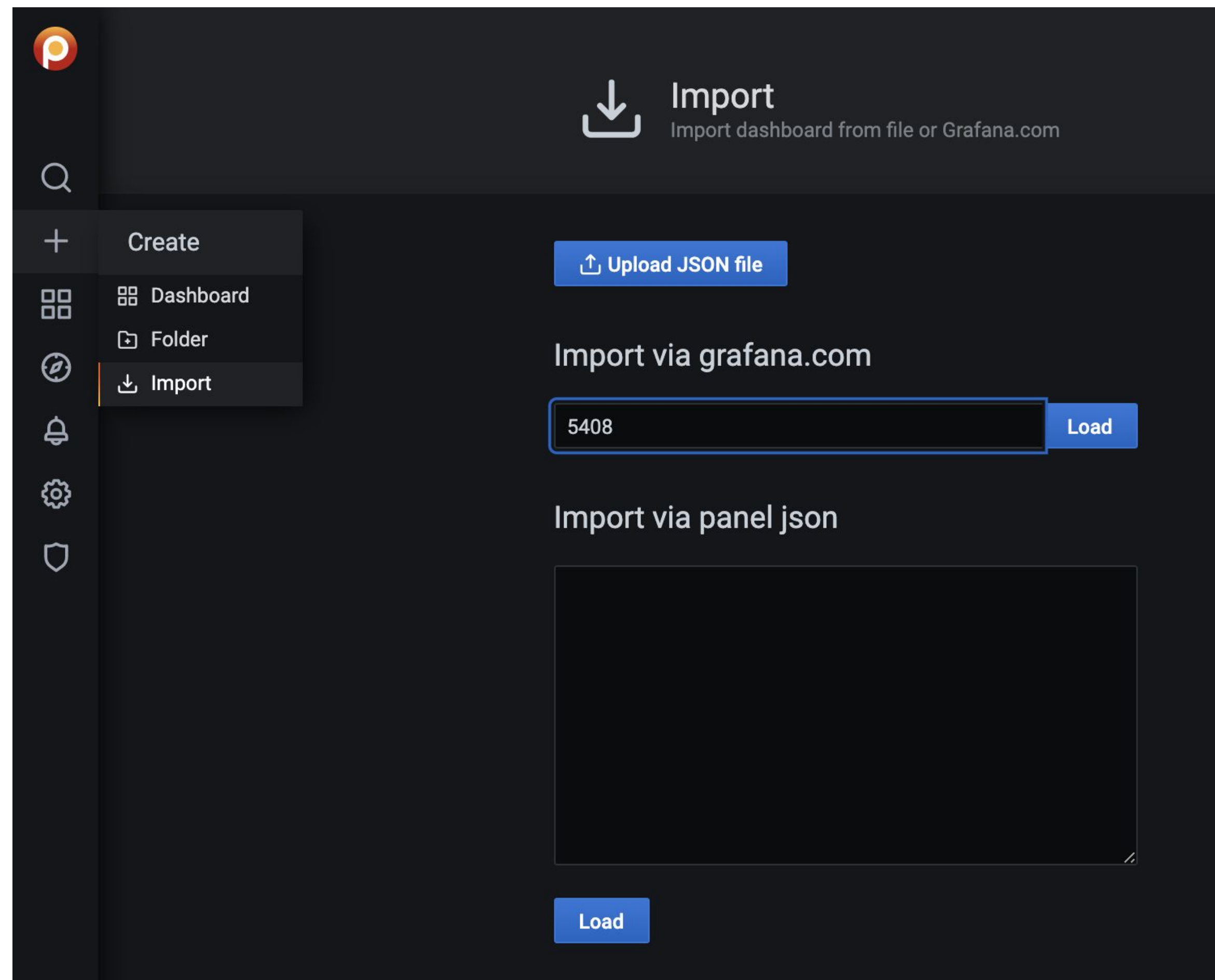
[GRAFANA 5.0.1](#)

[GRAPH 5.0.0](#)

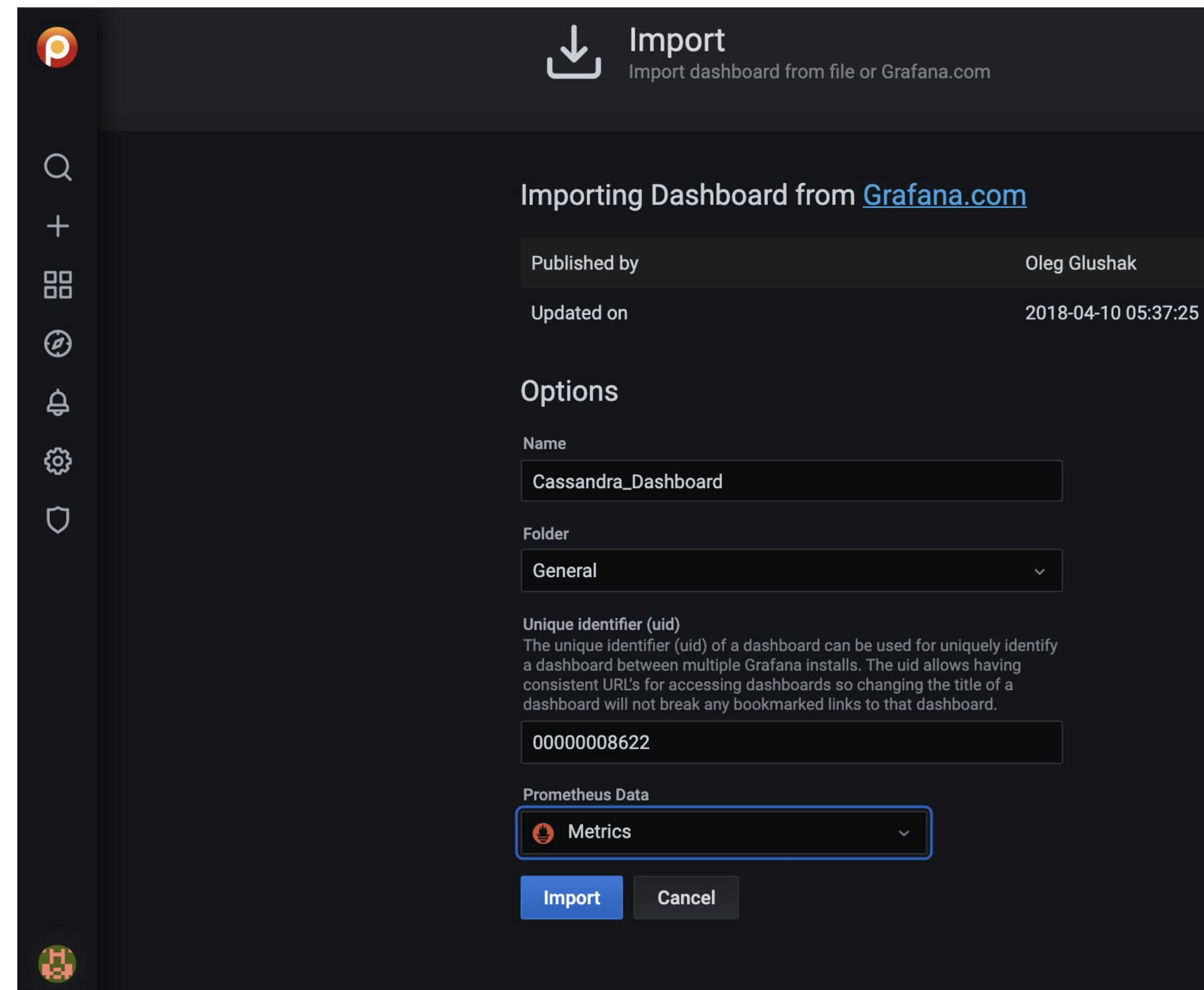
[PROMETHEUS 5.0.0](#)

[SINGLESTAT 5.0.0](#)

Adding a Cassandra Dashboard



Adding a Cassandra Dashboard



The screenshot shows the Grafana 'Import' interface. At the top, there's a dark header with the Grafana logo on the left and a download icon with the text 'Import' and 'Import dashboard from file or Grafana.com' on the right. Below the header, the main content area is titled 'Importing Dashboard from [Grafana.com](#)'. It displays metadata for a dashboard: 'Published by' Oleg Glushak and 'Updated on' 2018-04-10 05:37:25. Under the 'Options' section, there are three fields: 'Name' with the value 'Cassandra_Dashboard', 'Folder' with a dropdown menu showing 'General', and 'Unique identifier (uid)' with the value '00000008622'. A descriptive text for the uid explains its purpose. Below these is the 'Prometheus Data' section with a dropdown menu showing 'Metrics'. At the bottom, there are two buttons: 'Import' (highlighted in blue) and 'Cancel'.

Import
Import dashboard from file or Grafana.com

Importing Dashboard from [Grafana.com](#)

Published by	Oleg Glushak
Updated on	2018-04-10 05:37:25

Options

Name

Cassandra_Dashboard

Folder

General

Unique identifier (uid)

The unique identifier (uid) of a dashboard can be used for uniquely identify a dashboard between multiple Grafana installs. The uid allows having consistent URL's for accessing dashboards so changing the title of a dashboard will not break any bookmarked links to that dashboard.

00000008622

Prometheus Data

Metrics

Import Cancel

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

Metrics resolution, sec ⓘ

Rare	Standard	Frequent	Custom
------	----------	----------	--------

Low	60
Medium	10
High	5

Apply changes

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-mysqld.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

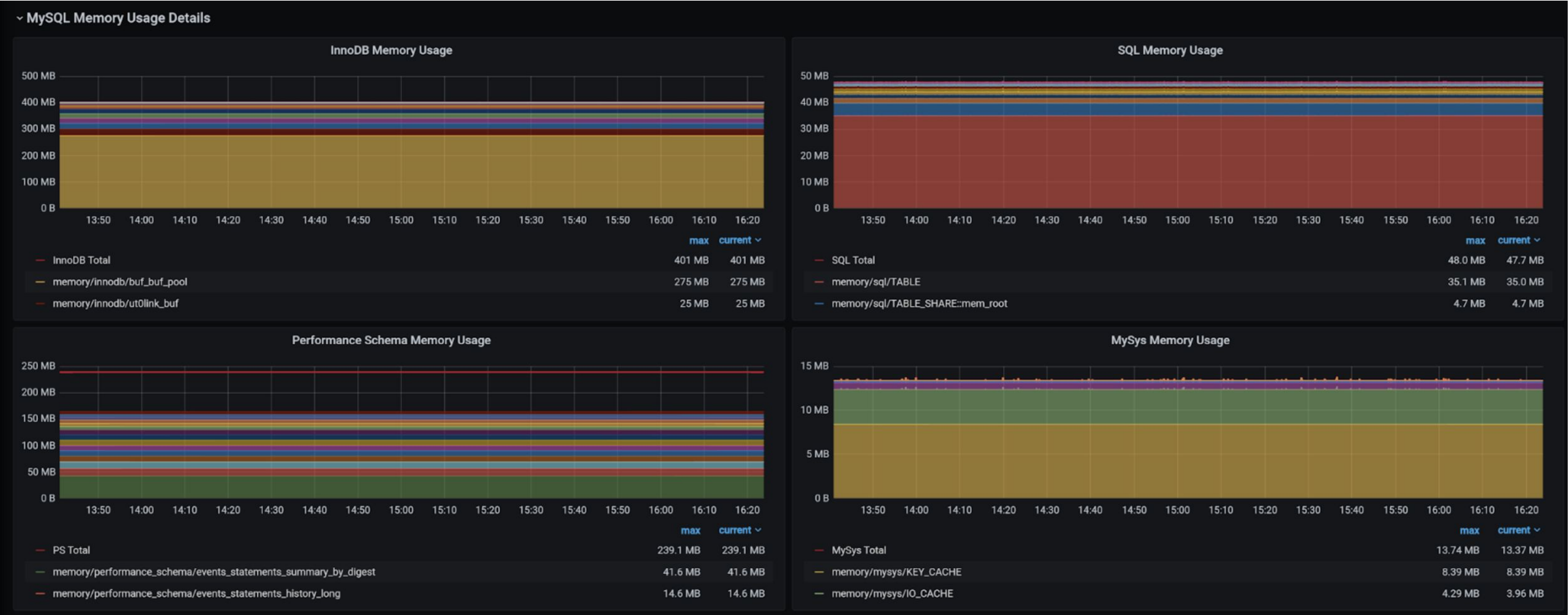
- Custom textfile collector

```
[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:
 - <https://www.percona.com/blog/2020/11/02/understanding-mysql-memory-usage-with-performance-schema/>
- External collector:
 - <https://github.com/Percona-Lab/pmm-custom-queries/blob/master/mysql/ps-memory-summary.yml>
- Grafana dashboard:
 - <https://grafana.com/grafana/dashboards/13266>

Example - MySQL Custom Query Collector



Thank You!